Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Parallel BMI Optimization using Unimodal Normal Distribution Crossover GA with Reduced-order Individual Expression ⋆

Michihiro Kawanishi * **Masaya Handa** **

* *Toyota Technological Institute, Tempaku, Nagoya 468-8511, Japan*
(Tel: +81-52-809-1815; e-mail: kawa@toyota-ti.ac.jp).
** *Kobe University, 1-1 Rokkodai-cho, Nada, Kobe 657-8501, Japan*

Abstract: This paper deals with a global optimization of BMIEP (Bilinear Matrix Inequalities Eigenvalue Problem) based on parallel UNDX (Unimodal Normal Distribution Crossover) GA. The several efficient conventional techniques for real-coded GA are also incorporated. Considering the BMIEP characteristic properties, we introduce a reduced-order individual expression with alternating SDP evaluation. The alternating SDP evaluation needs much computational burden compared to conventional eigenvalue evaluation. In order to reduce the computational burden, we consider the parallelization of the algorithm. Numerical experiments are carried out to confirm the effectiveness of proposed algorithms.

1. INTRODUCTION

Many control system design problems can be naturally described by Bilinear Matrix Inequalities (BMIs) (Goh et al., 1994; Safonov et al., 1994). Accordingly, efficient algorithms for solving BMIs have been studied by many researchers. Obtaining the exact solutions of BMIs is difficult because BMI optimization problem is in the class of NP hard. The problem becomes difficult to solve as the number of variables becomes large. The efficiency of currently available algorithms is not enough for solving practical BMI problems on control system design.

In this research, we consider to develop a practical BMI optimization algorithms based on real-coded genetic algorithms (RCGA) (Davis, 1990; Wright, 1991; Eshelman and Schaffer, 1993; Eshleman et al., 1997) for solving controller design problems. To this end, we apply "Unimodal Normal Distribution Crossover (UNDX)" (Ono and Kobayashi, 1997; Kita et al., 1998) to our BMI algorithms. The UNDX is one of the most efficient crossover methods for RCGA. We also apply other efficient GA techniques including "Extrapolation-directed Crossover (EDX)" (Sakuma and Kobayashi, 2001), "Minimal Generation Gap (MGG) model" (Satoh et al., 1996), and "Innately Split Model (ISM)" (Ikeda and Kobayashi, 2000) to achieve higher optimization performance.

Considering the problem structure of BMIs, we propose an BMI RCGA using a primary search direction with relaxed LMI (Kawanishi and Ikuyama, 2005), reduced-order individual expression, and alternating SDP (Semidefinite Programming) evaluation. Since alternating SDP evaluation for each individuals needs much computation amount, we here also consider the parallelization of the algorithm. Then, finally, we confirm the effectiveness of the proposed methods by numerical experiments.

⋆ This work was supported by KAKENHI (18560435) Grant-in-Aid for Scientific Research (C) (Sponsor and financial support acknowledgment goes The Ministry of Education, Culture, Sports, Science and Technology (MEXT)).

2. PRELIMINARY

In this section, we summarize the notations and describe the problem formulation.

2.1 BMI Problem

Given real valued vectors $\boldsymbol{x} \in \mathcal{R}^{n_x}$ and $\boldsymbol{y} \in \mathcal{R}^{n_y}$, we define a biaffine matrix valued function $F : \mathcal{R}^{n_x} \times \mathcal{R}^{n_y} \to \mathcal{R}^{m \times m}$ by

$$F(\boldsymbol{x}, \boldsymbol{y}) = F_{00} + \sum_{i=1}^{n_x} x_i F_{i0} + \sum_{j=1}^{n_y} y_j F_{0j} + \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} x_i y_j F_{ij} \ , \quad (1)$$

where x_i is the i-th element of $\boldsymbol{x}$, y_j is the j-th element of $\boldsymbol{y}$ and $F_{ij} = F_{ij}^T \in \mathcal{R}^{m \times m}$. Define the sets of indices as $\mathcal{I} := \{1, 2, \cdots, n_x\}$ and $\mathcal{J} := \{1, 2, \cdots, n_y\}$.

Many control problems can be naturally described by BMI (Bilinear Matrix Inequalities) as $F(\boldsymbol{x}, \boldsymbol{y}) \prec 0$. So, it is crucial to evaluate the feasibility of $F(\boldsymbol{x}, \boldsymbol{y}) \prec 0$ (BMI Feasibility Problem). It is also important to minimize a linear objective function $\boldsymbol{c}_x^T \boldsymbol{x} + \boldsymbol{c}_y^T \boldsymbol{y}$ under BMI constraints (BMI Linear Objectives Optimization Problem) (Goh et al., 1994; Safonov et al., 1994).

Since these BMI problems are transformed into a BMI eigenvalue problem (BMIEP), we must obtain the optimal solution $\lambda_{\rm opt}$ of BMIEP. Therefore, in this paper, we consider the global optimization of this non-convex BMIEP.

Given a closed hyper-rectangle $\mathcal{X} \times \mathcal{Y}$;

$$\mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_{n_x}, \qquad \mathcal{X}_i := [L_{x_i}, U_{x_i}], \quad (2)$$

$$\mathcal{Y} := \mathcal{Y}_1 \times \mathcal{Y}_2 \times \cdots \times \mathcal{Y}_{n_y}, \qquad \mathcal{Y}_i := [L_{y_i}, U_{y_i}] \ , \quad (3)$$

where $-\infty < L_{x_i} \leq U_{x_i} < \infty \quad (i = 1, \cdots, n_x)$ and $-\infty < L_{y_j} \leq U_{y_j} < \infty \quad (j = 1, \cdots, n_y)$, we can describe the BMIEP (BMI Eigenvalue Problem) as

$$\min_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{X}\times\mathcal{Y}} \overline{\lambda}\{F(\boldsymbol{x},\boldsymbol{y})\} =: \lambda_{\rm opt} \tag{4}$$

where $\overline{\lambda}(\cdot)$ denotes the maximal eigenvalue.

