Polynomial Optimization Methods for Robustness Certification Problems in Deep Neural Networks

Tong CHEN
M2 StatML, Université Paris Sud
Tutors: Edouard PAUWELS, Victor MAGRON, Jean-Bernard LASSERRE

1 Introduction

Artificial neural network is widely used for classification, when we talk about neural network classifier, we often refer to the parameters in the network. It is very important to know whether a classifier is reliable or not, in other words, whether it is robust with respect to inputs. This induces the problem to verify the robustness of a given classifier, that is, if we perturbate the inputs a little, the output should not be different. In fact, people have already discovered many adversarial examples [14] (instances with small, intentional feature perturbations that cause a machine learning model to make a false prediction) in neural networks. Historically, when adversarial example occurs, people tried to propose defenses based on transformations of test inputs [12], but the defense was broken in only five days [1]. Therefore, there is need for a systematic method allowing one to handle this problem. The task of this report is to deal with such robustness certification problems in neural networks using polynomial optimization tools. In [17], the authors translate this kind of problem into a quadratically constrained quadratic program (QCQP) and used the semidefinite programming (SDP) relaxation (an SDP with larger feasible domain and easier to solve) to get an upper bound of the optimal value of QCQP [17]. In Section 2.2, we will see that the certification task is indeed to identify whether the optimal value of an optimization problem is negative or not, so that if we get a negative upper bound of the optimal value, we could still certify robustness. What we use in this report is actually also a method relying on SDP, but with a finer approximation and convergence garantee. We propose a hierarchy of SDPs which gives a better upper bound than in [17] if the size of the SDP constraints is large enough.

1.1 Motivation

Although artificial neural network is powerful in many areas, the classifiers have been shown in some cases to fail in the presence of small adversarial perturbations of inputs [5]. A typical example is displayed in Figure 1, the original picture is a panda, we add a tiny perturbation to it and get a new picture which is visually exactly the same (a panda). However, the classifier we trained says that the new picture is a gibbon with high confidence. Such adversarial examples make machine learning models vulnerable to attacks. For instance, a self-driving car crashes into another car because it ignores a stop sign. Someone had placed a picture over the sign, which looks like a stop sign with a little dirt for humans, but was designed to look like a parking prohibition sign for the sign recognition software of the car.

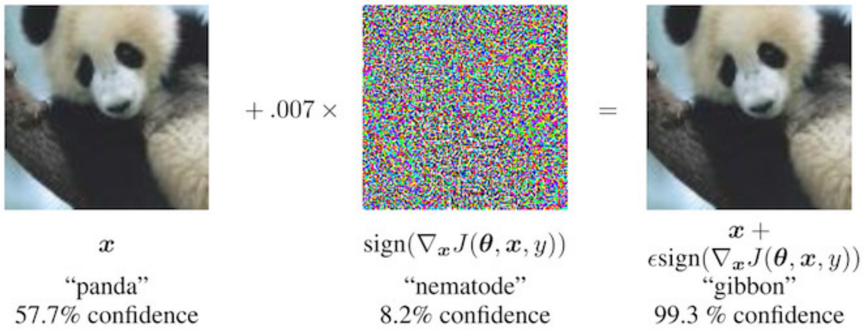

Figure 1: An adversarial example in deep neural network (Chapter 6.2, [14])

Since the discovery of such adversarial examples, it becomes critical that we examine not only whether the systems do not simply work "most of the time", but which are truly robust and reliable. Although many notions of robustness and reliability exist, one particular topic in this area that has raised a great deal of interest in recent years is that of adversarial robustness: can we train classifiers that are certifiably robust to all attacks within a fixed attack model? There are many approaches to certify robustness of neural networks, for many of which we construct convex relaxations in order to obtain an upper bound on the worst-case loss (which is the maximum of an optimization problem) over all valid attacks [2, 21, 20, 26, 25, 27, 29] - this upper bound serves as a certificate of robustness.

1.2 Related works

The mechanism to deal with such kind of certification problem is translating the ReLU equality (ReLU is the activation function in the neural network, $\text{ReLU}(x) = \max\{x, 0\}$) into a system of polynomial equations. In this report, we mainly refer to [17], where the authors use SDP relaxation to get an upper bound approximation of the robustness certification problems. The reason why people often relax a non-convex optimization problem to an SDP is because there are many existing solvers and toolboxes in different computer languages, which provide us an efficient way to solve it. The authors in [17] use the YALMIP toolbox [11] with MOSEK solver [15] to solve this convex program on a 4-core CPU, and compare the results with different robustness training techniques (SDP-cert [17], LP-cert [28], Grad-cert [16]) in different neural networks (Grad-NN [16], LP-NN [28], PGD-NN [13]). The SDP-cert provides non-vacuous certificates for all networks considered, and consistently performs better than both LP-cert and Grad-cert for all three networks.

1.3 Our contribution

We will see later that the SDP relaxation in [17] is in fact the first order moment relaxation in the framework of Lasserre's hierarchy [6]. In our work, we will do the similar comparison as in [17], the improvement is that we will use a higher order SDP relaxation to get a finer bound on the optimization problem. So that if we increase the order of moment relaxation, we will always get a better result, the only problem is the computational cost when we deal with networks with large number of neurons (for example, in [17], the authors use neural networks with more than 500 neurons). That is the reason why we need sparsity patterns and propose structured SDP relaxations (the dense relaxation 3.2, the sparse relaxation 3.3, the squared-constraint sparse relaxation 3.4 and the heuristic relaxation 3.5).

For this report, we only consider fully-connected neural networks as discussed in [17], but more complicated networks such as convolutional neural networks (CNN) can be considered in the future work.

1.4 Successful applications

The idea to transform a polynomial optimization problem (POP) into an equivalent quadratic optimization problem can be traced back to Shor [19]. There are already many successful applications of Lasserre's hierarchy: the optimal power flow (OPF) problem [4], the max-cut problem [9], etc.

2 Preliminaries and Problem Statement

2.1 Fully-connected neural networks

Consider a fully-connected multi-layer neural network, where we are interested in classification. Suppose we have an **input** $x \in \mathbb{R}^{p_0}$ and L **hidden layers**. For each hidden layer $i = 1, \ldots, L$, we associate a triple of parameters (A_i, b_i, σ_i), where $A_i \in \mathbb{R}^{p_i \times p_{i-1}}$ is the so-called **weight matrix**, $b_i \in \mathbb{R}^{p_i}$ is the **bias**, and $\sigma_i : \mathbb{R}^{p_i} \to \mathbb{R}^{p_i}$ is the **activation function**. We denote by $x_i \in \mathbb{R}^{p_i}$ the **activation vector** at layer i, i.e. $x_i = \sigma_i(A_i x_{i-1} + b_i)$ with $x_0 := x$. Suppose we have k classes, we denote by $F(x) \in \mathbb{R}^k$ the vector in the **softmax layer** where the score of each label is assumed to be a linear combination of the last activation vector, i.e., $F(x)_j = c_j^T x_L$ for $j = 1, \ldots, k$ and for some $c_j \in \mathbb{R}^{p_L}$. The final **output** y is the label assigned to the input x with the highest score, i.e. $y = \arg\max_{j=1,\ldots,k} F(x)_j$. See Figure 2.

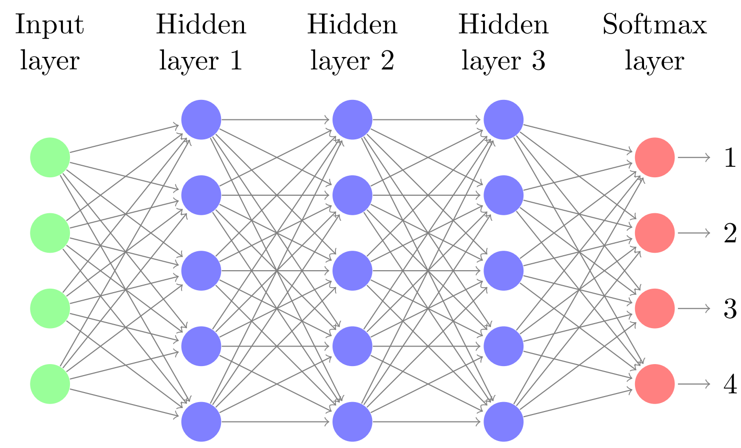

Figure 2: Fully-connected neural network, $L = 3, p_0 = 4, p_1 = p_2 = p_3 = 5$

2.2 Robustness margin certification problems (RMCP)

Now suppose we have already trained a network with parameters $W = (A_1, b_1; \ldots; A_L, b_L; c)$, here we set the activation functions as ReLU, i.e. $\sigma_i(x) = \text{ReLU}(x) = \max\{x, 0\}$. For a given input x, we denote by $F_W(x)$ the output of x with respect to the network W. Fixing the parameters (note that we do not care about the weights we obtained, they could be anything else), we want to study the robustness of this network with respect to the given input $\bar{x}$ and classification $\bar{y}$, that is, if we vary the input x in a small neighborhood of $\bar{x}$, will we get a different classification y? Or equivalently, for x close to $\bar{x}$, does there exist some $y \neq \bar{y}$ such that $F_W(x)_y > F_W(\bar{x})_{\bar{y}}$? In this paper, for simplicity, we focus on the inputs that are bounded in the l_∞ norm (i.e. $||x||_\infty = \max_i |x_i|$, but any other distance function using polynomial is possible). Let $\varepsilon > 0$, denote by $B(\bar{x}, \varepsilon)$ the ball centered at $\bar{x}$ with radius ε, then the problem is to solve the following optimization problem

$$l_y^*(\bar{x}, \bar{y}) = \max_{x \in B(\bar{x}, \varepsilon)} F_W(x)_y - F_W(\bar{x})_{\bar{y}}$$

for every $y \neq \bar{y}$. We say that a network is **certifiably robust** on $(\bar{x}, \bar{y})$ if $l_y^*(\bar{x}, \bar{y}) < 0$ for all $y \neq \bar{y}$. The **robustness margin certification problem (RMCP)** can be written as the following optimization problem

$$\begin{aligned} \max_x \quad & F_W(x)_y - F_W(\bar{x})_{\bar{y}} = (c_y - c_{\bar{y}})^T x_L \\ \text{s.t.} \quad & \begin{cases} x_i = \text{ReLU}(A_i x_{i-1} + b_i), i = 1, \ldots, L; \\ ||x - \bar{x}||_\infty \leq \varepsilon. \end{cases} \end{aligned}$$

2.3 Polynomial optimization problem (POP)

A general polynomial optimization problem in n variables takes the form

$$\begin{aligned} \inf_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \begin{cases} f_i(\mathbf{x}) \geq 0, i = 1, \ldots, p; \\ g_j(\mathbf{x}) = 0, j = 1, \ldots, q. \end{cases} \end{aligned} \tag{POP}$$

where f, $\{f_i\}_{i=1}^p$, $\{g_j\}_{j=1}^q$ are polynomials in n variables. This is a special case of nonlinear optimization problem where ojective function and all constraints are polynomials. In particular, by adding $g_j \geq 0$ and $-g_j \geq 0$, we may drop the equality constraints $g_j = 0$ and assume that we have only inequality constraints $f_i \geq 0$.

2.4 RMCP as a polynomial optimization problem (POP)

In [17], the authors set $b_i = 0$ and use the two following analytic equalities:

$$a = \max\{b, 0\} \Longleftrightarrow \begin{cases} a(a-b) = 0 \\ a \geq 0, a \geq b \end{cases} \tag{1}$$

$$b \leq a \leq c \Longleftrightarrow (a-b)(a-c) \leq 0 \Longleftrightarrow a^2 \leq (b+c)a - bc \tag{2}$$

so that the RMCP can be rewritten as

$$\begin{aligned}
\max \quad & c^T x_L \\
\text{s.t.} \quad & \begin{cases} x_i(x_i - A_i x_{i-1}) = 0, x_i \geq A_i x_{i-1}, x_i \geq 0, i = 1, \dots, L, \\ x_{i-1}^2 \leq (l_{i-1} + u_{i-1}) x_{i-1} - l_{i-1} u_{i-1}, i = 1, \dots, L. \end{cases}
\end{aligned} \tag{3}$$

where $l_0 = \bar{x} - \varepsilon, u_0 = \bar{x} + \varepsilon$ and $l_i = (A_i)_+ l_{i-1} - (A_i)_- u_{i-1}, u_i = (A_i)_+ u_{i-1} + (A_i)_- l_{i-1}$ for $i = 1, \dots, L-1$. In this paper, we also use the equivalence (1), while we keep the inequality $||x - \bar{x}||_\infty \leq \varepsilon$, so that we get another form of RMCP

$$\begin{aligned}
\max_{x,z} \quad & c^T x_L \\
\text{s.t.} \quad & \begin{cases} z_i - A_i x_{i-1} - b_i = 0, i = 1, \dots, L; \\ x_i(x_i - z_i) = 0, x_i \geq z_i, x_i \geq 0, i = 1, \dots, L; \\ -\varepsilon \leq x - \bar{x} \leq \varepsilon. \end{cases}
\end{aligned}$$

Here we introduce new variables z_i in order to create affine constraints $z_i - A_i x_{i-1} - b_i = 0$ and constraints $x_i(x_i - z_i) = 0, x_i \geq z_i, x_i \geq 0$ with only 2 variables.

2.5 Example of SDP-relaxation

This section is the SDP-relaxation the authors use in [17]. Notice that there are quadratic constraints $x_i(x_i - A_i x_{i-1}) = 0$ in RMCP, so that this problem is non-convex and NP-hard [10] in general. The main idea is to introduce an SDP relaxation which can be run efficiently with computer. Recall that an **semidefinite programming (SDP)** is a subfield of convex optimization concerned with the optimization of a linear objective function over the intersection of the cone of positive semidefinite matrices with an affine space. Here, a symmetric $n \times n$ real matrix M is said to be **positive semidefinite** if the scalar $z^T M z$ is strictly positive for every non-zero $z \in \mathbb{R}^n$. Generally, an SDP has the following form:

$$\begin{aligned}
\min_{X \in \mathbb{S}^n} \quad & \langle C, X \rangle_{\mathbb{S}^n} \\
\text{s.t.} \quad & \begin{cases} \langle A_k, X \rangle_{\mathbb{S}^n} \leq b_k, k = 1, \dots, m; \\ X \succeq 0. \end{cases}
\end{aligned}$$

where $\mathbb{S}^n$ denotes the the space of all $n \times n$ real symmetric matrices, and $\langle \cdot, \cdot \rangle_{\mathbb{S}^n}$ is the Frobenius product in $\mathbb{S}^n$. Let $v := [1\ x^T\ x_1^T\ \dots\ x_L^T]^T$ and $P := vv^T$, we use symbolic indexing $P(\cdot)$ to denote the sub-blocks of P, i.e.

$$P = \begin{bmatrix} P(1) & P(x^T) & P(x_1^T) & \dots & P(x_L^T) \\ P(x) & P(xx^T) & P(xx_1^T) & \dots & P(xx_L^T) \\ P(x_1) & P(x_1 x^T) & P(x_1 x_1^T) & \dots & P(x_1 x_L^T) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P(x_L) & P(x_L x^T) & P(x_L x_1^T) & \dots & P(x_L x_L^T) \end{bmatrix}$$

Then, the RMCP (3) is equivalent to

$$\begin{aligned}
\max_{P \in \mathbb{S}^n} \quad & c^T P(x_L) \\
\text{s.t.} \quad & \begin{cases} \operatorname{diag}(P(x_i x_i^T)) = \operatorname{diag}(A_i P(x_{i-1} x_i^T)), i = 1, \dots, L; \\ P(x_i) \geq A_i P(x_{i-1}), P(x_i) \geq 0, i = 1, \dots, L; \\ \operatorname{diag}(P(x_{i-1} x_{i-1}^T)) \leq (l_{i-1} + u_{i-1}) P(x_{i-1}) - l_{i-1} u_{i-1}, i = 1, \dots, L; \\ P(1) = 1, P \succeq 0, \boxed{\operatorname{rank}(P) = 1}. \end{cases}
\end{aligned}$$

The constraint $\operatorname{rank}(P) = 1$ makes the problem non-convex, and the typical way to relax it to a convex SDP is by dropping the rank-1 condition of P. Then the SDP relaxation of the RMCP (3) can be written in terms of the matrix P as

$$\max \quad c^T P(x_L)$$
$$\text{s.t.} \quad \begin{cases} \operatorname{diag}(P(x_i x_i^T)) = \operatorname{diag}(A_i P(x_{i-1} x_i^T)), i = 1, \dots, L; \\ P(x_i) \geq A_i P(x_{i-1}), P(x_i) \geq 0, i = 1, \dots, L; \\ \operatorname{diag}(P(x_{i-1} x_{i-1}^T)) \leq (l_{i-1} + u_{i-1}) P(x_{i-1}) - l_{i-1} u_{i-1}, i = 1, \dots, L; \\ P(1) = 1, P \succeq 0. \end{cases}$$

From now on, we call this SDP relaxation the **SDP-cert** relaxation in order to distinguish from the SDP relaxations we will introduce soon.

