ABSTRACT

The min-max problem, also known as the saddle point problem, can be used to formulate a wide range of applications in signal processing and wireless communication. However, existing optimization theory and methods, which mostly deal with problems with certain convex-concave structure, are not applicable for the aforementioned applications, which oftentimes involve non-convexity. In this work, we consider a general block-wise one-sided non-convex min-max problem, in which the minimization problem consists of multiple blocks and is non-convex, while the maximization problem is (strongly) concave. We propose two simple algorithms, which alternatingly perform one gradient descent-type step for each minimization block and one gradient ascent-type step for the maximization problem. For the first time, we show that such simple alternating min-max algorithms converge to first-order stationary solutions. We conduct numerical tests on a robust learning problem, and a wireless communication problem in the presence of jammers, to validate the efficiency of the proposed algorithms.

1. INTRODUCTION

Consider the following min-max (a.k.a. saddle point) problem:

\[
\min_{x} \max_{y} \ f(x_1, x_2, \ldots, x_K; y) + \sum_{i=1}^{K} h_i(x_i) - g(y)
\]

s.t. \( x_i \in X_i \), \( y \in Y \), \( i = 1, \ldots, K \)

where \( f \) is a continuously differentiable function; \( h_i \) and \( g \) are some convex possibly non-smooth function; \( x := [x_1; \ldots; x_K] \) and \( y \) are the block optimization variables; \( X_i \)‘s and \( Y \) are some convex feasible sets. We call the problem one-sided non-convex problem because we assume that \( f(x, y) \) is non-convex w.r.t. \( x \), and (strongly) concave w.r.t. \( y \). The challenge is to design effective algorithms that can deal with the non-convexity in the problem.

Problem (1) is quite general. It arises in many signal processing, communication and networking applications, as listed below.

1.1. Motivating Examples

Distributed non-convex optimization: Consider a network of \( K \) agents defined by a connected graph \( G \triangleq \{V, E\} \), where each agent \( i \) can communicate with its neighbors. The agents solve the following problem, \( \min_{x \in \mathbb{R}^d} \sum_{i=1}^{K} (f_i(y) + h_i(y)) \), where each \( f_i(y) : \mathbb{R}^d \to \mathbb{R} \) is a non-convex, smooth function, and \( h_i(y) : \mathbb{R}^d \to \mathbb{R} \) is convex non-smooth function that plays the role of the regularizer. Each agent \( i \) has access only to \( f_i, h_i \). Denote \( x \), as agent \( i \)’s local copy of \( y \), then we have the following equivalent formulation

\[
\min_{x \in \mathbb{R}^d} \ f(x) + h(x) = \sum_{i=1}^{K} (f_i(x_i) + h_i(x_i)) \quad \text{s.t. } (A \otimes I_d)x = 0,
\]

where \( x = [x_1; \ldots; x_K] \in \mathbb{R}^{Kd} \) and \( A \in \mathbb{R}^{d \times K} \) is the incidence matrix, i.e., assuming that the edge \( e \) is incident on vertices \( i \) and \( j \), with \( i > j \) we have that \( A_{ei} \equiv 1, A_{ej} \equiv -1 \) and \( A_{ce} \equiv 0 \) for all other vertices; \( \otimes \) denotes the Kronecker product. This task captures the formulation of many problems that appear in distributed machine learning and signal processing (e.g., \([1–4]\)).

Using duality theory we can rewrite the above problem as

\[
\min_{x \in \mathbb{R}^{Kd}} \max_{\lambda \in \mathbb{R}^d} \ f(x) + h(x) + \langle \lambda, (A \otimes I_d)x \rangle
\]

where \( \lambda \) are the multipliers. Clearly (2) is in the form of (1).

