Alternating Projection Method for Intersection of Convex Sets, Multi-Agent Consensus Algorithms, and Averaging Inequalities

Alternating Projection Method for Intersection of Convex Sets, Multi-Agent Consensus Algorithms, and Averaging Inequalities

A. V. Proskurnikov[a,*] **and I. S. Zabarianska**[b,**]

[a] *Politecnico di Torino, Turin, 10129 Italy*

[b] *St. Petersburg State University, St. Petersburg, 199178 Russia*

**e-mail: avp1982@gmail.com*

***e-mail: akshiira@yandex.ru*

Received November 3, 2023; revised November 11, 2023; accepted November 20, 2023

Abstract—The history of the alternating projection method for finding a common point of several convex sets in Euclidean space goes back to the well-known Kaczmarz algorithm for solving systems of linear equations, which was devised in the 1930s and later found wide applications in image processing and computed tomography. An important role in the study of this method was played by I.I. Eremin's, L.M. Bregman's, and B.T. Polyak's works, which appeared nearly simultaneously and contained general results concerning the convergence of alternating projections to a point in the intersection of sets, assuming that this intersection is nonempty. In this paper, we consider a modification of the convex set intersection problem that is related to the theory of multi-agent systems and is called the con-strained consensus problem. Each convex set in this problem is associated with a certain agent and, generally speaking, is inaccessible to the other agents. A group of agents is interested in finding a com-mon point of these sets, that is, a point satisfying all the constraints. Distributed analogues of the alter-nating projection method proposed for solving this problem lead to a rather complicated nonlinear system of equations, the convergence of which is usually proved using special Lyapunov functions. A brief survey of these methods is given, and their relation to the theorem ensuring consensus in a system of averaging inequalities recently proved by the second author is shown (this theorem develops convergence results for the usual method of iterative averaging as applied to the consensus problem).

Keywords: alternating projection method, convex programming, Fejér mappings, distributed algo-rithms, consensus, multi-agent systems

DOI: 10.1134/S0965542524700155

1. INTRODUCTION

A number of problems in numerical analysis and optimization are reduced to finding a point in the intersection of a family of closed convex sets in a finite-dimensional or Hilbert vector space (or to establishing the fact that this intersection is empty). A special case is verifying the consistency of a system of linear equations with finding a possible solution, as well as verifying the consistency of constraints in linear programming. An explosion of interest in fast algorithms for solving such problems was associated, in particular, with the advent of computed tomography and related algorithms for fast image reconstruction (see [1]). The problem of finding a point in the intersection of half-spaces of an affine space arises in the separation of two sets of points in space, which is an important problem in machine learning (see [2, 3]) and mathematical diagnostics (see [4–6]). Convex optimization leads to problems of more general form: at the initial stage of a primal optimization method, it is necessary to find an (arbitrary) point satisfying a system of convex constraints. Finally, intersections of convex sets arise in control theory in the context of estimating parameters and trajectories of systems under unknown disturbances (see [7]) as exemplified by systems of recurrent target inequalities (see [8, 9]).

Most of the algorithms available in the literature for computing a common point of convex sets are modifications and generalizations of the alternating projection method. This method is also known as projection onto convex sets (POCS). Its history goes back to the works by Kaczmarz [10] and Cimmino [11] concerning relaxation methods for solving systems of linear equations, and for the first time (in the special case of two linear subspaces) it was presumably mentioned in von Neumann's lectures on the geometry of Hilbert spaces published in [12]. The subsequently developed theory of alternating projection methods is based primarily on the results obtained by three outstanding specialists in optimization and numerical methods, namely, Eremin, Polyak, and Bregman, whose pioneering works [13–15] appeared nearly simultaneously. On the one hand, the study of alternating iterative projections led to the development of the theory of Fejér processes (see [16–19]). On the other hand, it motivated the study of the Bregman D-distance (divergence) (see [20]), which now plays an extremely important role in machine learning, optimization, and computational geometry.

At present, there are rather detailed surveys of the convex set intersection problem (SIP) and its generalizations (specifically, the convergence of Fejér approximations), among which particular mention should be made of the monographs [21, 22], the earlier work [23], and a more specific survey of estimation theory in [7]. Some generalizations of SIP to the nonconvex case (projection on manifolds) can be found in [24], which also provides a fairly complete review of the literature on this topic. However, these surveys do not cover *multi-agent* versions of SIP, which have been investigated in recent years under various names, for example, the problem of constrained consensus (see [25]) or optimal consensus (see [26]). In these problems, each convex set is possessed by a single agent and is inaccessible to the other agents (this can be caused by privacy requirements or the complexity of describing the sets themselves, information about which may be difficult to transmit through a network for some reason). As in the original problem, the goal of a group of agents is to find a point in the intersection of all the sets so that each agent can share information only with a certain set of neighbors. Neighborhood relations are described by a communication graph, which may change and may not be fully known. Distributed algorithms for solving the multi-agent SIP were studied mainly in control theory, and they are little known to specialists from other fields, while available topic reviews are mostly limited to special cases, such as solution algorithms for linear equations (see [27]).

The present paper fills this gap and provides a brief overview of distributed algorithms for solving the multi-agent SIP. We will show that

(i) in these algorithms (as in the centralized case) the projection operators can be replaced by arbitrary Fejér (paracontracting) mappings;

(ii) convergence to a consensus in the proposed algorithms can be proved using a unified approach within the framework of the theory of *averaging inequalities* on graphs. This theory was developed in the first author's dissertation [28] (see also [29, 30]). For the reader's convenience, we also give proofs of the main results, which have not been published previously in Russian.

The rest of this paper is organized as follows. In Section 2, we formulate the SIP problem and describe alternating projection algorithms and their generalizations (Fejér processes). Additionally, we provide a brief survey of the main results concerning the convergence of these algorithms, specifically, fundamental results of Eremin, Bregman, and Gurin–Polyak–Raik. The basic material of this paper regarding the multi-agent SIP formulation is given in Section 3, which also presents some convergence results for classical averaging consensus algorithms on variable graphs and for associated systems of averaging inequalities. The paper is completed with Section 4. The main results on the convergence of multi-agent algorithms are proved in the Appendix.

2. SIP AND ITS GENERALIZATIONS

In this paper, SIP is considered in a finite-dimensional Euclidean space, but a number of the results presented below can be extended to Hilbert spaces (with some modifications related to the type of convergence of iterative approximations). Projection methods were also studied on manifolds admitting efficient computation of the projection operator (see [24]). This generalization goes beyond the scope of the present paper.

2.1. Formulation of the Problem and the Projection Operator

As a rule, the classical SIP (or the problem of consistency of convex constraints) is formulated as follows.

Problem A. Given a family of convex sets $\{\Xi_i\}_{i\in\mathcal{V}}$ in $\mathbb{R}^d$, where $\mathcal{V}$ is a finite index set, which have a nonempty intersection, i.e.,

$$\Xi_* \triangleq \bigcap_{i\in\mathcal{V}} \Xi_i \neq \varnothing, \tag{1}$$

find at least one point $\xi_* \in \Xi_*$.

A more complicated problem is to additionally identify the situation where $\Xi_* = \varnothing$.

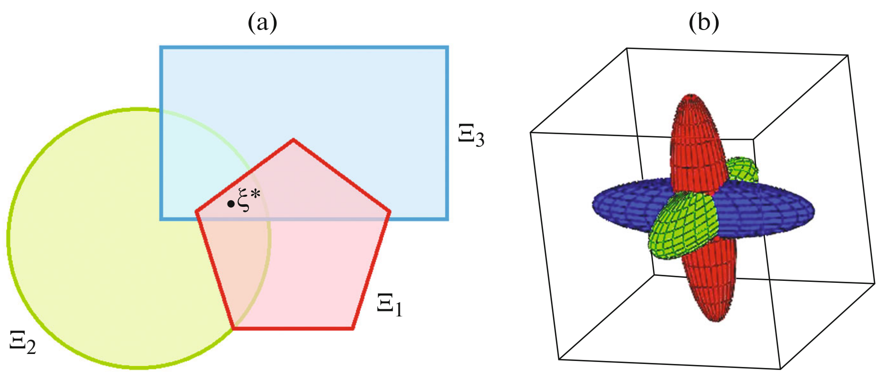

Fig. 1. Examples of intersections of convex sets: (a) in the plane and (b) in space.

Problem B. Given a family of convex sets $\{\Xi_i\}_{i\in\mathcal{V}}$, check the fulfillment of condition (1) and, if it holds, find at least one point $\xi_* \in \Xi_*$.

Note that, in the case of a one-dimensional space, i.e., $d = 1$, Problems A and B are easy to solve, since the only possible convex set in $\mathbb{R}$ is an interval (open, half-open, or closed). The intersection of a finite family of intervals is trivially found by sorting their left and right endpoints. In this case, we can determine the set Ξ_*. In the case $d = 2$, the set Ξ_* is usually easy to visualize on a computer, but completely describing its structure or analytically finding at least one of its points ξ_* is a rather complicated problem. In the case of three dimensions $d = 3$, even the visualization of Ξ_* is rather difficult (Fig. 1).

Algorithms for solving Problems A and B initially proposed in the literature were based on alternating projections. We recall some fundamental result about projection onto a convex set and the definition of a projection operator. For any closed convex set $\Omega \subset \mathbb{R}^d$ and $x \in \mathbb{R}^d$, the *projection operator* $P_\Omega : \mathbb{R}^d \to \Omega$ maps a point to the nearest element from Ω, i.e.,[1] $|x - P_\Omega(x)| = \min_{y\in\Omega}|x - y|$. The minimum of the distance is always attained, and the nearest point is unique. This statement holds in an arbitrary Hilbert space (see [31], Theorem 1.4.1).

It can be shown that[2] $\measuredangle(y - P_\Omega(x), x - P_\Omega(x)) \geq \pi/2\ \forall y \in \Omega$ (Fig. 2) and

$$|x - y|^2 \geq |x - P_\Omega(x)|^2 + |y - P_\Omega(x)|^2 \qquad \forall y \in \Omega. \tag{2}$$

In view of property (2), if $y \in \Omega$ (in other words, y is a fixed point of P_Ω), then

$$|P_\Omega(x) - y| \leq |x - y|;$$

moreover, the inequality is strict if and only if $|x - P_\Omega(x)| > 0$, i.e., $x \notin \Omega$. A continuous mapping with this property is called *Fejér* or (in the English-language literature) paracontracting. A formal definition will be given later.

2.2. First Works: Systems of Linear Equations and Inequalities

The simplest (and historically the first) example of SIP is solving a system of linear equations[3]

$$a_i^\top\xi = b_i \qquad \forall i \in \mathcal{V},$$

where $\xi \in \mathbb{R}^d$ is the unknown vector and $a_i \in \mathbb{R}^d, b_i \in \mathbb{R}$ are the coefficients of the ith equation. In this case, $\Xi_i = \{\xi : a_i^\top\xi = b_i\}$ is an affine hyperplane of dimension $d - 1$, and the search for the solution of this

[1] Throughout this paper, the symbol $|\cdot|$ denotes the usual Euclidean norm in $\mathbb{R}^d$: $|x|^2 = \sum_i x_i^2 = x^\top x$.

[2] The symbol $\measuredangle$ denotes the angle between vectors (in the range $[0, \pi]$).

[3] Here and below, elements of $\mathbb{R}^d$ are understood as column vectors; accordingly, $a^\top\xi$ is the inner product of two vectors, a and ξ.

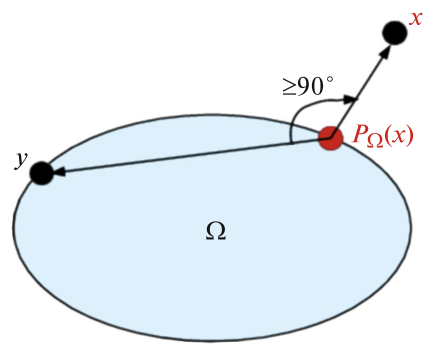

Fig. 2. Projection of the point x onto a closed convex set Ω.

