A New Lagrangian Dual Global Optimization Algorithm for Solving Bilinear Matrix Inequalities

H.D. Tuan[*], **P. Apkarian**[†] and **Y. Nakashima**[‡]

Abstract

A new global optimization algorithm for solving Bilinear Matrix Inequalities (BMI) problems is developed. It is based on a dual Lagrange formulation for computing lower bounds that are used in a branching procedure to eliminate partition sets in the space of complicating variables. The advantage of the proposed method is twofold. First, lower bound computations reduce to solving easily tractable Linear Matrix Inequality (LMI) problems. Secondly, the lower bounding procedure guarantees global convergence of the algorithm when combined with an exhaustive partitioning of the space of complicating variables. A rigorous proof of this fact is provided. Another important feature is that the branching phase takes place in the space of complicating variables only, hence limiting the overall cost of the algorithm. Also, an important point in the method is that separated LMI constraints are encapsulated into an augmented BMI for improving the lower bound computations. Applications of the algorithm to robust structure/controller design are considered.

1 Introduction

A general BMI problem can be formulated as follows

$$\min \langle c,x\rangle + \langle d,y\rangle \quad : \quad x \in X := [p^0, q^0] \subset R^n, \tag{1}$$

$$G_0 + \sum_{j=1}^{m} y_j G_j \leq 0, \tag{2}$$

$$L_0 + \sum_{i=1}^{n} x_i L_{i0} + \sum_{j=1}^{m} y_j L_{0j} + \sum_{i=1}^{n}\sum_{j=1}^{m} x_i y_j L_{ij} < 0, \tag{3}$$

where $G_0, G_j, L_0, L_{0i}, L_{j0}, L_{ij}$ are symmetric matrices of appropriate sizes, $p^0 \in R^n, q^0 \in R^n$ and x and y are the decision variables.

It is widely recognized that such BMI problems arises frequently in the study of control problems and thus they constitute a very important class of optimization problems with a vast array of potential applications. Unfortunately, in contrast to linear matrix inequality (LMI) problems which are convex and can be solved by polynomial-time interior-point

[*]Department of Control and Information, Toyota Institute of Technology, Hisakata 2-12-1, Tenpaku, Nagoya 468-8511, JAPAN; Email: tuan@toyota-ti.ac.jp

[†]ONERA-CERT, 2 av. Edouard Belin, 31055 Toulouse, FRANCE - Email: apkarian@cert.fr

[‡]FANUC Ltd., Oshino-mura, Yamanashi 401-0597, JAPAN; Email: nakasima@mbc.sphere.ne.jp

methods, BMI problems (1)-(3) are nonconvex and known to be NP-hard [19]. The hardness comes from the BMI constraint (3).

Alternatively, the BMI (3) can be rewritten as

$$L_0 + \sum_{i=1}^{n} x_i L_{i0} + \sum_{j=1}^{m} y_j L_{0j} + \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} L_{ij} < 0, \tag{4}$$

$$w_{ij} = x_i y_j,\ j = 1, 2, ..., n;\ j = 1, 2, ..., m, \tag{5}$$

where (4) is now a convex LMI constraint in (x, y, w) and (5) is a nonconvex indefinite quadratic constraint. Therefore techniques which can efficiently handle the quadratic constraint (5) are of special interest for solving (1)-(3).

Obviously, the LMI constraint (2) includes as a particular case the box constraint

$$y \in [r, s] \subset R^m \tag{6}$$

and other linear constraints for y.

Goh et al. [10] proposed a simple branch and bound (BB) algorithm for problem (1), (3), (6) with branching in the (x, y)-space of all variables hence of dimension $n + m$, and bounding based upon a relaxation of the nonconvex constraint (5) for $(x_i, y_j) \in [p_i, q_i] \times [r_j, s_j]$ by a coarse approximated constraint

$$w_{ij} \in [p_i r_j, q_i s_j]. \tag{7}$$

It can be assumed without loss of generality that $p_i \geq 0$ and $r_j \geq 0$. Note that the nonconvex quadratic constraint (5) has been well studied in global optimization and it is classically known that a much tighter relaxed constraint than (7) for (5) (see e.g. [15, 2, 12, 18]) is given as

$$\max\{r_j x_i + p_i y_j - p_i r_j, s_j x_i + q_i y_j - q_i s_j\} \leq w_{ij} \leq \min\{s_j x_i + p_i y_j - p_i s_j, r_j x_i + q_i y_j - q_i r_j\} \tag{8}$$

In [3, 21] it is further shown that functions max and min in (8) are in fact convex and concave envelopes of the bilinear function $x_i y_j$ on the rectangles $[p_i, q_i] \times [r_j, s_j]$. The constraint (8) is nothing else than the following constraint

$$(x_i, y_j, w_{ij}) \in \text{convex hull}\{(p_i, r_j, p_i r_j), (p_i, s_j, p_i s_j), (q_i, r_j, q_i r_j), (q_i, s_j, q_i s_j)\} \tag{9}$$

which is used in [11] to compute a lower bound of (1)-(3). However, the form (8) which is an explicit description of (9), is more convenient for computational implementation. Both (8) and (9) have been essentially established in [15].

On the other hand, the most fundamental convexity property of the BMI problem (1)-(3) is that it becomes convex when either y or x is fixed. Therefore, regarding only x (when $n \leq m$) or y (when $m < n$) as "complicating variables", BB algorithms with branching only on the space of complicating variables (instead of the whole space of all variables as done in [10, 11]) have been proposed in [20, 5, 6]. Such branching techniques are really important as the global search procedure is restricted to operate in the low dimensional space of dimension $\min\{n, m\}$ compared with the dimension $n + m$ of the space of all variables and thus bring to a more reasonable extent the difficulties of "the curse of dimensionality" inherent to most nonconvex problems. Let us remember here that in general the number of iterations of a BB algorithm is an exponential function of the branching space dimension. Therefore, in cases when $\min\{n,m\}$ is not excessively large but $\max\{n,m\}$ may be large, the algorithms of [20, 5, 6] can work well whereas those of [10, 11] are still handicapped by branching operations performed in a space of dimension $n+m$. An improved algorithm to that of [10, 11] has been proposed in [13]. It requires computing a local optimal solution of problem (1)-(3) at every iteration. Note that even this local optimization problem is itself NP-hard and therefore time consuming. The algorithm of [20] uses the relaxed constraint (8) for obtaining a lower bound and the d.c. (difference of convex functions) structure (see e.g [14, 21]) of the nonconvex constraint (5) to perform the decomposition. In [5, 6], the authors proposed lower bounds based upon $2^{\min\{n,m\}}$ LMI *relaxed* convex problems of the Lagrange dual problem of (1)-(3) by fixing the complicating variables followed by a generalized Bender decomposition. While the convergence (to a global optimal solution) of algorithm [20] has been shown, it remains a very delicate issue in the algorithm of [5, 6]. This again demonstrates that the convergence of global optimization algorithms is a difficult question that deserves careful attention since contrarily to what is often believed, not all relaxations produce globally convergent BB algorithms.

