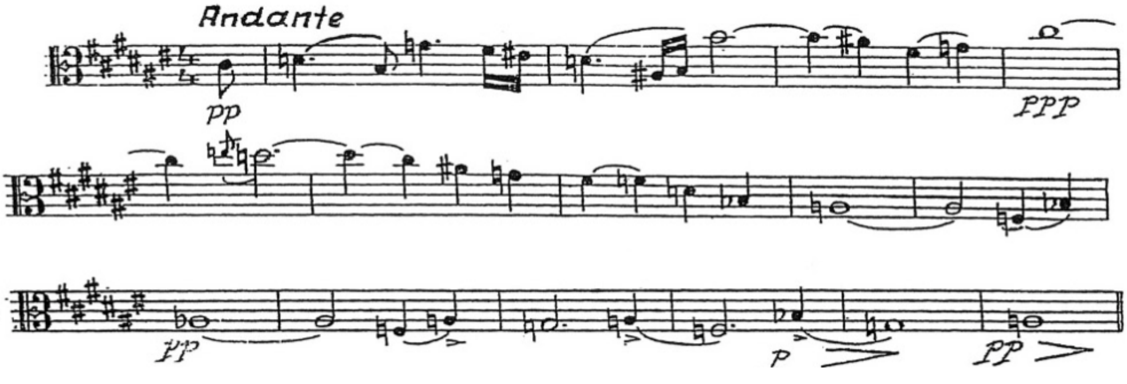
Aleksandar Cvetković
Gran Sasso Science Institute (GSSI)
L'Aquila, Italy

OPTIMISING THE PERRON EIGENVALUE WITH APPLICATIONS TO DYNAMICAL SYSTEMS AND GRAPHS

PhD Thesis
March 14, 2019

Thesis advisor: Vladimir Yu. Protasov
University of L'Aquila, Italy
Faculty of Computer Science of National Research University
Higher School of Economics, Moscow, Russia



G. Mahler - *Symphony no. 10, Adagio*

Contents

Chapter 1

Introduction

Metzler matrices are an important tool for applied mathematics. They play a crucial role in the modelling of the continuous positive linear systems, linear switching and dynamical systems [3, 4, 5]. Their array of applications spans across wide variety of fields: biology, medicine, electrodynamics, economics, social studies and many more [8, 9, 10]. *Non-negative matrices* represent the most well-known and well-studied subset of Metzler matrices [1]. Their spectrum of applications is even greater. The most famous examples include applications to the linear difference equations, Markov chains, discrete dynamical systems and graph theory [12, 13, 14, 15, 16, 17].

Non-negative and, more general, Metzler matrices are in the focus of the presented work. To put it less vaguely, we are concerned with optimizing their leading eigenvalues, their stability, and finding their closest (un)stable counterparts. Before starting out a formal exposition, we give a brief thesis overview. Along with this, we present the problems and the related work which are the foundation of the thesis.

1.1 Thesis overview

Regarding the non-negative matrices and their application, a considerable body of research is devoted to the following two problems:

`N1:` Optimizing the spectral radius on a given family of non-negative matrices;
`N2:` Finding the closest Schur weakly (un)stable non-negative matrix in a given norm (*Schur (de)stabilization*).

In general, both problems are notoriously hard. Spectral radius is neither convex nor concave function, nor is Lipschitz. Because of this, many points of local extrema might occur when searching for the closest (un)stable matrix. To make things worse, they can be quite difficult to identify [18]. However, for matrix families with special structure, i.e. *product (or uncertainty) structure*, it is possible to find an efficient algorithm for optimizing the spectral radius [20, 21]. These families have an application in varous fields such as graph theory, asynchronous systems, biology, economy, etc [15, 22, 23, 24]. In [25] and [27] two algorithms were introduced for optimizing the spectral radius on non-negative product families: *spectral simplex method* and *spectral greedy method*. Independently, a *policy iteration algorithm* was devised in [28], which works in the same fashion as the greedy method. The aforementioned algorithms work faultlessly on positive product families. In Chapter 2 we implement and compare simplex and greedy method by conducting numerical experiments, demonstrating that greedy method is strikingly more efficient.

Extending to the non-negative product families, both greedy and simplex methods may fail due to cycling, or may give a wrong answer. This is especially apparent when dealing with very sparse matrices. Both methods can be modified to accommodate for the (sparse) non-negative families, as it was done in [25, 27]. Unfortunately, these modifications come with several setbacks. Primarily, they considerably complicate fairly straightforward procedures. In addition, the modified greedy method can only be used for minimization of the spectral radius. To overcome this, we introduce *selective greedy method* in Chapter 3, which is one of the thesis central parts. This novel method is as efficient as the spectral greedy method for positive product families. We theoretically confirm its efficiency by proving its local quadratic convergence.

As already mentioned, the problem of Schur (de)stabilization is, in general, hard [18, 19]. However, in L_∞, L_1 and max- norms either an explicit solutions or an efficient algorithms can be provided, as shown in [27]. There, an algorithm based on the spectral greedy method was constructed to solve the Schur stabilization problem. In Chapter 4 we present a modification of this algorithm which, combined with the selective greedy method, significantly improves the computational time.

Further in Chapter 4 we demonstrate an application of Schur stabilization to graph theory. We present an algorithm for finding the closest acyclic directed subgraph. We take things a bit further and use this algorithm to tackle the problem of finding a *maximum acyclic subgraph* (MAS). This is a famous NP-hard problem that has inspired a large body of literature [29, 30]. Equivalent to this problem is the *minimum feedback arc set* problem, which has found its way in numerous applications [32, 33]. Our approach to approximating MAS is time efficient, and we believe it is as competitive as the latest algorithms when it comes to edge-preserving of the starting graph.

Moving on from non-negative matrices, we can set the analogues to the `N1` and `N2` problems for Metzler matrices. They are formulated as:

`M1`: Optimizing the spectral abscissa on a given family of Metzler matrices;
`M2`: Finding the closest Hurwitz weakly (un)stable Metzler matrix in a given norm (*Hurwitz (de)stabilization*).

The story is the same as with the non-negative case: both of these problems are hard in general [34]. However, when working with Metzler product families (`M1`) and L_∞, L_1 and max- norms (`M2`), they can be effectively solved. This is another central part of the thesis, and it is discussed throughout Chapters 5 and 6. In Chapter 5 we modify the selective greedy method to handle the optimization of spectral abscissa on Metzler families, while in Chapter 6 we deal with the Hurwitz (de)stabilization.

Some applications of the Hurwitz (de)stabilization are discussed in Chapter 7. We use Hurwitz stabilization to look for the closest stable Metzler *sign-matrix*. Metzler sign-matrices and sign-stability are interesting concepts with many applications [35, 38, 40]. We then demonstrate the application of sign-stabilization for stabilization of the linear switching systems.

All the algorithms presented in the thesis are supplemented with the corresponding numerical experiments. The experiments are performed on a standard laptop and coded in Python using NumPy and ScyPy packages. The codes can be found and downloaded on:
https://github.com/ringechpil/thesis_codes.

We should remark the change of the objective function in the optimization problems `N1` and `M1`. This change is due to the Perron-Frobenius theory. For non-negative matrices spectral radius and spectral abscissa coincide, and spectral radius acts as a *Perron (leading) eigenvalue.* For Metzler matrices with negative entries this is not true. In this case the spectral abscissa takes over the role of the Perron eigenvalue. Since the presented procedures actually optimize the leading eigenvalue and just exploit the fact that the spectral radius (abscissa) coincide with it, we make the aforementioned change.

Most of the thesis results were first presented in [41] and [42]. The main topic of [41] is introducing the selective greedy method for non-negative sparse families. [42] is primarily concerned with extending the applicability of the selective greedy method to Metzler families, and its use on dynamical system.

Throughout the thesis we will assume, unless stated otherwise, that all our matrices are $d \times d$ square, and all the vectors are from $\mathbb{R}^d$. If A is some matrix, $\boldsymbol{a}_i$ will denote its row, and a_{ij} its entry; if $\boldsymbol{v}$ is some vector, v_i denotes its entry.

Now we define some concepts fundamental to the thesis development.

Definition 1.1 *A matrix is said to be* Metzler *if all its off-diagonal elements are non-negative. We denote the set of all Metzler matrices by $\mathcal{M}$.*

Definition 1.2 *A Metzler matrix is* non-negative *if all of its elements are non-negative. A non-negative matrix is* strictly positive *if it contains no zero entry. If A is a non-negative matrix we write $A \geqslant 0$, and if A is strictly positive we write $A > 0$.*

For two matrices A and B we write $A \geqslant B$ if $A - B \geqslant 0$, and $A \leqslant B$ if $B - A \geqslant 0$. Analogously, we define strict relations $A > B$ and $A < B$. We define those relations for the vectors in the same manner.

Definition 1.3 Spectral radius *of a matrix A, denoted by $\rho(A)$, is the largest modulus of its eigenvalues, that is:*

$$\rho(A) = \max\{\ |\lambda| \mid \lambda \text{ is an eigenvalue of } A\}.$$

Definition 1.4 Spectral abscissa *of a matrix A, denoted by $\eta(A)$, is the largest real part of its eigenvalues, that is:*

$$\eta(A) = \max\{ \text{ Re}(\lambda) \mid \lambda \text{ is an eigenvalue of } A\}.$$

Definition 1.5 *A family $\mathcal{A}$ of matrices is called a* product family[1] *if there exist compact sets $\mathcal{F}_i \subset \mathbb{R}^d$, $i = 1, \dots, d$, such that $\mathcal{A}$ consists of all possible matrices with i-th row from $\mathcal{F}_i$, for all $i = 1, \dots, d$. The sets $\mathcal{F}_i$ are called* uncertainty sets.

The matrices belonging to the product families are constructed by independently taking ith row from the uncertainty set $\mathcal{F}_i$. Moreover, product families can be topologically seen as product sets of their uncertainty sets: $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_d$. For the rest of the text we will always assume that our uncertainty sets are either finite or have a polyhedral structure, meaning that they can be represented as a convex combinations of their vertices.

Example 1.6 Family of directed graphs where each vertex has prescribed number of edges can be viewed as a product family. To each vertex i having r_i edges we can associate an uncertainty set $\mathcal{F}_i$, containing all possible configurations of r_i outgoing edges from the vertex i.

If we observe the graph's adjacency matrix, then its rows represent the vertices, and corresponding uncertainty sets $\mathcal{F}_i$ sets of all (0,1)-vectors with ones in r_i places. Optimizing the spectral radius on such product families of (0,1)-matrices $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_d$ is a problem found in literature [22, 45], which we will address later. ○

Example 1.7 Let A be some non-negative matrix. An L_∞ ball of non-negative matrices of radius τ around A

$$\mathcal{B}_\tau^+(A) \;=\; \Big\{ X \geqslant 0 \;\Big|\; \|X - A\|_\infty \leq \tau \Big\}$$

is a product family with the polyhedral uncertainty sets

$$\mathcal{B}_{i,\tau}^+(A) = \Big\{ \boldsymbol{x} \in \mathbb{R}_+^d \;\Big|\; \sum_{j=1}^{d} |x_j - a_{ij}| \leqslant \tau \Big\}.$$

Similarly, an L_∞ ball of Metzler matrices $\mathcal{B}_\tau(A)$ around a Metzler matrix A is also a product family. Minimizing the spectral radius (abscissa) on the $\mathcal{B}_\tau^+$ (i.e. $\mathcal{B}_\tau$) will be a crucial point for finding the closest non-negative (Metzler) matrix, which we will see later on. ○

[1] Since in this work we primarily deal with product families, we will usually refer to *product families (of matrices)* simply as *families (of matrices)*, meaning the same.

Chapter 2

Positive product families

In this chapter we deal with the optimization of spectral radius on positive product families. We present two algorithms *spectral simplex method* and *spectral greedy method*. In essence, these methods are iterative procedures which bring the spectral radius closer to the optimal value with each iteration.

Definition 2.1 *A product familiy is* positive product family *if it contains only strictly positive matrices*

The presented methods rely heavily on the spectral properties of positive matrices, which are characterised by the following theorems:

Theorem 2.2 ([14]) *Let A be a positive matrix. Then, there exist an eigenvalue r such that:*
1° $r > 0$;
2° $r > |\lambda|$*, for every eigenvalue* $\lambda \neq r$;
3° r *is a simple eigenvalue;*
4° r *can be associated with a strictly positive (right) eigenvector* $\boldsymbol{v}$*;*
5° $\boldsymbol{v}$ *is a simple eigenvector, i.e. it is unique up to a multiplicative constant.* □

Definition 2.3 *Let A be a positive matrix. An eigenvalue r characterized by Theorem* 2.2 *is called* Perron (leading) eigenvalue*. Its corresponding eigenvector $\boldsymbol{v}$ is called* Perron (leading) eigenvector*.*

Looking at Definition 1.3 for spectral radius, we easily conclude:

Proposition 2.4 *For a positive matrix, its spectral radius is its Perron eigenvalue.* □

The next definition introduces a concept essential for determining if a given matrix is optimal.

Definition 2.5 *Let $\mathcal{F}$ be a positive product family. A matrix $A \in \mathcal{F}$ is said to be* maximal in each row with the respect to its leading eigenvector $\boldsymbol{v}$ *if* $(\boldsymbol{a}_i, \boldsymbol{v}) = \max_{\boldsymbol{b}_i \in \mathcal{F}_i}(\boldsymbol{b}_i, \boldsymbol{v})$, $i = 1, \ldots, d$*. It is* minimal in each row w.t.r. to its leading eigenvector $\boldsymbol{v}$ *if* $(\boldsymbol{a}_i, \boldsymbol{v}) = \min_{\boldsymbol{b}_i \in \mathcal{F}_i}(\boldsymbol{b}_i, \boldsymbol{v})$ *for all* $i = 1, \ldots, d$*.*

Proposition 2.6 [25] *Let $\mathcal{F}$ be a positive product family. If a matrix $A \in \mathcal{F}$ is maximal (minimal) in each row w.t.r. to its leading eigenvector, then it has the maximal (minimal) spectral radius among all matrices from $\mathcal{F}$.* □

2.1 Simplex and greedy method: an overview

Spectral simplex method originated in [21]. An algorithm for its implementation on positive and non-negative product families was constructed in [25]. We describe its version for positive families below:

`Alg. SP: Spectral simplex method for maximizing spectral radius over positive product families`

`Initialization.` Let $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_d$ be a positive product family. Taking arbitrary $\boldsymbol{a}_i \in \mathcal{F}_i\,,\ i = 1, \ldots, d$, form the matrix $A_1 \in \mathcal{F}$ with rows $\boldsymbol{a}_1^{(1)}, \ldots, \boldsymbol{a}_d^{(1)}$. Take its leading eigenvector $\boldsymbol{v}_1$.

`Main loop; the` k`th iteration.` We have a matrix $A_k \in \mathcal{F}$ composed of rows $\boldsymbol{a}_i^{(k)} \in \mathcal{F}_i, i = 1, \ldots, d$. Compute its leading eigenvector $\boldsymbol{v}_k$. Starting from $i = 1$ do:

Find a solution $\boldsymbol{a}_i^* \in \mathcal{F}_i$ to the problem

$$\left\{ \begin{array}{l} (\boldsymbol{a}_i, \boldsymbol{v}_k) \ \to \ \max \\ \boldsymbol{a}_i \in \mathcal{F}_i \end{array} \right.$$

If $(\boldsymbol{a}_i^*, \boldsymbol{v}_k) = (\boldsymbol{a}_i^{(k)}, \boldsymbol{v}_k)$, set $\boldsymbol{a}_i^{(k+1)} = \boldsymbol{a}_i^{(k)}$. Then, if $i < d$, set $i = i + 1$. Else, if $i = d$, stop the algorithm; the matrix A_k is maximal in each row, and therefore the optimal solution with $\rho(A_k) = \rho_{\max}$.

If $(\boldsymbol{a}_i^*, \boldsymbol{v}_k) > (\boldsymbol{a}_i^{(k)}, \boldsymbol{v}_k)$, set $\boldsymbol{a}_i^{(k+1)} = \boldsymbol{a}_i^*$ and $\boldsymbol{a}_j^{(k+1)} = \boldsymbol{a}_j^{(k)}$ for $j = i + 1, \ldots, d$. Then move onto the $(k + 1)$`st iteration`. ◇

The simplex method makes row-by-row checks to determine whether the current matrix is the optimal one. If the check fails in some row, the algorithm goes back to the row one. The greedy method, on the other hand, checks all the rows at once, and then changes the non-optimal ones. It was first introduced in [27], and we formally present it below:

`Alg. GP: Spectral greedy method for maximizing spectral radius over positive product families`

`Initialization.` Let $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_d$ be a positive product family. Taking arbitrary $\boldsymbol{a}_i \in \mathcal{F}_i\,,\ i = 1, \ldots, d$, form a matrix $A_1 \in \mathcal{F}$ with rows $\boldsymbol{a}_1^{(1)}, \ldots, \boldsymbol{a}_d^{(1)}$. Take its leading eigenvector $\boldsymbol{v}_1$.

`Main loop; the` k`th iteration.` We have a matrix $A_k \in \mathcal{F}$ composed of rows $\boldsymbol{a}_i^{(k)} \in \mathcal{F}_i, i = 1, \ldots, d$. Compute its leading eigenvector $\boldsymbol{v}_k$ and for $i = 1, \ldots d$, find a solution $\boldsymbol{a}_i^* \in \mathcal{F}_i$ of the problem

$$\left\{ \begin{array}{l} (\boldsymbol{a}_i, \boldsymbol{v}_k) \ \to \ \max \\ \boldsymbol{a}_i \in \mathcal{F}_i \end{array} \right.$$

For each $i = 1, \dots, d$, do:

If $(\boldsymbol{a}_i^*, \boldsymbol{v}_k) = (\boldsymbol{a}_i^{(k)}, \boldsymbol{v}_k)$, set $\boldsymbol{a}_i^{(k+1)} = \boldsymbol{a}_i^{(k)}$.
Otherwise, if $(\boldsymbol{a}_i^*, \boldsymbol{v}_k) > (\boldsymbol{a}_i^{(k)}, \boldsymbol{v}_k)$, set $\boldsymbol{a}_i^{(k+1)} = \boldsymbol{a}_i^*$.

Form the corresponding matrix A_{k+1}. If the first case took place for all $i = 1, \dots, d$, i.e. if $A_{k+1} = A_k$, finish the procedure; the matrix A_k is maximal in each row, and therefore the optimal solution with $\rho(A_k) = \rho_{\max}$. Otherwise, go to $(k+1)$`st iteration`. ◇

Remark. Both simplex and greedy method for minimizing the spectral radius are exactly the same as above, except that we change the row if $(\boldsymbol{a}_i^*, \boldsymbol{v}_k) < (\boldsymbol{a}_i^{(k)}, \boldsymbol{v}_k)$.

Let $\rho_k = \rho(A_k)$, where A_k is the matrix obtained in the kth iteration of Algorithm SP (GP). The described algorithms are strict relaxation schemes, in other words:

Proposition 2.7 [25] *For both the simplex and greedy method algorithms we have $\rho_{k+1} > \rho_k$ for every k. Analogously, for the minimization we have $\rho_{k+1} < \rho_k$ for every k.* □

The maximal scalar product $(\boldsymbol{a}_i, \boldsymbol{v}_k)$ over all $\boldsymbol{a} \in \mathcal{F}_i$ is found by solving the corresponding convex problem over $\mathrm{co}(\mathcal{F}_i)$. If the uncertainty set $\mathcal{F}_i$ is finite, this is done merely by exhaustion of all the elements of $\mathcal{F}_i$. If $\mathcal{F}_i$ is a polyhedron, then we find $\boldsymbol{a}_i^{(k+1)}$ among its vertices solving an LP problem by the (usual) simplex method. Each matrix A_k is therefore a vertex of the product set $\mathcal{F}$. From the fact that we have finite number of vertices, and that the spectral radius strictly increases with each iteration, we have

Theorem 2.8 [25, 27] *Algorithms SP and GP find an optimal solution in the finite number of iterations. The same is true for the minimization.* □

2.2 Simplex vs. greedy method: a comparison

In this section we compare how simplex and greedy methods perform on positive product families, as the dimension d and cardinality of uncertainty sets N are varied. All the product sets are randomly generated. The results are presented in the tables below.

Simplex method

$N \setminus d$	25	100	250
50	43.5	185.8	463
100	45.3	195.6	502

Greedy method

$N \setminus d$	25	100	250
50	3	3	3
100	3	3.2	3

Table 2.1.1: Simplex vs. Greedy, maximization, avg. number of iterations

Simplex method

$N \setminus d$	25	100	250
50	0.15s	3.61s	35.1s
100	0.27s	6.06s	52.11s

Greedy method

$N \setminus d$	25	100	250
50	0.02s	0.11s	0.36s
100	0.04s	0.21s	0.58s

Table 2.1.2: Simplex vs. Greedy, maximization, avg. running time

Simplex method

$N \setminus d$	25	100	250
50	50	190	243.3
100	57.7	222.2	518.3

Greedy method

$N \setminus d$	25	100	250
50	3.2	3.1	3.2
100	3.4	3.2	3.1

Table 2.2.1: Simplex vs. Greedy, minimization, avg. number of iterations

Simplex method

$N \setminus d$	25	100	250
50	0.16s	4.17s	18.42s
100	0.34s	6.97s	56.18s

Greedy method

$N \setminus d$	25	100	250
50	0.02s	0.13s	0.55s
100	0.05s	0.21s	0.61s

Table 2.2.2: Simplex vs. Greedy, minimization, avg. running time

It is not hard to see that the greedy method considerably outperforms the simplex method. For $d = 250$ and $N = 10$ the simplex method takes just a less than a minute to finish, while greedy method does it in less than a second. Nevertheless, a credit must be given to the spectral simplex method for being the original optimization method of this kind, and from which the greedy method was derived. Moreover, we will later make a comeback to the simplex method to prove its global linear convergence.

To further emphasize the efficiency of the greedy method, we performed additional experiments for the dimension of up to 2000, and for bigger uncertainty sets.

avg. number of iterations

$N \setminus d$	500	1000	2000
50	3.1	3	3.1
100	3.2	3.1	3
250	3.1	3.2	3.1

avg. running time

$N \setminus d$	500	1000	2000
50	1.07s	4.63s	20.69s
100	1.55s	6.36s	25.44s
250	2.8s	8.64s	390.27s

Table 2.3: Greedy method, maximization

avg. number of iterations

$N \setminus d$	500	1000	2000
50	3.2	3.1	3
100	3.1	3	3
250	3	3.1	3.1

avg. running time

$N \setminus d$	500	1000	2000
50	1.06s	4.57s	19.39s
100	1.64s	4.89s	24.6s
250	2.52s	7.29s	447.09s

Table 2.4: Greedy method, minimization

In all the cases, except for the last one, we have an amazing performance time: less then a half of a minute. As for the big blow-up in the $d = 2000$ and $N = 250$ case, we believe it is due to the combinatorics of the problem, since we have significantly more vectors to check for maximality (minimality) compared to the $N = 100$ size sets. Aside from this 'very big case', the most computationally demanding task is computing the leading eigenvector $\boldsymbol{v}_k$. The running times are therefore in big part determined by the numerical procedure for obtaining $\boldsymbol{v}_k$. We used the built-in method of the NumPy Python package. Using a more efficient method may further improve the running times. The choice of the method for computing the leading eigenvector plays even a greater role. As we will see in the following chapter, this choice can even affect the convergence of the greedy procedure and broaden its scope of application.

Chapter 3

Non-negative product families

Impressive performance of the greedy method on positive families gets overshadowed by the fact that strictly positive matrices occur seldom in applications. In real applications one needs to deal with sparse non-negative matrices. Unfortunately, the greedy method may fail in this case.

Definition 3.1 *A product family is* non-negative product family *if it contains only non-negative matrices.*

We introduce the concept of irreducibility, which will occasionally show up in the further text.

