Paper submitted to the IFAC Conference on System Structure and Control (Nantes, France, July 8-10 1998).

Nov 14, 1997. Revised Feb. 10, 1998.

Numerical computation of spectral elements in max-plus algebra

Jean Cochet-Terrasson[*] Guy Cohen[†] Stéphane Gaubert[‡][§]
Michael Mc Gettrick[¶] Jean-Pierre Quadrat[‖]

Abstract

We describe the specialization to max-plus algebra of Howard's policy improvement scheme, which yields an algorithm to compute the solutions of spectral problems in the max-plus semiring. Experimentally, the algorithm shows a remarkable (almost linear) average execution time.

I. Introduction

The max-plus semiring $\mathbb{R}_{\max}$ is the set $\mathbb{R} \cup \{-\infty\}$, equipped with max, written additively ($a \oplus b = \max(a, b)$), and $+$, written multiplicatively ($a \otimes b = a + b$). The zero element will be denoted by $\mathbb{0}$ ($\mathbb{0} = -\infty$), the unit element will be denoted by $\mathbb{1}$ ($\mathbb{1} = 0$). We will adopt the usual algebraic conventions, writing for instance ab for $a \otimes b$, $\mathbb{0}$ for the zero vector or zero matrix (the dimension being clear from the context), etc.

The *spectral problem* for a matrix $A \in (\mathbb{R}_{\max})^{n\times n}$ can be writen as

$$Ax = \lambda x \quad , \tag{1}$$

where $x \in (\mathbb{R}_{\max})^n \setminus \{\mathbb{0}\}$ and $\lambda \in \mathbb{R}_{\max}$, i.e. with the usual notation

$$\forall i \in \{1, \ldots, n\}, \quad \max_{1\leq j\leq n} (A_{ij} + x_j) = \lambda + x_i, \tag{2}$$

where $x \in (\mathbb{R} \cup \{-\infty\})^n$ has at least one finite entry, and $\lambda \in \mathbb{R} \cup \{-\infty\}$. As usual, we will call λ an eigenvalue, and x an associated eigenvector. Whereas the max-plus spectral theorem, which characterizes the solutions of (1), is one of the most studied max-plus results[1], comparatively little can be found about the numerical solving of (1). Unlike in usual algebra, the max-plus spectral problem can be solved exactly in a finite number of steps. The commonly received method to solve (1) relies on Karp's algorithm [20], which computes the (unique) eigenvalue of an *irreducible*[2] matrix A in $O(n^3)$ time[3] (in fact, $O(n \times E)$ time, where E is the number of non-$\mathbb{0}$ entries of A), and[4] $O(n)$ space[3]. Then, some additional manipulations allow one to obtain a generating family of the eigenspace, to compute other interesting spectral characteristics such as the spectral projector, the cyclicity, etc. (see [1, §3.7]). A good bibliography on the maximal cycle mean problem, and a comparison of Karp's algorithm with other classical algorithms, can be found in [9].

The purpose of this paper is to describe a very different algorithm, which seems more efficient, in practice.

We will show how the specialization to the max-plus case of Howard's multichain policy improvement algorithm (see e.g. [10], or [23] for a survey), which is well known in stochastic control, runs in time[5] $N_H O(E)$ and space $O(n)$, where N_H is the number of iterations of the algorithm. Although N_H, which depends on both n and the numerical values of the entries of A, seems difficult to evaluate, its average value is small (experimental tests on full matrices suggest $N_H = O(\log n)$).

In other words, it seems experimentally possible to solve in an *almost linear* (i.e. almost $O(E)$) average time a family of combinatorial problems for which the best standard algorithms run in $O(n \times E)$ time.

We conjecture that the worst case value of the number of iterations N_H is polynomial in E. Examples show that it is at least of order n.

The max-plus version of Howard's algorithm outperforms other known methods with good average execution time, such as linear programming. The only other fast method known to us is Cuninghame-Green and Yixun's algorithm [8], which runs in time $N_{CY} O(E)$, where the average value of the number of iterations N_{CY} is experimentally $O(n^{0.8})$ for full matrices, according to [8].

This work was partially supported by the European Community Framework IV program through the research network ALAPEDES ("The Algebraic Approach to Performance Evaluation of Discrete Event Systems").

[*] BPRE, État Major de l'Armée de l'Air, 24 Bd. Victor, 75015 Paris, France

[†] Centre Automatique et Systèmes, École des Mines de Paris, 35 rue Saint-Honoré, 77305 Fontainebleau Cedex, France. email: cohen@cas.ensmp.fr

[‡] Corresponding author.

[§] INRIA, Domaine de Voluceau, B.P. 105, 78153 Le Chesnay Cedex, France. email: Stephane.Gaubert@inria.fr

[¶] Centre Automatique et Systèmes. email: gettrick@cas.ensmp.fr

[‖] INRIA. email: Jean-Pierre.Quadrat@inria.fr

[1] See [24, 26, 6, 15, 16] for historical references. Recent presentations can be found in [1, §3.2.4,§3.7],[7],[14, § 3.7]. See [22, 21] for generalizations to the infinite dimension case.

[2] Irreducibility is defined in §III below.

[3] Throughout the paper, "time" and "space" refer to the execution time (on a sequential machine) and to the memory space required by the algorithm, respectively.

[4] The natural implementation of Karp's algorithm, described in [20], needs $O(n^2)$ space. However, it is easy to design a two passes variant, which needs a double time, and runs in only $O(n)$ space. As detailed in [9], it is also possible to optimize Karp's algorithm using the sometimes sparse character of the matrix that it builds.

[5] The family of Howard's algorithms works only for "non-degenerate" matrices with at least one non-$\mathbb{0}$ entry per row. For such matrices, $n \leq E$, and $O(E) = O(n + E)$.

Some parts of the present work were initiated in [3], and developed in a different direction in [4, 12]. It is remarkable that Howard's policy improvement scheme not only provides efficient algorithms, but also simple existence proofs. In particular, the existence of generalized eigenmodes for max-plus linear dynamical systems with several incommensurable delays, which is stated in § III below, seems new. A similar proof technique was applied to min-max functions in [12].

The paper is organized as follows.

In section II, we motivate the max-plus spectral problems, by showing how familiar problems in Discrete Event Systems theory and Operations Research reduce to the spectral problem (1), and to some of its extensions.

