Solving Two Stage Stochastic Programming Problems Using ADMM

Nouralden Mohammed (nouralden@aims.ac.za)
University of the Witwatersrand https://orcid.org/0000-0003-2010-0052
Montaz Ali
University of the Witwatersrand

Research Article

Keywords: Stochastic Programming, Progressive Hedging, Alternating Direction Method of Multipliers, Convergence

Posted Date: November 24th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-657331/v1

Mathematical Programming manuscript No.
(will be inserted by the editor)

Solving Two Stage Stochastic Programming Problems Using ADMM

Nouralden Mohammed · **M Montaz Ali**

Received: date / Accepted: date

Abstract In this paper, we have dealt with the solution of a two-stage stochastic programming problem using ADMM. We have formulated the problem into a deterministic three-block separable optimization problem, and then we applied ADMM to solve it. We have established the theoretical convergence of ADMM to the optimal solution based on the concept of lower semicontinuity and the Kurdyka-Lojasiewicz property. We have compared ADMM with Progressive Hedging in terms of performance criteria using five benchmark problems from the literature. The comparison shows that ADMM outperforms Progressive Hedging.

Keywords Stochastic Programming · Progressive Hedging · Alternating Direction Method of Multipliers · Convergence.

Mathematics Subject Classification (2020) 90C15 · 90C26 · 90C90

1 Introduction

In recent years, researchers have shown an increased interest in the Alternating Direction Method of Multipliers (ADMM) method [13] and its applications. The importance of ADMM arises from its ability to decompose the optimization problem into separable components, which can be solved in alternating manners. Various applied problems in machine learning form an application area of ADMM. The emergence

Nouralden Mohammed, Corresponding author
School of Computer Science and Applied Mathematics
University of The Witwatersrand
Johannesburg, South Africa
2356898@students.wits.ac.za

M Montaz Ali
School of Computer Science and Applied Mathematics
University of The Witwatersrand
Johannesburg, South Africa
montaz.ali@wits.ac.za

of machine learning and big data applications necessitates distributed computation and storage due to large problem sizes. Boyd et al. [5] have applied ADMM on various distributed large-scale convex optimization problems, including the least absolute shrinkage and selection operator (LASSO) [22], support vector machine (SVM), and sparse logistic regression. Wang et al. [21] have also used ADMM to replace stochastic gradient descent (SGD) optimizer in deep learning. Their new algorithm based on ADMM converges to the optimal solution.

The convergence properties and application of ADMM have been reported in the literature [5,8,9]. There is a solid relationship between Douglas-Rachford splitting [23] and ADMM. A careful formulation shows that the two methods are equivalent in their dual form. Arpón et al. [2] successfully implemented ADMM to the two-stage stochastic program (SP). They have not provided the theoretical convergence of ADMM but referred to the work of Sun et. al [19] for the convergence proof. We have taken a slightly different approach to establish the theoretical convergence of ADMM. Since the class of SPs we have considered are non-convex and non-smooth, our convergence proof is based on the concept of semicontinuity and the Kurdyka-Lojasiewicz property of the objective function [20]. Sun et al. [19] have introduced an intermediate step in the Gauss-Seidel cycle for updating the block coordinate descent for ADMM, which requires more computational time. They have argued that the directly extended ADMM for 3-blocks, a version of the classical ADMM developed by adding a third variable to the objective function and the constraints, is non-convergent. We have used the usual Gauss-Seidel approach where each variable is updated once; hence no extra computation step is needed. Also, we have proved that the directly extended ADMM is convergent under some suitable assumptions. Then we have applied ADMM to the two-stage SPs with a finite number of scenarios. The formulation is separable and dimension intensive, but the scenario-dependent variables and related dual residuals of ADMM are executable in parallel.

We have studied the convergence analysis of the three-block formulation of ADMM, and implemented it for solving the two-stage SP without any external solver. We have tested our algorithm on a number of problems from the literature [1,12]. Moreover, we have numerically compared our algorithm with state-of-art Progressive Hedging (PH) [17,4].

The rest of the paper is organized as follows. Section 2 introduces the three-block minimization problem, and in Section 3 we give a brief introduction to the two-stage SP. We present the theoretical convergence of the 3-block ADMM in Section 4. The implementation of ADMM to the two-stage SP is presented in Section 5. The numerical experiments are given in Section 6, followed by the concluding remarks in Section 7.

2 Three-block Minimization Problem

The classic 2-block ADMM, its numerical implementation and convergence properties have been well documented in the literature [5,8,9]. Sun et al. [19] have applied 3-block semi-proximal ADMM to a class of convex conic programming with several types of constraints. In this paper, we study the 3-block problem for a class of non-

convex programming problems using the directly extended ADMM and provide its convergence under some suitable assumptions 4.

Consider the three-block minimization problem in the form

$$\begin{aligned} \min_{x_1,x_2,x_3} \quad & f(x_1)+g(x_2)+h(x_3), \quad \text{subject to} \\ & A_1x_1+A_2x_2+A_3x_3=t, \end{aligned} \tag{1}$$

where $f:\mathbb{R}^{n_1}\to\mathbb{R}, g:\mathbb{R}^{n_2}\to\mathbb{R}$ are proper lower semi-continuous functions, $h:\mathbb{R}^{n_3}\to\mathbb{R}$ is a linear function; $t\in\mathbb{R}^m$, and A_1, A_2 and A_3 are in appropriate dimensions [20]. The augmented Lagrangian for Problem (1) is defined as

$$\begin{aligned} L_\rho(x_1,x_2,x_3,\mu) := & f(x_1)+g(x_2)+h(x_3)+\mu^T(A_1x_1+A_2x_2 \\ & +A_3x_3-t)+\frac{\rho}{2}\|A_1x_1+A_2x_2+A_3x_3-t\|^2. \end{aligned} \tag{2}$$

In the solution procedure of Problem (1) we include regularized terms. The regularized terms enhance the convergence by reducing the oscillation between progressive iterates [2]. We have used the following iterative procedure with the regularized term:

$$\begin{aligned} x_1^{k+1} &= \arg\min_{x_1}\left(L_\rho(x_1,x_2^k,x_3^k,\mu^k)+\frac{\gamma}{2}\left\|x_1-x_1^k\right\|^2\right), \\ x_2^{k+1} &= \arg\min_{x_2}\left(L_\rho(x_1^{k+1},x_2,x_3^k,\mu^k)+\frac{\gamma}{2}\left\|x_2-x_2^k\right\|^2\right), \\ x_3^{k+1} &= \arg\min_{x_3}\left(L_\rho(x_1^{k+1},x_2^{k+1},x_3,\mu^k)+\frac{\gamma}{2}\left\|x_3-x_3^k\right\|^2\right), \\ \mu^{k+1} &= \mu^k+\rho(A_1x_1^{k+1}+A_2x_2^{k+1}+A_3x_3^{k+1}-t), \end{aligned} \tag{3}$$

where γ is a penalty parameter linked with the regularized terms. The augmented regularized term added to each subproblem in (3) eliminates the need for a convex objective function [20].

Boyd et al. [5] state the necessary and sufficient optimality conditions for Problem (1) as follows

$$A_1x_1^*+A_2x_2^*+A_3x_3^*-t=0, \tag{4}$$

$$0\in\partial f(x_1^*)+A_1^T\mu^*, \tag{5}$$

$$0\in\partial g(x_2^*)+A_2^T\mu^*, \tag{6}$$

$$0\in\partial h(x_3^*)+A_3^T\mu^*, \tag{7}$$

where Equation (4) is the primal feasibility, and Equations (5)-(7) are dual feasibilities. Here, ∂f denotes the subdifferential set of f.

Provided that x_3^{k+1} minimizes $L_\rho(x_1^{k+1},x_2^{k+1},x_3,\mu^k)$, we have

$$\begin{aligned} 0 &\in \partial h(x_3^{k+1})+A_3^T\mu^k+\rho A_3^T(A_1x_1^{k+1}+A_2x_2^{k+1}+A_3x_3^{k+1}-t) \\ &= \partial h(x_3^{k+1})+A_3^T\mu^{k+1}, \qquad \text{by (3)}, \end{aligned}$$

which means that x_3^{k+1} and μ^{k+1} satisfy the duality condition in Equation (7).

In the case of x_2^{k+1}, we have

$$\begin{aligned} 0 &\in \partial g(x_2^{k+1}) + A_2^T \mu^k + \rho A_2^T (A_1 x_1^{k+1} + A_2 x_2^{k+1} + A_3 x_3^k - t) \\ &= \partial g(x_2^{k+1}) + A_2^T \mu^{k+1} + s_1^{k+1}, \end{aligned}$$

where $s_1^{k+1} = \rho A_2^T A_3 (x_3^k - x_3^{k+1})$ is the residual for the dual feasibility.

By repeating the same steps, we find the residual associated with x_1^{k+1} as

$$s_2^{k+1} = \rho A_1^T \left[A_2 (x_2^k - x_2^{k+1}) + A_3 (x_3^k - x_3^{k+1}) \right].$$

At iteration $k+1$, we denote $r^{k+1} = A_1 x_1^{k+1} + A_2 x_2^{k+1} + A_3 x_3^{k+1} - t$ as *primal residual* and s_1^{k+1}, s_2^{k+1} as *dual residuals*. We will see in Section 4 that all the residuals converge to zero as $k \to \infty$.

