Journal of Mathematical Imaging and Vision
https://doi.org/10.1007/s10851-022-01075-1

Toward a Sparsity Theory on Weighted Lattices

Nikolaos Tsilivis[1] · **Anastasios Tsiamis**[2] · **Petros Maragos**[1]

Received: 7 July 2021 / Accepted: 23 February 2022

Abstract
This paper studies issues of sparse representation in nonlinear vector spaces. In particular, we focus on complete weighted lattices Maragos (Math. Control Signals Syst 29: 21, 2017), a class of nonlinear spaces that generalizes mathematical morphology and max-plus algebra. We show how one can obtain sparse solutions to equations that arise in such spaces and discuss the computational hardness of the problem. Then, the focus shifts to max-plus algebra and, in particular, to sparse approximate solutions to max-plus equations. The developed theoretical tools allow us to make structured arguments about the pruning of a special class of neural networks, called morphological neural networks.

Keywords Sparsity · Lattices · Mathematical morphology · Submodularity

1 Introduction

Mathematical morphology focuses on the shape and structure of an image. It was developed initially throughout the 1970s and 1980s and was successfully applied to nonlinear image and signal analysis, while its mathematical foundations were further consolidated and expanded during the 1990's. A central motivation for its development as an image processing paradigm lies at the following observation: contrary to acoustic signals that obey a linear superposition, *visual signals do not combine linearly* [15]. There is an inherent **ordering** in the way humans (or sensors) receive such signals:

> "…*any object that is seen hides those that are placed beyond it with respect to the viewer* …" ([30]).

Mathematically, this simple but key observation suggests a need for structures different than the usual vector spaces. The notion of ordering is central in lattices (order theory) and, indeed, lattices serve as the theoretical basis of mathematical morphology [15,16,29,31]. We refer interested readers to prior work [27,30,31] for a classical treatment of mathematical morphology, while recent articles [3,14,22] cover a wide range of theory and applications of Mathematical Morphology.

In our work, we leverage the unifying framework of weighted lattices [23,24] that generalizes mathematical morphology. Conceptually, a weighted lattice resembles a vector (linear) space whose vector addition is substituted by pointwise vector maximum/minimum and scalar multiplication by an arbitrary scalar operation (that distributes over maxima/minima). The crucial difference is the nonlinearity of maximum/minimum. This enriches the space with a natural ordering, but deprives it of inverse operations. At the core of these spaces lies also the concept of *duality*, which stems from the concurrent treatment of both maximum and minimum operations. We study the problem of finding **sparse vectors** in such spaces. Similar to the study of sparsity in linear algebra [11], sparse vectors in weighted lattices represent the simplest explanations of our equations. It can be shown [24] that the archetypal morphological image operators (*dilations* and *erosions*) admit compact representations on a weighted lattice through simple matrix equations. Thus, by finding sparse solutions in their equations, we are able to find the *simplest* input that produces a specified output in a morphological system.

The most well-known special case that arises in this weighted lattice framework is *max-plus algebra*. Max-plus algebra also stems from the max-plus or tropical semiring that forms the arithmetic of tropical geometry [20]. Its two

✉ Nikolaos Tsilivis
ntsilivis96@gmail.com

Anastasios Tsiamis
atsiamis@seas.upenn.edu

Petros Maragos
maragos@cs.ntua.gr

1 School of ECE, National Technical University of Athens, Athens, Greece

2 ESE Department, SEAS, University of Pennsylvania, Philadelphia, USA

Published online: 07 April 2022

key operations is maximum and addition and it is defined over $\mathbb{R} \cup \{-\infty\}$. A variety of problems from optimization and optimal control have been solved via a max-plus treatment [1,4,13], while recently a connection between max-plus algebra and several areas of machine learning has been established [25,32,38]. Recently, sparsity in max-plus algebra was introduced [34]. In particular, it was shown that finding the sparsest solution to a max-plus equation is equivalent to the minimum set cover problem and, thus, NP-complete. Additionally, the problem of sparse ℓ_1 approximate solutions was studied, together with that of the recovery of a sparse vector. On a related note, the connection between max-plus equations and set covers has been observed before [6], while the pruning of the optimal value function in optimal control can be viewed as the search of an exact sparse representation of a max-plus basis of functions [13]. Herein, we extend ideas from max-plus algebra to any weighted lattice. Such a generic treatment will allow one to leverage sparse representations in more general spaces, such as the max-min ones that have been utilized for ultrametric image processing [2].

This work makes the following contributions:

- It introduces the concept of sparsity on a large class of nonlinear vector spaces.
- It studies the problem of finding sparse solutions to equations arising in these spaces.
- Focusing on max-algebra, it poses and solves *generalized* problems of computing the sparsest approximate solution to a matrix max-plus equation. Contrary to prior work, this allows the approximation error to be measured by any ℓ_p norm, while in the case of ℓ_∞ we consider a more "natural" optimization problem.
- Finally, we briefly present an application of the theory in the pruning of morphological neural networks.

The current paper is an extended version of our prior work [35], and its novel part involves the generalization of sparsity on Weighted Lattices (as opposed to only in max-plus algebra)—see Sects. 2.1 and 3, together with numerical computations at the end of Sect. 4.2. In particular, we address a hyperparameter selection problem of the ℓ_∞ optimization problem, while we, also, confirm the theoretical predictions for the approximation error of the resultant vector.

2 Background Concepts

Notation. We use roman letters for functions, signals and their arguments and Greek letters mainly for operators. Also, boldface roman letters for vectors (lowcase) and matrices (capital). Let $[n] = \{1, 2, \dots, n\}$.

2.1 Lattice Spaces

A *partially ordered set*, briefly **poset** $(\mathcal{P}, \leq)$, is a set $\mathcal{P}$ equipped with a binary relation $\leq$ that is a partial ordering on $\mathcal{P}$, i.e., $\leq$ is reflexive, antisymmetric and transitive. If $S \subseteq \mathcal{P}$, then an element $B \in \mathcal{P}$ is called an upper bound of S if $X \leq B$ for all $X \in S$. The *least upper bound* of S (if it exists) is called its **supremum** and is often denoted by $\bigvee S$. Likewise, we define lower bounds, *greatest lower bounds* or **infima** of S ($\bigwedge S$).

A **lattice** $(\mathcal{L}, \leq)$ is a poset whose finite subsets have a supremum and an infimum. A lattice $(\mathcal{L}, \leq)$ is *complete* if each of its subsets (even infinite) has a supremum and an infimum in $\mathcal{L}$. We denote the supremum and infimum of $\mathcal{L}$ by $\top = \bigvee \mathcal{L}$ and $\bot = \bigwedge \mathcal{L}$, respectively.

From any lattice $\mathcal{L}$, we can construct a new specific one in the following way: let $\mathcal{O}(\mathcal{L})$ be the set of all functions $f : \mathcal{L} \to \mathcal{L}$. Then, equipped with elementwise partial ordering $\preceq$ and elementwise supremum and infimum, this set is also a lattice. We denote it by $(\mathcal{O}(\mathcal{L}), \preceq)$ and is called an *operator lattice*. Special elements of this lattice that will be relevant to our analysis are the following:

- The identity operator $\mathbf{id}$: $\mathbf{id}(X) = X$ forall $X \in \mathcal{L}$,
- Extensive operators ψ : $\mathbf{id} \preceq \psi$,
- Antiextensive operators ψ : $\psi \preceq \mathbf{id}$,
- Idempotent operators ψ : $\psi^2 = \psi$ (where $\psi^2(X) = \psi(\psi(X))$),
- Increasing operators ψ : $X \leq Y \Rightarrow \psi(X) \preceq \psi(Y)$.

The fundamental blocks of morphological operators consist of four classes of increasing operators, namely **dilations**, **erosions**, **opening** and **closing**. An operator δ is called dilation if $\delta(\bigvee_i X_i) = \bigvee_i \delta(X_i)$ for any collection $\{X_i\}$. Dually, an operator ε is called erosion if $\varepsilon(\bigwedge_i X_i) = \bigwedge_i \varepsilon(X_i)$ for any collection $\{X_i\}$. An operator is an opening if it is increasing, antiextensive and idempotent, while an operator is a closing if it is increasing, extensive and idempotent. Such operators, and combinations of them, have been applied successfully to several problems in image processing and computer vision. Central to the analysis of complete lattices and mathematical morphology is, also, the following notion that pairs an operator with another which resembles its inverse. The pair (δ, ε) of operators on a complete lattice $\mathcal{L}$ is an **adjunction** on $\mathcal{L}$ if $\delta(X) \leq Y \iff X \leq \varepsilon(Y)$ for all $X, Y \in \mathcal{L}$.

We enrich the lattice structure with two additional binary operations. An algebra $(\mathcal{K}, \vee, \wedge, \star, \star')$ is called **clodum** (*complete lattice-ordered double monoid*) if:

(C1) $(\mathcal{K}, \vee, \wedge)$ is a complete, distributive lattice.
(C2) $(\mathcal{K}, \star)$ is a monoid where $\star$ distributes over suprema, that is $a\star(\bigvee_i x_i) = \bigvee_i (a\star x_i)$ for all $a, x_i \in \mathcal{K}$.
(C3) $(\mathcal{K}, \star')$ is a monoid where $\star'$ distributes over infima, that is $a\star'(\bigwedge_i x_i) = \bigwedge_i (a\star' x_i)$ for all $a, x_i \in \mathcal{K}$.

Examples of such spaces are

- The *Max-plus* space $(\overline{\mathbb{R}}, \vee, \wedge, +, +')$, where $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ and $+, +'$ are identical for real numbers, but $a+(-\infty) = (-\infty)+a = -\infty$ and $a+'\infty = \infty+'a = \infty$ for any $a \in \overline{\mathbb{R}}$. Note that here both $(\mathbb{R}, +)$ and $(\mathbb{R}, +')$ are groups, so this clodum has a richer structure. We call such a space **clog** (*complete lattice-ordered group*) and we refer to this specific clog as the **max-plus clog**.
- The **max-min clodum** $([0, 1], \vee, \wedge, \min, \max)$.