In this paper, we first convert the BMI problem (1)-(3) into a BMI problem with special structure. Exploiting this structure, the Lagrange dual problem is shown to be an LMI problem and is used to compute a new lower bound. This lower bound is then encapsulated in a general rectangular partitioning scheme to constitute the proposed global algorithm. With the proposed decomposition technique, we provide a rigorous proof of global convergence of our algorithm.

The organization of the paper is as follows. Section 2 investigates the dual BMI problem which is used in Section 3 to compute cheap lower bounds. The new BB algorithm is also established in this section and its global convergence is proved in Section 4. Finally, an application of the proposed algorithm to a robust structure/control design problem is given in Section 5. The notation in the paper is standard. $\langle .,. \rangle$ is the scalar product in a finite dimensional linear space. $A < 0$ ($A \leq 0$, resp.) for a symmetric matrix A, means that A is negative definite (semi-definite, resp.). Accordingly, $A < B$ for symmetric matrices A and B means that $A - B$ is negative definite. Also, $Tr(A)$ stands for the trace of A. To save a description space we shall use

$$\begin{bmatrix} A \\ B \end{bmatrix}_d \quad \text{for} \quad \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}.$$

Finally, $x \geq p$ for vectors in R^n indicates component-wise inequalities $x_i \geq p_i$ with $i = 1, 2, ..., n$.

2 The BMI Problem and its dual

Without loss of generality, we assume throughout that $n \leq m$ in (1)-(3). The generalized Bender's decomposition method of [5, 6] is based on the following projection method. Let $f(x)$ be the optimal value of (1)-(3) with $x \in X$ fixed. Then (1)-(3) can be recast as the projected problem

$$\min \ f(x) : \ x \in X,$$

which is a minimization of the nonconvex function $f(x)$ over the convex box constraint $x \in X$. Thus a BB algorithm involves at each iteration the lower bound computation for

$$\min\ f(x):\ x \in M, \tag{10}$$

where M is some rectangle $M = [p, q] \subset X$. For every fixed $x \in M$, a lower bound of $f(x)$ over M is the optimal value of the dual Lagrange problem of (1)-(3) (with x fixed) which reduces to an LMI problem. Therefore, in this context, infinitely many LMI dual Lagrange problems are required to compute a lower bound of (10). In [5, 6], 2^n *relaxed* problems are used to replace this infinite number of dual Lagrange problems. Thus a lower bounding step requires the solution of 2^n LMI problems. Such relaxations for the lower bound computation seems also to cause the main difficulty for the convergence of the corresponding BB algorithm (see Remark in Section 3 below). Here, we shall present a new technique using a special Lagrange dual problem which both bypasses the projection and numerous relaxation operations but also requires only one LMI problem to determine a new lower bound.

First, by translation if necessary, we can assume that $y \in R^m_+$. For fixed x problem (1)-(3) is a convex LMI problem, so x can be considered as the complicating variable[14, 21]. Based on this observation, our target is to develop a BB method for solving (15) with branching performed upon x. For this purpose, for $M = [p, q] \subset X$, we are interested in the subproblem

$$\min\ \langle c, x\rangle + \langle d, y\rangle \quad : \quad \begin{array}{l} x \in M = [p, q], \\ (2) - (3) \end{array}$$

which can be rewritten as

$$\min\ t \quad : \quad x \in M = [p, q], \tag{11}$$

$$A_{00}(x) + \sum_{j=1}^{m} y_j A_{j0}(x) \leq t Q_{00} \tag{12}$$

with

$$A_{00}(x) = \begin{bmatrix} G_0 \\ L_0 + \sum_{i=1}^{n} x_i L_{i0} \\ \langle x, c\rangle \end{bmatrix}_d, \quad A_{j0}(x) = \begin{bmatrix} G_j \\ L_{0j} + \sum_{i=1}^{n} x_i L_{ij} \\ d_j \end{bmatrix}_d, \quad j = 1, ..., n;\ Q_{00} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_d. \tag{13}$$

Note that the LMI structure of (2) is hidden in (12). It is possible to exploit this substructure and the box constraint $x \in M$ to augment the BMI constraint by additional BMI constraints which leaves the optimal value of the program (11)-(12) unchanged but leads to improved lower bounds. This comes from the fact that lower bounds are trivially increased when constraints are added. This procedure appears to have a significant impact on the efficiency of the global technique which is described in the sequel. The following proposition whose proof is immediate, clarifies this point.

Proposition 1 *LMI (2) with the box constraint (11) in force infer the additional set of BMI constraints*

$$
\begin{array}{rcl}
(x_i - p_i)G_0 + \sum_{j=1}^{n} y_j(x_i - p_i)G_j & \le & 0; \\
(q_i - x_i)G_0 + \sum_{j=1}^{n} y_j(q_i - x_i)G_j & \le & 0, \\
 & & i = 1, 2, ..., n.
\end{array} \tag{14}
$$

Thus we can convert (11)-(12) into the following BMI problem

$$
f_M^* := \min\ t:\ A_0(x,p,q) + \sum_{j=1}^{m} y_j A_j(x,p,q) \le tQ,\ y \ge 0,\ x \in M, \tag{15}
$$

where

$$
A_j(x,p,q) = \begin{bmatrix} A_{j0}(x) \\ A_{j1}(x,p,q) \end{bmatrix}_d,\ Q = \begin{bmatrix} Q_{00} \\ Q_{01} \end{bmatrix}_d,\ Q_{01} = 0,
$$

$$
A_{j1}(x,p,q) = \begin{bmatrix} (x_1 - p_1)G_j \\ (q_1 - x_1)G_j \\ .... \\ (x_n - p_n)G_j \\ (q_n - x_n)G_j \end{bmatrix}_d,\ j = 0, 1, ..., n \tag{16}
$$

The Lagrange dual of (15) is then described as

$$
\psi_M^* := \max_{Z \ge 0} \min_{t \in R, y \ge 0, x \in M} \{t + \mathrm{Tr}[Z(A_0(x,p,q) + \sum_{j=1}^{m} y_j A_j(x,p,q) - tQ)]\} \tag{17}
$$

The following alternate formulation turns out to be useful throughout this section.