3 POP and further SDP Relaxations

Before introducing the SOS decomposition and the SDP relaxation, we give a brief overview of the relaxation methods we use in our optimization problem. As already discussed in the previous section, we have the RMCP

$$\max_{x,z} \quad c^T x_L$$
$$\text{s.t.} \quad \begin{cases} z_i - A_i x_{i-1} - b_i = 0, i = 1, \dots, L; \\ x_i(x_i - z_i) = 0, x_i \geq z_i, x_i \geq 0, i = 1, \dots, L; \\ -\varepsilon \leq x - \bar{x} \leq \varepsilon. \end{cases} \tag{RMCP}$$

For the following discussion, we always refer to the above (RMCP) as the **robustness margin certification problem**. We also use an equivalent version of the RMCP, where we introduce squared constraints to replace the affine equalities:

$$\max_{x,z} \quad c^T x_L$$
$$\text{s.t.} \quad \begin{cases} ||z_i - A_i x_{i-1} - b_i)||^2 = 0, i = 1, \dots, L; \\ x_i(x_i - z_i) = 0, x_i \geq z_i, x_i \geq 0, i = 1, \dots, L; \\ -\varepsilon \leq x - \bar{x} \leq \varepsilon. \end{cases} \tag{SC-RMCP}$$

and we call this problem **squared constraint RMCP** (SC-RMCP). We explore some sparsity patterns for each of the two (RMCP) and (SC-RMCP) models: the **dense relaxation (DR)** is the method without sparsity [6], the **sparse relaxation (SR)** is the method with sparsity (which satisfies the running intersection property, see Section 3.3) [7], the **squared-constraint sparse relaxation (SCSR)** is a variant of sparse relaxation which reduces the order of localization matrices (see Section 3.1) of the constraints with large number of variables, and the **heuristic relaxation (HR)** is the method we proposed in order to handle this specific RMCP. The subsections describing the framework of these methods are indicated in the following table:

	Dense	Sparse	SC-Sparse	Heuristic
RMCP	3.2	3.3	-	-
SC-RMCP	-	-	3.4	3.5

We list the computational cost and convergence behavior of each method, to give to the reader a general intuition why we need to explore sparsity in the relaxation methods. In the table, for general POP, n denotes the number of variables in the model, d denotes the order of the relaxation methods (see Section 3.1), p denotes the number of inequality constraints, q denotes the number of equality constraints. Sparsity in POP relies on separability patterns of the constraints encoded by a set of cliques (or subsets) of variables. We denote by $I_k \subseteq \{1, \dots, n\}$ the clique number k, for $k = 1, \dots, m$, where m denotes the number of cliques. We use $n_k := |I_k|$ for the cardinality of clique number k. For instance, $f = x_1 x_2 + x_2 x_3$, here $n = 3$, $m = 2$, we can define the cliques $I_1 = \{1, 2\}$, $I_2 = \{2, 3\}$, and then $x_1 x_2 \in \mathbb{R}[\mathbf{x}_{I_1}], x_2 x_3 \in \mathbb{R}[\mathbf{x}_{I_2}]$. We translates these quantities in the context of RMCP problems for a given deep network architecture, p_i denotes the dimension of the i-th activation vector in our neural network, for $i = 1, \dots, L$ where L denotes the number of layers in the network.

Relax. methods	for general POP		for RMCP		Conv.
	size of SDP	# of SDP cstr.	size of SDP	# of SDP cstr.	
Dense	$\approx d^n$	$\approx p+q$	$\approx d^{\sum_i p_i}$	$\approx 4\sum_i p_i + p_0$	yes
Sparse	$\approx \max_k d^{n_k}$	$\approx m+p+q$	$\approx \max_i d^{p_i}$	$\approx 4\sum_i p_i + p_0 + L$	yes (RIP)
SC-sparse	$\approx \max_k d^{n_k}$	$\approx m+p$	$\approx \max_i d^{p_i}$	$\approx 3\sum_i p_i + p_0 + L$	not known
Heuristic	$\leq \max_k d^{n_k}$	$\geq m+q$	$\approx d^2$	$\leq \sum_i p_i^2 + 3\sum_i p_i + p_0$	not known

The size of SDP constraint is a computational bottleneck. For example, for a positive semidefinite matrix $P \in \mathbb{R}^{n\times n}$, the size of P is n, and we have n^2 unknown variables in P. Normally, when the size of SDP matrices are larger than 10^4, a typical laptop has a severe burden on solving SDP. For the netwoks considered in [17], we have more than 500 variables ($n \geq 500$), so that even for $d = 2$, it is out of reach. The game of using sparsity is to trade off between the size of SDP constraints and the number of SDP constraints. For dense relaxation, the computation is out of reach even for order 2, so that sparsity is really needed in implementation. But for fully-connected networks, the sparse relaxation method proposed in [7] still requires a lot of computation, we have to push it further (squared-constraint sparse and heuristic relaxation).

In our framework, we actually use a system of SDPs to approach the exact value. The convergence here means that the optimal value of the system of SDPs converges to the exact solution of the original problem. Otherwise, it still converges but to some value different from the exact solution. By adding localization constraints ($M^2 - ||x||^2 \geq 0$), the dense relaxation is convergent, we refer to [6] for its proof; under RIP condition, the sparse relaxation is also convergent, we refer to [7, 22] for its proof, [23] for the SparsePOP software, and [3] for the Gloptipoly software; the convergence proof of squared-constraint relaxation is stated in Appendix 7.1. Beyond convergence, all the described relaxations provide valid numerical upper bounds for the RMCP problem and hence can be used for certification of adversarial robustness, even though we do not have a proof for squared-constraint sparse relaxation and heuristic relaxation.

Now we start the story of SOS and moment relaxations.

3.1 Duality between sum-of-square (SOS) and SDP

Positivity certificates

Suppose we have a polynomial in n variables $f \in \mathbb{R}[\mathbf{x}]$, how can we check if $f(x) \geq 0$ for all $x \in \mathbb{R}^n$? Assume that f can be written as a **sum-of-square (SOS)** of polynomials, i.e. $f = \sum_{i=1}^k f_i^2$, then it is obvious that f is always non-negative. The converse is not always true: a non-negative polynomial cannot always be written as a sum of squares. A famous example is the **Motzkin polynomial** $p(x, y) = x^4y^2 + x^2y^4 - 3x^2y^2 + 1$, $p(x, y)$ is non-negative but it is not a sum of squares of polynomials [18]. Nevertheless, we may find some positivity certificates on a compact semi-algebraic set (a subset of $\mathbb{R}^n$ which is defined by several polynomials). Suppose we have $K \subseteq \mathbb{R}^n$ defined by

$$K := \{x \in \mathbb{R}^n : g_j(x) \geq 0, j = 1, \ldots, m\} \tag{4}$$

which is compact, we introduce the following certificate due to Schmüdgen:

Theorem 3.1. ***(Schmüdgen's Positivstellensatz [8, Theorem 2.13])*** *Let $K \subseteq \mathbb{R}^n$ be as in (4). If $f \in \mathbb{R}[\mathbf{x}]$ is strictly positive on K, then*

$$f = \sum_{J\subseteq\{1,\ldots,m\}} \sigma_J g_J$$

where σ_J is SOS and $g_J = \prod_{j\in J} g_j$ for all $J \subseteq \{1, \ldots, m\}$.

The drawback of Schmüdgen's Positivstellensatz is that there are exponentially many SOS terms, actually, under certain assumptions, we can improve Schmüdgen's results. First, let us introduce some notations in order to simplify the statements. Associated with the family $g_j \subseteq \mathbb{R}[\mathbf{x}]$, define

$$\begin{aligned}
\Sigma[\mathbf{x}] &= \left\{ \sum_{i=1}^{k} f_i^2, f_i \in \mathbb{R}[\mathbf{x}], i = 1, \ldots, k \right\} \\
\Sigma_d[\mathbf{x}] &= \left\{ \sum_{i=1}^{k} f_i^2, f_i \in \mathbb{R}[\mathbf{x}], deg(f_i) \leq d, i = 1, \ldots, k \right\} \\
Q(g) &:= \left\{ \sigma_0 + \sum_{j=1}^{m} \sigma_j g_j : \sigma_j \in \Sigma[\mathbf{x}], j = 0, \ldots, m \right\} \\
Q_d(g) &:= \left\{ \sigma_0 + \sum_{j=1}^{m} \sigma_j g_j : \sigma_j \in \Sigma_d[\mathbf{x}], j = 0, \ldots, m \right\}
\end{aligned}$$

Assumption 1. *There exists $u \in Q(g)$ such that $\{x \in \mathbb{R}^n : u(x) \geq 0\}$ is compact.*

Theorem 3.2. ***(Putinar's Positivstellensatz [8, Theorem 2.15])*** *Let $K \subseteq \mathbb{R}^n$ be as in (4). Under Assumption 1, if $f \in \mathbb{R}[\mathbf{x}]$ is strictly positive on K, then $f \in Q(g)$, i.e.*

$$f = \sigma_0 + \sum_{j=1}^{m} \sigma_j g_j \,,$$

where $\sigma_j \in \Sigma[\mathbf{x}], j = 0, \ldots, m$.

In particular, if we take $u_M(x) = M^2 - ||x||^2$, with $M = \sup\{||x|| : x \in K\}$, add u_M to the SOS cone $Q(g)$, then u_M satisfies Assumption 1. For a compact set K, we can always ensure that $u_M \in Q(g)$, yielding the following corollary:

Corollary 3.3. *Let $K \subseteq \mathbb{R}^n$ be as in (4). Select some $M > 0$ such that $||x^*|| \leq M$ and assume that $g_1(x) = M^2 - ||x||^2$. If $f \in \mathbb{R}[\mathbf{x}]$ is positive on K, then $f \in Q(g)$, i.e.*

$$f = \sigma_0 + \sum_{j=1}^{m} \sigma_j g_j \,,$$

for some $\sigma_j \in \Sigma[\mathbf{x}], j = 0, \ldots, m$.

Lasserre's Hierarchy[6]

Computing $f^* = \inf_{x \in K} f(x)$ is equivalent to solve $\sup\{t \in \mathbb{R} : f(x) - t \geq 0, \forall x \in K\}$. The idea is then to apply Corollary 3.3 to reformulate this problem as

$$\inf_{x \in K} f(x) = \sup\{t : f - t \in Q(g)\} \,. \tag{5}$$

Let us consider the following relaxation

$$(f_d^M)^* = \sup\{t : f - t \in Q_d(g)\} \tag{6}$$

of (5) and show that this relaxation is an SDP.
Let $f(x) = \sum_{\alpha \in \mathbb{N}_r^n} f_\alpha x^\alpha$ be a polynomial of degree r with n variables, define $\mathbf{f} = (f_\alpha)_{\alpha \in \mathbb{N}_r^n}$ and

$$v_r(x) = [1, x_1, x_2, \ldots, x_n, x_1^2, x_1 x_2, \ldots, x_n^2, \ldots, x_n^r]^T$$

be the vector of basis in $\mathbb{R}[\mathbf{x}]^{s(r)}$, $s(r) = \binom{n+r}{r}$, then $f(x) = \mathbf{f}^T v_r(x)$. An SOS polynomial of degree at most $2d$ can be written as $\sigma = v_d(x)^T Q v_d(x)$ with $Q \succeq 0$. Then (5) can be written as

$$\begin{aligned}
\sup_{t, Q_j} \quad & t \\
\text{s.t.} \quad & \begin{cases} \mathbf{f}^T v_d(x) - t = v_d(x)^T Q_0 v_d(x) + \sum_{j=1}^m g_j(x) v_{d-\omega_j}(x)^T Q_j v_{d-\omega_j}(x); \\ Q_j \succeq 0, j = 0, \ldots, m. \end{cases}
\end{aligned}$$

where $\omega_j = deg(g_j)/2$. let $\mathbf{y} = \{y_i\}_{i=1}^{+\infty}$ be the infinite sequence with $y_1 = 1$, and $\{y_i\}_{i=1}^{s(d)} = v_d(x)$ for all $d \geq 1$. Define the **moment matrix** $M_d(\mathbf{y}) = v_d(x)v_d(x)^T$, and the **localization matrix** (of g_j) $M_{d-\omega_j}(g_j\mathbf{y}) = g_j(x)v_{d-\omega_j}(x)v_{d-\omega_j}(x)^T$ where $\omega_j = deg(g_j)/2$. Notice that

$$v_d(x)^T Q_0 v_d(x) = \langle Q_0, v_d(x)v_d(x)^T\rangle = \langle Q_0, M_d(\mathbf{y})\rangle$$
$$g_j(x)v_{d-\omega_j}(x)^T Q_j v_{d-\omega_j}(x) = \langle Q_j, g_j(x)v_{d-\omega_j}(x)v_{d-\omega_j}(x)^T\rangle = \langle Q_j, M_d(g_j\mathbf{y})\rangle$$

so that we get an SDP problem

$$\begin{aligned}\sup_{t,Q} \quad & t \\ \text{s.t.} \quad & \begin{cases}\mathbf{f}^T\mathbf{y} - t = \langle Q_0, M_d(\mathbf{y})\rangle + \sum_{j=1}^m \langle Q_j, M_{d-\omega_j}(g_j\mathbf{y})\rangle; \\ \mathbf{y}(1) = 1, Q_j \succeq 0, j = 0,\ldots,m.\end{cases}\end{aligned}$$

This is exactly the dual of SDP problem

$$\begin{aligned}\inf_{y} \quad & \mathbf{f}^T\mathbf{y} \\ \text{s.t.} \quad & \begin{cases}M_d(\mathbf{y}) \succeq 0, \mathbf{y}(1) = 1; \\ M_{d-\omega_j}(g_j\mathbf{y}) \succeq 0, j = 1,\ldots,m.\end{cases}\end{aligned}$$

we call the above system **Lasserre's hierarchy** of SDP relaxations. In particular, if we set $d = 1$, i.e. we obtain the so-called **first order moment relaxation**, yielding

$$\begin{aligned}\inf_{y} \quad & \mathbf{f}^T\mathbf{y} \\ \text{s.t.} \quad & \begin{cases}M_1(\mathbf{y}) \succeq 0, \mathbf{y}(1) = 1; \\ L_{\mathbf{y}}(g_j) \succeq 0, j = 1,\ldots,m.\end{cases}\end{aligned}$$

which is exactly the SDP relaxation in [17]. For the same initial problem, as long as we use a higher moment relaxation, we are sure that we may get a better result compared to SDP-cert method.

Define the **Riesz linear functional** of f as $L_{\mathbf{y}}(f) = \mathbf{f}^T\mathbf{y}$, and let

$$(\rho_d^M)^* = \inf\{L_{\mathbf{y}}(f) : \mathbf{y}(1) = 1, M_d(\mathbf{y}) \succeq 0, M_{d-\omega_j}(g_j\mathbf{y}) \succeq 0, j = 1,\ldots,m\} \tag{7}$$

with the dual

$$(f_d^M)^* = \sup\{t : f - t \in Q_d(g)\}\,. \tag{8}$$

By weak duality, we have $(f_d^M)^* \leq (\rho_d^M)^* \leq f^*$. From [6, Theorem 3.4], we can successively approximate the optimal value f^* of our original problem with Lasserre's hierarchy of SDP relaxations.