3 Two-stage SP

The two-stage SP is defined as [12]

$$\begin{aligned} \min_x \quad & c^T x + \mathbf{E}_\xi \mathscr{Q}(x, \xi), \quad \text{subject to} \\ & Ax = b, \quad x \geq 0, \quad x \in \mathbb{R}^{n_1}, \end{aligned} \tag{8}$$

where $c \in \mathbb{R}^{n_1}$, $A \in \mathbb{R}^{m_1 \times n_1}$, $b \in \mathbb{R}^{m_1}$, are constants and ξ is a random variable of a known distribution. The term, $\mathscr{Q}(x, \xi)$, outlines the second-stage optimum value and is defined as [4]

$$\begin{aligned} \mathscr{Q}(x, \xi) = \min_y \quad & q(\xi)^T y, \quad \text{subject to} \\ & T(\xi)x + W(\xi)y = h(\xi), \\ & y \geq 0, \quad y \in \mathbb{R}^{n_2}, \end{aligned} \tag{9}$$

where $q(\xi) \in \mathbb{R}^{n_2}$, $T(\xi) \in \mathbb{R}^{m_2 \times n_1}$, $W(\xi) \in \mathbb{R}^{m_2 \times n_2}$, $h(\xi) \in \mathbb{R}^{m_2}$ encode the random variable data.

The value of the first-stage variable, x, needs to be determined before any future revealing of the random variable ξ; y is known as the second-stage decision variable, which corresponds to the decisions after the realizations of the random variable revealed; y is also known as recourse variable since it compensates any bad decisions that may occur at the first-stage.

When the random variable ξ has finitely many realizations, or scenarios $S = \{\xi_1, \xi_2 \ldots, \xi_s\}$, with corresponding probabilities $\{p_1, p_2, \ldots, p_s\}$, Problem (8) can be reformulated as a large linear programming (LP) problem [10] in the form

$$\begin{aligned} \min_{x,y} \quad & c^T x + \sum_{i=1}^{s} p_i q_i^T y_i, \quad \text{subject to} \\ & Ax = b, \\ & T_i x + W_i y_i = h_i, \quad i = 1, 2, \ldots, s, \\ & x, y_i \geq 0, \qquad i = 1, 2, \ldots, s, \end{aligned} \tag{10}$$

where $\xi = \xi_i$ is i-th the scenario.

Cutting-plane methods, e.g., the L-shaped method, are used to solve (10) when the recourse matrix W is not dependent on the random variable ξ. When the recourse matrix depends on the random variable, a different method is needed, e.g., Progressive Hedging. We have taken a different approach that can deal with the recourse matrix of either type. Problem (10) can be equivalently restated as

$$\min \quad c^T x + I(\hat{x}) + \sum_{i=1}^{s} \left(p_i q_i^T y_i + I(\hat{y}_i) \right), \quad \text{subject to} \tag{11}$$

$$Ax = b, \tag{12}$$

$$x - \hat{x} = 0, \tag{13}$$

$$y_i - \hat{y}_i = 0, \qquad i = 1,2,\ldots,s \tag{14}$$

$$T_i x + W_i y_i = h_i, \quad i = 1,2,\ldots,s, \tag{15}$$

where

$$I(z) = \begin{cases} 0, & z \geq 0 \\ +\infty, & \text{otherwise}, \end{cases}$$

is a convex function since it is defined on a convex set [15].

The two indicator functions in (11) for the pairs $(\hat{x}, \hat{y}_i)$ compensate for the non-negativity condition of the variables x and y_i in the constrains (13) and (14), respectively. The new formulation in Equations (11)-(15) is considered as a three-block ADMM model. The first-stage decision variable x is presented the first block, while the second stage decision variable y_i in the second block, and the third one for the pair $(\hat{x}, \hat{y}_i), i = 1,2,\cdots,s$.

The section below states the convergence analysis of ADMM applied to two-stage SP.

4 Convergence Analysis

We start by introducing some definitions and lemmas that will be needed here.

Definition 1 The function $f : \mathbb{R}^n \to \mathbb{R}$ has the Kurdyka-Lojasiewicz (K-L) [20] property at $\bar{x}$, if there exists positive constants $\eta > 0, \delta > 0$, and a concave function $\varphi : [0, \eta) \to \mathbb{R}^+$, such that $\forall x \in \mathcal{N}(\bar{x}, \delta) \cap \{x : f(\bar{x}) < f(x) < f(\bar{x}) + \eta\}$,

$$\varphi^{'}(f(x) - f(\bar{x}))\texttt{dist}(0, \partial f(x)) \geq 1,$$

where the function φ has the following characteristics: (i) φ is continuous on $[0, \eta)$; (ii) φ is smooth concave on $(0, \eta)$; (iii) $\varphi(0) = 0$, $\varphi^{'}(x) > 0, \forall x \in (0, \eta)$; $\texttt{dist}(x, D) ::= \inf\{\|x - d\| : d \in D, x \in \mathbb{R}^n\}$.

Since all the functions we are considering in the two-stage SP are real analytic (linear, and indicator functions), they satisfy K-L inequality. All the lemmas and theorems presented here applies to real analytic functions.

In our context, we define a function $V : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^{n_3} \times \mathbb{R}^m \times \mathbb{R}^{n_3} \to \mathbb{R}$ by

$$V(x_1, x_2, x_3, \mu, \bar{x}_3) ::= L_\rho(x_1, x_2, x_3, \mu) + \frac{\tau}{2} \|x_3 - \bar{x}_3\|^2, \tag{16}$$

where $\tau = \frac{4\gamma^2}{\rho\sigma}$, γ and σ are constants, and $\bar{x}_3$ is the immediate prior iteration of x_3.

Let V be a lower semi-continuous function with non-empty domain and $V(x) > -\infty$ for every $x \in \text{dom } V$, and a_1 and a_2 are fixed positive constants. We suppose that the iterates of ADMM in (3) have a Lyapunov function that verifies the following subgradient decent conditions of [3]:

(C1) $V(x^{k+1}) \leq V(x^k) - a_1 \left\|x^k - x^{k+1}\right\|^2$, $\forall k \in \mathbb{N}$; this means that $V(\cdot)$ is a Lyapunov function that decreases in each iteration.
(C2) $\texttt{dist}(0, \partial V(x^{k+1})) \leq a_2 \left\|x^k - x^{k+1}\right\|$, $\forall k \in \mathbb{N}$.
(C3) There exists a sub-sequence $\{x^{k_j}\}$ converge to x^* such that $V(x^{k_j}) \to V(x^*)$ as $j \to \infty$.

In the sequel, we assume the properties (A1)-(A4) hold, where the given matrices, functions, and parameters are subject to Equations (2)-(3):

(A1) V is a K-L function;
(A2) There is a constant $\sigma > 0$, such that $\sigma \|\mu\|^2 \leq \left\|A_3^T \mu\right\|^2, \forall \mu \in \mathbb{R}^m$;
(A3) The parameters are chosen such that $\rho > \frac{8\gamma}{\sigma}$;
(A4) All the matrices A_1, A_2, and A_3 are bounded under Frobenius norm.

We will demonstrate that the function V meets the conditions (C1)-(C3) once the sequence has been generated with the iterative procedure in Equation (3). The proofs of Lemmas 1 and 3 are based on our choice of V in (16).

The statements of lemmas 1 and 4 are given in [3] and [20], respectively, but we have taken a slight different approach in proving them for two-stage SP.

Lemma 1 *Let $\{x^k\}$ be a sequence that meets the conditions (C1)-(C3). If V is a K-L function, then the sequence $\{x^k\}$ converges to $\bar{x}$. Furthermore, the sequence $\{x^k\}$ has a finite length, i.e., $\sum_{k=1}^{\infty} \left\|x^{k+1} - x^k\right\| < \infty$ [Theorem 3.1, [3]]. It follow that $V(\bar{x}) = V(x^*)$, where x^* is the optimal point. Moreover, the augmented function L_ρ and Lyapunov function V has the same optimal value.*

Proof Condition (C1) states that the function V is a nonincreasing function, i.e., $V(x^{k+1}) \leq V(x^0)$. Also, condition (C1) and the fact $V(x^*) \leq V(x^{k+1})$ yield

$$\left\|x^{k+1} - x^k\right\| \leq \sqrt{\frac{V(x^k) - V(x^{k+1})}{a_1}} \leq \sqrt{\frac{V(x^k) - V(x^*)}{a_1}}, \quad \forall k \in \mathbb{N}. \tag{17}$$

Now, if $x^k \in \mathscr{N}(x^*, \delta)$ for $k \geq 1$, we claim that the following inequality holds

$$2\left\|x^{k+1} - x^k\right\| \leq \left\|x^k - x^{k-1}\right\| + \frac{a_2}{a_1}\left(\varphi(V(x^k) - V(x^*)) - \varphi(V(x^{k+1}) - V(x^*))\right), \tag{18}$$

where the constants a_1 and a_2 are given in the conditions (C1)-(C2). If $x^{k+1} = x^k$, the inequality is trivial. Let us assume that $x^{k+1} \neq x^k$, $V(x^*) < V(x^k)$. The quantity

$V(x^k)-V(x^*)>0$ and (C2) for x^k and x^{k-1}, $x^k \neq x^{k-1}$, combined with K-L inequality, $\varphi^{'}(V(x^k)-V(x^*))\texttt{dist}(0,\partial V(x^k) \geq 1$, $\texttt{dist}(0,\partial V(x^k)) \neq 0$, imply

$$\varphi^{'}(V(x^k)-V(x^*)) \geq \frac{1}{\text{dist}(0,\partial V(x^k)} \geq \frac{1}{a_2\left\|x^k-x^{k-1}\right\|}. \tag{19}$$

Since any concave differentiable function f satisfies the inequality

$$f(x)-f(y) \geq f^{'}(x)(x-y),$$

using this fact about the concave function combined with (C1) we have

$$\begin{aligned}\varphi(V(x^k)-V(x^*))-\varphi(V(x^{k+1})-V(x^*)) \geq& \varphi^{'}(V(x^k)-V(x^*))\left(V(x^k)-V(x^{k+1})\right)\\ \geq& \varphi^{'}(V(x^k)-V(x^*))\left(a_1\left\|x^{k+1}-x^k\right\|^2\right).\end{aligned} \tag{20}$$

By combining the inequalities in (19) and (20) we have

$$\frac{a_2}{a_1}\left(\varphi(V(x^k)-V(x^*))-\varphi(V(x^{k+1})-V(x^*))\right) \geq \frac{\left\|x^{k+1}-x^k\right\|^2}{\left\|x^k-x^{k-1}\right\|},$$

and by multiplying the both sides with $\left\|x^k-x^{k-1}\right\|$ and taking the square root we have

$$\left\|x^{k+1}-x^k\right\| \leq \sqrt{\left\|x^k-x^{k-1}\right\|\frac{a_2}{a_1}\left(\varphi(V(x^k)-V(x^*))-\varphi(V(x^{k+1})-V(x^*))\right)}. \tag{21}$$

Then, by applying Young's inequality $2\sqrt{\alpha\beta} \leq \alpha+\beta$ to Equation (21) and multiplying the both sides by 2 we get

$$\begin{aligned}2\left\|x^{k+1}-x^k\right\| \leq& 2\sqrt{\left\|x^k-x^{k-1}\right\|\frac{a_2}{a_1}\left(\varphi(V(x^k)-V(x^*))-\varphi(V(x^{k+1})-V(x^*))\right)}\\ \leq& \left\|x^k-x^{k-1}\right\|+\frac{a_2}{a_1}\left(\varphi(V(x^k)-V(x^*))-\varphi(V(x^{k+1})-V(x^*))\right),\end{aligned}$$

from which inequality (18) follows.

