Decomposition and Projection Methods for Distributed Robustness Analysis of Interconnected Uncertain Systems^{*}

Sina Khoshfetrat Pakazad^{*} Martin S. Andersen[†]
Anders Hansson^{*} Anders Rantzer[‡]

^{*} *Division of Automatic Control, Department of Electrical Engineering,
Linköping University, Sweden (e-mail: {sina.kh.pa,hansson}@isy.liu.se).*

[†] *Department of Informatics and Mathematical Modeling, Technical
University of Denmark (e-mail: mskan@imm.dtu.dk)*

[‡] *Department of Automatic Control, Lund University, Sweden (e-mail:
anders.rantzer@control.lth.se)*

Abstract: We consider a class of convex feasibility problems where the constraints that describe the feasible set are loosely coupled. These problems arise in robust stability analysis of large, weakly interconnected uncertain systems. To facilitate distributed implementation of robust stability analysis of such systems, we describe two algorithms based on decomposition and simultaneous projections. The first algorithm is a nonlinear variant of Cimmino's mean projection algorithm, but by taking the structure of the constraints into account, we can obtain a faster rate of convergence. The second algorithm is devised by applying the alternating direction method of multipliers to a convex minimization reformulation of the convex feasibility problem. Numerical results are then used to show that both algorithms require far less iterations than the accelerated nonlinear Cimmino algorithm.

Keywords: Robust stability, interconnected systems, convex projections, distributed computing, decomposition methods

1. INTRODUCTION

Methods for robust stability analysis of uncertain systems have been studied thoroughly, e.g., see Megretski and Rantzer (1997); Skogestad and Postlethwaite (2007); Zhou et al. (1997). One of popular methods for robust stability analysis is the so-called μ -analysis, an approximation of which solves a Convex Feasibility Problem, CFP, for each frequency on a finite grid of frequencies. For networks of interconnected uncertain systems, each CFP is a *global* feasibility problem that involves the entire network. In other words, each CFP depends on all the local models, i.e., the models associated with the subsystems in the network, as well as the network topology. In networks where all the local models are available to a central unit and the size of the problem is not computationally prohibitive, each feasibility problem can be solved efficiently in a centralized manner. However distributed or parallel methods are desirable in networks, for instance, where the interconnected system dimension is large or where the local models are either private or available to only a small number of subsystems in the network, Fang and Antsaklis (2008); Grasedyck et al. (2003); Rice and Verhaegen (2009a); VanAntwerp et al. (2001). In these references, for specific network structures, the system matrix is diagonalized and decoupled using decoupling transformations.

This decomposes the analysis and design problem and provides the possibility to produce distributed and parallel solutions. Under certain assumption on the adjacency of the network, the authors in Jönsson et al. (2007) also propose a decomposition for the analysis problem which allows them to use distributed computational algorithms for solving the analysis problem. Similarly, the authors in Kim and Braatz (2012) put forth a decomposable analysis approach which is applicable to interconnection of identical subsystems. An overview of distributed algorithms for analysis of systems governed by Partial Differential Equations, PDEs, is given in Rice and Verhaegen (2009b). In contrast to the above mentioned papers, in this paper, we investigate the possibility of decomposition of the analysis problem based on the sparsity in the interconnections. We then focus on computational algorithms to solve the analysis problem distributedly. This enables us to facilitate distributed robust stability analysis of large networks of weakly interconnected uncertain systems.

In case the interconnection among the subsystems is sparse, i.e., subsystems are weakly interconnected, it is often possible to decompose the analysis problem. Decomposition allows us to reformulate a single global feasibility constraint $x \in \mathcal{C} \subset \mathbb{R}^n$ involving m uncertain systems as a set of $N < m$ loosely coupled constraints

$$x \in C_i \equiv \{x \in \mathbb{R}^n \mid f_i(x) \preceq_{K_i} 0\}, \quad i = 1, \dots, N \quad (1)$$

where $f_i(x) \preceq_{K_i} 0$ is a linear conic inequality with respect to a convex cone K_i . We will assume that f_i

^{*} This research is supported by the Swedish department of education within the ELLIIT research program.

depends only on a small subset of variables $\{x_k | k \in J_i\}$, $J_i \subseteq \{1, \dots, n\}$. The number of constraints N is less than the number of systems m in the network, and hence the functions f_i generally involve problem data from more than one subsystem in the network. It is henceforth assumed that each f_i is described in terms of only a small number of local models such that the Euclidean projection $P_{C_i}(x)$ of x onto C_i involves just a small group of systems in the network. This assumption is generally only valid if the network is weakly interconnected.