Lemma 1 *The Lagrange dual problem (17) can be equivalently formulated as follows.*

$$
\max_{Z \ge 0}\{\min_{x \in M} Tr(ZA_0(x,p,q)) :\ Tr(ZQ) = 1,\ Tr(ZA_j(x,p,q)) \ge 0,\ \forall x \in M,\ j = 1, 2, ..., m\} \tag{18}
$$

Proof: For every $Z \ge 0$, we have

$$
\min\{t + \mathrm{Tr}[Z(A_0(x,p,q) + \sum_{j=1}^{m} y_j A_j(x,p,q) - tQ)] :\ t \in R,\ y \ge 0,\ x \in M\} =
$$

$$
\min\{t - t\mathrm{Tr}(ZQ) + \mathrm{Tr}(ZA_0(x,p,q)) + \sum_{j=1}^{m} y_j \mathrm{Tr}(ZA_j(x,p,q)) :\ t \in R,\ y \ge 0,\ x \in M\} =
$$

$$
\min_{t \in R} t(1 - \mathrm{Tr}(ZQ)) + \min_{y \ge 0,\ x \in M}[\mathrm{Tr}(ZA_0(x,p,q)) + \sum_{j=1}^{m} y_j \mathrm{Tr}(ZA_j(x,p,q))]. \tag{19}
$$

We note also that

$$
\min_{t \in R} t(1 - \mathrm{Tr}(ZQ)) = \begin{cases} 0 & \text{if } \mathrm{Tr}(ZQ) = 1 \\ -\infty & \text{otherwise}, \end{cases}
$$

while

$$\min_{y\geq 0}\min_{x\in M}[\mathrm{Tr}(ZA_0(x,p,q))+\sum_{j=1}^{m} y_j \mathrm{Tr}(ZA_j(x,p,q))] =$$

$$\min_{x\in M}\min_{y\geq 0}[\mathrm{Tr}(ZA_0(x,p,q))+\sum_{j=1}^{m} y_j \mathrm{Tr}(ZA_j(x,p,q))] =$$

$$\begin{cases} \min_{x\in M} \mathrm{Tr}(ZA_0(x,p,q)) & \text{if } \mathrm{Tr}(ZA_j(x,p,q)) \geq 0 \ j=1,2,...,m, \ \forall x \in M \\ -\infty & \text{otherwise} \end{cases}$$

Substituting the latter expressions into (19) yields (18). □

We note that generally $\psi_M^* < f_M^*$, since there is a duality gap in nonconvex problems. However, when M is just a singleton then (15) is convex and $g_M = f_M^*$ under the standard constraint qualification in semidefinite programming. Thus as shown in the next sections, we can reduce the duality gap $f_M^* - g_M$ by partitioning X. As shown hereafter, problem (18) actually is an easily tractable LMI optimization problem. Efficient interior-point algorithms [1, 9, 22] can thus be utilized to compute the lower bound (18). This is in sharp contrast with the generalized Bender's decomposition for (15) in [5, 6] where a nonconvex dual problem has to be considered.

3 A Branch and Bound Method

For every rectangle $M \subset X$ denote by $(Q(M))$ problem (18). Since $A_j(x,p,q)$ is affine in x, we infer

$$\min_{x\in M} \mathrm{Tr}(ZA_0(x,p,q)) = \min_{x\in \mathrm{vert}(M)} \mathrm{Tr}(ZA_0(x,p,q)),$$

and

$$\mathrm{Tr}(ZA_j(x,p,q)) \geq 0, \ \forall x \in M \Leftrightarrow \mathrm{Tr}(ZA_j(x,p,q)) \geq 0, \ \forall x \in \mathrm{vert}(M)\,,$$

where $\mathrm{vert}(M)$ stand for the set of vertices of M.

We deduce that $(Q(M))$ is equivalent to

$$\max_{Z\geq 0}\{ \min_{x\in \mathrm{vert}(M)} \mathrm{Tr}(ZA_0(x,p,q)) : \ \mathrm{Tr}(ZQ)=1, \ \mathrm{Tr}(ZA_j(x,p,q)) \geq 0 \ \ \forall x \in \mathrm{vert}(M), \ j=1,2,...,m\} \tag{20}$$

which is actually an *LMI program*, alternatively rewritten as

$$\psi_M^* = \max\{t: \ \mathrm{Tr}(ZA_0(x,p,q)) \geq t, \ \mathrm{Tr}(ZA_j(x,p,q)) \geq 0, \ \forall\, x \in \mathrm{vert}(M), \ j=1,2,...,m \ \ \mathrm{Tr}(ZQ)=1, \ Z \geq 0\}$$

Clearly, $\psi_M^* \leq f_M^*$, i.e. ψ_M^* is a lower bound for f_M^*. Using this lower bound, we can conceive the following BB algorithm for solving the BMI problem.

Algorithm

- *Initialization.* Let $(\bar{x}^1, \bar{y}^1, \bar{t}^1)$ be an initial feasible solution with $\bar{t}^1 = \lambda_{\max}[A_{00}(\bar{x}^1) + \sum_{j=1}^{m} \bar{y}_j^1 A_j(\bar{x}^1)]$. Set $M_0 = X, \mathcal{S}_1 = \mathcal{P}_1 = \{M_0\}$. Set $k=1$.

- *Step* 1. For each rectangle $M = [p, q] \in \mathcal{P}_k$ solve

  $$\max_{Z \geq 0}\{\min_{x \in M} \mathrm{Tr}(ZA_0(x, p, q)) : \ \mathrm{Tr}(ZQ) = 1, \ \mathrm{Tr}(ZA_j(x, p, q)) \geq 0, \ \forall x \in \mathrm{vert}(M)\}$$

  to obtain ψ^*_M.

- *Step* 2. Delete every rectangle $M \in \mathcal{S}_k$ such that $\psi^*_M \geq \bar{t}^k - \varepsilon$. Let $\mathcal{R}_k$ be the collection of remaining rectangles.

- *Step* 3. If $\mathcal{R}_k = \emptyset$ then terminate: $(\bar{x}^k, \bar{y}^k, \bar{t}^k)$ is an ε-optimal solution.

- *Step* 4. Let $M_k \in \mathrm{argmin}\{\psi^*_M | \ M \in \mathcal{R}_k\}$.

  Compute a new feasible solution by local search from the center of M_k (or by solving an LMI program, see Remark below). Let $(\bar{x}^{k+1}, \bar{y}^{k+1}, \bar{t}^{k+1})$ be the new incumbent with $\bar{t}^{k+1} = \lambda_{\max}[A_{00}(\bar{x}^{k+1} + \sum_{j=1}^m \bar{y}_j^{1+1} A_{j0}(\bar{x}^{k+1})]$.

- *Step* 5. Bisect M_k upon its longest edge. Let $\mathcal{P}_{k+1}$ be the partition of M_k.

- *Step* 6. Set $\mathcal{S}_{k+1} = (\mathcal{R}_k \setminus \{M_k\}) \cup \mathcal{P}_{k+1}, k \leftarrow k+1$ and go back to Step 1.

4 Global convergence

An important question concerns the convergence properties of the proposed BB algorithm. The answer turns out to be positive as shown below.