Defining a Linear Matrix valued function $F_L : \mathcal{R}^{n_x} \times \mathcal{R}^{n_y} \times \mathcal{R}^{n_x \times n_y} \to \mathcal{R}^{m \times m}$;

$$F_L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W}) = F_{00} + \sum_{i=1}^{n_x} x_i F_{i0} + \sum_{j=1}^{n_y} y_j F_{0j} + \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} w_{ij} F_{ij} \ , \tag{5}$$

where w_{ij} denotes the (i,j)-th element of $\boldsymbol{W}$, we obtain the lower bound of BMIEP $\lambda_{\rm Lopt}$ $(\leq \lambda_{\rm opt})$ by

$$\lambda_{\rm Lopt} = \min_{(\boldsymbol{x},\boldsymbol{y},{\rm vec}(\boldsymbol{W}))\in\mathcal{X}\times\mathcal{Y}\times\mathcal{W}} \overline{\lambda}\{F_L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{W})\} \ , \tag{6}$$

$$\mathcal{W} := \left\{ \begin{array}{l} {\rm vec}(\boldsymbol{W}) \ \mid \ (\boldsymbol{x},\boldsymbol{y}) \in \mathcal{X}\times\mathcal{Y}, \\ \left.\begin{array}{l} w_{ij} \geq L_{y_j} x_i + L_{x_i} y_j - L_{y_j} L_{x_i} \\ w_{ij} \geq U_{y_j} x_i + U_{x_i} y_j - U_{y_j} U_{x_i} \\ w_{ij} \leq U_{y_j} x_i + L_{x_i} y_j - L_{y_j} U_{x_i} \\ w_{ij} \leq L_{y_j} x_i + U_{x_i} y_j - U_{y_j} L_{x_i} \\ i = 1, \cdots, n_x, \quad j = 1, \cdots, n_y \end{array}\right\}, \end{array} \tag{7}$$

where ${\rm vec}(\boldsymbol{W})$ indicate $[w_{11} \ \ w_{21} \cdots w_{n_x 1} \ \ w_{12} \cdots w_{n_x n_x}]^{\rm T}$ (Fujioka and Hoshijima, 1997).

The lower bound of BMIEP is described by LMI (Linear Matrix Inequalities) optimization problem that is calculated by efficient interior point methods (Karmarker methods).

2.2 Real-coded Genetic Algorithm

Real-coded genetic algorithms (RCGA) (Davis, 1990; Wright, 1991; Eshelman and Schaffer, 1993; Eshleman et al., 1997) is suitable for BMIEP optimization compared to binary-coded genetic algorithms. In this subsection, we summarize several key notions and techniques for our BMI RCGA.

First of all, we define an individual, i.e. a real valued vector, as

$$\boldsymbol{g}_i = {\rm col}(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{X}\times\mathcal{Y}, \tag{8}$$

where ${\rm col}(\boldsymbol{x},\boldsymbol{y})$ indicate $[\boldsymbol{x}^{\rm T} \ \boldsymbol{y}^{\rm T}]^{\rm T}$. We define a population $\mathcal{P}$ consisting of $n_{\rm pop}$ individuals as

$$\mathcal{P} := \{\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_{n_{\rm pop}}\}. \tag{9}$$

Since the generations are alternated in GA procedure, we denote a population at the k-th generation by

$$\mathcal{P}(k) := \{\boldsymbol{g}_1^{(k)}, \boldsymbol{g}_2^{(k)}, \cdots, \boldsymbol{g}_{n_{\rm pop}}^{(k)}\}. \tag{10}$$

We also define an evaluation function $\Lambda(\boldsymbol{g}_i)$ and a fitness function $f_{\mathcal{P}}(\boldsymbol{g}_i) > 0$ for an individual $\boldsymbol{g}_i$ as

$$\begin{aligned} \Lambda(\boldsymbol{g}_i) &:= \overline{\lambda}\{F(\boldsymbol{x}_i, \boldsymbol{y}_i)\}, \\ f_{\mathcal{P}}(\boldsymbol{g}_i) &:= \max_{\boldsymbol{g}_j\in\mathcal{P}} \Lambda(\boldsymbol{g}_j) - \Lambda(\boldsymbol{g}_i). \end{aligned} \tag{11}$$

Note the value of the fitness function $f_{\mathcal{P}}(\boldsymbol{g}_i)$ depends on the population $\mathcal{P}$. This fitness function $f_{\mathcal{P}}(\boldsymbol{g}_i)$ is only used for roulette wheel selection procedure.

