KKT-INFORMED NEURAL NETWORK

A PARALLEL SOLVER FOR PARAMETRIC CONVEX OPTIMIZATION PROBLEM

A PREPRINT

Carmine Delle Femine
Department of Data Intelligence for Energy and Industrial Processes
Vicomtech Foundation
Donostia-San Sebastián, 20009
cdellefemine@vicomtech.org

September 5, 2024

ABSTRACT

This is the abstract

Keywords

Optimization

1 Introduction

2 Background

Consider a parametric convex optimization problem in the standard form:

$$\begin{aligned} \min_{x \in \mathcal{D} \subseteq \mathbb{R}^n} \quad & f(x, \theta) \\ \text{s.t.} \quad & g_i(x, \theta) \leq 0 \quad i = 1, \dots, m \\ & A(\theta)x - b(\theta) = 0 \end{aligned}$$

where $x \in \mathcal{D} \subseteq \mathbb{R}^n$ is the optimization variable; $\theta \in \mathcal{D}_\theta \subseteq \mathbb{R}^k$ are the parameters defining the problem; $f : \mathcal{D}_f \subseteq \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}$ is the convex cost function; $g_i : \mathcal{D}_{g_i} \subseteq \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}$ are the convex inequality constraints, $A : \mathcal{D}_\theta \to \mathbb{R}^{p \times n}$ and $b : \mathcal{D}_\theta \to \mathbb{R}^p$ defines the affine equality constraints and $\mathcal{D} = \bigcap_{i=1}^m \mathcal{D}_{g_i} \cap \mathcal{D}_f$ is the domain of the optimization problem.

Assume differentiable cost and constraints functions and that g_i satisfies Slater's condition. Given a set of parameters θ, $x^* \in \mathcal{D}$ is optimal if and only if there are λ^* and ν^* that, with x^*, satisy the Karush-Kuhn-Tucker conditions (KKT):

$$A(\theta)x^* - b(\theta) = 0 \tag{1}$$

$$g_i(x^*, \theta) \leq 0 \quad i = 1, \dots, m \tag{2}$$

$$\lambda_i^* \geq 0 \quad i = 1, \dots, m \tag{3}$$

$$\lambda_i^* g_i(x^*, \theta) = 0 \quad i = 1, \dots, m \tag{4}$$

$$\nabla_{x^*} f(x^*, \theta) + \sum_{i=1}^m \lambda_i^* \nabla_{x^*} g_i(x^*, \theta) + A^T \nu^* = 0 \tag{5}$$

3 Proposed method

KKT-Informed Neural Network (KINN) builds upon the principles of Physics-Informed Neural Networks (PINNs), incorporating mathematical conditions of the Karush-Kuhn-Tucker (KKT) conditions directly into the neural architecture. This integration facilitates a disciplined learning process where the network not only predicts optimization variables but also ensures these predictions are compliant with KKT conditions, essential for guaranteeing the optimality of solutions in convex optimization under exam.

Network architecture is a MLP designed to take a batch of problem parameters $\Theta = \{\theta^{(i)}\}_{i=1}^{N}$ as input and predict x^*, λ^*, ν^*. A ReLU function is placed at the end of the branch predicting λ^* to ensure its feasability.

$$(\hat{x}, \hat{\lambda}, \hat{\nu}) = \text{KINN}(\Theta) \tag{6}$$

$$\hat{\lambda} \in \mathbb{R}_{+}^{0} \tag{7}$$

Loss function is so defined:

$$\mathcal{L} = \mathcal{L}_S + \sum_{i=1}^{m} \mathcal{L}_{I,i} + \mathcal{L}_E + \sum_{i=1}^{m} \mathcal{L}_{C,i}$$

where:

$$\mathcal{L}_S = \|\nabla_{\hat{x}} f(\hat{x}, \theta) + \sum\nolimits_{i=1}^{m} \hat{\lambda}_i \nabla_{\hat{x}} g_i(\hat{x}, \theta) + A^T \hat{\nu}\|_2 \tag{8}$$

$$\mathcal{L}_{I,i} = \|\max(0, g_i(\hat{x}, \theta))\|_2 \tag{9}$$

$$\mathcal{L}_E = \|A(\theta)\hat{x} - b(\theta)\|_2 \tag{10}$$

$$\mathcal{L}_{C,i} = \|\hat{\lambda}_i g_i(\hat{x}, \theta)\|_2 \tag{11}$$

$$\tag{12}$$

4 Case study

Let us take such a problem as a test case for this approach:

We have a renewable energy generator in a power grid, whose active and reactive power injections are controllable. The set of injection points (P, Q) is limited by physical constraints, so the set-points (a_P, a_Q) must be projected onto that set.