From now on, the following standard regularity assumption is assumed.

$$(\mathbf{R}) \qquad (\forall x \in X) \quad (\exists Z_1 \geq 0) \quad \mathrm{Tr}(Z_1 Q_{00}) = 1, \ \mathrm{Tr}(Z_1 A_{j0}(x)) > 0, \ j = 1, 2, ..., m.$$

Note that $(\mathbf{R})$ simply means that the convex dual problem of (1)-(3) for fixed $x \in X$ is strictly feasible. It is just a standard technical assumption to ensure that whenever $x \in X$ is fixed there is no duality gap between (1)-(3) and its dual [16]. Such assumption is present in every primal-dual algorithms of semi-definite programming. When such assumption does not hold, there are many ways to reduce any original problem to another one which satisfies this assumption (see e.g. [7, section 2.5]). Alternatively, we can replace this assumption by the strict feasibility of (2)-(3) for every fixed $x \in X$.

Now, let $\{M_k\}$ = with $M_k = [p^k, q^k]$ be a nested sequence of subrectangles of X such that $\cap_{k=1}^\infty M_k = \{\bar{x}\}$. For convenience, we shall write $f^*_k, \psi^*_k, \psi^*_{\bar{x}}$ for $f^*_{M_k}, \psi^*_{M_k}, \psi^*_{\{\bar{x}\}}$. Clearly,

$$\psi^*_k \leq f^*_k \tag{21}$$

Using (16) we can write

$$\begin{aligned}
\psi^*_k = \max\{t \ \ : \ \ & \mathrm{Tr}(Z_1 A_{00}(x)) + \mathrm{Tr}(Z_2 A_{01}(x, p^k, q^k)) \geq t, \\
& \mathrm{Tr}(Z_1 A_{j0}(x)) + \mathrm{Tr}(Z_2 A_{j1}(x, p^k, q^k)) \geq 0 \\
& \forall x \in \mathrm{vert}(M_k), \ j = 1, 2, ...m; \\
& \mathrm{Tr}(Z_1 Q_{00}) = 1, \ Z_1 \geq 0, \ Z_2 \geq 0\} \qquad (22) \\
\psi^*_{\bar{x}} = \max\{t \ \ : \ \ & \mathrm{Tr}(Z_1 A_{00}(\bar{x})) \geq t, \ \mathrm{Tr}(Z_1 A_{j0}(\bar{x})) \geq 0, \ j = 1, 2, ..., m, \\
& \mathrm{Tr}(Z_1 Q_{00}) = 1, \ Z_1 \geq 0\} \qquad (23)
\end{aligned}$$

Since $\bar{x} \in M_k \; \forall k$, we have

$$f_k^* \leq \min\{t : \; A_{00}(\bar{x}) + \sum_{j=1}^{m} y_j A_{j0}(\bar{x}) \leq tQ_{00}, y \geq 0\}.$$

By virtue of the duality theorem for semidefinite programming (see e.g. [16, section 4.2])

$$\begin{aligned} & \min\{t : \; A_{00}(\bar{x}) + \sum_{j=1}^{m} y_j A_{j0}(\bar{x}) \leq tQ_{00}, y \geq 0\} \qquad (24) \\ = \; & \max\{\mathrm{Tr}(Z_1 A_{00}(\bar{x})) : \; \mathrm{Tr}(Z_1 Q_{00}) = 1, \; \mathrm{Tr}(Z_1 A_{j0}(\bar{x})) \geq 0, \; j = 1, \ldots, m, \; Z_1 \geq 0\} \\ = \; & \psi_{\bar{x}}^* \quad \text{(from the definition (23) of } \psi_{\bar{x}}^*). \end{aligned}$$

Hence,

$$f_k^* \leq \psi_{\bar{x}}^*. \qquad (25)$$

Also, setting

$$\begin{aligned} L &= \{Z_1 \geq 0 : \; \mathrm{Tr}(Z_1 Q_{00}) = 1, \; \mathrm{Tr}(Z_1 A_{j0}(\bar{x})) \geq 0, \; j = 1, 2, ..., m\}, \\ L_k &= \{Z_1 \geq 0 : \; \mathrm{Tr}(Z_1 Q_{00}) = 1, \; \mathrm{Tr}(Z_1 A_{j0}(x)) \geq 0 \; \forall x \in M_k, \; j = 1, 2, ...m\}, \end{aligned}$$

and

$$\tilde{\psi}_k^* = \max\{\min_{x \in M_k} \mathrm{Tr}(Z_1 A_0(x)) : \; Z_1 \in L_k\}, \qquad (26)$$

Obviously,

$$\tilde{\psi}_{\bar{x}}^* = \psi_{\bar{x}}^* = \max\{\mathrm{Tr}(Z A_0(\bar{x}) : \; Z \in L\},$$

and

$$L_1 \subset \ldots \subset L_k \subset \ldots \subset L, \quad \tilde{\psi}_1^* \leq \ldots \leq \tilde{\psi}_k^* \leq \ldots \leq \tilde{\psi}_{\bar{x}}^* = \psi_{\bar{x}}^*.$$

Finally, since every $Z_1 \in L_k$ results $(Z_1, 0)$ feasible to (22) we have also

$$\tilde{\psi}_k^* \leq \psi_k^* \qquad (27)$$

The next proposition is our main result which allows us to conclude global convergence of the BB algorithm.

Proposition 2 *If $\{M_k\}$ is any nested sequence of subrectangles of X such that $\cap_{k=1}^{\infty} M_k = \{\bar{x}\}$, then*

$$f_k^* - \psi_k^* \to 0. \qquad (28)$$

Proof: For all k we have, by (21), (25), $\tilde{\psi}_k^* \leq \psi_k^* \leq f_X^* \leq f_k^* \leq \psi_{\bar{x}}^*$, so it suffices to show that

$$\tilde{\psi}_k^* \nearrow \psi_{\bar{x}}^* \quad (k \to +\infty).$$

Denote $\alpha_i(Z_1, x) = \mathrm{Tr}(Z_1 A_{j0}(x))$ and $\alpha(Z_1, x) = \min_{j=1,2,\ldots,m} \alpha_i(Z_1, x)$. Observe that if $Z_1 \in L$ and $\alpha(Z_1, \bar{x}) > 0$ then by continuity of $\alpha(Z_1, x)$, one must have $\alpha(Z_1, x) > 0$ for all x in a sufficiently small ball around $\bar{x}$, hence for all $x \in M_k$ with sufficiently large k (since $\max_{x \in M_k} \|x - \bar{x}\| \to 0$ as $k \to +\infty$). Thus, if $Z_1 \in L$ with $\alpha(Z_1, \bar{x}) > 0$ then $Z_1 \in L_k$ for all sufficiently large k. Keeping this in mind let $\varepsilon > 0$ be an arbitrary positive scalar and

