Joint Multicast and Unicast Beamforming for the MISO Downlink Interference Channel

Ya-Feng Liu†, Cheng Lu†, Meixia Tao‡, Jiageng Wu§
†State Key Laboratory of Scientific and Engineering Computing, AMSS, CAS, Beijing 100190, China
‡School of Economics and Management, North China Electric Power University, Beijing 102206, China
§Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
Email: yafliu@lsec.cc.ac.cn, lucheng1983@163.com, mxtao@sjtu.edu.cn, chart_wu7@126.com

Abstract—In this paper, we consider the joint multicast and unicast beamforming design problem in the multi-input single-output downlink wireless network, where all base stations (BSs) potentially can cooperate (as a single virtual BS) to transmit a common multicast and multiple dedicated unicast data streams at the same time on the same frequency band. To reduce cooperation overhead (among different BSs), we prefer partial cooperation transmission such that each user’s data stream (either multicast or unicast) is served by only a small subset of BSs. We formulate the problem from a (group) sparse optimization perspective and propose a branch-and-bound (BB) algorithm for solving the problem. Our proposed BB algorithm is guaranteed to find the globally optimal solution of the problem. We also propose an efficient successive linear approximation (SLA) algorithm for solving the problem. Numerical results show that the SLA algorithm can perform very close to the BB algorithm but with significantly less computational complexity. It is also shown that the proposed mixed $\ell_2/\ell_1$ regularizer in the sparse formulation provides a flexible and effective tradeoff between the total transmission power cost and the cooperation cost.

Index Terms—BB algorithm, global optimality, group sparse, joint multicast and unicast beamforming.

I. INTRODUCTION

Transmit beamforming has been recognized as a powerful directional signal transmission scheme in multi-user multi-antenna wireless systems. Two scenarios that are extensively studied in the literatures are multicast beamforming [1] and unicast beamforming [2], [3]. In multicast beamforming, all users in a same multicast group want to receive a common message shaped by a single beamformer. In unicast beamforming, each user wishes to receive a private message shaped by a dedicated beamformer. The above two beamforming techniques can be combined as joint multicast and unicast beamforming in scenarios where users wish to receive both private and common messages.

Simultaneous multicast and unicast transmissions via superposition coding on the same frequency band at the same time has been proposed in [4]. Compared to frequency/time division multiplexing, joint multicast and unicast beamforming equipped with superposition coding can provide significant performance gains (especially when there is a large difference in terms of SINR in superposed signals) because superposing multicast and unicast allows for full bandwidth utilization and more efficient transmit power allocation [5]. However, an issue associated with superposing multicast and unicast is the cross-interferences between the two traffic streams, which, if not properly managed, may significantly affect the system performance. Related works on joint multicast and unicast beamforming dealing with the above issue include [6], [7], [8], [9], [10].

In this paper, we consider the joint multicast and unicast beamforming design problem in a multi-input single-output (MISO) downlink wireless network, where all BSs can potentially cooperate to transmit the multicast and unicast data to all users at the same time on the same frequency band. To reduce the data-exchange overhead for cooperation, or, effectively, to reduce the backhaul traffic load among different BSs, we consider partial cooperative transmission as in [11] and expect that each user is served by a small subset of BSs. More specifically, if all BSs form as a single virtual BS, then its antennas can be partitioned into multiple groups (each corresponding to one BS). Then, the expectation that each user’s data stream is served by a small subset of BSs mathematically translates to the objective that its virtual beamformer has a group sparse structure. This motivates us to formulate the joint multicast and unicast beamforming problem as the minimization of the total transmission power plus a mixed $\ell_2/\ell_1$ regularizer, subject to the quality of service constraints of all users’ multicast and unicast data. We propose two algorithms to solve the formulated problem. The first one is a branch-and-bound (BB) algorithm that is guaranteed to find the globally optimal solution of the problem. Our proposed BB algorithm can provide an important benchmark for evaluating the performance of suboptimal algorithms for the same problem. The second one is an efficient successive linear approximation (SLA) algorithm with a judicious choice of the initial point. The effectiveness and efficiency of the proposed algorithms are illustrated by numerical simulations.

