PENALTY DUAL DECOMPOSITION METHOD WITH APPLICATION IN SIGNAL PROCESSING

Qingjiang Shi[1,2] *and Mingyi Hong*[1,3]

[1] Dept. of IMSE, Iowa State University, Ames, IA, 50011

[2] School of Info. Sci. & Tech., Zhejiang Sci-Tech University, Hangzhou, China 310018.

[3] Dept. of Elect. & Compt. Eng. Iowa State University, Ames, IA, 50011.

ABSTRACT

Many problems of recent interest in signal processing, machine learning and wireless communications can be posed as nonconvex nonsmooth optimization problems. These problems are generally difficult to solve especially when the optimization variables are nonlinearly coupled in some nonconvex constraints. In this paper, we propose an algorithm named "penalty dual decomposition" (PDD) method, for the minimization of a nonconvex nonsmooth objective subject to nonconvex constraints. We show that the PDD converges to KKT solutions under certain constraint qualification condition. Simulations corroborate the excellent performance of the PDD method.

1. INTRODUCTION

Many important engineering problems arising from signal processing, wireless communications and machine learning can be modeled as nonconvex nonsmooth optimization problems. These problems are generally difficult to solve, especially when the optimization variables are nonlinearly coupled in some nonconvex constraints. This paper aims to provide a general algorithmic framework, that can exploit the problem structure as fully as possible, for the minimization of a nonconvex nonsmooth function subject to certain nonconvex *coupling* constraints.

Nonconvex coupling constraints often arise in contemporary engineering problems [1–4]. For example, in the joint source-relay design of various multiple-input-multiple-output (MIMO) relay systems [5–7], the relay power constraint takes a bi-quadratic form, meaning it is a quaratic constraint in either source or relay precoders; in dictionary learning [8, 9], nonnegative matrix factorization [10–12], and geometry-based blind source separation [13], the data fidelity requirements are often expressed as nonconvex bi-linear equality constraints. Despite their wide applicability, solving problems efficiently with nonconvex coupling constraints is very challenging, because it is difficult to explore their problem structures.

One approach that is often used in the literature to deal with coupling constraints, but often with poor performance, is the alternating optimization (AO) method. The simple idea behind the AO method is to replace difficult joint optimization over all variables with a sequence of simple optimization problems over a subset of variables. For instance, the works [14] and [5] applied the AO method to the joint source-relay design problem, where the source precoder and the relay precoder are coupled each other in SINR constraints or relay power constraint. However, the AO method can only provide feasible solutions in the coupling constraint case and cannot guarantee convergence to stationary solutions unless the problem has some special structure; see for example [15]. In particular, the AO method gets easily trapped in some unexpected points in the equality coupling constraint case.

Another popular approach that can deal with constraint coupling is penalty method [16]. The basic idea of penalty methods is to move the difficult coupling constraints to the objective function as a penalty term, which prescribes a high cost to infeasible points with a suitable penalty parameter. For example, in [2], Kuang *et. al.* used penalty method to approximate the solution of the symmetric nonnegative matrix factorization problem. In [6], Shi *et. al.* used penalty method to solve the joint source-relay design problem for full-duplex MIMO relay systems. The work [17] showed that penalty method can be applied to solve certain rank minimization problem. However, penalty methods could be very inefficient due to the issue of ill-conditioning for large penalty parameter. Alternatively, Augmented Lagrangian (AL) method [18, 19] was proposed to overcome the limitations of penalty methods by introducing an additional dual-related term. In the AL methods, a sequence of AL subproblems (i.e., the problems of minimization of the augmented Lagrangian) needs to be exactly or approximately solved. When the subproblems are easily solvable, the AL methods are attractive as they can be easily implemented (often matrix-free) [20] and have at least local convergence guarantees under relatively mild assumptions [21, 22]. However, the AL subproblems are generally hard to solve especially when they have complicated constraints. Further, most AL methods cannot deal with problems of nonconvex objective functions with nonconvex nonsmooth terms, thus limiting its application in many contemporary signal processing problems involving possibly nonconvex nonsmooth regularizers.

This paper proposes an algorithm named penalty dual decomposition (PDD) method, for the minimization of a nonconvex nonsmooth objective subject to nonconvex constraints. We show that the PDD has convergence to KKT solutions under certain constraint qualification. An application to max-min rate fairness multi-cast beamforming [23] corroborates the excellent performance of the PDD method. More applications will be reported elsewhere.

Notations: Unless otherwise specified, we use uppercase bold letters for matrices, lowercase bold letters for column vectors, and regular letters for scalars. The notation $\mathbb{R}^n/\mathbb{C}^n$ denotes the n-dimensional space of real/complex number. For a vector $\boldsymbol{x}$, $\|\boldsymbol{x}\|$ and $\|\boldsymbol{x}\|_\infty$ denotes Euclidean norm and element-wise infinity norm, respectively. For a scalar function $f(\cdot)$, $f'(\cdot)/\nabla f(\cdot)$ denote its derivative/gradient with respect to its argument. For a multivariate function $f(\boldsymbol{x},\boldsymbol{y})$, $\nabla_{\boldsymbol{x}} f(\boldsymbol{x},\boldsymbol{y})$ denotes its gradient with respect to $\boldsymbol{x}$. For vector functions $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x},\boldsymbol{y})$, $\nabla \boldsymbol{g}(\boldsymbol{x})$ denotes the Jacobian matrix of $\boldsymbol{g}(\boldsymbol{x})$ and $\nabla_{\boldsymbol{x}}\boldsymbol{h}(\boldsymbol{x},\boldsymbol{y})$ denotes the Jacobian matrix of $\boldsymbol{h}(\boldsymbol{x},\boldsymbol{y})$ with respect to $\boldsymbol{x}$. '$\otimes$' denotes Kronecker product. $\boldsymbol{e}_i$ denotes a vector of all zeros except the i-th entry being 1.

