Distributionally Robust Geometric Joint Chance-Constrained Optimization: Neurodynamic Approaches

Siham Tassouli, Abdel Lisser

Distributionally Robust Geometric Joint Chance-Constrained Optimization: Neurodynamic Approaches

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

This paper proposes a two-time scale neurodynamic duplex approach to solve distributionally robust geometric joint chance-constrained optimization problems. The probability distributions of the row vectors are not known in advance and belong to a certain distributional uncertainty set. In our paper, we study three uncertainty sets for the unknown distributions. The neurodynamic duplex is designed based on three projection equations. The main feature of our framework is to propose a neural network-based method to solve distributionally robust joint chance-constrained optimization problems that converges in probability to the global optimum without the use of standard state-of-the-art solving methods. In the numerical Section, we apply the proposed approach to solve a problem of shape optimization and a telecommunication problem.

Keywords: Dynamical neural network, Distributionally robust optimization, Joint chance constraints, Particle swarm optimization, Two-timescale system.

2010 MSC: 00-01, 99-00

1. Introduction

Chance-constrained programming appears with the increased need to include uncertainty in complex decision-making models. It was introduced for the first time by Charnes & Cooper [1]. Since then, chance-constrained optimization has been widely studied, and the range of applications is very large, e.g., resource allocation in communication and network systems [2, 3], information theory [4, 5], chemical engineering [6], computational finance [7, 8], metal cutting optimization [9], spatial gate sizing [10], profit maximization [11] and biochemical systems [12], portfolio selection [13], energy systems operations [14], water quality management [15] and transportation problems [16]. In this paper, we study chance-constrained geometric programs. Liu et al. [17] propose some convex based approximations to come up with lower and upper bounds for geometric programs with joint probabilistic constraints when the stochastic parameters are normally distributed and pairwise independent. Shiraz et al. [18] use a duality algorithm to solve fuzzy chance-constrained geometric programs. Tassouli & Lisser [19] propose a neurodynamic approach to solve geometric programs with joint probabilistic constraints with normally distributed coefficients and independent matrix row vectors. Liu et al. [20]

*Corresponding author.

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

propose convex approximation based algorithms to solve distributionally robust geometric programs with individual and joint chance constraints.

Geometric programming is a method for solving a class of nonlinear problems. It is used to minimize functions that are in the form of posynomials subject to constraints of the same type. It was introduced for the first time by Duffin et al. [21]. Since then geometric programming was employed to solve several optimization problems, e.g, resource allocation in communication and network systems [2, 3], information theory [4, 5], chemical engineering [6], computational finance [7, 8], metal cutting optimization [9], spatial gate sizing [10], profit maximization [11] and biochemical systems [12].

In this paper, we are interested in solving joint chance-constrained geometric optimization problems. We study the case where the distribution of the random parameters is unknown, aka distributionally robust optimization. In fact, we may only know partial information about the statistical properties of the stochastic parameters. El Ghaoui & Lebret [22] use second-order cone programming to solve least-squares problems where the stochastic parameters are not known but bounded. Bertsimas & Sim [23] introduce a less conservative approach to solve linear optimization problems with uncertain data. Bertsimas & Brown [24] propose a general scheme for designing uncertainty sets for robust optimization. Wiesemann et al. [25] propose standardized ambiguity sets for modeling and solving distributionally robust optimization problems. Peng et al. [26] study one density-based uncertainty set and four two-moments based uncertainty sets to solve games with distributionally robust joint chance constraints. Cheng et al. [27] solve a distributionally robust quadratic knapsack problem. Dou & Anitescu [28] propose a new ambiguity set tailored to unimodal and seemingly symmetric distributions to deal with distributionally robust chance constraints. Li & Ke [29] approximate a distributionally robust chance constraint by the worst-case Conditional Value-at-Risk. Hanasusanto et al. [30] approximate two-stage distributionally robust programs with binary recourse decisions. Georghiou et al. [31] propose a primal-dual lifting scheme for the solution of two-stage robust optimization problems.

Recent papers have considered the use of distributionally robust approaches in transportation network optimization problems [32], multistage distribution system planning [33], portfolio optimization problems [34, 35], planning and scheduling [36], risk measures [37], multimodal demand problems [38], appointment scheduling [39], vehicle routine problems [40] and energy and reserve dispatch [41].

The use of neural networks to solve optimization problems has been actively studied since the 1980s when the idea was first introduced by Tank & Hopfield [42]. Xia & Wang [43] present a recurrent neural network for solving nonlinear convex programming problems subject to nonlinear inequality constraints. Wang [44] proposes a deterministic annealing neural network for convex programming. Nazemi & Omedi [45] presents a neural network model for solving the shortest path problems. Tassouli & Lisser [19] propose a recurrent neural network to solve geometric joint chance-constrained optimization problems.

In addition to the significant accomplishments achieved by individual recurrent neural networks (RNNs), it is important to note that one-time-scale RNNs have limitations when it comes to constrained global optimization problems and more general problem domains. The dynamic behaviors

of one-time-scale RNNs can exhibit drastic changes and become unpredictable when dealing with certain optimization problems. Neuroscience studies have provided evidence of the existence of two-time-scales in the brain, where different processes operate at different time scales [46]. Therefore, a neurodynamic model with two-time-scales is considered more biologically plausible for emulating brain functions than a model with only one-time-scale. This paper proposes a two-timescale duplex neurodynamic approach for distributionally robust joint chance-constrained optimization problems, which is formulated using a biconvex reformulation. Unlike other existing methods that give lower or upper bounds to this kind of problems, the proposed approach employs two recurrent neural networks that operate collaboratively at two different timescales and converge almost surely to a global optimal solution of the given distributionally robust optimization problem. The main contributions of our work are threefold.

(i) On the formulation side, we derive the deterministic formulations of the distributionally robust initial problem of each uncertainty set. Then, we propose two neurodynamic approaches to solve the resulting problems.

(ii) On the theoretical side, we show that the proposed neurodynamic approaches are stable and convergent.

(iii) On the numerical side, we show that the proposed neurodynamic methods cover well the risk area induced by the distributionally robust chance constraints.

The rest of the paper is organized as follows. In Section 2, we study two uncertainty sets to solve a distributionally robust geometric chance-constrained optimization problem and give four deterministic equivalent problems. In Section 3, we propose a recurrent neural network to solve the first three resulting problems and prove its convergence and stability. We consider in Section 4 a duplex of two two-timescale recurrent neural networks to solve the last deterministic problem and prove its convergence almost surely to the global optimum. In Section 5, we evaluate the performances of the proposed neurodynamic approaches by solving a shape optimization problem and a telecommunication problem.

2. Problem statement and reformulation

A general form of a geometric program is given as follows

$$\begin{aligned} \min_{t\in\mathbb{R}^M_{++}} \ & \sum_{i=1}^{I_0} c_i^0 \prod_{j=1}^{M} t_j^{a_{ij}^0}, \\ \text{s.t} \ & \sum_{i=1}^{I_k} c_i^k \prod_{j=1}^{M} t_j^{a_{ij}^k} \leq 1, k = 1, ..., K, \end{aligned} \tag{1}$$

where $c_i^k,\ i = 1, ..., I_k,\ k = 0, ..., K$ are positive constants and the exponents $a_{ij}^k, i = 1, ..., I_k, j = 1, ..., M, k = 0, 1, ..., K$ are real constants.

In this paper, we consider the case where the coefficients c_i are not known. Consequently, we reformulate the optimization problem (1) as follows

$$\min_{t\in\mathbb{R}^M_{++}} \sup_{\mathcal{F}_0\in\mathcal{D}_0} \mathbb{E}_{\mathcal{F}_0}\left[\sum_{i=1}^{I_0} c_i^0 \prod_{j=1}^{M} t_j^{a_{ij}^0}\right], \tag{JCP}$$
$$\text{s.t} \quad \inf_{\mathcal{F}\in\mathcal{D}} \mathbb{P}_{\mathcal{F}}\left(\sum_{i=1}^{I_k} c_i^k \prod_{j=1}^{M} t_j^{a_{ij}^k} \le 1, k=1,....,K\right) \ge 1-\epsilon,$$

where $\mathcal{F}_0$ is the probabilistic distribution of vector $C^0 = (c_1^0, .., c_{I_0}^0)^T$, $\mathcal{F}$ is the joint distribution for $C^1 = (c_1^1, .., c_{I_1}^1)^T$, ..., $C^k = (c_1^k, .., c_{I_k}^k)^T$, $\mathcal{D}_0$ is the uncertainty set for the probability distribution $\mathcal{F}_0$, $\mathcal{D}$ is the uncertainty set for the probability distribution $\mathcal{F}$ and $1-\epsilon$, $\epsilon \in (0, 0.5]$, is the confidence parameter for the joint constraint.

This paper considers the distributionally robust geometric programs (JCP) using two different sets of uncertainty. The first set focuses on uncertainties in distributions, considering both known and unknown first two order moments. The second set incorporates first order moments along with nonnegative support constraints.

2.1. Uncertainty Sets with First Two Order Moments

We first consider that the mean vector of C^k, $k = 0, 1, ..., K$ lies in an ellipsoid of size $\gamma_1^k \ge 0$ with center μ_k and that the covariance matrix of C^k, $k = 0, 1, ..., K$ lies in a positive semidefinite cone of center $\Sigma_k = \left\{\sigma_{i,j}^k, i, j = 1, ..., I_k\right\}$. We define for every $k = 0, 1, ..., K$, $\mathcal{D}_k^2(\mu_k, \Sigma_k) = \left\{ \mathcal{F}_k \;\middle|\; \begin{array}{r} (\mathbb{E}_{\mathcal{F}_k}[C^k] - \mu_k)^T \Sigma_k^{-1} (\mathbb{E}_{\mathcal{F}_k}[C^k] - \mu_k) \le \gamma_1^k \\ \mathrm{COV}_{\mathcal{F}_k}(C^k) \preceq \gamma_2^k \Sigma_k \end{array} \right\}$, where F_k is the probability distribution of C^k, $\gamma_2^k \ge 0$ and $\mathrm{COV}_{\mathcal{F}_k}$ is a covariance operator under probability distribution $\mathcal{F}_k$ of C^k.

Based on whether the row vectors C^k, $k = 1, .., K$ are mutually independent or dependent, we have two cases.

