BI-ALTERNATING DIRECTION METHOD OF MULTIPLIERS

Guoqiang Zhang and Richard Heusdens

Department of Intelligent Systems
Delft University of Technology
Delft, the Netherlands
Email: {g.zhang-1,r.heusdens}@tudelft.nl

ABSTRACT

The alternating-direction method of multipliers (ADMM) has been widely applied in the field of distributed optimization and statistic learning. ADMM iteratively approaches the saddle point of an augmented Lagrangian function by performing three updates per-iteration. In this paper, we propose a bi-alternating direction method of multipliers (BiADMM) that iteratively minimizes an augmented bi-conjugate function. As a result, the convergence of BiADMM is naturally established. Unlike ADMM that always involves three updates per iteration, BiADMM opens up an avenue to perform either two or three updates per iteration, depending on the functional construction. As an application, we consider applying BiADMM for the lasso problem. Experimental results demonstrate the effectiveness of our new method.

Index Terms— Distributed optimization, Alternating Direction Method of Multipliers, Bi-Alternating Direction of Multipliers

1. INTRODUCTION

Consider a decomposable optimization problem with a linear constraint

$$\min_{x,z} \quad f(x) + g(z) \quad \text{subject to} \quad Mx = z, \tag{1}$$

where $f : \mathbb{R}^{n_1} \to \mathbb{R}\bigcup\{\infty\}$ and $g : \mathbb{R}^{n_2} \to \mathbb{R}\bigcup\{\infty\}$ are convex functions, and M is an $n_1 \times n_2$ matrix. In recent years, the above problem-formulation has found many applications in distributed optimization and statistic learning [1], such as network resource allocation [2], compressive sensing [3], channel coding [4]. The research challenge is how to efficiently reaches the optimal solution of (1) by exploiting the decomposable structure of the objective function.

In the literature, the dual-ascent method, proposed in the mid-1960s [5, 6, 7], is a classic approach for solving (1). The method iteratively approaches the saddle point of a Lagrangian function by alternatively updating the primal variables (x, z) in (1) and the Lagrangian multipliers. However, the convergence of the dual-ascent method requires strong assumptions on the objective function [8, 1] like strong convexity of $f(x)$ and $g(z)$, making it less popular.

In order to bring robustness to the dual-ascent method, the alternating direction method of multipliers (ADMM) was developed in the mid-1970s [9, 10]. Since then, the properties of ADMM have been well studied in a series of papers [11, 12, 13, 14]. A thorough review on ADMM has been provided in [1] by Boyd et al. ADMM considers an augmented Lagrangian function where a quadratic penalty function $\|Mx - z\|^2$ is introduced. At each iteration, ADMM always involve three updates, two coordinate-descent operations for (x, z) and one gradient-descent operation for the Lagrangian multipliers. It was shown that ADMM possesses a guaranteed convergence under very mild conditions [1]. Due to its simplicity and robustness, ADMM has been widely applied in distributed optimization and statistic learning [1]. Recently, it has been shown that ADMM possesses a linear convergence rate [15].

2. STATEMENT

We note that ADMM attempts to reach the saddle point of the augmented Lagrangian function, which is interpreted as a minmax problem. One natural question is if we can build an unconstrained minimization problem that is equivalent to (1). That is the unconstrained problem implicitly tackles the linear constraint in (1). In general, it is relatively easy to solve a minimization problem over a minmax problem. Our primary motivation is to provide an alterative framework to compute the optimal solution of (1) in a distributed fashion. Our research may shed light on how to solve more complicated problems with linear constraints.

Formally, in this paper, we reconsider to solve the decomposable problem (1). Inspired by the Fenchel's duality, we first construct an augmented bi-conjugate function. The new function involves $(f(x), g(z))$ and also their conjugates [16]. We show that the new function is lower bounded by zero. Further, the lower bound zero is achieved if and only if an optimal solution of (1) is reached.

We propose a bi-alternating direction method of multipliers (BiADMM) to iteratively minimize the augmented bi-conjugate function. At each iteration, BiADMM may involve two or three updates depending on the form of the augmented bi-conjugate function. Therefore, BiADMM is more flexible than ADMM w.r.t. the number of updates. The convergence of BiADMM is naturally established due to our functional construction. As an application, we consider solving the lasso problem by using BiADMM. Experimental results demonstrate the effectiveness of BiADMM as compared to ADMM.

