



A neurodynamic approach for distributionally robust joint chance-constrained optimization problems

Siham Tassouli, Abdel Lisser

► To cite this version:

Siham Tassouli, Abdel Lisser. A neurodynamic approach for distributionally robust joint chance-constrained optimization problems. 2022. hal-03798527

HAL Id: hal-03798527

<https://hal.science/hal-03798527>

Preprint submitted on 5 Oct 2022

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

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

A neurodynamic approach for distributionally robust joint chance-constrained optimization problems

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 studies a dynamical neural network approach to solve joint chance-constrained distributionally robust optimization problems. We consider that the row vectors of the matrix defining the constraints are independent. 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 two uncertainty sets for the unknown distributions. The main feature of our framework is to propose a neural network-based method to solve distributionally robust joint chance-constrained optimization problems without the use of standard state-of-the-art solving methods. We show the convergence and stability of the proposed neural network. In the numerical Section, we apply the proposed approach to solve a profit maximization problem to show the performances of our approach.

Keywords: Dynamical neural network, Distributionally robust optimization, Joint chance constraints, Lyapunov Theory, Ordinary differential equations.

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. 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 [2] use second-order cone programming to solve least-squares problems where the stochastic parameters are not known but bounded. Bertsimas & Sim [3] introduce a less conservative approach to solve linear optimization problems with uncertain data. Bertsimas & Brown [4] propose a general scheme for designing uncertainty sets for robust optimization. Wiesemann et al. [5] propose standardized ambiguity sets for modeling and solving distributionally robust optimization problems. Peng et al. [6] study one density-based uncertainty set and four two-moments based

^{*}Corresponding author.

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

15 uncertainty sets to solve games with distributionally robust joint chance constraints. Cheng et al. [7] solve a distributionally robust quadratic knapsack problem. Dou & Animescu [8] propose a new ambiguity set tailored to unimodal and seemingly symmetric distributions to deal with distributionally robust chance constraints. Li & Ke [9] approximate a distributionally robust chance constraint by the worst-case Conditional Value-at-Risk. Hanasusanto et al. [10] approximate two-stage distributionally
20 robust programs with binary recourse decisions. Georghiou et al. [11] 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 [12], multistage distribution system planning [13], portfolio optimization problems [14, 15], planning and scheduling [16], risk measures [17], multimodal demand problems
25 [18], appointment scheduling [19], vehicle routine problems [20] and energy and reserve dispatch [21].

This paper proposes a recurrent neural network to solve robust joint chance-constrained problems. The use of neural networks to solve optimization problems has been actively studied since the 1980s when the idea was firstly introduced by Tank & Hopfield [22]. Xia & Wang [23] present a recurrent neural network for solving nonlinear convex programming problems subject to nonlinear inequality
30 constraints. Wang [24] proposes a deterministic annealing neural network for convex programming. Nazemi & Omedi [25] presents a neural network model for solving the shortest path problems.

In this paper, we study two moment-based uncertainty sets and give the deterministic equivalent of the robust initial problem for each set. Based on the partial KKT system of the equivalent deterministic programs, we construct a recurrent system that converges to a partial optimum of the
35 proposed programs. The main contributions of our work are threefold.

- (i) On the formulation side, we reformulate the distributionally robust initial problem as a non-linear biconvex problem for each uncertainty set. Then, we study the optimality conditions for each deterministic program and based on the partial KKT system, we propose a recurrent neural network formulation. To the best of our knowledge, distributionally robust joint chance-
40 constrained optimization problems have not been reformulated as dynamical neural networks.
- (ii) On the theoretical side, we show that every equilibrium point of the dynamical system is a partial KKT point. We also prove the stability and convergence of the neural network.
- (iii) On the numerical side, we show that the neural network gives a robust upper bound to the initial problem.

45 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. Based on the partial KKT systems introduced in Section 2, we construct in Section 3 a stable recurrent neural network that converges to a partial optimal solution to the initial problem. In Section 4, we introduce a profit maximization problem to evaluate
50 the proposed approach.

