Decomposition and Simultaneous Projection Methods for Convex Feasibility Problems with Application to Robustness Analysis of Interconnected Uncertain Systems

Sina Khoshfetrat Pakazad, Anders Hansson, Martin S. Andersen, Anders Rantzer

Division of Automatic Control
E-mail: sina.kh.pa@isy.liu.se, hansson@isy.liu.se, martin.andersen@ucla.edu, anders.rantzer@control.lth.se

13th May 2011

Report no.: LiTH-ISY-R-3012

Address:
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

WWW: http://www.control.isy.liu.se

Technical reports from the Automatic Control group in Linköping are available from http://www.control.isy.liu.se/publications.
Abstract
In this paper a specific class of convex feasibility problems are considered and tailored algorithms to solve this class of problems are introduced. First, the Nonlinear Cimmino Algorithm is reviewed. Then motivated by the special structure of the problems at hand, a modification to this method is proposed. Next, another method for solving the dual problem of the provided problem is presented. This leads to similar update rules for the variables as in the modified Nonlinear Cimmino Algorithm. Then an application for the proposed algorithms on the robust stability analysis of large scale weakly interconnected systems is presented and the performance of the proposed methods are compared.

Keywords: Convex feasibility problems, Robust stability analysis, Decomposition, Simultaneous projection, Distributed.
Decomposition and Simultaneous Projection Methods for Convex Feasibility Problems with Application to Robustness Analysis of Interconnected Uncertain Systems

Sina Khoshfetrat Pakazad, Anders Hansson, Martin S.Andersen and Anders Rantzer

Abstract—In this paper a specific class of convex feasibility problems are considered and tailored algorithms to solve this class of problems are introduced. First, the Nonlinear Cimmino Algorithm is reviewed. Then motivated by the special structure of the problems at hand, a modification to this method is proposed. Next, another method for solving the dual problem of the provided problem is presented. This leads to similar update rules for the variables as in the modified Nonlinear Cimmino Algorithm. Then an application for the proposed algorithms on the robust stability analysis of large scale weakly interconnected systems is presented and the performance of the proposed methods are compared.

Index Terms—Convex feasibility problems, Robust stability analysis, Decomposition, Simultaneous projection, Distributed.

I. INTRODUCTION

Convex feasibility problems have been studied quite thoroughly through the years and there exist a substantial number of publications in the area, [14], [15], [10], [7], [9], [8], [17], [2], [20], [15], [16]. Most of the major proposed methods for solving the convex feasibility problem can be grouped into the following classes, successive projection methods which consider projecting into different constraint sets one at a time with cyclic or more general control sequences, [10], [15], [7], simultaneous projection methods with weighted control, [17], [20], and Subgradient projection methods with either cyclic or weighted control, [8], [20], [21]. The terms cyclic and weighted mainly refer to the sequential and simultaneous nature of the update rules in the corresponding algorithms, respectively. For more information on these terms and a survey on the algorithms for solving convex feasibility problems refer to [2].

This paper focuses on solving convex feasibility problems where none of the constraints involved in the problem is dependent the whole optimization vector. Considering this imposed structure, it is then possible to modify the already existing methods or tailor new algorithms, efficient for solving this type of problems. It is the aim of this paper to provide projection based methods that can be solved in a distributed manner and possibly over a network of agents.

Considering the fact that simultaneous projection methods with weighted control are easier to implement in a distributed framework than for instance methods with cyclic control, in this paper we only consider this type of methods. One of the widely known algorithms in this class of methods is the Nonlinear Cimmino Algorithm, [17], which provides the baseline for one of the proposed algorithms in this paper. This method is then referred to as the Modified Nonlinear Cimmino Algorithm.

The other proposed method in this paper is mainly inspired by the work in [5], which approaches the convex feasibility problem by formulating its Augmented Lagrangian and solving the corresponding dual problem via the Alternating Direction Method of Multipliers, ADMM, [5], [3].

The paper has the following formation. Section II describes the problem formulation and the structure of the problem of interest. In Section III, first Subsection III-A presents the Nonlinear Cimmino Algorithm which provides the necessary background for Section IV. That is followed by a description of the Augmented Lagrangian and the ADMM algorithm in Subsection III-B. This is later used in Section V. Section IV introduces modifications to the Nonlinear Cimmino Algorithm. Using a slightly different approach, Section V introduces another algorithm to solve the problem at hand. An application of the introduced methods is then presented in Section VI. Finally, the capabilities of the proposed approaches are tested in Section VII and the paper is concluded with some final remarks and suggestions for future work in Section VIII.