Theorem 3.4. *As $d \to +\infty$, we have $(\rho_d^M)^* \uparrow f^*$, and, when d is sufficiently large, there is no duality gap between the moment relaxation problem and its dual, i.e. $(f_d^M)^* = (\rho_d^M)^*$.*

Next, we consider different moment relaxations of (RMCP).

3.2 Dense relaxation (DR)[6]

First of all, we explain how the method applies in a general polynomial optimization problem as (POP). We have the following moment relaxation:

$$\begin{aligned}\inf_{\mathbf{y}} \quad & L_{\mathbf{y}}(f) \\ \text{s.t.} \quad & \begin{cases}M_d(\mathbf{y}) \succeq 0, \mathbf{y}(1) = 1; \\ M_{d-\omega_i}(f_i\mathbf{y}) \succeq 0, i = 1,\ldots,p; \\ M_{d-\tau_j}(g_j\mathbf{y}) = 0, j = 1,\ldots,q.\end{cases}\end{aligned}$$

In this case, the moment relaxation involves a positive semidefinite matrix $M_d(\mathbf{y})$ which is of size $\binom{n+d}{d} \approx d^n$ when the number of variables n is fixed. Thus, the computational cost increases polynomially with the relaxation order. Therefore, the dense relaxation is often out of reach especially when the order of relaxation is greater than 2. We need some sparsity structure to improve it.

3.3 Sparse relaxation (SR)[7]

Due to the computational complexity of dense relaxation, we need some sparsity property in our target POP. Assume that the set of variables in POP can be divided by several cliques, which are indexed by I_k, $k = 1, \cdots, m$, i.e. $\{1, \cdots, n\} = \cup_{k=1}^m I_k$. Each polynomial constraint involves only the variables indexed by a certain clique, i.e. $f_i \in \mathbb{R}[\mathbf{x}_{I_{k(i)}}]$ for some $k(i)$ and $g_j \in \mathbb{R}[\mathbf{x}_{I_{k(j)}}]$ for some $k(j)$. We assume that those I_k satisfy the **Running Intersection Property (RIP)**: for every $k = 1, \dots, m-1$,

$$I_{k+1} \cap \bigcup_{j=1}^{k} I_j \subseteq I_s, \text{ for some } s \leq k\,.$$

The RIP condition is used to ensure that the sparse relaxation is convergent. If the cliques do not satisfy RIP, we can still run the SDP relaxation on our computer and get results, the only problem is that the optimal value may not converge to the exact one.

Now we reformulate our problem with sparsity as

$$\begin{aligned} \inf_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \begin{cases} f_i(\mathbf{x}) \geq 0, i \in I_{k(i)}, i = 1, \dots, p; \\ g_j(\mathbf{x}) = 0, j \in I_{k(j)}, j = 1, \dots, q. \end{cases} \end{aligned}$$

with sparse moment relaxation

$$\begin{aligned} \inf_{\mathbf{y}} \quad & L_{\mathbf{y}}(f) \\ \text{s.t.} \quad & \begin{cases} M_d(\mathbf{y}, I_k) \succeq 0, \mathbf{y}(1) = 1, k = 1, \dots, m; \\ M_{d-\omega_i}(f_i\mathbf{y}, I_{k(i)}) \succeq 0, i \in I_{k(i)}, i = 1, \dots, p; \\ M_{d-\tau_j}(g_j\mathbf{y}, I_{k(j)}) = 0, j \in I_{k(j)}, j = 1, \dots, q. \end{cases} \end{aligned}$$

Apply the sparse relaxation on the full problem of RMCP

$$\begin{aligned} \max_{x,z} \quad & c^T x_L \\ \text{s.t.} \quad & \begin{cases} z_i - A_i x_{i-1} - b_i = 0, i = 1, \dots, L; \\ x_i(x_i - z_i) = 0, x_i \geq z_i, x_i \geq 0, i = 1, \dots, L; \\ -\varepsilon \leq x - \bar{x} \leq \varepsilon. \end{cases} \end{aligned}$$

Note that when adding the extra variables z_i, the result of the first order SDP relaxation is the same as when not adding these z_i (this statement requires proof, it will be added in the later work of my internship), so that we may use a higer order relaxation with this new formulation to compare the existed results. We define a sparsity pattern on it

$$\begin{aligned} I_i &= \{z_i^1, \dots, z_i^{p_i}; x_{i-1}^1, \dots, x_{i-1}^{p_{i-1}}\}, i = 1, \dots, L; \\ J_i &= \{z_i^1, \dots, z_i^{p_i}; x_i^1, \dots, x_i^{p_i}\}, i = 1, \dots, L. \end{aligned}$$

where

$$\begin{aligned} & c^T x_L \in \mathbb{R}[X_{J_L}]; \\ & z_i^j - A_i^{j,:} x_{i-1} - b_i^j \in \mathbb{R}[X_{I_i}], \forall j, i = 1, \dots, L; \\ & x_i^j(x_i^j - z_i^j), x_i^j - z_i^j, x_i^j \in \mathbb{R}[X_{J_i}], \forall j, i = 1, \dots, L; \\ & x^j - \bar{x}^j + \varepsilon, -x^j + \bar{x}^j + \varepsilon, \in \mathbb{R}[X_{I_1}], \forall j,. \end{aligned}$$

Unfortunately, even though the terms $x_i^j(x_i^j - z_i^j), x_i^j - z_i^j, x_i^j$ are extremely sparse, there's an affine constraint $z_i - A_i x_{i-1} - b_i = 0$ involving $p_{i-1} + 1$ variables where p_{i-1} is the dimension of the activation vector x_{i-1}. The size of the localizaiton matrix with respect to this constraint is $\binom{1 + p_{i-1} + d}{d} \approx d^{p_{i-1}}$. Normally, p_{i-1} can be very large (≥ 500), which gives severe burden to the solver. Therefore, we propose the following squared-constraint sparse relaxation and heuristic relaxation to improve it.

3.4 Squared-constraint sparse relaxation (SCSR)

The purpose of methods proposed in Subsection 3.4 and Subsection 3.5 is to solve problems with some sparsity patterns, but some of the constraints involve a large number of variables. Different from nearly-sparse method [24], our methods can be applied when more than one constraint violate the sparsity. Precisely, assume we have the problem

$$\begin{aligned}\inf_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \begin{cases} f_i(\mathbf{x}) \geq 0, i \in I_{k(i)}, i = 1, \ldots, p; \\ g_j(\mathbf{x}) = 0, j = 1, \ldots, q. \end{cases}\end{aligned}$$

where $g_j(\mathbf{x}) = \sum_{k=1}^{m} g_{j,k}(\mathbf{x}), g_{j,k} \in \mathbb{R}[X_{I_k}], j = 1, \ldots, q$. We assume that g_j involves a large number of variables, so that it destroys the sparsity. The problem is equivalent to

$$\begin{aligned}\inf_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \begin{cases} f_i(\mathbf{x}) \geq 0, i \in I_{k(i)}, i = 1, \ldots, p; \\ g_j(\mathbf{x})^2 = 0, j = 1, \ldots, q. \end{cases}\end{aligned}$$

by taking squares on each equality constraint. The idea is to replace the localization matrix of the large-size constraints $M_{d-\tau_i}(g_i^2 y) \succeq 0$ by moment constraints $L_{\mathbf{y}}(g_i^2) \geq 0$, i.e. we fix the order of this localization matrix to be 1.

For the squared-constraint sparse relaxation, we don't apply finer sparsity in the moment constraints, i.e. we consider the moment relaxation problem with the same sparsity pattern as in sparse relaxation:

$$\begin{aligned}\inf_{\mathbf{y}} \quad & L_{\mathbf{y}}(f) \\ \text{s.t.} \quad & \begin{cases} M_d(\mathbf{y}, I_k) \succeq 0, \mathbf{y}(1) = 1, k = 1, \ldots, m; \\ M_{d-\omega_i}(f_i\mathbf{y}, I_{k(i)}) \succeq 0, i \in I_{k(i)}, i = 1, \ldots, p; \\ L_{\mathbf{y}}(g_i^2) = 0, j = 1, \ldots, q. \end{cases}\end{aligned}$$

Lagrangian

$$\begin{aligned} L(\mathbf{y}, \mathbf{X}, \mu) &= \langle f, \mathbf{y} \rangle - \langle X_0, M_d(\mathbf{y}) \rangle - \sum_{i=1}^{p} \langle X_i, M_{d-\omega_i}(f_i \mathbf{y}) \rangle + \sum_{j=1}^{q} \mu_j \langle g_j, \mathbf{y} \rangle \\ &= \sum_{\alpha} f_\alpha y_\alpha - \langle X_0, \sum_{\alpha} B_\alpha y_\alpha \rangle - \sum_{i=1}^{p} \langle X_i, \sum_{\alpha} C_{i,\alpha} y_\alpha \rangle + \sum_{j=1}^{q} \mu_j \sum_{\alpha} g_{j,\alpha} y_\alpha \\ &= \sum_{\alpha \neq 0} \Big(f_\alpha - \langle X_0, B_\alpha \rangle - \sum_{i=1}^{p} \langle X_i, C_{i,\alpha} \rangle + \sum_{j=1}^{q} \mu_j g_{j,\alpha} \Big) y_\alpha \\ &\qquad - \langle X_0, B_0 \rangle - \sum_{i=1}^{p} \langle X_i, C_{i,0} \rangle + \sum_{j=1}^{q} \mu_j g_{j,0} \end{aligned}$$

$$\inf_{\mathbf{y}} L(\mathbf{y}, \mathbf{X}, \mu) = \begin{cases} -\langle X_0, B_0 \rangle - \sum_{i=1}^{p} \langle X_i, C_{i,0} \rangle + \sum_{j=1}^{q} \mu_j g_{j,0}, \\ \qquad \text{if } f_\alpha + \sum_{j=1}^{q} \mu_j g_{j,\alpha} = \langle X_0, B_\alpha \rangle + \sum_{i=1}^{p} \langle X_i, C_{i,\alpha} \rangle, \alpha \neq 0; \\ -\infty, \qquad \text{otherwise.} \end{cases}$$

and the SDP form of its dual problem is

$$\begin{aligned}\sup_{\mathbf{X},\mu} \quad & \lambda := -\sum_{k=1}^{p} \langle X_{0,k}, B_{k,0} \rangle - \sum_{k=1}^{p} \sum_{i \in I_k} \langle X_{i,k}, C_{i,k,0} \rangle + \sum_{j=1}^{q} \mu_j g_j(0) \\ s.t. \quad & \begin{cases} f_\alpha + \sum_{j=1}^{q} \mu_j g_{j,\alpha} = \sum_{k:\, supp(\alpha) \in I_k} \langle X_{0,k}, B_{k,\alpha} \rangle + \sum_{k:\, supp(\alpha) \in I_k} \sum_{i \in I_k} \langle X_{i,k}, C_{i,k,\alpha} \rangle, \alpha \neq 0; \\ X_{0,k} \succeq 0, X_{i,k} \succeq 0, i \in I_k, k = 1, \ldots, p; \\ \mu_j \in \mathbb{R}, j = 1, \ldots, q. \end{cases}\end{aligned}$$

Reformulization

$$\sum_\alpha f_\alpha x^\alpha + \sum_{j=1}^q \mu_j \sum_\alpha g_{j,\alpha} x^\alpha = \langle X_0, \sum_\alpha B_\alpha x^\alpha \rangle + \sum_{i=1}^p \langle X_i, \sum_\alpha C_{i,\alpha} x^\alpha \rangle$$
$$- \langle X_0, B_0 \rangle - \sum_{i=1}^p \langle X_i, C_{i,0} \rangle + \sum_{j=1}^q \mu_j g_{j,0}$$
$$\Rightarrow f(\mathbf{x}) + \sum_{j=1}^q \mu_j g_j(\mathbf{x}) = \langle X_0, \mathbf{v}_d(\mathbf{x})\mathbf{v}_d(\mathbf{x})^T \rangle + \sum_{i=1}^p \langle X_i, f_i(\mathbf{x})\mathbf{v}_d(\mathbf{x})\mathbf{v}_d(\mathbf{x})^T \rangle + \lambda$$
$$\Rightarrow f(\mathbf{x}) - \lambda = \sigma_0(\mathbf{x}) + \sum_{i=1}^p f_i(\mathbf{x})\sigma_i(\mathbf{x}) - \sum_{j=1}^q \mu_j g_j(\mathbf{x}), \quad \sigma_0, \sigma_i \in \Sigma_d, \mu_j \in \mathbb{R}$$

as well as the SOS form

$$\sup_{\sigma,\mu} \quad \lambda$$
$$s.t. \quad \begin{cases} f(\mathbf{x}) - \lambda = \sum_{k=1}^p \left(\sigma_{0,k}(\mathbf{x}) + \sum_{i\in I_k} \sigma_{i,k}(\mathbf{x}) f_i(\mathbf{x})\right) + \sum_{j=1}^q \mu_j g_j(\mathbf{x}); \\ \sigma_{0,k}, \sigma_{i,k} \in \Sigma_d[X_{I_k}], i \in I_k, k = 1, \dots, p; \\ \mu_j \in \mathbb{R}, j = 1, \dots, q. \end{cases}$$

Different from sparse relaxation, we change the affine constraints $z_i - A_i x_{i-1} - b_i = 0$ to the quadratic constraints $||z_i - A_i x_{i-1} - b_i||^2 = 0$ within (SC-RMCP), in order to create new cliques of size 2 that will be helpful to our modeling later (see Subsection 3.5). We use the fact that $a = b = 0$ if and only if $a^2 = b^2 = 0$ if and only if $a^2 + b^2 = 0$.

$$\max_{x,z} \quad c^T x_L$$
$$\text{s.t.} \quad \begin{cases} \sum_{i=1}^L ||z_i - A_i x_{i-1} - b_i||^2 = 0, & \text{(union version)} \\ ||z_i - A_i x_{i-1} - b_i||^2 = 0, i = 1, \dots, L, & \text{(separate version)} \\ x_i(x_i - z_i) = 0, x_i \geq z_i, x_i \geq 0, i = 1, \dots, L; \\ -\varepsilon \leq x - \bar{x} \leq \varepsilon. \end{cases}$$

The corresponding g_i is $||z_i - A_i x_{i-1} - b_i||^2$ (for the separate version) or $\sum_{i=1}^L ||z_i - A_i x_{i-1} - b_i||^2$ (for the union version). We propose the following sparsity pattern

$$I_i^j = \{z_i^j; x_{i-1}^1, \dots, x_{i-1}^{p_{i-1}}\}, i = 1, \dots, L, j = 1, \dots, p_i;$$
$$J_i^j = \{z_i^j; x_i^1, \dots, x_i^{p_i}\}, i = 1, \dots, L, j = 1, \dots, p_i.$$

where

$$c^T x_L \in \mathbb{R}[X_{J_L^1}];$$
$$z_i^j - A_i^{j,:} x_{i-1} - b_i \in \mathbb{R}[X_{I_i^j}], \forall j, i = 1, \dots, L;$$
$$x_i^j(x_i^j - z_i^j), x_i^j - z_i^j, x_i^j \in \mathbb{R}[X_{J_i^j}], \forall j, i = 1, \dots, L;$$
$$x^j - \bar{x}^j + \varepsilon, -x^j + \bar{x}^j + \varepsilon, \in \mathbb{R}[X_{I_1^1}], \forall j.$$

this method contains a linear constraint with $1 + p_{i-1}$ variables which runs faster than the sparse relaxation. However, we can even do better by expanding the quadratic terms further.