2. Problem statement and optimality conditions

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

$$\min_{x \in \mathbb{R}_+^n} \sup_{\mathcal{F}_0 \in \mathcal{D}_0} \mathbb{E} \begin{bmatrix} \tilde{\zeta}_0^T & x \end{bmatrix}, \quad (1)$$

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

where $\tilde{\zeta}_0 \in \mathbb{R}^n$ is an uncertain parameter, $[\tilde{\zeta}_1, \tilde{\zeta}_2, \dots, \tilde{\zeta}_K]^T$ 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 $\tilde{\zeta}_k$, $k = 1, \dots, K$ are not completely known and belong to \mathcal{D}_k . Thus, we take the worst-case scenario 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 $(\tilde{\zeta}_k)_{k \in \{1, \dots, K\}}$, we introduce nonnegative auxiliary variables z_k , $k = 1, \dots, K$ and rewrite the constraint (2) as

$$\begin{cases} \inf_{\mathcal{F}_k \in \mathcal{D}_k} \mathbb{P} \left(\tilde{\zeta}_k x \leq b_k \right) \geq \alpha^{z_k}, k = 1, \dots, K \\ \sum_{k=1}^K z_k = 1, \\ z_k \geq 0, k = 0, 1, \dots, K. \end{cases} \quad (3)$$

In the following subsections, we give deterministic equivalents problems for (1)-(2) using two moments based uncertainty sets to define \mathcal{D}_k , $k = 1, \dots, K$.

We first assume that we know the mean vector μ_k and the covariance matrix Σ_k of $\tilde{\zeta}_k^T$. We define
 55 for every $k = 0, 1, \dots, K$
 $\mathcal{D}_k^1(\mu_k, \Sigma_k) = \left\{ \mathcal{F}_k \left| \begin{array}{l} \mathbb{E}[\tilde{\zeta}_k^T] = \mu_k \\ \mathbb{E}[(\tilde{\zeta}_k^T - \mu_k)(\tilde{\zeta}_k^T - \mu_k)^T] = \Sigma_k \end{array} \right. \right\}$, 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 [7].

$$\begin{cases} \mu_k^T x + \sqrt{\frac{\alpha^{z_k}}{1 - \alpha^{z_k}}} \|\Sigma_k^{\frac{1}{2}} x\| \leq b_k, k = 1, \dots, K \\ \sum_{k=1}^K z_k = 1, \\ z_k \geq 0, k = 1, \dots, K. \end{cases} \quad (4)$$

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

$$\min \mu_0^T x, \quad (5)$$

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

$$\sum_{k=1}^K z_k = 1, x \geq 0, \quad (7)$$

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

60 **Lemma 1.** The function $z \mapsto \sqrt{\frac{\alpha^z}{1-\alpha^z}}$, with $0 \leq \alpha \leq 1$ is convex $\forall z \geq 0$.

Proof. Let $z \geq 0$ and $0 \leq \alpha \leq 1$, we have $\sqrt{\frac{\alpha^z}{1-\alpha^z}} = \exp\{\frac{1}{2}(z\log(\alpha) - \log(1-\alpha^z))\}$. 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 then 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 we conclude then
65 that $z \mapsto \exp\{\frac{1}{2}(z\log(\alpha) - \log(1-\alpha^z))\}$ is convex. The conclusion follows. \square

Corollary 2. Problem (5)-(8) is biconvex.

Now we consider that the mean vector μ_k lies in an ellipsoid of size $\gamma_{k1} \geq 0$ and that the covariance matrix Σ_k lies in a positive semidefinite cone. We define for every $k = 0, 1, \dots, K$,

$$\mathcal{D}_k^2(\mu_k, \Sigma_k) = \left\{ \mathcal{F}_k \left| \begin{array}{l} (\mathbb{E}_{\mathcal{F}_k}[\tilde{\zeta}_k^T] - \mu_k)^T \Sigma_k^{-1} (\mathbb{E}_{\mathcal{F}_k}[\tilde{\zeta}_k^T] - \mu_k) \leq \gamma_{k1} \\ \text{COV}_{\mathcal{F}_k}(\tilde{\zeta}_k^T) \preceq \gamma_{k2} \Sigma_k \end{array} \right. \right\},$$
where $\gamma_{k2} \geq 0$ and $\text{COV}_{\mathcal{F}_k}$ 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 [6] as follows.

$$\left\{ \begin{array}{l} \mu_k^T x + (\sqrt{\frac{\alpha^{z_k}}{1-\alpha^{z_k}}} \sqrt{\gamma_{k2}} + \sqrt{\gamma_{k1}}) \|\Sigma_k^{\frac{1}{2}} x\| \leq b_k, k = 1, \dots, K \\ \sum_{k=1}^K z_k = 1, \\ z_k \geq 0, k = 1, \dots, K. \end{array} \right. \quad (9)$$

We can formulate the objective function as [26]

$$\min_{x \in \mathbb{R}_+^n} \mu_0^T x + \sqrt{\gamma_{01}} \|\Sigma_0^{\frac{1}{2}} x\|. \quad (10)$$

The constraints set (9) is biconvex and the objective function (10) is convex.