One algorithm that is suited for distributed solution of the CFP is the nonlinear Cimmino algorithm, Censor and Elfving (1981); Cimmino (1938), which is also known as the Mean Projection Algorithm. This algorithm is a fixed-point iteration which takes as the next iterate $x^{(k+1)}$ a convex combination of the projections $P_{C_i}(x^{(k)})$, i.e.,

$$x^{(k+1)} := \sum_{i=1}^N \alpha_i P_{C_i}(x^{(k)}) \quad (2)$$

where $\sum_{i=1}^N \alpha_i = 1$ and $\alpha_1, \dots, \alpha_N > 0$. Notice that each iteration consists of two steps: a parallel projection step which is followed by a consensus step that can be solved by means of distributed weighted averaging, e.g., Lin and Boyd (2003); Nedic and Ozdaglar (2009); Rajagopalan and Shah (2011); Tsitsiklis (1984); Tsitsiklis et al. (1986). The authors in Iusem and De Pierro (1986) have proposed an accelerated variant of the nonlinear Cimmino algorithm that takes as the next iterate a convex combination of the projections of $x^{(k)}$ on only the sets for which $x^{(k)} \notin C_i$. This generally improves the rate of convergence when only a few constraints are violated. However, the nonlinear Cimmino algorithm and its accelerated variant may take unnecessarily conservative steps when the sets C_i are loosely coupled. We will consider two algorithms that can exploit this type of structure, and both algorithms are closely related to the nonlinear Cimmino in that each iteration consists of a parallel projection step and a consensus step.

The first algorithm that we consider is equivalent to the von Neumann Alternating Projection (AP) algorithm, Bregman (1965); von Neumann (1950), in a product space $\mathcal{E} = \mathbb{R}^{|J_1|} \times \dots \times \mathbb{R}^{|J_N|}$ of dimension $\sum_{i=1}^N |J_i|$. As a consequence, this algorithm converges at a linear rate under mild conditions, and its behavior is well-understood also when the CFP is infeasible, Bauschke and Borwein (1994). Using the ideas from Bertsekas and Tsitsiklis (1997); Boyd et al. (2011), we also show how this algorithm can be implemented in a distributed manner.

A CFP can also be formulated as a convex minimization problem which can be solved with distributed optimization algorithms; see e.g. Bertsekas and Tsitsiklis (1997); Boyd et al. (2011); Nedic et al. (2010); Tsitsiklis (1984). The second proposed algorithm is the Alternating Direction Method of Multipliers (ADMM) Bertsekas and Tsitsiklis (1997); Boyd et al. (2011); Gabay and Mercier (1976); Glowinski and Marroco (1975), applied to a convex minimization formulation of the CFP. Unlike AP, ADMM also makes use of dual variables, and when applied to the CFP, it is equivalent to Dykstra's alternating projection method, Bauschke and Borwein (1994); Dykstra (1983), in the product space $\mathcal{E}$. Although there exist problems for which

Dykstra's method is much slower than the classical AP algorithm, it generally outperforms the latter in practice.

Outline The paper is organized as follows. In Section 2, we present a product space formulation of CFPs with loosely coupled sets, and we describe an algorithm based on von Neumann AP method. In Section 3, a convex minimization reformulation of the CFP is considered, and we describe an algorithm based on the ADMM. Distributed implementations of both algorithms are discussed in Section 4, and in Section 5, we consider distributed solution of the robust stability analysis problem. We present numerical results in Section 6, and the paper is concluded in Section 7.

Notation We denote with $\mathbb{N}_p$ the set of positive integers $\{1, 2, \dots, p\}$. Given a set $J \subset \{1, 2, \dots, n\}$, the matrix $E_J \in \mathbb{R}^{|J| \times n}$ is the 0-1 matrix that is obtained from an identity matrix of order n by deleting the rows indexed by $\mathbb{N}_n \setminus J$. This means that $E_J x$ is a vector with the components of x that correspond to the elements in J . The distance between two sets $C_1, C_2 \subseteq \mathbb{R}^n$ is defined as $\text{dist}(C_1, C_2) = \inf_{y \in C_1, x \in C_2} \|x - y\|_2$. With $D = \text{diag}(a_1, \dots, a_n)$ we denote a diagonal matrix of order n with diagonal entries $D_{ii} = a_i$. The column space of a matrix A is denoted $\mathbb{C}(A)$. Given vectors v^k for $k = 1, \dots, N$, the column vector $(v^1, \dots, v^N)$ is all of the given vectors stacked.