3. AUGMENTED BI-CONJUGATE FUNCTION

In this section, we explain how to transform the problem (1) into an equivalent minimization problem with no explicit constraint. To achieve this goal, we make use of the conjugate of a function [16].

3.1. The original functions and their conjugate

We consider the problem (1) where the two functions $f(x)$ and $g(z)$ are closed, proper and convex functions. The Lagrangian function associated with (1) is defined by

$$L(x, z, \delta) = f(x) + g(z) + \delta^\top (Mx - z), \tag{2}$$

where δ is the Lagrangian multiplier. The Lagrangian function can be viewed as a convex function over (x, z) with fixed δ, and a concave function over δ with fixed (x, z). Through the rest of the paper, we make the following assumption.

Assumption 1. *There exists a saddle point (x^*, z^*, δ^*) to the Lagrangian function $L(x, z, \delta)$ such that for all $(x, z) \in \mathbb{R}^{n_1 \times n_2}$ and $\delta \in \mathbb{R}^{n_2}$ we have*

$$L(x^*, z^*, \delta) \leq L(x^*, z^*, \delta^*) \leq L(x, z, \delta^*).$$

Next we introduce the conjugate function of $(f(x), g(z))$. In particular, the conjugate of $f(x)$ and $g(z)$ are defined as [16]

$$\begin{aligned} f^*(\lambda) &= \max_x \lambda^\top x - f(x) & (3)\\ g^*(\delta) &= \max_z \delta^\top z - g(z). & (4) \end{aligned}$$

$f^*(\lambda)$ and $g^*(\delta)$ are always convex functions irrespective of $f(x)$ and $g(z)$. We note that the conjugate of $f^*(\lambda)$ and $g^*(\delta)$ yields the original function $f(x)$ and $g(z)$, respectively, since $(f(x), g(z))$ are closed, proper and convex functions [16]. In other words, $(f(x), g(z))$ and $(f^*(\lambda), g^*(\delta)))$ are conjugate to each other.

Given the conjugate functions (3)-(4), Fenchel's inequality states that for all $(x, \lambda) \in \mathbb{R}^{n_1 \times n_1}$ and $(z, \delta) \in \mathbb{R}^{n_2 \times n_2}$, there is

$$\begin{aligned} f(x) + f^*(\lambda) - \lambda^\top x \geq 0 & \quad (5)\\ g(z) + g^*(\delta) - \delta^\top z \geq 0, & \quad (6) \end{aligned}$$

where the equality holds when $0 \in \lambda - \nabla f(x)$ and $0 \in \delta - \nabla g(z)$. To clarify, $\nabla f(x)$ and $\nabla g(z)$ denotes the sub-differential of $f(x)$ and $g(x)$, respectively.

3.2. Function construction

With the conjugate functions $f^*(\lambda)$ and $g^*(\delta)$, we define the augmented bi-conjugate function as

$$\begin{aligned} & L_\Phi(x, z, \delta, \lambda)\\ = \;& f(x) + g(z) + \frac{\rho}{2}\|Mx - z\|_2^2 + f^*(\lambda) + g^*(\delta)\\ & + \frac{\eta}{2}\|M^\top \delta + \lambda\|_2^2 - \lambda^\top x - \delta^\top z, \end{aligned} \tag{7}$$

where the two parameters $\rho, \eta > 0$ and $\Phi = \{\rho, \eta\}$. The two quadratic penalty functions in (7) are introduced in order to enforce the equality constraints $z = Mx$ and $M^\top \delta = -\lambda$. The name *bi-conjugate* comes from the fact that $(f(x), g(z))$ and $(f^*(\lambda), g^*(\delta)))$ are conjugate to each other. Fig. 1 visualizes the relationship of the variables (x, z, δ, λ) in the function $L_\Phi(x, z, \delta, \lambda)$.

Next we study the properties of the function in (7), which is summarized in a theorem below:

Theorem 1. *The function $L_\Phi(x, z, \delta, \lambda)$ is lower bounded by zero for all x, z, δ and λ. Further, the lower bound zero is achieved if and only if an optimal solution of (1) is reached.*

Proof. The proof for the zero lower-bound is trivial. By applying the Fenchel's inequality (5)-(6), one can immediately show that $L_\Phi(x, z, \delta, \lambda)$ is lower bounded by zero.