Let $\varepsilon,\delta>0$ such that $\delta \in (0,\varepsilon)$. We assume that

$$\left\|x^*-x^0\right\|+2\sqrt{\frac{V(x^0)-V(x^*)}{a_1}}+\frac{a_2}{a_1}\varphi'(V(x^0)-V(x^*)) < \delta, \tag{22}$$

holds, where $x^0 \in \mathscr{N}(x^*,\delta)$ and

$$\forall k \in \mathbb{N}, x^k \in \mathscr{N}(x^*,\delta) \Longrightarrow x^{k+1} \in \mathscr{N}(x^*,\varepsilon) \quad \text{where } V(x^*) \leq V(x^{k+1}). \tag{23}$$

The intuition behind the assumption (22) is that if our initial starting solution is close to the optimal solution of $V(x)$, the generated sequence will also remain bounded. The properties of the concave function, φ, also play a role in it.

We claim that $x^j \in \mathscr{N}(x^*, \varepsilon)$ hold for $j = 1, 2, \ldots$. We prove this claim by induction. From (17), when $k = 0$, we have

$$\left\|x^1 - x^0\right\| \leq \sqrt{\frac{V(x^0) - V(x^*)}{a_1}}. \tag{24}$$

By combining inequality (24) with the assumption (22), and using triangle inequality we have

$$\left\|x^* - x^1\right\| \leq \left\|x^* - x^0\right\| + \left\|x^0 - x^1\right\| \leq \left\|x^* - x^0\right\| + \sqrt{\frac{V(x^0) - V(x^*)}{a_1}} < \delta,$$

which concludes that $x^1 \in \mathscr{N}(x^*, \delta)$. Suppose that the claim $x^j \in \mathscr{N}(x^*, \delta)$ holds for $j \geq 1$. We will now show $x^{j+1} \in \mathscr{N}(x^*, \delta)$. By taking the sum of (18) from $k = 1$ to j we get the following

$$\begin{aligned}
\sum_{k=1}^{j} 2\left\|x^{k+1} - x^k\right\| - \sum_{k=1}^{j}\left\|x^k - x^{k-1}\right\| &= \sum_{k=1}^{j}\left\|x^{k+1} - x^k\right\| + \sum_{k=1}^{j}\left(\left\|x^{k+1} - x^k\right\| - \left\|x^k - x^{k-1}\right\|\right) \\
&= \sum_{k=1}^{j}\left\|x^{k+1} - x^k\right\| + \left\|x^{j+1} - x^j\right\| - \left\|x^1 - x^0\right\| \\
&\leq \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*)) - \varphi(V(x^{j+1}) - V(x^*))\right),
\end{aligned}$$

which can be written visibly as

$$\sum_{k=1}^{j}\left\|x^{k+1} - x^k\right\| + \left\|x^{j+1} - x^j\right\| \leq \left\|x^1 - x^0\right\| + \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*)) - \varphi(V(x^{j+1}) - V(x^*))\right). \tag{25}$$

Then by adding and subtracting each $x^0, x^1, \cdots, x^j$ with the term $x^* - x^{j+1}$ and using triangle inequalities, and by (25) we have

$$\begin{aligned}
&\left\|x^* - x^{j+1}\right\| \\
&\leq \left\|x^* - x^0\right\| + \left\|x^0 - x^1\right\| + \sum_{k=1}^{j} \left\|x^k - x^{k+1}\right\| && \text{(26a)} \\
&\leq \left\|x^* - x^0\right\| + \left\|x^0 - x^1\right\| + \sum_{k=1}^{j} \left\|x^k - x^{k+1}\right\| + \left\|x^j - x^{j+1}\right\|, && \text{(26b)} \\
&\leq \left\|x^* - x^0\right\| + 2\left\|x^0 - x^1\right\| + \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*)) - \varphi(V(x^{j+1}) - V(x^*))\right), \quad \text{by (25)}, && \text{(26c)} \\
&\leq \left\|x^* - x^0\right\| + 2\sqrt{\frac{V(x^0) - V(x^*)}{a_1}} + \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*)) - \varphi(V(x^{j+1}) - V(x^*))\right), \quad \text{by (24)} && \text{(26d)} \\
&\leq \left\|x^* - x^0\right\| + 2\sqrt{\frac{V(x^0) - V(x^*)}{a_1}} + \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*))\right) && \text{(26e)} \\
&\leq \left\|x^* - x^0\right\| + 2\sqrt{\frac{V(x^0) - V(x^*)}{a_1}} + \frac{a_2}{a_1}\left(\varphi(V(x^0) - V(x^*))\right) < \delta, && \text{(26f)}
\end{aligned}$$

where we have added the positive term, $\left\|x^j - x^{j+1}\right\|$, to the right hand side of inequality (26a). Since the term $\varphi(V(x^{j+1}) - V(x^*))$ in inequality (26d) is positive, so removing it will results in (26e). Inequality (26f) follows from $V(x^1) \leq V(x^0)$, of Lyapunov function V. Thus, we conclude that $x^{j+1} \in \mathscr{N}(x^*, \delta)$. Therefore, inequality (18) holds for $k = j+1$, namely

$$2\left\|x^{(j+1)+1} - x^{j+1}\right\| \leq \left\|x^{j+1} - x^j\right\| + \frac{a_2}{a_1}\left(\varphi(V(x^{j+1}) - V(x^*)) - \varphi(V(x^{(j+1)+1}) - V(x^*))\right). \tag{27}$$

Inequality (25) implies

$$\begin{aligned}
\sum_{k=1}^{j} \left\|x^{k+1} - x^k\right\| &\leq \left\|x^1 - x^0\right\| + \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*))\right) - \left[\frac{a_2}{a_1}\varphi(V(x^{j+1}) - V(x^*)) + \left\|x^{j+1} - x^j\right\|\right] \\
&\leq \left\|x^1 - x^0\right\| + \frac{a_2}{a_1}\left(\varphi(V(x^1) - V(x^*))\right),
\end{aligned}$$

since the expression in bracket is positive.

Therefore, it follows that

$$\sum_{k=1}^{\infty} \left\|x^{k+1} - x^k\right\| < \infty,$$

which infers that the sequence $\{x^k\}$ converge to some $\bar{x}$. Then, from condition (C2) and the proven claim $x^j \in \mathscr{N}(x^*, \delta), j = 1, 2, \cdots$, we conclude that there exist $v^k \in \partial V(x^k)$ such that $v^k \to 0$ as $k \to \infty$ and

$$\liminf_{k \to \infty} V(x^k) = V(\bar{x}) \geq V(x^*), \quad \text{by lower semi-continuity.} \tag{28}$$

If $V(\bar{x}) > V(x^*)$, then using K-L inequality we have

$$\varphi^{'}(V(\bar{x}) - V(x^*)) \left\| v^k \right\| \geq 1,$$

which contradicts the fact that $v^k \to 0$ as $k \to \infty$. Therefore

$$V(\bar{x}) \leq V(x^*). \tag{29}$$

Then, the inequalities (28) and (29) conclude that $V(\bar{x}) = V(x^*)$. Due to the convergence of x^k to $\bar{x}$, the regularized term $\|x_3 - \bar{x}_3\| \to 0$ from where $V(x^*) = L_\rho(x^*)$ follows. □

Lemma 2 *For each $k \in \mathbb{N}$, there exists a positive constant $a_1 > 0$ such that $V(\hat{w}^{k+1}) \leq V(\hat{w}^k) - a_1 \left\| \hat{w}^{k+1} - \hat{w}^k \right\|$, where $\hat{w}^k = (x_1^k, x_2^k, x_3^k, \mu^k, x_3^{k-1})$.*

Proof Taking partial derivative with respect to x_3 at iteration $k+1$ and setting

$$\nabla_{x_3} V(x_1^{k+1}, x_2^{k+1}, x_3, \mu^k, x_3^k) = 0,$$

yields

$$\begin{aligned} &\nabla_{x_3} h(x_3^{k+1}) + A_3^T \mu^k + \rho A_3^T (A_1 x_1^{k+1} + A_2 x_2^{k+1} + A_3 x_3^{k+1} - t) + \\ &\gamma(x_3^{k+1} - x_3^k) = \nabla_{x_3} h(x_3^{k+1}) + A_3^T \mu^{k+1} + \gamma(x_3^{k+1} - x_3^k) = 0. \end{aligned} \tag{30}$$

