2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

A Suite of Distributed Methodologies to Solve the Sparse Analytic Hierarchy Process Problem

Marta Menci[1*], Gabriele Oliva[1,2], Marco Papi[1], Roberto Setola[1,2] and Antonio Scala[3]

Abstract— **In this paper we aim at finding effective distributed algorithms to solve the *Sparse Analytic Hierarchy Process* (SAHP) problem, where a set of networked agents (e.g., wireless sensors, mobile robots or IoT devices) need to be ranked based on their utility/importance. However, instead of knowing their absolute importance, the agents know their relative utility/importance with respect to their neighbors. Moreover, such a relative information is perturbed due to errors, subjective biases or incorrect information. Recently, the *Sparse Eigenvector Method* proved its effectiveness in tackling this problem. However, such a method has several drawbacks, such as demanding computation/communication requirements and lack of control on the magnitude of the computed estimate. With the aim to mitigate such issues, in this paper we inspect the possibility to resort to a suite of different methodologies, each inspired to well known algorithms in the literature, i.e., Metropolis-Hastings Markov chains, Heat-Bath Markov chains and formation control. The proposed methodologies are less demanding in terms of memory and communication capabilities; however, each approach has its own strength points and drawbacks. The aim of this paper is thus to provide a numerical comparison of their performances over networks with different characteristics.**

I. Introduction

The *Analytic Hierarchy Process* (AHP) [25] is an effective decision-making technique aimed at ranking a set of alternatives based on their utility or importance. This task is done by resorting to relative preference information, i.e., by considering the ratio of the utilities. Such ratios are typically perturbed or affected by subjective biases or errors, and do not represent a perfectly transitive ordering, thus calling for approximated solutions. In its standard formulation, the AHP problem requires knowledge on the ratio of the utilities of each pair of alternatives; however, in the literature several results have been provided in order to handle incomplete information [5], [12], [13], [24].

The AHP problem appears of particular interest in a distributed decision-making context; in particular, in this paper we are interested to the scenario envisaged in [24], where a set of networked agents need to rank themselves in a distributed manned, based on the knowledge of perturbed ratios of utilities with respect to their neighbors. An example in this sense is a network of mobile agents, each provided with different equipment or tools in order to perform their tasks; agent i, by comparing its equipment with the one of a neighboring agent j, is able to assess its relative priority; by composing such a local information, the agents are able to rank themselves and compute an absolute priority. Notice that, in the literature, typical distributed ranking approaches aim at developing a ranking based on the topological structure of the network (see, among other methodologies, the *pagerank* problem [15]). Conversely, the ranking achieved within the AHP approach has no relation with the topological structure, but is the result of the composition of the relative importance information involving the nodes and their neighbors.

We point out that the standard approach to solve the AHP problem, the so-called *eigenvector method*, has been often regarded as an arbitrary or questionable choice for approximating the unknown utilities (see for instance the debate in [7], [26]); the main reason for such a criticism is that the ranking obtained via the eigenvector method might be reversed when additional alternatives are considered. Moreover, the procedure developed in [24], although effective, has nontrivial computational requirements. To the best of our knowledge, no other approach has been provided in the literature in the distributed case.

In this paper we investigate the possibility of solving the distributed AHP problem in three alternative ways which are based on different approaches with respect to the eigenvector method and require less computational resources with respect to the approach in [24]. Specifically, the proposed methods are inspired to well known algorithms in the literature, i.e., Metropolis-Hastings Markov chains [19], Heat-Bath Markov chains [1] and formation control [9]. The aim of this paper is to identify strengths and weak points of such methodologies and to evaluate their performance on an experimental basis.

A. Paper Outline

The paper outline is as follows: Section 2 provides some preliminary concepts and definitions; Section 3 reviews the eigenvector method developed in [24], while Sections 4, 5 and 6 develop, respectively, three novel algorithms; Section 7 provides a simulation campaign aimed at comparing the effectiveness of the four methodologies; finally, we collect some conclusions and future work directions in Section 8.

[1] Unit of Automatic Control, Department of Engineering, Università Campus Bio-Medico di Roma, via Álvaro del Portillo 21, 00128, Rome, Italy.
[2]Consorzio Nazionale Interuniversitario per i Trasporti e la Logistica (NITEL), via Spalato 11, 00198, Rome Italy. This research was partially supported by SECUREWATER project, funded by the Ministry of Foreign Affairs and International Cooperation.
[3] ISC-CNR UoS "Sapienza",Piazzale Moro 5, 00185 Roma, Italy.
[*] Corresponding author, email: m.menci@unicampus.it
This research was partially supported by Project SECUREWATER, funded by the Italian Ministry of Foreign Affairs.

978-3-9524-2699-9 ©2018 EUCA

II. Notation and Preliminaries

A. General Notation and Definitions

We denote vectors via boldface letters, while matrices are shown with uppercase letters. Moreover, we denote by W_{ij} the (i,j)-th entry of a matrix W and by q_i the i-th entry of a vector $\mathbf{q}$. We denote by $\mathbf{1}_n$ a vector with n components, each equal to 1. Finally, we denote a normal variable with mean μ and standard deviation σ by $N(\mu,\sigma^2)$. The $n\times n$ matrix W is *irreducible* if there is no $n\times n$ invertible matrix P such that

$$P^{-1}WP = \begin{bmatrix} X & Y \\ 0 & Z \end{bmatrix}.$$

B. Kendall's Correlation Index

Let $\boldsymbol{a}\in\mathbb{R}^n$ and let $\boldsymbol{b}\in\mathbb{R}^n$ be a permutation of the elements in $\boldsymbol{a}$. Given two pairs of values (a_i,b_i) and (a_j,b_j), we say they are *concordant* if both $a_i > a_j$ and $b_i > b_j$ or if both $a_i < a_j$ and $b_i < b_j$; similarly the pairs are *discordant* if $a_i > a_j$ and $b_i < b_j$ or if $a_i < a_j$ and $b_i > b_j$. If $a_i = a_j$ or $b_i = b_j$ the pairs are neither concordant nor discordant. The *Kendall's correlation index* [16] τ is defined as $\tau = \frac{|\mathscr{C}|-|\mathscr{P}|}{n(n-1)/2}$, where $\mathscr{C}$ and $\mathscr{P}$ are the sets of concordant and discordant pairs (a_i,b_i) and (a_j,b_j), respectively. We point out that τ can be regarded as the degree of shuffling of $\boldsymbol{b}$ with respect to $\boldsymbol{a}$; in fact, for τ equal to one the two vectors are identical, while for $\tau=-1$ $\boldsymbol{b}$ is in reverse order with respect to $\boldsymbol{a}$ and for $\tau\approx 0$ the two vectors are independent.