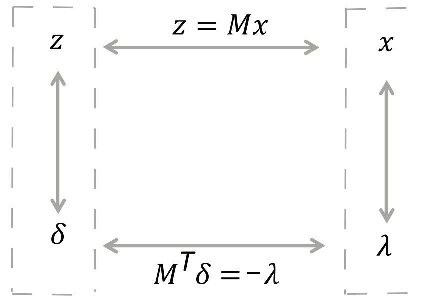

Fig. 1. The relationship of the variables (x, z, δ, λ) in (7). z and δ are related by the quantity $\delta^\top z$. On the other hand, x and λ are related by the quantity $\lambda^\top x$.

Next we show that an optimal solution of (1) leads to the lower-bound zero of the augmented bi-conjugate function. Suppose (x^*, z^*, δ^*) is an optimal solution. Then there must exist a δ^* such that

$$\begin{aligned} 0 &\in \nabla f(x^*) + M^\top \delta^*\\ 0 &\in \nabla g(z^*) - \delta^*\\ 0 &= z^* - Mx^*. \end{aligned}$$

In this situation, if we let $\lambda^* = -M^\top \delta^*$ in (7), we obtain $L_\Phi(x^*, z^*, \delta^*, \lambda^*) = 0$.

Conversely, if the augmented bi-conjugate function equals to zero, we must have $Mx = z$ and $M^\top \delta = -\lambda$. Further, from (5)-(6), the two optimality conditions $0 \in \lambda - \nabla f(x)$ and $0 \in \delta - \nabla g(z)$ hold. As a result, we obtain an optimal solution to the problem (1). □

Theorem 1 implies that one can minimize the augmented bi-conjugate function to reach the optimal solution of (1). Further, the optimal solution is certified if the function value equals to zero. The above property is highly valuable in practice as it can greatly simplify the design of termination criterion for an optimization method.

Remark 1. *It is worth noting that the augmented Lagrangian function for designing ADMM (see [1]) only introduces one auxiliary variable δ. ADMM essentially solves a minmax problem. On the other hand, the function $L_\Phi(x, z, \delta, \lambda)$ introduces two auxiliary variables δ and λ. The additional variable λ makes it possible to convert (1) into an unconstrained minimization problem. Further, we notice that the augmented bi-conjugate function $L_\Phi(x, z, \delta, \lambda)$ has two free parameters ρ and η while the augmented Lagrangian function for ADMM has one free parameter.*

Remark 2. *We note that the form of the augmented bi-conjugate function is not unique. One can construct other functions besides (7). For instance, one can introduce a replica $\bar{x}$ of x with the constraint $\bar{x} = x$ or a replica $\bar{\delta}$ of δ with the constraint $\bar{\delta} = \delta$. Correspondingly, the quadratic penalty functions have to adapted to tackle the new linear constraints. In some applications, one particular form of the function might be favorable over others. In this paper, we focus on (7) as an example.*

4. THE BI-ALTERNATING DIRECTION METHOD OF MULTIPLIERS

In this section, we first consider the situation that the two conjugate functions $f^*(\lambda)$ and $g^*(\delta)$ can be computed easily from (3)-(4). In this situation, the alternative minimization of the augmented bi-conjugate function over the variables arises naturally.

We also consider the situation that $f^*(\lambda)$ and/or $g^*(\delta)$ cannot be obtained easily. We then extend the augmented bi-conjugate function by incorporating feedback from last iteration in computing new estimates.

4.1. Basic bi-alternating direction updating

In this subsection, we assume that the two conjugate functions $f^*(\lambda)$ or $g^*(\delta)$ can be easily computed. This facilitates the minimization of $L_\Phi(x,z,\delta,\lambda)$ over δ or λ by simple computation.

We minimize the function $L_\Phi(x,z,\delta,\lambda)$ by performing Gauss-Seidel iteration. Each time we minimize the function over some variables while keeping all the others freezing. At each iteration, every variable receives a new estimate. Note that the function $L_\Phi(x,z,\delta,\lambda)$ is convex over (x,δ) if (z,λ) are fixed. Conversely, the function is also convex over (z,λ) if (x,δ) are fixed. One natural scheme for updating the estimate is as follows

$$(\hat{x}^{(k+1)},\hat{\delta}^{(k+1)}) \;=\; \arg\min_{x,\delta} L_\Phi(x,\hat{z}^{(k)},\delta,\hat{\lambda}^{(k)}) \tag{8}$$

$$(\hat{z}^{(k+1)},\hat{\lambda}^{(k+1)}) \;=\; \arg\min_{z,\lambda} L_\Phi(\hat{x}^{(k+1)},z,\hat{\delta}^{(k+1)},\lambda), \tag{9}$$

where $k \geq 0$ denotes the number of iterations. The iteration stops when the function value equals to zero.