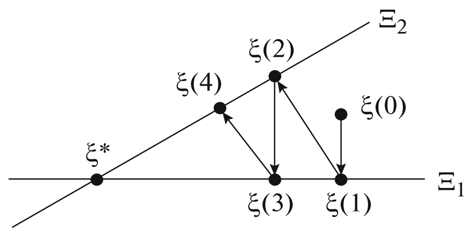

Fig. 3. Iterative Kaczmarz method.

system (more precisely, one of the solutions) can be treated as finding the intersection of these convex sets. It should be emphasized that the system of equations can be overdetermined ($d < |\mathcal{V}|$) or underdetermined ($d > |\mathcal{V}|$), the equations can be linearly dependent, and the number of equations and variables can be very large. Such systems of equations arise, for example, in computed tomography in the recovery of images from projections (see [1]), in Leontief input–output models (see [32]), and in the computation of the PageRank vector (see [33, 34]) and similar characteristics in ranking and searching problems and in finding eigenvectors. Two simple projection algorithms for solving linear equations were proposed nearly simultaneously in the 1930s by Kaczmarz and Cimmino.

2.2.1. Kaczmarz algorithm. According to the Kaczmarz algorithm[4] (see [10, 35]), a point is iteratively projected onto hyperplanes in cyclic order (in the case of two equations (i.e., $|\mathcal{V}| = 2$) and two variables (i.e., $d = 2$), the procedure is illustrated in Fig. 3).

It is easy to see that the projection operator onto the set $\Xi = \{\xi : a^\top \xi = b\}$ (where $|a| \neq 0$) is an affine map:

$$P_\Xi : x \mapsto x + \frac{b - a^\top x}{|a|^2} a. \tag{3}$$

Assume that the equations are somehow numbered from 1 to n, so that $\mathcal{V} = \{1, \ldots, n\}$, and there is an arbitrary approximation $\xi(0)$ to the solution. At the iteration with index $t = 0, 1, \ldots$, the approximation $\xi(t)$ is replaced by

$$\xi(t+1) = P_{\Xi_i}(\xi(t)) = \xi(t) + \frac{b_i - a_i^\top \xi(t)}{|a_i|^2} a_i, \quad \text{where} \quad t + 1 \equiv i \bmod n.$$

In the case of two sets, this algorithm was first mentioned in von Neumann's lectures (1933) published later as a monograph (see [12]); here, linear closed subspaces Ξ_1, Ξ_2 of a Hilbert space were considered. Von Neumann proved that the sequence $\xi(t)$ converges in the norm to the projection of the initial point

[4] This algorithm is also known as the row action method, since a single row of the constraint matrix is used at every step.

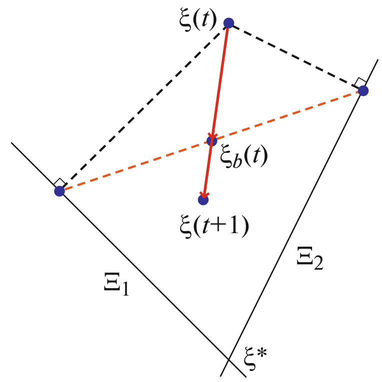

Fig. 4. Cimmino's method (illustration for the two-dimensional case).

$\xi(0)$ onto the space $\Xi_* = \Xi_1 \cap \Xi_2$ (see [12], Theorem 10.7). The convergence is exponential, and its rate is determined by the squared cosine of the angle between Ξ_1 and Ξ_2 (see [36]). This result, i.e., the exponential convergence of $\xi(t)$ to the projection $P_{\Xi_*}(\xi(0))$ remains valid $n > 2$ subspaces, but only upper bounds for the convergence rate are known in this case (see [22, 23, 36]), and, generally speaking, the convergence rate is not determined only by the angles between all possible pairs of subspaces (Ξ_i, Ξ_j). More details on the history of the Kaczmarz algorithm and its generalizations obtained in recent years can be found in [35].

2.2.2. Cimmino's algorithm. Cimmino's iterative algorithm was proposed in [11] and, in some sense, it is the prototype of the distributed multi-agent algorithms considered in Section 3. In contrast to the Kaczmarz algorithm, $\xi(t)$ at the iteration is projected onto all hyperplanes simultaneously. Then the center of mass $\xi_b(t)$ of the resulting projections is computed, which can be used as the next approximation $\xi(t+1)$. More generally, the next approximation can be obtained as

$$\xi(t+1) = \xi(t) + \alpha(\xi_b(t) - \xi(t)),$$

where $\alpha \in (0,2)$ is a fixed step size (Fig. 4).

A great advantage of Cimmino's algorithm is that the projection operations can be parallelized on several processors. However, the number of iterations required for finding a solution with the prescribed accuracy can be very large. There are accelerated versions of the algorithm, for example, the block Cimmino algorithm (see [37]) for sparse systems of equations; their consideration goes beyond the scope of this paper.

2.2.3. The Agmon–Motzkin–Schoenberg (AMS) algorithm for intersection of half-spaces. An important stage in the study of SIP was the works by Agmon [38] and Motzkin and Schoenberg [39], which appeared simultaneously. As was mentioned in [38], the idea underlying the algorithm was due to Motzkin. In contrast to Kaczmarz's and Cimmino's works, a system of linear inequalities is considered, i.e., a point is sought in the intersection of the half-spaces

$$\Xi_i = \{\xi \in \mathbb{R}^d : a_i^\top \xi \le b_i\}.$$

The projector onto the half-space $\Xi = \{\xi : a^\top \xi \le b\}$ is easy to compute. However, it is no longer an affine map:

$$P_\Xi : x \mapsto x' \triangleq \begin{cases} x + \dfrac{b - a^\top x}{|a|^2} a, & a^\top x > b, \\ x, & a^\top x \le b. \end{cases} \quad (4)$$

Obviously, the equality $a_i^\top \xi = b_i$ can be written as a pair of inequalities, so solutions of an arbitrary system of linear equations and inequalities can be found by applying the method from [38, 39] (assuming that such a solution exists). Note that numerous "linear" problems of learning (e.g., algorithms for tuning perceptron weights), classification, and mathematical diagnostics (see [2, 4–6]) are reduced to solving inequalities. In these problems, the task is to find a hyperplane separating two groups of points in a mul-

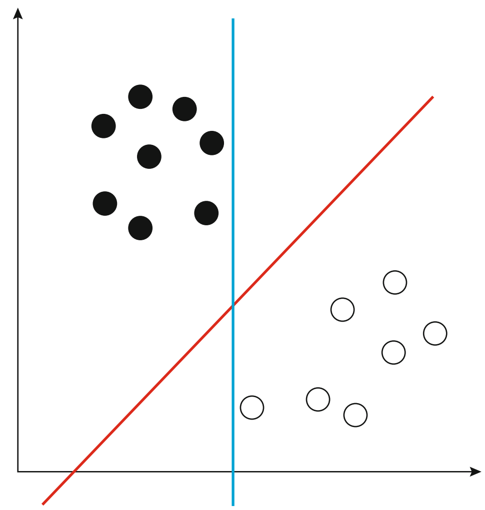

Fig. 5. Affine hyperplanes separating two sets of points in space.

tidimensional space (Fig. 5). If the hyperplane is defined by a linear equation, the separation condition is a system of inequalities on the coefficients of this equation.

The implementation of the method from [38, 39] resembles the Kaczmarz algorithm (with the projector given by (4), rather than by (3)), but there is a substantial difference: at every iteration $t = 0,1,\ldots$, the projection is performed onto the most *distant* set. Formally, it is necessary to find the index[5]

$$m(t) = \arg\max_{i\in\mathcal{V}} d(\xi(t),\Xi_i), \tag{5}$$

where $d(\xi,\Xi) \triangleq \min\{\|\xi - x\| : x \in \Xi\}$. Projection is performed onto the half-space with index $m(t)$, so that the next approximation has the form

$$\xi(t+1) = P_{\Xi_{m(t)}}(\xi(t)).$$

In [38, 39] it was proved (assuming that the sets have a nonempty intersection) that this algorithm and its relaxed version

$$\xi(t+1) = (1-\alpha)\xi(t) + \alpha P_{\Xi_{m(t)}}(\xi(t)), \quad \alpha \in (0,2), \tag{6}$$

converge exponentially. Note that the convergence rate can be determined explicitly (see [38]) and depends on the matrix A, although its computation is nontrivial. In terms of complexity, an AMS iteration is equivalent to a Cimmino iteration, since the distances to all half-spaces have to be computed before projection.

2.3. Alternating Projection Method for Arbitrary Sets

Alternating projection methods for finding a common point of arbitrary closed convex sets most frequently rely on the idea of the Kaczmarz algorithm (iterative projections onto sets in cyclic order, see Fig. 6) and the AMS algorithm (in this case, the most distant set is chosen at every step).

Since this class of methods has been addressed in numerous publications (see, e.g., the surveys in [22, 23]), we mention only major works that led to a number of studies in constrained convex optimization.

Historically, the first was Eremin's paper [40] (written and submitted in 1962), where the AMS method (5), (6) was applied to arbitrary closed convex sets Ξ_i. It was shown that the method converges if the sets have a nonempty intersection, and, vice versa, if the sequence $\xi(t)$ converges, then its limit automatically belongs to the intersection Ξ_* of all the sets.

Bregman [14] investigated the convergence of algorithms with iterative projections in cyclic order (as in the Kaczmarz algorithm) and with projections onto the most distant convex set, as in the AMS algo-

[5] In the case of nongeneric position, when a point is equidistant from several sets and arg max is defined ambiguously, it is possible to choose an arbitrary value; the smallest index is usually picked (see [13]).

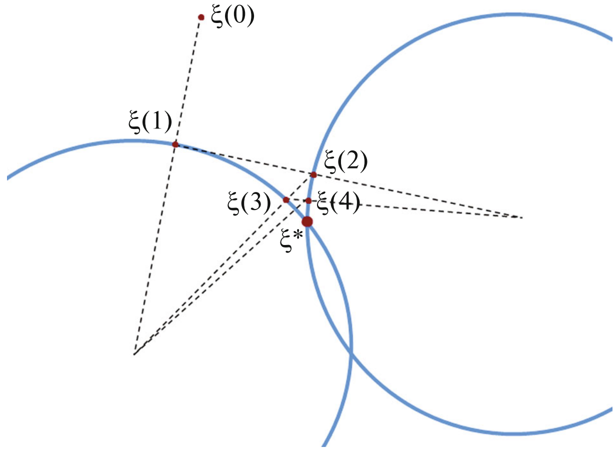

Fig. 6. Projections in cyclic order (illustration for two sets).

rithm. The convergence was analyzed in a Hilbert space, but it was understood only in the weak sense (in the case of a finite-dimensional space, it is equivalent to norm convergence) and was proved assuming that the sets have a nonempty intersection. Moreover, the convergence of the modified AMS algorithm in the case of half-spaces $\Xi_i = \{\xi : a_i^\top \xi \leq b_i\}$ was studied. In this modified method, projection is performed not onto the most distant half-space Ξ_i, but rather onto the set for which the constraint $a_i^\top \xi \leq b_i$ at the step t is violated to the highest degree; in other words, (5) is replaced by the condition

$$m(t) = \arg\max_{i \in \mathcal{V}} [a_i^\top \xi(t) - b_i],$$

while the projection rule (6) remains unchanged.

A further generalization of Eremin's and Bregman's results was obtained by Gurin,[6] Polyak, and Raik [15]. Following [14], they investigated alternating projection algorithms of two types (in cyclic order and onto the most distant set) for closed convex sets in an arbitrary Hilbert space. Conditions for *exponential* convergence (*in the norm of the Hilbert space*) of both algorithms were obtained in the case when each of the sets Ξ_i has a nonempty intersection with the *interiors* of the other sets:

$$\Xi_i \cap \bigcap_{j \neq i} \operatorname{int} \Xi_j \neq \varnothing \quad \forall i \in \mathcal{V}.$$

This condition can be dropped if Ξ_i are half-spaces (thus, the result of [38] was generalized to the infinite-dimensional case). Additionally, a convergence condition in the norm ("uniform convexity" of Ξ_i) was presented. In all these results, the exact projection operator can be replaced by its "relaxed" version, as in algorithm (6) (moreover, the step size α can be different at every iteration of the algorithm). In the case of an empty intersection of the sets Ξ_i, it was shown that the cyclic projection algorithm converges exponentially to a periodic trajectory. An accelerated version of the AMS method was also proposed (projection onto the most distant set). A number of interesting applications of the alternating projection method, specifically, to polynomial approximation of a function on an interval and to a special optimal control problem were described in [15].

