A Randomized Algorithm for Robust BMI Optimization

Takayuki Wada and Yasumasa Fujisaki

1 Graduate School of Information Science and Technology, Osaka University,
1–5 Yamadaoka, Suita, Osaka 565–0871, Japan
E-mail: {t-wada, fujisaki}@ist.osaka-u.ac.jp

Abstract
Robust bilinear matrix inequality (BMI) optimization, which is to minimize an objective function subject to a parameter dependent BMI constraint, is considered. A recursive algorithm employing a branch-and-bound technique and randomization of a parameter is provided for solving the problem. When the algorithm finds a solution, this solution satisfies the parameter dependent constraint with a prescribed accuracy in a probabilistic sense. Furthermore, the objective function value at that solution ensures that the feasible set whose objective function value is less than this value is too small to be found.

1 Introduction
Decision and control under uncertain environment can be recast as the robust optimization [1] which is to minimize an objective function subject to a parameter dependent constraint. For example, robust output feedback controller design [2], switching rule design of a linear switched system [3], operation planning of uncertain power systems [4] are rewritten as the robust optimization. However, the optimization is difficult due to its parameter dependency [5].

A randomization technique is one of the solutions for its difficulty [5]. For example, scenario approach [6] gives a standard convex optimization with many randomly sampled constraints. They show that an appropriate selection of the number of random samples leads to an approximate solution in a probabilistic sense. In addition to this, necessary number of random samples is of polynomial order of given accuracy parameters and the number of decision variables. This approach is extended to a robust nonconvex optimization in [7]. These researches clarify how many random samples is needed for finding an approximate solution with prescribed theoretical guarantee. However, these algorithms require us to solve a standard nonconvex optimization with many randomly sampled constraints. Furthermore, they do not provide any algorithm for solving this problem with some guarantee.

Another approach is a recursive algorithm employing randomly sampled constraints one by one. For a robust convex problem, there are some recursive type randomized algorithms [8] for robust convex optimization. It has also been extended to robust nonconvex feasibility problem [3, 9]. However, these algorithms are specialized for switching rule design of linear switched systems. Then, these do not deal with optimization problem.

In this paper, we consider a robust bilinear matrix inequality (BMI) optimization whose constraint is parameter dependent and bilinear with respect to decision variables. The BMI optimization is a standard form of nonconvex optimization [10–12] and this problem is its generalization. Our aim is to characterize difficulty of the problem from a viewpoint of randomized algorithms. To achieve this goal, we develop a randomized algorithm with guarantee on feasibility in a probabilistic sense. We employ branch-and-bound techniques [13]. Then, we show that this solution satisfies the constraint with a prescribed accuracy in a probabilistic sense. Furthermore, we clarify the volume of feasible set whose objective function value is less than the obtained objective function value is enough small with high probability. We hasten to note that these techniques do not lead to a solution within a reasonable computational time due to NP-hardness of the BMI optimization [14].

We have already developed a randomized algorithm for robust nonconvex problem with parameter dependent decision variable [4]. This algorithm utilizes a full randomization technique, that is, it employs random sampled constraints and random sampled decision variables. It stops within finite number of iterations and these numbers are of polynomial of the problem size. Furthermore, we can provide guarantee on feasibility and optimality in a probabilistic sense. However, in general, the feasible set whose objective function value is enough small to find its element by using the full randomization technique. In this case, our randomized branch-and-bound algorithm could find a solution.

This paper is organized as follows. In Section 2, we formulate the robust BMI optimization. In Section 3, we show a randomized branch-and-bound algorithm. A numerical example is shown in Section 4. Finally, we provide concluding remarks in Section 5.

