Paper:

AEGA: A New Real-Coded Genetic Algorithm Taking Account of Extrapolation

Kento Uemura and Isao Ono

Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama, 226-8502 Kanagawa, Japan
E-mail: {uemura@ic., isao@}dis.titech.ac.jp

[Received October 26, 2015; accepted January 27, 2016]

This study proposes a new real-coded genetic algorithm (RCGA) taking account of extrapolation, which we call adaptive extrapolation RCGA (AEGA). Real-world problems are often formulated as black-box function optimization problems and sometimes have ridge structures and implicit active constraints. mAREX/JGG is one of the most powerful RCGAs that performs well against these problems. However, mAREX/JGG has a problem of search inefficiency. To overcome this problem, we propose AEGA that generates offspring outside the current population in a more stable manner than mAREX/JGG. Moreover, AEGA adapts the width of the offspring distribution automatically to improve its search efficiency. We evaluate the performance of AEGA using benchmark problems and show that AEGA finds the optimum with fewer evaluations than mAREX/JGG with a maximum reduction ratio of 45%. Furthermore, we apply AEGA to a lens design problem that is known as a difficult real-world problem and show that AEGA reaches the known best solution with approximately 25% fewer evaluations than mAREX/JGG.

Keywords: real-coded genetic algorithms, adaptive extrapolation RCGA, black-box function optimization, ridge structures, implicit active constraints

1. Introduction

Function optimization is an major problem that occurs in various fields of engineering. The purpose in solving the problem is to identify the decision variable $\boldsymbol{x} \in \mathscr{X} \subseteq \mathbb{R}^n$ that minimizes a given objective function $f(\boldsymbol{x})$:

$$\text{minimize } f(\boldsymbol{x}) \text{ subject to } \boldsymbol{x} \in \mathscr{X} \subseteq \mathbb{R}^n, \quad . \quad . \quad . \quad (1)$$

where $\mathscr{X}$ is the search space. The problem is *unconstrained* when the search space is n-dimensional real-valued space, $\mathscr{X} = \mathbb{R}^n$, and *constrained* when constraint is imposed, $\mathscr{X} \subset \mathbb{R}^n$. A solution is *feasible* if it satisfies the constraint and *infeasible* if it violates the constraint.

In real-world problems, $f(\boldsymbol{x})$ is often given as *black-box* and sometimes has *ridge structures* [1]. When $f(\boldsymbol{x})$ is not given as an explicit form, $f(\boldsymbol{x})$ is called black-box. For example, $f(\boldsymbol{x})$ is black-box when some simulations such as of physical phenomena are required to calculate $f(\boldsymbol{x})$. The ridge structure is a class of landscapes of objective functions in which lie narrow and curved valleys, called ridges.

In real-world constrained problems, *implicit* constraints are sometimes imposed and they are often *active*. An implicit constraint is a constraint that is not explicitly given as constraint functions. When a constraint is implicit, the feasibility of $\boldsymbol{x}$ is determined by whether its evaluation value is defined. The implicit constrained function optimization can be formulated as:

$$\text{minimize } f_{\text{ic}}(\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & \boldsymbol{x} \in \mathscr{X} \\ +\infty & \boldsymbol{x} \notin \mathscr{X} \end{cases}, \quad . \quad . \quad . \quad (2)$$

where undefined objective function values are regarded as $+\infty$. A constraint is active when the optimum lies on the boundary between feasible and infeasible regions. Note that in black-box cases, the boundary is unknown in advance on implicit constrained problems.

The real-coded genetic algorithm (RCGA) [2–4] is known as a powerful black-box function optimization method. In the RCGA, a solution candidate is called an *individual* and a set of individuals is called a *population*. The RCGA updates its population stochastically through generations and, finally, is expected to have the population converge to the optimum. Many RCGAs have been proposed for unconstrained problems and they have yielded good performance [5–9]. RCGAs for unconstrained problems can be easily applied to those of implicit constrained by means of *the resampling technique* [10, 11]. The resampling technique rejects infeasible individuals and resamples them until feasible ones are generated.

Many conventional RCGAs generate new individuals, called *offspring*, to *interpolate* the current population to find the optimum. They assume that a population continues covering the optimum throughout the search [5–7, 9].

However, a population does not always cover the optimum. For problems with ridge structures, it is often observed that a population comes not to cover the optimum while searching on a ridge. As a result, it often prematurely converges on the ridge. For implicit active constrained problems, it is often observed that a population does not cover the optimum throughout the search. Thus, it often converges far from the optimum or on a wrong constraint boundary on which the optimum does not lie.

AREX/JGG [12, 13] used with the resampling technique, called *modified AREX/JGG* (*mAREX/JGG*), is one of the most promising methods used to correct the problems of interpolation-oriented RCGAs. mAREX/JGG generates offspring not only to interpolate but also to extrapolate the current population and has shown good performance on problems with ridge structures and implicit active constraints. *Extrapolation* enables a population not only to move along a ridge but also to approach the optimum on an active constraint or escape from a wrong constraint boundary.