Notation

The notation in this paper is fairly standard. Symbols \(\mathbb{R}\) and \(\mathbb{N}\) denote the real and positive integer numbers and \(\mathbb{S}^d\) denotes \(d \times d\) symmetric matrices. The Euclidian norm or 2-norm is denoted by \(\|\cdot\|_{2}\) and \(\bigcap\) represents the intersection operator. We use \(x'\) and \(x^\ast\) to represent the transpose and conjugate transpose of the vector \(x\). Unless clearly stated otherwise, by \(x_{j,k}^i\) we denote the \(j^\text{th}\) element in the vector \(x\) corresponding to the \(i^\text{th}\) subproblem at the \(k^\text{th}\) iteration. Let \(J \in \mathbb{N}^N\), then \(x_J\) represents a vector of elements of \(x\) with the indices marked by the elements in \(J\), with the same ordering.

II. PROBLEM FORMULATION

A. Problem Statement

Consider the following optimization problem

\[
\text{Find } x \\
\text{subj. to } x \in \bigcap_{i=1}^N C_i, \tag{1}
\]
where $x \in \mathbb{R}^n$ and $\mathcal{C}_i = \{x | \mathcal{F}_i(x) \preceq_k 0\}$, with either $\mathcal{F}_i : \mathbb{R}^n \mapsto \mathbb{R}^{d_i}$ representing a $K$-convex mapping or $\mathcal{F}_j : \mathbb{R}^n \mapsto \mathbb{S}^{d_j}$ describing an LMI, for $i = 1, \cdots, N$. Assume that $\mathcal{F}_i$ is only dependent on certain components of $x$. Let $J^i \in \mathbb{N}^{m_i}$ denote the vector of indices of components of $x$ present in $\mathcal{F}_i$, where $m_i \leq n$, and let $\mathcal{J}^i \subseteq \{1, \cdots, n\}$ be the set of those indices, for $i = 1, \cdots, N$.

It is also beneficial to define $I^i \in \mathbb{N}^{d_i}$, which represents the vector of indices of constraints, $\mathcal{F}_i$, that are dependent on the $i$th component of $x$, where $i \leq N$, and also $\mathcal{I}^i \subseteq \{1, \cdots, N\}$ which is the set of those indices, for $i = 1, \cdots, n$.

In this report, this class of problems are considered and methods for solving these problems are proposed and for the sake of simplicity only LMI constraints are considered. The extension to the other cases is straightforward. Next this structure is exploited in a more formal manner.

**B. Structure Exploitation**

Let $\tilde{\mathcal{C}}_i = \{s^i | \tilde{\mathcal{F}}_i(s^i) \preceq 0\}$ where $\tilde{\mathcal{F}}_i : \mathbb{R}^{m_i} \mapsto \mathbb{S}^{d_i}$, $s^i \in \mathbb{R}^{m_i}$ and let $S \in \mathbb{R}^{\sum_{i=1}^{N} m_i}$ with $S = \begin{bmatrix} s^1 \\ \vdots \\ s^N \end{bmatrix}$. Define the following optimization problem

$$\text{Find } S \text{ subject to } s^i \in \tilde{\mathcal{C}}_i, \text{ for } i = 1, \ldots, N,$$

(2)

where $x \in \mathbb{R}^n$ and for $J^i \in \mathbb{N}^{m_i}$, as defined in Section II-A, and where $x_{J^i}$ denotes the vector $\begin{bmatrix} x_{J^i_1} \\ \vdots \\ x_{J^i_{m_i}} \end{bmatrix}$.

It is possible to define $\tilde{\mathcal{F}}_i$ such that $\tilde{\mathcal{C}}_i = \tilde{\mathcal{C}}_i \times \mathbb{R}^{n-m_i}$, and as a result, a solution to the problem defined in Section II-A can be achieved by solving the problem defined in (2).

This problem can also be reformulated as below

$$\text{minimize } \sum_{i=1}^{N} g_i(s^i) \text{ subject to } s^i = x_{J^i}, \text{ for } i = 1, \ldots, N,$$

(3)

where

$$g_i(s^i) = \begin{cases} \infty, & s^i \notin \tilde{\mathcal{C}}_i, \\ 0, & s^i \in \tilde{\mathcal{C}}_i, \end{cases}$$

(4)

represents the indicator function for $\tilde{\mathcal{C}}_i$. Note that in problems (2) and (3), $s^i$ represents local copies of relevant global variables, $x_{J^i}$. From now on, vectors $s^i$ and $x_i$ are referred to as local and global variables, respectively.

This formulation is later used to provide iterative algorithms for solving the problem defined in section II-A.