In section III, we briefly recall the very classical characterization of eigenvalues of max-plus matrices. We discuss their relation with *cycle time* vectors, which govern the asymptotic behavior of max-plus linear dynamical systems. We show how these cycle times can by computed from *generalized eigenmodes*, which are a non classical useful extension of the notion of eigenvector, already used in [12].

In section IV, we describe the max-plus version of Howard's policy iteration algorithm, which computes generalized eigenmodes, and which in fact shows that such eigenmodes exist. The only noticeable originality, by comparison with the classical stochastic control case, is that a *value determination* step can be performed in time $O(n)$, using a special graph exploration algorithm that we present in detail.

In section V, illustrative examples and systematical numerical tests are presented.

A small prototype, written in C, which implements the max-plus policy iteration algorithm described here can be found currently on the web page http://amadeus.inria.fr/gaubert. This prototype will be integrated in the max-plus toolbox of SCILAB[6] which is under development.

II. What the max-plus spectral theory can do for you

In this section, we list several basic problems that reduce to the spectral problem (1) and to some of its extensions. Other applications of the max-plus spectral problem can be found e.g. in [22, 14], and in the references therein.

PROBLEM 1 (MAXIMAL CIRCUIT MEAN). *Given a directed graph*[7] $G = (\mathcal{N}, \mathcal{E})$, *equipped with a* valuation *map* $w : \mathcal{E} \to \mathbb{R}$, *compute the* maximal circuit mean

$$\rho = \max_{c} \frac{\sum_{e \in c} w(e)}{\sum_{e \in c} 1} \quad , \tag{3}$$

where the $\max$ *is taken over all the circuits* c *of* G, *and the sums are taken over all the edges* e *of* c.

The denominator of (3) is the *length* of circuit c. The numerator is the *valuation* or *weight* of circuit c.

By Theorem III.1 below, when G is strongly connected, ρ coincides with the (unique) eigenvalue of matrix $A \in (\mathbb{R}_{\max})^{\mathcal{N} \times \mathcal{N}}$, defined as follows:

$$A_{ij} = \begin{cases} w(i,j) & \text{if } (i,j) \in \mathcal{E}, \\ \mathbb{0} \quad (= -\infty) & \text{otherwise.} \end{cases} \tag{4}$$

Conversely[8], with any matrix $A \in (\mathbb{R}_{\max})^{n \times n}$, we will associate the graph G_A with set of nodes $\mathcal{N} = \{1, \ldots, n\}$ and set of edges $\mathcal{E} = \{(i,j) \mid A_{ij} \neq \mathbb{0}\}$, equipped with the valuation $w(i,j) = A_{ij}$. This bijective correspondence between valued graphs, on the one hand, and max-plus matrices, on the other hand, will be used systematically in the sequel.

PROBLEM 2 (CYCLE TIME). *Given a matrix* $A \in (\mathbb{R}_{\max})^{n \times n}$ *with at least one finite entry per row, compute the* cycle time vector

$$\chi(A) = \lim_{k \to \infty} \frac{1}{k} \times x(k) \quad , \tag{5}$$

where

$$x_i(k) = \max_{1 \leq j \leq n} (A_{ij} + x_j(k-1)), \quad \forall 1 \leq i \leq n, \quad \forall k \in \mathbb{N} \setminus \{0\} \quad , \tag{6}$$

and the initial condition $x(0) \in \mathbb{R}^n$ *is arbitrary.*

Of course, (6) is nothing but a linear system in the max-plus semiring:

$$x(k) = Ax(k-1), \qquad \forall k \in \mathbb{N} \setminus \{0\} \quad . \tag{7}$$

In other words, the cycle time vector $\chi(A)$ determines the linear growth rate of the trajectories of the max-plus linear dynamical system (7). The fact that $\chi(A)$ exists, that it is independent[9] of the initial condition $x(0) \in \mathbb{R}^n$, and that it can be computed from the eigenvalues of the submatrices associated with the strongly connected components of the graph of A, will be detailed in Prop. III.2 below.

We next describe a useful generalization of the max-plus spectral problem, which requires the definition of max-polynomials. A (formal, generalized) max-polynomial in the indeterminate γ is simply a formal sum $\bigoplus_{t \in \mathbb{R}^+} p_t \gamma^t$, where p is a map $\mathbb{R}^+ \to \mathbb{R}_{\max}$, $t \mapsto p_t$, such that $p_t = \mathbb{0}$ for all but finitely many values of $t \in \mathbb{R}^+$. We denote by $\mathbb{R}_{\max}\{\gamma\}$ the set of such polynomials.

The *generalized spectral problem* for a polynomial matrix $\mathcal{A} \in (\mathbb{R}_{\max}\{\gamma\})^{n \times n}$ can be written as:

$$\mathcal{A}(\lambda^{-1})x = x \quad , \tag{8}$$

---

[6] A free open MATLAB-analogue software, developed at INRIA. The current version of SCILAB (without the max-plus toolbox) can be found on http://www-rocq.inria.fr/scilab.

[7] A (finite, directed) graph can be described by a finite set of nodes $\mathcal{N}$ and a set of (oriented) edges $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$. In the sequel, we will use the familiar notions of (directed) path, (directed) circuit, etc., without further comments.

[8] Note that according to (4) and throughout the paper, there is an arc *from* i to j if $A_{ij} \neq \mathbb{0}$. This "direct" convention, which is standard in combinatorial matrix theory and automata theory, was already used in [14]. The "inverse" convention (with $A_{ji} \neq \mathbb{0}$ instead of $A_{ij} \neq \mathbb{0}$) was used in [1]. This "inverse" convention is standard and preferable for discrete event system applications, unless one accepts to deal with linear systems of the form $x(k) = x(k-1)A$, $x(k)$ being a row vector, and A a square matrix, instead of the more familiar $x(k) = Ax(k-1)$, $x(k)$ being a column vector. A consequence of the compromise made in this paper (choosing the "direct" convention, while considering dynamical systems of the second form) is that the accessibility relation, in Prop. III.2 and III.4 below, is the inverse of the one used e.g. in [1, 18].