However, mAREX/JGG has a problem of search inefficiency. mAREX/JGG cannot always extrapolate the population because mAREX/JGG chooses parents randomly from the current population to generate offspring.

In this study, we propose an RCGA, called adaptive extrapolation RCGA (AEGA)[1] that extrapolates the current population in a stable manner to overcome the problem of mAREX/JGG. AEGA generates offspring based on information about the current population to extrapolate that population when AEGA tries to move or expand it. Furthermore, AEGA adapts the width of the offspring distribution separately in two directions to improve the search efficiency. We show the effectiveness of AEGA compared to that of mAREX/JGG using benchmark problems and a lens design problem [15].

The reminder of the paper is organized as follows. We explain mAREX/JGG and point out its problem in Section 2. We then propose AEGA in Section 3 and evaluate it using benchmark and real-world problems in Sections 4 and 5, respectively. In Section 6, we conclude the study and suggest future research.

2. mAREX/JGG and its Problem

2.1. mAREX/JGG

AREX/JGG [12, 13] is one of the most powerful RCGAs for unconstrained benchmark problems. To handle implicit constrained problems, AREX/JGG is often used with the resampling technique. In this study, we call AREX/JGG with the resampling technique the *modified AREX/JGG* (mAREX/JGG). Note that mAREX/JGG works in exactly the same manner as AREX/JGG on unconstrained problems.

mAREX/JGG can generate offspring not only to interpolate but also to extrapolate the current population, as shown in **Fig. 1**. At the beginning of a generation, mAREX/JGG chooses $n+1$ parents randomly from the population, where n is the dimension of a problem. Then, mAREX/JGG generates the region spanned by the parents and translates the region so that its center matches the weighted mean of the parents. The weighted mean is calculated with a higher weight for a better parent and thus, mAREX/JGG translates the region toward one considered more promising. After the translation, mAREX/JGG expands the region isotropically with an expansion rate parameter that is adapted automatically using AER(M) [12] and generates $\lambda(> n+1)$ offspring according to the the multivariate normal distribution on the expanded region. Finally, mAREX/JGG replaces the parents with the best $n+1$ offspring selected from that of λ. Please see the study of [12, 13] for the details about this algorithm.

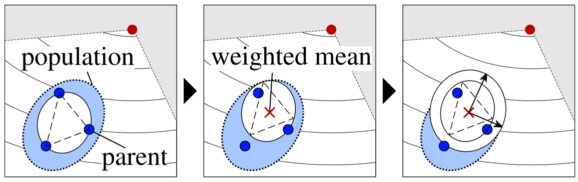

Fig. 1. Offspring generation in mAREX/JGG. Curved lines represent the contours of the objective function. The optimum (red point) lies on the boundary between feasible (lower white area) and infeasible (upper gray area) regions.

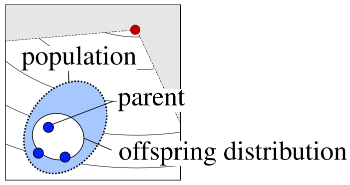

Fig. 2. Problem of mAREX/JGG.

2.2. Problem of mAREX/JGG

We believe that mAREX/JGG has a problem regarding search efficiency. mAREX/JGG generates offspring to extrapolate not the population but rather the parents that are chosen randomly from the current population. Therefore, if the parents are chosen unevenly as shown in **Fig. 2**, mAREX/JGG cannot extrapolate the population.

3. Adaptive Extrapolation RCGA

3.1. Basic Ideas

3.1.1. Generating Offspring and Updating a Population

To overcome the problem of mAREX/JGG, we propose an RCGA called AEGA. AEGA generates offspring to extrapolate the current population in a more stable manner when it needs to move or expand the population.

AEGA expands an offspring distribution separately in two directions, named *promising* and *orthogonal directions*. A promising direction is one from the mean of the worst half individuals to the mean of the best half individuals in the current population that consists of μ individuals. Orthogonal directions are all directions that are orthogonal to the promising direction.

AEGA determines the width of the offspring distribution separately in the promising and orthogonal directions

1. We have developed AEGA based on the prototype that we proposed in [14]. We have introduced a new adaptation mechanism of expansion rate, β, and a new definition of the promising direction toward which an offspring distribution spreads widely to the prototype.

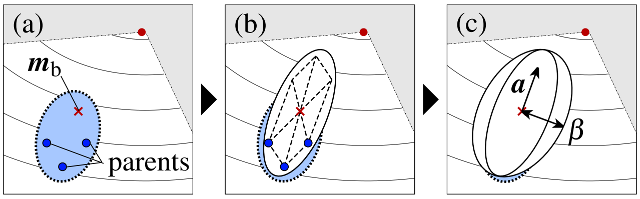

Fig. 3. Generating an offspring distribution in AEGA.