Depending on the M matrix and the parameter set Φ, we can also come up with alternative schemes to update the estimate. For instance, when M is a square nonsingular matrix and the parameters (ρ,η) are large enough, the function $L_\Phi(x,z,\delta,\lambda)$ is convex over (x,λ) if (z,δ) are fixed. Conversely, the function is also convex over (z,δ) if (x,λ) are fixed. In this situation, the estimate at kth iteration can be updated as

$$(\hat{x}^{(k+1)},\hat{\lambda}^{(k+1)}) \;=\; \arg\min_{x,\lambda} L_\Phi(x,\hat{z}^{(k)},\hat{\delta}^{(k)},\lambda) \tag{10}$$

$$(\hat{z}^{(k+1)},\hat{\delta}^{(k+1)}) \;=\; \arg\min_{z,\delta} L_\Phi(\hat{x}^{(k+1)},z,\delta,\hat{\lambda}^{(k+1)}). \tag{11}$$

Similarly, one can also consider the case that M is of rank n_1 or n_2. We will omit the details here.

Remark 3. *We note that the updating scheme (8)-(9) or (10)-(11) only involves coordinate-descent operations. This is different from that of ADMM which involves both gradient-descent and coordinate-descent operations.*

4.2. Extended bi-alternating direction updating

We note that for some function $f(x)$ (or $g(z)$), it may be difficult to compute its conjugate, making the updating scheme (8)-(9) or (10)-(11) time-consuming. However, the conjugate of $f(x)$ (or $g(z)$) coupled with a quadratic function may take a simple form. One such example is the function $\|Ax-b\|_2^2$ where the matrix A has more columns than rows (see Section 5 for the lasso problem). We consider the above case in this subsection.

At iteration k, we extend the function $L_\Phi(x,z,\delta,\lambda)$ in (1) by incorporating the estimate $\hat{x}^{(k)}$ and $\hat{z}^{(k)}$, which is expressed as

$$\begin{aligned} & L_\Psi^{(k)}(x,z,\delta,\lambda) \\ = \;& f(x)+\frac{\beta_x}{2}\|x-\hat{x}^{(k)}\|_2^2+g(z)+\frac{\beta_x}{2}\|z-\hat{z}^{(k)}\|_2^2 \\ & +\frac{\rho}{2}\|Mx-z\|_2^2+\bar{f}^{*(k)}(\lambda)+\bar{g}^{*(k)}(\delta) \\ & +\frac{\eta}{2}\|M^\top\delta+\lambda\|_2^2-\lambda^\top x-\delta^\top z, \end{aligned} \tag{12}$$

where $\Psi=\{\rho,\eta,\beta_x,\beta_z\}$, and $\bar{f}^{*(k)}(\lambda)$ and $\bar{g}^{*(k)}(\delta)$ are defined as

$$\bar{f}^{*(k)}(\lambda) \;=\; \max_x\left(\lambda^\top x-f(x)-\frac{\beta_x}{2}\|x-\hat{x}^{(k)}\|_2^2\right), \tag{13}$$

$$\bar{g}^{*(k)}(\delta) \;=\; \max_z\left(\delta^\top z-g(z)-\frac{\beta_x}{2}\|z-\hat{z}^{(k)}\|_2^2\right). \tag{14}$$

For the special case that $\beta_x=\beta_z=0$, $(\bar{f}^{*(k)}(\lambda),\bar{g}^{*(k)}(\delta))$ reduces to $(f^*(\lambda),g^*(\delta))$. We assume that $\bar{f}^{*(k)}(\lambda)$ and $\bar{g}^{*(k)}(\delta)$ can be easily computed due to the quadratic functions. In practice, we have found the functions in some classic problems have such a property, like the basis pursuit problem and the lasso problem.

Similarly, we minimize the new function $L_\Psi^{(k)}(x,z,\delta,\lambda)$ by performing Gauss-Seidel iteration. One natural updating scheme is to adapt (8)-(9) for the new function. One can also come up other schemes by making use of the parameter set Ψ. The key point is to make the function value decreasing when updating the estimate for the variables.

5. EXPERIMENTAL RESULTS

We first tested BiADMM for a toy example. Our primary motivation was to show that there is a problem of the form (1) for which BiADMM is much more efficient than ADMM in terms of the convergence rate. In ADMM, there is one free parameter to be specified, which we denote as γ. The parameter γ actually corresponds to ρ in BiADMM.