Definition 3.2 *A non-negative matrix is* irreducible *if it does not have a non-trivial invariant coordinate subspace, i.e. a subspase spanned by some elements $\boldsymbol{e}_i$ of the canonical basis.*

Definition 3.3 *A matrix family $\mathcal{A}$ is irreducible if its matrices do not share a common invariant subspace.*

From Definition 3.2, we see that having at least one irreducible matrix $A \in \mathcal{A}$ implies that the whole family $\mathcal{A}$ is irreducible. Positive families are the special case of irreducible non-negative families.

3.1 Greedy method for non-negative families

The Perron eigenvalue of a non-negative matrix loses some properties it has for the positive matrices. This is why the greedy method may fail when operating on non-negative product families.

Theorem 3.4 ([14]) *Let A be a non-negative matrix. Then, there exist an eigenvalue r such that:*
1° $r \geqslant 0$;
2° $r \geqslant |\lambda|$*, for every eigenvalue* λ*;*
3° r *can be associated with a non-negative (right) eigenvector* $\boldsymbol{v}$. □

Definition 3.5 *Let A be a non-negative matrix. An eigenvalue r characterized by Theorem* 3.4 *is called* Perron (leading) eigenvalue. *Its corresponding eigenvector $\boldsymbol{v}$ is called* Perron (leading) eigenvector.

Proposition 3.6 *For a non-negative matrix, its spectral radius is its Perron eigenvalue.* □

Comparing Theorems 2.2 and 3.4 we see that the Perron eigenvalue of a non-negative matrix is neither simple nor unique, and may contain zero entries. Moreover, there might exist several leading eigenvectors. All this can cause trouble when implementing the greedy method on non-negative families. In this case we may have cycling, or get the wrong answer.

Definition 3.7 *A subspace spanned by all the leading eigenvectors of a non-negative matrix is called* Perron subspace*.*

With a slight (but essential) modification, we extend the notions of maximality/minimality in each row, given in Definition 2.5.

Definition 3.8 *Let $\mathcal{F}$ be a non-negative product family. A matrix $A \in \mathcal{F}$ is said to be* maximal in each row w.t.r. to its leading eigenvector $\boldsymbol{v}$ *if* $\boldsymbol{v} > 0$ *and* $(\boldsymbol{a}_i, \boldsymbol{v}) = \max_{\boldsymbol{b}_i \in \mathcal{F}_i} (\boldsymbol{b}_i, \boldsymbol{v})$, $i = 1, \ldots, d$. *It is* minimal in each row w.t.r. to its leading eigenvector $\boldsymbol{v}$ *if* $(\boldsymbol{a}_i, \boldsymbol{v}) = \min_{\boldsymbol{b}_i \in \mathcal{F}_i} (\boldsymbol{b}_i, \boldsymbol{v})$ *for all* $i = 1, \ldots, d$.

Note that, in the non-negative case, these notions are not completely analogous: the minimality in each row is defined with the respect to an arbitrary leading eigenvector $\boldsymbol{v} \geqslant 0$, while the maximality needs a strictly positive $\boldsymbol{v}$. Having this in mind, Proposition 2.6 can be extended to non-negative families [25].

We now present the modified version of Algorithm GP for use on non-negative product families, and discuss under which conditions it is safe to use it.

`Alg. GNN: Spectral greedy method for maximizing spectral radius over non-negative families`

`Initialization.` Let $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_d$ be a non-negative product family. Taking arbitrary $\boldsymbol{a}_i \in \mathcal{F}_i$, $i = 1, \ldots, d$, form a matrix $A_1 \in \mathcal{F}$ with rows $\boldsymbol{a}_1^{(1)}, \ldots, \boldsymbol{a}_d^{(1)}$. Take its leading eigenvector $\boldsymbol{v}_1$ (or take any of them, if it is not unique).

`Main loop; the` k`th iteration.` We have a matrix $A_k \in \mathcal{F}$ composed of rows $\boldsymbol{a}_i^{(k)} \in \mathcal{F}_i, i = 1, \ldots, d$. Compute and choose one of its leading eigenvectors $\boldsymbol{v}_k$, and for $i = 1, \ldots d$, find a solution $\boldsymbol{a}_i^* \in \mathcal{F}_i$ of the problem

$$\begin{cases} (\boldsymbol{a}_i, \boldsymbol{v}_k) & \to \ \max \\ \boldsymbol{a}_i \in \mathcal{F}_i \end{cases}$$

For each $i = 1, \ldots, d$, do:

If $(\boldsymbol{a}_i^*, \boldsymbol{v}_k) = (\boldsymbol{a}_i^{(k)}, \boldsymbol{v}_k)$, set $\boldsymbol{a}_i^{(k+1)} = \boldsymbol{a}_i^{(k)}$.
Otherwise, if $(\boldsymbol{a}_i^*, \boldsymbol{v}_k) > (\boldsymbol{a}_i^{(k)}, \boldsymbol{v}_k)$, set $\boldsymbol{a}_i^{(k+1)} = \boldsymbol{a}_i^*$.

Form the corresponding matrix A_{k+1}. If the first case took place for all $i = 1, \ldots, d$, i.e. if $A_{k+1} = A_k$, go to `Termination`. Otherwise, go to $(k+1)$`st iteration`.

`Termination.` If $\boldsymbol{v}_k > 0$, finish the procedure; the matrix A_k is maximal in each row, and therefore the optimal solution with $\rho(A_k) = \rho_{\max}$. If $\boldsymbol{v}_k$ has some zero components, then the family $\mathcal{F}$ is reducible, and we need to exit the algorithm and factorize $\mathcal{F}$ (see Section 3.6 below). Otherwise, we can get the estimate for $\rho_{\max}$ from Proposition 3.9 given bellow. ◇

Remark. The procedure for minimizing the spectral radius is exactly the same, except that we change the row if $(\boldsymbol{a}_i^*, \boldsymbol{v}_k) < (\boldsymbol{a}_i^{(k)}, \boldsymbol{v}_k)$, and we omit the requirement that $\boldsymbol{v}_k > 0$.

Greedy method for non-negative families is also a relaxation scheme but, unlike for the positive families, we do not have a strict increase in spectral radius with every iteration. Because of this the algorithm may get stuck, making the optimal value unreachable. If this happens the algorithm can be interrupted at an arbitrary step, after which we apply the a posteriori estimates for $\rho_{\max}$ defined below.

Denote $\boldsymbol{v}_k = (v_{k,1}, \ldots, v_{k,d})$, so $(\boldsymbol{a}_i^{(k)}, \boldsymbol{v}_k) = \rho_k v_{k,i}$. Then for for an arbitrary matrix A and its leading eigenvector $\boldsymbol{v}_k$, we define the following values.

$$s_i(A) = \begin{cases} \max\limits_{\boldsymbol{b}\in\mathcal{F}_i} \frac{(\boldsymbol{b},\boldsymbol{v}_k)}{v_{k,i}} & ; \; v_{k,i} > 0; \\ +\infty & ; \; v_{k,i} = 0; \end{cases} \qquad s(A) = \max_{i=1,\ldots,d} s_i(A) \tag{3.1}$$

Thus, $s_i(A)$ is the maximal ratio between the value $(\boldsymbol{a}, \boldsymbol{v}_k)$ over all $\boldsymbol{a} \in \mathcal{F}_i$ and the ith component of $\boldsymbol{v}_k$. Similarly, for the minimum:

$$t_i(A) = \begin{cases} \min\limits_{\boldsymbol{b}\in\mathcal{F}_i} \frac{(\boldsymbol{b},\boldsymbol{v}_k)}{v_{k,i}} & ; \; v_{k,i} > 0; \\ +\infty & ; \; v_{k,i} = 0; \end{cases} \qquad t(A) = \min_{i=1,\ldots,d} t_i(A) \tag{3.2}$$

Then the following obvious estimates for $\rho_{\max}$ (and $\rho_{\min}$) are true:

Proposition 3.9 [25] *For the greedy method implemented on non-negative product families (for maximization and for minimization respectively), we have in kth iteration:*

$$\rho_k \leqslant \rho_{\max} \leqslant s(A_k)\ ; \qquad t(A_k) \leqslant \rho_{\min} \leqslant \rho_k\,. \tag{3.3}$$

□

3.2 The (in)applicability of the greedy method

As stated, the greedy method may cycle for general non-negative product families. We illustrate that by the following

Example 3.10 Consider the product family $\mathcal{F}$ of 3×3 matrices defined by the three uncertainty sets:

$$\mathcal{F}_1 = \left\{ \begin{array}{lrr} (1, & 1, & 1) \\ (0, & 5, & 10) \\ (0, & 10, & 5) \\ (12, & 0, & 0) \end{array} \right\}; \qquad \mathcal{F}_2 = \left\{ \begin{array}{lrr} (1, & 1, & 1) \\ (0, & 10, & 0) \end{array} \right\};$$

$$\mathcal{F}_3 = \left\{ \begin{array}{lrr} (1, & 1, & 3) \\ (0, & 0, & 10) \end{array} \right\}.$$

Taking the first row from each set we compose the first matrix A_1 (which is positive, hence the family is irreducible) and the algorithm runs as follows: $A_1 \to A_2 \rightleftarrows A_3$, where

$$\begin{tikzcd}[ampersand replacement=\&] \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 3 \end{pmatrix} \arrow[r, "\boldsymbol{v}_1=(1,1,2)^T"] \& \begin{pmatrix} 0 & 5 & 10 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix} \arrow[r, shift left, "\boldsymbol{v}_2=(2,2,1)^T"] \& \begin{pmatrix} 0 & 10 & 5 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix} \arrow[l, shift left, "\boldsymbol{v}_3=(2,1,2)^T"] \end{tikzcd}$$

The matrix A_1 has a unique eigenvector $\boldsymbol{v}_1 = (1, 1, 2)^T$. Taking the maximal row in each set $\mathcal{F}_i$ with the respect to this eigenvector, we compose the next matrix A_2. It has a multiple leading eigenvalue $\lambda = 10$. Choosing one of its leading eigenvectors $v_2 = (2, 2, 1)^T$, we make the next iteration and obtain the matrix A_3. It also has a multiple leading eigenvalue $\lambda = 10$. Choosing one of its leading eigenvectors $v_2 = (2, 1, 2)^T$, we make the next iteration and come back to the matrix A_2. The algorithm cycles.

We see that the greedy method is stuck on the cycle $A_2 \rightleftarrows A_3$ and on the value of the spectral radius $\rho = 10$. However, the maximal spectral radius $\rho_{\max}$ for this family is not 10, but 12. The value $\rho_{\max}$ is realized for the matrix

$$A_4 = \begin{pmatrix} 12 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 3 \end{pmatrix}$$

which is missed by the algorithm.

The algorithm never terminates and, when stopped, gives a wrong solution: $\rho(A_2) = \rho(A_3) = 10$ instead of $\rho_{\max} = 12$. The a posteriori estimate (3.3) gives $10 \leqslant \rho_{\max} \leqslant 12.5$. The error is 25 %, and this is the best the greedy method can do for this family. ○

The presented example rises a natural question: is the idea of the greedy algorithm in the end applicable for strictly positive matrices only? Can it be modified to work with sparse matrices? In [27] one such modification was given, but not without drawbacks. This modification somewhat complicates the greedy procedure and it can be used for the minimization only. To find another approach, we first need to reveal the difficulties caused by sparsity, and then to see how to treat them.

We saw that if some vectors from the uncertainty sets have zero components, then the greedy method may cycle and, which is still worse, may miss the optimal matrix and give a quite rough final estimate of the optimal value. Indeed, numerical experiments show that cycling often occurs for sparse matrices, especially for the minimization problem (for maximizing it is rare but also possible). Moreover, the sparser the matrix, the more often the cycling occurs.

The reason of cycling is hidden in multiple leading eigenvectors. If at some iteration we get a matrix A_k with a multiple leading eigenvector, then we have an uncertainty with choosing $\boldsymbol{v}_k$ from the Perron subspace. A 'bad choice' may cause cycling. In Example 3.10 both matrices A_2 and A_3 have Perron subspaces of dimension 2, and this is the reason of the trouble. Moreover, in practice, not only multiple but 'almost multiple' leading eigenvectors (when the second largest eigenvalue is very close to the first one) may occur because of the rounding errors and cause cycling, or at least significant slow-down of the algorithm. On the other hand, if in all iterations the leading eigenvectors $\boldsymbol{v}_k$ are simple, then cycling never occurs as the following proposition guarantees.

Proposition 3.11 *If in all iterations of the greedy method the leading eigenvectors $\boldsymbol{v}_k$ are simple, then the method does not cycle and (in case of finite or polyhedral sets $\mathcal{F}_i$) terminates within finite time.* □

By a well-known result of the Perron-Frobenius theory, irreducible matrices have strictly positive leading eigenvalue and a simple leading eigenvector [1]. Therefore, if we manage to obtain an irreducible matrix in each iteration of the greedy method, we will eventually finish with the right answer. This is exactly the case with the positive matrices, since they are irreducible. Unfortunately, for sparse matrices obtaining an irreducible one at every step is quite restrictive, and hardly eve r achievable in applications.

A naive idea to avoid cycling would be to slightly change all the uncertainty sets $\mathcal{F}_i$ making them strictly positive. For example, replacing them by the perturbed sets $\mathcal{F}_{i,\varepsilon} = \mathcal{F}_i + \varepsilon \boldsymbol{e}$, where $\boldsymbol{e} = (1, \ldots, 1) \in \mathbb{R}^d$ and $\varepsilon > 0$ is a small constant. This means adding a positive number ε to each entry of the vectors from the uncertainty sets. All matrices $A \in \mathcal{F}$ become strictly positive: $A \mapsto A_\varepsilon = A + \varepsilon E$, where $E = \boldsymbol{e}\boldsymbol{e}^T$ is the matrix of ones. By Proposition 3.11, the greedy method for the perturbed family $\mathcal{F}_\varepsilon$ does not cycle. However, this perturbation can significantly change the answer to the problem. For example, if a matrix A has $d-1$ ones over the main diagonal and all other elements are zeros ($a_{ij} = 1$ if $i - j = 1$ and $a_{ij} = 0$ otherwise), then $\rho(A) = 0$, while $\rho(A_\varepsilon) > \varepsilon^{1/d}$. Even if ε is very small, say, 10^{-20}, then for $d = 50$ we have $\rho(A_\varepsilon) > 0.4$, while $\rho(A) = 0$. Hence, we cannot merely replace the family $\mathcal{F}$ by $\mathcal{F}_\varepsilon$ without risking a big error.

Nevertheless, for the greedy method, a way to avoid cycling does exist even for sparse matrices. Further in the text we will present a strategy showing its efficiency both theoretically and numerically. Even for the very sparse matrices, it can work as fast as for the positive ones. Before introducing this strategy we will make a short overview about implementation of the simplex method on the non-negative families.

Remark. The authors of [28] came to the problem of spectral radius optimization by studying entropy games with fixed number of states. Independently of [27] they derived *policy iteration algorithm* which works the same way as the spectral greedy method. However, no advances on how to practically handle sparse matrices were made there.

Simplex method for non-negative families

Analogously to Algorithms GP and GNN, Algorithm SP can be modified to accommodate for the non-negative matrices. Unfortunately, all the troubles that follow the greedy method occur also for the simplex procedure. Overcoming those issues is the main topic of [25]. One strategy is to simply stop the procedure after some number of iteration and use the estimates from Proposition 3.9, since they also hold for the simplex method. However, as with the greedy case, we may end up getting quite rough estimates.

Proposition 3.11 is also true for the simplex method. Therefore, we can safely execute the simplex procedure on a non-negative family if we can provide an irreducible matrix at every step. As we already mentioned, this might be too much to ask for. Luckily, for maximization we are in much better situation: we just need to provide the irreducible matrix in the first step.

Theorem 3.12 [25] *Let $\mathcal{F}$ be a non-negative product family. Consider the spectral simplex method in the maximization problem. If the initial matrix $A_1 \in \mathcal{F}$ has a simple leading eigenvector, then so do all successive matrices $A_2, A_3, \ldots$ and the method does not cycle.* □

Theorem 3.12 does not apply for the greedy method. Example 3.10 clearly illustrates this: we start with a strictly positive matrix A_1, but we end up with the cycling. Theorem 3.12 is also inapplicable to the simplex method for minimization, but [25] also provides a modification of the simplex method for handling the sparse matrices when minimising the spectral radius.

With everything said, we can still use the simplex method on sparse matrices when working with small dimensions and small uncertainty sets. For bigger problems computations become too much time demanding. When minimizing spectral radius on a very sparse non-negative families for $d = 500$ and $N = 5$, the number of average iterations amounts to around 750 [25]. In order to efficiently optimize spectral radius on big and sparse families we need to find a way to 'save' the greedy method and make it perform at least to some degree as good as in the positive case.

3.3 Selective greedy method

Let $\mathcal{F}$ be a product family of sparse non-negative matrices, and $\mathcal{F}_\varepsilon$ the corresponding perturbed family. The crucial idea behind the new strategy is to optimise the spectral radius on the original family $\mathcal{F}$ by using the leading eigenvectors from the perturbed family $\mathcal{F}_\varepsilon$.

Definition 3.13 *The selected leading eigenvector of a non-negative matrix A is the limit $\lim\limits_{\varepsilon\to 0} \boldsymbol{v}_\varepsilon$, where $\boldsymbol{v}_\varepsilon$ is the normalized leading eigenvector of the perturbed matrix $A_\varepsilon = A + \varepsilon E$.*

The next result gives a way for finding the selected eigenvector explicitly. Before formulating it, we make some observations.

If an eigenvector $\boldsymbol{v}$ is simple, then it coincides with the selected eigenvector. If $\boldsymbol{v}$ is multiple, then the selected eigenvector is a vector from the Perron subspace. Thus, Definition 3.13 provides a way to select one vector from the Perron subspace of every non-negative matrix. We are going to see that this selection is successful for the greedy method.

Consider the power method for computing the leading eigenvectors: $\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k$, $k \geq 0$. In most of practical cases it converges to a leading eigenvector $\boldsymbol{v}$ linearly with the rate $O(|\frac{\lambda_2}{\lambda_1}|^k)$, where λ_1, λ_2 are the first and the second largest by modulus eigenvalues of A, or $O(\frac{1}{k})$, if the leading eigenvalue has non-trivial (i.e., of size bigger than one) Jordan blocks. Rarely, if there are several largest by modulus eigenvalues, then the "averaged" sequence $\frac{1}{r}\sum_{j=0}^{r-1} x_{k+j}$ converges to $\boldsymbol{v}$, where r is the *imprimitivity index* [1]. In all cases we will say that the power method converges and, for the sake of simplicity, assume that there is only one largest by modulus eigenvalue, maybe multiple, i.e., that $r = 1$.

Theorem 3.14 *The selected leading eigenvector of a non-negative matrix A is proportional to the limit of the power method applied to the matrix A with the initial vector $\boldsymbol{x}_0 = \boldsymbol{e}$.*

`Proof.` We may assume that $\rho(A) = 1$. The spectral radius of the matrix A_ε strictly increases in ε, so $\rho(A_\varepsilon) = 1 + \delta$, for some $\delta > 0$. We have

$$(A + \varepsilon\, \boldsymbol{e}\,\boldsymbol{e}^T)\, \boldsymbol{v}_\varepsilon \;=\; (1+\delta)\boldsymbol{v}_\varepsilon\,.$$

Denoting by $S_\varepsilon = (\boldsymbol{e}, \boldsymbol{v}_\varepsilon)$ the sum of entries of the vector $\boldsymbol{v}_\varepsilon$, we obtain

$$A\boldsymbol{v}_\varepsilon \;+\; \varepsilon\, S_\varepsilon\, \boldsymbol{e} \;=\; (1+\delta)\, \boldsymbol{v}_\varepsilon\,,$$

and hence

$$\left(I \;-\; \frac{1}{(1+\delta)}\, A\right) \boldsymbol{v}_\varepsilon \quad = \quad \frac{\varepsilon\, S_\varepsilon}{1+\delta}\, \boldsymbol{e}\,.$$

Since $\rho(A) = 1$, we have $\rho(\frac{1}{(1+\delta)}\,A) \;=\; \frac{1}{1+\delta} \;<\; 1$. Therefore, we can apply the power series expansion:

$$\left(I \;-\; \frac{1}{(1+\delta)}\, A\right)^{-1} \;=\; I \;+\; (1+\delta)^{-1}A \;+\; (1+\delta)^{-2}A^2 \;+\; \ldots$$

and obtain

$$\boldsymbol{v}_\varepsilon \quad = \quad \frac{1+\delta}{\varepsilon\, S_\varepsilon}\,\sum_{k=0}^{\infty}(1+\delta)^{-k}A^k\boldsymbol{e}\,.$$

Assume now that the power method for the matrix A and for $\boldsymbol{x}_0 = \boldsymbol{e}$ converges to some vector $\tilde{\boldsymbol{v}}$. In case $r = 1$ (r is the imprimitivity index), this means that $A^k\boldsymbol{e} \to \tilde{\boldsymbol{v}}$ as $k \to \infty$. Then, direction of the vector $\sum_{k=0}^{\infty}(1+\delta)^{-k}A^k\boldsymbol{e}$ converges as $\delta \to 0$ to the direction of the vector $\tilde{\boldsymbol{v}}$. Since $\delta \to 0$ as $\varepsilon \to 0$, the theorem follows. If $r > 1$, then $\frac{1}{r}\sum_{j=0}^{r-1}A^{k+j}\boldsymbol{e} \to \tilde{\boldsymbol{v}}$. Arguing as above, we conclude again that the direction of the vector $\sum_{k=0}^{\infty}(1+\delta)^{-k}A^k\boldsymbol{e}$ converges to the direction of the vector $\tilde{\boldsymbol{v}}$ as $\delta \to 0$. This completes the proof. □

Definition 3.15 *The greedy method with the selected leading eigenvectors $\boldsymbol{v}_k$ in all iterations is called selective greedy method.*

Now we arrive at the main result of this section. We show that using selected eigenvectors $\boldsymbol{v}_k$ avoids cycling.

Theorem 3.16 *The selective greedy method does not cycle.*

`Proof.` Consider the case of finding maximum. The case of minimum is proved in the same way. By the kth iteration of the algorithm we have a matrix A_k and its selected leading eigenvector $\boldsymbol{v}_k$. Assume the algorithm cycles: $A_1 = A_{n+1}$, and let $n \geqslant 2$ be the minimal length of the cycle (for $n = 1$, the equality $A_1 = A_2$ means that A_1 is the optimal matrix, so this is not cycling). Consider the chain of perturbed matrices $A_{1,\varepsilon} \to \cdots \to A_{n+1,\varepsilon}$, where $A_{k,\varepsilon} = A_k + \varepsilon E$ and $\varepsilon > 0$ is a small number. For the matrix A_k, in each row $\boldsymbol{a}_i^{(k)}$, we have one of two cases:

1) if $(\boldsymbol{a}_i^{(k)}, \boldsymbol{v}_k) = \max\limits_{\boldsymbol{a}_i \in \mathcal{F}_i}(\boldsymbol{a}_i, \boldsymbol{v}_k)$, then this row is not changed in this iteration: $\boldsymbol{a}_i^{(k+1)} = \boldsymbol{a}_i^{(k)}$. In this case $\boldsymbol{a}_i^{(k+1)} + \varepsilon\boldsymbol{e} = \boldsymbol{a}_i^{(k)} + \varepsilon\boldsymbol{e}$.

2) if $(\boldsymbol{a}_i^{(k)}, \boldsymbol{v}_k) < \max\limits_{\boldsymbol{a}_i \in \mathcal{F}_i}(\boldsymbol{a}_i, \boldsymbol{v}_k)$, then the row $\boldsymbol{a}_i^{(k)}$ is not optimal, and $(\boldsymbol{a}_i^{(k)}, \boldsymbol{v}_k) < (\boldsymbol{a}_i^{(k+1)}, \boldsymbol{v}_k)$. Hence the same inequality is true for the perturbed matrices $A_{k,\varepsilon}, A_{k+1,\varepsilon}$ and for the eigenvector $\boldsymbol{v}_{k,\varepsilon}$ of $A_{k,\varepsilon}$, whenever ε is small enough.