C. Graph Theory

Let $G=\{V,E\}$ be a *graph* with n nodes $V=\{v_1,\ldots,v_n\}$ and e edges $E\subseteq V\times V$, where $(v_i,v_j)\in E$ captures the existence of a link from node v_i to node v_j. A graph is said to be *undirected* if $(v_i,v_j)\in E$ whenever $(v_j,v_i)\in E$, and is said to be *directed* otherwise. In the following we will consider only undirected graphs. A graph is *connected* if for each pair of nodes v_i,v_j there is a path over G that connects them. Let the neighborhood $\mathscr{N}_i$ of a node v_i be the set of nodes v_j that are connected to v_i via an edge (v_j,v_i) in E. The *degree* d_i of a node v_i is the number of its incoming edges[1], i.e., $d_i=|\mathscr{N}_i|$. The *adjacency matrix* A of a graph $G=\{V,E\}$ with n nodes is the $n\times n$ matrix such that $A_{ij}=1$ if $(v_j,v_i)\in E$ and $A_{ij}=0$ otherwise; moreover, the $n\times n$ *degree matrix* D is an $n\times n$ diagonal matrix such that $D_{ii}=d_i$, for all $i\in\{1,\ldots,n\}$. The $n\times n$ *Laplacian matrix* of G is $L=D-A$; since, by construction, the rows of L sum to zero, it can be noted that L is singular. The *diameter* δ of a graph G is the length of the longest among the minimum paths between any pair of nodes. We say an $n\times n$ matrix W *has the same structure* as a graph $G=\{V,E\}$ with n nodes if $A_{ij}=0$ implies $W_{ij}=0$. It can be shown that, in the case of undirected graphs, the fact W is irreducible corresponds to the fact that W has the same structure as a connected graph G.

D. Markov Chains

Let a graph $G=\{V,E\}$ with n nodes; a (time-homogeneous) Markov chain [10], [11] is a dynamic system in the form $\boldsymbol{p}(k+1)=H^T\boldsymbol{p}(k)$, where $\boldsymbol{p}(0)\in\mathbb{R}^n$ is a *probability distribution*, i.e., it has just nonnegative entries and satisfies $\mathbf{1}_n^T\boldsymbol{p}(0)=1$, and the *transition probability matrix* H is such that $H_{ij}\in[0,1]$ for all $i,j\in\{1,\ldots,n\}$ and $H\mathbf{1}_n=\mathbf{1}_n$. A Markov chain is said to be irreducible if H is irreducible, i.e., if H has the same structure as a connected graph G. For each state variable i, the associated *period* is $h=\gcd\{n\in\mathbb{N}_+\,|\,(H^n)_{ii}>0\}$, where gcd is the greatest common divisor. From the definition, it follows that a sufficient condition for the period of the i-th variable to be $h=1$ is that $H_{ii}>0$. A Markov chain is said to be *aperiodic* if all the state variables have period equal to one.

Remark 1: Note that, if the Markov chain is irreducible, then all the states have the same period (see, for instance [10], [11] and references therein). Therefore, a sufficient aperiodicity condition is that matrix H has at least one nonzero diagonal entry.

Remark 2: If a Markov chain is irreducible but not aperiodic, and H has a vector $\boldsymbol{f}$ as its left dominant eigenvector, i.e., $H^T\boldsymbol{f}=\boldsymbol{f}$, it can be easily noted that for any $\alpha\in(0,1)$ the matrix $H^*=\alpha H+(1-\alpha)I$ is aperiodic and it holds $(H^*)^T\boldsymbol{f}=\alpha H^T\boldsymbol{f}+(1-\alpha)\boldsymbol{f}=\alpha\boldsymbol{f}+(1-\alpha)\boldsymbol{f}=\boldsymbol{f}$, i.e., H^* has the same left dominant eigenvector as the original matrix H.

A Markov chain is said to have a *limiting distribution* $\boldsymbol{p}_\infty$ if, for all probability distributions $\boldsymbol{p}(0)$ it holds $\lim_{k\to\infty}(H^T)^k\boldsymbol{p}(0)=\boldsymbol{p}_\infty$. Note that, in general, a Markov chain might not have a limiting distribution; a necessary and sufficient condition for its existence is that the Markov chain is irreducible and aperiodic.

The following remark shows that, in the general case when the initial condition is not a probability distribution vector, any Markov chain is *sum-preserving*.

Remark 3: Let us consider a Markov chain with transition probability matrix H, a vector $\boldsymbol{p}(0)\in\mathbb{R}^n$ with nonnegative entries and such that $\mathbf{1}_n^T\boldsymbol{p}(0)\notin\{0,1\}$, and a vector $\boldsymbol{p}'(0)=\boldsymbol{p}(0)/\mathbf{1}_n^T\boldsymbol{p}(0)$. Moreover, let $\boldsymbol{p}(k)$ and $\boldsymbol{p}'(k)$ be the state of the Markov chain when the initial condition is $\boldsymbol{p}(0)$ and $\boldsymbol{p}'(0)$, respectively. At each time step k it holds

$$\boldsymbol{p}(k)=(M^T)^k\boldsymbol{p}(0)=\left(\mathbf{1}^T\boldsymbol{p}(0)\right)(M^T)^k\boldsymbol{p}'(0)=\left(\mathbf{1}^T\boldsymbol{p}(0)\right)\boldsymbol{p}'(k),$$

and since $\boldsymbol{p}'(k)$ is a distribution for all times k, it follows that $\mathbf{1}^T\boldsymbol{p}(k)=\mathbf{1}^T\boldsymbol{p}(0),\quad\forall k=0,1,\ldots$.