Further development: Fejér operators and optimization. The above-considered algorithms for finding a common point of convex sets are based on the implicit assumption that the projection operator onto each of the sets can be effectively computed. In the case of linear hyperplanes and half-spaces, this operator has an analytical representation, while for more complicated sets (e.g., defined by a nonlinear convex inequality $f(\xi) \leq 0$) the projection can be computed only numerically and, generally speaking, the computation is reduced to a convex optimization problem. However, even in the first works (see [10, 13, 38, 39]) it was noted that the projection operator can be replaced by the relaxed projection (6), and, moreover, the basic

[6] Unfortunately, when the translation of the original version of [15] was digitized, the first author's surname was erroneously written as Gubin. This spelling mistake is still present in ScienceDirect and is contained in the majority of surveys of algorithms for SIP.

property used in the proof is that it has a weak contraction property and any point outside the chosen set Ξ approaches this set. This observation led to the development of the theory of *Fejér processes*, the foundations of which were laid by Eremin in his pioneering works [13, 16, 41–43]. The most complete description of this theory can be found in [21]. As will be shown in Section 3, the convergence results for iterative procedures with Fejér operators can be generalized to the multi-agent case (see Theorem 1). In the next subsection, we consider Fejér mappings in more detail and give some examples.

Frequently, it is of interest to find not an arbitrary point in the intersection of convex sets, but rather a point that optimizes some functional (possibly nonsmooth). The possibility of solving linear programming problems with the help of the alternating projection method was noted by Bregman (see [14]), who later considered iterative "projection" methods (see [20, 44]) in the sense of some "pseudometric" on a linear space, which was called the D-distance. The most frequently used type of D-distances is known as the Bregman *divergence* and is given by the formula

$$D_\varphi(x, y) = \varphi(x) - \varphi(y) - \nabla\varphi(y)(x - y)$$

for some convex differentiable function φ. Bregman showed (see [20]) that projection methods are able to solve convex programming problems with equality and inequality constraints in the sense of a suitably constructed D-distance. An alternative approach proposed by Eremin (see [42]) leads to Fejér shifted mappings. Note that optimization algorithms of this type have been poorly studied in the multi-agent case. As a rule, alternative algorithms obtained by decentralizing gradient projection methods are used (see [25, 45, 46]). Frequently, constraints are also replaced by barrier functions. The study of optimization algorithms with constraints goes beyond the scope of this paper. A fairly complete exposition of modern methods can be found, for example, in [47].

2.4. Fejér (Paracontracting) Mappings

Throughout this subsection, we assume that $P : \mathbb{R}^d \to \mathbb{R}^d$ is a given operator (generally speaking, nonlinear) and $\|\cdot\|$ is a given norm on $\mathbb{R}^d$ (not necessarily Euclidean). Recall that, in a finite-dimensional space, all norms are equivalent and generate the same topology. Accordingly, the concepts of openness, closedness, and continuity in various norms are also equivalent.

Definition 1. An operator P is called *Fejér*[7] with respect to a set $M \subset \mathbb{R}^d$, or M*-Fejér* if the following three conditions are satisfied:

(a) P is a continuous mapping on the entire space;

(b) the set M is fixed with respect to P;

(c) the operator satisfies the contraction condition with respect to M, i.e.,

$$\|P(\xi) - \xi_0\| = \|P(\xi) - P(\xi_0)\| < \|\xi - \xi_0\| \quad \forall \xi \not\in M, \quad \xi_0 \in M. \tag{7}$$

Obviously, it follows from (7) that P has the property

$$\|P(\xi) - P(\xi_0)\| \le \|\xi - \xi_0\| \quad \forall \xi \in \mathbb{R}^d, \quad \xi_0 \in M. \tag{8}$$

In the English-language literature, Fejér operators are often called paracontracting (see [49]); there are also other terms, for example, an operator attracting to the set M (see [23]). Obviously, if $\|x - y\| < \|P(x) - y\|$, then $x \ne P(x)$; thus, M in Definition 1 coincides with the set of all of fixed points of the M-Fejér operator $\mathcal{F}(P) \triangleq \{\zeta : P(\zeta) = \zeta\}$; therefore, if no explicit specification of this set is required, for brevity, we call the operator Fejér. It is also easy to see that the set M is automatically closed and convex. The first property follows from the continuity of the operator P: if a sequence $\{x_n\}$ of elements of M converges to some limit $x \in \mathbb{R}^d$, then, obviously, for an arbitrary element $a \in M$, we have $\|x - a\| = \lim_{n\to\infty} \|x_n - a\| = \lim_{n\to\infty} \|P(x_n) - a\| = \|P(x) - a\|$, which is possible only if $x \in M$. The second property is also easy to prove: if z is an arbitrary element of the segment joining points $x, y \in M$, then $\|x - y\| = \|x - z\| + \|z - y\| \ge \|x - P(z)\| + \|P(z) - y\| \ge \|x - y\|$. This inequality holds only if $z \in M$ (other-

[7] The term "M-Fejér operator" introduced by I.I. Eremin was motivated by the concept of a Fejér monotone (with respect to M) sequence introduced in [39]. A sequence $\{x_n\} \subset \mathbb{R}^d \backslash M$ is called so if $x_{n+1} \ne x_n$ for all n and $\|x_n - y\| \ge \|x_{n+1} - y\|$ for all n and $y \in M$. This concept, in turn, goes back to Fejér's work [48] on the location of roots of polynomials.

wise, $\|x - P(z)\| < \|x - z\|$ and $\|y - P(z)\| < \|y - z\|$). Therefore, together with two points, M contains a segment joining them. A Fejér mapping is a natural generalization of a contraction mapping P (in terms of Banach), for which $\|P(x) - P(y)\| \le q\|x - y\|\ \forall x, y \in \mathbb{R}^d$ with some constant $q \in (0,1)$. A fundamental difference between contraction and Fejér mappings is that the former have exactly one fixed point (by the well-known Banach theorem), while a Fejér mapping can have an arbitrary closed convex set of fixed points. Examples of Fejér operators are given in the next subsection.

2.4.1. Examples of Fejér operators.

Example 1. As was shown above, the projector P_Ω onto a closed convex set is a Fejér operator (in the standard Euclidean norm). More generally, it can be shown that for any coefficient $a \in (0,2)$ the operator $(1 - a)\mathrm{Id} + aP_\Omega$, where Id is the identity map, is also a Fejér operator in the Euclidean norm (see, for example, Lemma 2 in [40] and Corollary 2.5 in [23]).

Example 2. A convex combination of several Fejér mappings is a Fejér mapping. Let $P_1, \dots, P_n$ be Fejér operators with fixed point sets $M_1, \dots, M_n$, respectively, and let $a_1, \dots, a_n > 0$ be coefficients such that $\sum_{i=1}^n a_i = 1$. Then it is easy to see that $P = \sum_i a_i P_i$ is a Fejér operator with respect to $M = M_1 \cap \dots \cap M_n$ (assuming that M is nonempty). By combining this example with Example 1, we can show that the approximation $\xi(t+1)$ in Cimmino's algorithm described in Subsection 2.1 is obtained from $\xi(t)$ by applying a Fejér operator.

Example 3. Similarly, it is easy to verify that the composition of Fejér operators $P_n \circ \dots \circ P_1$ is also Fejér with respect to the intersection of the fixed point sets $M = M_1 \cap \dots \cap M_n$ (assuming that M is nonempty). Thus, n successive operations (where n is the number of sets) in the Kaczmarz algorithm are equivalent to the application of a Fejér operator.

Example 4. Let P be a nonexpanding operator in some strongly convex norm[8] $\|\cdot\|$ ($\|P(x) - P(y)\| \le 1$ for all $x, y \in \mathbb{R}^d$) with a nonempty fixed point set $M = \mathcal{F}(P)$. Then the operator $(1 - a)\mathrm{Id} + aP$ with $a \in (0,1)$ is also nonexpanding and M-Fejér. The first property is verified straightforwardly. Additionally, if $y \in M$ and $x \notin M$, then $P(x) - y \neq x - y$; therefore, one of the following three cases holds: $\|P(x) - y\| < \|x - y\|$, the vectors $P(x) - y$ and $x - y$ are noncollinear, or $P(x) - y = y - x$. In all the cases,

$$\|(1 - a)x + aP(x) - y\| = \|(1 - a)(x - y) + a(P(x) - y)\| < \|x - y\|.$$

In view of the remark made above, we obtain a well-known fact, namely, the set of fixed points of a nonexpanding (with respect to a strongly convex norm) mapping is always convex (see [23]).

Examples 2–4 are interesting in that, although the projection onto the set M can be difficult to calculate explicitly, a Fejér operator can be known under which a point not lying in M approaches this set. Actually, there are numerous examples of this type. The following one is concerned with the widespread case when a convex set is defined by a scalar convex inequality.

Example 5. Consider a continuously differentiable convex function $f : \mathbb{R}^d \to \mathbb{R}$ and assume that the set $M_f \triangleq \{x : f(x) \le 0\}$ is nonempty (obviously, it is convex and closed). Given a constant $\alpha \in (0,2)$, consider the mapping

$$P_{\alpha,f} : x \mapsto x' \triangleq \begin{cases} x - \alpha f(x)\dfrac{\nabla f(x)}{|\nabla f(x)|^2}, & f(x) > 0, \\ x, \quad f(x) \le 0. \end{cases}$$

Note that the gradient ∇f, in view of the convexity, can vanish only at (global) minimizers of f, and all minimizers (if they exist) must belong to M_f. In view of this, $P_{\alpha,f}(x)$ is well defined for $f(x) > 0$. It is also easy to show that $P_{\alpha,f}$ is a continuous (but, generally speaking, nondifferentiable) mapping with the fixed point set M_f. An important observation made in [13] is that $P_{\alpha,f}$ is a Fejér operator with respect to the Euclidean norm. This construction extends trivially to nondifferentiable convex mappings for which the subdifferential has a continuous branch. More precisely, if there exists a continuous mapping

[8] A norm is strictly convex if the strict triangle inequality $\|x + y\| < \|x\| + \|y\|$ holds for any linearly independent vectors x, y.

$g : \mathbb{R}^d \to \mathbb{R}^d$ such that $g(x) \in \partial f(x)$ for all x, then the gradient $\nabla f(x)$ in the definition of $P_{\alpha,f}$ can be replaced by $g(x)$ (see [13]).

Other examples of Fejér mappings are proximal operators and gradient descent operators corresponding to strongly convex functions (see [49]).

2.4.2. Iterations of Fejér operators. The following simple lemma shows that iterations of a Fejér operator, like iterations of a contracting (in the sense of Banach) mapping, converge to a fixed point.

Lemma 1. *Let $P : \mathbb{R}^d \to \mathbb{R}^d$ be a Fejér operator with a nonempty fixed point set $M = \mathcal{F}(P)$. Then the sequence $(\xi(t))_{t=0}^{\infty}$ defined by the recurrence relation*

$$\xi(t+1) = P(\xi(t)), \quad t = 0,1,\ldots,$$

with an arbitrary initial point $\xi(0)$ converges to an element of M.

Lemma 1 is proved in the Appendix. In view of Examples 2 and 3, it follows from Lemma 1 that the Kaczmarz and Cimmino algorithms converge (for projectors onto arbitrary sets whose structure is not necessarily linear). The concept of a Fejér operator and Lemma 1 can be generalized to set-valued mappings (arising, for example, in the AMS algorithm if the index $m(t)$ in (5) is allowed to take several values). The most general results on the convergence and divergence of Fejér processes can be found in [17, 21, 23, 50].

3. MULTI-AGENT ALGORITHMS FOR THE INTERSECTION OF SETS

Now we consider the multi-agent[9] formulation of SIP. In this problem, each closed convex set $\Xi_i \subseteq \mathbb{R}^d$ is associated with its own agent (which can be a software module, robot, or living being with some autonomy in decision making). Frequently, this set is treated as a constraint imposed by agent i. We assume that agent i has sufficient information on its set $\Xi_i \subseteq \mathbb{R}^d$. Specifically, agent i can compute the projection $P_{\Xi_i}(\xi)$ of an arbitrary point $\xi \in \mathbb{R}^d$ onto the set Ξ_i or, more generally, the value of some Fejér operator $P_i(\xi)$ with the fixed point set Ξ_i. However, agent i has no access to information about the other agents' sets Ξ_j, $j \neq i$. This can be caused by the complex structure of the sets, so that information about them is difficult to transmit through a network, or by privacy requirements (the sets can depend on confidential information). The goal of the agents is to find a point $\xi_* \in \Xi_* \triangleq \bigcap_{i \in \mathcal{V}} \Xi_i$ satisfying all the constraints.