$\bar{Z}_1$ an arbitrary element of L. Take any Z_1 corresponding to $\bar{x}$ according to Assumption (**R**) and for $\theta \in [0,1)$ define $Z_1^\theta := (1-\theta)Z_1 + \theta\bar{Z}_1$. Then $Z_1^\theta \in L$, $\mathrm{Tr}(Z_1^\theta A_{j0}(\bar{x})) = (1-\theta)\mathrm{Tr}(Z_1 A_{j0}(\bar{x})) + \theta\mathrm{Tr}(\bar{Z}_1 A_{j0}(\bar{x})) > 0,\ i = 1,\ldots,m$ and since $\lim_{\theta\to 1}\|Z_1^\theta - \bar{Z}_1\| = 0$, for θ sufficiently near to 1, we have $\mathrm{Tr}(Z_1^\theta A_{00}(\bar{x})) \geq \mathrm{Tr}(\bar{Z}_1 A_{00}(\bar{x})) - \varepsilon$. Furthermore, since $Z_1^\theta \in L$ with $\alpha(Z_1^\theta, \bar{x}) > 0$, it follows by the above that $Z_1^\theta \in L_k$ for all sufficiently large k and, consequently, $\min_{x\in M_k} \mathrm{Tr}(Z_1^\theta A_{00}(x)) \leq \tilde{\psi}_k^*$. In view of $\cap_{k=1}^\infty M_k = \bar{x}$, we thus have $\tilde{\psi}_k^* \geq \mathrm{Tr}(Z_1^\theta A_{00}(\bar{x})) - \varepsilon \geq \mathrm{Tr}(\bar{Z}_1 A_{00}(\bar{x})) - 2\varepsilon$ for all sufficiently large k. Since $\varepsilon > 0$ and $\bar{Z}_1 \in L$ are arbitrary, this shows that $\tilde{\psi}_k^* \to \psi_{\bar{x}}^*$, hence completing the proof. □

Theorem 1 *The above algorithm can be infinite only if $\varepsilon = 0$. In the latter case, the sequence $\psi_{M_k}^*$ tends to the global minimum of (15), and any cluster point $\bar{x}$ of the sequence $\{\bar{x}^k\}$ yields an optimal solution $(\bar{x}, \bar{y}, \bar{t})$ where $(\bar{y}, \bar{t})$ is an optimal solution of (24).*

Proof: By the selection rule of M_k, we have $\psi_{M_k}^* \leq f^*$ for all k. Since the subdivision is exhaustive if the procedure is infinite it generates at least one nested sequence of rectangles $\{M_{k_\nu}\}$ collapsing to a singleton $\{\bar{x}\}$. The conclusion then follows from Proposition 2. □

Remarks. It is important to note that the subdivision process is exhaustive when the diameter of the sequence of rectangles (the length of their longest edge) tends to zero as k tends to infinity. This is indeed the case when a bisection rule is used. That is, rectangles are splitted through the middle point of their longest edge. Relation (28) is the consistency condition which ensures convergence of most BB algorithms, since it guarantees that the smallest lower bound at iteration k tends to the sought global minimum of the overall problem. An difficulty for the convergence of the generalized Bender's decomposition algorithm of [5, 6] is clearly connected with such consistency condition. This also explains why not all relaxation can be used for producing a globally convergent BB algorithm.

In step 4 of the algorithm a new incumbent is easily generated as follows. Let x^k be any point of M_k, then an optimal solution (y^k, t^k) of the LMI program

$$\min\{t:\ A_{00} + \sum_{j=1}^m y_j A_{j0}(x^k) \leq tQ, y \geq 0\}$$

yields a feasible solution (x^k, y^k, t^k) of the primal problem.

Incidentally, we have shown that our algorithm remains convergent if $\tilde{\psi}_M$ defined by (26) is used for computing the lower bound at Step 1 instead of ψ_M. Clearly, $\tilde{\psi}_M$ is the optimal value of the Lagrangian dual problem for (11)-(12). However, by (27), we can see that the lower bound ψ_M is tighter than $\tilde{\psi}_M$ and thus speeds up the convergence of the algorithm. Thus, the additional BMI constraint (14) is very useful for improving the performance of the algorithm.

In the previous works [10, 11, 20] the lower bound computings are based on the primal like forms of LMIs while (20) is dual like one, i.e. they are complementary each either. Thus, with efficient primal-dual interior algorithms like SDPPACK[1] used, they seemly require the same computational effort.

For simplicity of descriptions, we have assumed that X in (1) is a box in R^n. Obviously, the algorithm remains valid when X is a simplex in R^n and the simplicial subdivision is accordingly used instead of the rectangular one together with an adequate bisection subdivision rule. With such simplicial partition, the number of the inequality constraint in (22) is proportional to $(n+1)$ instead of 2^n for the rectangular subdivision. Thus, even for the box constraint (1) with n larger it is recommended to partition X first into simplices and then solve (1)-(3) by the simplicial partition. When M in (12)-(13) is a simplex described by a linear inequality system

$$\langle a^\ell, x\rangle \geq \alpha_\ell,\ i = 1, 2, ..., h \tag{29}$$

then the following BMI constraint should be incorporated to (12)-(13)

$$(\langle a^\ell, x\rangle - \alpha_\ell)G_0 + \sum_{j=1}^{m} y_j(\langle a^\ell, x\rangle - \alpha_\ell)G_j \leq 0,\ \ell = 1, 2, ..., h, \tag{30}$$

instead of (14). Accordingly, A_{j1} in (16) is changed to

$$A_{j1}(x, M) = \begin{bmatrix} (\langle a^1, x\rangle - \alpha_1)G_j \\ (\langle a^2, x\rangle - \alpha_2)G_j \\ .... \\ (\langle a^h, x\rangle - \alpha_h)G_j \end{bmatrix}_d,\ j = 0, 1, ..., m. \tag{31}$$

5 Application to a robust control problem

In this section, we provide a simple illustration of the proposed method for solving BMI problems which can serve as our preliminary computational result. As claimed previously, the advantage of our BB algorithms for solving (1)-(3) is that the branching is performed in the reduced space R^m instead of the whole space R^{n+m} that make it practical for a small m though $n+m$ is large. Therefore, the best illustration for this is a a problem which can be reduced to the form (1)-(3) with n small compared with m. Thus, we consider an $\mathcal{H}_\infty$ control problem in which both the controller and some plant's coefficients or structure can be designed simultaneously. The design of the system structure consists, for example, in deciding what devices are used, where these are located, what values must be taken by some physical quantities. Classically, for a given plant the structure is determined first and the controller is designed afterwards. However, the simultaneous design of controller and structure, certainly constitutes a better policy to find answers to the overall control problem.

