A Neurodynamic Duplex for Distributionally Robust Joint Chance-Constrained Optimization

Siham Tassouli and Abdel Lisser
Laboratoire des Signaux et Systèmes (L2S), CentraleSupélec, Université Paris Saclay, 3, rue Joliot Curie, 91192 Gif Sur Yvette Cedex, France

Keywords: Dynamical Neural Network, Distributionally Robust Optimization, Joint Chance Constraints, Particle Swarm Optimization, Two-Timescale Systems.

Abstract: This paper introduces a new neurodynamic duplex approach to address distributionally robust joint chance-constrained optimization problems. We assume that the constraints' row vectors are independent, and their probability distributions belong to a specific distributional uncertainty set that is not known beforehand. Within our study, we examine two uncertainty sets for these unknown distributions. Our framework's key feature is the use of a neural network-based method to solve distributionally robust joint chance-constrained optimization problems, achieving an almost sure convergence to the optimum without relying on standard state-of-the-art solving methods. In the numerical section, we apply our proposed approach to solve a profit maximization problem, demonstrating its performance and comparing it against existing state-of-the-art methods.

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 (Charnes and Cooper, 1959). Since then, chance-constrained optimization has been widely studied, and the range of applications is very large. In this paper, we are interested in solving joint chance-constrained 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 (El Ghaoui and Lebret, 1997) use second-order cone programming to solve least-squares problems where the stochastic parameters are not known but bounded. Bertsimas & Sim (Bertsimas and Sim, 2004) introduce a less conservative approach to solve linear optimization problems with uncertain data. Bertsimas & Brown (Bertsimas and Brown, 2009) propose a general scheme for designing uncertainty sets for robust optimization. Wiesemann et al. (Wiesemann et al., 2014) propose standardized ambiguity sets for modeling and solving distributionally robust optimization problems. Peng et al. (Peng et al., 2021) 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. (Cheng et al., 2014) solve a distributionally robust quadratic knapsack problem. Dou & Anitescu (Dou and Anitescu, 2019) propose a new ambiguity set tailored to unimodal and seemingly symmetric distributions to deal with distributionally robust chance constraints. Li & Ke (Li and Ke, 2019) approximate a distributionally robust chance constraint by the worst-case Conditional Value-at-Risk. Hanasusanto et al. (Hanasusanto et al., 2016) approximate two-stage distributionally robust programs with binary recourse decisions. Georghiou et al. (Georghiou et al., 2020) 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 (Dai and Yang, 2020), multi-stage distribution system planning (Zare et al., 2018), portfolio optimization problems (Du et al., 2021), planning and scheduling (Shang and You, 2018), risk measures (Postek et al., 2016), multimodal demand problems (Hanasusanto et al., 2014), appointment scheduling (Zhang et al., 2017), vehicle routine problems (Ghosal and Wiesemann, 2020), risk minimization (Faury et al., 2020) and energy and reserve dispatch (Ordoudis et al., 2021).

Tassouli, S. and Lisser, A.
A Neurodynamic Duplex for Distributionally Robust Joint Chance-Constrained Optimization.
DOI: 10.5220/0012262100003639

In *Proceedings of the 13th International Conference on Operations Research and Enterprise Systems (ICORES 2024)*, pages 15-24
ISBN: 978-989-758-681-1; ISSN: 2184-4372

The neurodynamic system approach represents a significant methodology for addressing optimization problems. By employing artificial recurrent neural networks, optimization problems can be transformed into dynamic systems described by first-order differential equations. These dynamic systems are expected to converge to static states or equilibrium points, which correspond to the solutions of the original optimization problems, starting from given initial points. Moreover, neural networks designed for optimization purposes can be readily implemented in hardware using integrated circuits, allowing for easy deployment. Neural networks offer two compelling advantages when applied to optimization problems: parallel information processing and hardware implementability. Neural networks possess inherent parallel processing capabilities, enabling the simultaneous evaluation of multiple inputs and the computation of the corresponding outputs. This parallelism facilitates efficient and concurrent information processing, leading to faster optimization performance compared to sequential algorithms. Additionally, neural networks can be implemented using specialized hardware, such as integrated circuits or dedicated processing units. This hardware implementation leverages the parallel nature of neural networks, further enhancing computational speed and efficiency. By utilizing hardware resources, neural networks can be deployed in real-time applications or embedded systems, enabling efficient and rapid optimization across various domains. Over the past few decades, recurrent neural networks (RNNs) have received extensive attention for solving optimization problems. A notable early breakthrough in this field was achieved by Hopfield and Tank in 1985 (Hopfield and Tank, 1985), where they introduced a linear programming neural network specifically designed for online optimization applications. Since then, numerous RNN architectures have been proposed to address constrained optimization problems. Xia & Wang (Xia and Wang, 2004) present a recurrent neural network for solving nonlinear convex programming problems subject to nonlinear inequality constraints. Wang (Wang, 1994) proposes a deterministic annealing neural network for convex programming. Nazemi & Omedi (Nazemi and Omidi, 2013) presents a neural network model for solving the shortest path problems. Tassouli & Lisser (Tassouli and Lisser, 2023) propose a recurrent neural network to solve geometric joint chance-constrained optimization problems.

In this paper, we introduce a novel 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 problem, the proposed approach employs two recurrent neural networks that operate collaboratively at two different timescales and converge almost surely to an optimal solution value of the given distributionally robust optimization problem. The main contributions of our work are threefold.

(i) On the formulation side, we reformulate the distributionally robust initial problem as a nonlinear biconvex problem for each uncertainty set. Then, we propose a duplex of two recurrent neural networks to solve the resulting problems. To the best of our knowledge, distributionally robust joint chance-constrained optimization problems have not been addressed using dynamical neural networks.

(ii) On the theoretical side, we show that our neurodynamic duplex converges almost surely to a global optimum of the optimization problem.

(iii) On the numerical side, we show that our neurodynamic duplex gives robust solutions to the initial problem and outperforms the state-of-the-art solving methods.

