Automatica 134 (2021) 109924

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Distributed fixed point method for solving systems of linear algebraic equations☆

Dušan Jakovetić [a], Nataša Krejić [a], Nataša Krklec Jerinkić [a], Greta Malaspina [a,*], Alessandra Micheletti [b]

[a] Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia
[b] Department of Environmental Science and Policy - ESP, Università degli Studi di Milano, Via Saldini 50, 20133 Milano, Italy

ARTICLE INFO

Article history:
Received 10 January 2020
Received in revised form 4 June 2021
Accepted 18 August 2021
Available online xxxx

Keywords:
Distributed optimization
Systems of linear equations
Fixed point methods
Consensus
Kriging

ABSTRACT

We present a class of iterative fully distributed fixed point methods to solve a system of linear equations, such that each agent in the network holds one or several of the equations of the system. Under a generic directed, strongly connected network, we prove a convergence result analogous to the one for fixed point methods in the classical, centralized, framework: the proposed method converges to the solution of the system of linear equations at a linear rate. We further explicitly quantify the rate in terms of the linear system and network parameters. Next, we show that the algorithm provably works under time-varying directed networks provided that the underlying graph is connected over bounded iteration intervals, and we establish a linear convergence rate for this setting as well. A set of numerical results is presented, demonstrating practical benefits of the method over existing alternatives.

1. Introduction

The problem we consider is

$$Ay = b \tag{1}$$

where $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ and $b = [b_i] \in \mathbb{R}^n$ are given, and $y \in \mathbb{R}^n$ is the vector of the unknowns. The matrix A is assumed to be nonsingular, so that the problem has a unique solution. We also assume that the problem is solved in a distributed computational framework determined by a set of computational nodes which can communicate through a sequence of graphs. Let $A_i \in \mathbb{R}^{1\times n}$ and $b_i \in \mathbb{R}$ be the ith row of A and b respectively. It is assumed that each node i knows the corresponding A_i and b_i and needs to obtain the solution y^* through an iterative, distributed algorithm. This assumption is later relaxed in the sense that the number of nodes is $N \leq n$, and each node can hold several rows of the matrix A, see Section 3.1.

The considered problem is important as linear systems appear naturally in a number of control problems, like the estimation problems on graphs described in Barooah and Hespanha (2007), including localization problems, time synchronization and motion consensus and parameter identification in wireless sensor networks (Bolognani, Favero, Schenato, & Varagnolo, 2010). A particularly important example of application is Ordinary Kriging (OK) (Cortes, 2009; Cressie, 1993; Krige, 1951; Matheron, 1963), an optimal linear prediction technique of the expected value of a spatial random field. In Section 5 we will use a special case known as Simple Kriging as an example of application of our method. In Cortes (2009) a combination of Kriging and Kalman filters is applied to estimate a spatio-temporal random field, where at each iteration of the presented method a Kriging problem has to be solved distributedly by the network of agents.

There is a vast literature devoted to solving systems of linear equations in the conventional centralized environment (Greenbaum, 1997; Saad, 2003), as well as a number of results that cover parallelization of classical iterative methods which are applicable to the case of fully connected distributed environment, Frommer and Szyld (1992). Our interest in this paper is the class of fixed point methods (Greenbaum, 1997; Saad, 2003) and their extensions to the distributed framework, as described above. We develop a class of novel, fully distributed, iterative fixed point methods to solve (1), wherein each node can exchange messages only with those in its neighborhood in the communication graph,

☆ This work is supported by the BIGMATH project which has received funding by the European Union's Horizon 2020 research and innovation programme under the Marie Sk lodowska-Curie Grant Agreement no. 812912. The work of Jakovetić, Krejić and Krklec Jerinkić is partially supported by Serbian Ministry of Education, Science and Technological Development. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Julien M. Hendrickx under the direction of Editor Christos G. Cassandras.

* Corresponding author.
E-mail addresses: dusan.jakovetic@dmi.uns.ac.rs (D. Jakovetić), natasak@uns.ac.rs (N. Krejić), natasa.krklec@dmi.uns.ac.rs (N. Krklec Jerinkić), greta.malaspina@dmi.uns.ac.rs (G. Malaspina), alessandra.micheletti@unimi.it (A. Micheletti).

https://doi.org/10.1016/j.automatica.2021.109924

and each node obtains the estimate of the solution y^* of problem (1). Then (1) can be transformed into an equivalent fixed point problem

$$y = My + d, \tag{2}$$

and one can apply the Banach contraction principle and define the fixed point iterative method of the form $y^{k+1} = My^k + d$, for suitable choices of $M \in \mathbb{R}^{n\times n}$ and $d \in \mathbb{R}^n$ (see Section 2 for the details).

The sufficient and necessary condition for the convergence of such iterative sequence is $\rho(M) < 1$, where $\rho(M)$ is the spectral radius of M. Furthermore, a sufficient condition for the convergence of $\{y^k\}$ is given by $\|M\| < 1$ for an arbitrary matrix norm $\|\cdot\|$. Clearly, there is a number of suitable ways to define the iterative matrix M in such way that either $\rho(M) < 1$ or $\|M\| < 1$ for many matrix classes, like symmetric positive definite matrices, M-matrices, H-matrices, etc. (Berman & Plemmons, 1994). Typical methods are the Jacobi and Gauss–Seidel method as well as their modifications like Jacobi Overrelaxation (JOR), Successive Overrelaxation (SOR), Symmetric Successive Overrelaxation (SSOR) method and so on (Greenbaum, 1997; Saad, 2003). The convergence of fixed point methods is linear and the convergence factor is determined by the spectral radius or the norm of M. The main idea of relaxation methods is to introduce a parameter that reduces the norm (or the spectral radius) of the corresponding iterative matrix and ensures faster convergence.

There is a rich literature on parallelization of fixed point iterative methods, where the computational nodes communicate in an all-to-all fashion (Bertsekas & Tsitsiklis, 2015; Frommer & Mayer, 1989; Frommer & Szyld, 1992, 2000). In the case of very large dimensions one needs to split the computational effort between different computational nodes to speed up the algorithm. The total cost of solving the problem of interest is mainly dictated by the corresponding computational cost and the communication cost of exchanging messages between the parallelized nodes (processes) along iterations. Usually, major bottlenecks include waiting for the slowest node to complete an iteration, or latency incurred by the time to communicate a message. For this reason asynchronous methods, which allow for latency in communication and nonuniform distribution of computational work, are also considered, Frommer and Szyld (2000). The methods of this type are convergent under different communication latency conditions (Frommer & Szyld, 2000).

The framework we consider assumes a network of computational nodes which communicate through a generic directed graph, which can depend on time. Thus the results in Bertsekas and Tsitsiklis (2015), Frommer and Mayer (1989), Frommer and Szyld (1992, 2000) are not applicable. The same framework is also considered in Alaviani and Elia (2020), Anderson, Mou, Morse, and Helmke (2016), Liu, Morse, Nedić, and Basar (2017), Liu, Mou, and Morse (2018), Mou, Lin, Wang, Fullmer, and Morse (2016), Shi and Anderson (2016), Wang, Zhou, Mou, and Corless (2019), Xiao and Hu (2017), and a survey of the methods is presented in Wang, Mou, Lian, and Ren (2019). The focus of these methods is to ensure convergence of the local approximations to the global solution, in the presence of time-varying communication graphs. In the context of these algorithms, convergence is defined in two possible ways. In Mou et al. (2016), Xiao and Hu (2017) each node holds a local approximation of a subset of the variables and convergence of these local variables to the corresponding part of the solution is required. In Anderson et al. (2016), Liu et al. (2017, 2018), Shi and Anderson (2016), Wang et al. (2019) every node contains a vector of the same size as the unknown vector of the linear system, and the convergence of each local vector to the full solution in ensured. We are interested in the second scenario. The method presented in Frommer and Szyld (1992) is applicable to a general problem of the type (1) with loose restrictions on the matrix A and can be used to solve the linear least squares problem as well.

In this paper, we propose a novel distributed method to solve (1), which we refer to as DFIX (Distributed Fixed Point). DFIX assumes the same computational framework as Liu et al. (2017, 2018), Wang et al. (2019) but differs significantly from the above mentioned methods. We provide a canonical way to decentralize any fixed point method for solving linear systems and we extend the convergence theory of centralized fixed point methods to the distributed case in the sense of sufficient conditions. That is, we demonstrate that the condition $\|M\|_\infty < 1$ continues to be sufficient in the distributed environment, assuming that each network agent locally stores a vector of dimension n. The main convergence result is completely analogous to the centralized case — given an iterative matrix with infinity norm smaller than 1, the iterative sequence converges for an arbitrary starting point. The theory presented here thus covers a large class of linear systems. We prove R-linear convergence of DFIX under directed strongly connected networks and explicitly quantify the corresponding convergence factor in terms of network and linear system parameters. As detailed below, numerical simulations demonstrate advantages of DFIX over some state of the art methods.

With respect to the underlying graph, representing the connection among the computational agents, both the case when the graph is fixed (i.e., the connectivity among the nodes is the same at any time during the execution of the algorithm) and the case when the network can change at every iteration, are considered. In the fixed graph case we prove that convergence holds if the network is strongly connected, while in the time-varying case we give suitable assumptions over the sequence of networks. The time-independent case is a particular case of the time-varying case but for the sake of clarity we first present and analyze the algorithm assuming the network is fixed and then generalize the analysis to the time-varying case.

Any system of linear equation (1) with symmetric matrix A can be considered as the first order optimality condition of an unconstrained optimization problem with cost function $\frac{1}{2}x^TAx - b^Tx$. It is therefore of interest to compare the approach of solving (1) applying some distributed optimization method (Li & Qu, 2017; Nedic, Olshevsky, & Shi, 2016; Shi, Ling, Wu, & Yin, 2015; Sundararajan, Van Scoy, & Lessard, 2019) to the minimization of the quadratic function $\frac{1}{2}x^TAx - b^Ttx$ with DFIX. We thus compare computational and communication costs of DFIX with the state-of-the-art optimization method from these papers and show that the computational costs with DFIX are significantly lower, while the communication costs are comparable or go in favor of DFIX, depending on the connectivity of the underlying graph. Thus the numerical efficiency of DFIX is also shown. A comparison with the method from Liu et al. (2017) is also presented in Section 5, demonstrating the clear advantage of DFIX.

The contributions of this paper are the following: a novel fixed point iterative method for solving linear systems in the distributed environment is defined and convergence analysis that is analogous to the classical centralized case is presented, showing that the sufficient convergence condition is the same; the convergence factor depends on the norm of the iterative matrix, the diameter of the underlying communication graph and the weight matrix. The results are then extended to the case of directed communication graph and the time-varying graph under reasonable connectivity conditions. Extensive numerical tests are performed and the results confirm theoretical findings.