3.5 Heuristic relaxation (HR)

The heuristic relaxation handles the same problem as squared-constraint sparse relaxation:

$$\inf_{\mathbf{x}} \quad f(\mathbf{x})$$
$$\text{s.t.} \quad \begin{cases} f_i(\mathbf{x}) \geq 0, i \in I_{k(i)}, i = 1, \dots, p; \\ g_j(\mathbf{x})^2 = 0, j = 1, \dots, q. \end{cases}$$

where $g_j(\mathbf{x}) = \sum_{k=1}^m g_{j,k}(\mathbf{x}), g_{j,k} \in \mathbb{R}[X_{I_k}], j = 1,\dots,q$. The only difference is that in the moment relaxation, we add the moment of $g_{j,k}$ with respect to each clique separately,

$$\inf_{\mathbf{y}} \quad L_{\mathbf{y}}(f)$$
$$\text{s.t.} \quad \begin{cases} M_d(\mathbf{y}, I_k) \succeq 0, \mathbf{y}(1) = 1, k = 1,\dots,p; \\ M_{d-\omega_i}(f_i\mathbf{y}, I_k) \succeq 0, i \in I_k, k = 1,\dots,p; \\ \sum_{k=1}^p L_{\mathbf{y}}(g_{j,k}, I_k) = 0, j = 1,\dots,q. \end{cases}$$

Now in the moment problem, for the constraints involving $1 + p_{i-1}$ variables, we only have linear moment constraints with cliques of size 2. A direct advantage of this trick is that it reduces greatly the computation time, so that it is potential to deal with large neural networks.

When applying heuristic relaxation to (SC-RMCP),

$$\max_{x,z} \quad c^T x_L$$
$$\text{s.t.} \quad \begin{cases} \sum_{i=1}^L ||z_i - A_i x_{i-1} - b_i||^2 = 0, & \text{(union version)} \\ ||z_i - A_i x_{i-1} - b_i||^2 = 0, i = 1,\dots,L, & \text{(separate version)} \\ x_i(x_i - z_i) = 0, x_i \geq z_i, x_i \geq 0, i = 1,\dots,L; \\ -\varepsilon \leq x - \bar{x} \leq \varepsilon. \end{cases}$$

the corresponding g_i is $||z_i - A_i x_{i-1} - b_i||^2$ (for the separate version) or $\sum_{i=1}^L ||z_i - A_i x_{i-1} - b_i||^2$ (for the union version). We propose the following sparsity pattern

$$\begin{aligned} I_i^{j,k} &= \{z_i^j; x_{i-1}^k\}, i = 1,\dots,L, j = 1,\dots,p_i, k = 1,\dots,p_{i-1}; \\ J_{i-1}^{j,k} &= \{x_{i-1}^j, x_{i-1}^k\}, i = 1,\dots,L+1, j = 1,\dots,p_{i-1}, k = 1,\dots,p_{i-1}; \\ K_i^j &= \{z_i^j; x_i^j\}, i = 1,\dots,L, j = 1,\dots,p_i. \end{aligned}$$

for objective function, we have

$$c^T x_L = \sum_{i=1}^{p_L} c_i x_L^i = \underbrace{c_1 x_L^1}_{K_L^1} + \underbrace{c_2 x_L^2}_{K_L^2} + \cdots + \underbrace{c_{p_L} x_L^{p_L}}_{K_L^{p_L}}$$

for ReLU constraints and attack constraints, we have

$$\begin{aligned} &x_i^j(x_i^j - z_i^j), x_i^j - z_i^j, x_i^j \in K_i^j; \\ &x^j - \bar{x}^j + \varepsilon, -x^j + \bar{x}^j + \varepsilon \in I_1^{1,j}. \end{aligned}$$

for quadratic constraints, we expand the squares and classify the cliques explicitly

$$
\begin{aligned}
(z_i^j - A_i^{j,:}x_{i-1} - b_i^j)^2 = (z_i^j)^2 + \sum_{k=1}^{p_{i-1}}\sum_{l=1}^{p_{i-1}} A_i^{j,k}A_i^{j,l}x_{i-1}^k x_{i-1}^l + (b_i^j)^2 \\
+ 2\sum_{k=1}^{p_{i-1}} b_i^j A_i^{j,k} x_{i-1}^k - 2\sum_{k=1}^{p_{i-1}} A_i^{j,k} z_i^j x_{i-1}^k - 2b_i^j z_i^j \\
= \underbrace{(z_i^j)^2 + (b_i^j)^2 - 2b_i^j z_i^j - 2z_i^j A_i^{j,1}x_{i-1}^1 + (x_{i-1}^1)^2(A_i^{j,1})^2 + 2b_i^j A_i^{j,1}x_{i-1}^1}_{I_i^{j,1}} \\
\underbrace{-2z_i^j A_i^{j,2}x_{i-1}^2 + (x_{i-1}^2)^2(A_i^{j,2})^2 + 2b_i^j A_i^{j,2}x_{i-1}^2}_{I_i^{j,2}} \\
- \dots \underbrace{-2z_i^j A_i^{j,p_{i-1}}x_{i-1}^{p_{i-1}} + (x_{i-1}^{p_{i-1}})^2(A_i^{j,p_{i-1}})^2 + 2b_i^j A_i^{j,p_{i-1}}x_{i-1}^{p_{i-1}}}_{I_i^{j,p_{i-1}}} \\
+ \underbrace{2A_i^{j,1}A_i^{j,2}x_{i-1}^1 x_{i-1}^2}_{J_i^{1,2}} + \underbrace{2A_i^{j,1}A_i^{j,3}x_{i-1}^1 x_{i-1}^3}_{J_i^{1,3}} + \dots + \underbrace{2A_i^{j,1}A_i^{j,p_{i-1}}x_{i-1}^1 x_{i-1}^{p_{i-1}}}_{J_i^{1,p_{i-1}}} \\
+ \underbrace{2A_i^{j,2}A_i^{j,3}x_{i-1}^2 x_{i-1}^3}_{J_i^{2,3}} + \underbrace{2A_i^{j,2}A_i^{j,4}x_{i-1}^2 x_{i-1}^4}_{J_i^{2,4}} + \dots + \underbrace{2A_i^{j,2}A_i^{j,p_{i-1}}x_{i-1}^2 x_{i-1}^{p_{i-1}}}_{J_i^{2,p_{i-1}}} \\
+ \dots + \underbrace{2A_i^{j,p_{i-1}-1}A_i^{j,p_{i-1}}x_{i-1}^{p_{i-1}-1} x_{i-1}^{p_{i-1}}}_{J_i^{p_{i-1}-1,p_{i-1}}}
\end{aligned}
$$

In summary, for squared-constraint sparse relaxation and heuristic relaxation, we only consider the moment constraint (first-order of localization matrix) of the quadratic terms $||z_i - A_i x_{i-1} - b_i||^2$. No matter whether we consider the full clique or the separate couple of cliques, we save a lot of time in implementation.

4 Numerical Results

In this section, we apply the four methods stated before to some randomly generated neural networks, and compare the results with SDP-cert method in [17].

4.1 Review of the certification problem and the corresponding POP

First recall the original RMCP:

$$
\max_x \quad F_W(x)_y - F_W(\bar{x})_{\bar{y}} = (c_y - c_{\bar{y}})^T x_L
$$
$$
\text{s.t.} \quad \begin{cases} x_i = ReLU(A_i x_{i-1} + b_i), i = 1, \dots, L; \\ ||x - \bar{x}||_\infty \le \varepsilon. \end{cases}
$$

the reformulized RMCP:

$$
\max_{x,z} \quad c^T x_L
$$
$$
\text{s.t.} \quad \begin{cases} z_i - A_i x_{i-1} - b_i = 0, i = 1, \dots, L; \\ x_i(x_i - z_i) = 0, x_i \ge z_i, x_i \ge 0, i = 1, \dots, L; \\ -\varepsilon \le x - \bar{x} \le \varepsilon. \end{cases}
$$

and the squared-constraint RMCP (SC-RMCP):

$$\max_{x,z} \quad c^T x_L$$

$$\text{s.t.} \quad \begin{cases} ||z_i - A_i x_{i-1} - b_i||^2 = 0, i = 1, \ldots, L; & \text{(separate version)} \\ \sum_{i=1}^{L} ||z_i - A_i x_{i-1} - b_i||^2 = 0; & \text{(union version)} \\ x_i(x_i - z_i) = 0, x_i \geq z_i, x_i \geq 0, i = 1, \ldots, L; \\ -\varepsilon \leq x - \bar{x} \leq \varepsilon. \end{cases}$$

4.2 Experimental setting

For the implementation, we randomly generate neural networks in dimension 3, 4, 5, 6 with only one hidden layer ($L = 1$), set range of perturbation ε to 0, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 2e-1, 3e-1, 4e-1, 6e-1, 8e-1, 1, and do experiments with different relaxation methods of different orders d (dense relaxation: $d = 1, 2$; sparse relaxation: $d = 2$; squared-constraint sparse relaxation: $d = 2, 3, 4, 5$; heuristic relaxation: $d = 2$).

4.3 Results and discussion

Objective value of different methods in different dimensions (take dim. = 6 for an example)

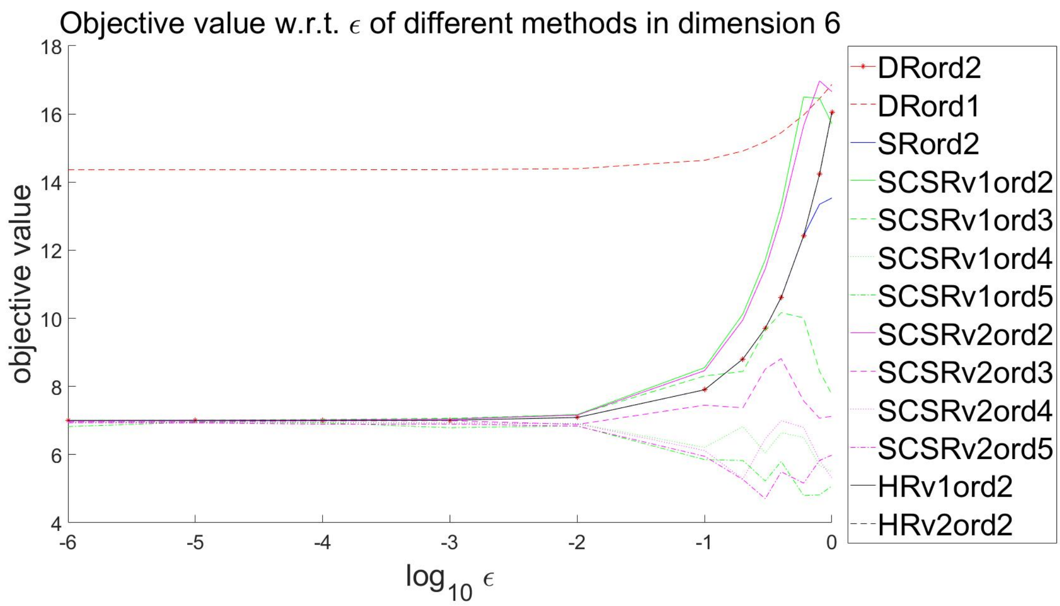

Figure 3: Objective value w.r.t. ε of different methods, $L = 1, p_0 = p_1 = 6$

In Figure 3, we use the log scale for ε. As we can see from the plot, when ε is small (for example, $\varepsilon < 0.01$, i.e. $\log_{10} \varepsilon \leq -2$, which represents the left part of Figure 3), there's only a tiny gap between the heuristic relaxation and the dense relaxation while the time reduction of heuristic method is remarkable (see Figure 5). As the dimension increases, at the same ε level, the gap between heuristic relaxation and the dense relaxation also increases, which is a bad sign for our implementation in networks with higher dimension. Nevertheless, from the preliminary experiments we have, when ε is small, the results of second-order heuristic relaxation (7.08, $\varepsilon = 0.01$) is **much** better than that of first-order dense relaxation, i.e. SDP-cert (14.4, $\varepsilon = 0.01$, the double of 7.08). This means we get a better results than in [17] in low dimension with less computation time. Also notice that sometimes there are **numerical issues**: we couldn't solve the problem or we couldn't tell whether the result we get is optimal or not . From Figure 3 we see that there are two factors that possibly cause numerical issues: one is the range of ε, as ε gets larger ($\varepsilon > 0.01$), numerical issue occurs in heuristic method; another one is dimension, as the dimension gets bigger, even at a level ε where numerical issues don't occur in lower dimension, we start observing some numerical issues, see Figure 4.

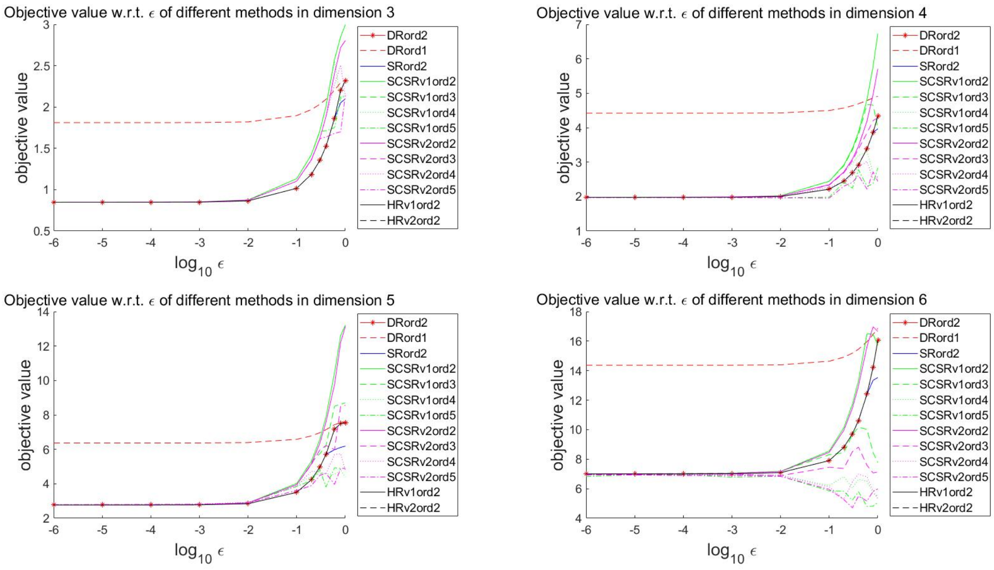

Figure 4: Running time w.r.t. ε of different methods in different dimensions

Average running time of different methods in different dimensions

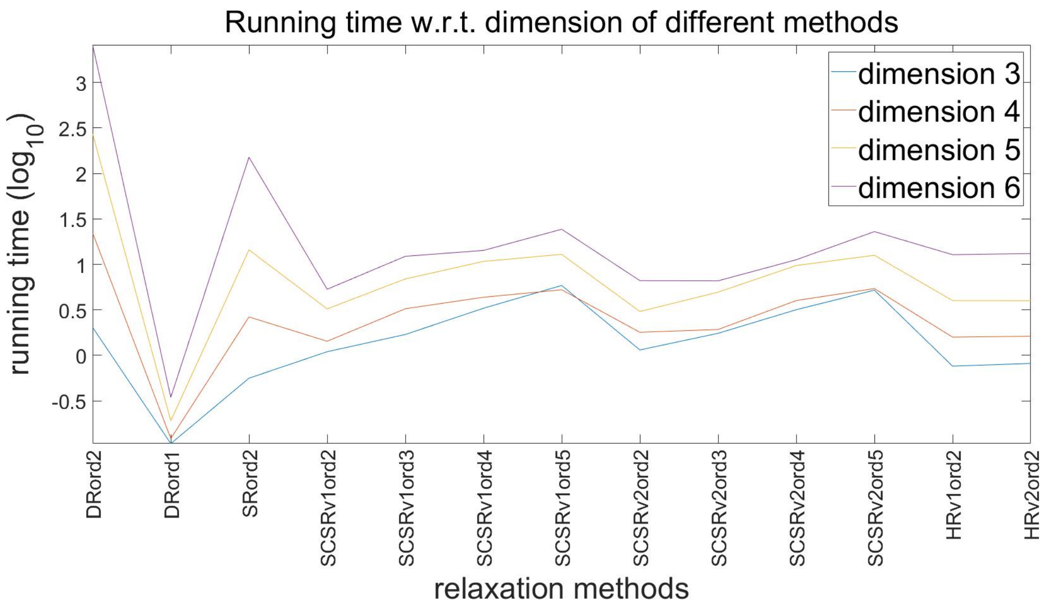

Figure 5: Running time of different methods in different dimensions

The timings in the plot are all log-scaled, which refers to the y-axis of Figure 5. From Figure 5, we can see that as the dimension increases, the computing timings of dense relaxation increase exponentially (from dimension 3 to 6, the log-scaled computation timings are 0.3, 1.3, 2.4, 3.4, respectively) while the heuristic relaxation increases nearly linearly, which allows us to deal with neural networks with more layers and more neurons. In the same dimension (remarkably in high dimension, take dim. = 6 for example), the computation timings of dense relaxation ($10^{3.4}$ seconds) and sparse relaxation ($10^{2.1}$ seconds) are very expensive, while the heuristic method is much cheaper ($10^{0.7} \approx 5$ seconds). Even though the heuristic relaxation still takes more time than first-order dense relaxation (SDP-cert, $10^{-0.46} \approx 0.34$ seconds), for ε small enough, the heuristic relaxation gets much better results (7.0, almost as good as second-order dense relaxation) than what SDP-cert gets (14.4, the double of 7), see Figure 3.