Also, we tested BiADMM for the classic lasso problem. For this problem, we constructed and minimized the function $L_\Psi^{(k)}(x,z,\delta,\lambda)$ in (12) instead. ADMM was again taken as a reference method for performance comparison.

5.1. A toy example

In the first experiment, we considered a simple problem where $f(x)$ and $g(z)$ in (1) are scalar quadratic functions. Specifically, the two functions take the form as

$$f(x)=\frac{1}{2}x^2-ax \tag{15}$$

$$g(z)=\frac{1}{2}z^2-bz, \tag{16}$$

where (x,z) satisfies the linear constraint $x=z$. It is immediate that $x^*=z^*=(a+b)/2$. For the quadratic functions in (15)-(16), the corresponding augmented bi-conjugate function is convex over (x,z,δ,λ). Despite the simplicity of the two functions, we applied BiADMM and ADMM to solve the problem.

In the implementation of BiADMM, we followed the updating scheme (10)-(11) since $L_\Phi(x,z,\delta,\lambda)$ is a convex function for any (ρ,η). For simplicity, we set $\eta=\rho$. On the other hand, we implemented ADMM by following [1]. To make a fair comparison, the initial value of $(\hat{x}^{(0)}, \hat{z}^{(0)}, \hat{\delta}^{(0)})$ for the two methods were set to be identical (For ADMM, there is no variable λ). To terminate the iteration of the two methods, the estimate-difference $|\hat{x}-\hat{z}|$ was measured. The convergence threshold were set as 10^{-5}.

The convergence results of the two methods are displayed in Table 1 for a set of ρ (or equivalently, γ) values. It is seen from the table that the minimum number of iterations required for BiADMM is 4. On the other hand, the minimum number of iterations for ADMM is 16. This suggests that BiADMM is able to enhance the information-exchange between the two variables x and z by choosing the parameters properly. The main reason behind it might be that the auxiliary variable λ in BiADMM provides more freedom to convey information between x and z.

$\rho = \gamma$	0.01	0.05	0.1	0.5	1	5	10	50
ADMM	657	137	72	22	16	28	43	116
BiADMM	4	5	5	9	12	34	58	225

Table 1. Number of iterations of the two methods for the toy example specified by (15)-(16). The parameter γ is from ADMM. The parameters (a, b) in the two functions of (15)-(16) were set as $a = -1$ and $b = 4$.

5.2. The lasso problem

In the second experiment, we considered solving the lasso problem by using BiADMM. The lasso problem is originated from bioinformatics and machine learning, which is expressed as

$$\min_{x,z} \frac{\alpha}{2}\|Ax-b\|_2^2 + \|z\|_1 \quad \text{subject to} \quad x = z, \tag{17}$$

where A is a $m \times n$ matrix ($n > m$), and $\alpha > 0$ is a scalar regularization parameter. We note that the conjugate of $\|z\|_1$ takes a simple form. However, the conjugate of $\frac{\alpha}{2}\|Ax-b\|_2^2$ is rather difficult to compute due to the fact that $n > m$. To tackle this issue, we compute the conjugate of $\frac{\alpha}{2}\|Ax-b\|_2^2$ coupled with a quadratic function as defined in (13).

Formally, by setting $(\beta_x > 0, \beta_z = 0)$ in (12), we construct the augmented bi-conjugate function as

$$\begin{aligned} & L_{\Psi}^{(k)}(x, z, \delta, \lambda) \\ &= \frac{\alpha}{2}\|Ax-b\|_2^2 + \frac{\beta_x}{2}\|x-\hat{x}^{(k)}\|_2^2 + \|z\|_1 + \frac{\rho}{2}\|x-z\|_2^2 \\ &\quad + \bar{f}^{*(k)}(\lambda) + 1_{\{-1 \preceq \delta \preceq 1\}} + \frac{\eta}{2}\|\lambda+\delta\|_2^2 - \lambda^\top x - \delta^\top z. \end{aligned} \tag{18}$$

where the indicator function $1_{\{-1 \preceq \delta \preceq 1\}}$ is in fact the conjugate of $\|z\|_1$. We note that the function in (12) is convex over (x, λ) if (z, δ) are fixed. By using the above property, we alternatively minimized the function in the experiment as

$$\begin{aligned} (\hat{x}^{(k+1)}, \hat{\lambda}^{(k+1)}) &= \arg\min_{x,\lambda} L_{\rho}^{(k)}(x, \hat{z}^{(k)}, \hat{\delta}^{(k)}, \lambda) \\ \hat{z}^{(k+1)} &= \arg\min_{z} L_{\rho}^{(k)}(\hat{x}^{(k+1)}, z, \hat{\delta}^{(k)}, \hat{\lambda}^{(k+1)}) \\ \hat{\delta}^{(k+1)} &= \arg\min_{\delta} L_{\rho}^{(k)}(\hat{x}^{(k+1)}, \hat{z}^{(k+1)}, \delta, \hat{\lambda}^{(k+1)}), \end{aligned}$$

where $k \geq 0$. The above BiADMM scheme involves three updates per-iteration, which is the same as ADMM.