This paper is organized as follows. Section 2 contains the description of the computational framework together with a brief overview of fixed point methods. The method DFIX is defined and analyzed in Section 3 for the fixed graph case. In Section 4 we

present the time-varying case. Numerical results that illustrate the theoretical analysis as well as an application of DFIX to a Kriging problem are presented in Section 5. Some conclusions are drawn in Section 6.

2. Preliminaries

Let us first briefly recall the theory of fixed point iterative methods for systems of linear equations. A generic method of type (2)

$$y^{k+1} = My^k + d, \tag{3}$$

is convergent if $\rho(M) < 1$, where $\rho(M)$ is the spectral radius of M, i.e., the largest eigenvalue of M in modulus. This condition is both necessary and sufficient for convergence. Given any matrix norm $\|\cdot\|$ one can also state the sufficient convergence condition as $\|M\| < 1$. There are many ways of transforming (1) to the fixed point form (2), depending on the properties of A, with Jacobi and Gauss–Seidel methods, as well as their relaxation versions being the most studied methods. To fix the idea before defining the distributed method we recall here the Jacobi and Jacobi Over-relaxation, JOR, method, keeping in mind that we will consider a generic M in the next section.

Assume that A is a nonsingular matrix with nonzero diagonal entries. Using the splitting $A = D - P$, with $D = diag(a_{11}, \ldots, a_{nn})$, the Jacobi iterative method is defined by (3) with $M = D^{-1}P := M_J$. In other words, given $d = D^{-1}b$ and denoting by $y^k = (y_1^k, \ldots, y_n^k)$ the estimate of solution to (1) at iteration k, the new iteration is defined by

$$y_i^{k+1} = -\frac{1}{a_{ii}} \sum_{j=1, j\neq i}^{n} a_{ij} y_j^k + d_i, \ i = 1, \ldots, n.$$

The method is linearly convergent for many classes of matrices, for example strictly diagonally dominant matrices, symmetric positive definite matrices etc. (Greenbaum, 1997; Saad, 2003), and the rate of convergence is determined by $\rho(M_J)$. To speed up convergence and extend the class of matrices for which the method is convergent, one can introduce a relaxation parameter $\alpha \in \mathbb{R}$ and define $M = \alpha D^{-1}P + (1-\alpha)I$. In other words, the JOR iteration is given by

$$y_i^{k+1} = (1-\alpha) y_i^k - \frac{\alpha}{a_{ii}} \Big(\sum_{j=1, j\neq i}^{n} a_{ij} y_j^k + b_i \Big), \ i = 1, \ldots, n. \tag{4}$$

If A is a symmetric positive definite matrix, the JOR method converges (Greenbaum, 1997; Saad, 2003) for $\alpha \in (0, 2\rho(M_J)^{-1})$. Assuming that each node can communicate directly with every other node, the method can be applied in parallel and asynchronous manner, see Bertsekas and Tsitsiklis (2015), Frommer and Szyld (2000).

Let us now define precisely the computational environment we consider. Assume that the network of nodes is a directed network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of all edges, i.e., all pairs (i, j) of nodes where node i can send information to node j through a communication link.

Definition 1. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *strongly connected* if for every couple of nodes i, j there exists an oriented path from i to j in $\mathcal{G}$. That is, if there exist $s_1, \ldots, s_l$ such that $(i, s_1), (s_1, s_2), \ldots, (s_l, j) \in \mathcal{E}$.

Assumption A1. The network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is directed, strongly connected, with self-loops at every node.

Remark 1. The case of undirected network $\mathcal{G}$ can be seen as the particular case of directed graph where $\mathcal{G}$ is symmetric. That is, $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$. In this case, the hypothesis that $\mathcal{G}$ is strongly connected is equivalent to $\mathcal{G}$ connected.

Let us denote by O_i the in-neighborhood of node i, that is, the set of nodes that can send information to node i directly. Since the graph has self loops at each node, then $i \in O_i$ for every i. We associate with $\mathcal{G}$ an $n \times n$ matrix W, such that the elements of W are all nonnegative and each row sums up to one. More precisely, we assume the following.

Assumption A2. The matrix $W \in \mathbb{R}^{n\times n}$ is row stochastic with elements w_{ij} such that $w_{ij} > 0$ if $j \in O_i$, $w_{ij} = 0$ if $j \notin O_i$.

Let w_{min} denote a constant such that all nonzero elements of W satisfy $w_{ij} \geq w_{\min} > 0$. Such constant exists if Assumption A1 holds. Moreover, we have $w_{\min} \in (0, 1)$. Therefore, for all elements of W we have

$$w_{ij} \neq 0 \Rightarrow w_{ij} \geq w_{\min}. \tag{5}$$

The diameter δ of a network $\mathcal{G}$ is defined as the largest distance in the graph between two of its nodes.

3. DFIX method

Let us consider a generic fixed-point method (3) for solving (1) with $M = [m_{ij}] \in \mathbb{R}^{n\times n}$, $d = [d_i] \in \mathbb{R}^n$ defined in such a way that each node i contains the ith row $M_i \in \mathbb{R}^{1\times n}$ and $d_i \in \mathbb{R}$. Moreover, assume that the fixed point y^* of (2) is a solution of (1). The algorithm is designed in such way that each node has its own estimate of the solution y^*. Thus at the iteration k each node i has its own estimate $x_i^k \in \mathbb{R}^n$ with components x_{ij}^k, $j = 1, \ldots, n$. The DFIX method is presented in the algorithm below.

Algorithm DFIX

Step 0 Initialization: Set $k = 0$. Each node chooses $x_i^0 \in \mathbb{R}^n$.

Step 1 Each node i computes

$$\hat{x}_{ii}^{k+1} = \sum_{j=1}^{n} m_{ij} x_{ij}^k + d_i, \ \hat{x}_{ij}^{k+1} = \hat{x}_{ij}^k, i \neq j. \tag{6}$$

Step 2 Each node i updates its solution estimate

$$x_i^{k+1} = \sum_{j=1}^{n} w_{ij} \hat{x}_j^{k+1} \tag{7}$$

and sets $k = k + 1$.

Notice that at Step 1 each node i updates only the ith component of its solution estimate and leaves all other components unchanged, while in Step 2 all nodes perform a consensus step (deGroot, 1974; Hendrickx, Jungers, Olshevsky, & Vankeerberghen, 2014; Touri & Nedic, 2012) using the set of vector estimates $\hat{x}_j^{k+1}$. Thus the network agents have to locally store (and reach consensus to) a vector (solution estimate) of dimension n which thus grows with the number of agents N. Defining the global iterative variable

$$X^k = \left(x_1^k; \ldots; x_n^k\right) \in \mathbb{R}^{n^2}.$$

Algorithm DFIX can be stated in the condensed form using X^k, $\widehat{M}_i$ and $\widehat{d}_i$ for $i = 1, \ldots, n$, where $\widehat{d}_i = (0, \ldots, d_i, \ldots, 0)^T \in \mathbb{R}^n$ and $\widehat{M}_i \in \mathbb{R}^{n\times n}$ has the ith row equal to M, the rest of diagonal elements equal to 1 and the remaining elements equal to 0. Now, Step 1 can be rewritten as $\hat{x}_i^{k+1} = \widehat{M}_i x_i^k + \hat{d}_i$, and we can rewrite the Steps 1–2 in matrix form as

$$X^{k+1} = (W \otimes I)(\mathcal{M} X^k + \hat{d}) \tag{8}$$

where $\otimes$ denotes the Kronecker product, $\mathcal{M} = diag\left(\widehat{M}_1, \ldots, \widehat{M}_n\right) \in \mathbb{R}^{n^2 \times n^2}$, and $\widehat{d} = \left(\hat{d}_1; \ldots; \hat{d}_n\right) \in \mathbb{R}^{n^2}$. Notice that Eq. (8) is only theoretical, in the sense that since each agent has access only to partial information, the global vector X^k, the matrix $\mathcal{M}$ and the vector $\widehat{d}$ are not computed at any node and it serves only for convergence analysis of Algorithm 1.

The following theorem shows that all local sequences $\{x_i^k\}$, $i \in \{1, \ldots, n\}$ converge to the fixed point y^* of (2). Denote $X^* = (y^*; \ldots; y^*) \in \mathbb{R}^{n^2}$.

Theorem 1. *Let Assumptions A1 and A2 hold, $\|M\|_\infty = \mu < 1$ and let $\{X^k\}$ be a sequence generated by (8). Then the global error $E^k = X^k - X^*$ satisfies*

$$\|E^{k+1}\|_\infty \leq \tau \|E^{k-\delta+1}\|_\infty, \tag{9}$$

where $\tau = 1 - w_{\min}^{\delta}(1-\mu)$ and δ denotes the diameter of the underlying computational graph $\mathcal{G}$.

Proof. Since W is assumed to be row stochastic there holds $(W \otimes I)X^* = X^*$. Moreover, using the fact that $\hat{d} = (I \otimes I - \mathcal{M})X^*$, we obtain the following recursion

$$E^{k+1} = (W \otimes I)\mathcal{M}E^k. \tag{10}$$

Notice that $\|(W \otimes I)\mathcal{M}\|_\infty \leq 1$, and therefore

$$\|E^{k+1}\|_\infty \leq \|E^k\|_\infty. \tag{11}$$

Now, denoting by e_i^k the ith block of E^k (the local error corresponding to node i) and by e_{ij}^k its jth component, we get from (10)

$$e_{ij}^{k+1} = w_{ij}M_j e_j^k + \sum_{s \neq j} w_{is} e_{sj}^k. \tag{12}$$

The thesis follows if we prove that

$$|e_{ij}^{k+1}| \leq \tau \|E^{k-l+1}\|_\infty \tag{13}$$

holds for every k if the distance between j and i in the graph is equal to l, with $\tau = (1 - w_{\min}^{l}(1-\mu))$. We proceed by induction over the distance l. If $l = 1$, that is, if there is an edge from j to i, then $w_{ij} \geq w_{min} > 0$. By (12) we get

$$\begin{aligned} |e_{ij}^{k+1}| &\leq w_{ij}|M_j e_j^k| + \sum_{s \neq j} w_{is}|e_{sj}^k| \leq \\ &\leq w_{ij}\mu\|E^k\|_\infty + \|E^k\|_\infty \sum_{s \neq j} w_{is} \leq \\ &\leq \left(1 - w_{ij}(1-\mu)\right)\|E^k\|_\infty \leq \\ &\leq \left(1 - w_{\min}(1-\mu)\right)\|E^k\|_\infty, \end{aligned}$$

and defining $\tau' = \left(1 - w_{\min}(1-\mu)\right) < 1$, we get

$$|e_{ij}^{k+1}| \leq \tau' \|E^k\|_\infty. \tag{14}$$

Assume that (13) holds for distance equal to $l-1$, and let us prove it for l. Let $(j, s_{l-1}, s_{l-2}, \ldots, s_1, i)$ be a path of length l from j to i. In particular we have that $w_{is_1} > 0$ and thus

$$|e_{ij}^{k+1}| \leq w_{is_1}|e_{s_1 j}^k| + \sum_{s \neq s_1} w_{is}|e_{sj}^k|. \tag{15}$$

