Fast primal-dual projected linear iterations for distributed consensus in constrained convex optimization

Ion Necoara, Ioan Dumitrache and Johan A.K. Suykens

***Abstract*— In this paper we study the constrained consensus problem, i.e. the problem of reaching a common point from the estimates generated by multiple agents that are constrained to lie in different constraint sets. First, we provide a novel formulation of this problem as a convex optimization problem but with coupling constraints. Then, we propose a primal-dual decomposition method for solving this type of coupled convex optimization problems in a distributed fashion given restrictions on the communication topology. The proposed algorithm is based on consensus principles (as an efficient strategy for information fusion in networks) in combination with local subgradient updates for the primal-dual variables. We show, for the first time, that the nonnegative weights corresponding to the consensus process can be interpreted as dual variables and thus they can be updated using arguments from duality theory. Therefore, in our algorithm the weights are updated following some precise rules, while in most of the existing distributed algorithms based on consensus principles the weights have to be tuned. Preliminary simulation results show that our algorithm works, an average, ten times faster than some existing methods.**

I. Introduction

There has been much interest in designing distributed or parallel algorithms for finding a feasible point in the intersection of some closed convex sets. Such problems appear in a wide range of applications such as rendezvous problems, motion planning and alignment problems, where the position of each agent (robot, car, etc) is limited to a certain region or range, networked control, estimation in sensor networks, and distributed multi-agent optimization problems. One distinguishes mainly two classes of optimization algorithms for this problem:

(I) "Centralized" optimization algorithms that address the feasibility problems from a central viewpoint: In this class the structure of the network has a particular form (in general star-shaped with a supervisor), which is exploited, as it represents considerable sparsity in the optimization problem. The main parts of the algorithms can be parallelized, but the parallelization in these "centralized" algorithms will not be restricted by e.g. limited communication between node values and is just for the sake of exploiting sparsity. These algorithms are mainly based on alternating projections [1] or primal-dual projections [2].

(II) Distributed optimization algorithms: In contrast to the "centralized" algorithms, for distributed algorithms we need to associate a graph and then the algorithm must satisfy an extra constraint: the computation shall be performed on all nodes in parallel, and the communication between nodes is restricted to the edges of the graph. The communication constraints might come from judicial or game-theoretic restrictions, or from hardware constraints.

An active area of research now is also the consensus problem for networks of dynamic agents, in which several autonomous agents try to reach a common objective. This is partly due to broad applications of multi-agent systems in many areas including distributed optimization (in particular convex feasibility problem), cooperative control, distributed sensor networks, etc (see e.g. [3]). A consensus algorithm is a distributed protocol where at each time step every agent updates its variable by taking a weighted average of its own value with values received from some of the other agents.

While "centralized" optimization is a widely diffused and well-understood advanced optimization technique, see e.g. the overview book [1], distributed optimization algorithms, that are mainly based on combining consensus negotiations (as an efficient method for information fusion) with subgradient methods, are less mature but subject to intensive ongoing research (see e.g. [4], [5], [6]). Despite much work on the consensus problem, most existing algorithms on this type of distributed optimization problems uses matrix theory, spectral graph theory and subgradient methods to prove their convergence and solve these problems in a distributed manner. The main drawback of the existing distributed methods is that they do not provide yet a mathematically sound way of choosing the weights from the consensus protocol, which has a very strong influence on the convergence rate of these methods. Such algorithms are very sensitive to the choice of these weights that must be tuned, since are considered as parameters in these methods. In addition to the papers already mentioned above, our paper is also related to recent work dealing with network design problems for achieving faster consensus algorithms [7], [8].

The purpose of this paper is to propose a new distributed method for the convex feasibility problem that combines, as in the methods already cited above, consensus protocols with subgradient iterations, but choosing the weights adequately. Based on a new interpretation of the weights in the consensus protocol as dual variables (or Lagrange multipliers), we derive a primal-dual distributed method that still preserves the structure of the problem but it provides, using theoretical arguments, a novel way for updating the weights. Up to our knowledge this is the first time when the weights are interpreted as dual variables and thus updated

Ion Necoara and Ioan Dumitrache are with Automation and Systems Engineering Department, University Politehnica Bucharest, Romania. Johan Suykens is with Electrical Engineering Department (ESAT-SCD), Katholieke Universiteit Leuven, Belgium. {i.necoara,ioandumitrache}@ics.pub.ro, johan.suykens@esat.kuleuven.be.

using optimization arguments. We hope this new method of updating the weights may open a new window of opportunity for algorithmic research in this area of distributed methods based on consensus arguments. The new algorithm is suitable for decomposition since it is highly parallelizable and thus it can be effectively implemented on parallel processors.