2 Problem Formulation
Let us consider a robust bilinear matrix inequality (BMI) optimization:

\[
\min_{x \in X, y \in Y} \quad c^T x + c^T y
\]

s.t. \( F(x, y, \delta) \leq 0 \quad \forall \delta \in \Delta, \quad (1) \)

where \( x \in X \subseteq \mathbb{R}^n \) and \( y \in Y \subseteq \mathbb{R}^m \) are decision variables, \( c_x \in \mathbb{R}^n \) and \( c_y \in \mathbb{R}^m \) are coefficients of the objective function, and \( \delta \in \Delta \subseteq \mathbb{R}^d \) is an uncertain parameter. The BMI
constraint \( F(x, y, \delta) \leq 0 \) is defined by
\[
F(x, y, \delta) = F_{00}(\delta) + \sum_{i=1}^{n_x} x_i F_{i0}(\delta) + \sum_{j=1}^{n_y} y_j F_{0j}(\delta)
+ \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} w_{ij} F_{ij}(\delta) \leq 0,
\]
where \( F_{ij}(\delta) = F_{ij}^\text{F}(\delta) \) for \( i = 0, 1, 2, \ldots, n_x \) and \( j = 0, 1, 2, \ldots, n_y \). Additionally, \( F_{ij}(\delta) \) could be nonlinear in \( \delta \in \Delta \). In this paper, we assume that the sets \( X \) and \( Y \) are hyper cubes which are defined by
\[
X = [\underline{x}_1, \overline{x}_1] \times [\underline{x}_2, \overline{x}_2] \times \ldots \times [\underline{x}_n, \overline{x}_n],
Y = [\underline{y}_1, \overline{y}_1] \times [\underline{y}_2, \overline{y}_2] \times \ldots \times [\underline{y}_n, \overline{y}_n],
\]
where \( \underline{x}_i \leq \overline{x}_i \) and \( \underline{y}_j \leq \overline{y}_j \). We can also deal with other general sets such as ellipsoids, polytopes, and so on, if we add appropriate constraints to \( F(x, y, \delta) \leq 0 \).

Even if there is no uncertainty, it is difficult to find a solution within a reasonable computing time. Furthermore, the problem is also difficult from a viewpoint of its parameter dependency. This is because we cannot verify whether given candidate solution \((x, y, \delta)\) is feasible, or not, within finite computational time.

3 A Randomized Branch-and-Bound Algorithm for Robust BMI Optimization

Now, we consider a randomized branch-and-bound algorithm for robust BMI optimization. To employ a branch-and-bound technique, we have to prepare for algorithms for finding upper and lower bounds of the optimal value of the problem (1) within a subset of \( X \times Y \).

3.1 Lower bounds

It is well known that a linear matrix inequality (LMI) relaxation approach for the standard BMI optimization leads to a lower bound of the optimal value of the original problem. Now, we apply this technique for the robust BMI optimization. An LMI relaxation of a parameter dependent BMI constraint is given by
\[
F_L(x, y, W, \delta) = F_{00}(\delta) + \sum_{i=1}^{n_x} x_i F_{i0}(\delta) + \sum_{j=1}^{n_y} y_j F_{0j}(\delta)
+ \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} w_{ij} F_{ij}(\delta) \leq 0,
\]
where \( w_{ij} \in \mathbb{R} \) is the \((i, j)\) element of a matrix \( W \in \mathbb{R}^{n_x \times n_y} \). If \( F(x, y, \delta) \leq 0 \) for a fixed \( \delta \in \Delta \), then \( F_L(x, y, x^T, \delta) \leq 0 \) holds. Since \( x \) and \( y(\delta) \) belong to hyper cubes \( X \) and \( Y \) respectively, \((x, y, \delta)\) has to satisfy
\[
-w_{ij} + x_i y_j + x_j y_i - x_i y_j \leq 0,
-w_{ij} - x_i y_j - x_j y_i + x_i y_j \leq 0,
-w_{ij} - x_i y_j + x_j y_i - x_i y_j \leq 0,
-w_{ij} + x_i y_j + x_j y_i - x_i y_j \leq 0,
\]
for any \( i = 1, 2, \ldots, n_x \) and \( j = 1, 2, \ldots, n_y \). Thus, the LMI relaxation of the problem (1) is
\[
\begin{aligned}
&\min c^T x + c^T y \\
\text{s.t.} &\quad F_L(x, y, W, \delta) \leq 0 \\
&\quad w_{ij} + x_i y_j + x_j y_i - x_i y_j \leq 0 \\
&\quad w_{ij} - x_i y_j - x_j y_i + x_i y_j \leq 0 \\
&\quad w_{ij} - x_i y_j + x_j y_i - x_i y_j \leq 0 \\
&\quad w_{ij} + x_i y_j + x_j y_i - x_i y_j \leq 0 \\
&\quad \forall \delta \in \Delta.
\end{aligned}
\]