A consequence of the above remark is that, if a Markov chain has a limiting distribution $\boldsymbol{p}_\infty$, then in the general case where $\boldsymbol{p}(0)$ is not a distribution vector the state asymptotically converges to a vector $\widehat{\boldsymbol{p}}_\infty=(\mathbf{1}^T\boldsymbol{p}(0))\boldsymbol{p}_\infty$, i.e., to a vector in the span of the limiting distribution $\boldsymbol{p}_\infty$ with $\mathbf{1}^T\widehat{\boldsymbol{p}}_\infty=\mathbf{1}^T\boldsymbol{p}(0)$.

III. Problem Statement

Let us consider a set of n agents, interconnected by an undirected and connected graph $G=\{V,E\}$, and suppose

[1] Over undirected graphs, for each node v_i the number of its incoming and outgoing edges coincide.

that each agent is characterized by an unknown utility or relevance $w_i > 0$. Suppose further that each agent v_i is provided with just relative information regarding the ratios between its utility w_i and the utility w_j of each of its neighbors $v_j \in \mathcal{N}_i$ over G. In particular, let us assume that each ratio is perturbed by a multiplicative error or bias $\varepsilon_{ij} > 0$; in other words, for each neighbor $v_j \in \mathcal{N}_i$, the i-th agent v_i knows just $S_{ij} = \frac{w_i}{w_j}\varepsilon_{ij}$. In particular, as implicitly done in [24], we assume that there is no vector $\boldsymbol{f} \in \mathbb{R}^n$ such that $\varepsilon_{ij} = f_i/f_j$ for all $(v_i, v_j) \in E$; in other terms, the perturbations ε_{ij} represent a distortion of the utilities and there is no obvious way to compute a vector $\widetilde{\boldsymbol{w}}$ that satisfies $\widetilde{w}_i/\widetilde{w}_j = \varepsilon_{ij} w_i/w_j$ for all links $(v_i, v_j) \in E$.

Let S be the $n \times n$ matrix collecting the overall information available to the agents; we assume that, for all $(v_i, v_j) \in E$, it holds $S_{ji} = S_{ij}^{-1}$, i.e., we assume that $\varepsilon_{ji} = \varepsilon_{ij}^{-1}$. Such an assumption is common practice in the AHP literature [3], [6], [25]. The aim of each agent v_i is to compute an estimate $\widetilde{w}_i$ for its utility w_i. Note that, since the agents are provided with ratios of utilities, we assume that such utilities are defined up to a multiplicative scaling factor that is the same for all the agents. In the following, we denote by $\boldsymbol{w} \in \mathbb{R}^n$ and $\widetilde{\boldsymbol{w}} \in \mathbb{R}^n$ the stack of all w_i and $\widetilde{w}_i$, respectively.

IV. Sparse Eigenvector Method

The Sparse Eigenvector Method (SEM), developed in [24], is an effective distributed methodology to solve the AHP problem in the case of a matrix S having the structure of a connected and undirected graph $G = \{V, E\}$. The main idea behind the approach is that, in the nominal case where all perturbations $\varepsilon_{ij} = 1$, the dominant eigenvector of matrix $D^{-1}S$ (we recall that D is a diagonal matrix having the degree d_i of the i-th node at its i-th diagonal entry) is indeed the desired utility vector $\boldsymbol{w}$, while the dominant eigenvalue associated to it is equal to one, i.e., it holds $D^{-1}S\boldsymbol{w} = \boldsymbol{w}$.

As demonstrated in [24], the dominant eigenvalue of $D^{-1}S$ is equal to one if and only if the perturbations are not present, while it is in general different from one. Following the path of the traditional eigenvector method for the complete information case [25], the algorithm in [24] aims at letting the agents compute the dominant eigenvector of $D^{-1}S$, and specifically each agent aims at computing the corresponding component of the eigenvector. This is done by implementing a power iteration [20], which corresponds to a distributed algorithm given the sparse nature of matrix $D^{-1}S$. However, since in general the dominant eigenvalue of $D^{-1}S$ is not equal to one, a naive power iteration (i.e., without normalization) would either converge to zero or diverge; in order to address this challenge, the algorithm in [24] aims at letting the agents compute also an estimate of the dominant eigenvalue of $D^{-1}S$, which is used as a normalizing factor in the power iteration. In more detail, the agents have knowledge on an upper bound $\widetilde{\delta}$ of the graph diameter δ and execute a max-consensus procedure which is reinitialized at time steps that are multiple of $\widetilde{\delta}$, in order to reach an agreement on the estimate of the dominant eigenvalue as of $\widetilde{\delta}$ steps earlier.

Algorithm 1 Sparse Eigenvector Method. The pseudocode represents the point of view of the i-th agent.

```
procedure $SEM(\widetilde{\delta})$
    ▷ Initial Condition
    $\widetilde{w}_i(0) =$ random positive real number
    $y_i(0) = \ell_i(0) = h_i(0) = 1$
    ▷ Synchronous Iteration
    $\widetilde{w}_i(k+1) = \frac{1}{h_i(k)d_i} \sum_{j \in \mathcal{N}_i} S_{ij}\widetilde{w}_j(k)$
    $y_i(k+1) = \frac{1}{\widetilde{w}_i(k)d_i} \sum_{j \in \mathcal{N}_i} S_{ij}\widetilde{w}_j(k)$
    $\ell_i(k+1) = \begin{cases} y_i(k), & \text{if } \mathrm{mod}(k, \widetilde{\delta}) = 0, \\ \max_{v_j \in \mathcal{N}_i \cup \{v_i\}} \{\ell_j(k)\}, & \text{otherwise.} \end{cases}$
    $h_i(k+1) = \begin{cases} \ell_i(k), & \text{if } \mathrm{mod}(k, \widetilde{\delta}) = 0, \\ h_i(k), & \text{otherwise.} \end{cases}$
end procedure
```

