A Parallel Algorithm for Energy Efficiency Maximization in Massive MIMO Networks

Yang Yang
Intel Deutschland GmbH, Neubiberg 85579, Germany
Email: yang1.yang@intel.com

Marius Pesavento
Darmstadt University of Technology, Darmstadt 64283, Germany
Email: pesavento@nt.tu-darmstadt.de

***Abstract*—In this paper, we propose a novel iterative algorithm based on successive convex approximation for the nonconvex energy efficiency optimization problem in massive MIMO networks. The stationary points of the original problem are found by solving a sequence of successively refined approximate problems, and the proposed algorithm has the following advantages: 1) fast convergence as the structure of the original energy efficiency function is preserved as much as possible in the approximate problem, and 2) easy implementation as each approximate problem is natural for parallel computation and all variable updates have a closed-form expression. The proposed algorithm is guaranteed to converge and its advantages are also illustrated numerically.**

***Index Terms*—Energy efficiency, fractional programming, massive MIMO, parallel algorithm, successive convex approximation**

I. Introduction

The increase in capacity, increase in the number of connected devices, increase in reliability, decrease in latency and increase in efficiency anticipated by 2020 are some of the trends pushing the limits of 4G [1]. With the advent of 5G by 2020, the number of connected devices is predicted to reach 50 millions and target of the data rate increase is 1000x. The increase in the data rate is expected to achieve at the same or even a lower level of energy consumption. Therefore the so-called energy efficiency (EE) is a key performance indicator that attracts extensive attention and it imposes stringent requirements on efficient transmission schemes enhancing the spectral and energy efficiency.

Massive MIMO is identified to be an enabling technique to increase the spectral efficiency by orders of magnitude, because the hundreds or thousands of antenna elements at the base stations can provide a spatial resolution for hundreds of user equipment per cell at a low level of inter-user interference [2]. However, large arrays are only attractive if the antenna elements are of cheap hardware, and hardware impairment is thus more likely, e.g., amplifier nonlinearities, phase noise and quantization errors generating self-interference [3].

In this paper, we study the EE maximization problem in massive MIMO networks and we adopt the notion of EE to be the sum rate of all users divided by the consumed energy (bits/Hz/s/Joule). It is well known that sum rate maximization in interference-limited networks is a nonconvex and NP-hard problem [4]. The EE maximization problem in interference-limited massive MIMO networks is an even more challenging problem because the EE is a fractional function (with the consumed energy in the denominator) while the sum rate function in the numerator is a nonconcave function [5]. As a result, the mathematical tool of fractional programming is no longer applicable.

In state-of-the-art studies, the EE is usually optimized under the interference-free assumption by orthogonal transmission scheme [6] or successive interference cancellation [7] and the sum rate function in the numerator is thus concave. However, orthogonal transmission schemes may decrease the spectral efficiency, especially in dense networks. Other approaches maximizing the EE in interference-limited networks such as [8] are largely based on heuristics and the iterative algorithms proposed therein are not guaranteed to converge to stationary points of the nonconvex EE function.

An iterative algorithm with guaranteed convergence for EE maximization problem in interference-limited networks is recently proposed in [5]. This itera12.925tive algorithm consists of solving a sequence of approximate problems, with increasing resemblance to the original EE maximization problem. In each approximate problem, the nonconcave sum rate function is replaced by its concave global lower bound function, and the approximate problem can then be solved by fractional programming tools.

Despite the guaranteed convergence, the complexity of the iterative algorithm proposed in [5] is very high. On one hand, the lower bound function is constructed based on logarithmic approximation, while logarithmic functions are not amenable for numerical optimization. On the other hand, the approximate problem does not have other structures that can be exploited to enable, e.g., parallel computation, and it can only be solved by general purpose optimization solvers and this may incur a large latency in the decision making process. Furthermore, the lower bound of the allocated power must be strictly positive, which may not be practical in multi-carrier systems where no power is allocated to a carrier with strong multi-user interference.

In this paper, we develop an iterative algorithm that overcome the above difficulties. The algorithm is based on the idea of successive convex approximation, recently advanced in [9], [10] and further generalized in [11]. In each iteration, the proposed approximate function only needs to be pseudo-convex, a weak form of convexity, and this weak assumption makes it possible to preserve as much structure available in the original EE function as possible, e.g., the partial concavity in the numerator function and the division operator. Besides this, the proposed approximate function is natural for parallel computation. In particular, the approximate problem can be decomposed into many independent scalar problems that can be solved in parallel and each scalar problem has a closed-form solution. The computational complexity of each variable step is thus much lower than that of [5]. Based on the line of analysis in [11], we show that the proposed algorithm is guaranteed to converge. The advantage of the proposed algorithm is also illustrated numerically.

II. System Model and Problem Formulation

Consider the massive MIMO network with K cells and each cell serves one user with a total number of N resource blocks. The achievable transmission rate for user k at resource block n in the uplink can be formulated into the following form:

$$r_{k,n}(\mathbf{p}) \triangleq \log\left(1 + \frac{w_{kk,n}p_{k,n}}{\sigma_{k,n}^2 + \phi_{k,n}p_{k,n} + \sum_{j\neq k} w_{kj,n}p_{j,n}}\right), \tag{1}$$

where $p_{k,n}$ is the transmission power for user k at resource block n, $\sigma_{k,n}^2$ is the covariance of the additive noise at the receiver of user k, while $\{\phi_{k,n}\}_{k,n}$ and $\{w_{kj,n}\}_{k,j,n}$ are positive constants that depend on the channel conditions only. In particular, $\phi_{k,n}p_{k,n}$ is the self-interference that accounts for the hardware impairments and/or channel estimation error, and $\sum_{j\neq k} w_{kj,n}p_{j,n}$ accounts for the interference from other users [5]. The form in (1) also arises in other system models, such as relay-assisted CoMP interference network [12, Sec. 4.1].