Thus, in both cases, each row of $A_{k+1,\varepsilon}$ makes a bigger or equal scalar product with $\boldsymbol{v}_{k,\varepsilon}$ than the corresponding row of $A_{k,\varepsilon}$. Moreover, at least in one row this scalar product is strictly bigger, otherwise $A_k = A_{k+1}$, which contradicts the minimality of the cycle. This, in view of strict positivity of $A_{k,\varepsilon}$, implies that $\rho_{k+1,\varepsilon} > \rho_{k,\varepsilon}$. We see that in the chain $A_{1,\varepsilon} \to \cdots \to A_{n+1,\varepsilon}$ the spectral radius strictly increases. Therefore $A_{1,\varepsilon} \neq A_{n+1,\varepsilon}$, and hence $A_1 \neq A_{n+1}$. The contradiction competes the proof. □

Exampe 3.10 [continued] We apply the selective greedy method to the family from Section 3.2, on which the (usual) greedy method cycles and arrives to a wrong solution. Now we have:

$$A_1 \to A_2 \xrightarrow{\boldsymbol{v}_2=(3,2,2)^T} A_3 = \begin{pmatrix} 12 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix}$$

$$A_3 \xrightarrow{\boldsymbol{v}_3=(1,0,0)^T} A_4 = \begin{pmatrix} 12 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 3 \end{pmatrix} \xrightarrow{\boldsymbol{v}_4=(49,5,6)^T} A_4$$

Thus, the selective greedy method arrives at the optimal matrix A_4 at the third iteration. This matrix is maximal in each row with respect to its leading eigenvector $\boldsymbol{v}_4$, as the last iteration demonstrates.

The selective greedy method repeats the first iteration of the (usual) greedy method, but in the second and in the third iteration it chooses different leading eigenvectors ("selected eigenvectors") by which it does not cycle, going directly to the optimal solution. ○

The greedy method does not cycle, provided it uses selected leading eigenvectors $\boldsymbol{v}_k$ in all iterations. If the leading eigenvector is unique, then it coincides with the selected leading eigenvector, and there is no problem. The difficulty occurs if the leading eigenvector is not simple. The following crucial fact is a straightforward consequence of Theorem 3.16.

Corollary 3.17 *The power method* $\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k$, $k \geqslant 0$, *applied to the initial vector* $\boldsymbol{x}_0 = \boldsymbol{e}$ *converges to the selected leading eigenvector.*

To realize the selective greedy method we compute all the leading eigenvectors $\boldsymbol{v}_k$ by the power method starting always with the same vector $\boldsymbol{x}_0 = \boldsymbol{e}$ (the vector of ones). After a sufficient number of iterations we come close to the limit, which is by Corollary 3.17 the selected eigenvector (up to normalization). So, we actually perform the greedy method with approximations for the selected eigenvectors, which are, in view of Theorem 3.16, the leading eigenvectors of perturbed matrices A_ε for some small ε.

Therefore, to avoid cycling we can compute all eigenvectors $\boldsymbol{v}_k$ by the power method starting with the same initial vector $\boldsymbol{x}_0 = \boldsymbol{e}$. Because of computer approximations, we actually deal with totally positive matrices of the family $\mathcal{F}_\varepsilon$ for some small ε.

We chose the precision parameter ε in advance and keep it the same in all iterations of the greedy algorithm. By Theorem 3.16, if $\varepsilon > 0$ is small enough, then the greedy method does not cycle and finds the solution within finite time. In practice, if the algorithm cycles for a given ε, then we reduce it taking, say, $\varepsilon/10$ and restart the algorithm.

Remark. To avoid any trouble with the power method for sparse matrices, we can use the following well-known fact: if $A \geq 0$ is a matrix, then the matrix $A + I$ has a unique (maybe multiple) eigenvalue equal by modulus to $\rho(A) + 1$. This implies that the power method applied for the matrix $A + I$ always converges to its leading eigenvector, i.e., $(A + I)^k \boldsymbol{e} \to \boldsymbol{v}$ as $k \to \infty$. Of course, $A + I$ has the same leading eigenvector as A. Hence, we can replace each uncertainty set $\mathcal{F}_i$ by the set $\boldsymbol{e}_i + \mathcal{F}_i$, where $\boldsymbol{e}_i$ is the ith canonical basis vector. After this each matrix A_k is replaced by $A_k + I$, and we will not have trouble with the convergence of the power method.

3.4 The convergence rate of the greedy method

In this section we address two issues: revealing the secret of the extremely fast convergence of the greedy method and getting an explanation why does the spectral simplex method converges slower (although very fast as well). We restrict ourselves to the maximization problems For minimization problems, the results and the proofs are the same. The results of this section apply to the original greedy procedure, as presented in Section 3.1, which is not necessarily equipped with selective power method.

First of all, let us remark that due to possible cycling, no efficient estimates for the rate of convergence exist. Indeed as we saw in Example 3.10 the greedy method may not converge to the solution at all. That is why we estimate the rate of convergence only under some favorable assumptions (positivity of the limit matrix, positivity of its leading eigenvector, etc.) These assumptions are not very restrictive since the selective greedy method has the same convergence rate as the greedy method for a strictly positive family $\mathcal{F}_\varepsilon$.

We are going to show that under those assumptions, both methods have global linear convergence, but the greedy method, has even a local quadratic convergence. In both cases we estimate the parameters of linear and quadratic convergence.

Global linear convergence

The following result is formulated for both the greedy and the spectral simplex method. We denote by A_k the matrix obtained by the kth iteration, by ρ_k its spectral radius and by $\boldsymbol{u}_k, \boldsymbol{v}_k$ its left and right leading eigenvectors (if there are multiple ones, we take any of them). We also denote $\rho_{\max} = \max_{A \in \mathcal{F}} \rho(A)$. As usual, $\{\boldsymbol{e}_j\}_{j=1}^d$ is the canonical basis in $\mathbb{R}^d$ and $\boldsymbol{e}$ is the vector of ones.

Theorem 3.18 *In both the greedy method and the spectral simplex method, in each iteration we have*

$$\rho_{k+1} \ \geqslant \ \rho_k \ + \ (\rho_{\max} - \rho_k) \max_{j=1,\dots,d} \frac{u_{k+1,j}\, v_{k,j}}{(\boldsymbol{u}_{k+1}, \boldsymbol{v}_k)}\,, \tag{3.4}$$

provided $\boldsymbol{v}_k > 0$.

`Proof.` Consider the value $s = s(A)$ defined in (3.1). Let the maximum in formula (3.1) be attained in the mth row. Then for every matrix $A \in \mathcal{F}$, we have $A\boldsymbol{v}_k \leqslant s\,\boldsymbol{v}_k$, and therefore $\rho(A) \leqslant s$. Thus, $\rho_{\max} \leqslant s$. On the other hand, in both greedy and simplex methods, the mth component of the vector $A_{k+1}\boldsymbol{v}_k$ is equal to the mth component of the vector $s\,\boldsymbol{v}_k$. In all other components, we have $A_{k+1}\boldsymbol{v}_k \geqslant A_k\boldsymbol{v}_k = \rho_k\,\boldsymbol{v}_k$. Therefore,

$$A_{k+1}\boldsymbol{v}_k \ \geqslant\ \rho_k\boldsymbol{v}_k \ +\ (s-\rho_k)v_{k,j}\boldsymbol{e}_j\,.$$

Multiplying this inequality by the vector $\boldsymbol{u}_{k+1}$ from the left, we obtain

$$\boldsymbol{u}_{k+1}^T A_{k+1}\boldsymbol{v}_k \ \geqslant\ \rho_k(\boldsymbol{u}_{k+1},\boldsymbol{v}_k) \ +\ (s-\rho_k)\,v_{k,j}\,(\boldsymbol{u}_{k+1},\boldsymbol{e}_j)\,.$$

Since $\boldsymbol{u}_{k+1}^T A_{k+1} = \rho_{k+1}\boldsymbol{u}_{k+1}^T$ and $(\boldsymbol{u}_{k+1},\boldsymbol{e}_j) = u_{k+1,j}$, we have

$$\rho_{k+1}(\boldsymbol{u}_{k+1},\boldsymbol{v}_k) \ \geqslant\ \rho_k(\boldsymbol{u}_{k+1},\boldsymbol{v}_k) \ +\ (s-\rho_k)v_{k,j}u_{k+1,j}\,.$$

Dividing by $(\boldsymbol{u}_{k+1},\boldsymbol{v}_k)$ and taking into account that $s \geqslant \rho_{\max}$, we arrive at (3.4). □

Rewriting (3.4) in the form

$$\rho_{\max} \ -\ \rho_{k+1} \quad\leqslant\quad (\rho_{\max}\ -\ \rho_k)\left(1 - \frac{u_{k+1,j}\,v_{k,j}}{(\boldsymbol{u}_{k+1},\boldsymbol{v}_k)}\right)\,,$$

we see that the kth iteration reduces the distance to the maximal spectral radius in at least $\left(1\ -\ \frac{u_{k+1,j}\,v_{k,j}}{(\boldsymbol{u}_{k+1},\boldsymbol{v}_k)}\right)$ times. Thus, we have established

Corollary 3.19 *Each iteration of the greedy method and of the spectral simplex method reduces the distance to the optimal spectral radius in at least* $\left(1\ -\ \frac{u_{k+1,j}\,v_{k,j}}{(\boldsymbol{u}_{k+1},\boldsymbol{v}_k)}\right)$ *times.*

If $A_k > 0$, then the contraction coefficient from Corollary 3.19 can be roughly estimated from above. Let m and M be the smallest and the biggest entries, respectively, of all vectors from the uncertainty sets. For each $A \in \mathcal{F}$, we denote by $\boldsymbol{v}$ its leading eigenvector and by ρ its spectral radius. Let also S be the sum of entries of $\boldsymbol{v}$. Then for each i, we have $v_i = \rho^{-1}(A\boldsymbol{v},\boldsymbol{e}_i) \leqslant \rho^{-1}MS$. Similarly, $v_i \geqslant \rho^{-1}mS$. Hence, for every matrix from $\mathcal{F}$, the ratio between the smallest and of the biggest entries of its leading eigenvector is at least m/M. The same is true for the left eigenvector. Therefore,

$$\frac{u_{k+1,j}v_{k,j}}{(\boldsymbol{u}_{k+1},\boldsymbol{v}_k)} \quad\geqslant\quad \frac{m^2}{m^2+(d-1)M^2}.$$

We have arrived at the following theorem on the global linear convergence

Theorem 3.20 *If all uncertainty sets are strictly positive, then the rate of convergence of both the greedy method and of the spectral simplex method is at least linear with the factor*

$$q \quad\leqslant\quad 1\ -\ \frac{m^2}{m^2+(d-1)M^2}\,, \qquad (3.5)$$

where m and M are the smallest and the biggest entries respectively of vectors from the uncertainty sets.

Thus, in each iteration of the greedy method and of the spectral simplex method, we have

$$\rho_{\max} - \rho_{k+1} \leqslant q^k (\rho_{\max} - \rho_0), \tag{3.6}$$

Local quadratic convergence

Now we are going to see that the greedy method has actually a quadratic rate of convergence in a small neighbourhood of the optimal matrix. This explains its fast convergence. We establish this result under the following two assumptions: 1) the optimal matrix A_*, for which $\rho(A_*) = \rho_{\max}$, is strictly positive; 2) in a neighbourhood of A_*, the uncertainty sets $\mathcal{F}_i$ are bounded by C^2 smooth strictly convex surfaces. The latter is, of course, not satisfied for polyhedral sets. Nevertheless, the quadratic convergence for sets with a smooth boundary explains the fast convergence for finite or for polyhedral sets as well.

The local quadratic convergence means that there are constants $B > 0$ and $\varepsilon \in (0, \frac{1}{B})$ for which the following holds: if $\|A_k - A_*\| \leqslant \varepsilon$, then $\|A_{k+1} - A_*\| \leqslant B\,\|A_k - A_*\|^2$. In this case, the algorithm converges whenever $\|A_1 - A_*\| \leqslant \varepsilon$, in which case for every iteration k, we have

$$\|A_k - A_*\| \leqslant \frac{1}{B}\,(B\,\|A_1 - A_*\|)^{2^k - 1}$$

(see, for instance, [43]).

Here we use Euclidean norm $\|\boldsymbol{x}\| = \sqrt{(\boldsymbol{x}, \boldsymbol{x})}$. We also need the L_∞-norm $\|\boldsymbol{x}\|_\infty = \max_{i=1,\dots,d}|x_i|$. $A \preceq B$ means that the matrix $B - A$ is positive semidefinite.

Theorem 3.21 *Suppose the optimal matrix A_* is strictly positive and all uncertainty sets $\mathcal{F}_i$ are strongly convex and C^2 smooth in a neighbourhood of A_*; then the greedy algorithm has a local quadratic convergence to A_*.*

In the proof we use the following auxiliary fact

Lemma 3.22 *If A is a strictly positive matrix with $\rho(A) = 1$ and with the leading eigenvector $\boldsymbol{v}$, then for every $\varepsilon > 0$ and for every vector $\boldsymbol{v}' \geqslant 0$, $\|\boldsymbol{v}'\| = 1$, such that $\|A\boldsymbol{v}' - \boldsymbol{v}'\|_\infty \leqslant \varepsilon$, we have $\|\boldsymbol{v} - \boldsymbol{v}'\| \leqslant C(A)\,\varepsilon$, where $C(A) = \frac{1}{m}\,(1 + (d-1)\frac{M^2}{m^2})^{1/2}$, m and M are minimal and maximal entry of A respectively.*

`Proof.` Let $\boldsymbol{v} = (v_1, \dots, v_d)$. Denote $\boldsymbol{v}' = (v_1 s_1, \dots, v_d s_d)$, where s_i are some nonnegative numbers. Choose one for which the value $|s_i - 1|$ is maximal. Let it be s_1 and let $s_1 - 1 > 0$ (the opposite case is considered in the same way). Since $\|\boldsymbol{v}'\| = \|\boldsymbol{v}\|$ and $s_1 > 1$, it follows that there exists q for which $s_q < 1$. Since $A\boldsymbol{v} = \boldsymbol{v}$, we have $\sum_{j=1}^d a_{1j} v_j = v_1$. Therefore,

$$\begin{aligned} (\boldsymbol{v}' - A\boldsymbol{v}')_1 \;&=\; s_1 v_1 \;-\; \sum_{j=1}^d a_{1j} v_j s_j \;= \\ &\sum_{j=1}^d a_{1j} v_j s_1 \;-\; \sum_{j=1}^d a_{1j} v_j s_j \;= \sum_{j=1}^d a_{1j} v_j (s_1 - s_j). \end{aligned} \tag{3.7}$$

The last sum is bigger than or equal to one term $a_{1q}v_m(s_1 - s_q) \geqslant a_{1q}v_q(s_1 - 1) \geqslant m(v_1' - v_1)$. Note that for all other i, we have $|s_i - 1| \leqslant |s_1 - 1|$, hence $|v_i' - v_i| \leqslant \frac{v_i}{v_1}|v_1' - v_1| \leqslant \frac{M}{m}|v_1' - v_1|$. Therefore, $\|\boldsymbol{v}' - \boldsymbol{v}\| \leqslant (1 + (d-1)\frac{M^2}{m^2})^{1/2}|v_1' - v_1|$. Substituting to (3.7), we get

$$(\boldsymbol{v}' - A\boldsymbol{v}')_1 \ \geqslant \ m|v_1' - v_1| \geqslant \ m\left(1 + (d-1)\frac{M^2}{m^2}\right)^{-1/2} \|\boldsymbol{v}' - \boldsymbol{v}\|$$

Hence, $\|A\boldsymbol{v}' - \boldsymbol{v}'\|_\infty \geqslant \frac{1}{C(A)}\|\boldsymbol{v}' - \boldsymbol{v}\|$, which completes the proof. □

`Proof of Theorem 3.21.` Without loss of generality it can be assumed that $\rho(A_*) = 1$. It is known that any strongly convex smooth surface Γ can be defined by a smooth convex function f so that $\boldsymbol{x} \in \Gamma \Leftrightarrow f(\boldsymbol{x}) = 0$ and $\|f'(\boldsymbol{x})\| = 1$ for all $\boldsymbol{x} \in \Gamma$. For example, one can set $f(\boldsymbol{x})$ equal to the distance from $\boldsymbol{x}$ to Γ for $\boldsymbol{x}$ outside Γ or in some neighbourhood inside it. For each $i = 1, \ldots, d$, we set $\Gamma_i = \partial\mathcal{F}_i$ and denote by f_i such a function that defines the surface Γ_i. Fix an index $i = 1, \ldots, d$ and denote by $\boldsymbol{a}^{(k)}, \boldsymbol{a}^{(k+1)}$, and $\boldsymbol{a}^*$ the ith rows of the matrices A_k, A_{k+1}, and A_* respectively (so, we omit the subscript i to simplify the notation). Since $\boldsymbol{a}^* = \mathrm{argmax}_{\boldsymbol{a} \in \Gamma_i}(\boldsymbol{a}, \boldsymbol{v}^*)$, it follows that $f_i'(\boldsymbol{a}^*) = \boldsymbol{v}^*$. Denote by L and ℓ the lower and upper bounds of the quadratic form $f''(\boldsymbol{a})$ in some neighbourhood of $\boldsymbol{a}^*$, i.e., $\ell I \preceq f''(\boldsymbol{a}) \preceq LI$ at all points $\boldsymbol{a}$ such that $\|\boldsymbol{a} - \boldsymbol{a}^*\| < \varepsilon$. Writing the Tailor expansion of the function f_i at the point $\boldsymbol{a}^*$, we have for small $\boldsymbol{h} \in \mathbb{R}^d$:

$$f_i(\boldsymbol{a}^* + \boldsymbol{h}) \ = \ f_i(\boldsymbol{a}^*) \ + \ (f_i'(\boldsymbol{a}^*), \boldsymbol{h}) \ + \ r(\boldsymbol{h})\|\boldsymbol{h}\|^2,$$

where the remainder $r(\boldsymbol{h}) \leqslant L$. Substituting $\boldsymbol{h} = \boldsymbol{a}^{(k)} - \boldsymbol{a}^*$ and taking into account that $f_i(\boldsymbol{a}^{(k)}) = f_i(\boldsymbol{a}^*) = 0$, because both $\boldsymbol{a}^{(k)}$ and $\boldsymbol{a}^*$ belong to Γ, we obtain $(f_i'(\boldsymbol{a}^*), \boldsymbol{h}) + r(\boldsymbol{h})\|\boldsymbol{h}\|^2 = 0$. Hence $|(\boldsymbol{v}^*, \boldsymbol{h})| \leqslant L\|\boldsymbol{h}\|^2$. Thus,

$$|(\boldsymbol{v}^*, \boldsymbol{a}^{(k)}) \ - \ (\boldsymbol{v}^*, \boldsymbol{a}^*)| \ \leqslant \ L\,\|\boldsymbol{h}\|^2\,.$$

This holds for each row of A_k, therefore $\|(A_k - I)\boldsymbol{v}^*\|_\infty \leqslant L\|\boldsymbol{h}\|^2$, and hence $\|\boldsymbol{v}_k - \boldsymbol{v}^*\| \leqslant C(A_k)\,L\,\|\boldsymbol{h}\|^2$, where the value $C(A_k)$ is defined in Lemma 3.22.

Further, $\boldsymbol{a}^{(k+1)} = \mathrm{argmax}_{\boldsymbol{a} \in \Gamma_i}(\boldsymbol{a}, \boldsymbol{v}_k)$, hence $f_i'(\boldsymbol{a}^{(k+1)}) = \boldsymbol{v}_k$. Consequently,

$$\|\boldsymbol{v}_k - \boldsymbol{v}^*\| \ = \ |f_i'(\boldsymbol{a}^{(k+1)}) - f_i'(\boldsymbol{a}^*)| \ \geqslant \ \ \ell\,\|\boldsymbol{a}^{(k+1)} - \boldsymbol{a}^*\|.$$

Therefore, $\|\boldsymbol{a}^{(k+1)} - \boldsymbol{a}^*\| \leqslant \frac{C(A_k)L}{\ell}\|\boldsymbol{h}\|^2 = \frac{C(A_k)L}{\ell}\|\boldsymbol{a}^{(k+1)} - \boldsymbol{a}^*\|^2$. This holds for each row of the matrices, hence the theorem follows. □

In the proof of Theorem 3.21 we estimated the parameter B of the quadratic convergence in terms of the parameters ℓ and L of the functions f_i''. This estimate, however, is difficult to evaluate, because the functions f_i are a priori unknown. To estimate B effectively, we need another idea based on geometrical properties of the uncertainty sets $\mathcal{F}_i$. Below we obtain such an estimate in terms of radii of curvature of the boundary of $\mathcal{F}_i$.

Assume that at each point of intersection of the surface $\Gamma_i = \partial\mathcal{F}_i$ and of the corresponding ε-neighbourhood of the point $\boldsymbol{a}_i^*$, the maximal radius of curvature does not exceed R and the minimal radius of curvature is at least $r > 0$. Denote also by C the maximal value of $C(A)$ over the corresponding neighbourhood of the matrix A^*, where $C(A)$ is defined in Lemma 3.22.

Theorem 3.23 *For the greedy method, we have* $B \leqslant \frac{RC}{2r\rho(A_*)}$.

`Proof.` As in the proof of Theorem 3.21, we assume that $\rho(A_*) = 1$ and denote by $\boldsymbol{a}^{(k)}, \boldsymbol{a}^{(k+1)}$, an $\boldsymbol{a}^*$ the ith rows of the matrices A_k, A_{k+1}, and A_* respectively. Since $\boldsymbol{a}^* = \operatorname{argmax}_{\boldsymbol{a}\in\Gamma_i} (\boldsymbol{a}, \boldsymbol{v}^*)$, it follows that $\boldsymbol{v}^*$ is the outer unit normal vector to Γ_i at the point $\boldsymbol{a}^*$. Denote $\boldsymbol{h} = \boldsymbol{a}^{(k)} - \boldsymbol{a}^*$. Draw a Euclidean sphere $\mathcal{S}_i$ of radius r tangent to Γ_i from inside at the point $\boldsymbol{a}^*$. Take a point $\boldsymbol{b}^{(k)}$ on $\mathcal{S}_i$ such that $\|\boldsymbol{b}^{(k)} - \boldsymbol{a}^*\| = \|\boldsymbol{h}\|$. Since the maximal radius of curvature of Γ_i is at least R, this part of the sphere is inside $\mathcal{F}_i$, hence the vector $\boldsymbol{a}^{(k)} - \boldsymbol{a}^*$ forms a smaller angle with the normal vector $\boldsymbol{v}^*$ than the vector $\boldsymbol{b}^{(k)} - \boldsymbol{a}^*$. The vectors $\boldsymbol{a}^{(k)} - \boldsymbol{a}^*$ and $\boldsymbol{b}^{(k)} - \boldsymbol{a}^*$ are of the same length, therefore

$$(\boldsymbol{a}^{(k)} - \boldsymbol{a}^*\,,\, \boldsymbol{v}^*) \ \geqslant\ (\boldsymbol{b}^{(k)} - \boldsymbol{a}^*\,,\, \boldsymbol{v}^*) \ =\ -\frac{\|\boldsymbol{h}\|^2}{2r}$$

On the other hand, $(\boldsymbol{a}^{(k)} - \boldsymbol{a}^*, \boldsymbol{v}^*) \leqslant 0$. Therefore,

$$\Big| (\boldsymbol{a}^{(k)}\,,\, \boldsymbol{v}^*) \ -\ (\boldsymbol{a}^*\,,\, \boldsymbol{v}^*) \Big| \ \leqslant\ \frac{\|\boldsymbol{h}\|^2}{2r}$$

This inequality holds for all rows of the matrix A_k, therefore

$$\|(A_k - I)\boldsymbol{v}^*\|_\infty = \ \|(A_k - A^*)\bar{\boldsymbol{v}}\|_\infty \leqslant \frac{\|\boldsymbol{h}\|^2}{2r},$$

and hence, by Lemma 3.22,

$$\|\boldsymbol{v}_k - \boldsymbol{v}^*\| \ \leqslant\ \frac{C}{2r}\,\|\boldsymbol{h}\|^2. \tag{3.8}$$

Further, $\boldsymbol{a}^{(k+1)} = \operatorname{argmax}_{\boldsymbol{a}\in\Gamma_i} (\boldsymbol{a}, \boldsymbol{v}_k)$, hence $\boldsymbol{v}_k$ is a normal vector to Γ_i at the point $\boldsymbol{a}^{(k+1)}$. Consequently

$$R\,\|\boldsymbol{v}_k - \boldsymbol{v}^*\| \geqslant \ \|\boldsymbol{a}^{(k+1)} - \boldsymbol{a}^*\|.$$

Combining this inequality with (3.8) we obtain $\|\boldsymbol{a}^{(k+1)} - \boldsymbol{a}^*\| \leqslant \frac{CR}{2r}\,\|\boldsymbol{h}\|^2$, which concludes the proof. □

Remark. The results of this section rely on the assumption that for the optimal matrix A_* we have $\rho(A_*) = 1$. This can always be done, without the loss of generality, if $\rho(A_*) > 0$. Thus, all the results apply for the optimal matrices with non-zero spectral radius. However, for the case $\rho(A_*) = 0$ we cannot guarantee that the derived convergence rates hold. Even tough selective greedy method will not cycle in this case either, it may considerably slow down, which is also noticed in numerical simulations.