[9] If some entries of $x(0)$ are infinite, the limit in (5) need not exist, see e.g. [11, Remark 1.1.10, Chap. VI] and [14, Th. 17]. The condition that all the entries of $x(0)$ are finite, and that A has at least one finite entry per row (which guarantees that A sends $\mathbb{R}^n$ to $\mathbb{R}^n$, i.e. that the image by A of a column vector with finite entries has finite entries) is frequently used since it seems practically relevant for discrete event systems and makes life simpler.

where $x \in (\mathbb{R}_{\max})^n \setminus \{\mathbb{0}\}$, $\lambda \in \mathbb{R}$, and $\mathcal{A}(\lambda^{-1}) \in (\mathbb{R}_{\max})^{n\times n}$ denotes the matrix obtained by replacing each occurrence of the indeterminate γ by λ^{-1} ($= -\lambda$, with the usual notation) in the formal expression of $\mathcal{A}$. If $\mathcal{A} = \bigoplus_{t\in\mathbb{R}^+} A_t\gamma^t$, with $A_t \in (\mathbb{R}_{\max})^{n\times n}$, the spectral problem (8) can be rewritten more explicitly as

$$\bigoplus_{t\in\mathbb{R}^+} A_t\lambda^{-t}x = x \quad , \tag{9}$$

where the sum is indeed a finite one, since A_t is $\mathbb{0}$ for all but finitely many values of t. When $\mathcal{A} = A\gamma$, (9) specializes to (1). For this reason, we will call x a *generalized eigenvector* of $\mathcal{A}$ and λ a *generalized eigenvalue*.

The appropriate graphical object to be associated with a polynomial matrix $\mathcal{A} \in (\mathbb{R}_{\max}\{\gamma\})^{n\times n}$ is not a valued directed graph, but the bi-valued directed *multigraph*[10] $G_{\mathcal{A}}$, with set of nodes $\mathcal{N} = \{1,\dots,n\}$, set of edges $\mathcal{E} = \{(i,t,j) \in \mathcal{N}\times\mathbb{R}^+\times\mathcal{N} \mid (A_t)_{ij} \neq \mathbb{0}\}$, initial node map $\mathrm{In}(i,t,j) = i$, terminal node map $\mathrm{Out}(i,t,j) = j$, first valuation $w : w(i,t,j) = (A_t)_{ij}$, and second valuation $\tau : \tau(i,t,j) = t$. Then, the generalized spectral problem (8) becomes

$$\forall i \in \mathcal{N}, \qquad x_i = \max_{(i,t,j)\in\mathcal{E}} (w(i,t,j) - \lambda \times \tau(i,t,j) + x_j) \ . \tag{10}$$

We will see in Theorem III.3 that the solution λ of (10) (which is unique under natural conditions) yields the solution ρ' of the following problem.

PROBLEM 3 (MAXIMAL CIRCUIT MEAN$'$). *Given a multigraph $G = (\mathcal{N}, \mathcal{E}, \mathrm{In}, \mathrm{Out})$, equipped with two valuations $w : \mathcal{E} \to \mathbb{R}$, $\tau : \mathcal{E} \to \mathbb{R}^+$, such that $\sum_{e\in c}\tau(e) > 0$, for all circuits c of G, compute the (generalized) maximal circuit mean:*

$$\rho' = \max_c \frac{\sum_{e\in c} w(e)}{\sum_{e\in c}\tau(e)} \quad , \tag{11}$$

where the $\max$ *is taken over all the circuits c of G.*

As shown in Prop. III.4 below, the generalized spectral problem (8) is also useful in the effective computation of cycle times of some max-plus linear dynamical systems, that are infinite dimensional (multi-delay) versions of (7).

We will say that $\mathcal{A} = \bigoplus_{t\in\mathbb{R}^+} A_t\gamma^t \in (\mathbb{R}_{\max}\{\gamma\})^{n\times n}$ is a *good polynomial matrix* if it has at least one non-$\mathbb{0}$ entry per row, and if there are no circuits in the graph of A_0.

PROBLEM 4 (CYCLE TIME$'$). *Given a good polynomial matrix $\mathcal{A} \in (\mathbb{R}_{\max}\{\gamma\})^{n\times n}$, compute the* cycle time vector

$$\chi(\mathcal{A}) = \lim_{k\to\infty} \frac{1}{k} \times x(k) \quad , \tag{12}$$

where the trajectory x is now given by the dynamics

$$x_i(k) = \max_{1\le j\le n} \max_{t\in\mathbb{R}^+} ((A_t)_{ij} + x_j(k-t)), \ \ \forall k \ge 0 \quad , \tag{13}$$

and $(x(k))_{0>k\ge -K_0}$ is a given (bounded) initial condition, with $K_0 = \max\{t \in \mathbb{R}^+ \mid A_t \neq \mathbb{0}\}$.

More algebraically, (13) can be rewritten as follows:

$$x(k) = \bigoplus_{t\in\mathbb{R}^+} A_t x(k-t), \ \ \forall k \ge 0 \ . \tag{14}$$

Remark II.1. Problems 4 and 2 are in fact two special versions of a more general problem (see e.g. [19]). If $\mathcal{X}$ is a normed vector space and $f : \mathcal{X} \to \mathcal{X}$ is a non-expansive map (i.e. $\|f(x) - f(y)\| \le \|x - y\|$), the limit $\chi(f) = \lim_k 1/k \times f^k(x)$, if it exists, is independent of the initial point x. Problem 2 deals with the case when $\mathcal{X}$ is equal to $\mathbb{R}^n$, equipped with the sup norm, and $f(x) = Ax$. In Problem 4, $\mathcal{X}$ is the set of bounded functions from $[-K_0, 0)$ to $\mathbb{R}^n$, equipped with the sup norm, and f is the evolution operator which with the piece of trajectory $\{x(k)\}_{-K_0\le k<0}$ (initial condition), associates the trajectory obtained after one unit of time: $\{x(k+1)\}_{-K_0\le k<0}$. The evolution operator is obviously well defined since there are no circuits in the graph of A_0. It is clearly monotone and homogeneous, hence, by a simple result [5], it is non-expansive for the sup-norm. Thus, the existence of the limit (12) for a particular bounded function $\{x(k)\}_{-K_0\le k<0}$, implies the existence of $\chi(f)$, which is equal to $\chi(\mathcal{A})$. Conversely, the existence of $\chi(f)$ clearly implies that the limit (12) exists, with $\chi(\mathcal{A}) = \chi(f)$.

III. Some classical and less classical elements of max-plus spectral theory

In all this section, with a matrix $A \in (\mathbb{R}_{\max})^{n\times n}$ we associate the graph $G_A = (\mathcal{N}, \mathcal{E})$, equipped with the valuation w, as defined in the discussion following Eqn 4. The strongly connected components of the graph of A are called *classes*. A matrix is *irreducible* if its graph is strongly connected, i.e. if it has a single class. The following result is classical [24, 26, 6, 15, 16]. See e.g. [1, 7] for recent presentations and proofs.