The rest of the paper is organized as follows. In Section 2, we study two uncertainty sets to solve a distributionally robust chance-constrained optimization problem and give the optimality conditions of the obtained deterministic programs. We propose in Section 3 a duplex of two two-timescale neurodynamic systems that converges to a global optimal solution of the initial problem. We study the convergence analysis in Section 4. Section 5 introduces a profit maximization problem to evaluate the proposed approach.

2 PROBLEM STATEMENT AND OPTIMALITY CONDITIONS

In this paper, we are interested in the optimization problem of the form.

$$\min_{x\in\mathbb{R}^n_+} \sup_{\mathcal{F}_0\in\mathcal{D}_0} \mathbb{E}_{\mathcal{F}_0}\left[\tilde{\zeta_0}^T x\right], \quad (1)$$

$$\text{s.t} \quad \inf_{\mathcal{F}\in\mathcal{D}} \mathbb{P}_{\mathcal{F}}\left(\tilde{\zeta}_k x \leq b_k, k=1,...,K\right) \geq \alpha. \quad (2)$$

where $\tilde{\zeta_0} \in \mathbb{R}^n$ is an uncertain parameter, $[\tilde{\zeta}_1, \tilde{\zeta}_2, ...\tilde{\zeta}_K]^T$ is a $K \times n$ set of pairwise independent random vectors in $\mathbb{R}^n$ and $b \in \mathbb{R}^K$ is a deterministic vector. We consider the case where the probability distribution $\mathcal{F}_0$ of $\tilde{\zeta_0}$ belongs to a certain uncertainty set $\mathcal{D}_0$ and the probability distributions

$\mathcal{F}_k$ of $\tilde{\zeta}_k$, $k = 1,...,K$ are not completely known and belong to $\mathcal{D}_k$. Thus, we take the worst-case where constraints (2) are jointly satisfied for all possible distributions in a given distributional uncertainty set $\mathcal{D}$ with a given probability level α. Based on the pairwise independence between the vectors $(\tilde{\zeta}_k)_{k\in\{1,..,K\}}$, we introduce nonnegative auxiliary variables z_k, $k = 1,...,K$ and rewrite constraint (2) as

$$\inf_{\mathcal{F}_k\in\mathcal{D}_k} \mathbb{P}_{\mathcal{F}_k}\left(\tilde{\zeta}_k x \le b_k\right) \ge \alpha^{z_k}, k = 1,...,K \quad (3)$$

$$\sum_{k=1}^{K} z_k = 1, \quad (4)$$

$$z_k \ge 0, k = 1,...,K. \quad (5)$$

In this section, we propose two uncertainty sets to solve (1)-(2) using two moments-based uncertainty sets to define $\mathcal{D}_k$, $k = 1,...,K$.

We first assume that we know the mean vector μ_k and the covariance matrix Σ_k of $\tilde{\zeta}_k^T$. We define for every $k = 0,1,...,K$

$$\mathcal{D}_k^1(\mu_k,\Sigma_k) = \left\{ \mathcal{F}_k \;\middle|\; \begin{array}{r} \mathbb{E}[\tilde{\zeta}_k^T] = \mu_k \\ \mathbb{E}[(\tilde{\zeta}_k^T-\mu_k)(\tilde{\zeta}_k^T-\mu_k)^T] = \Sigma_k \end{array} \right\},$$

where $\mathcal{F}_k$ is a probability distribution of $\tilde{\zeta}_k^T$. In this case, we have the following deterministic reformulation for the distributionally robust joint chance constraint (2) in (Cheng et al., 2014).

$$\mu_k^T x + \sqrt{\frac{\alpha^{z_k}}{1-\alpha^{z_k}}}||\Sigma_k^{\frac{1}{2}} x|| \le b_k, k = 1,...,K \quad (6)$$

$$\sum_{k=1}^{K} z_k = 1, \quad (7)$$

$$z_k \ge 0, k = 1,..,K. \quad (8)$$

We obtain the following deterministic equivalent problem for (1)-(2).

$$\min \mu_0^T x, \quad (9)$$

$$\text{s.t. } \mu_k^T x + \sqrt{\frac{\alpha^{z_k}}{1-\alpha^{z_k}}}||\Sigma_k^{\frac{1}{2}} x|| \le b_k, k = 1,...,K \quad (10)$$

$$\sum_{k=1}^{K} z_k = 1, x \ge 0, \quad (11)$$

$$z_k \ge 0, k = 1,..,K. \quad (12)$$

Lemma 1. The function $z \mapsto \sqrt{\frac{\alpha^z}{1-\alpha^z}}$, with $0 < \alpha < 1$ is convex $\forall z > 0$.

Proof. Let $z > 0$ and $0 < \alpha < 1$, we have $\sqrt{\frac{\alpha^z}{1-\alpha^z}} = \exp\left\{\frac{1}{2}(z\log(\alpha) - \log(1-\alpha^z))\right\}$. We have $z \mapsto \alpha^z$ is a convex function and the function $z \mapsto \log(1-z)$ is non-increasing and concave, there follows that $z \mapsto \log(1-\alpha^z)$ is concave. We have that $z \mapsto \frac{1}{2}(z\log(\alpha) - \log(1-\alpha^z))$ is convex as an addition of two convex functions. Furthermore, $z \mapsto e^z$ is a non-increasing convex function. Then we conclude that $z \mapsto \exp\left\{\frac{1}{2}(z\log(\alpha) - \log(1-\alpha^z))\right\}$ is convex. □

Corollary 2. Problem (9)-(12) is biconvex on (x,z)