**III. MATHEMATICAL PRELIMINARIES**

**A. Nonlinear Cimmino Algorithm**

Consider the problem in (1), and let $P_{\mathcal{C}_i}(x)$ represent the projection of $x$ into $\mathcal{C}_i$. This projection is defined as below

$$P_{\mathcal{C}_i}(x) = \text{argmin}_{\bar{x} \in \mathcal{C}_i} \|x - \bar{x}\|_2^2.$$ 

(5)

Define

$$P(x) = \sum_{i=1}^{N} \alpha_i P_{\mathcal{C}_i}(x),$$

(6)

where $\alpha_i > 0$ and $\sum_{i=1}^{N} \alpha_i = 1$. Then if (1) is strictly feasible, the sequence generated by $x^{k+1} = P(x^k)$ converges to a point in $\bigcap_{i=1}^{N} \mathcal{C}_i$. This method can be accelerated by choosing the weights, $\alpha_i$, wisely. The following way of choosing the weights, taken from [17], is proven to accelerate the Cimmino algorithm. Let $I(x) = \{i | x \notin \mathcal{C}_i\}$ and $C(x)$ denote the cardinality of $I(x)$. Define

$$\mu(x) = \begin{cases} \left(\sum_{i \in I(x)} \alpha_i\right)^{-1} C(x) \geq 2, \\ 1, \text{ otherwise}, \end{cases}$$

(7)

and

$$\bar{P}(x) = x + \mu(x) (P(x) - x).$$

(8)

Defining $v_i(x) = \alpha_i \mu(x)$, (8) can be rewritten as below

$$\bar{P}(x) = \begin{cases} \sum_{i \in I(x)} v_i(x) P_{\mathcal{C}_i(x)}, & C(x) \geq 2, \\ P_x, & \text{otherwise} \end{cases}$$

(9)

Note that $v_i(x) > 0$ and $\sum_{i \in I(x)} v_i(x) = 1$. Then the update $x^{k+1} = \bar{P}(x^k)$ will also converge to a point in the feasible set. For more details on the Accelerated Nonlinear Cimmino Algorithm refer to [17].

**B. Augmented Lagrangian and the ADMM Algorithm**

Consider the following optimization problem with equality constraints

$$\text{minimize } F(x) \text{ subject to } Ax = b,$$

(10)

The Augmented Lagrangian for this problem can be defined as

$$L_\mu(x, \lambda) = F(x) + \lambda^T (Ax - b) + \frac{\mu}{2} \|Ax - b\|_2^2,$$

(11)

which can also be rewritten as
Augmented Lagrangian for this problem is defined as below:

\[ L_\rho(x, \lambda) = F(x) + \lambda^T (Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2 \]

\[ = F(x) + \lambda^T (Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2 + \frac{1}{2\rho} \|\lambda\|_2^2 - \frac{1}{2\rho} \|\lambda\|_2^2 \]  

(12)

Consider the following optimization problem

\[ \text{minimize } F(S) \]

subj. to \( S \in C. \)  

(13)

This optimization problem can be reformulated as below:

\[ \text{minimize } F(S) + g(q) \]

subj. to \( S - q = 0, \)  

(14)

where \( g(q) \) represents the indicator function for the set \( C. \) Considering (14) and by defining \( \bar{\lambda} = \frac{\lambda}{\rho}, \) the normalized Augmented Lagrangian for this problem is defined as

\[ L_\rho(S, q, \bar{\lambda}) = F(S) + g(q) + \frac{\rho}{2} \|S - (q - \bar{\lambda})\|_2^2 - \frac{\rho}{2} \|\bar{\lambda}\|_2^2. \]  

(15)

This problem can be solved using the Alternating Direction Method of Multipliers, ADMM, by the following iterative scheme.

\[ x_{i}^{k+1} = \sum_{j \in \mathcal{I}_i} \alpha_j \left( P_{C_j}(x^k) \right)_i, \quad i = 1, \ldots, n. \]  

(18)

In order to tailor this algorithm for this class of problems, it seems intuitive to discard the terms that relate to the constraints that do not affect \( x_i, \) i.e. the second term in the right hand side of (18). Also for the modified algorithm to be consistent, new weights, \( \bar{\alpha}_{ij}, \) are introduced such that \( \sum_{j \in \mathcal{I}_i} \bar{\alpha}_{ij} = 1. \) As a result we end up with the following modified update rule

\[ x_{i}^{k+1} = \sum_{j \in \mathcal{I}_i} \bar{\alpha}_{ij} \left( P_{C_j}(x^k) \right)_i. \]  

(19)