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

$$\min f(x), \quad (11)$$

$$\text{s.t. } g(x, z) \leq 0, \quad (12)$$

$$h(z) \leq 0, \quad (13)$$

$$l(x) \leq 0, \quad (14)$$

$$\text{where, } f(x) = \begin{cases} \mu_0^T x, & \text{if } \mathcal{D}_k = \mathcal{D}_k^1 \\ \mu_0^T x + \sqrt{\gamma_{01}} \|\Sigma_0^{\frac{1}{2}} x\|, & \text{if } \mathcal{D}_k = \mathcal{D}_k^2 \end{cases}, h(z) = (\sum_{k=1}^K z_k - 1, 1 - \sum_{k=1}^K z_k, -z_1, -z_2, \dots, z_K)^T,$$

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

70 To derive the optimality conditions of problem (11)-(14), we use the partial KKT system defined as follows.

Definition 1. Let \mathcal{U} the feasible set of (11)-(14), let $(x^*, z^*) \in \mathcal{U}$. If there exists $\beta^{(1)}$, $\beta^{(2)}$, γ , and λ such that

$$\nabla_x f(x^*) + \beta^{(1)T} \nabla_x g(x^*, z^*) + \lambda^T \nabla_x l(x^*) = 0, \quad (15)$$

$$\beta^{(1)} \geq 0, \beta^{(1)T} g(x^*, z^*) = 0, \lambda \geq 0, \lambda^T l(x^*) = 0, \quad (16)$$

$$\beta^{(2)T} \nabla_z g(x^*, z^*) + \gamma^T \nabla_z h(z^*) = 0, \quad (17)$$

$$\beta^{(2)} \geq 0, \beta^{(2)T} g(x^*, z^*) = 0, \gamma \geq 0, \gamma^T h(z^*) = 0, \quad (18)$$

then (x^*, z^*) is called a partial KKT point of (11)-(14).

The following theorem gives the optimality conditions of problem (11)-(14).

Theorem 3. If (15)-(18) is satisfied with partial Slater constraints qualification at (x^*, z^*) , then (x^*, z^*) is a partial optimum of (11)-(14) if and only if (x^*, z^*) is a partial KKT point of (11)-(14). Furthermore, if $\beta^{(1)} = \beta^{(2)}$ then (x^*, z^*) is a KKT point of (11)-(14).

Remark 4. The lines of the proof of Theorem 3 are given in [27].

3. A dynamical neural network

Based on the partial KKT system obtained in the previous Section, we construct a dynamical neural network to solve the robust chance-constrained problem. The following ordinary system drives the constructed neural network.

$$\frac{dx}{dt} = -(\nabla f(x) + \nabla_x g(x, z)^T (\beta + g(x, z))_+ + \nabla_x l(x)^T (\lambda + l(x))_+), \quad (19)$$

$$\frac{dz}{dt} = -(\nabla_z g(x, z)^T (\beta + g(x, z))_+ + \nabla_z h(z)^T (\gamma + h(z))_+), \quad (20)$$

$$\frac{d\beta}{dt} = (\beta + g(x, z))_+ - \beta, \quad (21)$$

$$\frac{d\gamma}{dt} = (\gamma + h(z))_+ - \gamma, \quad (22)$$

$$\frac{d\lambda}{dt} = (\lambda + l(x))_+ - \lambda. \quad (23)$$

System (19)-(23) can be shortly written as $\frac{dy}{dt} = \Phi(y)$ where $y = (x, z, \beta, \gamma, \lambda)^T$ and

$$\Phi(y) = \begin{bmatrix} -(\nabla f(x) + \nabla_x g(x, z)^T (\beta + g(x, z))_+ + \nabla_x l(x)^T (\lambda + l(x))_+) \\ -(\nabla_z g(x, z)^T (\beta + g(x, z))_+ + \nabla_z h(z)^T (\gamma + h(z))_+) \\ (\beta + g(x, z))_+ - \beta \\ (\gamma + h(z))_+ - \gamma \\ (\lambda + l(x))_+ - \lambda \end{bmatrix}.$$

A generalized circuit implementation of neural network (19)-(23) is represented in Figure 1.