2. DECOMPOSITION AND PROJECTION METHODS

2.1 Decomposition and convex feasibility in product space

Given N loosely coupled sets $C_1, \dots, C_N$, as defined in (1), we define N lower-dimensional sets

$$\bar{C}_i = \{s^i \in \mathbb{R}^{|J_i|} \mid E_{J_i}^T s^i \in C_i\}, \quad i = 1, \dots, N \quad (3)$$

such that $s^i \in \bar{C}_i$ implies $E_{J_i}^T s^i \in C_i$. With this notation, the standard form CFP can be rewritten as

$$\begin{aligned} & \text{find} && s^1, s^2, \dots, s^N, x \\ & \text{subject to} && s^i \in \bar{C}_i, \quad i = 1, \dots, N \\ & && s^i = E_{J_i} x, \quad i = 1, \dots, N \end{aligned} \quad (4)$$

where the equality constraints are the so called coupling constraints that ensure that the variables $s^1, \dots, s^N$ are consistent with one another. In other words, if the sets C_i and C_j ($i \neq j$) depend on x_k , then the k th component of $E_{J_i}^T s^i$ and $E_{J_j}^T s^j$ must be equal.

The formulation in (4) decomposes the global variable x into N coupled variables $s^1, \dots, s^N$. This allows us to rewrite the problem as a CFP with two sets

$$\begin{aligned} & \text{find} && S \\ & \text{subject to} && S \in \mathcal{C}, \quad S \in \mathcal{D} \end{aligned} \quad (5)$$

where $S = (s^1, \dots, s^N) \in \mathbb{R}^{|J_1|} \times \dots \times \mathbb{R}^{|J_N|}$, $\mathcal{C} = \bar{C}_1 \times \dots \times \bar{C}_N$, $\mathcal{D} = \{\bar{E}x \mid x \in \mathbb{R}^n\}$ and $\bar{E} = [E_{J_1}^T \dots E_{J_N}^T]^T$. The formulation in (5) can be thought of as a "compressed" product space formulation of a CFP with the constraints in (1), and it is similar to the consensus optimization problems described in (Boyd et al., 2011, Sec. 7.2).

2.2 Von Neumann's alternating projection in product space

The problem in (5) can be solved using the von Neumann's AP method. If $X = \bar{E}x$, applying the von Neumann AP algorithm to the CFP in (5) results in the update rule

$$S^{(k+1)} = P_{\mathcal{C}}(X^{(k)}) \quad (6a)$$

$$\begin{aligned} &= (P_{\bar{C}_1}(E_{J_1}x^{(k)}), \dots, P_{\bar{C}_N}(E_{J_N}x^{(k)})) \\ X^{(k+1)} &= \bar{E} \underbrace{(\bar{E}^T \bar{E})^{-1} \bar{E}^T S^{(k+1)}}_{x^{(k+1)}}, \end{aligned} \quad (6b)$$

where (6a) is the projection onto $\mathcal{C}$, and (6b) is the projection onto the column space of $\bar{E}$, Khoshfetrat Pakazad et al. (2011). The projection onto the set $\mathcal{C}$ can be computed in parallel by N computing agents, i.e., agent i computes $s^{i,(k)} = P_{\bar{C}_i}(E_{J_i}x^{(k)})$, and the second projection can be interpreted as a consensus step that can be solved via distributed averaging. In case the problem is feasible, under mild conditions, the iterates converge to a feasible solution with a linear rate, (Beck and Teboulle, 2003, Thm. 2.2). Also in case the problem is infeasible, $X^{(k)} - S^{(k)}, X^{(k)} - S^{(k+1)} \rightarrow v, \|v\|_2^2 = \text{dist}(\mathcal{C}, \mathcal{D})$, Bauschke and Borwein (1994).

3. CONVEX MINIMIZATION REFORMULATION

The CFP in (4) is equivalent to a convex minimization problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^N g_i(s^i) \\ &\text{subject to} \quad s^i = E_{J_i}x, \quad i = 1, \dots, N, \end{aligned} \quad (7)$$

with variables S and x , where

$$g_i(s^i) = \begin{cases} \infty & s^i \notin \bar{C}_i \\ 0 & s^i \in \bar{C}_i \end{cases} \quad (8)$$

is the indicator function for the set $\bar{C}_i$. In the following, ADMM is applied to the problem (7).