In this example, the plant type is given beforehand, but parameter values are adjusted off-line to improve the performance of the overall system. Our aim is to attenuate the effect of disturbances on the mass in a mass-spring arrangement. The controlled system is shown in Figure 1, where x is the position of mass, u is the control force input, w is the disturbance force, m is the mass ($m = 4$), k and c are the spring and damping coefficients which are allowed to take values in a parameter box H,

$$p := (k, c) \in H = [4, 12] \times [0.5, 1.5].$$

The controlled output z and measurement output y are

$$z = [x, u]', \quad y = x.$$

Considering $[x, \dot{x}]'$ as the state vector, the generalized plant is given as

$$P(s) = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right] = \left[\begin{array}{cc|c|c} 0 & 1 & 0 & 0 \\ -\frac{k}{m} & -\frac{c}{m} & \frac{1}{m} & \frac{1}{m} \\ \hline 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \end{array}\right].$$

Now, the problem can be stated as to find an internally stabilizing proper controller $K(s)$ and a parameter $p \in H$ minimizing the H_∞ norm $\|T_{zw}\|_\infty$, where T_{zw} is the closed-loop transfer function from w to z. It is well known [8] that this problem has the BMI formulation

$$\begin{array}{rl} (\mathbf{P}) & \text{minimize} \quad \gamma : \\ & \mathcal{N}_1' \begin{bmatrix} A(p)\widehat{R} + \widehat{R}A(p)' & \widehat{R}C_1' & B_1 \\ C_1\widehat{R} & -\gamma I & 0 \\ B_1' & 0 & -\gamma I \end{bmatrix} \mathcal{N}_1 < 0 \\ & \mathcal{N}_2' \begin{bmatrix} A(p)'\widehat{S} + \widehat{S}A(p) & \widehat{S}B_1 & C_1' \\ B_1'\widehat{S} & -\gamma I & 0 \\ C_1 & 0 & -\gamma I \end{bmatrix} \mathcal{N}_2 < 0 \\ & \begin{bmatrix} \widehat{R} & I \\ I & \widehat{S} \end{bmatrix} \geq 0 \\ & p \in H\,, \end{array} \tag{32}$$

where $\mathcal{N}_1$ and $\mathcal{N}_2$ denote any bases of the null spaces of $[B_2', D_{12}', 0]$ and $[C_2, D_{21}, 0]$, respectively.

It is important to note that the dimension of the complicating variable p in (32) is just 2 and compares favorably with the dimension 9 of the all decision variables $(\gamma, \hat{R}, \hat{S}, p)$ in (32). Thus, our BB algorithm with branching performed in a subset of space R^2 is really advantageous compared with BB algorithms operating in R^9.

In order to transform problem (32) into the form (15), we use the change of variables

$$\widehat{R} = R_0 + R = R_0 + \sum_{i \leq j}^{2} R(i,j) I_{ij}, \qquad \widehat{S} = S_0 + S = S_0 + \sum_{i \leq j}^{2} S(i,j) I_{ij},$$

where I_{ij} is the symmetric 2×2 matrix with all zero entries but $I_{ij}(i,j) = I_{ij}(j,i) = 1$, and R_0, S_0 are constant matrices of the form

$$R_0 = \begin{bmatrix} 0 & -\xi_R \\ -\xi_R & 0 \end{bmatrix} = -\xi_R I_{12}, \qquad S_0 = \begin{bmatrix} 0 & -\xi_S \\ -\xi_S & 0 \end{bmatrix} = -\xi_S I_{12},$$

with ξ_R and ξ_S chosen large enough positive scalars, so that all entries of R and S can be assumed nonnegative.

Thus, (32) is rewritten in the form (15) as

$$\begin{bmatrix} L_0(p) \\ M_0(p) \\ N_0 \end{bmatrix}_d + \sum_{i \leq j}^{2} R(i,j) \begin{bmatrix} L_{ij}(p) \\ 0 \\ N_{1ij} \end{bmatrix}_d + \sum_{i \leq j}^{2} S(i,j) \begin{bmatrix} 0 \\ M_{ij}(p) \\ N_{2ij} \end{bmatrix}_d < \gamma \begin{bmatrix} Q_1 \\ Q_2 \\ 0 \end{bmatrix}_d, \tag{33}$$

where

$$
\begin{aligned}
L_0(p) &= \mathcal{N}_1' \begin{bmatrix} A(p)R_0 + R_0A(p)' & R_0C_1' & B_1 \\ C_1R_0 & 0 & 0 \\ B_1' & 0 & 0 \end{bmatrix} \mathcal{N}_1, \\
L_{ij}(p) &= \mathcal{N}_1' \begin{bmatrix} A(p)I_{ij} + I_{ij}A(p)' & I_{ij}C_1' & 0 \\ C_1I_{ij} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathcal{N}_1, \\
Q_1 &= \mathcal{N}_1' \begin{bmatrix} 0 \\ I \\ I \end{bmatrix}_d \mathcal{N}_1, \\
M_0(p) &= \mathcal{N}_2' \begin{bmatrix} A(p)'S_0 + S_0A(p) & S_0B_1 & C_1' \\ B_1'S_0 & 0 & 0 \\ C_1 & 0 & 0 \end{bmatrix} \mathcal{N}_2, \\
M_{ij}(p) &= \mathcal{N}_2' \begin{bmatrix} A(p)'I_{ij} + I_{ij}A(p) & I_{ij}B_1 & 0 \\ B_1'I_{ij} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathcal{N}_2, \\
Q_2 &= \mathcal{N}_2' \begin{bmatrix} 0 \\ I \\ I \end{bmatrix}_d \mathcal{N}_2, \\
N_0 &= -\begin{bmatrix} R_0 & I \\ I & S_0 \end{bmatrix}, \; N_{1ij} = \begin{bmatrix} -I_{ij} & 0 \\ 0 & 0 \end{bmatrix}, \; N_{2ij} = \begin{bmatrix} 0 & 0 \\ 0 & -I_{ij} \end{bmatrix}.
\end{aligned}
$$

The dual problem on a rectangle $M \subset H$ then becomes

$$
\begin{array}{rl}
(\mathbf{D}) \qquad \text{maximize} & t: \\
Z_1 \geq 0,\ Z_2 \geq 0,\ Z_3 \geq 0 & \\
\mathrm{Tr} Z_1 Q_1 + \mathrm{Tr} Z_2 Q_2 = 1 & \\
\mathrm{Tr} Z_1 L_0(p) + \mathrm{Tr} Z_2 M_0(p) - \mathrm{Tr} Z_3 N_0 \geq t & \\
\mathrm{Tr} Z_1 L_{ij}(p) - \mathrm{Tr} Z_3 N_{1ij} \geq 0 & \\
\mathrm{Tr} Z_2 M_{ij}(p) - \mathrm{Tr} Z_3 N_{2ij} \geq 0 & \\
1 \leq i \leq j \leq 2,\ \forall p \in \mathrm{vert}(M)\,. &
\end{array}
\tag{34}
$$

For practical computations, the selection of appropriate ξ_R and ξ_S is difficult since if they are unnecessarily too large, the dual problem $(\mathbf{D})$ may yield a poor lower bound. Hence we find a lower bound of $M = [p_1, q_1] \times [p_2, q_2]$, based on the idea that optimal solution of $(\mathbf{P})$ for M, $\widehat{R}^*(M), \widehat{S}^*(M)$ will be near to that of $(\mathbf{P})$ at p_c, $\widehat{R}^*(p_c), \widehat{S}^*(p_c)$ with $p_c = (\frac{p_1+q_1}{2}, \frac{p_2+q_2}{2})$. For this, first, we solve $(\mathbf{P})$ at p_c to obtain an optimal solution $\widehat{R}^*$ and $\widehat{S}^*$. Then set $m_R := \min_{i \leq j} \widehat{R}^*(i,j)$. If $m_R \geq 0$ then $\xi_R = 0$, otherwise $\xi_R = -\eta \times m_R$, where $\eta = 1.5 \sim 3$. The scalar ξ_S is handled similarly. With these values of ξ_R and ξ_S, we can obtain better lower bounds on M.