As a special case, if the weights are chosen such that \( \bar{\alpha}_{ij} = \frac{1}{|\mathcal{I}_i|}, \forall j \in \mathcal{I}_i, \) then (19) gives

\[ x_{i}^{k+1} = \frac{1}{|\mathcal{I}_i|} \sum_{j \in \mathcal{I}_i} \left( P_{C_j}(x^k) \right)_i. \]  

(20)

The update rule in (20) can also be obtained as follows. Consider the problem in (3). This problem is equivalent to the following optimization problem

\[ \text{minimize } \sum_{i=1}^{N} g_i(s^i) \]

subj. to \( \frac{1}{2} \sum_{i=1}^{N} \|s^i - x_{J_i}\|_2^2 \leq 0. \)  

(21)

The Lagrangian for this problem is defined as below

\[ L(S, x, \beta) = \sum_{i=1}^{N} \left\{ g_i(s^i) + \frac{\beta}{2} \|s^i - x_{J_i}\|_2^2 \right\} \]  

(22)

where \( \beta \in \mathbb{R}^+ \) is the dual variable.

Assuming feasibility of the problem in (1), then for the problem in (21) strong duality holds by definition, chapter 5 of [6] and this problem can be solved through its dual problem, based on (22), by a dual ascent method, chapter 6 [4], using the following block coordinate descent scheme,

\[ S^{k+1} = \arg \min_S \left\{ \sum_{i=1}^{N} \left( g_i(s^i) + \frac{\beta}{2} \|s^i - x_{J_i}\|_2^2 \right) \right\}, \]

\[ x^{k+1} = \arg \min_x \left\{ \sum_{i=1}^{N} \beta \|s^{i,k+1} - x_{J_i}\|_2^2 \right\}, \]

\[ \beta^{k+1} = \beta^k + \alpha \sum_{i=1}^{N} \|s^{i,k+1} - x_{J_i}\|_2^2, \]  

(23)

where \( \alpha \in \mathbb{R}^+ \) and the initial value for \( \beta \) should be positive, i.e. \( \beta^0 > 0. \) The \( S \)-iteration, or the update for the local variables, in (23) is separable with respect to \( s^i \) and can be decomposed into \( N \) local problems and solved in a distributed manner as below
\[
s^{i,k+1} = \arg\min_{s^i} \left\{ g_i(s^i) + \beta^k \left\| s^i - x_{j,i}^k \right\|^2_2 \right\} = P_{C_i} \left( x_{j,i}^k \right). \tag{24}
\]

Similarly, the \(x\)-iteration, or the update of the global variables, of the algorithm is also separable with respect to \(x_i\) and can be rewritten as \(N\) local problems

\[
x_i^{k+1} = \arg\min_{x_i} \left\{ \sum_{j \in I_i} \beta^k \left\| s_i^{j,k+1} - x_{j,i} \right\|^2_2 \right\}
= \arg\min_{x_i} \left\{ \sum_{j \in I_i} \beta^k \left\| s_i^{j,k+1} - x_{j,i} \right\|^2_2 \right\}
= \frac{1}{t_i} \sum_{j \in I_i} \left( s_i^{j,k+1} \right), \tag{25}
\]

where \(t_i^j\) is equal to the index of the component of \(s^j\) that corresponds to \(x_i\), for \(j = 1, \cdots, N\). Combining (24) and (25) results in the following update rule for the variables

\[
x_i^{k+1} = \frac{1}{t_i} \sum_{j \in I_i} \left( P_{C_j} \left( x_{j,i}^k \right) \right)_{t_i}^j . \tag{26}
\]

Considering the definition of \(C_i\) and the fact that both update rules in (20) and (26) only include constraints that affect \(x_i\), i.e. \(I_i\), and precisely pick out the terms relevant to \(x_i\), they are equivalent.

Here many details of the derivation has been omitted, but details are given in Section V for a very similar problem formulation.

The Nonlinear Cimmino Algorithm and the corresponding modified approach presented above, deal with the convex feasibility problem by investigating its primal problem or the dual problem based on the regular Lagrangian problem. Next, this problem is also approached through its corresponding dual problem which is defined based on its Augmented Lagrangian, and an iterative algorithm is proposed by solving this dual problem.

V. ADMM BASED ALGORITHM

Assume there exists a strictly feasible solution to the problem in (1). Then Slater’s condition holds and we have strong duality. As a result the problem in (3) can be solved through its dual, based on its Augmented Lagrangian, and using ADMM, [5],[3].