2.1.1. Case (JCP) with Jointly Independent Row Vectors.

Assumption 1. We assume that $\mathcal{D} = \{\mathcal{F} | \mathcal{F} = \mathcal{F}_1\mathcal{F}_2...\mathcal{F}_K\}$, where $\mathcal{F}$ is the joint distribution for mutually independent random vectors C^1 , C^2 , ..., C^K with marginals $\mathcal{F}_1$, $\mathcal{F}_2$, ..., $\mathcal{F}_K$.

Theorem 1. Given Assumption 1, (JCP) is equivalent to

$$(\mathrm{JCP}_{ind}) \min_{t\in\mathbb{R}^M_{++}, y\in\mathbb{R}^K_+} \sum_{i=1}^{I_0} \mu_i^0 \prod_{j=1}^{M} t_j^{a_{ij}^0} + \sqrt{\gamma_1^0}\sqrt{\sum_{i=1}^{I_0}\sum_{l=1}^{I_0} \sigma_{i,l}^0 \prod_{j=1}^{M} t_j^{a_{ij}^0 + a_{lj}^0}}, \tag{2}$$

$$\text{s.t} \quad \sum_{i=1}^{I_k} \mu_i^k \prod_{j=1}^{M} t_j^{a_{ij}^k} + \sqrt{\gamma_1^k}\sqrt{\sum_{i=1}^{I_k}\sum_{l=1}^{I_k} \sigma_{i,l}^k \prod_{j=1}^{M} t_j^{a_{ij}^k + a_{lj}^k}}$$
$$+\sqrt{\frac{y_k}{1-y_k}}\sqrt{\gamma_2^k}\sqrt{\sum_{i=1}^{I_k}\sum_{l=1}^{I_k} \sigma_{i,l}^k \prod_{j=1}^{M} t_j^{a_{ij}^k + a_{lj}^k}} \le 1, \; k = 1, ..., K, \tag{3}$$

$$\prod_{k=1}^{K} y_k \ge 1-\epsilon, \; 0 < y_k \le 1 \; , \; k = 1, ..., K. \tag{4}$$

Proof. As the row vectors C^k, $k = 1, ..., K$ are mutually independent, (JCP) is written equivalently by introducing K nonegative auxiliary variables y_k as [19].

$$\begin{aligned}
&\min_{t\in\mathbb{R}^M_{++}} \sup_{\mathcal{F}_0\in\mathcal{D}_0} \mathbb{E}_{\mathcal{F}_0}\left[\sum_{i=1}^{I_0} c_i^0 \prod_{j=1}^{M} t_j^{a_{ij}^0}\right], \\
&\text{s.t} \quad \inf_{\mathcal{F}_k\in\mathcal{D}_k} \mathbb{P}_{\mathcal{F}_k}\left(\sum_{i=1}^{I_k} c_i^k \prod_{j=1}^{M} t_j^{a_{ij}^k} \le 1\right) \ge y_k, , k = 1, ..., K, \\
&\qquad \prod_{k=1}^{K} y_k \ge 1-\epsilon,\ 0 < y_k \le 1\ ,\ k = 1, ..., K.
\end{aligned}$$

By Theorem 1 in [20], we conclude that (JCP) is equivalent to (JCP$_{ind}$). □

Problem (JCP$_{ind}$) is not convex. By applying the logarithmic transformation $r_j = log(t_j)$, $j = 1, ..., M$ and $x_k = \log(y_k)$, $k = 1, ..., K$, we have the following equivalent reformulation of (JCP$_{ind}$)

$$(\text{JCP}^{log}_{ind}) \min_{r\in\mathbb{R}^M, x\in\mathbb{R}^K} \sum_{i=1}^{I_0} \mu_i^0 \exp\left\{\sum_{j=1}^{M} a_{ij}^0 r_j\right\} + \sqrt{\gamma_1^0}\sqrt{\sum_{i=1}^{I_0}\sum_{l=1}^{I_0} \sigma_{i,l}^0 \exp\left\{\sum_{j=1}^{M}(a_{ij}^0 + a_{lj}^0) r_j\right\}}, \tag{5}$$

$$\begin{aligned}
\text{s.t} \quad & \sum_{i=1}^{I_k} \mu_i^k \exp\left\{\sum_{j=1}^{M} a_{ij}^k r_j\right\} + \sqrt{\gamma_1^k}\sqrt{\sum_{i=1}^{I_k}\sum_{l=1}^{I_k} \sigma_{i,l}^k \exp\left\{\sum_{j=1}^{M}(a_{ij}^k + a_{lj}^k) r_j\right\}} \\
& + \sqrt{\gamma_2^k}\sqrt{\sum_{i=1}^{I_k}\sum_{l=1}^{I_k} \sigma_{i,l}^k \exp\left\{\sum_{j=1}^{M}(a_{ij}^k + a_{lj}^k) r_j + \log\left(\frac{e^{x_k}}{1-e^{x_k}}\right)\right\}} \le 1,\ k = 1, ..., K,
\end{aligned}$$

$$\sum_{k=1}^{K} x_k \ge \log(1-\epsilon),\ x_k \le 0\ ,\ k = 1, ..., K. \tag{6}$$

Theorem 2. [20] If $\sigma_{i,l}^k \ge 0$ for all i, l and k, problem (JCP$^{log}_{ind}$) is a convex programming problem.

2.1.2. Case (JCP) with Jointly Dependent Row Vectors.

In this case, (JCP) is equivalent to [20]

$$(\text{JCP}_{dep}) \min_{t\in\mathbb{R}^M_{++}, y\in\mathbb{R}^K_+} \sum_{i=1}^{I_0} \mu_i^0 \prod_{j=1}^{M} t_j^{a_{ij}^0} + \sqrt{\gamma_1^0}\sqrt{\sum_{i=1}^{I_0}\sum_{l=1}^{I_0} \sigma_{i,l}^0 \prod_{j=1}^{M} t_j^{a_{ij}^0 + a_{lj}^0}}, \tag{7}$$

$$\begin{aligned}
\text{s.t} \quad & \sum_{i=1}^{I_k} \mu_i^k \prod_{j=1}^{M} t_j^{a_{ij}^k} + \sqrt{\gamma_1^k}\sqrt{\sum_{i=1}^{I_k}\sum_{l=1}^{I_k} \sigma_{i,l}^k \prod_{j=1}^{M} t_j^{a_{ij}^k + a_{lj}^k}} \\
& + \sqrt{\frac{y_k}{1-y_k}}\sqrt{\gamma_2^k}\sqrt{\sum_{i=1}^{I_k}\sum_{l=1}^{I_k} \sigma_{i,l}^k \prod_{j=1}^{M} t_j^{a_{ij}^k + a_{lj}^k}} \le 1,\ k = 1, ..., K,
\end{aligned} \tag{8}$$

$$\sum_{k=1}^{K} y_k \ge K-\epsilon,\ 0 < y_k \le 1\ ,\ k = 1, ..., K. \tag{9}$$

As for the independent case, we obtain the following biconvex equivalent problem for (JCP$_{dep}$)

$$(\text{JCP}^{log}_{dep}) \min_{r\in\mathbb{R}^M, x\in\mathbb{R}^K} \sum_{i=1}^{I_0}\mu_i^0 \exp\left\{\sum_{j=1}^{M} a_{ij}^0 r_j\right\} + \sqrt{\gamma_1^0}\sqrt{\sum_{i=1}^{I_0}\sum_{l=1}^{I_0}\sigma_{i,l}^0 \exp\left\{\sum_{j=1}^{M}(a_{ij}^0 + a_{lj}^0)r_j\right\}}, \tag{10}$$

$$\text{s.t} \quad \sum_{i=1}^{I_k}\mu_i^k \exp\left\{\sum_{j=1}^{M} a_{ij}^k r_j\right\} + \sqrt{\gamma_1^k}\sqrt{\sum_{i=1}^{I_k}\sum_{l=1}^{I_k}\sigma_{i,l}^k \exp\left\{\sum_{j=1}^{M}(a_{ij}^k + a_{lj}^k)r_j\right\}}$$

$$+\sqrt{\gamma_2^k}\sqrt{\sum_{i=1}^{I_k}\sum_{l=1}^{I_k}\sigma_{i,l}^k \exp\left\{\sum_{j=1}^{M}(a_{ij}^k + a_{lj}^k)r_j + \log\left(\frac{y_k}{1-y_k}\right)\right\}} \leq 1,\ k = 1, ..., K,$$

$$\sum_{k=1}^{K} y_k \geq K - \epsilon,\ 0 < y_k \leq 1\ ,\ k = 1, ..., K. \tag{11}$$

Theorem 3. [20] If $\epsilon \leq 0.5$ and $\sigma_{i,l}^k \geq 0$ for all i, l and k, problem (JCP^{log}_{dep}) is a convex programming problem.

2.2. Uncertainty Sets with Known First Order Moment and Nonnegative Support

In this section, we consider uncertainty sets with nonnegative supports and known first-order moments. The uncertainty sets for (JCP) can be formulated as follows

$$\mathcal{D}_k^3(\mu_k, \Sigma_k) = \left\{\mathcal{F}_k \middle| \begin{array}{r} \mathbb{E}[C^k] = \mu^k \\ \mathbb{P}_{\mathcal{F}_k}[C^k \geq 0] = 1 \end{array}\right\}, k = 0, 1, ..., K, \text{ where } \mu^k > 0.$$

2.2.1. Case (JCP) with Jointly Independent Row Vectors.

We first consider the case when the marginal distributions in the uncertainty set are jointly independent. Using the strong duality [20], (JCP) can be reformulated as follows

$$(\text{JCP}^{\text{ind}}_{NS}) \min_{t\in\mathbb{R}^M_{++},\lambda,\beta,\pi} \sum_{i=1}^{I_0}\mu_i^0 \prod_{j=1}^{M} t_j^{a_{ij}^0}, \tag{12}$$

$$\text{s.t} \quad \prod_{k=1}^{K} y_k \geq 1 - \epsilon,\ 0 \leq y_k \leq 1\ ,\ k = 1, ..., K, \tag{13}$$

$$y_k\lambda_k^{-1} - \lambda_k^{-1}\beta^{k^T}\mu^k \leq 1, k = 1, ..., K, \tag{14}$$

$$\beta_k \leq 0, 0 < \lambda \leq 1, k = 1, ..., K, \tag{15}$$

$$\lambda_k^{-1}\pi_k \geq 1, k = 1, ..., K, \tag{16}$$

$$(-\beta_k)^{-1}\pi_k \prod_{j=1}^{M} t_j^{a_{ij}^k} \leq 1, i = 1, ..., I_k, k = 1, ..., K, \tag{17}$$