The pseudocode of the algorithm developed in [24] is given in Algorithm 1. Specifically, Algorithm 1 is initialized with an estimate $h_i(0) = 1$ for the dominant eigenvalue that is the same for all agents; moreover, each agent relies on $\widetilde{w}_i(k)$ in order to compute a quantity $y_i(k)$ that tends to the dominant eigenvalue of $D^{-1}S$ as k approaches infinity. While doing so, the agents execute several max-consensus procedures, which are re-initialized every $\widetilde{\delta}$ steps, so that every time a max-consensus procedure terminates at a given time instant k, they compute an updated h_i that is the same for all agents and corresponds to the greatest among the values $y_i(k - \widetilde{\delta})$ over all agents i. By repeatedly doing so, the agent i is able to compute the i-th component of a finite and nonzero vector in the span of the dominant eigenvector of $D^{-1}S$, as well as the dominant eigenvalue of $D^{-1}S$.

A. Discussion

We point out that Algorithm 1, although effective, has several drawbacks: (1) the agents need to know a global parameter such as an upper bound $\widetilde{\delta}$ of the network diameter δ (e.g., computing it via the approach in [22]); (2) each agent needs to let four state variables evolve in parallel, relying at each step on the state variables of their neighbors; (3) the agents need to run a max-consensus iteration in parallel with the main iteration, and such a max-consensus procedure has to be reinitialized at prescribed time instants; (4) no guarantee on the magnitude of $\widetilde{\boldsymbol{w}}$ is given, and the agents might need to further interact to calculate a normalization of the result. However, the main advantage of Algorithm 1 is that, by computing the dominant eigenvalue of $D^{-1}S$, the agents become aware of valuable meta-information regarding the degree of inconsistency of the available data.

V. Metropolis-Hastings Method

In this section we inspect the possibility to resort to a *Metropolis-Hastings* (MH) Markov chains [14], [19] to solve the AHP problem in a distributed way. We point out that Metropolis-Hastings Markov chains are quite a popular tool in the literature to sample from a known discrete distribution (or, at least, a distribution for which probability ratios are known). However, to the best of our knowledge, the adoption of such a tool in order to rank a set of alternatives based on perturbed ratios can be regarded as an original contribution of this paper.

Let an $n \times n$ matrix M having the same structure as the graph G, such that, for all $i, j \in \{1, \ldots, n\}$ it holds

$$M_{ij} = \begin{cases} \frac{1}{d_i} \min\left\{1, \frac{d_i S_{ji}}{d_j}\right\}, & \text{if } (v_i, v_j) \in E \\ 1 - \frac{1}{d_i} \sum_{h \in \mathcal{N}_i} \min\left\{1, \frac{d_i S_{hi}}{d_h}\right\}, & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

It can be easily noted that, by construction, it holds $M_{ij} \in [0,1]$ and $M\mathbf{1}_n = \mathbf{1}_n$; hence, matrix M is the transition probability matrix of a Markov chain in the form

$$\boldsymbol{p}(k+1) = M^T \boldsymbol{p}(k). \tag{2}$$

The standard Metropolis-Hastings algorithm corresponds to the above Markov chain, in the special case when all terms $S_{ij} = w_i/w_j$, i.e., when no perturbation is considered. In such a special case, assuming that G is undirected and connected and that all $w_i > 0$, as shown in [14], [19], the Markov chain in Eq. (2) is aperiodic and has $\boldsymbol{w}$ as its limiting distribution.

Note that if the initial condition $\boldsymbol{p}(0)$ of the Markov chain is a probability distribution vector, then at each time k the state $\boldsymbol{p}(k)$ is a probability distribution vector. Conversely, as noted in Remark 3, when $\boldsymbol{p}(0)$ is not a probability distribution vector but it has nonnegative entries and nonzero sum, the state converges to a vector in the span of the limiting distribution vector having sum of the entries equal to the sum of the entries of $\boldsymbol{p}(0)$. This means that, differently from the SEM approach, if the agents select random positive initial conditions, there is a guarantee that the sum of the entries of the estimated utilities will remain constant during the execution of the algorithm. Moreover, since the algorithm exhibits asymptotic convergence, if a normalized vector with sum equal to one is required, it is more convenient to let the agents cooperate in order to select an initial distribution beforehand, rather than performing normalization after the execution of the asymptotic algorithm is truncated. In order to do this, the agents need to execute an initialization phase where $\widetilde{\boldsymbol{w}}(0)$ is constructed. Among other possible choices, a feasible approach is to elect a *leader* via max-consensus or other techniques (see [2] and references therein for recent works on this topic) and then select $\widetilde{w}_i(0) = 1$ if node v_i is the leader and $\widetilde{w}_i(0) = 0$ otherwise.

Let us now discuss the proposed algorithm when perturbations are present; the pseudocode is reported in Algorithm 2. As said above, the algorithm either requires the agents to choose an initial condition $\widetilde{w}_i(0)$ that corresponds to a distribution vector $\widetilde{\boldsymbol{w}}(0)$ or a positive random number.

Moreover, in order to compute the entries M_{ij} according to Eq. (1), the agents need to know the degree d_j of their neighbors $v_j \in \mathcal{N}_j$. Let $\alpha \in (0,1)$ be a parameter known to all agents and let us consider the stacked dynamics of all the agents, i.e., $\widetilde{\boldsymbol{w}}(k+1) = (\alpha M^T + (1-\alpha)I)\widetilde{\boldsymbol{w}}(k)$. As discussed in Remark 2, the dynamic matrix of the above discrete-time system is a convex combination of M^T and the identity matrix, hence $(\alpha M^T + (1-\alpha)I)$ is aperiodic. Moreover, since the graph G is connected for hypothesis, it follows that $(\alpha M^T + (1-\alpha)I)$ is irreducible. Therefore, the proposed dynamics corresponds to a Markov chain having a limiting distribution, which is given by the dominant left eigenvector of M.