5 Conclusion and Future Work

5.1 Summary of our contribution

In this work, we apply moment relaxation methods to the RMCP certification problem arising in deep networks. We propose a heuristic method for the RMCP with specifically structured squared constraints. We compare the results of moment relaxations based on Lasserre's hierarchy with the SDP-cert method from [17]. Our numerical experiments indicate that the heuristic relaxation performs better than the other ones.

5.2 Interpretation of the result

The preliminary results for the small networks are exciting. From Figure 3 and Figure 5, we see that the heuristic method takes much less time but gets much better results, even though the result is for some naive neural networks. The only problem is that we need to be careful with the numerical issues when we increase the dimension, when we increase the order of our relaxation and when we enlarge the range of perturbation.

5.3 Future work

The next step of our work is to train networks as in [17] (Grad-NN, LP-NN, PGD-NN), apply our heuristic methods to them, and compare our performance with SDP-cert. If the heuristic methods performs well, it is possible to use such moment relaxation in broader optimization problems (with other tricks of course). For more complicated networks such as CNN, specific sparsity structures shall allow us to apply our SDP-based methods.

6 Acknowledgement

I would like to express my great appreciation to my three advisors: Edouard Pauwels, Victor Magron, and Jean-Bernard Lasserre, for accepting me as a member of their project, for teaching me how to do research and write report, and for taking me into the world of optimization. I also would like to thank Christophe Giraud, for giving me the opportunity to do Master in StatML.

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017.

[2] Akhilan Boopathy, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. Cnn-cert: An efficient framework for certifying robustness of convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3240–3247, 2019.

[3] Didier Henrion, Jean-Bernard Lasserre, and Johan Löfberg. Gloptipoly 3: moments, optimization and semidefinite programming. *Optimization Methods & Software*, 24(4-5):761–779, 2009.

[4] Cédric Josz. Application of polynomial optimization to electricity transmission networks. *PhD manuscript, arXiv preprint arXiv:1608.03871*, 2016.

[5] Zico Kolter and Aleksander Madry. Adversarial robustness - theory and practice. `https://adversarial-ml-tutorial.org/`, 2018.

[6] Jean-Bernard Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.

[7] Jean-Bernard Lasserre. Convergent sdp-relaxations in polynomial optimization with sparsity. *SIAM Journal on Optimization*, 17(3):822–843, 2006.

[8] Jean Bernard Lasserre. *An introduction to polynomial and semi-algebraic optimization*, volume 52. Cambridge University Press, 2015.

[9] Monique Laurent. Semidefinite relaxations for max-cut. In *The sharpest cut: The Impact of Manfred Padberg and his work*, pages 257–290. SIAM, 2004.

[10] Monique Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging applications of algebraic geometry*, volume 149 of *IMA Vol. Math. Appl.*, pages 157–270. Springer, New York, 2009.

[11] Johan Löfberg. Yalmip: A toolbox for modeling and optimization in matlab. In *Proceedings of the CACSD Conference*, volume 3. Taipei, Taiwan, 2004.

[12] Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*, 2017.

[13] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[14] Christoph Molnar. Interpretable machine learning. `https://christophm.github.io/interpretable-ml-book/`, 2019.

[15] APS Mosek. The mosek optimization software. *Online at `http://www.mosek.com`*, 54(2-1):5, 2010.

[16] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.

[17] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems*, pages 10877–10887, 2018.

[18] Bruce Reznick. Some concrete aspects of hilbert's 17th problem. *Contemporary mathematics*, 253:251–272, 2000.

[19] Naum Zuselevich Shor. *Nondifferentiable optimization and polynomial problems*, volume 24. Springer Science & Business Media, 2013.

[20] Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. In *Advances in Neural Information Processing Systems*, pages 10802–10813, 2018.

[21] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):41, 2019.

[22] Hayato Waki, Sunyoung Kim, Masakazu Kojima, and Masakazu Muramatsu. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization*, 17(1):218–242, 2006.

[23] Hayato Waki, Sunyoung Kim, Masakazu Kojima, Masakazu Muramatsu, and Hiroshi Sugimoto. Algorithm 883: Sparsepop—a sparse semidefinite programming relaxation of polynomial optimization problems. *ACM Transactions on Mathematical Software (TOMS)*, 35(2):15, 2008.

[24] Tillmann Weisser and Jean-Bernard Lasserre. Solving nearly sparse polynomial optimization problems. `http://homepages.laas.fr/tweisser/drupal/sites/homepages.laas.fr.tweisser/files/u107/nearly_sparse_final.pdf`, 2017.

[25] Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Duane Boning, Inderjit S Dhillon, and Luca Daniel. Towards fast computation of certified robustness for relu networks. *arXiv preprint arXiv:1804.09699*, 2018.

[26] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Aurelie Lozano, Cho-Jui Hsieh, and Luca Daniel. On extensions of clever: A neural network robustness evaluation algorithm. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1159–1163. IEEE, 2018.

[27] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578*, 2018.

[28] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv preprint arXiv:1711.00851*, 2017.

[29] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in neural information processing systems*, pages 4939–4948, 2018.

7 Appendix

7.1 Proof of convergence of squared-constraint sparse relaxation

Suppose we have a squared-constraint problem (without sparsity):

$$f^* = \inf_{\mathbf{x}} \quad f(\mathbf{x}) \\ \text{s.t.} \quad \begin{cases} f_i(\mathbf{x}) \geq 0, i = 1, \ldots, p; \\ ||Ax - b||^2 = 0. \end{cases} \tag{9}$$

and the moment relaxation:

$$\rho_d = \inf_{\mathbf{y}} \quad L_{\mathbf{y}}(f) \\ \text{s.t.} \quad \begin{cases} M_d(\mathbf{y}) \succeq 0, \mathbf{y}(1) = 1; \\ M_{d-\omega_i}(f_i \mathbf{y}) \succeq 0, i = 1, \ldots, p; \\ L_{\mathbf{y}}(||Ax - b||^2) = 0. \end{cases} \tag{10}$$

For $d \geq 1$, let $\mathbf{y}^d$ be an optimal solution of (10). We assume that $f_1(x) = 1 - ||\mathbf{x}||^2$ by normalization, and that $f_1(\mathbf{x})$ can be expressed as a weighted sum of squares with respect to f_i. Then we have $L_{\mathbf{y}^d}(f_1) \geq 0$, thus $\mathbf{y}^d$ is in the unit ball B_1 of the normed space l_∞ (by completing $\mathbf{y}^d$ with zeros to make it an infinite vector in l_∞). By Banach-Alaoglu Theorem, there exists a subsequence $\{d_k\}_{k\geq 1}$ and $\mathbf{y}^*$ such that $\mathbf{y}^{d_k} \to \mathbf{y}^*$ as $k \to +\infty$ for the weak-$*$ topology $\sigma(l_\infty, l_1)$ of l_∞. This means that $\lim_{k\to+\infty} y_\alpha^{d_k} = y_\alpha^*$ for all α. We need a lemma for the following discussion:

Lemma 7.1. ***(Dual side of Putinar's Theorem)*** *The sequence $\mathbf{y}$ has a representing measure supported on $K := \{f_j \geq 0, j = 1, \ldots, q\}$ if and only if for all $d \geq 1$, $M_d(\mathbf{y}) \succeq 0, M_{d-\omega_j}(f_j \mathbf{y}) \succeq 0$.*

Proof. (Sketch) The proof is based on Riesz-Haviland Theorem (Theorem 2.34, [8]), Putinar's Representing Theorem (Theorem 2.15, [8]) and the fact that $\langle \mathbf{u}, M_d(\mathbf{y})\mathbf{u}\rangle = L_{\mathbf{y}}(\mathbf{u}^2), \langle \mathbf{u}, M_d(h\mathbf{y})\mathbf{u}\rangle = L_{\mathbf{y}}(h\mathbf{u}^2)$ for all $\mathbf{u} \in \mathbb{R}[\mathbf{x}]$. The detailed proof can be referred to [8] (Theorem 2.44). □

Now, let us prove that $M_d(\mathbf{y}^*) \succeq 0$. By definition of $\mathbf{y}^d$, we have $M_{d_k}(\mathbf{y}^{d_k}) \succeq 0$ for all $k \geq 1$. Then for all $d \leq d_k$, $M_d(\mathbf{y}^{d_k}) \succeq 0$ since $M_d(\mathbf{y}^{d_k})$ is a submatrix of $M_{d_k}(\mathbf{y}^{d_k})$. Let $k \to +\infty$, we get $M_d(\mathbf{y}^*) \succeq 0$ by continuity of $M_d(\cdot)$. Similarly, we can prove that $M_{d-\omega_i}(f_i \mathbf{y}^*) \succeq 0$. By lemma 7.1, $\mathbf{y}^*$ has a representing measure ϕ supported on $\Gamma := \{f_i \geq 0, i = 1, \ldots, p\}$, but then we have

$$0 = \lim_{k\to+\infty} L_{\mathbf{y}^{d_k}}(||Ax - b||^2) = L_{\mathbf{y}^*}(||Ax - b||^2) = \int_\Gamma ||Ax - b||^2 d\phi$$

Therefore, ϕ is also supported on $\{Ax - b = 0\}$, thus supported on $\Omega = \Gamma \cap \{Ax - b = 0\}$. Let f^* be an optimal value of (9), we have

$$f^* \geq \lim_{k\to+\infty} L_{\mathbf{y}^{d_k}}(f) = \int_\Omega f d\phi \geq f^*$$

which means $\lim_{k\to+\infty} \rho_{d_k} = f^* = \int_\Omega f d\phi$. In particular, if the global minimizer $\mathbf{x}^*$ is unique, then $\phi = \delta_{x^*}$ and $\lim_{d\to+\infty} \rho_d = f^*$.

7.2 Numerical results of Section 4

This section is the full information of all the preliminary experimental results in Section 4, where we randomly generate network parameters in dimension 3, 4, 5, 6 and set range of perturbation ε to 0, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 2e-1, 3e-1, 4e-1, 6e-1, 8e-1, 1, and do experiments with different parameters (dimension, ε, order, etc.).

Notations

- Ord. = relaxation order;
- ε = range of perturbation;

- Res. = results;
- obj. = objective value;
- stat. = status of solution (1 means that optimal value is extracted, 0 means that optimal value cannot be ensured, −1 means that no optimal value can be extracted);
- DR = dense relaxation;
- SR = sparse relaxation;
- SCSRv1 = squared-constraint sparse relaxation (union version);
- SCSRv2 = squared-constraint sparse relaxation (separate version);
- HRv1 = heuristic relaxation (union version);
- HRv2 = heuristic relaxation (separate version);
- rat. = ratio of gap (obj. of current method / obj. of dense method order 2 ×100%).

Robustness margin certification problems (RMCP)

$$\begin{aligned} \max \quad & F_W(x)_y - F_W(\bar{x})_{\bar{y}} = (c_y - c_{\bar{y}})^T x_L \\ \text{s.t.} \quad & \begin{cases} x_i = ReLU(A_i x_{i-1} + b_i), i = 1, \ldots, L; \\ ||x - \bar{x}||_\infty \le \varepsilon. \end{cases} \end{aligned}$$

POP form of RMCP

$$\begin{aligned} \max_{x,z} \quad & c^T x_L \\ \text{s.t.} \quad & \begin{cases} z_i - A_i x_{i-1} - b_i = 0, i = 1, \ldots, L; \\ x_i(x_i - z_i) = 0, x_i \ge z_i, x_i \ge 0, i = 1, \ldots, L; \\ -\varepsilon \le x - \bar{x} \le \varepsilon. \end{cases} \end{aligned}$$

Squared-constraint RMCP (SC-RMCP)

$$\begin{aligned} \max_{x,z} \quad & c^T x_L \\ \text{s.t.} \quad & \begin{cases} ||z_i - A_i x_{i-1} - b_i||^2 = 0, i = 1, \ldots, L; & \text{(separate version)} \\ \sum_{i=1}^L ||z_i - A_i x_{i-1} - b_i||^2 = 0; & \text{(union version)} \\ x_i(x_i - z_i) = 0, x_i \ge z_i, x_i \ge 0, i = 1, \ldots, L; \\ -\varepsilon \le x - \bar{x} \le \varepsilon. \end{cases} \end{aligned}$$

Table 1: 1 layer, 3×3, $A = \begin{bmatrix} 0.3810 & 0.0331 & 0.3888 \\ 0.1343 & 0.6749 & 0.5289 \\ 0.4262 & 0.9193 & 0.4403 \end{bmatrix}$, $b = \begin{bmatrix} 0.3287 \\ 0.2185 \\ 0.2622 \end{bmatrix}$, $c = \begin{bmatrix} 0.2569 \\ 0.0028 \\ 0.8297 \end{bmatrix}$, $\bar{x} = \begin{bmatrix} 0.4386 \\ 0.3342 \\ 0.1891 \end{bmatrix}$