(JCP) can be reformulated as a convex problem using a logarithmic transformation $x_j = \log(y_j)$, $t_j = \log(r_j)$, $\tilde{\lambda_k} = \log(\lambda_k)$, $\tilde{\beta_k} = \log(-\beta_k)$, $\tilde{\pi} = \log(\pi)$. Problem (JCP_{NS}) becomes,

$$(\text{JCP}^{log}_{NS-ind}) \min_{x,r,\tilde{\lambda},\tilde{\beta},\tilde{\pi}} \sum_{i=1}^{I_0} \mu_i^0 \exp\left\{\sum_{j=1}^{M} a_{ij}^0 r_j\right\}, \tag{18}$$

$$\text{s.t} \quad \sum_{k=1}^{K} x_k \geq \log(1-\epsilon),\ x_k \leq 0, k = 1, ..., K, \tag{19}$$

$$\exp(x_k - \tilde{\lambda}_k) + \sum_{i=1}^{I_k} \exp\left\{-\tilde{\lambda}_k + \tilde{\beta}_i^k + \log\mu_i^k\right\} \leq 1, k = 1, ..., K,$$

$$\tilde{\lambda}_k \leq 0, k = 1, ..., K, \tag{20}$$

$$\tilde{\lambda}_k \leq \tilde{\pi}_k, k = 1, ..., K, \tag{21}$$

$$\tilde{\pi}_k + \sum_{j=1}^{M} a_{ij}^k r_j - \tilde{\beta}_i^k \leq 0, i = 1, ..., I_k, k = 1, ..., K. \tag{22}$$

120 *2.2.2. Case (JCP) with Jointly Dependent Row Vectors.*

In the case where the constraints of (JCP) are jointly dependent, we have the following deterministic equivalent

$$(\text{JCP}^{\text{dep}}_{NS}) \min_{t\in\mathbb{R}^M_{++},\lambda,\beta,\pi} \sum_{i=1}^{I_0} \mu_i^0 \prod_{j=1}^{M} t_j^{a_{ij}^0}, \tag{23}$$

$$\text{s.t} \quad \prod_{k=1}^{K} y_k \geq K - \epsilon,\ 0 \leq y_k \leq 1\ ,\ k = 1, ..., K, \tag{24}$$

$$y_k \lambda_k^{-1} - \lambda_k^{-1} \beta^{k^T} \mu^k \leq 1, k = 1, ..., K, \tag{25}$$

$$\beta_k \leq 0, 0 < \lambda \leq 1, k = 1, ..., K, \tag{26}$$

$$\lambda_k^{-1} \pi_k \geq 1, k = 1, ..., K, \tag{27}$$

$$(-\beta_k)^{-1} \pi_k \prod_{j=1}^{M} t_j^{a_{ij}^k} \leq 1, i = 1, ..., I_k, k = 1, ..., K, \tag{28}$$

We apply a log transformation to convert (JCP^{log}_{NS}) into a biconvex problem. We take $t_j = \log(r_j)$, $\tilde{\lambda_k} = \log(\lambda_k)$, $\tilde{\beta_k} = \log(-\beta_k)$, $\tilde{\pi} = \log(\pi)$ and obtain

$$(\text{JCP}^{log}_{NS-dep}) \min_{x,r,\tilde{\lambda},\tilde{\beta},\tilde{\pi}} \sum_{i=1}^{I_0} \mu_i^0 \exp\left\{\sum_{j=1}^{M} a_{ij}^0 r_j\right\}, \tag{29}$$

$$\text{s.t} \quad \prod_{k=1}^{K} y_k \geq K - \epsilon,\ 0 \leq y_k \leq 1\ ,\ k = 1, ..., K, \tag{30}$$

$$y_k \exp(-\tilde{\lambda}_k) + \sum_{i=1}^{I_k} \exp\left\{-\tilde{\lambda}_k + \tilde{\beta}_i^k + \log\mu_i^k\right\} \leq 1, k = 1, ..., K,$$

$$\tilde{\lambda}_k \leq 0, k = 1, ..., K, \tag{31}$$

$$\tilde{\lambda}_k \leq \tilde{\pi}_k, k = 1, ..., K, \tag{32}$$

$$\tilde{\pi}_k + \sum_{j=1}^{M} a_{ij}^k r_j - \tilde{\beta}_i^k \leq 0, i = 1, ..., I_k, k = 1, ..., K. \tag{33}$$

3. A dynamical recurrent neural network for ($\mathbf{JCP}^{\boldsymbol{log}}_{\boldsymbol{dep}}$), ($\mathbf{JCP}^{\boldsymbol{log}}_{\boldsymbol{ind}}$) and ($\mathbf{JCP}^{\boldsymbol{log}}_{\boldsymbol{NS-ind}}$)

Observe that (JCP^{log}_{dep}), (JCP^{log}_{ind}) and ($\mathrm{JCP}^{log}_{NS_ind}$) can be written in the following general form

$$\begin{aligned} &\min_{r} f(z), \\ &\text{s.t. } g(z) \leq 0, \end{aligned} \tag{34}$$

where f and g are two convex functions.

For (JCP^{log}_{ind}), $z=(r,x)^T$, $f(z)=\sum_{i=1}^{I_0}\mu_i^0\exp\left\{\sum_{j=1}^{M}a_{ij}^0 r_j\right\}+\sqrt{\gamma_1^0}\sqrt{\sum_{i=1}^{I_0}\sum_{l=1}^{I_0}\sigma_{i,l}^0\exp\left\{\sum_{j=1}^{M}(a_{ij}^0+a_{lj}^0)r_j\right\}}$

and $g(z)=$

$$\begin{pmatrix} \sum\limits_{i=1}^{I_1}\mu_i^1\exp\left\{\sum_{j=1}^{M}a_{ij}^1 r_j\right\}+\sqrt{\gamma_1^1}\sqrt{\sum\limits_{i=1}^{I_1}\sum\limits_{l=1}^{I_1}\sigma_{i,l}^1\exp\left\{\sum\limits_{j=1}^{M}(a_{ij}^1+a_{lj}^1)r_j\right\}} \\ +\sqrt{\frac{e^{x_1}}{1-e^{x_1}}}\sqrt{\gamma_2^1}\sqrt{\sum\limits_{i=1}^{I_1}\sum\limits_{l=1}^{I_1}\sigma_{i,l}^1\exp\left\{\sum\limits_{j=1}^{M}(a_{ij}^1+a_{lj}^1)r_j\right\}}-1 \\ \vdots \\ \sum\limits_{i=1}^{I_K}\mu_i^K\exp\left\{\sum\limits_{j=1}^{M}a_{ij}^K r_j\right\}+\sqrt{\gamma_1^K}\sqrt{\sum\limits_{i=1}^{I_K}\sum\limits_{l=1}^{I_K}\sigma_{i,l}^K\exp\left\{\sum_{j=1}^{M}(a_{ij}^K+a_{lj}^K)r_j\right\}}+ \\ \sqrt{\frac{e^{x_K}}{1-e^{x_K}}}\sqrt{\gamma_2^K}\sqrt{\sum\limits_{i=1}^{I_K}\sum\limits_{l=1}^{I_K}\sigma_{i,l}^K\exp\left\{\sum\limits_{j=1}^{M}(a_{ij}^K+a_{lj}^K)r_j\right\}}-1 \\ \log(1-\epsilon)-\sum_{k=1}^{K}x_k \\ x_1 \\ \vdots \\ x_K \end{pmatrix}.$$

For (JCP^{log}_{dep}), $z=(r,x)^T$, $f(z)=\sum_{i=1}^{I_0}\mu_i^0\exp\left\{\sum_{j=1}^{M}a_{ij}^0 r_j\right\}+\sqrt{\gamma_1^0}\sqrt{\sum_{i=1}^{I_0}\sum_{l=1}^{I_0}\sigma_{i,l}^0\exp\left\{\sum_{j=1}^{M}(a_{ij}^0+a_{lj}^0)r_j\right\}}$

and $g(z)=$

$$\begin{pmatrix} \sum\limits_{i=1}^{I_1}\mu_i^1\exp\left\{\sum_{j=1}^{M}a_{ij}^1 r_j\right\}+\sqrt{\gamma_1^1}\sqrt{\sum\limits_{i=1}^{I_1}\sum\limits_{l=1}^{I_1}\sigma_{i,l}^1\exp\left\{\sum\limits_{j=1}^{M}(a_{ij}^1+a_{lj}^1)r_j\right\}} \\ +\sqrt{\frac{e^{x_1}}{1-e^{x_1}}}\sqrt{\gamma_2^1}\sqrt{\sum\limits_{i=1}^{I_1}\sum\limits_{l=1}^{I_1}\sigma_{i,l}^1\exp\left\{\sum\limits_{j=1}^{M}(a_{ij}^1+a_{lj}^1)r_j\right\}}-1 \\ \vdots \\ \sum\limits_{i=1}^{I_K}\mu_i^K\exp\left\{\sum\limits_{j=1}^{M}a_{ij}^K r_j\right\}+\sqrt{\gamma_1^K}\sqrt{\sum\limits_{i=1}^{I_K}\sum\limits_{l=1}^{I_K}\sigma_{i,l}^K\exp\left\{\sum_{j=1}^{M}(a_{ij}^K+a_{lj}^K)r_j\right\}}+ \\ \sqrt{\frac{e^{x_K}}{1-e^{x_K}}}\sqrt{\gamma_2^K}\sqrt{\sum\limits_{i=1}^{I_K}\sum\limits_{l=1}^{I_K}\sigma_{i,l}^K\exp\left\{\sum\limits_{j=1}^{M}(a_{ij}^K+a_{lj}^K)r_j\right\}}-1 \\ \log(K-\epsilon)-\sum_{k=1}^{K}x_k \\ x_1 \\ \vdots \\ x_K \end{pmatrix}.$$

For ($\mathrm{JCP}^{log}_{NS-ind}$), $z=(r,x,\tilde{\lambda},\tilde{\beta},\tilde{\pi})^T$, $f(z)=\sum\limits_{i=1}^{I_0}\mu_i^0\prod_{j=1}^{M}t_j^{a_{ij}^0}$

and $g(z) = \begin{pmatrix} \log(1-\epsilon) - \sum_{k=1}^{K} x_k \\ x_1 \\ \vdots \\ x_K \\ \exp(x_1 - \tilde{\lambda}_1) + \sum_{i=1}^{I_1} \exp\left\{-\tilde{\lambda}_1 + \tilde{\beta}_i^1 + \log\mu_i^1\right\} - 1 \\ \vdots \\ \exp(x_K - \tilde{\lambda}_K) + \sum_{i=1}^{I_K} \exp\left\{-\tilde{\lambda}_K + \tilde{\beta}_i^K + \log\mu_i^K\right\} - 1 \\ \tilde{\lambda}_1 \\ \vdots \\ \tilde{\lambda}_1 - \tilde{\pi}_1 \\ \vdots \\ \tilde{\lambda}_K - \tilde{\pi}_K \\ \tilde{\pi}_1 + \sum_{j=1}^{M} a_{ij}^1 r_j - \tilde{\beta}_i^1 \leq 0, i = 1, ..., I_1 \\ \vdots \\ \tilde{\pi}_K + \sum_{j=1}^{M} a_{ij}^K r_j - \tilde{\beta}_i^K \leq 0, i = 1, ..., I_K \end{pmatrix}$.