Using the expressions for $\nabla_{x_3} h(x^{k+1})$ and $\nabla_{x_3} h(x^k)$ from (30) and using the Cauchy-Schwarz and Young inequalities after re-arranging of terms we get

$$\begin{aligned} \left\| A_3^T(\mu^{k+1} - \mu^k) \right\|^2 &= \left\| (\nabla h(x_3^{k+1}) - \nabla h(x_3^k)) + \gamma(x_3^{k+1} - x_3^k) - \gamma(x_3^k - x_3^{k-1}) \right\|^2 \\ &\leq \left\| (\nabla h(x_3^{k+1}) - \nabla h(x_3^k)) \right\|^2 + \gamma^2 \left\| (x_3^{k+1} - x_3^k) - (x_3^k - x_3^{k-1}) \right\|^2 \\ &+ 2\gamma \left\| \nabla h(x_3^{k+1}) - \nabla h(x_3^k) \right\| \left\| (x_3^{k+1} - x_3^k) - \gamma(x_3^k - x_3^{k-1}) \right\| \\ &= \gamma^2 \left(\left\| (x_3^{k+1} - x_3^k) - (x_3^k - x_3^{k-1}) \right\|^2 \right) \\ &\leq \gamma^2 \left(\left\| x_3^{k+1} - x_3^k \right\|^2 + \left\| x_3^k - x_3^{k-1} \right\|^2 + 2 \left(\left\| x_3^{k+1} - x_3^k \right\| \left\| (x_3^k - x_3^{k-1} \right\| \right) \right) \\ &\leq 2\gamma^2 \left(\left\| x_3^{k+1} - x_3^k \right\|^2 + \left\| x_3^k - x_3^{k-1} \right\|^2 \right), \quad \text{by Young inequality} \end{aligned}$$

where $\nabla h(x^{k+1}) - \nabla h(x^k) = 0$ because h is a linear function. Then it follows from property $(A2)$ that

$$\left\| \mu^{k+1} - \mu^k \right\|^2 \leq \frac{2\gamma^2}{\sigma} \left(\left\| x_3^{k+1} - x_3^k \right\|^2 + \left\| x_3^k - x_3^{k-1} \right\|^2 \right). \tag{31}$$

Also, we have

$$L_\rho(x_1^{k+1},x_2^k,x_3^k,\mu^k)+\frac{\gamma}{2}\left\|x_1^{k+1}-x_1^k\right\|^2 \le L_\rho(x_1^k,x_2^k,x_3^k,\mu^k), \tag{32a}$$

$$L_\rho(x_1^{k+1},x_2^{k+1},x_3^k,\mu^k)+\frac{\gamma}{2}\left\|x_2^{k+1}-x_2^k\right\|^2 \le L_\rho(x_1^{k+1},x_2^k,x_3^k,\mu^k), \tag{32b}$$

$$L_\rho(x_1^{k+1},x_2^{k+1},x_3^{k+1},\mu^k)+\frac{\gamma}{2}\left\|x_3^{k+1}-x_3^k\right\|^2 \le L_\rho(x_1^{k+1},x_2^{k+1},x_3^k,\mu^k), \tag{32c}$$

$$L_\rho(x_1^{k+1},x_2^{k+1},x_3^{k+1},\mu^{k+1}) = L_\rho(x_1^{k+1},x_2^{k+1},x_3^{k+1},\mu^k)+\frac{1}{\rho}\left\|\mu^{k+1}-\mu^k\right\|^2 \tag{32d}$$

where the inequalities (32a)-(32c) follows from (3), and the equality (32d) follows from (2) and (3). By adding up (32a)-(32d), we obtain

$$L_\rho(w^{k+1}) \le L_\rho(w^k)-\frac{\gamma}{2}\left\|u^{k+1}-u^k\right\|^2+\frac{1}{\rho}\left\|\mu^{k+1}-\mu^k\right\|^2, \tag{33}$$

where $u^k=(x_1^k,x_2^k,x_3^k)$, and $w^k=(x_1^k,x_2^k,x_3^k,\mu^k)$.

Dividing both sides of inequality (31) by ρ and adding up with inequality (33) it follows that

$$L_\rho(w^{k+1}) \le L_\rho(w^k)-\frac{\gamma}{2}\left\|u^{k+1}-u^k\right\|^2+\frac{2\gamma^2}{\rho\sigma}\left(\left\|x_3^{k+1}-x_3^k\right\|^2+\left\|x_3^k-x_3^{k-1}\right\|^2\right) \tag{34}$$

Therefore, by rearranging the terms of Equation (34) using $\tau=4\frac{\gamma^2}{\rho\sigma}$, we get

$$\begin{aligned}
V(\hat{w}^{k+1}) &= L_\rho(w^{k+1})+\frac{\tau}{2}\left\|x_3^{k+1}-x_3^k\right\|^2 \\
&\le L_\rho(w^k)+\frac{\tau}{2}\left\|x_3^k-x_3^{k-1}\right\|^2-\frac{\gamma}{2}\left\|u^{k+1}-u^k\right\|^2+\tau\left\|x_3^{k+1}-x_3^k\right\|^2 \\
&= V(\hat{w}^k)-\frac{\gamma}{2}\left\|u^{k+1}-u^k\right\|^2+\tau\left\|x_3^{k+1}-x_3^k\right\|^2 \\
&\le V(\hat{w}^k)-\frac{\gamma}{2}\left\|u^{k+1}-u^k\right\|^2+\tau\left\|u_3^{k+1}-u_3^k\right\|^2 \\
&\le V(\hat{w}^k)-a_1\left\|\hat{w}^{k+1}-\hat{w}^k\right\|^2,
\end{aligned} \tag{35}$$

where $a_1 \in (0,\frac{\gamma}{2}-\tau]$, which is positive, sine $\frac{\gamma}{2}>\tau$, $\rho>\frac{8\gamma}{\sigma}$ from Assumption (A3). □

Lemma 3 (Lemma 3, [20]) *Let u^k and w^k respectively denote (x_1^k,x_2^k,x_3^k) and $(x_1^k,x_2^k,x_3^k,\mu^k)$. If the sequence $\{u^k\}$ is bounded, then $\sum_{k=1}^{\infty}\left\|\hat{w}^k-\hat{w}^{k+1}\right\|^2<\infty$. In fact, w^k is asymptotically regular, indeed, $\left\|w^k-w^{k+1}\right\|\to 0$ as $k\to\infty$. Moreover, any accumulation point of $\{w^k\}$ is a stationary point of the augmented Lagrangian function L_ρ.*

Proof From Assumption (A2), and Equation (30) we have for $c=\left\|\nabla h(x_3^k)\right\|$

$$\sqrt{\sigma}\left\|\mu^k\right\| \le \left\|A_3^T\mu^k\right\| \le \left\|\nabla h(x_3^k)\right\|+\gamma\left\|x_3^k-x_3^{k-1}\right\| \le c+\gamma\left\|x_3^k-x_3^{k-1}\right\|.$$

It follows that $\{\mu^k\}$ is bounded since $\{u^k\}$ is bounded. Hence $\{\hat{w}^k\}$ is bounded since $\hat{w}^k = (x_1^k, x_2^k, x_3^k, \mu^k, x_3^{k-1})$ is a combination of u^k and μ^k. Thus, for being the iterates of lower semicontinuous function V, there exist a subsequence $\{\hat{w}^{k_j}\}$ which converges to $\hat{w}^*$; it follows $\liminf_{j\to\infty} V(\hat{w}^{k_j}) \geq V(\hat{w}^*)$. Lemma 1 showed that the function V decreases in each iteration, and thus $\{\hat{w}^{k_j}\}$ is convergent. Taking the summation for Equation (35) it follows that

$$a_1 \sum_{i=1}^{k} \left\|\hat{w}^{i+1} - \hat{w}^i\right\|^2 \leq V(\hat{w}^1) - V(\hat{w}^{k+1}) \leq V(\hat{w}^1) - V(\hat{w}^*),$$

which implies that $\sum_{k=1}^{\infty} \left\|\hat{w}^{k+1} - \hat{w}^k\right\|^2 < \infty$; then it follows that $\left\|\hat{w}^{k+1} - \hat{w}^k\right\|^2 \to 0$ as $k \to \infty$. This implies that primal and dual residuals converge to zero as $k \to \infty$. □

The following lemma states that $\texttt{dist}(0, \partial V(\hat{w}^{k+1}))$ decreases with the iteration $k+1$.

Lemma 4 *There is a positive constant a_2 such that $\texttt{dist}(0, \partial V(\hat{w}^{k+1})) \leq a_2 \left\|\hat{w}^{k+1} - \hat{w}^k\right\|$.*

Proof By taking the partial derivative of V in (16) with respect x_1 and equating it with zero and substituting $\mu^k = \mu^{k+1} - \rho\left(A_1 x_1^{k+1} + A_2 x_2^{k+1} + A_3 x_3^{k+1} - t\right)$ we have

$$\begin{aligned} 0 &= \partial f(x_1^{k+1}) + A_1^T \mu^k + \rho A_1^T \left(A_1 x_1^{k+1} + A_2 x_2^k + A_3 x_3^k - t\right) \\ &= \partial f(x_1^{k+1}) + A_1^T \mu^{k+1} + \rho A_1^T A_2 (x_2^k - x_2^{k+1}) + \rho A_1^T A_3 (x_3^k - x_3^{k+1}), \end{aligned}$$

by adding and subtracting the term $A_1^T\left(\mu^{k+1} - \mu^k\right)$ to the right hand-side yields

$$\begin{aligned} 0 = \underbrace{\partial f(x_1^{k+1}) + A_1^T \mu^{k+1} + A_1^T \left(\mu^{k+1} - \mu^k\right)}_{\nabla_{x_1} V(\hat{w}^{k+1})} + \rho A_1^T A_2 (x_2^k - x_2^{k+1}) + \\ \rho A_1^T A_3 (x_3^k - x_3^{k+1}) - A_1^T \left(\mu^{k+1} - \mu^k\right), \end{aligned}$$

where in $\nabla_{x_1} V(x_1, x_2^{k+1}, x_3^{k+1}, \mu^{k+1}, x_3^k)\,|_{x_1 = x_1^{k+1}}$ we have used

$$\mu^k = \mu^{k+1} - \rho\left(A_1 x_1^{k+1} + A_2 x_2^{k+1} + A_3 x_3^{k+1} - t\right)$$

.