ε	Res.	Relaxation Methods														
		DR		SR	SCSRv1					SCSRv2					HRv1	HRv2
		ord2	ord1		ord2	ord3	ord4	ord5	ord6	ord2	ord3	ord4	ord5	ord6		
0e+00	obj.	**8.47e-01**	**1.81e+00**	8.47e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01
	time (s)	2.28e+00	1.40e-01	6.65e-01	1.20e+00	1.57e+00	2.53e+00	4.67e+00	8.47e+00	1.37e+00	1.54e+00	2.73e+00	5.09e+00	8.60e+00	7.10e-01	7.81e-01
	stat. (-1/0/1)	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	rat. (%)	-	1.14e+02	1.57e-07	2.71e-02	2.42e-02	2.31e-02	2.21e-02	1.75e-02	2.96e-02	3.39e-02	2.63e-02	2.20e-02	2.01e-02	1.75e-02	1.93e-02
1e-06	obj.	**8.47e-01**	**1.81e+00**	8.47e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01
	time (s)	2.68e+00	1.21e-01	6.65e-01	1.41e+00	2.07e+00	3.40e+00	5.98e+00	9.92e+00	1.38e+00	1.87e+00	3.32e+00	5.25e+00	9.94e+00	8.02e-01	9.20e-01
	stat. (-1/0/1)	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	rat. (%)	-	1.14e+02	-1.00e-06	7.93e-02	8.71e-02	8.22e-02	8.09e-02	7.61e-02	7.24e-02	7.80e-02	7.27e-02	6.40e-02	6.80e-02	2.64e-02	2.37e-02
1e-05	obj.	**8.47e-01**	**1.81e+00**	8.47e-01	8.49e-01	8.48e-01	8.49e-01	8.49e-01	8.49e-01	8.49e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01	8.48e-01
	time (s)	2.74e+00	1.28e-01	6.67e-01	1.94e+00	2.14e+00	3.94e+00	8.11e+00	1.48e+01	1.70e+00	2.18e+00	4.26e+00	6.07e+00	1.20e+01	9.80e-01	8.93e-01
	stat. (-1/0/1)	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	rat. (%)	-	1.14e+02	-2.99e-08	1.71e-01	1.23e-01	1.31e-01	1.34e-01	1.38e-01	1.46e-01	1.23e-01	1.16e-01	9.82e-02	1.11e-01	2.33e-02	2.38e-02
1e-04	obj.	**8.48e-01**	**1.81e+00**	8.48e-01	8.50e-01	8.50e-01	8.49e-01	8.49e-01	8.50e-01	8.50e-01	8.49e-01	8.49e-01	8.49e-01	8.49e-01	8.48e-01	8.48e-01
	time (s)	2.57e+00	1.22e-01	6.60e-01	1.70e+00	2.35e+00	4.31e+00	8.69e+00	1.61e+01	1.61e+00	3.02e+00	5.18e+00	6.83e+00	1.34e+01	9.03e-01	9.87e-01
	stat. (-1/0/1)	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	rat. (%)	-	1.14e+02	7.24e-08	2.71e-01	2.50e-01	2.18e-01	2.17e-01	2.64e-01	2.30e-01	2.08e-01	1.98e-01	1.60e-01	1.69e-01	1.97e-02	1.97e-02
1e-03	obj.	**8.49e-01**	**1.81e+00**	8.49e-01	8.53e-01	8.52e-01	8.52e-01	8.52e-01	8.53e-01	8.52e-01	8.51e-01	8.51e-01	8.51e-01	8.52e-01	8.49e-01	8.49e-01
	time (s)	2.58e+00	1.18e-01	6.69e-01	1.63e+00	2.93e+00	5.06e+00	1.04e+01	1.63e+01	1.72e+00	2.47e+00	5.12e+00	6.60e+00	1.68e+01	9.30e-01	1.09e+00
	stat. (-1/0/1)	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	rat. (%)	-	1.13e+02	-4.32e-07	4.10e-01	3.52e-01	3.09e-01	3.16e-01	4.60e-01	3.24e-01	2.66e-01	2.48e-01	2.41e-01	3.05e-01	1.67e-02	1.71e-02
1e-02	obj.	**8.64e-01**	**1.82e+00**	8.64e-01	8.76e-01	8.76e-01	8.76e-01	8.76e-01	8.76e-01	8.73e-01	8.73e-01	8.73e-01	8.73e-01	8.73e-01	8.64e-01	8.64e-01
	time (s)	2.55e+00	1.25e-01	6.27e-01	1.54e+00	2.30e+00	4.10e+00	6.37e+00	1.21e+01	1.49e+00	2.21e+00	3.79e+00	8.58e+00	1.25e+01	9.51e-01	1.06e+00
	stat. (-1/0/1)	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	rat. (%)	-	1.11e+02	2.17e-07	1.33e+00	1.32e+00	1.32e+00	1.32e+00	1.37e+00	9.80e-01	9.73e-01	9.72e-01	9.72e-01	1.01e+00	1.79e-02	1.66e-02
1e-01	obj.	**1.02e+00**	**1.90e+00**	1.02e+00	1.13e+00	1.13e+00	1.13e+00	1.13e+00	1.13e+00	1.10e+00	1.10e+00	1.10e+00	1.10e+00	1.10e+00	1.02e+00	1.02e+00
	time (s)	2.39e+00	1.21e-01	6.23e-01	1.23e+00	1.52e+00	2.93e+00	7.25e+00	1.19e+01	1.52e+00	1.67e+00	3.51e+00	9.96e+00	1.04e+01	6.33e-01	6.99e-01
	stat. (-1/0/1)	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1
	rat. (%)	-	8.65e+01	-5.12e-07	1.14e+01	1.14e+01	1.14e+01	1.14e+01	1.13e+01	8.45e+00	8.45e+00	8.45e+00	8.43e+00	8.36e+00	1.45e-02	1.45e-02
2e-01	obj.	**1.19e+00**	**1.97e+00**	1.19e+00	1.42e+00	1.42e+00	1.42e+00	1.42e+00	1.42e+00	1.36e+00	1.36e+00	1.36e+00	1.36e+00	1.36e+00	1.19e+00	1.19e+00
	time (s)	1.27e+00	8.43e-02	4.30e-01	5.17e-01	8.76e-01	3.56e+00	9.46e+00	1.17e+01	9.86e-01	8.64e-01	1.93e+00	3.55e+00	5.92e+00	6.73e-01	6.99e-01
	stat. (-1/0/1)	1	0	1	0	0	0	0	-1	0	0	0	0	0	1	1
	rat. (%)	-	6.60e+01	1.49e-07	2.00e+01	2.00e+01	2.00e+01	1.99e+01	1.94e+01	1.48e+01	1.48e+01	1.48e+01	1.45e+01	1.45e+01	1.11e-02	1.06e-02
3e-01	obj.	**1.35e+00**	**2.03e+00**	1.35e+00	1.71e+00	1.71e+00	1.71e+00	1.71e+00	1.53e+00	1.62e+00	1.62e+00	1.62e+00	1.62e+00	1.59e+00	1.36e+00	1.36e+00
	time (s)	1.35e+00	9.19e-02	4.13e-01	6.76e-01	9.00e-01	2.08e+00	3.76e+00	4.47e+00	5.74e-01	9.33e-01	2.73e+00	3.58e+00	5.06e+00	6.80e-01	7.15e-01
	stat. (-1/0/1)	1	0	1	0	0	0	0	-1	0	0	0	0	-1	1	1
	rat. (%)	-	5.01e+01	-7.92e-06	2.65e+01	2.65e+01	2.65e+01	2.61e+01	1.29e+01	1.98e+01	1.98e+01	1.98e+01	1.93e+01	1.75e+01	1.13e-02	1.13e-02
4e-01	obj.	**1.52e+00**	**2.09e+00**	1.52e+00	2.01e+00	2.01e+00	2.01e+00	1.71e+00	1.55e+00	1.88e+00	1.88e+00	1.88e+00	1.64e+00	1.53e+00	1.52e+00	1.52e+00
	time (s)	1.36e+00	8.29e-02	4.08e-01	5.01e-01	1.31e+00	3.70e+00	2.86e+00	4.11e+00	4.79e-01	1.54e+00	3.14e+00	3.30e+00	4.95e+00	7.38e-01	6.92e-01
	stat. (-1/0/1)	1	0	1	0	0	0	-1	-1	0	0	0	-1	-1	1	1
	rat. (%)	-	3.74e+01	-4.89e-06	3.16e+01	3.16e+01	3.16e+01	1.23e+01	1.40e+00	2.37e+01	2.37e+01	2.36e+01	7.43e+00	1.20e-01	7.78e-03	1.02e-02
6e-01	obj.	**1.86e+00**	**2.20e+00**	1.86e+00	2.58e+00	2.58e+00	2.56e+00	1.75e+00	1.52e+00	2.41e+00	2.41e+00	2.27e+00	1.68e+00	1.79e+00	1.86e+00	1.86e+00
	time (s)	1.36e+00	8.50e-02	4.52e-01	6.67e-01	1.36e+00	3.77e+00	2.82e+00	4.71e+00	7.29e-01	1.65e+00	1.44e+00	2.74e+00	4.81e+00	6.58e-01	7.18e-01
	stat. (-1/0/1)	1	0	1	0	0	-1	-1	-1	0	0	-1	-1	-1	1	1
	rat. (%)	-	1.82e+01	7.82e-08	3.85e+01	3.85e+01	3.72e+01	-5.99e+00	-1.82e+01	2.93e+01	2.93e+01	2.20e+01	-9.75e+00	-3.91e+00	8.47e-03	6.89e-03
8e-01	obj.	**2.20e+00**	**2.30e+00**	2.05e+00	2.85e+00	2.85e+00	2.10e+00	2.12e+00	1.56e+00	2.72e+00	2.72e+00	2.51e+00	1.70e+00	1.80e+00	2.20e+00	2.20e+00
	time (s)	1.45e+00	8.78e-02	5.04e-01	6.67e-01	1.49e+00	1.51e+00	2.70e+00	5.36e+00	5.79e-01	1.47e+00	1.54e+00	3.19e+00	4.54e+00	6.39e-01	7.44e-01
	stat. (-1/0/1)	1	0	1	0	0	-1	-1	-1	0	0	-1	-1	-1	1	1
	rat. (%)	-	4.28e+00	-6.76e+00	2.94e+01	2.94e+01	-4.54e+00	-3.62e+00	-2.92e+01	2.36e+01	2.36e+01	1.40e+01	-2.28e+01	-1.82e+01	6.69e-03	6.10e-03
1e+00	obj.	**2.32e+00**	**2.33e+00**	2.10e+00	3.00e+00	3.00e+00	2.12e+00	2.14e+00	1.56e+00	2.80e+00	2.80e+00	2.13e+00	2.14e+00	1.83e+00	2.32e+00	2.32e+00
	time (s)	1.84e+00	8.64e-02	5.13e-01	5.85e-01	1.24e+00	1.99e+00	3.17e+00	5.83e+00	7.73e-01	1.30e+00	2.59e+00	2.95e+00	4.60e+00	5.97e-01	6.01e-01
	stat. (-1/0/1)	1	0	1	0	0	-1	-1	-1	0	0	-1	-1	-1	1	1
	rat. (%)	-	4.09e-01	-9.50e+00	2.92e+01	2.92e+01	-8.41e+00	-7.89e+00	-3.29e+01	2.09e+01	2.09e+01	-8.17e+00	-7.76e+00	-2.13e+01	2.99e-03	2.51e-03

Table 2: 1 layer, $A = \begin{bmatrix} 0.2630 & 0.4444 & 0.4803 & 0.6979 \\ 0.1783 & 0.4423 & 0.0111 & 0.3726 \\ 0.7771 & 0.4647 & 0.8002 & 0.1375 \\ 0.7494 & 0.5672 & 0.4868 & 0.9341 \end{bmatrix}$, $b = \begin{bmatrix} 0.0646 \\ 0.3565 \\ 0.1843 \\ 0.9440 \end{bmatrix}$, $c = \begin{bmatrix} 0.4584 \\ 0.5267 \\ 0.4215 \\ 0.0160 \end{bmatrix}$, $\bar{x} = \begin{bmatrix} 0.6487 \\ 0.9113 \\ 0.8957 \\ 0.3579 \end{bmatrix}$

ε	Res.	Relaxation Methods														
		DR		SR	SCSRv1					SCSRv2					HRv1	HRv2
		ord2	ord1		ord2	ord3	ord4	ord5	ord6	ord2	ord3	ord4	ord5	ord6		
0e+00	obj.	**1.98e+00**	**4.42e+00**	1.98e+00	1.98e+00	1.98e+00	1.98e+00	1.97e+00	1.96e+00	1.98e+00	1.98e+00	1.98e+00	1.97e+00	1.97e+00	1.98e+00	1.98e+00
	time (s)	1.81e+01	1.41e-01	2.15e+00	9.70e-01	2.50e+00	3.36e+00	5.39e+00	1.01e+01	1.46e+00	2.06e+00	2.63e+00	6.40e+00	1.08e+01	8.72e-01	9.23e-01
	stat. (-1/0/1)	1	0	1	1	1	1	-1	-1	1	1	1	-1	-1	1	1
	rat. (%)	-	1.24e+02	-5.40e-07	4.06e-02	2.01e-02	1.57e-02	-3.37e-01	-8.31e-01	2.53e-02	3.19e-02	2.32e-02	-2.68e-01	-2.90e-01	1.72e-02	1.61e-02
1e-06	obj.	**1.98e+00**	**4.42e+00**	1.98e+00	1.98e+00	1.98e+00	1.98e+00	1.97e+00	1.97e+00	1.98e+00	1.98e+00	1.98e+00	1.97e+00	1.97e+00	1.98e+00	1.98e+00
	time (s)	2.38e+01	1.19e-01	2.62e+00	1.14e+00	4.10e+00	3.36e+00	5.37e+00	1.46e+01	1.57e+00	2.21e+00	3.15e+00	4.98e+00	1.11e+01	1.04e+00	1.07e+00
	stat. (-1/0/1)	1	0	1	1	1	1	-1	-1	1	1	1	-1	-1	1	1
	rat. (%)	-	1.24e+02	-3.20e-05	9.95e-02	6.76e-02	1.88e-02	-3.40e-01	-1.77e-01	6.41e-02	5.01e-02	3.37e-02	-3.69e-01	-2.91e-01	2.89e-02	2.90e-02
1e-05	obj.	**1.98e+00**	**4.42e+00**	1.98e+00	1.98e+00	1.98e+00	1.98e+00	1.97e+00	1.96e+00	1.98e+00	1.98e+00	1.98e+00	1.97e+00	1.97e+00	1.98e+00	1.98e+00
	time (s)	2.36e+01	1.13e-01	2.73e+00	1.32e+00	3.77e+00	4.04e+00	5.15e+00	9.04e+00	1.94e+00	2.22e+00	3.15e+00	6.12e+00	1.16e+01	1.16e+00	1.18e+00
	stat. (-1/0/1)	1	0	1	1	1	1	-1	-1	1	1	1	-1	-1	1	1
	rat. (%)	-	1.24e+02	7.69e-06	1.60e-01	1.07e-01	3.54e-02	-3.23e-01	-6.72e-01	1.04e-01	1.16e-01	3.84e-02	-2.63e-01	-2.76e-01	3.26e-02	3.49e-02
1e-04	obj.	**1.98e+00**	**4.42e+00**	1.98e+00	1.98e+00	1.98e+00	1.98e+00	1.97e+00	1.96e+00	1.98e+00	1.98e+00	1.98e+00	1.97e+00	1.96e+00	1.98e+00	1.98e+00
	time (s)	2.51e+01	1.16e-01	2.74e+00	1.53e+00	5.08e+00	4.67e+00	5.28e+00	1.04e+01	1.94e+00	2.05e+00	4.03e+00	4.94e+00	9.20e+00	1.40e+00	1.31e+00
	stat. (-1/0/1)	1	0	1	1	1	1	-1	-1	1	1	1	-1	-1	1	1
	rat. (%)	-	1.24e+02	6.50e-06	2.64e-01	2.04e-01	9.47e-02	-3.38e-01	-6.26e-01	2.37e-01	2.77e-01	7.52e-02	-3.57e-01	-8.11e-01	2.74e-02	2.73e-02
1e-03	obj.	**1.98e+00**	**4.42e+00**	1.98e+00	1.99e+00	1.98e+00	1.98e+00	1.97e+00	1.93e+00	1.99e+00	1.99e+00	1.98e+00	1.97e+00	1.97e+00	1.98e+00	1.98e+00
	time (s)	2.36e+01	1.15e-01	2.62e+00	1.62e+00	6.58e+00	4.69e+00	5.14e+00	7.50e+00	1.81e+00	2.46e+00	4.40e+00	4.81e+00	1.21e+01	1.53e+00	1.52e+00
	stat. (-1/0/1)	1	0	1	1	1	1	-1	-1	1	1	1	-1	-1	1	1
	rat. (%)	-	1.23e+02	-3.63e-06	4.45e-01	2.62e-01	1.94e-01	-5.22e-01	-2.48e+00	5.35e-01	3.08e-01	1.23e-01	-5.47e-01	-5.02e-01	2.65e-02	1.92e-02
1e-02	obj.	**2.00e+00**	**4.43e+00**	2.00e+00	2.02e+00	2.02e+00	2.01e+00	1.96e+00	1.93e+00	2.01e+00	2.02e+00	1.99e+00	1.96e+00	1.95e+00	2.00e+00	2.00e+00
	time (s)	2.36e+01	1.22e-01	2.50e+00	1.63e+00	3.94e+00	4.45e+00	7.25e+00	6.85e+00	2.42e+00	1.69e+00	3.60e+00	5.50e+00	9.53e+00	1.59e+00	1.62e+00
	stat. (-1/0/1)	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1	1	1
	rat. (%)	-	1.21e+02	-6.08e-06	1.17e+00	1.16e+00	4.23e-01	-1.78e+00	-3.47e+00	6.57e-01	7.94e-01	-6.65e-01	-1.86e+00	-2.45e+00	2.82e-02	2.07e-02
1e-01	obj.	**2.21e+00**	**4.50e+00**	2.21e+00	2.44e+00	2.44e+00	2.34e+00	1.99e+00	1.86e+00	2.34e+00	2.33e+00	2.28e+00	1.95e+00	1.83e+00	2.21e+00	2.21e+00
	time (s)	2.13e+01	1.14e-01	2.51e+00	1.50e+00	2.90e+00	5.03e+00	4.48e+00	6.61e+00	1.47e+00	1.82e+00	4.95e+00	5.35e+00	9.18e+00	2.00e+00	1.74e+00
	stat. (-1/0/1)	1	0	1	0	0	-1	-1	-1	0	0	-1	0	-1	1	1
	rat. (%)	-	1.03e+02	-3.78e-06	1.05e+01	1.03e+01	5.84e+00	-1.03e+01	-1.60e+01	5.90e+00	5.38e+00	2.87e+00	-1.17e+01	-1.75e+01	2.28e-02	1.50e-02
2e-01	obj.	**2.45e+00**	**4.57e+00**	2.45e+00	2.92e+00	2.90e+00	2.70e+00	2.41e+00	1.78e+00	2.71e+00	2.69e+00	2.41e+00	2.30e+00	1.81e+00	2.45e+00	2.45e+00
	time (s)	2.03e+01	1.27e-01	2.51e+00	1.63e+00	2.41e+00	5.75e+00	5.69e+00	7.06e+00	1.50e+00	1.67e+00	7.34e+00	5.92e+00	8.27e+00	1.88e+00	1.90e+00
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	0	0	-1	-1	-1	-1	1	1
	rat. (%)	-	8.67e+01	-1.25e-05	1.91e+01	1.84e+01	1.04e+01	-1.43e+00	-2.74e+01	1.08e+01	9.93e+00	-1.35e+00	-5.97e+00	-2.63e+01	1.99e-02	1.37e-02
3e-01	obj.	**2.68e+00**	**4.64e+00**	2.68e+00	3.39e+00	3.36e+00	2.97e+00	2.24e+00	2.20e+00	3.09e+00	3.04e+00	2.61e+00	2.41e+00	2.02e+00	2.68e+00	2.68e+00
	time (s)	2.01e+01	1.27e-01	2.62e+00	1.86e+00	2.25e+00	4.64e+00	3.72e+00	7.64e+00	1.69e+00	1.84e+00	4.41e+00	5.90e+00	8.50e+00	1.97e+00	1.95e+00
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	-1	0	-1	-1	-1	-1	1	1
	rat. (%)	-	7.28e+01	-2.69e-05	2.64e+01	2.51e+01	1.06e+01	-1.64e+01	-1.78e+01	1.50e+01	1.33e+01	-2.66e+00	-1.01e+01	-2.46e+01	1.48e-02	1.24e-02
4e-01	obj.	**2.92e+00**	**4.70e+00**	2.92e+00	3.87e+00	3.79e+00	3.20e+00	2.78e+00	2.38e+00	3.46e+00	3.38e+00	2.87e+00	2.61e+00	2.23e+00	2.92e+00	2.92e+00
	time (s)	2.04e+01	1.15e-01	2.66e+00	1.01e+00	2.53e+00	5.63e+00	4.60e+00	8.10e+00	1.45e+00	1.85e+00	3.83e+00	5.77e+00	8.42e+00	1.98e+00	2.03e+00
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	-1	0	-1	-1	-1	-1	1	1
	rat. (%)	-	6.09e+01	-8.64e-05	3.25e+01	3.00e+01	9.68e+00	-4.62e+00	-1.84e+01	1.86e+01	1.59e+01	-1.76e+00	-1.05e+01	-2.36e+01	1.68e-02	1.10e-02
6e-01	obj.	**3.39e+00**	**4.79e+00**	3.39e+00	4.83e+00	4.67e+00	3.29e+00	2.29e+00	2.42e+00	4.20e+00	3.85e+00	3.02e+00	2.22e+00	2.26e+00	3.39e+00	3.39e+00
	time (s)	2.14e+01	1.23e-01	2.66e+00	1.41e+00	3.11e+00	4.09e+00	5.06e+00	9.88e+00	1.26e+00	1.94e+00	3.90e+00	4.79e+00	9.79e+00	1.76e+00	1.86e+00
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1
	rat. (%)	-	4.13e+01	-1.58e-04	4.24e+01	3.78e+01	-2.92e+00	-3.24e+01	-2.87e+01	2.38e+01	1.35e+01	-1.10e+01	-3.47e+01	-3.33e+01	1.08e-02	9.88e-03
8e-01	obj.	**3.86e+00**	**4.86e+00**	3.86e+00	5.78e+00	4.66e+00	2.69e+00	2.37e+00	1.99e+00	4.97e+00	4.21e+00	2.49e+00	2.72e+00	2.06e+00	3.86e+00	3.86e+00
	time (s)	2.26e+01	1.29e-01	2.90e+00	1.44e+00	1.62e+00	3.87e+00	5.63e+00	6.94e+00	2.56e+00	1.70e+00	3.59e+00	4.85e+00	8.95e+00	1.73e+00	2.07e+00
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	-1	0	-1	-1	-1	-1	1	1
	rat. (%)	-	2.59e+01	-3.24e-05	4.97e+01	2.06e+01	-3.02e+01	-3.85e+01	-4.83e+01	2.87e+01	8.97e+00	-3.54e+01	-2.96e+01	-4.66e+01	1.01e-02	7.43e-03
1e+00	obj.	**4.33e+00**	**4.91e+00**	3.97e+00	6.73e+00	4.00e+00	2.54e+00	2.83e+00	2.37e+00	5.71e+00	4.28e+00	2.58e+00	2.44e+00	2.03e+00	4.33e+00	4.33e+00
	time (s)	2.51e+01	1.24e-01	3.16e+00	1.53e+00	1.50e+00	2.99e+00	5.62e+00	8.32e+00	2.26e+00	1.49e+00	3.07e+00	5.15e+00	9.03e+00	1.71e+00	1.90e+00
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1
	rat. (%)	-	1.34e+01	-8.43e+00	5.54e+01	-7.73e+00	-4.15e+01	-3.46e+01	-4.52e+01	3.18e+01	-1.11e+00	-4.05e+01	-4.37e+01	-5.32e+01	7.54e-03	6.47e-03

