Low Nonconvex Rank Bilinear Matrix Inequalities: Algorithm and Applications

H.D. Tuan*and **P. Apkarian**†

Abstract

A new Branch and Bound (BB) algorithm for solving a general class of Bilinear Matrix Inequality (BMI) problem is proposed. First, Linear Matrix Inequality (LMI) constraints are incorporated into BMI constraints in a special way to take advantage of useful informations on nonconvex terms. Then, the nonconvexity of BMI is centralized in coupling constraints so that when the latter are omitted, we get a relaxed LMI problem for computing lower bounds. As in our previous developments, the branching is performed in a reduced dimensional space of complicating variables. This makes the approach practical even with a large dimension of overall variables. Applications of the algorithm to several test problems of robust control are discussed.

1 Introduction

In this paper, we shall focus on the following BMI optimization problem

---

$$\min_{x,y} \ \langle c, x\rangle + \langle d, y\rangle \ : \tag{1}$$

$$F_{00} + \sum_{i=1}^{N_1} x_i F_{i0} + \sum_{j=1}^{N_2} y_j F_{0j} + \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} x_i y_j F_{ij} < 0, \tag{2}$$

$$G_0 + \sum_{j=1}^{N_2} y_j G_j \le 0, \tag{3}$$

$$x \in [a,b], \ \ [a,b] \subset R^{N_1}, \ \ a > 0, \tag{4}$$

---

where F_{ij}, G_i are symmetric matrices, $c \in R^{N_1}, d \in R^{N_2}$, and the interval inclusion (4) must be understood componentwise, that is, $a_i \le x_i \le b_i, \ \ i = 1,\dots,N_1$. Here (2) is BMI in (x, y), while (3) is an LMI in y. From a practical point of view, we can assume that the set of feasible solutions to the LMI (3) is compact.

*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 Pierre.Apkarian@cert.fr

Clearly, the constraint

$$y \in [c, d] \subset R^{N_2}, c > 0 \tag{5}$$

is a particular case of LMI constraint (3). The simpler feasibility problem (BFP) comprising (2), (4) and (5) has been first considered in [8], using general-purpose BB procedures. It has been later improved in [5] with a tighter relaxation inspired by indefinite quadratic programming results [12, 13, 1, 9]. The branching in [8, 5] and most related works is performed in the space of all variables (x, y) of dimension $N_1 + N_2$. This entails serious limitations for the practicability of these results in the control field. Indeed, BMIs arising from control problems have large $N_1 + N_2$ even in the simplest applications. This fact is also well recognized in global optimization. The efficiency of a global optimization algorithm critically depends on the used branching space and this motivates the development of recent decomposition methods in global optimization (see e.g. [10, 21]). This idea is natural since in general the iteration number of a BB algorithm is mainly sensitive to the branching space dimension. In [18] the following fundamental features of BFP (2), (4)-(5) has been used: it becomes just *convex* LMI feasibility problem when either variable x or variable y is held fixed. Thus, assuming $N_1 \leq N_2$ (when $N_1 > N_2$ we just exchange the roles of x and y) only x can be regarded as the complicating variables responsible for the hardness/nonconvexity of BFP and hence N_1 is regarded as the nonconvex rank of BFP (2), (4) and (5) [10]. As a result, the branching in [18] is performed only in the space of complicating variables x that makes the algorithm practical even for the case of large $N_1 + N_2$, providing the nonconvex rank of BFP ($\min\{N_1, N_2\}$) is low. The algorithm of [18] has been used successfully in [16] for solving BMIs arising from robust constrained nonlinear control problems.

On the other hand, one can see that forms like (2), (4) and (5) are not the best way for expressing BMIs arising from control problems. Instead, form (2)-(4) is more natural and preferable. Of course, one may argue that LMI (3) is a particular case of BMI (2) meaning that (2)-(4) can be equivalently recast as (4 and (3). This transformation, however, obliterate the useful LMI structure of (3). On the other hand, the well-known bounding techniques of global optimization [12, 13, 1, 9] used in [5, 18] exploit the box structure of constraints (4) and (5) to handle the nonconvex constraint (2). Alternatively, in [20], BFP (2)-(3) is transformed into a special form, whose dual Lagrangian is an LMI optimization program used for computing lower bounds. The idea of [20] is that as the space of complicating variables is iteratively partitioned, the resulting nonconvex duality gap is reduced and this constitutes the basis for the convergence of the algorithm.

In this paper, similarly to [18, 20] the branching is performed only in the x-space of complicating variables instead of the entire space of (x, y). With a new direct bounding technique, we propose a novel BB algorithm solving problem (1)-(4), which is practical even for large dimension of y provided that the dimension of x is relatively small. Our idea here is to exploit maximally LMI constraints (3) to draw useful informations on the nonconvex terms $x_i y_j$ causing the nonconvexity of (2). Then, the nonconvexity of BMI is reflected in these coupling constraints. When they are omitted, we get a relaxed LMI problem, useful for lower bound computations.