3.5 Selective greedy method: numerical results

We now demonstrate and analyse numerical results of the selective greedy method. Several types of problems are considered: maximizing and minimizing the spectral radius over finite and polyhedral uncertainty sets.

Product families with finite uncertainty sets

We first test the selective greedy method on positive product families, and then on non-negative product families with density parameters $\gamma_i \in (0, 1)$ (the percentage of nonzero entries of the vectors from the uncertainty set $\mathcal{F}_i$.) The results of tests on positive product families are given in Tables 3.1 and 3.2, for maximization and minimization respectively. In Tables 3.3 and 3.4 we report the behaviour of the selective greedy algorithm as dimension d and the size of the uncertainty sets N vary. For each uncertainty set $\mathcal{F}_i$ the density parameter γ_i is randomly chosen from the interval $9 - 15\%$ and the elements of the set $\mathcal{F}_i$ are randomly generated in accordance to the parameter γ_i. Table 3.3 contains the data for the maximisation, while Table 3.4 presents the results for the minimization problem. The precision of the power method for obtaining the leading eigenvalue is set to $\varepsilon = 10^{-7}$.

avg. number of iterations

$N \setminus d$	25	250	500	2000
50	3	3.2	3.1	3
100	3.2	3.2	3.1	3.1
250	3.1	3.1	3.1	3.1

avg. running time

$N \setminus d$	25	250	500	2000
50	0.02s	0.23s	0.46s	2.15s
100	0.04s	0.47s	0.89s	5.19s
250	0.1s	1.14s	2.1s	416.85s

Table 3.1: Selective greedy method, maximization, positive families

avg. number of iterations

$N \setminus d$	25	250	500	2000
50	3.2	3.1	3.1	3
100	3.2	3.1	3.1	3.1
250	3.5	3.2	3.2	3.1

avg. running time

$N \setminus d$	25	250	500	2000
50	0.01s	0.22s	0.45s	2.06s
100	0.05s	0.43s	0.87s	4.91s
250	0.1s	1.05s	2s	392.29s

Table 3.2: Selective greedy method, minimization, positive families

The performed experiments on positive families demonstrate how the choice of the numerical method for computing the leading eigenvector affects the computational time. This can be clearly seen by comparing the above results with the ones given in Section 2.2. For sparse matrices we have obtained the following results.

avg. number of iterations

$N \setminus d$	25	250	500	2000
50	5.7	4.2	4.2	4.2
100	5.9	4.4	4.4	4.1
250	6.2	4.6	4.6	4.1

avg. running time

$N \setminus d$	25	250	500	2000
50	0.04s	0.31s	0.61s	2.93s
100	0.08s	0.61s	1.28s	8.16s
250	0.18s	1.51s	3.05s	149.33s

Table 3.3: Selective greedy method, maximization, non-negative families

avg. number of iterations

$N \setminus d$	25	250	500	2000
50	8	4.5	4.4	4.2
100	7.2	4.5	4.5	4.2
250	7.4	4.7	4.6	4.1

avg. running time

$N \setminus d$	25	250	500	2000
50	0.06s	0.31s	0.66s	2.96s
100	0.11s	0.69s	1.27s	6.41s
250	0.24s	1.62s	2.96s	149.33s

Table 3.4: Selective greedy method, minimization, non-negative families

Table 3.5 shows how the sparsity affects the computations. The dimension is kept fixed at $d = 700$ and the cardinality of each product set at $|\mathcal{F}_i|= 200$, while we vary the percentage from which the density parameter takes value.

%	0 - 8	9 - 15	16 - 21	22 - 51	52 - 76	77 - 100
MAX	5.2	4.4	4	4	3.9	3.3
MIN	4.3	4.5	4.1	4.1	3.9	3.6

Table 3.5.1: Effects of sparsity, number of iterations

%	0 - 8	9 - 15	16 - 21	22 - 51	52 - 76	77 - 100
MAX	3.98s	3.36s	3.01s	3.01s	2.89s	2.96s
MIN	13.71s	3.27s	2.98s	3s	2.87s	2.64s

Table 3.5.2: Effects of sparsity, number of iterations

Although the number of iterations remains almost unchanged, we can notice a significant increase in the running time when minimizing on very sparse families. This is because the power method tends to slow down when computing the leading eigenvector for matrices with zero spectral radius, which are usually the solution in this case. We can avoid this by simply quitting the greedy procedure once a matrix with zero spectral radius is obtained. Even tough this matrix might not be minimal in each row, we can take it as a minimal solution, since spectral radius cannot go below zero.

Apart from the very sparse case, the number of iterations seems to depend neither on the dimension, nor on the cardinality of the uncertainty sets nor on the sparsity itself. The usual number of iterations is 3 - 4. The robustness to sparsity can be explained by the fact that the selective greedy method actually works with positive perturbations of matrices. The independence on the cardinality of the uncertainty sets is, most likely, explained by the quadratic convergence[1]. The distance to the optimal matrix quickly decays to the tolerance parameter $\varepsilon = 10^{-7}$, and the algorithm terminates.

[1] In the tables we see that the number of iterations may even decrease in cardinality.

Polyhedral product families

Here we test the selective greedy method on product families with uncertainty sets given by systems of $2d + N$ linear constraints of the form:

$$\begin{cases} (\boldsymbol{x}, \boldsymbol{b}_j) \leqslant 1 & j = 1, \dots, N \\ 0 \leqslant x_i \leqslant 1 & i = 1, \dots, d \end{cases}$$

where vectors $\boldsymbol{b}_j \in \mathbb{R}^d_+$ are randomly generated and normalized, and $\boldsymbol{x} = (x_1, \dots, x_d)$.

avg. number of iterations

$N \setminus d$	5	10	50
10	4.7	4.5	4.1
25	4.4	4.9	4.9
75	4.8	5.1	5.1

avg. running time

$N \setminus d$	5	10	50
10	1.05s	1.25s	2.26s
25	3.49s	5.17s	15.56s
75	26.32s	49.19s	206.38s

Table 3.6: Selective greedy method, maximization, polyhedral sets

In polyhedral uncertainty sets the algorithm searches the optimal rows among the vertices. However, the number of vertices can be huge. Nevertheless, the selective greedy method still needs about 5 iterations to find an optimal matrix.

3.6 Optimizing spectral radius on reducible families

If we finish the (selective) greedy procedure for maximization with a leading eigenvector $\boldsymbol{v}_k$ that has zero entries, we might not obtain an optimal solution. In this case the product family $\mathcal{F}$ is reducible. One way to resolve this is to use the Frobenius factorization and run the maximization procedure on irreducible blocks $\mathcal{F}^{(j)}$, constructing the optimal solution as the block matrix. The optimal value $\rho_{\max}$ is the largest of all optimal values $\rho^{(j)}_{\max}$ among all blocks [26]. The algorithm for the Frobenius factorization can be found in [48].

In practice, however, the Frobenius factorisation usually takes more time than one run of the greedy algorithm for the whole family. That is why it makes sense to avoid the factorisation by making the family irreducible. Since having only one irreducible matrix is enough to make the whole family irreducible, a simple strategy is to just include a matrix αP to the family $\mathcal{F}$, where P is a cyclic permutation matrix and $\alpha > 0$. The matrix P is irreducible, and so the family $\mathcal{F} \cup \{\alpha P\}$ is irreducible as well, hence the greedy method has to finish with a positive leading eigenvector. If the final matrix has no rows from αP, then it is optimal for the original family $\mathcal{F}$. Otherwise we have to make α smaller. However, if after trying out several values of α produces the final matrix with some rows from αP, we then need to resort to Frobenius factorization.

Chapter 4

Applications of the selective greedy method to non-negative matrices

This chapter considers some applications of optimizing the spectral radius on non-negative families. First section deals with the problem of finding the closest stable non-negative matrix (i.e. *Schur stabilization*). This issue was already dealt with in [27] by using the greedy method and a combinatorial approach to handle the sparse matrices. Here we implement the selective greedy method instead, along with some changes, improving the computational time.

Further on, we apply the selective greedy method to (directed) graphs. We use the maximization for obtaining the graph with maximum spectral radius among all the graphs with prescribed outgoing edges. Minimization is, on the other hand, used to find a *closest acyclic subgraph* to a given graph. We also tackle the problem of *maximal acyclic subgraph (MAS).*

We will use the following in this chapter:

Lemma 4.1 [27] *A non-negative matrix A is Schur stable if and only if the matrix $(I - A)^{-1}$ is well defined and non-negative.* □

Definition 4.2 *A* support *of a vector $\boldsymbol{v} \geqslant 0$, denoted by $\mathcal{S}$, is the set of all indices i for which $v_i > 0$.*

4.1 Closest stable non-negative matrix

Definition 4.3 *A non-negative matrix A is* strongly Schur stable *if $\rho(A) < 1$, and* weakly Schur stable *if $\rho(A) \leqslant 1$. Otherwise, if $\rho(A) > 1$ we say it is* strongly Schur unstable*, and if $\rho(A) \geqslant 1$ we say it is* weakly Schur unstable*.*

The Schur stabilization problem is well known and has been studied in the literature. Finding the closest *non-negative* matrix is important in applications such as dynamical systems, mathematical economy, population dynamics, etc. [15, 18, 27]. Usually the closest matrix is defined in the Euclidean or in the Frobenius norms.

In both cases the problem is hard. It was first noted in [27] that in L_∞-norm there are efficient methods for finding the global minimum[1]. Let us recall that $\|A\|_\infty = \max_{i=1,\dots,d} \sum_{j=1}^{d} |a_{ij}|$. Formally, for a non-negative matrix A with $\rho(A) > 1$ we need to solve

$$\begin{cases} \|X - A\|_\infty \to \min : \ X \geqslant 0, \\ \rho(X) = 1. \end{cases} \tag{4.1}$$

The key observation is that for every $\tau > 0$, the ball of radius τ in L_∞-norm centered in A, i.e., the set of non-negative matrices

$$\mathcal{B}_\tau^+(A) \ = \ \Big\{ X \geqslant 0 \ \Big| \ \|X - A\|_\infty \leq \tau \Big\}$$

is a product family with the polyhedral uncertainty sets

$$\mathcal{B}_{i,\tau}^+(A) = \Big\{ \boldsymbol{x} \in \mathbb{R}_+^d \ \Big| \ \sum_{j=1}^{d} |x_j - a_{ij}| \leqslant \tau \Big\}.$$

Hence, for each τ, the problem $\rho(X) \to \min$, $X \in \mathcal{B}_\tau^+(A)$, can be solved by the selective greedy method. Then the minimal τ such that $\rho(X) \leqslant 1$ is the solution of problem (6.10). This τ can be found merely by bisection.

Now we propose a modification of the algorithm for Schur stabilization, given in [27].

`Alg.  SCS: Schur stabilization in` L_∞ `norm`

`Step 0.` Take $\frac{\|A\|}{2}$ as the starting value for τ.

`Step 1.` Start the selective greedy method procedure for minimizing the spectral radius on the ball of non-negative matrices $\mathcal{B}_\tau^+(A)$. Iterate until a matrix X is obtained with $\rho(X) < 1$. When this is done, stop the greedy procedure, compute its leading eigenvector $\boldsymbol{v}$ and rearrange its positive entries $\boldsymbol{v}$: $v_{j_1} \geqslant \cdots \geqslant v_{j_m}$, where $\mathcal{S}$ is support of $\boldsymbol{v}$, and $|\mathcal{S}| = m$. Proceed to the next step.

Else, if the greedy procedure finishes finding the Schur unstable matrix as the optimal one on $\mathcal{B}_\tau^+(A)$, keep implementing the bisection in τ, until the ball $\mathcal{B}_\tau^+(A)$ containing strongly Schur stable matrix is reached.

`Step 2.` We construct matrices $C = (c_{ij})$ and R, as follows. For each $i \in \mathcal{S}$

$$c_{ij_k} = \begin{cases} 0, & k < l_i \\ \sum_{s=1}^{l_i} a_{ij_s}, & k = l_i \\ a_{ij_k}, & k > l_i \end{cases}$$

where l_i the minimal index for which $\sum_{s=1}^{l_i} a_{ij_s} > \tau$. If this l_i does not exist, we put $l_i = m$. R is a boolean matrix with ones on positions (i, l_i) and zeros in all other places. If some of the indices i, j are not in the support, then we put $c_{ij} = x_{ij}$. We

[1] The results from L_∞-norm can also be applied for L_1-norm; we elaborate this in Chapter 6.

can write $X = C - \tau R$.

Denote by τ_* a potential optimal value of the problem (6.10). To determine it, proceed to the next step.

`Step 3.` Because $\rho(X) < 1$, we have by Lemma 4.1 that $I - X$ is invertible and

$$(I - X)^{-1} = [I - (C - \tau R)]^{-1} \geqslant 0.$$

Since $1 = \rho(C - \tau_* R) = \rho(C - \tau R + (\tau - \tau_*)R)$, it follows that $\det(I - (C - \tau R) - (\tau - \tau_*)R) = 0$. From here we have

$$\det\Big(\frac{1}{\tau - \tau_*} I - [I - (C - \tau R)]^{-1} R\Big) = 0.$$

Matrix $[I - (C - \tau R)]^{-1} R$ is non-negative and $\lambda = \frac{1}{\tau - \tau_*}$ is its (positive) leading eigenvalue.
We find the potential optimal value τ_* by computing the leading eigenvalue λ of the matrix $[I - (C - \tau R)]^{-1} R$, and then calculating

$$\tau_* = \tau - \frac{1}{\lambda}.$$

`Step 4.` To check if τ_* is really optimal, start iterating through the greedy procedure on the ball $\mathcal{B}^+_{\tau_*}(A)$, as in the `Step 1`.

If in some iteration a matrix Y is obtained with $\rho(Y) < 1$, then τ_* is not optimal. Stop the greedy procedure, return to `Step 1`, and continue doing the bisection taking now τ_* as the starting value.

Else, if we finish the greedy procedure obtaining the matrix X_* with minimal spectral radius $\rho(X_*) = 1$ on the ball $\mathcal{B}^+_{\tau_*}(A)$, we are done: τ_* is the optimal value for the (6.10), with X_* as the corresponding optimal solution. $\diamond$

In Tables 4.1 and 4.2 we compare the running times for Schur stabilization reported in [27] (t_{np}) with the performance of Algorithm SCS (t_{scs}).

d	50	250	500	1000
t_{np}	0.15s	9.08s	44.02s	302.43s
t_{scs}	0.45s	7.12s	17.93s	221.28s

Table 4.1: Schur stabilization for positive matrices

d	50	250	500	1000
t_{np}	0.15s	20.26s	140.9s	717.25s
t_{scs}	0.4s	7.89s	23.21s	170.81s

Table 4.2: Schur stabilization for sparse matrices, with 9-15% sparsity of each row

If we compare these results with the numerical results for Schur stabilization from [27], we can observe a remarkable speed-up in the computational time. The first reason for this is because in the updated algorithm we do not need to conduct the greedy method until the end: we quit it as soon as the strongly Schur stable matrix is obtained. By quitting the greedy method before it finishes, we usually avoid obtaining matrices with zero spectral radius. As practical experiments showed, computation of the leading eigenvector for the zero spectral radius matrices can be drastically slow, especially for the very sparse big matrices. The big chunk of computational time in the implementation of the algorithm from [27] in fact goes for iterating through zero spectral radius matrices. In the updated procedure this is effectively avoided, providing us with significantly faster computations. The following table shows how the sparsity of the starting matrix affects the running time (of Algorithm SCS).

γ_i	3 - 8	9 - 15	16 - 21	22 - 51	52 - 76	77 - 100
t	137.49s	91.71s	179.73s	228.07s	292.15s	238.38s

Table 4.3: Effects of sparsity

Certainly, the dimension has a major effect on the computational time. However, how often we come across zero spectral radius matrices has also an important impact on the running duration.

We remark that for the big majority of the performed experiments we set the precision of the power method to $\varepsilon = 10^{-7}$. It is important to have a high precision, since for higher dimensions the procedure gets really sensitive, because the entries of the leading eigenvector get pretty close to one another. This affects the procedure which is dependant on the ordering of the entries. Therefore, the higher precision of the power method guarantees us the computed ordering, if not the same, then very similar to the exact one, thus giving us overall more precise result. On the other hand the bad ordering can lead to the bad behaviour of the algorithm. In few cases this precision was not enough and we needed to set $\varepsilon = 10^{-8}$ so the code could finish.

Another thing to point out is that in our implementation of the SCS algorithm we set the tolerance parameter θ to $\pm 10^{-4}$. This means that we accepted a matrix X_* as the solution if $\rho(X_*) = 1 \pm 10^{-4}$. If we want to have a better precision of the solution, then we need to set the precision parameter ε even smaller, which can increase the computational time.

Example 4.4 For the Schur unstable matrix

$$A = \begin{pmatrix} 0 & 0 & 5 & 0 & 0 & 9 & 0 & 0 & 8 & 0 \\ 2 & 0 & 0 & 5 & 8 & 0 & 8 & 0 & 4 & 0 \\ 0 & 3 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 9 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 7 & 0 & 0 & 6 & 5 & 7 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 & 2 & 0 & 5 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 4 & 9 & 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 0 & 2 \end{pmatrix}$$

we obtain the matrix X

$$X = \begin{pmatrix} 0 & 0 & 5 & 0 & 0 & 9 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 5 & 0 & 0 & 8 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 & 0 & 5 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 4 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

as the closest stable, with minimal distance $\tau_* = 10$. ○

Schur stabilization of $\alpha E, \ \alpha > 0$

[18] deals with the problem of finding a (locally) closest stable matrix in Frobenius norm. One example considers stabilizing the matrix αE, where E is the matrix of ones, and $\alpha > 0$. Here we are interested in the same problem, just working with the L_∞ norm.

Let $d \geqslant 2$ be a dimension of E. For $\alpha \leqslant \frac{1}{d}$ the matrix αE is already stable per se. So, the interesting case is when $\alpha > \frac{1}{d}$.

First, we analyse the $d = 2$ case, as in [18]. The solutions obtained there are also the solutions in L_∞ norm. However, using Algorithm SCS we come to a new solution which will prove useful in generalizing the result for higher dimensions.

For $\alpha \in [\frac{1}{2}, 1)$ the solution is also the matrix

$$X_\alpha = \begin{pmatrix} \alpha & 1-\alpha \\ \alpha & 1-\alpha \end{pmatrix}.$$

Note that X_α is the closest stable matrix to αE in L_∞ norm, but not in the Frobenius norm.

For $\alpha = 1$ Algorithm SCS finds the matrix

$$Y = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$$

as a solution. Moreover, the matrix

$$Y_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

is a solution too.

Matrix Y is also closest stable to αE for $\alpha > 1$. Additionally,

$$Y_\alpha = \begin{pmatrix} 1 & \alpha \\ 1 & 0 \end{pmatrix}$$

can also be regarded as a solution in this case.

Before generalizing these observations, we will give some illustrative examples for the case of $d = 4$. As said before, we are interested in case when $\alpha > \frac{1}{4}$. For $\alpha \in \{0.3, 0.4, 0.8\}$ we have following solutions:

$$X_{0.3} = \begin{pmatrix} 0.3 & 0.3 & 0.3 & 0.1 \\ 0.3 & 0.3 & 0.3 & 0.1 \\ 0.3 & 0.3 & 0.3 & 0.1 \\ 0.3 & 0.3 & 0.3 & 0.1 \end{pmatrix}, \quad X_{0.4} = \begin{pmatrix} 0.4 & 0.4 & 0.2 & 0 \\ 0.4 & 0.4 & 0.2 & 0 \\ 0.4 & 0.4 & 0.2 & 0 \\ 0.4 & 0.4 & 0.2 & 0 \end{pmatrix},$$
$$X_{0.8} = \begin{pmatrix} 0.8 & 0.2 & 0 & 0 \\ 0.8 & 0.2 & 0 & 0 \\ 0.8 & 0.2 & 0 & 0 \\ 0.8 & 0.2 & 0 & 0 \end{pmatrix}.$$

For $\alpha \geqslant 1$ the algorithm returns

$$Y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

However, the matrix

$$Y_\alpha = \begin{pmatrix} 1 & \alpha & \alpha & \alpha \\ 0 & 1 & \alpha & \alpha \\ 0 & 0 & 1 & \alpha \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

is also a solution, and a more representative one, since it requires less changes of entries.

Motivated by previous discussion, we can make a conjecture for the general case. When $\alpha \in (\frac{1}{d}, 1)$, we conjecture that the closest Schur stable matrix to αE in the the L_∞ norm is the matrix X_α, obtained by d times stacking the vector $\boldsymbol{v}_\alpha$, where

$$\boldsymbol{v}_\alpha = (\alpha, \dots, \alpha, 1 - \alpha M, 0, \dots, 0).$$

Here, the entries α are repeated M times, where M is the highest integer such that $\alpha M < 1$. For $\alpha > 1$, we conjecture that the upper triangular matrix having the form of Y_α is a solution. The numeric experiments done in higher dimensions fully support our conjectures, although we still lack the formal confirmation. Even so, the assumed solutions in L_∞ norm can be used as a starting point in search of the solutions in Frobenius norm.

4.2 Applications to graphs

Optimizing the spectral radius of a graph

A directed graph[2] $G = (V, E)$ with vertex set V and edge set E, can be represented by a (0,1)-matrix A_G, called *adjacency matrix*, defined by

$$a_{ij}^G = \begin{cases} 1, & (i,j) \in E \\ 0, & (i,j) \notin E. \end{cases}$$

The spectral radius of a directed graph G is the spectral radius of its adjacency matrix A_G. The problem of optimizing the spectral radius under some restrictions on the graph is well known in the literature [16, 22, 44, 45, 46, 47] Some of them deal with product families of adjacency matrices and hence can be solved by the greedy method. One example of such application is finding the maximal spectral radius of a directed graph with d vertices and prescribed numbers of outgoing edges $n_1, \dots, n_d$. For this problem, all the uncertainty sets $\mathcal{F}_i$ are polyhedral and are given by the systems of inequalities:

$$\mathcal{F}_i \; = \; \left\{ x \, \in \, \mathbb{R}^d \;\middle|\; \sum_{k=1}^{d} x_k \, \le \, n_i \, , \, 0 \leqslant x_k \leqslant 1, \, k = 1, \dots, d \right\}. \tag{4.2}$$

The minimal and maximal spectral radii are both attained at extreme points of the uncertainty sets, i.e., precisely when the the ith row has n_i ones and all other entries are zeros. If we need to minimize the spectral radius, we define $\mathcal{F}_i$ in the same way, with the only one difference: $\sum_{k=1}^{d} x_k \, \geqslant \, n_i$.