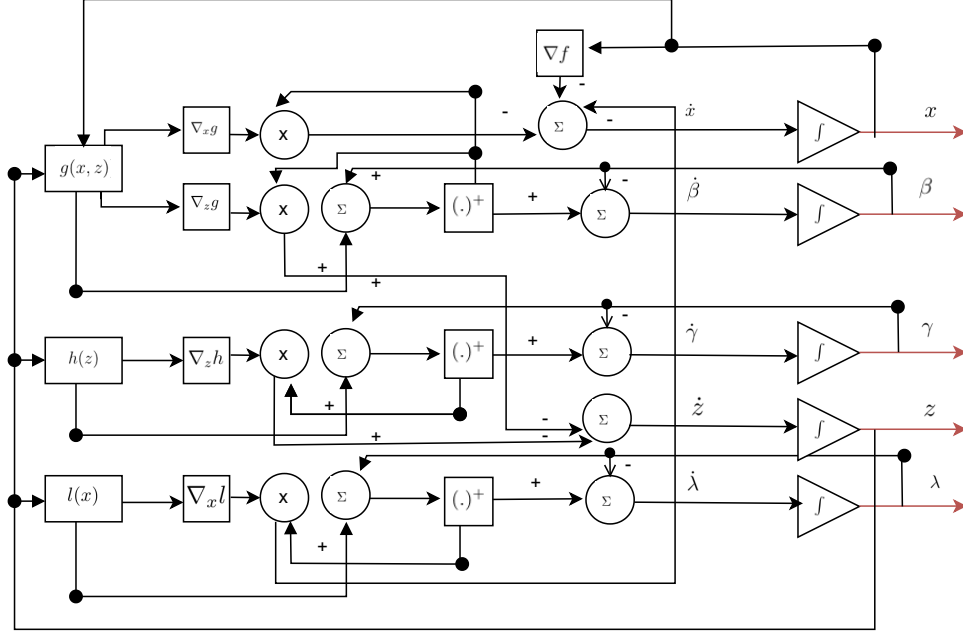


Figure 1: The implementation of the neural network (19)-(23)

Theorem 5. Let $y^* = (x^*, z^*, \beta^*, \gamma^*, \lambda^*)^T$ an equilibrium point of (19)-(23), then (x^*, z^*) is a partial optimum of (11)-(14). If $(x^*, z^*, \beta^*, \gamma^*, \lambda^*)$ is feasible for the partial KKT system (15)-(18) with $\beta^{(1)} = \beta^{(2)} = \beta$ then $y^* = (x^*, z^*, \beta^*, \gamma^*, \lambda^*)^T$ is an equilibrium point of (19)-(23).

Proof. Observe that $(\beta + g(x, z))_+ - \beta = 0 \Leftrightarrow (\beta \geq 0 \text{ and } g(x, z) \leq 0)$, the same observation goes for
 90 $(\gamma + h(z))_+ - \gamma$ and $(\lambda + l(x))_+ - \lambda$. Let $(x^*, z^*, \beta^*, \gamma^*, \lambda^*)^T$ an equilibrium point of (19)-(23), we have then that $\frac{dx^*}{dt} = \frac{dy^*}{dt} = \frac{d\beta^*}{dt} = \frac{d\gamma^*}{dt} = \frac{d\lambda^*}{dt} = 0$.

Therefore, We have $\frac{d\beta^*}{dt} = 0 \Leftrightarrow (\beta^* \geq 0 \text{ and } g(x^*, z^*) \leq 0 \text{ and } \beta^{*T} g(x^*, z^*) = 0)$, we obtain then equation (16) with $\beta^{(1)} = \beta$ of the partial KKT system (15)-(18). We replace $(\beta + g(x, z))_+$ by β in (19) to obtain (11). We obtain the remaining equations of the KKT system following the same steps.

95 The converse part of the proof is straightforward. \square

To prove the stability of the neural network (19)-(23), we first need to introduce the following definition and lemma.

Definition 2. [28] A mapping $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is said to be monotonic if

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

100 **Lemma 6.** [28] A differentiable mapping $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is monotonic, if and only if the jacobian matrix $\nabla F(x)$, $\forall x \in \mathbb{R}^n$, is positive semidefinite.

Theorem 7. Let (x^*, z^*) a partial optimum of (11)-(14), and $y^* = (x^*, z^*, \beta^*, \gamma^*, \lambda^*)^T$ the corresponding partial KKT point. Then a necessary condition for the neural network in (19)-(23) to be stable at y^* is that $\nabla\Phi(y)$ is negative semidefinite.