Replicating this step for the rest of variables respectively yields

$$\begin{aligned} 0 &= \partial g(x_2^{k+1}) + A_2^T \mu^{k+1} + A_2^T \left(\mu^{k+1} - \mu^k\right) + \rho A_2^T A_3 (x_3^k - x_3^{k+1}) - A_2^T \left(\mu^{k+1} - \mu^k\right), \\ 0 &= \partial h(x_3^{k+1}) + A_3^T \left(\mu^{k+1} - \mu^k\right) + A_3^T \mu^{k+1} - A_3^T \left(\mu^{k+1} - \mu^k\right), \\ 0 &= \nabla_\mu V(\hat{w}^{k+1}) = \frac{1}{\rho}(\mu^{k+1} - \mu^k) \quad \text{and} \\ 0 &= \nabla_{\bar{x}_3} V(\hat{w}^{k+1}) = \gamma(x_3^k - x_3^{k+1}). \end{aligned}$$

Adding up all the previous equations associated with each variable and applying the Frobenius norm to the both sides results in

$$\texttt{dist}(0,\partial V(\hat{w}^{k+1}) \leq b_1 \left\| x_2^k - x_2^{k+1} \right\| + b_2 \left\| x_3^k - x_3^{k+1} \right\| + b_3 \left\| \mu^{k+1} - \mu^k \right\|,$$

where $b_1 = \left\| \rho A_1^T A_2 \right\|_F, b_2 = \rho \left(\left\| A_1^T A_3 \right\|_F + \left\| A_2^T A_3 \right\|_F + \frac{\gamma}{\rho} \right)$ and , $b_3 = \|A_1\|_F + \|A_2\|_F + \|A_3\|_F + \frac{1}{\rho}$. Since the matrices A_1, A_2, A_3 are all bounded in Frobenius norm, implies that $\exists a_2 > 0$ such that $a_2 = \max\{b_1, b_2, b_3\}$. Then, adding the appropriate norm for x_1^{k+1} and x_3^{k-1} the inequality holds. □

Lemma 5 *Suppose the sequence* $\{w^{k_j}\}$ *generated by the iterative procedure* (3) *converge to* $\hat{w}^*$*and* V *is lower semicontinuous, i.e.* $\liminf_{j\to\infty} V(\hat{w}^{k_j}) \geq V(\hat{w}^*)$*, then* $\lim_{j\to\infty} V(\hat{w}^{k_j}) = V(\hat{w}^*)$.

Proof The proof of convergence follows directly from Lemma 1, in which the function V satisfies the conditions (C1)-(C3). We have shown in Lemmas 2-4 that V meets all conditions C1)-(C3). □

5 ADMM for Two-Stage SP

We consider deterministic version of the two-stage SP (11)-(15) and the notations used thereof. We redefine the variables x_1, x_2, and x_3 of Equation (1) as $x_1 := (y_1, y_2, \cdots, y_s)^T$, $x_2 := (\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_s, \hat{x})^T$, and $x_3 := x$. Correspondingly, the functions will be defined as follows:

$$\begin{aligned} f(x_1) &:= \sum_{i=1}^{s} p_i q_i^T y_i, \\ g(x_2) &:= \sum_{i=1}^{s} I(\hat{y}_i) + I(\hat{x}), \quad \text{and} \\ h(x_3) &:= c^T x. \end{aligned}$$

The vector of multipliers is defined as

$$\mu = (\alpha, \beta, \gamma_1, \cdots, \gamma_s, \delta_1, \cdots, \delta_s)^T.$$

In what follows, we derive the detailed steps to find the solution to Problem (11)-(15), and then we present a step by step algorithm for the solution. Consider the following augmented Lagrangian function

$$\begin{aligned} L_\rho(x,\hat{x},y,\hat{y}) =& c^T x + \beta^T (x-\hat{x}) + I(\hat{x}) + \sum_{i=1}^{s} \left(p_i q_i^T y_i + \gamma_i^T (y_i - \hat{y}_i) + I(\hat{y}_i) \right) \\ &+ \alpha^T (Ax-b) + \sum_{i=1}^{s} \delta_i^T (T_i x + W_i y_i - h_i) + \frac{\rho}{2} (\|Ax-b\|^2 \\ &+ \|x-\hat{x}\|^2 + \sum_{i=1}^{s} (\|T_i x + W_i y_i - h_i\|^2 + \|y_i - \hat{y}_i\|^2)), \end{aligned} \tag{36}$$

where $\beta \in \mathbb{R}^{n_1}$, $\alpha \in \mathbb{R}^{m_1}$, $\gamma_i \in \mathbb{R}^{n_2}$, and $\delta_i \in \mathbb{R}^{m_2}$, $i = 1,2,\cdots,s$.

Solving $\nabla_x L_\rho = 0$ yields:

$$\nabla_x L_\rho(x,\hat{x},y,\hat{y}) = c+\beta+A^T\alpha+\sum_{i=1}^{s}T_i^T\delta_i+\rho(A^T(Ax-b)+(x-\hat{x})+\sum_{i=1}^{s}T_i^T(T_ix+W_iy_i-h_i))=0$$

$$c+\beta+A^T\alpha+\sum_{i=1}^{s}T_i^T\delta_i+\rho(-A^Tb-\hat{x}+\sum_{i=1}^{s}T_i^T(W_iy_i-h_i))+\rho(A^TA+\mathbf{I}+\sum_{i=1}^{s}T_i^TT_i)x=0$$

$$\begin{aligned} x =&(A^TA+\mathbf{I}+\sum_{i=1}^{s}T_i^TT_i)^{-1}(A^Tb+\hat{x}+\sum_{i=1}^{s}T_i^T(h_i-W_iy_i) \\ &+\frac{-1}{\rho}(c+\beta+A^T\alpha+\sum_{i=1}^{s}T_i^T\delta_i)). \end{aligned} \tag{37}$$

Similarly, $\nabla_{y_i} L_\rho = 0, \quad i = 1,2,\cdots,s$, yields:

$$\begin{aligned} \nabla_{y_i} L_\rho(x,\hat{x},y,\hat{y}) = p_iq_i+\gamma_i+W_i^T\delta_i+ \ &+\rho\left(W_i^T(T_ix+W_iy_i-h)+(y_i-\hat{y}_i)\right)=0 \\ p_iq_i+\gamma_i+W_i^T\delta_i+\rho\left(W_i^T(T_ix-h_i)-\hat{y}_i\right)&+\rho\left(W_i^TW_i+\mathbf{I}\right)y_i=0 \end{aligned}$$

$$\begin{aligned} y_i =&\left(W_i^TW_i+\mathbf{I}\right)^{-1}(W_i^T(h_i-T_ix)+\hat{y}_i+ \\ &\frac{-1}{\rho}(p_iq_i+\gamma_i+W_i^T\delta_i)), \quad i=1,2,\cdots,s. \end{aligned} \tag{38}$$

And by differentiating L_ρ with respect to $\hat{x},\hat{y}_i$ respectively yields

$$\begin{aligned} \hat{x} &= \max\{x+\frac{\beta}{\rho},0\}, \\ \hat{y}_i &= \max\{y_i+\frac{\gamma_i}{\rho},0\}, \quad i=1,2,\cdots,s. \end{aligned}$$

The updates for the dual variables will be expressed as

$$\begin{aligned} \beta &= \beta+\rho\,(x-\hat{x}), \\ \alpha &= \alpha+\rho\,(Ax-b), \\ \gamma_i &= \gamma_i+\rho\,(y_i-\hat{y}_i), \\ \delta_i &= \delta_i+\rho\,(T_ix+W_iy_i-h_i), \quad i=1,2,\cdots,s. \end{aligned} \tag{39}$$

Algorithm 1 presents the steps of ADMM for solving two-stage SP in Equations (11)-(15). An important parameter of Algorithm 1 is the penalization parameter ρ. The convergence of the ADMM algorithm is very sensitive to such a choice; poor selection may lead to slow or non-convergence in practical problems [18]. A variant of ADMM, residual balancing, where the penalty parameter ρ_k changes at each iteration k is proposed by He et al [11]. The intuition behind the method is based on making the primal and dual residual norms to have similar magnitudes. By doing so, the primal and dual residual will have small values at the stage of convergence. This approach makes the performance less dependent on the initial choice of ρ. The superlinear convergence with $\rho_k \to \infty$ has been achieved by Rockefellar [16]. We

Algorithm 1 ADMM for two stage stochastic programs

Require: $\rho > 0$, $k_{max} > 0$, $\varepsilon > 0$. All the initialization to the variables $\hat{x}^0, y_i^0, \alpha^0, \beta^0, \delta_i^0, \gamma_i^0, \quad i = 1, \cdots s$, are obtained by first phase Linear programming step.