The organization of the paper is as follows. Section 2 gives some BMI characterizations of LMI constraints which serve as the basic tool for developing the BB method in Section 3. BMI feasibility problems are discussed in Section 4. Finally, Section 5 provides numerical examples illustrating the viability of our method.

2 BMI unification for LMI constraints

Set

$$w_{ij} = x_i y_j, \ i = 1, \ldots, N_1, j = 1, \ldots, N_2 \tag{6}$$

then (1)-(4) is equivalent to

$$\begin{aligned} &\min_{x,y,w_{ij}} \langle c, x\rangle + \langle d, y\rangle : \ (3),(4),(6), \\ &F_{00} + \sum_{i=1}^{N_1} x_i F_{i0} + \sum_{j=1}^{N_2} y_j F_{0j} + \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} w_{ij} F_{ij} < 0. \end{aligned} \tag{7}$$

Thus the difficulty of problem (7) is concentrated in the nonconvex constraint (6). For bounding, we have to compute a lower bound of (7) for

$$x \in M = [p, q], \qquad q > p, \tag{8}$$

with a given box or hyper-rectangle $M = [p, q] \subset [a, b]$.
Our new idea here is to recast LMI constraints (3) and (8) independent of $x_i y_j$ into new LMI constraints involving the coupling $x_i y_j$. Thus, by setting $w_{ij} = x_i y_j$ we will get a tight relaxed problem for computing a lower bound of (7) over $x \in [p, q]$.

Proposition 1 *LMI constraints (3) and (8) are equivalent to the BMI constraints*

$$(x_i - p_i)[G_0 + \sum_{j=1}^{N_2} y_j G_j] \le 0; \quad (q_i - x_i)[G_0 + \sum_{j=1}^{N_2} y_j G_j] \le 0. \tag{9}$$

Proof Since the implication $(3),(8) \Rightarrow (9)$ is obvious, let us prove the reverse implication. By (9),

$$(q_i - p_i)[G_0 + \sum_{j=1}^{N_2} y_j G_j] = (q_i - x_i)[G_0 + \sum_{j=1}^{N_2} y_j G_j] + (x_i - p_i)[G_0 + \sum_{j=1}^{N_2} y_j G_j] \le 0$$

and since $q_i > p_i$, we have (3). Then (3) and (9) imply (8). □

Using (6), we can rewrite (9) as (6) with

$$\begin{aligned} x_i G_0 - p_i[G_0 + \sum_{j=1}^{N_2} y_j G_j] + \sum_{j=1}^{N_2} w_{ij} G_j \le 0; \\ -x_i G_0 + q_i[G_0 + \sum_{j=1}^{N_2} y_j G_j] - \sum_{j=1}^{N_2} w_{ij} G_j \le 0. \end{aligned} \tag{10}$$

Hereafter,we shall use the definitions:

$$r_j = \min\{y_j | (3)\}, \quad s_j = \max\{y_j | (3)\}. \tag{11}$$

The following proposition is central in the proposed approach.

Proposition 2 *With the definitions (11), the constraints (10) implies*

$$p_i y_j - w_{ij} - p_i r_j + r_j x_i \le 0, \tag{12}$$
$$p_i s_j - x_i s_j - p_i y_j + w_{ij} \le 0, \tag{13}$$
$$x_i s_j - q_i s_j - w_{ij} + q_i y_j \le 0, \tag{14}$$
$$w_{ij} - q_i y_j - x_i r_j + q_i r_j \le 0\,. \tag{15}$$

Proof Let e^j denote the jth unit vector in R^{N_2}. By the Duality Theorem of Semi-Definite Programming, there are matrices Z and Z_1 such that

$$\begin{array}{c} \mathrm{Trace}(ZG_0) = r_j,\ Z \ge 0, \\ (\mathrm{Trace}(ZG_1), \mathrm{Trace}(ZG_2), \ldots, \mathrm{Trace}(ZG_{N_2}))' = -e^j, \end{array} \tag{16}$$

$$\begin{array}{c} \mathrm{Trace}(Z_1 G_0) = -s_j, Z_1 \ge 0, \\ (\mathrm{Trace}(Z_1G_1), \mathrm{Trace}(Z_1G_2), \ldots, \mathrm{Trace}(Z_1G_{N_2}))' = e^j. \end{array} \tag{17}$$

Using (16) and the first LMI in (10) we have

$$\begin{array}{rcl} 0 & \ge & \mathrm{Trace}[Z(x_iG_0 - p_i(G_0 + \sum_{j=1}^{N_2} y_jG_j) + \sum_{j=1}^{N_2} w_{ij}G_j)] \\ & = & x_i\mathrm{Trace}(ZG_0) - p_i[\mathrm{Trace}(ZG_0) + \sum_{j=1}^{N_2} y_j \mathrm{Trace}(ZG_j)] + \sum_{j=1}^{N_2} \mathrm{Trace}(ZG_j) \\ & = & x_i r_j - p_i r_j + p_i y_j - w_{ij}, \end{array}$$

which leads (12).