The paper is organized as follows. In Section II we introduce the convex feasibility problem that we want to solve, we provide new reformulations of this problem as separable optimization problems, so that we can use optimization methods to solve it, and then we show that the gradient projected algorithm can be viewed as a distributed consensus protocol. In Section III we introduce a primal-dual algorithm where the nonnegative weights corresponding to the consensus process can be interpreted as dual variables and thus they can be updated using arguments from duality theory. Convergence of this primal-dual distributed algorithm follows from standard arguments from optimization theory. Finally, extensions to general separable convex optimization problems are given.

II. Consensus algorithms for the convex feasibility problem with fixed weights

A. Problem definition

In this paper we focus on solving the following convex feasibility problem: find in a distributed way a common point

$$\text{(FP):} \qquad x^* \in \cap_{i=1}^N X_i.$$

We assume that $X_i \subseteq \mathbb{R}^n$ are given compact convex sets and simple enough (simple in the sense that the projection can be done easily on these sets). Moreover, we assume that the intersection of these convex sets is nonempty.

Note that this feasibility problem (FP) can be posed equivalently as a separable convex problem (see e.g. [9], [10] for more details):

$$f^* = \min_{x_i = x_j,\ i \neq j} \sum_{i=1}^N f_i(x_i), \tag{1}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is the indicator function of the convex set X_i, i.e.

$$f_i(x_i) = \begin{cases} 0 & \text{if } x_i \in X_i \\ \infty & \text{otherwise.} \end{cases}$$

It is clear that $f^* = 0$ if and only if $\cap_{i=1}^N X_i \neq \emptyset$.

One approach for solving distributively this feasibility problem is as follows (see [1]):

$$\min_{x_i \in X_i, x \in \mathbb{R}^n} \sum_{i=1}^N a_i \|x_i - x\|^2,$$

where $a_i > 0$ such that $\sum_i a_i = 1$ are given. Here, $\|\cdot\|$ denotes the Euclidian norm. If we apply the nonlinear Gauss-Seidel algorithm, we obtain the following parallel iteration:

$$x^{k+1} = \sum_{j=1}^N a_j x_j^k, \quad x_i^{k+1} = [x^{k+1}]_{X_i}, \ \ \forall i$$

or equivalently

$$x_i^{k+1} = [\sum_{j=1}^N a_j x_j^k]_{X_i}, \ \ \forall i \tag{2}$$

where $[\cdot]_{X_i}$ denotes the projection on the set X_i. However this algorithm is not fully distributed since each "agent" needs to communicate with all the others in a ring-shaped topology network.

Theorem 1: Any limit point of the sequence generated by the iteration (2) belongs to the set $\cap_{i=1}^N X_i$, for all i.

Proof: The proof follows from Proposition 2.5 in [1], since each function $x_i \mapsto a_i\|x_i - x\|^2$ and $x \mapsto \sum_{i=1}^N a_i\|x_i - x\|^2$ is differentiable and strictly convex in x_i and x respectively, so that $x_i^k \to x^* \in \cap_{i=1}^N X_i$ as $k \to \infty$, for all i. ■

In what follows we will develop fully distributed algorithms with given restrictions on the communication topology (i.e. the structure of the graph is fixed) that uses only vector operations and projections (thus very cheap computations) and we will also analyze convergence of such algorithms.

The goal is to solve (1) in a fully distribution fashion, i.e. the agents use only local information from their neighbors, given a fixed topology on the communication network. For this purpose we introduce an information exchange model. We consider a fixed network with the associated undirected graph $G = (V, E)$, where the node (agent) set is $V = \{1, \cdots, N\}$ and $E \subseteq V \times V$ denotes the set of edges. We also define the adjacency matrix of the graph as $A = [a_{ij}] \in \mathbb{R}^{N \times N}$ such that $a_{ij} > 0$ if and only if $(i, j) \in E$, and $a_{ij} = 0$ otherwise. We denote the neighbors of i with $\mathcal{N}(i) = \{j \in V : a_{ij} > 0\}$.

