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

An augmented Lagrangian Method for Distributed Optimization

Nikolaos Chatzipanagiotis · Darinka Dentcheva · Michael M. Zavlanos

Received: date / Accepted: date

Abstract We propose a novel distributed method for convex optimization problems with a certain separability structure. The method is based on the augmented Lagrangian framework. We analyze its convergence and provide an application to two network models, as well as to a two-stage stochastic optimization problem. The proposed method compares favorably to two augmented Lagrangian decomposition methods known in the literature, as well as to decomposition methods based on the ordinary Lagrangian function.

Keywords convex optimization · monotropic programming · alternating direction method · network optimization · diagonal quadratic approximation · stochastic programming

1 Introduction

Let $\mathscr{X}_i \subseteq \mathbb{R}^{n_i}$, $i \in \mathscr{I} = \{1,2,\dots,N\}$ be nonempty closed, convex subsets of n_i-dimensional Euclidean space respectively, and $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$, $i \in \mathscr{I}$ be convex functions. Furthermore let $\mathbf{A}_i$ be $m \times n_i$ matrices, $i = 1,2,\dots,N$. The focus of our investigation is the following convex optimization problem

$$\begin{aligned} \min \ & \sum_{i=1}^{N} f_i(\mathbf{x}_i) \\ \text{subject to } & \sum_{i=1}^{N} \mathbf{A}_i \mathbf{x}_i = \mathbf{b}, \\ & \mathbf{x}_i \in \mathscr{X}_i, \quad i = 1,2,\dots,N. \end{aligned} \tag{1}$$

This work was supported by the NSF awards DMS #1311978, CNS #1261828 and CNS #1302284.

Nikolaos Chatzipanagiotis
Dept. of Mechanical Engineering and Materials Science, Duke University, Durham, NC, 27708, USA
E-mail: n.chatzip@duke.edu

Darinka Dentcheva
Dept. of Mathematics, Stevens Institute of Technology, Hoboken, NJ 07030, USA
E-mail: darinka.dentcheva@stevens.edu

Michael M. Zavlanos
Dept. of Mechanical Engineering and Materials Science, Duke University, Durham, NC, 27708, USA
E-mail: michael.zavlanos@duke.edu

Problems of the form (1) are called *extended monotropic optimization problems* in [3]. The term monotropic optimization was introduced for the special case of $n_i = 1$, $i = 1, 2, \ldots, N$ in [19, 20], where problems of this type were analysed. In this paper, we propose a new decomposition method for solving problem (1) in a distributed fashion, which we call the *Accelerated Distributed Augmented Lagrangian* method.

The ever increasing size and complexity of modern day problems, coupled with the ongoing advancements in massively parallel processing capabilities of contemporary computers, have motivated recent advances in developing efficient, distributed computing methods. Distributed algorithms decompose the original problem into smaller, more manageable subproblems that are solved iteratively, either in a parallel or in a sequential fashion. Furthermore, in certain problems arising in communication networks, sensor networks, networked robotics, and other areas, it is desirable that the method relies only on local information exchanges between the decomposed subproblems during the iterative solution procedure, without the need to maintain a global "supervising" and coordinating unit.

Many decomposition methods rely on the decomposable structure of the dual function. We refer the interested reader to [4] for an overview. However, such dual methods suffer from well-documented disadvantages, such as slow convergence rates, and, also, non-uniqueness of solutions, which necessitates the application of advanced techniques of non-smooth optimization in order to ensure numerical stability and efficiency of the procedure. These drawbacks are alleviated by the application of regularization techniques such as bundle methods and by the augmented Lagrangian framework, which is akin to the regularization of the dual function. A significantly smaller number of works is available on decomposition of augmented Lagrangians. The convergence speed and the numerical advantages of augmented Lagrangian methods (see, e.g., [13, 16, 17, 21, 22]) provide a strong motivation for creating decomposed versions of them. Early specialized techniques that allow for decomposition of the augmented Lagrangian can be traced back to the works [6, 11, 12, 30–32]. More recent literature involves the *Diagonal Quadratic Approximmation* (DQA) algorithm [2, 15, 24] and the *Alternating Direction Method of Multipliers* (ADMM) [4, 5, 7, 9, 10]. The DQA method replaces each minimization step in the augmented Lagrangian algorithm by a separable approximation of the augmented Lagrangian function. The ADMM methods are based on the relations between splitting methods for monotone operators, such as Douglas-Rachford splitting, and the proximal point algorithm [10, 11]. In [7, 9], the authors develop the *Alternating Step Method* (ASM), which is a specialized, equivalent version of ADMM when applied on problems of the form (1). Note that the analysis in [9] is limited to cases where $n_i = 1$ for all i, however, the extension for $n_i > 1$ is straightforward.

Our paper is organized as follows. In section 2, we introduce the necessary notions and notation. We also provide a description of the DQA and ASM and recall their convergence properties. In section 3, we describe the proposed method ADAL and contrast it to the ASM and DQA methods. In section 4, we analyze the convergence of ADAL. Section 5 contains numerical results for a network utility maximization problem, a network flow problem, and a two-stage stochastic capacity expansion problem.

2 Preliminaries

We denote

$$f(\mathbf{x}) = \sum_{i=1}^{N} f_i(\mathbf{x}_i),$$

where $\mathbf{x} = [\mathbf{x}_1^\top, \dots, \mathbf{x}_N^\top]^\top \in \mathbb{R}^n$ with $n = \sum_{i=1}^N n_i$. Furthermore, we denote $\mathbf{A} = [\mathbf{A}_1 \dots \mathbf{A}_N] \in \mathbb{R}^{m\times n}$. The constraint $\sum\limits_{i=1}^{N} \mathbf{A}_i\mathbf{x}_i = \mathbf{b}$ of problem (1) takes on the form $\mathbf{Ax} = \mathbf{b}$. We associate Lagrange multipliers $\lambda \in \mathbb{R}^m$ with that constraint. The Lagrange function is defined as

$$L(\mathbf{x},\lambda) = f(\mathbf{x}) + \langle \lambda, \mathbf{Ax} - \mathbf{b}\rangle = \sum_{i=1}^{N} L_i(\mathbf{x}_i,\lambda) - \langle \mathbf{b},\lambda\rangle,$$

where

$$L_i(\mathbf{x}_i,\lambda) = f_i(\mathbf{x}_i) + \langle \lambda, \mathbf{A}_i\mathbf{x}_i\rangle.$$

The dual function has the form

$$g(\lambda) = \inf_{\mathbf{x}\in\mathscr{X}} L(\mathbf{x},\lambda) = \sum_{i=1}^{N} g_i(\lambda) - \langle \mathbf{b},\lambda\rangle,$$

where $\mathscr{X} = \mathscr{X}_1 \times \mathscr{X}_2 \cdots \times \mathscr{X}_N$ and

$$g_i(\lambda) = \inf_{\mathbf{x}_i\in\mathscr{X}_i} \Big[f_i(\mathbf{x}_i) + \langle \lambda, \mathbf{A}_i\mathbf{x}_i\rangle \Big].$$

The dual function is decomposable and this gives rise to various decomposition methods addressing the *dual problem*, which is given by

$$\max_{\lambda\in\mathbb{R}^m} \sum_{i=1}^{N} g_i(\lambda) - \langle \mathbf{b},\lambda\rangle. \tag{2}$$

The augmented Lagrangian associated with problem (1) has the form:

$$\Lambda_\rho(\mathbf{x},\lambda) \;=\; f(\mathbf{x}) \;+\; \langle \lambda, \mathbf{Ax}-\mathbf{b}\rangle \;+\; \frac{\rho}{2}\|\mathbf{Ax}-\mathbf{b}\|^2, \tag{3}$$

where $\rho > 0$ is a penalty parameter. We recall the standard augmented Lagrangian method (sometimes refered to as the "Method of Multipliers" in existing literature):

Augmented Lagrangian Method
Step 0. Set $k = 1$ and define initial Lagrange multipliers λ^1.
Step 1. For a fixed vector λ^k, calculate $\hat{\mathbf{x}}^k$ as a solution of the problem:

$$\min_{\mathbf{x}\in\mathscr{X}} \Lambda_\rho(\mathbf{x},\lambda^k). \tag{4}$$

Step 2. If the constraints $\sum\limits_{i=1}^{N} \mathbf{A}_i\hat{\mathbf{x}}_i^k = \mathbf{b}$ are satisfied, then stop (optimal solution found). Otherwise, set :

$$\lambda^{k+1} = \lambda^k + \rho\left(\sum_{i=1}^{N} \mathbf{A}_i\hat{\mathbf{x}}_i^k - \mathbf{b}\right), \tag{5}$$

Increase k by one and return to Step 1.

We refer to the standard augmented Lagrangian method (4)-(5) as the "centralized" augmented Lagrangian method in the rest of this paper. Similarly, we will use the term "centralized" to refer to solving a problem without using any decomposition approach.

We use $\mathscr{N}_{\mathscr{X}}(\mathbf{x})$ to denote the normal cone to the set $\mathscr{X}$ at the point $\mathbf{x}$ [26], i.e.,

$$\mathscr{N}_{\mathscr{X}}(\mathbf{x}) \;=\; \{\mathbf{h} \in \mathbb{R}^n : \langle \mathbf{h}, \mathbf{y} - \mathbf{x} \rangle \leq 0, \;\; \forall\, \mathbf{y} \in \mathscr{X}\}.$$

The convex subdifferential of a convex function f at a point $\mathbf{x}$ is denoted by $\partial f(\mathbf{x})$.

The convergence of the Augmented Lagrangian Method is ensured when problem (2) has an optimal solution independently of the starting point λ^1. Under convexity assumptions and a constrain qualification condition, every accumulation point of the sequence $\{\mathbf{x}^k\}$ is an optimal solution of problem (1). Furthermore, the augmented Lagrangian method exhibits convergence properties also in a non-convex setting assuming that the functions f_i, $i = 1, \dots N$ are twice continuously differentiable and the strong second-order conditions of optimality are satisfied. We refer to [18] for the analysis of the augmented Lagrangian method in the convex case and to [26] for the non-convex case.

A major drawback of the Augmented Lagrangian Method stems from the fact that (4) is not amenable to decomposition due to the quadratic penalty term in (3). This issue is addressed by creating successive separable approximations of the quadratic term in [2, 15, 24] and by using alternating linearization techniques in [14].

For a given monotropic problem (1), we define the *maximum degree* q as a measure of sparsity of the total constraint matrix $\mathbf{A}$. For each constraint $j = 1, \dots, m$, we introduce a measure of involvement. We denote the number of locations associated with this constraint by q_j, that is, q_j is the number of all $i \in \mathscr{I} : [\mathbf{A}_i]_j \neq \mathbf{0}$. Here, $[\mathbf{A}_i]_j$ denotes the j-th row of matrix $\mathbf{A}_i$ and $\mathbf{0}$ stands for a zero vector of proper dimension. We define q to be the maximum over all q_j, i.e.

$$q = \max_{1 \leq j \leq m} q_j. \tag{6}$$

It will be shown in Section 4 that q plays a critical role in the convergence properties of the proposed method.

In [24], the Diagonal Quadratic Approximation (DQA) method based on the augmented Lagrangian function is developed for problems of the form (1) and its convergence is analyzed. The method has found applications in stochastic programming, engineering, and finance. The idea of DQA is to produce a separable approximation of the primal step of the centralized augmented Lagrangian Method (4)-(5), which iteratively converges to the actual primal step of the Augmented Lagrangian Method. This is achieved by introducing an inner loop of minimization and correction steps. For $i = 1, \dots, N$, the *local augmented Lagrangian* function $\Lambda_\rho^i : \mathbb{R}^{n_i} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is defined according to

$$\Lambda_\rho^i(\mathbf{x}_i, \mathbf{x}^k, \lambda) \;=\; f_i(\mathbf{x}_i) \;+\; \langle \lambda, \mathbf{A}_i \mathbf{x}_i \rangle \;+\; \frac{\rho}{2} \|\mathbf{A}_i \mathbf{x}_i + \sum_{j \in \mathscr{I}}^{j \neq i} \mathbf{A}_j \mathbf{x}_j^k - \mathbf{b}\|^2. \tag{7}$$

The DQA method uses a parameter $\tau \in (0, 1)$, which is utilized as a stepsize in updating the primal variables. It works as follows.

Diagonal Quadratic Approximation (DQA)

Step 0. Set $k = 1$, $s = 1$ and define initial Lagrange multipliers λ^1 and initial primal variables $\mathbf{x}^{1,1}$.

Step 1. For fixed Lagrange multipliers λ^k and for every $i \in \mathscr{I}$, determine $\hat{\mathbf{x}}_i^{k,s}$ as the solution of:

$$\min_{\mathbf{x}_i \in \mathscr{X}_i} \Lambda_\rho^i(\mathbf{x}_i, \mathbf{x}^{k,s}, \lambda^k). \tag{8}$$

Step 2. For every $i \in \mathscr{I}$, if $\mathbf{A}_i\hat{\mathbf{x}}_i^{k,s} = \mathbf{A}_i\mathbf{x}_i^{k,s}$, then go to step 3; otherwise, set

$$\mathbf{x}_i^{k,s+1} = \mathbf{x}_i^{k,s} + \tau(\hat{\mathbf{x}}_i^{k,s} - \mathbf{x}_i^{k,s}), \tag{9}$$

increase s by 1 and go to Step 1.