1:

$$x^1 \leftarrow (A^T A + \mathtt{I} + \sum_{i=1}^{s} T_i^T T_i)^{-1} (A^T b + \hat{x}^0 + \sum_{i=1}^{s} T_i^T (h_i - W_i y_i^0) + \frac{-1}{\rho}(c + \beta^0 + A^T \alpha^0 + \sum_{i=1}^{s} T_i^T \delta_i^0)), \qquad \text{by (37)}$$

2: $k \leftarrow 1$
3: **while** True **do**
4: **for** $i \leftarrow 1$ to s **do**
5:

$$y_i^k \leftarrow \left(W_i^T W_i + \mathtt{I}\right)^{-1} \left(W_i^T (h_i - T_i x^k) + \hat{y}_i^k + \frac{-1}{\rho}\left(p_i q_i + \gamma_i^k + W_i^T \delta_i^k\right)\right), \qquad \text{by (38)}$$

6: **end for**
7:

$$\hat{x}^{k+1} \leftarrow \max\{x^k + \frac{\beta^k}{\rho}, 0\}$$

8:

$$\hat{y}_i^{k+1} \leftarrow \max\{y_i^k + \frac{\gamma_i^k}{\rho}, 0\}, \quad i = 1, \cdots, s.$$

9:

$$x^{k+1} \leftarrow (A^T A + \mathtt{I} + \sum_{i=1}^{s} T_i^T T_i)^{-1} (A^T b + \hat{x}^{k+1} + \sum_{i=1}^{s} T_i^T (h_i - W_i y_i^{k+1}) + \frac{-1}{\rho}(c + \beta^k + A^T \alpha^k + \sum_{i=1}^{s} T_i^T \delta_i^k)), \qquad \text{by (37)}$$

10: $\alpha^{k+1} \leftarrow \alpha^k + \rho\left(Ax^{k+1} - b\right)$
11: $\beta^{k+1} \leftarrow \beta^k + \rho\left(x^{k+1} - \hat{x}^{k+1}\right)$
12: $\delta_i^{k+1} \leftarrow \delta_i^k + \rho\left(T_i x^{k+1} + W_i y_i^{k+1} - h_i\right), \quad i = 1, \cdots, s.$
13: $\gamma_i^{k+1} \leftarrow \gamma_i^k + \rho\left(y_i^{k+1} - \hat{y}_i^{k+1}\right), \quad i = 1, \cdots, s.$
14: $r_0 \leftarrow Ax^{k+1} - b$
15: $r_i \leftarrow T_i x^{k+1} + W_i y_i^{k+1} - h_i, \quad i = 1, \cdots, s$
16: $s_i^1 \leftarrow \rho(x^{k+1} - x^k)$
17: $s_0^2 \leftarrow \rho(\hat{x}^k - \hat{x}^{k+1})$
18: $s_i^2 \leftarrow \rho(\hat{y}_i^k - \hat{y}_i^{k+1})$
19:

$$r = \left(r_0, r_1, \ldots, r_s,\right)^T, \quad s^1 = \left(s_1^1, s_2^1, \ldots, s_s^1,\right)^T, \quad s^2 = \left(s_0^2, s_1^2, \ldots, s_s^2,\right)^T$$

20: If $\|r\| \leq \varepsilon$, $\left\|s^1\right\| \leq \varepsilon$, and $\left\|s^2\right\| \leq \varepsilon$ stop; otherwise $k \leftarrow k + 1$
21: If k equals k_{max}, stop; the algorithm does not converge.
22: **end while**

have implemented iteration dependent adaptive ρ_k as suggested by He et al. [11]. In Algorithm 1, the preconditioning step initializes penalty parameter ρ, the maximum number of iterations k_{max}, and the convergence tolerance ε. The initialization in Line 1 provides an initial first-stage solution for the primary iterations $k \geq 1$. The initial penalty parameter will be adaptively updated by He et al. [11] to maintain the gap difference between the primal and dual residuals, see Equation (40). Algorithm 1 runs on deterministic initializations, in which the first phase procedure of linear programming is applied to bring the starting point to a feasible region. Since Algorithm 1 decomposes the problem by scenarios, Lines $5, 8, 12, 13, 15$, and 18 are implemented in parallel. Line 5 provides a scenario-wise solution to the second-stage variable in which all the components of the second-stage variable are solved concurrently. The second-stage solution needs to be assembled in one variable to find the first-stage solution, which is given in Line 9. Lines 10-13 give the updates for the dual variables using (39). The primal and dual residuals of Algorithm 1 are given in Lines 14-19. Algorithm 1 terminates in two ways; the convergence case, where all the residuals are less than the threshold ε, Line 20, or the non-convergence case, Line 21, where the algorithm hits the maximum number of iterations without meeting the convergence requirements. In the non-convergence case, the algorithm is enforced to stop by the maximum iteration.

6 Numerical Results

This section discusses the numerical experiments of Algorithm 1 on five benchmarks two-stage SP. The convergence and CPU time of the algorithm are also discussed.

The computer specification that runs the experiments has CPU Intel ® Core ^{TM}i7-7700 CPU @ 3.60GHz $\times 8$ and 15.5 GB of RAM. All the experiments have been carried out in MATLAB 2020. In the initialization step, we have defined the penalty parameter ρ as suggested by He et al. [11]:

$$\rho_{k+1} = \begin{cases} \nu\rho_k, & ||r_k|| > \mu \max\{||s_k^1||, ||s_k^2||\} \\ \rho_k/\nu, & \min\{||s_k^1||, ||s_k^2||\} > \mu||r_k|| \\ \rho_k, & \text{otherwise,} \end{cases} \tag{40}$$

where $\nu > 1$, $\mu > 1$. The value $\varepsilon = 10^{-3}$ and $k_{max} = 5 \times 10^4$ were used, respectively, in steps 20 and 21 of Algorithm 1. The primal and dual residuals that appear in all the relevant figures in this section are given in Line 19 of Algorithm 1, where dual1 and dual2 denotes s^1 and s^2, respectively. In all the tested problems, we have found Algorithm 1 always converges to the optimal value.

Each problem is solved with an initial starting point obtained using the phase 1 procedure of linear programming. Therefore, Algorithm 1 has been executed with on each problem with a feasible point. The CPU times are recorded for all problems. All the graphs of the residuals are provided in the log/log scale.

Table 1 summarizes features of the tested problems used. All the tested problems are linear in their first and second stages. The problems *LandS* and *Gbd* are solved by Linderoth et al. [12] using Sample Average Approximation; the authors provided

lower and upper bounds for optimal solutions. On the other hand, the problems *Assets* and *Phone* are taken from [1] for which the data can be found in the website stated in the paper. All the input data of the problems in Table 1 are provided in SMPS (Stochastic Mathematical Programming System) format for stochastic linear programs.

Table 1: Test Problem Data [12]

Name	Application	Scenarios	First stage size	Second stage size
LandS	Electricity planning	10^6	$(2,4)$	$(7,12)$
Assets	Asset management	3.75×10^4	$(5,8)$	$(5,8)$
Phone	Network planning	2^{15}	$(1,9)$	(23,93)
Gbd	aircraft allocation	6.5×10^5	$(4,17)$	$(5,10)$

In Table 1, the data in the last two columns are given in pair (m,n), where m is the number of rows, and n is the number of columns. The first stage size refers to the size of matrix A, and the second stage size refers to the size of matrix W, see Equation (10).

Table 2 reports the statistical bounds for the optimal value for problems *LandS* and *Gdb* found by Linderoth et al. [12].

Table 2: Optimal values for *LandS* and *gbd* for various scenarios.

Problem	Scenarios	95% confidence intervals
LandS	50	$[225.71\pm0.12]$
	100	$[225.55\pm0.12]$
	500	$[225.16\pm0.12]$
	1000	$[225.70\pm0.13]$
Gbd	50	$[1655.86\pm1.34]$
	100	$[1656.35\pm1.19]$
	500	$[1654.90\pm1.46]$
	1000	$[1655.70\pm1.49]$

6.1 Electrical Investment Planning (LandS Problem)

6.1.1 Small Size LandS

This example consists of four technologies and three different modes. The variable x_i describes the new capacity of technology i, and y_{ij} is the production rate from technology i for mode j. The investment cost, generate cost and load duration are $c=(10,7,16,6)^T$, $q=(4,4.5,3.2,5.5)^T$, and $\tau=\{10,6,1\}$, respectively [4]. All the variables are deterministic except the first demand $\mathbf{d}_1=\xi$ which takes the values $\{3,5,7\}$ with the corresponding probabilities $\{0.3,0.4,0.3\}$. The other demands are

deterministic and takes the values $d_2 = 3$ and $d_3 = 2$. So, the resulting problem takes the following form:

$$\begin{aligned} \min \quad & c^T x + E_\xi \sum_{j=1}^{3} \tau_j q^T y \qquad \text{subject to} \\ & 10x_1 + 7x_2 + 16x_3 + 6x_4 \leq 120, \\ & \sum_{i=1}^{4} x_i = 12, \\ & \sum_{i=1}^{4} y_{i1} = \xi, \\ & \sum_{i=1}^{4} y_{ij} = d_j, \qquad j = 2,3 \\ & x \geq 0, \quad y \geq 0. \end{aligned} \tag{41}$$

Louveaux and Smeers [14] report the optimal solution to Problem (41) to be

$$x^* = (2.666666, 4.0, 3.3333, 2.0)^T,$$

with the objective value of 381.853.

Using Algorithm 1, the optimal solution was found to be

$$x^* = (2.6664, 4.0001, 3.3335, 1.9999)^T,$$

with the optimum value of 381.67.

The result, as shown in Figure 1, presents the convergence of the residuals in 100 iterations, approximately.

6.1.2 Large Size LandS

This problem is a modified and extended version of the problem taken from [14].

The minimum total capacity and budget constraints are given in the first stage conditions, while the capacity restriction for each of the four technologies and the demands are in the second-stage. The demand constraint is given by

$$\sum_{i=1}^{4} y_{ij} \geq d_j, \quad j = 1,2,3,$$

where d_j's are given by

$$d_j = 0.04(k-1), \quad k = 1,2,\cdots,100, \quad j = 1,2,3,$$

all with a probability of 0.01. Demands are assumed independent, and therefore there are $(100)^3 = 10^6$ scenarios with equal probability of 10^{-6}.

Figure 2 indicates that Algorithm 1 takes 3000 iterations to converge. The obtained optimal value for this problem is 225.85, which agrees with the result in Table 2.

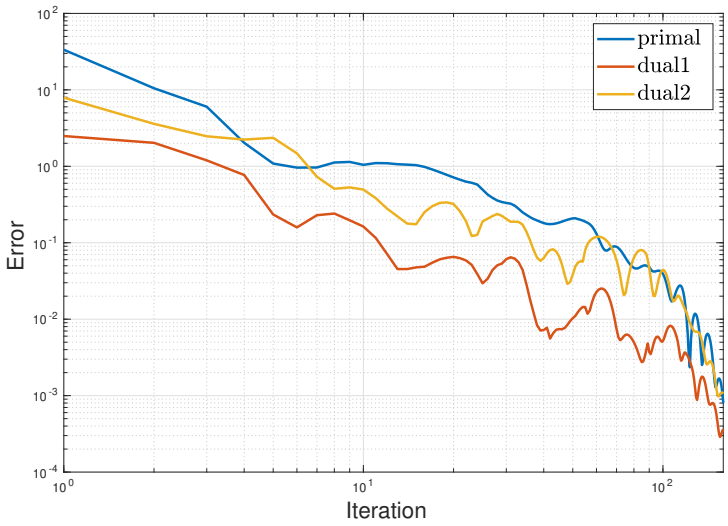


Fig. 1: Convergence behaviour of LandS-small problem.