A clodum serves as a scalar arithmetic in our analysis. Based on it, we now define the nonlinear vector spaces that are the central object of our work. Consider a set V and a clodum $(\mathcal{K}, \vee, \wedge, \star, \star')$ with identity elements of e, e', respectively, and infimum, supremum O, I, respectively. V equipped with:

(a) a vector supremum operation $(\vee : V \times V \to V)$ and a vector infimum operation $(\wedge : V \times V \to V)$, and
(b) two operations of "scalar multiplication" $(\star : \mathcal{K} \times V \to V$ and $\star' : \mathcal{K} \times V \to V)$

is called a *Weighted Lattice space* over $\mathcal{K}$ if for any $X, Y, Z \in V$, $a, b \in \mathcal{K}$ the following axioms hold [24]:

(WL1) $X \vee Y \in V$ and $X \wedge Y \in V$.
(WL2) $X \vee Y = Y \vee X$ and $X \wedge Y = Y \wedge X$.
(WL3) $X \vee (Y \vee Z) = (X \vee Y) \vee Z$ and $X \wedge (Y \wedge Z) = (X \wedge Y) \wedge Z$.
(WL4) $\exists O \in V : O \vee X = X, O \wedge X = O$ and $\exists I \in V : I \wedge X = X, I \vee X = I$.
(WL5) $X \leq Y \iff Y = X \vee Y$ and $X \leq Y \iff X = X \wedge Y$, where $\leq$ is a partial ordering of V.
(WL6) $X\vee(Y\wedge Z) = (X\wedge Y)\vee(X\wedge Z)$ and $X\wedge(Y\vee Z) = (X \wedge Y) \vee (X \wedge Z)$.
(WL7) $a\star X \in V$ and $a\star' X \in V$.
(WL8) $a\star(b\star X) = (a\star b)\star X$ and $a\star'(b\star' X) = (a\star' b)\star' X$.
(WL9) $e\star X = X$ and $e'\star' X = X$.
(WL10) $a\star(X \vee Y) = (a\star X) \vee (a\star Y)$ and $a\star'(X \wedge Y) = (a\star' X) \wedge (a\star' Y)$.
(WL11) $(a \vee b)\star X = (a\star X) \vee (b\star X)$ and $(a \wedge b)\star' X = (a\star' X) \wedge (b\star' X)$.

Additionally, if V is closed under infinite suprema and infima, and (WL10), (WL11) hold even for infinite collections of (scalar or vector) suprema and infima, then V is called a **Complete Weighted Lattice (CWL)** space.

One may notice the similarities of the aforementioned axioms with those of a vector (linear) space. In this work, we will focus on finite-dimensional cases where $V = \mathcal{K}^n, n \in \mathbb{N}$. For instance, over the max-plus clog $(\overline{\mathbb{R}}, \vee, \wedge, +, +')$ consider the set $V = \overline{\mathbb{R}}^n$ of n-dimensional vectors, the elementwise supremum $\vee$ and infimum $\wedge$ operations, and the extension of $+, +'$: $a+\mathbf{v} = [a+v_i]_{i=1}^n$, $a+'\mathbf{v} = [a+'v_i]_{i=1}^n$ for all $a \in \overline{\mathbb{R}}$, $\mathbf{v} \in \overline{\mathbb{R}}^n$. Then V is a complete weighted lattice, as it satisfies axioms (WL1-WL11) and the completeness properties. This space includes both max-plus and min-plus algebras, while the concurrent treatment of them highlights the *duality* of their principles.

The main operators that arise in such spaces are max-$\star$ and min-$\star'$ multiplications of a matrix $\mathbf{A} \in \mathcal{K}^{m\times n}$ with a vector $\mathbf{x} \in \mathcal{K}^n$:

$$\delta_{\mathbf{A}}(\mathbf{x}) \triangleq \mathbf{A} \boxed{\vee}\, \mathbf{x} = [\bigvee_{j=1}^{n} a_{ij}\star x_j]_{i=1}^m \tag{1}$$

and

$$\varepsilon_{\mathbf{A}}(\mathbf{x}) \triangleq \mathbf{A} \boxed{\wedge}'\, \mathbf{x} = [\bigwedge_{j=1}^{n} a_{ij}\star' x_j]_{i=1}^m, \tag{2}$$

respectively. It can be proved that these operators are dilations and erosions, and they are the fundamental blocks of sup-$\star$ and inf-$\star$ superpositions (Theorem 1 at [24]). When these operators are viewed as systems that operate on signals, a representation theory of a whole class of nonlinear systems can be revealed. In systems theory, inverse problems are, typically, of interest. That is, one searches for the unknown input that produced a measured outcome through our system. In our treatment, this can be formulated as a max-$\star$ equation:

$$\delta_{\mathbf{A}}(\mathbf{x}) = \mathbf{b} \tag{3}$$

Of course the same can be done for systems satisfying inf-$\star$ superpositions via min-$\star'$ equations. Let $S(\mathbf{A}, \mathbf{b}) = \{\mathbf{x} \in \mathcal{K}^n \mid \delta_{\mathbf{A}}(\mathbf{x}) = \mathbf{b}\}$ be the set of its solutions. Next theorem associates to equation (3) a vector $\varepsilon_{\mathbf{A}^*}(\mathbf{b})$, where $\varepsilon_{\mathbf{A}^*}$ is called the *adjoint* operator of $\delta_{\mathbf{A}}$ and can be expressed as follows:

$$\varepsilon_{\mathbf{A}^*}(\mathbf{y}) = [\bigvee_{i=1}^{m} \zeta(a_{ij}, y_i)]_{j=1}^n, \tag{4}$$

where $\zeta(a, w) \triangleq \sup\{u \in \mathcal{K} : a\star u \leq w\}$. We say that $\delta_{\mathbf{A}}$ together with $\varepsilon_{\mathbf{A}^*}$ form an adjunction.

Theorem 1 *[24] If $S(\mathbf{A}, \mathbf{b}) \neq \emptyset$, then $\varepsilon_{\mathbf{A}^*}(\mathbf{b}) \in S(\mathbf{A}, \mathbf{b})$ and $\mathbf{x} \leq \varepsilon_{\mathbf{A}^*}(\mathbf{b})$ for all $\mathbf{x} \in S(\mathbf{A}, \mathbf{b})$.*

These are exactly the equations whose sparse solutions will be studied.

2.2 Max-Plus Algebra

Max-plus arithmetic consists of the idempotent semiring $(\mathbb{R}_{\max}, \max, +)$, where $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$ is equipped with the standard maximum and sum operations, respectively. *Max-plus algebra* consists of vector operations that extend max-plus arithmetic to $\mathbb{R}^n_{\max}$. They include the pointwise operations of partial ordering $\mathbf{x} \leq \mathbf{y}$ and pointwise supremum $\mathbf{x} \vee \mathbf{y} = [x_i \vee y_i]$, together with a class of vector transformations defined below. Max-plus algebra is isomorphic to the *tropical algebra*, namely the min-plus semiring $(\mathbb{R}_{\min}, \min, +)$, $\mathbb{R}_{\min} = \mathbb{R} \cup \{\infty\}$ when extended to $\mathbb{R}^n_{\min}$ in a similar fashion. Vector transformations on $\mathbb{R}^n_{\max}$ (resp. $\mathbb{R}^n_{\min}$) that distribute over max-plus (resp. min-plus) vector superpositions can be represented as a max-plus $\boxplus$ (resp. min-plus $\boxplus'$) product of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}_{\max} (\mathbb{R}^{m \times n}_{\min})$ with an input vector $\mathbf{x} \in \mathbb{R}^n_{\max} (\mathbb{R}^n_{\min})$:

$$[\mathbf{A} \boxplus \mathbf{x}]_i \triangleq \bigvee_{k=1}^{n} a_{ik} + x_k, \quad [\mathbf{A} \boxplus' \mathbf{x}]_i \triangleq \bigwedge_{k=1}^{n} a_{ik} + x_k \tag{5}$$

In the case of a max-plus matrix equation $\mathbf{A} \boxplus \mathbf{x} = \mathbf{b}$, there is a solution if and only if the vector

$$\hat{\mathbf{x}} = (-\mathbf{A})^{\intercal} \boxplus' \mathbf{b} \tag{6}$$

satisfies it [6,8,24]. We call this vector the *principal solution* of the equation. It also satisfies the inequality $\mathbf{A} \boxplus \hat{\mathbf{x}} \leq \mathbf{b}$.

2.3 Submodularity

Let U be a universe of elements. A set function $f : 2^U \to \mathbb{R}$ is called *submodular* [19] if $\forall A \subseteq B \subseteq U,\ k \notin B$ holds:

$$f(A \cup \{k\}) - f(A) \geq f(B \cup \{k\}) - f(B). \tag{7}$$

A set function f is called *supermodular* if $-f$ is submodular. Submodular functions occur as models of many real world evaluations in a number of fields and allow many hard combinatorial problems to be solved fast and with strong approximation guarantees [5,18]. It has been suggested that their importance in discrete optimization is similar to convex functions' in continuous optimization [19].

The following definition captures the idea of how far a given function is from being submodular and generalizes the notion of submodularity.