Step 3. Set $\mathbf{x}^{k,s} = \mathbf{x}^{k+1}$. If the constraint $\sum_{i=1}^{N} \mathbf{A}_i\mathbf{x}_i^{k+1} = \mathbf{b}$ is satisfied, then stop (optimal solution found). Otherwise, set :

$$\lambda^{k+1} = \lambda^k + \rho\left(\sum_{i=1}^{N} \mathbf{A}_i\mathbf{x}_i^{k+1} - \mathbf{b}\right), \tag{10}$$

and $s = 1$, $\mathbf{x}^{k+1,1} = \mathbf{x}^{k+1}$, increase k by one and return to Step 1.

Convergence of the DQA method is guaranteed if the stepsize τ satisfies $0 < \tau < \frac{1}{q}$, where q is defined in (6). The inner-loop termination criterion in Step 2 of DQA requires that $\mathbf{A}_i\hat{\mathbf{x}}_i^{k,s} = \mathbf{A}_i\mathbf{x}_i^{k,s}$ for every $i \in \mathscr{I}$, which in practice is achieved within a given numerical accuracy ε. This entails that the augmented Lagrangian is calculated with an error bounded by $\frac{1}{\rho}\varepsilon q$ (see [24, Lemma 1]).

Another dual decomposition approach is that of alternating directions. Splitting methods such as the Alternating Directions Method of Multipliers (ADMM) could also be considered a form of augmented Lagrangian algorithm, although their convergence mechanism is quite different. In [7, 9], the authors apply the generalized ADMM on monotropic optimization problems and derive a simplified algorithmic form of ADMM, which they call the Alternating Step Method (ASM). We refer to ASM in our presentation instead of generically refering to ADMM because it is most similar to ADAL.

The ASM algorithm uses the following form of local augmented Lagrangian $\bar{\Lambda}_\rho^i(\mathbf{x}_i, \mathbf{x}^k, \lambda^k)$

$$\bar{\Lambda}_\rho^i(\mathbf{x}_i, \mathbf{x}^k, \lambda) = f_i(\mathbf{x}_i) + \langle \lambda, \mathbf{A}_i\mathbf{x}_i \rangle + \frac{\rho}{2}\sum_{l=1}^{m}\left([\mathbf{A}_i]_l\mathbf{x}_i - [\mathbf{A}_i]_l\mathbf{x}_i^k + \frac{1}{q_l}\left(\sum_{j\in\mathscr{I}}[\mathbf{A}_j]_l\mathbf{x}_j^k - b_l\right)\right)^2,$$

where q_l denotes the degree of constraint l, as per the aforementioned definition. The ASM method works as follows.

Alternating Step Method (ASM)

Step 0. Set $k = 1$ and define initial Lagrange multipliers λ^1 and initial primal variables $\mathbf{x}^1$.

Step 1. For fixed Lagrange multipliers λ^k and for every $i = 1, \dots, N$, determine $\hat{\mathbf{x}}_i^k$ as the solution of the following problem:

$$\min_{\mathbf{x}_i \in \mathscr{X}_i} \bar{\Lambda}_\rho^i(\mathbf{x}_i, \mathbf{x}^k, \lambda^k) \tag{11}$$

Step 2. For every $i \in \mathscr{I}$, set

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \sigma(\hat{\mathbf{x}}_i^k - \mathbf{x}_i^k), \tag{12}$$

where σ is a non-negative stepsize satisfying $\sigma \in (0,2)$.

Step 3. If the constraint $\sum_{i=1}^{N} \mathbf{A}_i \hat{\mathbf{x}}_i^k = \mathbf{b}$ is satisfied, then stop (optimal solution found). Otherwise, set :

$$\lambda_j^{k+1} = \lambda_j^k + \frac{\rho\sigma}{q_j} \left(\sum_{i=1}^{N} [\mathbf{A}_i \hat{\mathbf{x}}_i^k]_j - \mathbf{b}_j \right), \tag{13}$$

increase k by one and return to Step 1.

The ASM method allows for adapted stepsizes in Step 3, containing the degrees of each constraint. Furthermore, some terms involving q_j have found their way into the quadratic penalty term. The method introduces the relaxation factor $\sigma \in (0,2)$ from the theory of the generalized ADMM [8, 10] and utilizes it as a stepsize for the primal update. Note that, for $\sigma = 1$, we obtain the classical ADMM [8, 10]. We refer to [5, 8, 10] for a discussion on the general properties of the ADMM and its applications.

3 The accelerated Distributed Augmented Lagrangian Method

In this paper, we propose a new Augmented Lagrangian decomposition method, which we call *Accelerated Distributed Augmented Lagrangian* (ADAL). The method uses the same local Lagrangian approximation of DQA, but eliminates the inner loop of the DQA procedure. Nonetheless, the method enjoys convergence to the optimal solution of (1).

The ADAL has two parameters: a positive penalty parameter ρ and a stepsize parameter $\tau \in (0,1)$. Each iteration of ADAL is comprised of three steps: i) a minimization step of all the local augmented Lagrangians, ii) an update step for the primal variables, and iii) an update step for the dual variables. The computations at each step are performed in a parallel fashion, so that ADAL resembles a *Jacobi*-type algorithm; see [4] for more details on Jacobi and Gauss-Seidel type algorithms. ADAL works as follows.

Accelerated Distributed Augmented Lagrangian (ADAL)

Step 0. Set $k = 1$ and define initial Lagrange multipliers λ^1 and initial primal variables $\mathbf{x}^1$.

Step 1. For fixed Lagrange multipliers λ^k, determine $\hat{\mathbf{x}}_i^k$ for every $i \in \mathscr{I}$ as the solution of the following problem:

$$\min_{\mathbf{x}_i \in \mathscr{X}_i} \Lambda_\rho^i(\mathbf{x}_i, \mathbf{x}^k, \lambda^k). \tag{14}$$

Step 2. Set for every $i \in \mathscr{I}$

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \tau(\hat{\mathbf{x}}_i^k - \mathbf{x}_i^k). \tag{15}$$

Step 3. If the constraints $\sum_{i=1}^{N} \mathbf{A}_i \mathbf{x}_i^{k+1} = \mathbf{b}$ are satisfied and $\mathbf{A}_i \hat{\mathbf{x}}_i^k = \mathbf{A}_i \mathbf{x}_i^k$, then stop (optimal solution found). Otherwise, set:

$$\lambda^{k+1} = \lambda^k + \rho\tau \left(\sum_{i=1}^{N} \mathbf{A}_i \mathbf{x}_i^{k+1} - \mathbf{b} \right), \tag{16}$$

increase k by one and return to Step 1.

After the local calculations (14) have been performed, the primal variables $\mathbf{x}_i^k$ are updated in Step 2. In Section 4, we show that the method converges for a stepsize $\tau \in (0, \frac{1}{q})$.

A critical point for the convergence of our method is the choice of the $\mathbf{x}^{k+1}$ variables for performing the dual update (16), instead of the minimizers $\hat{\mathbf{x}}^k$ of the local augmented Lagrangians (14). In the DQA method, the terms $\mathbf{A}\hat{\mathbf{x}}^k$ are equal to $\mathbf{A}\mathbf{x}^{k+1}$ in (10) due to the loop between Step 1 and Step 2 of DQA, which is eliminated in the ADAL algorithm. The dual update in the centralized augmented Lagrangian Method (5) directly uses the minimizers of the augmented Lagrangian. Similarly, the ADMM methods (and, consequently, ASM) [5, 7–10] use $\hat{\mathbf{x}}^k$ as well.

In the rest of this section, we briefly discuss what information is needed to perform each step of ADAL at a given iteration k. Obviously, the primal update step (15) is local to each subproblem i and does not require any message exchanges to be performed, so we limit our discussion to steps 1 and 3. The particular information exchange patterns are relevant when problems need to be solved in the absence of a central processing unit that has access to and coordinates all the information generated by the problem. Such situations appear, for example, when solving optimization problems over networked systems, where every node of the network is a processing unit that can only access its own local information as well as information that is available from its one-hop neighbors.

According to (16), the update law for the dual variable of the j-th constraint is

$$\lambda_j^{k+1} = \lambda_j^k + \rho\tau \left(\sum_{i=1}^{N} \left[\mathbf{A}_i \mathbf{x}_i^{k+1}\right]_j - b_j \right).$$

This implies that the udpate of λ_j needs only information from those i for which $[\mathbf{A}_i]_j \neq \mathbf{0}$. Furthermore, recall that

$$\Lambda_\rho^i(\mathbf{x}_i, \mathbf{x}^k, \lambda) \;=\; f_i(\mathbf{x}_i) \;+\; \langle \lambda, \mathbf{A}_i \mathbf{x}_i \rangle \;+\; \frac{\rho}{2} \|\mathbf{A}_i \mathbf{x}_i + \sum_{j\in\mathscr{I}}^{j\neq i} \mathbf{A}_j \mathbf{x}_j^k - \mathbf{b}\|^2.$$

Since $\langle \lambda, \mathbf{A}_i \mathbf{x}_i \rangle \;=\; \sum_{j=1}^m \lambda_j [\mathbf{A}_i \mathbf{x}_i]_j$, we see that, in order to compute (14), each subproblem i needs access only to those λ_j for which $[\mathbf{A}_i]_j \neq \mathbf{0}$. Moreover, the penalty term of each Λ_ρ^i can be equivalently expressed as

$$\|\mathbf{A}_i \mathbf{x}_i + \sum_{j\in\mathscr{I}}^{j\neq i} \mathbf{A}_j \mathbf{x}_j^k - \mathbf{b}\|^2 = \sum_{l=1}^{m} \left(\left[\mathbf{A}_i \mathbf{x}_i\right]_l + \sum_{j\in\mathscr{I}}^{j\neq i} \left[\mathbf{A}_j \mathbf{x}_j^k\right]_l - b_l \right)^2.$$

The above penalty term is involved only in the minimization computation (14). Hence, for those l such that $[\mathbf{A}_i]_l = \mathbf{0}$, the terms $\sum_{j\in\mathscr{I}}^{j\neq i} \left[\mathbf{A}_j \mathbf{x}_j^k\right]_l - b_l$ are just constant terms in the minimization step, and can be excluded. This implies that subproblem i needs access only to the decisions $\left[\mathbf{A}_j \mathbf{x}_j^k\right]_l$ from all subproblems $j \neq i$ that are involved in the same constraints l as i.

4 Convergence.

In order to prove convergence of ADAL, we need the following two assumptions:

(A1) The functions $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$, $i \in \mathscr{I} = \{1,2,\dots,N\}$ are convex and $\mathscr{X}_i \subseteq \mathbb{R}^{n_i}$, $i = 1,\dots,N$ are nonempty closed convex sets.

(A2) The Lagrange function L has a saddle point $(\mathbf{x}^*,\lambda^*) \in \mathbb{R}^n \times \mathbb{R}^m$:

$$L(\mathbf{x}^*,\lambda) \ \le\ L(\mathbf{x}^*,\lambda^*) \ \le\ L(\mathbf{x},\lambda^*) \quad \forall\, \mathbf{x} \in \mathscr{X}, \forall\, \lambda \in \mathbb{R}^m. \tag{17}$$

(A3) All subproblems (14) are solvable at any iteration $k \in \mathbb{N}$.

Assumption (A2) implies that the point $\mathbf{x}^*$ is a solution of problem (1), the point λ^* is a solution of (2) and the strong duality relation holds, i.e., the optimal values of both problems are equal.

Assumption (A3) is satisfied if for every $i = 1,\dots,N$, either the set $\mathscr{X}_i$ is compact, or the function $f_i(\mathbf{x}_i) + \frac{\rho}{2}\|\mathbf{A}_i\mathbf{x}_i - \tilde{\mathbf{b}}\|^2$ is inf-compact for any vector $\tilde{\mathbf{b}}$. The latter condition, means that the level sets of the function are compact sets, implying that set $\{x \in \mathscr{X}_i : f_i(\mathbf{x}_i) + \frac{\rho}{2}\|\mathbf{A}_i\mathbf{x}_i - \tilde{\mathbf{b}}\|^2 \le \alpha\}$ is compact for any $\alpha \in \mathbb{R}$.

Define the *residual* $\mathbf{r}(\mathbf{x}) \in \mathbb{R}^m$ as the vector containing the amount of all constraint violations with respect to primal variable $\mathbf{x}$, i.e. $\mathbf{r}(\mathbf{x}) = \sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i - \mathbf{b}$.