2. PROBLEM SETUP AND KKT CHARACTERIZATION

Consider the following problem

$$(P) \qquad \begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{y}} \quad & f(\boldsymbol{x},\boldsymbol{y}) + \sum_{j=1}^{m} \tilde{\phi}(\boldsymbol{y}_j) \\ \text{s.t.} \quad & \boldsymbol{h}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{0}, \\ & \boldsymbol{g}_i(\boldsymbol{x}_i) \le \boldsymbol{0},\ i = 1,\cdots,N \end{aligned} \tag{1}$$

where the optimization variables are given by $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ with $\boldsymbol{x}_i \in \mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ and $\sum_{i=1}^n n_i = N$; $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m) \in \mathbb{R}^M$ and $\boldsymbol{y}_j \in \mathbb{R}^{m_j}$, $j = 1, 2, \ldots, m$, with $\sum_{j=1}^m m_j = M$; $\tilde{\phi}(\boldsymbol{y}_j)$ is a composite function in the form of $\phi_j(s_j(\boldsymbol{y}_j))$; the feasible set $\mathcal{X}$ is the Cartesian product of n simple *closed convex* sets: $\mathcal{X} \triangleq \Pi_{i=1}^n \mathcal{X}_i$ with $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ and $\sum_{i=1}^n n_i = N$; $f(\boldsymbol{x},\boldsymbol{y})$ is a scalar *continuously differentiable* function; for each j, $s_j(\boldsymbol{y}_j)$ is a *convex* but possibly nondifferentiable function while $\phi_j(\boldsymbol{x})$ is a *nondecreasing* and *continuously differentiable* function; for each i, $\boldsymbol{g}_i(\boldsymbol{x}_i) \in \mathbb{R}^{q_i}$ is a vector of q_i *continuously differentiable* functions with $q \triangleq \sum_{i=1}^n q_i$; $\boldsymbol{h}(\boldsymbol{x},\boldsymbol{y}) \in \mathbb{R}^p$ is a vector of p *continuously differentiable* functions. We assume that the feasible set of problem (P) is nonempty and the objective function is regular [24] throughout the rest of this paper.

The term $\tilde{\phi}(\boldsymbol{y}_j)$ could take the form of sparsity promoting function such as the ℓ_1 penalty function, the SCAD penalty or the logarithm function; see [25, TABLE I] for details. Since the term $\sum_{j=1}^{n_y} \tilde{\phi}(\boldsymbol{y}_j)$ could be neither convex nor differentiable, we use generalized gradient/direcgtional derivative [26] to characterize the first-order optimality condition. The following result characterizes the first-order optimality condition of problem (P), under certain *constraint qualification* called the *Robinson's condition* [27, Chapter 3]. The proof is omitted for brevity.

Theorem 2.1 *Let $(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})$ be a local minimum of problem (P). Assume that Robinson's condition holds for problem (P) at $(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})$. Then there exist multipliers $\hat{\boldsymbol{\mu}} \in \mathbb{R}^p$ and $\hat{\boldsymbol{\nu}}_i \in \mathbb{R}^{q_i}$, $j = 1, 2, \ldots, q$ that together with $(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})$ satisfy the following KKT system*

$$\Big(\nabla_{\boldsymbol{x}_i} f(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}) + \nabla_{\boldsymbol{x}_i}\boldsymbol{h}(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})^T\hat{\boldsymbol{\mu}} + \nabla_{\boldsymbol{x}_i}\boldsymbol{g}_i(\hat{\boldsymbol{x}}_i)^T\hat{\boldsymbol{\nu}}_i\Big)^T \times (\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i) \ge 0, \forall \boldsymbol{x}_i \in \mathcal{X}_i, \tag{2a}$$

$$0 \in \bar{\partial}\tilde{\phi}(\boldsymbol{y}_j) + \nabla_{\boldsymbol{y}_j} f(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}) + \nabla_{\boldsymbol{y}_j}\boldsymbol{h}(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}})^T\hat{\boldsymbol{\mu}}, \forall j, \tag{2b}$$

$$(\hat{\boldsymbol{\nu}}_i)^T\boldsymbol{g}_i(\hat{\boldsymbol{x}}_i) = 0,\ \boldsymbol{g}_i(\hat{\boldsymbol{x}}_i) \le 0,\ \hat{\boldsymbol{\nu}}_i \ge 0, \boldsymbol{h}(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}) = 0. \tag{2c}$$

where $\bar{\partial}\tilde{\phi}(\boldsymbol{y}_j)$ denotes the set of generalized gradients [26] of $\bar{\partial}\tilde{\phi}(\cdot)$ at $\boldsymbol{y}_j$.