Analogously, (15) is a consequence of (16) and the second LMI in (10), while (13) ((14), resp.) follows from (17) and the first (the second, resp.) LMI in (10). □

The main result in this section is the following theorem.

Theorem 1 *(i) (x, y) is feasible for the constraints (3) and (8) if and only if there exists a symmetric matrix $W = [w_{ij}]$ such that (x, y, W) satisfies (10).*
(ii) If (x, y, W) is feasible for (10) with $x_i \in \{p_i, q_i\}$ then (x, y, W) is feasible for (6).

Proof It is obvious that if (x, y) is feasible for (3) and (8) then (x, y, W) with $W = [x_iy_j]$ is feasible for (10). Now, suppose (x, y, W) is feasible for (10). Then add the first and the second LMIs of (10). We get (3) since $q_i > p_i$. Moreover, by (12)-(13),

$$\begin{array}{rl} & p_iy_j - p_ir_j + r_jx_i \le w_{ij} \le -p_is_j + x_is_j + p_iy_j \\ \Rightarrow & r_j(x_i - p_i) \le s_j(x_i - p_i) \\ \Leftrightarrow & (r_j - s_j)(x_i - p_i) \le 0 \\ \Rightarrow & x_i \ge p_i \end{array} \tag{18}$$

Analogously, (14)-(15) imply $x_i \le q_i$. Hence (8) is proved. Thus, the part (i) of the above Theorem has been proved.

To prove the part (ii), note that when $x_i = p_i$ then (18) implies $w_{ij} = p_iy_j = x_iy_j$, i.e. (x_i, y_j) satisfies (6). Analogously, with $x_i = q_i$ then (14)-(15) gives $w_{ij} = q_iy_j = x_iy_j$. The proof is thus complete.

$\square$

The equivalence between (3), (8) and (10) in Theorem 1 will be very important for computing upper bounds and in the subdivision strategy of our method.

Before closing this section, let us mention the following useful consequence of Theorem 1.

Corollary 1 *A matrix P and scalar λ satisfies conditions*

$$\lambda \in [\lambda_1, \lambda_2], \qquad ||P - P_0|| \leq \nu\,, \tag{19}$$

if and only if there is a matrix W of the same size of P such that (W, P, λ) is feasible for the LMIs

$$\begin{array}{c} \begin{bmatrix} (\lambda-\lambda_1)\nu I & W - \lambda_1 P - (\lambda - \lambda_1) P_0 \\ W' - \lambda_1 P' - (\lambda - \lambda_1) P_0' & (\lambda - \lambda_1)\nu I \end{bmatrix} \geq 0 \\ \begin{bmatrix} (\lambda_2-\lambda)\nu I & \lambda_2 P - W - (\lambda_2 - \lambda) P_0 \\ \lambda_2 P' - W' - (\lambda_2 - \lambda) P_0' & (\lambda_2 - \lambda)\nu I \end{bmatrix} \geq 0. \end{array} \tag{20}$$

Moreover, if (W, P, λ) is feasible for (20) with $\lambda \in \{\lambda_1, \lambda_2\}$ then

$$W = \lambda P. \tag{21}$$

Thus, instead of separately handling each element of matrix variables W and P in (19), which may destroy their useful structures and results in many linear constraints, the result in Corollary 1 maximally exploits the matrix structures of variables and is very convenient in the LMI-based setting. Interestingly enough, the well-known techniques of indefinite quadratic program (see e.g. [12, 21]) used in [5, 20] are unable to handle nonconvex constraints like (21).

3 BB Algorithm

Returning back to (7), by considering x as the complicating variables, we shall solve BFP by a BB method in which branching is upon x.

Bounding

With a given $M = [p, q] \in [a, b]$, we have to compute a lower bound of problem (7) for $x \in M$. By Theorem 1, the *exact* optimal value $\gamma(M)$ of (7) for $x \in M$ is computed by the nonconvex program

$$CQ(M) \left\{ \begin{array}{l} \min\limits_{x,y,w_{ij}} \ \langle c, x\rangle + \langle d, y\rangle : \ (8), (6), (10), \\ F_{00} + \sum\limits_{i=1}^{N_1} x_i F_{i0} + \sum\limits_{j=1}^{N_2} y_j F_{0j} + \sum\limits_{i=1}^{N_1}\sum\limits_{j=1}^{N_2} w_{ij} F_{ij} < 0. \end{array} \right. \tag{22}$$

Thus, by merely omitting the nonconvex constraint (6) in $CQ(M)$, we get the following LMI optimization problem which trivially provides a lower bound $\beta(M)$ for (22),

$$LB(M) \quad \min_{x,y,w_{ij}} \ \langle c, x\rangle + \langle d, y\rangle : \quad (22), (8), (10). \tag{23}$$

Let $(x(M), y(M), W(M))$ be an optimal solution of LB(M) (23), then an upper bound of (7) is provided by the following LMI program

$$UB(M) \begin{cases} \min\limits_{x,y,w_{ij}} \langle c, x(M)\rangle + \langle d, y\rangle : \ (3), \\ F_{00} + \sum\limits_{i=1}^{N_1} x_i(M) F_{i0} + \sum\limits_{j=1}^{N_2} y_j F_{0j} + \sum\limits_{i=1}^{N_1}\sum\limits_{j=1}^{N_2} x_i(M) y_j F_{ij} < 0. \end{cases} \tag{24}$$