Robust learning over multiple domains: In [5] the authors introduce a robust optimization framework, in which training sets from \( K \) different domains are used to train a machine learning model. Let \( S_k \triangleq \{ (x_i^k, y_i^k) \} \), \( 1 \leq k \leq K \) be the individual training sets with \( x_i^k \in \mathbb{R}^n \), \( y_i^k \in \mathbb{R} \); \( w \) be the parameter of the model we intend to learn, \( l(w) \) a non-negative loss function and \( f_k(w) = \frac{1}{|S_k|} \sum_{i=1}^{|S_k|} l(x_i^k, y_i^k, w) \) the non-convex (in general) empirical risk in the \( k \)-th domain. The following problem formulates the task of finding the parameter \( w \) that minimizes the empirical risk, while taking into account the worst possible distribution over the \( K \) different domains:

\[
\min_{w \in \mathbb{R}^d} \max_{p \in \Delta} p^T F(w),
\]

where \( F(w) := [f_1(w), \ldots, f_K(w)]^T \), \( p \) describes the adversarial distribution over the different domains and \( \Delta := \{p \in \mathbb{R}^K \mid 0 \leq p_i \leq 1, i = 1. \ldots, K, \sum_{i=1}^{K} p_i = 1\} \) is the standard simplex. Furthermore, it is also common to add a regularizer that imposes structures on the adversarial distribution (\( \lambda > 0 \) is some parameter):

\[
\min_{w \in \mathbb{R}^d} \max_{p \in \Delta} p^T F(w) - \frac{\lambda}{2} D(p || q)
\]

where \( D \) is some distance between probability distributions and \( q \) is some prior probability distribution. The objective function consists of a linear coupling between \( p \) and the non-convex function \( F(w) \) and a convex or strongly convex regularizer.

Power control problem: Consider a problem in wireless transceiver design, where \( K \) transmitter-receiver pairs transmit over \( N \) channels to maximize their minimum rates. User \( k \) transmits with power \( p_k := [p_k^1; \ldots; p_k^N] \), and its rate is given by: \( R_k(p_1, \ldots, p_K) = \sum_{n=1}^{N} \log \left( 1 + \frac{a_{nk}^i p_k^i}{\sigma^2 + \sum_{j=1,j \neq k}^K a_{nk}^j p_j^i} \right) \) (assuming Gaussian signaling), which is a non-convex function on \( p := [p_1; \ldots; p_K] \). Here \( a_{nk}^i \)‘s denote the channel gain between pair \( (\ell, k) \) on \( n \)-th channel, and \( \sigma^2 \) is the noise power. The classical max-min fair power control problem is: \( \max_{p \in \mathbb{R}^+} \ min_{y \in \Delta} R_k(p_1, \ldots, p_K) \times y_k \), where \( \Delta := \{p \mid 0 \leq p_k \leq p, \forall k \} \) denotes the feasible power allocations. The above max-min rate problem can be equivalently formulated as (1):

\[
\min_{\sum_{k=1}^{K} R_k(p_1, \ldots, p_K) \times y_k}
\]
where the set $\Delta$ is again the standard simplex. Note that $R_k(p)$ is a non-convex function in $p$, and there is a linear coupling between $y$ and the set of functions $\{R_k(p)\}_{k=1}^K$ in the objective function.

**Power control in the presence of a jammer:** Consider an extension of the above scenario (which is first described in [6]), where a jammer participates in a $K$-user $N$-channel interference channel transmission. Differently from a regular user, the jammer’s objective is to reduce the total sum-rate of the other users by properly transmitting noises. Because there are $N$ channels, we use $p_k^n$ to denote user $k$th transmission on $n$th channel. The corresponding sum-rate maximization-minimization problem can be formulated as:

$$\min_{p_k} \max_{p_0} \sum_{k,n} \log \left( 1 + \frac{a_{kk}^np_k^n p_0^k}{\sigma^2 + \sum_{j=1, j \neq k} a_{kj}^p p_j^n + a_{0k}^p p_0^k} \right),$$  \hspace{1cm} (6)

where $p_k$ and $p_0$ are the power allocation of user $k$ and the jammer, respectively; the set $\mathcal{P} := \mathcal{P}_1 \times \ldots \times \mathcal{P}_K$, where $\mathcal{P}_k$ are defined similarly as before.

### 1.2. Related Work

Motivated by these applications, it is of interest to develop efficient algorithms for solving these problems with theoretical convergence guarantees. There has been a long history of studying the min-max optimization problems. When the problem is convex-concave, previous works [7–10] and the references therein have shown that certain primal-dual type algorithms, which alternate between the update of $x$ and $y$ variables, can solve the convex-concave saddle problem optimally. However, when the problem is non-convex, the convergence behavior of the primal-dual algorithm has not been well understood. One challenge is that the primal and dual variables are coupled in a general nonlinear way, therefore it is difficult to characterize the progress of the iterates as the algorithm proceeds.