Remark 2.1 *The Robinson's condition assumed here is a kind of constraint qualification (CQ) condition. Although it is generally hard to check as many CQ conditions, such an condition is standard in constrained optimization literature, e.g., [17, 27–29]. Moreover, it is related to commonly used CQs, such as Mangasarian-Fromovitz constraint qualification (MFCQ) condition, linear independence constraint qualification (LICQ) condition, Slater's condition, etc. See details in [24].*

3. PDD METHOD FOR PROBLEM (P)

In addition to the nonconvexity and nondifferentiability, the constraint coupling due to $\boldsymbol{h}(\boldsymbol{x},\boldsymbol{y}) = 0$ further complicates problem (P). If no such constraints exist, the classical BCD-type algorithms can be applied to decompose problem (P) into a sequence of small-scale problems. This observation motivates us to develop a primal-dual based framework, which dualize the difficult coupling constraint by appropriate penalty function, and use coordinate-decomposition to perform fast computation, hence the name penalty dual decomposition method. In what follows, we present the basic PDD method and its convergence result.

Table 1. Algorithm 1: PDD method for problem (P)

0. initialize $\boldsymbol{z}^0$, $\varrho_0 > 0$, $\boldsymbol{\lambda}_0$, and set $0 < c < 1$, $k = 1$
1. **repeat**
2. $\quad \boldsymbol{z}^k = \mathrm{BSUM}(P_{\varrho_k,\boldsymbol{\lambda}_k}, \tilde{\mathcal{L}}_k, \boldsymbol{z}^{k-1}, \epsilon_k)$
3. $\quad$ **if** $\|\boldsymbol{h}(\boldsymbol{z}^k)\| \le \eta_k$ $\quad$ *// case 1*—AL method
4. $\quad\quad \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \frac{1}{\varrho_k}\boldsymbol{h}(\boldsymbol{z}^k)$
5. $\quad\quad \varrho_{k+1} = \varrho_k$
6. $\quad$ **else** *// case 2*—penalty method
7. $\quad\quad \boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k$
8. $\quad\quad \varrho_{k+1} = c\varrho_k$
9. $\quad$ **end**
10. $\quad k = k + 1$
11. **until** some termination criterion is met

3.1. PDD Method

The basic PDD method is a double-loop algorithm where the inner loop is to approximately solve an augmented Lagrangian (AL) subproblem while the outer loop is to update the dual variable or the penalty parameter in terms of the constraint violation [30]. Specifically, we update the dual variable $\boldsymbol{\lambda}_k$ using dual ascend method [31] when the constraint violation is small (i.e., Step 4); otherwise decrease the penalty parameter ϱ_k (i.e., Step 8). Moreover, to exploit the problem structure as fully as possible, we use the BSUM algorithm [32], which is a generalized version of block coordinate descent (BCD) method [16], to solve the AL subproblem. In the BSUM algorithm, each time one block variable is selected to be updated while fixing the others by minimizing a locally tight upper bound of the AL. Thus the PDD method perform adaptive switching between the AL method and penalty method. This adaptive strategy is expected to find an appropriately large penalty, with which, the AL method could eventually converge. The detailed steps of the PDD method are presented in TABLE I, where the parameter η_k measures the constraint violation and the parameter ϵ_k controls the solution accuracy of the BSUM algorithm, with both parameters going to zero as the number of outer iterations k increases.

The main effort of the PDD lies in Step 2. The operator '$\mathrm{BSUM}(P_{\varrho_k,\boldsymbol{\lambda}_k}, \tilde{\mathcal{L}}_k, \boldsymbol{z}^{k-1}, \epsilon_k)$' means that, starting from $\boldsymbol{z}^{k-1}$, the BSUM algorithm [32] or its variant shown in [24] is invoked to iteratively solve the problem $(P_{\varrho_k,\boldsymbol{\lambda}_k})$, given below

$$(P_{\varrho_k,\boldsymbol{\lambda}_k}) \quad \min_{\boldsymbol{x}_i\in\tilde{\mathcal{X}}_i,\boldsymbol{y}} \Big\{ \mathcal{L}_k(\boldsymbol{x},\boldsymbol{y}) \triangleq f(\boldsymbol{x},\boldsymbol{y}) + \sum_{j=1}^{n_y} \phi_j(s(\boldsymbol{y}_j)) + \boldsymbol{\lambda}_k^T\boldsymbol{h}(\boldsymbol{x},\boldsymbol{y}) + \frac{1}{2\varrho_k}\|\boldsymbol{h}(\boldsymbol{x},\boldsymbol{y})\|^2 \Big\} \tag{3}$$

where $\tilde{\mathcal{X}}_i = \{\boldsymbol{x}_i \in \mathcal{X}_i \mid \boldsymbol{g}_i(\boldsymbol{x}_i) \le 0\}$ and $\mathcal{L}_k(\boldsymbol{x},\boldsymbol{y})$ is the augmented Lagrange function with dual variable $\boldsymbol{\lambda}_k$ and penalty parameter ϱ_k. Further, the BSUM algorithm utilizes a locally tight upper bound of $\mathcal{L}_k(\boldsymbol{x},\boldsymbol{y})$, denoted as $\tilde{\mathcal{L}}_k$, and it terminates when certain solution accuracy ϵ_k is reached.