We know that z^* is an optimal solution of (34) if and only if the following Karush–Kuhn–Tucker (KKT) conditions are satisfied.

$$\nabla f(z) + \nabla g(z)^T \gamma = 0 \tag{35}$$

$$\gamma \geq 0, \gamma^T g(z) = 0 \tag{36}$$

To solve problem (34), we propose a dynamical recurrent neural network driven by the following ODE system

$$\kappa \frac{dz}{dt} = -\left(\nabla f(z) + \nabla g(z)^T (\gamma + g(z))_+\right) \tag{37}$$

$$\kappa \frac{d\gamma}{dt} = -\gamma + (\gamma + g(z))_+ \tag{38}$$

where $z(.)$ and $\gamma(.)$ are two time-dependent variables, κ is a given convergence rate and $(x)_+ = \max(x, 0)$.

125 **Theorem 4.** If (z^*, γ^*) is an equilibrium point of (37)-(38) if and only if z^* is an optimal solution of (34) where γ^* is the corresponding Lagrange multiplier.

Proof. Let (z^*, γ^*) is an equilibrium point of (37)-(38), then $\frac{dz^*}{dt} = 0$ and $\frac{d\gamma^*}{dt} = 0$.

$$\frac{dz^*}{dt} = 0 \Leftrightarrow \nabla f(z^*) + \nabla g(z^*)^T (\gamma^* + g(z^*))_+ = 0, \tag{39}$$

$$\frac{d\gamma^*}{dt} = 0 \Leftrightarrow -\gamma^* + (\gamma^* + g(z^*))_+ = 0 \tag{40}$$

Observe that $\gamma^* = (\gamma^* + g(z^*))_+$ if and only if $\gamma^* \geq 0$, $g(z^*) \leq 0$ and ${\gamma^*}^T g(z^*) = 0$, we obtain then (35) of the KKT system (35)- (36). Furthermore, we replace $(\gamma^* + g(z^*))_+$ by γ^* in the right hand

side of (39) we obtain then $\nabla f(z^*) + \nabla g(z^*)^T \gamma^* = 0$ which is equation (36) of the KKT system (35)-(36). For the converse part of the theorem, it is straightforward that if z^* is an optimal solution of (34) where γ^* is the corresponding Lagrange multiplier, then (z^*, γ^*) is an equilibrium point of (37)-(38). □

Lemma 1. For any initial point $(z(t_0), \gamma(t_0))$, there exists a unique continuous solution $(z(t), \gamma(t))$ for (37)-(38).

Proof. The right-hand side of system (37)-(38) is locally Lipschitz continuous, given that ∇f, ∇g and $(\gamma + g)_+$ are locally Lipschitz continuous. By applying the local existence theorem of ordinary differential equations, we can conclude that there exists a unique continuous solution trajectory $(z(t), \gamma(t))$ for (37)-(38). □

Theorem 5. The neural network proposed in equations (37)-(38) exhibits global stability in the Lyapunov sense. Furthermore, the dynamical network globally converges to a KKT point denoted (z^*, γ^*) where z^* is the optimal solution of the problem (34).

Proof. Let $\zeta = (z, \gamma)$, we define $U(\zeta) = \begin{bmatrix} -\left(\nabla f(z) + \nabla g(z)^T (\gamma + g(z))_+\right) \\ -\gamma + (\gamma + g(z))_+ \end{bmatrix}$.
First, consider the following Lyapunov function

$$E(\zeta) = ||U(\zeta)||^2 + \frac{1}{2}||\zeta - \zeta^*||, \tag{41}$$

where $\zeta^* = (z^*, \gamma^*)$ is an equilibrium point of (37)-(38).
$\frac{dE(\zeta(t))}{dt} = \frac{dU}{dt}^T U + U^T \frac{dU}{dt} + (\zeta - \zeta^*)^T \frac{d\zeta}{dt}$. Observe that $\frac{dU}{dt} = \frac{dU}{d\zeta} \times \frac{d\zeta}{dt} = \nabla U(\zeta) U(\zeta)$. Without loss of generality suppose that there exists $p \in \mathbb{N}$ such that $(\gamma + g(z))_+ = (\gamma_1 + g_1(z)), ..., (\gamma_p + g_p(z)), 0, ..., 0)$, and we define $g^p = (g_1, ..., g_p)$.
We have $\nabla U(\zeta) = \begin{bmatrix} -\left(\nabla^2 f(z) + \sum_{i=1}^p \nabla^2 g^p(z) (\gamma_p + g_p(z)) + \nabla g(z)^T \nabla g(z)\right) & -\nabla g^p(z)^T \\ \nabla g^p(z) & S_p \end{bmatrix}$. where $S_p = \begin{bmatrix} O_{p\times p} & O_{p\times(N-p)} \\ O_{(N-p)\times q} & I_{(N-p)\times(N-p)} \end{bmatrix}$, where N is the length of vector γ.
Since f and g are convex, then the Hessian matrices $\nabla^2 f$ and $\nabla^2 g^p$ are positive semidefinite. Furthermore $\nabla g^T \nabla g$ is positive semidefinite, we conclude that ∇U is negative semidefinite.
Back to the expression of $\frac{dE(\zeta(t))}{dt}$, we have $\frac{dE(\zeta(t))}{dt} = \underbrace{U^T(\nabla U + \nabla U^T)U}_{\leq 0 \text{ since } \nabla U \text{ is negative semidfinite}} + \underbrace{(\zeta - \zeta^*)^T(U(\zeta) - U(\zeta^*))}_{\leq 0 \text{ by Lemma 4 in [19]}} \leq 0$. Then, the neural network (37)-(38) is globally stable in the sense of Lyapunov. Next similarly to the proof of Theorem 5 in [19], we prove that the dynamical neural network (37)-(38) is globally convergent to (z^*, γ^*) where z^* is the optimal solution of (34).

□

4. A two-time scale neurodynamic duplex for ($\mathbf{JCP}^{\boldsymbol{log}}_{\boldsymbol{NS-dep}}$)

($\mathrm{JCP}^{log}_{NS-dep}$) can be written in the following general form

$$\begin{aligned} &\min_{z,y} f(z), \\ &\text{s.t. } g(z,y) \leq 0, \end{aligned} \tag{42}$$

where f is a convex function and g is a biconvex function, $z = (r, \tilde{\lambda}, \tilde{\beta}, \tilde{\pi})^T$, $f(z) = \sum_{i=1}^{I_0} \mu_i^0 \prod_{j=1}^{M} t_j^{a_{ij}^0}$

and $g(z,y) = \begin{pmatrix} K - \epsilon - \prod_{k=1}^{K} y_k \\ -y_1 \\ \vdots \\ -y_K \\ y_1 - 1 \\ \vdots \\ y_K - 1 \\ y_1 \exp(-\tilde{\lambda}_1) + \sum_{i=1}^{I_1} \exp\left\{-\tilde{\lambda}_1 + \tilde{\beta}_i^1 + \log\mu_i^1\right\} - 1 \\ \vdots \\ y_K \exp(-\tilde{\lambda}_K) + \sum_{i=1}^{I_K} \exp\left\{-\tilde{\lambda}_K + \tilde{\beta}_i^K + \log\mu_i^K\right\} - 1 \\ \tilde{\lambda}_1 \\ \vdots \\ \tilde{\lambda}_1 - \tilde{\pi}_1 \\ \vdots \\ \tilde{\lambda}_K - \tilde{\pi}_K \\ \tilde{\pi}_1 + \sum_{j=1}^{M} a_{ij}^1 r_j - \tilde{\beta}_i^1 \leq 0, i = 1, ..., I_1 \\ \vdots \\ \tilde{\pi}_K + \sum_{j=1}^{M} a_{ij}^K r_j - \tilde{\beta}_i^K \leq 0, i = 1, ..., I_K \end{pmatrix}$.

We denote $\mathcal{U} = \{z, y \mid g(z,y) \leq 0\}$ the feasible set of (42).The Lagrangian function of problem (42) is defined as follows:

$$\mathcal{L}(z, y, \omega) = f(z) + \omega^T g(z, y). \tag{43}$$

For any $(z,y) \in \mathcal{U}$, the KKT conditions are stated as follows:

$$\nabla L(z, y, \omega) = 0, \tag{44}$$

$$\omega \geq 0, \omega^T g(z, y) = 0. \tag{45}$$

Definition 1. Let $(z,y) \in \mathcal{U}$, (z,y) is called a partial optimum of (42) if and only if

$$f(z) \leq f(\tilde{z}), \forall \tilde{z} \in \mathcal{U}_y, \tag{46}$$

where $\mathcal{U}_y = \{z \mid g(z,y) \leq 0\}$.

