

# Maximizing Signal to Interference Noise Ratio for Massive MIMO : A Stochastic Neurodynamic Approach

Siham Tassouli, Abdel Lisser

▶ **To cite this version:**

Siham Tassouli, Abdel Lisser. Maximizing Signal to Interference Noise Ratio for Massive MIMO : A Stochastic Neurodynamic Approach. 2022. hal-03838584

**HAL Id: hal-03838584**
**https://hal.science/hal-03838584**

Preprint submitted on 3 Nov 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Maximizing Signal to Interference Noise Ratio for Massive MIMO : A Stochastic Neurodynamic Approach

Siham Tassouli*[a], Abdel Lisser[a]

[a]*Laboratoire des Signaux et Systèmes (L2S), CentraleSupélec, Université Paris Saclay, 3, rue Joliot Curie, 91192 Gif sur Yvette cedex, France*

**Abstract**

In this paper, we consider the problem of maximizing the worst user signal to interference noise ratio (SINR) for massive multiple input multiple output (MaMIMO). We reformulate the nonlinear optimization model as a joint chance-constrained geometric program. We propose a neurodynamic approach to solve the obtained problem. Our numerical results indicate that our approach outperforms the state-of-art convex approximations used to solve joint chance-constrained geometric problems.

*Keywords:* Dynamical neural network, Geometric programming, Wireless networks, Joint chance constraints, Lyapunov Theory, Ordinary differential equations.

*2010 MSC:* 00-01, 99-00

## 1. Introduction

The Massive Multiple Input Multiple Output (MaMIMO) is an emerging technology for new communication systems and the Internet of Things (IoT). It is based on the use of hundreds of antennas interfering with each other. It is one of the candidate techniques for 5G and also a candidate to succeed 4G LTE and LTE-A. The introduction of MaMIMO insured higher connectivity, the ability to adapt to high density environments, reduced transmission latency for augmented reality, energy efficiency meeting green communications guidelines and a better quality of signal paths and security.

In recent years, MaMIMO resource allocation has been studied in several works. Xuanhong et al. [1] investigate a joint resource allocation algorithm to improve spectrum efficiency and throughput. Mosleh et al. [2] study a resource allocation problem for downlink cell-free massive MIMO networks. Yin et al. [3] deal with the Mobility Problem of Massive MIMO using Extended Prony's Method. Dikmen & Kulac [4] examine power allocation algorithms for MIMO systems. Salah et al. [5] propose an adaptive optimization technique focusing on maximizing Energy Efficiency in adaptive massive MIMO networks.

In this paper, we propose a neurodynamic approach to solve a joint chance constrained nonlinear optimization model where the aim is to maximize the worst user SINR. Adasme et al. [6] propose a local search algorithm that allows obtaining feasible solutions for the problem of maximizing the worst

*Corresponding author.

*Email addresses:* siham.tassouli@centralesupelec.fr (Siham Tassouli*), abdel.lisser@centralesupelec.fr (Abdel Lisser)

user SINR for Massive MIMO. Mei & Zhang [7] derive a tractable lower bound of the average signal-to-interference-plus-noise ratio (SINR) at the receiver of each user, based on which two average-signal-to-average-interference-plus-noise ratio (ASAINR) balancing problems are formulated to maximize the minimum ASAINR among all users.

We reformulate our problem as a geometric optimization problem. Geometric programming was introduced by Duffin et al. [8] in 1967. Since then, geometric programming have been used to model and solve several optimization problems in several areas, i.e, aircraft design [9], communication systems [10], digital circuit optimization [11], information theory [12] ... . To model the uncertainty within the optimization problem, we use a joint chance constrained approach. Joint chance-constrained programming has been widely studied and applied to model real world problems, i.e, lot-sizing problems with stochastic demand [13], call center workforce scheduling under uncertain call arrival forecasts [14], optimal power flow [15] ... . To solve optimization problems with joint chance constraints many works were conducted to give deterministic equivalents and study the optimality conditions. Cheng & Lisser propose a second-order cone programming approach for linear programs with joint probabilistic constraints. You et al. [16] use data-driven models to solve programs with joint chance constraints. Ono et al. [17] present a novel dynamic programming algorithm to approximate joint chance constraints.

The rest of the paper is organized as follows. In Section 2, we first give a brief description of the MaMIMO resource allocation problem we are studying. Then, we present a joint probabilistic geometric formulation of the problem of maximizing the worst user Signal to Interference Noise Ratio and we give the optimality conditions of the obtained problem. Based on the partial KKT system obtained in Section 2, we propose in Section 3 a neurodynamic approach to solve the initial problem. In Section 4, we conduct some numerical results in order to evaluate the performances of our approach.

## 2. Problem formulation

We consider a single cell area, see Figure 1, which is composed of a set of $\mathcal{U} = \{1, ..., K\}$ users. We assume that each user uses only one antenna to receive the data from the base station. The base station is equipped with $T$ antennas. We aim to maximize the worst user SINR subject to some limits on the power assigned to each user. The $\text{SINR}_i$ for user $i$ can be expressed as follows [18]

$$\text{SINR}_i = \frac{p_i |g_i^H g_i|^2}{\sum\limits_{j \in \mathcal{U}, j \neq i} p_j |g_i^H g_j|^2 + |\sigma_i|^2} \tag{1}$$

We formulate our optimization problem as follows

$$\max_{p \in \mathbb{R}^K_{++}} \min_{i \in \mathcal{U}} \frac{p_i |g_i^H g_i|^2}{\sum\limits_{j \in \mathcal{U}, j \neq i} p_j |g_i^H g_j|^2 + |\sigma_i|^2}, \tag{2}$$

$$\text{s.t} \qquad P_{min} \leq p_i \leq P_{max}, \forall i \in \mathcal{U}, \tag{3}$$

where $p_i$ is the power to be assigned for each user $i \in \mathcal{U}$. $g_i \in \mathbb{C}^{T \times 1}$, $g_i^H \in \mathbb{C}^{1 \times T}$ and $\sigma_i^2$ are the beam domain channel vector associated to user $i \in \mathcal{U}$, its Hermitian transpose and Additive White

Gaussian Noise (AWGN), respectively. We finally assume that the AWGN behaves according to an independent complex Gaussian distribution with zero mean and unit variance ($\sigma_i \sim \mathcal{CN}(0,1)$) while each entry in vectors $g_i$ and $g_i^H$ is a complex number that is assumed to behave as a quasi-static independent and identically distributed Rayleigh fading channel. $P_{min}$ and $P_{max}$ define the lower and the upper bounds for each power variable, respectively.