Definition 1 [9] Let U be a set and $f : 2^U \to \mathbb{R}^+$ be an increasing, non-negative, function. The submodularity ratio of f is

$$\gamma_{U,k}(f) \triangleq \min_{L \subseteq U, S : |S| \leq k, S \cap L = \emptyset} \frac{\sum_{x \in S} f(L \cup \{x\}) - f(L)}{f(L \cup S) - f(L)} \tag{8}$$

Proposition 1 *[9] An increasing function $f : 2^U \to \mathbb{R}$ is submodular if and only if $\gamma_{U,k}(f) \geq 1,\ \forall\, U, k$.*

In [9], the authors used the submodularity ratio to analyze the properties of greedy algorithms in discrete optimization problems with functions that are only approximately submodular ($\gamma \in (0, 1)$). They proved that the performance of the algorithms degrade gradually as a function of γ, thus allowing guarantees for a wider variety of objective functions.

3 Sparsity on Complete Weighted Lattices

Let $(\mathcal{K}^n, \vee, \wedge, \star, \star')$ be a complete weighted lattice over a scalar clodum $(\mathcal{K}, \vee, \wedge, \star, \star')$. First, we define sparsity in this space.

Definition 2 We call a vector $\mathbf{x} \in \mathcal{K}^n$ *sparse* if it contains many $\perp$ elements, where $\perp = \bigwedge \mathcal{K}$ (the infimum of $\mathcal{K}$). We define its *support set*, $\mathrm{supp}(\mathbf{x})$, to be the set of positions where vector $\mathbf{x}$ has values greater than $\perp$, that is $\mathrm{supp}(\mathbf{x}) = \{i \mid x_i \neq \perp\}$.

Let $\mathbf{A} \in \mathcal{K}^{m \times n}, \mathbf{b} \in \mathcal{K}^m$. Without loss of generality, we assume $b_i \neq \perp$ and $\bigvee_j a_{ij} \neq \perp$ for all $i \in [m]$. We are interested in the sparsest solution of equation (3). This can be expressed as the following optimization problem:

$$\begin{aligned} \arg\min_{\mathbf{x} \in \mathcal{K}^n} & |\mathrm{supp}(\mathbf{x})| \\ \text{s.t. } & \delta_{\mathbf{A}}(\mathbf{x}) = \mathbf{b}, \end{aligned} \tag{9}$$

where $\mathrm{supp}(\mathbf{x}) = \left\{ j \in [n] \mid x_j \neq \perp \right\}$ is the *support set* of $\mathbf{x}$. Of course, we will study (9) in the cases where $S(\mathbf{A}, \mathbf{b}) \neq \emptyset$. Therefore, from Theorem 1, we know that $\varepsilon_{\mathbf{A}^*}(\mathbf{b}) \in S(\mathbf{A}, \mathbf{b})$.

We now define for each $\mathbf{x} \in S(\mathbf{A}, \mathbf{b})$ n sets that reveal the combinatorial notion of (9):

$$I_j(\mathbf{x}) = \left\{ i \in [m] \mid\ a_{ij} \star x_j = b_i \right\}, \quad j \in [n]. \tag{10}$$

Each set $I_j(\mathbf{x})$ contains the rows of the equation that the j-th component of $\mathbf{x}$ satisfies (or *covers*). Notice that it must hold $\bigcup_{j \in [n]} I_j(\mathbf{x}) = [m]$ for any $\mathbf{x} \in S(\mathbf{A}, \mathbf{b})$ (each row of the equation must be satisfied).

Let now $C \subset [n]$ be a set of *minimum cardinality* for which $\bigcup_{j \in C} I_j(\varepsilon_{\mathbf{A}^*}(\mathbf{b})) = [m]$ holds. That is, C contains the bare minimum of elements of $\varepsilon_{\mathbf{A}^*}(\mathbf{b})$ that are needed to satisfy the constraint of (9). C defines a solution of Problem (9):

Proposition 2 *Vector* $\mathbf{x}^{\star} \in \mathcal{K}^n$*, defined as*

$$x_j^\star = \begin{cases} [\varepsilon_{\mathbf{A}^*}(\mathbf{b})]_j, & j \in C \\ \bot, & \textit{otherwise} \end{cases} \tag{11}$$

is a solution to Problem (9).

Proof Let $\mathbf{x}' \in S(\mathbf{A}, \mathbf{b})$, then $\bigcup_{j\in \text{supp}(\mathbf{x}')} I_j(\mathbf{x}') = [m]$. Then, it can be seen (Lemma 1) that also $\bigcup_{j\in \text{supp}(\mathbf{x}')} I_j(\varepsilon_{\mathbf{A}^*}(\mathbf{b})) = [m]$ holds. Thus, by the definition of C, it is $|\text{supp}(\mathbf{x}')| \geq |C| = |\text{supp}(\mathbf{x})|$.

The proof of Lemma 1 concludes the proof. □

Lemma 1 $I_j(\mathbf{x}) \subseteq I_j(\varepsilon_{\mathbf{A}^*}(\mathbf{b}))$ *for all* $\mathbf{x} \in S(\mathbf{A}, \mathbf{b})$.

Proof Let $\mathbf{x} \in S(\mathbf{A}, \mathbf{b})$ and $i \in I_j(\mathbf{x})$, then $a_{ij}\star x_j = b_i$. It holds $x_j \leq [\varepsilon_{\mathbf{A}^*}(\mathbf{b})]_j$ (from Theorem 1) and, since $\star$ is an increasing operation (as a dilation in $\mathcal{K}$), we have $a_{ij}\star x_j \leq a_{ij}\star[\varepsilon_{\mathbf{A}^*}(\mathbf{b})]_j \iff b_i \leq a_{ij}\star[\varepsilon_{\mathbf{A}^*}(\mathbf{b})]_j$. But $a_{ij}\star[\varepsilon_{\mathbf{A}^*}(\mathbf{b})]_j \leq b_i$ for all i, j (since $\varepsilon_{\mathbf{A}^*}(\mathbf{b}) \in S(\mathbf{A}, \mathbf{b})$), so $a_{ij}\star[\varepsilon_{\mathbf{A}^*}(\mathbf{b})]_j = b_i$, or $i \in I_j(\varepsilon_{\mathbf{A}^*}(\mathbf{b}))$. □

Therefore, finding the sparsest solution of a dilation equation $\delta_{\mathbf{A}}(\mathbf{x}) = \mathbf{b}$ requires the following steps:

- Computing an erosion, $\varepsilon_{\mathbf{A}^*}(\mathbf{b})$.
- Computing n sets of indices $I_j(\varepsilon_{\mathbf{A}^*}(\mathbf{b}))$ for all $j \in [n]$.
- Finding the *minimum set cover* of $[m]$ from the $\{I_j(\varepsilon_{\mathbf{A}^*}(\mathbf{b}))\}_{j=1}^n$ collection.

Example 1 Consider the scalar max-min clodum ([0, 1], ∨, ∧, min, max). We are searching for the sparsest solution of the following equation:

$$\begin{pmatrix} 1 & 0.4 & 0 \\ 0.3 & 1 & 0.5 \\ 0.7 & 0.2 & 1 \end{pmatrix} \boxtimes \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0.8 \\ 0.4 \\ 0.7 \end{pmatrix}. \tag{12}$$

The adjoint vector erosion is defined through the scalar erosion ζ (see (4) and [24] for further details):

$$\zeta(a, w) = \begin{cases} w, & w < a \\ 1, & w \geq a. \end{cases} \tag{13}$$

Hence:

$$\begin{aligned} \boldsymbol{\varepsilon}(\boldsymbol{b}) &= \begin{pmatrix} \zeta(1, 0.8) \wedge \zeta(0.3, 0.4) \wedge \zeta(0.7, 0.7) \\ \zeta(0.4, 0.8) \wedge \zeta(1, 0.4) \wedge \zeta(0.2, 0.7) \\ \zeta(0, 0.8) \wedge \zeta(0.5, 0.4) \wedge \zeta(1, 0.7) \end{pmatrix} \\ &= \begin{pmatrix} 0.8 \wedge 1 \wedge 1 \\ 1 \wedge 0.4 \wedge 1 \\ 1 \wedge 0.4 \wedge 0.7 \end{pmatrix} = \begin{pmatrix} 0.8 \\ 0.4 \\ 0.4 \end{pmatrix}, \end{aligned} \tag{14}$$

which is a solution to equation (12), since:

$$\begin{aligned} \delta(\varepsilon(\mathbf{b})) &= \begin{pmatrix} 1 & 0.4 & 0 \\ 0.3 & 1 & 0.5 \\ 0.7 & 0.2 & 1 \end{pmatrix} \boxtimes \begin{pmatrix} 0.8 \\ 0.4 \\ 0.4 \end{pmatrix} \\ &= \begin{pmatrix} \min(1, 0.8) \vee \min(0.4, 0.4) \vee \min(0, 0.4) \\ \min(0.3, 0.8) \vee \min(1, 0.4) \vee \min(0.5, 0.4) \\ \min(0.7, 0.8) \vee \min(0.2, 0.4) \vee \min(1, 0.4) \end{pmatrix} \\ &= \begin{pmatrix} 0.8 \vee 0.4 \vee 0 \\ 0.3 \vee 0.4 \vee 0.4 \\ 0.7 \vee 0.2 \vee 0.4 \end{pmatrix} = \begin{pmatrix} 0.8 \\ 0.4 \\ 0.7 \end{pmatrix}. \end{aligned} \tag{15}$$

Its three indices sets are $I_1(\varepsilon(\mathbf{b})) = \{1, 3\}$, $I_2(\varepsilon(\mathbf{b})) = I_3(\varepsilon(\mathbf{b})) = \{2\}$; thus, the sparsest solutions are the vectors $\left(0.8,\ 0.4,\ 0\right)^T$ and $\left(0.8,\ 0,\ 0.4\right)^T$. Notice that the sparsest solution is not unique.