In the following, we address the convergence issue of the PDD method. To do so, we define $\boldsymbol{e}^k$ and $\boldsymbol{\Delta}_j^k$ in (4) and (5) (see the top of the next page), where $\boldsymbol{g}(\boldsymbol{x}) \triangleq (\boldsymbol{g}_i(\boldsymbol{x}_i))_i$. It is readily known that, when these two terms go to zero, the first order optimality condition with respect to $\boldsymbol{x}$ and $\boldsymbol{y}$ (i.e., (2a) and (2b)) holds true. The main convergence result is presented in Theorem 3.1. See [24] for the detailed proof.

$$\boldsymbol{e}^k = \mathcal{P}_{\mathcal{X}}\{\boldsymbol{x}^k - \nabla_{\boldsymbol{x}}\mathcal{L}_k(\boldsymbol{x}^k, \boldsymbol{y}^k) - \nabla \boldsymbol{g}(\boldsymbol{x}^k)^T\boldsymbol{\nu}^k\} - \boldsymbol{x}^k, \tag{4}$$

$$\Delta_j^k = \boldsymbol{y}_j^k - \arg\min_{\boldsymbol{y}_j} \left\{ \begin{array}{c} \phi_j'(s_j(\boldsymbol{y}_j^k))s_j(\boldsymbol{y}_j) + \frac{1}{2}||\boldsymbol{y}_j - \boldsymbol{y}_j^k||^2 \\ + \left(\nabla_{\boldsymbol{y}_j} f(\boldsymbol{x}^k, \boldsymbol{y}^k) + \nabla_{\boldsymbol{y}_j}\boldsymbol{h}(\boldsymbol{x}^k, \boldsymbol{y}^k)^T\left(\frac{1}{\varrho_k}\boldsymbol{h}(\boldsymbol{x}^k, \boldsymbol{y}^k) + \boldsymbol{\lambda}^k\right)\right)^T(\boldsymbol{y}_j - \boldsymbol{y}_j^k) \end{array} \right\} \tag{5}$$

Theorem 3.1 *Let $\{\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{\nu}^k\}$ be the sequence generated by Algorithm 1 for problem (P), where $\boldsymbol{\nu}^k = (\boldsymbol{\nu}_i^k)_i$ denotes the Lagrange multipliers associated with the constraints $\boldsymbol{g}_i(\boldsymbol{x}_i) \leq 0, \forall i$. The stop criterion for the BSUM algorithm involved in Algorithm 1 is*

$$\max\left(\|\boldsymbol{e}^k\|_\infty, \| \Delta^k \|_\infty\right) \leq \epsilon_k, \quad \forall k \tag{6}$$

with $\epsilon_k, \eta_k \to 0$ as $k \to \infty$. Suppose that $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a limit point of the sequence $\{\boldsymbol{x}^k, \boldsymbol{y}^k\}$ and Robinson's condition *holds for problem (P) at $(\boldsymbol{x}^*, \boldsymbol{y}^*)$. Then $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a KKT point of problem (P), i.e., it satisfies the KKT system* (2) *of problem (P).*

Remark 3.1 *A way to set η_k is to make it explicitly related to the constraint violation of the last iteration or the current minimum constraint violation. For example, we set $\eta_k = 0.9\|\boldsymbol{h}(\boldsymbol{z}^{k-1})\|_\infty$ in our simulations later. Moreover, for practical implementation, it is more reasonable to terminate the BSUM algorithm based on the progress of the objective value $\mathcal{L}_k(\boldsymbol{z}^r)$, i.e., $\frac{|\mathcal{L}_k(\boldsymbol{z}^r) - \mathcal{L}_k(\boldsymbol{z}^{r-1})|}{|\mathcal{L}_k(\boldsymbol{z}^{r-1})|} \leq \epsilon_k$. Here, $\boldsymbol{z}^r$ denotes the BSUM iterations. In addition, since the penalty term $\|\boldsymbol{h}(\boldsymbol{x})\|_\infty$ vanishes eventually, a practical choice of the termination condition for the PDD method is $\|\boldsymbol{h}(\boldsymbol{z}^k)\|_\infty \leq \epsilon_O$. Here, ϵ_O is some prescribed small constant.*

4. APPLICATION: MAX-MIN RATE FAIRNESS MULTI-CAST BEAMFORMING

4.1. Problem Statement

We here illustrate an important example of $(P1)$ in wireless communications—the multi-group multi-cast beamfroming problem [23]. Consider a single-cell multi-user multiple-input-multiple-output (MIMO) downlink system, where a base station (BS) equipped with N_t antennas wants to transmit $n_g > 1$ independent data streams to n_g group of users over a common frequency band. Suppose that the i-th group, denoted by $\mathcal{G}_i$, has m_i single-antenna users, each of which is interested in receiving a common data stream. Let s_i denote the data stream for group $\mathcal{G}_i$, $i = 1, 2, \ldots, n_g$ and $\boldsymbol{w}_i \in \mathbb{C}^{N_t}$ be the beamforming weight for the i-th group. The transmitted signal at the BS is given by $\sum_{i=1}^{n_g} \boldsymbol{w}_i s_i(t)$. Let $\boldsymbol{h}_k \in \mathbb{C}^{N_t}$ denote the channel vector between the BS and the receiver $k \in \mathcal{G}_i$. The received signal at the receiver $k \in \mathcal{G}_i$ is given by

$$r_k = \boldsymbol{h}_k^H \boldsymbol{w}_i s_i + \sum_{j \neq i} \boldsymbol{h}_k^H \boldsymbol{w}_j s_j + z_k, \quad k \in \mathcal{G}_i \tag{7}$$

where z_k denotes additional Gaussian white noise (AWGN) with variance σ_k^2.

Assume that s_i's are i.i.d Gaussian random variable with zero mean and unit variance, and s_i's and z_k's are independent of each other. Then the signal-to-interference-plus-noise-ratio (SINR) can be expressed as

$$\text{SINR}_k = \frac{\boldsymbol{w}_i^H \mathbf{R}_k \boldsymbol{w}_i}{\sum_{j \neq i} \boldsymbol{w}_j^H \mathbf{R}_k \boldsymbol{w}_j + \sigma_k^2}, k \in \mathcal{G}_i, i = 1, 2, \ldots, n_g \tag{8}$$

where $\mathbf{R}_k \triangleq \boldsymbol{h}_k \boldsymbol{h}_k^H$ or equals to the covariance matrix of $\boldsymbol{h}_k$.