THEOREM III.1 (MAX-PLUS SPECTRAL THEOREM). *An irreducible matrix $A \in (\mathbb{R}_{\max})^{n\times n}$ has a unique eigenvalue, given by* (3).

In general, there are several non-proportional eigenvectors (see e.g. [1] or [14]). A reducible matrix A has in general several distinct eigenvalues, and the maximal circuit mean (3) yields precisely the maximal eigenvalue (see e.g. [11, Ch.IV], [14], [2] for characterizations of the spectrum of reducible matrices).

We say that i has access to j if there is a path from i to j in the graph of A. We say that i has access to a class C if it has access to any $j \in C$ (this property is obviously independent of the choice of $j \in C$, by definition of a class). By “eigenvalue of a class C”, we mean the eigenvalue of the $C \times C$ submatrix of A, which is unique by Theorem III.1.

The following result appeared in [18, Prop. 7], and, in a stochastic context, in [1, Th. 7.36].

PROPOSITION III.2 (CYCLE TIME FORMULA). *Let $A \in (\mathbb{R}_{\max})^{n\times n}$, with at least one finite entry per row. The i-th entry $\chi_i(A)$ of the cycle time vector is equal to the maximum of the eigenvalues of the classes to which i has access*[8].

The next statement uses the correspondence between polynomial matrices and multigraphs, described in § II above. We will say that a polynomial matrix $\mathcal{A}$ is *irreducible* if its multigraph is strongly connected. More generally, we will naturally extend the notions of accessibility, classes, etc. to polynomial matrices (these notions are defined as in the case of ordinary matrices, but

---

[10] Loosely speaking, a multigraph is a graph in which several edges can link the same pair of nodes. Formally, a (finite) multigraph can be defined by a (finite) set of nodes $\mathcal{N}$, a (finite) set of edges $\mathcal{E}$, and two maps $\mathrm{In} : \mathcal{E} \to \mathcal{N}$ and $\mathrm{Out} : \mathcal{E} \to \mathcal{N}$, which give the initial node and terminal node of an edge, respectively.

replacing the graph G_A by the multigraph $G_{\mathcal{A}}$). The following result is taken from [1, Th. 3.28]

THEOREM III.3 (SPECTRAL THEOREM$'$). *An irreducible polynomial matrix $\mathcal{A} = \bigoplus_{t\in\mathbb{R}^+} A_t\gamma^t \in (\mathbb{R}_{\max}\{\gamma\})^{n\times n}$, such that the graph of A_0 has no circuits*[11]*, admits a unique generalized eigenvalue λ, given by* (11).

The following extension of Prop. III.2 is immediate.

PROPOSITION III.4 (CYCLE TIME FORMULA$'$). *Let $\mathcal{A}$ denote a good polynomial matrix. The i-th entry $\chi_i(\mathcal{A})$ of the cycle time vector is equal to the maximum of the generalized eigenvalues of the classes to which i has access*[8]*.* ☐

Since the decomposition of a directed graph or multigraph in strongly connected components can be done in linear time using Tarjan's algorithm [25], Prop. III.2 and Prop. III.4 reduce in linear time the computation of the cycle time vector to the computation of the (possibly generalized) eigenvalues of irreducible (possibly polynomial) matrices. In particular, the traditional way to compute the cycle time vector $\chi(A)$ is to compute the eigenvalues of the classes of A via Karp's algorithm [20], and then to apply Prop. III.2. This method does not work for the generalized dynamics (13), since Karp's algorithm cannot compute *generalized* eigenvalues. There are two traditional ways to overcome this difficulty. — When A_t is zero except for *integer values* of t, an elimination of the implicit part and a familiar augmentation of state reduces the generalized spectral problem for $\mathcal{A}$ to an ordinary spectral problem for a larger matrix A'. This method, which is presented in [1, § 2.5.3,§ 2.5.4], is not so expensive when the number of values of t for which $A_t \neq \mathbb{0}$ is small, particularly if it is implemented with some refinements, as in [13], for $\mathcal{A} = A_0 \oplus A_1\gamma$. — The second method relies on the general techniques presented in [17, Appendix V], which allow one to maximize in pseudo-polynomial time a ratio of the form $w(c)/\tau(c)$ for c in a finite set S, provided that for any value of $\lambda\in\mathbb{R}$, we know how to maximize in polynomial time the ratio $w(c) - \lambda\tau(c)$ for c in the same set S.

We will not discuss in detail these two more or less classical approaches, but rather show how a different generalization of the spectral problem allows us to determine directly and in full generality cycle time vectors. All the remaining part of this paper, and in particular, the max-plus version of Howard's policy improvement algorithm, will be based on this new spectral problem.

We consider a good polynomial matrix $\mathcal{A}$. We say that $(\eta, x) \in (\mathbb{R}^n)^2$ is a *generalized eigenmode*[12] if there exists $K\in\mathbb{R}$ such that

$$k\in\mathbb{R},\ k\geq K \ \Rightarrow\ D^k x = \mathcal{A}(D^{-1})D^k x \ , \tag{15}$$

where $D \overset{\text{def}}{=} \text{diag}(\eta_1,\dots,\eta_n)$ and $D^k = \text{diag}(k\times\eta_1,\dots,k\times\eta_n)$.

When $\mathcal{A} = A\gamma$, (15) becomes

$$k\in\mathbb{R},\ k\geq K \ \Rightarrow\ D^k x = AD^{k-1}x \ . \tag{16}$$

That is, the action of A coincides with the action of D on the orbit $\{D^k x\}_{k\geq K-1}$. As detailed in footnote 12, the eigenmode equation (15) is obtained by looking for an ultimately affine solution of (13), $x(k) = D^k x = k\times\eta + x$. If such a solution exists, $\chi(\mathcal{A}) = \lim_k \frac{1}{k}\times x(k) = \eta$. The next lemma follows readily from this observation, and from the fact, mentioned in Remark II.1 above, that the limit $\lim_k \frac{1}{k}\times x(k) = \eta$ is independent of the particular bounded initial condition.

LEMMA III.5. *If a good polynomial matrix $\mathcal{A}$ has a generalized eigenmode (η, x), then $\chi(\mathcal{A}) = \eta$.* ☐

In particular, if $\mathcal{A}$ is irreducible, Prop. III.4 implies that $\eta = \chi(\mathcal{A}) = (\lambda,\dots,\lambda)$, where λ is the generalized eigenvalue of $\mathcal{A}$. Therefore, (15) reduces to the (generalized) spectral problem (8), and x is a (generalized) eigenvector of $\mathcal{A}$. I.e., for irreducible matrices, finding generalized eigenmodes is equivalent to finding generalized eigenvectors.