The Augmented Lagrangian for this problem is defined as

\[
L_{\rho}(s, x, \lambda) = \sum_{i=1}^{N} \left\{ g_i(s^i) + \frac{\rho}{2} \left\| s^i - (x_{j,i} - \lambda^i) \right\|^2_2 - \frac{\rho}{2} \left\| \lambda^i \right\|^2_2 \right\}, \tag{27}
\]

where \(\lambda^i \in \mathbb{R}^{m_i}\) is the vector of normalized Lagrange multipliers corresponding to the constraint \(s^i - x_{j,i} = 0\) and \(\bar{\lambda} = \left[ \bar{\lambda}_1^1 \right] \in \mathbb{R}^{N} \sum_{i=1}^{N} m_i\) is the normalized Lagrange multipliers for the problem in (3). From now on we will omit the word normalized.

Applying the ADMM algorithm of Section III-B to (27), results in the following iterative method

\[
S^{k+1} = \arg\min_{s^i} \left\{ \sum_{i=1}^{N} \left( g_i(s^i) + \frac{\rho}{2} \left\| s^i - (x_{j,i} - \lambda^{i,k}) \right\|^2_2 \right) \right\},
\]

\[
x_i^{k+1} = \arg\min_{x_i} \left\{ \sum_{j \in I_i} \frac{\rho}{2} \left\| s_i^{j,k+1} - (x_{j,i} - \lambda^{i,k}) \right\|^2_2 \right\},
\]

\[
\lambda^{i,k+1} = \lambda^{i,k} + (s^{i,k+1} - x_{j,i}^k).
\tag{28}
\]

Similar to the algorithm presented in Section IV, the \(S\)-iteration and \(x\)-iteration for (28) can also be written as below

\[
s_i^{j,k+1} = \arg\min_{s_i^j} \left\{ g_i(s^i) + \frac{\rho}{2} \left\| s^i - (x_{j,i} - \lambda^{i,k}) \right\|^2_2 \right\}
= P_{C_i} \left( x_{j,i}^k - \lambda^{i,k} \right) . \tag{29}
\]

\[
x_i^{k+1} = \arg\min_{x_i} \left\{ \sum_{j \in I_i} \frac{\rho}{2} \left\| s_i^{j,k+1} - (x_{j,i} - \lambda^{i,k}) \right\|^2_2 \right\}
= \frac{1}{t_i} \sum_{j \in I_i} \left( s_i^{j,k+1} + \lambda^{i,k} \right) . \tag{30}
\]

where \(t_i^j\) is equal to the index of the component of \(s^j\) that corresponds to \(x_i\), for \(j = 1, \cdots, N\).

In order to further clarify the remaining steps of the algorithm, the description of those steps is accompanied by an example. Consider the problem defined in (2) with the following information. Let \(x \in \mathbb{R}^6\), \(J^1 = \{1, 2, 3\}\), \(J^2 = \{2, 4, 6\}\), \(J^3 = \{1, 3, 5\}\), as a result \(m_i = 3\) for \(i = 1, 2, 3\), and \(T^1 = \{1\}\), \(T^2 = \{1, 2, 3\}\), \(T^3 = \{1, 3\}\), \(T^4 = \{2\}\), \(T^5 = \{3\}\) and \(T^6 = \{2\}\). Let \(s^i \in \mathbb{R}^3\) for \(i = 1, 2, 3\), and \(x_{j,1} = x_{j,2} = x_{j,3} = 0\), \(x_{j,4} = x_{j,5} = x_{j,6} = 0\). Then for instance, \(t_1^2 = 2\), \(t_2^3 = 1\), \(t_3^1 = 1\) and \(t_1^3 = 3\) and \(t_3^6 = 2\). As a result the update rule for \(x_{j,2}\) and \(x_{j,3}\) can be written as below

\[
x_{j,2}^{k+1} = \arg\min_{x_{j,2}} \left\{ \sum_{j \in I_2} \frac{\rho}{2} \left\| s^{j,k+1} - (x_{j,2} - \lambda^{j,k}) \right\|^2_2 \right\}
= \frac{1}{t_2^2} \sum_{j \in I_2} \left( s_i^{j,k+1} + \lambda^{j,k} \right) . \tag{31}
\]
Let the following holds for the update of the Lagrange multipliers. Let $j^* \in I_3$, then the update for $\bar{\lambda}_{j^*}^i$ can be written as

$$
\bar{\lambda}_{j^*}^{i,k+1} = \bar{\lambda}_{j^*}^{i,k} + \left( s_{j^*}^{i,k+1} - \frac{1}{l_3} \sum_{j \in I_3} (s_{j}^{i,k+1} + \lambda_{j}^{i,k}) \right)
$$

$$
= \bar{\lambda}_{j^*}^{i,k} + \left( s_{j^*}^{i,k+1} - \frac{1}{l_3} \left\{ \left( s_{j}^{i,k+1} + \bar{\lambda}_{j}^{i,k} \right) + \left( s_{j}^{i,k+1} + \bar{\lambda}_{j}^{i,k} \right) \right\} \right)
$$