Algorithm 2 Distributed Metropolis-Hastings Algorithm. The pseudocode represents the point of view of the *i*-th agent.

```
procedure $MH(\widetilde{\delta}, \alpha, normalize)$
    ▷ Initialization
    send $d_i$ to all neighbors;
    receive $d_j$ from each neighbor $v_j \in \mathcal{N}_i$;
    ▷ Choose initial condition $\widetilde{\boldsymbol{w}}(0)$
    if normalize then
        elect a leader, e.g., via max-consensus($\widetilde{\delta}$)
        select $\widetilde{w}_i(0) = 1$ if $i$ leader $w_i(0) = 0$, otherwise
    else
        $\widetilde{w}_i(0) =$ random positive real number
    end
    ▷ Synchronous Iteration
    $\widetilde{w}_i(k+1) = (\alpha M_{ii} + 1 - \alpha)\widetilde{w}_i(k) + \alpha \sum_{j \in \mathcal{N}_i} M_{ji}\widetilde{w}_j(k)$
end procedure
```

A. Discussion

The approach discussed in this section has several advantages with respect to Algorithm 1: (1) except for an initialization in order to select a starting distribution, the algorithm requires less memory and communication among the agents and a less complex interaction, since only a standard discrete-time linear iteration is executed; (2) the vector computed is intrinsically normalized (provided that the initial state is a distribution) or, in any case, the sum of the entries of the estimated utilities is constant over time. However, the main drawback of this approach is that the parameter α, which is used to guarantee the existence of a limiting distribution, has to be known to all agents; moreover, if a distribution is required, the agents need to know a global parameter (e.g., the network diameter, in order to execute a leader election procedure).

VI. Heat-Bath Method

The *Heat-Bath* (HB) method, similarly to the Metropolis-Hastings approach, is yet another popular way to design a Markov chain that has a desired limiting distribution (see among others [1], [8]). Note that the adoption of a Heat-Bath Markov chain for the distributed ranking can be regarded as an original contribution of this paper.

Let an $n \times n$ matrix Q having the same structure as the graph G, such that, for all $i, j \in \{1, \ldots, n\}$ it holds

$$Q_{ij} = \begin{cases} \frac{\gamma}{1+S_{ij}}, & \text{if } (v_i, v_j) \in E \\ 1 - \gamma \sum_{h \in \mathcal{N}_i} \frac{1}{1+S_{ih}}, & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where γ is a global parameter known to all agents and such that $\gamma < 1/\max_{i=1,\ldots,n} d_i$. It can be easily noted that, by construction, it holds $Q_{ij} \in [0, \gamma] \subseteq [0,1]$ and $Q\mathbf{1}_n = \mathbf{1}_n$; hence, matrix Q is the transition probability matrix of a

Markov chain. Moreover, we observe that it holds

$$\sum_{j\in\mathcal{N}_i} Q_{ij} = \gamma \sum_{j\in\mathcal{N}_i} \frac{1}{1+S_{ij}} < \frac{1}{\max\{d_i\}} \sum_{j\in\mathcal{N}_i} 1 = \frac{d_i}{\max\{d_i\}} \leq 1;$$

hence, we have that, for all $i \in \{1,\dots,n\}$ it holds $Q_{ii} > 1 - d_i/\max\{d_i\} \geq 0$. Therefore, as discussed in Remark 1, the Heat-Bath Markov chain, in the general case, is intrinsically aperiodic and since G is connected we conclude that such a dynamics always reaches a limiting distribution. As a consequence, differently from the Metropolis-Hastings case, there is no need to implement convex combinations with the identity matrix. We point out that, in the nominal case where all terms $\varepsilon_{ij} = 1$, the Heat-Bath Markov chain is known[2] to have $\boldsymbol{w}$ as its dominant left eigenvector [1], [8]; therefore, when no perturbation is present, the Heat-Bath Markov chain has a limiting distribution that corresponds to $\boldsymbol{w}$.

Algorithm 3 Distributed Heat-Bath Algorithm. The pseudocode represents the point of view of the *i*-th agent.

```
procedure $HB(\gamma, \widetilde{\delta}, normalize)$
    ▷ Initialization
    ▷ Choose initial condition $\widetilde{\boldsymbol{w}}(0)$
    if $normalize$ then
        elect a leader, e.g., via max-consensus($\widetilde{\delta}$)
        select $\widetilde{w}_i(0) = 1$ if $i$ leader $w_i(0) = 0$, otherwise
    else
        $\widetilde{w}_i(0) =$ random positive real number
    end
    ▷ Synchronous Iteration
    $\widetilde{w}_i(k+1) = \sum_{j\in\mathcal{N}_i\cup\{i\}} Q_{ji}\widetilde{w}_j(k)$
end procedure
```

The pseudocode of the proposed Heat-Bath algorithm is reported in Algorithm 3. Similarly to the MH algorithm, if an estimated utility vector having sum of the entries equal to one is required, a feasible path is to resort to leader election; otherwise, if the agents select random positive initial condition then the sum of the entries is preserved as discussed in Remark 3.

A. Discussion

Similarly to the MH algorithm, the Heat-Bath approach has several benefits with respect to the eigenvector method, i.e., lower memory and communication requirements. Moreover, differently from MH, there is no need to force aperiodicity and the agents need not to know the degrees of their neighbors. However, just like MH, in case there is a need to select an initial condition, the agents must be aware of a global parameter such as the diameter.