Now we consider that the mean of $\tilde{\zeta}_k$ lies in an ellipsoid of size $\gamma_{k1} \ge 0$ with center μ_k and that the covariance matrix of $\tilde{\zeta}_k$ lies in a positive semidefinite cone of center Σ_k. We define for every $k = 0,1,...,K$, $\mathcal{D}_k^2(\mu_k,\Sigma_k) = \left\{ \mathcal{F}_k \;\middle|\; \begin{array}{r} (\mathbb{E}_{\mathcal{F}_k}[\tilde{\zeta}_k^T]-\mu_k)^T\Sigma_k^{-1}(\mathbb{E}_{\mathcal{F}_k}[\tilde{\zeta}_k^T]-\mu_k) \le \gamma_{k1} \\ \mathrm{COV}_{\mathcal{F}_k}(\tilde{\zeta}_k^T) \preceq \gamma_{k2}\Sigma_k \end{array} \right\}$, where $\gamma_{k2} \ge 0$ and $\mathrm{COV}_{\mathcal{F}_k}$ is a covariance operator under probability distribution $\mathcal{F}_k$. The deterministic reformulation for the distributionally robust joint chance constraint (2) in this case is given in (Peng et al., 2021) as follows.

$$\mu_k^T x + (\sqrt{\frac{\alpha^{z_k}}{1-\alpha^{z_k}}}\sqrt{\gamma_{k2}} + \sqrt{\gamma_{k1}})||\Sigma_k^{\frac{1}{2}} x|| \le b_k, \quad (13)$$

$$k = 1,...,K, \quad (14)$$

$$\sum_{k=1}^{K} z_k = 1, \quad (15)$$

$$z_k \ge 0, k = 1,..,K. \quad (16)$$

We can formulate the objective function as (Liu et al., 2022)

$$\min_{x\in\mathbb{R}^n_+} \mu_0^T x + \sqrt{\gamma_{01}}||\Sigma_0^{\frac{1}{2}} x||. \quad (17)$$

The constraints set (14) is biconvex and the objective function (17) is convex.

To study the optimality conditions of the robust joint chance-constrained problem. We give the equivalent deterministic problem for each uncertainty set in a general form as follows.

$$\min f(x), \quad (18)$$

$$\text{s.t. } g_k(x,z) \le 0, k = 1,...,K, \quad (19)$$

$$h(z) \le 0, \quad (20)$$

$$l(x) \le 0, \quad (21)$$

where,

$$f(x) = \begin{cases} \mu_0^T x, \text{ if } \mathcal{D}_k = \mathcal{D}_k^1 \\ \mu_0^T x + \sqrt{\gamma_{01}}||\Sigma_0^{\frac{1}{2}} x||, \text{ if } \mathcal{D}_k = \mathcal{D}_k^2 \end{cases}$$

$h(z) = \left(\sum_{k=1}^{K} z_k - 1, 1 - \sum_{k=1}^{K} z_k, -z_1, -z_2, ..., z_K\right)^T$, $l(x) = -x$ and

$$g_k(x,z) = \begin{cases} \mu_k^T x + \sqrt{\frac{\alpha^{z_k}}{1-\alpha^{z_k}}}||\Sigma_k^{\frac{1}{2}} x|| - b_k, \text{ if } \mathcal{D}_k = \mathcal{D}_k^1 \\ \mu_k^T x + (\sqrt{\frac{\alpha^{z_k}}{1-\alpha^{z_k}}}\sqrt{\gamma_{k2}} + \sqrt{\gamma_{k1}})||\Sigma_k^{\frac{1}{2}} x|| - b_k, \\ \text{if } \mathcal{D}_k = \mathcal{D}_k^2 \end{cases}$$

Definition 1. Let $\mathcal{U}$ the feasible set of (18)-(21), we define $\mathcal{U}_x = \{z \mid g_k(x,z) \leq 0, h(z) \leq 0, k = 1,...,K\}$ and $\mathcal{U}_z = \{x \mid g_k(x,z) \leq 0, l(x) \leq 0, k = 1,...,K\}$. (x^*,z^*) is a partial optimum of (18)-(21) if $f(x^*) \leq f(x), \forall x \in \mathcal{U}_{z^*}$.

Definition 2. Let $\mathcal{U}$ the feasible set of (18)-(21) and $(x^*,z^*) \in \mathcal{U}$. If there exists $\beta^{(1)}$, $\beta^{(2)}$, γ and λ such that (x^*,z^*) verifies

$$\nabla_x f(x) + {\beta^{(1)}}^T \nabla_x g(x,z) + \lambda^T \nabla_x l(x) = 0, \tag{22}$$

$$\lambda \geq 0, \lambda^T l(x) = 0, \beta^{(1)} \geq 0, {\beta^{(1)}}^T g(x,z) = 0, \tag{23}$$

$${\beta^{(2)}}^T \nabla_z g(x,z) + \gamma^T \nabla_z h(z) = 0, \tag{24}$$

$$\beta^{(2)} \geq 0, {\beta^{(2)}}^T g(x,z) = 0, \gamma \geq 0, \gamma^T h(z) = 0, \tag{25}$$

where $g(x,z) = (g_1(x,z),...,g_K(x,z))$ and $(x)_+ = \max(0,x)$. Then (x^*,z^*) is called a partial KKT point of (18)-(21).

Theorem 3. The partial KKT system (22)-(25) is equivalent to the following system

$$\begin{aligned}
&\nabla_x f(x) + \nabla_x g(x,z)^T (\beta^{(1)} + g(x,z))_+ + \nabla_x l(x)(\lambda + l(x))_+ = 0\\
&\nabla_z g(x,z)^T (\beta^{(2)} + g(x,z))_+ + \nabla_z h(z)^T (\gamma + h(z))_+ = 0\\
&(\beta^{(1)} + g(x,z))_+ - \beta^{(1)} = 0\\
&(\beta^{(2)} + g(x,z))_+ - \beta^{(2)} = 0\\
&(\lambda + l(x))_+ - \lambda = 0\\
&(\gamma + h(z))_+ - \gamma = 0
\end{aligned} \tag{26}$$

Proof. By $(\beta^{(1)} + g(x,z))_+ = \beta^{(1)}$ and $(\lambda + l(x))_+ = \lambda$, we have

$$\begin{aligned}
&\Big(\nabla_x f(x) + \nabla_x g(x,z)^T (\beta^{(1)} + g(x,z))_+ +\\
&\qquad \nabla_x l(x)(\lambda + l(x))_+ = 0\Big)\\
&\qquad\qquad \Leftrightarrow\\
&\Big(\nabla_x f(x) + {\beta^{(1)}}^T \nabla_x g(x,z) + \lambda^T \nabla_x l(x) = 0\Big).
\end{aligned}$$

We obtain the equation (25) of the partial KKT system.