The existence of generalized eigenmodes was proved in [12] when $\mathcal{A} = A\gamma$, as a special case of a more general result for min-max functions. In the next section, we will show how the max-plus version of Howard's policy improvement algorithm allows us to compute generalized eigenmodes. In particular, the termination of the algorithm will prove the existence of such eigenmodes, for good polynomial matrices.

IV. The max-plus policy improvement algorithm

In this section, $\mathcal{A}$ will be a good polynomial matrix. We will use systematically the multigraph $G_{\mathcal{A}} = (\mathcal{N},\mathcal{E},\text{In},\text{Out})$ equipped with the valuations w,τ, canonically associated with $\mathcal{A}$ in § II.

It can be checked that the eigenmode equation (15) which seems deceivingly to involve an infinite number of conditions, is equivalent to the following finite system:

$$\eta_i = \max_{(i,t,j)\in\mathcal{E}} \eta_j \tag{17}$$

$$x_i = \max_{(i,t,j)\in\overline{\mathcal{E}}} (w(i,t,j) - \tau(i,t,j)\times\eta_j + x_j) \ , \tag{18}$$

$$\text{where } \ \overline{\mathcal{E}} = \{(i,t,j)\in\mathcal{E} \mid \ \eta_i = \eta_j\} \ .$$

In loose terms, the multichain policy iteration algorithm will solve this system by trying to guess the arcs that attain the maximum. A precise statement of this idea needs the definition of

---

[11] In [1, Th. 3.28], it is only required that the circuits of the graph of A_0 have negative weights. We will not need this degree of generality here. In terms of the associated dynamical systems (13), the condition of the theorem simply means that there are no circuits involving zero-delay causality relations.

[12] This spectral notion is obtained by *two successive* generalizations of ordinary spectral problems. The *first generalization* consists in replacing ordinary dynamical systems of the form (7) (with unitary delays) by systems of the form (14) (with multiple delays). The ordinary spectral problem (1) and its generalization (9) are obtained by looking for solutions of the form $x(k) = \lambda^k x$, where λ is a scalar and $x \in (\mathbb{R}_{\max})^n\setminus\{0\}$. But the definition of cycle-time vectors requires x to have finite coordinates. Then, in the general case, a simple affine regime $x(k) = \lambda^k x = k\times(\lambda,\dots,\lambda)^T + x$ need not exist, but a more general affine regime $x(k) = D^k x$, where D is a diagonal matrix, is expected. In other words, we expect the different entries of $x(k)$ to have different (linear) growth rates, given by the diagonal entries of D. Hence, the *second generalization* consists in substituting $x(k) = D^k x$ for k large enough (i.e. $k\geq K+K_0$) in (14): then, one obtains precisely the generalized eigenmode equation (15). Contrary to the case when $D=\lambda$, D^{-1} need not commute with the matrices A_t, and thus, the relation $x = \mathcal{A}(D^{-1})x = \bigoplus_{t\in\mathbb{R}^+} A_t D^{-t}x$ need not imply that $D^k x = \mathcal{A}(D^{-1})D^k x$, for $k\geq 0$. This is why (15) has to be stated *for all* large k.

policy, which is a map

$$\pi : \mathcal{N} \to \mathcal{E}, \quad \text{such that} \quad \mathrm{In}(\pi(i)) = i, \ \forall i \in \mathcal{N} \ .$$

That is, a policy is just a map which with a node associates an edge starting from this node.

With a policy π, we associate the special polynomial matrix $\mathcal{A}^\pi = \bigoplus_{t \in \mathbb{R}^+} A_t^\pi \gamma^t$:

$$(A_t^\pi)_{ij} = \begin{cases} w(\pi(i)) & \text{if } j = \mathrm{Out}(\pi(i)) \text{ and } t = \tau(\pi(i)) \\ \mathbb{0} & \text{otherwise.} \end{cases}$$

Hence, the matrix $\mathcal{A}^\pi$ has exactly one non-zero entry per row, which corresponds to the edge selected by π, i.e. in the multigraph of $\mathcal{A}^\pi$, $\pi(i)$ is the unique edge starting from i. It has the same valuations w and τ as in the original multigraph of $\mathcal{A}$. E.g., we have depicted in Fig. 2 below the multigraph (in fact, the graph) of $\mathcal{A}^{\pi_1}$, where π_1 is the policy $1 \to 1, i \to 2$, for $i = 2, 3, 4$ and $\mathcal{A} = A\gamma$, where A is displayed in Fig. 1.

We first show how a generalized eigenmode (η, x) of a matrix of the form $\mathcal{A}^\pi$ can be computed in time $O(n)$.

Algorithm IV.1 (Value Determination). Input: a good polynomial matrix $\mathcal{A}$ and a policy π. Output: a generalized eigenmode of $\mathcal{A}^\pi$.

1. Find a circuit c in the multigraph of $\mathcal{A}^\pi$
2. Set
$$\overline{\eta} = \frac{\sum_{e \in c} w(e)}{\sum_{e \in c} \tau(e)} \ . \tag{19}$$
3. Select an arbitrary node i in c, set $\eta_i = \overline{\eta}$ and set x_i to an arbitrary value, say $x_i = 0$.
4. Visiting all the nodes j that have access to i in backward topological order, set
$$\eta_j = \overline{\eta} \tag{20}$$
$$x_j = w(\pi(j)) - \overline{\eta} \times \tau(\pi(j)) + x_{\mathrm{Out}(\pi(j))} \tag{21}$$
5. If there is a nonempty set C of nodes j that do not have access to i, repeat the algorithm using the $C \times C$ submatrix of $\mathcal{A}$ and the restriction of π to C.

The algorithm should be specified as follows.

Step 1 is very easy to implement: we can start from an arbitrary node i, move to node $j = \mathrm{Out}(\pi(i))$, then possibly to node $\mathrm{Out}(\pi(j))$, etc., until a node that is already visited is found. Then, a circuit has been found. This requires a linear time.