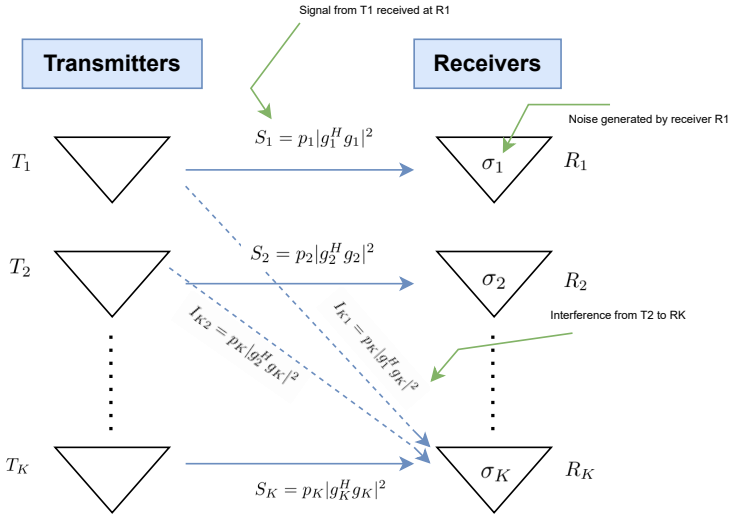



Figure 1: Signal to Interference plus Noise Ratio, illustration

Let $a_{ij} = |g_i^H g_j|^2 |g_i^H g_i|^{-2}$ and $b_i = |\sigma_i|^2 |g_i^H g_i|^{-2}$ and by introducing an additional variable $w$ we rewrite (2)-(3) as

$$\max_{p \in \mathbb{R}_{++}^K, w \in \mathbb{R}_{++}} \quad w, \tag{4}$$

$$\text{s.t} \quad \sum_{j \in \mathcal{U}, j \neq i} a_{ij} p_j p_i^{-1} w + b_i p_i^{-1} w \leq 1, \forall i \in \mathcal{U}, \tag{5}$$

$$P_{min} \leq p_i \leq P_{max}, \forall i \in \mathcal{U}. \tag{6}$$

An equivalent minimization problem is given by

$$\min_{p \in \mathbb{R}_{++}^K, w \in \mathbb{R}_{++}} \quad w^{-1}, \tag{7}$$

$$\text{s.t} \quad \sum_{j \in \mathcal{U}, j \neq i} a_{ij} p_j p_i^{-1} w + b_i p_i^{-1} w \leq 1, \forall i \in \mathcal{U}, \tag{8}$$

$$P_{min} \leq p_i \leq P_{max}, \forall i \in \mathcal{U}. \tag{9}$$

We consider the case where the coefficients $a_{ij}$ and $b_i$ are not completely known and normally distributed and pairwise independent, i.e., $a_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma_{ij}^2)$ and $b_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. We then replace the deterministic constraint (8) with the following joint constraint

$$\mathbb{P}\left\{ \sum_{j \in \mathcal{U}, j \neq i} a_{ij} p_j p_i^{-1} w + b_i p_i^{-1} w \leq 1, \forall i \in \mathcal{U} \right\} \geq 1 - \epsilon. \tag{10}$$

with $1-\epsilon$ is a given confidence level. We use joint chance constraint instead of using individual constraints because the joint chance constraint ensures that the constraint as a whole is satisfied to a certain confidence level. The individual chance constraints even if they are easier to solve, they only guarantee that each constraint is satisfied to a certain confidence level.

Using the pairwise independence between the coefficients and by introducing auxiliary variables $y_i \in \mathbb{R}^+$, $\forall i \in \mathcal{U}$, we give the following deterministic equivalent for the joint constraint (10)

$$\sum_{j\in\mathcal{U},j\neq i} \mu_{ij}p_j p_i^{-1}w + \mu_i p_i^{-1}w + \phi^{-1}(y_i)\left\{\sqrt{\sum_{j\in\mathcal{U},j\neq i}\sigma_{ij}^2 p_j^2 p_i^{-2}w^2 + \sigma_i^2 p_i^{-2}w^2}\right\} \leq 1, \forall i \in \mathcal{U}, \tag{11}$$

$$\prod_{i\in\mathcal{U}} y_i \geq 1-\epsilon, \tag{12}$$

$$0 \leq y_i \leq 1, \forall i \in \mathcal{U}, \tag{13}$$

We write then (7)-(9) equivalently as

$$\begin{aligned}
\min_{p\in\mathbb{R}^K_{++}, w\in\mathbb{R}_{++}} \quad & w^{-1}, \\
\text{s.t} \quad & \sum_{j\in\mathcal{U},j\neq i} \mu_{ij}p_j p_i^{-1}w + \mu_i p_i^{-1}w + \\
& \phi^{-1}(y_i)\left\{\sqrt{\sum_{j\in\mathcal{U},j\neq i}\sigma_{ij}^2 p_j^2 p_i^{-2}w^2 + \sigma_i^2 p_i^{-2}w^2}\right\} \leq 1, \forall i \in \mathcal{U}, \\
& 1-\epsilon-\prod_{i\in\mathcal{U}} y_i \leq 0, \\
& -y_i \leq 0, y_i - 1 \leq 0, \forall i \in \mathcal{U}, \\
& P_{min} - p_i \leq 0, p_i - P_{max} \leq 0, \forall i \in \mathcal{U}.
\end{aligned} \tag{SP}$$

The obtained equivalent deterministic problem (SP) is nonconvex, we apply then the logarithmic transformation $r_i = \log(p_i)$, $x_i = \log(y_i)$, $\forall i \in \mathcal{U}$ and $t = log(w)$ and obtain the following problem

$$\min \exp(-t), \tag{14}$$

$$\begin{aligned}
\text{s.t} \quad & \sum_{j\in\mathcal{U},j\neq i} \mu_{ij}\exp(r_j - r_i + t) + \mu_i\exp(t - r_i) \\
& +\phi^{-1}(e^{x_i})\left\{\sqrt{\sum_{j\in\mathcal{U},j\neq i}\sigma_{ij}^2\exp(2r_j - 2r_i + 2t) + \sigma_i^2\exp(2t - 2r_i)}\right\} \leq 1, \forall i \in \mathcal{U},
\end{aligned} \tag{15}$$

$$\log(1-\epsilon) - \sum_{i\in\mathcal{U}} x_i \leq 0,\ x_i \leq 0, i \in \mathcal{U}, \tag{16}$$

$$\log(P_{min}) - r_i \leq 0, r_i - \log(P_{max}) \leq 0, \forall i \in \mathcal{U}. \tag{17}$$