To achieve rate fairness among users, a popular criterion for beamforming design is to maximize the minimum user rate subject to the BS power constraint $\sum_{i=1}^{n} \|\boldsymbol{w}_i\|^2 \leq P_{BS}$, where P_{BS} denotes the total available power at the BS. Since the power constraint must be active at the optimality, we can write the max-min rate fairness multi-cast beamforming problem equivalently as

$$\max_{\{\boldsymbol{w}_i\}} \min_i \min_{k \in \mathcal{G}_i} \quad \log_2\left(1 + \frac{\boldsymbol{w}^H \mathbf{A}_{i_k} \boldsymbol{w}}{\boldsymbol{w}^H \mathbf{B}_{i_k} \boldsymbol{w}}\right), \quad \text{s.t. } \|\boldsymbol{w}\|^2 = 1 \tag{9}$$

where $\boldsymbol{w} = (\boldsymbol{w}_i)_i$, $\mathbf{A}_{i_k} = \text{diag}\{\boldsymbol{e}_i\} \otimes \mathbf{R}_k$, and

$$\mathbf{B}_{i_k} = (\mathbf{I} - \text{diag}\{\boldsymbol{e}_i\}) \otimes \mathbf{R}_k + \frac{\sigma_k^2}{P_{BS}}\mathbf{I}.$$

This problem is NP-hard [23]. A popular method to address this problem is using semidefinite relaxation method coupled with bisection method [23], referred to as BisecSDR method, where, in each bisection, it is required to solve a semidefinite programming, requiring complexity at most $O\left(B \log(\frac{1}{\epsilon})\sqrt{n_g N_t}(n_g^3 N_t^6 + n_g N_t^2 K\right)$ where $K \triangleq \sum_{i=1}^{n_g} m_i$, the parameter ϵ represents the solution accuracy at the interior-point algorithm's termination, and B denotes the number of bisections.

4.2. PDD-based Algorithm

For convenience, let us consider a more general equivalent formulation of problem (9), which is given by

$$(P1) \quad \max_{\boldsymbol{w}} \min_{k \in \mathcal{K}} \quad \frac{\boldsymbol{w}^H \mathbf{A}_k \boldsymbol{w}}{\boldsymbol{w}^H \mathbf{B}_k \boldsymbol{w}}, \quad \text{s.t.} \quad \|\boldsymbol{w}\|^2 = 1 \tag{10}$$

where $\mathcal{K} \triangleq \{1, 2, \ldots, K\}$, and the matrices $\mathbf{A}_k$'s are all positive semidefinite and $\mathbf{B}_k$'s are all positive definite. It is worth mentioning that, besides the multi-cast beamforming, many other engineering problems can be formulated as $(P1)$ [33]. In what follows, we present the PDD-based algorithm for problem $(P1)$.

First, we can recast problem $(P1)$ as follows

$$\begin{array}{ll} \max\limits_{\boldsymbol{t} \geq 0, \boldsymbol{w}} & \min\limits_k t_k \\ \text{s.t.} & \|\mathbf{A}_k^{\frac{1}{2}} \boldsymbol{w}\| = t_k \|\mathbf{B}_k^{\frac{1}{2}} \boldsymbol{w}\|, \forall k, \\ & \|\boldsymbol{w}\|^2 = 1, \end{array} \tag{11}$$

which is a special case of problem (P) satisfying LICQ condition. In problem (11), the first K equality constraints are difficult coupling constraints. By moving these constraints into the objective, we obtain the corresponding augmented Lagrangian problem as follows

$$\begin{array}{ll} \max\limits_{\boldsymbol{t} \geq 0, \boldsymbol{w}} & \min\limits_k t_k - \dfrac{1}{2\rho}\displaystyle\sum_{k=1}^{K}\left(\|\mathbf{A}_k^{\frac{1}{2}} \boldsymbol{w}\| - t_k \|\mathbf{B}_k^{\frac{1}{2}} \boldsymbol{w}\| + \rho\lambda_k\right)^2 \\ \text{s.t.} & \|\boldsymbol{w}\|^2 = 1. \end{array} \tag{12}$$

where ρ is a penalty parameter and λ_k is a Lagrange multiplier associated with the k-th constraint.

The key to using the PDD method is to find appropriate locally tight upper bounds for the objective function, so that BSUM can be applied to optimize the AL. For problem (12), we can simply decouple the variables into two blocks $\boldsymbol{w}$ and $\boldsymbol{t}$, leading to two subproblems: i.e., solve (12) for $\boldsymbol{t}$ while fixing $\boldsymbol{w}$, and solve (12) for $\boldsymbol{w}$ while fixing $\boldsymbol{t}$, which are respectively referred to as *$\boldsymbol{t}$-subproblem and $\boldsymbol{w}$-subproblem*. The $\boldsymbol{t}$-subproblem is strictly convex and thus has a unique solution. By applying KKT analysis, we can obtain a closed-form solution to the $\boldsymbol{t}$-subproblem. The main difficulty lies in solving the $\boldsymbol{w}$-subproblem given by

$$\min_{\boldsymbol{w}} \ \vartheta(\boldsymbol{w}) \triangleq \sum_{k=1}^{K}\left(\|\mathbf{A}_k^{\frac{1}{2}} \boldsymbol{w}\| - t_k \|\mathbf{B}_k^{\frac{1}{2}} \boldsymbol{w}\| + \rho\lambda_k\right)^2 \ \text{s.t.} \ \|\boldsymbol{w}\|^2 = 1.$$