Eqn (21) requires visiting the nodes in backward topological order, starting from i, since the value of $x_{\mathrm{Out}(\pi(j))}$ must be already fixed when we visit node j and set x_j. This complete visit can be done in linear time, at the price of an a priori tabulation of the (multi-valued)-inverse of the map $\mathcal{N} \to \mathcal{N}, \ j \mapsto \mathrm{Out}(\pi(j))$. Computing the inverse of this map also requires a linear time. The handling of this inverse is in fact the only part of the algorithm which requires more refined data types than simple arrays (in our implementation, we used linearly chained lists).

Step 5 is formulated in a recursive way only to simplify the statement of the algorithm, which is essentially non recursive.

The above considerations justify the following theorem.

Theorem IV.2. *Algorithm IV.1 computes a generalized eigenmode of $\mathcal{A}^\pi$ in time and space $O(n)$.* □

The second ingredient of Howard's algorithm is a policy improvement routine, which given a policy π and a generalized eigenmode (η, x) of $\mathcal{A}^\pi$, either finds a new policy π' such that $\chi(\mathcal{A}^{\pi'}) \geq \chi(\mathcal{A}^\pi)$, or proves that (η, x) is a generalized eigenmode of $\mathcal{A}$.

Algorithm IV.3 (Policy Improvement). Input: a good polynomial matrix $\mathcal{A}$, a policy π, together with a generalized eigenmode (η, x) of $\mathcal{A}^\pi$. Output: a policy π', such that $\chi(\mathcal{A}^{\pi'}) \geq \chi(\mathcal{A}^\pi)$.

1. Let[13]
$$\begin{aligned} J &= \{i \mid \max_{(i,t,j) \in \mathcal{E}} \eta_j > \eta_i\} \\ K(i) &= \operatorname*{arg\,max}_{(i,t,j) \in \mathcal{E}} \eta_j, \ \text{for } i = 1 \dots n, \\ I &= \{i \mid \max_{e=(i,t,j) \in K(i)} (w(e) - \tau(e)\eta_j + x_j) > x_i\} \\ L(i) &= \operatorname*{arg\,max}_{e=(i,t,j) \in K(i)} (w(e) - \tau(e)\eta_j + x_j) \ , \\ & \quad \text{for } i = 1 \dots n. \end{aligned}$$
2. If $I = J = \emptyset$, (η, x) is a generalized eigenmode of $\mathcal{A}$. Stop.
3. (a) If $J \neq \emptyset$, we set:
$$\pi'(i) = \begin{cases} \text{any } e \text{ in } K(i) & \text{if } i \in J, \\ \pi(i) & \text{if } i \notin J. \end{cases}$$
   (b) If $J = \emptyset$ but $I \neq \emptyset$, we set
$$\pi'(i) = \begin{cases} \text{any } e \in L(i) & \text{if } i \in I, \\ \pi(i) & \text{if } i \notin I. \end{cases}$$

The policy improvement rules 3a and 3b simply mean that one selects for the new policy the edges which realize the maximum in Eqns (17), (18). This maximum is taken hierarchically: Eqn (17) has priority on Eqn (18) in a policy improvement step. Only when Eqn (17) is satisfied Eqn (18) is used to determine the new policy. The other conditions in steps 3a and 3b simply mean that the preceding values of π should be kept in π', whenever possible. These technical tricks will guarantee the termination of the policy iteration algorithm below, even when "degenerate" policy improvements in which $\chi(\mathcal{A}^{\pi'}) = \chi(\mathcal{A}^\pi)$ occur.

The sets $K(i)$ and $L(i)$, which are introduced to simplify the statement of the algorithm, need not be explicitly tabulated. Clearly, Algorithm IV.3 runs in $O(|\mathcal{E}|)$ time[14] and $O(n)$ space[15]

We next state the max-plus version of Howard's policy iteration algorithm.

Algorithm IV.4 (Max-plus Policy Iteration). Input: a good polynomial matrix $\mathcal{A}$. Output: a generalized eigenmode of $\mathcal{A}$.

1. *Initialization.* Select an arbitrary policy π_1. Compute a generalized eigenmode (η^1, x^1) of $\mathcal{A}^{\pi_1}$, using Algorithm IV.1. Set $k = 1$.

---

[13] Recall that by $\arg\max_{e \in \mathcal{E}} f(e)$, we mean as usual the set of elements $m \in \mathcal{E}$ such that $f(m) = \max_{e \in \mathcal{E}} f(e)$.

[14] $|\mathcal{E}|$ simply denotes the number of edges of the multigraph.

[15] The algorithm needs less internal memory ($O(n)$ space) than the coding of the input itself, which requires $O(|\mathcal{E}|)$ space.

2. *Policy improvement*. Improve the policy π_k, using Algorithm IV.3 with input $\pi = \pi_k, \eta = \eta^k, x = x^k$. If the stopping condition of Algorithm IV.3 is satisfied, (η^k, x^k) is a generalized eigenmode of $\mathcal{A}$. Stop. Otherwise, set $\pi_{k+1} = \pi'$ (the output of Algorithm IV.3).
3. *Value determination*. Find a generalized eigenmode (η^{k+1}, x^{k+1}) of $\mathcal{A}^{\pi_{k+1}}$ using Algorithm IV.1, taking the special value $x_i^{k+1} = x_i^k$ in step 3,IV.1.
4. Increment k by one and go to step 2.

The algorithm builds a sequence of generalized eigenmodes (η^k, x^k) that is strictly increasing for the lexicographic order on $(\mathbb{R}^n)^2$, defined by $(x, y) <_{\text{lex}} (x', y')$ if $x < x'$ or $x = x'$ and $y < y'$. The fact that x_i^{k+1} must be set to x_i^k in step 3 of Algorithm IV.1 is a conservative trick analogous to the fact that the values of π are kept in π' whenever possible, in Algorithm IV.3. This technical condition is essential to guarantee the strict monotonicity of the sequence (η^k, x^k), which is needed in the proof that the algorithm terminates.

The proof of the following result is similar to the proof of the main theorem of [10]. It relies essentially on a version of the maximum principle for transient Markov chains. A more algebraic version of this proof, using germs of affine functions, appears in [4], and in the proof of the results announced therein.

THEOREM IV.5. *The max-plus policy iteration algorithm terminates in a number of iterations N_H which is less than the number of policies. One iteration requires*[14] *$O(|\mathcal{E}|)$ time. The algorithm requires $O(n)$ space*[15]. □

Indeed, the same policy is never selected twice. Bounding N_H by the number of policies which is finite but exponential is very coarse. On experimental random examples, N_H is very small, as detailed in section V below. The following result is an immediate consequence of the termination of the policy iteration algorithm and of Lemma III.5.