For each of the terms $|e_{sj}^k|$ in the sum, by (11), we have

$$|e_{sj}^k| \leq \|E^k\|_\infty \leq \|E^{k-l+1}\|_\infty. \tag{16}$$

Let us now consider the term $|e_{s_1 j}^k|$. Since $(j, s_{l-1}, s_{l-2}, \ldots, s_1, i)$ is a path of length l from j to i and the distance between j and i is equal to l, we have that the distance between j and s_1 is equal to $l-1$ and therefore, by inductive hypothesis we have for $\tau' = (1 - w_{\min}^{l-1}(1-\mu))$

$$|e_{s_1 j}^k| \leq \tau' \|E^{k-(l-1)}\|_\infty = \tau' \|E^{k-l+1}\|_\infty. \tag{17}$$

Putting (16) and (17) in (15), we get

$$\begin{aligned} |e_{ij}^{k+1}| &\leq w_{is_1}\tau' \|E^{k-l+1}\|_\infty + \sum_{s \neq s_1} w_{is}\|E^{k-l+1}\|_\infty = \\ &= \left(1 - w_{s_1 j}(1-\tau')\right)\|E^{k-l+1}\|_\infty \leq \\ &\leq \left(1 - w_{min}(1-\tau')\right)\|E^{k-l+1}\|_\infty \end{aligned} \tag{18}$$

and defining $\tau := \left(1 - w_{min}(1-\tau')\right)$ we get (13). Now the thesis follows directly from the fact that the distance between any two nodes is smaller or equal than the diameter δ of the graph. □

The previous analysis shows that the global error in nonexpanding and that we have a decrease after at most δ iterations, where δ is the diameter of the underlying graph. Next we quantify the R-linear convergence factor.

Corollary 1. *Suppose that the assumptions of Theorem 1 are satisfied. Then each node's solution estimate x_i^k converges to the solution y^* of the problem (2) R-linearly with the factor $\gamma = \tau^{1/\delta}$, i.e., for each $i \in \{1, 2, \ldots, N\}$ there holds $\|x_i^k - y^*\|_\infty = \mathcal{O}(\gamma^k)$, where $\gamma = \left(1 - w_{min}^{\delta}(1-\mu)\right)^{1/\delta}$.*

Proof. Denote $\xi_k := \|X^k - X^*\|_\infty = \|E^k\|_\infty$. Notice that (11) implies that $\xi_{k+1} \leq \xi_k$ for every k. Moreover, every iteration k can be represented as $k = s\delta + c$, where $s, c \in \mathbb{N}_0$ and $c < \delta$. Then,

$$\xi_k \leq \xi_{k-c} \leq \tau^s \xi_0 = \tau^{(k-c)/\delta}\xi_0 \leq \tau^{k/\delta}\tau^{-1}\xi_0 := \gamma^k C,$$

where

$$\gamma = \tau^{1/\delta} = \left(1 - w_{min}^{\delta}(1-\mu)\right)^{1/\delta}$$

and

$$C = \frac{\xi_0}{\tau} = \frac{\|X^0 - X^*\|_\infty}{1 - w_{min}^{\delta}(1-\mu)}.$$

By definition of X^k and X^* we have $\|x_i^k - y^*\|_\infty \leq \xi_k$, $i = 1, \ldots, N$ and the result follows. □

3.1. DFIX method - multirow case

The previous result can be generalized in such a way that each node holds more than one row as follows. We consider now a generic fixed-point method for solving (1) by the fixed point iterative method (3), with N nodes and $M = [m_{ij}] \in \mathbb{R}^{n \times n}$, $d = [d_i] \in \mathbb{R}^n$ defined in such a way that each node $i \in \{1, 2, \ldots, N\}$ contains several rows of M and d. Let us denote the rows assigned to node i by $R_i \subset \{1, 2, \ldots, n\}$. We assume $R_i \bigcap R_j = \varnothing$ for $i \neq j$ and $\bigcup_{i=1}^N R_i = \{1, 2, \ldots, n\}$. More precisely, each node i holds $M_j \in \mathbb{R}^{1 \times n}$ and $d_i \in \mathbb{R}$, for all $j \in R_i$. The algorithm is again designed in such way that each node has its own estimate of the solution y^*. Thus, at the iteration k each node i has its own estimate $x_i^k \in \mathbb{R}^n$ with components x_{ij}^k, $j = 1, \ldots, n$.

Algorithm DFIXM

Step 0 Initialization: Set $k = 0$. Each node chooses $x_i^0 \in \mathbb{R}^n$.
Step 1 Each node i computes

$$\hat{x}_{ij}^{k+1} = \sum_{l=1}^{n} m_{jl}x_{il}^k + d_j, \quad j \in R_i, \hat{x}_{ij}^{k+1} = \hat{x}_{ij}^k, j \notin R_i. \tag{19}$$

Step 2 Each node i updates its solution estimate

$$x_i^{k+1} = \sum_{j=1}^{N} w_{ij}\hat{x}_j^{k+1} \tag{20}$$

and sets $k = k + 1$.

Notice that at Step 1 each node i updates only the components $j \in R_i$ of its solution estimate and leaves all other components unchanged, while in Step 2 all nodes perform a consensus step using the set of vector estimates $\hat{x}_j^{k+1}$ obtained from the immediate neighbors.

Defining the global variable at iteration k as before, $X^k = \left(x_1^k; \ldots; x_n^k\right) \in \mathbb{R}^{Nn}$, Algorithm DFIXM can be stated in the condensed form with $\widehat{d}_i = \left(0, \ldots, d_j, d_{j+1}, \ldots, d_{j+q_i}, \ldots, 0\right)^T \in \mathbb{R}^n$ and $\widehat{M}_i \in \mathbb{R}^{n\times n}$ such that the jth row of $\widehat{M}_i$ is equal to the jth row of M for all $j \in R_i$, the rest of diagonal elements are equal to 1 and the remaining elements are equal to 0.

Now, Step 1 can be rewritten as $\hat{x}_i^{k+1} = \widehat{M}_i x_i^k + \hat{d}_i$, and each iteration of Algorithm DFIXM can be written as

$$X^{k+1} = (W \otimes I)(\mathcal{M}X^k + \hat{d}) \tag{21}$$

where $\mathcal{M} = diag\left(\widehat{M}_1, \ldots, \widehat{M}_n\right) \in \mathbb{R}^{Nn\times Nn}$, and $\widehat{d} = \left(\hat{d}_1; \ldots; \hat{d}_n\right) \in \mathbb{R}^{Nn}$. Again, (8) is never computed at any node and it is derived only for theoretical analysis.

The following theorem shows that for every $i \in \{1, \ldots, N\}$ the local sequence $\{x_i^k\}$ converges to the fixed point y^* of (2) as in the case of DFIX.

Theorem 2. *Let Assumptions A1–A2 hold, $\|M\|_\infty = \mu < 1$ and let $\{X^k\}$ be a sequence generated by (21). Then, for every k, the global error $E^k = X^k - X^*$ satisfies*

$$\|E^{k+1}\|_\infty \le \tau \|E^{k-\delta+1}\|_\infty, \tag{22}$$

where δ denotes the diameter of the underlying computational graph $\mathcal{G}$ and

$$\tau = 1 - w_{min}^{\delta}(1-\mu) \in (0, 1). \tag{23}$$

Proof. The proof is essentially the same as the proof of Theorem 1 with some technical changes. The error expression is now

$$e_{ij}^{k+1} = w_{ij}M_h e_j^k + \sum_{s\neq j} w_{is}e_{sj}^k, \tag{24}$$

where h depends on i and j. As in the previous case, we prove the thesis by proving that if the distance between j and i in the graph is equal to l, then

$$|e_{ij}^{k+1}| \le \tau \|E^{k-l+1}\|_\infty, \tag{25}$$

for every k, with $\tau = 1 - w_{min}^{l}(1-\mu) \in (0, 1)$. Let us proceed by induction over the distance l. If $l = 1$, that is, if there is an edge from j to i, then $w_{ij} \ge w_{min} > 0$. By (24) we get

$$\begin{aligned}
|e_{ij}^{k+1}| &\le w_{ij}|M_h e_j^k| + \sum_{s\neq j} w_{is}|e_{sj}^k| \le \\
&\le w_{ij}\|E^k\|_\infty \sum_{l=1}^{n} |m_{hl}| + \|E^k\|_\infty \sum_{s\neq j} w_{is} \le \\
&\le w_{ij}\mu\|E^k\|_\infty + \|E^k\|_\infty(1 - w_{ij}) \\
&\le \left(1 - w_{ij}(1-\mu)\right)\|E^k\|_\infty \le \\
&\le \left(1 - w_{\min}(1-\mu)\right)\|E^k\|_\infty,
\end{aligned}$$

and defining $\tau' = 1 - w_{\min}(1-\mu) < 1$, we get

$$|e_{ij}^{k+1}| \le \tau'\|E^k\|_\infty. \tag{26}$$

The rest of the proof is completely analogous to the proof of Theorem 1 and hence omitted here. □

Analogously, we can quantify the convergence factor in the same way as before and the corollary below holds.

Corollary 2. *Suppose that the assumptions of Theorem 2 are satisfied. Then each node's solution estimate x_i^k converges to the solution y^* of the problem (2) R-linearly with the factor $\gamma = \tau^{1/\delta}$, i.e., for each $i \in \{1, 2, \ldots, N\}$ there holds $\|x_i^k - y^*\|_\infty = \mathcal{O}(\gamma^k)$, where $\gamma = \left(1 - w_{min}^{\delta}(1-\mu)\right)^{1/\delta}$.*

DFIXM is a generalization of DFIX that might be of practical importance as it allows us to solve an n dimensional linear system with an arbitrary number of nodes $N \le n$ which might be the case in many applications. However we will continue with DFIX method for time-varying networks in the next Section to avoid notation cluttering and to facilitate reading. The changes in the proofs are of the same type as above.

4. Time-varying network

The method discussed in the previous sections is valid only if the graph representing the communication among the agents is the same at each iteration. If some failure of the communication link between two agents occurs during the execution of the algorithm, the underlying network changes, and Theorem 1 does not apply anymore. Thus DFIX is extended to the framework of time-varying communication networks and conditions on the sequence of communication graphs that yield convergence result analogous to the fixed network case are given. In particular we show that, in order to achieve convergence, strong connectivity is not necessary at any time.

Assume that a sequence of directed graphs $\{\mathcal{G}_k\}_k$ is given, such that at iteration k, $\mathcal{G}_k$ represents the network of nodes. That is, at iteration k, each node can communicate with its neighbors in $\mathcal{G}_k$. The DFIX algorithm described by Eqs. (6) and (7) can be applied in this case if we replace (7) with

$$x_i^{k+1} = \sum_{j=1}^{n} w_{ij}^k \hat{x}_j^{k+1} \tag{27}$$

where W^k is the consensus matrix associated with the graph $\mathcal{G}_k$, that is, W^k satisfies Assumption A2 with $\mathcal{G} = \mathcal{G}_k$. With this modification, the equation describing the global iteration becomes

$$X^{k+1} = (W^k \otimes I)(\mathcal{M}X^k + \hat{d}). \tag{28}$$

Let us first discuss the sequence of communication graphs.