Note in this example, the optimal solution $\widehat{S}$ seems to have negative elements with large absolute value, which causes large value of ξ_S. To overcome this difficulty, we treat two distinguished problems with $\widehat{S}(1,2) \geq 0$ and $\widehat{S}(1,2) \leq 0$, respectively. We have used the LMI toolbox [9] for solving the primal $(\mathbf{P})$ while the solver SDPPACK [1] has been used for the dual (**D**). All computations were performed on a PC with with CPU Pentium-II 333 MHz. The computational results with tolerance $\epsilon = 0.01$ for both cases are the following.

Case 1. $\widehat{S}(1,2) = \widehat{S}(2,1) \geq 0$ which corresponds to setting $\xi_S = 0$. The performance of the BB algorithm at every iteration is displayed in Table 1. The optimal solution in this case is

$$(k_{opt}, c_{opt}) = (11.969, 1.469),$$

$$R_{opt} = \begin{bmatrix} 0.1351 & -0.0248 \\ -0.0248 & 0.4043 \end{bmatrix}, \qquad S_{opt} = \begin{bmatrix} 9.2882e^{7} & 1.4606e^{-8} \\ 1.4606e^{-8} & 3.2739e^{-7} \end{bmatrix}.$$

Case 2. $\widehat{S}(1,2) = \widehat{S}(2,1) \leq 0$. Compared with case 1, we only change $\widehat{S} = S(1,1)I_{11} - S(1,2)I_{12} + S(2,2)I_{22}$. Again the performance of our algorithm at every iteration is given in Table 2.
The optimal solution in this case is

$$(k_{opt}, c_{opt}) = (11.969, 1.469),$$

$$R_{opt} = \begin{bmatrix} 0.1351 & -0.0248 \\ -0.0248 & 0.4044 \end{bmatrix}, \qquad S_{opt} = \begin{bmatrix} 7.3723e^{7} & -1.1381e^{-2} \\ -1.1381e^{-2} & 2.7994e^{-5} \end{bmatrix}.$$

From Table 1 and Table 2, we can see how the duality gap is reduced along iterations. Observe that due some computational instabilities/inaccuracies, the lower bound in case 2 are only slightly decreasing at some few iterations 3, 8 and 12.

The benefit of the structure design is clear by looking at Table 3, where the performance of the plant with the optimal structure has been improved compared with the nominal plant. The step responses of given in Figure 2 confirms that tracking performance of the optimal structure systems is much better than that of the nominal plant.

The H_∞ performance level as a functions of the parameters k and c is described in Figure 3. One can easily verified that the global optimal solution has been obtained.

Before closing this section let us mention some our computational experiences. We have tested up to 100 randomly generated examples for problems like (32) with different sizes of $\hat{R}, \hat{S}$ and the size of p ranged from 2 to 4. As expected, the number of iterations needed for solving every problem is not sensitive to the size of $\hat{R}, \hat{S}$ and only sensitive to the size of p (i.e. the size of the branching space), i.e. our algorithm may work well for problems with a small dimension of the complicating variables. Particularly, the iteration performance given in Tab.1 and Tab.2 is quite typical for case $m = 2$ in (1)-(3). For $m = 3$ and $m = 4$ the averaged iteration number is increasing to 92.5 and 210.3, respectively, while of course the execution time also depends strongly on the efficiency of the LMI solvers that are used at every iteration. This again confirms the well-known fact in global optimization that the performance of an algorithm heavily depends on the performed branching space and it is a main motivation for developing BB algorithm with branching performed in a reduced space. Note that we solve the optimization problem under the BMI constraints which is much harder than the BMI feasibility problems in the context of global optimization. This is due to the fact that like many other global optimization algorithms, our algorithm could find a value close to the optimal value at an early computational stage and the main effort is to confirm its optimality or to refine its level of optimality. Clearly, such an issue is completely absent in the BMI feasibility problem since in such a case the algorithm can stop earlier when constraints hold.

6 Concluding remarks

A new convergent BB algorithm of global optimization for solving BMI problems is proposed. First of all, the original problem is reduced to a convenient form such that its Lagrangian dual problem is just a convex LMI optimization problem which is used for computing lower bounds. Exploiting the most useful characterization of BMI that it becomes convex when a certain number of so called "complicating" are held fixed, the duality gap is reduced by an exhaustive partitioning on a reduced space of these variables which favorably limits the overall cost of the algorithm. The viability of the proposed algorithms is confirmed by solving a robust structure/control design and other computational examples.
Acknowledgments. We are greatly grateful to Professor Hoang Tuy for his frequent support and corrections during this work.

References

[1] F.Alizadeh, J.P.Haeberly, M.V.Nayakkankuppam, M.L.Overton, *SDPPACK*, version 0.8 beta, 1997.

[2] F.A. Al-Khayyal, J.E. Falk, Jointly constrained biconvex programming, *Mathematics of Operation Research* 8(1983), 273-286.

[3] F.A. Al-Khayyal, Jointly constrained bilinear programs and related problems: an overview, *Computers and Mathematics with Applications* 19(1990), 53-62.

[4] P. Apkarian, H.D. Tuan, Robust control via concave optimization: local and global algorithms, *Proc. of 37th CDC*, 3855-3860; To appear in *IEEE Trans. Autmomatic Control.*

[5] E. Beran, Methods for optimization based fixed-order control design, *Ph.D. thesis*, Technical University of Denmark, 1997.

[6] E. Beran, L. Vandenberge, S. Boyd, A global BMI algorithm based on the generalized Benders decomposition, *Proc. of ECC*, 1997.

[7] S. Boyd, L. ElGhaoui, E. Feron, V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory*, SIAM Studies in Applied Mathematics, Philadelphia, 1994.

[8] P. Gahinet, P. Apkarian, A linear matrix inequality approach to H_∞ control, *Inter. J. of Robust and Nonlinear Control* 4(1994), 421-448.

[9] P. Gahinet, A. Nemirovski, A.J.Laub, M.Chilali, *LMI Control Toolbox,* The MathWorks Inc., 1994.

[10] K.C. Goh, M.G. Safonov, G.P. Papavassilopoulos, A global optimization approach for the BMI problem, *Journal of Global Optimization*, 7(1995), 365-380.