In 5G wireless communication networks, the EE (the ratio of the sum rate and the sum power) is a key performance indicator. To address this issue, we look for the optimal power allocation that maximizes the EE:

$$\begin{array}{ll} \underset{\mathbf{p}}{\text{maximize}} & \dfrac{\sum_{k=1}^K \sum_{n=1}^N r_{k,n}(\mathbf{p})}{P_c + \sum_{k=1}^K \sum_{n=1}^N p_{k,n}} \\ \text{subject to} & \mathbf{p}_k \in \mathcal{P}_k,\ k = 1, \ldots, K, \end{array} \tag{2}$$

where $\mathbf{p}_k = (p_{k,n})_{n=1}^N$, $\mathbf{p} = (\mathbf{p}_k)_{k=1}^K$, and P_c is a positive constant representing the total circuit power dissipated in the network. It is straightforward to assign a weight factor $\alpha_{k,n} \geq 0$ for $r_{k,n}(\mathbf{p})$, but for the simplicity of presentation, we assume equal weights $\alpha_{k,n} = 1$ for all k and n. Note that $\mathcal{P}_k$ in (2) denotes the feasible power allocation strategies for user k:

$$\mathcal{P}_k \triangleq \{\mathbf{p}_k : \underline{\mathbf{p}}_k \leq \mathbf{p}_k \leq \bar{\mathbf{p}}_k,\ \sum_{n=1}^N p_{k,n} \leq P_k\}, \tag{3}$$

which consists of bound constraint ($\underline{\mathbf{p}}_k$ and $\overline{\mathbf{p}}_k$ is the lower and upper bound, respectively) and sum power constraint where P_k is the sum power budget. We assume $\mathcal{P}_k$ has a nonempty interior and problem (2) has a solution.

III. The Unified Successive Convex Approximation Framework

In this section, we give a brief review of the unified successive convex approximation method proposed in [11] to solve the following optimization problem:

$$\underset{\mathbf{x}\in\mathcal{X}}{\text{minimize}} \quad f(\mathbf{x}), \tag{4}$$

where $f(\mathbf{x}) : \mathcal{C}^n \to \mathcal{R}$ is a proper and differentiable function with a continuous gradient, and $\mathcal{X} \subseteq \mathcal{R}^n$ is a closed and convex set. We assume that problem (4) has a solution.

We start with the definition of *pseudo-convex functions:* A function $f(\mathbf{x})$ is said to be pseudo-convex if

$$f(\mathbf{y}) < f(\mathbf{x}) \Longrightarrow (\mathbf{y} - \mathbf{x})^T \nabla f(\mathbf{x}) < 0. \tag{5}$$

In other words, $f(\mathbf{y}) < f(\mathbf{x})$ implies $\mathbf{y} - \mathbf{x}$ is a *descent direction* of $f(\mathbf{x})$. We remark that the (strong) convexity of a function implies that the function is pseudo-convex, which in turn implies that the function is quasi-convex. That is:

$$\begin{array}{rcl} f(\mathbf{x}) \text{ is strongly convex} & \rightarrow & f(\mathbf{x}) \text{ is convex} \\ & & \downarrow \\ f(\mathbf{x}) \text{ is quasi-convex} & \leftarrow & f(\mathbf{x}) \text{ is pesudo-convex.} \end{array} \tag{6}$$

We solve (4) as a sequence of successively refined approximate problems, each of which is presumably much easier to solve than the original problem (4). In iteration t, let $\tilde{f}(\mathbf{x}; \mathbf{x}^t)$ be the approximate function of $f(\mathbf{x})$ around the point $\mathbf{x}^t$. Then the approximate problem is

$$\underset{\mathbf{x}\in\mathcal{X}}{\text{minimize}} \quad \tilde{f}(\mathbf{x}; \mathbf{x}^t).$$

Let us define the operator $\mathbb{B}\mathbf{x}^t$ as the minimizer of the approximate function $\tilde{f}(\mathbf{x}; \mathbf{x}^t)$ over $\mathbf{x} \in \mathcal{X}$:

$$\mathbb{B}\mathbf{x}^t \in \mathcal{S}(\mathbf{x}^t) \triangleq \left\{\mathbf{x}^\star \in \mathcal{X} :\ \tilde{f}(\mathbf{x}^\star; \mathbf{x}^t) = \min_{\mathbf{x}\in\mathcal{X}} \tilde{f}(\mathbf{x}; \mathbf{x}^t)\right\}. \tag{7}$$

It is shown in [11, Proposition 1] that $\mathbb{B}\mathbf{x}^t - \mathbf{x}^t$ is a descent direction of $f(\mathbf{x})$ at $\mathbf{x} = \mathbf{x}^t$ if the approximate function $\tilde{f}(\mathbf{x}; \mathbf{y})$ satisfies the following technical conditions:
(A1) $\tilde{f}(\mathbf{x}; \mathbf{y})$ is pseudo-convex in $\mathbf{x} \in \mathcal{X}$ for any $\mathbf{y} \in \mathcal{X}$;
(A2) $\tilde{f}(\mathbf{x}; \mathbf{y})$ is continuously differentiable in $\mathbf{x} \in \mathcal{X}$ for any given $\mathbf{y}$ and continuous in $\mathbf{y}$ for any $\mathbf{x} \in \mathcal{X}$;
(A3) The gradient of $\tilde{f}(\mathbf{x}; \mathbf{y})$ and the gradient of $f(\mathbf{x})$ are identical at $\mathbf{x} = \mathbf{y}$ for any $\mathbf{y} \in \mathcal{X}$, i.e., $\nabla_{\mathbf{x}}\tilde{f}(\mathbf{x}; \mathbf{y})\big|_{\mathbf{x}=\mathbf{y}} = \nabla_{\mathbf{x}} f(\mathbf{x})\big|_{\mathbf{x}=\mathbf{y}}$.