$$
= \frac{l_3 - 1}{l_3} \bar{\lambda}_{j}^{i,k} + \frac{l_3 - 1}{l_3} s_{j}^{i,k+1} - \frac{1}{l_3} \left( s_{j}^{i,k+1} + \bar{\lambda}_{j}^{i,k} \right)
$$

$$
= \frac{l_3 - 1}{l_3} \bar{\lambda}_{j}^{i,k} + \frac{l_3 - 1}{l_3} s_{j}^{i,k+1} - \frac{1}{l_3} \left( s_{j}^{i,k+1} + \bar{\lambda}_{j}^{i,k} \right)
$$

$$
= \frac{1}{2} \left( \bar{\lambda}_{j}^{i,k} + s_{j}^{i,k+1} \right) - \frac{1}{2} \left( s_{j}^{i,k+1} + \bar{\lambda}_{j}^{i,k} \right). \tag{35}
$$

The remaining $l_3 - 1$ Lagrange multipliers present in (30) are also updated in the same manner, resulting in. [5]

$$
\sum_{j \in I_3} \bar{\lambda}_{j}^{i,k} = 0. \tag{34}
$$

This fact can also be illustrated for the example presented above. For instance, consider the dual variables corresponding to $x_3$, i.e. $\bar{\lambda}_{j}^{i} \forall j \in I_3$

$$
\bar{\lambda}_{j}^{3,k+1} = \bar{\lambda}_{j}^{3,k} + \left( s_{j}^{3,k+1} - \frac{1}{l_3} \sum_{j \in I_3} (s_{j}^{3,k+1} + \lambda_{j}^{3,k}) \right)
$$

$$
= \bar{\lambda}_{j}^{3,k} + \left( s_{j}^{3,k+1} - \frac{1}{l_3} \left\{ \left( s_{j}^{3,k+1} + \bar{\lambda}_{j}^{3,k} \right) + \left( s_{j}^{3,k+1} + \bar{\lambda}_{j}^{3,k} \right) \right\} \right)
$$

$$
= \frac{l_3 - 1}{l_3} \bar{\lambda}_{j}^{3,k} + \frac{l_3 - 1}{l_3} s_{j}^{3,k+1} - \frac{1}{l_3} \left( s_{j}^{3,k+1} + \bar{\lambda}_{j}^{3,k} \right)
$$

$$
= \frac{l_3 - 1}{l_3} \bar{\lambda}_{j}^{3,k} + \frac{l_3 - 1}{l_3} s_{j}^{3,k+1} - \frac{1}{l_3} \left( s_{j}^{3,k+1} + \bar{\lambda}_{j}^{3,k} \right)
$$

$$
= \frac{1}{2} \left( \bar{\lambda}_{j}^{3,k} + s_{j}^{3,k+1} \right) - \frac{1}{2} \left( s_{j}^{3,k+1} + \bar{\lambda}_{j}^{3,k} \right). \tag{36}
$$

Consequently

$$
\sum_{j \in I_3} \bar{\lambda}_{j}^{3,k} = \left( \bar{\lambda}_{j}^{3,k+1} + \bar{\lambda}_{j}^{3,k+1} \right) = 0. \tag{37}
$$

Considering (30), the update for the global variables can be written as below

$$
x_{l_3}^{k+1} = \frac{1}{l_3} \sum_{j \in I_3} (s_{j}^{i,k+1}) \tag{38}
$$

Having (29), (38)

$$
x_{l_3}^{k+1} = \frac{1}{l_3} \sum_{j \in I_3} (s_{j}^{i,k+1}) \tag{39}
$$

By comparing (20) and (39), the following points surface

1) Due to the fact that $x_{l_3}^{k+1}$ is merely a reordering of the global variables, the update rules (20) and (39) have a similar structure.

2) Unlike (20), the update rule in (39) includes the Lagrange multipliers corresponding to the constraint set $s^i - x_{l_3}^{k+1} = 0$, which act as Dykstra like correction terms [10], [15].

3) The proposed simultaneous projection methods in this paper, namely the Modified Nonlinear Cimmino algorithm and the ADMM based approach, can also be considered as the corresponding approaches to Von-Neumann and Dykstra successive projection methods, respectively. This is due to the existence of a similar correction term in the update rule of both the ADMM based and Dykstra algorithms. It is also well established that Dysktra algorithm performs better than the Von-Neumann approach, specifically due to the existence of this correction term. Consequently, the same is also expected for the approaches presented in this paper.