VII. Sparse Logarithmic Least Squares Method

In this section we develop an algorithm, namely *Sparse Logarithmic Least Squares* (SLLS) Method that extends the *Logarithmic Least Squares* (LLS) Method developed in [3], [6] for the complete information case; such an algorithm is based on Fax and Murray's formation control algorithm [9], [21]. Specifically, within the SLLSM algorithm, the agents aim at finding a logarithmic least squares approximation $\widetilde{\boldsymbol{w}}$ of the vector $\boldsymbol{w}$ that is the stack of the utilities w_i. In other words, the problem corresponds to finding the vector $\widetilde{\boldsymbol{w}}$ such that

$$\widetilde{\boldsymbol{w}} = \arg\min_{\boldsymbol{q}\in\mathbb{R}^n_+} \left\{ \sum_{i=1}^{n} \sum_{j\in\mathcal{N}_i}^{n} \left(\ln(S_{ij}) - \ln\left(\frac{q_i}{q_j}\right)\right)^2 \right\}; \qquad (4)$$

notice that, since the above function is convex, such a vector always exists. We now provide a theorem that is the basis for the extension of the LLSM method to a sparse information context[3].

Theorem 1: Let us consider a set of n agents, interconnected by an undirected and connected graph $G = \{V, E\}$, and let S be the $n \times n$ matrix collecting the overall information available to the agents. Let us define $s_i = \sum_{j\in\mathcal{N}_i}^{n} \ln(S_{ij})$ and let $\boldsymbol{s}$ be the stack of all s_i. Moreover, let $\boldsymbol{r}^* \in \mathbb{R}^n$ be the vector that satisfies $L\boldsymbol{r}^* = \boldsymbol{s}$, where L is the laplacian matrix corresponding to the graph G. It holds $\widetilde{\boldsymbol{w}} = \exp(\boldsymbol{r}^*)$, where by $\exp(\boldsymbol{r}^*)$ we mean the component-wise exponentiation with base equal to e of the vector $\boldsymbol{r}^*$.

Proof: Let $r_i = \ln(q_i)$; Eq. (4) can be rearranged as

$$\begin{aligned} \widetilde{\boldsymbol{w}} &= \exp\left(\arg\min_{\boldsymbol{r}\in\mathbb{R}^n} \left\{ \sum_{i=1}^{n} \sum_{j\in\mathcal{N}_i}^{n} (\ln(S_{ij}) - r_i + r_j)^2 \right\} \right) = \\ &= \exp\left(\arg\min_{\boldsymbol{r}\in\mathbb{R}^n} \{ f(\boldsymbol{r}) \} \right). \end{aligned} \qquad (5)$$

According to Eq. (5), $\boldsymbol{y}^*$ is the solution of a convex and unconstrained minimization problem. Therefore the vector $\boldsymbol{r}^*$ that minimizes $f(\boldsymbol{r})$ is such that

$$\left.\frac{\partial f(\boldsymbol{r})}{\partial r_i}\right|_{\boldsymbol{r}=\boldsymbol{r}^*} = 0, \quad \forall i = 1,\dots,n,$$

which corresponds to $\sum_{j\in\mathcal{N}_i}^{n}(r_i^* - r_j^*) = s_i$, for all $i = 1,\dots,n$. Stacking the above equation for all agents i we get $L\boldsymbol{r}^* = \boldsymbol{s}$. This completes our proof. ■

We point out that, since the Laplacian matrix L is singular by definition, the vector $\boldsymbol{r}^*$ can not be computed by matrix inversion. A possible way to compute $\boldsymbol{r}^*$ is thus to resort to an asymptotic algorithm, whose pseudocode is reported in Algorithm 4.

Algorithm 4 Sparse Logarithmic Least Squares Method. The pseudocode represents the point of view of the *i*-th agent.

```
procedure $SLLSM(\theta)$
    ▷ Initialization
    $r_i(0) =$ random positive real number
    $\widetilde{w}_i(0) = \exp(r_i(0))$
    $s_i = \sum_{j\in\mathcal{N}_i}^{n} \ln(S_{ij})$
    ▷ Synchronous Iteration
    $r_i(k+1) = r_i(k) + \theta \sum_{j\in\mathcal{N}_i} (r_j(k) - r_i(k)) + \theta s_i$

    $\widetilde{w}_i(k+1) = \exp(r_i(k+1))$
end procedure
```

[2] This is a consequence of the fact that the Heat-Bath Markov chain satisfies the *detailed balance* $w_i Q_{ij} = w_j Q_{ji}$ for all $(v_i, v_j) \in E$; the interested reader is referred to [1], [8] for further information.

[3] We point out that some of the ideas of this theorem are similar to the results in [9], [21]; the theorem is given with a proof for the sake of self-containedness and clarity.

Remark 4: We point out that Algorithm 4 and, in particular, the dynamics chosen for $r_i(k)$, amounts to a discrete-time version of the continuous time *formation control* algorithm developed by Fax and Murray [9], [21], which has the form $\dot{\boldsymbol{r}}(t) = -L\boldsymbol{r}(t) + \boldsymbol{s}$; over connected undirected graphs, such a continuous time equation is known to converge to a vector $\boldsymbol{r}^*$ that satisfies $L\boldsymbol{r}^* = \boldsymbol{s}$. Let us consider a discrete-time setting and let us write down the stack of the dynamics for all the agents within Algorithm 4, which is $\boldsymbol{r}(k+1) = (I - \theta L)\boldsymbol{r}(k) + \theta\boldsymbol{s}$. The above dynamics is indeed a discrete-time average consensus dynamics (plus a constant exogenous input that does not affect stability); such a dynamics is known to converge asymptotically [21] over connected undirected graphs if the parameter θ, which can be regarded as a sampling time, satisfies $\theta \leq 1/\max_{i=1,\ldots,n} d_i$.