The following assumptions will be valid in this section:

Assumption 1:
1) $\cap_{i=1}^N X_i \neq \emptyset$
2) the graph G is connected and symmetric (i.e. the weights satisfy $a_{ij} = a_{ji}$ for all i, j)
3) $a_{ii} = 0$ for all i.

In this section we assume the adjacency matrix A to be fixed and we consider varying later in Section III.

Note that we can rewrite in a new form the feasibility problem (FP) "equivalently"[1] as the following convex optimization problem:

$$\min_{x_i \in X_i} \sum_{i;j>i} a_{ij}\|x_i - x_j\|^2. \tag{3}$$

This novel reformulation of the feasibility problem (FP) as an optimization problem will allow us in the sequel to solve (FP) using distributed gradient based algorithms.

The following lemma, whose proof is straightforward, shows the "equivalence" between the feasibility problem (FP) and the optimization problem (3):

Proposition 1: If the graph G is connected and symmetric the solution of the optimization problem (3) leads to a solution for the feasibility problem (FP).

[1]The equivalence here is understood in the sense that any solution of the optimization problem (3) leads to a solution of the feasibility problem (FP).

If $\cap_i X_i \neq \emptyset$, it is also clear that the set of optimal solutions for (3) is $X^* = \{e_N \otimes x^* : x^* \in \cap_i X_i\}$, where $\otimes$ denotes the Kronecker product and $e_N \in \mathbb{R}^N$ denotes the vector with all entries equal to 1. The goal is to solve the convex optimization problem (3) distributively, where each agent i uses only local information from its neighbors (given by the connected graph G) and we also perform only vector operations (no operations with matrices). Note that (3) can be written compactly as

$$\min_{x_i \in X_i} x^T (L \otimes I_n) x, \tag{4}$$

where $x = [x_1^T \cdots x_N^T]^T$ and the matrix $L \in \mathbb{R}^{N \times N}$, usually called the *Laplacian* matrix, is defined as follows $L = [\ell_{ij}]$:

$$\ell_{ij} = \begin{cases} \sum_l a_{il} & \text{if } i = j \\ -a_{ij} & \text{if } i \neq j. \end{cases}$$

It follows that $Le_N = 0$, where the vector $e_N \in \mathbb{R}^N$, i.e. L has a zero eigenvalue. Since the graph is connected this eigenvalue has multiplicity one [3]. It follows automatically that the matrix $Q = L \otimes I_n$ has a zero eigenvalue with multiplicity n. Note also that the matrix Q is symmetric and positive semidefinite, since

$$x^T (L \otimes I_n) x = x^T Q x = \sum_{i<j} a_{ij} \|x_i - x_j\|^2 \geq 0$$

for all x, but

$$e_{Nn}^T Q e_{Nn} = 0$$

because using simple rules for the Kronecker product we have that $Qe_{Nn} = (L \otimes I_n)(e_N \otimes e_n) = (Le_N) \otimes (I_n e_n) = 0$. We conclude that the function

$$f(x_1, \cdots, x_N) = \frac{1}{2} x^T (L \otimes I_n) x$$

is convex quadratic. Corresponding to the variable x we also consider the separable convex set $X = \prod_{i=1}^N X_i$.

B. Distributed gradient projection algorithm

In this section we solve the optimization problem (4) using the gradient projected algorithm (see e.g. [1], [11]). We assume in the sequel that the graph G and the adjacency matrix A to be fixed and satisfy the properties given in Assumption 1. We obtain a fully distributed algorithm which is also convergent. Let us define the gradient projection algorithm for the constrained convex quadratic problem:

$$\min_{x_i \in X_i} f(x_1, \cdots, x_N)$$

i.e. we have the following iterative algorithm

$$x^{k+1} = [x^k - \alpha_k (L \otimes I_n) x^k]_X,$$

where $\alpha_k \geq 0$ is the step-size and it will be defined in the sequel. Due to the special separable structure of our set X the previous iterations can be written equivalently for each node i as:

$$x_i^{k+1} = [(1 - \alpha_k \sum_{j \in \mathcal{N}(i)} a_{ij}) x_i^k + \sum_{j \in \mathcal{N}(i)} \alpha_k a_{ij} x_j^k]_{X_i}, \quad \forall i. \tag{5}$$