Let $z = (r,t)^T$, for the sake of simplicity we write the optimization problem as

$$\min f(z), \tag{18}$$

$$\text{s.t}\ \ g_i(z,x) \leq 0, \forall i \in \mathcal{U}, \tag{19}$$

$$l(x) \leq 0,\ h_i(x) \leq 0, i \in \mathcal{U}, \tag{20}$$

$$v_i(z) \leq 0, w_i(z) \leq 0, \forall i \in \mathcal{U}. \tag{21}$$

where $f(z) = \exp(-t)$,$l(x) = \log(1-\epsilon) - \sum_{i\in\mathcal{U}} x_i$, $h_i(x) = x_i$, $v_i(z) = \log(P_{min}) - r_i$, $w_i(z) = r_i - \log(P_{max})$ and

$$\begin{aligned} g_i(z,x) &= \sum_{j\in\mathcal{U},j\neq i} \mu_{ij}\exp(r_j - r_i + t) + \mu_i\exp(t - r_i) \\ &\quad + \phi^{-1}(e^{x_i})\left\{\sqrt{\sum_{j\in\mathcal{U},j\neq i} \sigma_{ij}^2\exp(2r_j - 2r_i + 2t) + \sigma_i^2\exp(2t - 2r_i)}\right\} - 1 \end{aligned}$$

**Lemma 1.** Problem (18)-(21) is biconvex on $(z, x)$.

*Proof.* The convexity on $z$ is straightforward. We have $x \mapsto e^x$ is convex and $\phi^{-1}(.)$ is non decreasing, then $x \mapsto \phi^{-1}(e^x)$ is convex. (18)-(21) is then convex on $x$. The conclusion follows. □

**Definition 1.** Let $z^* \in \mathbb{R}^{K+1}$, $x^* \in \mathbb{R}^K$, $\alpha_i^{(1)}$, $\alpha_i^{(2)}$, $\beta$, $\gamma_i$, $\lambda_i$ and $\zeta_i$, $i \in \mathbb{U}$ such that

$$\nabla_z f(z^*) + \sum_{i\in\mathcal{U}} \alpha_i^{(1)}\nabla_z g_i(z^*, x^*) + \sum_{i\in\mathcal{U}} \gamma_i\nabla_z v_i(z^*) + \sum_{i\in\mathcal{U}} \lambda_i\nabla_z w_i(z^*) = 0, \tag{22}$$

$$\beta\nabla_x l(x^*) + \sum_{i\in\mathcal{U}} \alpha_i^{(2)}\nabla_x g_i(z^*, x^*) + \sum_{i\in\mathcal{U}} \zeta_i\nabla_x h_i(x^*) = 0, \tag{23}$$

$$\alpha_i^{(1)} g_i(z^*, x^*) = 0, \gamma_i v_i(z^*) = 0, \lambda_i w_i(z^*) = 0, \alpha_i^{(1)} \geq 0, \gamma_i \geq 0, \lambda_i \geq 0, i \in \mathcal{U}, \tag{24}$$

$$\beta l(x^*) = 0, \alpha_i^{(2)} g_i(z^*, x^*) = 0, \zeta_i h_i(x^*) = 0, \beta \geq 0, \alpha_i^{(2)} \geq 0, \zeta_i \geq 0, i \in \mathcal{U}, \tag{25}$$

then $(z^*, x^*)$ is a partial KKT point of (SP).

The optimality conditions of problem (SP) are given in the following theorem

**Theorem 2.** Let $\alpha^{(1)} = (\alpha_1^{(1)}, ..., \alpha_N^{(1)})^T$, $\alpha^{(2)} = (\alpha_1^{(2)}, ..., \alpha_N^{(2)})^T$, $\gamma = (\gamma_1, ..., \gamma_N)^T$, $\lambda = (\lambda_1, ..., \lambda_N)^T$, $\zeta = (\zeta_1, ..., \zeta_N)^T$, $g = (g_1, ..., g_N)^T$, $v = (v_1, ..., v_N)^T$, $w = (w_1, ..., w_N)^T$ and $h = (h_1, ..., h_N)^T$, we write (22)-(25) equivalently as

$$\nabla_z f(z^*) + \nabla_z g_(z^*, x^*)^T\alpha^{(1)} + \nabla_z v(z^*)^T\gamma + \nabla_z w(z^*)^T\lambda = 0, \tag{26}$$

$$\beta\nabla_x l(x^*) + \nabla_x g(z^*, x^*)^T\alpha^{(2)} + \nabla_x h(x^*)^T\zeta = 0, \tag{27}$$

$$g_i(z^*, x^*)^T\alpha^{(1)} = 0, v(z^*)^T\gamma = 0, w(z^*)^T\lambda = 0, \alpha^{(1)} \geq 0, \gamma \geq 0, \lambda \geq 0, \tag{28}$$

$$\beta l(x^*) = 0, g(z^*, x^*)^T\alpha^{(2)} = 0, h(x^*)^T\zeta = 0, \beta \geq 0, \alpha^{(2)} \geq 0, \zeta \geq 0, \tag{29}$$

Let $z^* \in \mathbb{R}^{K+1}$, $x^* \in \mathbb{R}^K$, $(z^*, x^*)$ is a partial optimum of (SP) if and only if $(z^*, x^*)$ is a partial KKT point of (SP). Moreover, if $\alpha_i^{(1)} = \alpha_i^{(2)}$ then $(z^*, x^*)$ is a KKT point of (SP).

**Remark 3.** The main lines of the proof of Theorem 2 are given in [19].

## 3. Neurodynamic approach

In this Section, we aim to construct a continuous-time dynamical system that converges to a KKT point of (SP). Therefore, we propose a dynamical neural network described by the following dynamical system

$$\frac{dz}{dt} = -(\nabla_z f(z) + \nabla_z g(z,x)^T(\alpha + g(z,x))_+ + \nabla_z v(z)^T(\gamma + v(z))_+ + \nabla_z w(z)^T(\lambda + w(z))_+), \tag{30}$$

$$\frac{dx}{dt} = -(\nabla_x l(x)^T(\beta + l(x))_+ + \nabla_x g(z,x)^T(\alpha + g(z,x))_+ + \nabla_x h(x)^T(\zeta + h(x))_+), \tag{31}$$

$$\frac{d\alpha}{dt} = (\alpha + g(z,x))_+ - \alpha, \tag{32}$$

$$\frac{d\gamma}{dt} = (\gamma + v(z))_+ - \gamma, \tag{33}$$

$$\frac{d\lambda}{dt} = (\lambda + w(z))_+ - \lambda, \tag{34}$$

$$\frac{d\beta}{dt} = (\beta + l(x))_+ - \beta, \tag{35}$$

$$\frac{d\zeta}{dt} = (\zeta + h(x))_+ - \zeta. \tag{36}$$

For convenience, let $y = (z, x, \alpha, \gamma, \lambda, \beta, \zeta)$ we write the dynamical system (30)-(36) shortly as

$$\frac{dy}{dt} = \eta(y) \tag{37}$$

$$y(t_0) = y_0, \tag{38}$$

where $y_0$ is a given initial point. A generalized circuit implementation of neural network (30)-(36) is given in Figure 2

Now we study the stability and convergence properties for (30)-(36).