Apparently, the $\boldsymbol{w}$-subproblem is difficult to solve. Instead of exactly minimizing $\vartheta(\boldsymbol{w})$, we try to find a locally tight upper bound $u(\boldsymbol{w};\tilde{\boldsymbol{w}})$ for $\vartheta(\boldsymbol{w})$ and minimize this upper bound to update $\boldsymbol{w}$ given $\boldsymbol{t}$. Observing the constraint $\|\boldsymbol{w}\|=1$, we expect the upper bound to be a *homogeneous quadratic function* in the form of $\boldsymbol{w}^H\mathbf{C}\boldsymbol{w}$ or $\boldsymbol{w}_{eq}^T\mathbf{C}\boldsymbol{w}_{eq}$ where $\boldsymbol{w}_{eq}\triangleq(\Re e\{\boldsymbol{w}\},\Im m\{\boldsymbol{w}\})$, so that the resulting problem is an easily solvable eigenvalue problem.

By expanding $\vartheta(\boldsymbol{w})$, we can find that $\vartheta(\boldsymbol{w})$ includes the following four kinds of terms: **1)** $\boldsymbol{w}^H\mathbf{A}_k\boldsymbol{w}+t_k^2\boldsymbol{w}^H\mathbf{B}_k\boldsymbol{w}$; **2)** $-2t_k\|\mathbf{A}_k^{\frac{1}{2}}\boldsymbol{w}\|\|\mathbf{B}_k^{\frac{1}{2}}\boldsymbol{w}\|$; **3)** $2\rho\lambda_k\|\mathbf{A}_k^{\frac{1}{2}}\boldsymbol{w}\|$; **4)**$-2\rho\lambda_k t_k\|\mathbf{B}_k^{\frac{1}{2}}\boldsymbol{w}\|$. Clearly, we need to make efforts to bound the last three terms with homogenous quadratic functions. Unfortunately, since the multiplier λ_k's could be either negative or positive, it is challenging to bound the last two terms with homogenous quadratic functions. Thanks to the fact that $\|\boldsymbol{w}\|=1$, we can modify the third term as $2\rho\lambda_k\|\mathbf{A}_k^{\frac{1}{2}}\boldsymbol{w}\|\,\|\boldsymbol{w}\|$ when $\lambda_k<0$ and the fourth term as $-2\rho\lambda_k t_k\|\mathbf{B}_k^{\frac{1}{2}}\boldsymbol{w}\|\,\|\boldsymbol{w}\|$ when $\lambda_k>0$. Hence, essentially, $\vartheta(\boldsymbol{w})$ includes two kinds of terms in the forms of $\|\mathbf{Q}_1\boldsymbol{w}\|$ and $-\|\mathbf{Q}_1\boldsymbol{w}\|\,\|\mathbf{Q}_2\boldsymbol{w}\|$ with some appropriate $\mathbf{Q}_1$ and $\mathbf{Q}_2$. To bound these two terms, we resort to the following lemma.

Lemma 4.1 *For real vectors $\boldsymbol{x}$, $\boldsymbol{y}$, $\tilde{\boldsymbol{x}}$, $\tilde{\boldsymbol{y}}$, the following inequalities*

1) $\|\boldsymbol{x}\|\,\|\boldsymbol{y}\|\geq\frac{1}{\|\tilde{\boldsymbol{x}}\|\|\tilde{\boldsymbol{y}}\|}\boldsymbol{x}^T\tilde{\boldsymbol{x}}\tilde{\boldsymbol{y}}^T\boldsymbol{y},\ \forall\,\tilde{\boldsymbol{x}}\neq\mathbf{0},\tilde{\boldsymbol{y}}\neq\mathbf{0},\boldsymbol{x},\boldsymbol{y}$;

2) $\|\boldsymbol{x}\|\leq\frac{1}{2\|\tilde{\boldsymbol{x}}\|}\|\boldsymbol{x}\|^2+\frac{1}{2}\|\tilde{\boldsymbol{x}}\|,\ \forall\,\tilde{\boldsymbol{x}}\neq\mathbf{0},\boldsymbol{x}$

hold true with equality satisfied at $\boldsymbol{x}=\tilde{\boldsymbol{x}}$ and $\boldsymbol{y}=\tilde{\boldsymbol{y}}$.

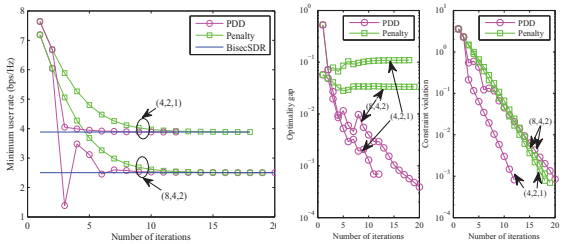
Proof: Part 1) follows directly from the Cauchy-Schwartz inequality, while Part 2) follows from the property of concave function by noting that $\|\boldsymbol{x}\|=\sqrt{\|\boldsymbol{x}\|^2}$ is a concave function of $\|\boldsymbol{x}\|^2$. ■