Although there are lots of recent works on the non-convex minimization problems [11], only few works focus on the non-convex min-max problems. For example, a robust optimization from multiple distributions is proposed recently in [5], where the coupling between the iterates in the minimization and maximization problems is linear and the variable of the minimization problem is unconstrained. A simple min-max algorithm in the stochastic settings is also considered, which updates the iterates of the both problems independently and converges to the stationary point of the min-max problems; please see [5, Theorem 1] for details about the convergence rate. In [12], a proximally guided stochastic mirror descent method (PG-SMD) is proposed, which also optimizes the minimization and maximization components separately and updates the corresponding iterates at the same time. From the convergence analysis, it turns out that PG-SMD is provable to converge to an approximate stationary point of the minimization problem. Recently, an oracle based non-convex stochastic gradient descent for generative adversarial networks was proposed in [13], where the algorithm assumes that the maximization subproblem can be solved up to some small error. None of the above works are of the alternating type we consider in this work, and none of them are for deterministic problems.

### 1.3. Contribution of this work

In this work, we consider the min-max problem from an alternating minimization/maximization perspective. The studied problems are general, allowing non-convexity and non-smoothness in the objective, non-convex coupling between variables as well. The proposed algorithm solves the maximization and minimization problems in an alternating way. The main contributions of this paper are listed as follows:

[1] A class of machine learning and resource allocations problems is formulated in the framework of non-convex min-max problems.

[2] Two types of min-max problems are studied, and two simple algorithms which alternate between the minimization and maximization steps, are proposed with provable convergence guarantees.

[3] To the best of our knowledge, it is the first time that the convergence rate of the alternating min-max algorithm is quantified for the (one-sided) non-convex min-max problem (1).

### 2. THE PROPOSED ALGORITHMS

In this section, we present our main convergence results for proposed algorithms. Due to space limits, we will omit the proof details, but instead presenting a few key lemmas leading to the main results.

#### 2.1. Maximization problem is strongly concave

First, we consider the following min-max problem,

$$\min_{x_i \in \mathcal{X}_i, \forall i, y \in \mathcal{Y}} \max_{x_1 \in \mathcal{X}_1, \ldots, x_K \in \mathcal{X}_K, y} f(x_1, \ldots, x_K, y)$$

where $x_i \in \mathcal{X}_i$ with $\mathcal{X}_i, \mathcal{Y}$ convex compact feasible sets and $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_K$. Moreover, $x$ and $y$ are coupled in a general way by function $f(x, y)$, the objective function $f(x, y)$ is non-convex with respect to $x$ and it has Lipschitz continuous gradient w.r.t. to $x$, with constants $L_x, \forall i$. Finally, $f(x, y)$ is strongly concave with respect to $y$ with modulus $\gamma > 0$, that is

$$f(x, y) - f(x, z) \leq \langle \nabla f(x, z), y - z \rangle - \frac{\gamma}{2} \|y - z\|^2, \quad \forall y \in \mathcal{Y}.$$

For the above problem, we propose the following algorithm:

$$x_i^{r+1} = \arg \min_{x_i \in \mathcal{X}_i} \langle \nabla_i f(w_i^r, y^r), x_i - x_i^r \rangle + \frac{\beta}{2} \|x_i - x_i^r\|^2, \quad i = 1, \ldots, K,$$

$$y^{r+1} = \arg \max_{y \in \mathcal{Y}} \langle \nabla y f(x^{r+1}, y), y - y^r \rangle - \frac{1}{2\rho} \|y - y^r\|^2$$

where $r$ is the iterate index, $w_i^r := [x_1^{r+1}; \ldots; x_i^{r+1}; x_{i+1}^{r+1}; \ldots]$, $\beta$ and $\rho$ are the regularizers and will be discussed further in Lemma 2.