Unimodal normal distribution crossover (UNDX) (Ono and Kobayashi, 1997) is reported to demonstrate higher performance than the other crossover methods in RCGA. The UNDX procedure is summarized as follows;

(1) Select 3 individuals as parents $\boldsymbol{g}_{p1}, \boldsymbol{g}_{p2}, \boldsymbol{g}_{p3} \in \mathcal{P}$, where $\boldsymbol{g}_{p1}, \boldsymbol{g}_{p2}$ are main parents, and $\boldsymbol{g}_{p3}$ is a sub parent.
(2) Define the middle point of main parents $\boldsymbol{g}_m := \frac{(\boldsymbol{g}_{p1}+\boldsymbol{g}_{p2})}{2}$.
(3) Let the direction from $\boldsymbol{g}_{p1}$ to $\boldsymbol{g}_{p2}$ be primary search direction $\boldsymbol{d}$, i.e. $\boldsymbol{d} := \boldsymbol{g}_{p2} - \boldsymbol{g}_{p1}$.
(4) Define D as the distance from $\boldsymbol{g}_{p3}$ to the primary search direction $\boldsymbol{d}$.
(5) Define $\boldsymbol{e}_i$ be the orthogonal basis vectors spanning the subspace perpendicular to primary search direction $\boldsymbol{d}$.
(6) Two children $\boldsymbol{g}_{c1}$ and $\boldsymbol{g}_{c2}$ are now generated as follows;

$$\boldsymbol{g}_{c1} = \boldsymbol{g}_m + \xi\boldsymbol{d} + D\sum_{i=1}^{n-1} \eta_i \boldsymbol{e}_i, \tag{12}$$

$$\boldsymbol{g}_{c2} = \boldsymbol{g}_m - \xi\boldsymbol{d} - D\sum_{i=1}^{n-1} \eta_i \boldsymbol{e}_i, \tag{13}$$

$$\xi \sim \ N(0, \sigma_\xi^2), \quad \eta_i \sim \ N(0, \sigma_\eta^2), \tag{14}$$

where n is a dimension of search space, $N(a,v)$ represents a normal distribution whose average is a and variance is v, σ_ξ and σ_η are constant parameters that are recommended to be set as $\frac{1}{2}|\boldsymbol{d}|$ and $\frac{0.35D}{\sqrt{n}}$ respectively. We describe the UNDX procedure as $(\boldsymbol{g}_{c1}, \boldsymbol{g}_{c2}) = {\rm UNDX}(\boldsymbol{g}_{p1}, \boldsymbol{g}_{p2}, \boldsymbol{g}_{p3})$. □

The values of σ_ξ and σ_η were firstly decided based on numerical experiments heuristically in Ono and Kobayashi (1997). After the research, the theoretical optimality of these values is confirmed later, which preserve the stochastic properties of the parent population, i.e. average, variance and covariance (Kita et al., 1998).

For the selection of an individual from populations, there exist several selection methods. One of the standard selection methods is Roulette Wheel Selection method. In roulette wheel selection procedure, an individual $\boldsymbol{g}_i$ is selected from a population $\mathcal{P}$ by the probability ${\rm P}[\boldsymbol{g}_i]$;

$${\rm P}[\boldsymbol{g}_i] := \frac{f_{\mathcal{P}}(\boldsymbol{g}_i)}{\sum_{i=1}^{n_{\rm pop}} f_{\mathcal{P}}(\boldsymbol{g}_i)}. \tag{15}$$

We denote an individual $\boldsymbol{g}_i$ that is selected by the roulette wheel selection as $\boldsymbol{g}_i = {\rm RWS}(\mathcal{P})$ □

One of the most important key notions for GA is the preservation of the diversity of the population. MGG (Minimal Generation Gap) Model (Satoh et al., 1996) is

appropriate for keeping the diversity of the population. The MGG model consists of the following five steps.

(1) Generate an initial population.
(2) Select a pair of individuals randomly from the population as parents.
(3) Generate $2n_{\rm cross}$ offspring by carrying out $n_{\rm cross}$ times crossover,
(4) Select two individuals from the family containing the two main parents and their $2n_{\rm cross}$ offspring. One is the best individual and the other is an individual selected by the roulette wheel selection (Goldberg, 1989). Replace the two main parents in step (2) with the two individuals.
(5) Repeat the procedure from step (2) to step (4) until a stopping condition is satisfied.

There exists sampling bias on unimodal normal distribution crossover. The algorithm has the tendency to mainly search in the center area of the search space. This property may cause deceptive phenomena and leads to failure of GA search in promising valley. In order to relax the sampling bias, we introduce extrapolation-directed crossover(EDX) (Sakuma and Kobayashi, 2001). EDX is summarized as follows;

(1) [UNDX] step (1) and (3)∼(5). In addition, suppose $\Lambda(\boldsymbol{g}_{p1}) < \Lambda(\boldsymbol{g}_{p2})$.
(2) With EDX, the child $\boldsymbol{g}_c$ is now generated as follows;

$$\boldsymbol{g}_c = \boldsymbol{g}_{p1} + D\sum_{i=1}^{n-1} v_i \boldsymbol{e}_i, \tag{16}$$

$$v_i \sim \ N(0, \sigma_\eta^2), \tag{17}$$

where n is a dimension of search space, σ_η is also recommended to be set by $\frac{0.35D}{\sqrt{n}}$. We describe the EDX procedure as $\boldsymbol{g}_c = \mathrm{EDX}(\boldsymbol{g}_{p1}, \boldsymbol{g}_{p2}, \boldsymbol{g}_{p3})$. □

If there exists a promising solution in steep valley (V-valley), it might be missed. This kind of search failure is called as deceptive phenomena. In order to overcome the deceptive phenomena, concentrative search scheme is needed. To this end, we introduce Innately Split Model (ISM) (Ikeda and Kobayashi, 2000) as follows.

(1) Carry out the optimization with small multi population.
(2) Initialize each population in specified small search area.
(3) Independently, carry out crossover and natural selection in each small population.
(4) When there exist multi populations that search the same valley, eliminate populations except one.
(5) Eliminate the populations whose fitness function value cannot be improved for long term.

3. GENETIC ALGORITHM FOR BMIEP

In this section, we propose a real-coded genetic algorithm for BMIEP. The proposed algorithm is carefully designed considering the structure of BMIEP.