With the descent direction $\mathbb{B}\mathbf{x}^t - \mathbf{x}^t$, the vector update $\mathbf{x}^{t+1}$ in the $(t + 1)$-th iteration is defined as follows:

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \gamma^t(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t), \tag{8}$$

where $\gamma^t \in (0, 1]$ is an appropriate stepsize that is determined by the exact line search or successive line search.
Exact line search: The stepsize is selected such that the function $f(\mathbf{x})$ is decreased to the largest extent along the descent direction $\mathbb{B}\mathbf{x}^t - \mathbf{x}^t$:

$$\gamma^t \in \underset{0\leq\gamma\leq 1}{\arg\min}\ f(\mathbf{x}^t + \gamma(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)). \tag{9}$$

Successive line search: If no structure in $f(\mathbf{x})$ (e.g., convexity) can be exploited to efficiently compute γ^t according to the exact line search (9), the successive line search can instead be employed: given scalars $0 < \alpha < 1$ and $0 < \beta < 1$, the stepsize

Algorithm 1 The iterative convex approximation algorithm for problem (4)

```
Data: $t = 0$ and $\mathbf{x}^0 \in \mathcal{X}$; stop criterion $\delta$.
S1: Compute $\mathbb{B}\mathbf{x}^t$ according to (7).
S2: Determine the stepsize $\gamma^t$ by exact/successive line search.
S3: Update $\mathbf{x}$ according to (8).
S4: If $\left|(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)^T \nabla f(\mathbf{x}^t)\right| \leq \delta$, STOP; otherwise go to S1.
```

γ^t is set to be $\gamma^t = \beta^{m_t}$, where m_t is the smallest nonnegative integer m satisfying the following inequality:

$$f(\mathbf{x}^t + \beta^m(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)) \leq f(\mathbf{x}^t) + \alpha\beta^m(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)^T \nabla f(\mathbf{x}^t). \tag{10}$$

The existence of a finite m_t is always guaranteed if $(\mathbb{B}\mathbf{x}^t - \mathbf{x}^t)^T \nabla f(\mathbf{x}^t) < 0$ [13].

The algorithm is summarized in Algorithm 1. It is shown in [11, Theorem 1] that if:

(A4) The solution set $\mathcal{S}(\mathbf{x}^t)$ is nonempty for $t = 1, 2, \ldots$;

(A5) Given any convergent subsequence $\{\mathbf{x}^t\}_{t\in\mathcal{T}}$ where $\mathcal{T} \subseteq \{1, 2, \ldots\}$, the sequence $\{\mathbb{B}\mathbf{x}^t\}_{t\in\mathcal{T}}$ is bounded,

then any limit point of $\{\mathbf{x}^t\}$ is a stationary point of (4).

Apart from the standard assumptions (A2) and (A3) in successive convex optimization (cf. [9]), the approximate function $\tilde{f}(\mathbf{x}; \mathbf{x}^t)$ only needs to be pseudo-convex (cf. Assumption (A1)). As a result, it enables us to construct new approximate functions that can often be optimized more easily or even in closed-form. This results in a significant reduction of the computational cost if the approximate problem must otherwise only be optimized by iterative algorithms as in standard solvers and we show in the next section how this advantage can be exploited.

IV. Energy Efficiency Maximization in Massive MIMO Systems

Problem (2) is nonconvex and it is a NP-hard problem to find a globally optimal point [4]. Therefore we aim at finding a stationary point of (2) using the proposed algorithm. To begin with, we propose the following approximate function at $\mathbf{p} = \mathbf{p}^t$ in iteration t (we can set $\mathbf{p}^0 = \underline{\mathbf{p}}$):

$$\tilde{f}(\mathbf{p}; \mathbf{p}^t) = \frac{\sum_{k=1}^K \sum_{n=1}^N \tilde{r}_{k,n}(p_{k,n}; \mathbf{p}^t)}{P_c + \sum_{k=1}^K \sum_{n=1}^N p_{k,n}}, \tag{11}$$

where

$$\begin{aligned}\tilde{r}_{k,n}(p_{k,n}; \mathbf{p}^t) \triangleq{}& r_{k,n}(p_{k,n}, \mathbf{p}^t_{-k,n}) \\ &+ \sum_{j\neq k}(r_{j,n}(\mathbf{p}^t) + (p_{k,n} - p^t_{k,n})\nabla_{p_{k,n}} r_{j,n}(\mathbf{p}^t))\end{aligned}$$