**Theorem 4.** Let $y = (z, x, \alpha, \gamma, \lambda, \beta, \zeta)$ an equilibrium point of (30)-(36), then $(z, x)$ is a KKT point of (SP). Furthermore, if $(z, x)$ is a KKT point of (SP) then there exists $(\alpha, \gamma, \lambda, \beta, \zeta)$ such that $(z, x, \alpha, \gamma, \lambda, \beta, \zeta)$ is an equilibrium point of (30)-(36).

*Proof.* Let $y = (z, x, \alpha, \gamma, \lambda, \beta, \zeta)$ an equilibrium point of (30)-(36), then $\frac{dz}{dt} = 0$, $\frac{dx}{dt} = 0$, $\frac{d\alpha}{dt} = 0$, $\frac{d\gamma}{dt} = 0$, $\frac{d\lambda}{dt} = 0$ $\frac{d\beta}{dt} = 0$ and $\frac{d\zeta}{dt} = 0$.

We have that $\frac{d\alpha}{dt} = 0 \iff (\alpha + g(z,x))_+ - \alpha \iff \{\alpha \geq 0\,,\ g(z,x) \leq 0 \text{ and } \alpha^T g(z,x) = 0\}$,
Similarly, we have $\frac{d\gamma}{dt} = 0 \iff \{\gamma \geq 0\,,\ v(z) \leq 0 \text{ and } \gamma^T v(z) = 0\}$ and $\frac{d\lambda}{dt} = 0 \iff \{\lambda \geq 0\,,$ $w(z) \leq 0$ and $\lambda^T w(z) = 0\}$.
Furthermore, we have $\frac{dz}{dt} = 0 \iff -(\nabla_z f(z) + \nabla_z g(z,x)^T(\alpha + g(z,x))_+ + \nabla_z v(z)^T(\gamma + v(z))_+ +$ $\nabla_z w(z)^T(\lambda + w(z))_+) = 0 \iff \nabla_z f(z^*) + \nabla_z g(z^*, x^*)^T\alpha + \nabla_z v(z^*)^T\gamma + \nabla_z w(z^*)^T\lambda = 0$. We obtain then, equations (26) and (28) of the partial KKT system (26)-(29). Following the same steps we obtain equations (27) and (29).
The controverse part of the theorem is straightforward. □

**Theorem 5.** For any initial point $(z(t_0), x(t_0), \alpha(t_0), \gamma(t_0), \lambda(t_0), \beta(t_0), \zeta(t_0))$, there exists an unique continuous solution $(z(t), x(t), \alpha(t), \gamma(t), \lambda(t), \beta(t), \zeta(t))$for (30)-(36).

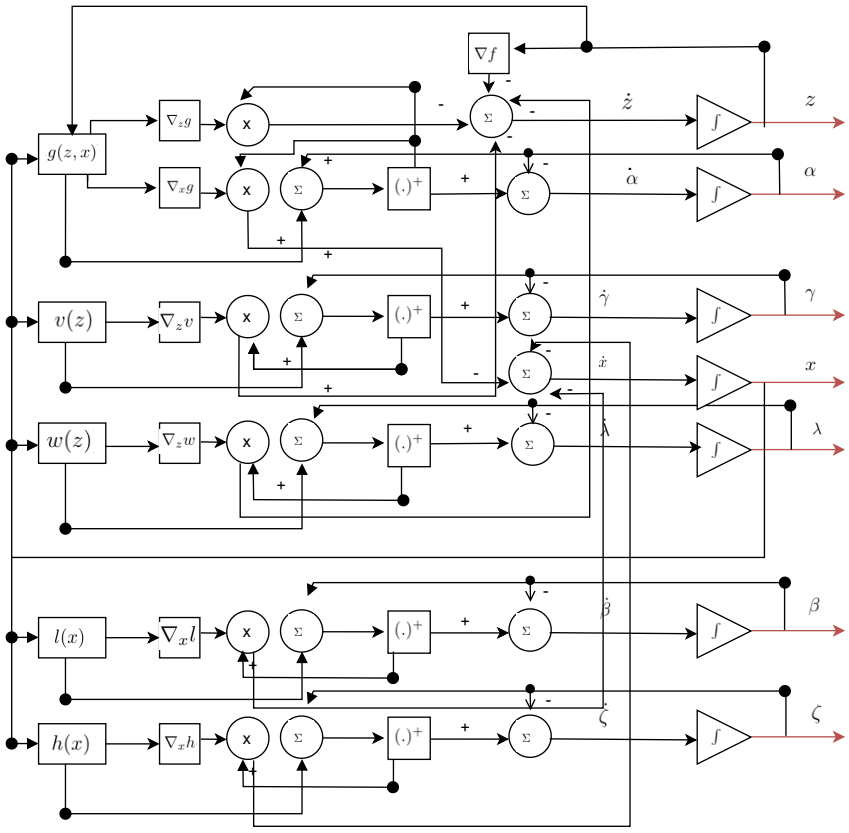



Figure 2: A block diagram for the neural network (30)-(36)

*Proof.* Since $\nabla_z f(z)$, $\nabla_z g(z,x)$, $\nabla_x g(z,x)$, $\nabla_z v(z)$, $\nabla_z w(z)$, $\nabla_x l(x)$ and $\nabla_x h(x)$ are continuously differentiable on open sets, then all the second terms of the differential equations (30)-(36) are locally Lipschitz continuous. According to the local existence of ordinary differential equations also known as *Picard–Lindelöf Theorem* [20], the neural network (30)-(36) has a unique continuous solution $(z(t), x(t), \alpha(t), \gamma(t), \lambda(t), \beta(t), \zeta(t))$. □

To prove the stability and convergence of the dynamical neural network (30)-(36), we first show the negative semidefiniteness of the Jacobian matrix $\nabla\eta(y)$ that we are going to use while defining the Lyapunov functions.