#### 2.2. Maximization problem is linear

Second, we consider the following min-max problem,

$$\max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \langle f(x), y \rangle$$

where $\mathcal{X}, \mathcal{Y}$ are still convex compact sets, the objective function $f(x)$ is non-convex, and it is Lipschitz and has Lipschitz gradient, with constants $L_x$ and $L_y$. Because the maximization problem over variable $y$ is linear rather than strongly concave, we propose a novel algorithm which adds some (diminishing) perturbation term in the maximization problem and uses the adaptive step-sizes (i.e., $\beta^r$ and $\gamma^r$ shown in the following section):

$$x_i^{r+1} = \arg \min_{x_i \in \mathcal{X}_i} \langle \nabla_i f(x^r, y^r), x - x^r \rangle + \frac{\beta^r}{2} \|x - x^r\|^2, \quad (10a)$$

$$y^{r+1} = \arg \max_{y \in \mathcal{Y}} \langle f(x^{r+1}), y - y^r \rangle - \frac{1}{2\rho} \|y - y^r\|^2 - \frac{\gamma^r}{2} \|y^r\|^2$$

where $\beta^r$ is the regularizer of the minimization problem and $\rho$ and $\gamma^r$ are the regularizers of maximization problems but $\gamma^r$ is a decreasing sequence.

### 3. THEORETICAL PROPERTIES AND DISCUSSIONS

In this section, we present our main convergence results for proposed algorithms. Due to space limits, we will omit the proof details, but instead presenting a few key lemmas leading to the main results.
Lemma 1 \textit{(Descent lemma)} Assume that \( f(x, y) \) has Lipschitz continuous gradient w.r.t \( y \) with constant \( L_y \) and is strongly concave with respect to \( y \) with modulus \( \gamma > 0 \). Let \((x^r, v^r)\) be a sequence generated by (8). Then we have the following descent estimate
\[
\begin{align*}
   f(x^{r+1}, y^{r+1}) - f(x^r, y^r) &\leq -\gamma\|y^{r+1} - y^r\|^2 \\
   - (\gamma - \frac{\rho}{2})\|y^{r+1} - y^r\|^2 &\leq \frac{\rho L_y^2}{2}\|x^{r+1} - x^r\|^2.
\end{align*}
\]
From Lemma 1, it is not clear whether the objective function is decreased or not, since the minimization step will consistently decrease the objective value while the maximization step will increase the objective value. The key in our analysis is to identify a proper potential function, which can capture the essential dynamics of the algorithm.

Lemma 2 Let \((x^r, y^r)\) be a sequence generated by (8). When the following conditions are satisfied,
\[
   \rho < \frac{\gamma}{2L_y}, \quad \beta \geq \max\left\{\frac{4}{\gamma^2}, \frac{L_y}{\gamma^2} + \rho, L_{x, \text{max}}\right\},
\]
then there exist \( c_1, c_2 > 0 \) such that the potential function will monotonically decrease, i.e.,
\[
   \mathcal{P}^{r+1} - \mathcal{P}^r \leq -c_1\|y^{r+1} - y^r\|^2 - c_2\|x^{r+1} - x^r\|^2,
\]
where \( \mathcal{P}^r := f(x^{r+1}, y^{r+1}) + \left(\frac{\sqrt{2}}{v^r} - 4\right)\|y^{r+1} - y^r\|^2. \)

3.1. Convergence rate of algorithm (8)

To state our main result, let us define the proximal gradient of the objective function as
\[
   \nabla \mathcal{L}(x, y) \triangleq \left[\begin{array}{c}
x - \nabla_x f(x, y) \\
y - \nabla_y f(x, y)
\end{array}\right]
\]
where \( \nabla \mathcal{L}(x, y) = 0 \), then we have the following convergence rate of the proposed algorithm.

Theorem 1 \textit{Suppose that the sequence \((x^r, y^r)\) is generated by (8) and \( \rho, \beta \) satisfy the conditions (11). For a given small constant \( \epsilon \), let \( T(\epsilon) \) denote the first iteration index, such that the following inequality is satisfied:}
\[
   T(\epsilon) \triangleq \min\{t \mid \|\nabla \mathcal{L}(x^t, y^t)\|^2 \leq \epsilon, \ t \geq 1\}.
\]
\textit{Then there exists some constant \( C > 0 \) such that \( \epsilon \leq C/(2 - P/T(\epsilon)) \) where \( P \) denotes the lower bound of \( P \).}