and $\mathbf{p}_{-k,n} \triangleq (p_{j,n})_{j=1,j\neq k}^N$. Note that the numerator function of $\tilde{f}(\mathbf{p}; \mathbf{p}^t)$ consists of KN separable component functions, one for each scalar variable $p_{k,n}$, and $\tilde{r}_{k,n}(p_{k,n}; \mathbf{p}^t)$ is constructed as follows: since $r_{k,n}(\mathbf{p})$ is concave in $p_{k,n}$ (shown shortly later in the right column of this page) but $r_{j,n}(\mathbf{p})$ is not concave in $p_{k,n}$ for $j \neq k$ (as a matter of fact, $r_{j,n}(\mathbf{p})$ is convex in $p_{k,n}$ for $j \neq k$), the concave function $r_{k,n}(p_{k,n}, \mathbf{p}^t_{-k,n})$ is preserved in $\tilde{r}_{k,n}(p_{k,n}; \mathbf{p}^t)$ with $\mathbf{p}_{-k,n}$ fixed to be $\mathbf{p}^t_{-k,n}$ while the nonconcave functions $(r_{j,n}(\mathbf{p}))_{j\neq k}$ are linearized w.r.t. $p_{k,n}$ at $\mathbf{p} = \mathbf{p}^t$. In this way, the partial concavity in $r_k(\mathbf{p})$ is preserved. Similarly, since $P_c + \sum_{n=1}^N \sum_{j=1}^K p_{k,n}$ in the denominator is linear in $\mathbf{p}$ and thus left intact. Furthermore, the division operator in the original problem (2) is kept in the approximate function (11). Although it will destroy the concavity of the approximate function $\tilde{f}(\mathbf{p}; \mathbf{p}^t)$ in (11) is not a concave function[1], it presents the pseudo-concavity as we show in two steps.

Step 1: The function $r_{k,n}(p_{k,n}, \mathbf{p}^t_{-k,n})$ is concave in $p_{k,n}$. For the simplicity of notation, we define two constants $c_1 \triangleq w_{kk,n}/\phi_{k,n} > 0$ and $c_2 \triangleq (\sigma^2_{k,n} + \sum_{j\neq k} w_{kj,n} p^{(t)}_{j,n})/\phi_{k,n} > 0$. The first-order derivative and second-order derivative of $r_{k,n}(p_{k,n}, \mathbf{p}^t_{-k,n})$ w.r.t. $p_{k,n}$ are

$$\begin{aligned}\nabla_{p_{k,n}} r_{k,n}(p_{k,n}, \mathbf{p}^t_{-k,n}) &= \frac{1 + c_1}{(1 + c_1)p_{k,n} + c_2} - \frac{1}{p_{k,n} + c_2}, \\ \nabla^2_{p_{k,n}} r_{k,n}(p_{k,n}, \mathbf{p}^t_{-k,n}) &= \frac{1}{(p_{k,n}+c_2)^2} - \frac{(1 + c_1)^2}{((1+c_1)p_{k,n}+c_2)^2} \\ &= -\frac{2c_1c_2p_{k,n}(1 + c_1) + (c_1^2 + 2c_1)c_2^2}{((1 + c_1)p_{k,n} + c_2)^2(p_{k,n} + c_2)^2}.\end{aligned}$$

Since $\nabla^2_{p_{k,n}} r_{k,n}(p_{k,n}, \mathbf{p}^t_{-k,n}) < 0$ when $p_{k,n} \geq 0$, $r_{k,n}(p_{k,n}, \mathbf{p}^t_{-k,n})$ is a concave function of $p_{k,n}$ in the nonnegative axis $p_{k,n} \geq 0$ [14].

Step 2: Given the concavity of $r_{k,n}(p_{k,n}, \mathbf{p}^t_{-k,n})$, the function $\tilde{r}_{k,n}(p_{k,n}; \mathbf{p}^t)$ is concave in $p_{k,n}$. Since the component functions $(r_{k,n}(p_{k,n}; \mathbf{p}^t))_{k,n}$ are separable in $(p_{k,n})_{k,n}$, the numerator function of $\tilde{f}(\mathbf{p}; \mathbf{p}^t)$, namely, $\sum_{k=1}^K \sum_{n=1}^N \tilde{r}_{k,n}(p_{k,n}; \mathbf{p}^t)$, is concave in $\mathbf{p}$. Since the denominator function of $\tilde{f}(\mathbf{p}; \mathbf{p}^t)$ is a convex function of $\mathbf{p}$, it follows from [15, Lemma 3.8] that $\tilde{f}(\mathbf{p}; \mathbf{p}^t)$, which represents a quotient of a concave and a nonnegative convex function, is pseudo-concave. Therefore Assumption (A1) is satisfied.

Then we verify that the gradient of the approximate function and that of the original objective function are identical at $\mathbf{p} = \mathbf{p}^t$. It follows that

$$\begin{aligned}&\nabla_{p_{k,n}} \tilde{f}(\mathbf{p}; \mathbf{p}^t)\Big|_{\mathbf{p}=\mathbf{p}^t} \\ &= \nabla_{p_{k,n}} \left(\frac{\tilde{r}_{k,n}(p_{k,n}; \mathbf{p}^t)}{P_c + \sum_{j=1}^K \sum_{m=1}^N p_{j,m}}\right)\Bigg|_{p_{k,n}=p^t_{k,n}} \\ &= \sum_{j=1}^K \frac{\nabla_{p_{k,n}} r_{j,n}(\mathbf{p}^t)(P_c + \sum_{j=1}^K \sum_{m=1}^N p^t_{j,m}) - r_{j,n}(\mathbf{p}^t)}{(P_c + \sum_{j=1}^K \sum_{m=1}^N p^t_{j,m})^2} \\ &= \nabla_{p_{k,n}} \left(\frac{\sum_{j=1}^K \sum_{m=1}^N r_{j,m}(\mathbf{p})}{P_c + \sum_{j=1}^K \sum_{m=1}^N p_{j,m}}\right)\Bigg|_{\mathbf{p}=\mathbf{p}^t}, \ \forall\, k,\, n.\end{aligned}$$

Therefore Assumption (A3) is satisfied. Since both the numerator function and the (nonzero) denominator function of $\tilde{f}(\mathbf{p}; \mathbf{p}^t)$ are continuously differentiable, Assumption (A2) is satisfied.