3.1 Solution via Alternating Direction Method of Multipliers

By applying ADMM, e.g., see Boyd et al. (2011) and (Bertsekas and Tsitsiklis, 1997, Sec. 3.4), to the problem in (7), we obtain the following update rules

$$S^{(k+1)} = P_{\mathcal{C}}(X^{(k)} - \bar{\lambda}^{(k)}) \quad (9a)$$

$$X^{(k+1)} = \bar{E} (\bar{E}^T \bar{E})^{-1} \bar{E}^T S^{(k+1)} \quad (9b)$$

$$\bar{\lambda}^{(k+1)} = \bar{\lambda}^{(k)} + (S^{(k+1)} - X^{(k+1)}), \quad (9c)$$

where $\bar{\lambda} = (\bar{\lambda}^1, \dots, \bar{\lambda}^N)$, Khoshfetrat Pakazad et al. (2011). Note that similar to the von Neumann's AP method, the projection onto $\mathcal{C}$ can be computed in parallel. We remark that this algorithm is a special case of the algorithm developed in (Boyd et al., 2011, Sec. 7.2) for consensus optimization. We also note that this algorithm is equivalent to Dykstra's alternating projection method for two sets where one of the sets is affine, Bauschke and Borwein (1994). Moreover, it is possible to detect infeasibility in the same way that we can detect infeasibility for the AP method, i.e., $X^{(k)} - S^{(k)} \rightarrow v$ where $\|v\|_2^2 = \text{dist}(\mathcal{C}, \mathcal{D})$, Bauschke and Borwein (1994). Note that, unlike the AP method, the iterations in (9) does not necessarily converge with a linear rate.

Algorithm 1 ADMM based algorithm

- 1: Given $x^{(0)}, \bar{\lambda}^{(0)} = 0$, for all $i = 1, \dots, N$, each agent i should
 - 2: **for** $k = 0, 1, \dots$ **do**
 - 3: $s^{i,(k+1)} \leftarrow P_{\bar{C}_i}(x_{J_i}^{(k)} - \bar{\lambda}_i^{(k)})$.
 - 4: Communicate with all agents r belonging to $\text{Ne}(i)$.
 - 5: **for all** $j \in J_i$ **do**
 - 6: $x_j^{(k+1)} = \frac{1}{|J_j|} \sum_{q \in J_j} (E_{J_q}^T s^{q,(k+1)})_j$.
 - 7: **end for**
 - 8: $\bar{\lambda}_i^{(k+1)} \leftarrow \bar{\lambda}_i^{(k)} + (s^{i,(k+1)} - x_{J_i}^{(k+1)})$
 - 9: **end for**
-

4. DISTRIBUTED IMPLEMENTATION

In this section, we describe how the algorithms can be implemented in a distributed manner using techniques that are similar to those described in Boyd et al. (2011) and (Bertsekas and Tsitsiklis, 1997, Sec. 3.4). Specifically, the parallel projection steps in the algorithms expressed by the update rules in (6) and (9) are amenable to distributed implementation. We will henceforth assume that a network of N computing agents is available.

Let $\mathcal{I}_i = \{k \mid i \in J_k\} \subseteq \mathbb{N}_n$ denote the set of constraints that depend on x_i . Then it is easy to verify that $\bar{E}^T \bar{E} = \text{diag}(|\mathcal{I}_1|, \dots, |\mathcal{I}_n|)$, and consequently, the j th component of x in the update rules (6b) and (9b) is of the form

$$x_j^{(k+1)} = \frac{1}{|J_j|} \sum_{q \in J_j} (E_{J_q}^T s^{q,(k+1)})_j. \quad (10)$$

In other words, the agents with indices in the set J_j must solve a distributed averaging problem to compute $x_j^{(k+1)}$. Let $x_{J_i} = E_{J_i}x$. Since the set $\bar{C}_i$ associated with agent i involves the variables with indices in J_i , agent i should update x_{J_i} by performing the update in (10) for all $j \in J_i$. This requires agent i to communicate with all agents in the set $\text{Ne}(i) = \{j \mid J_i \cap J_j \neq \emptyset\}$, which we call the neighbors of agent i . Each agent $j \in \text{Ne}(i)$ shares one or more variables with agent i . Distributed variant of the algorithm presented in the Section 3.1 is summarized in Algorithm 1. Note that in Algorithm 1, if we neglect $\bar{\lambda}$ and its corresponding update rule, we arrive at the distributed variant of the algorithm presented in Section 2.2. We refer to this algorithm as Algorithm 2.

4.1 Feasibility Detection

For strictly feasible problems, algorithms 1 and 2 converge to a feasible solution. We now discuss different techniques for checking feasibility.

Global Feasibility Test Perhaps the easiest way to check feasibility is to directly check the feasibility of $x^{(k)}$ with respect to all the constraints. This can be accomplished by explicitly forming $x^{(k)}$. This requires a centralized unit that receives the local variables from the individual agents.