Proof. We consider the following Lyapunov function.

$$V(y) = \|\Phi(y)\|^2 + \frac{1}{2} \|y - y^*\|^2 \quad (24)$$

105 Notice that $\frac{d\Phi}{dt} = \frac{\nabla\Phi}{\nabla y} \frac{dy}{dt} = \nabla\Phi(y)\Phi(y)$, we have then $\frac{dV(y(t))}{dt} = (\frac{d\Phi}{dt})^T \Phi + \Phi^T \frac{d\Phi}{dt} + (y - y^*)^T \frac{dy}{dt} = \Phi^T (\nabla\Phi(y)^T + \nabla\Phi(y)) \Phi + (y - y^*)^T \Phi(y)$. Since, $\nabla\Phi$ is negative semidefinite we have $\Phi^T (\nabla\Phi(y)^T + \nabla\Phi(y)) \Phi \leq 0$ and $(y - y^*)^T \Phi(y) = (y - y^*)^T (\Phi(y) - \Phi(y^*)) \leq 0$ by Lemma 6. There follows that $\frac{dV(y)}{dt} \leq 0$. According to the Lyapunov theory, the neural network in (19)-(23) is stable at y^* . \square

Theorem 8. The Jacobian matrix $\nabla\Phi(y)$ is negative semidefinite.

110 *Proof.* Without loss of generality, We assume that there exists $p, q, m \in \mathbb{N}$ such that

$$\begin{aligned} (\beta + g)_+ &= (\beta_1 + g_1(x, z), \beta_2 + g_2(x, z), \dots, \beta_p + g_p(x, z), \underbrace{0, \dots, 0}_{K-p}), \\ (\lambda + h)_+ &= (\lambda_1 + h_1(z), \lambda_2 + h_2(z), \dots, \lambda_q + h_q(z), \underbrace{0, \dots, 0}_{2-q}), \\ (\gamma + l)_+ &= (\gamma_1 + l_1(z), \gamma_2 + l_2(z), \dots, \gamma_m + l_m(z), \underbrace{0, \dots, 0}_{K-m}). \end{aligned}$$

The Jacobian matrix $\nabla\Phi(y)$ can be written as $\nabla\Phi(y) = \begin{bmatrix} A_1 & A_2 & A_3 & 0 & 0 \\ B_1 & B_2 & B_3 & B_4 & B_5 \\ C_1 & C_2 & C_3 & 0 & 0 \\ 0 & D_2 & 0 & D_4 & 0 \\ 0 & E_2 & 0 & 0 & E_5 \end{bmatrix}$

where

$$A_1 = -(\nabla^2 f(x) + \sum_{i=1}^p ((\beta_i + g_i) \nabla_x^2 g_i^p(x, z)) + \nabla_x g^p(x, z)^T \nabla_x g^p(x, z)),$$

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

$$A_3 = -\nabla_x g(x, z)^T,$$

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

$$\begin{aligned} B_2 &= -(\sum_{i=1}^p ((\beta_i + g_i) \nabla_z^2 g_i^p(x, z)) + \nabla_z g^p(x, z)^T \nabla_z g^p(x, z) + \sum_{i=1}^q ((\lambda_i + h) \nabla_z^2 h_i^q(z)) + \nabla_z h^q(z)^T \nabla_z h^q(z)) \\ &\quad + \sum_{i=1}^m ((\gamma_i + l) \nabla_z^2 l_i^m(z)) + \nabla_z l^m(z)^T \nabla_z l^m(z), \end{aligned}$$

$$B_3 = -\nabla_z g(x, z)^T, \quad B_4 = -\nabla_z h(z)^T, \quad B_5 = -\nabla_z l(x)^T, \quad C_1 = \nabla_x g(x, z), \quad C_2 = \nabla_z g(x, z),$$

$$C_3 = -S_p, \quad D_2 = \nabla_z h(z), \quad D_4 = -S_q, \quad E_2 = \nabla_z l(x), \quad E_5 = -S_m,$$

$$\begin{aligned}
115 \quad \text{where } S_p &= \begin{bmatrix} O_{p \times p} & O_{p \times (K-p)} \\ O_{(K-p) \times p} & I_{(K-p) \times (K-p)} \end{bmatrix}, \quad S_q = \begin{bmatrix} O_{q \times q} & O_{q \times (2-q)} \\ O_{(2-q) \times q} & I_{(2-q) \times (2-q)} \end{bmatrix} \\
\text{and} \quad S_m &= \begin{bmatrix} O_{m \times m} & O_{m \times (K-m)} \\ O_{(K-m) \times m} & I_{(K-m) \times (K-m)} \end{bmatrix}.
\end{aligned}$$