4.1 Problem description

The feasibile set $\mathcal{D}$ is defined by the physical parameters of the generator $\overline{P}_g, \underline{P}_g, P_g^+, \overline{Q}_g, \underline{Q}_g, Q_g^+, Q_g^-$, characterizing the minimum and maximum possible values and the relationships between active and reactive power, and the dynamic value $P_{g,t}^{(\max)}$ which indicates the maximum power that can be generated at that time given the external conditions (e.g. wind speed, solar radiation, etc.):

$$\mathcal{D} = \{(P, Q) \in \mathbb{R}^2 | \underline{P}_g \le P \le P_{g,t}^{(\max)}, Q_g \le Q \le \overline{Q}_g, Q \le \tau_g^{(1)} P + \rho_g^{(1)}, Q \ge \tau_g^{(2)} P + \rho_g^{(2)}\} \tag{13}$$

where:

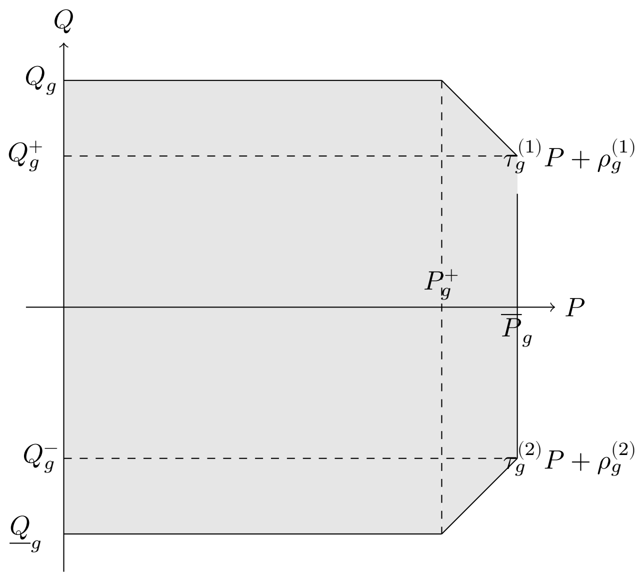
$$\tau_g^{(1)} = \frac{Q_g^+ - \overline{Q}_g}{\overline{P}_g - P_g^+} \tag{14}$$

$$\rho_g^{(1)} = \overline{Q}_g - \tau_g^{(1)} P_g^+ \tag{15}$$

$$\tau_g^{(2)} = \frac{Q_g^- - \underline{Q}_g}{\overline{P}_g - P_g^+} \tag{16}$$

$$\rho_g^{(2)} = \underline{Q}_g - \tau_g^{(2)} P_g^+ \tag{17}$$

$$\tag{18}$$



The problem could be stated in standard form as:

$$\begin{aligned} \min_{x \in \mathcal{D} \subseteq \mathbb{R}^2} \quad & \frac{1}{2}\|a - x\|_2^2 \\ \text{s.t.} \quad & Gx - h \leq 0 \end{aligned}$$

with $a = (a_P, a_Q)$, $x = (P, Q)$ and:

$$G = \begin{pmatrix} -1 & 1 & 1 & 0 & 0 & -\tau_g^{(1)} & \tau_g^{(2)} \\ 0 & 0 & 0 & -1 & 1 & 1 & 1 \end{pmatrix}^T \tag{19}$$

$$h = \begin{pmatrix} -\underline{P}_g & \overline{P}_g & P_{g,t}^{(\max)} & -\underline{Q}_g & \overline{Q}_g & \rho_g^{(1)} & -\rho_g^{(2)} \end{pmatrix}^T \tag{20}$$

With associated KKT conditions:

$$G\hat{x} - h \leq 0 \tag{21}$$
$$\lambda_i^* \geq 0 \quad i = 1, \dots, 7 \tag{22}$$
$$G^T \hat{\lambda} = 0 \tag{23}$$
$$(a - \hat{x}) + G^T \hat{\lambda} = 0 \tag{24}$$