Performing the greedy method we solve in each iteration an LP problem $(\boldsymbol{a}_i, \boldsymbol{v}_k) \to \max, \; \boldsymbol{a}_i \in \mathcal{F}_i, \; i = 1, \dots, d$, with the sets $\mathcal{F}_i$ defined in (4.2). In fact, this LP problem is solved explicitly: the (0,1)-vector $\boldsymbol{a}_i$ has ones precisely at positions of the n_i maximal entries of the vector $\boldsymbol{v}_k$, while all other components of $\boldsymbol{a}_i$ are zeros.

Thus, in each iteration, instead of solving d LP problems, which can get computationally demanding even for not so big d, we just need to deal with finding

[2] In this work we deal only with directed graphs. Therefore, we will always refer to them shortly as graphs. Furthermore, we deal only with the graphs containing no self-loops.

the corresponding number of highest entries of the eigenvector $\boldsymbol{v}_k$. In Table 4.4 we present the numerical results for applying the selective greedy method to this problem, where the number of iteration and the time required for the algorithm to finish are shown.

d	500	1500	3000	5000
#	3	3	3	3
t	0.16s	0.83s	3.08s	6.62s

Table 4.4: Row sums are randomly selected integers from the segment $[0.1, 0.75] \times d$

Example 4.5 We apply the greedy method to find the maximal spectral radius of a graph with 7 vertices and with the numbers of incoming edges $(n_1, \ldots, n_7) = (3, 2, 3, 2, 4, 1, 1)$. The algorithm finds the optimal graph with the adjacency matrix:

$$A_G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

with $\rho(A) = 3.21432$.

In terms of application, knowing the optimal matrix A_G can tell us which vertices we should prioritise, i.e. supply first, in order to maximize the effect. In our example it is the fifth vertex. ◦

Closest acyclic subgraph and MAS

We denote the set of all (0,1)-matrices with $\mathbb{B}$. Optimizing the spectral radius on a ball of (0,1)-matrices $\mathcal{B}_\varepsilon^{(0,1)}$,

$$\mathcal{B}_\varepsilon^{(0,1)} = \mathcal{B}_\varepsilon^{+} \cap \mathbb{B},$$

can be used for finding the closest (in L_∞ norm) acyclic subgraph to a given graph. By knowing the spectral radius of the graph's adjacency matrix, one can easily see if it contains cycles. Moreover, the following theorem holds:

Theorem 4.6 [35] *Let A_Γ be adjacency matrix of a directed graph Γ. The following statements are equivalent:*

1° *A_Γ is strongly Schur stable;*
2° *$\rho(A_\Gamma) = 0$;*
3° *the graph Γ is acyclic and all diagonal elements of A_Γ are zero;*
4° *there exists a permutation matrix P such that $P^T A_\Gamma P$ is upper-triangular and has zero elements on the main diagonal;*
5° *Depth First Search algorithm applied to the graph Γ returns a spanning tree with no back edge.*

For more details about Depth First Search (DFS) algorithm we refer the reader to [48]. In short, a DFS algorithm applied to an acyclic graph will rearrange its vertices, outputting a spanning tree with no edges directed from a vertex back to its ancestor.

We now formulate the problem of finding the closest acyclic graph. For a given graph G that contains cycles, we are interested in finding its subgraph Γ, such that:

$$\begin{cases} \|A_G - A_\Gamma\|_\infty \to \min \\ \rho(A_\Gamma) = 0, \end{cases} \tag{4.3}$$

where A_G and A_Γ are the adjacency matrices of graphs G and Γ, respectively.

The L_∞ norm is a good choice for the problem (4.3), since it gives us the information on:

1) the required number of edges one needs to cut from at least one vertex of the graph G to render it acyclic;
2) the upper bound on the number of edges one should cut from each vertex to render the graph G acyclic.

In other words, if τ_* is the optimal value of the problem (4.3), then one should cut exactly τ_* edges from at least one vertex of G, and one needs not to cut more than τ_* edges from each vertex.

Related to the poblem (4.3) is the issue of making a graph G acyclic, so that the obtained subgraph has maximal number of edges. This is the *maximal acyclic subgraph (MAS) problem.* It is an NP-hard problem and there exists a vast body of literature dedicated to it [29, 30, 31, 32, 33]. One can only hope to get, in polynomial time, a good approximation of the maximal acyclic subgraph. Our solution to the problem of finding the closest acyclic subgraph may also prove to give a good approximation of the MAS.

Before setting out to describe the procedure for solving the (4.3), we give the following result:

Proposition 4.7 [45] *Let r and R be, respectively, a minimum and maximum row sum of the non-negative matrix A. Then*

$$r \leqslant \rho(A) \leqslant R.$$

□

From Theorem 4.6 and Proposition 4.7 we conclude that the adjacency matrix of an acyclic graph must contain a row of zeros. Further, we can conclude that the distance between a graph G and its closest acyclic subgraph is at least r. From this point we develop our procedure for solving (4.3), described below.

`Alg. ACY: Finding the closest acyclic subgraph`

`Step 0.` Let A_G be adjacency matrix of a graph G containing cycles, and let r and R be, respectively, a minimum and maximum row sums of matrix A_G. Take $k = \frac{R-r}{2}$.

`Step 1.` Do the greedy procedure for minimizing spectral radius on the ball of (0,1)-matrices $\mathcal{B}_k^{(0,1)}$.

We can also give the explicit solution to the LP problems occurring in greedy procedure when applied to the ball $\mathcal{B}_k^{(0,1)}$. Let X' be a (0,1)-matrix obtained in some iteration with $\boldsymbol{v} \geqslant 0$ as its leading eigenvector. We rearrange positive entries of $\boldsymbol{v}$: $v_{j_1} \geqslant \cdots \geqslant v_{j_m}$, where $\mathcal{S}$ is the support of $\boldsymbol{v}$, and $|\mathcal{S}| = m$. In the next iteration of the greedy procedure we construct the (0,1)-matrix $X = (x_{ij})$ as follows:

For each $i \in \mathcal{S}$ we have LP problem

$$(\boldsymbol{x}_i, \boldsymbol{v}) \to \min, \quad \boldsymbol{x}_i \in \mathcal{F}_i, \tag{4.4}$$

which we solve by setting the first κ non-zero entries x'_{ij_k} of $\boldsymbol{x}_i$ equal to zero, where

$$\kappa = \min\Big\{k,\ \sum_{k=1}^{m} x'_{ij_k}\Big\}.$$

`Step 2.` Doing a bisection in $k \in [r, R]$, repeat `Step 1`, until a value $\tilde{k}$ is reached, for which matrix $\tilde{X}$ is obtained, with $\rho(\tilde{X}) \leqslant \alpha$. Usually, α is chosen to be $1 \leqslant \alpha \leqslant 3$.

`Step 3.` If $\rho(\tilde{X}) > 0$, then, increasing $\tilde{k}$ by one repeat `Step 1`, until a matrix Y with $\rho(Y) = 0$ is obtained. Alternatively, for $\rho(\tilde{X}) = 0$, repeat `Step 1`, decreasing $\tilde{k}$ by one, until the matrix with positive spectral radius is obtained. Take the last matrix Y with $\rho(Y) = 0$. The distance $k_* = \|A_G - Y\|_\infty$ is the optimal value.

We can now take Y as an optimal solution to the problem (4.3). However, in the light of the MAS problem, Y might not present the best approximation, since it will preserve less than a half of starting graph's edges[3] (see Tables 4.6 and 4.7 of the numerical results). To improve upon this, proceed to the

`Step 4.` Run DFS algorithm on the acyclic graph generated by the matrix Y to obtain a spanning tree (Theorem 4.6). For each vertex of the spanning tree, restore all edges present in the starting graph G, so that no cycle is created. Take the newly obtained graph Γ as the optimal solution.

Alternatively, one can find a permutation P and bring the matrix Y to an upper triangular form $P^T Y P$. We then restore all ones present in the original matrix A_G, without breaking the upper triangular structure. More formally put, we permute the indices $\{1, \ldots, d\} \to \{i_1, \ldots, i_d\}$ (so that Y becomes upper triangular), and form the matrix Y' with entries

[3] Dividing the adjacency matrix into two triangular halves and choosing the one with greater number of edges will produce an acyclic graph.

$$y'_{i_k j_l} = \begin{cases} a^G_{i_k j_l}, & i_k < j_l \\ 0, & i_k \geqslant j_l. \end{cases}$$

We put $Y' = A_\Gamma$, and take the corresponding graph Γ as the optimal solution. This graph will be much better approximate of the MAS than the matrix obtained in `Step 3` (see Tables 4.8 and 4.9 of the numerical results). ◇

Numerical results

Algorithm ACY can be used to give answer to both the closest acyclic graph and MAS problems. If we stop with `Step 3`, we can obtain a distance to the closest acyclic graph in relatively short time, as can be seen from the tables below. Here, Δ represents a fraction of the edges preserved in the solution graph Γ, compared to the starting graph G.

9-51% sparsity

d	50	250	500	1000
t	0.5s	4.23s	20.13s	299.25s
Δ	0.508	0.491	0.498	0.494

26-95% sparsity

d	50	250	500	1000
t	0.65s	5.29s	30.51s	381.52s
Δ	0.448	0.453	0.46	0.461

Table 4.5: Numerical results for Alg. ACY, reduced version

%	3 - 21	9 - 51	16 - 75	25 - 95
t	15.52s	41.11s	47.61s	48.17s
Δ	0.52	0.499	0.484	0.462

Table 4.6: Alg. ACY, reduced version, varying the sparsity, $d = 500$

However, the obtained graph may be unsatisfactory, especially in the light of the MAS problem. Implementing the full Algorithm ACY improves upon this, but it prolongs the computational time. This is because DFS algorithms may have, in the worst-case scenario, computational time of $O(n^2)$, where n is number of edges [48]. Below, we report the results for implementation of the full Algorithm ACY.

9-51% sparsity

d	50	250	500	1000
t	0.53s	4.7s	37.1s	527.79s
Δ	0.64	0.62	0.619	0.617

26-95% sparsity

d	50	250	500	1000
t	0.66s	5.73s	50.42s	674.27s
Δ	0.598	0.592	0.593	0.593

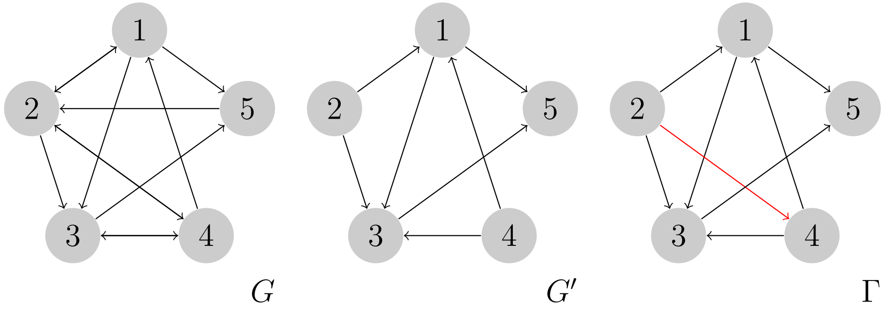
Table 4.7: Numerical results for Alg. ACY, full version

%	3 - 21	9 - 51	16 - 75	25 - 95
t	27.16s	70.69s	81.02s	88.78s
Δ	0.639	0.619	0.608	0.594

Table 4.8: Alg. ACY, reduced version, varying the sparsity, $d = 500$

To the best of our knowledge, the most effective algorithms found in literature for approximating MAS have $\Delta = 0.5 + \Omega(\frac{\delta}{\log d})$, where $0.5 + \delta$ is the fraction of the original edges contained in MAS [29, 31]. The numerical results presented here show that our approach outputs graphs with $\Delta \approx 0.6$, which seems to remain the same even for the large dimensions or big density. We can conjecture that our algorithm gives $\Delta = 0.5 + \Omega(\frac{1}{\log d})$ (or maybe even $\Delta = 0.5 + \Omega(\frac{\delta}{\log d})$). However, checking this conjecture will require numerical experiments on a very big scale, or theoretical confirmation. In any case, our procedure could prove useful for constructing more effective algorithms for approximating MAS. We also remark that our procedure can be made even faster by using an improved DFS algorithm.

Example 4.8 In the figure below, we apply Algorithm ACY to a graph G containing cycles. First we obtain its closest acyclic subgraph G' and then an approximation to its MAS, Γ.



Actually Γ is exactly maximum acyclic subgraph to G; restoring any of the edges in Γ creates a cycle. ○

Example 4.9 For a graph G containing cycles, with adjacency matrix

$$A_G = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Algorithm ACY first returns graph G' with

$$A_{G'} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

as the closest acyclic subgraph with $\tau_* = 2$. Continuing further, it finishes with the graph Γ,

$$A_\Gamma = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

as the graph approximating MAS of G. For the graph G', we have $\Delta(G') = 0.578$, while the graph Γ saves $\Delta(\Gamma) = 0.711$ of the original edges. Moreover, if we manually add back to Γ any of the edge from the original graph G, we create a cycle. This suggests that Γ might actually be the exact maximum acyclic subgraph to G. ∘

Chapter 5

Metzler product families

The second part of the thesis is concerned with Metzler matrices, i.e. real matrices with non-negative off-diagonal entries. As already mentioned, they are an important mathematical tool and found quite often in applications [3, 4, 5, 8, 9, 10]. Our main goal of this chapter is to generalize the selective greedy method and use it to optimize spectral abscissa on Metzler product families.

Definition 5.1 *A Metzler matrix is* strict *if it has a strictly negative diagonal entry. A Metzler matrix is* full *if it has positive off-diagonal elements. A Metzler matrix is* irreducible *if it does not have a non-trivial invariant coordinate subspace.*

By $\mathcal{M}$ we denote the set of all the Metzler matrices, and by $\mathcal{M}_i$ the sets of vectors

$$\mathcal{M}_i = \{\ \boldsymbol{a} \in \mathbb{R}^d \mid a_{j \neq i} \geqslant 0\ \}, \qquad i = 1, \ldots, d.$$

Definition 5.2 *A family $\mathcal{F}$ of Metzler matrices is a* Metzler product family *if there exist compact sets $\mathcal{F}_i \subset \mathcal{M}_i$, $i = 1, \ldots, d$, such that $\mathcal{F}$ consists of all possible matrices with i-th row from $\mathcal{F}_i$, for every $i = 1, \ldots, d$.*

We stated in the introduction that we change focus from spectral radius ρ to spectral abscissa η for Metzler matrices. This is because the spectral radius is no longer a Perron eigenvalue of a Metzler matrix, which is seen from

Theorem 5.3 ([14]) *Let A be a full (or at least irreducible) Metzler matrix. Then, there exist an eigenvalue r such that:*
1° $r \in \mathbb{R}$;
2° $r > \mathrm{Re}\lambda$*, for every eigenvalue* $\lambda \neq r$;
3° *r is a simple eigenvalue;*
4° *r can be associated with a strictly positive (right) eigenvector $\boldsymbol{v}$;*
5° *$\boldsymbol{v}$ is a simple eigenvector.* □

Theorem 5.4 ([14]) *Let A be a Metzler matrix. Then, there exist an eigenvalue r such that:*
1° $r \in \mathbb{R}$;
2° $r \geqslant \mathrm{Re}\lambda$*, for every eigenvalue λ;*
3° *r can be associated with a non-negative (right) eigenvector $\boldsymbol{v}$.* □

Definition 5.5 *Let A be a (full/irreducible) Metzler matrix. An eigenvalue r characterized by Theorem* (5.3) 5.4 *is called* Perron (or leading) eigenvalue*. Its corresponding eigenvector $\boldsymbol{v}$ is called* Perron (or leading) eigenvector*.*

Recalling Definition 1.4 for spectral abscissa, we conclude

Proposition 5.6 *For a Metzler matrix, its spectral abscissa is its Perron eigenvalue.* □

Unlike for the non-negative case, the Perron eigenvalue of a Metzler matrix can take on negative values. Of course, for non-negative matrices spectral radius and spectral abscissa coincide.

Proposition 5.7 *If A is a non-negative matrix, then*

$$\rho(A) = \eta(A).$$

□

Another important thing to point out is the behaviour of the Perron eigenvector. For (strict) Metzler matrices it remains non-negative, as with the non-negative matrices. Moreover, for the full Metlzer case it behaves the same way as for the positive matrices: it is strictly positive and unique.

A property of a strict Metzler matrix, which we will amply exploit, is that it can be translated to a non-negative matrix: i.e. if A is a strict Metzler matrix, then there exists $h > 0$ such that $A + hI$ is non-negative. Having this in mind, along with the fact that the translation by identity matrix does not change the set of eigenvectors, we have from Proposition 5.7

Lemma 5.8 *If A is strict Metzler, then*

$$\rho(A + hI) = \eta(A) + h,$$

for any $h > 0$ such that $A + hI$ is non-negative. □

5.1 Greedy method for full Metzler matrices

We saw that the Perron eigenvector for both the positive and full Metzler matrices keeps the same properties. This is why it can make sense to use the non-selective greedy method presented in Section 2.1 (Algorithm GP) for full Metzler matrices. The goal of this section is to formally show that the greedy method can be safely and effectively used for optimizing the spectral abscissa on the full Metzler product families. We must remark that selective greedy method, as given in Section 3.3, is inapplicable to general Metzler matrices without modifications. We will address and resolve this issue in the next section. As with the positive and non-negative product families, the results of this section are the extension of the results from [21].

It is known that spectral radius is monotone on the set of non-negative matrices [27]. We now prove that the same holds for the spectral abscissa on the set of Metzler matrices, that is:

Lemma 5.9 *Let A and B be two Metzler matrices such that $B \geqslant A$. Then $\eta(B) \geqslant \eta(A)$.*

`Proof.` Let $h > 0$ be such that both $A+hI$ and $B+hI$ are non-negative. Since B is entrywise bigger than A we have $B+hI \geqslant A+hI$. By the monotonicity of spectral radius $\rho(B+hI) \geqslant \rho(A+hI)$ holds. Using Lemma 5.8, we obtain $\eta(B) \geqslant \eta(A)$. □

Lemma 5.10 *Let A be a Metzler matrix, $\boldsymbol{u} \geqslant 0$ a vector, and $\lambda \geqslant 0$ a real number. Then $A\boldsymbol{u} \geqslant \lambda\boldsymbol{u}$ implies that $\eta(A) \geqslant \lambda$. If for a strictly positive vector $\boldsymbol{v}$ we have $A\boldsymbol{v} \leqslant \lambda\boldsymbol{v}$, then $\eta(A) \leqslant \lambda$.*

`Proof.` Let $A\boldsymbol{u} \geqslant \lambda\boldsymbol{u}$. Since A is Metzler, there exists $h \geqslant 0$ such that $A+hI \geqslant 0$. We have $(A+hI)\boldsymbol{u} \geqslant (\lambda+h)\boldsymbol{u}$, and since the analogous of this Lemma for non-negative matrices and spectral radii holds, we obtain $\rho(A+hI) \geqslant \lambda+h$, and therefore $\eta(A) \geqslant \lambda$ (Lemma 5.8). The proof of the second statement follows by the same reasoning. □

Corollary 5.11 *Let A be a Metzler matrix, $\boldsymbol{u} \geqslant 0$ a vector, and $\lambda \geqslant 0$ a real number. Then $A\boldsymbol{u} > \lambda\boldsymbol{u}$ implies that $\eta(A) > \lambda$. If for a strictly positive vector $\boldsymbol{v}$ we have $A\boldsymbol{v} < \lambda\boldsymbol{v}$, then $\eta(A) < \lambda$.*

`Proof.` Let $A\boldsymbol{u} > \lambda\boldsymbol{u}$. The statement $\eta(A) < \lambda$ is in direct contradiction with Lemma 5.10 and $\eta(A) = \lambda$ would imply that λ is an eigenvalue of matrix A, which is not true. Hence, $\eta(A) > \lambda$ has to hold. The proof of the second statement follows the same reasoning. □

Since the behaviour of the leading eigenvector remains unchanged, we can extend the definitions of minimality/maximality in each row as given by Definition 3.8 (2.5) to account for (full) Metzler matrices.

The following Proposition stems directly from Lemma 5.10.

Proposition 5.12 *Let matrix A belong to a Metzler product family $\mathcal{F}$ and $\boldsymbol{v} \geqslant 0$ be its leading eigenvector. Then*
1) if $\boldsymbol{v} > 0$ and A is maximal in each row with the respect to $\boldsymbol{v}$, then $\eta(A) = \max\limits_{X\in\mathcal{F}} \eta(X)$.
2) if A is minimal in each row with the respect to $\boldsymbol{v}$, then $\eta(A) = \min\limits_{X\in\mathcal{F}} \eta(X)$. □

Proposition 5.13 *If a Metzler matrix $A \in \mathcal{F}$ is full, then it has the maximal spectral abscissa in $\mathcal{F}$ if and only if A is maximal in each row with the respect to its (unique) leading eigenvector. The same is true for minimization.*

`Proof. Maximization.` Let A be maximal in each row w.t.r. to $\boldsymbol{v}$. Since for a full Metzler matrix its leading eigenvector is strictly positive, we can apply 1) of Proposition 5.12, and therefore the maximal spectral abscissa is indeed achieved for matrix A.

Conversely, assume that A maximizes the spectral abscissa on $\mathcal{F}$, but it is not maximal w.t.r. to its leading eigenvector $\boldsymbol{v}$ in each row. Construct matrix $A' \in \mathcal{F}$ by changing all the necessary rows of the matrix A, so that A' becomes maximal w.t.r. to $\boldsymbol{v}$ in each row. Thus we have $A'\boldsymbol{v} > A\boldsymbol{v} = \eta(A)\boldsymbol{v}$. This, in turn, results in $\eta(A') > \eta(A)$ (Corollary 5.11), which is incorrect. So, A has to be maximal in each row w.t.r. to its eigenvector.

`Minimization.` Suppose now that A is minimal in each row w.t.r. to $\boldsymbol{v}$. One direction of the equivalence is basically 2) of Proposition 5.12. The proof for the other direction goes the similar way as with the maximization, taking into account the strict positivity of the vector $\boldsymbol{v}$. □

Proposition 5.14 *For every Metzler product family there exists a matrix which is maximal (minimal) in each row with the respect to the one of its leading eigenvalues.*

`Proof.` Let $\varepsilon' > 0$ be such that the shifted product family $\mathcal{F}_{\varepsilon'} = \mathcal{F} + \varepsilon' E$ is full Metzler family, and let $A_{\varepsilon'} \in \mathcal{F}_{\varepsilon'}$ be the matrix with maximal spectral abscissa. Since this matrix is full, by Proposition 5.13 it has to be maximal in each row. For all $0 < \varepsilon \leqslant \varepsilon'$, we associate one such a matrix A_ε. By compactness, there exists a sequence $\{\varepsilon_k\}_{k\in\mathbb{N}}$ such that $\varepsilon_k \to 0$ as $k \to +\infty$. Hence, matrices A_{ε_k} converges to a matrix $A \in \mathcal{F}$ and their respective leading eigenvectors converge to a nonzero vector $\boldsymbol{v}$. By continuity, $\boldsymbol{v}$ is a leading eigenvector of A, and A is maximal in each row w.t.r. to $\boldsymbol{v}$. □

Theorem 5.15 *If $\mathcal{F}$ is a product family of full Metzler matrices, the greedy spectral simplex method terminates in finite time.*

`Proof.` Assume we use the greedy method for maximizing the spectral abscissa (the same goes for minimization). Proposition 5.14 guarantees the existence of a matrix maximal in each row, and Proposition 5.13 that this matrix is the optimal one. Starting from matrix $A_1 \in \mathcal{F}$ and iterating trough the algorithm we obtain the sequence of matrices $A_2, A_3, \ldots$ with the sequence of their respective spectral abscissas $\eta(A_2), \eta(A_3), \ldots$, which is increasing. Moreover, each ith row of the given matrices is some vertex of the polyhedron $\mathcal{F}_i$, so the number of total states is finite. Therefore, we arrive at our solution in finite number of iterations, increasing the spectral abscissa in each one, until we reach the optimal matrix. □

Remark. Since Perron eigenvectors/eigenvalues of full and irreducible Metlzer matrices have the same properties, we can safely use greedy method on Metzler matrices which are not full, provided we obtain an irreducible matrix in each iteration. Of course, this condition is too restrictive and quite difficult to achieve in applications.