Local Feasibility Test It is also possible to check feasibility locally. Instead of sending the local variables to a central unit, each agent declares its feasibility status with respect to its local constraint. This type of feasibility detection is based on the following lemmas.

Lemma 1. If $x_{J_i}^{(k)} \in \bar{C}_i$, for all $i = 1, \dots, N$ then using these vectors a feasible solution, x , can be constructed for the original problem, i.e., $x \in \bigcap_{i=1}^N C_i$.

Proof. See Khoshfetrat Pakazad et al. (2011).

Lemma 2. If $\|X^{(k+1)} - X^{(k)}\|_2^2 = 0$ and $\|S^{(k+1)} - X^{(k+1)}\|_2^2 = 0$, then a feasible solution, x , can be constructed for the original problem, i.e., $x \in \bigcap_{i=1}^N C_i$.

Proof. See Khoshfetrat Pakazad et al. (2011).

Remark 1. For Algorithm 2, the conditions in lemmas 1 and 2 are equivalent. However, this is not the case for Algorithm 1. For this algorithm, satisfaction of the conditions in Lemma 2 implies the conditions in Lemma 1.

Remark 2. Feasibility detection using Lemma 1, requires additional computations for Algorithm 1. These computations include local feasibility check of the iterates x_{J_i} . Note that this check does not incur any additional cost for Algorithm 2.

Infeasibility Detection Recall from sections 2.2 and 3.1 that if the CFP is infeasible, then the sequence $\|S^{(k)} - X^{(k)}\|_2^2 = \sum_{i=1}^N \|s^{i,(k)} - x_{J_i}^{(k)}\|_2^2$ will converge to a nonzero constant $\|v\|_2^2 = \text{dist}(\mathcal{C}, \mathcal{D})$. Therefore, in practice, it is possible to detect infeasible problems by limiting the number of iterations. In other words, in case there exists an agent i for which $\|s^{i,(k)} - x_{J_i}^{(k)}\|_2^2$ is not sufficiently small after the maximum number of iterations has been reached, the problem is deemed to be infeasible.

5. ROBUST STABILITY ANALYSIS

Robust stability of uncertain large-scale weakly interconnected uncertain systems with structured uncertainty can be analyzed through the so-called μ -analysis framework Fan et al. (1991). This leads to a CFP which is equivalent to a semidefinite program (SDP) involving the system matrix. In this section, we show, through an example, how the structure in the interconnections can be used to decompose the problem. Consider the system description $Y(s) = M(s)U(s)$, where $M(s)$ is a $m \times m$ transfer function matrix, and let $U(s) = \Delta Y(s)$, where $\Delta = \text{diag}(\delta_i)$, with $\delta_i \in \mathbb{R}$, $|\delta_i| \leq 1$ for $i = 1, \dots, m$, representing the uncertainties in the system. The system is said to be robustly stable if there exists a diagonal positive definite $\mathbf{X}(\omega)$ and $0 < \mu < 1$ such that

$$M(j\omega)^* \mathbf{X}(\omega) M(j\omega) - \mu^2 \mathbf{X}(\omega) \prec 0, \quad (11)$$

for all $\omega \in \mathbb{R}$. Note that this problem is infinite dimensional, and in practice, it is often solved approximately by discretizing the frequency variable.

In the following, we investigate only a single frequency point such that the dependence on the frequency is dropped. Moreover, for the sake of simplicity, we assume that M is real-valued. The extension to complex valued M is straightforward. As a result, feasibility of the following CFP is a sufficient condition for robust stability of the system

$$\begin{aligned} & \text{find} \quad \mathbf{X} \\ & \text{subject to} \quad M' \mathbf{X} M - \mathbf{X} \preceq -\varepsilon I \\ & \quad \quad \quad x_i \geq \varepsilon_i, \quad i = 1, \dots, m \end{aligned} \quad (12)$$

for $\varepsilon_i, \varepsilon > 0$ and where x_i are the diagonal elements of $\mathbf{X}$. In the case of weakly interconnected system, the system

Fig. 1. An example of a weakly interconnected uncertain system.