**Theorem 6.** The Jacobian matrix $\nabla\eta(y)$ is negative semidefinite.

*Proof.* Without loss of generality, we assume that $\beta + l(x) \geq 0$ and that there exists $0 \leq p, q, r, s \leq K$ such that

$$(\alpha + g)_+ = (\alpha_1 + g_1(z,x), \alpha_2 + g_2(z,x), ....., \alpha_p + g_p(z,x), \underbrace{0, ...., 0}_{K-p}),$$

$$(\gamma + v)_+ = (\gamma_1 + v_1(z), \gamma_2 + v_2(z), ....., \gamma_q + v_q(z), \underbrace{0, ...., 0}_{K-q}),$$

$$(\lambda + w)_+ = (\lambda_1 + w_1(z), \lambda_2 + w_2(z), ....., \lambda_r + w_r(z), \underbrace{0, ...., 0}_{K-r}),$$

$$(\zeta + h)_+ = (\zeta_1 + h_1(x), \zeta_2 + h_2(x), ....., \zeta_s + h_s(x), \underbrace{0, ...., 0}_{K-s}).$$

We represent the Jacobian matrix of $\eta$ in the following form $\nabla\eta(y) = \begin{bmatrix} A_1 & A_2 & A_3 & A_4 & A_5 & 0 & 0 \\ B_1 & B_2 & B_3 & 0 & 0 & B_6 & B_7 \\ C_1 & C_2 & C_3 & 0 & 0 & 0 & 0 \\ D_1 & 0 & 0 & D_4 & 0 & 0 & 0 \\ E_1 & 0 & 0 & 0 & E_5 & 0 & 0 \\ 0 & F_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & G_2 & 0 & 0 & 0 & 0 & G_7 \end{bmatrix}$,

where

$$A_1 = -\Big(\nabla_z^2 f(z) + \sum_{i=1}^{p}((\alpha_i + g_i)\nabla_z^2 g_i^p(z,x)) + \nabla_z g^p(z,x)^T \nabla_z g^p(z,x) + \sum_{i=1}^{q}((\gamma_i + v_i)\nabla_z^2 v_i^q(z)) + \nabla_z v^q(z)^T \nabla_z v^q(z) + \sum_{i=1}^{r}((\lambda_i + w_i)\nabla_z^2 w_i^r(z)) + \nabla_z w^r(z)^T \nabla_z w^r(z)\Big),$$

$$A_2 = -\Big(\sum_{i=1}^{p}((\alpha_i + g_i)\nabla_x \nabla_z g_i^p(z,x)) + \nabla_x g^p(z,x)^T \nabla_z g^p(z,x)\Big),$$

$$A_3 = -\nabla_z g^p(z,x)^T, \qquad A_4 = -\nabla_z v^q(z)^T, \qquad A_5 = -\nabla_z w^r(z)^T,$$

$$B_1 = -\Big(\sum_{i=1}^{p}(\alpha_i + g_i)\nabla_z \nabla_x g_i^p(z,x) + \nabla_z g^p(z,x)^T \nabla_x g^p(z,x)\Big),$$

$$B_2 = -\Big(\sum_{i=1}^{p}((\alpha_i + g_i)\nabla_x^2 g_i^p(z,x)) + \nabla_x g^p(z,x)^T \nabla_x g^p(z,x) + \nabla_x^2 l(x) + \nabla_x l(x)^T \nabla_x l(x) + \sum_{i=1}^{s}((\zeta_i + h_i)\nabla_x^2 \zeta_i^s(x)) + \nabla_x h^s(x)^T \nabla_x h^s(x)\Big),$$

$$B_6 = -\nabla_x l(x)^T, \qquad B_7 = -\nabla_x h^s(x)^T, \qquad C_1 = \nabla_z g^p(z,x),$$

$$C_2 = \nabla_x g^p(z,x), \qquad C_3 = S_p = -\begin{bmatrix} O_{p\times p} & O_{p\times(K-p)} \\ O_{(K-p)\times p} & I_{(K-p)\times(K-p)} \end{bmatrix}, \qquad D_1 = \nabla_z v^q(z),$$

$$D_4 = S_q = -\begin{bmatrix} O_{q\times q} & O_{q\times(K-q)} \\ O_{(K-q)\times q} & I_{(K-q)\times(K-q)} \end{bmatrix}, \quad E_1 = \nabla_z w^r(z), \quad E_5 = S_r = -\begin{bmatrix} O_{r\times r} & O_{r\times(K-r)} \\ O_{(K-r)\times r} & I_{(K-r)\times(K-r)} \end{bmatrix},$$

$$F_2 = \nabla_x l(x), \qquad G_2 = \nabla_x h^s(x), \qquad G_7 = S_s = -\begin{bmatrix} O_{s\times s} & O_{s\times(K-s)} \\ O_{(K-s)\times s} & I_{(K-s)\times(K-s)} \end{bmatrix}.$$

We rewrite then the Jacobian matrix $\nabla\eta$ as

$$\nabla\eta(y) = \begin{bmatrix} A_1 & A_2 & A_3 & A_4 & A_5 & 0 & 0 \\ A_2^T & B_2 & B_3 & 0 & 0 & B_6 & B_7 \\ -A_3^T & -B_3^T & S_p & 0 & 0 & 0 & 0 \\ -A_4^T & 0 & 0 & S_q & 0 & 0 & 0 \\ -A_5^T & 0 & 0 & 0 & S_r & 0 & 0 \\ 0 & -B_6^T & 0 & 0 & 0 & 0 & 0 \\ 0 & -B_7^T & 0 & 0 & 0 & 0 & S_s \end{bmatrix} = \left(\begin{array}{cc|c} A_1 & A_2 & \\ A_2^T & B_2 & \mathbb{B} \\ \hline \multicolumn{2}{c|}{-\mathbb{B}^T} & \mathbb{S} \end{array}\right),$$

where $\mathbb{B} = \begin{bmatrix} A_3 & A_4 & A_5 & 0 & 0 & \\ B_2 & B_3 & 0 & 0 & B_6 & B_7 \end{bmatrix}$ and $\mathbb{S} = \begin{bmatrix} S_p & 0 & 0 & 0 & 0 \\ 0 & S_q & 0 & 0 & 0 \\ 0 & 0 & S_r & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & S_s \end{bmatrix}$. Since $g$ is biconvex, then $\nabla_z^2 g^p$ and $\nabla_x^2 g^p$ are positive semidefinite. Using the convexity of $v$, $w$, $l$, and $h$, we have that the matrices $\nabla_z^2 v$, $\nabla_z^2 w$, $\nabla_x^2 l$ and $\nabla_x^2 h$ are positive semidefinite. Furthermore, observe that for any square matrix $M$, we have that $M^T M$ is positive semidefinite. We conclude then that $\begin{bmatrix} A_1 & A_2 \\ A_2^T & B_2 \end{bmatrix}$ is negative semidefinite [21]. It is clear that $\mathbb{S}$ is negative semidefinite, we have then $\nabla\eta$ is negative semidefinite [21]. □

The following definition and lemma are used later to prove the stability of the dynamical neural network (30)-(36).