Theorem 6. The KKT system (44)-(45) is equivalent to the following system

$$\nabla f(z) + \nabla_z g(z,y)^T(\omega + g(x,z))_+ = 0 \tag{47}$$

$$\nabla_y g(z,y)^T(\omega + g(z,y))_+ = 0 \tag{48}$$

$$(\omega + g(x,z))_+ - \omega = 0 \tag{49}$$

Proof. The proof of Theorem 6 follows the same lines as the proof of Theorem 4. □

Based on the equations (47)-(49), we consider the following two-time-scale recurrent neural network model

$$\kappa_1 \frac{dz}{dt} = -\left(\nabla f(z) + \nabla_z g(z,y)^T(\omega + g(x,z))_+\right), \tag{50}$$

$$\kappa_2 \frac{dy}{dt} = -\left(\nabla_y g(z,y)^T(\omega + g(z,y))_+\right), \tag{51}$$

$$\kappa_2 \frac{d\omega}{dt} = -\omega + (\lambda + g(z,y))_+, \tag{52}$$

where (z, y, ω) are now time-dependent variables and κ_1 and κ_2 are two time scaling constants with $\kappa_1 \neq \kappa_2$. We propose a duplex of two two-time-scale recurrent neural network (50)-(52) for solving (34) one with $\kappa_1 > \kappa_2$ and the second with $\kappa_1 < \kappa_2$ as shown in Figure 1.

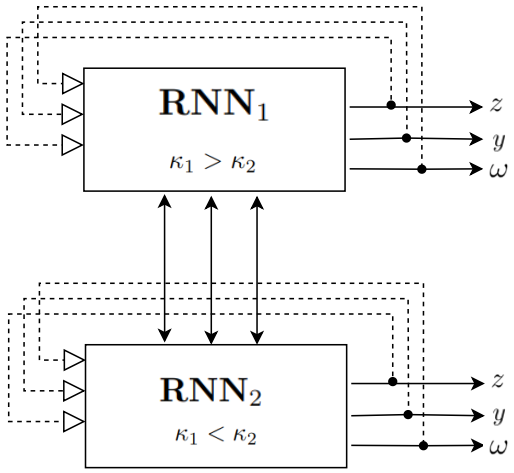

Figure 1: A block diagram depicting a duplex neurodynamic system with a two-timescale configuration

Theorem 7. (z, y, ω) is an equilibrium point of (50)-(52) if and only if (z, y, ω) is a KKT point of (42).

Proof. Let (z, y, ω) is an equilibrium point of (50)-(52). We have then

$$\frac{dz}{dt} = 0 \Leftrightarrow -\left(\nabla f(z) + \nabla_z g(z,y)^T(\omega + g(x,z))_+\right) = 0, \tag{53}$$

$$\frac{dy}{dt} = 0 \Leftrightarrow -\left(\nabla_y g(z,y)^T(\omega + g(z,y))_+\right) = 0, \tag{54}$$

$$\frac{d\omega}{dt} = 0 \Leftrightarrow -\omega + (\lambda + g(z,y))_+ = 0. \tag{55}$$

We obtain system (47)-(49). By Theorem 6, the conclusion follows. The converse part of the Theorem is straightforward. □

The process begins by initializing the state variables of the neurodynamic models. Subsequently, each model undergoes a precise local search based on its dynamics to optimize its performance. Once all neurodynamic models have converged to their equilibria, the initial states of the recurrent neural networks are optimized using the particle swarm optimization (PSO) updating rule. In this context, we represent the position of the i^{th} particle as $\Lambda_i = (\Lambda_{i1}, ..., \Lambda_{in})^T$, and its velocity as $v_i = (v_{i1}, ..., v_{in})^T$. The inertia weight $w \in [0, 1]$ determines the extent to which the particle retains its previous velocity. The best previous position that yielded the maximum fitness value for the i^{th} particle is denoted as $\tilde{\Lambda}_i = (\tilde{\Lambda}_{i1}, ..., \tilde{\Lambda}_{in})^T$, and the best position in the entire swarm that yielded the maximum fitness value is represented by $\hat{\Lambda} = (\hat{\Lambda}_1, ..., \hat{\Lambda}_n)^T$. The initial state of each neurodynamic model is updated using the PSO updating rule, as described in reference [47].

$$v_i(j+1) = wv_i(j) + c_1 r_1(\tilde{\Lambda}_i - \Lambda_i(j)) + c_2 r_2(\hat{\Lambda}_i - \Lambda_i(j)), \tag{56}$$

$$\Lambda_i(j+1) = \Lambda_i(j) + v_i(j+1). \tag{57}$$

where the iterative index is represented by j, while the two weighting parameters are denoted as c_1 and c_2 and r_1 and r_2 represent two random values from the interval $[0, 1]$.

To achieve global convergence, the diversity of initial neuronal states is crucial. One approach to enhance this diversity is by introducing a mutation operator, which generates a random $\Lambda_i(j+1)$. This random generation of $\Lambda_i(j+1)$ helps increase the variation among the initial neuronal states. To measure the diversity of these states, we employ the following function

$$d = \frac{1}{n} \sum_{i=1}^{n} \|\Lambda_i(j+1) - \hat{\Lambda}(j)\|. \tag{58}$$

We utilize the wavelet mutation operator proposed in [48], which is performed for the i-th particle if $d < \zeta$. The mutation operation is carried out as follows

$$\Lambda_i(j+1) = \begin{cases} \Lambda_i(j) + \mu(h_i - \Lambda_i(j)) & , \mu > 0 \\ \Lambda_i(j) + \mu(\Lambda_i(j) - l_i) & , \mu < 0 \end{cases} \tag{59}$$

where h_i and l_i are the upper and the lower bounds for Λ_i, respectively. $\zeta > 0$ is a given threshold and μ is defined using a wavelet function

$$\mu = \frac{1}{\sqrt{a}} e^{-\frac{\phi}{2a}} \cos(5\frac{\phi}{a}) \tag{60}$$

When the value of μ goes to 1, the mutated element of the particle moves towards the maximum value of $\Lambda_i(j+1)$. On the other hand, as μ approaches -1, the mutated element moves towards the minimum value of $x_i(j+1)$. The magnitude of $|\mu|$ determines the size of the search space for $x_i(j+1)$, with larger values indicating a wider search space. Conversely, smaller values of $|\mu|$ result in a smaller search space, allowing for fine-tuning.

To achieve fine-tuning, the dilation parameter a is adjusted based on the current iteration j relative to the total number of iterations T. Specifically, a is set as a function of j/T, with $a = e^{10\frac{j}{T}}$. Additionally, ϕ is randomly generated from the interval $[-2.5a, 2.5a]$.

175 The algorithm details are given in Algorithm 1 where $\Lambda = (z, y, \omega)$

Algorithm 1 The neurodynamic duplex

Initialize

- Let $\Lambda_1(0)$ and $\Lambda_2(0)$ randomly in the feasible region.
- Let the initial best previous position and best position $\tilde{\Lambda}(0) = \hat{\Lambda}(0) = y = \Lambda(0)$.
- Set the convergence error ζ.

while $||\Lambda(j+1) - \Lambda(j)|| \geq \epsilon$ **do**

Compute the equilibrium points $\bar{\Lambda}_1(j)$ and $\bar{\Lambda}_2(j)$ of RNN_1 and RNN_2.

if $f(\bar{z}_1(j)) < f(\tilde{z}(j))$ **then**

$\tilde{\Lambda}(j+1) = \bar{\Lambda}_1(j)$

else

$\tilde{\Lambda}(j+1) = \tilde{\Lambda}(j)$

end if

if $f(\bar{z}_2(j)) < f(\tilde{z}(j))$ **then**

$\tilde{\Lambda}(j+1) = \bar{\Lambda}_2(j)$

else

$\tilde{\Lambda}(j+1) = \tilde{\Lambda}(j)$

end if

if $f(\tilde{z}(j)) < f(\hat{z}(j))$ **then**

$\hat{\Lambda}(j+1) = \tilde{\Lambda}(j+1)$

else

$\hat{\Lambda}(j+1) = \hat{\Lambda}(j)$

end if

Compute the value of $\Lambda(j+1)$ following (56)-(57).

if $d < \zeta$ **then**

Perform the wavelet mutation (59).

end if

j=j+1

end while

Lemma 2. [49] Suppose that the objective function f is measurable, and the feasible region $\mathcal{U}$ is a measurable subset, and for any Borel subset $\mathcal{B}$ of $\mathcal{U}$ with positive Lebesgue measure we have $\prod_{k=1}^{\infty}(1 - \mathbb{P}_k(\mathcal{B})) = 0$. Let $\{y(k)\}_{k=1}^{\infty}$ be a sequence generated by a stochastic optimization algorithm. If $\{y(k)\}_{k=1}^{\infty}$ is a nonincreasing sequence, then it converges in probability to the global optimum set.

Theorem 8. If the state of the neurodynamic model with a single timescale, described by the

following equations

$$\kappa\frac{dz}{dt} = -\left(\nabla f(z) + \nabla_z g(z,y)^T(\omega + g(x,z))_+\right), \tag{61}$$

$$\kappa\frac{dy}{dt} = -\left(\nabla_y g(z,y)^T(\omega + g(z,y))_+\right), \tag{62}$$

$$\kappa\frac{d\omega}{dt} = -\omega + (\lambda + g(z,y))_+, \tag{63}$$

converges to an equilibrium point, then the state of the neurodynamic model with two timescales, as described by equations (50)-(52), globally converges to a partial optimum of problem (42).

Proof. We recall the Lagrangian function of (42)

$$\mathcal{L}(z,y,\omega) = f(z) + \omega^T g(z,y). \tag{64}$$

An equilibrium point (z^*, y^*, ω^*) of (61)-(63) corresponds to a KKT point of (42). We fix y^*, and take $z \in \mathcal{U}_{y^*}$, (42) becomes a convex optimization problem and we have

$$\mathcal{L}(z^*, y^*, \omega^*) \leq \mathcal{L}(z, y^*, \omega^*), \tag{65}$$

which is equivalent to

$$f(z^*) + {\omega^*}^T g(z^*, y^*) \leq f(z) + {\omega^*}^T g(z, y^*). \tag{66}$$

As ${\omega^*}^T g(z, y^*) \leq {\omega^*}^T g(z^*, y^*) = 0$, we have $f(z^*) \leq f(z)$. By Definition 1, (z^*, y^*) is a partial optimum of 42. □

Theorem 9. The duplex of two two-timescale neural networks in Figure 1 is globally convergent to a global optimal solution of problem (34).