Furthermore, observe that

- $(\beta^{(1)} + g(x,z))_+ - \beta^{(1)} = 0$ if and only if $\beta^{(1)} \geq 0$, $g(x,z) \leq 0$ and ${\beta^{(1)}}^T g(x,z) = 0$,
- $(\lambda + l(x))_+ - \lambda = 0$ if and only if $\lambda \geq 0$, $l(x) \leq 0$ and $\lambda^T l(x) = 0$,

which leads to the equation (23) of the partial KKT system. We obtain the remaining equations following the same lines. The converse part of the theorem is straightforward.

□

Definition 3. Let (x^*,z^*) a feasible point of (18)-(21). If $x^* \geq 0$, $h(z^*) \leq 0$ and there exists $(\bar{x},\bar{z})$ such that $g(x^*,\bar{z}) \leq 0$ and $g(\bar{x},z^*) \leq 0$, then (18)-(21) satisfies partial Slater constraint qualification at (x^*,z^*).

The following theorem gives the optimality conditions of problem (18)-(21).

Theorem 4. If partial Slater constraint qualification hold for (18)-(21) at (x^*,z^*), then (x^*,z^*) is a partial optimum of (18)-(21) if and only if (x^*,z^*) is a partial KKT point of (18)-(21). Furthermore, if $\beta^{(1)} = \beta^{(2)}$ then (x^*,z^*) is a KKT point of (18)-(21).

Remark 5. The proof of Theorem 4 follows the lines of Theorem 1 in (Shen et al., 2020).

3 A NEURODYNAMIC DUPLEX

Based on the system (26), we propose a duplex of two two-time-scale recurrent neural network models for solving (18)-(21). Every recurrent neural network of the duplex is driven by the following ODE system.

$$\begin{aligned}
\kappa_1 \frac{dx}{dt} = &-(\nabla_x f(x) + \nabla_x g(x,z)^T (\beta + g(x,z))_+\\
&+ \nabla_x l(x)(\lambda + l(x))_+),
\end{aligned} \tag{27}$$

$$\begin{aligned}
\kappa_2 \frac{dz}{dt} = &-(\nabla_z g(x,z)^T (\beta + g(x,z))_+\\
&+ \nabla_z h(z)^T (\gamma + h(z))_+),
\end{aligned} \tag{28}$$

$$\kappa_2 \frac{d\beta}{dt} = -\beta + (\beta + g(x,z))_+, \tag{29}$$

$$\kappa_2 \frac{d\lambda}{dt} = -\lambda + (\lambda + l(x))_+. \tag{30}$$

$$\kappa_2 \frac{d\gamma}{dt} = -\gamma + (\gamma + h(z))_+. \tag{31}$$

where $(x,z,\beta,\gamma,\lambda)$ are now time-dependent variables and κ_1 and κ_2 are two time scaling constants with $\kappa_1 \neq \kappa_2$. We consider a duplex of two two-time-scale recurrent neural network (27)-(31) RNN_1 and RNN_2 for solving (18)-(21) one with $\kappa_1 > \kappa_2$ and the second with $\kappa_1 < \kappa_2$ as shown in Figure 1. The zoom on RNN_1 shows the circuit implementation of a single two-timescale recurrent neural network (27)-(31).

Theorem 6. $(x^*,z^*,\beta^*,\gamma^*,\lambda^*)$ is an equilibrium point of (27)-(31) if and only if (x^*,z^*) is a KKT point of (18)-(21) and β^*,γ^* and λ^* are the associated Lagrange variables.

Proof. $(x^*,z^*,\beta^*,\gamma^*,\lambda^*)$ is an equilibrium point of (27)-(31) if and only if $\frac{dx}{dt} = 0$, $\frac{dz}{dt} = 0$, $\frac{d\beta}{dt} = 0$, $\frac{d\lambda}{dt} = 0$ and $\frac{d\gamma}{dt} = 0$, we obtain system (26). By Theorems 3 and 4 the conclusion follows. □

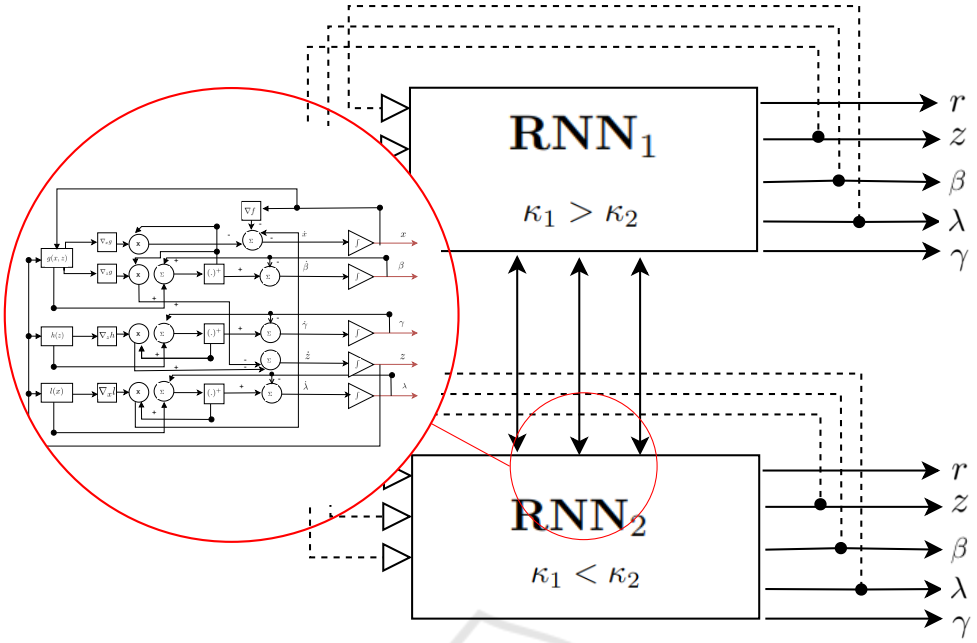

Figure 1: A block diagram depicting a duplex neurodynamic system with a two-timescale configuration.