**Definition 2.** A mapping $\mathrm{F} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is said to be monotonic if

$$(x-y)^T(\mathrm{F}(x)-\mathrm{F}(y)) \geq 0, \qquad \forall x, y \in \mathbb{R}^n$$

**Lemma 7.** [22] A differentiable mapping $\mathrm{F} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is monotonic, if and only if the Jacobian matrix $\nabla\mathrm{F}(x)$, $\forall x \in \mathbb{R}^n$, is positive semidefinite.

**Theorem 8.** The dynamical neural network (30)-(36) is stable in the sense of Lyapunov and converges to a KKT point of (SP).

*Proof.* Let $\tilde{y}$ an equilibrium point of (30)-(36) and let $V_1$ the following Lyapunov function $V_1(y) = ||\eta(\tilde{y})|| + \frac{1}{2}||y-\tilde{y}||^2$.

We have $\frac{dV_1(y)}{dt} = \frac{d\eta(y)}{dt}^T \eta(y) + \eta(y)^T \frac{d\eta(y)}{dt} + (y-\tilde{y})^T \frac{dy}{dt}$. On the other hand, $\frac{d\eta}{dt} = \frac{d\eta}{dy}\frac{dy}{dt}$. We have then, $V_1(y) = \eta(y)^T(\nabla\eta(y)^T + \nabla\eta(y))\eta(y) + (y-\tilde{y}))^T\eta(y)$. Since $\nabla\eta$ is negative semidefinite, then $\eta(y)^T(\nabla\eta(y)^T + \nabla\eta(y))\eta(y) \leq 0$. Moreover, by Lemma 7 we have $(y-\tilde{y}))^T\eta(y) \leq 0$. We conclude that $\frac{dV_1(y)}{dt} \leq 0$ and since $V_1$ is positive we have that the dynamical neural network (30)-(36) is stable in the sense of Lyapunov [23].

Observe that $\frac{1}{2}||y-\tilde{y}||^2 \leq V_1(y)$, then there exists a convergent subsequence $(y(t_k))_{k\geq 0}$ such that $\lim_{k\longrightarrow\infty} t_k = +\infty$ and $\lim_{k\longrightarrow\infty} y(t_k) = \hat{y}$ where $\hat{y}$ satisfies $\frac{dV_1(\hat{y})}{dt} = 0$.

We have by LaSalle's invariance principle [24] that the neural network converges to the largest invariant set contained in $S$ which is defined by $S = \{y(t) | \frac{dV_1(y)}{dt} = 0\}$.

Notice that $\frac{dy}{dt} = 0 \Leftrightarrow \frac{dV_1(y)}{dt}$, we have then that $\hat{y}$ is an equilibrium point of the dynamical system (30)-(36).

We introduce a second Lyapunov function defined as follows $V_2(y) = ||\eta(\tilde{y})|| + \frac{1}{2}||y-\hat{y}||^2$. Since $V_2$ is continuously differentiable, $\eta(\hat{z}) = 0$ and $\lim_{k\longrightarrow\infty} y(t_k) = \hat{y}$ then $\lim_{t\longrightarrow\infty} V_2(y(t)) = V_2(\hat{y}) = 0$. On the other hand, we have $\frac{dV_2(y)}{dt} \leq 0$ which leads to $\frac{1}{2}||y-\hat{y}||^2 \leq V_2(y)$. We conclude that $\lim_{t\longrightarrow\infty} \|y-\hat{y}\| = 0$ and then $\lim_{t\longrightarrow\infty} y(t) = \hat{y}$. We proved then, that the neural network (30)-(36) is convergent in the sense of Lyupanov to an equilibrium point $\hat{y} = (\hat{z}, \hat{x}, \hat{\alpha}, \hat{\gamma}, \hat{\lambda}, \hat{\beta}, \hat{\zeta})$ where $(\hat{z}, \hat{x})$ is a KKT point of of problem (SP). □

## 4. Numerical experiments

In this Section, we conduct preliminary numerical results in order to evaluate the performances of our approach. For this purpose, all the numerical experiments were done using Python. To compute the partial derivatives and the jacobians, we use the package autograd. To generate the random instances, we use the package numpy.random. The ODEs of the recurrent dynamical neural networks are solved using the function solve_ivp of scipy.integrate library. We run our algorithms on Intel(R) Core(TM) i7-10610U CPU @ 1.80GHz. For the numerical experiments, we set $P_{min} = 0.1$, $P_{max} = 0.5$, $\epsilon = 0.1$, we generate the complex vectors $g_i \in \mathbb{C}^{T\times 1}$ and $g_i^H \in \mathbb{C}^{1\times T}$ for each $i \in \mathcal{U}$ according to an independent complex Gaussian distribution function with zero mean and variance equal to one. Then, we multiply each of these vectors by a factor in the set $\{3.0, 4.0, 5.0, 7.0\}$. We generate the parameter $\sigma_i$ for each $i \in \mathcal{U}$ according to an independent complex Gaussian distribution function with zero mean and variance equal to one. The variables $a_{ij}$ and $b_i$ are then computed as explained in Section 2. We assume that $\mu_{ij} = a_{ij}$, $\mu_i = b_i$ and we vary the values of $\sigma_{ij}$ and $\sigma_i$ in $\{0.1, 0.2, 0.3\}$. We compare our neural network with the state-of-the-art based convex approximations approach [25]. We only account for the quality of the solution and do not record the CPU time as current ODE solvers are time consuming.

### *4.1. Convergence analysis*

We first solve (SP) for $K = 5$ for different feasible initial point $y_0$, we observe the convergence process of the neural network for each case. We observe, see Figure 3, that the neural network converges to the same final value for the different starting points.