Notations. We use $\mathbb{C}^{n \times 1}$ ($\mathbb{R}^{n \times 1}$) to denote the set of the $n$-dimensional complex (real) column vectors. For a given complex number $c$, $\text{Re}(c)$, $\text{Im}(c)$, and $\text{arg}(c)$ stand for its real part, its imaginary part, and its argument, respectively. For a
given complex vector $x$, $\|x\|$ denotes its Euclidean norm.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a MISO downlink wireless network which consists of a set $B := \{1, 2, \ldots, B\}$ of BSs and a set $K := \{1, 2, \ldots, K\}$ of mobile users. We assume that all BSs have $M$ antennas and all mobile users have a single antenna. We assume frequency-flat quasi-static complex channels. Let $h_{k,b} \in \mathbb{C}^{M \times 1}$ denote the channel vector between BS $b$ to user $k$. We use $s_k$ to denote an encoded unicast symbol intended for user $k$ and $s$ to denote an encoded multicast symbol intended for all users. Then, the signal transmitted by BS $b$ is

$$
\sum_{k \in K} v_{k,b} s_k + w_b s,
$$

where $v_{k,b} \in \mathbb{C}^{M \times 1}$ is the unicast beamforming vector applied at BS $b$ for user $k$ and $w_b$ is the multicast beamforming vector applied at the same BS. Let $n_b$ be the additive white Gaussian noise with variance $\sigma_b^2$ at user $k$. Then, the received signal at user $k$ can be expressed as

$$
y_k = \sum_{b \in B} h_{k,b}^H w_b s + \sum_{b \in B} h_{k,b}^H v_{j,b} s_j + n_k.
$$

The users perform successive interference cancelation [4] to first decode the multicast signal (treating the unicast signals as noise) and then decode the unicast signals. More specifically, assuming that both multicast and unicast signals have unit power, the multicast SINR for user $k$ is

$$
\text{SINR}_k^M = \frac{\|h_k^H w\|^2}{\sum_{j \in K} \|h_k^H v_j\|^2 + \sigma_k^2}, \quad k \in K,
$$

where $h_k = [h_{k,1}^T, h_{k,2}^T, \ldots, h_{k,B}^T]^T \in \mathbb{C}^{BM \times 1}$ denotes the network-wide channel vector between all BSs and user $k$, $w = [w_1^T, w_2^T, \ldots, w_B^T]^T \in \mathbb{C}^{BM \times 1}$ denotes the accumulated multicast beamforming vector from all BSs, and $v_j = [v_{j,1}^T, v_{j,2}^T, \ldots, v_{j,B}^T]^T \in \mathbb{C}^{BM \times 1}$ denotes the accumulated unicast beamforming vector from all BSs to user $j$; the unicast SINR for user $k$ is

$$
\text{SINR}_k^U = \frac{\|h_k^H v_k\|^2}{\sum_{j \neq k} \|h_k^H v_j\|^2 + \sigma_k^2}, \quad k \in K.
$$

Notice that in SINRs (1) and (2), we implicitly assume that all BSs share the multicast/unicast data streams for each user $k$, $k \in K$. Unfortunately, this will cause significant data-exchange overhead or, effectively, backhaul burden, especially when the number of users and BSs becomes large. To reduce overhead, partial cooperative transmission is desired whereby each user’s multicast/unicast transmission is served by a small subset of BSs. Mathematically, this translates to the group sparse structure of the accumulated multicast/unicast beamforming vectors $w$ and $v_k$, i.e., most of block components $\{w_b\}$ in $w$ and $\{v_{k,b}\}$ in $v_k$ should be zero. One popular approach to enforcing the sparsity of the solution in an optimization problem is to penalize its objective function with a group sparse encouraging penalty such as the mixed $\ell_2/\ell_1$ norm [12]. In our case, the group sparsity of $w$ and $v_k$ can be expressed as $\sum_{b \in B} \|w_b\|$ and $\sum_{b \in B} \|v_{k,b}\|$, respectively.