Table 3: 1 layer, $A \in \mathbb{R}^{5\times 5}$, $A = \begin{bmatrix} 0.5170 & 0.9898 & 0.3506 & 0.0375 & 0.8955 \\ 0.4844 & 0.3974 & 0.7774 & 0.8110 & 0.1989 \\ 0.9469 & 0.1234 & 0.5375 & 0.6920 & 0.6093 \\ 0.4953 & 0.9927 & 0.5437 & 0.7207 & 0.3242 \\ 0.2610 & 0.9350 & 0.5323 & 0.2581 & 0.4252 \end{bmatrix}$, $b = \begin{bmatrix} 0.4268 \\ 0.0632 \\ 0.3252 \\ 0.4319 \\ 0.6100 \end{bmatrix}$, $c = \begin{bmatrix} 0.5170 \\ 0.5828 \\ 0.8641 \\ 0.1573 \\ 0.5597 \end{bmatrix}$, $\bar{x} = \begin{bmatrix} 0.4996 \\ 0.4439 \\ 0.1962 \\ 0.0233 \\ 0.0174 \end{bmatrix}$

ε	Res.	Relaxation Methods														
		DR		SR	SCSRv1					SCSRv2					HRv1	HRv2
		ord2	ord1		ord2	ord3	ord4	ord5	ord6	ord2	ord3	ord4	ord5	ord6		
0e+00	obj.	**2.77e+00**	**6.37e+00**	2.77e+00	2.77e+00	2.77e+00	2.77e+00	2.79e+00	2.78e+00	2.77e+00	2.77e+00	2.78e+00	2.79e+00	2.78e+00	2.77e+00	2.77e+00
	time (s)	2.14e+02	1.87e-01	1.19e+01	1.60e+00	3.21e+00	1.69e+01	8.37e+00	1.50e+01	1.57e+00	2.47e+00	1.18e+01	9.02e+00	1.11e+01	2.35e+00	2.47e+00
	stat. (-1/0/1)	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	rat. (%)	-	1.30e+02	-3.14e-07	4.51e-02	3.44e-02	5.92e-02	7.10e-01	2.36e-01	4.31e-02	3.16e-02	3.23e-01	7.65e-01	2.96e-01	1.79e-02	2.04e-02
1e-06	obj.	**2.77e+00**	**6.37e+00**	2.77e+00	2.77e+00	2.77e+00	2.77e+00	2.79e+00	2.78e+00	2.77e+00	2.77e+00	2.78e+00	2.79e+00	2.78e+00	2.77e+00	2.77e+00
	time (s)	2.54e+02	1.86e-01	1.39e+01	1.87e+00	5.65e+00	2.54e+01	8.24e+00	1.77e+01	1.87e+00	3.67e+00	1.05e+01	9.02e+00	1.09e+01	3.04e+00	3.00e+00
	stat. (-1/0/1)	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	rat. (%)	-	1.30e+02	-5.66e-06	1.32e-01	1.10e-01	9.63e-02	7.49e-01	2.65e-01	1.20e-01	9.78e-02	3.82e-01	7.98e-01	3.23e-01	3.16e-02	3.13e-02
1e-05	obj.	**2.77e+00**	**6.37e+00**	2.77e+00	2.77e+00	2.77e+00	2.78e+00	2.79e+00	2.78e+00	2.78e+00	2.77e+00	2.78e+00	2.79e+00	2.78e+00	2.77e+00	2.77e+00
	time (s)	2.55e+02	1.80e-01	1.39e+01	2.18e+00	8.06e+00	1.70e+01	1.01e+01	1.22e+01	3.07e+00	4.14e+00	2.17e+01	9.07e+00	2.14e+01	3.47e+00	3.61e+00
	stat. (-1/0/1)	1	0	1	1	1	1	-1	1	1	1	1	1	1	1	1
	rat. (%)	-	1.30e+02	-1.41e-06	2.27e-01	1.90e-01	3.66e-01	9.24e-01	3.57e-01	2.38e-01	1.68e-01	2.63e-01	9.48e-01	4.20e-01	2.96e-02	2.47e-02
1e-04	obj.	**2.77e+00**	**6.37e+00**	2.77e+00	2.78e+00	2.78e+00	2.80e+00	2.80e+00	2.79e+00	2.78e+00	2.78e+00	2.79e+00	2.80e+00	2.79e+00	2.77e+00	2.77e+00
	time (s)	2.53e+02	2.02e-01	1.38e+01	4.14e+00	2.06e+01	1.19e+01	1.81e+01	2.30e+01	3.94e+00	9.75e+00	1.61e+01	1.47e+01	1.97e+01	3.62e+00	3.26e+00
	stat. (-1/0/1)	1	0	1	1	1	-1	1	1	1	1	1	1	1	1	1
	rat. (%)	-	1.30e+02	-2.84e-06	4.31e-01	3.12e-01	1.04e+00	9.32e-01	6.03e-01	4.01e-01	3.03e-01	6.23e-01	1.14e+00	6.54e-01	2.56e-02	1.96e-02
1e-03	obj.	**2.78e+00**	**6.37e+00**	2.78e+00	2.79e+00	2.79e+00	2.82e+00	2.81e+00	2.80e+00	2.79e+00	2.79e+00	2.83e+00	2.81e+00	2.80e+00	2.78e+00	2.78e+00
	time (s)	2.41e+02	1.87e-01	1.33e+01	3.43e+00	1.80e+01	1.45e+01	2.07e+01	1.86e+01	2.93e+00	1.34e+01	9.62e+00	2.56e+01	2.21e+01	4.22e+00	3.57e+00
	stat. (-1/0/1)	1	0	1	1	1	-1	1	1	1	1	-1	1	1	1	1
	rat. (%)	-	1.30e+02	3.69e-07	6.42e-01	5.16e-01	1.62e+00	1.27e+00	9.19e-01	4.81e-01	4.90e-01	1.78e+00	1.29e+00	8.34e-01	2.03e-02	1.82e-02
1e-02	obj.	**2.84e+00**	**6.39e+00**	2.84e+00	2.89e+00	2.89e+00	2.90e+00	2.90e+00	2.89e+00	2.89e+00	2.89e+00	2.92e+00	2.89e+00	2.88e+00	2.84e+00	2.84e+00
	time (s)	2.40e+02	1.86e-01	1.26e+01	3.22e+00	8.33e+00	1.16e+01	2.41e+01	2.08e+01	3.26e+00	7.17e+00	1.25e+01	2.61e+01	2.16e+01	4.62e+00	4.68e+00
	stat. (-1/0/1)	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1
	rat. (%)	-	1.25e+02	-4.20e-06	1.80e+00	1.77e+00	2.12e+00	2.12e+00	1.77e+00	1.52e+00	1.76e+00	2.62e+00	1.83e+00	1.51e+00	2.19e-02	2.00e-02
1e-01	obj.	**3.50e+00**	**6.58e+00**	3.50e+00	4.02e+00	4.02e+00	4.01e+00	3.85e+00	3.88e+00	3.95e+00	3.94e+00	3.86e+00	3.61e+00	3.71e+00	3.50e+00	3.50e+00
	time (s)	2.28e+02	2.43e-01	1.33e+01	3.63e+00	6.49e+00	1.06e+01	1.31e+01	1.46e+01	6.65e+00	4.04e+00	6.11e+00	9.57e+00	1.79e+01	4.42e+00	4.41e+00
	stat. (-1/0/1)	1	0	1	0	0	-1	-1	-1	0	0	-1	-1	-1	1	1
	rat. (%)	-	8.79e+01	2.27e-06	1.49e+01	1.49e+01	1.46e+01	9.89e+00	1.08e+01	1.27e+01	1.26e+01	1.01e+01	3.08e+00	5.87e+00	1.51e-02	1.38e-02
2e-01	obj.	**4.24e+00**	**6.78e+00**	4.24e+00	5.30e+00	5.29e+00	4.40e+00	4.18e+00	3.83e+00	5.15e+00	5.12e+00	4.73e+00	3.89e+00	3.59e+00	4.24e+00	4.24e+00
	time (s)	2.40e+02	2.03e-01	1.32e+01	3.97e+00	4.10e+00	6.06e+00	1.08e+01	1.69e+01	2.45e+00	4.88e+00	6.87e+00	1.15e+01	1.71e+01	4.34e+00	4.23e+00
	stat. (-1/0/1)	1	0	1	0	0	-1	-1	-1	0	-1	-1	-1	-1	1	1
	rat. (%)	-	6.00e+01	-2.13e-05	2.51e+01	2.48e+01	3.76e+00	-1.28e+00	-9.54e+00	2.16e+01	2.08e+01	1.17e+01	-8.21e+00	-1.52e+01	1.21e-02	1.14e-02
3e-01	obj.	**4.97e+00**	**6.97e+00**	4.97e+00	6.59e+00	5.75e+00	4.60e+00	4.82e+00	3.58e+00	6.37e+00	5.97e+00	4.67e+00	4.50e+00	4.08e+00	4.97e+00	4.97e+00
	time (s)	2.41e+02	1.73e-01	1.41e+01	2.88e+00	2.65e+00	5.48e+00	1.30e+01	1.30e+01	1.81e+00	3.72e+00	7.83e+00	1.29e+01	1.38e+01	4.69e+00	3.81e+00
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	-1	0	0	-1	-1	-1	1	1
	rat. (%)	-	4.02e+01	-1.88e-05	3.26e+01	1.57e+01	-7.42e+00	-3.03e+00	-2.80e+01	2.81e+01	2.01e+01	-6.01e+00	-9.58e+00	-1.79e+01	8.94e-03	1.22e-02
4e-01	obj.	**5.71e+00**	**7.14e+00**	5.71e+00	7.88e+00	6.21e+00	4.58e+00	3.81e+00	4.23e+00	7.59e+00	6.22e+00	4.57e+00	4.60e+00	3.81e+00	5.71e+00	5.71e+00
	time (s)	2.70e+02	1.91e-01	1.53e+01	2.48e+00	3.29e+00	4.79e+00	8.20e+00	1.56e+01	2.66e+00	3.04e+00	4.93e+00	1.15e+01	1.49e+01	4.46e+00	5.28e+00
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	-1	0	-1	-1	-1	-1	1	1
	rat. (%)	-	2.52e+01	-3.42e-05	3.81e+01	8.74e+00	-1.97e+01	-3.32e+01	-2.59e+01	3.29e+01	9.01e+00	-1.99e+01	-1.93e+01	-3.33e+01	5.98e-03	6.69e-03
6e-01	obj.	**7.18e+00**	**7.43e+00**	5.98e+00	1.04e+01	8.51e+00	4.54e+00	4.93e+00	3.74e+00	9.77e+00	5.98e+00	5.70e+00	3.97e+00	3.65e+00	7.18e+00	7.18e+00
	time (s)	3.27e+02	1.74e-01	1.83e+01	3.80e+00	3.51e+00	5.49e+00	1.11e+01	1.61e+01	3.30e+00	2.20e+00	7.14e+00	8.34e+00	1.59e+01	4.07e+00	4.80e+00
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	-1	0	-1	-1	-1	-1	1	1
	rat. (%)	-	3.60e+00	-1.67e+01	4.55e+01	1.86e+01	-3.67e+01	-3.13e+01	-4.78e+01	3.61e+01	-1.66e+01	-2.05e+01	-4.47e+01	-4.91e+01	6.86e-03	5.35e-03
8e-01	obj.	**7.50e+00**	**7.56e+00**	6.12e+00	1.26e+01	8.62e+00	4.80e+00	4.89e+00	3.61e+00	1.22e+01	8.50e+00	5.74e+00	4.88e+00	3.71e+00	7.50e+00	7.50e+00
	time (s)	4.12e+02	1.80e-01	1.74e+01	6.08e+00	3.11e+00	6.24e+00	1.06e+01	1.49e+01	3.92e+00	2.99e+00	6.25e+00	8.06e+00	1.59e+01	4.47e+00	4.67e+00
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	-1	0	-1	-1	-1	-1	0	0
	rat. (%)	-	7.74e-01	-1.84e+01	6.81e+01	1.49e+01	-3.60e+01	-3.48e+01	-5.19e+01	6.28e+01	1.33e+01	-2.34e+01	-3.49e+01	-5.05e+01	1.59e-02	2.13e-02
1e+00	obj.	**7.53e+00**	**7.60e+00**	6.19e+00	1.32e+01	8.70e+00	4.39e+00	4.90e+00	3.58e+00	1.31e+01	8.57e+00	4.78e+00	4.90e+00	3.69e+00	7.54e+00	7.54e+00
	time (s)	3.85e+02	1.91e-01	1.67e+01	2.88e+00	3.01e+00	4.27e+00	1.12e+01	1.61e+01	2.04e+00	2.96e+00	4.95e+00	8.28e+00	1.64e+01	4.27e+00	4.08e+00
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	-1	0	-1	-1	-1	-1	0	0
	rat. (%)	-	9.92e-01	-1.78e+01	7.55e+01	1.55e+01	-4.17e+01	-3.49e+01	-5.24e+01	7.45e+01	1.38e+01	-3.66e+01	-3.50e+01	-5.10e+01	1.85e-01	1.85e-01