VI. APPLICATION

One of the possible applications of the proposed methods is in robust stability analysis of uncertain large scale interconnected systems with structured uncertainty. This problem can be investigated through a $\mu$ analysis framework, [11], which can be formulated as a Semi Definite Programming, SDP, problem involving the system matrix. Consider the following system description

$$
Y(s) = M(s)U(s), \tag{40}
$$

where $M(s)$ is a $m \times m$ transfer matrix, and let

$$
U(s) = \Delta Y(s), \tag{41}
$$

where $\Delta = \text{diag}(\delta_i)$, with $\delta_i \in \mathbb{R}$, $|\delta_i| \leq 1$ for $i = 1, \cdots, m$, represents the uncertainty in the system. This system is said to be robustly stable if there exists a positive definite $X(\omega)$ and $0 < \mu < 1$ such that

$$
M(j\omega)^*X(\omega)M(j\omega) - \mu^2X(\omega) < 0, \tag{42}
$$
for all $\omega$. Note that this problem is infinite dimensional.
However, since problems at different frequencies are independent from each other, in practice this problem is solved only for sufficiently many frequency points. Next, this problem is investigated only for a single frequency point and the dependence on the frequency is dropped. Moreover, for the sake of simplicity, it is assumed that $M$ is real-valued. The extension to complex valued $M$ is straightforward. It is also assumed that $X$ is a diagonal matrix. As a result the following convex feasibility problem provides sufficient conditions for robust stability of the system under consideration

\[
\text{Find } X \quad \text{subj. to } X - M'XM \preceq -\epsilon I,
\]
\[x_i \geq \epsilon, \text{ for } i = 1, \ldots, m. \tag{43}\]

for $\epsilon > 0$.

A large scale network of weakly interconnected uncertain systems can also be cast in the form presented in (40) and (41). In the case of weakly interconnected system, the system matrix relating the input to output signals, i.e. $M$, is sparse. As an example, the case of a tri-diagonal $M$-matrix is considered

\[
M = \begin{bmatrix}
g_1 & h_1 & 0 & 0 & 0 \\
f_1 & g_2 & h_2 & 0 & 0 \\
0 & f_2 & \ddots & \ddots & \ddots \\
0 & 0 & \ddots & g_{m-1} & h_{m-1} \\
0 & 0 & 0 & f_{m-1} & g_m
\end{bmatrix}. \tag{44}\]

which describes the setup where diagonal elements $m_{i,i}$ of $M$ have a feedback around them with the uncertainty $\delta_i$, and that each of the adjacent systems, i.e. the systems with index $i - 1$ and $i + 1$ provide inputs to the system with index $i$. This is the only coupling between the subsystems, see Figure 1.

Considering this setup then the LMI defined in (43) becomes banded which is a special case of Chordal sparsity patterns. Chordal sparsity patterns have been addressed within SDP by several authors, see [1], [12], [19], [13].

Using the range space conversion method presented in Section 5.1 of [19], a banded matrix can be decomposed as a sum of matrices, as illustrated in Figure 2. Then the banded matrix is negative semi-definite if and only if there exists a decomposition, as in Figure 2, such that each of the non-zero blocks in the right hand side matrices are negative semi-definite [18]. Note that the nonzero blocks marked in the matrices on the right hand side of the presentation in Figure 2 are different from the blocks marked in the left hand side matrix. Using similar procedures, the problem in (43) can be reformulated as in (2). Let

\[
q = \begin{bmatrix}
x_1 \\
x_2 \\
w_1 \\
y_1 \\
z_1 \\
x_3 \\
w_2 \\
y_2 \\
z_2 \\
\vdots \\
x_{m-2} \\
w_{m-3} \\
y_{m-3} \\
z_{m-3} \\
x_{m-1} \end{bmatrix}, \tag{45}\]

define the global variables vector. Then the reformulated problem can be written as below

\[
\text{Find } S \quad \text{subj. to } \Pi_i(s^i) \preceq -\epsilon I
\]
\[s^i = x_{\mu} \text{ for } i = 1, \ldots, m - 2, \]
\[x_j \geq \epsilon \text{ for } j = 1, \ldots, m. \tag{46}\]

where $\mathcal{J}_1 = \{1, \ldots, 5\}$, $\mathcal{J}_2 = \{3, \ldots, 9\}$, $\mathcal{J}_3 = \{7, \ldots, 13\}$, $\ldots$, $\mathcal{J}_{N-1} = \{5 + 7(N - 3) + 1, \ldots, 5 + 7(N - 3) + 7\}$ and $\mathcal{J}_N = \{5 + 7(N - 2) + 1, \ldots, 5 + 7(N - 2) + 5\}$. Accordingly, $m_1 = 5$, $m_2 = 7$, $\ldots$, $m_{N-1} = 7$ and $m_N = 5$. Also the constraints in (46) are defined as below.
The norm of primal residuals, dual residuals and dual variables vector with respect to the iteration number.