COROLLARY IV.6. *A good polynomial matrix $\mathcal{A}$ has a generalized eigenmode (η, x). In particular, the cycle time $\chi(\mathcal{A}) = \eta$ exists.* □

Remark IV.7. Howard's algorithm is not limited to spectral problems. It is possible to design policy iteration algorithms for fixed points equations of the form $x = Ax \oplus b$, where A is a square matrix with maximal eigenvalue strictly less than $\mathbb{1}$, and b a column vector. This will be detailed elsewhere.

V. Examples and numerical tests

A. Illustrative example

We apply the max-plus policy iteration algorithm to determine the eigenvalue of the matrix displayed in Fig. 1. This corresponds to the case where $\mathcal{A} = A\gamma$, and $\tau \equiv 1$. In particular, the multigraph of $\mathcal{A}$ will be identified with the graph of A.

$$A = \begin{bmatrix} 1 & 2 & \mathbb{0} & 7 \\ \mathbb{0} & 3 & 5 & \mathbb{0} \\ \mathbb{0} & 4 & \mathbb{0} & 3 \\ \mathbb{0} & 2 & 8 & \mathbb{0} \end{bmatrix}$$

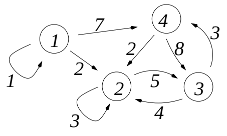

Fig. 1. A matrix and its graph

The following run of the algorithm is visualized in Fig. 2. We choose the initial policy π_1: $1 \to 1$, $i \to 2$, for $i = 2, 3, 4$. Applying Algorithm IV.1, we find a first circuit $c_1 : 1 \to 1$, with $\overline{\eta} = w(c_1)/\tau(c_1) = 1$. We set $\eta_1^1 = 1$, $x_1^1 = 0$. Since 1 is the only node which has access to 1, we apply Algorithm IV.1 to the subgraph of $G_{\mathcal{A}}$ with nodes $2, 3, 4$. We find the circuit $c_2 : 2 \to 2$ and set $\overline{\eta} = w(c_2)/\tau(c_2) = 3$, $\eta_2^1 = 3$ and $x_2^1 = 0$. Since $3, 4$ have access to 2, we set $\eta_i^1 = 3$ for $i = 3, 4$. Moreover, an application of (21) yields $x_3^1 = 4 - 3 + x_2^1$, $x_4^1 = 2 - 3 + x_2^1$. To summarize:

$$\eta^1 = \begin{bmatrix} 1 & 3 & 3 & 3 \end{bmatrix}^T, \quad x^1 = \begin{bmatrix} 0 & 0 & 1 & -1 \end{bmatrix}^T .$$

We improve the policy using Algorithm IV.3. Since $J = \{1\} \neq \emptyset$, we have a type 3a improvement. This yields $\pi_2 : i \to 2$, for $i = 1, 2, 3, 4$. Only the entry 1 of x^1 and η^1 has to be modified, which yields

$$\eta^2 = \begin{bmatrix} 3 & 3 & 3 & 3 \end{bmatrix}^T, \quad x^2 = \begin{bmatrix} -1 & 0 & 1 & -1 \end{bmatrix}^T .$$

We next tabulate with less details the end of the run of the algorithm. *Algorithm IV.3, type 3b policy improvement.* $\pi_3 : 1 \to 4, 2 \to 3, 3 \to 2, 4 \to 3$. *Algorithm IV.1. Value determination.* Circuit found, $c : 3 \to 2 \to 3$, $\overline{\eta} = (w(2,3) + w(3,2))/2 = 9/2$.

$$\eta^3 = \begin{bmatrix} \frac{9}{2} & \frac{9}{2} & \frac{9}{2} & \frac{9}{2} \end{bmatrix}^T, \quad x^3 = \begin{bmatrix} \frac{11}{2} & 0 & -\frac{1}{2} & 3 \end{bmatrix}^T .$$

Algorithm IV.3, type 3b policy improvement. The only change is $\pi_4(3) = 4$. *Algorithm IV.1. Value determination.* Circuit found, $c : 3 \to 4 \to 3$, $\overline{\eta} = (w(3,4) + w(4,3))/2 = 11/2$.

$$\eta^4 = \begin{bmatrix} \frac{11}{2} & \frac{11}{2} & \frac{11}{2} & \frac{11}{2} \end{bmatrix}^T, \quad x^4 = \begin{bmatrix} 4 & -\frac{1}{2} & 0 & \frac{5}{2} \end{bmatrix}^T .$$

Algorithm IV.3. Stop. $11/2$ is an eigenvalue of A, and x^4 is an eigenvector.

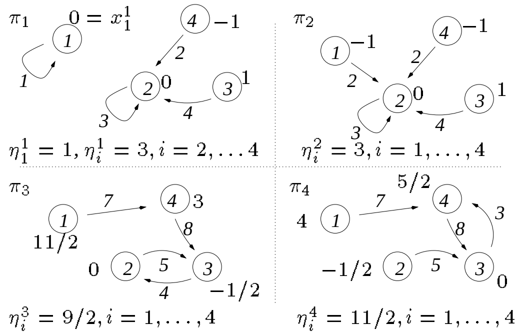
π_1
$0 = x_1^1$
1
4
-1
2
1
2
0
3
1
3
4
$\eta_1^1 = 1, \eta_i^1 = 3, i = 2, \ldots 4$
π_2
4
-1
1
-1
2
2
2
0
3
1
3
4
$\eta_i^2 = 3, i = 1, \ldots, 4$
π_3
7
4
3
1
8
$11/2$
0
2
5
3
4
$-1/2$
$\eta_i^3 = 9/2, i = 1, \ldots, 4$
π_4
$5/2$
7
4
3
4
1
8
$-1/2$
2
5
3
0
$\eta_i^4 = 11/2, i = 1, \ldots, 4$

Fig. 2. The sequence of policies built by the max-plus policy iteration algorithm, for the matrix A displayed in Fig. 1. The valuations of the nodes indicate the vectors x^k, $k = 1 \ldots, 4$.

B. Numerical Tests

The results of the numerical experiments displayed in Fig. 3,4,5 should be self-explanatory.

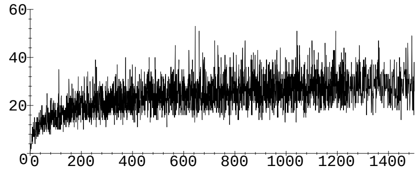

Fig. 3. Number of iterations N_H of Howard's algorithm as a function of the dimension, for full random matrices, with i.i.d entries distributed uniformly on an interval.