To avoid cluttering the notation, we will use the simplified notation $\sum_i$ to denote summation over all $i \in \mathscr{I}$, i.e. $\sum_i = \sum_{i=1}^N$, unless explicitly noted otherwise. Also, we define the auxiliary variables:

$$\hat{\lambda}^k = \lambda^k + \rho\mathbf{r}(\hat{\mathbf{x}}^k), \tag{18}$$

available at iteration k. Note that this happens to be the dual update rule in the centralized augmented Lagrangian Method.

The basic idea of the proof is to introduce the Lyapunov (merit) function

$$\phi(\mathbf{x}^k,\lambda^k) \ =\ \sum_{i=1}^N \rho\|\mathbf{A}_i(\mathbf{x}_i^k - \mathbf{x}_i^*)\|^2 \ +\ \frac{1}{\rho}\|\lambda^k + \rho(1-\tau)\mathbf{r}(\mathbf{x}^k) - \lambda^*\|^2. \tag{19}$$

We will show in Theorem 1 that this merit function is strictly decreasing during the execution of the ADAL algorithm (14)-(16), given that the stepsize τ satisfies the condition $0 < \tau < 1/q$. Then, in Theorem 2 we argue that this strict decrease property implies convergence of the primal and dual variables to their respective optimal values.

We begin the proof by utilizing the first order optimality conditions of all the subproblems (14) in order to derive some necessary inequalities.

Lemma 1 *Assume (A1)–(A3). The following inequality holds:*

$$\frac{1}{\rho}(\hat{\lambda}^k - \lambda^*)^\top(\lambda^k - \hat{\lambda}^k) \ \ge\ \rho\sum_i \left[(\mathbf{A}_i\hat{\mathbf{x}}_i^k - \mathbf{A}_i\mathbf{x}_i^*)^\top \sum_{j\neq i}\mathbf{A}_j(\mathbf{x}_j^k - \hat{\mathbf{x}}_j^k)\right] \tag{20}$$

where $(\mathbf{x}^*,\lambda^*)$ *is a saddle point of the Lagrangian L and* λ^k, $\hat{\lambda}^k$, $\hat{\mathbf{x}}_i^k$, *and* $\mathbf{x}_j^k$ *are calculated at iteration k.*

Proof The first order optimality conditions for problem (14) imply the following inclusion for the minimizer $\hat{\mathbf{x}}_i^k$

$$0 \in \partial f_i(\hat{\mathbf{x}}_i^k) + \mathbf{A}_i^\top\lambda^k + \rho\mathbf{A}_i^\top\Big(\mathbf{A}_i\hat{\mathbf{x}}_i^k + \sum_{j\neq i}\mathbf{A}_j\mathbf{x}_j^k - \mathbf{b}\Big) + \mathscr{N}_{\mathscr{X}_i}(\hat{\mathbf{x}}_i^k) \tag{21}$$

We infer that subgradients $\mathbf{s}_i^k \in \partial f_i(\hat{\mathbf{x}}_i^k)$ and normal elements $\mathbf{z}_i^k \in \mathcal{N}_{\mathscr{X}_i}(\hat{\mathbf{x}}_i^k)$ exist such that we can express (21) as follows:

$$0 \;=\; \mathbf{s}_i^k + \mathbf{A}_i^\top \lambda^k + \rho \mathbf{A}_i^\top \Big(\mathbf{A}_i\hat{\mathbf{x}}_i^k + \sum_{j\neq i}\mathbf{A}_j\mathbf{x}_j^k - \mathbf{b}\Big) + \mathbf{z}_i^k. \tag{22}$$

Taking inner product with $\mathbf{x}_i^* - \hat{\mathbf{x}}_i^k$ on both sides of this equation and using the definition of a normal cone, we obtain

$$\langle \mathbf{s}_i^k + \mathbf{A}_i^\top \lambda^k + \rho \mathbf{A}_i^\top \Big(\mathbf{A}_i\hat{\mathbf{x}}_i^k + \sum_{j\neq i}\mathbf{A}_j\mathbf{x}_j^k - \mathbf{b}\Big), \mathbf{x}_i^* - \hat{\mathbf{x}}_i^k\rangle \;=\; \langle -\mathbf{z}_i^k, \mathbf{x}_i^* - \hat{\mathbf{x}}_i^k\rangle \;\geq\; 0. \tag{23}$$

Using the variables $\hat{\lambda}^k$ defined in (18), we substitute λ^k in (23) and obtain:

$$\begin{aligned} 0 \;\leq\; & \langle \mathbf{s}_i^k + \mathbf{A}_i^\top \left[\hat{\lambda}^k - \rho\Big(\sum_j \mathbf{A}_j\hat{\mathbf{x}}_j^k - \mathbf{b}\Big) + \rho\Big(\mathbf{A}_i\hat{\mathbf{x}}_i^k + \sum_{j\neq i}\mathbf{A}_j\mathbf{x}_j^k - \mathbf{b}\Big)\right], \mathbf{x}_i^* - \hat{\mathbf{x}}_i^k\rangle \\ =\; & \langle \mathbf{s}_i^k + \mathbf{A}_i^\top \left[\hat{\lambda}^k + \rho\Big(\sum_{j\neq i}\mathbf{A}_j\mathbf{x}_j^k - \sum_{j\neq i}\mathbf{A}_j\hat{\mathbf{x}}_j^k\Big)\right], \mathbf{x}_i^* - \hat{\mathbf{x}}_i^k\rangle \end{aligned} \tag{24}$$

The assumptions (A1) and (A2) entail that the following optimality conditions are satisfied at the point $(\mathbf{x}^*, \lambda^*)$:

$$0 \in \partial f_i(\mathbf{x}_i^*) + \mathbf{A}_i^\top \lambda^* + \mathcal{N}_{\mathscr{X}_i}(\mathbf{x}_i^*) \quad \text{for all } i = 1,\dots,N. \tag{25}$$

Inclusion (25) implies that subgradients $\mathbf{s}_i^* \in \partial f_i(\mathbf{x}_i^*)$ and normal vectors $\mathbf{z}_i^* \in \mathcal{N}_{\mathscr{X}_i}(\mathbf{x}_i^*)$ exist, such that we can express (25) as:

$$0 \;=\; \mathbf{s}_i^* + \mathbf{A}_i^\top \lambda^* + \mathbf{z}_i^*$$

Taking inner product with $\hat{\mathbf{x}}_i^k - \mathbf{x}_i^*$ on both sides of this equation and using the definition of a normal cone, we infer

$$\langle \mathbf{s}_i^* + \mathbf{A}_i^\top \lambda^*, \hat{\mathbf{x}}_i^k - \mathbf{x}_i^*\rangle \;\geq\; 0, \quad \text{for all } i = 1,\dots,N. \tag{26}$$

Combining (24) and (26), we obtain the following inequalities for all $i = 1,\dots,N$:

$$(\hat{\mathbf{x}}_i^k - \mathbf{x}_i^*)^\top \Big(\mathbf{s}_i^* - \mathbf{s}_i^k + \mathbf{A}_i^\top(\lambda^* - \hat{\lambda}^k) - \rho\mathbf{A}_i^\top\big[\sum_{j\neq i}\mathbf{A}_j\mathbf{x}_j^k - \sum_{j\neq i}\mathbf{A}_j\hat{\mathbf{x}}_j^k\big]\Big) \;\geq\; 0. \tag{27}$$

Using the monotonicity of the subdifferential mapping, we take out the terms involving the subgradients $\big(\hat{\mathbf{x}}_i^k - \mathbf{x}_i^*\big)^\top\big(\mathbf{s}_i^* - \mathbf{s}_i^k\big) \leq 0$ and arrive at:

$$(\hat{\mathbf{x}}_i^k - \mathbf{x}_i^*)^\top \Big[\mathbf{A}_i^\top(\lambda^* - \hat{\lambda}^k) - \rho\mathbf{A}_i^\top\big(\sum_{j\neq i}\mathbf{A}_j\mathbf{x}_j^k - \sum_{j\neq i}\mathbf{A}_j\hat{\mathbf{x}}_j^k\big)\Big] \;\geq\; 0 \quad \forall i = 1,\dots,N. \tag{28}$$

Adding the inequalities for all $i = 1,\dots,N$ and rearranging terms, we get:

$$(\lambda^* - \hat{\lambda}^k)^\top\Big[\sum_i \mathbf{A}_i(\hat{\mathbf{x}}_i^k - \mathbf{x}_i^*)\Big] \;\geq\; \rho\sum_i\Big[(\mathbf{A}_i\hat{\mathbf{x}}_i^k - \mathbf{A}_i\mathbf{x}_i^*)^\top\sum_{j\neq i}(\mathbf{A}_j\mathbf{x}_j^k - \mathbf{A}_j\hat{\mathbf{x}}_j^k)\Big] \tag{29}$$

Substituting $\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i^* = \mathbf{b}$ and $\sum_{i=1}^N \mathbf{A}_i\hat{\mathbf{x}}_i^k - \mathbf{b} = \frac{1}{\rho}(\hat{\lambda}^k - \lambda^k)$ in (29), we conclude that

$$\frac{1}{\rho}(\hat{\lambda}^k - \lambda^*)^\top(\lambda^k - \hat{\lambda}^k) \;\geq\; \rho\sum_i\Big[(\mathbf{A}_i\hat{\mathbf{x}}_i^k - \mathbf{A}_i\mathbf{x}_i^*)^\top\sum_{j\neq i}(\mathbf{A}_j\mathbf{x}_j^k - \mathbf{A}_j\hat{\mathbf{x}}_j^k)\Big]$$

as required. $\square$

In the following two lemmata, we exploit the result from Lemma 1 and perform some necessary manipulations that will allow us to prove the strict decrease property of the Lyapunov (merit) function in Theorem 1 later on.

Lemma 2 *Under assumptions (A1)–(A3), the following estimate holds:*

$$\begin{aligned}\rho\sum_i\Big[(\mathbf{A}_i\mathbf{x}_i{}^k-\mathbf{A}_i\mathbf{x}_i^*)^\top(\mathbf{A}_i\mathbf{x}_i^k-\mathbf{A}_i\hat{\mathbf{x}}_i^k)\Big]\;+\;\frac{1}{\rho}(\lambda^k-\lambda^*)^\top(\lambda^k-\hat{\lambda}^k)&\\ \geq\;\sum_i\rho\|\mathbf{A}_i(\mathbf{x}_i^k-\hat{\mathbf{x}}_i^k)\|^2\;+\;\frac{1}{\rho}\|\hat{\lambda}^k-\lambda^k\|^2\;+\;(\hat{\lambda}^k-\lambda^k)^\top[\mathbf{r}(\mathbf{x}^k)-\mathbf{r}(\hat{\mathbf{x}}^k)].&\end{aligned}\tag{30}$$

Proof Consider the result of Lemma 1 and add the term $\rho\sum_i\Big[(\mathbf{A}_i\hat{\mathbf{x}}_i^k-\mathbf{A}_i\mathbf{x}_i^*)^\top(\mathbf{A}_i\mathbf{x}_i^k-\mathbf{A}_i\hat{\mathbf{x}}_i^k)\Big]$ to both sides of inequality (20), which gives us

$$\begin{aligned}&\rho\sum_i\Big[(\mathbf{A}_i\hat{\mathbf{x}}_i^k-\mathbf{A}_i\mathbf{x}_i^*)^\top(\mathbf{A}_i\mathbf{x}_i^k-\mathbf{A}_i\hat{\mathbf{x}}_i^k)\Big]\;+\;\frac{1}{\rho}(\hat{\lambda}^k-\lambda^*)^\top(\lambda^k-\hat{\lambda}^k)\\ &\quad\geq\;\rho\sum_i\Big[(\mathbf{A}_i\hat{\mathbf{x}}_i^k-\mathbf{A}_i\mathbf{x}_i^*)^\top(\mathbf{A}_i\mathbf{x}_i^k-\mathbf{A}_i\hat{\mathbf{x}}_i^k)\Big]\;+\;\rho\sum_i\Big[(\mathbf{A}_i\hat{\mathbf{x}}_i^k-\mathbf{A}_i\mathbf{x}_i^*)^\top\sum_{j\neq i}(\mathbf{A}_j\mathbf{x}_j^k-\mathbf{A}_j\hat{\mathbf{x}}_j^k)\Big].\end{aligned}$$

Grouping the terms at the right-hand side of the inequality by their common factor, we transform the estimate as follows:

$$\begin{aligned}\rho\sum_i\Big[(\mathbf{A}_i\hat{\mathbf{x}}_i^k-\mathbf{A}_i\mathbf{x}_i^*)^\top(\mathbf{A}_i\mathbf{x}_i^k-\mathbf{A}_i\hat{\mathbf{x}}_i^k)\Big]\;+\;\frac{1}{\rho}(\hat{\lambda}^k-\lambda^*)^\top(\lambda^k-\hat{\lambda}^k)&\\ \geq\;\rho\sum_i\Big[(\mathbf{A}_i\hat{\mathbf{x}}_i^k-\mathbf{A}_i\mathbf{x}_i^*)^\top\sum_j(\mathbf{A}_j\mathbf{x}_j^k-\mathbf{A}_j\hat{\mathbf{x}}_j^k)\Big]&\end{aligned}\tag{31}$$