3.1 GA for BMIEP with conventional techniques

In order to overcome the difficulties caused by GA-hard structure, i.e. big-valley, UV-valley and ridge line structure, we here utilize UNDX with MGG model, EDX and ISM techniques. A UNDX RCGA algorithm for BMIEP is described in the following **Algorithm 1**;

[**Algorithm 1**]
0 Set $k = 0$ and $p_{\rm EDX}$ $(0 < p_{\rm EDX} \le 1)$.
1 Set p $(0 < p \le 1)$, $c_{x_i} \in \mathcal{X}_i$ and $c_{y_j} \in \mathcal{Y}_j$ randomly.
2 Generate an initial population $\mathcal{P}(k)$ randomly.
$\mathcal{P}(k) := \{\boldsymbol{g}_1^{(k)}, \boldsymbol{g}_2^{(k)}, \cdots, \boldsymbol{g}_\ell^{(k)} \cdots \boldsymbol{g}_{n_{\rm pop}}^{(k)}\}$,
$\boldsymbol{g}_\ell^{(k)} := (\boldsymbol{x}_\ell^{(k)}, \boldsymbol{y}_\ell^{(k)})$,
$x_{\ell i}^{(k)} \in [pL_{x_i} - c_{x_i}\ pU_{x_i} - c_{x_i}] \cap \mathcal{X}_i,\ i \in \mathcal{I}$
$y_{\ell j}^{(k)} \in [pL_{y_j} - c_{y_j}\ pU_{y_j} - c_{y_j}] \cap \mathcal{Y}_j,\ j \in \mathcal{J}$
3 Select $\boldsymbol{g}_{p1}, \boldsymbol{g}_{p2}, \boldsymbol{g}_{p3}$ from the population $\mathcal{P}(k)$ randomly. Suppose $\Lambda(\boldsymbol{g}_{p1}) < \Lambda(\boldsymbol{g}_{p2})$. Eliminate the main parents $\mathcal{P}(k) := \mathcal{P}(k) \backslash \{\boldsymbol{g}_{p1}, \boldsymbol{g}_{p2}\}$.
4 In probability $p_{\rm EDX}$, carry out EDX for $n_{\rm EDX}$ times.
for $i = 1 : n_{\rm EDX}$
Calculate $\boldsymbol{g}_c = \mathrm{EDX}(\boldsymbol{g}_{p1}, \boldsymbol{g}_{p2}, \boldsymbol{g}_{p3})$.
If $\Lambda(\boldsymbol{g}_c) < \Lambda(\boldsymbol{g}_{p1})$ holds, replace $\boldsymbol{g}_{p1}$ by $\boldsymbol{g}_c$.
End for. Go to Step 8.
5 Carry out UNDX for $n_{\rm cross}$ times.
Initialize $\mathcal{P}_{\rm undx}(k) := \{\boldsymbol{g}_{p1}, \boldsymbol{g}_{p2}\}$
for $i = 1 : n_{\rm cross}$
$(\boldsymbol{g}_{c1}, \boldsymbol{g}_{c2}) = \mathrm{UNDX}(\boldsymbol{g}_{p1}, \boldsymbol{g}_{p2}, \boldsymbol{g}_{p3})$
$\mathcal{P}_{\rm undx}(k) := \mathcal{P}_{\rm undx}(k) \cup \{\boldsymbol{g}_{c1}, \boldsymbol{g}_{c2}\}$
End for.
6 Replace $\boldsymbol{g}_{p1}$ by the best individual in $\mathcal{P}_{\rm undx}(k)$.
7 Replace $\boldsymbol{g}_{p2}$ by $\boldsymbol{g}_{\rm rws} = \mathrm{RWS}(\mathcal{P}_{\rm undx}(k))$
8 $\mathcal{P}(k) := \mathcal{P}(k) \cup \{\boldsymbol{g}_{p1}, \boldsymbol{g}_{p2}\}$ and Let $k := k + 1$.
9 Repeat step 3 - 8 until convergence criterion is satisfied.
10 Repeat step 1 - 9 until stopping criterion is satisfied. (ISM procedure)

where $x_{\ell i}^{(k)}$ indicates the i-th element of the vector $\boldsymbol{x}_\ell^{(k)}$. The populations $\mathcal{P}(k)$ and $\mathcal{P}_{\rm undx}(k)$ consist of $n_{\rm pop}$ and $2n_{\rm cross}$ individuals respectively.

Steps 1-2 and 10 in Algorithm 1 are for ISM. EDX is carried out in Steps 3-4. UNDX is executed in step 5. Steps 6-8 are for the implementation of MGG Model.

3.2 A new primal search direction with relaxed LMIs

In this subsection, we state a new primary search direction which is proposed in Kawanishi and Ikuyama (2005). Suppose $\mathcal{P} := \{\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_\ell, \cdots, \boldsymbol{g}_{n_{\rm pop}}\}$ and $\boldsymbol{g}_\ell := \mathrm{col}(\boldsymbol{x}_\ell, \boldsymbol{y}_\ell)$. First, we define a region $\mathcal{Q}_x \times \mathcal{Q}_y$ including all individuals of the population $\mathcal{P}$ as follows.