Finding the minimum set cover of a set is an NP-complete problem. Nevertheless, we can resort to known polynomial time approximation algorithms for solving it. In particular, we can create iteratively a cover by picking each time the set that contains the most currently non-covered elements [36]. This requires $\mathcal{O}(n^2)$ time and produces a set cover whose size is at most $H(d_{\max})$ times the cardinality of the optimal cover (where $d_{\max}$ the largest cardinality of a $I_j(\varepsilon(\mathbf{b}))$ and $H(n)$ denotes the n-th harmonic number).

Now we show that Problem (9) is indeed NP-hard for a certain class of cloda, that satisfy the following condition:

Assumption I

$$\exists c \geq e, c \neq e : \bigvee \{u \in \mathcal{K} \mid c\star u = c\} = e \tag{16}$$

Note that each clog satisfies Assumption I, since $c\star u = c \iff u = e,\ \forall c$. However, it doesn't hold in the max-min clodum ([0, 1], ∨, ∧, min, max) that we just studied, because $\nexists c \geq e, c \neq e$ (since $e = \top$).

Theorem 2 *Let **k-sparse** be the decision problem counterpart of (9): Given matrices* $\mathbf{A} \in \mathcal{K}^{m\times n}$, $\mathbf{b} \in \mathcal{K}^m$ *with values from a scalar clodum* $\mathcal{K}$ *that satisfies Assumption I, is there a solution* $\mathbf{x} \in \mathcal{K}^n$ *of equation (3) with* $|\text{supp}(\mathbf{x})| \leq k$*? k-sparse is NP-hard.*

Proof Let k-Set Cover be the following problem: Let $U = [m]$ be a set and $\{M_j\}_{j=1}^n$ a collection of n other sets. Is there a set of indices S, such that $|S| \leq k$ and $\cup_{j\in S} M_j = U$? We will show that k-Set Cover reduces in polynomial time to k-sparse.

We construct $\mathbf{A} \in \mathcal{K}^{m\times n}$ and $\mathbf{b} \in \mathcal{K}^m$ as following:

$$\begin{aligned} a_{ij} &= \begin{cases} c, & i \in M_j \\ e, & \text{otherwise} \end{cases} \\ b_i &= c, \quad \forall i \in [m], \end{aligned} \tag{17}$$

where $c \in \mathcal{K}$ is a witness of formula (16). First, we calculate the vector erosion $\varepsilon_{\mathbf{A}^\star}(\mathbf{b})$ (denoted as $\varepsilon(\mathbf{b})$ for the sake of simplicity):

$$[\varepsilon(\mathbf{b})]_j = \bigwedge_i \zeta(a_{ji}, b_i) = \zeta(c, c) \wedge \zeta(e, c). \tag{18}$$

We have:

$$\begin{aligned} \zeta(e, c) &= \sup\{u \in \mathcal{K} \mid e\star u \leq c\} \\ &= \sup\{u \in \mathcal{K} \mid u \leq c\} = c, \\ \zeta(c, c) &= \sup\{u \in \mathcal{K} \mid c\star u \leq c\}. \end{aligned} \tag{19}$$

Let $u \geq e$, then $c\star u \geq c$, so if $c\star u \leq c$, then $c\star u = c$. Hence, from Assumption I: $\zeta(c, c) = e$. Therefore, $[\varepsilon(\mathbf{b})]_j = e, \ \forall j \in [n]$. We can now show the two directions of the reduction:

$\Rightarrow$: If $(U, \{M_j\}, k)$ has a cover S (meaning $\bigcup_{j \in S} M_j = U$ with $|S| \leq k$), then we define $\mathbf{x}$ as:

$$x_j = \begin{cases} [\varepsilon(\mathbf{b})]_j = e, & j \in S \\ \perp, & \text{otherwise} \end{cases} \tag{20}$$

and $\bigvee_j a_{ij}\star x_j = c\star e \vee e\star\perp = c$, so $\delta(\mathbf{x}) = \mathbf{b}$ with $|\text{supp}(\mathbf{x})| = |S| \leq k$; thus, $(\mathbf{A}, \mathbf{b})$ has a k-sparse solution.

$\Leftarrow$: In the opposite direction, if $(\mathbf{A}, \mathbf{b})$ has a k-sparse solution $\mathbf{x}$, then $\bigcup_{j \in \text{supp}(\mathbf{x})} I_j(\mathbf{x}) = [m]$. We have:

$$\begin{aligned} I_j(\mathbf{x}) &= \{i \mid a_{ij}\star x_j = b_i\} \\ &= \{i \mid a_{ij}\star x_j = c\} \end{aligned} \tag{21}$$

From Theorem 1, it is $x_j \leq [\varepsilon(\mathbf{b})]_j = e$; hence, $I_j(\mathbf{x}) = \{i \mid a_{ij} = c\} = M_j$. So, $(U, \{M_j\})$ has a cover of cardinality at most k. □

4 Sparsity in Max-plus Algebra

We now turn our attention to the max-plus clog (referred to as max-plus algebra for the rest of this section). Sparsity in this space was introduced in [34]. A vector $\mathbf{x} \in \mathbb{R}^n_{\max}$ is called *sparse* if it contains many $-\infty$ elements and we define its *support set*, $\text{supp}(\mathbf{x})$, to be the set of positions where vector $\mathbf{x}$ has finite values, that is $\text{supp}(\mathbf{x}) = \{i \mid x_i \neq -\infty\}$. We know from Theorem 2 (its max-plus version was proved in [34]) that finding an exact solution to a max-plus matrix equation:

$$\mathbf{A} \boxplus \mathbf{x} = \mathbf{b} \tag{22}$$

is an NP-hard problem. However, in many problems and applications an *approximate* solution may be sufficient (or even preferable). For instance, one may use max-plus equations to fit a convex function to data, by utilizing the representation as maxima of hyperplanes that each convex function admits [26]. Solving these equations gives us the exact hyperplanes that form our function, meaning that, in the presence of noisy data, an approximate solution might be beneficial. This is because such a solution would be less susceptible to the variance of the data.

4.1 Sparse ℓ_p, $p < \infty$ Approximate Solutions to Max-plus Equations

Hence, we will study now the problem of finding the sparsest ℓ_p *approximate* solution to the max-plus matrix equation $\mathbf{A} \boxplus \mathbf{x} = \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$. Such a solution should i) have minimum support set $\text{supp}(\mathbf{x})$, and ii) have small enough approximation error $\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}\|_p^p$, for some ℓ_p, $p < \infty$, norm. For this reason, given a prescribed constant ϵ, we formulate the following optimization problem:

$$\begin{aligned} \arg \min_{\mathbf{x} \in \mathbb{R}^n_{\max}} |\text{supp}(\mathbf{x})|, \ \text{s.t.} \ \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}\|_p^p \leq \epsilon, \ p < \infty \\ \mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b}. \end{aligned} \tag{23}$$

Note that we add an additional constraint $\mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b}$, also known as the "lateness" constraint. This constraint makes problem (23) more tractable; it enables the reformulation of problem (23) as a set optimization problem in (31). In many applications, this constraint is desirable–see [34]. However, in other situations, it might lead to less sparse solutions or higher residual error. A possible way to overcome this constraint is explored in Sect. 4.2.

Even with the additional lateness constraint, problem (23) is very hard to solve. Observe, for example, that when $\epsilon = 0$, we recover the initial exact problem that is NP-hard. Thus, we do not expect to find an efficient algorithm which solves (23) exactly. Instead, as we prove next, there is a polynomial time algorithm which finds an *approximate* solution, by leveraging its supermodular properties. By approximate solution, we now mean a vector that has a support set of approximately minimum cardinality. This approximation is quantified in Proposition 4.

First, let us show that the above problem can be formed as a discrete optimization problem over a set. The analysis is similar to prior work [34], where the case $p = 1$ was examined. For the rest of this section, let $J = [n]$.

Lemma 2 ***(Projection on the support set, ℓ_p case)*** *Let $T \subseteq J$,*

$$X_T = \{\mathbf{x} \in \mathbb{R}^n_{max} : \text{supp}(\mathbf{x}) = T, \ \mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b}\}. \tag{24}$$

and $\mathbf{x}|_T$ be defined as $\hat{\mathbf{x}}$ inside T and $-\infty$ otherwise, where $\hat{\mathbf{x}}$ is the principal solution defined in (6). Then, it holds:

1. $\mathbf{x}|_T \in X_T$.
2. $\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}|_T\|_p^p \leq \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}\|_p^p \quad \forall \mathbf{x} \in X_T$.

Proof 1. It suffices to show that $\mathbf{A} \boxplus \mathbf{x}|_T \leq \mathbf{b}$. For $j \in T$ it is $[\mathbf{x}|_T]_j = \hat{x}_j$ and for $j \in J \setminus T$, $[\mathbf{x}|_T]_j = -\infty \leq \hat{x}_j$. Thus,

$$\mathbf{x}|_T \leq \hat{\mathbf{x}} \iff \mathbf{A} \boxplus \mathbf{x}|_T \leq \mathbf{A} \boxplus \hat{\mathbf{x}} \implies \mathbf{A} \boxplus \mathbf{x}|_T \leq \mathbf{b}. \tag{25}$$

Hence, $\mathbf{x}|_T \in X_T$.