Recall that $\sum_j\mathbf{A}_j(\mathbf{x}_j^k-\hat{\mathbf{x}}_j^k)=\mathbf{r}(\mathbf{x}^k)-\mathbf{r}(\hat{\mathbf{x}}^k)$, which means that this term is a constant factor with respect to the summation over i in the right hand side of (31). Moreover, $\sum_i\mathbf{A}_i\mathbf{x}_i^*=\mathbf{b}$. Substituting these terms at the right-hand side of (31), yields

$$\begin{aligned}\rho\sum_i\Big[(\mathbf{A}_i\hat{\mathbf{x}}_i^k-\mathbf{A}_i\mathbf{x}_i^*)^\top(\mathbf{A}_i\mathbf{x}_i^k-\mathbf{A}_i\hat{\mathbf{x}}_i^k)\Big]\;+\;\frac{1}{\rho}(\hat{\lambda}^k-\lambda^*)^\top(\lambda^k-\hat{\lambda}^k)&\\ \geq\;\rho\Big[\sum_i(\mathbf{A}_i\hat{\mathbf{x}}_i^k-\mathbf{A}_i\mathbf{x}_i^*)\Big]^\top\big[\mathbf{r}(\mathbf{x}^k)-\mathbf{r}(\hat{\mathbf{x}}^k)\big]&\\ =\;\rho\Big[(\sum_i\mathbf{A}_i\hat{\mathbf{x}}_i^k-\mathbf{b})^\top\big[\mathbf{r}(\mathbf{x}^k)-\mathbf{r}(\hat{\mathbf{x}}^k)\big]\Big]\;=\;(\hat{\lambda}^k-\lambda^k)^\top[\mathbf{r}(\mathbf{x}^k)-\mathbf{r}(\hat{\mathbf{x}}^k)]&\end{aligned}\tag{32}$$

Next, we represent

$$\begin{aligned}(\mathbf{A}_i\hat{\mathbf{x}}_i^k-\mathbf{A}_i\mathbf{x}_i^*)\;&=\;(\mathbf{A}_i\mathbf{x}_i^k-\mathbf{A}_i\mathbf{x}_i^*)+(\mathbf{A}_i\hat{\mathbf{x}}_i^k-\mathbf{A}_i\mathbf{x}_i^k)\\ \text{and}\quad\hat{\lambda}^k-\lambda^*\;&=\;(\lambda^k-\lambda^*)+(\hat{\lambda}^k-\lambda^k),\end{aligned}$$

in the left-hand side of (32). We obtain

$$\begin{aligned}&\rho\sum_i\Big[(\mathbf{A}_i\mathbf{x}_i{}^k-\mathbf{A}_i\mathbf{x}_i^*)^\top(\mathbf{A}_i\mathbf{x}_i^k-\mathbf{A}_i\hat{\mathbf{x}}_i^k)\Big]\;+\;\frac{1}{\rho}(\lambda^k-\lambda^*)^\top(\lambda^k-\hat{\lambda}^k)\\ &\qquad\geq\;\rho\sum_i\|\mathbf{A}_i(\mathbf{x}_i^k-\hat{\mathbf{x}}_i^k)\|^2+\frac{1}{\rho}\|\hat{\lambda}^k-\lambda^k\|^2+(\hat{\lambda}^k-\lambda^k)^\top[\mathbf{r}(\mathbf{x}^k)-\mathbf{r}(\hat{\mathbf{x}}^k)],\end{aligned}$$

which completes the proof. □

In the next lemma, we obtain a modified version of (30) whose right-hand side is non-negative. This result is utilized in Theorem 1 to show that the Lyapunov (merit) function (19) is strictly decreasing. We shall use the following "pseudo-dual" variable

$$\bar{\lambda}^k = \lambda^k + \rho(1-\tau)\mathbf{r}(\mathbf{x}^k). \tag{33}$$

Lemma 3 *Under the assumptions (A1)–(A3), the following estimate holds*

$$\rho\sum_i \Big[(\mathbf{A}_i\mathbf{x}_i{}^k - \mathbf{A}_i\mathbf{x}_i^*)^\top(\mathbf{A}_i\mathbf{x}_i^k - \mathbf{A}_i\hat{\mathbf{x}}_i^k)\Big] \;+\; \frac{1}{\rho}(\bar{\lambda}^k - \lambda^*)^\top(\lambda^k - \hat{\lambda}^k)$$
$$\geq\; \frac{\rho}{2}\sum_i \|\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)\|^2 \;+\; \Big(\frac{\tau}{\rho} - \frac{\tau^2 q}{2\rho}\Big)\|\lambda^k - \hat{\lambda}^k\|^2, \tag{34}$$

where $\bar{\lambda}^k$ *are defined in* (33).

Proof Adding $\frac{1}{\rho}\big[\rho(1-\tau)\mathbf{r}(\mathbf{x}^k)\big]^\top(\lambda^k - \hat{\lambda}^k) = -(1-\tau)\rho\mathbf{r}(\mathbf{x}^k)^\top\mathbf{r}(\hat{\mathbf{x}}^k)$ to both sides of inequality (30) we get:

$$\rho\sum_i \Big[(\mathbf{A}_i\mathbf{x}_i{}^k - \mathbf{A}_i\mathbf{x}_i^*)^\top(\mathbf{A}_i\mathbf{x}_i^k - \mathbf{A}_i\hat{\mathbf{x}}_i^k)\Big] \;+\; \frac{1}{\rho}(\bar{\lambda}^k - \lambda^*)^\top(\lambda^k - \hat{\lambda}^k)$$
$$\geq\; \rho\sum_i \|\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}^k)\|^2 \;+\; \frac{1}{\rho}\|\hat{\lambda}^k - \lambda^k\|^2 \;+\; (\hat{\lambda}^k - \lambda^k)^\top\big[\mathbf{r}(\mathbf{x}^k) - \mathbf{r}(\hat{\mathbf{x}}^k)\big]$$
$$-(1-\tau)\rho\mathbf{r}(\mathbf{x}^k)^\top\mathbf{r}(\hat{\mathbf{x}}^k). \tag{35}$$

Consider the term $(\hat{\lambda}^k - \lambda^k)^\top\big[\mathbf{r}(\mathbf{x}^k) - \mathbf{r}(\hat{\mathbf{x}}^k)\big] \;-\; (1-\tau)\rho\mathbf{r}(\mathbf{x}^k)^\top\mathbf{r}(\hat{\mathbf{x}}^k)$ at the right hand side of (35). We manipulate it to yield:

$$(\hat{\lambda}^k - \lambda^k)^\top\big[\mathbf{r}(\mathbf{x}^k) - \mathbf{r}(\hat{\mathbf{x}}^k)\big] - (1-\tau)\rho\mathbf{r}(\mathbf{x}^k)^\top\mathbf{r}(\hat{\mathbf{x}}^k) = \tag{36}$$
$$= \rho\mathbf{r}(\hat{\mathbf{x}}^k)\Big[\mathbf{r}(\mathbf{x}^k) - \mathbf{r}(\hat{\mathbf{x}}^k)\Big] - (1-\tau)\rho\mathbf{r}(\mathbf{x}^k)^\top\mathbf{r}(\hat{\mathbf{x}}^k)$$
$$= \rho\mathbf{r}(\hat{\mathbf{x}}^k)^\top\Big[\mathbf{r}(\mathbf{x}^k) - \mathbf{r}(\hat{\mathbf{x}}^k)\Big] - (1-\tau)\rho\Big[\mathbf{r}(\mathbf{x}^k) - \mathbf{r}(\hat{\mathbf{x}}^k) + \mathbf{r}(\hat{\mathbf{x}}^k)\Big]^\top\mathbf{r}(\hat{\mathbf{x}}^k)$$
$$= \tau\rho\mathbf{r}(\hat{\mathbf{x}}^k)^\top\Big[\mathbf{r}(\mathbf{x}^k) - \mathbf{r}(\hat{\mathbf{x}}^k)\Big] - (1-\tau)\rho\|\mathbf{r}(\hat{\mathbf{x}}^k)\|^2$$
$$= \tau(\hat{\lambda}^k - \lambda^k)^\top\sum_i \mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k) - (1-\tau)\frac{1}{\rho}\|\hat{\lambda}^k - \lambda^k\|^2.$$

Substituting back in (35), we obtain:

$$\rho\sum_i \Big[(\mathbf{A}_i\mathbf{x}_i{}^k - \mathbf{A}_i\mathbf{x}_i^*)^\top(\mathbf{A}_i\mathbf{x}_i^k - \mathbf{A}_i\hat{\mathbf{x}}_i^k)\Big] \;+\; \frac{1}{\rho}(\bar{\lambda}^k - \lambda^*)^\top(\lambda^k - \hat{\lambda}^k)$$
$$\geq\; \rho\sum_i \|\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)\|^2 \;+\; \frac{\tau}{\rho}\|\hat{\lambda}^k - \lambda^k\|^2 \;+\; \tau\sum_i(\hat{\lambda}^k - \lambda^k)^\top\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k). \tag{37}$$

Each of the terms $\tau(\hat{\lambda}^k - \lambda^k)^\top\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)$ on the right hand side of (37) can be bounded below by considering

$$\tau(\hat{\lambda}^k - \lambda^k)^\top(\mathbf{A}_i\mathbf{x}_i^k - \mathbf{A}_i\hat{\mathbf{x}}_i^k) \;=\; \tau\sum_{j=1}^m(\hat{\lambda}_j^k - \lambda_j^k)^\top\big[\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)\big]_j$$
$$\geq\; -\frac{1}{2}\sum_{j=1}^m\left(\rho\Big[\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)\Big]_j^2 \;+\; \frac{\tau^2}{\rho}(\hat{\lambda}_j^k - \lambda_j^k)^2\right),$$

where $\big[\,\cdot\,\big]_j$ denotes the j-th row of a matrix, and λ_j indicates the Lagrange multiplier of the j-th constraint. Note, however, that some of the rows of $\mathbf{A}_i$ might be zero. If $[\mathbf{A}_i]_j = \mathbf{0}$, then it follows that $(\hat{\lambda}_j^k - \lambda_j^k)^\top \big[\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)\big]_j = 0$. Hence, denoting the set of nonzero rows of $\mathbf{A}_i$ as $\mathscr{Q}_i$, i.e., $\mathscr{Q}_i = \{j = 1,\dots,m \;:\; [\mathbf{A}_i]_j \neq \mathbf{0}\}$, we can obtain a tighter lower bound for each $\tau(\hat{\lambda}^k - \lambda^k)^\top \mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)$ as

$$\tau(\hat{\lambda}^k - \lambda^k)^\top(\mathbf{A}_i\mathbf{x}_i^k - \mathbf{A}_i\hat{\mathbf{x}}_i^k) \;\geq\; -\frac{1}{2}\sum_{j\in\mathscr{Q}_i}\left(\rho\Big[\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)\Big]_j^2 + \frac{\tau^2}{\rho}(\hat{\lambda}_j^k - \lambda_j^k)^2\right). \tag{38}$$

Now, recall from (6) that q denotes the maximum number of non-zero blocks $[\mathbf{A}_i]_j$ over all j (in other words, q is the maximum number of locations i that are involved in the constraint j). Then, summing inequality (38) over all i, we observe that each quantity $(\hat{\lambda}_j^k - \lambda_j^k)^2$ is included in the summation at most q times.

This observation leads us to the bound

$$\tau\sum_i(\hat{\lambda}^k - \lambda^k)^\top\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k) \;\geq\; -\frac{1}{2}\Big(\sum_i\rho\|\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)\|^2 + \frac{\tau^2 q}{\rho}\|\hat{\lambda}^k - \lambda^k\|^2\Big). \tag{39}$$

Finally, we substitute (39) into (37) to get

$$\begin{aligned}\rho\sum_i\Big[(\mathbf{A}_i\mathbf{x}_i^k - \mathbf{A}_i\mathbf{x}_i^*)^\top(\mathbf{A}_i\mathbf{x}_i^k - \mathbf{A}_i\hat{\mathbf{x}}_i^k)\Big] &+ \frac{1}{\rho}(\bar{\lambda}^k - \lambda^*)^\top(\lambda^k - \hat{\lambda}^k)\\ &\geq \sum_i\frac{\rho}{2}\|\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)\|^2 + \big(\frac{\tau}{\rho} - \frac{\tau^2 q}{2\rho}\big)\|\lambda^k - \hat{\lambda}^k\|^2,\end{aligned} \tag{40}$$

which completes the proof. □

We are ready to prove the key result pertaining to the convergence of our method. We shall show that the function ϕ defined in (19) is a Lyapunov function for ADAL.