[1] A concave function divided by a linear function is no longer a concave function.

$$p_{k,n}(\lambda_\tau^t) = \left[\mathtt{int}_{k,n}(\mathbf{p}^t) \frac{\sqrt{(2\phi_{k,n} + w_{kk,n})^2 - 4\phi_{k,n}\left(\frac{w_{kk,n}}{(\pi_{k,n}(\mathbf{p}^t) - \lambda_\tau^t - \mu_\tau^t(k))\mathtt{int}_{k,n}(\mathbf{p}^t)} + 1\right)} - 1}{2\phi_{k,n}(\pi_{k,n}(\mathbf{p}^t) - \lambda_\tau^t - \mu_\tau^t(k))(\phi_{k,n} + w_{kk,n})} \right]_{\underline{p}_{k,n}}^{\overline{p}_{k,n}}. \tag{17}$$

Given the approximate function (11), the approximate problem in iteration t is thus

$$\mathbb{B}\mathbf{p}^t = \underset{(\mathbf{p}_k \in \mathcal{P}_k)_{k=1}^K}{\arg\max} \; \frac{\sum_{k=1}^K \sum_{n=1}^N \tilde{r}_{k,n}(p_{k,n}; \mathbf{p}^t)}{P_c + \sum_{k=1}^K \sum_{n=1}^N p_{k,n}}, \tag{12}$$

where $\mathcal{P}_k$ denotes the feasible power allocation strategies for user k as defined in (3). Since $\mathcal{P}_k$ is a closed and bounded set for all $k = 1, \ldots, K$, Assumptions (A4) and (A5) are satisfied.

Since the objective function in (2) is nonconvex, it may not be computationally affordable to perform the exact line search. Instead, the successive line search can be applied to calculate the stepsize. As a result, the convergence of the proposed algorithm with approximate problem (12) and successive line search readily follows from [11, Theorem 1].

The maximization problem in (12) is a fractional programming problem and thus can be solved iteratively by the Dinkelbach's algorithm, cf. [5, Algorithm 5]: given λ_τ^t, the following optimization problem is solved in iteration $\tau + 1$:

$$\mathbf{p}(\lambda_\tau^t) = \underset{\{\mathbf{p}_k \in \mathcal{P}_k\}_{k=1}^K}{\arg\max} \left\{ \begin{array}{c} \sum_{k=1}^K \sum_{n=1}^N \tilde{r}_{k,n}(p_{k,n}; \mathbf{p}^t) \\ -\lambda_\tau^t (P_c + \sum_{k=1}^K \sum_{n=1}^N p_{k,n}) \end{array} \right\}. \tag{13}$$

The variable λ_τ^t is then updated in iteration $\tau + 1$ as

$$\lambda_{\tau+1}^t = \frac{\sum_{k=1}^K \sum_{n=1}^N \tilde{r}_{k,n}(p_{k,n}(\lambda_\tau^t); \mathbf{p}^t)}{P_c + \sum_{k=1}^K \sum_{n=1}^N p_{k,n}(\lambda_\tau^t)}. \tag{14}$$

It follows from the convergence properties of the Dinkelbach's algorithm that

$$\lim_{\tau \to \infty} \mathbf{p}(\lambda_k^{t,\tau}) = \mathbb{B}\mathbf{p}^t$$

at a superlinear convergence rate.

It is easy to see that the optimization problem in (13) can naturally be decomposed into independent subproblems that can be solved in parallel: $\mathbf{p}(\lambda_\tau^t) = (\mathbf{p}_k(\lambda_\tau^t))_{k=1}^K$ and for all $k = 1, \ldots, K$,

$$\mathbf{p}_k(\lambda_\tau^t) = \underset{\mathbf{p}_k \in \mathcal{P}_k}{\arg\max} \left\{ \sum_{n=1}^N \tilde{r}_{k,n}(p_{k,n}; \mathbf{p}^t) - \lambda_\tau^t \sum_{n=1}^N p_{k,n} \right\}. \tag{15}$$

Since the optimization problem in (15) is convex and $\mathcal{P}_k$ has an nonempty interior (by assumption), strong duality holds [14] and the optimization problem can be further decomposed in the dual domain by relaxing the sum power constraint $\sum_{n=1}^N p_{k,n} \leq P_k$ in $\mathcal{P}_k$ into the Lagrangian [16]: $\mathbf{p}_k(\lambda_\tau^t) = (p_{k,n}(\lambda_\tau^t))_{n=1}^N$ and for all $n = 1, \ldots, N$,

$$p_{k,n}(\lambda_\tau^t) = \underset{\underline{p}_{k,n} \leq p_{k,n} \leq \overline{p}_{k,n}}{\arg\max} \; \tilde{r}_{k,n}(p_{k,n}; \mathbf{p}^t) - \lambda_\tau^t p_{k,n} - \mu_\tau^t(k) p_{k,n}, \tag{16}$$