Branching

Let $M = [p, q]$ be the candidate for further subdivision at a given iteration and let $(x(M), y(M), W(M))$ be an optimal solution of $LB(M)$ as before. In view of Theorem 1, if $x_i(M) \in \{p_i, q_i\}, i = 1, 2, ..., N_1$, then $(x(M), y(M), W(M))$ satisfies (6), i.e. $\beta(M)$ is the exact minimum of $CP(M)$ and thus provides the global minimum value of (7) in our BB process. Thus, to speed up the algorithm, we provide the following subdivision rule.
Subdivision rule. *Define*

$$\begin{array}{l} \mu_i = \min\{x_i(M) - p_i, q_i - x_i(M)\}, \ i = 1, 2, ..., N_1, \\ i_M = arg \max\limits_{i=1,2,...,N_1} \mu_i \end{array} \tag{25}$$

Subdivide M by the line $x_{i_M} = x_{i_M}(M)$.

We are now in a position to state our algorithm for solving the global optimal value of problem (1)-(4) with a given tolerance $\epsilon > 0$.

BMI optimization Algorithm

Initialization. Start with $M_0 = [a, b]$ and $\gamma^0 = +\infty$ and any feasible solution (x^0, y^0) of (3)-(4). Set $\mathcal{S}_1 = \mathcal{N}_1 = \{M_0\}$. Set $\kappa = 1$.

Step 1. For each $M \in \mathcal{N}_\kappa$ solve $LB(M)$ to obtain the optimal value $\beta(M)$ and an optimal solution $(x(M), y(M), W(M))$. Solve $UB(M)$ to update the best current value γ^κ with the corresponding current best solution (x^κ, y^κ).

Step 2. In $\mathcal{S}_\kappa$ delete all M such that $\beta(M) \geq \gamma^\kappa - \epsilon$. Let $\mathcal{R}_\kappa$ be the set of remaining rectangles. If $\mathcal{R}_\kappa = \emptyset$, terminate: γ^κ is the $\epsilon-$suboptimal value with the corresponding solution (x^κ, y^κ).

Step 3. Choose $M_\kappa \in \text{argmin}\{\beta(M) | \ M \in \mathcal{R}_\kappa\}$ and divide it into two smaller rectangles $M_{\kappa,1}, M_{\kappa,2}$ according to the above subdivision rule. Let $\mathcal{N}_{\kappa+1} = \{M_{\kappa,1}, M_{\kappa,2}\}$, $\mathcal{S}_{\kappa+1} = (\mathcal{R}_\kappa \setminus M_\kappa) \cup \mathcal{N}_{\kappa+1}$.

Set $\kappa \leftarrow \kappa + 1$ and go back to Step 1.

The (global) convergence of the above algorithm is proved in the following theorem.

Theorem 2 *The above algorithm will terminate after a finitely many iterations, yielding the $\epsilon-$suboptimal value of problem (7).*

Proof Suppose that the algorithm is infinite. Then by [21, Th. 5.5], the above subdivision rules guarantees that the algorithm generates a nested of rectangles $M_\nu = [p^\nu, q^\nu] \to M^* = [p^*, q^*]$ such that $x(M_\nu) \to x^*$ with $x_i^* \in \{p_i^*, q_i^*\}, i = 1, 2, ..., N_1$. Thus, as mentioned before $\beta(M^*) = \gamma(M^*) \geq \gamma^*$, where γ^* is the global minimal value of (1)-(4). But by construction procedure in Step 2, we have $\beta(M_\nu) < \gamma^* - \epsilon$ by which we infer $\beta(M^*) \leq \gamma^* - \epsilon$, a contradiction.

□

4 Remarks on the feasibility problems

An important particular case of BMI optimization problem (1)-(4) is the problem of checking the feasibility of system (2)-(4). This can be reduced to the form

$$\begin{aligned} \min\ t: \quad & F_{00} + \sum_{i=1}^{N_1} x_i F_{i0} + \sum_{j=1}^{N_2} y_j F_{0j} + \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} x_i y_j F_{ij} < tI, \\ & (3), (4) \end{aligned} \tag{26}$$

Obviously, the system (2)-(4) is feasible if and only if the optimal value t^* of (26) is negative. However, in contrast to the optimization problem (1)-(4), we can stop our search as soon as we can draw the conclusion $t^* < 0$ or $t^* > 0$, without reaching the exact value of t^*. Also, the following features can be exploited. For every M, $LB(M)$ is adjusted to

$$\begin{aligned} \min\ t: \quad & F_{00} + \sum_{i=1}^{N_1} x_i F_{i0} + \sum_{j=1}^{N_2} y_j F_{0j} + \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} w_{ij} F_{ij} < tI, \\ & (8), (10) \end{aligned} \tag{27}$$