Assuming that the agents can communicate unlimitedly, a common point can theoretically be found by applying the usual alternating projection method (generalized Kaczmarz algorithm). For this purpose, the agents have to be somehow indexed and, after each projection operation, the resulting approximation $\xi(t)$ to the point has to be transmitted to the next agent (in cyclic order). Agent 1 projects the initial point $\xi(0)$ onto the set Ξ_1 or, more generally, calculates some Fejér operator $P_1(\xi(0))$ and transmits the result to agent 2, who applies some operator P_2 and so on; agent n applies an operator P_n and transmits the result to agent 1. However, this procedure is unsatisfactory for a number of reasons. First, being formally independent, the agents cannot carry out their operations in parallel: each of them is idle for $n - 1$ out of n successive iterations. Second, according to the paradigm of multi-agent systems, the agents generally have to be equivalent and they should not have distinctive features such as a unique index. Accordingly, a special procedure is required to obtain an ordering of the agents from 1 to n. Finally, as was mentioned above, at step t, there should be the possibility of communication between the agents with indices $1 + (t \bmod n)$ and $1 + (t + 1 \bmod n)$. In practice, this is not always guaranteed: it is possible that the communication graph is incomplete, changes with time, or unknown in advance (e.g., some communication lines may be switched off due to failures or the necessity of transmitting more important information).

From the point of view of a multi-agent implementation, it is more convenient to use the generalized Cimmino method (projection onto all sets and computation of the barycenter) and the generalized AMS method (finding the most distant set). However, they assume that the agents can communicate with some central node, which collects the values $P_i(\xi(t))$ and then computes their barycenter (or the most distant

[9] In this survey. we do not consider the general philosophy of the theory of multi-agent systems and do not give a rigorous definition of agents, which differs in control theory and computer sciences. The historical background and the main problems in the theory of multi-agent systems (also known as network systems) can be found, for example, in [51–57].

projection) to find $\xi(t+1)$. Such a central node in a network is not always possible; moreover, the complexity of such a centralized system increases with the number of agents.

In view of what was said above, alternative algorithms were proposed for solving the multi-agent SIP. Their underlying idea is similar to Cimmino's algorithm, but they are based on the concept of a *consensus* reached by computing iterative averages (a short introduction to consensus algorithms is given in the next subsection). In contrast to usual projection algorithms, each agent i has its own state $\xi^i(t) \in \mathbb{R}^d$, which is available at every step to its neighbors (in terms of the communication graph), is updated depending on the neighbors' states, and eventually converges to a point in the intersection of the sets; moreover, this steady state is identical for all the agents.

The rest of this section is organized as follows. A brief description of consensus algorithms and a condition for reaching a consensus in a multi-agent network with a variable directed graph is given in Subsection 3.1. We will use this result in a somewhat stronger form (robustness of a consensus to vanishingly small perturbations). Another important result—a lemma about a consensus guaranteed by averaging inequalities—is presented in Subsection 3.2. In Subsection 3.3, we describe several classes of algorithms proposed in the literature for solving the multi-agent SIP and formulate the main result of this paper, which is proved in the Appendix.

3.1. Classical Consensus Algorithms and Preliminaries from Graph Theory

Averaging algorithms have a long story going back, on the one hand, to the dynamics of Markov processes (see [58]) and, on the other hand, to multi-agent (microscopic) models of opinion dynamics (see [59, 60]). In this subsection, we briefly describe linear averaging algorithms and present the basic (to date) condition for reaching a consensus.

3.1.1. Dynamics of iterative averaging. Consensus. Consider a set $\mathcal{V}$ of agents, each described by a scalar variable $x_i(t)$, $i \in \mathcal{V}$, called the agent's *state*, which varies in discrete time $t = 0,1,\ldots$. At every step t, each of the agents calculates the *weighted average* of its own state and the other agents' states. However, the agent has information not about all states, but rather about agents from some subset $\mathcal{V}_i(t) \subseteq \mathcal{V}$, which we call the *neighbor* set of agent i at the time t. Throughout this paper, we assume that $i \in \mathcal{V}_i(t)$. At every time $t \in \mathbb{N}$, agent i assigns *averaging weights* $a_{ij}(t)$ to all its neighbors. The weights of each agent i satisfy the constraints

$$a_{ij}(t) \geq 0 \quad \forall j \in \mathcal{V}_i(t), \quad \sum_{j \in \mathcal{V}_i(t)} a_{ij}(t) = 1. \tag{9}$$

Then the agent updates its value as follows:

$$x_i(t+1) = \sum_{j \in \mathcal{V}_i(t)} a_{ij}(t) x_j(t), \quad t = 0,1,\ldots. \tag{10}$$

Setting $a_{ij}(t) \triangleq 0$ for $j \not\in \mathcal{V}_i(t)$ yields a *stochastic* matrix of averaging weights $A(t) = (a_{ij})_{i,j \in \mathcal{V}}$. It is convenient to rewrite (10) in compact matrix form as

$$x(t+1) = A(t)x(t), \quad t = 0,1,\ldots, \tag{11}$$

where $x(t)$ denotes the column vector made up of the agents' states $x_i(t)$.

In the case of a constant matrix A, the behavior of system (11) is obviously determined by the structure of the matrix A^t, which has been well studied in the theory of Markov chains. In applications, however, it is utterly important to have the possibility of working with a variable matrix, since communication channels between the agents can open and close, resulting in changed neighborhood relations (sets $\mathcal{V}_i$) and, as a consequence, in the necessity to redistribute the weights. In the general case, the behavior of systems with a variable matrix, or infinite matrix products, remains poorly studied. The main results can be found in [61, 58] and in more recent works [62–64]. Below, we present only one result, which will be used in what follows.

At every step, the agent's state approaches its neighbors' states, so it can be expected that all states eventually become identical; in other words, a consensus is reached:

$$\bar{x}_1 = \ldots = \bar{x}_n, \quad \text{where} \quad \bar{x}_i \triangleq \lim_{t \to \infty} x_i(t) \quad \forall i \in \mathcal{V}. \tag{12}$$

Sometimes a consensus is defined in a formally weaker sense, for example, as asymptotic synchronization of the agents' values:

$$\lim_{t\to\infty} \max_{i,j\in\mathcal{V}} |x_j(t) - x_i(t)| = 0. \tag{13}$$

Note that, in our case, the last requirement is equivalent to a "strong" consensus for algorithm (11). This fact is well known in the theory of Markov chains (see [61], Theorem 4.17) (in this case, synchronization is weak ergodicity of the inverse matrix product, which is equivalent to strong ergodicity, thus implying a consensus). This is associated with the fact that, in view of Eqs. (10), the maximum (minimum) element of the vector $x(t)$ does not increase (does not decrease, respectively) with growing t and converges to a finite limit. Therefore, it is easy to see that

$$\lim_{t\to\infty} \max_{i,j\in\mathcal{V}} |x_j(t) - x_i(t)| = \lim_{t\to\infty} \max_i x_i(t) - \lim_{t\to\infty} \min_i x_i(t);$$

hence, (13) is equivalent to $\lim_{t\to\infty} \max_i x_i(t) = \lim_{t\to\infty} \min_i x_i(t)$, i.e., all the elements of $x(t)$ converge to the same finite value.

In what follows, (10) (or (11)) is referred to as the *iterative averaging algorithm* defined by the sequence of stochastic matrices $A(t)$. Although the matrices $A(t)$ "hide" the structure of the original sets $\mathcal{V}_i(t)$ ("neighborhood" relations), in fact, it is these sets that are primary, since they determine the information structure of the multi-agent algorithm (which agents send their states to which agents), while the weights of the averaging are significantly less important in the consensus problem considered below, although they cannot be chosen completely arbitrarily. Instead of the matrix, one often speaks of the (weighted) influence graph between the agents. In the following subsection, we give definitions from graph theory.

3.1.2. Influence graphs and related concepts.

Definition 2. A *graph* (formally, a directed or oriented graph) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a pair of two finite sets $\mathcal{V}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ (which can be arbitrary). Elements of $\mathcal{V}$ are vertices of the graph, and elements of $\mathcal{E}$ are its edges. It is usually assumed that the vertices are indexed from 1 to n, so that $\mathcal{V} = [1:n]$. The edge (i, j) joins vertex i to vertex j (denoted as $i \to j$). In all graphs considered below, each vertex has a *loop*, i.e., an edge (i, i).

The following two concepts refer to types of connectivity of (directed) graphs.

Definition 3. A sequence of edges $v_0 \to v_1 \to v_2 \to \dots \to v_s$ is called a route joining the vertex v_0 to the vertex v_s, and the number of edges in the route s is called its length. A graph is said to be *strongly connected* if each pair of its distinct vertices is joined by a route. A graph is called *quasi-strongly connected* if there are routes from some vertex (called the root) to the other vertices.

Note that a quasi-strongly connected graph has an outgoing *spanning tree* (see [65]) and, vice versa, the existence of an outgoing spanning tree in a graph implies that the graph is quasi-strongly connected. Accordingly, quasi-strong connectivity in consensus criteria is often formulated as the existence of a spanning tree in the corresponding graph (see [66, 67]).

As is customary in the theory of multi-agent systems, we consider graphs whose vertices are associated with agents and the edge $j \to i$ is associated with the influence of agent j on agent i at a given time. Concerning algorithm (10), the influence means that the state of agent j participates in the formation of the next state of agent i, i.e., $a_{ij}(t) \neq 0$. Accordingly, the weight $a_{ij}(t)$ can be treated as the degree of influence or the weight of the corresponding edge in the graph. This motivates the following definition.

Definition 4. Given a square matrix $A = (a_{ij})_{i,j\in\mathcal{V}}$, we define an associated graph $(\mathcal{V}, \mathcal{E}[A])$ whose vertices are in one-to-one correspondence with the index set $\mathcal{V}$ and $\mathcal{E}[A] = (j, i) : a_{ij} \neq 0$. In what follows, the triple $\mathcal{G}[A] \triangleq (\mathcal{V}, \mathcal{E}[A], A)$ is called the *weighted graph* determined by the matrix A.

3.1.3. Assumptions about influence weights. Although the choice of the weight matrix is, in fact, significantly less important than the structure of the influence graph, a consensus cannot be reached with an arbitrary choice of weights. For example, the cyclic algorithm

$$x_1(t+1) = x_2(t), \quad x_2(t+1) = x_3(t), \quad \dots, \quad x_n(t+1) = x_1(t)$$

not only fails to reach a consensus, but also does not converge at all in the case of pairwise distinct initial states $x_i(0)$, since the vector $x(t+1)$ is obtained from $x(t)$ by a cyclic permutation. To eliminate such behavior, we need some conditions similar to aperiodicity of Markov chains. The most widespread condition imposed on matrices in most works is that the diagonal elements of the stochastic matrix are uniformly positive:

$$a_{ii}(t) \geq \eta > 0 \quad \forall i \in \mathcal{V} \quad \forall t. \tag{14}$$

Additionally, it is clear that, to reach a consensus, we need that the relation between the agents should not be broken completely; moreover, the influence weights cannot become extremely small over time. This can easily be illustrated by an example of two agents and matrices: $a_{11}(t) = a_{22}(t) = 1 - a(t)$ and $a_{12}(t) = a_{21}(t) = a(t)$. It is easy to see that

$$x_1(t+1) - x_2(t+1) = (1 - 2a(t))(x_1(t) - x_2(t)) = \dots = (x_1(0) - x_2(0))\prod_{s=0}^{t}(1 - 2a(s)).$$

If $|a(t)| \neq 1/2$ for all t and $\sum_{t\geq 0} a(t) < \infty$, then the product on the right-hand side converges to a nonzero value; accordingly, no consensus is reached.

In this context, a "stronger" connectivity of the influence graph is usually introduced, which has to be preserved when too weak relations between the agents are discarded. We consider only the most widespread property of uniform (or recurring) connectivity.