In this paper, we are interested in minimizing the total transmission power at the BSs as well as reducing data-exchange overhead among different BSs, satisfying the multicast and unicast SINR constraints of all users. Mathematically, the problem can be formulated as follows:

$$
\begin{align*}
\min_{(w,v)} & \quad f_\lambda(w, v) \\
\text{s.t.} & \quad \text{SINR}_k^M \geq \gamma_k^M, \quad k \in K, \\
& \quad \text{SINR}_k^U \geq \gamma_k^U, \quad k \in K, \\
\end{align*}
$$

where

$$
f_\lambda(w, v) = (1 - \lambda) \left(\|w\|^2 + \|v\|^2\right) + \lambda \left(\sum_{b \in B} \|w_b\| + \sum_{k \in K} \sum_{b \in B} \|v_{k,b}\|\right),
$$

$$
v = [v_1, v_2, \ldots, v_K]^T \in \mathbb{C}^{KBM \times 1}
$$

is a collection of all unicast beamforming vectors; $\lambda \in [0, 1]$ is a parameter that controls the sparsity of the beamforming vectors; $\gamma_k^M$ and $\gamma_k^U$ are the desired multicast and unicast SINR targets of user $k$, respectively. Problem (3) is non-convex due to the non-convex multicast SINR constraints. We remark that the above formulated problem (3) is different from all previously mentioned works [6], [7], [8], [9], [10] mainly in the group sparsity regularizer in the objective function. Next, we develop two algorithms for solving problem (3).

III. PROPOSED BB ALGORITHM

In this section, we first briefly introduce the basic idea of the BB algorithm in Section III-A. Then, we propose a convex relaxation of problem (3) in Section III-B, which will provide efficient lower bounds in our proposed BB algorithm. Finally, we present our proposed BB algorithm for solving problem (3) in Section III-C.

A. Overview of the BB Algorithm

A BB algorithm (for solving a minimization problem) is generally a smart enumeration procedure, which recursively partitions the feasible region of the original problem to smaller subregions and constructs subproblems over the partitioned subregions. In the enumeration procedure, a lower bound for each subproblem is often computed by solving a convex relaxation problem defined over the corresponding subregion; an upper bound is obtained from the best known feasible solution generated by the enumeration procedure or by some other local optimization/heuristic algorithms. A well-designed BB algorithm is generally much more efficient than “brute force” enumeration, because a subproblem with a lower bound being larger than the obtained upper bound, which does not contain the globally optimal solution of the original problem in its feasible region and thus is often called as an inactive subproblem, will not be further enumerated. The BB algorithm terminates until all active subproblems have been enumerated, and then an optimal solution (within any given error tolerance) can be obtained.
B. Convex Relaxation of Problem (3)

We first give an equivalent reformulation of problem (3). Define $q_k(v) := \sum_{v \in K} \left| h^H_k v \right|^2 + \sigma_k^2$, $k \in K$. According to [3], all unicast SINR constraints can be rewritten as

\[
\begin{align*}
\text{Re}(h^H_k v_k) & \geq \sqrt{\frac{\gamma^M}{1 + \gamma^M}} q_k(v), \quad k \in K, \\
\text{Im}(h^H_k v_k) & = 0, \quad k \in K;
\end{align*}
\]

the multiset SINR constraint of user $K$ can be rewritten as

\[
\text{Re}(h^H_K w) \geq \sqrt{\gamma^M} q_K(v) \quad \text{and} \quad \text{Im}(h^H_K w) = 0.
\]

Hence, problem (3) can be reformulated as

\[
\begin{align*}
\min_{w, v} & \quad f_\lambda(w, v) \\
\text{s.t.} & \quad |h^H_K w| \geq \sqrt{\gamma^M} q_K(v), \quad k \in K \setminus \{K\}, \\
& \quad (4) \quad \text{and} \quad (5).
\end{align*}
\]

Problem (6) is still nonconvex because the first $K - 1$ multiset SINR constraints in it are nonconvex. Fortunately, from [13, Proposition 1, Proposition 2], we have the following proposition, based on which we can obtain an efficient convex relaxation of problem (6).