Definition 2. Given graphs $\mathcal{G}_1, \mathcal{G}_2$ with $\mathcal{G}_i = (\mathcal{V}, \mathcal{E}_i)$, their *composition* is defined as $\mathcal{G}_2 \circ \mathcal{G}_1 = (\mathcal{V}, \mathcal{E})$ where $\mathcal{E} := \{(j, i) \in \mathcal{V}^2 \mid \exists\, s \in \mathcal{V}$ such that $(j, s) \in \mathcal{E}_1, (s, i) \in \mathcal{E}_2\}$.

In other words an edge from j to i in $\mathcal{G}_2 \circ \mathcal{G}_1$ exists if we can find a path from j to i such that the first edge of the path is in $\mathcal{G}_1$ and the second edge is in $\mathcal{G}_2$. This definition can be extended to finite sequences of graphs of arbitrary length.

Remark 2. Let us consider a generic set of graphs $\mathcal{G}_1, \ldots, \mathcal{G}_m$. It is easy to see that if for every index j the graph $\mathcal{G}_j$ has self-loops at every node then the set of edges of the composition $\mathcal{G}_1 \circ \ldots \circ \mathcal{G}_m$ contains the set of edges of $\mathcal{G}_j$ for every j. In particular, if there exists an index $\hat{j} \in \{1, \ldots, m\}$ such that $\mathcal{G}_{\hat{j}}$ is fully connected, then $\mathcal{G}_1 \circ \ldots \circ \mathcal{G}_m$ is also fully connected.

Definition 3. Given an infinite sequence of networks $\{\mathcal{G}_k\}_k$ and a positive integer $\bar{m}$, we say that the sequence is *jointly fully (respectively, strongly) connected for sequences of length* $\bar{m}$ if for every index k, the composition $\mathcal{G}_k \circ \mathcal{G}_{k+1} \circ \ldots \circ \mathcal{G}_{k+\bar{m}-1}$ is fully (respectively, strongly) connected.

Definition 4. Given an infinite sequence of networks $\{\mathcal{G}_k\}_k$ and two integers τ_0, l, we say that the sequence is *repeatedly jointly strongly connected with constants* τ_0, l, if for every index k, the composition $\mathcal{G}_{\tau_0+kl} \circ \mathcal{G}_{\tau_0+kl+1} \circ \ldots \circ \mathcal{G}_{\tau_0+(k+1)l}$ is strongly connected.

Definition 5. Given two vertices i, j we say that there is a *joint path* of length l from i to j in $\mathcal{G}_k, \ldots, \mathcal{G}_{k+\bar{m}-1}$ if there exist $s_1, \ldots, s_{l-1}$ such that $(i, s_1) \in \mathcal{E}_{k+\bar{m}-1}$, $(s_1, s_2) \in \mathcal{E}_{k+\bar{m}-2}, \ldots, (s_{l-1}, j) \in \mathcal{E}_{k+\bar{m}-l}$, and we say that i, j have *joint distance* l in $\mathcal{G}_k, \ldots, \mathcal{G}_{k+\bar{m}-1}$ if the shortest joint path from i to j is of length l.

Our analysis is based on the following assumption.

Assumption A3. $\{\mathcal{G}_k\}$ is a sequence of directed graphs, with self-loops at every node, jointly fully connected for sequences of length $\bar{m}$, for some positive integer $\bar{m}$.

The algorithm presented in Liu et al. (2018) works for time-varying network in a similar framework. Formally, the hypothesis on $\{\mathcal{G}_k\}$ in Liu et al. (2018) is the following.

Assumption A3'. $\{\mathcal{G}_k\}$ is a sequence of directed graphs, with self-loops at every node, jointly *strongly* connected for sequences of length $\bar{p}$, for some positive integer $\bar{p}$.

We show now that Assumptions A3 and A3' are equivalent, in the sense specified by Proposition 1. In the following, given an integer m, we denote with $\mathcal{G}^m$ the composition of m copies of $\mathcal{G}$.

Lemma 1. *If $\mathcal{G}$ is a directed strongly connected graph with self-loops at every node and diameter δ, then $\mathcal{G}^\delta$ is fully connected.*

Proof. By definition of composition we have that (i, j) is an edge in $\mathcal{G}^\delta$ if and only if $\exists s_1, \ldots, s_{\delta-1} \in \mathcal{V}$ such that

$$(i, s_1), (s_1, s_2), \ldots, (s_{\delta-1}, j) \in \mathcal{G}. \tag{29}$$

We want to prove that for every $i, j \in \mathcal{V}$ a sequence of nodes s_h as in (29) exists.

Since $\mathcal{G}$ is fully connected with diameter δ, there exists a path in $\mathcal{G}$ from i to j of length $l \leq \delta$. That is, there exist a set of nodes $v_1, \ldots, v_{l-1}$ such that $(i, v_1), (v_1, v_2), \ldots, (v_{l-1}, j)$ are edges in $\mathcal{G}$ and therefore a sequence satisfying (29) is given by $s_h = v_h$ for $h = 1 : l-1$ and $s_h = j$ for $h = l : \delta$. □

Proposition 1. *Let $\{\mathcal{G}_k\}$ be a sequence of graphs where, for each k, $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$ is a directed graph with self-loops at every node. The following are equivalent:*

(1) there exist $\tau_0, l \in \mathbb{N}$ such that $\{\mathcal{G}_k\}$ is repeatedly jointly strongly connected with constants τ_0, l
(2) there exists $\bar{p} \in \mathbb{N}$ such that $\{\mathcal{G}_k\}$ is strongly connected for sequences of length $\bar{p}$
(3) there exists $\bar{m} \in \mathbb{N}$ such that $\{\mathcal{G}_k\}$ is fully connected for sequences of length $\bar{m}$

Proof. It is easy to see that (2) $\Rightarrow$ (1) with $\tau_0 = 0$ and $l = \bar{p}$ and since full connectivity clearly implies strong connectivity, we have that (3) $\Rightarrow$ (2) with $\bar{p} = \bar{m}$.

We now prove that (1) $\Rightarrow$ (2) with $\bar{p} = 2l$. That is, we prove that if (1) holds, then for every index s the composition $\mathcal{G}_s \circ \ldots \circ \mathcal{G}_{s+2l-1}$ is strongly connected. Given an index s, we denote with $\bar{r}$ the remainder of the division of $(s - \tau_0)$ by l, we define $\bar{h} := l^{-1}(s - \tau_0 + l - \bar{r})$. By definition of $\bar{r}$ and $\bar{h}$ and applying (1) with $k = \bar{h}$ we have that the graph $H := \mathcal{G}_{s+l-\bar{r}} \circ \ldots \circ \mathcal{G}_{s+2l-\bar{r}-1} = \mathcal{G}_{\tau_0+\bar{h}l} \circ \ldots \circ \mathcal{G}_{\tau_0+(\bar{h}+1)l-1}$ is strongly connected and thus $\mathcal{G}_s \circ \ldots \circ \mathcal{G}_{s+2l-2} = \mathcal{G}_s \circ \ldots \circ \mathcal{G}_{s+l-\bar{r}-1} \circ H \circ \circ \mathcal{G}_{s+2l-\bar{r}} \circ \ldots \circ \mathcal{G}_{s+2l-1}$ is strongly connected. Since $2l - \bar{r} \in l+1, \ldots, 2l$ we have the thesis.

Finally, we prove that (2) $\Rightarrow$ (3). Since the size of $\mathcal{V}$ is finite, there exists a finite number of graphs with vertices $\mathcal{V}$. In particular, there exists a finite integer L equal to the number of strongly connected graphs with vertices $\mathcal{V}$. We denote with $H_1, \ldots H_L$ such graphs, with δ_j the diameter of H^j and with $\bar{\delta} := \max \delta_j$. Given any index k, we consider $(\bar{\delta}-1)L+1$ sequences of length $\bar{p}$ as follows:

$$S_1 = \mathcal{G}_k \circ \mathcal{G}_{k+1} \ldots \circ \mathcal{G}_{k+\bar{p}-1},$$
$$S_2 = \mathcal{G}_{k+\bar{p}} \circ \mathcal{G}_{k+\bar{p}+1} \ldots \circ \mathcal{G}_{k+2\bar{p}-1},$$
$$\vdots$$
$$S_{(\bar{\delta}-1)L+1} = \mathcal{G}_{k+(\bar{\delta}-1)L\bar{p}} \ldots \circ \mathcal{G}_{k+(\bar{\delta}-1)L\bar{p}+\bar{p}-1}.$$

Statement (2) implies that, for every $j \in \{1, \ldots, (\bar{\delta}-1)L+1\}$, $S_j \in \{H_1, \ldots H_L\}$ and thus there exists an index $\hat{\imath} \in \{1, \ldots, L\}$ such that at least $\bar{\delta}$ elements of $\{S_1, \ldots, S_{(\bar{\delta}-1)L+1}\}$ are equal to $H_{\hat{\imath}}$. Using the fact that, by Lemma 1, $H_{\hat{\imath}}^{\delta_{\hat{\imath}}}$ is fully connected and Remark 2, we have $\mathcal{G}_k \circ \mathcal{G}_{k+1} \circ \ldots \circ \mathcal{G}_{k+(\bar{\delta}-1)L\bar{p}+\bar{p}-1} = S_1 \circ \ldots \circ S_{(\bar{\delta}-1)L+1}$ fully connected, so (3) holds with $\bar{m} = (\bar{\delta}-1)L\bar{p} + \bar{p}$. □

To conclude the considerations on the sequence of networks we remark that, since we are assuming that the linear system (1) has unique solution and that each node contains exactly one row of the coefficient matrix, the D-connectivity hypothesis introduced in Liu et al. (2017) is equivalent to Assumption A3' and thus, by Proposition 1, to Assumption A3.

Theorem 3. *Assume that a sequence of networks $\{\mathcal{G}_k\}_k$ is given, satisfying Assumption A3, and that for every index k the corresponding consensus matrix W^k satisfies Assumption A2. Let $\{X^k\}$ be a sequence generated by (28) with $\|M\|_\infty = \mu < 1$. For every $k \in \mathbb{N}$ the global error $E^k = X^k - X^*$ satisfies*

$$\|E^{k+1}\|_\infty \leq \sigma \|E^{k-\bar{m}+1}\|_\infty, \tag{30}$$

where $\sigma = (1 - w_{min}^{\bar{m}}(1-\mu))$ and $\bar{m}$ is the constant given by Assumption A3.