2. Let $\mathbf{x} \in X_T$, then $\mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b} \iff \mathbf{x} \leq \hat{\mathbf{x}}$, which implies (since both $\mathbf{x}$, $\mathbf{x}|_T$ have $-\infty$ values outside of T):

$$\mathbf{x} \leq \mathbf{x}|_T \iff \mathbf{b} - \mathbf{A} \boxplus \mathbf{x}|_T \leq \mathbf{b} - \mathbf{A} \boxplus \mathbf{x}. \tag{26}$$

Hence,

$$\begin{aligned} &\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}|_T\|_p^p \\ &= \sum_{j\in T} (\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}|_T)_j^p \\ &\leq \sum_{j\in T} (\mathbf{b} - \mathbf{A} \boxplus \mathbf{x})_j^p = \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}\|_p^p. \end{aligned} \tag{27}$$

□

The previous lemma informs us that we can fix the finite values of a solution of Problem (23) to be equal to those of the principal solution $\hat{\mathbf{x}}$. Indeed,

Proposition 3 *Let $\mathbf{x}_{OPT}$ be an optimal solution of (23), then we can construct a new one with values inside the support set equal to those of the principal solution $\hat{\mathbf{x}}$.*

Proof Define

$$\mathbf{z} = \begin{cases} \hat{x}_j, & j \in \text{supp}(\mathbf{x}_{\text{OPT}}) \\ -\infty, & \text{otherwise} \end{cases}, \tag{28}$$

then $\text{supp}(\mathbf{x}_{\text{OPT}}) = \text{supp}(\mathbf{z})$ and, from Lemma 2, $\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{z}\|_p^p \leq \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}_{\text{OPT}}\|_p^p$ and $\mathbf{A} \boxplus \mathbf{z} \leq \mathbf{b}$. Thus, $\mathbf{z}$ is also an optimal solution of (23). □

Therefore, the only variable that matters in Problem (23) is the support set. To further clarify this, let us proceed with the following definitions:

Definition 3 Let $T \subseteq J$ be a candidate support and let $\mathbf{A}_j$ denote the j-th column of $\mathbf{A}$. The error vector $\mathbf{e} : 2^J \to \mathbb{R}^m$ is defined as:

$$\mathbf{e}(T) = \begin{cases} \mathbf{b} - \bigvee_{j\in T} (\mathbf{A}_j + \hat{x}_j), & T \neq \emptyset \\ \bigvee_{j\in J} \mathbf{e}(\{j\}), & T = \emptyset. \end{cases} \tag{29}$$

Observe that for any T, it holds $\bigvee_{j\in T} (\mathbf{A}_j + \hat{x}_j) \leq \bigvee_{j\in J} (\mathbf{A}_j + \hat{x}_j) \leq \mathbf{b}$, which means that the above vector $\mathbf{e}(T) = (e_1(T), e_2(T), \ldots, e_m(T))^\intercal$ is always non-negative. We also define the corresponding error function $E_p : 2^J \to \mathbb{R}$ as:

$$E_p(T) = \|\mathbf{e}(T)\|_p^p = \sum_{i=1}^m (e_i(T))^p. \tag{30}$$

Problem (23) can now be written as:

$$\begin{aligned} &\arg\min_{T\subseteq J} |T| \\ &\quad \text{s.t. } E_p(T) \leq \epsilon \end{aligned} \tag{31}$$

The main results of this section are based on the following properties of E_p.

Theorem 3 *Error function E_p is decreasing and supermodular.*

Proof Regarding the monotonicity, let $\emptyset \neq C \subseteq B \subset J$, then

$$\bigvee_{j\in C} (\mathbf{A}_j + \hat{x}_j) \leq \bigvee_{j\in B} (\mathbf{A}_j + \hat{x}_j) \iff \mathbf{e}(B) \leq \mathbf{e}(C), \tag{32}$$

thus raising the, non-negative, components of the two vectors to the p-th power and adding the inequalities together yield $E_p(B) \leq E_p(C)$. The case for $C = \emptyset$ easily follows from the definition of $\mathbf{e}$.

Let $S, L \subseteq U \subseteq J$, with $|S| \leq K$, $S \cap L = \emptyset$ and define $f(U) = -E_p(U), \forall\, U$. Then:

$$\gamma_{U,K}(f) = \min_{L,S} \frac{\sum_{s_k\in S} f(L \cup \{s_k\}) - f(L)}{f(L \cup S) - f(L)}, \tag{33}$$

where $f(L) = \sum_{i=1}^m [b_i - \bigvee_{j\in L} (A_{ij} + \hat{x}_j)]^p$.

Let now I_1 be the set:

$$I_1 = \{i \mid \bigvee_{j\in L\cup S} (A_{ij} + \hat{x}_j) = \bigvee_{j\in L} (A_{ij} + \hat{x}_j)\} \tag{34}$$

and for each $s_k \in S$, we define two sets of indices:

$$\begin{aligned} I_2(s_k) = \{i \mid &\bigvee_{j\in L\cup\{s_k\}} (A_{ij} + \hat{x}_j) = \\ &\bigvee_{j\in L\cup S} (A_{ij} + \hat{x}_j) > \bigvee_{j\in L} (A_{ij} + \hat{x}_j)\} \end{aligned} \tag{35}$$

and

$$I_3(s_k) = \{i \mid \bigvee_{j\in L\cup S} (A_{ij} + \hat{x}_j) > \bigvee_{j\in L\cup\{s_k\}} (A_{ij} + \hat{x}_j) > \bigvee_{j\in L} (A_{ij} + \hat{x}_j)\}. \tag{36}$$

Then, if

$$\Sigma_1(L, S) = \sum_{s_k\in S} \sum_{i\in I_1, I_2(s_k)} \{-[b_i - \bigvee_{j\in L\cup\{s_k\}} (A_{ij} + \hat{x}_j)]^p + [b_i - \bigvee_{j\in L} (A_{ij} + \hat{x}_j)]^p\} \tag{37}$$

and

$$\Sigma_2(L, S) = \sum_{s_k\in S} \sum_{i\in I_3(s_k)} -[b_i - \bigvee_{j\in L\cup\{s_k\}} (A_{ij} + \hat{x}_j)]^p + [b_i - \bigvee_{j\in L} (A_{ij} + \hat{x}_j)]^p, \tag{38}$$

the ratio becomes:

$$\gamma_{U,K}(f) = \min_{L,S} \frac{\Sigma_1(L, S) + \Sigma_2(L, S)}{\Sigma_1(L, S)} \geq 1, \ \forall\, U, K \tag{39}$$

meaning (Proposition 1) that f is submodular or, equivalently, $E_p = -f$ is supermodular. □

Algorithm 1: Approximate solution of problem (23)

Input: $\mathbf{A}$, $\mathbf{b}$
Compute $\hat{\mathbf{x}} = (-\mathbf{A})^\intercal \boxplus' \mathbf{b}$
if $E_p(J) > \epsilon$ **then**
  return Infeasible
Set $T_0 = \emptyset$, $k = 0$
while $E_p(T_k) > \epsilon$ **do**
  $j = \arg\min_{s\in J\setminus T_k} E_p(T_k \cup \{s\})$
  $T_{k+1} = T_k \cup \{j\}$
  $k = k + 1$
end
$x_j = \hat{x}_j$, $j \in T_k$ and $x_j = -\infty$, otherwise
return $\mathbf{x}$, T_k

Setting $\tilde{E}_p(T) = \max(E_p(T), \epsilon)$[1] and leveraging the previous theorem, we are able to formulate problem (31), and thus the initial one (23), as a cardinality minimization problem subject to a supermodular equality constraint [36], which allows us to approximately solve it by the greedy Algorithm 1. Algorithm 1 selects greedily at each step the index j whose inclusion on the support set yields the greatest decrease on the error function. The calculation of the principal solution requires $\mathcal{O}(nm)$ time and the greedy selection of the support set of the solution costs $\mathcal{O}(n^2)$ time. We call the solutions of problem (23) *Sparse Greatest Lower Estimates* of $\mathbf{b}$. Regarding the approximation ratio between the optimal solution and the output of Algorithm 1, the following proposition holds.

Proposition 4 *Let $\mathbf{x}$ be the output of Algorithm 1 after $k > 0$ iterations of the inner while loop and T_k the respective support set. Then, if T^* is the support set of the optimal solution of (23), the following inequality holds:*

$$\frac{|T_k|}{|T^*|} \leq 1 + \log\left(\frac{m\Delta^p - \epsilon}{E_p(T_{k-1}) - \epsilon}\right), \tag{40}$$

where $\Delta = \bigvee_{i,j}(b_i - A_{ij} - \hat{x}_j)$.

Proof From [36], the following bound holds for the cardinality minimization problem subject to a supermodular and decreasing constraint, defined as function $f : 2^J \to \mathbb{R}$, by the greedy algorithm:

$$\frac{|T_k|}{|T^*|} \leq 1 + \log\left(\frac{f(\emptyset) - f(J)}{f(T_{k-1}) - f(J)}\right) \tag{41}$$

For our problem, it is $f = \tilde{E}_p$. Observe now that, since $k > 0$, $\tilde{E}_p(\emptyset) = E_p(\emptyset) \leq m\Delta^p$, $0 \leq \tilde{E}_p(J) = \epsilon$ and $\tilde{E}_p(T_{k-1}) > \epsilon$. Therefore, the result follows. □

The ratio warns us to expect less optimal and, thus, less sparse vectors when increasing the norm p that we use to measure the approximation. It also hints toward an inapproximability result when $p \to \infty$, which is formalized next.

4.2 Sparse Vectors with Minimum ℓ_∞ Errors

Although in some settings the $\mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b}$ constraint is needed [34], in other cases it could disqualify potentially sparser vectors from consideration. Omitting the constraint, on the other hand, makes it unclear how to search for minimum error solutions for any ℓ_p $(p < \infty)$ norm. For instance, it has recently been reported that it is NP-hard to determine if a given point is a local minimum for the ℓ_2 norm [17]. For that reason, we shift our attention to the case of $p = \infty$. It is well known [6,8] that the problem $\min_{\mathbf{x}\in\mathbb{R}^n_{\max}} \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}\|_\infty$ has a closed form solution; it can be calculated in $\mathcal{O}(nm)$ time by adding to the principal solution element-wise the half of its ℓ_∞ error. Note that this new vector does not necessarily satisfy $\mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b}$, so it shows a way to overcome the aforementioned limitation [2]

---

[1] The new, truncated, error function remains supermodular [18].