**Proposition 1.** Suppose $\arg\max \{h^H_k w\} \in [l_k, u_k]$, $k \in K \setminus \{K\}$. Let $S^{l_k, u_k}_k$ (the subscript $k$ is introduced here because the set depends on $h_k$ and $q_k(v)$) be the set defined by the inequality $|h^H_k w| \geq \sqrt{\gamma^M} q_k(v)$ and let $S^{l_k, u_k}_k$ be the set defined by the following set of inequalities

\[
\begin{align*}
\sin(l_k)\text{Re}(h^H_k w) - \cos(l_k)\text{Im}(h^H_k w) & \leq 0, \\
\sin(u_k)\text{Re}(h^H_k w) - \cos(u_k)\text{Im}(h^H_k w) & \geq 0, \\
a_k\text{Re}(h^H_k w) + b_k\text{Im}(h^H_k w) & \geq (a_k^2 + b_k^2) \sqrt{\gamma^M} q_k(v),
\end{align*}
\]

where $a_k = (\cos(l_k) + \cos(u_k))/2$ and $b_k = (\sin(l_k) + \sin(u_k))/2$. Then, $S^{l_k, u_k}_k$ is the convex envelope of $S^{l_k, u_k}_k$ for $k \in K \setminus \{K\}$.

Proposition 1 immediately implies that, if $\arg\max \{h^H_k w\} \in [l_k, u_k]$ for $k \in K \setminus \{K\}$ in problem (6), then

\[
\begin{align*}
\min_{w, v} & \quad f_\lambda(w, v) \\
\text{s.t.} & \quad (w, v) \in S^{l_k, u_k}_k, \quad k \in K \setminus \{K\}, \\
& \quad (4) \quad \text{and} \quad (5),
\end{align*}
\]

is a convex relaxation of problem (6). Moreover, an effective approach to tightening the above relaxation is to reduce the width of the corresponding intervals $[l_k, u_k]$.

Problem (7) can be equivalently reformulated as a convex optimization problem over two self-dual (symmetric) cones: (standard) second-order cone and rotated second-order cone. Therefore, problem (7) can be solved efficiently and globally by using the interior-point algorithm [14], which has been nicely implemented in sedumi [15].

C. Proposed BB Algorithm

In this subsection, we propose a BB algorithm for globally solving problem (3). The basic idea of the proposed algorithm is to relax the original problem to the conic program (7) and gradually tighten the relaxation by reducing the width of the associated intervals.

For ease of presentation, we introduce the following notations. Let $A = \bigcup_{k=1}^{K-1} [l_k, u_k]$ and let $CP(A)$ denote the conic program relaxation (7) defined over the set $A$; let $P$ denote the constructed problem list and let $\{A, w, v, L\}$ denote a problem instance from the list $P$, where $L$ is the optimal value of $CP(A)$ and $(w, v)$ is its optimal solution; let superscript $t$ denote the iteration number; and let $(w^*, v^*)$ denote the best known feasible solution and let $U^*$ denote the objective value of problem (3) at $(w^*, v^*)$.