We write $\nabla\Phi(y)$ as,

$$\nabla\Phi(y) = \left(\begin{array}{cc|c} A_1 & B_1^T & \mathbb{B} \\ B_1 & B_2 & \\ \hline -\mathbb{B}^T & & \mathbb{S} \end{array} \right), \text{ where } \mathbb{B} = \begin{bmatrix} A_3 & 0 & 0 \\ B_3 & B_4 & B_5 \end{bmatrix} \text{ and } \mathbb{S} = \begin{bmatrix} -S_p & 0 & 0 \\ 0 & -S_q & 0 \\ 0 & 0 & -S_m \end{bmatrix}.$$

Recall that g is biconvex, f , l and h are convex functions then we have $\nabla^2 f$, $\nabla_{xg}^2 g$, $\nabla_z^2 g$, $\nabla_z^2 l$ and $\nabla_z^2 h$ are positive semidefinite. Moreover, it is straightforward that $\nabla_x g^T \nabla_x g$, $\nabla_z g^T \nabla_z g$, $\nabla_z h^T \nabla_z h$ and $\nabla_z l^T \nabla_z l$ are positive semidefinite. It follows that A_1 and B_2 are negative semidefinite matrix, and then $\begin{bmatrix} A_1 & B_1^T \\ B_1 & B_2 \end{bmatrix}$ is negative semidefinite. \mathbb{S} is obviously negative semidefinite. We conclude that $\nabla\Phi$ is negative semidefinite. \square

Theorem 9. The neural network (19)-(23) converges globally to a KKT point of (11)-(14).

Proof. Let y^* an equilibrium point of (19)-(23). We have $V(y) \geq \frac{1}{2} \|y - y^*\|^2$ (defined in (24)). Consequently, there exists a convergent subsequence $(y(t_k)_{k \geq 0})$ where $\lim_{k \rightarrow \infty} y(t_k) = \hat{y}$ and \hat{y} satisfies $\frac{dV(y(t))}{dt} = 0$. We define $S = \{y | \frac{dV(y)}{dt} = 0\}$, by LaSalle's invariance principle [29] the neural network converges to the largest invariant set contained in S . Notice that

$$\frac{dy}{dt} = 0 \Leftrightarrow \frac{dV(y)}{dt} = 0, \tag{25}$$

125 then \hat{y} is an equilibrium point of (19)-(23).

Now, to show that the neural network converges to \hat{y} , we introduce a new Lyapunov function defined as follows.

$$W(y) = \|\Phi(y)\|^2 + \frac{1}{2} \|y - \hat{y}\|^2. \tag{26}$$

Notice that W is continuously differentiable and $\lim_{k \rightarrow \infty} y(t_k) = \hat{y}$ and since $W(\hat{y}) = 0$, we have

$$\lim_{t \rightarrow \infty} W(y(t)) = W(\hat{y}) = 0. \tag{27}$$

There follows from $\frac{dW(y)}{dt} \leq 0$ that $\frac{1}{2} \|y - \hat{y}\|^2 \leq W(y)$. From (27) we conclude that $\lim_{t \rightarrow \infty} y(t) = \hat{y}$. The neural network (19)-(23) is then convergent in the sense of Lyapunov to a KKT point of (11)-(14). \square

4. Numerical experiments

130 To evaluate the performances of our approach, we solve 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_i and the covariance matrix Σ_i describing the uncertainty sets of the time required to manufacture one unit of each of the n products and the daily capacity of the N machines are given. The objective of the study is to determine the daily

135 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.

$$\max \sup_{\mathcal{F}_0 \in \mathcal{D}_0} \mathbb{E} [c^T x], \quad (28)$$

$$\text{s.t. } \inf_{\mathcal{F} \in \mathcal{D}} \mathbb{P} \left(\sum_{i=1}^n t_{ij} x_i \leq b_j, j = 1, \dots, N \right) \geq p, \quad (29)$$

$$x \geq 0, \quad (30)$$

where vector 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 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`.

We solve problem (28)-(30) for different values of n and N with the proposed neural network (19)-(23). The values of μ_j and c are uniformly generated in $[12.0, 14.0]$, 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 $\gamma_{k2} = 5$. We first solve problem (28)-(30) for $N=7$ machines and $n = 10$ products. In order to compare the performances of our approach with the literature existing methods, we use the tangent approximation to solve (28)-(30) using ten tangent points [7]. Table 1 recapitulates the obtained results for different values of p . The first column gives the value of p , the second and third columns give the objective value and the CPU time in seconds obtained with the neural network when the uncertainty set $\mathcal{D} = \mathcal{D}^1(\mu, \Sigma)$, respectively. Columns four and five represent the objective value and the CPU time in seconds for the tangent approximation, respectively. Column six computes the gap between the objective values of the two approaches ($\text{GAP} = \frac{\text{Tangent approximation} - \text{NN}}{\text{Tangent approximation}}$). Table 2 represents the same results when $\mathcal{D} = \mathcal{D}^2(\mu, \Sigma)$. We observe that the neural network converges for the different levels of probability and that the gap with the lower bound given by the tangent approximation remains very tight, i.e., it does not exceed 2.7%.