We describe the working process of the neurodynamic duplex as follows:
First, the state variables of the neurodynamic models are initialized. Then, each model undergoes a precise local search based on its dynamics for the optimization process. Once all neurodynamic models have converged to their equilibria, the initial states of the recurrent neural networks are optimized using the updating rule of particle swarm optimization (PSO). We denote $y_i = (y_{i1}, ..., y_{in})^T$ the position of the i^{th} particle and $v_i = (v_{i1}, ..., v_{in})^T$ its velocity. The inertia weight $w \in [0,1]$ determines the degree to which the particle's previous velocity is retained. The best previous position yielding the maximum fitness value for the i^{th} particle is denoted as $\tilde{y}_i = (\tilde{y}_{i1}, ..., \tilde{y}_{in})^T$, and the best position yielding the maximum fitness value in the swarm is represented by $\hat{y} = (\hat{y}_1, ..., \hat{y}_n)^T$. The initial state of each neurodynamic model is updated using the PSO updating rule given by (Clerc and Kennedy, 2002), i.e,

$$v_i(j+1) = wv_i(j) + c_1 r_1(\tilde{y}_i - y_i(j)) + c_2 r_2(\hat{y}_i - y_i(j)), \tag{32}$$

$$y_i(j+1) = y_i(j) + v_i(j+1). \tag{33}$$

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

The diversity of initial neuronal states plays a critical role in achieving global optimization. Introducing a mutation operator, which generates a random $y_i(j+1)$, can enhance the diversity of initial neuronal states. To quantify the diversity of these states, we use the following function

$$d = \frac{1}{n}\sum_{i=1}^{n} \|y_i(j+1) - \hat{y}(j)\|. \tag{34}$$

We use wavelet mutation operator from (Ling et al., 2008) and performing for the i^{th} particle if $d < \zeta$ as follows

$$y_i(j+1) = \begin{cases} y_i(j) + \mu(h_i - y_i(j)) & , \rho > 0 \\ y_i(j) + \mu(y_i(j) - l_i) & , \rho < 0 \end{cases} \tag{35}$$

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

$$\rho = \frac{1}{\sqrt{a}} e^{-\frac{\phi}{2a}} \cos(5\frac{\phi}{a}) \tag{36}$$

As the value of ρ goes to 1, the mutated element of the particle will move towards the maximum value of $y_i(j+1)$, whereas is close to -1, the mutated element goes towards the minimum value of $x_i(j+1)$. The magnitude of $|\rho|$ determines the size of the search space for $x_i(j+1)$, with larger values indicating a wider search space. Conversely, smaller values of $|\mu|$ lead to a smaller search space for fine-tuning. To achieve fine-tuning, the value of the dilation parameter a is adjusted based on the current number of iterations j relative to the total number of iterations T. Specifically, a is a function of j/T, we take $a = e^{10\frac{j}{T}}$. We generate ϕ randomly from $[-2.5a, 2.5a]$.

The algorithm details are given in Algorithm 1 where $y = (x, z, \beta, \gamma)$.

Algorithm 1: The neurodynamic duplex.

- Let $y_1(0)$ and $y_2(0)$ be randomly generated in the feasible region.
- Let $\tilde{y}(0) = \hat{y}(0) = y(0)$ the initial best previous position and best position, respectively.
-Set the convergence error ε.
while $||y(j+1) - y(j)|| \geq \varepsilon$ **do**
  Compute the equilibrium points $\bar{y}_1(j)$ and $\bar{y}_2(j)$ of RNN_1 and RNN_2 based on (27)-(31).
  if $f(\bar{x}_1(j)) < f(\tilde{x}(j))$ **then**
    $\tilde{y}(j+1) = \bar{y}_1(j)$
  else
    $\tilde{y}(j+1) = \tilde{y}(j)$
  end if
  if $f(\bar{x}_2(j)) < f(\tilde{x}(j))$ **then**
    $\tilde{y}(j+1) = \bar{y}_2(j)$
  else
    $\tilde{y}(j+1) = \tilde{y}(j)$
  end if
  if $f(\tilde{x}(j)) < f(\hat{x}(j))$ **then**
    $\hat{y}(j+1) = \tilde{y}(j+1)$
  else
    $\hat{y}(j+1) = \hat{y}(j)$
  end if
  Compute the value of $y(j+1)$ following (32)-(33).
  if $d < \zeta$ **then**
    Perform the wavelet mutation (35).
  end if
  j=j+1
end while

4 CONVERGENCE ANALYSIS

Lemma 7. (Uryasev and Pardalos, 2013) 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 $\{f(y(k))\}_{k=1}^{\infty}$ is a nonincreasing sequence, then it converges in probability to the set of global optimal solutions.

Theorem 8. If the state of the following neurodynamic model with a single timescale

$$\begin{aligned}
\kappa\tfrac{dx}{dt} &= -\left(\nabla_x f(x) + \nabla_x g(x,z)^T(\beta + g(x,z))_+ + \nabla_x l(x)(\lambda + l(x))_+\right),\\
\kappa\tfrac{dz}{dt} &= -\left(\nabla_z g(x,z)^T(\beta + g(x,z))_+ + \nabla_z h(z)^T(\gamma + h(z))_+\right),\\
\kappa\tfrac{d\beta}{dt} &= -\beta + (\beta + g(x,z))_+,\\
\kappa\tfrac{d\lambda}{dt} &= -\lambda + (\lambda + l(x))_+.\\
\kappa\tfrac{d\gamma}{dt} &= -\gamma + (\gamma + h(z))_+.
\end{aligned} \tag{37}$$

converges to an equilibrium point, then the state of a neurodynamic model with two timescales, as described by equations (27)-(31), globally converges to a partial optimum of problem (18)-(21).