Theorem 1 *Assume (A1)–(A3). If the ADAL method uses stepsize τ satisfying*

$$0 < \tau < \frac{1}{q},$$

then, the sequence $\{\phi(\mathbf{x}^k, \lambda^k)\}$, with $\phi(\mathbf{x}^k, \lambda^k)$ defined in (19)*, is strictly decreasing.*

Proof We show that the dual update step (16) in the ADAL method results in the following update rule for the variables $\bar{\lambda}^k$, which are defined in (33):

$$\bar{\lambda}^{k+1} \;=\; \bar{\lambda}^k + \tau\rho\mathbf{r}(\hat{\mathbf{x}}^k) \tag{41}$$

Indeed,

$$\begin{aligned}\lambda^{k+1} &= \lambda^k + \tau\rho\mathbf{r}(\mathbf{x}^{k+1})\\ &= \lambda^k + \tau\rho\Big[(1-\tau)\mathbf{r}(\mathbf{x}^k) + \tau\mathbf{r}(\hat{\mathbf{x}}^k)\Big]\\ &= \lambda^k + \tau\Big[-(1-\tau)\rho\Big(\mathbf{r}(\hat{\mathbf{x}}^k) - \mathbf{r}(\mathbf{x}^k)\Big) + \rho\mathbf{r}(\hat{\mathbf{x}}^k)\Big]\\ &= \lambda^k - (1-\tau)\rho\tau\Big(\mathbf{r}(\hat{\mathbf{x}}^k) - \mathbf{r}(\mathbf{x}^k)\Big) + \tau\rho\mathbf{r}(\hat{\mathbf{x}}^k)\end{aligned} \tag{42}$$

Adding $(1-\tau)\rho\mathbf{r}(\mathbf{x}^k)$ to both sides of (42) and rearranging terms, we obtain

$$\lambda^{k+1} + (1-\tau)\rho\Big[\mathbf{r}(\mathbf{x}^k) + \tau\Big(\mathbf{r}(\hat{\mathbf{x}}^k) - \mathbf{r}(\mathbf{x}^k)\Big)\Big] \;=\; \lambda^k + (1-\tau)\rho\mathbf{r}(\mathbf{x}^k) + \tau\rho\mathbf{r}(\hat{\mathbf{x}}^k).$$

This is equivalent to

$$\lambda^{k+1} + (1-\tau)\rho\mathbf{r}(\mathbf{x}^{k+1}) \;=\; \lambda^k + (1-\tau)\rho\mathbf{r}(\mathbf{x}^k) + \tau\rho\mathbf{r}(\hat{\mathbf{x}}^k),$$

which is the update rule (41).

We define the progress at each iteration k of the ADAL method as

$$\theta_k(\tau) \;=\; \phi(\mathbf{x}^k,\lambda^k) - \phi(\mathbf{x}^{k+1},\lambda^{k+1}).$$

We substitute $\bar{\lambda}^k$ in the formula for calculating the function ϕ and use relation (41). The progress $\theta_k(\tau)$ can be evaluated as follows:

$$\begin{aligned}
\theta_k(\tau) \;&=\; \sum_{i=1}^N \rho\|\mathbf{A}_i(\mathbf{x}_i^k - \mathbf{x}_i^*)\|^2 + \frac{1}{\rho}\|\bar{\lambda}^k - \lambda^*\|^2 - \sum_{i=1}^N \rho\|\mathbf{A}_i(\mathbf{x}_i^{k+1} - \mathbf{x}_i^*)\|^2 - \frac{1}{\rho}\|\bar{\lambda}^{k+1} - \lambda^*\|^2 \\
&=\; \sum_{i=1}^N \rho\|\mathbf{A}_i(\mathbf{x}_i^k - \mathbf{x}_i^*)\|^2 \;+\; \frac{1}{\rho}\|\bar{\lambda}^k - \lambda^*\|^2 \\
&\quad -\sum_{i=1}^N \rho\|\mathbf{A}_i(\mathbf{x}_i^k - \mathbf{x}_i^*) \;+\; \tau\mathbf{A}_i(\hat{\mathbf{x}}_i^k - \mathbf{x}_i^k)\|^2 - \frac{1}{\rho}\|\bar{\lambda}^k - \lambda^* + \tau\rho\mathbf{r}(\hat{\mathbf{x}}^k)\|^2 \\
&=\; 2\tau\Bigg[\rho\sum_i\Big[(\mathbf{A}_i\mathbf{x}_i{}^k - \mathbf{A}_i\mathbf{x}_i^*)^\top(\mathbf{A}_i\mathbf{x}_i^k - \mathbf{A}_i\hat{\mathbf{x}}_i^k)\Big] \;+\; \frac{1}{\rho}(\bar{\lambda}^k - \lambda^*)^\top(\lambda^k - \hat{\lambda}^k)\Bigg] \\
&\quad -\tau^2\Bigg[\sum_i \rho\|\mathbf{A}_i(\hat{\mathbf{x}}_i^k - \mathbf{x}_i^k)\|^2 \;+\; \frac{1}{\rho}\|\hat{\lambda}^k - \lambda^k\|^2\Bigg].
\end{aligned} \tag{43}$$

We use Lemma 3 to substitute the positive term in (43) by its lower bound and obtain that the progress at each iteration is estimated as follows:

$$\begin{aligned}
\theta_k(\tau) \;&\geq\; 2\tau\Bigg[\sum_i \frac{\rho}{2}\|\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)\|^2 \;+\; (\frac{\tau}{\rho} - \frac{\tau^2 q}{2\rho})\|\lambda^k - \hat{\lambda}^k\|^2\Bigg] \qquad (44) \\
&\quad -\tau^2\Bigg[\sum_i \rho\|\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)\|^2 \;+\; \frac{1}{\rho}\|\lambda^k - \hat{\lambda}^k\|^2\Bigg] \\
&\geq\; \tau\Bigg[\sum_i (1-\tau)\rho\|\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)\|^2 \;+\; \Big(\frac{\tau}{\rho} - \frac{\tau^2 q}{\rho}\Big)\|\lambda^k - \hat{\lambda}^k\|^2\Bigg] \\
&\geq\; 0 \quad \text{whenever } 0 < \tau < \frac{1}{q} < 1.
\end{aligned}$$

Relation (44) implies that $\theta_k > 0$ during the execution of ADAL. Indeed, the coefficients of all terms at the right-hand side are positive due to the choice of parameters $\rho > 0$ and $0 < \tau < \frac{1}{q} \leq 1$. This means that the lower bound on $\theta_k(\tau)$ can be zero only if all terms are equal to zero, which means that $\lambda^k - \hat{\lambda}^k = \rho\mathbf{r}(\hat{\mathbf{x}}^k) = 0$, and $\mathbf{A}_i\mathbf{x}_i^k = \mathbf{A}_i\hat{\mathbf{x}}_i^k$, for all $i = 1,\dots,N$. In such a case, the ADAL method stops at Step 3. Thus, $\theta_k > 0$ during the execution of ADAL, which in turn means that the merit function $\phi(\mathbf{x}^k,\lambda^k)$ is strictly decreasing. □

Theorem 2 *In addition to (A1)-(A3), assume that the sets $\mathscr{X}_i$ are bounded for all $i = 1,\dots N$. Then the ADAL method either stops at an optimal solution of problem* (2) *or generates a sequence of λ^k converging to an optimal solution of it. Any sequence $\{\mathbf{x}^k\}$ generated by the ADAL algorithm has an accumulation point and any such point is an optimal solution of problem* (1).

Proof If the method stops at Step 3 in some iteration k_0, then $\mathbf{r}(\hat{\mathbf{x}}^{k_0}) = 0$ and $\mathbf{A}_i\hat{\mathbf{x}}_i^{k_0} = \mathbf{A}_i\mathbf{x}_i^{k_0}$. In this case, the optimality conditions (21) for problems (14) become

$$0 \in \partial f_i(\hat{\mathbf{x}}_i^{k_0}) + \mathbf{A}_i^\top \lambda^{k_0} + \mathscr{N}_{\mathscr{X}_i}(\hat{\mathbf{x}}_i^k), \quad i = 1\dots,N$$

which implies that $\hat{\mathbf{x}}_i^{k_0}$ is an optimal solution of the problem $\min_{\mathbf{x}_i\in\mathscr{X}_i} L_i(\mathbf{x}_i,\lambda^{k_0})$ with optimal Lagrange multiplier λ^{k_0} for all $i = 1\dots,N$. Thus, λ^{k_0} is a solution of problem (2) and $\mathbf{x}^{k_0}$ is a solution of problem (1).

Now, we consider the case, in which the method generates an infinite sequence of iterates. Relation (44) implies that

$$\phi(\mathbf{x}^{k+1},\lambda^{k+1}) \le \phi(\mathbf{x}^k,\lambda^k) - \tau\Big[\sum_i (1-\tau)\rho\|\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)\|^2 + \Big(\frac{\tau}{\rho} - \frac{\tau^2 q}{\rho}\Big)\|\lambda^k - \hat{\lambda}^k\|^2\Big] \tag{45}$$

Iterating inequality (45) for $k = 1,2,\dots$ and dividing by $\tau > 0$, we obtain:

$$\sum_{k=1}^{\infty}\left[\sum_{i=1}^{N}(1-\tau)\rho\|\mathbf{A}_i(\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k)\|^2 + \Big(\frac{\tau}{\rho} - \frac{\tau^2 q}{\rho}\Big)\|\lambda^k - \hat{\lambda}^k\|^2\right] < \frac{1}{\tau}\phi(\mathbf{x}^1,\lambda^1) \tag{46}$$

This implies that the sequence $\{\mathbf{A}_i\hat{\mathbf{x}}_i^k - \mathbf{A}_i\mathbf{x}_i^k\}$ converges to zero as $k\to\infty$. We substitute $\hat{\lambda}^k - \lambda^k = \rho\mathbf{r}(\hat{\mathbf{x}}^k)$ at the left-hand side of (46) and infer that the sequence $\{\mathbf{r}(\hat{\mathbf{x}}^k)\}$ converges to zero whenever $k\to\infty$. By the monotonicity and boundedness properties of $\phi(\mathbf{x}^k,\lambda^k)$, we conclude that the sequence $\{\lambda^k\}$ is convergent as well. We denote $\lim_{k\to\infty}\lambda^k = \mu$.

Due to the boundedness of $\mathscr{X}_i$ all sequences $\{\mathbf{x}_i^k\}$, $i = 1,\dots N$, are bounded. This means that the sequences $\{\mathbf{x}_i^k\}$ have accumulation points $\tilde{\mathbf{x}}_i$, which are also accumulation points of $\{\hat{\mathbf{x}}_i^k\}$ due to Step 2 of the ADAL algorithm. We can choose a subsequence $\mathscr{K}\subset\{1,2,\dots\}$ so that $\{\mathbf{x}_i^k\}_{k\in\mathscr{K}}$ and $\{\hat{\mathbf{x}}_i^k\}_{k\in\mathscr{K}}$ converge to $\tilde{\mathbf{x}}_i$ for all $i = 1,\dots,N$. Denoting $\tilde{\mathbf{x}} = (\tilde{\mathbf{x}}_1,\dots,\tilde{\mathbf{x}}_N)^\top$, we observe that the point $\tilde{\mathbf{x}}$ is feasible due to the closedness of the sets $\mathscr{X}_i$ and the continuity of $\mathbf{r}(\cdot)$.

For any $i = 1,\dots,N$, consider the sequence $\{\mathbf{s}_i^k\}_{k\in\mathscr{K}}$, where $\mathbf{s}_i^k$ are the subgradients of f_i from the optimality condition (22) for problems (14). The subdifferential mapping $\mathbf{x}\rightrightarrows\partial f(\mathbf{x})$ of any finite-valued convex function defined on $\mathbb{R}^n$ is upper semi-continuous and has compact images. Therefore, the sequences $\{\mathbf{s}_i^k\}_{k\in\mathscr{K}}$ have convergent subsequences due to a fundamental result that goes back to [1]. We can choose $\mathscr{K}_1\subset\mathscr{K}$ such that $\{\mathbf{s}_i^k\}_{k\in\mathscr{K}_1}$ converge to some $\tilde{\mathbf{s}}_i\subset\partial f_i(\tilde{\mathbf{x}}_i)$ for all $i = 1,\dots,N$.

Passing to the limit in equation (22), we infer that each sequence $\{\mathbf{z}_i^k\}_{k\in\mathscr{K}_1}$ converges to a point $\tilde{\mathbf{z}}_i$, $i = 1,\dots,N$. The mapping $\mathbf{x}_i\rightrightarrows\mathscr{N}_{\mathscr{X}_i}(\mathbf{x}_i)$ has closed graph and, hence, $\tilde{\mathbf{z}}_i\in\mathscr{N}_{\mathscr{X}_i}(\tilde{\mathbf{x}}_i)$.

After the limit pass in (22) over $k\in\mathscr{K}_1$, we conclude that

$$0 = \tilde{\mathbf{s}}_i + \mathbf{A}_i^\top\mu + \tilde{\mathbf{z}}_i^k, \quad \forall\, i = 1\dots,N.$$

This relation together with the feasibility of $\tilde{\mathbf{x}}$ implies that $\tilde{\mathbf{x}}$ is a solution of problem (1) and μ is a solution of problem (2). □

Note that if $\mathbf{r}(\hat{\mathbf{x}}^{k_0}) = 0$ at some iteration k_0 but $\mathbf{A}_i\hat{\mathbf{x}}_i^{k_0} \neq \mathbf{A}_i\mathbf{x}_i^{k_0}$, then the ADAL method will iterate with the same Lagrange multipliers until the information is synchronized ($\mathbf{A}_i\hat{\mathbf{x}}_i^k = \mathbf{A}_i\mathbf{x}_i^k$ for all $i = 1,\dots N$) resembling one inner loop of the DQA method.