Set

$$f(x, y) = \lambda_{\max}[F_{00} + \sum_{i=1}^{N_1} x_i F_{i0} + \sum_{j=1}^{N_2} y_j F_{0j} + \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} x_i y_j F_{ij}]$$

where $\lambda_{\max}[.]$ stands for the maximal eigenvalue of a symmetric matrix. It is obvious that the optimal solution $(x(M), y(M))$ of (27) is feasible for (3), (4) and by Proposition 2, $(\tilde{x}(M), y(M))$ with $\tilde{x}_i(M) = w_{ij}(M)/y_j(M)$ is feasible for (3) and (4) too. Thus we can use both $f(x(M), y(M))$ and $f(\tilde{x}(M), y(M))$ for updating the upper bound for BB process solving (26) instead of solving a LMI optimization problem like $UB(M)$.

5 Numerical examples

5.1 Linear Parameter-Varying system analysis

Consider the linear time-varying system

$$\dot{x} = A(t), \ A(t) \in \text{convex hull}\{A_1, A_2\} \tag{28}$$

with given matrices A_1, A_2 of dimension $n \times n$. In the last few years, linear time-varying as well as gain-scheduling systems have motivated new interests for developing tools (see e.g. [2] and references therein). The main issue here is how to choose an appropriate Lyapunov function for checking the stability of the system (28). It is known that fixed quadratic Lyapunov functions $V(x) = x'Px$ with some positive definite matrix P may lead to very conservative answers. In [3, 19] parameter-dependent Lyapunov functions have been used. This way, the stability analysis problem is recast as the feasibility problem of parameterized LMIs. An alternative method for the stability problem of (28) has been considered in [23] by means of piecewise quadratic Lyapunov functions of the forms

$$
\begin{aligned}
& V(x) = \max\{x'P_1x, x'P_2x\},\ P_1 > 0,\ P_2 > 0 && (29)\\
\text{or}\quad & V(x) = \min\{x'P_1x, x'P_2x\},\ P_1 > 0,\ P_2 > 0. && (30)
\end{aligned}
$$

Using so called $\mathcal{S}$-procedure with a variable reduction technique, it is shown [23] that system (28) is stable if and only if the following matrix inequality conditions with decision variables $P_1, P_2, \delta_1, \delta_2$ are feasible

$$
\begin{aligned}
(1-\delta_2)(A_1'P_2 + P_2A_1) + \delta_2(P_2 - P_1) < 0 && (31)\\
(1-\delta_1)(A_2'P_1 + P_1A_2) - \delta_1(P_2 - P_1) < 0, && (32)\\
A_1'P_1 + P_1A_1 < 0,\ A_2'P_2 + P_2A_2 < 0, \quad 0 < P_1 < I, \quad 0 < P_2 < I, && (33)\\
0 \le \delta_i \le 1,\ i = 1, 2 && (34)
\end{aligned}
$$

Now it is clear that (31)-(34) has the exact form of (2)-(4) where (δ_1, δ_2) are the complicating variables. Thus the nonconvexity rank of (31)-(34) is just 2 while the overall problem dimension is $n(n+1)+2$, thus much larger. Hence our BB algorithm is readily applied to solve (31)-(34) independently of the dimension n of matrices in (28), whereas most earlier algorithms of [8, 11, 5] have a trouble even for the smallest dimension $n = 2$ since they require branching performed in the space of dimension $n(n+1)+2 = 8$. It is also clear that rewriting (31)-(34) in form (2)-(5) for applying the results of [8, 11, 5] will destroy some useful LMI structure of (33).

Take the following data from [23]

$$
A_1 = \begin{bmatrix} 0 & 1 \\ -2 & -1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 1 \\ -2-k & -1 \end{bmatrix} \tag{35}
$$

In [23], the authors used an heuristic gridding method with sampling interval 0.1 for (δ_1, δ_2) to solve (31)-(34) and found the largest $k > 0$ such that the corresponding conditions (31)-(34) are feasible. $k = 4.7$ has been obtained this way with $(\delta_1, \delta_2) = (0.9, 0.8)$. It is obvious that such a method requires solving not only many LMI problems but also is likely to miss the optimal solutions. In contrast, our BB algorithm found, after a few iterations, that the maximal value of k is $k = 4.75$. For this k, a solution to (31)-(34) is

$$
\delta_1 = 0.85775, \qquad \delta_2 = 0.79578,
$$
$$
P_1 = \begin{bmatrix} 0.93375 & 0.16119 \\ 0.16119 & 0.23355 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 0.99311 & 0.07355 \\ 0.07355 & 0.21465 \end{bmatrix}
$$

Note that we have used the Matlab LMI Control Toolbox [7] for all LMI-related computations.