Comparing the iteration (5) with the iteration (2) we see that now we have a fully distributed scheme, where the agents must exchange information according to our information exchange model given by the graph G. Clearly, we can recover (2) from (5) when the graph is complete (i.e. each agent exchanges information with all the others). Let us define the matrix $P^k = [p_{ij}^k]$ as:

$$p_{ij}^k = \begin{cases} 1 - \alpha_k \sum_{l=1}^N a_{il} & \text{if } i = j \\ \alpha_k a_{ij} & \text{if } i \neq j. \end{cases}$$

It follows immediately that matrix P^k is symmetric and $P^k e_N = e_N$, i.e. it has an eigenvalue $\lambda = 1$. Moreover, if the following inequality holds

$$\alpha_k \leq \frac{1}{\max_i \{\sum_{j \in \mathcal{N}(i)} a_{ij}\}},$$

then all the entries of P^k are nonnegative and thus P^k is a *doubly stochastic* matrix. From (5) we obtain the following update rule (constrained consensus protocol):

$$x_i^{k+1} = [\sum_{j \in \mathcal{N}(i) \cup \{i\}} p_{ij}^k x_j^k]_{X_i} \tag{6}$$

We want to prove convergence for the iteration (5) or equivalently (6). It is easy to see that the function f is convex and has a Lipschitz continuous gradient (since it is a quadratic function). We denote the Lipschitz constant with L_f. Using standard arguments from linear algebra (in particular from Gershgorin Theorem) we derive that

$$L_f = \|Q\| = \|L\| \leq 2 \max_i \{\sum_{j \in \mathcal{N}(i)} a_{ij}\}.$$

We now show that if the step length α_k is chosen appropriately, this distributed gradient projected scheme is convergent:

Theorem 2: If Assumption 2 holds and the step length $\alpha_k \in (\epsilon, \ \frac{2}{L_f} - \epsilon)$ for all k, where $\epsilon > 0$ is sufficiently small, then the sequence $\{x_i^k\}_k$ has a common limit point $x^* \in \cap_{i=1}^N X_i$ for all i.

Proof: Let $x^* \in \cap_i X_i$ and denote $\mathbf{x}^* = e_N \otimes x^* \in X$. Note that the gradient of f at $\mathbf{x}^*$ satisfies $\nabla f(\mathbf{x}^*) = Q\mathbf{x}^* = (L \otimes I_n)(e_N \otimes x^*) = 0$. Based on this identity we can derive the following sequence of inequalities:

$$\begin{aligned} &\|x^{k+1} - \mathbf{x}^*\|^2 \leq \\ &\|x^k - \mathbf{x}^* - \alpha_k (\nabla f(x^k) - \nabla f(\mathbf{x}^*))\|^2 \leq \\ &\|x^k - \mathbf{x}^*\|^2 - \alpha_k (\frac{2}{L_f} - \alpha_k) \|\nabla f(x^k) - \nabla f(\mathbf{x}^*)\|^2 = \\ &\|x^k - \mathbf{x}^*\|^2 - \alpha_k (\frac{2}{L_f} - \alpha_k) \|\nabla f(x^k)\|^2. \end{aligned}$$

The first inequality follows from the non-expansive property of the projection and second inequality follows from well-known properties of a function with Lipschitz continuous gradient (see[2] [11]). Note that the largest decrease

[2]If the gradient of the function f is Lipschitz, then the following inequality holds for all x, y:

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq \frac{1}{L_f} \|\nabla f(x) - \nabla f(y)\|^2.$$

in the previous inequalities is obtained by maximizing $\max_{\alpha>0} \alpha(\frac{2}{L_f} - \alpha)$, i.e. for

$$\alpha_k = \frac{1}{L_f}.$$

Since the sets X_i are bounded, it follows from the previous inequalities that $\|x^k - \mathbf{x}^*\| \leq \|x^0 - \mathbf{x}^*\|$ and thus the sequence x^k is also bounded. Therefore it contains a convergent subsequence. For simplicity of the exposition we assume that the entire sequence x^k is convergent to some $y^* = [{y_1^*}^T \cdots {y_N^*}^T]^T \in \mathbb{R}^{Nn}$, i.e. $\lim_k x_i^k = y_i^*$. Since $x_i^k \in X_i$ with X_i compact sets, it follows that $y_i^* \in X_i$. It remains to show that $y_i^* = y_j^*$ for all $i \neq j$.