[2] An additional motivation for the adoption of the ℓ_∞ comes from recent work on Supermodular Optimization that shows that certain problems that are NP-hard under general ℓ_p normed error functions are actually solvable in polynomial time when $p = \infty$ [21].

First, let us demonstrate that problem (23), when considering the ℓ_∞ norm, becomes non-approximable by the greedy Algorithm 1. Hence, consider now the following optimization problem:

$$\begin{aligned} &\arg\min_{\mathbf{x}\in\mathbb{R}^n_{\max}} |\mathrm{supp}(\mathbf{x})| \\ &\quad \text{s.t. } \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}\|_\infty \leq \epsilon. \end{aligned} \tag{42}$$

Thanks to a similar construction as in the previous section, this problem can be recast as a set-search problem.

Lemma 3 ***(Projection on the support set,*** ℓ_∞**case)** *Let* $T \subseteq J$, $\mathbf{x}|_T$ *defined as* $\hat{\mathbf{x}}$ *inside* T *and* $-\infty$ *otherwise and* $\mathbf{x}^* = \mathbf{x}|_T + \frac{\|\mathbf{b}-\mathbf{A}\boxplus\mathbf{x}|_T\|_\infty}{2}$. *Then* $\forall\ \mathbf{z} \in \mathbb{R}^n_{max}$ *with* $\mathrm{supp}(\mathbf{z}) = T$, *it holds:*

$$\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{z}\|_\infty \geq \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}^*\|_\infty = \frac{\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}|_T\|_\infty}{2}. \tag{43}$$

proof (Sketch) By fixing the support set of the considered vectors equal to T, equivalently we omit the columns and indices of $\mathbf{A}$ and $\mathbf{x}$, respectively, that do not belong in T (since they will not be considered at the evaluation of the maximum). By doing so, we get a new equation with same vector $\mathbf{b}$ and restricted $\mathbf{A}, \mathbf{x}$. The vector $\mathbf{x}^*$ that minimizes the ℓ_∞ error of this equation is obtained from its principal solution plus the half of its ℓ_∞ error. But now observe that the new principal solution shares the same values with the original principal solution (follows from Lemma 2) inside T, which is exactly vector $\mathbf{x}|_T$. Extending $\mathbf{x}^*$ back to $\mathbb{R}^n_{\max}$ yields the result. □

So, a similar result to Proposition 3 holds.

Proposition 5 *Let* $\mathbf{x}_{OPT}$ *be an optimal solution of (42), then we can construct a new one with values inside the support set equal to those of the principal solution* $\hat{\mathbf{x}}$ *plus the half of its* ℓ_∞ *error.*

By defining $E_\infty(T) = \frac{\|\mathbf{b}-\mathbf{A}\boxplus\mathbf{x}|_T\|_\infty}{2}$, (42) becomes:

$$\begin{aligned} &\arg\min_{T\subseteq J} |T| \\ &\quad \text{s.t. } E_\infty(T) \leq \epsilon \end{aligned} \tag{44}$$

Unfortunately this problem does not admit an approximate solution by the greedy Algorithm 1 (to be precise, the modified version of Algorithm 1 when E_p becomes E_∞), as its error function, although decreasing, is not supermodular. The following example also reveals that the submodularity ratio (8) of E_∞ is 0. Therefore, it is not even approximately supermodular and a solution by Algorithm 1 can be arbitrarily bad [9].

Example 2 Let $A = \begin{pmatrix} 0 & 5 & 2 \\ 4 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix}$, then principal solution $\hat{\mathbf{x}}$ is:

$$\hat{\mathbf{x}} = \begin{pmatrix} 0 & -4 & 0 \\ -5 & -1 & -1 \\ -2 & 0 & 0 \end{pmatrix} \boxplus' \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -3 \\ -2 \\ 0 \end{pmatrix}.$$

We calculate now the error function on different sets:

- When $T = \{3\}$, then $\hat{\mathbf{x}}|_{\{3\}} = (-\infty, -\infty, 0)^\mathsf{T}$ and $E_\infty(\{3\}) = \frac{1}{2}\|\mathbf{b} - \bigvee_{j\in\{3\}}(\mathbf{A}_j + \hat{x}|_{\{3\},j})\|_\infty = \frac{1}{2}\| \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \|_\infty = \frac{1}{2}$.
- When $T = \{1, 3\}$, $E_\infty(\{1,3\}) = \frac{1}{2}\| \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} -3 \\ 1 \\ -3 \end{pmatrix} \bigvee \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \|_\infty = \frac{1}{2}$.
- When $T = \{2, 3\}$, $E_\infty(\{2,3\}) = \frac{1}{2}\| \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 3 \\ -1 \\ -1 \end{pmatrix} \bigvee \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \|_\infty = \frac{1}{2}$.
- When $T = \{1, 2, 3\}$, $E_\infty(\{1,2,3\}) = \frac{1}{2}\| \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} -3 \\ 1 \\ -3 \end{pmatrix} \bigvee \begin{pmatrix} 3 \\ -1 \\ -1 \end{pmatrix} \bigvee \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \|_\infty = 0$.

Let now $f = -E_\infty$, $L = \{3\}$, $S = \{1, 2\}$, then, by (8), we have:

$$\begin{aligned} &\frac{f(\{3\} \cup \{1\} - f(\{3\}) + f(\{3\} \cup \{2\}) - f(\{3\})}{f(\{3\} \cup \{1,2\}) - f(\{3\})} \\ &= \frac{-1/2 + 1/2 - 1/2 + 1/2}{0 + 1/2} = 0, \end{aligned} \tag{45}$$

meaning that f has submodularity ratio 0 or E_∞ is not even approximately supermodular.

Although the previous discussion denies from problem (42) a greedy solution with any guarantees, we propose next a practical alternative to get a sparse enough vector.

We first obtain a sparse vector $\mathbf{x}_{p,\epsilon}$ by solving problem (23) for a fixed ℓ_p, $p < \infty$, norm. Then, we add to this vector element-wise half of its ℓ_∞ error $\|\mathbf{b}-\mathbf{A}\boxplus\mathbf{x}_{p,\epsilon}\|_\infty/2$. Interestingly, this new solution minimizes the ℓ_∞ error among all vectors with the same support, as formalized in the following result. We will call such a solution *Sparse Minimum Max Absolute Error (SMMAE)* estimate of $\mathbf{b}$ and will denote it by $\mathbf{x}_{\text{SMMAE}}$.

Proposition 6 *Let* $\mathbf{x}_{\text{SMMAE}} \in \mathbb{R}^n_{max}$ *be defined as:*

$$\mathbf{x}_{\text{SMMAE}} = \mathbf{x}_{p,\epsilon} + \frac{\|\mathbf{b}-\mathbf{A}\boxplus\mathbf{x}_{p,\epsilon}\|_\infty}{2}, \tag{46}$$

where $\mathbf{x}_{p,\epsilon}$ *is a solution of problem (23) with fixed* (p,ϵ). *Then* $\forall\, \mathbf{z} \in \mathbb{R}^n_{max}$ *with* $\text{supp}(\mathbf{z}) = \text{supp}(\mathbf{x}_{p,\epsilon})$, *it holds*

$$\begin{aligned}\|\mathbf{b}-\mathbf{A}\boxplus\mathbf{z}\|_\infty &\geq \|\mathbf{b}-\mathbf{A}\boxplus\mathbf{x}_{\text{SMMAE}}\|_\infty \\ &= \frac{\|\mathbf{b}-\mathbf{A}\boxplus\mathbf{x}_{p,\epsilon}\|_\infty}{2}\end{aligned} \tag{47}$$

and, also,

$$\|\mathbf{b}-\mathbf{A}\boxplus\mathbf{x}_{\text{SMMAE}}\|_\infty \leq \frac{\sqrt[p]{\epsilon}}{2}. \tag{48}$$

Proof Observe that $\mathbf{x}_{p,\epsilon}$ is equal to the principal solution $\hat{\mathbf{x}}$ inside $\text{supp}(\mathbf{x}_{p,\epsilon})$. So the first inequality holds from Lemma 3. Regarding the second one, we have:

$$\begin{aligned}\|\mathbf{b}-\mathbf{A}\boxplus\mathbf{x}_{\text{SMMAE}}\|_\infty &= \frac{\|\mathbf{b}-\mathbf{A}\boxplus\mathbf{x}_{p,\epsilon}\|_\infty}{2} \\ &= \frac{\bigvee_i (b_i - [\mathbf{A}\boxplus\mathbf{x}_{p,\epsilon}]_i)}{2}.\end{aligned} \tag{49}$$

But, notice that:

$$\begin{aligned}\left(\bigvee_i b_i - [\mathbf{A}\boxplus\mathbf{x}_{p,\epsilon}]_i\right)^p &= \bigvee_i (b_i - [\mathbf{A}\boxplus\mathbf{x}_{p,\epsilon}]_i)^p \\ &\leq \sum_i (b_i - [\mathbf{A}\boxplus\mathbf{x}_{p,\epsilon}]_i)^p \leq \epsilon,\end{aligned} \tag{50}$$

so

$$\bigvee_i (b_i - [\mathbf{A}\boxplus\mathbf{x}_{p,\epsilon}]_i) \leq \sqrt[p]{\epsilon} \tag{51}$$

and the result follows from (49). Note that the bound tightens, as p increases. □

The above method provides sparse vectors that are approximate solutions of the equation with respect to the ℓ_∞ norm without the need of the lateness constraint. After computing $\mathbf{x}_{p,\epsilon}$, $\mathbf{x}_{\text{SMMAE}}$ requires $\mathcal{O}(m|\text{supp}(\mathbf{x}_{p,\epsilon})| + |\text{supp}(\mathbf{x}_{p,\epsilon})|) = \mathcal{O}((m+1)|\text{supp}(\mathbf{x}_{p,\epsilon})|)$ time.