5.2 Structure/controller design examples

BMIs like (31)-(34) also arise naturally from the robust structure/controller design [20, 15] where the structure parameters (stiffness, spring and damping coefficients etc.) play the role of complicating variables. Consider the mass-spring-damper system described by Figure 1, where x is the position of mass, u is the control force input, w is the disturbance force, m, k and c are the mass, spring and damping coefficients satisfying

$$\begin{array}{l} 0.125k + c \geq 2,\ 0.125k + c \leq 2.5,\ (k - 8.5)m + 1 \leq 0,\ (c - 1.5)m \leq 0, \\ (k, c, m) \in H = [4, 12] \times [0.5, 1.5] \times [3, 5] \end{array} \tag{36}$$

Our aim is to attenuate the effect of disturbances on the mass in a mass-spring-damper arrangement. Note that (36) prevents increasing k, c while decreasing m simultaneously.

With the controlled output $z = [x, u]'$, the measurement output $y = x$, and the state vector $[x, \dot{x}]'$ the generalized plant is described as

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

where

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

Now, the problem can be stated as to find an internally stabilizing proper controller $K(s)$ and parameter $h \in H$ minimizing the H_∞ norm of the closed-loop transfer function T_{zw} from w to z. Note that this problem is harder than that in [15, 20] as it additionally includes the mass optimization requirement. From the main results in [6], this problem has the nonlinear matrix inequality formulation

$$\begin{array}{rl} \min\limits_{\gamma, m, k, c, \widehat{R}, S} & \gamma :\ (36) \\ & \mathcal{N}_1' \begin{bmatrix} A(k,c)\widehat{R}M + M\widehat{R}A(k,c)' & M\widehat{R}C_1' & B_1 \\ C_1\widehat{R}M & -\gamma I & 0 \\ B_1' & 0 & -\gamma I \end{bmatrix} \mathcal{N}_1 < 0 \\ & \mathcal{N}_2' \begin{bmatrix} A(k,c)'M^{-1}S + SM^{-1}A(k,c) & SM^{-1}B_1 & C_1' \\ B_1'M^{-1}S & -\gamma I & 0 \\ C_1 & 0 & -\gamma I \end{bmatrix} \mathcal{N}_2 < 0 \\ & \begin{bmatrix} \widehat{R} & I \\ I & S \end{bmatrix} \geq 0, \end{array} \tag{37}$$

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. The form (37) is not yet convenient for optimization purpose, so we perform the congruence transformation diag$[\,M \quad I\,]$ in the third inequality in (37) and the changes of variables

$$R = M\widehat{R}M, \quad \bar{m} = 1/m$$

to transform (37) to the problem

$$
\begin{array}{ll}
(\mathbf{P}) & \min_{\gamma,\bar{m},k,c,R,S} \gamma : \ (k,c,\bar{m}) \in \bar{H} = [4,12]\times[0.5,1.5]\times[0.2,1/3] \\
& 0.125k + c \geq 2,\ 0.125k + c \leq 2.5,\ k+\bar{m} \leq 8.5,\ c+\bar{m} \leq 1.5, \\
& \mathcal{N}_1' \begin{bmatrix} A_1(\bar{m},k,c)R + RA_1'(\bar{m},k,c) & RC_1' & B_1 \\ C_1R & -\gamma I & 0 \\ B_1' & 0 & -\gamma I \end{bmatrix} \mathcal{N}_1 < 0, \\
& \mathcal{N}_2' \begin{bmatrix} A_2(\bar{m},k,c)'S + SA_2(\bar{m},k,c) & S\bar{B}_1(\bar{m}) & C_1' \\ \bar{B}(\bar{m})_1'S & -\gamma I & 0 \\ C_1 & 0 & -\gamma I \end{bmatrix} \mathcal{N}_2 < 0 \\
& \qquad\qquad \begin{bmatrix} R & \bar{M}(1/\bar{m}) \\ \bar{M}(1/\bar{m}) & S \end{bmatrix} \geq 0 \\
& A_1(\bar{m},k,c) = \begin{bmatrix} 0 & \bar{m} \\ -k & -c\bar{m} \end{bmatrix},\ A_2(\bar{m},k,c) = \begin{bmatrix} 0 & 1 \\ -k\bar{m} & -c\bar{m} \end{bmatrix}, \\
& \bar{B}_1(\bar{m}) = \begin{bmatrix} 0 \\ \bar{m} \end{bmatrix},\ \bar{M}(1/\bar{m}) = \begin{bmatrix} 1 & 0 \\ 0 & 1/\bar{m} \end{bmatrix}.
\end{array} \tag{38}
$$

Obviously, the complicating variables of the above problem are $(\bar{m}, k, c)$, while the nonlinear terms arising in nonlinear matrix inequality (38) are

$$R_k = kR,\ R_{\bar{m}} = \bar{m}R, R_{c\bar{m}} = c\bar{m}R,\ S_{\bar{m}} = \bar{m}S,\ S_{c\bar{m}} = c\bar{m}S,\ S_{k\bar{m}} = k\bar{m}S, t = 1/\bar{m} \tag{39}$$