Please notice that the relaxation problem (2) is still difficult due to its parameter dependency. Now, we introduce the probability measure \( P_s \) on the set \( \Delta \). Then, we employ randomization techniques for solving the problem (2) in a probabilistic sense. We summarize an algorithm in Appendix.

3.2 Upper bounds

In this section, we introduce a full randomization algorithm for finding an upper bound of the robust BMI optimization within a subset \( X_s \times Y_s \) of \( X \times Y \). We assume that \( X_s \) and \( Y_s \) are also hyper cubes. We employ randomization of all decision variables \( x \) and \( y \) and an uncertain parameter \( \delta \) from the sets \( X_s \), \( Y_s \), and \( \Delta \), respectively. To generate random samples, we introduce a probability measure \( P_s \) on \( \Delta \) and the uniform probability measure \( P_{y_s} \) over \( X_s \times Y_s \).

An algorithm is shown in Algorithm 1. The algorithm stops within a finite number of random samples from the sets \( X_s \), \( Y_s \), and \( \Delta \). In fact, the maximum number of random samples from the set \( X_s \times Y_s \) is \( k \ell \) and that from the set \( \Delta \) is \( \ell \). When we select appropriate numbers \( k \) and \( \ell \), we can guarantee quality of an obtained solution.

Algorithm 1 A randomized algorithm without branching

Require: \( \ell \in \mathbb{N} \) and \( \ell \in \mathbb{N} \)
0: Set \( \hat{x} := \infty, \hat{\delta} := \text{Null}, \) and \( \check{x} := \text{Null} \);
1: for \( \ell := 1 \) to \( \ell \) do
2: Draw \((x^{(\ell)}, y^{(\ell)})\) from \( X_s \times Y_s \) according to \( P_{y_s} \);
3: for \( k := 1 \) to \( k \) do
4: Draw \( a^{(k)} \) from \( P_s \);
5: if \( F(x^{(\ell)}, y^{(\ell)}, a^{(k)}) \geq 0 \) or \( \check{\delta} \leq c^T x^{(\ell)} + c^T y^{(\ell)} \) then
6: goto Step 10;
7: end if
8: end for
9: Update \( \check{x} := c^T x^{(\ell)} + c^T y^{(\ell)}, \hat{x} := x^{(\ell)}, \hat{\delta} := y^{(\ell)} \);
10: end for
11: return \((\check{x}, \hat{x}, \hat{\delta})\).

Lemma 1 ([4]) For given \( \alpha_a \in (0, 1), \beta_a \in (0, 1), \alpha_b \in (0, 1), \) and \( \beta_b \in (0, 1), \) select \( k \in \mathbb{N} \) and \( \ell \in \mathbb{N} \) such that
\[
\check{k} \geq \ln \frac{7}{\beta_a} \ln \frac{1}{1 - \alpha_a}, \quad \ell \geq \ln \frac{1}{\beta_b} \ln \frac{1}{1 - \alpha_b}.
\]

Then, the following statements hold.
1. The probability that \((\hat{x}, \hat{y})\) satisfies
\[
P_{d}(\delta \in \Delta : F(\hat{x}, \hat{y}, \delta) \leq 0 \leq 1 - \alpha_{c} \text{ and } \hat{x} \neq \text{Null}) \leq 1 - \alpha_{b}.
\]
is less than or equal to \(\beta_{b}\).