[11] H. Fujioka, K. Hoshijima, Bounds for the BMI eigenvalues problems-A good lower bound and a cheap upper bound, *Trans. of SICE* 33(1997), 616-621.

[12] R. Horst, H. Tuy, *Global optimization: deterministic approaches* (3rd edition), Springer, 1996.

[13] M. Kawanishi, T. Sugie, H. Kanki, BMI global optimization based on branch and bound method taking account of the property of local minima, *Proc. of CDC 1997*, 781-786.

[14] H. Konno, P.T. Thach, H. Tuy, *Optimization on Low Rank Nonconvex Structures,* Kluwer Academic, 1997

[15] G.P. McCormic, Computability of global solutions to factorable nonconvex programs: part I-convex underestimating problems, *Mathematical Programming* 10(1976), 147-175.

[16] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Methods for Convex Programming: Theory and Applications*, SIAM, Philadelphia, 1994.

[17] M.G. Safonov, K.C. Goh, J.H. Ly, Control systems synthesis via bilinear matrix inequalities, *Proc. of the American Control Conference 1994*, pp. 45–49.

[18] H.D. Sherali, A. Alameddine, A new reformulation-linearization technique for bilinear programming problems, *J. of Global Optimization* 2(1992), 379-410.

[19] O. Toker, H. Ozbay, On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback, *Proc. of ACC 1995*, 2525-2526.

[20] H.D. Tuan, S. Hosoe, H. Tuy, D.C. optimization approach to robust controls: the feasibility problem, *Preprint 9601*, Department of Electronic-Mechanical Engineering, Nagoya University, Japan, October 1996; To appear in *Int. J. Control.*

[21] H. Tuy, *Convex analysis and global optimization,* Kluwer Academic 1998.

[22] L. Vandenberghe, S. Boyd, Semidefinite programming, *SIAM Review*, 38(1996), 49-95.

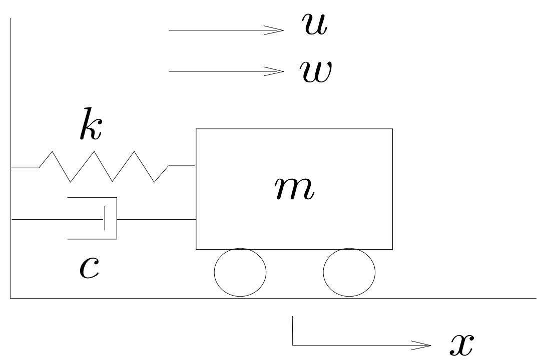

Figure 1: The mass-spring-damper system

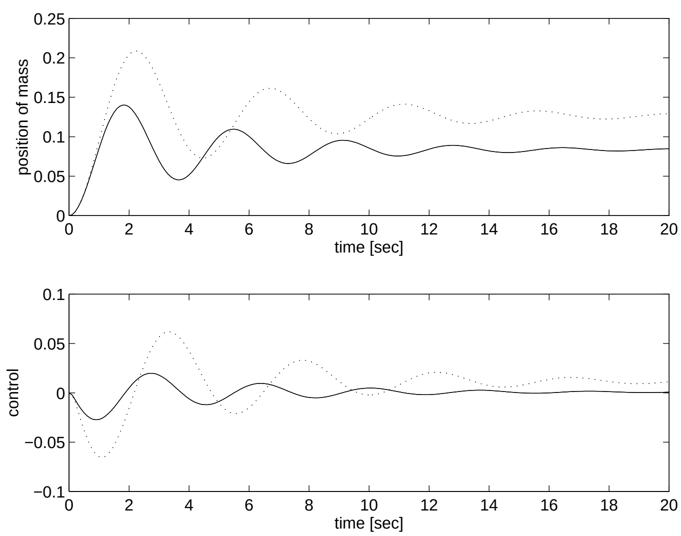

Figure 2: The tracking performance of the optimal structure plant (solid) and the nominal plant (dot) under the step disturbance

iteration	lower bound	current best value	# of remaining rectangles
1	0.240	0.579	1
2	0.293	0.536	2
3	0.321	0.518	3
4	0.323	0.518	4
5	0.339	0.509	5
6	0.336	0.428	5
7	0.344	0.424	6
8	0.347	0.424	7
9	0.347	0.392	6
10	0.351	0.391	7
11	0.347	0.391	7
12	0.353	0.391	8
13	0.353	0.391	8
14	0.353	0.376	6
15	0.354	0.376	7
16	0.356	0.375	7
17	0.357	0.375	7
18	0.357	0.369	3
19	0.357	0.369	2
20	0.359	0.368	0

Table 1: Performance of the algorithm for case 1. Total cputime is 598 sec.

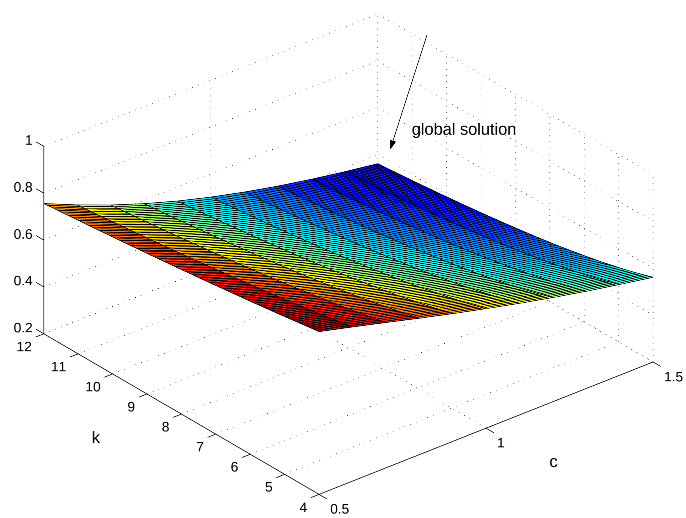

Figure 3: H_∞ performance versus parameters (k, c)

iteration	lower bound	current best value	#remaining rectangles
1	0.236	0.579	1
2	0.292	0.536	2
3	(0.026)	0.517	3
4	0.320	0.517	4
5	(0.034)	0.517	5
6	0.323	0.517	6
7	0.338	0.509	7
8	(0.336)	0.428	5
9	0.347	0.424	6
10	0.347	0.392	6
11	0.350	0.391	7
12	(0.347)	0.391	7
13	0.352	0.391	8
14	0.353	0.376	8
15	0.353	0.376	6
16	0.354	0.375	7
17	0.356	0.375	7
18	0.357	0.368	7
19	0.357	0.368	3
20	0.357	0.368	2
21	0.359	0.368	0

Table 2: Performance of the algorithm for case 2. Total cputime is 891 sec.

parameter (k, c)	performance level γ_{opt}
(11.969,1.469) (optimal)	0.3681
(8,1) (nominal)	0.5791

Table 3: comparison with nominal case