Table 4: 1 layer, $A \in \mathbb{R}^{6\times 6}$, $A = \begin{bmatrix} 0.2419 & 0.7956 & 0.0694 & 0.8522 & 0.5951 & 0.2218 \\ 0.0482 & 0.3083 & 0.4253 & 0.2143 & 0.6650 & 0.1313 \\ 0.0555 & 0.9675 & 0.5292 & 0.0502 & 0.5443 & 0.9281 \\ 0.2823 & 0.4443 & 0.4296 & 0.1750 & 0.1924 & 0.2726 \\ 0.8313 & 0.4863 & 0.2385 & 0.8544 & 0.1700 & 0.8923 \\ 0.7432 & 0.3947 & 0.6313 & 0.4729 & 0.4360 & 0.7347 \end{bmatrix}$, $b = \begin{bmatrix} 0.6350 \\ 0.9009 \\ 0.1451 \\ 0.9094 \\ 0.1103 \\ 0.8666 \end{bmatrix}$, $c = \begin{bmatrix} 0.2051 \\ 0.2648 \\ 0.5850 \\ 0.7410 \\ 0.5136 \\ 0.9041 \end{bmatrix}$, $\bar{x} = \begin{bmatrix} 0.0721 \\ 0.8379 \\ 0.9495 \\ 0.3981 \\ 0.5584 \\ 0.4610 \end{bmatrix}$

ε	Res.	Relaxation Methods														
		DR		SR	SCSRv1					SCSRv2					HRv1	HRv2
		ord2	ord1		ord2	ord3	ord4	ord5	ord6	ord2	ord3	ord4	ord5	ord6		
0e+00	obj.	**7.00e+00**	**1.44e+01**	7.00e+00	7.00e+00	7.01e+00	6.96e+00	6.91e+00	6.78e+00	7.00e+00	6.97e+00	6.94e+00	6.92e+00	6.68e+00	7.00e+00	7.00e+00
	time (s)	1.83e+03	3.13e-01	1.12e+02	2.56e+00	9.77e+00	1.28e+01	2.04e+01	3.16e+01	3.07e+00	5.16e+00	9.38e+00	2.13e+01	3.51e+01	6.14e+00	6.87e+00
	stat. (-1/0/1)	1	0	1	1	1	-1	-1	-1	1	-1	-1	-1	-1	1	1
	rat. (%)	-	1.05e+02	-4.32e-07	3.48e-02	1.91e-01	-5.16e-01	-1.24e+00	-3.10e+00	2.51e-02	-3.79e-01	-7.76e-01	-1.08e+00	-4.55e+00	7.98e-03	1.07e-02
1e-06	obj.	**7.00e+00**	**1.44e+01**	7.00e+00	7.01e+00	7.01e+00	6.96e+00	6.82e+00	6.79e+00	7.01e+00	6.97e+00	6.95e+00	6.94e+00	6.49e+00	7.00e+00	7.00e+00
	time (s)	2.39e+03	2.85e-01	1.43e+02	3.35e+00	9.67e+00	1.28e+01	1.76e+01	3.17e+01	3.75e+00	5.09e+00	9.33e+00	2.05e+01	3.25e+01	7.27e+00	8.20e+00
	stat. (-1/0/1)	1	0	1	1	0	-1	-1	-1	1	-1	-1	-1	-1	1	1
	rat. (%)	-	1.05e+02	-2.05e-05	1.14e-01	1.28e-01	-5.23e-01	-2.50e+00	-3.00e+00	9.62e-02	-3.81e-01	-7.16e-01	-8.76e-01	-7.22e+00	2.05e-02	2.42e-02
1e-05	obj.	**7.00e+00**	**1.44e+01**	7.00e+00	7.01e+00	7.01e+00	6.96e+00	6.95e+00	6.77e+00	7.01e+00	6.97e+00	6.94e+00	6.93e+00	6.72e+00	7.00e+00	7.00e+00
	time (s)	2.39e+03	2.92e-01	1.53e+02	4.29e+00	1.54e+01	1.25e+01	2.18e+01	3.36e+01	4.74e+00	5.12e+00	9.38e+00	2.12e+01	3.37e+01	8.58e+00	9.93e+00
	stat. (-1/0/1)	1	0	1	1	1	-1	-1	-1	1	-1	-1	-1	-1	1	1
	rat. (%)	-	1.05e+02	-3.47e-06	1.90e-01	2.22e-01	-5.87e-01	-6.88e-01	-3.23e+00	1.58e-01	-4.11e-01	-7.90e-01	-9.92e-01	-4.00e+00	2.36e-02	2.51e-02
1e-04	obj.	**7.00e+00**	**1.44e+01**	7.00e+00	7.02e+00	7.03e+00	6.95e+00	6.93e+00	6.57e+00	7.01e+00	6.99e+00	6.94e+00	6.89e+00	6.49e+00	7.00e+00	7.00e+00
	time (s)	2.33e+03	3.19e-01	1.47e+02	4.70e+00	1.17e+01	1.32e+01	2.30e+01	3.03e+01	5.48e+00	6.73e+00	9.39e+00	1.99e+01	3.13e+01	9.35e+00	1.10e+01
	stat. (-1/0/1)	1	0	1	1	1	-1	-1	-1	1	-1	-1	-1	-1	1	1
	rat. (%)	-	1.05e+02	1.49e-06	2.54e-01	4.90e-01	-6.70e-01	-9.68e-01	-6.15e+00	2.24e-01	-1.54e-01	-8.03e-01	-1.51e+00	-7.25e+00	2.33e-02	1.91e-02
1e-03	obj.	**7.01e+00**	**1.44e+01**	7.01e+00	7.04e+00	7.06e+00	6.93e+00	6.79e+00	6.56e+00	7.04e+00	7.00e+00	6.93e+00	6.90e+00	6.73e+00	7.01e+00	7.01e+00
	time (s)	2.28e+03	3.09e-01	1.43e+02	6.61e+00	1.47e+01	1.25e+01	1.75e+01	3.08e+01	8.37e+00	7.68e+00	9.23e+00	1.83e+01	3.35e+01	1.17e+01	1.23e+01
	stat. (-1/0/1)	1	0	1	1	1	-1	-1	-1	1	-1	-1	-1	-1	1	1
	rat. (%)	-	1.05e+02	-1.20e-06	5.36e-01	7.95e-01	-1.09e+00	-3.10e+00	-6.37e+00	4.48e-01	-1.45e-01	-1.14e+00	-1.58e+00	-4.00e+00	1.85e-02	1.31e-02
1e-02	obj.	**7.09e+00**	**1.44e+01**	7.09e+00	7.17e+00	7.17e+00	6.92e+00	6.84e+00	6.58e+00	7.16e+00	6.88e+00	6.90e+00	6.83e+00	6.63e+00	7.09e+00	7.09e+00
	time (s)	2.33e+03	3.42e-01	1.31e+02	5.48e+00	1.46e+01	1.63e+01	2.24e+01	3.29e+01	7.52e+00	7.15e+00	1.52e+01	1.94e+01	7.56e+01	2.33e+01	2.54e+01
	stat. (-1/0/1)	1	0	1	1	1	-1	-1	-1	1	-1	-1	-1	-1	1	1
	rat. (%)	-	1.03e+02	-3.28e-05	1.20e+00	1.21e+00	-2.44e+00	-3.45e+00	-7.18e+00	1.00e+00	-2.88e+00	-2.61e+00	-3.58e+00	-6.49e+00	1.49e-02	1.45e-02
1e-01	obj.	**7.90e+00**	**1.46e+01**	7.90e+00	8.55e+00	8.31e+00	6.21e+00	5.85e+00	5.77e+00	8.46e+00	7.45e+00	6.12e+00	5.95e+00	5.83e+00	7.90e+00	7.90e+00
	time (s)	2.57e+03	5.68e-01	1.63e+02	1.18e+01	2.79e+01	2.20e+01	4.40e+01	8.60e+01	1.64e+01	1.27e+01	1.98e+01	4.76e+01	1.03e+02	2.22e+01	2.26e+01
	stat. (-1/0/1)	1	0	1	0	0	0	-1	-1	0	-1	0	-1	-1	1	1
	rat. (%)	-	8.52e+01	-4.69e-05	8.21e+00	5.14e+00	-2.14e+01	-2.59e+01	-2.70e+01	7.11e+00	-5.71e+00	-2.26e+01	-2.47e+01	-2.62e+01	1.37e-02	1.24e-02
2e-01	obj.	**8.81e+00**	**1.49e+01**	8.81e+00	1.01e+01	8.44e+00	6.82e+00	5.83e+00	5.16e+00	9.95e+00	7.37e+00	5.29e+00	5.27e+00	4.94e+00	8.81e+00	8.81e+00
	time (s)	2.65e+03	4.71e-01	1.64e+02	8.93e+00	1.91e+01	3.21e+01	5.83e+01	6.18e+01	1.39e+01	1.03e+01	1.02e+01	2.91e+01	6.61e+01	1.31e+01	1.11e+01
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	-1	0	-1	-1	-1	-1	1	1
	rat. (%)	-	6.93e+01	-1.25e-04	1.50e+01	-4.15e+00	-2.25e+01	-3.38e+01	-4.15e+01	1.30e+01	-1.63e+01	-4.00e+01	-4.02e+01	-4.39e+01	1.11e-02	1.19e-02
3e-01	obj.	**9.71e+00**	**1.52e+01**	9.71e+00	1.17e+01	9.65e+00	6.05e+00	5.23e+00	4.87e+00	1.15e+01	8.51e+00	6.47e+00	4.71e+00	4.51e+00	9.71e+00	9.71e+00
	time (s)	2.27e+03	3.45e-01	1.43e+02	4.38e+00	9.47e+00	1.23e+01	2.10e+01	3.46e+01	7.00e+00	7.32e+00	1.13e+01	2.32e+01	4.39e+01	1.35e+01	1.27e+01
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	-1	0	-1	-1	-1	-1	1	1
	rat. (%)	-	5.63e+01	-1.52e-04	2.07e+01	-5.92e-01	-3.77e+01	-4.62e+01	-4.98e+01	1.80e+01	-1.24e+01	-3.34e+01	-5.15e+01	-5.35e+01	9.49e-03	1.04e-02
4e-01	obj.	**1.06e+01**	**1.54e+01**	1.06e+01	1.33e+01	1.02e+01	6.64e+00	5.80e+00	4.64e+00	1.30e+01	8.82e+00	7.00e+00	5.49e+00	4.96e+00	1.06e+01	1.06e+01
	time (s)	2.33e+03	3.11e-01	1.41e+02	5.79e+00	8.45e+00	1.21e+01	2.65e+01	3.31e+01	6.90e+00	6.31e+00	1.44e+01	2.15e+01	3.42e+01	1.34e+01	1.24e+01
	stat. (-1/0/1)	1	0	1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1
	rat. (%)	-	4.55e+01	-9.83e-05	2.55e+01	-4.22e+00	-3.75e+01	-4.54e+01	-5.63e+01	2.21e+01	-1.69e+01	-3.41e+01	-4.82e+01	-5.33e+01	1.03e-02	1.06e-02
6e-01	obj.	**1.24e+01**	**1.60e+01**	1.24e+01	1.65e+01	1.00e+01	6.50e+00	4.80e+00	4.39e+00	1.57e+01	7.57e+00	6.80e+00	5.16e+00	4.73e+00	1.24e+01	1.24e+01
	time (s)	5.17e+03	3.29e-01	1.72e+02	7.11e+00	8.02e+00	9.60e+00	1.71e+01	2.41e+01	3.74e+00	4.58e+00	1.19e+01	2.12e+01	2.85e+01	1.18e+01	1.42e+01
	stat. (-1/0/1)	1	0	1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	1	1
	rat. (%)	-	2.85e+01	-1.27e-02	3.28e+01	-1.94e+01	-4.77e+01	-6.14e+01	-6.47e+01	2.60e+01	-3.90e+01	-4.53e+01	-5.85e+01	-6.19e+01	5.89e-03	-1.58e-04
8e-01	obj.	**1.42e+01**	**1.65e+01**	1.33e+01	1.65e+01	8.45e+00	5.67e+00	4.82e+00	4.15e+00	1.70e+01	7.07e+00	5.84e+00	5.83e+00	5.11e+00	1.42e+01	1.42e+01
	time (s)	2.52e+03	2.96e-01	1.64e+02	2.14e+00	6.00e+00	8.54e+00	1.24e+01	2.19e+01	3.12e+00	4.06e+00	1.02e+01	1.76e+01	2.96e+01	1.23e+01	1.28e+01
	stat. (-1/0/1)	1	0	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1
	rat. (%)	-	1.56e+01	-6.21e+00	1.57e+01	-4.06e+01	-6.01e+01	-6.62e+01	-7.08e+01	1.92e+01	-5.03e+01	-5.90e+01	-5.90e+01	-6.41e+01	3.77e-03	-3.81e-02
1e+00	obj.	**1.60e+01**	**1.69e+01**	1.35e+01	1.57e+01	7.77e+00	5.50e+00	5.08e+00	4.18e+00	1.67e+01	7.13e+00	5.32e+00	5.98e+00	5.05e+00	1.60e+01	1.60e+01
	time (s)	2.61e+03	3.14e-01	1.82e+02	2.19e+00	4.71e+00	8.27e+00	1.44e+01	2.09e+01	2.24e+00	3.60e+00	6.59e+00	1.74e+01	2.85e+01	1.35e+01	1.17e+01
	stat. (-1/0/1)	1	0	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1
	rat. (%)	-	5.13e+00	-1.56e+01	-2.03e+00	-5.16e+01	-6.57e+01	-6.84e+01	-7.39e+01	3.84e+00	-5.56e+01	-6.69e+01	-6.27e+01	-6.85e+01	2.45e-03	-4.85e-02