2. The probability that this problem solution
\[
P_{s}(x, y) \in X \times Y : c_{T}^{T}x + c_{R}^{T}y \leq \hat{y},
\]
\[
F(x, y, \delta) \leq 0, \forall \delta \in \Delta \leq 1 - \alpha_{b}
\]
is less than or equal to \(\beta_{b}\).

That is, obtained solution \((\hat{x}, \hat{y})\) is a feasible solution in a probabilistic sense with high probability \(1 - \beta_{b}\). This implies that \((\hat{x}, \hat{y})\) is a feasible solution in a probabilistic sense with the same probability level. We can also see that objective function value at \((\hat{x}, \hat{y})\) is an upper bound of the optimal value of the problem (1) with this probability. Furthermore, the objective function value \(\hat{y}\) at \((\hat{x}, \hat{y})\) is an approximate minimum in a probabilistic sense with probability \(1 - \beta_{b}\).

Implementation of the algorithm is easy because of simplicity of the algorithm. This is one of the advantages of the algorithm. However, since the feasible set of the problem is generally too small, the algorithm cannot find a solution in many cases. To solve this issue, we employ a branch-and-bound technique.

### 3.3 A randomized branch-and-bound algorithm

Now, we show a randomized branch-and-bound algorithm for the problem (1). We utilize Algorithms 1 and 3 for finding upper and lower bounds. In Algorithm 2, we prepare for a provisional optimal value \(\hat{y}\) and a provisional optimal solution \((\hat{x}, \hat{y})\). Furthermore, we construct a subproblem set \(S\). Then, we iterate the following procedures. We first extract a subproblem \((C, \hat{y}(C))\) from \(S\) such that the lower bound of the optimal value of the problem (1) within \(C\) attains minimum among all subproblems in \(S\).

Then, we split the set \(C\) into two distinct sets \(C_{1}\) and \(C_{2}\) such that \(C = C_{1} \cup C_{2}\). We solve the original problems (1) and the relaxation problems (2) in \(C_{1}\) and \(C_{2}\), respectively. Based on the results, upper and lower bounds of the optimal value are updated. If obtained lower bound is greater than the current upper bound, we prune this region. This is because there is no optimal value in this region. Otherwise, we further subdivide this region into the problem set \(S\). In this loop, if the upper bound \(\hat{y}\) and the minimum lower bound \(L_{1}(\kappa)\) of the optimal value in the solution set \(S\) become close enough, the algorithm stops with \(\hat{y}\) and \((\hat{x}, \hat{y})\) as outputs. Properties of the algorithm are as follows.

**Theorem 1** For given \(\alpha \in (0, 1), \beta \in (0, 1), \varepsilon \in (0, \infty), \) and \(r_{m} \in (0, r_{S})\), select \(\tilde{L}_{1}(\kappa) \in \mathbb{N}, \tilde{L}_{2}(\kappa) \in \mathbb{N}, \tilde{L}_{1}(\kappa) \in \mathbb{N} \) and \(\tilde{L}_{2}(\kappa) \in \mathbb{N} \) such that
\[
\tilde{L}_{1}(\kappa) \geq \left( \frac{2 \ln \pi \kappa + \ln \left( \frac{1}{6\beta} \right)}{L_{1}(\kappa)} \right) \ln \frac{1}{1 - \alpha},
\]
\[
\tilde{L}_{2}(\kappa) \geq \left( \frac{2 \ln \pi \kappa + \ln \left( \frac{1}{6\beta} \right)}{L_{1}(\kappa)} \right) \ln \frac{1}{1 - \alpha},
\]
\[
\tilde{L}_{2}(\kappa) \geq \max \left( \frac{50(n_{x} + n_{y})}{\kappa}, \frac{n_{x} n_{y} \kappa^{2} \exp \left( \frac{104 \ln \left( r_{m} / r_{S} \right)}{25} \right)}{\kappa} \right),
\]

where \(r_{m}\) is the maximum radius of a ball in \(X \times Y\) and \(r_{S}\) is the maximum radius of the feasible set of the problem 1. Then, the following statements hold.