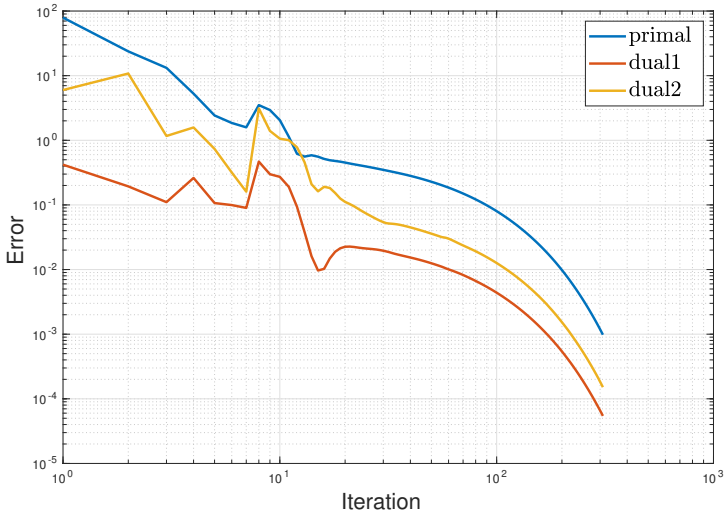


Fig. 2: Convergence behaviour of LandS-Large problem.

6.2 Network Model for Asset Management (Asset Problem)

In this problem, the assets are represented by nodes, while arcs represent the transactions. There are five nodes for which costs are fixed in every stage. These are checking, saving, certificate of deposit, cash, and loads [1]. This problem has a two-stage stochastic setting because the purchase or sale has deterministic costs, while the return on investment is random.

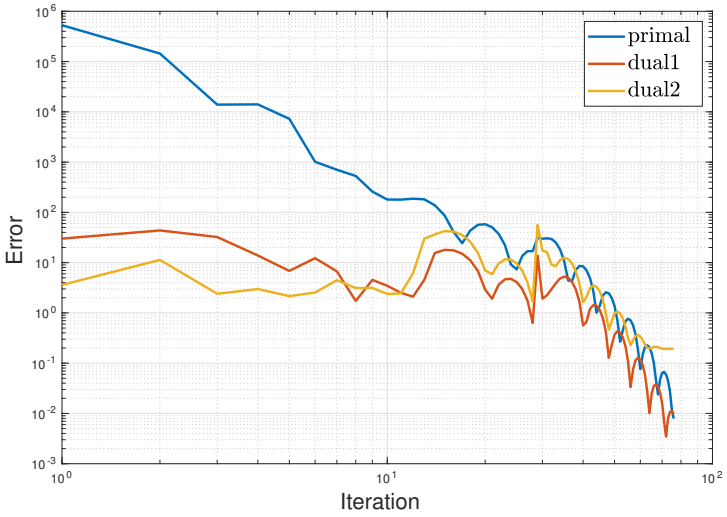


Fig. 3: Convergence behaviour of Asset problem.

We denote the set of nodes by $\mathscr{N}$ and the set of arcs by $\mathscr{A}$. The objective value will is calculated over the terminal arcs, which is denoted by $\mathscr{T} \subset \mathscr{A}$.

The objective function is defined as $f_s(v, y(s)) \in C^1$, which is a convex function under scenario $s \in S$, where $v := \{v_{ij} : (i,j) \in \mathscr{A}_1\} \in \mathbf{R}^{n_1}$ and $y(s) := \{y_{ij}(s) : (i,j) \in \mathscr{A}_2\} \in \mathbf{R}^{n_2}$, where $n_1 = |A_1|$ and $n_2 = |A_2|$, respectively, $n = n_1 + n_2$. Additionally, the deterministic and supply nodes are encoded in the vectors $b \in \mathbb{R}^{m_1}$ and $h \in \mathbb{R}^{m_2 \times |S|}$, where $m_1 = |\mathscr{N}_1|$ and $m_2 = |\mathscr{N}_2|$, respectively. The two-stage SP is given in the form [1]

$$\min \quad \sum_{s \in S} p_s f_s(v, y(s)) \qquad \text{subject to} \tag{42}$$

$$Av = b, \tag{43}$$

$$T(s)v + W(s)y(s) = h(s), \quad \forall s \in S, \tag{44}$$

$$l^1 \leq v \leq u^1, \quad l^2 \leq y(s) \leq u^2, \quad \forall s \in S. \tag{45}$$

For the complete mathematical description, we refer to [1].

This problem has two different sizes; the first one has 100 scenarios, and the second has 37,500 scenarios. The numerical results depicted in Figure 3 corresponding to the $37,500$ scenarios; the 100 scenarios problem shows the similar results. Figure 3 indicates that the algorithm took less than 100 iterations to converge. The algorithm converges to the optimal value of -696.74, which coincides with the value reported in [1],[7].

6.3 Telecommunication Network Planning (Phone Problem)

The manager of a telecommunication network has to plan for future growth. Decisions about where and how much to expand the capacity must be chosen optimally. In the formulation of this problem, the "how much" is decided before the future realization.

These types of networks are very interconnected in which several routes can provide services to the demand of point-to-point requests. Besides, each route has one or more direct links. The objective is to minimize the unserved requests by adding new links while staying within the budget constraint.

The problem is expressed mathematically as

$$\min_{x} \quad \mathbf{E}[Q(x,d)] \qquad \text{subject to}$$
$$\sum_{j=1}^{n} x_j \leq b, \quad x \geq 0,$$

where $Q(x,d)$ is expressing the number of unserved requests. The full description of the problem can be found in [1].

The number of scenarios, in this case, is 1000. We have found that all the sizes of $50, 100, 500, 1000$, and 5000 scenarios attain the same optimal value in our numerical experiment. Figure 4 represents that the algorithm converges in 100 iterations. The algorithm converges to the optimal value of 36.89. The optimal solution reported by the authors is 36.9 [1], [7].

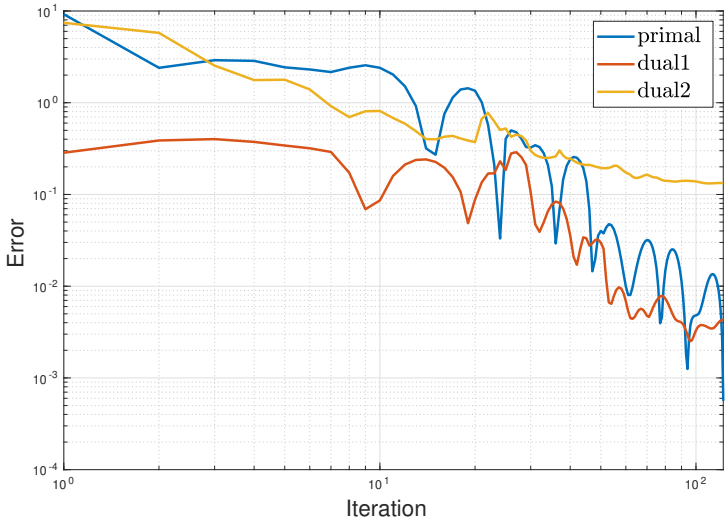


Fig. 4: Convergence behaviour for Phone problem.

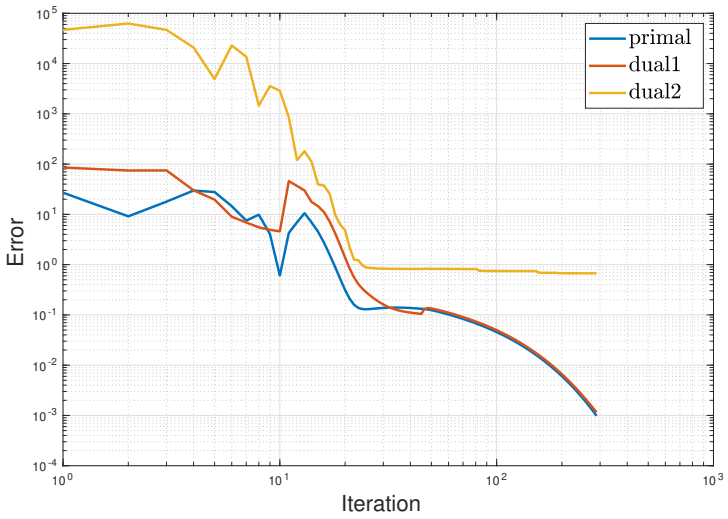


Fig. 5: Convergence behaviour for Gbd problem with 1000 scenarios

6.4 Aircraft Allocation (Gbd Problem)

In this problem, four types of aircraft have to be assigned to five routes to maximize the revenue under the stochastic demand. The first-stage variable encodes how many aircraft are assigned to each route, while the first-stage constraints bound each type's possible number of aircraft. On the other hand, the second-stage variables denote the number of transported passengers on every five routes, and the random demands represent the second-stage constraints. This model has 646,425 possible scenarios. For the detailed mathematical model, see [6].

We have run the algorithm on the problem of different sizes. We have chosen to present results obtained for the moderate size of 1000 scenarios as the other sizes show similar results. Figure 5 indicates that the algorithm converges in less than 260 iterations. The optimal value for this problem is 1649.9, which agrees with the result in Table 2.

6.5 Numerical Comparisons and Discussion

This section provides a comparison of the Algorithm 1 with the Progressive Hedging (PH) algorithm with respect to CPU time needed for convergence, the number of iterations, percentage gap, and whether the method is converged or not for each problem. The results of this comparison are presented in Table 3 and Figure 6.