Remark 1 The ADAL method and its convergence analysis are presented with a single penalty parameter $\rho \in \mathbb{R}_+$. Nevertheless, the constraints can be re-scaled beforehand, which amounts to using different penalty parameters ρ_j, $j = 1,\dots m$ for each constraint. The convergence follows in such a case by exactly the same line of arguments. However, the penalty parameter ρ should be kept fixed from iteration to iteration.

Since we have now established convergence of ADAL in theory, it is also important to state the following observations regarding its behavior in practical applications.

Although, we have established which stepsizes guarantee theoretical convergence, we have undertaken numerical experiments, in which we multiply the primal and dual update stepsizes with some relaxation factors β_p, β_d, in an effort to accelerate convergence. Here, β_p is applied to the primal variables update step and β_d to the dual update step. Our numerical experiments indicate that, at least for the applications considered here, we can employ relaxation factors $\beta_p \in [1, 2.5)$ and $\beta_d \in [1, q)$. The employment of such relaxations provides reasonable acceleration to the overall convergence speed.

5 Numerical Experiments.

In the previous sections, we saw that the algorithmic forms of ADAL, DQA and ASM possess some interesting similarities and differences, despite the fact that their convergence proofs follow completely different paths. In what follows, we briefly discuss some of the connections between the algorithmic forms of these methods.

A key idea in the convergence analysis of Diagonal Quadratic Approximation (DQA) method, presented in [24], is the calculation of the value of the augmented Lagrangian function from (4) by a sequence of successive separable approximations in the inner loop of DQA. In other words, the minimization of the augmented Lagrangian (4) is approximated iteratively by successive minimizations of the local augmented Lagrangians (8) and primal variable update steps (9). Furthermore, convergence of DQA is guaranteed if the stepsize τ satisfies $0 < \tau < \frac{1}{q}$ and the convergence rate is related to the scarcity of the matrix A. In [24], values of $\tau = \frac{1}{2q}$ are recommended. In ADAL, we find that the inner loop of DQA can be terminated after just one iteration and obtain the fastest convergence, as simulations suggest. Interestingly, numerical experiments have shown that convergence accelerates with the increase of τ and in most cases the factor τ in the dual updates can be neglected without compromising convergence.

The main difference between the algorithmic forms of ASM and ADAL is that ASM uses different local augmented Lagrangians $\bar{\Lambda}_\rho^i(\mathbf{x}_i, \mathbf{x}^k, \lambda^k)$ in Step (11), which involve the constraint degrees in the quadratic terms. Furthermore, the steps with the dual variables are performed with the use of the $\hat{\mathbf{x}}_i^k$ variables in ASM. In contrast, the convergence of ADAL is compromised if we perform the dual update with the variables $\hat{\mathbf{x}}_i^k$ instead of $\mathbf{x}_i^{k+1}$. Finally, ASM introduces the relaxation factor $\sigma \in (0, 2)$ from the theory of the generalized ADMM and utilizes it as a stepsize for the primal update. In contrast, ADAL uses $\tau = 1/q$.

We have implemented ADAL, DQA and ASM on two popular network optimization problems: Network Utility Maximization [28] and Network flow problem [4]. Comparative results between all algorithms are presented, which illustrate that ADAL is significantly faster than both the DQA and ASM.

Additionally, we apply all algorithms to solve a two-stage stochastic optimization problem. Dedicated decomposition methods are used for the numerical solution of such problems due to their large dimensions in any realistic situation.

In all simulations, the maximum residual $\max_j \mathbf{r}_j(\mathbf{x}^k)$, i.e., the maximum constraint violation among all constraints $j = 1,\dots,m$, was monitored as a criterion of convergence. The examined networks were randomly generated with the agents uniformly distributed in rectangle boxes. The inner loop termination criterion for DQA is set to be $\|\mathbf{A}_i\hat{\mathbf{x}}_i^k - \mathbf{A}_i\mathbf{x}_i^k\|_\infty \leq 10^{-2}$ for every $i = 1,\dots,N$, unless otherwise noted.

We note that, after extensive simulations, we have found that ASM and DQA required relatively larger values of the penalty coefficient in comparison with ADAL, e.g., $\rho_{ASM} \geq 3\rho_{ADAL}$, and $\rho_{DQA} \geq 5\rho_{ADAL}$ typically, (where the subscripts denote the respective ρ used for each method), at least for the problems considered here.

5.1 Network Utility Maximization problem

Consider an undirected graph $G = (N,A)$ with a set of nodes N and a set of arcs A. The set of nodes is consisted of two subsets $N = \{S,D\}$, where S is the set of source nodes and D the set of destination nodes. Let s_i denote the rate of resource production at node $i \in S$ and also let t_{ij} denote the rate of a commodity flowing through arc (i,j). Each arc (i,j) has a feasible range of flows $a_{ij} \leq t_{ij} \leq b_{ij}$, where a_{ij}, b_{ij} are given numbers. Denote the neighborhood of node i as $\mathscr{C}_i = \{j : (i,j) \in A\}$. At this point, note that $q = \max_i |\mathscr{C}_i|$ and, according to the convergence analysis, the stepsize in the ADAL algorithm must be $\tau < \frac{1}{q}$. Nevertheless, as mentioned in Section 4 the numerical experiments indicate that significant acceleration is achieved if the stepsizes for the primal and dual updates are relaxed to $\tau = \frac{\beta_p}{q}$ and $\tau = \frac{\beta_d}{q}$, respectively, where $\beta_p \in [1, 2.5)$ and $\beta_d \in [1, q)$.

The NUM problem entails solving

$$\text{(NUM)} \qquad \begin{aligned} \max \quad & U(\mathbf{s}) = \sum_{i\in S} U_i(s_i) \\ \text{subject to} \quad & \sum_{\{j\in\mathscr{C}_i\}} t_{ij} - \sum_{\{j|i\in\mathscr{C}_j\}} t_{ji} = s_i, \quad \forall\, i \in S \\ & a_{ij} \leq t_{ij} \leq b_{ij}, \quad \forall\, (i,j) \in A \end{aligned}$$

The NUM problem maximizes the amount of resources produced at the source nodes and route the resources to the destination nodes. The constraints $\sum_{\{j\in\mathscr{C}_i\}} t_{ij} - \sum_{\{j|i\in\mathscr{C}_j\}} t_{ji} = s_i$ express the conservation of commodity flow at each source node. Note that, in our consideration, the destination nodes are modeled as sinks and can absorb any amount of incoming rates. We consider normalized rates $s_i, t_{ij} \in [0,1]$, without any loss of generality.

The requirement on the utility functions $U_i(s_i)$ is that they are monotonically non-decreasing expressing preference to larger transmission rates, i.e. an increase in the rate of one node does not decrease the value of the total utility function to be maximized. In our simulations, we choose $U(\mathbf{s}) = \prod_{i\in S}(s_i)$ in order to maximize the product of rates, which can be recast as the sum of logarithms $U(\mathbf{s}) = \sum_{i\in S}\log(s_i)$. This choice is typical in NUM problems [28] and aims to produce a fairer resource allocation among all source nodes in the network. Note that the choice $U(\mathbf{s}) = \sum_{i\in S} s_i$, would result in a problem where the maximum rates are rewarded. In our case, this would lead to a trivial problem, in which the nodes in communication range of the destinations are rewarded with the maximum rate 1 and the rest with 0.

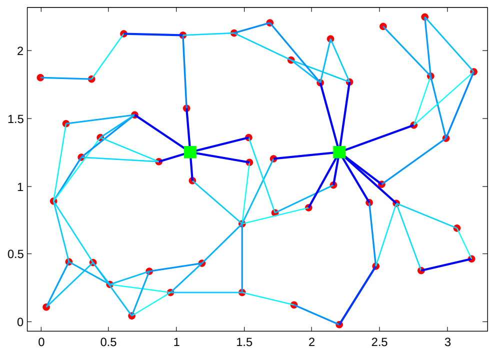


Fig. 1 Random network of 50 sources (red dots) and 2 sinks (green squares). The rate of flow t_{ij} through arc (i, j) defines the thickness and color of the corresponding drawn line. Thin, light blue lines indicate weaker links.

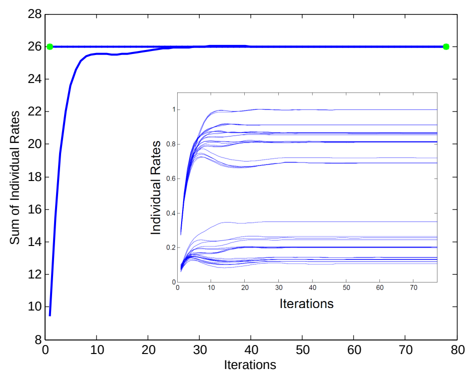


Fig. 2 Evolution of the sum of rates $\sum_{i\in S} s_i$ during implementation of ADAL. The horizontal line depicts the value obtained by solving the centralized problem. We observe that the distributed utility function converges to the optimal solution very fast. Also included is the subfigure illustrating the evolution of individual rates for every source.

Fig. 1 shows a network consisting of 50 sources and 2 sinks that is returned after applying the ADAL algorithm on (NUM). All subsequent simulation results involve networks of this form, unless otherwise noted. Fig. 2 shows the evolution of the individual and total rates, s_i and $\sum_{i\in S} s_i$, respectively, corresponding to maximization of the utility $U(\mathbf{s}) = \sum_{i\in S} \log(s_i)$ for fair allocation. We observe that the utility converges sufficiently to its optimal value in only about 25 iterations. Next, we plot in Fig. 3 the evolution of the maximum residual during the execution of ADAL. The figure contains results for networks of different sizes. An encouraging observation is that network size does not appear to affect speed of convergence dramatically, at least for the NUM problem considered here. Repeated simulations have shown that convergence speed remains at this level of magnitude. Note that, in all cases, the ratio of sources-to-sinks has been maintained the same at 25/1, in an effort to keep the randomly generated networks as similar as possible.

As already mentioned, ADAL can be viewed as a truncated form of the DQA algorithm. In Fig. 4, we explore the connections between the two methods. Fig.4(a) compares results of truncating the inner loop of DQA at a predefined number of iterations M. For $M = 1$,

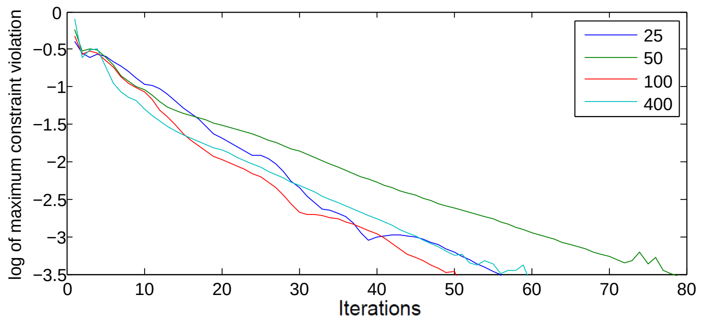
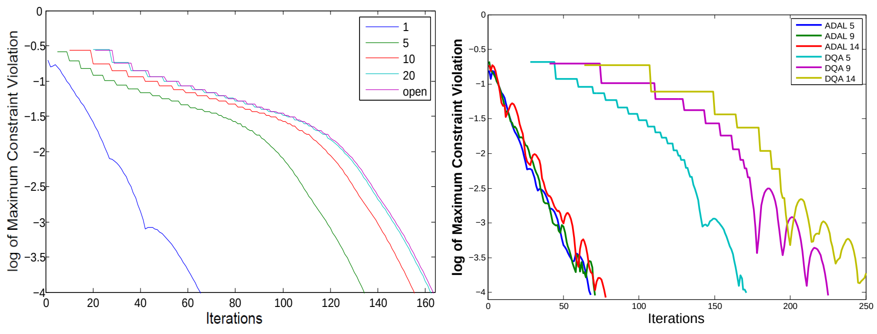


Fig. 3 Constraint violation convergence of ADAL for different network sizes of 25, 50, 100 and 400 source nodes. The ratio of sources-to-destinations is kept at 25/1 and the maximum degree $q=6$ for all cases.



(a) (b)

Fig. 4 Constraint violation convergence for: a) Different exit criteria from the inner loop of DQA. The line labeled 'open' accounts for repeating the DQA inner loop until $\|\mathbf{A}_i\hat{\mathbf{x}}_i^k - \mathbf{A}_i\mathbf{x}_i^k\|_\infty \leq 10^{-2}$ for every $i=1,\ldots,N$. In the other instances we force exit if either the aforementioned criterion is satisfied or if the indicated amount of iterations M has been surpassed. For $M=1$ we obtain the ADAL method. The results correspond to a network of 50 sources and 2 sinks with $q=6$, b) Different network densities for the ADAL and DQA methods. The results correspond to networks of 50 sources and 2 sinks with $q=5,9,14$, respectively. In both figures, the horizontal axis depicts the inner loop iterations for the case of DQA. Also, note that the step shape of the DQA graphs in the figures is caused by the dual updates at each outer loop iteration.