1. The probability that \(P_{o}(\delta \in \Delta : F(\hat{x}, \hat{y}, \delta) \leq 0 \leq 1 - \alpha) \leq 1 - \alpha_{b} \) or \((\hat{x}, \hat{y}) \neq (\text{Null}, \text{Null})\) holds is less than or equal to \(\beta\).

2. When \(\hat{y} \neq \text{Null}\), \(\hat{y}_{L} \leq \hat{y} \leq \hat{y}_{U} + \epsilon\) holds.

3. The probability that
\[
\hat{y}_{L} \leq \gamma^{*} + \mu
\]
holds is greater than 1 - \(\beta\), where \(\gamma^{*}\) is the optimal solution of the problem (1) and \(\mu\) is a tolerance which is defined by
\[
\mu = \max_{(x, y) \in S} \frac{\max_{(x, y) \in S} (c_{T}^{T}x + c_{R}^{T}y) - \min_{(x, y) \in S} (c_{T}^{T}x + c_{R}^{T}y)}{r_{m}}.
\]

**Proof:** Let us consider two events, \(F_{d}\) and \(B_{d}\):

\(F_{d}: \) At the \(k\)-th executions of Algorithm 1, the \(\ell\)-th random sampled pair of \(x^{(\ell)}(\kappa)\) and \(y^{(\ell)}(\kappa)\) satisfies \(\tilde{L}_{1}(\kappa)\) random sampled constraints, that is,
\[
F(x^{(\ell)}(\kappa), y^{(\ell)}(\kappa), \delta_{k}) \leq 0, \quad k = 1, 2, \ldots, \tilde{L}_{1}(\kappa)
\]

Note that this event implies the event \(\{\hat{x} \neq \text{Null}\}\).

\(B_{d}: \) A pair of \(\hat{x}(\kappa)\) and \(\hat{y}(\kappa)\) satisfies
\[
P_{d}(\delta \in \Delta : F(\hat{x}(\kappa), \hat{y}(\kappa), \delta) \leq 0) \leq 1 - \alpha.
\]

That is, a pair of \(\hat{x}(\kappa)\) and \(\hat{y}(\kappa)\) is not a probabilistic solution with a given accuracy \(\alpha\).

Our aim is to show the probability that any pair of \(\hat{x}(\kappa)\) and \(\hat{y}(\kappa)\), \(\kappa = 1, 2, \ldots\), which is output of the algorithm is not a probabilistic solution is less than or equal to \(\beta\). That is, we try to show
\[
(\cap_{\kappa=1}^{\infty} P_{s}\cap) \hat{y}(\kappa) \leq \beta
\]
holds.

For each \(\kappa = 1, 2, \ldots, \tilde{L}_{1}(\kappa)\),
\[
P_{s}(\hat{x}^{(\ell)}(\kappa)) \leq \sum_{\ell=1}^{\tilde{L}_{1}(\kappa)} \left( \frac{1}{\alpha} \hat{y}(\kappa) \right)
\]

\[
\leq \left( \frac{1}{\alpha} \hat{y}(\kappa) \right)
\]