3.2. Convergence rate of perturbed algorithm (10)

We have the following convergence analysis for the algorithm (10). Let \((x^r, y^r)\) be a sequence generated by (10). We have
\[
   \lim_{r \to \infty} \frac{1}{\gamma^{r+1}} \leq \frac{1}{\gamma} \leq \frac{1}{\gamma} \sum_{r=1}^{\infty} \frac{1}{\alpha^r} = \infty, \quad \alpha^r - \alpha^{r+1} \leq 0, \quad \lim_{r \to \infty} \frac{1}{\alpha^r} \to 0,
\]
then there exists a sequence \( \{c^r\} \) bounded away from zero, such that the potential function will monotonically decrease, i.e.,
\[
   \mathcal{P}^{r+1} - \mathcal{P}^r \leq -\frac{1}{10\rho}\|y^{r+1} - y^r\|^2 - c^r\|x^{r+1} - x^r\|^2.
\]

4. NUMERICAL RESULTS

We test our algorithms on two applications: the robust learning problem and the rate maximization problem in the presence of a jammer. Consider a learning scenario where we have datasets from two different domains and adopt a regularized logistic regression model in order to solve a binary classification problem. We aim to learn the model parameters using the following two approaches:

[1] \textit{Robust Learning :} Apply the robust learning model (4) and optimize the cost function using the algorithm (8).
[2] \textit{Multitask Learning :} Apply a multitask learning model, where the weights associated with each loss function/task are fixed to 1/2. The problem is optimized using gradient descent.

We evaluate the accuracy of the above algorithms as the worst hit rate across the two domains, i.e.,
\[
   \text{accuracy} = \min\{\text{hit rate on domain 1, hit rate on domain 2}\}.
\]

We run two sets of experiments. In the first we generate randomly two datasets using the Gaussian and the Laplacian distribution respectively. The first dataset contains 200 training and 40 testing data points, while in the second one the number is 1000 and 200 respectively. At each dataset the data points are generated in a way
that the two classes are approximately linearly separable. In Fig. 1 the accuracy of the two algorithms is illustrated.

In the second set of experiments we use two different parts of the MNIST dataset [14] as the two different domains. For the 1st domain we use 200 images for training and 50 for testing, and in the second 800 and 200 images respectively. Also, we contaminate the second dataset with Gaussian noise, in order to differentiate it from the first one. We train the two aforementioned models on those datasets to classify correctly the digit ‘1’ from the rest of them. The results are presented in Fig. 3. It is apparent that the performance of the robust model is comparable to that of the multitask learning model, which shows that the proposed algorithm is capable of attaining good solutions of min-max optimization problems.

**Power control in the presence of a jammer.** We consider the multi-channel and multi-user formulation (6) where there are \( N \) parallel channels available, and there are \( K \) normal users and one jammer in the system. For this problem it is easy to verify that the jammer problem has a strongly concave objective function, therefore we can directly apply Algorithm (8).

In our test we will compare the proposed method with the well-known interference pricing method [15, 16], and the WMMSE algorithm [17], both of which are designed to solve the \( K \)-user \( N \)-channel sum-rate optimization problem without the jammer. Our problem is tested using the following setting. Consider a network with \( K = 10 \), and the interference channel among the users and the jammer is generated using uncorrelated fading channel model with channel coefficients generated from the complex \( \mathcal{C}N(0,1) \) [17]. The users’ power budget is assumed to be \( P \) for all transmitters, where \( P = 10^{3} \text{SNR}/10 \). For test cases without a jammer, we set \( \sigma_{k}^{2} = 1 \) for all \( k \). For test cases with a jammer, we set \( \sigma_{k}^{2} = 1/2 \) for all \( k \), and let the jammer has the rest of the noise power, i.e., \( p_{0,\text{max}} = N/2 \). Note that the consideration behind splitting the noise power is mostly for fairness comparison between the cases with and without the jammer. However, we do note that it is not possible to be completely fair because even though the total noise budgets are the same, the noise power transmitted by the jammer also has to go through the channel, therefore the total received noise power could still be different.

From the Fig. 4 (top), we see that the pricing algorithm monotonically increases the sum rate (as is predicted by theory), while our proposed algorithm behaves very differently. Further in Fig. 4 (bottom), we do see that by using the proposed algorithm, the jammer is able to effectively reduce the total sum rate of the system.
5. REFERENCES