$$L_{x_i} := \min_{\ell=1,\cdots,n_{\rm pop}} x_{\ell i},\quad U_{x_i} := \max_{\ell=1,\cdots,n_{\rm pop}} x_{\ell i}$$
$$\mathcal{Q}_{x_i} := [L_{x_i}\ \ U_{x_i}] \subset \mathcal{X}_i,\ i \in \mathcal{I}$$
$$\mathcal{Q}_x := \mathcal{Q}_{x_1} \times \mathcal{Q}_{x_2} \times \cdots \times \mathcal{Q}_{x_{n_x}}$$

$$L_{y_j} := \min_{\ell=1,\cdots,n_{\rm pop}} y_{\ell j},\quad U_{y_j} := \max_{\ell=1,\cdots,n_{\rm pop}} y_{\ell j}$$

$$\mathcal{Q}_{y_j} := [L_{y_j} \;\; U_{y_j}] \subset \mathcal{Y}_j,\ j \in \mathcal{J}$$
$$\mathcal{Q}_y := \mathcal{Q}_{y_1} \times \mathcal{Q}_{y_2} \times \cdots \times \mathcal{Q}_{y_{n_y}}$$

where $x_{\ell i}$ indicates the i-th element of a vector $\boldsymbol{x}_\ell$ and $y_{\ell j}$ indicates the j-th element of a vector $\boldsymbol{y}_\ell$. The region $\mathcal{Q}_x \times \mathcal{Q}_y$ represents a minimal orthogonal hyper rectangle containing the population $\mathcal{P}$. Replacing $\mathcal{X} \times \mathcal{Y}$ by $\mathcal{Q}_x \times \mathcal{Q}_y$, we define $\mathcal{W}_\mathcal{Q}$ by (7). Carrying out (6), we can define a special individual $\boldsymbol{g}_{\rm lmi}$ as follows;

$$\boldsymbol{g}_{\rm lmi} := \arg_{(\boldsymbol{x},\boldsymbol{y})} \min_{\mathcal{Q}_x \times \mathcal{Q}_y \times \mathcal{W}_\mathcal{Q}} F_L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W}) \tag{18}$$
$$=: \mathrm{LMI}(\mathcal{P}), \tag{19}$$

where $\arg_{(\boldsymbol{x},\boldsymbol{y})}$ denotes the partial arguments $(\boldsymbol{x}, \boldsymbol{y})$ of the entire arguments $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{W})$. Now, considering the structure of BMIEP, a new effective primal search direction is obtained by

$$\boldsymbol{d}_{\rm lmi} := \boldsymbol{g}_{\rm lmi} - \boldsymbol{g}_i. \tag{20}$$

In view point of preserving the diversity of populations, we do not directly apply the new primary search direction. Along with the procedure of MGG model that is superior for keeping the population diversity, we utilize the new primary search direction as follows.

[Algorithm 2]
0 [Algorithm 1] step 0 - 6
7 Replace $\boldsymbol{g}_{p2}$ by $\boldsymbol{g}_{\rm lmi} = \mathrm{LMI}(\mathcal{P}_{\rm undx}(k))$.
8 [Algorithm 1] step 8 - 10

In the proposed algorithm, the new primary search direction $\boldsymbol{d}_{\rm lmi}$ is exploited for generating local populations $\mathcal{P}_{\rm undx}(k)$. Only in case that the individual $\boldsymbol{g}_{\rm lmi}$ is selected in the selection procedure, the new primary search direction is applied and becomes valid.

3.3 Reduced-order search with alternating SDP evaluation

Here, we consider a genetic algorithm in the reduced-order space. Without loss of generality, we can suppose $n_x < n_y$ on variables vector $\boldsymbol{x} \in \mathcal{R}^{n_x}$ and $\boldsymbol{y} \in \mathcal{R}^{n_y}$. We here define individuals $\boldsymbol{h}_i$ by

$$\boldsymbol{h}_i = \boldsymbol{x} \in \mathcal{X}. \tag{21}$$

Note that the search space dimension is reduced to n_x from $n_x + n_y$ by this individual expression. In this case, since the variable vector $\boldsymbol{y}$ is not specified by the individuals $\boldsymbol{h}_i$, we cannot evaluate $\Lambda(\boldsymbol{h}_i) = \overline{\lambda}\{F(\boldsymbol{x}, \cdot)\}$. To evaluate the reduced-order individuals, we must also define a new evaluation function. To this end, we here consider to utilize the bi-convex property of BMIEP. The following SDP (Semi-definit programming) evaluation enables us to evaluate the reduced-order individuals $\boldsymbol{h}_i$;

$$\min_{\boldsymbol{y} \in \mathcal{Y}, \lambda} \lambda \qquad \text{s.t.} \quad \lambda I - F(\boldsymbol{h}_i, \boldsymbol{y}) \succ 0 \tag{22}$$

Moreover, in order to evaluate the individuals more exactly, we consider to utilize the alternating SDP method. A new evaluation function $\Gamma(\boldsymbol{h}_i)$ for the reduced-order individuals is defined as follows;

1. Fix $\delta > 0$. Set $k = 0$.
2. Set $(\boldsymbol{y}^{(0)}, \lambda_{g_i}^{(k)}) := \arg \min_{\boldsymbol{y} \in \mathcal{Y}, \lambda} \{\lambda : \lambda I - F(\boldsymbol{h}_i, \boldsymbol{y}) \succ 0\}$.
3. Repeat {
4. $(\boldsymbol{x}^{(k+1)}, \lambda_{g_i}^{(k)}) = \arg \min_{\boldsymbol{x} \in \mathcal{X}, \lambda} \{\lambda : \lambda I - F(\boldsymbol{x}, \boldsymbol{y}^{(k)}) \succ 0\}$
5. $(\boldsymbol{y}^{(k+1)}, \lambda_{g_i}^{(k)}) = \arg \min_{\boldsymbol{y} \in \mathcal{Y}, \lambda} \{\lambda : \lambda I - F(\boldsymbol{x}^{(k+1)}, \boldsymbol{y}) \succ 0\}$
6. $k = k + 1$.
7. }until $\lambda_{g_i}^{(k-1)} - \lambda_{g_i}^{(k)} < \delta |\lambda_{g_i}^{(k)}|$
8. $\Gamma(\boldsymbol{h}_i) := \lambda_{g_i}^{(k)}$