\[
(1 - \alpha) \hat{y}(\kappa)
\]
since each \( \hat{L} \) we obtain
\[
\sum_{k=1}^{\infty} \ell_k(\alpha) (1 - \alpha \gamma) \leq \sum_{k=1}^{\infty} \frac{6\beta}{\pi^2 k^2} = \beta,
\]
we obtain
\[
\left( \mathcal{P}_C P_{\beta}^{\gamma}(\alpha), \hat{L}(\alpha) \right)^{\infty} \left( \bigcup_{k=1}^{\infty} \left( \mathcal{F}_k \cap \mathcal{B}_k \right) \right) \leq \beta
\]
which shows the statement 1.
2. We can see that this statement holds due to a property of branch and bound techniques.
3. When we select \( \bar{k}_2(\alpha) \) and \( \bar{k}_2(\alpha) \) according to (5) and (6), Lemma 2 in Appendix implies that \( \hat{L} \) satisfies
\[
\hat{L} \leq \hat{L}_2(\alpha) + \frac{\max \left( e_i^T x + c_i^T y \right) - \min \left( e_i^T x + c_i^T y \right)}{r_L(C)}
\]
with probability \( 1 - 6\beta/(\pi^2 k^2) \), where \( \gamma \leq (C) \) is the optimal value of the problem (2) over \( C \), \( \hat{L} \) is its lower bound given at Step 5 of Algorithm 2 and \( r_L(C) \) is the largest radius of a ball within the set
\[
S(C) = \{(x, y, W) \in C : F_L(x, y, W, \rho) \leq 0, \forall \rho \in \Delta \}
\]
That is,
\[
r_L(C) = \max_{x, y, W} \{ (x, y, W) : (x - x_0)^T (x - x_0) + (y - y_0)^T (y - y_0) \}
\]
Since each \( \hat{L} \) satisfies the above inequality with probability \( 1 - 6\beta/(\pi^2 k^2) \) the probability that all of \( \hat{L} \) satisfy the above inequality is greater than or equal to
\[
1 - \sum_{k=1}^{\infty} \frac{6\beta}{\pi^2 k^2} \leq 1 - \beta.
\]
Notice that \( \hat{L} \) is the minimum in \( S \). That is,
\[
\hat{L} \leq \min_{(C, \gamma) \in S} \left\{ \gamma \left( \frac{\max \left( e_i^T x + c_i^T y \right) - \min \left( e_i^T x + c_i^T y \right)}{r_L(C)} \right) \right\}
\]
holds with probability \( 1 - \beta \). Then, when we define the set
\[
S' = S \setminus \left( \bigcup_{(C, \gamma) \in S} C \right),
\]
we cannot find a feasible solution in \( S' \). Thus, we can interpret \( \gamma(S') = \infty \). Replacing \( \min_{(C, \gamma) \in S} \gamma(C) \) with \( \gamma \), we therefore see that
\[
\hat{L} \leq \gamma + \max_{(C, \gamma) \in S} \frac{\max \left( e_i^T x + c_i^T y \right) - \min \left( e_i^T x + c_i^T y \right)}{r_L(C)} \leq \gamma + \frac{\max \left( e_i^T x + c_i^T y \right) - \min \left( e_i^T x + c_i^T y \right)}{r_L(C)} r_m
\]
holds with probability \( 1 - \beta \).
4 A Numerical Example

In this section, we solve a simple robust BMI optimization which is given by

\[ \min \gamma \text{ s.t. } F(x, y, \delta) = F_{00}(\delta) + F_{10}(\delta)x + F_{01}(\delta)y + F_{11}(\delta)xy \leq \gamma I, \quad \forall \delta \in \Delta. \]

That is, \( x_1 = 1 \) and \( n_x = 1 \). This model is taken from [10]. The nominal values of the coefficient matrices are

\[
F_{00} = \begin{bmatrix}
-10 & -0.5 & -2 \\
-0.5 & 4.5 & 0 \\
-2 & 0 & 0
\end{bmatrix}, \quad
F_{10} = \begin{bmatrix}
9 & 0.5 & 0 \\
-0.5 & 0 & -3 \\
0 & -3 & -1
\end{bmatrix},
\]

\[
F_{01} = \begin{bmatrix}
-1.8 & -0.1 & -0.4 \\
-0.1 & 1.2 & -1 \\
-0.5 & -1 & 0
\end{bmatrix}, \quad
F_{11} = \begin{bmatrix}
0 & 0 & 2 \\
0 & -5.5 & 3 \\
2 & 3 & 0
\end{bmatrix}.
\]

All parameters are allowed to vary 10\% around these nominal values. We have set the set \( X \) and \( Y \) according to the paper [10]:

\[ \{(x, y, \gamma) : -0.5 \leq x \leq 2, \quad -3 \leq y \leq 7, \quad -2 \leq \gamma \leq 2\} \]