In terms of the above analysis and using Lemma 4.1, we can obtain $\vartheta(\boldsymbol{w})\leq u(\boldsymbol{w},\tilde{\boldsymbol{w}})\triangleq\boldsymbol{w}_{eq}^T\mathbf{C}\boldsymbol{w}_{eq}$ in the real domain. Moreover, it can be verified that $u(\boldsymbol{w},\tilde{\boldsymbol{w}})$ is a locally tight upper bound [32] of $\vartheta(\boldsymbol{w})$ over the set $\{\boldsymbol{w}|\,\|\boldsymbol{w}\|=1\}$. Due to space limitation, we omit the detailed form of $\mathbf{C}$ which is a $2n_gN_t$ by $2n_gN_t$ matrix function of $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$. With such an upper bound function, we update $\boldsymbol{w}$ by solving an eigenvalue problem, i.e., $\min_{\boldsymbol{w}_{eq}}\boldsymbol{w}_{eq}^T\mathbf{C}\boldsymbol{w}_{eq}$, s.t. $\|\boldsymbol{w}_{eq}\|=1$. Denote by $\boldsymbol{v}_{min}(\mathbf{C})$ the eigenvector of $\mathbf{C}$ corresponding to its minimum eigenvalue. Once we get $\boldsymbol{v}_{min}(\mathbf{C})$, we can construct the corresponding $\boldsymbol{w}$. It can be shown that the most costly step of the BSUM algorithm lies in calculating $\boldsymbol{v}_{min}(\mathbf{C})$, requiring complexity of $O(Kn_g^2N_t^2)+O(n_g^3N_t^3)$, where the first term corresponds to the computation of $\mathbf{C}$ while the second term corresponds to the eigenvalue decomposition. It is easily seen that the PDD method has lower complexity than the BisecSDR method in [23].

4.3. Numerical Results

In the simulations, the noise power is set to unit for all receivers and $P_{BS}=10$ dB. For convenience, we denote by (N_t,n_g,m_g) a multi-user multi-cast network with N_t BS antennas, n_g multi-cast groups each with m_g single-antenna users, hence $K=n_gm_g$ users in total. Furthermore, we set $\rho_0=K$, $c=0.6$, $\epsilon_O=1e-3$, as well as $\epsilon_k=\epsilon_{k-1}c$ with $\epsilon_0=1e-3$ for the PDD method. Moreover, we compare the PDD method with the BisecSDR method[1] in [23] and the penalty-BSUM method[2] proposed in [6] (abbreviated as 'Penalty' in the plot). The penalty-BSUM shares the same parameter setting with the PDD and the BisecSDR is terminated when the relative size of the bisection interval is smaller than $1e-3$.

Two examples of convergence behavior are illustrated in Fig. 1, where the final result of the BisecSDR method is presented. It is seen that the PDD method exhibits better convergence behavior than the penalty-BSUM method in terms of both the objective value and the optimality gap, while both achieving similar constraint violation. Here the optimality gap measures how well the solution satisfies the KKT condition of problem (9) or $(P1)$. Moreover, the PDD can achieve the upper bound value provided by the BisecSDR in these two examples, implying the excellent performance of the PDD.



(a) The minimum user rate. (b) The optimality and feasibility gap.

Fig. 1. Convergence behavior of the PDD method.

Table 2 compares the performance of three methods in terms of the CPU time and the achieved minimum rate averaged over 100 random channel realizations. In the table, R_{PDD}, R_{SDR}, and $R_{Penalty}$ denote the minimum rate achieved by the PDD method, the BisecSDR method, and the penalty-BSUM method, respectively, while T_{PDD}, T_{SDR}, and $T_{Penalty}$ denote the corresponding cpu time required by each method. It can be observed that the PDD method requires much less cpu time than the BisecSDR method while achieving almost global optimality. Moreover, it performs more efficient than the penalty-BSUM method.

Table 2. The Average CPU Time And Min. Rate Comparison

Network	$\frac{R_{PDD}}{R_{SDR}}$	$\frac{T_{SDR}}{T_{PDD}}$	$\frac{R_{PDD}}{R_{Penalty}}$	$\frac{T_{Penalty}}{T_{PDD}}$
$(2,2,2)$	99.79%	21.56	100.20%	1.95
$(4,2,2)$	99.93%	28.47	100.00%	2.43
$(8,4,2)$	99.89%	10.27	100.00%	1.96

5. CONCLUSION

We have provided an optimization framework for a class of nonsmooth nonconvex optimization problems. Its convergence to KKT points has been established under Robinson's condition. Resorting to the BSUM-type algorithm, our framework can make use of the problem structure as fully as possible, thus it scales well to the problem size. It can be used to address many difficult problems arising from signal processing.

[1] Since the semidefinite relaxation method cannot guarantee a rank-one solution, the work [33] proposed using Gaussian randomization procedure (GRP) to recover a good rank-one solution in the end of bisection method. Note that the BisecSDR method here includes no GRP. Hence, it serves as an upper bound performance in the comparison.

[2] The penalty-BSUM algorithm is similar to the PDD method but does not include the dual update as in the PDD method. The algorithm in [33] is in essence the penalty-BSUM algorithm, with the only difference in that some fixed penalty parameter was used in [33] while the penalty-BSUM algorithm uses increasing penalty. However, fixed penalty parameter cannot guarantee a KKT solution. Moreover, it is generally difficult to choose a penalty parameter which works well for all cases. Hence, we modify the algorithm in [33] to the exact penalty-BSUM algorithm by using increasing penalty.

6. REFERENCES

[1] B. Dai and W. Yu, "Sparse beamforming and user-centric clustering for downlink cloud radio access network," *IEEE Access*, vol. 2, pp. 1326–1339, Nov 2014.