Now, we are in position to state our new reduced-order search algorithms as follows;

[Algorithm 3]
0 [Algorithm 1] step 0
1 Set p $(0 < p \leq 1)$ and $c_{x_i} \in \mathcal{X}_i$ randomly.
2 Generate an initial population $\mathcal{P}(k)$ randomly.
$\mathcal{P}(k) := \{\boldsymbol{h}_1^{(k)}, \boldsymbol{h}_2^{(k)}, \cdots, \boldsymbol{h}_\ell^{(k)} \cdots \boldsymbol{h}_{n_{\rm pop}}^{(k)}\}$, $\boldsymbol{h}_\ell^{(k)} := \boldsymbol{x}_\ell^{(k)}$, $x_{\ell i}^{(k)} \in [pL_{x_i} - c_{x_i}\ pU_{x_i} - c_{x_i}] \cap \mathcal{X}_i,\ i \in \mathcal{I}$
3 Select $\boldsymbol{h}_{p1}, \boldsymbol{h}_{p2}, \boldsymbol{h}_{p3}$ from the population $\mathcal{P}(k)$ randomly. Suppose $\Gamma(\boldsymbol{h}_{p1}) < \Gamma(\boldsymbol{h}_{p2})$. Eliminate the main parents $\mathcal{P}(k) := \mathcal{P}(k) \backslash \{\boldsymbol{h}_{p1}, \boldsymbol{h}_{p2}\}$.
4 In probability $p_{\rm EDX}$, carry out EDX for $n_{\rm EDX}$ times.
for $i = 1 : n_{\rm EDX}$
Calculate $\boldsymbol{h}_c = \mathrm{EDX}(\boldsymbol{h}_{p1}, \boldsymbol{h}_{p2}, \boldsymbol{h}_{p3})$.
If $\Gamma(\boldsymbol{h}_c) < \Gamma(\boldsymbol{h}_{p1})$ holds, replace $\boldsymbol{h}_{p1}$ by $\boldsymbol{h}_c$.
End for. Go to Step 8.
5 Carry out UNDX for $n_{\rm cross}$ times.
Initialize $\mathcal{P}_{\rm undx}(k) := \{\boldsymbol{h}_{p1}, \boldsymbol{h}_{p2}\}$
for $i = 1 : n_{\rm cross}$
$(\boldsymbol{h}_{c1}, \boldsymbol{h}_{c2}) = \mathrm{UNDX}(\boldsymbol{h}_{p1}, \boldsymbol{h}_{p2}, \boldsymbol{h}_{p3})$
$\mathcal{P}_{\rm undx}(k) := \mathcal{P}_{\rm undx}(k) \cup \{\boldsymbol{h}_{c1}, \boldsymbol{h}_{c2}\}$
End for.
6 Replace $\boldsymbol{h}_{p1}$ by the best individual in $\mathcal{P}_{\rm undx}(k)$.
7 Replace $\boldsymbol{h}_{p2}$ by $\boldsymbol{h}_{\rm rws} = \mathrm{RWS}(\mathcal{P}_{\rm undx}(k))$
8 $\mathcal{P}(k) := \mathcal{P}(k) \cup \{\boldsymbol{h}_{p1}, \boldsymbol{h}_{p2}\}$ and Let $k := k + 1$.
9 [Algorithm 1] step 9 - 10

In Algorithm 3, all full-order individuals $\boldsymbol{g}_i$ in Algorithm 1 are replaced by reduced-order individuals $\boldsymbol{h}_i$. The evaluation function $\Lambda(\cdot)$ in Algorithm 1 is also replaced by the avaluation function $\Gamma(\cdot)$ with alternating SDP method. For the calculation of $\Gamma(\cdot)$, many LMI iterations are needed. The utilization of $\Gamma(\cdot)$ impose us heavy computational burden compared to the conventional eigenvalue-based evaluation function $\Lambda(\cdot)$. In order to reduce the computational burden, we consider the parallelization of the algorithm in the next section.

4. PARALLEL ALGORITHM

In Algorithm 3, we need to carry out many LMI optimizations for evaluating individuals with alternating SDP evaluation method. However, these calculations can be efficiently parallelizable. The parallelized evaluation function $\Gamma_{\rm para}(\cdot)$ is now defined as follows;

$$\{\lambda_1, \lambda_2, \cdots, \lambda_n\} = \Gamma_{\rm para}(\{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n\}), \tag{23}$$
$$\lambda_i := \Gamma(\boldsymbol{h}_i), \quad i = 1, 2, \cdots, n, \tag{24}$$

where each evaluation $\Gamma(\boldsymbol{h}_i)$ is calculated simultaneously at each worker nodes. An algorithm of the master node is summarized as follows;

[Parallelized Algorithm 3]