Proof. We follow the proof of Theorem 1. For every index k, the matrix W^k is row stochastic and $\|(W^k \otimes I)\mathcal{M}\|_\infty \leq 1$, so we get

$$E^{k+1} = (W^k \otimes I)\mathcal{M}E^k. \tag{31}$$

and

$$\|E^{k+1}\|_\infty \leq \|E^k\|_\infty. \tag{32}$$

For every node i, j and for every iteration index k, we have

$$e_{ij}^{k+1} = w_{ij}^k M_j e_j^k + \sum_{s \neq j} w_{is}^k e_{sj}^k. \tag{33}$$

We now prove that if the joint distance between j and i in $\mathcal{G}_{k-\bar{m}+1}$, $\mathcal{G}_{k-\bar{m}+2}, \ldots, \mathcal{G}_k$ is equal to l, then for every k

$$|e_{ij}^{k+1}| \leq \sigma \|E^{k-l+1}\|_\infty, \tag{34}$$

where $\sigma = (1 - w_{\min}^l(1-\mu))$. We proceed by induction over the joint distance l. If $l = 1$, that is, if $w_{ij}^k > 0$, proceeding as in the derivation of (14) we get

$$|e_{ij}^{k+1}| \leq \big(1 - w_{ij}^k(1-\mu)\big)\|E^k\|_\infty \leq \big(1 - w_{\min}(1-\mu)\big)\|E^k\|_\infty =: \sigma\|E^k\|_\infty.$$

Assume now that (34) holds for distance equal to $l-1$. Let $(j, s_{l-1}, s_{l-2}, \dots, s_1, i)$ be a joint path of length l from j to i in $\mathcal{G}_{k-\bar{m}+1}, \mathcal{G}_{k-\bar{m}+2}, \dots, \mathcal{G}_k$ In particular we have that $w^k_{is_1} > 0$ and thus

$$|e^{k+1}_{ij}| \leq w_{is_1}|e^k_{s_1 j}| + \sum_{s \neq s_1} w_{is}|e^k_{sj}|. \tag{35}$$

Using the fact that $(j, s_{l-1}, s_{l-2}, \dots, s_1)$ is a joint path of length $l-1$ from j to s_1 in $\mathcal{G}_{k-\bar{m}+1}, \mathcal{G}_{k-\bar{m}+2}, \dots, \mathcal{G}_{k-1}$, applying the inductive hypothesis and proceeding as in the proof of the previous theorem, we get

$$|e^{k+1}_{ij}| \leq \left(1 - w_{min}(1-\sigma')\right) \|E^{k-l+1}\|_\infty \tag{36}$$

with σ' given by (34) for distance $l-1$, and defining $\sigma := \left(1 - w_{min}(1-\sigma')\right) = (1 - w^l_{min}(1-\mu))$ we get (34) for distance equal to l.

Since the sequence $\{\mathcal{G}_k\}$ is fully connected for sequences of length $\bar{m}$ we have that for every couple of nodes i, j the joint distance between j and i in $\mathcal{G}_{k-\bar{m}+1}, \mathcal{G}_{k-\bar{m}+2}, \dots, \mathcal{G}_k$ is smaller or equal than $\bar{m}$ and the statement follows. □

Theorem 3 shows that the time-independent case fits in the time-varying framework with $\mathcal{G}_k$ is equal to $\mathcal{G}$. If the diameter of $\mathcal{G}$ is δ, then Assumption 3 holds with $\bar{m} = \delta$ and the two theorems coincide.

5. Numerical results

In this section we present testing results for DFIX method and comparison with the state-of-the-art distributed optimization algorithms from Li and Qu (2017), Nedic et al. (2016), Shi et al. (2015), Sundararajan et al. (2019) referred to here as DIGing, EXTRA and SVL respectively, and the method for solving systems of linear equations presented in Liu et al. (2017), abbreviated here as Projection. The test set consists of two types of problems: Simple Kriging problems and linear systems with strictly diagonally dominant coefficient matrix. The influence of the connectivity of the underlying network on computational and communication cost of DFIX is studied in Section 5.1. We also compare DFIX with the mentioned methods on a simple Kriging problem. In Section 5.2 we repeat the comparison considering a randomly generated linear system. The DFIXM method is considered as well and we discuss the situation of fixed linear system and increasing number of the nodes. In Sections 5.3 and 5.4 we consider the cases of directed and time-varying networks, respectively.

The results demonstrate that DFIX, analogously to the classical results, outperforms the optimization method for solving the unconstrained quadratic problem both in terms of computation and communication. With respect to the method from Liu et al. (2017) the comparison is again favorable for DFIX. Clearly, the method from Liu et al. (2017) is designed for a wider class of problems, but in the case of unique solution and a suitable iterative matrix its efficiency is significantly lower than DFIX.

In the following, the DFIX method we consider is defined using Jacobi Overrelaxation, as specified in Section 2, as the underlying fixed point method. The iteration k of the distributed method at each node is given by

$$\hat{x}^{k+1}_{ii} = (1-\alpha)x^k_{ii} - \frac{\alpha}{a_{ii}}\left(\sum_{j \neq i} a_{ij}x^k_{ij} - b_i\right), \tag{37}$$

$$\hat{x}^{k+1}_{ij} = x^k_{ij} \quad \text{for } j \neq i,$$

and

$$x^{k+1}_i = \sum_{j=1}^n w_{ij}\hat{x}^{k+1}_j. \tag{38}$$

In the rest of the section we refer to the method defined by Eqs. (37), (38) as DFIX-JOR, and we choose the relaxation parameter α in (37) as $2/\|D^{-1}A\|_\infty$ where $D = diag(a_{11}, \dots, a_{nn})$. The methods for distributed optimization DIGing, EXTRA and SVL are applied to solve the unconstrained problem with quadratic objective function given by $\frac{1}{2}x^TAx - b^Tx$, which is equivalent to finding a solution of (1). The step-size parameter for DIGing and EXTRA is chosen as $\eta = 1/(3L)$ where $L = \max_{i=1:n} 2\|A_i\|^2_2$, while the parameters for SVL method are computed through the procedure described in Sundararajan et al. (2019). We remark that the relaxation parameter for DFIX and the step-size η for DIGing and EXTRA can be easily computed in the distributed framework, the computation of the optimal parameters for SVL requires knowledge of the extremal eigenvalues of the matrix A and the spectral gap of the consensus matrix W. Finally, Projection method deals with the linear system (1) directly and it does not require the computation of any additional parameter, but it requires a local initial vector x^0_i for each node i.

5.1. Simple Kriging problem

The first problem we consider is Simple Kriging (Cressie, 1993). Let us consider a physical process modeled as a spatial random field and assume that a network of sensors is given in the region of interest, taking measurements of the field. The goal is to estimate the field in any given point of the region. Assuming that the field is Gaussian and stationary, and that the expected value and covariance function are known at any point, this kind of problem can be solved by Simple Kriging method.

Denote with $\mathcal{Z}(s)$ the value of the random field at the point s, and with $\mu(s)$ its expected value, which is assumed to be known. Moreover, by the stationarity assumption, the covariance between the value of $\mathcal{Z}$ at two points is given by

$$Cov(\mathcal{Z}(s_1), \mathcal{Z}(s_2)) = \mathcal{K}(\|s_1 - s_2\|_2)$$

for some nonnegative function $\mathcal{K}$. Given the positions in space $\{s_1, \dots, s_n\} \subset \mathbb{R}^2$ of the n sensors of the network, let $\{\mathcal{Z}(s_1), \dots, \mathcal{Z}(s_n)\}$ be the sampled values at those points and define the covariance matrix $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ as $a_{ij} = \mathcal{K}(\|s_i - s_j\|_2)$. Now, given a point $\bar{s}$ where we want to estimate the field, the vector $b \in \mathbb{R}^n$ is defined as $b_i = \mathcal{K}(\|s_i - \bar{s}\|_2)$. The predicted value of $\mathcal{Z}(\bar{s})$ is then given by

$$\hat{p}(s) := \mu(\bar{s}) + \sum_{i=1}^n x_i(\mathcal{Z}(s_i) - \mu(s_i))$$

where $(x_1, \dots, x_n)$ is the approximate solution of the linear system

$$Ax = b. \tag{39}$$

Clearly, the matrix W plays an important role in the DFIX-JOR method. So let us first illustrate the influence of connectivity within the network in terms of communication traffic and computational cost for the kriging problem, with covariance function given by

$$\mathcal{K}(t) := \exp(-5t^2). \tag{40}$$

We assume that a set $\{s_1, \dots, s_{100}\} \subset [-30, 30]^2$ of agents is given and for any $m \in \{2, 4, \dots, 48, 50\}$ we take the m-regular graph with vertices $\{s_1, \dots, s_{100}\}$. The matrix W is defined using the Metropolis weights (Xiao, Boyd, & Lall, 2006), which in the m-regular case are given by

$$w_{ij} = \begin{cases} (m+1)^{-1}, & \text{if } j = i \text{ or } j \in \mathcal{O}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{41}$$

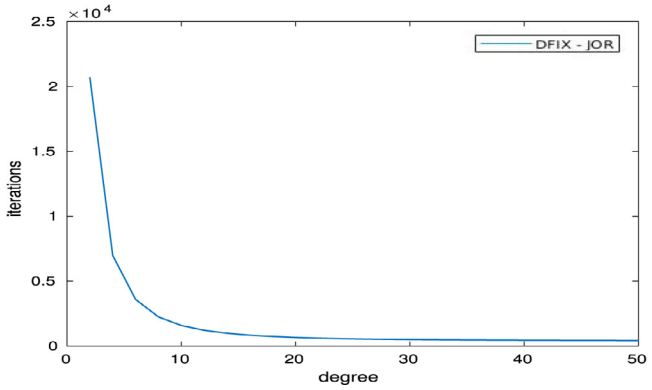

Fig. 1. Number of iterations.

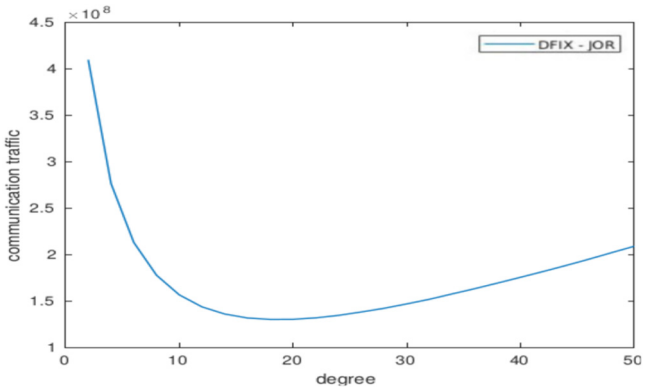

Fig. 2. Communication cost.