For a sub-rectangle $\tilde{H} = [k_1,k_2]\times[c_1,c_2]\times[\bar{m}_1,\bar{m}_2]$, we have to relax the nonconvex constraints caused by nonlinear terms (39) for computing a lower bound of $(\mathbf{P})$ over $\tilde{H}$. The nonlinear constraint $t = 1/\bar{m}$ can be relaxed according to [18] as

$$\begin{bmatrix} t & 1 \\ 1 & \bar{m} \end{bmatrix} \geq 0, \bar{m}_1\bar{m}_2 t + \bar{m} \leq \bar{m}_1 + \bar{m}_2. \tag{40}$$

On the other hand, without loss of generality, we can assume that the optimal solution R, S of $(\mathbf{P})$ with $(k,c,\bar{m}) \in \tilde{H}$ satisfies the "trust region" condition

$$||R - R_{\tilde{H}}|| \leq \nu_{\tilde{H}},\ ||S - S_{\tilde{H}}|| \leq \nu_{\tilde{H}} \tag{41}$$

with $\nu_{\tilde{H}}$ chosen depending on the size of $\tilde{H}$. Here $R_{\tilde{H}}, S_{\tilde{H}}$ is the solution of $(\mathbf{P})$ corresponding to $(\bar{m},k,c) = (\bar{m}_1+\bar{m}_2, k_1+k_2, c_1+c_2)/2$ which can be used for updating the current best value (upper bound) in our BB algorithm if they satisfies the constraint (36). Then nonconvex terms $R_k, R_{\bar{m}}, R_c = cR, S_k = kS, S_{\bar{m}} = \bar{m}S, S_c = cS$ arising in (39) are relaxed using Corollary 1, while $R_{c\bar{m}}$ according to Theorem 1 is relaxed to

$$
\begin{array}{l}
R_{c\bar{m}} - c_1R_{\bar{m}} - \bar{m}_1R_c + c_1\bar{m}_1R \geq 0 \\
R_{c\bar{m}} - c_2R_{\bar{m}} - \bar{m}_2R_c + c_2\bar{m}_2R \geq 0 \\
-R_{c\bar{m}} + c_1R_{\bar{m}} + \bar{m}_2R_c - c_1\bar{m}_2R \geq 0 \\
-R_{c\bar{m}} + c_2R_{\bar{m}} + \bar{m}_1R_c - c_2\bar{m}_1R \geq 0
\end{array} \tag{42}
$$

and it is analogous for $S_{c\bar{m}}, S_{k\bar{m}}$. The computational result for an optimal solution with tolerance $\epsilon = 0.01$ is given in Table 1. One can see the benefit of both spring-damper

problem	optimal (k, c, m)	opt. perform. γ	# of iter.
nominal plant	(8,1,4)	0.5791	1
spring-damper optimiz.	(8.2510,1.2510,4)	0.4885	15
mass-spring-damper optimiz.	(8.1595,1.1779,3.0953)	0.4659	29

Table 1: Computational results for mass-spring-damper system

($m = 4$ is fixed) and mass-spring-damper optimization. Note that the performance of our algorithm is much worse (i.e. requiring many more iterations) if the standard bisection rule is used instead of the subdivision rule (25).

Finally, let's consider another widely used system of two masses connected by a spring [22]

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k/m_1 & k/m_1 & 0 & 0 \\ k/m_2 & -k/m_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/m_1 \\ 0 \end{bmatrix} (u + \alpha_d d), \\ z &= [\alpha_x x_2 \quad u]', \\ y &= x_2 + \alpha_v v, \end{aligned} \tag{43}$$

with the designed parameters (k, m_1, m_2) satisfying the constraints

$$(k - 0.5)m_1 - 1 \le 0, (k - 3.5)m2 + 1 \le 0, \ (k, m_1, m_2) \in [0.5, 1.5] \times [0.5, 1.5] \times [0.25, 0.75]. \tag{44}$$

Here m_1 and m_2 are masses with position x_1, x_2, while k is the spring constant, u is the control force input, d is the plant disturbance, v is the process and sensor noises and y is the sensor measurement. The constants α_v and α_d represent the noise to signal ratios, while α_x is a constant weight. In contrast to the ACC benchmark problem considering k, m_1, m_2 as uncertainties [17], our problem is to find optimal parameters (k, m_1, m_2) in the set (44) and a stabilizing control $K(s)$ to regulate the position x_2 with minimal control energy, i.e. find (k, m_1, m_2) and $K(s)$ minimizing the H_∞ norm of $T_{zw}(s)$, transfer function from w to z. By a similar argument, this problem can be transformed to the form like (**P**) containing the nonlinear terms

$$m_1 R, \ kR, \ km_2 R, \ km_1 S, \ km_2 S\,,$$

which can be handled as in the previous mass-spring-damper example. Note that (44) prevents either increasing k, m_1 simultaneously or increasing k, m_1 while decreasing m_2 simultaneously. The computational results with $(\alpha_x, \alpha_v, \alpha_d) = (1, 0.1, 0.1)$ are given in Table 2. In the problem of optimizing (k, m_1) the value $m_2 = 0.5$ is held fixed. From Table 1 and 2 one can see that the number of iterations is sensitive only to the number of complicating variables (i.e. the dimension of the branching space).