A. Discussion

We notice that Algorithm 4 has several advantages with respect to Algorithm 1: (1) instead of approximating the unknown utilities with the dominant eigenvector of $D^{-1}S$, a procedure that has raised some criticism in the literature, the solution computed by Algorithm 4 represents a clear log-quadratic minimization of the error between the perturbed ratios S_{ij} and the ratios $\widetilde{w}_i/\widetilde{w}_j$; (2) the algorithm requires less memory and communication among the agents and a less complex interaction, since only a standard discrete-time average consensus iteration (with an exogenous constant input) is executed. We point out that Algorithm 4, being essentially an asymptotic average-consensus algorithm, can be the base for several extensions, such as distributed stopping criteria [18], [28], finite-time [23], [27] or asynchronous implementations [4], [17]. However, Algorithm 4 has some drawbacks: (1) there is no control on the norm of the estimate, which might need to undergo normalization; (2) each agent needs to known the same global parameter θ.

VIII. Experimental Comparison

In this section we analyze in an experimental way the performances of the four methodologies discussed in this paper, considering random networks of different typologies with $n = 50$ nodes. In more detail, we consider a Watts-Strogatz small-world network with 3 links per node and rewiring probability 30%, a Barabási-Albert scale free network with 3 preferential attachments per node, a random geometric network (i.e., a network where the nodes are generated at random in the unit square and are connected if their Euclidean distance is less than a radius ρ) with $\rho = 0.3$ and an Erdős-Renyí random network model with connection probability 30%. In order to evaluate the performance of the four methodologies, we consider log-normal perturbation terms $\varepsilon_{ij} = exp(\phi_{ij})$, where $\phi_{ij} = N(0, \sigma^2)$ is sampled from a Gaussian distribution with zero mean and standard deviation σ; we consider different values of $\sigma \in [0,1]$ and for each choice of σ we show the results over $m = 50$ random networks. Note that we select random terms ϕ_{ij} for $i < j$, while we set $\phi_{ji} = -\phi_{ij}$, in order to obtain $\varepsilon_{ji} = \varepsilon_{ij}^{-1}$.

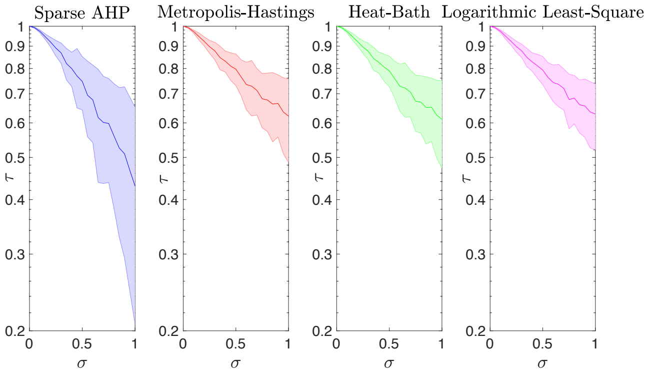

Fig. 1. Kendall correlation coefficient between the nominal and perturbed ranking, plotted against the standard deviation of the log-normal perturbations in the case of Small-World networks with $n = 50$ nodes, 3 links per node and rewiring probability equal to 30%. Results are the average over $m = 50$ runs.

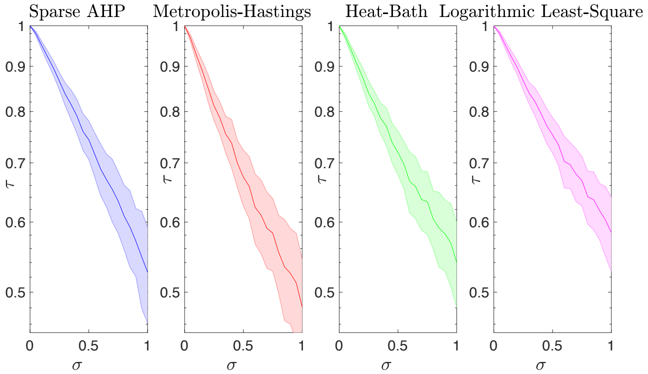

Fig. 2. Kendall correlation coefficient between the nominal and perturbed ranking, plotted against the standard deviation of the log-normal perturbations in the case of Scale-Free networks with $n = 50$ nodes and 3 preferential attachments per node. Results are the average over $m = 50$ runs.

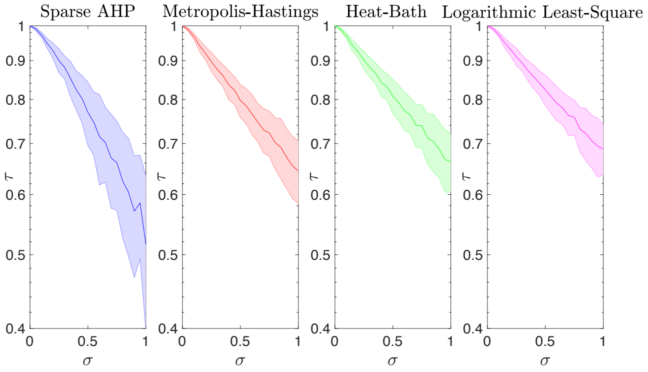

Fig. 3. Kendall correlation coefficient between the nominal and perturbed ranking, plotted against the standard deviation of the log-normal perturbations in the case of Random Geometric networks with $n = 50$ nodes and $\rho = 0.3$. Results are the average over $m = 50$ runs.

In Figures 1–4 we show, plotted against the standard deviation of the terms ϕ_{ij}, the Kendall correlation coefficient τ between the ranking obtained via the nominal utility vector and the one obtained based on the approximation

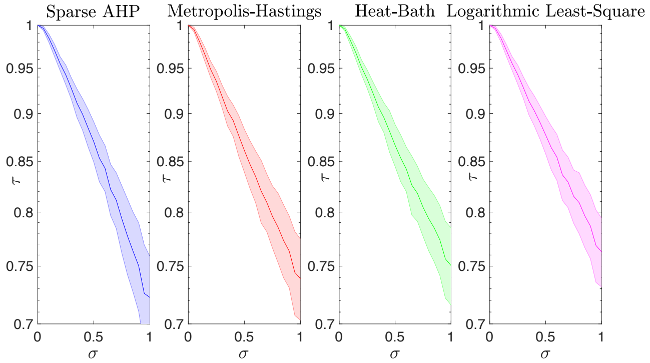

Fig. 4. Kendall correlation coefficient between the nominal and perturbed ranking, plotted against the standard deviation of the log-normal perturbations in the case of Erdős-Renyí networks with $n = 50$ nodes and connection probability equal to 30%. Results are the average over $m = 50$ runs.