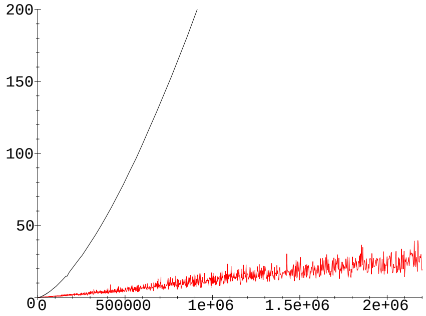

Fig. 4. Cpu time (in sec.) of Howard's algorithm (in red) vs Karp's algorithm (in black) on a pentium 200Mhz with 500Mb of RAM, as a function of the number of arcs for full random matrices (same probabilistic characteristics as in Fig. 3).

References

[1] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity*. Wiley, 1992.

[2] R.B. Bapat, D. Stanford, and P. van den Driessche. Pattern properties and spectral inequalities in max algebra. *SIAM Journal of Matrix Analysis and Applications*, 16(3):964–976, 1995.

[3] J. Cochet-Terrasson. Étude et mise en œuvre de l algorithme de Howard sous des hypothèses faibles d'accessibilité. Rapport de stage, ENSTA et DEA "Modélisation et Méthodes Mathématiques en Économie", Université de Paris I, July 1996.

[4] J. Cochet-Terrasson, S. Gaubert, and J. Gunawardena. Dynamics of min-max functions. 1997. Submitted for publication. Also Technical Report HPL-BRIMS-97-13.

[5] M.G. Crandall and L. Tartar. Some relations between non expansive and order preserving maps. *Proceedings of the AMS*, 78(3):385–390, 1980.

[6] R.A. Cuninghame-Green. *Minimax Algebra*. Number 166 in Lecture notes in Economics and Mathematical Systems. Springer, 1979.

[7] R.A Cuninghame-Green. Minimax algebra and applications. *Advances in Imaging and Electron Physics*, 90, 1995.

[8] R.A. Cuninghame-Green and Lin Yixun. Maximum cycle-means of weighted digraphs. *Appl. Math. JCU*, 11B:225–234, 1996.

[9] A. Dasdan and R. Gupta. Faster maximum and minimum mean cycle algorithms for system performance analysis. Technical report 97-07, Univ. of California, Irvine, 1997. Submitted to IEEE-Transactions on Computer-Aided Design.

[10] E.V. Denardo and B.L. Fox. Multichain Markov renewal programs. *SIAM J.Appl.Math*, 16:468–487, 1968.

[11] S. Gaubert. *Théorie des systèmes linéaires dans les dioïdes*. Thèse, École des Mines de Paris, July 1992.

[12] S. Gaubert and J. Gunawardena. The duality theorem for min-max functions. *C.R.A.S*, 1997. Accepted.

[13] S. Gaubert and J. Mairesse. Modelling and analysis of timed petri nets using heaps of pieces. 1997. Submitted for publication. Also technical report LITP/97-14. An abridged version can be found in the Proceedings of the ECC'97, Bruxells, 1997.

[14] S. Gaubert and M. Plus. Methods and applications of (max,+) linear algebra. In R. Reischuk and M. Morvan, editors, *STACS'97*, number 1200 in LNCS, Lübeck, March 1997. Springer.

[15] M. Gondran and M. Minoux. Valeurs propres et vecteurs propres en théorie des graphes. In *Problèmes combinatoires et théorie des graphes*, number 260 in Colloques internationaux CNRS, Orsay, 1976.

[16] M. Gondran and M. Minoux. Valeurs propres et vecteurs propres dans les dioïdes et leur interprétation en théorie des graphes. *EDF, Bulletin de la Direction des Etudes et Recherches, Serie C, Mathématiques Informatique*, 2:25–41, 1977.

[17] M. Gondran and M. Minoux. *Graphes et algorithmes*. Eyrolles, Paris, 1979. Engl. transl. *Graphs and Algorithms*, Wiley, 1984.

[18] J. Gunawardena. Cycle time and fixed points of min-max functions. In G. Cohen and J.P. Quadrat, editors, *11th International Conference on Analysis and Optimization of Systems*, number 199 in LNCIS, pages 266–272. Springer, 1994.

[19] J. Gunawardena and M. Keane. On the existence of cycle times for some nonexpansive maps. Technical Report HPL-BRIMS-95-003, Hewlett-Packard Labs, 1995.

[20] R.M. Karp. A characterization of the minimum mean-cycle in a digraph. *Discrete Maths.*, 23:309–311, 1978.

[21] V. Kolokoltsov and V Maslov. *Idempotent analysis and applications*. Kluwer Acad. Publisher, 1997.

[22] V. Maslov and S. Samborskiĭ, editors. *Idempotent analysis*, volume 13 of *Adv. in Sov. Math.* AMS, RI, 1992.

[23] M.L. Puterman. Markov decision processes. *Handbboks in operations research and management science*, 2:331–434, 1990.

[24] I.V. Romanovskiĭ. Optimization and stationary control of discrete deterministic process in dynamic programming. *Kibernetika*, 2:66–78, 1967. Engl. transl. in Cybernetics 3 (1967).

[25] R.E. Tarjan. Depth first search and linear graph algorithms. *SIAM J. Comput*, 1:146–160, 1972.

[26] N.N. Vorobyev. Extremal algebra of positive matrices. *Elektron. Informationsverarbeitung und Kybernetik*, 3, 1967. in Russian.

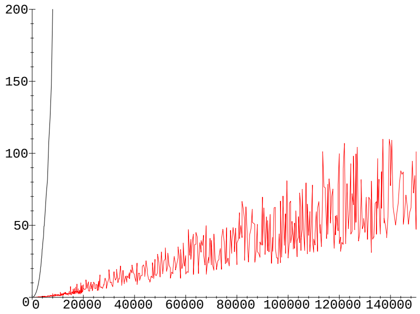

Fig. 5. Cpu time (in sec.) of Howard's algorithm (in red) vs Karp's algorithm (in black), on a pentium 200Mhz with 500Mb of RAM, as a function of the number of nodes n, for sparse random matrices, with exactly 5 arcs starting from each node. The 5 successors of a given node are drawn randomly from $\{1, \ldots, n\}$ with the uniform distribution. The corresponding valuations of the arcs are i.i.d., with an uniform distribution on an interval.