4.2 Experimental results

The problem described has a two-dimensional optimization variable, ten scalar parameters and seven constraints:

$$(X, \Lambda) = \text{KINN}(\Theta) \tag{25}$$

with:

$$\Theta \in \mathbb{R}^{N \times 10}, \quad \Theta_i = (a_P^{(i)}, a_Q^{(i)}, \overline{P}_g^{(i)}, \underline{P}_g^{(i)}, P_g^{+(i)}, \overline{Q}_g^{(i)}, \underline{Q}_g^{(i)}, Q_g^{+(i)}, Q_g^{-(i)}, P_{g,t}^{(\max)^{(i)}}) \tag{26}$$
$$X \in \mathbb{R}^{N \times 2}, \quad X_i = (P^{(i)}, Q^{(i)}) \tag{27}$$
$$\Lambda \in \mathbb{R}_+^{0^{N \times 7}}, \quad \Lambda_i = \lambda^{(i)} \tag{28}$$

At each training step, a random batch of parameters Θ was sampled:

$$
\begin{aligned}
a_P^{(i)} &\sim U(0\,\text{p.u.}, 1\,\text{p.u.}) && (29)\\
a_Q^{(i)} &\sim U(-1\,\text{p.u.}, 1\,\text{p.u.}) && (30)\\
\overline{P}_g^{(i)} &\sim U(0.2\,\text{p.u.}, 0.8\,\text{p.u.}) && (31)\\
P_g^{+(i)} &\sim U(0\,\text{p.u.}, \overline{P}_g^{(i)}) && (32)\\
\overline{Q}_g^{(i)} &\sim U(0.2\,\text{p.u.}, 0.8\,\text{p.u.}) && (33)\\
Q_g^{+(i)} &\sim U(0\,\text{p.u.}, \overline{Q}_g^{(i)}) && (34)\\
P_{g,t}^{(\max)^{(i)}} &\sim U(0\,\text{p.u.}, \overline{P}_g^{(i)}) && (35)
\end{aligned}
$$

Models parameters were update to minimize the following loss function:

$$
\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{L}_S^{(i)} + \mathcal{L}_I^{(i)} + \mathcal{L}_C^{(i)})
$$

where:

$$
\begin{aligned}
\mathcal{L}_S^{(i)} &= \|(a^{(i)} - \hat{x}^{(i)}) + G^{(i)^T}\hat{\lambda}^{(i)}\|_2 && (36)\\
\mathcal{L}_I^{(i)} &= \|\max(0, G^{(i)}\hat{x} - h^{(i)})\|_2 && (37)\\
\mathcal{L}_C^{(i)} &= \|G^{(i)^T}\hat{\lambda}^{(i)}\|_2 && (38)\\
& && (39)
\end{aligned}
$$