Definition 5. Given a nonnegative matrix $A \in \mathbb{R}^{\mathcal{V}\times\mathcal{V}}$, its ε-*skeleton* is the matrix A^ε obtained by setting all elements smaller than ε to zero, i.e.,

$$a_{ij}^\varepsilon \triangleq \begin{cases} a_{ij} & \text{if} \quad a_{ij} \geq \varepsilon, \\ 0 & \text{otherwise.} \end{cases}$$

The weighted graph $\mathcal{G}[A^\varepsilon]$ is called the ε-skeleton of the weighted graph $\mathcal{G}[A]$. We say that a weighted graph is (quasi-)strongly ε-connected if its ε-skeleton is (quasi-)strongly connected.[10]

Definition 6. Given a sequence of matrices $A(t) \in \mathbb{R}^{\mathcal{V}\times\mathcal{V}}$, where $t \in [t_0 : t_1]$, the *union* of the corresponding graphs over $[t_0 : t_1]$ is defined as the graph

$$\bigcup_{t=t_0}^{t_1} \mathcal{G}[A(t)] \triangleq \mathcal{G}\left[\sum_{t=t_0}^{t_1} A(t)\right].$$

The graphs $\mathcal{G}[A(t)]$, $t \in \mathcal{T}$, where $\mathcal{T} = [t_0 : t_1]$, are called *jointly* (quasi-)strongly ε-connected on the interval $\mathcal{T}$ if their union on $\mathcal{T}$ is (quasi-)strongly ε-connected.

Definition 7. A time-varying graph $\mathcal{G}[A(\cdot)]$ is *uniformly* (quasi-)strongly connected if there are constants $\varepsilon > 0$ and $T > 0$ such that the graphs $\mathcal{G}[A(t)]$ are jointly (quasi-)strongly ε-connected at each time $[t_0 : t_0 + T]$, $t_0 \geq 0$.

Note that the weights of the average are often subjected to an additional constraint, which usually holds in practice. Specifically, it is assumed that the nonzero elements of the weight matrix cannot be too small:

$$a_{ij}(t) \in \{0\} \cup [\varepsilon, 1] \quad \forall t = 0, 1, \dots. \tag{15}$$

Then uniform connectivity means that the union of the graphs over a sufficiently long period T is connected (in the strong or quasi-strong sense).

3.1.4. Sufficient condition for a robust consensus. In the study of multi-agent algorithms for the intersection of sets, we will use the following condition for a consensus, which actually guarantees that the consensus is robust to perturbations vanishing at infinity. Along with (11), we consider the perturbed system

$$x(t+1) = A(t)x(t) + f(t), \quad t = 0, 1, \dots. \tag{16}$$

Lemma 2 *Let* $A(t)$ *be stochastic matrices with uniformly positive diagonal elements* (14). *Then algorithm* (11) *establishes consensus* (12) *if the variable graph* $\mathcal{G}[A(\cdot)]$ *is uniformly quasi-strongly connected. Moreover, if the perturbation* $f(\cdot)$ *vanishes at infinity, i.e.,* $f(t) \xrightarrow[t\to\infty]{} 0$, *then*

$$\max x(t) - \min x(t) \xrightarrow[t\to\infty]{} 0 \tag{17}$$

[10]In other words, a graph is (*quasi-*)*strongly* ε-connected if it remains (quasi-)strongly connected after deleting all arcs with a weight smaller than ε.

for any solution of (16).

Lemma 2 is usually proved under the additional assumption (15); in this form, it can be found in [45]. As was shown in [28], this condition can be dropped, but the proof then becomes much more complicated. Furthermore, the argument from [68], which proves the continuous analogue of Lemma 2 (along with more general criteria for a robust consensus), can be extended to discrete-time systems. However, all these proofs are rather complicated technically, so they are omitted.

3.2. Consensus in Averaging Inequalities

In this subsection, we introduce an important tool for proving the main result of this paper. Specifically, along with Eqs. (11), we consider *recurrent averaging inequalities* of the form

$$x(t+1) \le A(t)x(t), \quad t \in \mathbb{N}. \tag{18}$$

Here, the inequality between vectors is understood componentwise.

It should be noted that a recurrent inequality cannot be treated as a control algorithm for a group of agents, since its solution is not uniquely determined. However, such inequalities provide a convenient tool for analyzing multi-agent algorithms (see [30, 60, 69]). It may seem surprising that such a weak constraint on the state vector of the system of agents makes it possible to establish some properties, but it will be shown later that, as in the case of equations, consensus (12) can be proved, assuming that the graph is uniformly strongly connected. In contrast to Eqs. (11), whose solutions are bounded, solutions of the inequalities are only semibounded from above. Specifically, some components can converge to $-\infty$ as $t \to \infty$. For example, a consensus in the case of unbounded solutions means that $\bar{x}_i = -\infty \; \forall i$. In applications, however, an a priori lower bound for the solution is often known; thus, there is no consensus at infinity.

As will be seen later, in the study of solutions to inequalities, it is frequently important to know the behavior of the residual vector between the right- and left-hand sides:

$$\Delta_i(t) \triangleq \sum_j a_{ij}(t)x_j(t) - x_i(t+1) \ge 0, \tag{19}$$

in particular, the convergence of these residuals to zero:

$$\Delta_i(t) \xrightarrow[t\to\infty]{} 0. \tag{20}$$

The following key lemma guarantees that a consensus is reached and the residual converges to zero for any bounded solution of (18).

Lemma 3. *Assume that the sequence of stochastic matrices* $A(t)$ *has uniformly positive diagonal elements* $a_{ii}(t) \ge \eta > 0$ *and the variable graph* $\mathcal{G}[A(\cdot)]$ *is uniformly strongly connected. Then consensus* (12) *is reached for any solution of inequality* (18) *and, if the solution is bounded below, then* (20) *holds for any* $i \in \mathcal{V}$.

The proof of Lemma 3 is given in the Appendix. Special cases of this lemma (assuming that all nonzero matrix elements, rather than only diagonal ones, are uniformly positive) were proved in [30, 69]. Note that the uniform strong connectivity requirement can be significantly weakened. The most general result is given in [28].

3.3. Consensus Algorithms for the Intersection of Convex Sets

In this subsection, we consider several classes of algorithms proposed in the literature for solving the multi-agent SIP. It will be shown that, in each of the indicated algorithms, the convergence of the states of all agents to a consensus point lying in the intersection of the sets can fairly easily be proved by applying the technique of averaging inequalities.

Specifically, assume that a convex closed set $\Xi_i \subset \mathbb{R}^d$ associated with agent $i \in \mathcal{V}$ is the set of fixed points of a Fejér mapping P_i with respect to some norm $\|\cdot\|$ (in the most widespread case, P_i is the projector onto the set Ξ_i and the norm is Euclidean). Assume that agent i is capable of computing the value of $P_i(\xi)$ at an arbitrary given point ξ.

3.3.1. Classes of consensus algorithms. We will show that, under certain constraints imposed on the graph, a point belonging to Ξ_* can be computed using one of the following algorithms based on iterative averaging with simultaneous application of Fejér operators:

$$\xi^i(t+1) = P_i\left[\sum_{j\in\mathcal{V}} a_{ij}(t)\xi^j(t)\right], \quad i \in \mathcal{V}, \tag{21}$$

$$\xi^i(t+1) = P_i\left[\sum_{j\in\mathcal{V}} a_{ij}(t)P_j(\xi^j(t))\right], \quad i \in \mathcal{V}, \tag{22}$$

$$\xi^i(t+1) = \sum_{j\in\mathcal{V}} a_{ij}(t)P_j(\xi^j(t)), \quad i \in \mathcal{V}, \tag{23}$$

$$\xi^i(t+1) = a_{ii}(t)P_i(\xi^i(t)) + \sum_{j\neq i} a_{ij}(t)\xi^j(t), \quad i \in \mathcal{V}. \tag{24}$$

In all these algorithms, as before, $A(t) = (a_{ij}(t))$ are stochastic matrices. Note that, in the degenerate case when the agents have no constraints ($\Xi_i = \mathbb{R}^d$ and $P_i = \mathrm{Id}$) algorithms (21)–(24) turn into the standard iterative averaging procedure for reaching a consensus (see [53, 58, 70]) (with the only difference being that the agent's state is a multidimensional vector, rather than a scalar; obviously, this in no way changes the convergence conditions for the algorithm). However, in the general case, the convergence condition is much more restrictive, namely, we need uniform strong connectivity, rather than quasi-strong connectivity, as in Lemma 2. This constraint on the graph is imposed in all works known to the authors. The method for proving convergence described in this paper reduces each of the algorithms to a system of averaging inequalities; strong connectivity is required for applying Lemma 3.

Origin and features of algorithms (21)–(24). Algorithm (21) was proposed for the case of projection operators P_i in the seminal paper [25] on distributed optimization and was further developed in [45], where some assumptions (in particular, the symmetry of the matrix a_{ij}) were weakened and the robustness of the algorithm from [25] to delays was examined.[11] Later, algorithm (21) was used in [49] to find a common point of Fejér[12] mappings P_i. A substantial feature of this algorithm is that an agent first receives the states of neighbors and then computes P_i using the averaged value. Note that, if P_i is the projection onto Ξ_i, then the state of agent i does not leave the set Ξ_i for $t \geq 0$. In this case, the agent's state tends to approach the neighbors' states without violating the agent's own constraint.

Algorithm (24) was initially proposed in [71] for solving systems of linear equations, but it also works for a more general case, namely, when Ξ_i are linear hyperplanes and the projection operators P_i can be computed analytically. As in algorithm (21), the agents communicate their states to each other. However, there are two fundamental differences. First, an agent can compute P_i without waiting for neighboring measurements (in practice, this can speed up the computations). Second, the state of agent i, as a rule, does not belong to Ξ_i at any step.

Algorithm (23) is similar in structure to Cimmino's algorithm with the only difference being that each agent, first, has its own approximation to the point ξ^i from the intersection of the sets, and second, the computation of the barycenter of all projections is replaced by computing a convex combination of the projections of neighboring agents. Distributed algorithms with a similar structure were studied in [17], but agents were not introduced explicitly and the case of a variable graph was not considered. Note that, in this algorithm, the agents do not share information about their own sets with their neighbors; moreover, they do not share their states (this can be important for data protection).

Finally, algorithm (22) was proposed in [72] also as applied to algebraic equations. As in algorithm (23), the agents share their values P_i, but then agent i repeatedly computes P_i. In the case of projections,

[11] It follows from the results of [28, 30] that all algorithms (21)–(24) are actually robust to communication delays, since such delays do not violate the properties of averaging algorithms and inequalities. In this survey, we do not consider robustness issues, which are mainly studied in control theory for multi-agent systems and are of interest if the agents are limited in their ability to communicate quickly (e.g., they are linked via the Internet).

[12] Note that [49] imposes a number of constraints; in particular, the Fejér property has to hold with respect to the ℓ_p norm on $\mathbb{R}^d$. As will be seen later, this constraint can be removed.

as was discussed above, this guarantees that the state of agent i does not leave the desired set. It will be shown below that this technique also allows us to avoid some pathological equilibrium points arising in system (23) (when the intersection of the sets is empty: $\Xi_* \neq \varnothing$).

3.3.2. Constrained consensus theorem. Assume that $\Xi_* = \bigcap \Xi_i = \bigcap \mathcal{F}(P_i)$ is nonempty. We say that a constrained consensus (see [25]) is established if, for arbitrary initial states of the agents, the following limits exist and equal each other:

$$\lim_{t\to\infty} \xi^1(t) = \dots = \lim_{t\to\infty} \xi^n(t) \in \Xi_*. \tag{25}$$

An analysis of algorithms (21)–(24) relies on the convergence properties of the averaging inequalities (18) and on Lemma 2 about the robustness of a consensus. Below is the main result, which will be proved in the Appendix.

Theorem 1. *Assume that the mappings* $P_i : \mathbb{R}^d \to \mathbb{R}^d$ *are Fejér with respect to some common norm* $\|\cdot\|$ *and have at least one common fixed point, i.e.,* $\Xi_* = \bigcap_{i\in\mathcal{V}} \mathcal{F}(P_i) \neq \varnothing$. *Assume that the sequence of stochastic matrices* $A(t)$ *has uniformly positive diagonal elements* $a_{ii}(t) \geq \eta > 0$ *and the variable graph* $\mathcal{G}[A(\cdot)]$ *corresponding to* $A(t)$ *is uniformly strongly connected. Then each of the algorithms* (21)–(24) *finds a common fixed point of* $\{P_i\}$, *i.e., the agents' states* ξ^i *converge to consensus* (25).