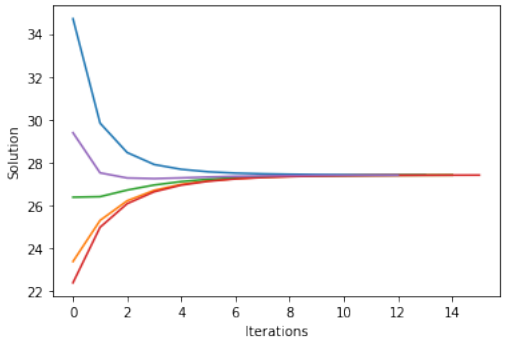



Figure 3: Convergence of the neural network different starting points $y_0$

### *4.2. Joint constraints vs. individual constraints*

In order to show the advantage of using joint constraints instead of individual constraints to deal with the uncertainty in constraints (8), We solve (SP) for different values of users, i.e., from $K = 2$ to $K = 20$ for both joint and individual chance constraints. We generate 100 instances of the stochastic variables $a_{ij}$ and $b_i$ and observe the number of times where the constraints (8) were not respected and we call them violated scenarios (VS for short). We recapitulate the obtained results in Table

| K | Individual constraints | VS | Joint constraints | VS |
|---|---|---|---|---|
| 2 | 5.45 | 16 | 5.99 | 6 |
| 3 | 6.87 | 21 | 7.43 | 9 |
| 5 | 35.57 | 39 | 36.99 | 8 |
| 7 | 48.98 | 53 | 50.43 | 11 |
| 10 | 39.40 | 62 | 41.68 | 10 |
| 15 | 82.30 | 84 | 85.21 | 12 |
| 20 | 113.65 | 82 | 117.33 | 10 |

Table 1: Individual constraints vs. Joint constraints for different values of $K$

1. Column one gives the number of users $K$, columns two and three give the optimal solution and the number of VS obtained using the individual constraints. Columns four and five represent the the optimal solution and the number of VS obtained using the joint constraints. We observe that the number of VS while using individual constraint is larger than the number of VS while using the joint constraints. The difference in VS number becomes more important as the value of $K$ increases. Using joint chance constraints ensures a better cover for the risk area.

### *4.3. The dynamical neural network vs. a sequential algorithm*

For the sake of comparison we solve problem (SP) using the neurodynamic approach in addition to the sequential algorithm proposed in [25]. The obtained results are recapitulated in Table 2. Column one gives the number of users $K$, columns two and three give the optimal solution and the number of VS obtained using the sequential algorithm. Columns four and five represent the optimal solution and the number of VS obtained using the dynamical neural network. Finally, column six gives the gap between the two solutions which is computed as follows $\text{GAP} = \frac{(\text{Solution}_{\text{SA}} - \text{Solution}_{\text{NN}})}{\text{Solution}_{\text{SA}}} \times 100$, with $\text{Solution}_{\text{NN}}$ and $\text{Solution}_{\text{SA}}$ are the objective values obtained with the neural network and the sequential algorithm, respectively. We observe that the dynamical neural network gives better solutions compared to the sequential algorithm. Moreover, the number of violated scenarios for the solutions obtained using the neurodynamical approach is slightly fewer than this obtained using the sequential algorithm.

Now we consider the case where $K = 5$ and we vary the value of $\epsilon$ in $[0.05, 0.4]$. We recapitulate the obtained results in Table 3. We observe that as $\epsilon$ increases the problem becomes less conservative. Moreover, we observe that the gap between the two approaches increases as $\epsilon$ increases as shown in Figure 4 and the number of violated scenarios increases see Figure 5. The difference in the number of violated scenarios becomes more significant as $\epsilon$ increases, hence the neurodynamical approach ensures a better robustness.

| K | Sequential Algorithm | VS | Neural Network | VS | GAP |
|---|---|---|---|---|---|
| 2 | 5.40 | 12 | 5.10 | 6 | 5.88 |
| 3 | 25.77 | 10 | 25.61 | 8 | 0.62 |
| 5 | 28.97 | 11 | 28.88 | 9 | 0.31 |
| 7 | 68.79 | 10 | 68.56 | 8 | 0.33 |
| 10 | 70.81 | 21 | 69.68 | 14 | 1.62 |
| 15 | 84.43 | 7 | 84.39 | 6 | 0.04 |
| 20 | 117.37 | 13 | 117.33 | 10 | 0.03 |

Table 2: Neural network vs. the sequential algorithm for different values of $K$

| $\epsilon$ | Sequential Algorithm | VS | Neural Network | VS | GAP |
|---|---|---|---|---|---|
| 0.05 | 30.23 | 4 | 29.89 | 2 | 1.13 |
| 0.1 | 29.47 | 15 | 29.07 | 9 | 1.37 |
| 0.15 | 28.96 | 22 | 28.53 | 11 | 1.50 |
| 0.2 | 28.56 | 32 | 28.10 | 19 | 1.63 |
| 0.3 | 27.87 | 54 | 27.40 | 26 | 1.71 |
| 0.4 | 27.30 | 63 | 26.81 | 34 | 1.82 |

Table 3: Neural network vs. the sequential algorithm for different values of $\epsilon$

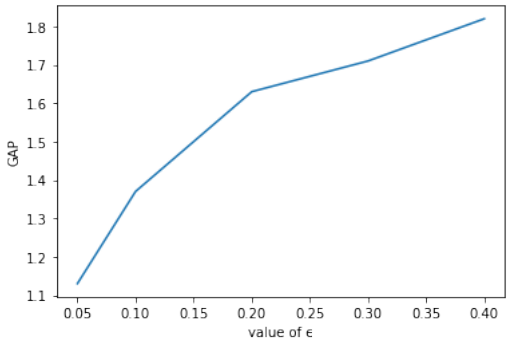



Figure 4: Evolution of GAP function to $\epsilon$

.

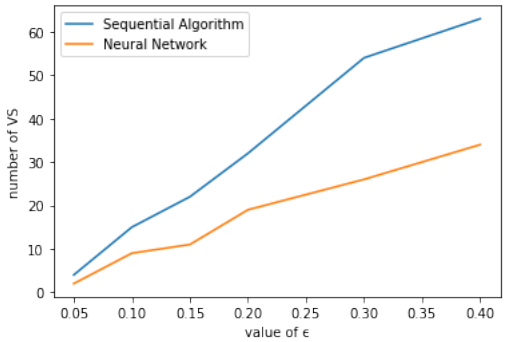



Figure 5: Evolution of VS function to $\epsilon$

.