The previous approach, however, depends on the norm order p that we use to compute the sparse vector $\mathbf{x}_{p,\epsilon}$. Since the previous theoretical treatment does not give a definite answer on how to make this choice, we address the problem numerically, and we find that in general a higher order norm yields smaller supports, without affecting the fidelity of the solutions (measured by the ℓ_∞ error of the approximation).

We generate random 50×100 matrices $\mathbf{A}$ with elements taking values in the set $\{0,\ldots,98\}$ and 50-dimensional vectors $\mathbf{b}$ with values in $\{0,\ldots,105\}$. We first compute the solution of problem (23) for $p = 1$, setting $\epsilon = \|\mathbf{b} - \mathbf{A}\boxplus\hat{\mathbf{x}}\|_1 + 1$ to guarantee feasibility ($\hat{\mathbf{x}}$ is the principal solution defined in (6)). Let $\mathbf{x}_1$ be the resultant sparse vector. Then, we solve (23) for increasing values of p and we set $\epsilon = \|\mathbf{b}-\mathbf{A}\boxplus\mathbf{x}_1\|_p$ in order to force the solution to be similar to $\mathbf{x}_1$ (so that the comparison between different values of p is fair).

First, we assess the sparsity of the resultant vectors (Fig. 1 (left)). We find that the larger the norm order p, the sparser the vector is, even though the behavior is not strictly monotonic (it has no reason to be from the theory). Second, we examine the ℓ_∞ approximation error of this vector across p (Fig. 1 (right)). Interestingly, the decreased cardinality of the support set of the solution does not harm approximation performance as measured by ℓ_∞. We see almost no increase in the error, as p increases. These two insights together suggest that, when searching for sparse vectors with small ℓ_∞ error, it is beneficial to consider large norm order p to find $\mathbf{x}_{p,\epsilon}$ before adding to it the half of its ℓ_∞ approximation error. Finally, in Fig. 1 (right) we also show the ℓ_∞ error of the SMMAE estimate of $\mathbf{b}$, and, as the theory predicts, it is exactly half of its corresponding $\mathbf{x}_{p,\epsilon}$ vector.

5 Application in Neural Network Pruning

Recently, there has been a renewed interest in Morphological Neural Networks [7,12,28,33] which consist of neural networks with layers performing morphological operations (dilations or erosions). While they are theoretically appealing because of the success that morphological operations had in traditional computer vision tasks and the universal approximation property that these networks possess, they have also shown an ability to be pruned and produce interpretable models [7,10,39]. Herein, we propose a way to do this systematically, by formulating the pruning problem as a system of max-plus equations and leveraging the theory of the previous section.

Let a morphological network be a multi-layered network that contains layers of linear transformations followed by max-plus operations. The authors of [39] call this sequence of layers as a *Max-plus block*. If $\mathbf{x} \in \mathbb{R}^d$ represents the input and

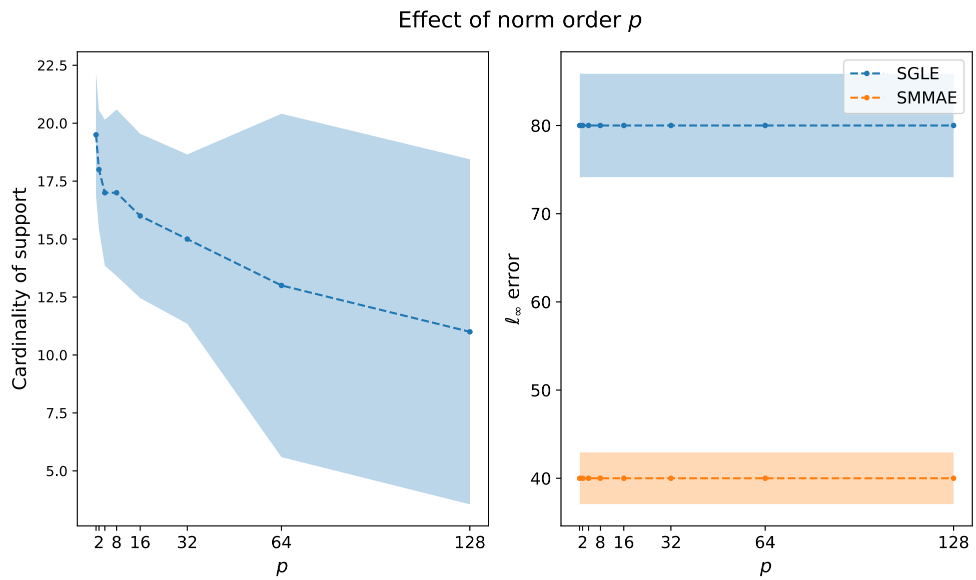
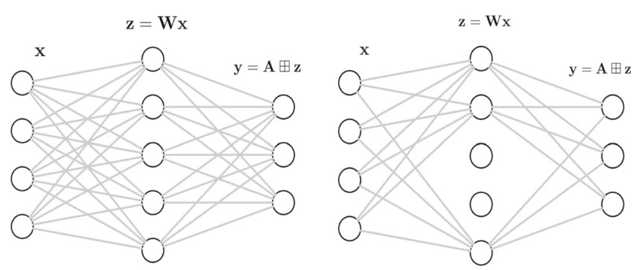


Fig. 1 Effect of norm on the support and the ℓ_∞ error of the solution. We report median values, while the shaded regions denote one standard deviation across 100 different random pairs of $\mathbf{A}$, $\mathbf{b}$ matrices. **Left:** Cardinality of support as a function of the selected norm order p. **Right:** ℓ_∞ approximation error as a function of p. It shows the error of (1) solutions of (23) (*SGLE*), and (2) the optimal max absolute error estimates defined in (46) (*SMMAE*)



(a) A simple Max-plus block with $d = 4, n = 5, k = 3$.

(b) The same Max-plus block, after pruning two neurons from its first layer.

Fig. 2 Morphological neural networks

k is the output's dimension, then a simple network of 1 Max-plus block (see Fig. 2) performs the following operations:

$$\begin{aligned} \mathbf{z} &= \mathbf{W}\mathbf{x}, \\ \mathbf{y} &= \mathbf{A} \boxplus \mathbf{z}, \end{aligned} \tag{52}$$

where $\mathbf{W} \in \mathbb{R}^{n \times d}$ and $\mathbf{A} \in \mathbb{R}_{\max}^{k \times n}$. Suppose now that this network has been trained successfully, possibly with a redundant number n of neurons and we wish to maintain its accuracy while minimizing its size. For each training sample $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, it holds $\tilde{\mathbf{y}}^{(i)} = \mathbf{A} \boxplus \mathbf{z}^{(i)}$, where $\tilde{\mathbf{y}}^{(i)}$ is the network's prediction. We keep now fixed the prediction (that we wish to maintain) and the matrix $\mathbf{A}$ and we find a sparse approximate solution of this equation with respect to vector $\mathbf{z}^{(i)}$. Observe that if a value of $\mathbf{z}$ equals $-\infty$, then equivalently we can set the corresponding column of $\mathbf{A}$ to -∞, thus pruning the whole unit. Of course, this naive technique would prune units that are important for other training samples. We propose overcoming this by finding sparse solutions for each sample, counting how many times each index $j \in \{1, \ldots, n\}$ has been found inside the support set of a solution and then keeping only the k most frequent values.

The proposed method enables one to fully prune neurons from any layer that performs a max-plus operation, without harming its performance, and produce compact, interpretable networks. We support the above analysis by providing an experiment on MNIST and FashionMNIST datasets. Both datasets are balanced and contain 10 different classes.

Example 3 We train 2 networks for each dataset, containing 1 max-plus block with 64 and 128 neurons, respectively, inside the hidden layer, for 20 epochs with Stochastic Gradient Descent optimizing the Cross Entropy Loss.

After the training, we pick at random 10000 samples from the training dataset (which account to 17% of the whole training data), we perform a forward pass over the network for each one of them to obtain predictions and then run Algorithm 1 with $p = 20$ and $\epsilon = 2^{20}$, so that we acquire sparse vectors $\mathbf{z}$ (and their support sets). Then, we simply find the 10 (same as the number of classes) most frequent indices inside the support sets of the solutions, keep the units that correspond to those indices and prune the rest of them. As shown in Table 1, all of the pruned networks record the same test accuracy as the full models, while having 54 and 118 *less* neurons, respectively. Note that trying to train from scratch networks with $n = 10$, under the same training setting, produces significantly worse results (around 60% for both datasets for 5 different random seeds).

Table 1 Test set accuracy before and after pruning

	MNIST		FashionMNIST	
	64	128	64	128
Full model	92.21	92.17	79.27	83.37
Pruned ($n = 10$)	92.21	92.17	79.27	83.37

6 Conclusions and Future Work

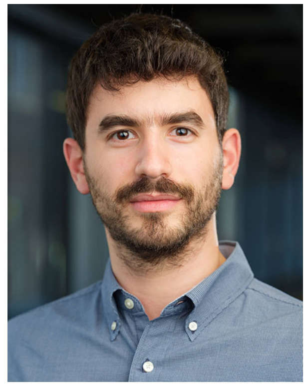
In this work, we performed a few steps toward a complete theory of sparsity in a specific class of nonlinear spaces, called complete weighted lattices. We introduced the concept of sparsity, explained how one can find sparse solutions to equations of this kind and discussed the computational aspects of it. Then, we focused on the case of max-plus algebra, a specific subcase of such a space. We posed generalized optimization problems for the computation of sparse approximate solutions of max-plus matrix equations and proposed methods for solving them. We briefly presented then how this sparsity framework might be utilized in the pruning of a special class of Neural Networks. It is a subject of future work to expand the general sparse framework to cover sparse approximate solutions to any weighted lattice, study other notions of approximations such as the range semi-metric, investigate the applications of sparsity in more areas of applications, and perform further experiments on the proposed pruning technique in deeper networks or more general, max-min neural networks [37].