To evaluate the robustness of the proposed neural network for the two uncertainty sets \mathcal{D}^1 and \mathcal{D}^2 , we additionally solve problem (28)-(30) 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

p	NN for $\mathcal{D}^1(\mu, \Sigma)$		Tangent approximation		GAP
	Obj value	CPU Time	Obj value	CPU Time	
0.95	26.77	2.29	27.09	0.19	1.1%
0.9	32.96	5.55	33.30	0.11	1.0%
0.85	32.68	4.62	33.20	0.22	1.5%
0.8	37.86	6.33	38.22	0.17	0.9%

Table 1: Results for different values of α

p	NN for $\mathcal{D}^2(\mu, \Sigma)$		Tangent approximation		GAP
	Obj value	CPU Time	Obj value	CPU Time	
0.95	15.85	1.17	16.22	0.18	2.2%
0.9	20.31	0.82	20.73	0.19	2.0%
0.85	21.22	3.45	21.83	0.14	2.7%
0.8	25.48	0.60	25.84	0.17	1.3%

Table 2: Results for different values of α

robust solutions. Table 3 recapitulates the obtained results, where column one gives the true distribu-
tion, columns two, three, four and five give the number of violated scenarios for the solution obtained
170 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
175 the joint constraint. In fact, the average numbers 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.

Now in order to evaluate the performances of our approach, we solve problem (28)-(30) for different

		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

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

180 numbers of machines N and products n . Tables 4 and 5 represent the obtained results. We observe that neural networks converge and that the gap increases as the problem size increases but does not exceed 6.8%. Gaps for $\mathcal{D}^2(\mu, \Sigma)$ are more significant than those for $\mathcal{D}^1(\mu, \Sigma)$. In fact, \mathcal{D}^2 gives less information about the stochastic parameters than \mathcal{D}^1 , and thus it has to be more conservative to ensure the robustness of the solutions.

N	n	NN for $\mathcal{D}^1(\mu, \Sigma)$		Tangent approximation		GAP
		Obj value	CPU Time	Obj value	CPU Time	
5	10	32.50	18.58	33.18	0.90	2.0%
10	15	35.53	73.09	36.54	12.10	2.7%
10	20	32.33	198.61	33.76	13.06	4.2%
15	25	33.44	506.62	35.22	29.64	5.0%

Table 4: Results for different values of n and N

N	n	NN for $\mathcal{D}^2(\mu, \Sigma)$		Tangent approximation		GAP
		Obj value	CPU Time	Obj value	CPU Time	
5	10	26.27	3.47	26.78	1.07	1.9%
10	15	26.11	22.10	26.95	6.24	3.9%
10	20	24.82	152.38	26.08	16.97	4.8%
15	25	26.53	288.46	28.49	38.74	6.8%

Table 5: Results for different values of n and N

185 We observe that the neural network has a significant CPU time compared to the tangent approximation since the tangent approximation uses very powerful convex solvers. Thereby, developing faster ODE solvers might decrease significantly the total CPU time. Notice that our approach does not replace the existing convex approximations but gives some promising results and opens the way for a new vision of the joint probabilistic problems.

5. Conclusion

190 This paper studies a distributionally robust joint constrained optimization problem for two different moments based uncertainty sets. We transform the initial stochastic problems into biconvex deterministic equivalent problems. We then study the optimality conditions of the obtained problems by introducing the partial KKT system. We use our approach to solve the problem of profit maximization in the numerical Section. We evaluate the performances of the robust solutions by comparing them to the state-of-the-art solving methods. We show that the neurodynamic approach gives better and more robust solutions. We note that the performances of our approach 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] 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. [arXiv:https://doi.org/10.1137/S0895479896298130](https://arxiv.org/abs/https://doi.org/10.1137/S0895479896298130), doi:10.1137/S0895479896298130.
URL <https://doi.org/10.1137/S0895479896298130>
- [3] D. Bertsimas, M. Sim, The price of robustness, *Operations Research* 52 (1) (2004) 35–53.
URL <http://www.jstor.org/stable/30036559>
- [4] 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>
- [5] W. Wiesemann, D. Kuhn, M. Sim, Distributionally robust convex optimization, *Operations Research* 62 (6) (2014) 1358–1376. [arXiv:https://doi.org/10.1287/opre.2014.1314](https://arxiv.org/abs/https://doi.org/10.1287/opre.2014.1314), doi:10.1287/opre.2014.1314.
URL <https://doi.org/10.1287/opre.2014.1314>
- [6] 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.
URL <https://doi.org/10.1007/s11590-021-01700-9>
- [7] J. Cheng, E. Delage, A. Lisser, Distributionally robust stochastic knapsack problem, *SIAM Journal on Optimization* 24 (3) (2014) 1485–1506. [arXiv:https://doi.org/10.1137/130915315](https://arxiv.org/abs/https://doi.org/10.1137/130915315), doi:10.1137/130915315.
URL <https://doi.org/10.1137/130915315>
- [8] 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>.
URL <https://www.sciencedirect.com/science/article/pii/S0167637719301099>
- [9] 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>
- [10] 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>