where $\mu_\tau^t(k)$ is the optimal dual variable that is associated with the sum power constraint $\sum_{n=1}^N p_{k,n} \leq P_k$ and that satisfies the complementary slackness condition $0 \leq \mu_\tau^t(k) \perp \sum_{n=1}^N p_{k,n}(\lambda_\tau^t) - P_k \leq 0$ ($a \perp b$ means $a \cdot b = 0$). The variable $\mu_\tau^t(k)$ is related to the water level as in the classic waterfilling algorithm that determines the amount of power allocated to each resource block [17] and can easily be found by the efficient bisection method [16], [18]. Note that $p_{k,n}(\lambda_\tau^t)$ can be expressed in closed-form, as it is simply the projection of the point that sets the gradient of the objective function in (16) to zero onto the interval $[\underline{p}_{k,n}, \overline{p}_{k,n}]$, and it can be verified that finding that point is equivalent to finding the root of a polynomial with order 2. We omit the detailed derivations and directly give the expression of $p_{k,n}(\lambda_k^{t,\tau})$ in (17) at the top of this page, where $\pi_{k,n}(\mathbf{p}^t) \triangleq \sum_{j \neq k} \nabla_{p_{k,n}} r_{j,n}(\mathbf{p}^t)$ and $\mathtt{int}_{k,n}(\mathbf{p}^t) \triangleq \sigma_{k,n}^2 + \sum_{j \neq k} w_{kj,n} p_{j,n}^t$. The above procedures are summarized in Algorithm 2.

Algorithm 2 The iterative convex approximation algorithm for the EE maximization problem (2)

```
S0: $t = 0$ and $\mathbf{p}^0 = \underline{\mathbf{p}}$; stop criterion $\delta_p$ and $\delta_\lambda$.
S1: Compute $\mathbb{B}\mathbf{p}^t$ according to S1.0-S1.3:
    S1.0 $\tau = 0$ and $\lambda_0^t = 0$;
    S1.1 Compute $p_{k,n}(\lambda_\tau^t)$ according to (17), where $\mu_\tau^t(k)$
         is found by the bisection method.
    S1.2 Compute $\lambda_{\tau+1}^t$ according to (14).
    S1.3 If $|\lambda_{\tau+1}^t - \lambda_\tau^t| \leq \delta_\lambda$, go to S2; Otherwise $\tau \leftarrow \tau + 1$
         and go to S1.1.
S2: Determine the stepsize $\gamma^t$ by successive line search.
S3: Update $\mathbf{p}$ according to (8).
S4: If $\left|(\mathbb{B}\mathbf{p}^t - \mathbf{p}^t)^T \nabla f(\mathbf{p}^t)\right| \leq \delta_p$, STOP; otherwise $t \leftarrow t + 1$
    and go to S1.
```

Some comments on Algorithm 2 are drawn as follows:

On the approximate function: In the approximate function (11), the desirable structure in the original objective function is preserved as much as possible, namely, the partial concavity in the numerator function $\sum_{n=1}^N \sum_{k=1}^K r_{k,n}(\mathbf{p})$, the linearity in the denominator function $P_c + \sum_{n=1}^N \sum_{k=1}^K p_{k,n}$ and the division operator. Therefore, the proposed algorithm is of a best-response nature and expected to converge faster than plain gradient based methods which linearizes the original objective function $\sum_{j=1}^K \sum_{n=1}^N r_{k,n}(\mathbf{p}) / (P_c + \sum_{j=1}^K \sum_{n=1}^N p_{k,n})$ completely in each iteration, as observed in some previous works [9], [10], [18]. However, the convergence of Algorithm 2 cannot be derived from existing works [9], [10], since the approximate function (11) presents only a weak form of convexity, namely, the pseudo-convexity, which is much weaker than those required in state-of-the-art convergence analysis, e.g., uniform strong convexity in [9], [10].

On the parallel implementation: In each iteration of the proposed algorithm, the approximate problem (12) is decomposed into many independent scalar problems (16) that can then be solved in parallel. This leads to a natural exploitation of parallel computing architecture that is nowadays extensively used to accelerate the computation.

On the computation: The variable update in each iteration has a simple closed-form expression (up to a Lagrange multiplier that can be found by the bisection method), cf. (17). On one hand, the convergence speed is greatly enhanced than state-of-the-art method [5] as we will illustrate numerically. On the other hand, only basic algebraic operations are required at each processor and the hardware cost is significantly reduced. The massive deployment of the proposed scheme at the base stations is thus made possible.

On the structure of the proposed algorithm: The proposed algorithm has three layers: outer layer with index t, middle layer with index τ and inner layer where the bisection method is implemented to search the optimal dual variable $\mu_\tau^t(k)$. Since the algorithms in the inner and middle layers converge very fast (typically in a few iterations) and the convergence speed is independent of the problem dimension (the dimension of $\mathbf{p}$), and each update has a simple closed-form expression, the convergence speed is still very competitive as we show numerically. In the case of a single resource block ($N = 1$), $\mu_\tau^t(k) = 0$ and the inner layer consists of one iteration only. A centralized coordinator is foreseen to coordinate the information exchange, and this can be performed by the base stations.