The pseudo-code of our proposed BB algorithm for solving problem (3) is given in Algorithm 1, where

\[
w^t = S(w^t, v^t) := \min \left\{ \frac{|h^H_k w^t|}{\sqrt{\gamma^M q_k(v^t)}}, \cdots, \frac{|h^H_K w^t|}{\sqrt{\gamma^M q_K(v^t)}} \right\}
\]

and

\[
k^* = \arg \min_{k \in \{1, \ldots, K-1\}} \left\{ |h^H_k w^t| - \sqrt{\gamma^M q_k(v^t)} \right\}.
\]

**Definition 1** ($\epsilon$-Optimal Solution). For any given $\epsilon > 0$, a feasible point $(w, v)$ is called an $\epsilon$-optimal solution of problem (3) if it satisfies $(f_\lambda(w, v) - v^*)/v^* \leq \epsilon$, where $v^*$ is the optimal value of problem (3).

**Theorem 1.** For any given $\epsilon > 0$ and any given feasible instance of problem (3), Algorithm 1 will return an $\epsilon$-optimal solution of the given instance.

IV. PROPOSED SLA ALGORITHM

In this section, we propose an efficient SLA algorithm for solving problem (6) (equivalent to problem (3)). The basic idea of the SLA algorithm is to approximate the term $|h^H_k w|^2$ by its first order Taylor expansion and then the nonconvex constraints in (P') become convex ones. More specifically, the SLA algorithm introduces the auxiliary variables

\[
x_k := \left[ \text{Re}(h^H_k w), \text{Im}(h^H_k w) \right]^T \in \mathbb{R}^2, \quad k \in K \setminus \{K\};
\]

given the current point $\{x_k^t \}$ (at the $t$-th iteration), the algorithm first approximates the nonconvex constraint $|h^H_k w|^2 \geq \gamma^M q_k(v)$ by the convex quadratic constraint

\[
\|x_k^t\|^2 + 2(x_k^t)^T(x_k - x_k^t) \geq \gamma^M q_k(v), \quad k \in K \setminus \{K\},
\]

and then solves the following approximation problem to obtain the next iterate $\{x_k^{t+1} \}$:

\[
\begin{align*}
\min_{w, v, x} & \quad f_\lambda(w, v) \\
\text{s.t.} & \quad \|x_k\|^2 + 2(x_k)^T(x_k - x_k^t) \geq \gamma^M q_k(v), \quad k \in K \setminus \{K\}, \\
& \quad x_k = \left[ \text{Re}(h^H_k w), \text{Im}(h^H_k w) \right]^T, \quad k \in K \setminus \{K\}, \\
& \quad (4) \quad \text{and} \quad (5),
\end{align*}
\]

where $x$ is a collection of $\{x_k\}^{K-1}_{k=1}$.
One potential drawback of the SLA algorithm is that its performance is sensitive to the choice of the initial point. The SLA algorithm might converge to a (bad) local minimizer or subproblem (10) in it might become infeasible if the initial point is not carefully chosen. To overcome this issue, we propose the following two-stage approach to finding a good initial point:

I: Solve the following unicast beamforming problem for its solution $\bar{v}$:

$$
\begin{align*}
\min_{\{v\}} & \ (1 - \lambda)\|v\|^2 + \lambda \sum_{k \in K} \sum_{b \in B} \|v_{k,b}\| \\
\text{s.t.} & \ \ (4);
\end{align*}
$$

II: Given an integer $N > 1$, randomly generate points $\{w^{(n)}\}_{n=1}^{N}$ and scale them as

$$
\bar{w}^{(n)} = \frac{\sqrt{\gamma^M}w^{(n)}}{\min\left\{\frac{h_{k,b}^T\bar{w}^{(n)}}{q_{k}(\bar{v})}, \ldots, \frac{|h_{k,b}^T|}{q_{k}(\bar{v})}\right\}}.
$$

Let $\bar{w}$ be the one among $\{\bar{w}^{(n)}\}_{n=1}^{N}$ with minimum $(1 - \lambda)\|w\|^2 + \lambda \sum_{b \in B} \|w_{k,b}\|$. It is simple to show that the point $\{\bar{w}, \bar{v}\}$ is feasible to the original problem (3).