$\widetilde{\boldsymbol{w}}$ computed via the different algorithms. According to the figures, as the perturbation magnitude grows, the Sparse AHP methodology exhibits a comparatively large degree of shuffling with respect to the nominal ranking; conversely, the other methodologies have remarkably better results, especially for large perturbation. We point out that the MH, HB and SLLS methodologies have comparable results, although the SLLS approach has slightly better correlation with the nominal ranking than the other approaches, especially for large perturbations.

IX. Conclusions and Future Work

In this paper we have provided three different methodologies to solve the Sparse Analytic Hierarchy Process problem in a distributed way, based on popular algorithms in the literature such has Metropolis-Hastings Markov chains, Heat-Bath Markov chains and formation control. Future work will be aimed at providing a formal characterization of the performance of such algorithms as a function of the perturbations, as well as to identify topological conditions that can be used to select the most appropriate methodology depending on the structure of the network.

References

[1] D. Achlioptas, M. Molloy, C, Moore, and F. Van Bussel, “Rapid mixing for lattice colourings with fewer colours", *Journal of Statistical Mechanics: Theory and Experiment*, vol. 10, P10012, 2005.

[2] K. Altisen, A. Cournier, S. Devismes, A. Durand, and F. Petit, “Self-stabilizing leader election in polynomial steps", *Information and Computation*, vol. 254, pp. 330–366, 2017.

[3] J. Barzilai, W. D. Cook, and B. Golany, “Consistent weights for judgements matrices of the relative importance of alternatives", *Operations research letters*, vol. 6, n. 3, pp. 131–134, 1987.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms", *IEEE/ACM Transactions on Networking*, vol. 14, pp, 2508–2530, 2006.

[5] F. J. Carmone, A. Kara, and S. H. Zanakis, “A Monte Carlo investigation of incomplete pairwise comparison matrices in AHP", *European Journal of Operational Research*, vol. 102, no. 3, pp. 538–553, 1997.

[6] G. B. Crawford, “The geometric mean procedure for estimating the scale of a judgement matrix", *Mathematical Modelling*, vol. 9, no. 3-5, pp. 327–334, 1987.

[7] J. S. Dyer, “Remarks on the Analytic Hierarchy Process", *Management science*, vol. 36, no. 3, pp. 249–258, 1990.

[8] M. Dyer, C. Greenhill, and M. Ullrich, “Structure and Eigenvalues of Heat-Bath Markov Chains", *Linear Algebra and its Applications*, vol. 454, pp. 57–71, 2014.

[9] J. A. Fax and R. M. Murray, “Information flow and cooperative control of vehicle formations", *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[10] W. R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov chain Monte Carlo in practice*, CRC press, 1995.

[11] O. Häggström, *Finite Markov chains and algorithmic applications*, vol. 52, Cambridge University Press, 2002.

[12] P. T. Harker, “Alternative modes of questioning in the analytic hierarchy process", *Mathematical Modelling*, vol. 9, no. 3-5, pp. 353–360, 1987.

[13] P. T. Harker, “Incomplete pairwise comparisons in the Analytic Hierarchy Process", *Mathematical Modelling*, vol. 9, no. 11, pp. 837–848, 1987.

[14] W. K. Hastings, “Monte Carlo sampling methods using Markov chains and their applications", *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[15] H. Ishii and R. Tempo, “The pagerank problem, multiagent consensus, and web aggregation: A systems and control viewpoint", *IEEE Control Systems*, vol. 34, no. 3, pp. 34–53, 2014.

[16] M. G. Kendall, “A new measure of rank correlation", *Biometrika*, vol. 30 no. 1/2, pp. 81–93, 1938.

[17] A. Khosravi and Y. S. Kavian, “Broadcast gossip ratio consensus: Asynchronous distributed averaging in strongly connected networks", *IEEE Transactions on Signal Processing*, vol. 65, no. 1, pp. 119–129, 2017.

[18] N. E. Manitara and C. N. Hadjicostis, “Distributed stopping for average consensus in digraphs", *IEEE Transactions on Control of Network Systems*, 2017 (Article in press).

[19] N.Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines", *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[20] R. V. Mises and H. Pollaczek-Geiringer, “Praktische verfahren der gleichungsauflösung", *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 9, no. 2, pp, 152–164, 1929.

[21] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems", *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[22] G. Oliva, R. Setola, and C. N. Hadjicostis, “Distributed finite-time calculation of node eccentricities, graph radius and graph diameter", *Systems & Control Letters*, vol. 92, pp. 20–27, 2016.

[23] G. Oliva, R. Setola, and C. N. Hadjicostis, “Distributed finite-time average-consensus with limited computational and storage capability", *IEEE Transactions on Control of Network Systems*, vol. 4, no. 2, pp. 380–391, 2017.

[24] G. Oliva, R. Setola, and A.Scala, “Sparse and distributed analytic hierarchy process", *Automatica*, vol. 85, pp. 211 – 220, 2017.

[25] T. L. Saaty, “A scaling method for priorities in hierarchical structures", *Journal of mathematical psychology*, vol. 15, no. 3, pp. 234–281, 1977.

[26] T. L. Saaty, “An exposition of the AHP in reply to the paper “remarks on the analytic hierarchy process", *Management science*, vol. 36, no. 3, pp. 259–268, 1990.

[27] L. Wang and F. Xiao, “Finite-time consensus problems for networks of dynamic agents", *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 950–955, 2010.

[28] P. Xie, K. You, and C. Wu, “How to stop consensus algorithms, locally?", *arXiv preprint arXiv:1703.05106*, 2017.