In the experiment, we set $(m, n) = (60, 100)$ and $\alpha = 0.9$. The elements in (A, b) were generated randomly from normal Gaussian distribution. In the implementation of BiADMM, we let $(\beta_x, \rho, \eta) = (5, 1, 0.005)$ in (12). To a make a fair comparison, we also chose $\gamma = 1$ in ADMM. Except the above parameters, the experiment-setup is the same as for the toy example.

The convergence results are displayed in Table 2. Seven pairs of (A, b) for the lasso problem were generated and tested by the two methods. It is seen that the convergence speed of the two methods is comparable. In some cases, BiADMM outperforms ADMM considerably.

	1	2	3	4	5	6	7
ADMM	395	407	515	649	535	401	435
BiADMM	230	232	538	348	294	377	484

Table 2. Number of iterations of the two methods for seven realizations of the lasso problem (17).

6. CONCLUSION

We have proposed a new distributed computational framework for solving the decomposable optimization problem (1) with linear constraint. Namely, we have transformed the original problem into an unconstrained minimization problem. The augmented bi-conjugate function to be minimized implicitly tackles the linear constraint. While ADMM iteratively solves a minmax problem, BiADMM we have proposed iteratively minimizes the augmented bi-conjugate function. The key point in constructing the augmented bi-conjugate function is to introduce one more auxiliary variable λ compared to the function in the minmax problem.

We have evaluated BiADMM and ADMM for a toy example. The experimental results suggest that the auxiliary variable λ may be helpful to accelerate the convergence rate. Finally we have successfully applied BiADMM in solving the lasso problem, of which the performance is comparable to that of ADMM.

7. REFERENCES

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *In Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[2] D. P. Palomar and M. Chiang, "A Tutorial on Decomposition Methods for Network Utility Maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, 2006.

[3] Joo F. C. Mota, Joo M. F. Xavier, P. M. Q. Aguiar, and M. Pschel, "Distributed Basis Pursuit," *IEEE Trans. on Signal Processing*, vol. 60, no. 4, pp. 1942–1956, 2012.

[4] S. Barman, X. Liu, S. Draper, and B. Recht, "Decomposition methods for large scale LP decoding," in Arxiv, 2012.

[5] G. B. Dantzig., *Linear programming and extensions*, RAND Corporation, 1963.

[6] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, no. 3, pp. 399417, 1963.

[7] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, p. 4:238252, 1962.

[8] D. P. Bertsekas and J. N. Tsitsikis, *Parallel and distributed Computation: Numerical Methods*, Belmont, MA: Athena Scientific, 1997.

[9] R. Glowinski and A. Marrocco, "Sur lapproximation par elements finis dordre un, et la resolution par penalisation-dualite dune classe de problems de Dirichlet nonlineaires," *Revue Franccaise d'Automatique, Informatique, et Recherche Operationelle*, vol. 9, pp. 41–76, 1975.

[10] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite-element approximations," *Computers and Mathematics with Applications*, vol. 2, pp. 17–40, 1976.

[11] D. Gabay, "Applications of the method of multipliers to variational inequalities," *in M. Fortin and R. Glowinski, editors, Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems*, 1983.

[12] P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM Journal on Control and Optimization*, vol. 29, no. 1, pp. 119–138, 1991.

[13] R. Glowinski and P. Le Tallec, "Augmented Lagrangian methods for the solution of variational problems," Technical Report 2965, 1987.

[14] J. Eckstein and M. Fukushima, "Some reformulations and applications of the alternating direction method of multipliers," *Large Scale Optimization: State of the Art*, pp. 119–138, 1993.

[15] X. M. Yuan B. S. He and X. M. Xu, "On the O(1/n) convergence rate of DouglasRachford alternating direction method," *SIAM J. Numer. Anal.*, vol. 50, no. 2, pp. 700709, 2012.

[16] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.