5.2 Greedy method for sparse Metzler matrices

Comparing Theorems 5.3 and 5.4, we see that the Perron subspace of a general Metzler matrix might be spanned by more than one leading eigenvector. Because of this, the greedy method may choose a 'bad' leading eigenvalue from a Perron subspace and cycle, as with the sparse non-negative matrices. Using the selective greedy method on sparse Metzler matrices to avoid cycling seems like a logical move. However, it may happen that the power method applied to a strict Metzler matrix, starting with the initial vector $\boldsymbol{v}_0 = \boldsymbol{e}$, does not converge.

Example 5.16 We apply the power method to the matrix

$$A = \begin{pmatrix} -2 & 2 & 0 \\ 0 & -6 & 5 \\ 2 & 2 & -9 \end{pmatrix}.$$

Starting from the vector $\boldsymbol{v}_0 = \boldsymbol{e}$, and obtain the following sequence of vectors:

$\boldsymbol{v}_1 = (0.242,\ 0,\ -0.97)$
$\boldsymbol{v}_2 = (-0.025,\ -0.507,\ 0.862)$
$\boldsymbol{v}_3 = (-0.094,\ -0.649,\ -0.755)$
$\boldsymbol{v}_4 = (0.138,\ -0.694,\ 0.707)$
$\dots$

It is clear that the sequence of vectors $\boldsymbol{v}_k$ is not going to converge, because the sign pattern keeps changing with every iteration. ○

Having in mind that a Metzler matrix can be translated to a corresponding non-negative matrix, we propose *translative power method.*

Definition 5.17 *Let A be a Metzler matrix. A power method applied to a matrix $\tilde{A} = A + hI$, where $h \geqslant 0$ is the minimal number for which $\tilde{A}$ is non-negative is called* translative power method.

We extend Definition 3.13 to include Metlzer matrices and give the following

Proposition 5.18 *The translative power method $\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k$, $k \geq 0$, where A is Metzler matrix, applied to the initial vector $\boldsymbol{x}_0 = \boldsymbol{e}$ converges to the selected leading eigenvector.*

`Proof.` If A is non-negative, then translative power method is just the regular power method and we can use Corollary 3.17. Suppose now that A is strictly Metzler. Applying a translative power method on it, taking $\boldsymbol{e}$ as an initial vector, we obtain the selected leading eigenvector $\boldsymbol{v}$ of a non-negative matrix $\tilde{A} = A + hI$. This eigenvector $\boldsymbol{v} = \lim\limits_{\varepsilon \to 0} \boldsymbol{v}_\varepsilon$ is a limit of a sequence of leading eigenvectors of perturbed positive matrices $\tilde{A}_\varepsilon = \tilde{A} + \varepsilon E$. Since the leading eigenvectors $\boldsymbol{v}_\varepsilon$ for matrices $\tilde{A}_\varepsilon$ are also the leading eigenvectors for the perturbed Metzler matrices $A_\varepsilon = A + \varepsilon E$, we have, by the definition, that $\boldsymbol{v}$ is a selected leading eigenvector of the matrix A. □

Proposition 5.18 allows us to use the selective greedy method when dealing with Metzler matrices as well, having in mind that in this case we need to resort to the translative power method. Moreover,

Theorem 5.19 *The selective greedy method, equipped with translative power method and applied to Metzler matrices, does not cycle.*

`Proof.` Assume we have a product family $\mathcal{F}$ that contains strict Metzler matrices, since the case of non-negative families is already resolved by Theorem 3.16. Cycling of the selective greedy method on this family would also imply the cycling of the spectral greedy method on the family $\mathcal{F} + \varepsilon E$ of full Metzler matrices, which is, given Theorem 3.16, impossible. □

Remark. As discussed in Section 3.3, when implementing selective greedy method we compute the selected leading eigenvectors by applying the power method not on the non-negative matrices A_k, obtained iterating through the procedure, but on the matrices $A_k + I$ instead. For the same reasons, if we obtain a strict Metzler matrix A_k in some iteration, we will actually compute the selected leading eigenvector by applying power method ta a non-negative matrix $A_k+(h+1)I$, where $h = |\min_i a_{ii}^{(k)}|$.

Remark. If using the selective greedy method on Metlzer family $\mathcal{F}$ for maximization produces as a solution a matrix A_* having the leading eigenvector $\boldsymbol{v}$ with some zero entries, then we are dealing with reducible family. We can deal with this issue in completely analogous manner as described in Section 3.6. However, in this case, we need to add the matrix $\alpha P - \beta I$ instead, where $\alpha, \beta > 0$.

5.3 Numerical results

We test the selective greedy method, equipped with translative power method, on both full and sparse Metzler product families. The results of tests on full Metzler product families, as dimension d and the size of the uncertainty sets N vary, are given in Tables 5.1 and 5.2, for maximization and minimization, respectively.

avg. number of iterations

$N \setminus d$	25	250	500	2000
50	3	3.2	3	3.1
100	3.1	3.1	3	3
250	3.2	3.1	3.2	3

avg. running time

$N \setminus d$	25	250	500	2000
50	0.02s	0.24s	0.51s	2.19s
100	0.04s	0.45s	0.89s	5.12s
250	0.1s	1.05s	2.06s	382.12s

Table 5.1: Selective greedy method, maximization, full Metzler matrices

avg. number of iterations

$N \setminus d$	25	250	500	2000
50	3.2	3	3.1	3.1
100	3.2	3.3	3.1	3
250	3.3	3.2	3.2	3.1

avg. running time

$N \setminus d$	25	250	500	2000
50	0.02s	0.21s	0.46s	2.21s
100	0.04s	0.44s	0.89s	4.9s
250	0.1s	1.06s	2.05s	436.82s

Table 5.2: Selective greedy method, minimization, full Metzler matrices

In Tables 5.3 and 5.4 we report the behaviour of the selective greedy algorithm on sparse Metzler families. For each uncertainty set $\mathcal{F}_i$ the density parameter γ_i is randomly chosen from the interval $9 - 15\%$ and the elements of the set $\mathcal{F}_i$ are randomly generated in accordance to it.

avg. number of iterations

$N \setminus d$	25	250	500	2000
50	5.4	4.3	4.1	4
100	5.8	4.4	4.2	4.1
250	5.8	4.6	4.4	4.1

avg. running time

$N \setminus d$	25	250	500	2000
50	0.02s	0.3s	0.33s	2.89s
100	0.1s	0.6s	1.2s	6.02s
250	0.2s	1.54s	2.91s	167.89s

Table 5.3: Selective greedy method, maximization, sparse Metzler matrices

avg. number of iterations

$N \setminus d$	25	250	500	2000
50	7.8	4.5	4.3	4.1
100	8.6	4.5	4.3	4.2
250	9.6	4.7	4.5	4.4

avg. running time

$N \setminus d$	25	250	500	2000
50	1.35s	0.31s	0.63s	3.03s
100	0.58s	0.66s	1.26s	6.04s
250	0.32s	1.6s	2.99s	159.29s

Table 5.4: Selective greedy method, minimization, sparse Metzler matrices

Table 5.5 shows how the number of iterations and computing time vary as the density parameter is changed. The dimension is kept fixed at $d = 600$ and the cardinality of each product set at $|\mathcal{F}_i| = 200$, while we vary the interval from which γ_i takes value. We may conclude that the greedy method behaves on Metzler families as good as for the non-negative ones.

%	0 - 8	9 - 15	16 - 21	22 - 51	52 - 76	77 - 100
MAX	5	4.2	4.1	4	3.8	3.4
MIN	6.2	4.4	4.1	4	4	3.8

Table 5.3.1: Effects of sparsity, number of iterations

%	0 - 8	9 - 15	16 - 21	22 - 51	52 - 76	77 - 100
MAX	3.18s	2.71s	2.6s	2.54s	2.36s	2.14s
MIN	42.58s	2.69s	2.6s	2.55s	2.48s	2.32s

Table 5.3.2: Effects of sparsity, number of iterations

Chapter 6

Closest (un)stable Metzler matrix

The problem of finding the closest (un)stable Metzler matrix (i.e. *Hurwitz (de)stabilization*), bein hard in general [19, 34], allows efficient solutions when working in L_1, L_∞ and max- norms. These norms are given by:

$$\begin{aligned}
\|X\|_{\max} &= \max_{1\leqslant i,j\leqslant d} |x_{ij}|;\\
\|X\|_1 &= \max_{1\leqslant j\leqslant d} \textstyle\sum_{i=1}^{d} |x_{ij}|;\\
\|X\|_\infty &= \max_{1\leqslant i\leqslant d} \textstyle\sum_{j=1}^{d} |x_{ij}|.
\end{aligned}$$

As it can be easily seen $\|X\|_\infty = \|X^T\|_1$. Consequently, all the results for the L_∞ norm apply for the L_1 norm as well: we just need to take the transpose of the matrix. Having this in mind, we shall only develop results for L_∞ and max norms[1].

Dealing with Metzler matrices means that we are considering stability in the sense of Hurwitz. While Schur stability gets determined by the spectral radius, the Hurwitz stability depends on the sign of spectral abscissa.

Definition 6.1 *A Metzler matrix A is* strongly Hurwitz stable *if $\eta(A) < 0$, and* weakly Hurwitz stable *if $\eta(A) \leqslant 0$. Otherwise, if $\eta(A) > 0$ we say it is* strongly Hurwitz unstable*, and if $\eta(A) \geqslant 0$ we say it is* weakly Hurwitz unstable*.*

We denote by $\mathcal{H}$ the set of all weakly Hurwitz stable Metzler matrices, and by $\mathcal{H}_s$ the set of all strongly Hurwitz stable Metzler matrices.

Remark. When searching for the closest (un)stable Metzler matrix to the matrix A, the starting matrix need not necessarily be Metzler. It is easy to check that both the real matrix A and matrix A', with entries

$$a'_{ij} = \begin{cases} a_{ij}, & a_{ij} \geqslant 0 \text{ or } i = j \\ 0, & \text{otherwise} \end{cases}$$

have the same solution. Therefore, without the loss of generality, we shall always assume that our starting matrix is Metzler.

[1] In this chapter we will regard norms without a subscript as the L_∞ norm (i.e $\|\cdot\| \equiv \|\cdot\|_\infty$).

6.1 Hurwitz (de)stabilization in max-norm

Closest Hurwitz unstable Metzler matrix in the max-norm

For a given Hurwitz stable Metzler matrix A we consider the problem of finding its closest (weakly) Hurwitz unstable matrix X, with the respect to the max-norm. In other words, for a matrix $A \in \mathcal{H}_s$ find a matrix X that satisfies:

$$\begin{cases} \|X - A\|_{\max} \to \min \\ \eta(X) = 0, \; X \in \mathcal{M}. \end{cases}$$

The following set of results provides the answer to the posed problem. We start from

Lemma 6.2 [35] *A Metzler matrix A is Hurwitz stable if and only if it is invertible and the matrix $-A^{-1}$ is non-negative.* □

Lemma 6.3 *Let $A \in \mathcal{H}_s$ and $H \geqslant 0$, $H \neq 0$. Then*

$$\eta(A + \alpha H) < 0, \qquad 0 \leqslant \alpha < \frac{1}{\rho(-A^{-1}H)} \tag{6.1}$$

and $\eta\Big(A + \frac{H}{\rho(-A^{-1}H)}\Big) = 0$.

`Proof.` Define matrix $B = -A^{-1}H$. Since A is Hurwitz stable, from Lemma 6.2 we have $B \geqslant 0$. Define

$$W(\alpha) = -A - \alpha H = -A - \alpha(-AB) = -A(I - \alpha B).$$

From Lemma 4.1 and Lemma 6.2, $W(\alpha)$ is invertible for every $\alpha \in \Big[0, \frac{1}{\rho(B)}\Big)$. Moreover, for every α from the given interval, $W^{-1}(\alpha)$ is non-negative. Since $-W(\alpha) = A + \alpha H$ is Metzler, by Lemma 6.2, it is Hurwitz stable, so we have (6.1).

For the proof of the second part, we assume that matrix B is strictly positive. Then its leading eigenvector $\boldsymbol{v}$ is also strictly positive, and we have $-W\Big(\frac{1}{\rho(B)}\Big)\boldsymbol{v} = 0$. Hence, by the continuity of spectral abscissa on the set of Metzler matrices, we have $\eta\Big(A + \frac{H}{\rho(B)}\Big) = 0$. □

The following result is a direct consequence of Lemma 6.3:

Corollary 6.4 *Let $A \in \mathcal{H}_s$ and $H \geqslant 0$, $H \neq 0$. Then all $X \in \mathcal{M}$ that satisfy*

$$X < A + \frac{H}{\rho(-A^{-1}H)}$$

are stable. □

Applying Corollary 6.4 to the special case of the non-negative matrix $H = E$, where E is matrix of all ones, we arrive to the explicit formula for Hurwitz destabilization in the max-norm.

Theorem 6.5 *Let A be a Hurwitz stable Metzler matrix and $\boldsymbol{e}$ vector of all ones. Then all Metzler matrices X that satisfy*

$$X < A - \frac{E}{(A^{-1}\boldsymbol{e}, \boldsymbol{e})}$$

are Hurwitz stable. Moreover, matrix $A - \frac{E}{(A^{-1}\boldsymbol{e}, \boldsymbol{e})}$ is the closest Hurwitz unstable Metzler matrix with the distance to the starting matrix A equal to

$$\tau_* = \frac{1}{\sum\limits_{i,j} |a'_{ij}|},$$

where a'_{ij} are entries of the matrix A^{-1}.

`Proof.` The proof stems directly from the fact that

$$\rho(-A^{-1}E) = -(A^{-1}\boldsymbol{e}, \boldsymbol{e}).$$

□

Closest Hurwitz stable Metzler matrix in the max-norm

We now deal with the problem of Hurwitz stabilization in the max-norm. First, consider the following spectral abscissa minimization problem for a given matrix $A \in \mathcal{M}$ and parameter $\tau \geqslant 0$:

$$\min_{X \in \mathcal{M}} \{\eta(X) \mid \|X - A\|_{\max} \leqslant \tau\}. \tag{6.2}$$

Lemma 6.6 *The optimal solution of the problem* (6.2) *is a matrix $A(\tau) \in \mathcal{M}$ with the following entries:*

$$a_{ij}(\tau) = \begin{cases} a_{ii} - \tau, \ i = j \\ \max\{0, a_{ij} - \tau\}, \ i \neq j. \end{cases}$$

`Proof.` Clearly, a Metzler matrix $A(\tau)$ is feasible for (6.2). Moreover, for any other feasible solution X, we have $X \geqslant A(\tau)$. From the monotonicity of spectral abscissa we have that $A(\tau)$ is truly the optimal solution to our problem. □

Now, for a given Hurwitz unstable Metzler matrix A, consider the following minimization problem:

$$\min_{X \in \mathcal{M}} \|X - A\|_{\max}. \tag{6.3}$$

Denote by τ_* its optimal value.

Lemma 6.7 *τ_* is a unique root of the equation $\eta(A(\tau)) = 0$.*

`Proof.` The statement follows directly from Lemma 6.6 and the fact the function $\eta(A(\tau))$ is monotonically decreasing in τ. □

The matrix $A(\tau_*)$ is the closest Hurwitz stable Metzler matrix to a given matrix $A \in \mathcal{M}$ with $\eta(A) > 0$, and τ_* is the minimal distance. We now present the algorithm for computing the value τ_* and solving (6.3).

`Alg. HUS-max: Finding the closest Hurwitz stable Metzler matrix in max-norm`

`Step 1.` Sort all positive entries of matrix A in an increasing order. Check if the highest entry is on the main diagonal. If so, take $\tau_* = \max_i a_{ii}$ and finish the procedure. Else, continue to the

`Step 2.` Using the monotonicity of the spectral abscissa, find by bisection in the entry number the value τ_1, which is the largest between all positive entries a_{ij} and zero, such that $\eta(A(a_{ij})) > 0$. Find value τ_2, which is the smallest entry of A with $\eta(A(a_{ij})) < 0$.

`Step 3.` Form the matrix

$$H = \frac{A(\tau_1) - A(\tau_2)}{\tau_2 - \tau_1}.$$

`Step 4.` Compute

$$\tau_* = \tau_2 - \frac{1}{\rho(-A^{-1}(\tau_2)H)}.$$

◇

Theorem 6.8 *Algorithm HUS-max computes an optimal solution to the problem* (6.3).

`Proof.` First, assume that the highest entry is on the main diagonal. Then, the matrix $A(\max_i a_{ii})$ will be non-positive diagonal matrix such that $\eta(A(\max_i a_{ii})) = 0$. Since τ_*, by Lemma 6.7, is the unique root of the equation $\eta(A(\tau)) = 0$, we can put $\tau_* = \max_i a_{ii}$.

Now, assume that the highest entry is off-diagonal. Then, the matrix $A(\max_{ij} a_{ij})$ is negative diagonal. Therefore, we have

$$\eta\Big(A\Big(\max_{ij} a_{ij}\Big)\Big) < 0.$$

On the other hand, $\eta(A(0)) = \eta(A) > 0$. So, it is possible to find two values $\tau_1 < \tau_2$ from the set

$$\{0\} \cup \{a_{ij} \mid a_{ij} > 0\}$$

such that $\eta(A(\tau_1)) > 0$ and $\eta(A(\tau_2)) < 0$. $A(\tau)$ is linear in τ, and for every $\tau \in [\tau_1, \tau_2]$ we have

$$A(\tau) = A(\tau_2) + \frac{\tau_2 - \tau}{\tau_2 - \tau_1}(A(\tau_1) - A(\tau_2)) = A(\tau_2) + (\tau_2 - \tau)H.$$

Applying Lemma 6.3 for Metzler matrix $A(\tau_2)$ and non-negative matrix H concludes the proof. □

6.2 Hurwitz destabilization in L_∞ norm

Let A be a Metzler matrix, and let

$$\mathcal{B}_\varepsilon(A) = \{X \in \mathcal{M} \mid \|A - X\| \leqslant \varepsilon\}$$

denote an ε-ball of Metzler matrices around A. Optimization of the spectral abscissa on $\mathcal{B}_\varepsilon(A)$ is a crucial tool for the Hurwitz (de)stabilization. The selective greedy method developed in Chapter 5 can be applied on $\mathcal{B}_\varepsilon$, since it can be considered as a product set of the balls

$$\mathcal{B}_\varepsilon^{(i)} = \{\boldsymbol{x} \in \mathcal{M}_i \mid \|\boldsymbol{a}_i - \boldsymbol{x}\| \leqslant \varepsilon\}, \qquad i = 1, \ldots, d.$$

Let us first consider the problem of Schur destabilization of non-negative matrix in L_∞ norm: if A is a non-negative matrix with $\rho(A) < 1$, find a matrix X that satisfies

$$\begin{cases} \|X - A\| \to \min \\ \rho(X) = 1. \end{cases} \tag{6.4}$$

An explicit solution to (6.4) was given in [27]. An important point is that the solution is always a non-negative matrix, entrywise bigger than A.

Theorem 6.9 [27] *The optimal value τ_* of the problem* (6.4) *is the reciprocal of the biggest component of the vector $(I - A)^{-1}\boldsymbol{e}$. Let k be the index of that component. Then the optimal solution is the matrix*

$$X = A + \tau_* E_k.$$

□

We can also solve a more general problem: for a given $h > 0$ and non-negative matrix A with $\rho(A) < h$, find the closest matrix X having $\rho(X) = h$, i.e., find a solution to

$$\begin{cases} \|X - A\| \to \min \\ \rho(X) = h. \end{cases} \tag{6.5}$$

Here, as with the $h = 1$ case, the solution X is also non-negative and entrywise bigger than A. We can also prove a generalization of Theorem 6.9:

Theorem 6.10 *The optimal value τ_* of the problem* (6.5) *is the reciprocal of the biggest component of the vector $(hI - A)^{-1}\boldsymbol{e}$. Let k be the index of that component. Then the optimal solution is the matrix*

$$X = A + \tau_* E_k. \tag{6.6}$$

`Proof.` The optimal matrix X_* for (6.5) is also a solution to the maximization problem

$$\begin{cases} \rho(X) \ \to \max \\ \|X - A\| \leqslant \tau, \ X \geqslant A. \end{cases}$$

having the optimal value $\tau = \tau_*$. Let us characterize this matrix for arbitrary τ. From Proposition 5.12 and Proposition 5.14, applied to the non-negative product families and spectral radii, we can conclude that X is maximal in each row for the product family with the uncertainty sets:

$$\begin{aligned}\mathcal{B}_\tau^+(A) &= \mathcal{B}_\tau(A) \cap \{X \geqslant 0 \,|X \geqslant A\} \\ &= \{X \geqslant 0 \,|X \geqslant A, ((X-A)\boldsymbol{e}, \boldsymbol{e}_i) \leqslant \tau,\ i = 1, \dots, d\}.\end{aligned}$$

Conversely, every matrix X with $\rho(X) = h$ which is maximal in each row w.r.t. a strictly positive leading eigenvector, solves (6.5).

Any matrix $X \in \mathcal{B}_\tau^+(A)$ with the leading eigenvector $\boldsymbol{v}$ is optimal in the ith row if and only if the scalar product $(\boldsymbol{x}_i - \boldsymbol{a}_i, \boldsymbol{v})$ is maximal under the constraint $(\boldsymbol{x}_i - \boldsymbol{a}_i, \boldsymbol{e}) = \tau$. This maximum is equal to $r\tau$, where r is the maximal component of the leading eigenvector $\boldsymbol{v}$. Denote the index of this component by k. Then

$$\boldsymbol{x}_i - \boldsymbol{a}_i = \tau \boldsymbol{e}_k, \qquad i = 1, \dots, d.$$

Hence, if X is maximal in each row, we have

$$X = A + \tau \boldsymbol{e}\boldsymbol{e}_k^T = A + \tau E_k.$$

Furthermore, since each set $\mathcal{B}_\tau^+(\boldsymbol{a}_i)$ contains a strictly positive point, we have

$$v_i = (X\boldsymbol{v})_i = \max_{\boldsymbol{x} \in \mathcal{B}_\tau(\boldsymbol{a}_i)} (\boldsymbol{x}, \boldsymbol{v}) > 0.$$

Therefore, the leading eigenvector $\boldsymbol{v}$ is strictly positive so, by Proposition 5.12, the matrix X maximizes the spectral radius on the product family $\mathcal{B}_\tau^+(A)$.

Thus, the optimal matrix has the form (6.6) for some k, and $\|X - A\|= \tau_*$. It remains to find k for which the value of τ is minimal.

Since $\rho(A + \tau_* E_k) = h$, it follows that τ_* is the smallest positive root of the equation

$$\det(A - hI + \tau E_k) = 0. \tag{6.7}$$

Since $\rho(\frac{A}{h}) < 1$ we have $(hI - A)^{-1} = \frac{1}{h}(I - \frac{A}{h})^{-1} = \frac{1}{h}\sum_{j=0}^{\infty}(\frac{A}{h})^j \geqslant 0$. Multiplying equation (6.7) by $\det(-(hI - A)^{-1})$ we obtain

$$\det(I - \tau(hI - A)^{-1}E_k) = 0.$$

The matrix $\tau(hI - A)^{-1}E_k$ has only one nonzero column. This is the kth column, equal to $\tau(hI - A)^{-1}\boldsymbol{e}$. Hence,

$$\det(I - \tau(hI - A)^{-1}E_k) = 1 - \tau[(hI - A)^{-1}\boldsymbol{e}]_k.$$

We conclude that the minimal τ corresponds to the biggest component of this vector. □

We can now move to a Hurwitz destabilization, or more formally put: for a given Metzler matrix A with $\eta(A) < 0$, find a solution to

$$\begin{cases} \|X - A\| \to \min \\ \eta(X) = 0. \end{cases} \tag{6.8}$$

Lemma 6.11 *Let a matrix X be a solution to the problem (6.8). Then, X is Metzler and $X \geqslant A$.*

`Proof.` First, let us show that X is Metzler. Assume the contrary, that it has some negative off-diagonal entries. We consider a matrix X' with entries

$$x'_{ij} = \begin{cases} x_{ij}, & x_{ij} \geqslant 0 \text{ or } i = j \\ |x_{ij}|, & x_{ij} < 0 \text{ and } i \neq j. \end{cases}$$

We have $\|X' - A\| < \|X - A\|$ and $\eta(X') < 0$. Since X' is Metzler, there exists $h > 0$ such that $X'_h = X' + hI$ is non-negative. In addition, from Lemma 5.8, we obtain $\rho(X'_h) < h$.