Adding the inequalities from above, taking into account that $\alpha_k \in (\epsilon,\ \frac{2}{L_f} - \epsilon)$, and then considering $k \to \infty$, we obtain $\lim_{k\to\infty} \nabla f(x^k) = 0$ or equivalently $\lim_{k\to\infty}(L \otimes I_n)x^k = 0$. We conclude that

$$\lim_{k\to\infty} {x^k}^T (L \otimes I_n) x^k = \lim_{k\to\infty} \sum_{i<j} a_{ij} \|x_i^k - x_j^k\|^2 = 0.$$

Since $a_{ij} > 0$ and the graph is connected, we have that $\lim_{k\to\infty} x_i^k - x_j^k = 0$ for all $i \neq j$. Therefore, there exists some $x^* \in \cap_{i=1}^N X_i$ such that $y_i^* = x^*$, i.e. $\lim_{k\to\infty} x_i^k = x^*$. ■

III. Consensus algorithms for the convex feasibility problem with variable weights

A. A primal-dual distributed consensus algorithm

The theory that we will develop in this section is related to recent work on network design problems for achieving faster consensus algorithms. For example, in [7] design of the weights of a network is considered and solved using semi-definite convex programming. This leads to an increase in algebraic connectivity of a network that is a measure of speed of convergence of consensus algorithms. An alternative approach is to keep the weights fixed and design the topology of the network to achieve high algebraic connectivity based on random rewiring [8]. In the previous section we have provided a distributed gradient algorithm in which the topology of the graph G and the corresponding weights are fixed but the step size is varying. In this section we devise a primal-dual distributed method for solving the feasibility problem (FP) using varying weights corresponding to a fixed topology for the graph G and also a fixed step size.

The following assumption is valid in this section:

Assumption 2: We consider a directed graph $G = (V, E)$ that is strongly connected.

We introduce first some definitions and then we provide a novel reformulation of problem (FP) as a structured optimization problem. For a positive integer p we denote the standard simplex in $\mathbb{R}^p$ with Δ, i.e.

$$\Delta = \{a \in \mathbb{R}^p : \sum_{i=1}^p a_i = 1,\ a_i \geq 0\ \forall i\}.$$

We start from a well-known relation in the optimization field, namely that the maximum over a finite number of points can be recast as a linear program with constraints over the simplex:

Lemma 1 ([11]): Given the nonnegative numbers $\alpha_1, \cdots, \alpha_p$, then the following holds

$$\max\{\alpha_1, \cdots, \alpha_p\} = \max_{a\in\Delta} \sum_{i=1}^p a_i \alpha_i.$$

Based on the previous lemma, we formulate a new convex optimization problem for solving the feasibility problem (FP). Let us note that the optimal solution of the following convex optimization problem

$$\min_{x_i \in X_i} \sum_{i=1}^N \max_{j\in\mathcal{N}(i)} \{\|x_i - x_j\|^2\} \tag{7}$$

is also a solution of our original problem (FP) (since the graph is strongly connected). Note that we fix in advance the communication topology of the associated network. Using now the previous lemma for the convex optimization problem (7), we obtain the following $\min - \max$ optimization problem with a convex-concave objective function in the variables x_i and a_{ij}:

$$\min_{x_i\in X_i} \max_{a_i\in\Delta_i} \sum_{i=1}^N \sum_{j\in\mathcal{N}(i)} a_{ij}\|x_i - x_j\|^2,$$

where $a_i = [a_{ij}]_{j\in\mathcal{N}(i)} \in \mathbb{R}^{|\mathcal{N}(i)|}$ with $|\mathcal{N}(i)|$ denotes the number of neighbors of node i in the graph G and the simplex $\Delta_i \subset \mathbb{R}^{|\mathcal{N}(i)|}$. Since the sets X and $\Delta = \Delta_1 \times \cdots \times \Delta_N$ are compact and since the objective function is convex in x and concave in $a = [a_{ij}]^T \in \mathbb{R}^{\sum_i |\mathcal{N}(i)|}$ we can interchange $\min$ with $\max$ in the previous optimization problem (see e.g. [12]), i.e

$$\max_{a_i\in\Delta_i} \min_{x_i\in X_i} \sum_{i;j\in\mathcal{N}(i)} a_{ij}\|x_i - x_j\|^2. \tag{8}$$