The proposed SLA algorithm is summarized as Algorithm 2. Some remarks are in order. First, its convergence to a stationary point can be established as in [16]. Second, for any given $x_k$, there holds $\|x_k\|^2 \geq \|x_k\|^2 + 2(x_k^T(x_k - x_k^*))$, $\forall \ x_k$. Hence, the feasible region of subproblem (10) is a subset of that of the original problem (3). This, together with the fact that the initial point is feasible, immediately implies that the subproblem at each iteration is feasible and the solution of each subproblem is feasible to the original problem (3). Finally, both subproblems (10) and (11) can be reformulated as convex conic programs and thus can be solved efficiently and globally by the interior-point algorithm [14], [15].

V. NUMERICAL SIMULATIONS

In this section, we present some numerical simulation results to illustrate the performance of the proposed BB and SLA algorithms. We consider a multi-user MISO downlink interference channel, where there are $B = 6$ BSs and each BS is equipped with $M = 2$ antennas. The channel coefficients are generated according to the complex Gaussian distribution $h_{k,b} \sim \mathcal{CN}(0, I_2)$, $k \in K$, $b \in B$. The noise power is set to be $\sigma_k^2 = 1$, $k \in K$. The multicast and unicast SINR targets are set to be $\gamma^M = 10$ and $\gamma^{(b)}_{k} = 1$ for all users. The error tolerance $\epsilon$ in both Algorithms 1 and 2 are set to $1 e - 3$ and the sample size $N$ in Algorithm 2 is set to be 1000. We use sedumi [15] to solve all conic subproblems.

All figures are obtained by averaging over 200 Monte-Carlo runs. Figs. 1 and 2 plot the average total cost (i.e., the objective value of problem (3)) and the average CPU time comparison of the proposed two algorithms versus different numbers of users. We can observe from Fig. 2 that the CPU time of the BB algorithm increases exponentially with the number of users. However, we highlight here that the BB algorithm is guaranteed to find the globally optimal solution of problem (3) (within any given error tolerance). Compared to the BB algorithm, the SLA algorithm can achieve almost the same objective value but with significantly less CPU time. The good performance of the SLA algorithm is mainly due to the judicious choice of the initial point.

Fig. 3 plots the tradeoff curve of the average total power and the ratio of the number of active links and the number of total links with $K = 6, 8, 10$. Each point in Fig. 3 corresponds to the results of problem (3) obtained by the SLA algorithm with $\lambda = 0.1, \ell = 0, 1, \ldots, 10$. In particular, the top left (lower right) point in each curve corresponds to the case where $\lambda = 1 (\ell = 0)$. Fig. 3 demonstrates that the mixed $\ell_2/\ell_1$ regularizer in problem (3) indeed is effective in
achieving a tradeoff between the total transmission power and the number of active links, i.e., the cooperation cost. We can also observe from Fig. 3 that: when \( \lambda = 0 \), all links are active, i.e., full cooperation; as \( \lambda \) increases, the active ratio, i.e., the ratio of the number of active links and the number of links, decreases and the total transmission power generally increases; for a fixed \( \lambda \), both of the active ratio and the total transmission power increase with the number of users. All these results are consistent with our engineering practice.

Algorithm 1: Proposed BB Algorithm for Globally Solving Problem (3)