Define the matrix $X_h = X + hI$ which contains off-diagonal negative entries. From the inequality $\|X_h^k\| < \|(X'_h)^k\|$, using Gelfand's formula for spectral radius [2], we arrive at $\rho(X_h) < \rho(X'_h)$. Since the largest real part of all the eigenvalues of X is equal to zero, we have

$$0 = \eta(X) \leqslant \rho(X).$$

Adding h on both sides of inequality and using $\rho(X) + h = \rho(X + hI)$, we obtain $h < \rho(X'_h)$, which is a contradiction. Therefore, matrix X must not contain negative off-diagonal entries, i.e. it has to be Metzler.

Now let us prove that $X \geqslant A$. Assume the converse, that there exist some entries x_{ij} of X such that $x_{ij} < a_{ij}$. Define matrix $\tilde{X}$ with entries

$$\tilde{x}_{ij} = \begin{cases} x_{ij}, & x_{ij} \geqslant a_{ij} \\ a_{ij}, & x_{ij} < a_{ij} \end{cases}$$

$\tilde{X}$ is Metzler, $\tilde{X} \geqslant A$ and $\tilde{X} \geqslant X$. By the monotonicity of spectral abscissa we have $\eta(\tilde{X}) \geqslant \eta(X)$. X is the closest Hurwitz unstable Metzler matrix, so $\|\tilde{X} - A\| \geqslant \|X - A\|$. This is impossible, since $\|\tilde{X} - A\| < \|X - A\|$ holds. Hence, X has to be entrywise bigger than A. □

Theorem 6.12 *The optimal value τ_* of the problem* (6.8) *is the reciprocal of the biggest component of the vector $-A^{-1}\boldsymbol{e}$. Let k be the index of that component. Then the optimal solution is the matrix*

$$X = A + \tau_* E_k. \tag{6.9}$$

`Proof.` Lemma 6.2 ensures invertibility of the matrix A. We also remark that a Hurwitz stable Metzler matrix is necessary a strict Metzler matrix: if it were non-negative, than the biggest real part of its eigenvalues would also be non-negative, which is not the case.

Now, let $h > 0$ be such that the matrix $\tilde{A} = A + hI$ is non-negative. We have $\rho(\tilde{A}) < h$. Using Theorem 6.10 we obtain the closest non-negative matrix to $\tilde{A}$. Denote it by $\tilde{X}$. We have $\rho(\tilde{X}) = h$ and

$$\tilde{X} = \tilde{A} + \tau_* E_k,$$

where τ_* is the reciprocal of the biggest component of the vector $(hI - \tilde{A})^{-1}\boldsymbol{e} = -A^{-1}\boldsymbol{e}$. Let $X = \tilde{X} - hI$. X is Metzler with $\eta(X) = 0$, and satisfies (6.9).

Now we need to check that X is really the closest Hurwitz unstable Metzler matrix for A. Assume that $Y \neq X$ is the true solution of (6.8). We have $\|Y - A\| < \|X - A\| = \tau_*$. Define $\tilde{Y} = Y + hI$. $\tilde{Y}$ is entrywise bigger than $\tilde{A}$, and $\rho(\tilde{Y}) = h$. Since $\tilde{X}$ is the closest non-negative matrix to $\tilde{A}$ having spectral radius equal to h, the following is true:

$$\|Y - A\| = \|\tilde{Y} - \tilde{A}\| \geqslant \|\tilde{X} - \tilde{A}\| = \tau_*,$$

which is a contradiction. Therefore, our X is really the optimal solution. □

Example 6.13 For a given Hurwitz stable Metzler matrix

$$A = \begin{pmatrix} -4 & 0 & 0 & 0 & 4 \\ 0 & -2 & 0 & 2 & 0 \\ 0 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & 3 & -9 \end{pmatrix}$$

with $\eta(A) = -1$, the biggest component of the vector $-A^{-1}\boldsymbol{e}$ is the third one, and it is equal to 2.5. Thus we get the matrix

$$X = \begin{pmatrix} -4 & 0 & 0.4 & 0 & 4 \\ 0 & -2 & 0.4 & 2 & 0 \\ 0 & 2 & -0.6 & 0 & 0 \\ 0 & 0 & 0.4 & -4 & 0 \\ 0 & 0 & 0.4 & 3 & -9 \end{pmatrix}$$

as the closest Hurwitz unstable. ○

6.3 Hurwitz stabilization in L_∞ norm

The problem of Schur stabilization for non-negative matrices in L_∞ norm was also considered in [27]: if A is a given non-negative matrix with $\rho(A) > 1$, find a non-negative matrix X that satisfies

$$\begin{cases} \|X - A\| \to \min : \ X \geqslant 0, \\ \rho(X) = 1. \end{cases} \tag{6.10}$$

Notice that, in contrast to the problem of Schur destabilization (6.4), here we have to impose the non-negativity condition on our solution.

The story is the same with the Hurwitz (de)stabilization of Metzler matrix. For Hurwitz destabilization (6.8), requesting for the solution to be Metzler is redundant, a fact confirmed by Lemma 6.11. However, this is not the case with the Hurwitz stabilization; here, we have to explicitly impose the restriction on our solution to be Metzler. So, our problem can be written in the following manner:

$$\begin{cases} \|X - A\| \to \min : \ X \in \mathcal{M}, \\ \eta(X) = 0. \end{cases} \tag{6.11}$$

To obtain the solution, we solve the related problem

$$\begin{cases} \eta(X) \to \min : \ \|X - A\| \leqslant \tau, \\ A \geqslant X, \ X \in \mathcal{M}. \end{cases}$$

and use the strategy that implements the bisection in τ, together with the greedy procedure for minimizing the spectral abscissa described in Chapter 5.

First, we describe how to compute matrix X explicitly in each iteration of the greedy method for minimization, thus finding the optimal solution on the ball of radius τ.

Fix some $\tau > 0$, and let X' be a matrix obtained in some iteration with $\boldsymbol{v} \geqslant 0$ as its leading eigenvector. We rearrange positive entries of $\boldsymbol{v}$: $v_{j_1} \geqslant \cdots \geqslant v_{j_m}$, where $\mathcal{S}$ is support of $\boldsymbol{v}$, and $|\mathcal{S}| = m$. In the next iteration of the greedy procedure we construct the matrix $X = (x_{ij})$:

For each $i \in \mathcal{S}$ we solve

$$(\boldsymbol{x}_i, \boldsymbol{v}) \to \min, \qquad \boldsymbol{x}_i \in \mathcal{F}_i$$

which we can rewrite as

$$\sum_{k=1}^{m} x_{ij_k} v_{j_k} \to \min : \begin{cases} \sum_{k=1}^{m} x_{ij_k} \geqslant -\tau + \sum_{k=1}^{m} a_{ij_k}, \\ x_{ij_k} \geqslant 0, \ i \neq j_k. \end{cases} \tag{6.12}$$

We can solve (6.12) explicitly:

$$x_{ij_k} = \begin{cases} 0, & k < l_i \\ -\tau + \sum_{s=1}^{l_i} a_{ij_s}, & k = l_i \\ a_{ij_k}, & k > l_i \end{cases} \tag{6.13}$$

where

$$l_i = \min\Big\{i, \ \min\Big\{l \in \mathcal{S} \Big| \sum_{s=1}^{l} a_{ij_s} > \tau\Big\}\Big\}. \tag{6.14}$$

For $i \in \mathcal{S}$, but $j \notin \mathcal{S}$, we take $x_{ij} = a_{ij}$; and if $i \notin \mathcal{S}$ we put $\boldsymbol{x}_i = \boldsymbol{x}'_i$.

`Alg. HUS: Computing the closest Hurwitz stable Metzler matrix in` L_∞ `norm`

`Step 0.` Take $\frac{\|A\|}{2}$ as the starting value for τ.

`Step 1.` Minimize the spectral abscissa on the ball of Metzler matrices $\mathcal{B}_\tau(A)$ by using the selective greedy method and applying (6.13). Let X be a matrix with the minimal spectral abscissa on $\mathcal{B}_\tau(A)$. Keep implementing the bisection on τ, until the ball $\mathcal{B}_\tau(A)$ for which $\eta_{\min} = \eta(X) < 0$ is obtained. Compute the leading eigenvector $\boldsymbol{v}$ of the matrix X and proceed to the

`Step 2.` Construct matrices $C = (c_{ij})$ and R, as follows. For each $i \in \mathcal{S}$

$$c_{ij_k} = \begin{cases} 0, & k < l_i \\ \sum_{s=1}^{l_i} a_{ij_s}, & k = l_i \\ a_{ij_k}, & k > l_i \end{cases}$$

where l_i is given by (6.14), while R is a boolean matrix having ones on positions (i, l_i) and zeros in all other places. If some of the indices i, j are not in the support, then we put $c_{ij} = x_{ij}$. We can write $X = C - \tau R$.

Denote by τ_* a potential optimal value of the problem (6.11). To determine this value we proceed to the next step.

`Step 3.` Since $\eta(X) < 0$, we have by Lemma 6.2 that X is invertible and

$$-X^{-1} = -(C - \tau R)^{-1} \geqslant 0.$$

Since $0 = \eta(C - \tau_* R) = \eta(C - \tau R + (\tau - \tau_*)R)$, it follows that $\det(-(C - \tau R) - (\tau - \tau_*)R) = 0$. From here we have

$$\det\Big(\frac{1}{\tau - \tau_*} I + (C - \tau R)^{-1} R\Big) = 0.$$

Matrix $-(C - \tau R)^{-1} R$ is non-negative and $\lambda = \frac{1}{\tau - \tau_*}$ is its (positive) leading eigenvalue.
We find the potential optimal value τ_* by computing the leading eigenvalue λ of the matrix $-(C - \tau R)^{-1} R$, and then calculating

$$\tau_* = \tau - \frac{1}{\lambda}.$$

`Step 4.` To check if τ_* is really optimal, start iterating through the greedy procedure on the ball $\mathcal{B}_{\tau_*}(A)$.

If during some iteration a matrix with negative spectral abscissa is obtained, τ_* is not the optiaml value. Stop the greedy procedure and return to `Step 1`, taking now τ_* as the starting value for τ.

Else, if we finish the greedy procedure obtaining the matrix X_* with minimal spectral abscissa $\eta(X_*) = 0$ on the ball $\mathcal{B}_{\tau_*}(A)$, we are done: τ_* is the optimal value for the problem (6.11), with X_* as the corresponding optimal solution. ◇

Remark. Contrary to Algorithm SCS for Schur stabilization, here we let the greedy procedure finish its course in Step 1, without interrupting it. We do this since numerical experiments showed that for Hurwitz stabilization this practice leads us to the solution much quicker, than when doing the interruptions.

Example 6.14 For a given Hurwitz unstable Metzler matrix

$$A = \begin{pmatrix} 3 & 0 & 2 & 1 & 4 \\ 7 & -4 & 6 & 5 & 7 \\ 3 & 4 & 2 & 3 & 0 \\ 2 & 1 & 1 & -1 & 8 \\ 8 & 0 & 0 & 4 & 9 \end{pmatrix}$$

with $\eta(A) > 0$, the described procedure computes the matrix

$$X = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 7 & -7 & 6 & 5 & 0 \\ 3 & 0 & -4 & 3 & 0 \\ 2 & 0 & 0 & -1 & 0 \\ 8 & 0 & 0 & 4 & -1 \end{pmatrix}$$

as the closest stable Metzler with $\tau_* = 10$. ○

Numerical results

In this subsection we report the numerical results for the implementation of Algorithm HUS, as the dimension of the starting matrix A is varied. By t_{full} we denote running time of the algorithm applied on a full starting matrix; t_{sparse} is the running time for sparse matrices, having the density parameter between $9 - 15\%$. The precision parameter of the power method was set to $\varepsilon = 10^{-7}$ in most of the experiments, except in some cases when it was required to set it to 10^{-8}, so the code could finish.

d	50	250	500	1000
t_{full}	0.4s	10.96s	49.22s	273.21s
t_{sparse}	0.31s	8.38s	12.29s	66.42s

Table 6.1: Numerical results for Hurwitz stabilization

The next table shows how the sparsity affects the computations. The dimension is kept fixed at $d = 850$ in all experiments, while the density parameter γ_i for each row of the starting matrix is randomly chosen from the given interval.

γ_i	3 - 8	9 - 15	16 - 21	22 - 51	52 - 76	77 - 100
t	69.44s	60.9s	127.98s	115.63s	152.12s	132.85s

Table 6.2: Effects of sparsity

The importance of the solution set

The following example illustrates the importance of imposing the set of allowable solutions for the stabilization problems.

Example 6.15 Observe the non-negative matrix

$$A = \begin{pmatrix} 1 & 9 \\ 6 & 0 \end{pmatrix}$$

with spectral abscissa (i.e. spectral radius) greater than one. If we want to find a closest *non-negative* matrix having spectral abscissa (i.e. spectral radius) equal to one, we have a Schur stabilization problem. Solving it, we obtain the matrix

$$X_* = \begin{pmatrix} 0 & 4.236 \\ 0.236 & 0 \end{pmatrix}$$

as the closest Schur stable, with $\tau_* = 5.764$. However if we expand our set of admissible solution to a set of Metzler matrices, we get

$$Y_* = \begin{pmatrix} -4.4 & 9 \\ 0.6 & 0 \end{pmatrix}$$

as the solution. In this case the distance to the starting matrix will be $\tau_* = 5.4$, which is smaller than for the closest non-negative matrix. ○

Remark In the previous example we found the closest Metzler matrix to a matrix A having $\eta(A) = 1$ by slightly modifying Step 4 of the Algorithm SCS: we used the greedy procedure on the balls of Metzler matrices $\mathcal{B}_\tau(A)$, instead on the balls of non-negative matrices $\mathcal{B}_\tau^+(A)$. Further, we changed spectral radius to spectral abscissa, and determined the indices l_i using the formula (6.14).

Chapter 7

Applications of selective greedy method to Metzler matrices

Checking the reliability of a given data

One of the most prominent uses of Metzler matrices is for mathematical modelling [3, 4, 35]. Numerous phenomena can be modelled by positive linear dynamical systems. In most simple case these systems have the form $\dot{\boldsymbol{x}} = A\boldsymbol{x}$, where $\boldsymbol{x} = \boldsymbol{x}(t)$ is a vector of time-dependent unknowns, and A is a time-independent Metzler matrix. The coefficients of A are determined from the gathered data. Often, researchers are interested in examining the stability of the model, which is reflected by the Hurwitz stability of matrix A. However, gathered data might contain some errors (i.e. due to the measurement imprecisions), which can lead us to the model with qualitative behaviour completely different than the real picture. To check if our data is reliable, we can optimize spectral abscissa on the ball $\mathcal{B}_\varepsilon(A)$.

For example, let us assume that the matrix of our model-system A is Hurwitz stable and that each entry contains an error not bigger than ε. To check if our model is reliable, we need to maximize the spectral abscissa on $\mathcal{B}_\varepsilon(A)$.

Let X_* be the optimal matrix. If $\eta(X_*) < 0$, we are safe and the qualitative behaviour of our model will agree with the real state of affair, even in the case of making the biggest predicted errors. On the other hand, if we obtain $\eta(X_*) > 0$, we cannot claim that our model reliably describes the phenomenon. In this case one needs to further refine our data gathering methods.

Similar reasoning can be applied if for our model we have $\eta(A) > 0$, and we want to check if the real system is unstable as well. However, in this case we would actually need to minimize spectral abscissa on $\mathcal{B}_\varepsilon$.

Stabilization of 2D postitive linear switching systems

Positive linear switching systems (LSS) are an important tool of mathematical modelling. They are a point of an extensive research with many applications [5, 6, 7]. Let $A_i,\ i = 1, \dots, N$ be a family of Metzler matrices. A positive LSS is given with:

$$\begin{array}{rcl} \dot{x}(t) & = & A_\sigma(t)x(t) \\ x(0) & = & x_0, \end{array} \tag{7.1}$$

where $x \in \mathbb{R}^d$ and $\sigma : \mathbb{R}_{\geqslant 0} \to \{1, \ldots, N\}$ is a piecewise constant switching signal.

A crucial question in the theory of LSS is the asymptotic stability under the arbitrary switching signal. The following theorem provides us with the necessary condition:

Theorem 7.1 [5] *If the positive LSS* (7.1) *is asymptotically stable (under the arbitrary switching), then all the matrices in the convex hull* $\mathrm{co}\{A_1, \ldots, A_N\}$ *are Hurwitz stable.* □

The converse of Theorem 7.1 is true only for the two-dimensional case [6]. Therefore, we can use Hurwitz stabilization on the unstable convex combinations to build an algorithm for the stabilization of the 2D positive LSS. By this we mean constructing a stable LSS from the original system, as shown below.

Assume that 2×2 Metzler matrices $A_i,\ i = 1, \ldots, N$ are Hurwitz stable. If some of them is not, find and replace it with its closest stable. Suppose that there exists a matrix in $\mathrm{co}\{A_1, \ldots, A_N\}$ that is not Hurwitz stable.

Let $A = \sum_{i=1}^N \alpha_i A_i$ be a convex combination in $\mathrm{co}\{A_1, \ldots, A_N\}$ with the largest spectral abscissa ($\sum_{i=1}^N \alpha_1 = 1,\ \alpha_i \in [0, 1],\ i = 1, \ldots, N$). Find its closest Hurwitz stable matrix A', and denote by τ_* the optimal distance.

We now need to decompose the matrix $A' = \sum_{i=1}^N \alpha_i A_i'$, while taking care about matrices A_i. We have

$$\tau_* = \|A - A'\| \leqslant \sum_{i=1}^N \alpha_i \|A_i - A_i'\| = \sum_{i=1}^N \alpha_i \tau_i. \tag{7.2}$$

Choose values τ_i and matrices $A_i' \in \mathcal{B}_{\tau_i}(A_i)$, so that (7.2) and $A' = \sum_{i=1}^N \alpha_i A_i'$ are satisfied.

Let A_* be a convex combination in $\mathrm{co}\{A_1', \ldots, A_N'\}$ with the largest spectral abscissa. If $\eta(A_*) < 0$, then we are done: the switching system built from matrices A_i' is asymptotically stable. Else, we should make a different choice of τ_i, or restart the procedure, but now starting with matrices A_i'.

7.1 Closest stable sign-matrix and its application to positive LSS

The notions of sign-matrices and sign-stability originated from the problems in ecology [10, 11], economy [38, 39] and chemistry [40]. Ever since, those concepts have been a point of interest in the mathematical literature [35, 36, 37]. Metzler sign-matrices are very useful for the dynamical system modelling. They come in handy if we do not posses quantitative data, but just the information on the sign of coefficients of the observed system. By analysing the sign-matrix of the system, we can discern its qualitative behaviour. As we will see later on, Metzler sign-matrices can be used as a tool for stability analysis of the linear switching systems.

Denote by $\mathcal{M}_{\text{sgn}}$ set of all real Metzler matrices with entries from $\{-1, 0, 1\}$. Hurwitz stable matrices from $\mathcal{M}_{\text{sgn}}$ have one peculiar property: replacing any entry by an arbitrary real number of the same sign does not influence the stability.

Example 7.2 The matrix $A \in \mathcal{M}_{\text{sgn}}$

$$A = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

is Hurwitz stable with $\eta(A) = -1$. Metzler matrix A' with the same sign pattern, given by

$$A' = \begin{pmatrix} -10^{-9} & 10^{9} & 0 & 0 & 0 \\ 0 & -10^{-9} & 0 & 0 & 10^{9} \\ 0 & 10^{9} & -10^{-9} & 0 & 0 \\ 10^{9} & 10^{9} & 0 & -10^{-9} & 10^{9} \\ 0 & 0 & 0 & 0 & -10^{-9} \end{pmatrix}$$

has $\eta(A') = -10^{-9}$, i.e., is also Hurwitz stable. It remains stable in spite of very big changes (order of 10^9) of all non-negative entries. ○

The same can be said for Schur stable matrices: replacing a positive entry of a zero spectral radius matrix by an arbitrary non-negative number will not change its spectral radius. We illustrate this by

Example 7.3 The (0,1)-matrix

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

is Schur stable with $\rho(A) = 0$. A non-negative matrix matrix A' with the same distribution of zero entries, given by

$$A' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 3 \\ 4 & 0 & 0 & 5 & 6 \\ 7 & 0 & 0 & 0 & 8 \\ 9 & 0 & 0 & 0 & 0 \end{pmatrix}$$

has also zero spectral radius, i.e., is also Schur stable. ○

The secret behind this behaviour can be discerned by examining the sign-matrices and their graphs.

Definition 7.4 *A matrix is called a* sign-matrix *if its entries take values from the set $\{-, 0, +\}$. If $-$ appears only on the main diagonal, we say it is* Matzler sign-matrix. *If it does not contain $-$, then it is* non-negative sign-matrix.

In analogy with the real Metzler and non-negative matrices, we can also consider Hurwitz and Schur stability of sign-matrices.

Definition 7.5 *Let M be a sign-matrix. We say that real matrix X belongs to a* qualitative class of a matrix M, *denoted by $\mathcal{Q}(M)$, if*

$$x_{ij} \begin{cases} < 0, & m_{ij} = - \\ = 0, & m_{ij} = 0 \\ > 0, & m_{ij} = +. \end{cases}$$

Definition 7.6 *A Metzler sign-matrix M is* strongly (weakly) Hurwitz stable *if all the matrices from the qualitative class $\mathcal{Q}(M)$ are strongly (weakly) Hurwitz stable. A non-negative sign-matrix M is* strongly (weakly) Schur stable *if all the matrices from the qualitative class $\mathcal{Q}(M)$ are strongly (weakly) Schur stable.*

Theorem 7.7 [35] *Let M be a Metzler sign-matrix, and* $\operatorname{sgn}(M)$ *be a real matrix given by*

$$m_{ij}^{\text{sgn}} = \begin{cases} -1, & m_{ij} = -, \\ 0, & m_{ij} = 0, \\ 1, & m_{ij} = +. \end{cases}$$

M is strongly (weakly) Hurwitz stable if and only if $\operatorname{sgn}(M)$ *is strongly (weakly) Hurwitz stable. Moreover, M is strongly Hurwitz stable if and only if* $\operatorname{sgn}(M) + I$ *is the adjacency matrix of an acyclic graph.*
If M is a non-negative sign-matrix, then M is strongly (weakly) Schur stable if and only if $\operatorname{sgn}(M)$ *is strongly (weakly) Schur stable. Moreover, M is strongly Schur stable if and only if* $\operatorname{sgn}(M)$ *is the adjacency matrix of an acyclic graph.* □

From Theorem 7.7 we see why the Hurwitz stable matrices from $\mathcal{M}_{\text{sgn}}$ do not change their stability, even as we change their non-zero entries.

The problem of Hurwitz stabilization of real Metzler matrices can be formulated for the Metzler sign-matrices as well. Theorem 7.7 gives a way to do so: for a given

Hurwitz unstable Metzler sign-matrix M, we need to find *closest Hurwitz stable Metzler sign-matrix* X. In other words, X should solve

$$\begin{cases} \|\mathrm{sgn}(M) - \mathrm{sgn}(X)\| \to \min \\ \eta(\mathrm{sgn}(X)) \leqslant 0. \end{cases} \tag{7.3}$$

The norm considered is the L_∞ norm.