Note that from this formulation we can view the variables a_{ij} as dual variables. We introduce the following notation:

$$\Psi(a, x) = \sum_{i;j\in\mathcal{N}(i)} a_{ij}\|x_i - x_j\|^2,$$

which is a convex-concave function.

Using for example the Uzawa iteration adoption (see [13]) of the Arrow-Hurwicz gradient method for finding saddle points for the $\max - \min$ problem (8) we obtain the following iterations:

$$x^{k+1} = [x^k - \alpha\partial_x\Psi(a^k, x^k)]_X$$
$$a^{k+1} = [a^k + \alpha\partial_a\Psi(a^k, x^k)]_\Delta,$$

where $\alpha > 0$ is a fixed step size, $\partial_x\Psi$ denotes the partial derivative of the function Ψ with respect to x. Let us note that this is a fully distributed algorithm where we update both the primal variables x_i but also the weights (dual variables) a_{ij} using just information from the neighbors. Indeed, explicitly we can write for all $i = 1 \cdots N$:

$$x_i^{k+1} = [(1 - \alpha \sum_{j \in \mathcal{N}(i)} a_{ij}^k) x_i^k + \sum_{j \in \mathcal{N}(i)} \alpha a_{ij}^k x_j^k]_{X_i} \quad (9)$$

$$a_i^{k+1} = [a_i^k + \alpha d_i^k]_{\Delta_i}, \quad (10)$$

where the components of $d_i^k \in \mathbb{R}^{|\mathcal{N}(i)|}$ are given by $\|x_i^k - x_j^k\|^2$ for $j \in \mathcal{N}(i)$. We should remark that the only information that the neighbor $j \in \mathcal{N}(i)$ must send to i are their own updates x_j^k.

Note that in the proposed algorithm the associated network leads to a directed graph (the adjacency matrix is not symmetric anymore) having a dynamic topology. This topology depends on the state of all agents (nodes) x_i and is determined locally for each agent , i.e. the topology is a state-dependent graph. Moreover, it might be possible that at some iteration k the strong connectivity of the graph G to be lost since we can have $a_{ij}^k = 0$ for some $j \in \mathcal{N}(i)$ (see also the example in (12)). We can also remark that our iteration (9) is similar to the update rule for constrained consensus given in [5]:

$$x_i^{k+1} = [p_{ii}^k x_i^k + \sum_{j \in \mathcal{N}(i)} p_{ij}^k x_j^k]_{X_i}, \quad (11)$$

where for convergence of such consensus iteration to a common point $x^* \in \cap_i X_i$ the varying weights must satisfy certain conditions, such as: doubly stochasticity, connectivity, bounded intercommunication interval, etc (see [5] for more details). However, the main difference between our iterations (9)–(10) and the iteration (11) from above derived in the paper [5] is that in our case we provide also update rules for the weights a_{ij}^k while there are no rules for updating the weights p_{ij}^k in [5].

Convergence of our primal-dual distributed algorithm follows from standard optimization theory:

Theorem 3: [13] The primal-dual distributed projected algorithm given by the iteration (9)-(10) is convergent, i.e. $\lim_{k \to \infty} x_i^k = x^* \in \cap_{i=1}^N X_i$ for all i.

Note that for our new algorithm, given by the above two iterations (9)-(10), the computations are also very cheap, consisting on vector operations and projections on simple sets. There exist very efficient algorithms for projection of a point on the simplex and moreover the dimensions of each simplex is small $\Delta_i \subset \mathbb{R}^{|\mathcal{N}(i)|}$, with $|\mathcal{N}(i)| \ll N$.