Note that it is the first time that Theorem 1 has been published in its complete generality. The convergence of algorithms (21), (22), and (24) was first proved in [30] with the help of recurrent averaging inequalities under the additional assumption (15), which was later dropped in [28]. However, algorithm (23) was not examined.

3.3.3. Remarks on Theorem 1. If the sets Ξ_i have no common points, the convergence of the vectors $\xi^i(t)$ to a common value ξ_* as $t \to \infty$ in algorithms (21), (22), and (24) is obviously impossible. For algorithms (21) and (22), this is explained by the fact that $\xi^i(t) \in \Xi_i$ for any $i \in \mathcal{V}$, so the common limit is automatically a common point of all Ξ_i. For algorithm (24), this conclusion follows from the continuity of P_i: assuming that $\xi^i(t) \xrightarrow[t\to\infty]{} \xi_*$, we would have

$$a_{ii}(t)[P_i(\xi^i(t)) - \xi^i(t)] = \xi^i(t+1) - \sum_{j\in\mathcal{V}} a_{ij}(t)\xi^j(t) \xrightarrow[t\to\infty]{} 0,$$

which, in view of the uniform positivity of a_{ii}, means once again that $\xi_* = P_i(\xi_*)$ for all i, i.e., ξ_* is a common fixed point of the family of operators. However, algorithm (23) is an exception: a consensus in this algorithm is theoretically possible even if the intersection of the sets Ξ_i is empty. Consider, for example, a system of two agents with constraint sets $\Xi_1 = \{-1\}$, $\Xi_2 = \{1\}$ and weighting coefficients $a_{11} = a_{12} = a_{21} = a_{22} = 1/2$. Obviously, $\xi^1(t) \equiv \xi^2(t) \equiv 0$ is an equilibrium point of system (23) that belongs to none of the sets Ξ_i.

However, in the case of no consensus reached, the behavior of trajectories of the system remains, to our knowledge, an open question. Specifically, even in the case of a constant weight matrix, it is unknown whether convergence to a periodic or quasi-periodic trajectory always holds. Moreover, as can be seen from the proof given in the Appendix, even the boundedness of solutions is based on the nonemptiness of $\Xi_* = \bigcap_i \Xi_i$ and, except for special cases (e.g., P_i is a projector onto a bounded set Ξ_i), is generally not an obvious property.

It should also be noted that the convergence rate of algorithms (21)–(24) is rather difficult to estimate explicitly even in the case of a constant graph. In the case of a variable graph, even the exact convergence rate of usual consensus algorithms ($P_i = \mathrm{Id}$) has not been fully investigated. Some convergence rate estimates can be found in [73]. As was shown in [74], the estimation of the convergence rate of consensus algorithms is reduced to calculating the largest ergodicity coefficient for some compact set of stochastic matrices. There are "accelerated" modifications of algorithm (21) for solving systems of linear equations (see [27, 72]) for which exponential convergence to a consensus has been proved. These algorithms were not studied in the case of arbitrary Fejér operators, and their analysis requires a technique going beyond the scope of this survey.

In this paper, we do not consider stochastic versions of algorithms for convex set intersection, for example, the randomized version of algorithm (24) from [75] or randomized modifications of the Kaczmarz algorithm (see, e.g., the literature review in [76]). It should be noted that some of the above-considered algorithms have continuous-time analogues. For example, the following analogue of algorithm (24) was presented in [26]:

$$\dot{\xi}^i(t) = \sum_{j\in\mathcal{V}} a_{ij}(t)(\xi^j(t) - \xi^i(t)) + P_i(\xi^i(t)) - \xi^i(t). \tag{26}$$

This algorithm can be analyzed using the theory of averaging inequalities in continuous time (see [28, 29]). The corresponding results are not presented, since they are entirely similar to discrete-time convergence criteria.

4. CONCLUSIONS

We are grateful to the editors of the special issue dedicated to B.T. Polyak for the opportunity of presenting this survey, which is an expanded lecture given by the first author at the Polyak Traditional Summer School on control and optimization in 2023. The survey describes the history of the SIP in a space and covers the first author's recent results concerning the multi-agent SIP formulation and decentralized algorithms for its solution. Polyak regarded article [15] as one of the most important of his works, which contributed a number of important results on the exponential convergence of projection algorithms. In particular, he talked about this article at the seminar of the Trapeznikov Institute of Control Sciences of the Russian Academy of Sciences organized in honor of awarding him the Khachiyan Prize.[13] According to the Google Scholar database, the English version of this article has been cited more than 1000 times.

It should be noted that, although 90 years have passed since the proof of the von Neumann's first theorem, which establishes the convergence of the alternating projection method for two linear subspaces, the theory of projection algorithms is far from being complete. This is concerned with both the classical (centralized) version of the problem and its multi-agent (decentralized) formulation. As was mentioned above, numerous questions related to the convergence rate of algorithms remain open. In particular, there is no exact convergence estimate even for the Kaczmarz algorithm. When projection operators are replaced by Fejér mappings associated with sets, convergence estimates can be obtained only in special cases (see [43]). For decentralized algorithms, the estimation of the convergence rate of agent's states to a consensus value is a nontrivial problem even for an unconstrained consensus (when all sets coincide with the entire space) (see [73]).

Another open question is finding minimum (necessary and sufficient) conditions for the graph to be connected. As was shown in [28], uniform connectivity in Lemmas 2 and 3 can be significantly relaxed. However, the given condition is still only sufficient, but not necessary, and is difficult to verify for an arbitrary variable graph.

A promising research direction is to extend projection methods for manifolds (see [24]) to the multi-agent case. This task is associated with the convergence of consensus algorithms on manifolds (see [77]).

APPENDIX

Proof of Lemma 3

Assume that all the conditions of Lemma 3 are satisfied, in particular, $a_{ii}(t) \geq \eta$. First, consider the situation when the uniform connectivity condition holds with $T = 1$, i.e., all graphs $\mathcal{G}[A(t)]$ are strongly ε-connected. Let the number of agents be denoted by $n = |\mathcal{V}|$. For each $t = 0,1,\ldots$, the elements of $\mathcal{V}$ are indexed in ascending order of the components $x(t)$:

$$\mathcal{V} = \{\sigma_1(t),\ldots,\sigma_N(t)\},$$
$$y_1(t) \triangleq x_{\sigma_1(t)}(t) \leq y_2(t) \triangleq x_{\sigma_2(t)}(t) \leq \cdots \leq y_n(t) \triangleq x_{\sigma_n(t)}(t).$$

We will prove the auxiliary inequality

$$y_{k+1}(t+1) \leq (1-\varepsilon_0)y_n(t) + \varepsilon_0 y_k(t) \quad \forall k = 0,1,\ldots,n-1, \tag{27}$$

[13] Video of the seminar is available at https://www.youtube.com/watch?v=_6p5qQ15fpw.

where $\varepsilon_0 = \min(\varepsilon, \eta)$. Indeed, define $I \triangleq \{\sigma_{k+1}(t), \ldots, \sigma_n(t)\}$. Since $\mathcal{G}[A(t)]$ is a strongly ε-connected graph, there exist $i \in I$ and $m \in J \triangleq \mathcal{V} \setminus I$ such that $a_{im}(t) \geq \varepsilon \geq \varepsilon_0$. Therefore,

$$x_i(t+1) \leq a_{im}(t) \underbrace{x_m(t)}_{\leq y_k(t)} + \sum_{j \in \mathcal{V} \setminus \{m\}} a_{ij}(t) \underbrace{x_j(t)}_{\leq y_n(t)} \leq (1-\varepsilon_0) y_n(t) + \varepsilon_0 y_k(t).$$

On the other hand, using the inequality $a_{ii}(t) \geq \eta \geq \varepsilon_0$, for each $j \in J$ we have

$$x_j(t+1) \leq a_{jj}(t) \underbrace{x_j(t)}_{\leq y_k(t)} + \sum_{\ell \in \mathcal{V} \setminus \{j\}} a_{j\ell}(t) \underbrace{x_\ell(t)}_{\leq y_n(t)} \leq (1-\varepsilon_0) y_n(t) + \varepsilon_0 y_k(t).$$

Therefore, all components x with indices from the set $J \cup \{i\}$ are no greater than the right-hand side of (27). To prove (27), it remains to be noted that the cardinality of the last set is $k+1$.

Since $y_n(t) = \max x(t)$ does not increase, there exists a limit $y_* = \lim_{t \to \infty} y_n(t) \geq -\infty$. If $y_* = -\infty$, then, obviously, $x_i(t) \xrightarrow[t \to \infty]{} -\infty$. Otherwise, we can use (27) to prove that $y_k(t) \xrightarrow[t \to \infty]{} y_* > -\infty$, using backward induction on $k = n, n-1, \ldots, 1$; in other words, consensus (12) is reached.

The case of an arbitrary period $T > 1$ is reduced to the previously considered case: it suffices to note that in view of uniform positivity of diagonal elements the graphs of the matrices

$$\tilde{A}(k) = A(kT) A(kT+1) \cdots A(kT+T-1)$$

are strongly $\tilde{\varepsilon}$-connected for some $\tilde{\varepsilon} = \tilde{\varepsilon}(\varepsilon, \eta) > 0$. Therefore, convergence to consensus takes place for a subsequence $\tilde{x}(k) = x(kT)$ satisfying the recurrent inequality

$$\tilde{x}(k+1) \leq \tilde{A}(k) \tilde{x}(k).$$

Since the inequalities

$$\tilde{x}(k+1) = x(kT+T) \leq A(kT+T-1) \cdots A(kT+\ell) x(kT+\ell),$$
$$x(kT+\ell) = A(kT+\ell-1) \cdots A(kT) \tilde{x}(k),$$

hold for any $\ell = 1, 2, \ldots, T-1$, it follows from the convergence $\tilde{x}(k) \xrightarrow[k \to \infty]{} \tilde{c}\mathbf{1}_n$ (where $\tilde{c} \in \mathbb{R}$ and $\mathbf{1}_n$ denotes the all-one column vector of dimension n) and the stochasticity of the matrices $A(t)$ that $x(kT+\ell) \xrightarrow[k \to \infty]{} \tilde{c}\mathbf{1}_n$ for any $\ell = 1, 2, \ldots, T-1$; therefore, a consensus is reached among the agents.

Finally, property (20) for bounded solutions follows from the fact that the matrices $A(t)$ are stochastic (and, in particular, uniformly bounded): if $x(t) \xrightarrow[t \to \infty]{} c\mathbf{1}_n$ (for some $c \in \mathbb{R}$), then $A(t)\mathbf{1}_n = \mathbf{1}_n$; therefore,

$$A(t)x(t) = A(t)(x(t) - c\mathbf{1}_n) + c\mathbf{1}_n \xrightarrow[t \to \infty]{} c\mathbf{1}_n.$$

Proofs of Lemma 1 *and Theorem* 1

Lemma on a Fejér mapping. The proofs of the theorem and the lemma rely on the following technical result.

Lemma 4. *Let P be a Fejér operator in some norm $\|\cdot\|$ and ξ^0 be its fixed point. Define $d(\xi) \triangleq \|\xi - \xi^0\| - \|P\xi - \xi^0\| \geq 0$ and consider a bounded sequence of vectors $\xi(t)$ such that $d(\xi(t)) \xrightarrow[t \to \infty]{} 0$. Then*

$$\|P\xi(t) - \xi(t)\| \xrightarrow[t \to \infty]{} 0.$$

Proof. Assume the opposite: $\|P\xi(t_r) - \xi(t_r)\| \geq \varepsilon$ for any $r \in \mathbb{N}$, any $\varepsilon > 0$, and any sequence $t_r \to \infty$. By passing to a smaller subsequence, it is possible to assume without loss of generality that the vectors $\xi(t_r)$ converge to a limit $\xi_* \in \mathbb{R}^d$. Since P is a continuous mapping, we have $\|P\xi_* - \xi_*\| \geq \varepsilon$ for $d(\xi_*) = 0$. We have obtained a contradiction to (7), since $\xi_* \notin \mathcal{F}(P)$, while $\|P\xi_* - \xi^0\| = \|\xi_* - \xi^0\|$. Lemma 4 is proved.