0 [Algorithm 3] step 0-2
3 Select $\boldsymbol{h}_{p1}, \boldsymbol{h}_{p2}, \boldsymbol{h}_{p3}$ from the population $\mathcal{P}(k)$ randomly. Calculate $\{\lambda_{p1}, \lambda_{p2}\} = \Gamma_{\rm para}(\{\boldsymbol{h}_{p1}, \boldsymbol{h}_{p2}\})$. Suppose $\lambda_{p1} < \lambda_{p2}$. Eliminate the main parents $\mathcal{P}(k) := \mathcal{P}(k) \backslash \{\boldsymbol{h}_{p1}, \boldsymbol{h}_{p2}\}$.
4 In probability $p_{\rm EDX}$, carry out EDX for $n_{\rm EDX}$ times.
  for $i = 1 : n_{\rm EDX}$
    Calculate $\boldsymbol{h}_{ci} = {\rm EDX}(\boldsymbol{h}_{p1}, \boldsymbol{h}_{p2}, \boldsymbol{h}_{p3})$.
  End for.
  $\{\lambda_{c1}, \cdots, \lambda_{cn_{\rm EDX}}\} := \Gamma_{\rm para}(\{\boldsymbol{h}_{c1}, \boldsymbol{h}_{c2}, \cdots, \boldsymbol{h}_{cn_{\rm EDX}}\})$.
  If $\min_i \lambda_{ci} < \Gamma(\boldsymbol{h}_{p1})$ holds, replace $\boldsymbol{h}_{p1}$ by $\boldsymbol{h}_{ci}$.
  Go to Step 8.
5 [Algorithm 3] step 5
6 Replace $\boldsymbol{h}_{p1}$ by $\boldsymbol{h}_{\rm best} := \arg\min \Gamma_{\rm para}(\mathcal{P}_{\rm undx}(k))$.
7 [Algorithm 1] step 7 - 10

In step 4 and 6 in parallelized Algorithm 3, we evaluate all individuals in the populations where the population in step 4 consists of $n_{\rm EDX}$ individuals and the population $\mathcal{P}_{\rm undx}(k)$ in step 6 consists of $2n_{\rm cross}$ individuals. In the evaluation function $\Gamma_{\rm para}$, these individuals are distributed to each worker nodes and simultaneously evaluated at each nodes.

5. NUMERICAL EXPERIMENTS

In this section, the effectiveness of the proposed methods is evaluated by numerical experiments.

5.1 Reduced-order search

We here consider an example of Helicopter stabilizing problem with static output feedback control firstly described in Keel et al. (1988). The problem is briefly summarized as follows. The dynamics of the Helicopter is represented by

$$\begin{cases} \dot{\boldsymbol{x}} = A\boldsymbol{x} + Bu \\ y = C\boldsymbol{x} \end{cases}, \tag{25}$$

$$\left[\begin{array}{cc} A & B \\ C & 0 \end{array}\right] = \left[\begin{array}{cccc|cc} -0.0366 & 0.0271 & 0.0188 & -0.4555 & 0.4422 & 0.1761 \\ 0.0482 & -1.0100 & 0.0024 & -0.4555 & 3.5446 & -7.5922 \\ 0.1002 & 0.3681 & -0.7070 & 1.4200 & -5.5200 & 4.4900 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{26}$$

The stabilizing problem with static output feedback is described by BMI;

$$\Phi(K, X) := \begin{bmatrix} (A+BKC)^{\rm T}X + X(A+BKC) & 0 \\ 0 & -X \end{bmatrix} < 0. \tag{27}$$

If $\min_{K,X} \overline{\lambda}\{\Phi(K, X)\} < 0$ holds, the static output feedback gain K stabilizes the closed-loop control system. By transforming (27) to the standard BMI form (1), we can see $n_x = 2$, $n_y = 10$ and $m = 8$.

The numerical parameters are set by $p = 0.1$, $n_{\rm pop} = 30$, $p_{\rm EDX} = 0.5$, $n_{\rm EDX} = 30$ and $n_{\rm cross} = 15$. The region of variables is defined by $L_{x_i} = L_{y_j} = -100$ and $U_{x_i} = U_{y_j} = 100$. If the progress in 100 generations is less than 10^{-7}, we terminate the algorithm. The termination parameter in alternating SDP evaluation method is set to $\delta = 0.02$. CPU of our computational environment for this numerical experiments is 1.83 GHz AMD Sempron 2600+. The size of memory is 512 MB. We carry out the numerical calculations 10 times for each algorithm to evaluate the average performance and the worst case (robustness). Our experimental results are summarized in Table 1 and Table 2.

Table 1. Results for Helicopter Problem without ISM (one iteration)

Alg.#	Alg.1	Alg.2	Alg.3
Max	68.2060	8.4712	-12.5800
Average	32.4101	-2.6770	-12.5800
Min	5.6964	-12.2429	-12.5801
CPU time [s]	9.4	14.5	386.1

Table 2. Results for Helicopter Problem with ISM in 650 s CPU time

Alg.#	Alg.1	Alg.2	Alg.3
Max	1.06416	-11.9698	-12.5638
Average	-0.6551	-12.3934	-12.5782
Min	-8.2283	-12.5297	-12.5800
Convergence freq. (ISM)	26	33.6	1.9

Table 1 shows the results at the end of the first convergence (without ISM). From this table, although Alg. 3 takes the longest time compared to other algorithms, algorithm 3 achieves the highest performance. We can see the improvement by the reduced-order search is larger than the improvement by the algorithm 2 with LMI relaxation technique which we proposed in Kawanishi and Ikuyama (2005).

In order to compare the performance in the same time interval, we carry out the second experiments. We compare the performance in 650 s CPU time. The experimental results are shown in Table 2. Algorithm 3 with reduced-order search is resulting the highest performance in three algorithms.

The resultant controller K and the Lyapunov variable X that achieves the minimum eigenvalue -12.5801 are

$$K = \begin{bmatrix} -18.7822 \\ 99.2710 \end{bmatrix}, \tag{28}$$

$$X = \begin{bmatrix} 100.0000 & -0.4819 & 13.2491 & 5.8518 \\ -0.4819 & 41.2086 & -27.5647 & -0.0869 \\ 13.2491 & -27.5647 & 45.2944 & -6.3143 \\ 5.8518 & -0.0869 & -6.3143 & 25.1968 \end{bmatrix}. \tag{29}$$

Closed-loop poles by the controller are -821.28, -0.09, and $-0.32 \pm 1.06i$.