It is important to know also that we can apply other first order methods for finding saddle points for the $\min \max$ optimization problem (8), that can be more efficient than Uzawa iteration presented in this paper, such as Korpelevich's algorithm [14]. Note that any first order method for finding saddle points for (8) is fully distributed, i.e. for the iteration in node i we just need to use information from its neighbors.

B. Extensions to separable convex optimization problems

The theory presented here can be used to solve distributively more complex separable convex problems. For example we consider solving the following separable convex optimization problem in a distributed way:

$$\min_{x \in \cap_i X_i} \sum_{i=1}^N f_i(x).$$

This problem can be written equivalently as a separable convex problem (see [10]):

$$\min_{x_i \in X_i, x_i = x_j} \sum_{i=1}^N f_i(x_i).$$

Using a penalty type method for removing the coupling linear constraints we obtain:

$$\min_{x_i \in X_i} \sum_{i=1}^N f_i(x_i) + \mu \max_{j \in \mathcal{N}(i)} \{\|x_i - x_j\|^2\},$$

where $\mu > 0$ is a penalty parameter. Proceeding as before, we obtain a distributed primal-dual gradient projected algorithm whose convergence can be proved using again standard arguments from optimization theory. Note that in theory μ should vary and converge to ∞.

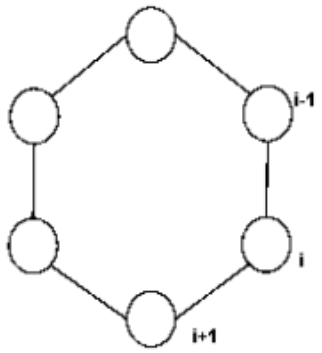


Fig. 1. A graph having a ring topology.

IV. Preliminary numerical results

In this section we compare, in terms of the number of iterations, the distributed gradient algorithm given in (5) with our new primal-dual distributed consensus algorithm given by the iterations (9)–(10). We consider a network for which the associated graph has N nodes and each node has two neighbors, i.e. a ring topology (see Fig. 1):

$$\mathcal{N}(i) = \begin{cases} \{2, N\} & \text{if } i = 1 \\ \{i-1, i+1\} & \text{if } 1 < i < N \\ \{N-1, 1\} & \text{if } i = N. \end{cases}$$

Note that for such a graph in the form of a ring the graph connectivity (which characterizes the performance of a consensus algorithm), i.e. the second smallest eigenvalue of the normalized Laplacian is close to 0, namely $1 - \cos(2\pi/N)$. In conclusion, the convergence of the classical consensus protocol for such a graph is slow (see [3] for more details). From our simulations (see Table I) we can see however, that even for such a graph our primal-dual distributed consensus algorithm works very well.

N	n	nr. it. alg. (5)	nr. it. alg. (9)–(10)
10	100	2.381	219
10	300	2.728	236
25	50	17.683	1.618
50	30	20.231	2.324
50	100	48.570	9.685

TABLE I

COMPUTATIONAL RESULTS FOR THE CONVEX FEASIBILITY PROBLEM (FP): NUMBER OF ITERATIONS FOR DISTRIBUTED GRADIENT ALGORITHM (5) AND NEW PRIMAL-DUAL DISTRIBUTED CONSENSUS ALGORITHM (9)–(10).

For both algorithms the step size takes the same value, i.e for the distributed gradient algorithm given by the iteration (5) we take the optimal step size $\alpha_k = \frac{1}{L_f}$ for all $k \geq 0$ (according to Theorem 2) and for the primal-dual distributed consensus algorithm given by the iteration (9)-(10), the step size is $\alpha = \frac{1}{L_f}$ as well. Furthermore, the sets X_i are considered to be half-spaces:

$$X_i = \{x_i \in \mathbb{R}^n : a_i^T x_i \leq b_i\},$$

generated randomly (to generate a_i we use normal distribution, while to generate b_i we use uniform distribution) such that $0 \in X_i$, and thus $0 \in \cap_i^N X_i$.