Proof. By Theorem 8, the two-timescale neurodynamic models RNN_1 and RNN_2 are proven to converge to a partial optimum. From Algorithm 1, the solution sequence is generated as follows

$$\begin{cases} \hat{\Lambda}(j+1) & = \tilde{\Lambda}(j+1) \text{ if } f(\tilde{z}(j)) < f(\hat{z}(j)), \\ \hat{\Lambda}(j+1) & = \hat{\Lambda}(j) \text{ else.} \end{cases}$$

We observe that the generated solution sequence is monotonically increasing $\{f(\tilde{\Lambda}(j))\}_{j=1}^{\infty}$. Let $\mathcal{M}_{i,j}$ represent the supporting set of the initial state of RNNi at iteration j. According to equation (59), the mutation operation ensures that the initial states of the recurrent neural networks are constrained to the feasible region $\mathcal{U}$. Therefore, for every iteration index $J \geq 1$, the supporting sets satisfy the following condition:

$$\mathcal{U} \subseteq \mathcal{M} = \bigcup_{j=1}^{J}\bigcup_{i=1}^{2}\mathcal{M}_{i,j}. \tag{67}$$

Consequently, we have $v(\mathcal{U}) = v(\mathcal{M}) > 0$.

By Lemma 2, we have

$$\lim_{j->\infty} \mathbb{P}(\hat{\Lambda}(j) \in \Phi) = 1 \tag{68}$$

where Φ is the set of the global optimal solutions of (34). The conclusion follows.

□

5. Numerical experiments

We consider three geometric optimization problems to evaluate the performance of our neurodynamic approaches. All the algorithms in this Section are implemented in Python. We run our algorithms on Intel(R) Core(TM) i7-10610U CPU @ 1.80GHz. The random instances are generated with numpy.random, and we solve the ODE systems with solve_ivp of scipy.integrate. The deterministic equivalent programs are solved with the package gekko and the gradients and partial derivatives are computed with autograd.grad and autograd.jacobian. For the following numerical experiments, we set $\gamma_1^k = 2$, $\gamma_2^k = 2$ and the error tolerance for the neurodynamic duplex $\zeta = 10^{-4}$. In the second subsection, we evaluate the quality of our neurodynamic duplex by comparing the obtained solutions with the ones given by the Convex Alternate Search (CAR) from [50]. The gap between the two solutions is computed as follows GAP $= \frac{\text{Sol}_{\text{CAR}} - \text{Sol}_{\text{Duplex}}}{\text{Sol}_{\text{CAR}}}$, where Sol_{CAR} and $\text{Sol}_{\text{Duplex}}$ are the solutions obtained using the CAR and the neurodynamic duplex, respectively. For the neurodynamical duplex, we take $\frac{\kappa_1}{\kappa_2} = 0.1$ for the first dynmical neural network and $\frac{\kappa_1}{\kappa_2} = 10.0$ for the second one.

5.1. Uncertainty Sets with First Two Order Moments

5.1.1. A three-dimension shape optimization problem

We first consider a transportation problem involving the shifting of grain from a warehouse to a factory. The grain is transported within an open rectangular box, with dimensions of length x_1 meters, width x_2 meters, and height x_3 meters, as illustrated in Figure 2. The objective of the problem is to maximize the volume of the rectangular box, given by the product of its length, width, and height $(x_1x_2x_3)$. However, two constraints must be satisfied. The first constraint relates to the floor area of the box, and the second constraint relates to the wall area. These constraints are necessary to ensure that the shape of the box aligns with the requirements of a given truck. In our analysis, we assume that the wall area A_{wall} and the floor area A_{floor} are random variables. We

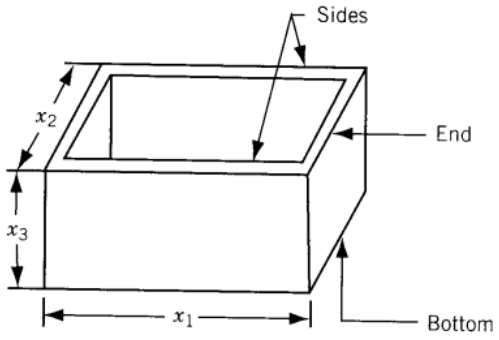

Figure 2: 3D-box shape [51]

formulate our shape optimization problem as follows

Independent case			Dependent case		
Obj value	CPU Time	VS	Obj value	CPU Time	VS
0.296	0.43	0	0.298	0.46	0

Table 1: Results of solving problem (69) when $\mathcal{D} = \mathcal{D}^2$

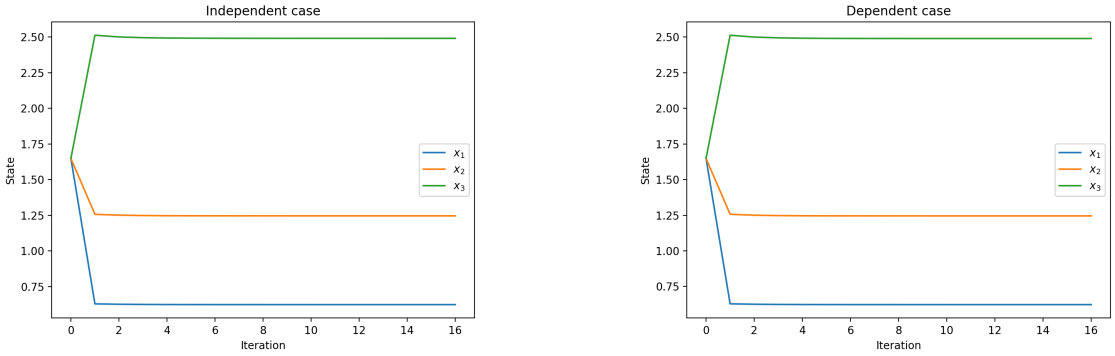

Figure 3: Transient behaviors of the state variables

$$\min_{x\in\mathbb{R}_{++}^3} x_1^{-1}x_2^{-1}x_3^{-1}, \tag{69}$$
$$\text{s.t} \quad \inf_{\mathcal{F}\in\mathcal{D}} \mathbb{P}\left(\frac{1}{A_{wall}}(2x_3x_2 + 2x_1x_3) \leq 1, \frac{1}{A_{floor}}x_1x_2 \leq 1\right) \geq 1-\epsilon.$$

where $\mathcal{F}$ is the joint distribution for $\frac{1}{A_{wall}}$ and $\frac{1}{A_{floor}}$ and $\mathcal{D}$ is the uncertainty set for the probability distribution $\mathcal{F}$. We solve problem (69) when the uncertainty set is equal to $\mathcal{D}^2$ using the dynamical neural network (37)-(38). For the numerical experiments, we take the mean and the covariance describing the uncertainty sets for $\frac{1}{A_{wall}}$ $m_{wall} = 0.05$, $\sigma_{wall} = 0.01$, respectively and for $\frac{1}{A_{floor}}$ $m_{floor} = 0.5$, $\sigma_{floor} = 0.1$, respectively. We recapitulate the obtained results in Table 1. Columns one, two and three give the optimal value, the CPU time and the number of violated scenarios (VS) in the independent case, respectively. Columns four, five and six show the optimal value, the CPU time and the number VS in the dependent case, respectively. The dynamic neural network covers well the risk region in both cases. Figure 3 show the convergence of the state variables.

5.1.2. Multidimensional shape optimization problem

To further assess the performance of our dynamical neural network, we use the multidimensional shape optimization problem with joint chance constraints from [20].

$$\min_{x\in\mathbb{R}_{++}^M} \prod_{i=1}^{m} x_i^{-1},$$
$$\text{s.t} \quad \inf_{\mathcal{F}\in\mathcal{D}} \mathbb{P}_{\mathcal{F}}\left(\sum_{j=1}^{m-1}\left(\frac{m-1}{A_{wall\,j}}x_1 \prod_{i=1,i\neq j}^{m} x_i\right), \frac{1}{A_{floor}}\prod_{j=2}^{m} x_j \leq 1\right) \geq 1-\epsilon, \tag{70}$$
$$\frac{1}{\gamma_{i,j}}x_i x_j^{-1} \leq 1, 1 \leq i \neq j \leq m.$$

In our numerical experiments, we fixed the following parameters $\frac{1}{\gamma_{i,j}} = 0.5$ and $\epsilon = 0.15$. The inverse of floor's area ($\frac{1}{A_{floor}}$) and the inverse of wall area ($\frac{1}{A_{wall_j}}$) for each $j = 1, ..., m$ were considered as random variables. We test the robustness of the different approaches by creating 100 random samples of the variables $\frac{1}{A_{wall_j}}$ and $\frac{1}{A_{floor}}$. We then examine if the solutions meet the constraints of (70) for all 100 cases for the Gaussian distribution, for example. If the solutions are not feasible for a particular case, it is referred to as a violated scenario (VS).

We first solve (70) for $m = 5$ and when the uncertainty set is $\mathcal{D}^2$ in the independent case for different initial points, we observe that the dynamical neural network (37)-(38) converges to the same final value independently from the starting value as shown in Figure 4.

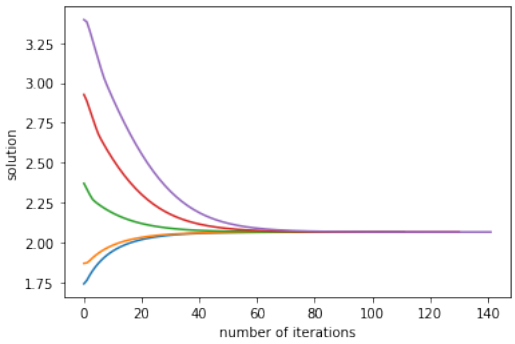

Figure 4: Convergence of the dynamical neural network (37)-(38) for different initial points for (70).

Now we solve (70) for known first-order moments of $\frac{1}{A_{floor}}$ and $\frac{1}{A_{wall_j}}$ for both the dependent and the independent case. We present the obtained results in Table 2. We observe again that the dependent case is more conservative compared to the independent one.

m	Independent case	CPU Time	Dependent case	CPU Time
3	1.03	1.05	1.30	1.39
5	2.09	5.11	2.15	5.20
10	14.79	4.83	5.04	15.10
15	7.76	47.80	7.99	58.04
20	10.68	97.72	10.87	100.91

Table 2: Results for different values of m

5.2. Uncertainty Sets with Known First Order Moment and Nonnegative Support

5.2.1. A generalized shape optimization problem

We solve (70) when the uncertainty set is $\mathcal{D}^3$ for both the independent and the dependent case.