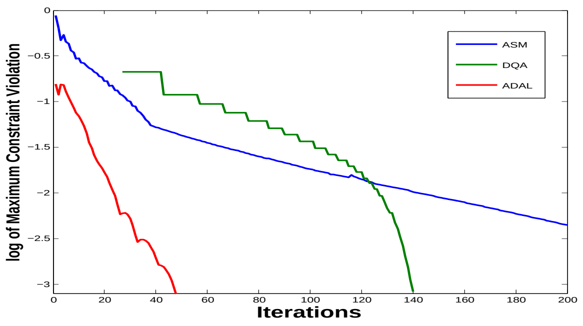


Fig. 5 Comparison between the ASM, DQA and ADAL methods, for a network of 50 sources, 2 sinks and $q=7$.

we obtain the ADAL method. We observe that truncating the inner loop yields accelerated convergence for DQA, with the fastest case being the ADAL algorithm. Note that no theoretical proof for the convergence of DQA for intermediate values of M exists. Moreover, since the performance of both algorithms appears to depend on the maximum degree q we have conducted simulations for different values of q. The results are illustrated in Fig. 4(b). Surprisingly and in contrast to DQA, it appears that ADAL is not greatly affected by the value of q, at least for the NUM problems considered here. Finally, in Fig. 5, we compare the performance of the three methods. The ASM was implemented for $\sigma = 1.9$, the maximum value of σ that did not compromise convergence, while returning the best results. Also, in all three methods the penalty parameter and initialization points were the same, in order to preserve the homogeneity of results. We notice that ADAL performs significanlty better than both ASM and DQA.

5.2 Linear Network flow problem

Closely related to the NUM problem is the optimal network flow problem. Here, we examine the Linear Network Flow (LNF) case, where the arc costs are linear. LNF is a classical problem that has been studied extensively. The assignment, max-flow and shortest path problems are special cases of LNF [4].

Consider a directed graph $G = (N,A)$, with a set of nodes N and a set of arcs A. Each arc (i, j) has associated with it a scalar c_{ij} referred to as the cost coefficient of (i, j). Let t_{ij} denote the flow of arc (i, j) and consider the problem

$$\text{(LNF)} \qquad \begin{aligned} \min \quad & \sum_{(i,j)\in A} c_{ij}t_{ij} \\ \text{subject to} \quad & \sum_{\{j|(i,j)\in A\}} t_{ij} - \sum_{\{j|(j,i)\in A\}} t_{ji} = s_i, \quad \forall\, i \in N \\ & a_{ij} \le t_{ij} \le b_{ij}, \quad \forall\, (i,j) \in A \end{aligned}$$

Essentially, the difference here is that we do not seek to maximize the s_i production rates as in the NUM, but rather set some desired levels of s_i and seek to find the flows that keep the problem feasible while minimizing the total cost. Moreover, the objective function is linear. For the set of S source nodes, we have $s_i > 0$, $\forall\, i \in S$, while for the set of D destination nodes we have $s_i < 0$, $\forall\, i \in D$. The conservation of flow in the network requires that $\sum_{i\in N} s_i = 0$. In the examples shown below, we also set a set of R nodes to be relays, that is $s_i = 0$, $\forall\, i \in R$. In addition, we set the cost coefficients $c_{ij} = 1$ and the arc flow bounds $0 \le t_{ij} \le 1$ for simplicity, without any loss of generality.

Fig. 6 depicts the two typical, 50 node networks that were considered here. Also shown are the corresponding flows as solved by the ADAL method. In Fig. 6(a) we consider a case with 5 sources and 5 destinations, which are set a network diameter apart. On the other hand, Fig. 6(b) depicts a case with 7 sources and 7 destinations, which are set half the network diameter apart. In Fig. 7, we plot the evolution of the objective functions after applying DQA, ASM and ADAL on the two networks depicted in Fig. 6. The evolutions of the respective maximum residuals are shown in Fig. 8. In addition, the evolution of the maximum residual for a larger network of 100 nodes is depicted in Fig. 9(a). We observe that ADAL is still faster than both the ASM and DQA, albeit the gap in convergence speed has decreased. In Fig. 9(b) we plot the evolution of the sequence $\phi(\mathbf{t}^k, \lambda^k)$ to verify the

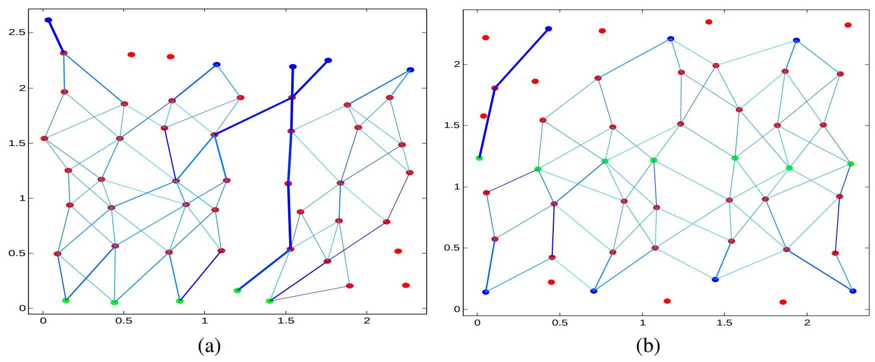


Fig. 6 Two typical LNF cases considered, with $N = 50$ nodes. Blued dots denote source nodes, while green dots correspond to sinks and red dots to relays. The flow t_{ij} through arc (i, j) defines the thickness of the corresponding drawn line. Thin lines indicate weaker links. a) For this case $S = D = 5,\ R = 40$ and the source nodes are positioned to be as far away from the destinations as possible. b) For this case $S = D = 7,\ R = 36$ and the source nodes are only half the network diameter apart from the destinations.

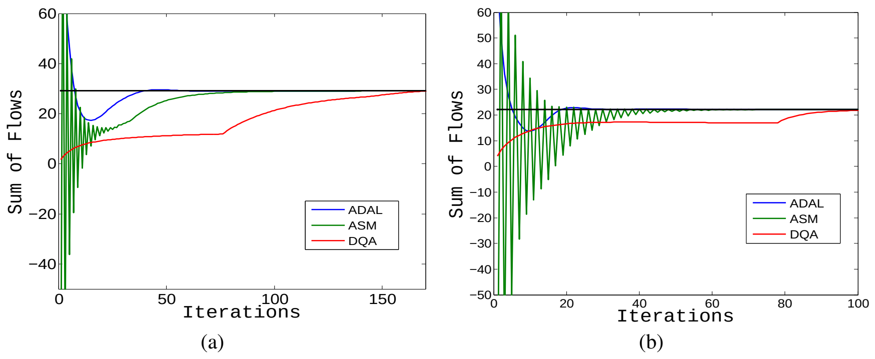


Fig. 7 Evolution of the sum of flows $\sum_{(i,j)\in A}\ t_{ij}$ after implementation of ADAL, ASM and DQA on: a) the case depicted in Fig. 6(a) and b) the case depicted in Fig. 6(b). The horizontal lines depict the objective function values obtained after solving the corresponding centralized problem. Note how the ASM oscillates between positive and negative values of the total flow, which normally should not be the case (since $t_{ij} \geq 0$). This is caused by the fact that we have used stepsize $\sigma = 1.9$ (recall (12)) in our simulations. This choice of σ returned the fastest convergence for ASM, even though it lead to this "counter-intuitive" behavior.

correctness of our proof. The sequence is strictly monotonically decreasing at each iteration, as required.

A possible modification of DQA, ASM and ADAL is to implement these methods in a "Gauss-Seidel" fashion, where the corresponding minimization steps of each method are performed in a sequential fashion for every $i = 1, \ldots, N$. The convergence proofs of all methods are only valid for the "Jacobi" type implementation, where the minimization steps are executed in parallel, however, it is interesting to compare the relative performances between these two approaches. Towards this goal, Fig. 10 depicts the convergence results after applying the Gauss-Seidel version of DQA, ASM and ADAL on the network depicted in Fig. 6(b). Note that the respective results for the "normal" Jacobi versions of these methods are depicted in Fig. 7(b) and Fig. 8(b). We observe that the Gauss-Seidel versions converge faster, in terms of the number of iterations, than the Jacobi ones for all algorithms. Never-

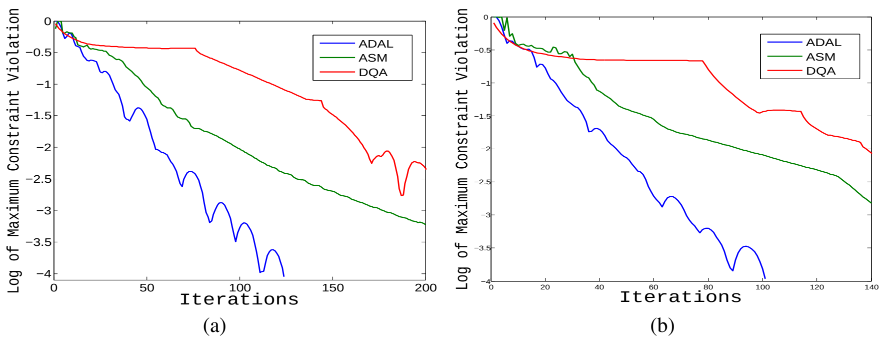


Fig. 8 Evolution of the maximum residual after implementation of ADAL and ASM on: a) the case depicted in Fig. 6(a) and b) the case depicted in Fig. 6(b).

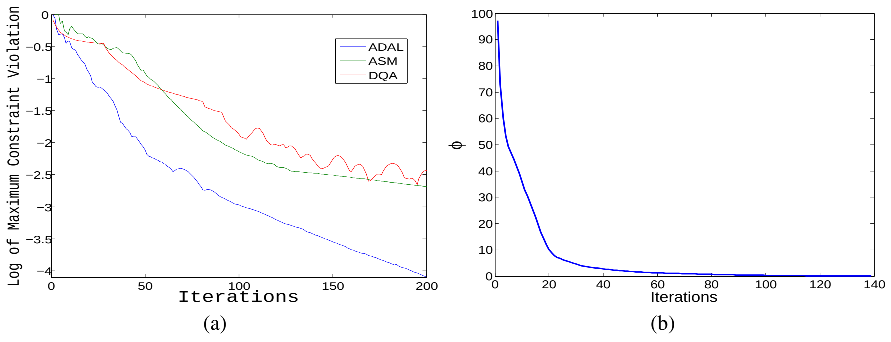


Fig. 9 a) Evolution of the maximum residual after implementation of ADAL, ASM and DQA on a large network with $N = 100$ and $S = D = 10$, The sources were set half a diameter apart from the destinations. b) Evolution of $\phi(\mathbf{t}^k, \lambda^k)$ for ADAL applied on the aforementioned network.

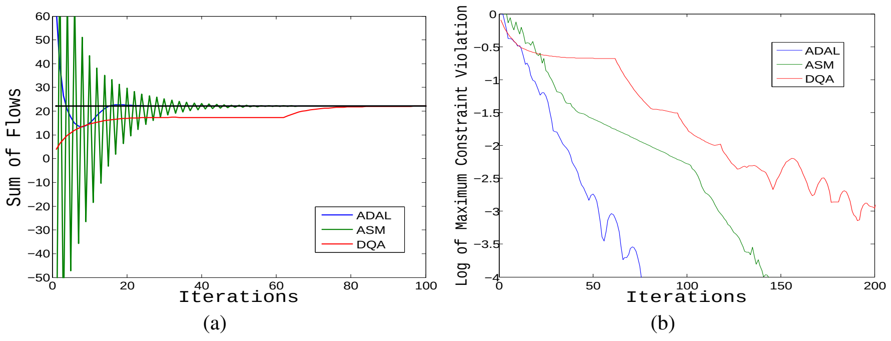


Fig. 10 Convergence results for the Gauss-Seidel type implementation of ADAL, ASM and DQA on the network depicted in Fig. 6(b): a) Objective function convergence, and b) Constraint violation convergence.

theless, the sequential nature of the Gauss-Seidel means that in applications where parallel computation is available, the Jacobi type implementations will converge faster in terms of real-time computation, with the difference increasing for increasing problem sizes.

5.3 Two-Stage Stochastic Optimization problems

Two-stage stochastic optimization problems are among the most popular optimization model for decisions under uncertainty. Problems of this type occur frequently in applications, e.g. investment planning problems, control of water systems or energy systems. The size of a two-stage stochastic programming problem grows very quickly with the number of events (scenarios) incorporated into the model and general optimization solvers may not be able to handle problems with realistic size, which has motivated the development of decomposition methods as the only effective alternative. For more information about the structure and properties of the two-stage models we refer to [27, 29]. Decomposition approaches and numerical methods for solving two-and multi-stage problems are discussed in [25].