For each value of the degree m the system $Ax = b$ is solved with DFIX-JOR. In Figs. 1 and 2 we plot the number of iterations and the total communication cost, respectively, until the stopping criterion

$$\max_{i=1,\dots,n} \|Ax_i^k - b\| \leq 10^{-4} \tag{42}$$

is satisfied, for graphs of increasing degree. The communication cost is computed as follows. At each iteration, Step 1 does not require any communication, while in Step 2 node i shares x_i^k with all the agents in its neighborhood. The per-iteration traffic is thus given by $n^2m = 2|\mathcal{E}|n$, where $\mathcal{E}$ is the set of edges of the underlying network and m is the degree. Note that here we implicitly assume that there is a dedicated communication link between any pair of agents. This reflects practical scenarios where dedicated peer-to-peer channels are ensured, e.g., through frequency division multiple access or similar schemes. In the other tests that we present the broadcasting scenario will also be considered: in that case, the per-iteration communication cost is independent to the number of edges in the network and it is given by the number of nodes times the size of the shared vectors, thus it is proportional to the number of performed iterations.

From Figs. 1 and 2 one can see that, as the degree of the network increases, the number of iterations required to satisfy (42) decreases, while the total communication traffic first decreases then increases again. As the connectivity of the graph improves, the local information is distributed through the network more efficiently, and a smaller number of iterations is necessary. On the other hand, if the degree is larger, the consensus step (7) of the algorithm requires each node to share its local vector with a larger number of neighbors, yielding a higher communication traffic at each iteration. The fact that the overall communication traffic (Fig. 2) is nonmonotone suggests that for large values of the degree, the decrease in the number of iterations in not enough to balance the higher per-iteration traffic. Let us now compare the DFIX-JOR with DIGing (Li & Qu, 2017; Nedic et al., 2016), EXTRA (Shi et al., 2015), SVL (Sundararajan et al., 2019) and Projection method (Liu et al., 2017). We consider a 10×10 grid of nodes located at $\{s_1, \dots, s_{100}\} \subset [-3, 3]^2$ and, given a communication radius $R > 0$, we define the network so that nodes i and j are neighbors if and only if their distance is smaller than R. The linear system that we consider is derived by the kriging problem described at the beginning of this section. That is, we consider again $Ax = b$ with

$$a_{ij} = \mathcal{K}(\|s_i - s_j\|_2), \quad b_i = \mathcal{K}(\|s_i - \bar{s}\|_2) \tag{43}$$

where $\mathcal{K}$ is given by (40) and $\bar{s}$ is a fixed random point in $[-3, 3]^2$. Proceeding as in the previous test, we compute the communication traffic and computational cost required by the three methods

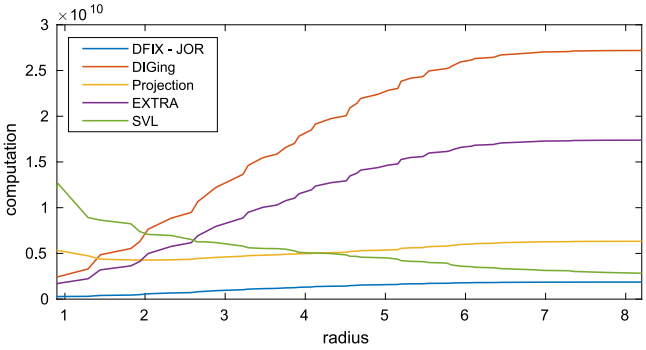

Fig. 3. Simple kriging problem (43), computational cost.

to achieve the tolerance specified at (42), for different values of the communication radius R. For each method, the overall computational cost is given by the number of iterations performed times the per-iteration cost, calculated as the number of scalar operations in one iteration. Similarly, the communication traffic is given by the number of iterations times the total number of vectors shared by the nodes during one iteration, times the length n of the vector. The matrix W is defined as in Xiao et al. (2006), with off-diagonal elements $w_{ij} = \frac{1}{1+\max\{m_i, m_j\}}$ if $j \in \mathcal{O}_i$, and $w_{ij} = 0$ otherwise, where m_i denotes the degree of node i. The diagonal elements are $w_{ii} = 1 - \sum_{j \neq i} w_{ij}$. The resulting matrix W is stochastic. The stopping criterion is the same as in the previous test. The initial point at each node is the same for all the methods, $x_{ii}^0 = b_i/a_{ii}$ and $x_{ij}^0 = 0$ for every $j \neq i$. In Figs. 3, 4 and 5 we plot the obtained results. As we can see, in this framework, DFIX method is more efficient than the methods we compare with, both in terms of computational and communication costs.

5.2. Strictly diagonally dominant systems

Let us now consider a linear system $Ax = b$ of order $n = 100$, where A and b are generated as follows. For every index i we take b_i randomly generated with uniform distribution in $(0, 1)$ and A is a symmetric diagonally dominant random matrix obtained as follows: take $\hat{a}_{ij} \in (0, 1)$ with uniform distribution and then set $\tilde{A} = \frac{1}{2}(\hat{A} + \hat{A}^T)$ and finally $A = \hat{A} + (n-1)I$, where I is the identity matrix of order n. The underlying network is an m-regular graph with n nodes. For every fixed value of the degree m, 10 random linear systems are generated, solved with all methods and the average number of iterations needed to fulfill (42) is computed. The total amount of computation and communication for each method are then obtained multiplying the average number of

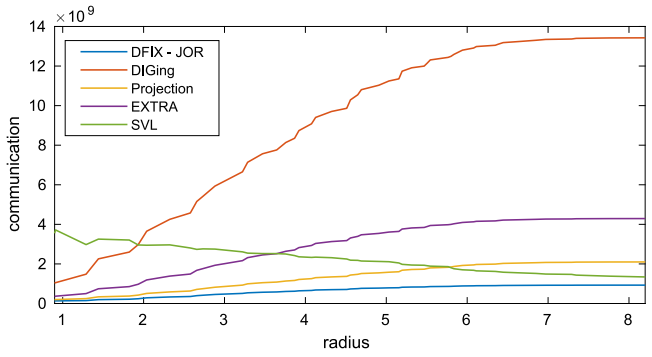

Fig. 4. Simple kriging problem (43), peer-to-peer communication traffic.

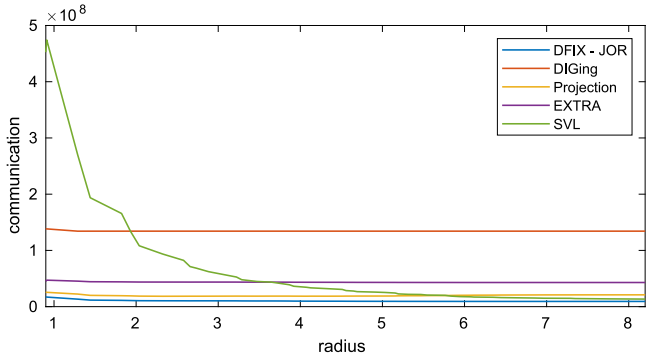

Fig. 5. Simple kriging problem (43), broadcasting communication traffic.

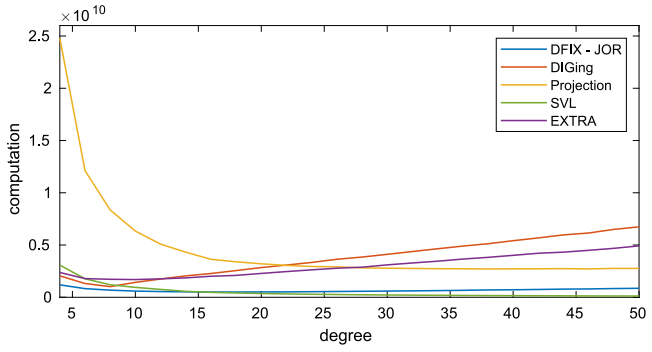

Fig. 6. m-regular graph, computational cost.

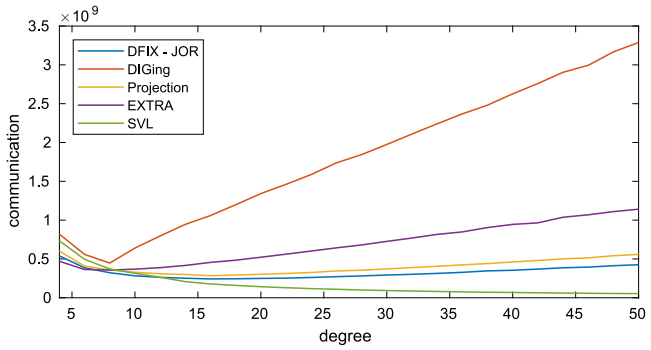

Fig. 7. m-regular graph, peer-to-peer communication traffic.

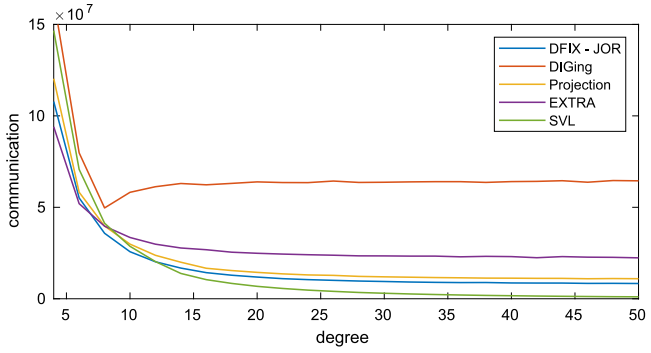

Fig. 8. m-regular graph, broadcasting communication traffic.

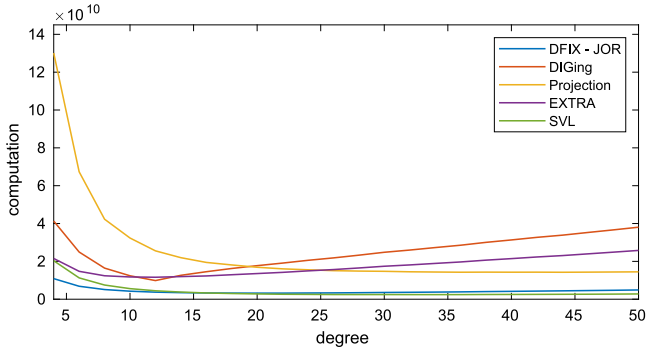

Fig. 9. Computational cost.

iterations and the per-iteration computational cost and communication traffic, respectively. The matrix W is defined as in (41), the parameter of the methods are computed as described at the beginning of the section, while the initial guess at each node and the termination condition are as in the previous test. In Figs. 6, 7 and 8 we plot the results for $m \in \{2, 4, \ldots, 48, 50\}$. DFIX outperforms DIGing, EXTRA and Projection method in terms of computation and communication both in the peer-to-peer and in the broadcasting scenario, while SVL method performs better than DFIX for values of the degree larger than 15. We remark again that SVL method is run with the optimal choice of the parameters, exploiting information on the eigenvalues of A and W.

The same tests were performed on random Erdos–Renyi graphs with given expected average degree for a sequence of increasing degrees. In these tests all methods are more expensive in terms of both communication and computational effort but the mutual comparison is the same as in the case of m-regular graphs.