We have selected parameters as \( \alpha = 0.01, \beta = 0.01, \varepsilon = 0.01 \), and \( n_{\max} = 1.0 \times 10^4 \). We have obtained the following solution \( (\hat{x}, \hat{y}, \hat{\gamma}) = (-0.1913, 1.0476, 1.4639) \). In this example, we have generated about 800,000 \( \tilde{u}(k) \) random samples from the set \((x, y)\) at Step 9 in Algorithm 2. On the other hand, \( \bar{u}(k) \) was about 2,500. By employing posteriori analysis, we have examined this solution satisfies the statements of the theorem with probability 1 – 0.001. Furthermore, when we have employed Algorithm 1 for solving the problem (1), we have found 0.1725 which is clearly infeasible. In this case, selecting \( a_x = a_y = a_y = b_y = 0.01 \), the maximum number of random samples is about 500,000. Even if the full randomization technique cannot find a solution, our algorithm could find a solution.

5 Concluding Remarks

We have proposed a branch-and-bound randomized algorithm for robust BMI optimization. When the algorithm stops with the output, the obtained solution satisfies a parameter dependent constraint with high probability. Furthermore, it is shown that the obtained objective function value is close to optimal value in a probabilistic sense. In fact, the volume of feasible set whose objective function value is less than the obtained suboptimal value is enough small with high probability. In addition to this, we have shown that erroneous pruning in the branch-and-bound algorithm occurs with small probability.

Acknowledgement This work was partially supported by JSPS KAKENHI Numbers JP26420416 and JP15K18088.

References

[1] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski: Robust Optimization, Princeton Series in Applied Mathematics, Princeton University Press 2009.

[2] S. Kanev, C. Scherer, M. Verhaegen, and B. De Schutter: Robust output-feedback controller design via local BMI optimization, Automatica, Vol. 40, pp. 1115–1127 2004.

[3] H. Ishii, T. Başar, and R. Tempo: Randomized algorithms for synthesis of switching rules for multimodal systems, IEEE Transactions on Automatic Control, Vol. 50, No. 6, pp. 754–767 2005.

[4] T. Wada, R. Morita, T. Asai, I. Masubuchi, and Y. Fujisaki: Randomized solutions of robust optimal power flow, Proceedings of the 53rd IEEE Conference on Decision and Control, pp. 5195–5200 2014.

[5] R. Tempo, G. Calafiore, and F. Dabbene: Randomized Algorithms for Analysis and Control of Uncertain Systems with Applications, Communications and Control Engineering, Springer, second edition 2013.

[6] G. Calafiore and M. C. Campi: The scenario approach to robust control design, IEEE Transactions on Automatic Control, Vol. 51, No. 5, pp. 742–753 2006.

[7] S. Grammatico, X. Zhang, K. Margellos, P. Goulart, and J. Lygeros: A scenario approach for non-convex control design, IEEE Transactions on Automatic Control, Vol. 61, No. 2, pp. 334–345 2016.

[8] T. Wada and Y. Fujisaki: Sequential randomized algorithms for robust convex optimization, IEEE Transactions on Automatic Control, Vol. 60, No. 12, pp. 3356–3361 2015.

[9] T. Wada and Y. Fujisaki: Probabilistic design of switched systems, Proceedings of European Control Conference 2007, pp. 2611–2618 2007.

[10] K.-C. Goh, M. G. Safonov, and G. P. Papavassilopoulos: Global optimization for the biaffine matrix inequality problem, Journal of Global Optimization, Vol. 7, pp. 365–380 1995.

[11] A. Hassibi, J. How, and S. Boyd: A path-following method for solving BMI problems in control, Proceedings of American Control Conference, pp. 1385–1389 1999.

[12] M. Fukuda and M. Kojima: Branch-and-cut algorithms for the bilinear matrix inequality eigenvalue problem, Computational Optimization and Applications, Vol. 19, pp. 79–105 2001.

[13] H. D. Sherali and C. H. Tuncbilek: A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique, Journal of Global Optimization, Vol. 2, No. 1, pp. 101–112 1991.

[14] O. Toker and H. Özbay: On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback, Proceedings of American Control Conference, pp. 2525–2526 1995.