Remark. Notice that in Hurwitz stabilization of sign-matrix we allow our optimal solution to have negative spectral abscissa as well. We do this since in some cases a solution with zero spectral abscissa does not exist.

Denote $\mathcal{B}_k^{\mathrm{sgn}} = \mathcal{B}_k \cap \mathcal{M}_{\mathrm{sgn}}$. Using minimization of spectral abscissa on $\mathcal{B}_k^{\mathrm{sgn}}$ we can present a simple procedure for solving problem (7.3).

`Algorithm HSM: Finding the closest Hurwitz stable sign-matrix`

Let M be a Hurwitz unstable sign-matrix. Set $k = \lfloor \frac{\|\mathrm{sgn}(M)\|}{2} \rfloor$. By X_k denote the sign-matrix such that $\eta_k = \eta(\mathrm{sgn}(X_k))$ is minimal on $\mathcal{B}_k^{\mathrm{sgn}}$.

Doing a bisection in k, minimize the spectral abscissa on the ball $\mathcal{B}_k^{\mathrm{sgn}}$. Do this until sign-matrices X_{k-1} and X_k are obtained, with the (minimal) spectral abscissas $\eta_{k-1} > 0$ and $\eta_k \leqslant 0$ on the balls $\mathcal{B}_{k-1}^{\mathrm{sgn}}$ and $\mathcal{B}_k^{\mathrm{sgn}}$, respectively.

We can finish the procedure by taking $k = k_*$ as the optimal value, and $X_k = X_*$ as the optimal solution of (7.3).

Additionally, we can run a DFS algorithm on matrix X_* if it has no cycles (we consider $-$ entries as no-edge). In this case we obtain a spanning tree, and then restore all the $+$ entries present in the original matrix M without creating cycles. The matrix obtained this way will still be the closest stable, and it will contain more $+$ entries, resembling the starting matrix even better. ◇

Example 7.8 For the unstable Metzler sign-matrix

$$M = \begin{pmatrix} 0 & + & + & + & 0 \\ + & + & 0 & + & + \\ + & + & 0 & 0 & + \\ + & 0 & 0 & - & + \\ 0 & 0 & + & + & + \end{pmatrix}$$

we find

$$M_* = \begin{pmatrix} 0 & 0 & 0 & + & 0 \\ + & - & 0 & + & + \\ + & 0 & - & 0 & + \\ 0 & 0 & 0 & - & 0 \\ 0 & 0 & 0 & + & 0 \end{pmatrix}$$

with $\eta(\mathrm{sgn}(M_*)) = 0$ as the closest sign-stable with optimal distance $k_* = 2$. ○

Example 7.9 For the unstable Metzler sign-matrix

$$M = \begin{pmatrix} - & + & 0 & 0 & + \\ + & 0 & 0 & + & + \\ + & 0 & 0 & + & 0 \\ + & + & + & 0 & 0 \\ 0 & + & + & 0 & - \end{pmatrix}$$

we find

$$M_* = \begin{pmatrix} - & 0 & 0 & 0 & 0 \\ + & - & 0 & + & + \\ + & 0 & - & 0 & 0 \\ + & 0 & + & - & 0 \\ 0 & 0 & 0 & 0 & - \end{pmatrix}$$

with $\eta(\mathrm{sgn}(M_*)) = -1$ as the closest sign-stable with optimal distance $k_* = 2$. The optimal solution with zero spectral abscissa is impossible to find, because at the distance $k = 1$ we obtain the matrix

$$M' = \begin{pmatrix} - & 0 & 0 & 0 & + \\ + & 0 & 0 & 0 & + \\ + & 0 & 0 & 0 & 0 \\ + & + & + & - & 0 \\ 0 & 0 & + & 0 & - \end{pmatrix}$$

with $\eta(\mathrm{sgn}(M')) = 0.46$ as the one with the minimal spectral abscissa. ○

For a non-negative sign-matrix M we can set the problem of finding the *closest Schur stable non-negative sign-matrix*, analogous to (7.3):

$$\begin{cases} \|\mathrm{sgn}(M) - \mathrm{sgn}(X)\| \to \min \\ \rho(\mathrm{sgn}(X)) = 0. \end{cases} \tag{7.4}$$

Inspecting Theorem 7.7 we see that the problem (7.4) is equivalent to the problem of finding the closest acyclic subgraph, and therefore it can be solved by using Algorithm ACY given in Chapter 4.

We have seen that Theorem 7.1 works both ways only in the two-dimensional case. However, there is a criterion of asymptotic stability, valid in any dimension, which involves sign-matrices and sign-stability.

Theorem 7.10 [35] *Let M_i, $i = 1, \ldots, N$ be a family of Metzler sign-matrices. For all $A_i \in \mathcal{Q}(M_i)$, $i = 1, \ldots, N$, the LSS* (7.1) *is asymptotically stable if and only if all the diagonal entries of M_i, $i = 1, \ldots, N$ are negative and $\sum_{i=1}^{N} M_i$ is strongly Hurwitz sign-stable.* □

Now, assume that the sign-matrices M_i, $i = 1, \dots, N$ have negative diagonal entries, that they are Hurwitz stable, but their sum $M = \sum_{i=1}^N M_i$ is not. Using Algorithm HSM, we can find closest Hurwitz sign-stable matrix M' to M. Let k_* be the optimal value. Now we need to decompose M' into the sum $\sum_{i=1}^N M_i'$, while taking into the consideration the structure of matrices M_i. We have

$$k_* = \|\mathrm{sgn}(M) - \mathrm{sgn}(M')\| \leqslant \sum_{i=1}^N \|\mathrm{sgn}(M_i) - \mathrm{sgn}(M_i')\| = \sum_{i=1}^N k_i. \tag{7.5}$$

We need to choose values k_i and matrices $M_i' \in \mathcal{B}_{k_i}^{\mathrm{sgn}}(M_i)$, so that (7.5) and $M' = \sum_{i=1}^N M_i'$ are satisfied. Matrices M_i', including their sum, will have negative entries on the main diagonal and be Hurwitz stable. Therefore, by Theorem 7.10, any LSS we construct from matrices in $\mathcal{Q}(M_i')$ will be asymptotically stable, while keeping the similar structure to the initial system.

The following example illustrates how to apply Algorithm HSM to stabilize the positive LSS.

Example 7.11 Observe a positive linear switching system, switching between the following three strongly Hurwitz stable matrices:

$$A_1 = \begin{pmatrix} -8 & 0 & 0 & 0 \\ 0 & -4 & 2 & 0 \\ 7 & 0 & -8 & 3 \\ 0 & 1 & 9 & -9 \end{pmatrix} \quad A_2 = \begin{pmatrix} -5 & 0 & 0 & 2 \\ 0 & -3 & 0 & 0 \\ 2 & 2 & -1 & 3 \\ 0 & 5 & 0 & -8 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} -9 & 0 & 2 & 0 \\ 0 & -9 & 0 & 2 \\ 0 & 7 & -2 & 3 \\ 0 & 0 & 0 & -4 \end{pmatrix}.$$

Their corresponding sing-matrices are given by

$$M_1 = \begin{pmatrix} - & 0 & 0 & 0 \\ 0 & - & + & 0 \\ + & 0 & - & + \\ 0 & + & + & - \end{pmatrix} \quad M_2 = \begin{pmatrix} - & 0 & 0 & + \\ 0 & - & 0 & 0 \\ + & + & - & + \\ 0 & + & 0 & - \end{pmatrix}$$

$$M_3 = \begin{pmatrix} - & 0 & + & 0 \\ 0 & - & 0 & + \\ 0 & + & - & + \\ 0 & 0 & 0 & - \end{pmatrix},$$

and their sum $M = M_1 + M_2 + M_3$ with

$$M = \begin{pmatrix} - & 0 & + & + \\ 0 & - & + & + \\ + & + & - & + \\ 0 & + & + & - \end{pmatrix}.$$

Since sign-matrix M is Hurwitz unstable (with $\eta(\text{sgn}(M)) = 1.303$), our positive LSS will not be asymptotically stable. Applying Algorithm HSM, we find its closest strongly Hurwitz stable sing-matrix

$$M' = \begin{pmatrix} - & 0 & 0 & + \\ 0 & - & 0 & + \\ + & + & - & 0 \\ 0 & + & 0 & - \end{pmatrix},$$

with an optimal distance $k_* = 1$. We select $k_1 = k_2 = k_3 = 1$ and decompose $M' = M'_1 + M'_2 + M'_3$ by choosing $M'_i \in \mathcal{B}^{\text{sgn}}_{k_i}(M_i),\ i = 1, 2, 3$. We have:

$$M'_1 = \begin{pmatrix} - & 0 & 0 & 0 \\ 0 & - & 0 & 0 \\ + & 0 & - & + \\ 0 & + & 0 & - \end{pmatrix} \quad M'_2 = \begin{pmatrix} - & 0 & 0 & + \\ 0 & - & 0 & 0 \\ + & + & - & 0 \\ 0 & + & 0 & - \end{pmatrix}$$

$$M'_3 = \begin{pmatrix} - & 0 & 0 & 0 \\ 0 & - & 0 & + \\ 0 & + & - & 0 \\ 0 & 0 & 0 & - \end{pmatrix}.$$

Following the sign pattern of matrices M'_i and correspondingly cutting the interdependencies in matrices A_i, we get

$$A'_1 = \begin{pmatrix} -8 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 \\ 7 & 0 & -8 & 0 \\ 0 & 1 & 0 & -9 \end{pmatrix} \quad A'_2 = \begin{pmatrix} -5 & 0 & 0 & 2 \\ 0 & -3 & 0 & 0 \\ 2 & 2 & -1 & 0 \\ 0 & 5 & 0 & -8 \end{pmatrix}$$

$$A'_3 = \begin{pmatrix} -9 & 0 & 0 & 0 \\ 0 & -9 & 0 & 2 \\ 0 & 7 & -2 & 0 \\ 0 & 0 & 0 & -4 \end{pmatrix}.$$

A newly obtained LSS built from matrices A'_i will be asymptotically stable under the arbitrary switching.

Using the just described procedure we can obtain the information on which interdependencies we should cut from the single systems comprising the LSS, in order to make it stable. ◦

Having matrices of LSS with all negative diagonal entries might not always be the case in real applications. Fortunately, the described procedure can also be used when there are some zero entries on the main diagonal. The essential thing is to cut the interdependencies of the matrices comprising the LSS, so that the sum of the newly obtained matrices (i.e. its graph) contains no cycles. We are not in trouble even if our starting matrices have positive diagonal entries: we just need to make them equal to zero.

Appendix: Implementation and strategies of the presented algorithms

The last part of the thesis is devoted to technical details about the implementation of the presented algorithms. The codes for the algorithms are done in Python, using NuPy and SciPy packages, and can be found on: https://github.com/ringechpil/thesis_codes. Each code comes with the set of comments containing explanations. We also present here one of the most important implementation strategies for countering the cycling caused by calculation imprecisions.

Overcoming the cycling caused by the rounding errors

Even tough the selective greedy method, in theory, can safely and effectively be used on Metzler product families, the cycling might occur due to the computational errors, especially in the case of very sparse matrices. The following example sheds some light on this issue.

Example 7.12 We run the selective greedy algorithm to the product family $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_7$ of $(0,1)$-matrices of dimension 7, for solving the minimization problem. Each uncertainty set $\mathcal{F}_i$ consists of $(0,1)$-vectors containing from 1 to 4 ones. The selective greedy algorithm cycles between the following two matrices:

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \rightleftarrows \quad A' = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Both matrices have the same leading eigenvalue $\lambda_{max} = 1$, and the algorithm computes the same leading eigenvector $\boldsymbol{v} = \boldsymbol{v}'$. Both matrices are minimal in the second, fifth, and sixth rows (the rows that get changed) with the respect to the vector $\boldsymbol{v}$, so we have $(\boldsymbol{a}_i, \boldsymbol{v}) = (\boldsymbol{a}'_i, \boldsymbol{v})$ for $i \in \{2, 5, 6\}$. According to the greedy algorithm, those rows should remain unchanged, and algorithm have actually to terminate with the matrix A. The explanation for this contradictory behaviour is that the machine, due to the calculation imprecisions, keeps miscalculating the above dot products, first taking the "problematic" rows $\boldsymbol{a}'_i$ of the second matrix as the optimal, and then rolling back to the rows $\boldsymbol{a}_i$, seeing them as optimal. $\circ$

One of the most straightforward strategy to resolve this issue is to simply round all calculation to some given number of decimals. This approach is particularly efficient if we are not concerned with high precision. However, if we want our calculations to have a higher order of precision, we need to resort to other strategies.

Another strategy to deal with this issue is to modify the part of the algorithm that dictates the conditions under which the row changes. Here we use notation for the maximization, although for everything is completely analogous for the minimization case.

Let $\boldsymbol{a}_i^{(k)}$ be the i-th row of a matrix A_k obtained in the kth iteration, $\boldsymbol{v}_k$ its leading eigenvector, and $\boldsymbol{a}_i^{(k+1)} = \max_{a \in \mathcal{F}_i}(\boldsymbol{a}, \boldsymbol{v}_k)$, different from $\boldsymbol{a}_i^{(k)}$. Unless the computed dot products $(\boldsymbol{a}_i^{(k)}, v_k)$ and $(\boldsymbol{a}_i^{(k+1)}, v_k)$ are the same, the row $\boldsymbol{a}_i^{(k)}$ will get replaced by $\boldsymbol{a}_i^{(k+1)}$. However, as can be seen from Example 7.12 above, the change may occur even if those dot products are the same. The undesired change may also occur even if the vector $\boldsymbol{a}_i^{(k+1)}$ is not truly optimal, but the computed dot products are really close to each other. This make the algorithm roll back to the vector $\boldsymbol{a}_i^{(k)}$ in the next iteration, and thus cycling.

To counter this, we impose the rule that the row $\boldsymbol{a}_i^{(k)}$ gets replaced by the row $\boldsymbol{a}_i^{(k+1)}$ if and only if $\boldsymbol{a}_i^{(k+1)} = \max_{a \in \mathcal{F}_i}(\boldsymbol{a}, \boldsymbol{v}_k)$ and $|(\boldsymbol{a}_i^{(k)}, \boldsymbol{v}_k) - (\boldsymbol{a}_i^{(k+1)}, \boldsymbol{v}_k)| \geqslant \delta$, where δ is some small parameter. In other words, in the $(k+1)$st iteration, the row $\boldsymbol{a}_i^{(k)}$ will not be replaced by the new row $\boldsymbol{a}_i^{(k+1)}$ if their computed scalar products with the vector $\boldsymbol{v}_k$ are really close to each other. Of course, we need to take care not to make δ too big, since it can lead to the incorrect answer and bad behaviour of the algorithm.

Thesis summary

Before closing the presentation, we make a brief summary by going through the four questions posed in the introductory chapter, listing the thesis contributions on the matter, and giving some ideas for the possible future research.

`N1:` Optimizing spectral radius on a given family of non-negative matrices.

By supplying the greedy method with the selective power method for computing the Perron eigenvalue, we were able to come up with the solution for optimizing the spectral radius on a product family of non-negative matrices. This new approach, the selective greedy method, can handle even the sparse matrices very fast. Moreover, we proved its local quadratic and global linear convergence. A possible further research in this direction would be supplying the greedy algorithm with some other, more efficient, method which will always choose a 'good' leading eigenvector from the Perron subspace.

`N2:` Finding the closest Schur weakly (un)stable non-negative matrix in a given norm.

Using few tricks we have managed to significantly improve the performance of the algorithm for finding the closest stable non-negative matrix in the L_∞ norm. Furthermore, we used the Schur stabilization to graph theory, and even found a way to apply it to approximate the MAS. The shortcomings of our approach for MAS approximation is that it still requires more in-depth theoretical research and large-scale numerical experiments to confirm its efficiency. Of course, this offers a good point for a further research.

`M1:` Optimizing spectral abscissa on a given family of Metzler matrices.

Introducing the translative power method we have managed to modify selective greedy method and make it applicable for optimizing the spectral abscissa on the product families of Metzler matrices. The performance of the selective greedy method is still remarkable as in the non-negative case. The only problematic ground are the very sparse and big zero spectral abscissa matrices. There, the translative method can really slow down. Finding an effective way to overcome this presents another direction for the further development of the work.

`M2:` Finding the closest Hurwitz weakly (un)stable Metzler matrix in a given norm.

We solved the problems of Hurwitz (de)stabilization in L_∞ (and therefore, L_1) and max- norms by providing either an explicit solutions or an effective algorithms. Bringing all pieces of the work together, we used the connection between sign-matrices, graph theory and positive LSS, together with Hurwitz stabilization, to provide a strategy for stabilizing an unstable LSS of any dimension.

Bibliography

[1] F.R.Gantmacher, *The theory of matrices*, Chelsea, New York (2013)

[2] R. A. Horn, C. R. Johnson, *Matrix analysis*, Cambridge University Press (1990)

[3] L. Farina and S. Rinaldi, *Positive linear systems: Theory and applications*, John Wiley & Sons (2000)

[4] D. G. Luenberger, *Introduction to dynamic systems: Theory, models and applications*, John Wiley & Sons (1979)

[5] D. Liberzon, *Switching in systems and control*, Birkhauser, New York (2003)

[6] L. Gurvits, R. Shorten, O. Mason *On the stability of switched positive linear systems*, IEEE Transactions on Automatic Control, 52(6):1099-1103 (2007)

[7] V. S. Kozyakin, *A short introduction to asynchronous systems*, In: Aulbach, B., Elaydi, S., Ladas, G. (eds.) Proceedings of the Sixth International Conference on Difference Equations (Augsburg,Germany 2001): New Progress in Difference Equations, pp. 153-166. CRC Press, Boca Raton (2004)

[8] K. J. Arrow, M. McManus *A note on dynamic stability*, Econometrica, Vol. 26, No. 3, pp. 448-454 (1958)

[9] A. Roberts, *The stability of a feasible random ecosystem*, Nature, 251:607-608 (1974)

[10] R. M. May, *Will a large complex system be stable?*, Nature, 238:413-414 (1972)

[11] R. Levins, *Problems of signed digraphs in ecological theory*, Ecosystem Analysis and Prediction (S. Levin, ed.), 264–277 (1974)

[12] A. Berman and R. J. Plemmons, *Nonnegative matrices in the mathematical sciences*, Acedemic Press, Now York (1979)

[13] K. S. Miller, *Linear difference equations*, Elsevier, Amsterdam (2000)

[14] E. Seneta, *Non-negative matrices and Markov chains*, Springer series in statistics (2006)

[15] D. O. Logofet, *Matrices and graphs: Stability problems in mathematical ecology*, CRC Press, Boca Raton (1993)

[16] D.Cvetković, M.Doob, H.Sachs, *Spectra of graphs. Theory and application*, Academic Press, New York (1980)

[17] W. Leontief, *Input-output economics, 2nd ed.*, Oxford Uni. Press, NY (1986)

[18] N. Guglielmi, V. Protasov, *On the closest stable/unstable nonnegative matrix and related stability radii*, arXiv:1802.03054v1 (2018)

[19] N. Gillis and P. Sharma, *On computing the distance to stability for matrices using linear dissipative Hamiltonian systems*, Automatica 85, 113–121, (2017)

[20] V. D. Blondel, Y. Nesterov, *Polynomial-time computation of the joint spectral radius for some sets of nonnegative matrices*, SIAM J. Matrix Anal. Appl. 31, no 3, 865–876 (2009)

[21] Y. Nesterov, V. Yu. Protasov, *Optimizing the spectral radius*, SIAM J. Matrix Anal. Appl. 34(3), 999-1013 (2013)

[22] S. Friedland, *The maximal eigenvalue of 0-1 matrices with prescribed number of ones*, Linear Alg. Appl., 69, 33–69 (1985)

[23] A. G. Vladimirov, N. Grechishkina, V. Kozyakin, N. Kuznetsov, A. Pokrovskii, and D. Rachinskii, *Asynchronous systems: Theory and practice*, Inform. Processes, 11, 1–45 (2011)

[24] D.G.Cantor and S.A.Lippman, *Optimal investment selection with a multiple of projects*, Econometrica, 63, 1231–1240 (1995)

[25] V. Yu. Protasov, *Spectral simplex method*, Math. Program. 156:485–511 (2016)

[26] N. E. Barabanov, *Absolute characteristic exponent of a class of linear non-stationary systems of differential equations*, Sib. Math. J. 29 521-530, (1988)

[27] Y. Nesterov, V. Yu. Protasov, *Computing closest stable non-negative matrix*, submitted to SIMAX (2017)

[28] M. Akian, S. Gaubert, J. Grand-Clment, J. Guillaud, *The operator approach to entropy games*, 34th International Symposium on Theoretical Aspects of Computer Science (STACS 2017), Hannover, Germany. Article No 6, 6:1 – 6:14 (2017)

[29] M. Charikar, K. Makarychev, Yu. Makarychev *On the advantage over random for maximum acyclic subgraph*, 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07) (2007)

[30] B. Berger, P. W. Shor *Tight bounds for the Maximum Acyclic Subgraph problem*, J. Algorithms, 25(1):118 (1997)

[31] R. Hassin, S. Rubinstein *Approximations for the maximum acyclic subgraph problem*, Information Processing Letters Volume 51, Issue 3, Pages 133–140 (1994)

[32] A. Baharev, H. Schichl, A. Neumaier *An exact method for the minimum feedback arc set problem* https://www.mat.univie.ac.at/ neum/ms/minimum˙feedback˙arc˙set.pdf (2015)

[33] J. Sun, D. Ajwani, P. K. Nicholson, A. Sala, S. Parthasarathy *Breaking cycles in noisy hierarchies* The 9th International ACM Web Science Conference, At Troy, New York, USA (2017)

[34] J. Anderson, *Distance to the nearest stable Metzler matrix*, arXiv:1709.02461v1 (2017)

[35] C. Briat, *Sign properties of Metzler matrices with applications*, arXiv:1512.07043 (2016)

[36] C. Jeffries, V. Klee, P. van den Driessche, *When is a matrix sign-stable?* Canadian Journal of Mathematics, 29: 315-326 (1977)

[37] J. Maybee, J. Quirk, *Qualitative problems in matrix theory* SIAM Reviews, 11(1):30-51 (1969)

[38] J. Quirk and R. Ruppert, *Qualitative economics and the stability of equilibrium*, The review of economic studies, 32(4):311-326 (1965)

[39] P. A. Samuelson, *Foundations of economic analysis*, Harvard Uiv. Press, Cambridge, Mass. (1955)

[40] B. L. Clarke, *Theorems on chemical network stability*, The Journal of Chemical Physics, 62:773-775 (1975)

[41] A. Cvetković, V. Yu. Protasov, *The greedy strategy in optimizing the Perron eigenvalue*, https://arxiv.org/pdf/1807.05099.pdf (2018)
Submitted to Mathematical Programming

[42] A. Cvetković *Stabilising the Metzler matrices with applications to dynamical systems*, https://arxiv.org/pdf/1901.05522.pdf (2019)
Submitted to CALCOLO

[43] K.E.Atkinson, *An introduction to numerical analysis*, John Wiley & Sons (1989)

[44] R. A. Brualdi, A. J. Hoffman, *On the spectral radius of* $(0,1)$*-matrices*, Linear Alg. Appl., 65, 133–146 (1985)

[45] G. M. Engel, H. Schneider, and S. Sergeev, *On sets of eigenvalues of matrices with prescribed row sums and prescribed graph*, Linear Alg. Appl., 455, 187-209 (2014)

[46] B. Liu, *On an upper bound of the spectral radius of graphs*, Discrete Math., 308, no 2, 5317–5324 (2008)

[47] D. D. Olesky, A. Roy, and P. van den Driessche, *Maximal graphs and graphs with maximal spectral radius*, Linear Alg. Appl., 346, 109–130 (2002)

[48] R.Tarjan, *Depth first search and linear graph algorithms*, SIAM J. Comput. 1 (2), 146-160 (1972)