Fig. 2. A positive semidefinite band matrix of order n with half-bandwidth p can be decomposed as a sum of $n - p$ positive semidefinite matrices.

matrix M that relates the input and output signals can be sparse. As an example, we consider a chain of systems which leads to a tri-diagonal system matrix

$$M = \begin{bmatrix} g_1 & h_1 & 0 & 0 & 0 \\ f_2 & g_2 & h_2 & 0 & 0 \\ 0 & f_3 & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & g_{m-1} & h_{m-1} \\ 0 & 0 & 0 & f_m & g_m \end{bmatrix}. \quad (13)$$

This system matrix is obtained if the input-output relation for the underlying systems are given by

$$\begin{aligned} p_1 &= g_1 q_1 + h_1 z_2, & p_m &= g_m q_m + f_m z_{m-1} \\ z_1 &= q_1, & z_m &= q_m \\ q_1 &= \delta_1 p_1, & q_m &= \delta_m p_m \end{aligned}$$

and

$$\begin{aligned} p_i &= g_i q_i + f_i z_{i-1} + h_i z_{i+1} \\ z_i &= q_i \\ q_i &= \delta_i p_i, \end{aligned}$$

for $i = 2, \dots, m - 1$. The tri-diagonal structure in the system matrix implies that the LMI defined in (12) becomes banded. This is a special case of a so-called chordal sparsity pattern, and these have been exploited in SDP by several authors; see Andersen et al. (2010); Fujisawa et al. (2009); Fukuda et al. (2000); Kim et al. (2010). Positive semidefinite matrices with chordal sparsity patterns can be decomposed into a sum of matrices that are positive semidefinite (Kakimura, 2010; Kim et al., 2010, Sec. 5.1). Notice that general sparse interconnection of uncertain systems will not yield a sparsity pattern in (12). We have chosen a simple example to demonstrate what can be done if (12) happens to be sparse in order to not clutter the presentation. The general case can be handled similarly as is described for a centralized solution in Andersen et al. (2012). A positive semidefinite band matrix can be decomposed into a sum of positive semidefinite matrices, as illustrated in Figure 2. Note that the nonzero blocks marked in the matrices on the right-hand side in Figure 2 are structurally equivalent to the block in the left-hand side, but the numerical values are generally different. Using this decomposition technique, the problem in (12) can be reformulated as in (5). By introducing auxiliary variables $w, y, z \in \mathbb{R}^{m-3}$ and letting $q = (x, w, y, z)$, the five-diagonal matrix $M' X M - X$ can be decomposed as

$$f_1(q) = \begin{bmatrix} g_1 & f_2 \\ h_1 & g_2 \\ 0 & h_2 \end{bmatrix} \begin{bmatrix} x_1 & 0 \\ 0 & x_2 \end{bmatrix} \begin{bmatrix} g_1 & f_2 \\ h_1 & g_2 \\ 0 & h_2 \end{bmatrix}' - \begin{bmatrix} x_1 & 0 & 0 \\ 0 & x_2 - w_1 & -y_1 \\ 0 & -y_1 & -z_1 \end{bmatrix} \quad (14a)$$

$$f_{m-1}(q) = \begin{bmatrix} f_{m-1} & 0 \\ g_{m-1} & f_m \\ h_{m-1} & g_m \end{bmatrix} \begin{bmatrix} x_{m-1} & 0 \\ 0 & x_m \end{bmatrix} \begin{bmatrix} f_{m-1} & 0 \\ g_{m-1} & f_m \\ h_{m-1} & g_m \end{bmatrix}' - \begin{bmatrix} w_{m-3} & y_{m-3} & 0 \\ y_{m-3} & w_{m-3} + x_{m-1} & 0 \\ 0 & 0 & x_m \end{bmatrix} \quad (14b)$$

and

$$f_i(q) = \begin{bmatrix} f_{i+1} \\ g_{i+1} \\ h_{i+1} \end{bmatrix} \begin{bmatrix} x_{i+1} \end{bmatrix} \begin{bmatrix} f_{i+1} \\ g_{i+1} \\ h_{i+1} \end{bmatrix}' - \begin{bmatrix} w_{i-1} & y_{i-1} & 0 \\ y_{i-1} & z_{i-1} + x_{i+1} - w_i & -y_i \\ 0 & -y_i & -z_i \end{bmatrix} \quad (14c)$$

for $i = 2, \dots, m-3$. This approach for decomposing the problem is a special case of the so-called range space decomposition method, which is used for handling chordal SDPs, Kim et al. (2010). Notice that (14a) and (14b) depend on data from two subsystems whereas (14c) depends on data from only one subsystem. The right-hand side of the LMI in (12) can be decomposed in a similar manner

$$F_1 = \begin{bmatrix} \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix}, F_{m-2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix}, F_i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix}, \quad i = 2, \dots, m-3.$$

With this decomposition, we can reformulate the LMI in (12) as a set of $m-2$ LMIs, $q \in C_i = \{q \mid f_i(q) \preceq -F_i\}$, $i = 1, \dots, m-2$, or equivalently, as the constraints

$$s^i \in \bar{C}_i = \{E_{J_i} q \mid f_i(q) \preceq -F_i\} \subseteq \mathbb{R}^{|J_i|}, \quad s^i = E_{J_i} q, \quad (15)$$

for $i = 1, \dots, m-2$. Here J_i is the set of indices of the entries of q that are required to evaluate f_i . Notice that (15) is of the form (4). The CFP defined by the constraints in (15) can be solved over a network of $m-2$ agents.