problem	optimal (k, m_1, m_2)	opt. perform. γ	# of iter.
nominal	(1,1,0.5)	0.3907	1
optimiz. k, m_1	(1.4251,1.0785,0.5)	0.3440	21
(43)-(44)	(1.4635,1.0390,0.7642)	0.3396	34

Table 2: Computational results for two mass systems

6 Concluding remarks

A new BB algorithm for solving a general class of BMI problems is developed. A key idea has been to incorporate LMI constraints into BMI constraint in a special way to take advantage of useful informations on nonconvex terms. Moreover, the nonconvexity of BMI is concentrated in coupling constraints so when the latter are omitted, we get the relaxed LMI problem for computing lower bounds. As in the our previous developments, the branching is performed in the space of complicating variables only to guarantee the practicability of the algorithm even with a large dimension of overall variables. The viability of the proposed algorithm is confirmed by its applications to several problems of robust control.

Acknowledgments. We are greatly grateful to Professor Hoang Tuy for his frequent support and corrections during this work.

References

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

[2] P. Apkarian, R.J. Adams, Advanced gain-scheduling techniques for uncertain systems, *IEEEE Trans. Control Systems Technology* 6(1998), 21-32

[3] P. Apkarian, H.D. Tuan, Parameterized LMIs in control theory, To appear in *SIAM J. Control and Optimization.*

[4] P. Apkarian, H.D. Tuan, Robust control via concave optimization: local and global algorithms, To appear in *IEEE Trans. Automat. Control.*

[5] 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.

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

[7] P. Gahinet, A. Nemirovski, A. Laub, M. Chilali, *LMI control toolbox,* The Math. Works Inc..

[8] 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.

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

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

[11] S.M. Lim, G.P. Papavassilopoulos, Numerical experience with parallel algorithms for solving the BMI problems, *Proc. of 13-th IFAC Congress*, San Fransisco, 1996, pp. 387-392.

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

[13] G.P. McCormic, *Nonlinear programming: theory, algorithms and applications*, John Willey and Sons, 1982.

[14] 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.

[15] G. Shi, R.E. Skelton, An algorithm for integrated structure and control design with variance bound, *Proc. of 35th IEEE Conference on Decision and Control*, 1996, pp. 167-172.

[16] O. Slupphaug, On robust constrained nonlinear control and hybrid control: BMI and MPC-based state feedback schemes, *Ph.D. Thesis*, Department of Engineering Cybernetics, Norwegian university of Science and Technology, Norway, October 1998.

[17] H.D. Tuan, S. Hosoe, H. Tuy, New global optimization algorithms for solving the robust performance problem of robust controls, *Proceedings of 2-nd Asian Control Conference*,1997, Vol. 1, 350-353.

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

[19] H.D. Tuan, P. Apkarian, Relaxation of parameterized LMIs with control applications, *Int. J. of Nonlinear Robust Controls* 9(1999), 59-84.

[20] H.D. Tuan, P. Apkarian, Y. Nakashima, A new Lagrangian dual global optimization algorithm for solving bilinear matrix inequalities, *Proc. of 1999 American Control Conference,* 1851-1855; To appear in *Int. J. of Nonlinear Robust Controls.*

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

[22] B. Wie, D.S. Bernstein, A benchmark problem for robust controller design, *Proc. ACC*, 961-962, 1990.

[23] L. Xia, S. Shishkin, M. Fu, Piecewise Lyapunov functions for robust stability of linear time-varying systems, *Systems and Control Letters* 31(1997), 165-171.

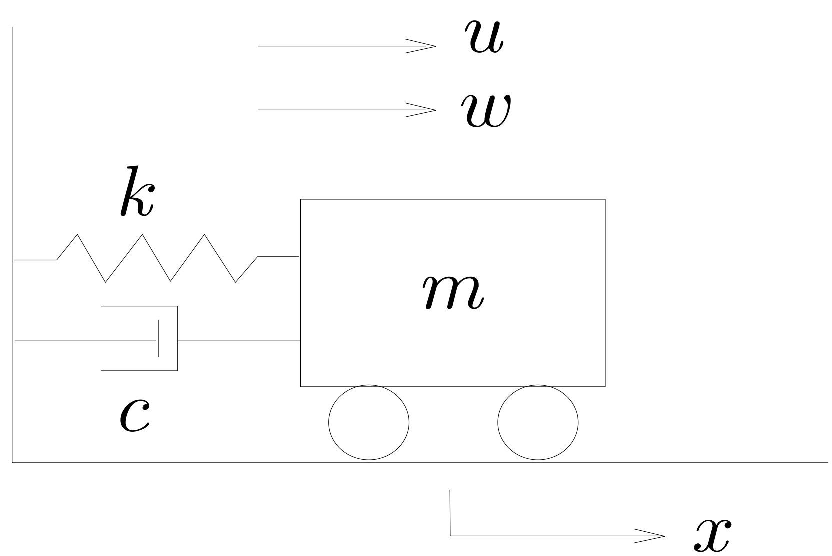

Figure 1: mass-spring-damper system