A A Randomized Algorithm for Robust Convex Optimization

In this section, we summarize a randomized algorithm for robust convex optimization [8]. We consider the LMI relaxation problem (2). To solve this problem, we introduce a
scalar function

\[ f_L(x, y, W, \delta) = [F_L(x, y, W, \delta)]^T + p(x, y, W, \delta) + q(x, y, W, \delta) + r(x, y, W, \delta) + s(x, y, W, \delta), \]

where \([\cdot]^T\) is the projection onto the cone of positive semidefinite matrices [5] and \(p, q, r, s\) are defined by

\[
p(x, y, W, \delta) = \text{diag}(p_{11}, p_{12}, \ldots, p_{1n}, p_{21}, p_{22}, \ldots, p_{2n}),
\]

\[
q(x, y, W, \delta) = \text{diag}(q_{11}, q_{12}, \ldots, q_{1n}, q_{21}, q_{22}, \ldots, q_{2n}),
\]

\[
r(x, y, W, \delta) = \text{diag}(r_{11}, r_{12}, \ldots, r_{1n}, r_{21}, r_{22}, \ldots, r_{2n}),
\]

\[
s(x, y, W, \delta) = \text{diag}(s_{11}, s_{12}, \ldots, s_{1n}, s_{21}, s_{22}, \ldots, s_{2n}),
\]

where \(\delta \in [0, \delta]\) and \(\alpha, \ell \in \mathbb{N}\). The analytic center is the minimum point of the logarithmic barrier

Algorithm 3 Randomized algorithm for robust convex optimization

1: Set \(\hat{\gamma} := \text{Null} \) and \(\hat{\varepsilon} := \text{Null}\). Select \(k \in \mathbb{N}, \ell \in \mathbb{N}\). Set \(z_0\) as the center of the hyper cube \(\mathbb{Z}\). Set the initial barrier function as

\[ B_0(z) := -\sum_{i=1}^{2(n_1+n_2+n_3)} \ln(b_i - a_i^Tz); \]

2: for \(\ell := 1 \) to \(\ell \) do
3: Draw \(\delta_k\) from the set \(\Delta\) according to the probability measure \(P_\delta\);
4: if \(f_L(z_{\ell-1}, \delta_k) \neq 0\) then
5: Compute a subgradient \(g := \partial L(z_{\ell-1}, \delta_k)\) and goto Update;
6: else if \(\hat{\gamma} \neq \text{Null}\) and \(c^Tz_{\ell-1} \geq \hat{\gamma}\) then
7: Set the gradient \(g := c\) of the objective function and goto Update;
8: end if
9: end for
10: Compute the gradient \(g := c\) of the objective function.
11: Update the provisional optimal value \(\hat{\gamma}\) and the provisional optimal solution \(\hat{z}\) as \(\hat{\gamma} := c^Tz_{\ell-1}\) and \(\hat{z} := z_{\ell-1}\);
12: Update: Update the candidate solution and the barrier function as

\[ z_{\ell} := \min B_\ell(z), \]

\[ B_\ell(z) := B_{\ell-1}(z) - \ln(b_{2(n_1+n_2+n_3)\ell} - a_{2(n_1+n_2+n_3)\ell}^Tz_{\ell-1}) - \ln(b_{2(n_1+n_2+n_3)\ell} - a_{2(n_1+n_2+n_3)\ell}^Tz_{\ell-1}). \]

Then, if \(\hat{\gamma} \neq \text{Null}\), the probability that \(\hat{\gamma}, \hat{z}, \hat{\gamma}\) and \(\hat{W}\) satisfies

\[ P_{\delta}(f_L(\hat{\gamma}, \hat{z}, \hat{W}, \delta) \leq 0) \geq 1 - \alpha_n \]

is greater than or equal to \(1 - \beta\), where \(\gamma_n\) is the optimal value of the chance constrained problem

\[
\min c^T x + c^T y \\
\text{s.t. } P_{\delta}(f_L(x, y, W, \delta) \leq 0) \geq 1 - \alpha_n.
\]