To confirm the effectiveness of DFIXM, we repeat the previous test with a linear system of size $n = 500$ and $N = 100$ nodes, where each node is assigned 5 equations. As we can see in Figs. 9, 10 and 11 the results for all the methods are completely analogous to the case where each node holds one equation.

Let us now show the influence of the number of nodes in the network on performance of the five methods. We consider a linear system of size $n = 100$ generated as described above, and for $N = 10, 20, \ldots, 100$ consider a regular network of size N. For each value of N the degree of the network is chosen so that the ratio between N and the degree is constant. The results are plotted in Figs. 12–14. The amount of both computation and communication of all the methods increases together with the number of nodes. Moreover, DFIX seems to outperform all the methods that we compare to in terms of computational costs, while in terms of communication it seems to be comparable with Projection and both methods seem to be cheaper than the optimization methods.

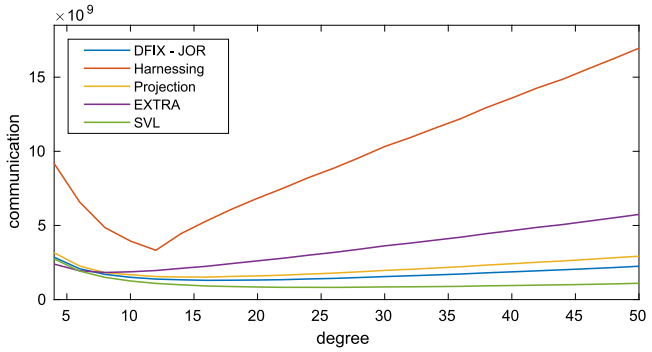

Fig. 10. Peer-to-peer communication traffic.

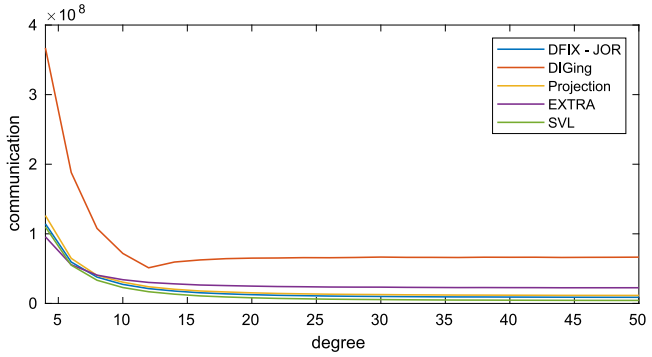

Fig. 11. Broadcasting communication traffic.

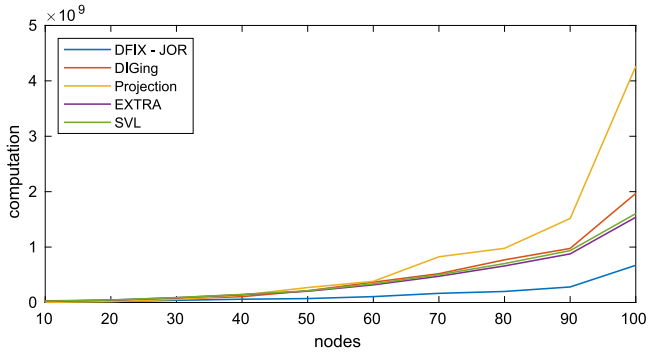

Fig. 12. Computational cost.

5.3. Directed networks

We now consider underlying directed networks. Let $n = 100$ be the size of linear system generated as in the previous tests. We consider a randomly generated directed network of size n such that the average out-degree of the nodes is equal to a fixed m. The consensus matrix W is defined with off-diagonal elements $w_{ij} = 1/(1+\hat{m}_i)$ if $j \in \mathcal{O}_i$, and $w_{ij} = 0$ otherwise, where $\hat{m}_i$ denotes the out-degree of node i, and the diagonal elements are $w_{ii} = 1 - \sum_{j \neq i} w_{ij}$. The resulting matrix W is row-stochastic. In Figs. 15, 16, 17 we plot the results for $m = 8, \ldots, 50$ for DFIX, DIGing, EXTRA and Projection. The SVL method fails to converge in this framework. The resulting comparison among the four methods is analogous to the case of undirected networks: DFIX seems to require the smallest computational effort among all methods and a similar communication traffic as Projection.

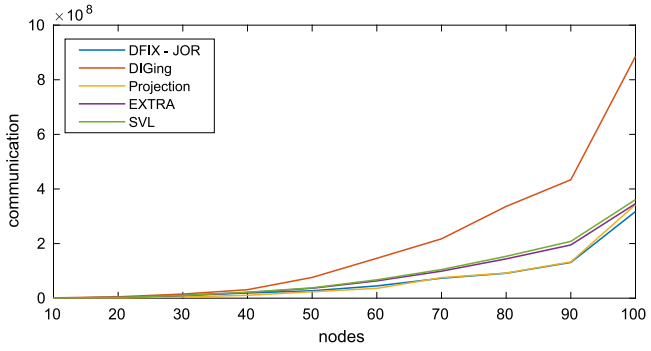

Fig. 13. Peer-to-peer communication traffic.

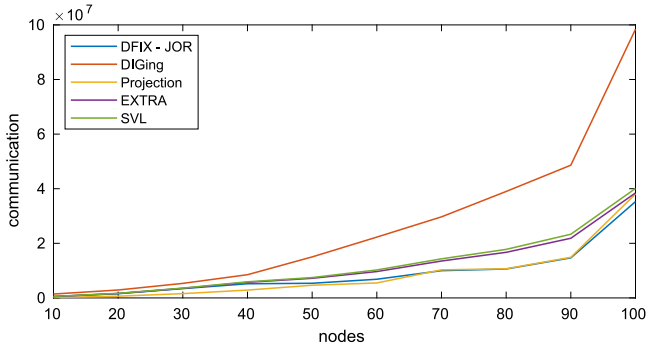

Fig. 14. Broadcasting communication traffic.

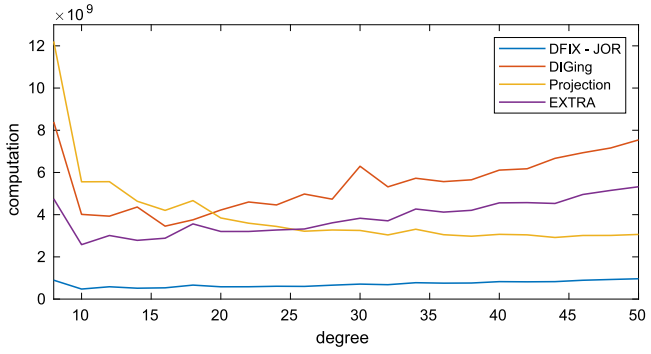

Fig. 15. Computational cost.

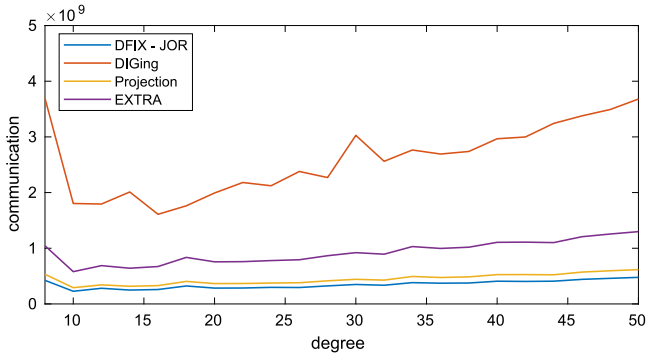

Fig. 16. Peer-to-peer communication traffic.

5.4. Time-varying network

We now compare the performance of the five methods in the time-varying case described in Section 4. The sequence $\{\mathcal{G}_k\}$

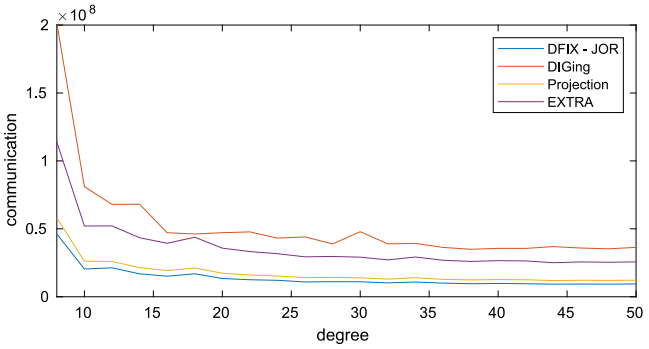

Fig. 17. Broadcasting communication traffic.

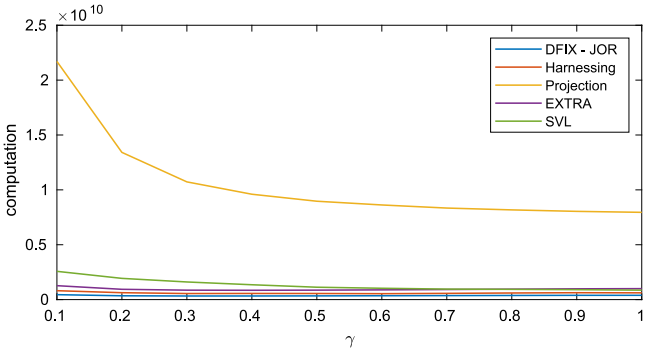

Fig. 18. Computational cost.

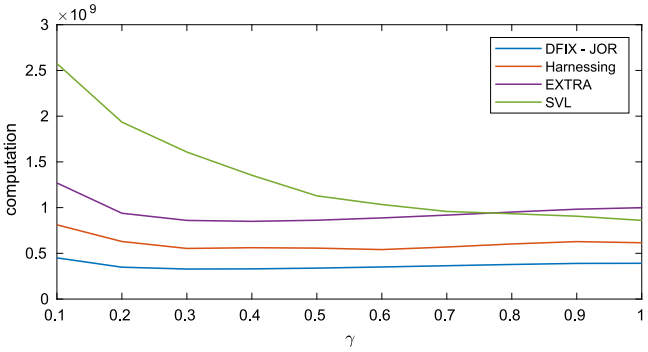

Fig. 19. Computational cost.

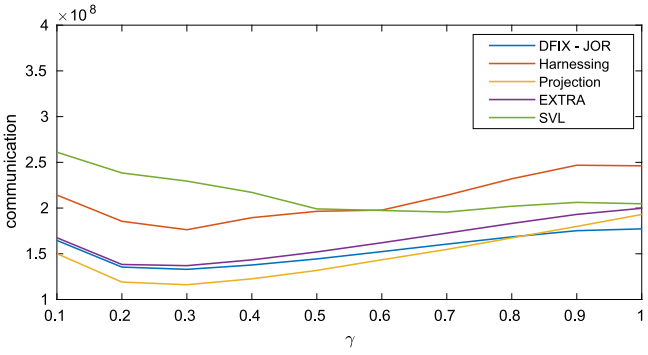

Fig. 20. Peer-to-peer communication traffic.