To assess the performance of ADMM and PH algorithms in terms of computational time, we have run both algorithms using the same programming language (MATLAB) in one device. A direct comparison of the running time of the two methods is unfair due to the different meaning of the stopping criteria used by the algorithms. For this reason, we used the following methodology: the two algorithms stop

when the error (a combination of primal and dual residuals) is less than convergence tolerance $\varepsilon = 10^{-3}$ or when they reach the maximum number of iterations.

In Table 3, columns 2-5 present the results of two algorithms. Columns 2 and 3 contain the CPU times needed for convergence; columns 4 and 5 present the number of iterations required. The same number of maximum iterations was used in both algorithms. The symbols "N" and "C" in columns 6 and 7 denote non-convergence and convergence case, respectively. The percentage gaps, columns 8 and 9, are calculated using the formula $\left|\frac{\phi - z^*}{z^*}\right| \times 100$, where ϕ is the computed optimal value and z^* the known optimal.

Table 3 shows that ADMM converged in all the problems while PH failed to converge in two problems; ADMM gives excellent approximations to the optimal values. The percentage gap infers that the convergence of ADMM is superior to that of PH in all the problems.

Table 3: Result summary of the comparison between PH and ADMM methods

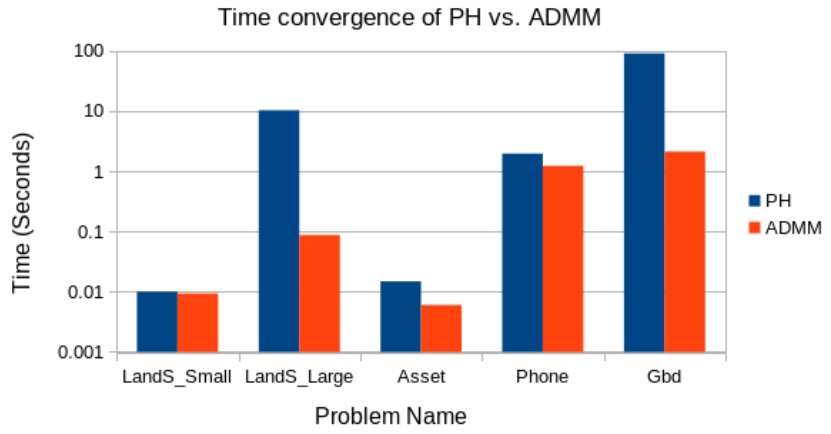
Problem	CPU Times		# of Iterations		Termination		Percentage gap	
	PH	ADMM	PH	ADMM	PH	ADMM	PH	ADMM
LandS-Small	0.01	0.00931	23	90	C	C	0.168%	0.047%
LandS-Large	10.1989	0.087	5000	309	N	C	0.177%	0.13%
Asset	0.0148	0.006	35	37	C	C	0.12%	0.09%
Phone	1.95	1.22	13	106	C	C	0.29%	0.001%
Gbd	89.56	2.11	5000	78	N	C	0.27%	0.52%

The graphs in Figure 6 are provided on a logarithmic scale because of the variations of results of different problems. Figure 6a presents the time that algorithms require to stop. We observe that ADMM converges faster than PH in four problems, while they are equivalent in the CPU time regarding the first one. Figure 6b compares the number of iterations showings that ADMM outperforms PH on two problems. On the other hand, PH performs better on two problems. Both methods are comparable in the number of iteration on the Asset problem.

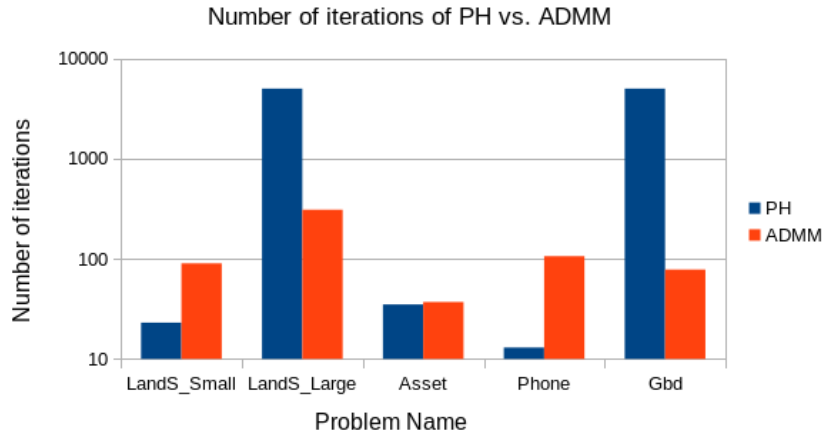
7 Conclusion

We have established the theoretical convergence of the iterative ADMM to the optimal solution of the two-stage stochastic programming problem. We have then implemented the method to three block formulation of the problem where its decomposable structure has been fully exploited in solving large-size problems. Numerical investigations of the algorithm using five benchmark problems have justified the theoretical convergence of the proposed algorithm.

The penalty parameter plays an essential role in the speed of the convergence of ADMM. We have shown that the adaptive penalty generates an accurate solution and always leads to convergence.



(a) Time for convergence



(b) Number of iterations

Fig. 6: Comparison of PH with ADMM using 5 problems from Table 1.

We have compared the performance of ADMM with the Progressive Hedging algorithm using the probability of success in obtaining the optimal solution, the accuracy of the solution obtained, and the CPU times. The comparison showed that ADMM outperforms PH in the probability of success and CPU times.

Our future research will involve the application of ADMM to multi-stage stochastic programming problems.

Acknowledgements This research was partially supported by African Institute for Mathematical Sciences (AIMS), South Africa.

References

1. KA Ariyawansa and Andrew J Felt. On a New Collection of Stochastic Linear Programming Test Problems. *INFORMS Journal on Computing*, 16(3):291–299, 2004.
2. Sebastián Arpón, Tito Homem-de Mello, and Bernardo K Pagnoncelli. An ADMM Algorithm for Two-stage Stochastic Programming Problems. *Annals of Operations Research*, 286(1):559–582, 2020.
3. Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal Alternating Minimization and Projection Methods for Nonconvex Problems: An Approach Based on The Kurdyka-Łojasiewicz Inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.
4. John R Birge and Francois Louveaux. *Introduction to Stochastic Programming*. Springer Science & Business Media, 2011.
5. Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed Optimization and Statistical Learning via The Alternating Direction Method of Multipliers*. Now Publishers Inc, 2011.
6. George B Dantzig and Mukund N Thapa. *Linear Programming 2: Theory and Extensions*. Springer Science & Business Media, 2006.
7. Andy Felt. Test-Problem Collection for Stochastic Linear Programming. `https://www4.uwsp.edu/math/afelt/slptestset/download.html`, 2003. [Online; accessed 14-October-2020].
8. Daniel Gabay and Bertrand Mercier. A Dual Algorithm for the Solution of Nonlinear Variational Problems via Finite Element Approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.
9. Roland Glowinski and A Marroco. Sur L'approximation, Par Éléments Finis D'ordre Un, Et La Résolution, Par Pénalisation-dualité D'une Classe de Problèmes de Dirichlet Non Linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 9(R2):41–76, 1975.
10. Jacek Gondzio, Pablo González-Brevis, and Pedro Munari. Large-scale Optimization with The Primal-dual Column Generation Method, 2016.
11. BS He, Hai Yang, and SL Wang. Alternating Direction Method with Self-Adaptive Penalty Parameters for Monotone Variational Inequalities. *Journal of Optimization Theory and Applications*, 106(2):337–356, 2000.
12. Jeff Linderoth, Alexander Shapiro, and Stephen Wright. The Empirical Behavior of Sampling Methods for Stochastic Programming. *Annals of Operations Research*, 142(1):215–241, 2006.
13. Jing Liu, Yongrui Duan, and Min Sun. A symmetric Version of the Generalized Alternating Direction Method of Multipliers for Two-block Separable Convex Programming. *Journal of Inequalities and Applications*, 2017(1):1–21, 2017.
14. François V Louveaux and Yves Smeers. Optimal Investments for Electricity Generateion: A Stochastic Model and A Test Problem. *Numerical Techniques for Stochastic Optimization*, 1988.
15. R Tyrrell Rockafellar. *Convex Analysis*. Number 28. Princeton university press, 1970.
16. R Tyrrell Rockafellar. Monotone Operators and the Proximal Point Algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.
17. R Tyrrell Rockafellar and Roger J-B Wets. Scenarios and Policy Aggregation in Optimization Under Uncertainty. *Mathematics of Operations Research*, 16(1):119–147, 1991.
18. Sarah M Ryan, Roger J-B Wets, David L Woodruff, César Silva-Monroy, and Jean-Paul Watson. Toward Scalable, Parallel Progressive Hedging for Stochastic Unit Commitment. In *2013 IEEE Power & Energy Society General Meeting*, pages 1–5. IEEE, 2013.
19. Defeng Sun, Kim-Chuan Toh, and Liuqin Yang. A Convergent 3-Block Semiproximal Alternating Direction Method of Multipliers for Conic Programming with 4-Type Constraints. *SIAM journal on Optimization*, 25(2):882–915, 2015.
20. Fenghui Wang, Wenfei Cao, and Zongben Xu. Convergence of Multi-block Bregman ADMM for Nonconvex Composite Problems. *Science China Information Sciences*, 61(12):122101, 2018.
21. J. Wang, Z. Chai, Y. Cheng, and L. Zhao. Toward Model Parallelism for Deep Neural Network Based on Gradient-Free ADMM Framework. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 591–600, 2020.
22. Z. Wang, B. Hall, J. Xu, and X. Shi. A Sparse Learning Framework for Joint Effect Analysis of Copy Number Variants. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(5):1013–1027, 2017.
23. Zheng Xu, Mario Figueiredo, and Tom Goldstein. Adaptive ADMM with Spectral Penalty Parameter Selection. In *Artificial Intelligence and Statistics*, pages 718–727. PMLR, 2017.