Proof. The Lagrangian function of problem (18)-(21) is given by

$$\mathcal{L}(x, z, \beta, \lambda, \gamma) = f(x) + \beta^T g(x,z) + \gamma^T l(x) + \lambda^T h(z).$$

As an equilibrium point of (37) corresponds to a KKT point $(x^*, z^*, \beta^*, \lambda^*, \gamma^*)$ of (18)-(21) (Xia and Wang, 2004) verifying

$$\nabla_x \mathcal{L}(x^*, z^*, \beta^*, \lambda^*, \gamma^*) = 0, \tag{38}$$
$$\nabla_z \mathcal{L}(x^*, z^*, \beta^*, \lambda^*, \gamma^*) = 0, \tag{39}$$
$$\nabla_\beta \mathcal{L}(x^*, z^*, \beta^*, \lambda^*, \gamma^*) = 0, \tag{40}$$
$$\nabla_\gamma \mathcal{L}(x^*, z^*, \beta^*, \lambda^*, \gamma^*) = 0, \tag{41}$$
$$\nabla_\lambda \mathcal{L}(x^*, z^*, \beta^*, \lambda^*, \gamma^*) = 0, \tag{42}$$
$$\beta^{*T} g(x^*, z^*) = 0, \beta^* \geq 0, \tag{43}$$
$$\gamma^{*T} l(x^*) = 0, \gamma^* \geq 0. \tag{44}$$
$$\lambda^{*T} h(z^*) = 0, \lambda^* \geq 0. \tag{45}$$

We fix x^* and take $z \in \mathcal{U}_x^*$, problem (18)-(21) becomes convex, we have

$$\mathcal{L}(x^*, z^*, \beta^*, \lambda^*, \gamma^*) \leq \mathcal{L}(x^*, z, \beta^*, \lambda^*, \gamma^*),$$

which leads to

$$f(x^*) + \beta^{*T} g(x^*, z^*) + \lambda^{*T} h(z^*) \leq f(x^*) + \beta^{*T} g(x^*, z) + \lambda^{*T} h(z).$$

As $\lambda^{*T} l(x) \leq \lambda^{*T} l(x^*) = 0$, $\gamma^{*T} h(z) \leq \gamma^{*T} h(z^*) = 0$ and $\beta^{*T} g(x^*, z) \leq \beta^{*T} g(x^*, z^*) = 0$ from the KKT conditions, then $f(x^*) \leq f(x)$ and this for every $z \in \mathcal{U}_x^*$. By Definition 2, we have that x^* is a partial optimum of (18)-(21). □

Theorem 9. If the state of the two-timescale neurodynamic model (27)-(31) converges to a partial optimum, the initial states and time constants of the two neurodynamic models are different. Then, the duplex of two two-timescale neural networks in Figure 1 system is globally convergent to a global optimal solution of problem (18)-(21).

Proof. By Theorem 4, 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

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

We observe that the generated solution sequence $\{f(\hat{y}(j))\}_{j=1}^{\infty}$ is monotonically increasing.

Let $\mathcal{M}_{i,j}$ be the supporting set of the initial state of RNN$_i$ at iteration j. As indicated by equation (35), the mutation operation ensures that the initial states of the RNNs are forced to be in the feasible region $\mathcal{U}$. Hence, for every iteration index $J \geq 1$, the supporting sets fulfill the following condition

$$\mathcal{U} \subseteq \mathcal{M} = \bigcup_{j=1}^{J}\bigcup_{i=1}^{2}\mathcal{M}_{i,j}. \tag{46}$$

We have $v(\mathcal{U}) = v(\mathcal{M}) > 0$. By Lemma 7, we have

$$\lim_{j->\infty} \mathbb{P}(\hat{y}(j) \in \Phi) = 1 \tag{47}$$

where Φ is the set of the global optimal solutions of (18)-(21). The conclusion follows. □

5 NUMERICAL EXPERIMENTS

To evaluate the performance of our approach, we consider a standard profit maximization problem. A manufacturing firm produces n products with N different machines. The times required to manufacture each unit are random variables. The mean vector μ_j and the covariance matrix Σ_j describing the uncertainty sets of the time vector $t_j = \{t_{ij}\}_{1\leq i\leq n}$, where t_{ij} is the time required to manufacture one unit of each of product i using machine j and the daily capacity of each machine j given by b_j are given. The objective of the study is to determine the daily number of units to be manufactured for each product without exceeding the available machining times. We write our robust joint chance-constrained maximization problem as follows.

$$\min_{x\geq 0} \sup_{\mathcal{F}_0\in\mathcal{D}_0} -\mathbb{E}[\tilde{c}^T x], \tag{48}$$

$$\text{s.t. } \inf_{\mathcal{F}\in\mathcal{D}} \mathbb{P}\left(\sum_{i=1}^{n} t_{ij}x_i \leq b_j, j = 1,...,\text{N}\right) \geq p, \tag{49}$$

where vector $\tilde{c}$ is a random variable and corresponds to the profit per unit for each product, t_{ij} is the time required to manufacture one unit of product i using machine j, b_j is the time capacity of machine j, p is a given probability level, $\mathcal{D}_0$ is an uncertainty set for the distribution $\mathcal{F}_0$ of $\tilde{c}$ and $\mathcal{D}$ is an uncertainty set for the distribution $\mathcal{F}$ of the random variables.