5.2 Parallelization efficiency

In this subsection, we confirm the parallelization efficiency by parallelized algorithm 3. We utilize a Beowulf cluster computer system which consists of 31 nodes. CPU of each node is AMD AthlonXP 2500+ 2.0 GHz. The size of memory on each nodes is 256 MB. The nodes are connected by TCP/IP Gigabit ethernet. The algorithm

Table 3. Results of Parallelization Efficiency

# of nodes	1	6	11	16	31
Eval. time [s]	178.4	48.80	33.60	25.73	19.51
Total time [s]	208.3	79.61	65.26	56.29	51.57
Ratio [%]	85.63	61.23	51.42	45.72	37.83
Opt. value	-12.58	-12.58	-12.58	-12.58	-12.58

is implemented by C++ with MPICH, CLAPACK and SDPA6.0 Libraries. We here evaluate the parallelization efficiency of the proposed parallelized algorithm 3 in 100-generation as the number of nodes increases.

Table.3 shows the result of parallelization efficiency of the developed parallelized algorithm 3, where "Eval. time" indicates the entire time for individuals evaluation, "Total time" represents the total execution time of the algorithm, "Ratio" means the ratio of "Eval. time" against "Total time", and "Opt. value" is the best value of the evaluation function, i.e. BMI eigenvalue.

In the case of one node, 85.63% of the coputational power is spent for the evaluation of individuals. The high level of "Ratio" indicates the importance of the parallelization of the evaluation. Comparing "Eval. time" for one node and 31 nodes, we can see that the parallelized algorithm 3 succeeds to reduce the evaluation time to 19.51 s from 178.4 s. From the result of "Total time" in Table.3, we can also see that the reduction of total execution time is almost saturated at 51.57 s due to the communication burden between the nodes.

6. CONCLUSION

In this paper, taking account of BMIEP's characteristic properties, we developed BMI oriented real-coded GA algorithms. The developed real-coded genetic algorithm is based on UNDX(Unimodal Normal Distribution Crossover), EDX(Extrapolation-directed Crossover), MGG(Minimal Generation Gap) model, and ISM(Innately Split Model). In order to utilize BMI structure characteristics effectively, we introduced a new primary search direction with relaxed LMI, reduced-order individual expression, and alternating SDP evaluation.

The alternating SDP evaluation of individuals needs much computational burden compared to conventional BMI eigenvalue evaluation. In order to reduce the computational burden, we considered the parallelization of our BMI real-coded genetic algorithm and developed a parallelized algorithm.

By the numerical experiments, we showed the utilization of BMI characteristic can improve the performance of general-purpose genetic algorithms. Real-coded GA approach is promissing for lage-scale practical BMIEP. The further research is needed.

ACKNOWLEDGEMENTS

This work was supported by KAKENHI (18560435) Grant-in-Aid for Scientific Research (C). Sponsor and financial support acknowledgment goes The Ministry of Education, Culture, Sports, Science and Technology (MEXT).

REFERENCES

L. Davis. *The handbook of genetic algorithms.* Van Nostrand Reinhold, New York, 1990.

L. J. Eshelman and J. D. Schaffer. Real coded genetic algorithms and interval-schemata. *Foundation of Genetic Algorithms*, pages 187–202, 1993.

L. J. Eshleman, K. E. Mathias, and J. D. Schaffer. Crossover operator biases, exploiting the population distribution. *Proc. of the 7th International Conference on Genetic Algorithms*, pages 354–361, 1997.

H. Fujioka and K. Hoshijima. Bounds for the BMI eigenvalue problem -A good lower bound and a cheap upper bound-. *Transactions of the Society of Instrument and Control Engineers*, 33(7):616–621, 1997.

K. C. Goh, L. Turan, M. G. Safonov, G. P. Papavassilopoulos, and J. H. Ly. Biaffine matrix inequality properties and computational methods. *Proc. American Control Conference*, pages 850–855, 1994.

D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* AddisonWsley Publishing Company Inc., 1989.

K. Ikeda and S. Kobayashi. GA based on the UV-structure hypothesis and its application to JSP. *Proc. Of Parallel Problem Solving from Nature*, pages 273–282, 2000.

M. Kawanishi and Y. Ikuyama. BMI optimization based on unimodal normal distribution crossover GA with relaxed LMI convex estimation. *Proc. of IEEE American Control Conference*, pages 2299–2304, 7 2005.

L. H. Keel, S. P. Bhattacharyya, and Jo W. Howze. Robust control with structured perturbations. *IEEE Trans. Automatic Control*, 33(1):68–78, 1988.

H. Kita, I. Ono, and S. Kobayashi. Theoretical analysis of the unimodal normal distribution crossover for real-coded genetic algorithms. *Proc. of IEEE International Conference on Evolutionary Computation*, pages 529–534, 1998.

I. Ono and S. Kobayashi. A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover. *Proc. of the 7th International Conference on Genetic Algorithms*, pages 246–253, 1997.

M. G. Safonov, K. C. Goh, and J. H. Ly. Control system synthesis via bilinear matrix inequalities. *Proc. American Control Conference*, pages 45–49, 1994.

J. Sakuma and S. Kobayashi. Extrapolation-directed crossover for real-coded GA: Overcoming deceptive phenomena by extrapolative search. *Proc. of IEEE International Conference on Evolutionary Computation*, pages 656–662, 2001.

H. Satoh, M. Yamamura, and S. Kobayashi. Minimal generation gap model for GAs considering both exploration and exploitation. *Proc. of the IIZUKA: Methodologies for the Conception, Design and Application of Intelligent Systems*, pages 494–497, 1996.

A. Wright. Genetic algorithms for real parameter optimization. *Foundations of Genetic Algorithms*, pages 205–218, 1991.