References

1. Akian, M., Gaubert, S., Guterman, A.: Tropical polyhedra are equivalent to mean payoff games. Int'l J. Algebra and Computation **22**(1) (2012)
2. Angulo, J.: Eigenfunctions of ultrametric morphological openings and closings. In: Lindblad, Joakim, Malmberg, Filip, Sladoje, Nataša (eds.) Discrete geometry and mathematical morphology, pp. 499–511. Springer International Publishing, Cham (2021)
3. Angulo, J.: Some open questions on morphological operators and representations in the deep learning era. In: Lindblad, Joakim, Malmberg, Filip, Sladoje, Nataša (eds.) Discrete geometry and mathematical morphology, pp. 3–19. Springer International Publishing, Cham (2021)
4. Baccelli, F., Cohen, G., Olsder, G.J., Quadrat, J.P.: Synchronization and linearity: an algebra for discrete event systems. Wiley, Hoboken (1992)
5. Bach, F.: Learning with submodular functions: a convex optimization perspective (2013)
6. Butkovič, P.: Max-linear systems: theory and algorithms. Springer, Berlin (2010)
7. Charisopoulos V, Maragos P: Morphological perceptrons: geometry and training algorithms. In: J. Angulo, et al. (eds.) Proc. Int'l Symp. Mathematical Morphology (ISMM), *LNCS*, vol. 10225, pp. 3–15. Springer, Cham (2017)
8. Cuninghame-Green, R.: Minimax algebra. Springer-Verlag, Berlin (1979)
9. Das, A., Kempe, D.: Approximate submodularity and its applications: aubset selection, sparse approximation and dictionary selection. J. Mach. Learn. Res. **19**(1), 74–107 (2018)
10. Dimitriadis, N., Maragos, P.: Advances in morphological neural networks: training, pruning and enforcing shape constraints. In: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3825–3829 (2021)
11. Elad, M.: Sparse and redundant representations: from theory to applications in signal and image processing. Springer, Berlin (2010)
12. Franchi, G., Fehri, A., Yao, A.: Deep morphological networks. Pattern Recognit. **102**, 107246 (2020)
13. Gaubert, S., McEneaney, W., Qu, Z.: Curse of dimensionality reduction in max-plus based approximation methods: Theoretical estimates and improved pruning algorithms. In: Proc. IEEE Conf. on Decision and Control and Eur. Control Conf. (2011)
14. Guichard, F., Maragos, P., Morel, J.: Partial differential equations for morphological operators, pp. 369–390. Springer New York, New York, NY (2005)
15. Heijmans, H.: Morphological image operators. Academic Press, Boston (1994)
16. Heijmans, H., Ronse, C.: The algebraic basis of mathematical morphology part I: dilations and erosions. Computer Vision Graphics Image Process **50**, 245–295 (1990)
17. Hook, J.: Max-plus linear inverse problems: 2-norm regression and system identification of max-plus linear dynamical systems with gaussian noise (2019)
18. Krause, A., Golovin, D.: Submodular function maximization. In: Tractability (2014)
19. Lovász, L.: Submodular functions and convexity mathematical programming the state of the art. Springer, Berlin (1983)
20. Maclagan, D., Sturmfels, B.: Introduction to tropical geometry. Amer. Math, Soc (2015)
21. Malmberg, F., Ciesielski, K.: Two polynomial time graph labeling algorithms optimizing max-norm-based objective functions. Journal of Mathematical Imaging and Vision **62** (2020)
22. Maragos, P.: Morphological filtering for image enhancement and feature detection. In: A. Bovik (ed.) Image and Video Processing Handbook, 2 edn., pp. 135–156. Elsevier Acad. Press (2005)
23. Maragos, P.: Representations for morphological image operators and analogies with linear operators, pp. 45–187. Elsevier, Hoboken (2013)
24. Maragos, P.: Dynamical systems on weighted lattices: General theory. Math. Control Signals Syst. **29**(21) (2017)
25. Maragos, P., Charisopoulos, V., Theodosis, E.: Tropical geometry and machine learning. Proc IEEE **109**(5), 728–755 (2021)
26. Maragos, P., Theodosis, E.: Multivariate tropical regression and piecewise-linear surface fitting. In: Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP), pp. 3822–3826 (2020)
27. Matheron, G.: Random sets and integral geometry. Wiley, New York (1975)
28. Ritter, G.X., Urcid, G.: Lattice algebra approach to single-neuron computation. IEEE Trans. Neural Netw. **14**(2), 282–295 (2003)
29. Ronse, C., Heijmans, H.: The algebraic basis of mathematical morphology. Part II: openings and closings. Comput. Vision Graphics Image Process.: Image Underst. **54**, 74–97 (1991)
30. Serra, J.: Image analysis and mathematical morphology. Acad Press, USA (1982)
31. Serra, J.: Image analysis and mathematical morphology theoretical advances. Acad Press, USA (1988)

32. Smyrnis, G., Maragos, P.: Multiclass neural network minimization via tropical newton polytope approximation. In: Proc. 37th Int'l Conf. on Machine Learning (ICML-2020), July 2020, vol. 119, pp. 9068–9077. PMLR (2020)
33. Sussner, P., Esmi, E.L.: Morphological perceptrons with competitive learning: lattice-theoretical framework and constructive learning algorithm. Inform. Sci. **181**, 1929–1950 (2011)
34. Tsiamis, A., Maragos, P.: Sparsity in max-plus algebra and systems. Discrete Events Dyn. Syst. **29**, 163–189 (2019)
35. Tsilivis, N., Tsiamis, A., Maragos, P.: Sparse approximate solutions in max-plus equations. In: Lindblad, Joakim, Malmberg, Filip, Sladoje, Nataša (eds.) Discrete geometry and mathematical morphology, pp. 538–550. Springer International Publishing, Cham (2021)
36. Wolsey, L.: An analysis of the greedy algorithm for the submodular set covering problem. Combinatorica **2**, 385–393 (1982)
37. Yang, P.F., Maragos, P.: Min-max classifiers: learnability, design and application. Pattern Recognit. **28**(6), 879–899 (1995)
38. Zhang, L., Naitzat, G., Lim, L.H.: Tropical geometry of deep neural networks. In: Proc. Int'l Conf. on Machine Learning, vol. 80, pp. 5824–5832. PMLR (2018)
39. Zhang, Y., Blusseau, S., Velasco-Forero, S., Bloch, I., Angulo, J.: Max-plus operators applied to filter selection and model pruning in neural networks. In: B. Burgeth, et al. (eds.) Proc. Int'l Symp. Mathematical Morphology (ISMM), *LNCS*, vol. 11564, pp. 310–322. Springer Nature (2019)

Nikolaos Tsilivis received the M.Eng. degree in electrical and computer engineering from the National Technical University of Athens, Greece in 2021, conducting his master thesis, titled "Sparse Representations in Tropical Mathematics", under the supervision of Professor Petros Maragos. He is currently a Ph.D. student at the Center for Data Science, New York University, where he is advised by Prof. Julia Kempe.



Anastasios Tsiamis received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 2014. Currently, he is a Ph.D. student in the Department of Electrical and Systems Engineering at the University of Pennsylvania, advised by George J. Pappas. His research interests include statistical learning for control, risk-aware control and optimization, and networked control systems. Anastasios Tsiamis was a finalist for the International Federation of Automatic Control (IFAC) Young Author Prize in IFAC 2017 World Congress and a finalist for the Best Student Paper Award in American Control Conference (ACC) 2019.

Petros Maragos received the M.Eng. Diploma in E.E. from the National Technical University of Athens (NTUA) in 1980 and the M.Sc. and Ph.D. degrees from Georgia Tech, Atlanta, in 1982 and 1985. In 1985, he joined the faculty of the Division of Applied Sciences at Harvard University, where he worked for eight years as professor of electrical engineering, affiliated with the Harvard Robotics Lab. In 1993, he joined the faculty of the School of ECE at Georgia Tech, affiliated with its Center for Signal and Image Processing. During periods of 1996-98 he had a joint appointment as director of research at the Institute of Language and Speech Processing in Athens. Since 1999, he has been working as professor at the NTUA School of ECE, where he is currently the director of the Intelligent Robotics and Automation Lab. He has held visiting positions at MIT in 2012 and at UPenn in 2016. His research and teaching interests include signal processing, computer vision and speech, machine learning, and robotics. In the above areas he has published numerous papers, book chapters, and has also co-edited three Springer research books, one on multimodal processing and two on shape analysis. He has served as: Associate Editor for the IEEE Transactions on ASSP and the Transactions on PAMI, and editorial board member for several journals on signal processing, image analysis and vision; Co-organizer of several conferences, including recently EUSIPCO-2017 and ICASSP-2023 as general chair; Member of three IEEE SPS technical committees and recently of the IEEE SPS Education Board. He has also served as member of the Greek National Council for Research and Technology. He is the recipient or co-recipient of several awards for his academic work, including: a 1987-1992 US NSF Presidential Young Investigator Award; the 1988 IEEE ASSP Young Author Best Paper Award; the 1994 IEEE SPS Senior Best Paper Award; the 1995 IEEE W.R.G. Baker Prize for the most outstanding original paper; the 1996 Pattern Recognition Society's Honorable Mention best paper award; the CVPR-2011 Workshop on Gesture Recognition best paper award; the 2007 EURASIP Technical Achievement Award. For his research contributions he was elected Fellow of IEEE in 1995 and Fellow of EURASIP in 2010. He was elected IEEE SPS Distinguished Lecturer for 2017-2018.