- 235 [11] A. Georghiou, A. Tsoukalas, W. Wiesemann, A primal–dual lifting scheme for two-stage robust optimization, *Operations Research* 68 (2) (2020) 572–590. arXiv:<https://doi.org/10.1287/opre.2019.1873>, doi:10.1287/opre.2019.1873.
URL <https://doi.org/10.1287/opre.2019.1873>
- [12] Q. Dai, J. Yang, A distributionally robust chance-constrained approach for modeling demand
240 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>
- [13] 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*
245 *Transactions on Power Systems* 33 (5) (2018) 5248–5262. doi:10.1109/TPWRS.2018.2792938.
- [14] 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.
URL <https://doi.org/10.1007/s10287-011-0132-0>
- [15] S. Wang, L. Pang, H. Guo, H. Zhang, Distributionally robust optimization with multivari-
250 ate second-order stochastic dominance constraints with applications in portfolio optimization,
Optimization 0 (0) (2022) 1–24. arXiv:<https://doi.org/10.1080/02331934.2022.2048382>,
doi:10.1080/02331934.2022.2048382.
URL <https://doi.org/10.1080/02331934.2022.2048382>
- [16] C. Shang, F. You, Distributionally robust optimization for planning and scheduling under uncer-
255 tainty, *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>
- [17] K. Postek, D. den Hertog, B. Melenberg, Computationally tractable counterparts of distribution-
ally robust constraints on risk measures, *SIAM Review* 58 (4) (2016) 603–650. arXiv:<https://doi.org/10.1137/151005221>, doi:10.1137/151005221.
260 URL <https://doi.org/10.1137/151005221>
- [18] G. Hanasusanto, D. Kuhn, S. Wallace, S. Zymmler, Distributionally robust multi-item newsvendor
problems with multimodal demand distributions, *Mathematical Programming* 152. doi:10.
1007/s10107-014-0776-y.
- 265 [19] 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>

- [20] S. Ghosal, W. Wiesemann, The distributionally robust chance-constrained vehicle routing problem, *Operations Research* 68 (3) (2020) 716–732. [arXiv:https://doi.org/10.1287/opre.2019.1924](https://doi.org/10.1287/opre.2019.1924), doi:10.1287/opre.2019.1924.
URL <https://doi.org/10.1287/opre.2019.1924>
- [21] 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>
- [22] D. W. Tank, J. J. Hopfield, Simple ‘neural’ optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit, 1986.
- [23] 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.
- [24] 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](https://doi.org/10.1016/0893-6080(94)90041-8).
URL <https://www.sciencedirect.com/science/article/pii/0893608094900418>
- [25] A. Nazemi, F. Omid, 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>
- [26] J. Liu, A. Lisser, Z. Chen, Distributionally robust chance constrained geometric optimization, *Mathematics of Operations Research* (2022) 0364–765XarXiv:<https://doi.org/10.1287/moor.2021.1233>, doi:10.1287/moor.2021.1233.
URL <https://doi.org/10.1287/moor.2021.1233>
- [27] M. Jiang, Z. Meng, R. Shen, Partial exactness for the penalty function of biconvex programming, *Entropy* 23 (2). doi:10.3390/e23020132.
URL <https://www.mdpi.com/1099-4300/23/2/132>
- [28] R. T. Rockafellar, R. J.-B. Wets, *Variational Analysis*, Springer, Berlin, Heidelberg, 1998. doi:10.1007/978-3-642-02431-3.
- [29] J.-J. E. Slotine, *Applied nonlinear control*, Prentice Hall, Englewood Cliffs (N.J.), 1991.