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, the values of μ_j and $\bar{c}$ the mean of $\tilde{c}$ are uniformly generated in $[2.0, 4.0]$, the components of the matrix Σ_j are uniformly drawn in the interval $[1.0, 3.0]$ and we generate the values of b_j uniformly in $[50.0, 60.0]$, $\gamma_{k1} = 5$ and $\gamma_{k2} = 5$.

The resulting deterministic equivalent problems of (48)-(49), where the uncertainty sets are $\mathcal{D}^1$ and $\mathcal{D}^2$ are given respectively by

$$\begin{aligned}&\min -\bar{c}^T x,\\ &\text{s.t. } \mu_j^T x + \sqrt{\frac{p^{z_j}}{1-p^{z_j}}}||\Sigma_j^{\frac{1}{2}} x|| \leq b_j, j = 1,...,N,\\ &\sum_{j=1}^{N} z_j = 1,\\ &x \geq 0, z_j \geq 0, j = 1,...,N,\end{aligned}$$

and

$$\begin{aligned}&\min -\bar{c}^T x,\\ &\text{s.t. } \mu_j^T x + \left(\sqrt{\frac{p^{z_j}}{1-p^{z_j}}}\sqrt{\gamma_{k2}} + \sqrt{\gamma_{k1}}\right)||\Sigma_j^{\frac{1}{2}} x|| \leq b_j, j = 1,...,N,\\ &\sum_{j=1}^{N} z_j = 1,\\ &x \geq 0, z_j \geq 0, j = 1,...,N,\end{aligned}$$

5.1 The Neurodynamic Duplex vs. Convex Approximations

Cheng et al. (Cheng et al., 2014) propose two convex approximations to solve problem (48)-(49). A linear approximation that gives an upper bound to the minimization problem and a tangent approximations that leads to a lower bound. In this first subsection, we compare the objective value obtained using the neurodynamic duplex with those obtained using the linear and the tangent approximations. We compute the gap between the two bounds and the global optimum given by the neurodynamic duplex by GAP = $\frac{\text{Bound}_{\text{lower, upper}} - \text{ND}}{\text{Bound}_{\text{lower, upper}}}$, where Bound$_{\text{lower}}$ is the value of the lower bound, Bound$_{\text{upper}}$ is the value of the upper bound, and ND is the value obtained using the neurodynamic duplex. We recapitulate the obtained results in Table 1. Column one gives the value of the confidence parameter p. Column two gives the final value of the neurodynamic duplex. Columns three and four

Table 1: Results for different values of p for $\mathcal{D}^1(\mu,\Sigma)$.

p	Neurodynamic duplex	Tangent approximation		Linear approximation	
	Obj value	Obj value	GAP	Obj value	GAP
0.95	-36.25	-36.41	0.43%	-36.20	-0.13%
0.9	-40.48	-40.51	0.07%	-40.46	-0.04%
0.8	-45.30	-45.41	0.24%	-45.22	-0.17%
0.7	-47.31	-47.38	0.14%	-47.28	-0.06%
0.6	-48.09	-48.13	0.08%	-48.07	-0.06%

Table 2: Results for different values of n and N for $\mathcal{D}^1(\mu,\Sigma)$.

n	N	Neurodynamic duplex	Tangent approximation		Linear approximation	
		Obj value	Obj value	GAP	Obj value	GAP
7	4	-22.86	-22.97	0.47%	-22.77	-0.39%
10	5	-22.51	-22.65	0.61%	-22.44	-0.31%
15	10	-21.36	-21.61	1.15%	-20.82	-2.59%
20	15	-21.28	-21.78	2.29%	-20.93	-1.67%
25	20	-19.79	-20.78	4.67%	-19.01	-4.10%

show the lower bound and its gap with the neurodynamic duplex, respectively. Finally, columns five and six present the upper bound and the gap with the neurodynamic approach. We observe that the final value obtained with the dynamical duplex remains between the two bounds for the different values of p with gaps less than 0.5%, demonstrating that the neurodynamic approach effectively converges to the global optimum. Moreover, we remark that as p increases, the value of the objective function increases which is coherent since lower values of p induce larger risk area.

5.2 The Distributionally Robust Optimization Approach vs. Stochastic Optimization Approaches

To evaluate the robustness of the proposed duplex for the two uncertainty sets $\mathcal{D}^1$ and $\mathcal{D}^2$, we additionally solve problem (48)-(49) when the random variables follow uniform and normal distributions and $p = 0.95$. We compare the solution of our proposed distributionally robust approach with the solution of the stochastic programming approach. We generate 100 instances for $(t_{ij})_{1\leq I\leq n, 1\leq j\leq N}$ using the mean vectors and the covariance matrix when the true distribution of the stochastic variables is one of the five following distributions: uniform distribution, normal distribution, log-normal distribution, logistic distribution and Gamma distribution. We calculate the number of times when the constraints were violated over the 100 generated scenarios for each stochastic and robust solutions. Table 3 recapitulates the obtained results, where column one gives the true distribution, columns two, three, four and five give the number of violated scenarios for the solution obtained using the uniform approach, the normal approach, the first robust approach and the second robust approach, respectively. The relative expected profit is computed relatively to the value achieved by the solution of the stochastic program with uniform distribution.

We observe that the distributionally robust approaches are more conservative compared to the stochastic approaches. We invest between 4.3% and 12.2% of the expected profit in order to ensure the joint constraint. In fact, the average number of violated scenarios for the robust approaches are 0 while the numbers of violated scenarios for the stochastic solutions are significant, i.e., when Gamma is the true distribution of the random variables, the average number of the violated scenarios are 24 and 9 for the uniform and the normal solutions, respectively.

6 CONCLUSION

This paper studies a distributionally robust joint-constrained optimization problem for two different moments-based uncertainty sets. We propose a two-timescale neurodynamic duplex to solve the distributionally robust problems. We prove that the proposed approach converges almost surely to a global optimum. Finally, we use our method to solve a problem of profit maximization. We evaluate the performances of the neurodynamic duplex by comparing it to the state-of-the-art solving methods.