1: input: An instance of problem (3), and an error tolerance \( \epsilon > 0 \).
2: Initialize \( \mathcal{P} = \emptyset \), \( A^0 = \prod_{k=1}^{K-1} \{ t_k^0, u_k^0 \} = \{ 0, 2\pi \}^{K-1} \), and set \( t = 0 \). // Initialization.
3: Solve \( \text{CP}(A^t) \) for its optimal solution \((w^0, v^0)\) and its optimal value \( L^0 \).
4: Compute \( \hat{w}^0 = S(w^0, v^0) \), where \( S(\cdot, \cdot) \) is defined in (8).
5: Set \( U^* = f_\lambda(\hat{w}^0, v^0) \) and \((w^*, v^*) = (\hat{w}^0, v^0) \). // Initial Upper Bound and Solution.
6: Add \( \{A^0, w^0, v^0, L^0\} \) into the problem list \( \mathcal{P} \). // Initial Problem List.
7: loop
8: Set \( t \leftarrow t + 1 \).
9: Choose a problem from \( \mathcal{P} \), denoted as \( \{A^t, w^t, v^t, L^t\} \), such that the bound \( L^t \) is the smallest one in \( \mathcal{P} \). // Lower Bound.
10: Delete the chosen subproblem from \( \mathcal{P} \). // Update Problem List.
11: if \( (U^* - L^t)/L^t < \epsilon \) then return \((w^*, v^*)\) and terminate the algorithm.
12: end if
13: end if
14: Select \( k \) according to (9) and let \( z_k^* = \frac{1}{2}(t_k^* + u_k^*) \).
15: Branch \( A^t \) into two sets \( A^t_k = \{ \theta \in A^t | \theta_k^* \leq z_k^* \} \) and \( A^t_k = \{ \theta \in A^t | \theta_k^* > z_k^* \} \), where \( \theta_k^* \) is the \( k \)-th component of \( \theta \in \mathbb{R}^{(K-1) \times 1} \). // Branch.
16: Solve \( \text{CP}(A^t_k) \) for its optimal solution \((w^t_k, v^t_k)\) and its optimal value \( L^t_k \).
17: Compute \( \hat{w}^t_k = S(w^t_k, v^t_k) \), where \( S(\cdot, \cdot) \) is defined in (8).
18: if \( L^t_k \leq L^* \) then
19: add \( \{A^t_k, w^t_k, v^t_k, L^t_k\} \) into \( \mathcal{P} \). // Update Problem List.
20: end if
21: if \( U^* > f_\lambda(\hat{w}^t_k, v^t_k) \) then
22: set \( U^* = f_\lambda(\hat{w}^t_k, v^t_k) \) and \((w^*, v^*) = (\hat{w}^t_k, v^t_k) \). // Update Upper Bound and Solution.
23: end if
24: end if
25: Solve \( \text{CP}(A^t_k) \) for its optimal solution \((w^t_k, v^t_k)\) and its optimal value \( L^t_k \).
26: Compute \( \hat{w}^t_k = S(w^t_k, v^t_k) \), where \( S(\cdot, \cdot) \) is defined in (8).
27: if \( L^t_k \leq U^* \) then
28: add \( \{A^t_k, w^t_k, v^t_k, L^t_k\} \) into \( \mathcal{P} \).
29: end if
30: if \( U^* > f_\lambda(\hat{w}^t_k, v^t_k) \) then
31: set \( U^* = f_\lambda(\hat{w}^t_k, v^t_k) \) and \((w^*, v^*) = (\hat{w}^t_k, v^t_k) \).
32: end if
33: end loop

Algorithm 2: Proposed SLA Algorithm for Solving Problem (3)

1: input: An instance of problem (3), an integer number \( N \geq 1 \) and an error tolerance \( \epsilon > 0 \).
2: Solve problem (11) for \( v \), use (12) to obtain \( w \), and compute \( x_k^* = \left[ \text{Re}(h_k^H w), \text{Im}(h_k^H w) \right]^T \), \( k \in K \setminus \{ K \} \). Set \( t = 0 \). // Initialization.
3: loop
4: Set \( t \leftarrow t + 1 \).
5: Solve problem (10) for its optimal solution \((w^t, v^t, x^t)\).
6: if \( \| x^t - x^{t-1} \|_2 / \| x^{t-1} \|_2 < \epsilon \) then
7: return \((w^t, v^t)\) and terminate the algorithm. // Termination.
8: end if
9: end loop

REFERENCES