We consider a two-stage stochastic problems of network capacity expansion, which is described in [27, Example 4, p.14]. Given a network of available routes, the problem consists in allocating proper capacities to arcs at the first stage before observing a random demand for traffic on the network. At the second stage, a shipment plan is determined utilizing the available network capacity so that the demand is satisfied. The problem objective is to plan the optimal shipment routes and also allocate capacities to arcs in a cost efficient manner.

Consider a directed graph with node set $\mathscr{N}$ and arc set $\mathscr{A}$. The capacity of each arc $a \in \mathscr{A}$ is a first-stage decision variable designated by x_a. There is a cost c_a for installing a unit of capacity on arc a.

For each pair of nodes $(m,n) \in \mathscr{N} \times \mathscr{N}$, we observe a random demand D^{mn} for shipments from m to n. We denote the shipment from m to n sent through arc a by y_a^{mn}, which is a part of the second stage decisions. The unit cost for shipments on each arc a is denoted by q_a. Our objective is to assign arc capacities in such a way that the expected total cost of capacity expansion and future shipping cost in a period of time is minimized. For each node $i \in \mathscr{N}$ denote by $\mathscr{A}_-(i) \subseteq \mathscr{A}$ and $\mathscr{A}_+(i) \subseteq \mathscr{A}$ the sets of incoming and outgoing arcs for this node, respectively. The second stage problem is the following multicommodity network flow problem

$$
\begin{aligned}
\min \quad & \sum_{m,n \in \mathscr{N}} \sum_{a \in \mathscr{A}} q_a y_a^{mn} \\
\text{subject to} \quad & \sum_{a \in \mathscr{A}_+(i)} y_a^{mn} - \sum_{a \in \mathscr{A}_-(i)} y_a^{mn} = \begin{cases} D^{mn}, & \text{if} \quad i = m, \\ -D^{mn}, & \text{if} \quad i = n, \\ 0, & \text{otherwise}, \end{cases} \\
& \sum_{m,n \in \mathscr{N}} y_a^{mn} \le x_a, \quad \forall\, a \in \mathscr{A} \\
& y_a^{mn} \ge 0, \quad \forall\, a \in \mathscr{A},\ i,m,n \in \mathscr{N}
\end{aligned}
\tag{49}
$$

Denote the optimal value of (49) as $Q(\mathbf{x},\mathbf{D})$, where $\mathbf{x},\mathbf{D}$ are the vectors of all capacity allocations and demands, respectively. The first stage problem has the form

$$
\min_{\mathbf{x} \ge 0} \sum_{a \in \mathscr{A}} c_a x_a + \mathbb{E}\left[Q(\mathbf{x},\mathbf{D})\right].
$$

In [27], the size of such a model is calculated as follows. If the number of nodes is N, the demand vector has $N(N-1)$ components. If each of these components has R possible realizations, which are independent, we include $K = R^{N(N-1)}$ scenarios into the problem. For each scenario, the second-stage decision has $N(N-1)|\mathscr{A}|$ components and the second stage problem has $N^2(N-1) + |\mathscr{A}|$ not counting the nonnegativity constraints. Therefore, the large scale linear programming formulation has $|\mathscr{A}| + N(N-1)|\mathscr{A}|R^{N(N-1)}$ variables and $(N^2(N-1) + |\mathscr{A}|)R^{N(N-1)}$ constraints.

The dual decomposition methods in stochastic optimization replace the first stage decision vector $\mathbf{x}$ by R vectors x_k (one for each scenario realization) and introduce additional constraints ensuring that the first stage decision variables do not depend on the second stage realizations of the random demand data. We obtain the linear optimization problem

$$\begin{aligned}
\min_{\{\mathbf{x}_r,\mathbf{y}_r\}_{r=1}^R} \quad & \sum_{r=1}^R p_r \left(\sum_{a\in\mathscr{A}} c_a x_a^r + \sum_{m,n\in\mathscr{N}} \sum_{a\in\mathscr{A}} q_a y_a^{r,mn} \right) \\
\text{subject to} \quad & \sum_{a\in\mathscr{A}_+(i)} y_a^{r,mn} - \sum_{a\in\mathscr{A}_-(i)} y_a^{r,mn} = \begin{cases} D^{r,mn}, & \text{if} \quad i=m, \\ -D^{r,mn}, & \text{if} \quad i=n, \\ 0, & \text{otherwise}, \end{cases} \\
& \sum_{m,n\in\mathscr{N}} y_a^{r,mn} \le x_a^r, \quad \forall\, a\in\mathscr{A},\ r=1,\dots,R \\
& y_a^{r,mn} \ge 0, \quad \forall\, a\in\mathscr{A},\ i,m,n\in\mathscr{N},\ r=1,\dots,R \\
& \sum_{r=1}^R A_r \mathbf{x}^r = 0.
\end{aligned} \tag{50}$$

where p_r denotes the probability of scenario r and $\mathbf{x}^r, \mathbf{y}^r \in \mathbb{R}^{|\mathscr{A}|}$ denote the capacity and flow decision vectors for scenario r. The nonanticipativity constraint $\sum_{r=1}^R A_r \mathbf{x}^r = 0$ ensures equality among the capacity decisions of all scenarios and can be expressed in various ways, one possibility being

$$\mathbf{x}^r = \mathbf{x}^s, \quad \forall\, 1 \le r < s \le R,$$

or another

$$\mathbf{x}^r = \mathbf{x}^{r+1}, \quad \forall\, 1 \le r \le R-1. \tag{51}$$

After assigning Lagrange multipliers to the nonanticipativity constraints, the problem splits into scenario subproblems. Note that, we do not necessarily need to decompose the original problem (50) into R subproblems in total. Instead, each subproblem can include a set of scenarios. In [24], the formulation (51) was adopted and DQA was applied to obtain a dual decomposition method for two- and multi-stage problems. In [23], the nonanticipativity constrants are expressed as follows

$$\mathbf{x}^r = \frac{1}{R} \sum_{s=1}^R \mathbf{x}^s,$$

and the ASM method is applied for solving the two-stage problem. The decomposition method constructed in this way is known in the area of stochastic programming as progressive hedging. We applied ADAL, ASM and DQA on a two-stage network capacity expansion problem with 50 nodes, 10 source-sink pairs and 200 demand realizations. The sources were positioned one network diameter apart from their respective sinks, in order to prevent trivial setups. Note that solving the LP problem (50) using CPLEX was not possible for this setup, since we ran out of memory on a 48GB computer. The distributed algorithms were implemented after we decomposed the problem into 10 subproblems, each one involved with 20 demand realizations. The nonanticipativity constraints were of the form $\mathbf{x}^r = \mathbf{x}^{r+1}, \quad \forall\, 1 \le r \le R-1$. The convergence results for the three methods are depicted in Fig. 11. We observe that, for the problem considered here, ADAL and ASM exhibit similar behavior, with ADAL converging slightly faster to better constraint violation accuracies.

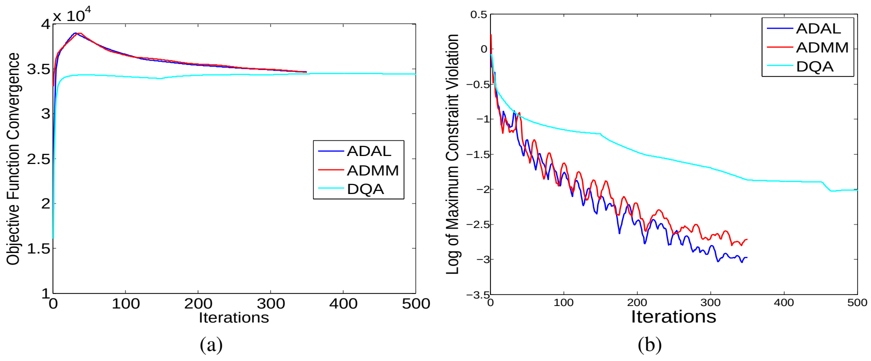


Fig. 11 Comparative convergence results between the ADAL, ASM and DQA methods for a two-stage network capacity expansion problem with 50 nodes, 10 source-sink pairs and 200 demand realizations: a) Objective value convergence and b) Maximum constraint violation.

On the other hand, DQA appears to converge faster to a reasonable value of the objective function, but is slower to constraint violation convergence.

6 Acknowledgements

The authors would like to thank the two anonymous referees whose comments helped improve the paper.

References

1. Berge, C.: Espaces Topologiques Functions Multivoques Dunod. Paris (1959)
2. Berger, A.J., Mulvey, J.M., Ruszczyński, A.: An extension of the DQA algorithm to convex stochastic programs. SIAM J. Optim. **4**(4), 735–753 (1994)
3. Bertsekas, D.: Extended monotropic programming and duality. Journal of Optimization Theory and Applications **139**(2), 209–225 (2008)
4. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Athena Scientific (1997)
5. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning **3**(1), 1–122 (2011)
6. Chen, G., Teboulle, M.: A proximal-based decomposition method for convex minimization problems. Mathematical Programming **64**, 81–101 (1994)
7. Eckstein, J.: The alternating step method for monotropic programming on the connection machine CM-2. ORSA Journal on Computing **5**(1), 84 (1993)
8. Eckstein, J.: Augmented Lagrangian and Alternating Direction methods for convex optimization: A tutorial and some illustrative computational results. Rutcor Research Report, Rutgers (2012)
9. Eckstein, J., Bertsekas, D.P.: An alternating direction method for linear programming. LIDS, MIT (1990)
10. Eckstein, J., Bertsekas, D.P.: On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. Mathematical Programming, **55**, 293–318 (1992)
11. Fortin, M., Glowinski, R.: Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems. North-Holland, Amsterdam (1983)
12. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximations. Computers and Mathematics with Applications **2**, 17–40 (1976)
13. Hestenes, M.: Multiplier and gradient methods. Journal of Optimization Theory and Applications **4**, 303–320 (1969)

14. Kiwiel, K.C., Rosa, C.H., Ruszczynski, A.: Proximal decomposition via alternating linearization. SIAM J. on Optimization **9**(3), 668–689 (1999)
15. Mulvey, J., Ruszczyński, A.: A diagonal quadratic approximation method for large scale linear programs. Operations Research Letters **12**, 205–215 (1992)
16. Powell, M.J.D.: A method for nonlinear constraints in minimization problems. Optimization, Academic press, London (1969)
17. Rockafellar, R.: Augmented Lagrange multiplier functions and duality in nonconvex programming. SIAM Journal on Control **12**, 268–285 (1973)
18. Rockafellar, R.: Augmented lagrangians and applications of the proximal point algorithm in convex programming. Mathematics of Opererations Research pp. 97–116 (1976)
19. Rockafellar, R.: Network Flows and Monotropic Optimization. Wiley-Interscience, New York (1984)
20. Rockafellar, R.: Monotropic programming: A generalization of linear programming and network programming. In: Convexity and Duality in Optimization, *Lecture Notes in Economics and Mathematical Systems*, vol. 256, pp. 10–36. Springer Berlin Heidelberg (1985)
21. Rockafellar, R.T.: Augmented lagrangians and applications of the proximal point algorithm in convex programming. Mathematics of Operations Research **1**, 97–116 (1976)
22. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. SIAM Journal on Control and Optimization **14**, 877–898 (1976)
23. Rockafellar, R.T., Wets, R.B.: Scenarios and policy aggregation in optimization under uncertainty. Mathematics of Operations Research **16**, 1–23 (1991)
24. Ruszczyński, A.: On convergence of an Augmented Lagrangian decomposition method for sparse convex optimization. Mathematics of Operations Research **20**, 634–656 (1995)
25. Ruszczyński, A.: Decompostion methods. In: A. Ruszczyński, A. Shapiro (eds.) Stochastic Programming, Handbooks in Operations Research and Management Science, pp. 141–211. Elsevier, Amsterdam (2003)
26. Ruszczyński, A.: Nonlinear Optimization. Princeton University Press, Princeton, NJ, USA (2006)
27. Ruszczyński A., S.A.: Optimality and duality in stochastic programming. In: S.A. Ruszczyński A. (ed.) Stochastic Programming, Handbooks in Operations Research and Management Science, pp. 65–139. Elsevier, Amsterdam (2003)
28. Shakkottai, S., Srikant, R.: Network optimization and control. Found. Trends Netw. **2**(3), 271–379 (2007)
29. Shapiro A. Dentcheva D., R.A.: Lectures on Stochastic Programming. MPS-SIAM Series on Optimization. MPS-SIAM (2009)
30. Stephanopoulos, G., Westerberg, A.W.: The use of hestenes method of multipliers to resolve dual gaps in engineering system optimization. Journal of Optimization Theory and Applications **15**, 285–309 (1975)
31. Tatjewski, P.: New dual-type decomposition algorithm for nonconvex separable optimization problems. Automatica **25**(2), 233–242 (1989)
32. Watanabe, N., Nishimura, Y., Matsubara, M.: Decomposition in large system optimization using the method of multipliers. Journal of Optimization Theory and Applications **25**(2), 181–193 (1978)