For the numerical experiments, we take $\epsilon = 0.2$. We solve problem (70) using the neurodynamic duplex in the dependent case. We recapitulate the obtained results in Table 3. Column one gives the number of variables m. Columns two, three and four give the objective value, the CPU time and the number of VS in the independent case, respectively. Columns five, six and seven give the objective

m	Independent case			Dependent case		
	Obj value	CPU Time	VS	Obj value	CPU Time	VS
3	0.204	2.28	3	0.491	10.12	0
5	1.03	6.25	2	1.82	98.68	0
10	6.99	15.26	2	9.79	86.35	0
15	18.43	23.84	3	23.45	201.13	0
20	32.09	94.76	5	38.71	744.26	0
30	42.37	100.23	3	51.56	1155.42	0

Table 3: (70) for different values of m for $\mathcal{D} = \mathcal{D}^3$

value, the CPU time and the number of VS for the dependent case, respectively. We observe that the problem with dependent variables is more conservative. Nevertheless, the solution, in this case, covers well the risk area as the number of VS is equal to 0 for all the values of m. Now we additionally solve problem (70) using the assumption that the random variables follow a normal distribution [19] for $m = 5$. In order to compare the solutions obtained with the stochastic and the robust approaches, we evaluate the robustness of the solutions for different hypotheses on the true distribution of the random parameters, i.e., the uniform distribution, the normal distribution, the log-normal distribution, the logistic distribution and Gamma distribution. The obtained results are presented in Table 4 which gives the number of violated scenarios for both the normal solutions and the robust ones and the objective value obtained by each solution. We can infer that the distributionally robust approaches are a conservative approximation of the stochastic programs. We observe that the solutions obtained by the nonnegative support are more conservative compared to the stochastic ones. Notice that the distributionally robust solutions are more robust, i.e., the number of VS when the true distribution is the Logistic distribution is equal to 23 and 19 for the nonnegative support solutions and is equal to 0 for the robust solutions.

		Normal solutions		Robust solutions	
		Independent	Dependent	Independent	Dependent
	Objective Value	0.86	0.99	2.43	4.14
Number	Uniform distribution	22	15	0	0
of	Normal distribution	18	11	1	0
violated	Log-normal distribution	7	4	2	1
scenarios	Logistic distribution	23	19	0	0
	Gamma distribution	16	12	2	2

Table 4: Number of violated scenarios for the stochastic and the robust solutions

5.2.2. Maximizing the worst user signal-to-interference noise ratio

We consider the problem of maximizing the worst user signal-to-interference noise ratio (SINR) for Massive Multiple Input Multiple Output (MaMIMO) systems subject to antenna assignment and multiuser interference constraints taken from [52] and given by

$$\max_{p\in\mathbb{R}^K_{++}} \min_{i\in\mathcal{U}} \frac{p_i|g_i^H g_i|^2}{\sum\limits_{j\in\mathcal{U},j\neq i} p_j|g_i^H g_j|^2+|\sigma_i|^2}, \tag{71}$$

$$\text{s.t} \qquad P_{min} \leq p_i \leq P_{max}, \forall i \in \mathcal{U}, \tag{72}$$

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 σ_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.

Let $a_{ij} = |g_i^H g_j|^2 |g_i^H g_i|^{-2}$ and $b_i = |\sigma_i|^2 |g_i^H g_i|^{-2}$, we derive a geometric reformulation of (71)-(72)

$$\min_{p\in\mathbb{R}^K_{++}, w\in\mathbb{R}_{++}} w^{-1}, \tag{73}$$

$$\text{s.t} \qquad \sum_{j\in\mathcal{U},j\neq i} a_{ij}p_j p_i^{-1} w + b_i p_i^{-1} w \leq 1, \forall i \in \mathcal{U}, \tag{74}$$

$$P_{min} \leq p_i \leq P_{max}, \forall i \in \mathcal{U}. \tag{75}$$

We assume that the coefficients a_{ij} and b_i are independent random variables and we propose the following optimization problem with individual and joint chance constraints

$$\begin{aligned} \min_{p\in\mathbb{R}^K_{++}, w\in\mathbb{R}_{++}} \; & w^{-1}, \\ \text{s.t} \quad & \inf_{\mathcal{F}_i\in\mathcal{D}-i} \mathbb{P}_{\mathcal{F}_i}\left\{\sum_{j\in\mathcal{U},j\neq i} a_{ij}p_j p_i^{-1} w + b_i p_i^{-1} w \leq 1\right\} \geq 1-\epsilon_i, \forall i \in \mathcal{U}, \\ & P_{min} \leq p_i \leq P_{max}, \forall i \in \mathcal{U}. \end{aligned} \tag{POI}$$

and

$$\begin{aligned} \min_{p\in\mathbb{R}^K_{++}, w\in\mathbb{R}_{++}} \; & w^{-1}, \\ \text{s.t} \quad & \inf_{\mathcal{F}\in\mathcal{D}} \mathbb{P}_{\mathcal{F}}\left\{\sum_{j\in\mathcal{U},j\neq i} a_{ij}p_j p_i^{-1} w + b_i p_i^{-1} w \leq 1, \forall i \in \mathcal{U}\right\} \geq 1-\epsilon, \\ & P_{min} \leq p_i \leq P_{max}, \forall i \in \mathcal{U}. \end{aligned} \tag{POJ}$$

We assume that the uncertainty set for the distributionally robust problems (POI) and (POJ) is $\mathcal{D}^3$. We fix $\epsilon = 0.2$. We first solve problem (POJ) for $K = 10$. Figure 5 shows the convergence of the power variables. Next, we solve (POI) and (POJ) for different values of the number of users K. Table 5 presents the obtained results. Column one gives the number of users K. Columns two and three give the optimal value and the number of VS for (POI), respectively. Columns four and five show the optimal value and the number of VS for (POJ), respectively. As observed in the previous section, the use of joint constraints leads to a more conservative minimization problem but covers well the risk area compared to the problem with individual constraints since the number of VS is lower.

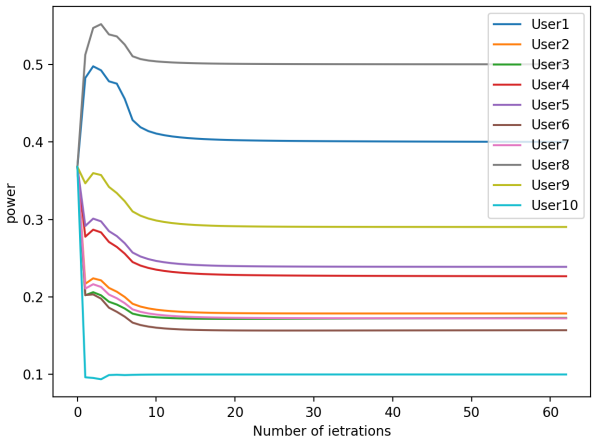

Figure 5: Convergence of the power variables

K	Individual constraints		Joint constraints	
	Obj value	VS	Obj value	VS
5	27.27	5	29.07	0
10	47.36	4	50.23	0
15	66.03	5	68.76	1
20	123.48	3	127.43	0

Table 5: Results for different values of K

6. Conclusion

This paper studies a distributionally robust joint-constrained geometric optimization problem for two different moments-based uncertainty sets. We propose two neurodynamic approaches to solve the resulting optimization problems. To assess the performances of the proposed approaches, we solve a problem of shape optimization and a telecommunication problem. We note that the performances of our approaches can be significantly increased with the development of new ODE solvers mainly based on machine learning techniques.

References

[1] A. Charnes, W. W. Cooper, Chance-constrained programming, Management Science 6 (1) (1959) 73–79.
URL https://EconPapers.repec.org/RePEc:inm:ormnsc:v:6:y:1959:i:1:p:73-79

[2] S. Kandukuri, S. Boyd, Optimal power control in interference-limited fading wireless channels with outage-probability specifications, IEEE Transactions on Wireless Communications 1 (1) (2002) 46–55. doi:10.1109/7693.975444.

[3] M. Hadi, M. R. Pakravan, Resource allocation for elastic optical networks using geometric optimization, J. Opt. Commun. Netw. 9 (10) (2017) 889–899. doi:10.1364/JOCN.9.000889.
URL https://opg.optica.org/jocn/abstract.cfm?URI=jocn-9-10-889

[4] J. G. Posada, A. Vani, M. Schwarzer, S. Lacoste-Julien, Gait: A geometric approach to information theory, in: S. Chiappa, R. Calandra (Eds.), Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, Vol. 108 of Proceedings of Machine Learning Research, PMLR, 2020, pp. 2601–2611.
URL https://proceedings.mlr.press/v108/posada20a.html

[5] A. Muqattash, M. Krunz, T. Shu, Performance enhancement of adaptive orthogonal modulation in wireless cdma systems, IEEE Journal on Selected Areas in Communications 24 (3) (2006) 565–578. doi:10.1109/JSAC.2005.862406.

[6] A. Sonmez, A. Baykasoglu, T. Dereli, I. Filiz, Dynamic optimization of multipass milling operations via geometric programming, International Journal of Machine Tools and Manufacture 39 (2) (1999) 297–320. doi:https://doi.org/10.1016/S0890-6955(98)00027-3.
URL https://www.sciencedirect.com/science/article/pii/S0890695598000273

[7] Y. Kabanov, C. Klüppelberg, A geometric approach to portfolio optimization in models with transaction costs, Finance Stochastics 8 (2) (2004) 207–227. doi:10.1007/s00780-003-0114-3.

[8] M. Zhang, J. Nan, G. Yuan, The geometric portfolio optimization with semivariance in financial engineering, Systems Engineering Procedia 3 (2012) 217–221, information Engineering and Complexity Science - Part I. doi:https://doi.org/10.1016/j.sepro.2011.10.034.
URL https://www.sciencedirect.com/science/article/pii/S2211381911001238

[9] J. Dupacová, Stochastic geometric programming with an application, Kybernetika 46 (2010) 374–386.

[10] J. Singh, Z.-Q. Luo, S. S. Sapatnekar, A geometric programming-based worst case gate sizing method incorporating spatial correlation, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 27 (2) (2008) 295–308. doi:10.1109/TCAD.2007.913391.