Proof of Lemma 1. Obviously, the sequence $\xi(t)$ from Lemma 1 is bounded, since the distances from $\xi(t)$ to any point from M do not increase. For an arbitrary element $\xi^0 \in M$, the nonincreasing sequence

$$a(t) \triangleq \left\|\xi(t) - \xi^0\right\| \geq 0$$

has a finite limit as $t \to \infty$. Specifically, since $P\xi(t) = \xi(t+1)$, we have

$$a(t) - a(t+1) = \left\|\xi(t) - \xi^0\right\| - \left\|P\xi(t) - \xi^0\right\| = d(\xi(t)) \xrightarrow[t\to\infty]{} 0.$$

By Lemma 4, we prove that $\|P\xi(t) - \xi(t)\| \xrightarrow[t\to\infty]{} 0$.

Note that, in view of the boundedness, there exists a subsequence $\xi(t_k)$ converging to some element $\xi_* \in \mathbb{R}^d$. Obviously,

$$\left\|P(\xi_*) - \xi_*\right\| = \lim_{k\to\infty} \|P(\xi(t_k)) - \xi(t_k)\| = 0,$$

i.e., $\xi_* \in M$. Since ξ^0 in the above argument is arbitrary, we can choose $\xi^0 = \xi^*$. Since for $t \geq t_k$,

$$0 \leq \left\|\xi(t) - \xi_*\right\| \leq \left\|\xi(t_k) - \xi_*\right\|,$$

where the right-hand side can be made arbitrarily small for large k, we have $\left\|\xi(t) - \xi_*\right\| \xrightarrow[t\to\infty]{} 0$. Lemma 1 is proved.

Proof of Theorem 1. First, we introduce some auxiliary notation. Given an arbitrary point $\xi^0 \in \Xi_*$, we define $\delta_i(\xi) \triangleq \left\|\xi - \xi^0\right\| - \left\|P_i\xi - \xi^0\right\| \geq 0$. Let $\zeta^i(t) = \sum_{j\in\mathcal{V}} a_{ij}(t)\xi^j(t)$. The central idea of the proof is to examine the properties of the vectors $x(t) = (x_i(t))_{i\in\mathcal{V}}$ whose components $x_i(t) \triangleq \left\|\xi^i(t) - \xi^0\right\|$ are the distances from the agents' states $\xi^i(t)$ to the chosen fixed point.

Step 1. First, we show that, for each of the algorithms indicated in the theorem, the vectors $x(t)$ satisfy the recurrent averaging inequality (18).

In the case of algorithm (21), we have

$$x_i(t+1) = \left\|P_i(\zeta^i(t)) - \xi^0\right\| \overset{(8)}{\leq} \left\|\zeta^i(t) - \xi^0\right\| = \left\|\sum_j a_{ij}(t)(\xi^j(t) - \xi^0)\right\| \leq \sum_j a_{ij}(t)\left\|\xi^j(t) - \xi^0\right\| = \sum_j a_{ij}(t)x_j(t). \quad (28)$$

The case of (22) is considered in a similar manner. Introducing $\overline{\zeta}^i(t) \triangleq \sum_{j\in\mathcal{V}} a_{ij}(t)P_j(\xi^j(t))$, we have

$$\begin{aligned} x_i(t+1) &= \left\|P_i(\overline{\zeta}^i(t)) - \xi^0\right\| \overset{(8)}{\leq} \left\|\overline{\zeta}^i(t) - \xi^0\right\| = \left\|\sum_j a_{ij}(t)(P_j(\xi^j(t)) - \xi^0)\right\| \leq \sum_j a_{ij}(t)\left\|P_j(\xi^j(t)) - \xi^0\right\| \\ &\overset{(8)}{\leq} \sum_j a_{ij}(t)\left\|\xi^j(t) - \xi^0\right\| = \sum_j a_{ij}(t)x_j(t). \end{aligned} \quad (29)$$

In the case of algorithm (23), it is easy to see that

$$x_i(t+1) = \left\|\sum_j a_{ij}(t)(P_j(\xi^j(t)) - \xi^0)\right\| \leq \sum_j a_{ij}(t)\left\|P_j(\xi^j(t)) - \xi^0\right\| \overset{(8)}{\leq} \sum_j a_{ij}(t)\left\|\xi^j(t) - \xi^0\right\| = \sum_j a_{ij}(t)x_j(t). \quad (30)$$

For algorithm (24), we have

$$\begin{aligned} x_i(t+1) &= \left\|a_{ii}(t)(P_i(\xi^i(t)) - \xi^0)) + \sum_{j\neq i} a_{ij}(t)(\xi^j(t) - \xi^0)\right\| \leq a_{ii}(t)\left\|P_i(\xi^i(t)) - \xi^0)\right\| + \sum_{j\neq i} a_{ij}(t)\left\|\xi^j(t) - \xi^0\right\| \\ &\overset{(8)}{\leq} a_{ii}(t)\left\|\xi^i(t) - \xi^0\right\| + \sum_{j\neq i} a_{ij}(t)x_j(t) = \sum_{j\in\mathcal{V}} a_{ij}(t)x_j(t). \end{aligned} \quad (31)$$

Note that $x_i(t)$ are nonnegative by definition. By Theorem 3, the averaging inequality (18) establishes a consensus, i.e., $x_i(t) \xrightarrow[t\to\infty]{} c \geq 0\ \forall i$, where c depends on the particular solution. Theorem 3 also implies that

$$\Delta_i(t) \xrightarrow[t\to\infty]{} 0 \quad \forall i \in \mathcal{V}, \tag{32}$$

where the residuals $\Delta_i(t)$ are given by formula (19).

Step 2. Investigating the structure of the residuals Δ_i, we show that

$$\left\|\xi^i(t+1) - \zeta^i(t)\right\| \xrightarrow[t\to\infty]{} 0 \quad \forall i, \tag{33}$$

$$\left\|P_i(\xi^i(t)) - \xi^i(t)\right\| \xrightarrow[t\to\infty]{} 0 \quad \forall i. \tag{34}$$

The vectors $\xi^i(t)$ (and, hence, $\zeta^i(t)$) are bounded, since the distances $x_i(t)$ from them to ξ^0 are bounded above in view of (18). For algorithm (21), it follows from inequality (28) that

$$\Delta_i(t) \geq \left\|\zeta^i(t) - \xi^0\right\| - \left\|P_i(\zeta^i(t)) - \xi^0\right\| = \delta_i(\zeta^i(t)). \tag{35}$$

Applying Lemma 4 to $P = P_i$ from (32), we obtain $\left\|P_i(\zeta^i(t)) - \zeta^i(t)\right\| \xrightarrow[t\to\infty]{} 0$, which is equivalent to (33), since $\xi^i(t+1) = P_i(\zeta^i(t))$. To derive (34), we note that

$$0 \leq \left\|P_i(\xi^i(t+1)) - \xi^i(t+1)\right\| \leq \left\|P_i(\zeta^i(t)) - \zeta^i(t)\right\| + \left\|\zeta^i(t) - \xi^i(t+1)\right\| \xrightarrow[t\to\infty]{} 0.$$

In the case of algorithm (24), we note that (31) yields

$$\Delta_i(t) \geq a_{ii}(t)\left(\left\|\xi^i(t) - \xi^0\right\| - \left\|P_i(\xi^i(t)) - \xi^0\right\|\right) \geq \eta\delta_i(\xi^i(t)), \tag{36}$$

where $\eta > 0$ is the constant from the assumption of the theorem. Applying Lemma 4, we can prove (34), which implies (33), since

$$\xi^i(t+1) = \zeta^i(t) + a_{ii}(t)(P_i(\xi^i(t)) - \xi^i(t)).$$

In the case of algorithm (22), the proof combines two estimates mentioned above. First, (29) leads to the inequality

$$\Delta_i(t) \geq \left\|\overline{\zeta}^i(t) - \xi^0\right\| - \left\|P_i(\overline{\zeta}^i(t)) - \xi^0\right\| = \delta_i(\overline{\zeta}^i(t)), \tag{37}$$

which is similar to (35) and, since $P_i(\overline{\zeta}^i(t)) = \xi^i(t+1)$, it implies that

$$\left\|\xi^i(t+1) - \overline{\zeta}^i(t)\right\| \xrightarrow[t\to\infty]{} 0. \tag{38}$$

Second, (29) also implies (36), which, in turn, implies (34). Thus,

$$\left\|\overline{\zeta}^i(t) - \zeta^i(t)\right\| \xrightarrow[t\to\infty]{} 0. \tag{39}$$

This proves (33), since $\left\|\xi^i(t+1) - \zeta^i(t)\right\| \leq \left\|\xi^i(t+1) - \overline{\zeta}^i(t)\right\| + \left\|\overline{\zeta}^i(t) - \zeta^i(t)\right\|$.

The proof for algorithm (23) is similar to the proof for algorithm (22). In this case, (38) holds by the definition of $\overline{\zeta}^i$ (the left-hand side is zero), and inequality (36) easily follows from (30), since

$$a_{ij}(t)\left(\left\|\xi^j(t) - \xi^0\right\| - \left\|P_j(\xi^j(t)) - \xi^0\right\|\right) \geq 0 \quad \forall j \neq i.$$

The fulfillment of Eqs. (34) and (33) can be checked by analogy with algorithm (22).

Thus, we have proved that, for all algorithms mentioned in the theorem, the solutions have the property

$$f^i(t) \triangleq \xi^i(t+1) - \zeta^i(t) = \xi^i(t+1) - \sum_{j\in\mathcal{V}} a_{ij}(t)\xi^j(t) \xrightarrow[t\to\infty]{} 0.$$

By the definition of $f^i(t)$, we note that the states of the agents ξ^i can be treated as trajectories of the perturbed *averaging system*

$$\xi^i(t+1) = \sum_j a_{ij}(t)\xi^j(t) + f^i(t),$$

where $f^i(t)$ converges to zero. By Lemma 2, the presence of perturbations[14] vanishing at infinity does not violate the synchronization between the agents: $\left\|\xi^i(t) - \xi^j(t)\right\| \xrightarrow[t\to\infty]{} 0$.

Step 3. Consider the state of the agent with index $i \in \mathcal{V}$. Since the vectors $\xi^i(t)$ are bounded, there exists a sequence $t_r \to \infty$ such that $\xi^i(t_r) \xrightarrow[r\to\infty]{} \xi_* \in \mathbb{R}^d$. In view of the synchronization, we have $\xi^j(t_r) \xrightarrow[r\to\infty]{} \xi_*$ for each $j \in \mathcal{V}$. Property (34) implies that $P_i(\xi_*) = \xi_*\ \forall i$, whence $\xi_* \in \Xi_*$. It remains to show that $\xi^i(t) \xrightarrow[t\to\infty]{} \xi_*$. Note that at Step 1 we did not specify the choice of ξ^0, which can be an arbitrary point of Ξ_*. We proved that, for any such point, the distance $x_i(t) = \left\|\xi^i(t) - \xi^0\right\|$ converges to some value c depending on ξ^0 and the initial conditions. Specifically, substituting $\xi^0 = \xi_*$, we find that there are limits

$$x_i(t) = \lim_{t\to\infty}\left\|\xi^i(t) - \xi_*\right\| = c_* \quad \forall i.$$

Recalling that $x_i(t_r) \xrightarrow[r\to\infty]{} 0$, we have $c_* = 0$, which proves that consensus (25) is reached; moreover, the common limit of the agents' states in (25) is equal to $\xi_* \in \Xi_*$.