6. NUMERICAL RESULTS

In this section, we apply algorithms 1 and 2 to a family of random problems involving a chain of uncertain systems. These problems have the same band structure as described in Section 5. We use the local feasibility detection method introduced in Section 4.1. Note that in order to avoid numerical problems and unnecessarily many projections, the projections are performed for slightly tighter bounds than those used for feasibility checks of the local iterates. This setup is the same for all the experiments presented in this section. Figures 3 and 4 show the behavior of Algorithm 1 for 50 randomly generated problems with $m = 52$. These problems decompose into 50 subproblems which 50 agents solve collaboratively. Figure 3 shows the number of agents that are locally infeasible at each iteration. Figure 4, illustrates the norm of iterate residuals used in Lemma 2. As can be seen from these figures, the feasibility detection based on lemmas 1 and 2 requires at most 7 and 27 iterations to detect convergence to a feasible solution, respectively. We performed the same experiment with Algorithm 2, and the results are illustrated in Figures 5 and 6. Figure 5 demonstrates that the global variables satisfy the constraints of the original problem after at most 22 iterations. This can be used to detect convergence to a feasible solution by checking the satisfaction of the conditions in Lemma 1. Considering Remark 1 the same performance should also be observed by checking the conditions in Lemma 2. This is confirmed by the results illustrated in Figure 4. In our experiments, Algorithm 1 is faster when feasibility detection is based on the conditions in Lemma 1, and Algorithm 2 is faster when the condition in Lemma 2 is used for feasibility detection. It is also worth mentioning that with the accelerated

Fig. 3. Algorithm 1: Number of violated constraints in the CFP after each update of the global variables. This figure illustrates the results for 50 randomly generated problems.

Fig. 4. Algorithm 1: $\|X^{(k)} - X^{(k-1)}\|_2^2$ and $\|S^{(k)} - X^{(k)}\|_2^2$ with respect to the iteration number. This figure illustrates the results for 50 randomly generated problems.

Fig. 5. Algorithm 2: Number of violated constraints in the CFP after each update of the global variables. This figure illustrates the results for 50 randomly generated problems.

nonlinear Cimmino algorithm, more than 1656 iterations were needed to obtain a feasible solution.

7. CONCLUSION

In this paper, we have shown that it is possible to solve CPFs with loosely coupled constraints efficiently in a distributed manner. The algorithms used for this purpose are based on von Neumann's AP method and ADMM. The former enjoys the linear convergence rate when applied to strictly feasible problems. The latter generally outperforms the AP method in practice in terms of the number of iterations required to obtain a feasible solution. Both methods can detect infeasible problems. For structured problems that arise in robust stability analysis of a large-scale weakly interconnected uncertain systems, our numerical results show that both algorithms outperform the classical projection-based algorithms. As future research direction, we will consider a more rigorous study of range space decomposition for general sparse network interconnections.

Fig. 6. Algorithm 2: $\|X^{(k)} - X^{(k-1)}\|_2^2$ and $\|S^{(k)} - X^{(k)}\|_2^2$ with respect to the iteration number. This figure illustrates the results for 50 randomly generated problems.

We will also investigate the possibility of employing other splitting methods for solving the decomposed problem in a distributed manner.