[11] V. Kojić, Z. Lukač, Solving profit maximization problem in case of the cobb-douglas production function via weighted ag inequality and geometric programming, in: 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2018, pp. 1900–1903. doi:10.1109/IEEM.2018.8607446.

[12] C.-S. Liu, G. Xu, , L. Wang, An improved geometric programming approach for optimization of biochemical systems, Journal of Applied Mathematics 2014. doi:10.1155/2014/719496.

[13] Y. Han, P. Li, An empirical study of chance-constrained portfolio selection model, Procedia Computer Science 122 (2017) 1189–1195, 5th International Conference on Information Technology and Quantitative Management, ITQM 2017. doi:https://doi.org/10.1016/j.procs.2017.

11.491.
URL https://www.sciencedirect.com/science/article/pii/S1877050917327485

[14] D. Huo, C. Gu, D. Greenwood, Z. Wang, P. Zhao, J. Li, Chance-constrained optimization for integrated local energy systems operation considering correlated wind generation, International Journal of Electrical Power & Energy Systems 132 (2021) 107153. doi:https://doi.org/10.1016/j.ijepes.2021.107153.
URL https://www.sciencedirect.com/science/article/pii/S0142061521003926

[15] A. Dhar, B. Datta, Chance constrained water quality management model for reservoir systems, ISH Journal of Hydraulic Engineering 12 (3) (2006) 39–48. doi:10.1080/09715010.2006.10514848.

[16] N. Sluijk, A. M. Florio, J. Kinable, N. Dellaert, T. Van Woensel, A chance-constrained two-echelon vehicle routing problem with stochastic demands, Transportation Science 0 (0) (0) null. doi:10.1287/trsc.2022.1162.

[17] 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.
URL https://www.sciencedirect.com/science/article/pii/S0167637716300761

[18] R. K. Shiraz, M. Tavana, H. Fukuyama, D. D. Caprio, Fuzzy chance-constrained geometric programming: the possibility, necessity and credibility approaches, Operational Research 17 (1) (2017) 67–97. doi:10.1007/s12351-015-0216-7.
URL https://ideas.repec.org/a/spr/operea/v17y2017i1d10.1007_s12351-015-0216-7.html

[19] S. Tassouli, A. Lisser, A neural network approach to solve geometric programs with joint probabilistic constraints, Mathematics and Computers in Simulation 205 (2023) 765–777. doi:https://doi.org/10.1016/j.matcom.2022.10.025.
URL https://www.sciencedirect.com/science/article/pii/S0378475422004384

[20] J. Liu, A. Lisser, Z. Chen, Distributionally robust chance constrained geometric optimization, Mathematics of Operations Research (2022) 0364–765Xdoi:10.1287/moor.2021.1233.

[21] R. J. Duffin, E. Peterson, C. Zener, Geometric Programming, Wiley, New York, 1967.

[22] L. El Ghaoui, H. Lebret, Robust solutions to least-squares problems with uncertain data, SIAM Journal on Matrix Analysis and Applications 18 (4) (1997) 1035–1064. doi:10.1137/S0895479896298130.

[23] D. Bertsimas, M. Sim, The price of robustness, Operations Research 52 (1) (2004) 35–53.
URL http://www.jstor.org/stable/30036559

[24] D. Bertsimas, D. B. Brown, Constructing uncertainty sets for robust linear optimization, Operations Research 57 (6) (2009) 1483–1495.
URL https://EconPapers.repec.org/RePEc:inm:oropre:v:57:y:2009:i:6:p:1483-1495

[25] W. Wiesemann, D. Kuhn, M. Sim, Distributionally robust convex optimization, Operations Research 62 (6) (2014) 1358–1376. doi:10.1287/opre.2014.1314.

[26] S. Peng, A. Lisser, V. V. Singh, N. Gupta, E. Balachandar, Games with distributionally robust joint chance constraints, Optim. Lett. 15 (6) (2021) 1931–1953. doi:10.1007/s11590-021-01700-9.

[27] J. Cheng, E. Delage, A. Lisser, Distributionally robust stochastic knapsack problem, SIAM Journal on Optimization 24 (3) (2014) 1485–1506. doi:10.1137/130915315.

[28] X. Dou, M. Anitescu, Distributionally robust optimization with correlated data from vector autoregressive processes, Operations Research Letters 47 (4) (2019) 294–299. doi:https://doi.org/10.1016/j.orl.2019.04.005.

[29] X. Li, J. Ke, Robust assortment optimization using worst-case cvar under the multinomial logit model, Operations Research Letters 47 (5) (2019) 452–457. doi:https://doi.org/10.1016/j.orl.2019.07.010.
URL https://www.sciencedirect.com/science/article/pii/S016763771830169X

[30] G. A. Hanasusanto, D. Kuhn, W. Wiesemann, K-adaptability in two-stage distributionally robust binary programming, Operations Research Letters 44 (1) (2016) 6–11. doi:https://doi.org/10.1016/j.orl.2015.10.006.
URL https://www.sciencedirect.com/science/article/pii/S0167637715001376

[31] A. Georghiou, A. Tsoukalas, W. Wiesemann, A primal–dual lifting scheme for two-stage robust optimization, Operations Research 68 (2) (2020) 572–590. doi:10.1287/opre.2019.1873.

[32] Q. Dai, J. Yang, A distributionally robust chance-constrained approach for modeling demand uncertainty in green port-hinterland transportation network optimization, Symmetry 12 (9). doi:10.3390/sym12091492.
URL https://www.mdpi.com/2073-8994/12/9/1492

[33] A. Zare, C. Y. Chung, J. Zhan, S. O. Faried, A distributionally robust chance-constrained milp model for multistage distribution system planning with uncertain renewables and loads, IEEE Transactions on Power Systems 33 (5) (2018) 5248–5262. doi:10.1109/TPWRS.2018.2792938.

[34] R. J. Fonseca, W. Wiesemann, B. Rustem, Robust international portfolio management, Comput. Manag. Sci. 9 (1) (2012) 31–62. doi:10.1007/s10287-011-0132-0.

[35] S. Wang, L. Pang, H. Guo, H. Zhang, Distributionally robust optimization with multivariate second-order stochastic dominance constraints with applications in portfolio optimization, Optimization 0 (0) (2022) 1–24. doi:10.1080/02331934.2022.2048382.

[36] C. Shang, F. You, Distributionally robust optimization for planning and scheduling under uncertainty, Computers & Chemical Engineering 110 (2018) 53–68. doi:https://doi.org/10.1016/j.compchemeng.2017.12.002.
URL https://www.sciencedirect.com/science/article/pii/S009813541730426X

[37] K. Postek, D. den Hertog, B. Melenberg, Computationally tractable counterparts of distributionally robust constraints on risk measures, SIAM Review 58 (4) (2016) 603–650. doi:10.1137/151005221.

[38] G. Hanasusanto, D. Kuhn, S. Wallace, S. Zymler, Distributionally robust multi-item newsvendor problems with multimodal demand distributions, Mathematical Programming 152. doi:10.1007/s10107-014-0776-y.

[39] Y. Zhang, S. Shen, S. A. Erdogan, Distributionally robust appointment scheduling with moment-based ambiguity set, Operations Research Letters 45 (2) (2017) 139–144. doi:https://doi.org/10.1016/j.orl.2017.01.010.
URL https://www.sciencedirect.com/science/article/pii/S0167637717300688

[40] S. Ghosal, W. Wiesemann, The distributionally robust chance-constrained vehicle routing problem, Operations Research 68 (3) (2020) 716–732. doi:10.1287/opre.2019.1924.

[41] C. Ordoudis, V. A. Nguyen, D. Kuhn, P. Pinson, Energy and reserve dispatch with distributionally robust joint chance constraints, Operations Research Letters 49 (3) (2021) 291–299. doi:https://doi.org/10.1016/j.orl.2021.01.012.
URL https://www.sciencedirect.com/science/article/pii/S0167637721000213

[42] D. W. Tank, J. J. Hopfield, Simple 'neural' optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit, 1986.

[43] Y. Xia, J. Wang, A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints, IEEE Transactions on Circuits and Systems I: Regular Papers 51 (7) (2004) 1385–1394. doi:10.1109/TCSI.2004.830694.

[44] J. Wang, A deterministic annealing neural network for convex programming, Neural Networks 7 (4) (1994) 629–641. doi:https://doi.org/10.1016/0893-6080(94)90041-8.
URL https://www.sciencedirect.com/science/article/pii/0893608094900418

[45] A. Nazemi, F. Omidi, An efficient dynamic model for solving the shortest path problem, Transportation Research Part C: Emerging Technologies 26 (2013) 1–19. doi:https://doi.org/10.1016/j.trc.2012.07.005.
URL https://www.sciencedirect.com/science/article/pii/S0968090X12000964

[46] M. Spitmaan, H. Seo, D. Lee, A. Soltani, Multiple timescales of neural dynamics and integration of task-relevant signals across cortex, Proceedings of the National Academy of Sciences 117 (36)

(2020) 22522–22531. doi:10.1073/pnas.2005993117.
URL https://www.pnas.org/doi/abs/10.1073/pnas.2005993117

[47] M. Clerc, J. Kennedy, The particle swarm - explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6 (1) (2002) 58–73. doi:10.1109/4235.985692.

[48] S. H. Ling, H. H. C. Iu, K. Y. Chan, H. K. Lam, B. C. W. Yeung, F. H. Leung, Hybrid particle swarm optimization with wavelet mutation and its industrial applications, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 38 (3) (2008) 743–763. doi:10.1109/TSMCB.2008.921005.

[49] S. Uryasev, P. Pardalos, Stochastic Optimization: Algorithms and Applications, Applied Optimization, Springer US, 2013.
URL https://books.google.fr/books?id=B_fiBwAAQBAJ

[50] P. F. Gorski Jochen, K. Kathrin, Biconvex sets and optimization with biconvex functions: a survey and extensions, Mathematical Methods of Operations Research (2007) 373–467doi:10.1007/s00186-007-0161-1.

[51] Geometric Programming, John Wiley and Sons, Ltd, 2009, Ch. 8, pp. 492–543. doi:https://doi.org/10.1002/9780470549124.ch8.

[52] P. Adasme, A. Lisser, A stochastic geometric programming approach for power allocation in wireless networks, Wireless Networksdoi:10.1007/s11276-023-03295-8.