Next, the proposed methods are applied to the problem introduced (46).

VII. NUMERICAL RESULTS

In this section the achieved results from applying the proposed methods to similar problems to the one defined in Section VI is presented. These problems have the same structure as the one defined in (47) with different data describing the constraints. In the analysis of the convergence of the proposed methods the following terms are essential. The primal residuals refer to the difference between the local variables or local copies of the global variables and the global variables at each iteration, and the dual residuals refer to the difference between the global variables from two consecutive iterations [5]. Figures 3 and 4 illustrate the general behavior of the ADMM base algorithm for 60 similar problems with $m = 52$, where the data describing the constraints are generated randomly. As a result this problem is decomposed into 50 problems which is solved via 50 collaborating agents. Figure 3 illustrates how the algorithm converges by investigating the norm of primal residuals, dual residuals and dual variables vector.

Figure 4 presents how many of the constraints are still violated after each update of the global variables. As can be seen from this figure after the third iteration, the global variables satisfy the constraints of the original problem. The remaining iterations are required so that all the agents agree on the same solution, i.e. they reach a consensus. The establishment of consensus is checked by investigating either of the dual or primal residuals.

The same analysis is also performed for the Modified Nonlinear Cimmino Algorithm and the results are illustrated in Figure 5. Considering the update rule for this algorithm, the information provided by the dual and primal residuals are similar so only the evolution of the primal residuals are illustrated. Also due to the fact that the behavior of the dual variables do not affect the behavior of the algorithm their presentation is also skipped. As can be seen from this figure
the convergence to a feasible point takes at most 16 iterations which is slower than the ADMM based algorithm. However, considering the update rules for the local and global variables for this method, after convergence of the global variables to a feasible solution, the consensus is achieved immediately. It is also worth mentioning that with the Accelerated Nonlinear Cimmino Algorithm a feasible solution was achieved after more than 835 iterations.

From Figures 4 and 5 it can be inferred that the correction terms, i.e. the dual variables present in the ADMM based algorithm, speed up the convergence of the global variables to a feasible point. However, for the agents to reach a consensus, it is essential to drive these dual variables to zero. Consequently, this algorithm requires more iterations so that the agents would agree on a single solution. It is also worth mentioning that solely by observation and as can also be seen from the Figure 4, after convergence of global variables to a feasible solution, they remain feasible through the rest of the iterations. Note that if the main aim for using these proposed methods is purely computational, the ADMM based algorithm can be stopped as soon as the global variables converge to a feasible solution.

Next, the performance of the proposed algorithms are tested for different problem sizes. As can be seen from Figures 6 and 7 the size of the problem does not affect the performance of the ADMM based approach and mainly the number of required iterations to reach consensus is slightly increased. Figure 8 also presents similar behavior for the Modified Nonlinear Cimmino algorithm.

**VIII. CONCLUSION AND FUTURE WORKS**

In this paper it is shown that by exploiting the existing structure in the problem, i.e. the dependence of constraints on different overlapping sets of the optimization variables, it is possible to improve the performance of the already existing methods very considerably and also propose algorithms better suited for the specific problem at hand. This is confirmed through application of the proposed methods to the robust stability analysis of a large-scale weakly interconnected uncertain system.
A possible future direction for this work is to investigate more general frameworks in projection based algorithms such as methods utilizing Bregman distances and projections. Also considering the fact that in many cases calculating the actual projection is computationally costly it is appealing to look into subgradient based methods, [8], [21], or inexact projection methods [20]. Another important issue that should also be addressed is the analysis of the behavior of the proposed methods in the case of infeasible or inconsistent problems.

ACKNOWLEDGMENT

This research is supported by the Department of Education within the ELLIIT research program.

REFERENCES


Decomposition and Simultaneous Projection Methods for Convex Feasibility Problems with Application to Robustness Analysis of Interconnected Uncertain Systems

Sina Khoshfetrat Pakazad, Anders Hansson, Martin S. Andersen, Anders Rantzer

In this paper a specific class of convex feasibility problems are considered and tailored algorithms to solve this class of problems are introduced. First, the Nonlinear Cimmino Algorithm is reviewed. Then motivated by the special structure of the problems at hand, a modification to this method is proposed. Next, another method for solving the dual problem of the provided problem is presented. This leads to similar update rules for the variables as in the modified Nonlinear Cimmino Algorithm. Then an application for the proposed algorithms on the robust stability analysis of large scale weakly interconnected systems is presented and the performance of the proposed methods are compared.