## 5. Conclusion

This paper proposes a neurodynamic approach to maximize the worst user signal to interference noise ratio. We first give a geometric formulation for the maximization problem then we derive a stochastic formulation to deal with the uncertainty of wireless channels. Based on the partial KKT system of the obtained deterministic equivalent problem for the stochastic formulation, we propose a convergent dynamical system to solve the problem of maximizing the worst user signal to interference noise ratio. The dynamical neural network has the advantage of converging directly to a solution without using any convex approximation, unlike the state-of-art methods. In the numerical Section, we compare the performances of our neurodynamic approach to a sequential algorithm and show that

our method gives better upper bounds for the optimal solution and covers better the risk area.

## References


[1] X. Lin, F. Xu, J. Fu, Y. Wang, Resource allocation for tdd cell-free massive mimo systems, Electronics 11 (12). doi:10.3390/electronics11121914.
URL https://www.mdpi.com/2079-9292/11/12/1914

[2] S. Mosleh, H. Almosa, E. Perrins, L. Liu, Downlink resource allocation in cell-free massive mimo systems, in: 2019 International Conference on Computing, Networking and Communications (ICNC), 2019, pp. 883–887. doi:10.1109/ICCNC.2019.8685542.

[3] H. Yin, H. Wang, Y. Liu, D. Gesbert, Dealing with the mobility problem of massive mimo using extended prony's method, in: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1–6. doi:10.1109/ICC40277.2020.9149225.

[4] O. Dikmen, S. Kulac, Power allocation algorithms for massive mimo system, Avrupa Bilim ve Teknoloji Dergisi (28) (2021) 444 – 452. doi:10.31590/ejosat.1005325.

[5] I. Salah, M. Mourad, K. Rahouma, A. Hussein, Energy efficiency optimization in adaptive massive mimo networks for 5g applications using genetic algorithm, Optical and Quantum Electronics 54. doi:10.1007/s11082-021-03507-5.

[6] P. Adasme, I. Soto, E. S. Juan, F. Seguel, A. D. Firoozabadi, Maximizing signal to interference noise ratio for massive mimo: A mathematical programming approach, in: 2020 South American Colloquium on Visible Light Communications (SACVC), 2020, pp. 1–6. doi:10.1109/SACVLC50805.2020.9129889.

[7] W. Mei, R. Zhang, Performance analysis and user association optimization for wireless network aided by multiple intelligent reflecting surfaces, IEEE Transactions on Communications 69 (9) (2021) 6296–6312. doi:10.1109/TCOMM.2021.3087620.

[8] R. J. Duffin, E. L. Peterson, C. M. Zener, Geometric programming : theory and application, New York (N.Y.) : Wiley, 1967.

[9] W. Hoburg, P. Abbeel, Geometric programming for aircraft design optimization, AIAA Journal 52 (11) (2014) 2414–2426. arXiv:https://doi.org/10.2514/1.J052732, doi:10.2514/1.J052732.
URL https://doi.org/10.2514/1.J052732

[10] M. Chiang, 2005.

[11] S. P. Boyd, S.-J. Kim, D. D. Patil, M. A. Horowitz, Digital circuit optimization via geometric programming, Operations Research 53 (6) (2005) 899–932.
URL http://www.jstor.org/stable/25146929

[12] C. H. Scott, T. R. Jefferson, A generalisation of geometric programming with an application to information theory, Inf. Sci. 12 (1977) 263–269.

[13] C. Gicquel, J. Cheng, A joint chance-constrained programming approach for the single-item capacitated lot-sizing problem with stochastic demand, Annals of Operations Research 264 (2018) 123–155.

[14] M. Excoffier, C. Gicquel, O. Jouini, A joint chance-constrained programming approach for call center workforce scheduling under uncertain call arrival forecasts, Computers Industrial Engineering 96 (2016) 16–30. doi:https://doi.org/10.1016/j.cie.2016.03.013.
URL https://www.sciencedirect.com/science/article/pii/S0360835216300766

[15] K. Baker, A. Bernstein, Joint chance constraints in ac optimal power flow: Improving bounds through learning, IEEE Transactions on Smart Grid 10 (6) (2019) 6376–6385. doi:10.1109/TSG.2019.2903767.

[16] B. You, E. Esche, J. Weigert, J.-U. Repke, Joint chance constraint approach based on data-driven models for optimization under uncertainty applied to the williams-otto process, in: M. Türkay, R. Gani (Eds.), 31st European Symposium on Computer Aided Process Engineering, Vol. 50 of Computer Aided Chemical Engineering, Elsevier, 2021, pp. 523–528. doi:https://doi.org/10.1016/B978-0-323-88506-5.50083-8.
URL https://www.sciencedirect.com/science/article/pii/B9780323885065500838

[17] M. Ono, Y. Kuwata, J. Balaram, Joint chance-constrained dynamic programming, in: 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), 2012, pp. 1915–1922. doi:10.1109/CDC.2012.6425906.

[18] S. T. Valduga, L. Deneire, R. Aparicio-Pardo, A. L. F. de Almeida, T. F. Maciel, J. C. M. Mota, Low-complexity heuristics to beam selection and rate adaptation in sparse massive mimo systems, Transactions on Emerging Telecommunications Technologies 29 (9) (2018) e3459, e3459 ett.3459. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.3459, doi:https://doi.org/10.1002/ett.3459.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3459

[19] M. Jiang, Z. Meng, R. Shen, Partial exactness for the penalty function of biconvex programming, Entropy 23 (2). doi:10.3390/e23020132.
URL https://www.mdpi.com/1099-4300/23/2/132

[20] R. K.Miller, A. N. Michel, Ordinary Differential Equations, Academic Press, 1981.

[21] C. Foias, A. E. Frazho, Positive Definite Block Matrices, Birkhäuser Basel, Basel, 1990, pp. 547–586. doi:10.1007/978-3-0348-7712-1_16.
URL https://doi.org/10.1007/978-3-0348-7712-1_16

[22] R. T. Rockafellar, R. J.-B. Wets, Variational Analysis, Springer, Berlin, Heidelberg, 1998. doi:10.1007/978-3-642-02431-3.

[23] R. M. Murray, S. S. Sastry, L. Zexiang, A Mathematical Introduction to Robotic Manipulation, 1st Edition, CRC Press, Inc., USA, 1994.

[24] J.-J. E. Slotine, W. Li, Applied Nonlinear Control, Prentice Hall, 1991.
URL https://books.google.fr/books?id=cwpRAAAAMAAJ

[25] J. Liu, A. Lisser, Z. Chen, Stochastic geometric optimization with joint probabilistic constraints, Operations Research Letters 44 (5) (2016) 687–691. doi:https://doi.org/10.1016/j.orl.2016.08.002.