A key advantage of this research is its capability to solve distributionally robust joint-constrained programs without relying on convex or linear approxima-

Table 3: Number of violated scenarios for the stochastic and the robust solutions.

		Stochastic solutions		Robust solutions	
		Uniform	Normal	$\mathcal{D}^1(\mu,\Sigma)$	$\mathcal{D}^2(\mu,\Sigma)$
	Relative expected profit	-0%	-0.5%	-4.3%	-12.2%
Number of violated scenarios	Uniform distribution	2	0	0	0
	Normal distribution	8	5	0	0
	Log-normal distribution	15	6	0	0
	Logistic distribution	23	5	0	0
	Gamma distribution	24	9	0	0

tion techniques. The results reveal that our approach outperforms some existing state-of-the-art methods. Furthermore, our method effectively covers the risk area by generating robust solutions, thus ensuring reliable outcomes in uncertain scenarios.

However, it is crucial to acknowledge that the current iteration of the algorithm is time-consuming, primarily due to the iterative solutions required by the dynamical differential system that describes the model. Nonetheless, there are opportunities to enhance both the efficiency and quality of the algorithm through further research and development efforts. One potential approach to improve the algorithm is to implement artificial intelligence techniques, such as neural networks or reinforcement learning, in the Ordinary Differential Equation (ODE) solvers. By incorporating these AI techniques, there is a possibility to enhance the speed and accuracy of solving the dynamical differential system. Furthermore, other computational techniques, such as parallel computing, GPU acceleration, or distributed computing, can be leveraged to further reduce the execution time of the algorithm. These techniques can make use of advancements in hardware to process computations in parallel, leading to significant time savings in solving the system.

REFERENCES

Bertsimas, D. and Brown, D. B. (2009). Constructing uncertainty sets for robust linear optimization. *Operations Research*, 57(6):1483–1495.

Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research*, 52(1):35–53.

Charnes, A. and Cooper, W. W. (1959). Chance-constrained programming. *Management Science*, 6(1):73–79.

Cheng, J., Delage, E., and Lisser, A. (2014). Distributionally robust stochastic knapsack problem. *SIAM Journal on Optimization*, 24(3):1485–1506.

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

Dai, Q. and Yang, J. (2020). A distributionally robust chance-constrained approach for modeling demand uncertainty in green port-hinterland transportation network optimization. *Symmetry*, 12(9).

Dou, X. and Anitescu, M. (2019). Distributionally robust optimization with correlated data from vector autoregressive processes. *Operations Research Letters*, 47(4):294–299.

Du, N., Liu, Y., and Liu, Y. (2021). A new data-driven distributionally robust portfolio optimization method based on wasserstein ambiguity set. *IEEE Access*, 9:3174–3194.

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

Faury, L., Tanielian, U., Dohmatob, E., Smirnova, E., and Vasile, F. (2020). Distributionally robust counterfactual risk minimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3850–3857.

Georghiou, A., Tsoukalas, A., and Wiesemann, W. (2020). A primal–dual lifting scheme for two-stage robust optimization. *Operations Research*, 68(2):572–590.

Ghosal, S. and Wiesemann, W. (2020). The distributionally robust chance-constrained vehicle routing problem. *Operations Research*, 68(3):716–732.

Hanasusanto, G., Kuhn, D., Wallace, S., and Zymler, S. (2014). Distributionally robust multi-item newsvendor problems with multimodal demand distributions. *Mathematical Programming*, 152.

Hanasusanto, G. A., Kuhn, D., and Wiesemann, W. (2016). K-adaptability in two-stage distributionally robust binary programming. *Operations Research Letters*, 44(1):6–11.

Hopfield, J. and Tank, D. (1985). Neural computation of decisions in optimization problems. *Biological cybernetics*, 52:141–52.

Li, X. and Ke, J. (2019). Robust assortment optimization using worst-case cvar under the multinomial logit model. *Operations Research Letters*, 47(5):452–457.

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

Liu, J., Lisser, A., and Chen, Z. (2022). Distributionally robust chance constrained geometric optimization. *Mathematics of Operations Research*, pages 0364–765X.

Nazemi, A. and Omidi, F. (2013). An efficient dynamic model for solving the shortest path problem. *Transportation Research Part C: Emerging Technologies*, 26:1–19.

Ordoudis, C., Nguyen, V. A., Kuhn, D., and Pinson, P. (2021). Energy and reserve dispatch with distributionally robust joint chance constraints. *Operations Research Letters*, 49(3):291–299.

Peng, S., Lisser, A., Singh, V. V., Gupta, N., and Balachandar, E. (2021). Games with distributionally robust joint chance constraints. *Optim. Lett.*, 15(6):1931–1953.

Postek, K., den Hertog, D., and Melenberg, B. (2016). Computationally tractable counterparts of distributionally robust constraints on risk measures. *SIAM Review*, 58(4):603–650.

Shang, C. and You, F. (2018). Distributionally robust optimization for planning and scheduling under uncertainty. *Computers & Chemical Engineering*, 110:53–68.

Shen, R., Meng, Z., and Jiang, M. (2020). Smoothing partially exact penalty function of biconvex programming. *Asia-Pacific Journal of Operational Research*, 37(04):2040018.

Tassouli, S. and Lisser, A. (2023). A neural network approach to solve geometric programs with joint probabilistic constraints. *Mathematics and Computers in Simulation*, 205:765–777.

Uryasev, S. and Pardalos, P. (2013). *Stochastic Optimization: Algorithms and Applications*. Applied Optimization. Springer US.

Wang, J. (1994). A deterministic annealing neural network for convex programming. *Neural Networks*, 7(4):629–641.

Wiesemann, W., Kuhn, D., and Sim, M. (2014). Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376.

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

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

Zhang, Y., Shen, S., and Erdogan, S. A. (2017). Distributionally robust appointment scheduling with moment-based ambiguity set. *Operations Research Letters*, 45(2):139–144.