4.2.1 Training

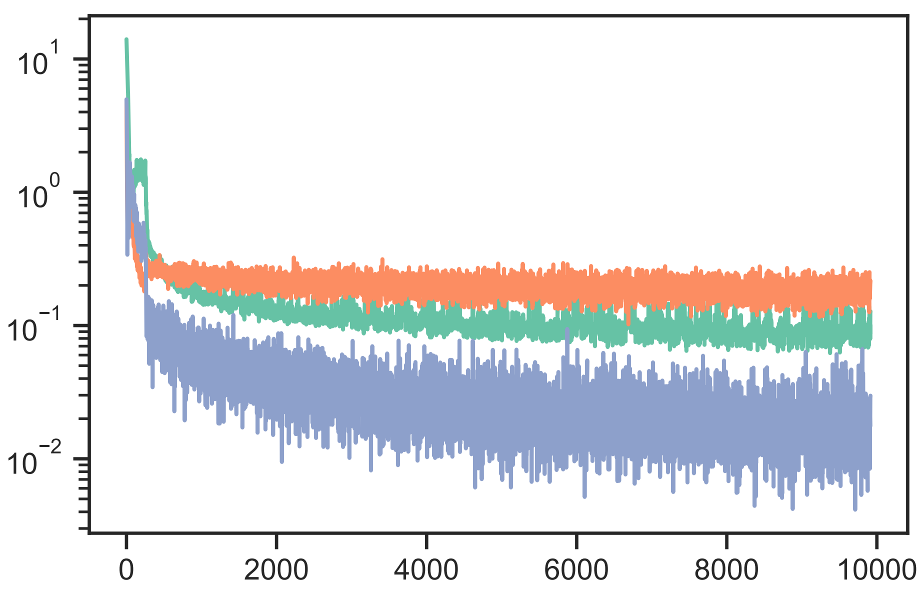


Figure 1: A line plot on a polar axis

4.2.2 Evaluation

4.2.3 Comparison

5 Conclusions

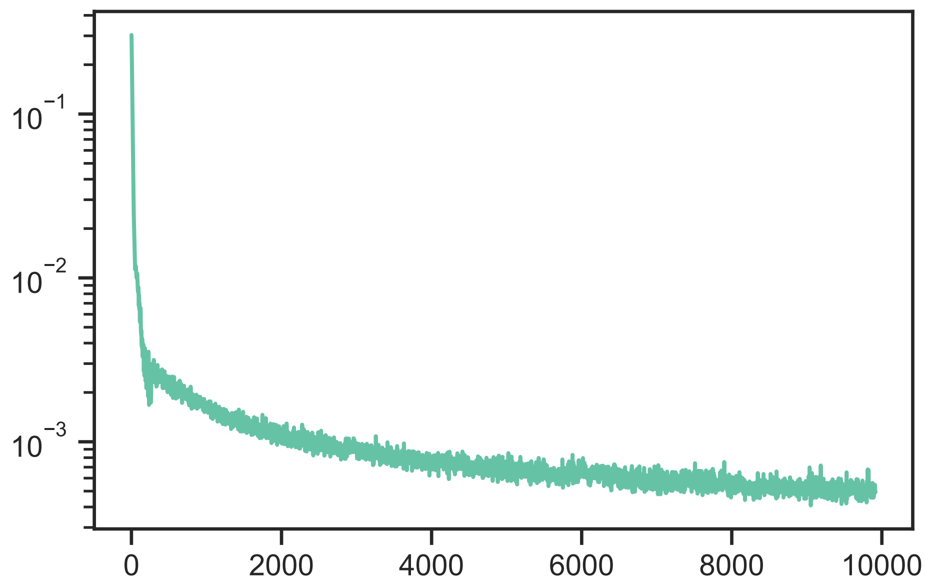


Figure 2: A line plot on a polar axis

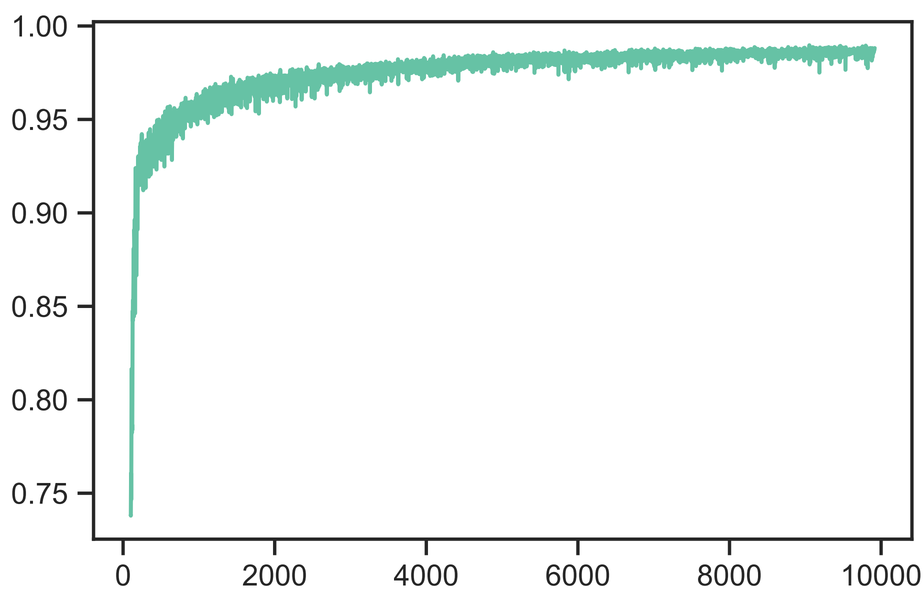


Figure 3: A line plot on a polar axis