REFERENCES

1. R. Gordon, R. Bender, and G. T. Herman, “Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography,” J. Theor. Biol. **29** (3), 471–481 (1970).
2. A. Kh. Gelig and A. S. Matveev, *Introduction to the Mathematical Theory of Learning Recognition Systems and Neural Networks* (Sankt-Peterb. Gos. Univ., St. Petersburg, 2014) [in Russian].
3. V. A. Yakubovich, “Some general theoretical principles for constructing trainable identification systems,” *Computer Science and Programming Issues* (Leningr. Gos. Univ., Leningrad, 1965), pp. 3–71 [in Russian].
4. V. F. Demyanov, “Mathematical diagnostics via nonsmooth analysis,” Optim. Methods Software **20** (2–3), 197–218 (2005).
5. K. I. Anan’ev et al., “Optimization methods in diagnosis problems,” Vestn. Sankt-Peterb. Gos. Univ. Prikl. Mat. Inf. Prots. Upr. **10** (3), 3–12 (2011).
6. V. N. Malozemov and A. V. Plotkin, “Strict polynomial separation of two sets,” Vestn. Sankt-Peterb. Gos. Univ. Mat. Mekh. Astron. **6** (2), 232–240 (2019).
7. P. L. Combettes, “The foundations of set theoretic estimation,” Proc. IEEE **81** (2), 182–208 (1993).
8. I. P. Petrov and A. V. Timofeev, “Finitely-convergent recurrent algorithms for solving target inequalities with constraints,” USSR Comput. Math. Math. Phys. **15** (6), 214–221 (1975).
9. V. N. Fomin, A. L. Fradkov, and V. A. Yakubovich, *Adaptive Control of Dynamic Objects* (Nauka, Moscow, 1981) [in Russian].
10. S. Kaczmarz, “Angenäherte Auflösung von Systemen linearer Gleichungen,” Bull. Int. Acad. Pol. Sci. Lett. **35**, 355–357 (1937).

[14] Formally, Lemma 2 was formulated for the case of agents with scalar states $x_i \in \mathbb{R}$, but it obviously remains valid for the multidimensional case (it suffices to consider the projections of the agents’ states onto each coordinate axis).

11. G. Cimmino, “Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari,” Ric. Sci. **2** (9), 326–333 (1938).

12. J. von Neumann, *Functional Operators*, Vol. 2: *The Geometry of Orthogonal Spaces* (Princeton Univ. Press, Princeton, NJ, 1950).

13. I. I. Eremin, “Relaxation method for solving systems of inequalities with convex functions on left-hand sides,” Dokl. Akad. Nauk SSSR **160** (5), 994–996 (1965).

14. L. M. Bregman, “Finding the common point of convex sets by the method of successive projection,” Dokl. Akad. Nauk SSSR **162** (3), 487–490 (1965).

15. L. G. Gurin, B. T. Polyak, E. V. Raik, “The method of projections for finding the common point of convex sets,” USSR Comput. Math. Math. Phys. **7** (6), 1–24 (1967).

16. I. I. Eremin and V. D. Mazurov, “Iterative method for solving convex programming problems,” Dokl. Akad. Nauk SSSR **170** (1), 57–60 (1966).

17. E. A. Berdnikova, I. I. Eremin, and L. D. Popov, “Distributed Fejér processes for systems of linear inequalities and problems of linear programming,” Autom. Remote Control **65** (2), 168–183 (2004).

18. I. I. Eremin and L. D. Popov, “Closed Fejér cycles for incompatible systems of convex inequalities,” Russ. Math. **52** (1), 8–16 (2008).

19. I. I. Eremin and L. D. Popov, “Fejér processes in theory and practice: Recent results,” Russ. Math. **53** (1), 36–55 (2009).

20. L. M. Bregman, “The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming,” USSR Comput. Math. Math. Phys. **7** (3), 200–217 (1967).

21. V. V. Vasin and I. I. Eremin, *Operators and Iterative Processes of Fejér Type: Theory and Applications* (Regulyarnaya i khaoticheskaya dinamika, Izhevsk, 2005; Walter de Gruyter, Berlin, 2009).

22. R. Escalante and M. Raydan, *Alternating Projection Methods* (SIAM, Philadelphia, 2011).

23. H. H. Bauschke and J. M. Borwein, “On projection algorithms for solving convex feasibility problems,” SIAM Rev. **38** (3), 367–426 (1996).

24. A. S. Lewis and J. Malick, “Alternating projections on manifolds,” Math. Oper. Res. **33** (1), 216–234 (2008).

25. A. Nedic, A. Ozdaglar, and P. Parrilo, “Constrained consensus and optimization in multi-agent networks,” IEEE Trans. Autom. Control **55** (4), 922–938 (2010).

26. G. Shi, K. Johansson, and Y. Hong, “Reaching an optimal consensus: Dynamical systems that compute intersections of convex sets,” IEEE Trans. Autom. Control **58** (3), 610–622 (2013).

27. P. Wang et al., “Solving a system of linear equations: From centralized to distributed algorithms,” Ann. Rev. Control. **47**, 306–322 (2019).

28. A. V. Proskurnikov, “Averaging algorithms and inequalities in problems of multi-agent control and simulation,” Doctoral Dissertation in Physics and Mathematics (St. Petersburg, 2021).

29. A. Proskurnikov and M. Cao, “Differential inequalities in multi-agent coordination and opinion dynamics modeling,” Automatica **85**, 202–210 (2017).

30. A. Proskurnikov, G. Calafiore, and M. Cao, “Recurrent averaging inequalities in multi-agent control and social dynamics modeling,” Ann. Rev. Control **49**, 95–112 (2020).

31. A. V. Balakrishnan, *Applied Functional Analysis* (Springer-Verlag, New York, 1976).

32. B. Peterson and M. Olinick, “Leontief models, Markov chains, substochastic matrices, and positive solutions of matrix equations,” Math. Model. **3** (3), 221–239 (1982).

33. B. Polyak and A. Tremba, “Regularization-based solution of the PageRank problem for large matrices,” Autom. Remote Control **73** (11), 1877–1894 (2012).

34. A. V. Gasnikov and D. Yu. Dmitriev, “On efficient randomized algorithms for finding the PageRank vector,” Comput. Math. Math. Phys. **55** (3), 349–365 (2015).

35. R. Sznajder, “Kaczmarz algorithm revisited,” Tech. Trans. Fundam. Sci., No. 2, 248–254 (2015).

36. F. Deutsch, “The angle between subspaces of a Hilbert space,” *Approximation Theory, Wavelets and Applications*, Ed. by S. P. Singh (Springer, Dordrecht, 1995), pp. 107–130.

37. M. Arioli et al., “A block projection method for sparse matrices,” SIAM J. Sci. Stat. Comput. **13**, 326–333 (1938).

38. S. Agmon, “The relaxation method for linear inequalities,” Can. J. Math. **6**, 382–392 (1954).

39. T. S. Motzkin and I. J. Schoenberg, “The relaxation method for linear inequalities,” Can. J. Math. **6**, 393–404 (1954).

40. I. I. Eremin, “Generalization of the relaxation method of Motzkin and Agmon,” Usp. Mat. Nauk **20** (2), 183–187 (1965).

41. I. I. Eremin, "On the speed of convergence in the method of Fejér approximation," Math. Notes **4** (1), 522–527 (1968).

42. I. I. Eremin, "Fejér transformations and a problem of convex programming," Sib. Math. J. **10** (5), 762–772 (1969).

43. I. I. Eremin, "The application of the method of Fejér approximations to the solution of problems of convex programming with non-smooth constraints," USSR Comput. Math. Math. Phys. **9** (5), 225–235 (1969).

44. L. M. Bregman, "Relaxation method for finding a common point of convex sets and its application in optimization problems," Dokl. Akad. Nauk SSSR **171** (5), 1019–1022 (1966).

45. P. Lin and W. Ren, "Constrained consensus in unbalanced networks with communication delays," IEEE Trans. Autom. Control **59** (3), 775–781 (2014).

46. G. Notarstefano, I. Notarnicola, and A. Camisa, "Distributed optimization for smart cyber-physical networks," Found. Trends Syst. Control **7** (3), 253–383 (2019).

47. A. V. Gasnikov, *Modern Numerical Optimization Methods: Universal Gradient Descent Method* (Mosk. Fiz.-Tekh. Inst., Moscow, 2018) [in Russian].

48. L. Fejér, "Über die Lage der Nullstellen von Polynomen, die aus Minimumforderungen gewisser Art entspringen," Math. Ann. **85**, 41–48 (1922).

49. D. Fullmer and A. S. Morse, "A distributed algorithm for computing a common fixed point of a finite family of paracontractions," IEEE Trans. Autom. Control **63** (9), 2833–2843 (2018).

50. V. V. Vasin and I. I. Eremin, *Operators and Iterative Processes of Fejér Type: Theory and Applications* (De Gruyter, Berlin, 2009).

51. G. Rzevski and P. Skobelev, *Managing Complexity* (WIT, Southampton, UK, 2014).

52. M. Wooldridge, *An Introduction To Multiagent Systems* (Wiley, Chichester, England, 2002).

53. A. V. Proskurnikov and A. L. Fradkov, "Problems and methods of network control," Autom. Remote Control **77** (10), 1711–1740 (2016).

54. O. N. Granichin, "What is the structure of complex information-controlling systems?," Stokhast. Optim. Inf. **12** (1), 3–19 (2016).

55. A. V. Proskurnikov and O. N. Granichin, "Evolution of clusters in large-scale dynamics networks," Cybern. Phys. **7** (3), 102–129 (2018).

56. A. L. Fradkov et al., *Issues of Network Control* (IKI, Izhevsk, 2015) [in Russian].

57. V. Marik, V. Gorodetsky, and P. Skobelev, "Multi-agent technology for industrial applications: Barriers and trends," *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics* (2020), pp. 1980–1987.

58. R. P. Agaev and P. Yu. Chebotarev, "Convergence and stability in problems of matching characteristics," Upr. Bol'shimi Sist. **30** (1), 470–505 (2010).

59. A. Proskurnikov and R. Tempo, "A tutorial on modeling and analysis of dynamic social networks. Part I," Ann. Rev. Control **43**, 65–79 (2017).

60. A. Proskurnikov and R. Tempo, "A tutorial on modeling and analysis of dynamic social networks: Part II," Ann. Rev. Control **45**, 166–190 (2018).

61. E. Seneta, *Non-Negative Matrices and Markov Chains* (Springer-Verlag, New York, 1981).

62. A. Leizarowitz, "On infinite products of stochastic matrices," Linear Algebra Appl. **168**, 189–219 (1992).

63. S. Bolouki and R. P. Malhame, "Consensus algorithms and the decomposition-separation theorem," IEEE Trans. Autom. Control **61** (9), 2357–2369 (2016).

64. B. Touri and A. Nedic, "On backward product of stochastic matrices," Automatica **48** (8), 1477–1488 (2012).

65. P. Chebotarev and R. Agaev, "Forest matrices around the Laplacian matrix," Linear Algebra Appl. **356**, 253–274 (2002).

66. W. Ren and R. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control: Theory and Applications* (Springer-Verlag, London, 2008).

67. W. Ren and Y. Cao, *Distributed Coordination of Multi-Agent Networks* (Springer, Cham, 2011).

68. G. Shi and K. Johansson, "Robust consensus for continuous-time multi-agent dynamics," SIAM J. Control Optim. **51** (5), 3673–3691 (2013).

69. A. Proskurnikov and M. Cao, "Modulus consensus in discrete-time signed networks and properties of special recurrent inequalities," *Proceedings of IEEE Conference on Decision and Control* (2017), pp. 2003–2008.

70. A. Proskurnikov and A. Matveev, "Popov-type criterion for consensus in nonlinearly coupled networks," IEEE Trans. Cybern. **45** (8), 1537–1548 (2015).

71. K. You, S. Song, and R. Tempo, “A networked parallel algorithm for solving linear algebraic equations,” *Proceedings of IEEE Conference on Decision and Control* (2016), pp. 1727–1732.
72. S. Mou, J. Liu, and A. Morse, “A distributed algorithm for solving a linear algebraic equation,” IEEE Trans. Autom. Control **60** (11), 2863–2878 (2015).
73. A. Olshevsky and J. Tsitsiklis, “Convergence speed in distributed consensus and averaging,” SIAM Rev. **53** (4), 747–772 (2011).
74. A. V. Proskurnikov and G. C. Calafiore, “Delay robustness of consensus algorithms: Continuous-time theory,” IEEE Trans. Autom. Control **68** (9), 5301–5316 (2023).
75. G. Shi and K. Johansson, “Randomized optimal consensus of multi-agent systems,” Automatica **48** (12), 3018–3030 (2012).
76. S. Steinerberger, “Randomized Kaczmarz converges along small singular vectors,” SIAM J. Matrix Anal. Appl. **42** (2), 608–615 (2021).
77. A. Sarlette and R. Sepulchre, “Consensus optimization on manifolds,” SIAM J. Control Optim. **48** (1), 56–76 (2009).