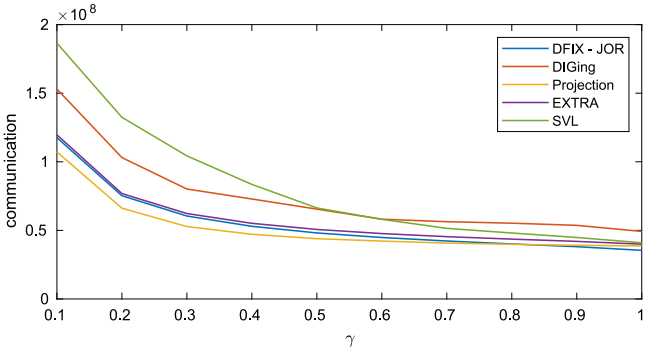

Fig. 21. Broadcasting communication traffic.

is generated as follows. For a fixed strongly connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a scalar $\gamma \in (0, 1]$, at every iteration k we randomly generate $\mathcal{E}_k$ by uniformly sampling $\gamma|E|$ edges from $\mathcal{E}$ and we set $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$. This construction can be interpreted as having a fixed underlying graph $\mathcal{G}$ that represents the available communication links among the nodes, and employing at each iteration only a fraction γ of the links. In particular, $\gamma = 1$ corresponds to the case $\mathcal{G}_k = \mathcal{G}$ for every k. As remarked in Section 4, this is equivalent to the time-independent case. The tests we present here are compare the communication and computational costs required by the five methods to solve a given linear system using the same sequence of networks $\{\mathcal{G}_k\}$. We generated the linear system as in Section 5.2 and chose $\mathcal{G}$ as the undirected m-regular graph with $n = 100$ vertices and degree $m = 8$. The same test is repeated for γ in $\{0.1, 0.2, \ldots, 1\}$. For every k the consensus matrix W^k associated with $\mathcal{G}_k$ is defined as in (41), the termination condition and all the parameters of the methods are chosen as in the previous sections. In Figs. 18–21 we plot the results (note that Fig. 19 repeats the results of Fig. 18, excluding Projection method). The computational cost and the communication traffic are calculated as described in Section 5.2. DFIX outperforms the three methods for distributed optimization both in terms of computation and communication in this framework. Comparing with Projection, for every value of the parameter γ, the computational cost of DFIX is significantly lower, but it requires a smaller amount of communication only for large values of γ (that is, when each graph $\mathcal{G}_k$ is equal or close to $\mathcal{G}$). Moreover, we can see that for all the methods except for SVL there is an optimal value of $\gamma < 1$, that minimizes the communication traffic, suggesting that using the whole graph $\mathcal{G}$ at every iteration (that is, setting $\gamma = 1$) is inefficient. A similar phenomena happens for DIGing, EXTRA and DFIX also for the computational cost (Fig. 19), while we can see in Figs. 18 and 19 that Projection and SVL methods are most efficient when all the available communication links are used at each iterations. For $\gamma < 1$ the networks $\mathcal{G}_k$ are in general not connected, but the joint connectivity of the overall sequence is enough to ensure the convergence of the methods.

6. Conclusions

A class of novel, iterative, distributed methods for the solution of linear systems of equations, derived upon classical fixed point methods. We proved linear convergence for strongly connected communication network and showed that the convergence rate depends on the diameter of the network and on the norm of the underlying iterative matrix. In particular, if the graph is strongly connected the obtained result is analogous to the classical, centralized case. The presented method is extended to the time-varying case and an analogous convergence result is proved under suitable joint connectivity assumptions, comparable with assumptions required by different methods in literature. The algorithm is compared with the relevant optimization methods

presented in Li and Qu (2017), Liu et al. (2017), Nedic et al. (2016), Shi et al. (2015), Sundararajan et al. (2019). The numerical results show good performance of DFIX in comparison with the mentioned methods. In particular, in the vast majority of the considered tests, DFIX outperformed all the methods in terms of both computational cost and communication traffic.

References

Alaviani, S. S., & Elia, N. (2020). A distributed algorithm for solving linear algebraic equations over random networks. *IEEE Transactions on Automatic Control*.

Anderson, B. D. O., Mou, S., Morse, A. S., & Helmke, U. (2016). Decentralized gradient algorithm for solution of a linear equation. *Numerical Algebra, Control and Optimization*, *6*, 319–328.

Barooah, P., & Hespanha, J. P. (2007). Estimation on graphs from relative measurements. *IEEE Control Systems Magazine*, *57*(8), 57–74.

Berman, A., & Plemmons, R. J. (1994). *Nonnegative matrices in the mathematical sciences*. SIAM.

Bertsekas, D. P., & Tsitsiklis, J. N. (2015). *Parallel and distributed computation: Numerical methods*. Athena Scientific.

Bolognani, S., Favero, S. D., Schenato, L., & Varagnolo, D. (2010). Consensus-based distributed sensor calibration and least-square parameter identification in WSNs. *International Journal of Robust and Nonlinear Control*, *20*(2), 176–193.

Cortes, J. (2009). Distributed kriged Kalman filter for spatial estimation. *Transactions on Automatic Control*, *54*(12), 2816–2827.

Cressie, N. A. C. (1993). *Statistics for spatial data*. New York: Wiley.

deGroot, M. H. (1974). Reaching a consensus. *Journal of the American Statistical Association*, *69*(345), 118–121.

Frommer, A., & Mayer, G. (1989). Convergence of relaxed parallel multisplitting methods. *Linear Algebra and its Applications*, *119*, 141–152.

Frommer, A., & Szyld, D. B. (1992). H-splittings and two-stage iterative methods. *Numerische Mathematik*, *63*, 345–356.

Frommer, A., & Szyld, D. B. (2000). On asynchronous iterations. *Journal of Computational and Applied Mathematics*, *123*, 201–216.

Greenbaum, A. (1997). *Iterative methods for solving linear systems*. SIAM.

Hendrickx, J. M., Jungers, R. M., Olshevsky, A., & Vankeerberghen, G. (2014). Graph diameter, eigenvalues, and minimum-time consensus. *Automatica*, *50*, 635–640.

Krige, D. G. (1951). A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of Chemical Metallurgical and Mining Society of South Africa*, *52*, 119–139.

Li, N., & Qu, G. (2017). Harnessing smothness to accelerate distributed optimization. *IEEE Transactions Control of Network Systems*, *5*(3), 1245–1260.

Liu, Ji., Morse, A. S., Nedić, A., & Basar, T. (2017). Exponential convergence of a distributed algorithm for solving linear algebraic equations. *Automatica*, *83*, 37–46.

Liu, J., Mou, S., & Morse, A. S. (2018). Asynchronous distributed algorithms for solving linear algebraic equations. *IEEE Transactions on Automatic Control*, *63*(2), 372–385.

Matheron, G. (1963). *Le krigeage: Memoires du Bureau de Recherches Géologiques et Miniéres, no. 24, Traité de Geostatistique Appliquée (Vol. II)*. Paris: Editions Bureau de Recherche Géologiques et Miniéres.

Mou, S., Lin, Z., Wang, L., Fullmer, D., & Morse, A. S. (2016). A distributed algorithm for efficiently solving linear equations and its applications (special issue JCW). *Systems & Control Letters*, *91*, 21–27.

Nedic, A., Olshevsky, A., & Shi, W. (2016). Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, *27*.

Saad, Y. (2003). *Iterative methods for sparse linear systems*. SIAM.

Shi, G., & Anderson, B. D. O. (2016). Distributed network flows solving linear-algebraic equations. In *Proc. of American Control Conf. Boston marriott Copley Place, July 6-8, Boston, Massachusetts, USA* (pp. 2864–2869).

Shi, W., Ling, Q., Wu, G., & Yin, W. (2015). Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, *25*, 944–966.

Sundararajan, A., Van Scoy, B., & Lessard, L. (2019). Analysis and design of first-order distributed optimization algorithms over time-varying graphs. Arxiv preprint, arXiv:1907.05448.

Touri, B., & Nedic, A. (2012). On backward product of stochastic matrices. *Automatica*, *48*, 1477–1488.

Wang, P., Mou, S., Lian, J., & Ren, W. (2019). Solving a system of linear equations: From centralized to distributed algorithms. *Annual Reviews in Control*, *47*, 306–322.

Wang, X., Zhou, J., Mou, S., & Corless, M. J. (2019). A distributed algorithm for least square solutions of linear equations. *IEEE Transactions on Automatic Control*, *64*(10), 4217–4222.

Xiao, L., Boyd, S., & Lall, S. (2006). Distributed average consensus with time-varying Metropolis weights.

Xiao, Y., & Hu, J. (2017). Distributed solutions of convex feasibility problems with sparsely coupled constraints. In *2017 IEEE 56th annual conference on decision and control* (pp. 3386–3392).

Dušan Jakovetić obtained a dipl. ing. degree from the School of Electrical Engineering, University of Belgrade, in 2007, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, and Instituto de Sistemas e Robotica (ISR), Instituto Superior Tecnico (IST), Lisbon, Portugal, in 2013. He is currently an associate professor at the Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Serbia. He was a research fellow at the BioSense Institute, Novi Sad, Serbia, from 2013 to 2015, and a postdoctoral researcher at IST from June to September 2013. His research interests include distributed inference, learning and optimization.

Nataša Krejić received the Ph.D. degree in mathematics form the University of Novi Sad, Novi Sad, Serbia. Currently, she is a full Professor at the University of Novi Sad and head of the Numerical Analysis Chair. Her research interests include numerical optimization and applications. She serves as the president of European Consortium for Mathematics in Industry for the period 2021–2023 as well as a member of the Editorial Board of Computational and Applied Mathematics , Publications de l'Institut Mathematique, and NSJOM.

Nataša Krklec Jerinkić obtained a diploma in mathematics in September 2007 and the Ph.D. degree in numerical mathematics in January 2014 from the Faculty of Sciences, University of Novi Sad. She is an Associate Professor at the Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad since July 2019. Her research interests include stochastic optimization.

Greta Malaspina received a master degree in mathematics in 2018 from the University of Florence, Italy. Since March 2019 she is a Ph.D. student at the Faculty of Sciences, University of Novi Sad, Serbia, in the Horizon 2020 MSCA project BIGMATH - "Big data challenges for Mathematics". Her research interests include large-scale and distributed optimization.

Alessandra Micheletti is Associate Professor of Probability and Mathematical Statistics at the Dept. of Environmental Science and Policy, Università degli Studi di Milano, Italy. She got a Ph.D. in computational mathematics, she is vice-president of the European Consortium for Mathematics in Industry, member of the Data Science Research Center of University of Milan and coordinator of the H2020 MSCA project BIGMATH- "Big data challenges for Mathematics". She is managing editor of the Journal of Mathematics in Industry and guest editor of a special issue entitled "Mathematical models of the spread and consequences of the SARS-CoV-2 pandemics. Effects on health, society, industry, economics and technology. "Her main research interests are in the field of Statistics and Topological Data Analysis applied to industrial and life sciences problems.