REFERENCES

- Andersen, M.S., Dahl, J., and Vandenberghe, L. (2010). Implementation of nonsymmetric interior-point methods for linear optimization over sparse matrix cones. *Mathematical Programming Computation*, 2, 167–201.
- Andersen, M.S., Hansson, A., Pakazad, S.K., and Rantzer, A. (2012). Distributed robust stability analysis of interconnected uncertain systems. In *Proceedings of the 51st IEEE Conference on Decision and Control*.
- Bauschke, H.H. and Borwein, J.M. (1994). Dykstra’s alternating projection algorithm for two sets. *Journal of Approximation Theory*, 79(3), 418–443.
- Beck, A. and Teboulle, M. (2003). Convergence rate analysis and error bounds for projection algorithms in convex feasibility problems. *Optimization Methods and Software*, 18(4), 377–394.
- Bertsekas, D.P. and Tsitsiklis, J.N. (1997). *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), 1–122.
- Bregman, L. (1965). The method of successive projection for finding a common point of convex sets. *Soviet Math. Dokl.*, 162, 487–490.
- Censor, Y. and Elfving, T. (1981). A nonlinear Cimmino algorithm. Technical Report Report 33, Department of Mathematics, University of Haifa.
- Cimmino, G. (1938). Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari. *La Ricerca Scientifica Roma*, 1, 326–333.
- Dykstra, R.L. (1983). An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78(384), 837–842.
- Fan, M.K.H., Tits, A.L., and Doyle, J.C. (1991). Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. *IEEE Trans. Autom. Control*, 36(1), 25–38.
- Fang, H. and Antsaklis, P.J. (2008). Distributed control with integral quadratic constraints. In *Proceedings of the 17th IFAC World Congress*, volume 17, 574–580.
- Fujisawa, K., Kim, S., Kojima, M., Okamoto, Y., and Yamashita, M. (2009). User’s manual for SparseCoLo: Conversion methods for SPARSE COnnic-form Linear Optimization. Technical report, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo.
- Fukuda, M., Kojima, M., Murota, K., and Nakata, K. (2000). Exploiting sparsity in semidefinite programming via matrix completion i: General framework. *SIAM Journal on Optimization*, 11, 647–674.
- Gabay, D. and Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1), 17–40.
- Glowinski, R. and Marroco, A. (1975). Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de dirichlet non linéaires. *Revue française d’automatique, informatique, recherche opérationnelle*, 9(2), 41–76.
- Grasedyck, L., Hackbusch, W., and Khoromskij, B.N. (2003). Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices. *Computing Vienna/New York*, 70(2), 121–165.
- Iusem, A.N. and De Pierro, A.R. (1986). Convergence results for an accelerated nonlinear Cimmino algorithm. *Numerische Mathematik*, 49, 367–378.
- Jönsson, U., Kao, C., and Fujioka, H. (2007). A popov criterion for networked systems. *Systems & Control Letters*, 56(9–10), 603–610.
- Kakimura, N. (2010). A direct proof for the matrix decomposition of chordal-structured positive semidefinite matrices. *Linear Algebra and its Applications*, 433(4), 819–823.
- Khoshfetrat Pakazad, S., S. Andersen, M., Hansson, A., and Rantzer, A. (2011). Decomposition and projection methods for distributed robustness analysis of interconnected uncertain systems. Technical Report LiTH-ISY-R-3033. URL <http://www.control.isy.liu.se/research/reports/2011/3033.pdf>.
- Kim, K.K. and Braatz, R.D. (2012). On the robustness of interconnected or networked uncertain linear multi-agent systems. *20th International Symposium on Mathematical Theory of Networks and Systems*.
- Kim, S., Kojima, M., Mevissen, M., and Yamashita, M. (2010). Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion. *Mathematical Programming*, 1–36.
- Lin, X. and Boyd, S. (2003). Fast linear iterations for distributed averaging. In *Proceedings 42nd IEEE Conference on Decision and Control, 2003.*, volume 5, 4997–5002 Vol.5.
- Megretski, A. and Rantzer, A. (1997). System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, 42(6), 819–830.
- Nedic, A. and Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control*, 54(1), 48–61.
- Nedic, A., Ozdaglar, A., and Parrilo, P. (2010). Constrained consensus and optimization in multi-agent networks. *IEEE Trans. Autom. Control*, 55(4), 922–938.
- Rajagopalan, S. and Shah, D. (2011). Distributed averaging in dynamic networks. *IEEE Journal of Selected Topics in Signal Processing*, 5(4), 845–854.
- Rice, J. and Verhaegen, M. (2009a). Robust control of distributed systems with sequentially semi-separable structure. In *Proceedings of the European Control Conference*.
- Rice, J. and Verhaegen, M. (2009b). Distributed control: A sequentially semi-separable approach for spatially heterogeneous linear systems. *IEEE Transactions on Automatic Control*, 54(6), 1270–1283.
- Skogestad, S. and Postlethwaite, I. (2007). *Multivariable feedback control*. Wiley.
- Tsitsiklis, J.N. (1984). *Problems in decentralized decision making and computation*. Ph.D. thesis, MIT.
- Tsitsiklis, J., Bertsekas, D., and Athans, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9), 803–812.
- VanAntwerp, J.G., Featherstone, A.P., and Braatz, R.D. (2001). Robust cross-directional control of large scale sheet and film processes. *Journal of Process Control*, 11(2), 149–177.
- von Neumann, J. (1950). *The Geometry of Orthogonal Spaces*, volume 2 of *Functional Operators*. Princeton University Press.
- Zhou, K., Doyle, J.C., and Glover, K. (1997). *Robust and optimal control*. Prentice Hall.