Simulations: We consider a single resource block, i.e., $N = 1$ and the subscript n is dropped for simplicity of presentation. The number of antennas at the BS in each cell is $M = 50$, and the channel from user j to cell k is $\mathbf{h}_{kj} \in \mathbb{C}^{M\times 1}$. We assume a similar setup as [5]: $w_{kk} = \left|\mathbf{h}_{kk}^H\mathbf{h}_{kk}\right|^2$, $w_{kj} = \left|\mathbf{h}_{kk}^H\mathbf{h}_{kj}\right|^2 + \epsilon\mathbf{h}_{kk}^H\mathbf{D}_j\mathbf{h}_{kk}$ for $j \neq k$ and $\phi_k = \epsilon\mathbf{h}_{kk}^H\mathbf{D}_k\mathbf{h}_{kk}$, where $\epsilon = 0.01$ is the error magnitude of hardware impairments at the BS and $\mathbf{D}_j = \text{diag}(\{|h_{jj}(m)|^2\}_{m=1}^M)$. The noise covariance $\sigma_k^2 = 1$, and the hardware dissipated power p_c is 10dBm, while $\underline{p}_k$ is -10dBm and $\overline{p}_k$ is 10dBm for all users. The benchmark algorithm is [5, Algorithm 1], which successively maximizes the following lower bound function of the objective function in (2), which is tight at $\mathbf{p} = \mathbf{p}^t$:

$$\begin{aligned} \underset{\mathbf{q}}{\text{maximize}}\ & \frac{\sum_{k=1}^K b_k^t + a_k^t \log w_{kk}}{P_c + \sum_{k=1}^K e^{q_k}} + \\ & \frac{\sum_{k=1}^K a_k^t (q_k - \log(\sigma_k^2 + \phi_k e^{q_k} + \sum_{j\neq k} w_{kj} e^{q_j}))}{P_c + \sum_{k=1}^K e^{q_k}} \\ \text{subject to}\ & \log(\underline{p}_k) \leq q_k \leq \log(\overline{p}_k),\ k = 1,\ldots,K, \end{aligned} \tag{18}$$

where

$$a_k^t \triangleq \frac{\text{sinr}_k(\mathbf{p}^t)}{1 + \text{sinr}_k(\mathbf{p}^t)},$$

$$b_k^t \triangleq \log(1 + \text{sinr}_k(\mathbf{p}^t)) - \frac{\text{sinr}_k(\mathbf{p}^t)}{1 + \text{sinr}_k(\mathbf{p}^t)}\log(\text{sinr}_k(\mathbf{p}^t)),$$

and

$$\text{sinr}_k(\mathbf{p}) \triangleq \frac{w_{kk}p^t}{\sigma_k^2 + \phi_k p_k + \sum_{j\neq k} w_{kj}p_j}.$$

Denote the optimal variable of (18) as $\mathbf{q}^t$ (which can be found by the Dinkelbach's algorithm); then the variable $\mathbf{p}$ is updated as $p_k^{t+1} = e^{q_k^t}$ for all $k = 1,\ldots,K$. We thus coin [5, Algorithm 1] as the successive lower bound maximization (SLBM) method.

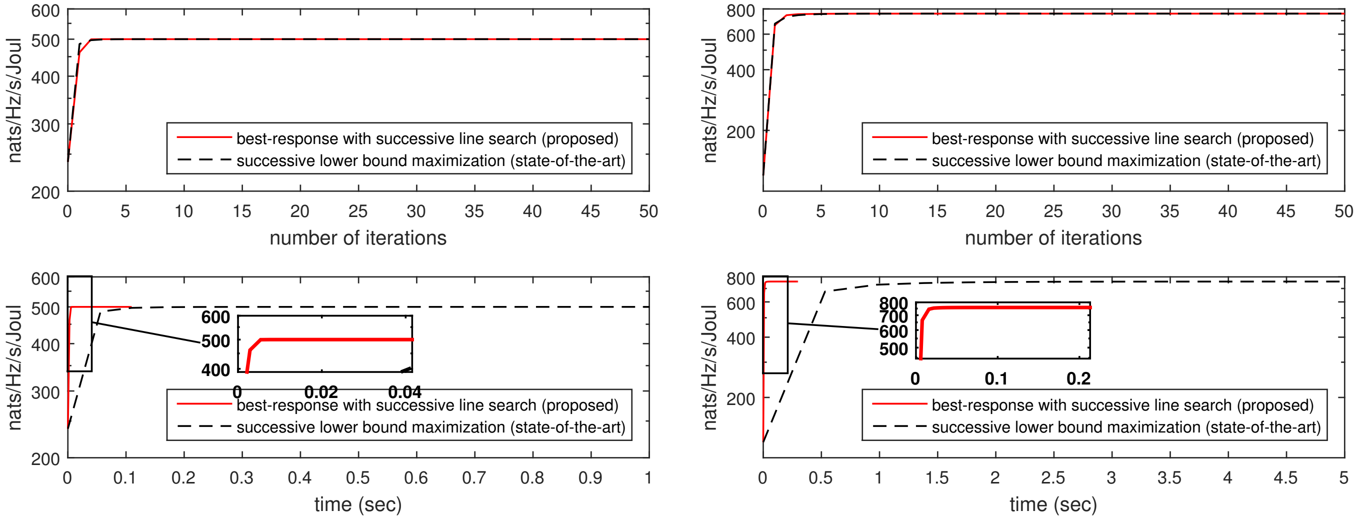
In Fig. 1, we compare the convergence behavior of the proposed method and the SLBM method in terms of both the number of iterations (the upper subplots) and the CPU time (the lower subplots), for two different number of users: $K = 10$ in Fig. 1 (a) and $K = 50$ in Fig. 1 (b). It is obvious that the convergence speed of the proposed algorithm in terms of the number of iterations is comparable to that of the SLBM method. However, we remark that the approximate problem (12) of the proposed algorithm is superior to that of the SLBM method in the following aspects:

Firstly, the approximate problem of the proposed algorithm consists of independent subproblems that can be solved in parallel, cf. (16), while each subproblem has a closed-form solution, cf. (17). However, the optimization variable in the approximate problem of the SLBM method (18) is a vector $\mathbf{q} \in \mathbb{R}^{K\times 1}$ and the approximate problem can only be solved by a general purpose solver.