[2] D. Kuang, S. Yun, and H. Park, "SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering," *Journal of Global Optimization*, vol. 62, no. 3, pp. 545–574, 2015.

[3] Q. Tran Dinh, S. Gumussoy, W. Michiels, and M. Diehl, "Combining convex-concave decompositions and linearization approaches for solving BMIs, with application to static output feedback," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1377–1390, June 2012.

[4] J. Wang and J. Q. Zhang, "A globally optimal bilinear programming approach to the design of approximate hilbert pairs of orthonormal wavelet bases," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 233–241, Jan 2010.

[5] K. T. Truong, P. Sartori, and R. W. Heath, "Cooperative algorithms for mimo amplify-and-forward relay networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1272–1287, March 2013.

[6] Q. Shi, M. Hong, X. Gao, E. Song, Y. Cai, and W. Xu, "Joint source-relay design for full–duplex mimo af relay systems," *IEEE Transactions on Signal Processing*, vol. PP, no. 99, Sept. 2016.

[7] Y. Rong, "Joint source and relay optimization for two-way linear non-regenerative mimo relay communications," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6533–6546, Dec 2012.

[8] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov 2006.

[9] B. Friedlander and T. Strohmer, "Bilinear compressed sensing for array self-calibration," in *2014 48th Asilomar Conference on Signals, Systems and Computers*, Nov 2014, pp. 363–367.

[10] D. L. Sun and C. Fvotte, "Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6201–6205.

[11] Y. Xu, W. Yin, Z. Wen, and Y. Zhang, "An alternating direction algorithm for matrix completion with nonnegative factors," *Frontiers of Mathematics in China*, vol. 7, no. 2, pp. 365–384, 2012.

[12] D. Hajinezhad, T. H. Chang, X. Wang, Q. Shi, and M. Hong, "Nonnegative matrix factorization using admm: Algorithm and convergence analysis," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 4742–4746.

[13] X. Fu, W. K. Ma, K. Huang, and N. D. Sidiropoulos, "Blind separation of quasi-stationary sources: Exploiting convex geometry in covariance domain," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp. 2306–2320, May 2015.

[14] R. Zhang, C. C. Chai, and Y. C. Liang, "Joint beamforming and power control for multiantenna relay broadcast channel with qos constraints," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 726–737, Feb 2009.

[15] Q. Shi, M. Razaviyayn, M. Hong, and Z. Q. Luo, "Sinr constrained beamforming for a mimo multi-user downlink system: Algorithms and convergence analysis," *IEEE Transactions on Signal Processing*, vol. 64, no. 11, pp. 2920–2933, June 2016.

[16] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 2 edition, 1999.

[17] Z. Lu, Y. Zhang, and X. Li, "Penalty decomposition methods for rank minimization," *Optimization Methods and Software*, vol. 30, no. 3, pp. 531–558, 2015.

[18] M. R. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, vol. 4, no. 5, pp. 303–320, 1969.

[19] M. J. D. Powell, "A method for nonlinear constraints in minimization problems," in *Optimization*, R. Fletcher, Ed., pp. 283–298. Academic Press, New York, 1969.

[20] M. Kočvara and M. Stingl, *PENNON: A generalized augmented Lagrangian method for semidefinite programming*, pp. 303–321, Springer US, Boston, MA, 2003.

[21] A. F. Izmailov and M. V. Solodov, "On attraction of linearly constrained lagrangian methods and of stabilized and quasi-newton sqp methods to critical multipliers," *Mathematical Programming*, vol. 126, no. 2, pp. 231–257, 2011.

[22] D. Fernndez and M. V. Solodov, "Local convergence of exact and inexact augmented lagrangian methods under the second-order sufficient optimality condition," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 384–407, 2012.

[23] E. Karipidis, N. D. Sidiropoulos, and Z. Q. Luo, "Quality of service and max-min fair transmit beamforming to multiple cochannel multicast groups," *IEEE Transactions on Signal Processing*, vol. 56, no. 3, pp. 1268–1279, March 2008.

[24] Q. Shi and M. Hong, "Penalty dual decomposition method for nonconvex nonsmooth optimization–Part I: Theory," *Technical Report*, available at arXiv.org, 2016.

[25] G. Gasso, A. Rakotomamonjy, and S. Canu, "Recovering sparse signals with a certain family of nonconvex penalties and dc programming," *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4686–4698, Dec 2009.

[26] E. J. Balder, "On generalized gradients and optimization," http://www.staff.science.uu.nl/ balde101/cao10/cursus08_3.pdf, Sept. 2008.

[27] A. Ruszczynski, *Nonlinear optimization*, Princeton University Press, 2011.

[28] Z. Lu and Y. Zhang, "Sparse approximation via penalty decomposition methods," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2448–2478, 2013.

[29] A. F. Izmailov and M. V. Solodov, "Optimality conditions for irregular inequality-constrained problems," *SIAM Journal on Control and Optimization*, vol. 40, no. 4, pp. 1280–1295, 2002.

[30] M. P. Friedlander and M. A. Saunders, "A globally convergent linearly constrained lagrangian method for nonlinear optimization," *SIAM J. on Optimization*, vol. 15, no. 3, pp. 863–897, 2005.

[31] D.P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA, 1996.

[32] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[33] M. Soltanalian, A. Gharanjik, M. R. B. Shankar, and B. Ottersten, "Grab-n-pull: An optimization framework for fairness-achieving networks," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 3301–3305.