It is straightforward to show that the projection of a point x_0 on the half-space $Y = \{x \in \mathbb{R}^n : a^T x \leq b\}$ can be done explicitly, i.e. if

$$[x_0]_Y = \arg\min_x \{\|x - x_0\|^2 : a^T x \leq b\},$$

then

$$[x_0]_Y = x_0 - \frac{a^T x_0 - b}{a^T a} a,$$

provided that $x_0 \notin Y$. Similarly, the projection of a point $a = [a_1 \ a_2]^T \in \mathbb{R}^2$ on the simplex $\Delta \subset \mathbb{R}^2$ can be computed explicitly as:

$$[a]_\Delta = \begin{cases} \begin{bmatrix} \frac{1+a_1-a_2}{2} \\ \frac{1-a_1+a_2}{2} \end{bmatrix} & \text{if} \quad 0 < \frac{1+a_1-a_2}{2} < 1 \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \text{if} \quad \frac{1+a_1-a_2}{2} \geq 1 \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \text{if} \quad \frac{1+a_1-a_2}{2} \leq 0. \end{cases} \tag{12}$$

We illustrate the efficiency of our primal-dual distributed consensus algorithm in Table 1. Note that the distributed gradient algorithm almost needs an average more than ten times the number of iterations than our proposed primal-dual distributed consensus algorithm.

V. CONCLUSIONS

In this paper we have discussed two gradient type algorithms for solving the convex feasibility problem. We have shown that both algorithms are fully distributed having many similarities with a consensus protocol. Moreover, we have succeeded to provide a way to update the weights in the consensus algorithm by reformulation of the problem in a primal-dual form and thus interpreting these weights as dual variables. Therefore, in our algorithm the weights are updated following some precise rules, while in most existing distributed algorithms based on consensus principles the weights have to be tuned since they are considered parameters for those methods. Convergence of these distributed schemes follows using standard arguments from convex optimization theory. We hope that this new interpretation of the weights may open a new window of opportunity for algorithmic research in this area of distributed methods based on consensus arguments. Extensions to more general convex optimization problems were also given. Preliminary simulation results show that our primal-dual distributed consensus algorithm is sometimes even ten times faster in the number of iterations than the distributed gradient algorithm.

ACKNOWLEDGMENTS

The research leading to these results has received funding from: the European Union, Seventh Framework Programme (FP7/2007–2013) under grant agreement no 248940; CNCSIS-UEFISCSU (project TE, no. 19/11.08.2010); ANCS (project PN II, no. 80EU/2010).

It was also supported by Research Council KUL: GOA AMBioRICS, CoE EF/05/006, OT/03/12, PhD/postdoc & fellow grants; Flemish Government: FWO PhD/postdoc grants, FWO projects G.0499.04, G.0211.05, G.0226.06, G.0302.07.

REFERENCES

[1] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical Methods.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[2] J. E. Spingarn, "A primal-dual projection method for solving systems of linear inequalities," *Linear Algebra and its Applications*, vol. 65, pp. 45–62, 1985.

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

[4] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, January 2009.

[5] A. Nedic, A. Ozdaglar, and P. Parrilo, "Constrained consensus and optimization in multi-agent networks," *to appear in IEEE Transactions on Automatic Control*, LIDS report 2779, 2009.

[6] B. Johansson, T. Keviczky, M. Johansson, and K. Johansson, "Subgradient methods and consensus algorithms for solving convex optimization problems," in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, Dec. 2008.

[7] L. Xiao and S. Boyd, "Fastest linear iteration for distributed averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, 2004.

[8] R. Olfati-Saber, "Ultrafast consensus in small-world networks," in *Proceedings of the 2005 American Control Conference*, June 2005, pp. 2371–2378.

[9] I. Necoara and J. A. K. Suykens, "Application of a smoothing technique to decomposition in convex optimization," *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2674–2679, 2008.

[10] ——, "An interior-point Lagrangian decomposition method for separable convex optimization," *Journal of Optimization Theory and Applications*, vol. 143, no. 3, pp. 567–588, 2009.

[11] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Boston: Kluwer, 2004.

[12] R. T. Rockafellar, *Convex Analysis*, ser. Princeton Mathematics. Princeton University Press, 1970, vol. 28.

[13] H. Uzawa, "Iterative methods for concave programming," in *Studies in Linear and Nonlinear Programming*, K. Arrow, L. Hurwicz, and H. Uzawa, Eds., 1958, pp. 154–165.

[14] G. Korpelevich, "The extragradient method for finding saddle points and other problems," *Matecon*, vol. 12, pp. 747–756, 1976.