In the simulations, we use the Matlab optimization toolbox to solve (18) and the iterative update specified in (13)-(14) to solve (12), where the solution of (13) is given by (16) (with $\mu_\tau^t(k) = 0$ for all t, τ, k) and the stopping criterion for (14) is $\left\|\boldsymbol{\lambda}_\tau^t\right\|_\infty \leq 10^{-5}$. The upper subplots in Fig. 1 show that the numbers of iterations required for convergence is approximately the same for the SLBM method when $K = 10$ in Fig. 1 (a) and when $K = 50$ in Fig. 1 (b). However, we see from the lower subplots in Fig. 1 that the CPU time of each iteration of the SLBM method is dramatically increased when K is increased from 10 to 50. On the other hand, the CPU time of the proposed algorithm is not notably changed because the operations are parallelizable[2] and the required CPU time is thus not affected by the problem size.

Secondly, since a variable substitution $q_k = e^{p_k}$ is adopted in the SLBM method (we refer to [5] for more details), the lower bound constraint $\underline{p}_k = 0$ (which corresponds to $q_k = -\infty$) cannot be handled by the SLBM method numerically.

[2] By stacking the $p_k(\lambda_\tau^t)$'s into the vector form $\mathbf{p}(\lambda_\tau^t) = (p_k(\lambda_\tau^t))_{k=1}^K$ we can see that only element wise operations between vectors and matrix vector multiplications are involved. The simulations on which Fig. 1 are based are not performed in a real parallel computing environment with K processors, but only make use of the efficient linear algebraic implementations available in Matlab which already implicitly admits a certain level of parallelism.



(a) Number of users: $K = 10$

(b) Number of users: $K = 50$

Fig. 1. EE Maximization: achieved EE versus the number of iterations

This limitation impairs the applicability of the SLBM method in many practical scenarios.

V. Conclusions

In this paper, we have proposed a novel iterative algorithm based on successive convex approximation for the nonconvex energy efficiency optimization problem in massive MIMO networks. The proposed algorithm is guaranteed to converge to a stationary point of the nonconvex EE maximization problem, and we have shown its advantages from the theoretical perspective: fast convergence and easy implementation as the approximate problems are natural for parallel computation and all variable updates have a closed-form expression. These advantages are finally illustrated by numerical simulations.

Acknowledgement

The work of Yang is supported by the framework of the Horizon 2020 project FANTASTIC-5G (ICT-671660). The work of Pesavento is supported by the Seven Framework for Research of the European Commission under grant number: ADEL 619647.

References

[1] ITU-R, "IMT vision – framework and oveall objectives of the future development of imt for 2020 and beyond," Tech. Rep., 9 2015.

[2] E. Björnson, E. G. Larsson, and M. Debbah, "Massive mimo for maximal spectral efficiency: How many users and pilots should be allocated?" *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1293–1308, Feb 2016.

[3] E. Björnson, J. Hoydis, M. Kountouris, and M. Debbah, "Massive mimo systems with non-ideal hardware: Energy efficiency, estimation, and capacity limits," *IEEE Transactions on Information Theory*, vol. 60, no. 11, pp. 7112–7139, Nov 2014.

[4] Z.-Q. Luo and S. Zhang, "Dynamic Spectrum Management: Complexity and Duality," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 57–73, Feb. 2008.

[5] A. Zappone, L. Sanguinetti, G. Bacci, E. Jorswieck, and M. Debbah, "Energy-efficient power control: A look at 5g wireless technologies," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1668–1683, April 2016.

[6] Q. Xu, X. Li, H. Ji, and X. Du, "Energy-efficient resource allocation for heterogeneous services in ofdma downlink networks: Systematic perspective," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2071–2082, Jun 2014.

[7] J. Xu and L. Qiu, "Energy efficiency optimization for mimo broadcast channels," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 690–701, February 2013.

[8] S. He, Y. Huang, L. Yang, and B. Ottersten, "Coordinated multicell multiuser precoding for maximizing weighted sum energy efficiency," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 741–751, Feb 2014.

[9] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang, "Decomposition by Partial Linearization: Parallel Optimization of Multi-Agent Systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 641–656, Feb. 2014.

[10] Y. Yang, G. Scutari, P. Song, and D. P. Palomar, "Robust MIMO Cognitive Radio Systems Under Interference Temperature Constraints," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 11, pp. 2465–2482, Nov. 2013.

[11] Y. Yang and M. Pesavento, "A unified successive pseudo-convex approximation framework," Jun. 2015, submitted to *IEEE Transactions on Signal Processing*. [Online]. Available: http://arxiv.org/abs/1506.04972

[12] A. Zappone and E. Jorswieck, "Energy efficiency in wireless networks via fractional programming theory," *Foundations and Trends in Communications and Information Theory*, vol. 11, no. 3-4, pp. 185–396, 6 2015.

[13] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

[14] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ Pr, 2004.

[15] C. Olsson, A. P. Eriksson, and F. Kahl, "Efficient optimization for L_∞-problems using pseudoconvexity," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[16] W. Yu, "Sum-capacity computation for the Gaussian vector broadcast channel via dual decomposition," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 754–759, Feb. 2006.

[17] R. Gallager, *Information Theory and Reliable Communication*. New York: John Wiley and Sons, 1968.

[18] Y. Yang, G. Scutari, D. P. Palomar, and M. Pesavento, "A parallel approximation method for nonconvex stochastic multi-agent optimization problems," Dec. 2015, to appear in *IEEE Transactions on Signal Processing*. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7412752