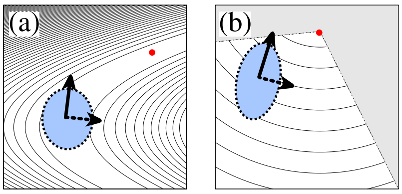

Fig. 4. Situations in which the expansion along the orthogonal direction (shown in dotted arrows) seems to be effective.

as shown in **Fig. 3**. At the beginning of a generation, AEGA chooses $n+1$ parents randomly from the worst half individuals in the population, where n is the dimension of a problem, and calculates the mean of the best half individuals, $\boldsymbol{m}_\mathrm{b}$, as shown in **Fig. 3(a)**. Then, AEGA constructs the region spanned by the parents and the mirrored parents that locate symmetric with respect to $\boldsymbol{m}_\mathrm{b}$ as shown in **Fig. 3(b)**. AEGA is expected to generate the offspring widely in the promising direction, $\boldsymbol{a}$, when the solutions having good evaluation values are biased in the population. Finally, AEGA expands the region in the orthogonal directions using an expansion rate parameter, β, as show in **Fig. 3(c)**, and generates offspring according to the multivariate normal distribution on the expanded region. Note that AEGA uses the resampling technique to handle constrained problems.

To control the width of the offspring distribution in orthogonal directions, AEGA adapts β so that β increases when a population moves along a ridge and approaches a wrong boundary without the optimum as shown in **Fig. 4**, and decreases when a population converges into a promising region. A large β helps the population to change the search direction along a ridge as shown in **Fig. 4(a)** and to escape from the convergence on a wrong boundary as shown in **Fig. 4(b)**. By contrast, a small β helps the population to converge into the promising region quickly. AEGA guarantees that β is greater than or equal to 1 because the purpose of the expansion is to extrapolate the population and excessive convergence may degrade performance. Therefore, AEGA adapts β according to two basic ideas: 1) β increases when the population changes the search direction and escaping from a wrong boundary, and 2) decreases when the population converges into a promising region. We call an adaptation mechanism based on the first idea *Adaptation 1* and that based on the second *Adaptation 2*. We explain in detail the two adaptation mechanisms in Section 3.1.2.

After generating offspring, AEGA replaces all $n+1$

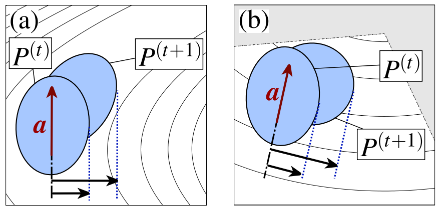

Fig. 5. Expected changes of populations when they: (a) advances along a ridge, and (b) approaches a wrong boundary. $P^{(t)}$ is a population of the tth generation.

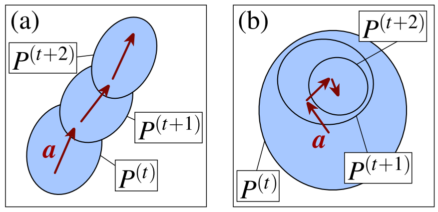

Fig. 6. Expected changes of populations and promising directions, $\boldsymbol{a}$, when the population is: (a) moving and (b) converging. $P^{(t)}$ is the population of the t-th generation.

parents randomly chosen from the worst half individuals in the population with the best $n+1$ individuals selected from the offspring to accelerate the search.

3.1.2. Adapting the Expansion Rate

Adaptation 1 increases β in order to change the search direction or expand the population when the population advances along a ridge for problems with ridge structures or approaches a wrong constraint boundary without the optimum for implicit active constrained problems. **Fig. 5** depicts the expected changes of successive populations in these situations. The population is expected to move or expand in orthogonal directions when it advances along a ridge as shown in **Fig. 5(a)** or approaches a wrong boundary without the optimum as shown in **Fig. 5(b)**. Therefore, Adaptation 1 increases β according to the degree of movement or expansion of the two successive populations, $P^{(t)}$ and $P^{(t+1)}$, in orthogonal directions.

Adaptation 2 decreases β to prevent extrapolation when the population is converging into a promising region. **Fig. 6(a)** depicts the expected changes of successive populations and their promising directions when the population moves in a certain direction. On the other hand, **Fig. 6(b)** depicts expected changes when the population is converging into a promising region. The promising directions are expected to point in various directions when the population is converging as shown in **Fig. 6(b)** and point in similar directions when moving in a certain direction as shown in **Fig. 6(a)**. Therefore, Adaptation 2 decreases β according to the degree of variation of the promising directions in recent generations.

In the following section, we describe the implementation of generating offspring, Adaptation 1, Adaptation 2, and combining the two adaptation mechanisms.

3.2. Generating Offspring

First, AEGA calculates the mean of the best half individuals, $\boldsymbol{m}_{\mathrm{b}}$, and the worst half individuals, $\boldsymbol{m}_{\mathrm{w}}$, and the promising direction, $\boldsymbol{a}$, of the current population as:

$$\boldsymbol{m}_{\mathrm{b}} := \frac{1}{\mu_{\mathrm{h}}}\sum_{k=1}^{\mu_{\mathrm{h}}} \boldsymbol{p}_{k:\mu},$$
$$\boldsymbol{m}_{\mathrm{w}} := \frac{1}{\mu-\mu_{\mathrm{h}}}\sum_{k=\mu_{\mathrm{h}}+1}^{\mu} \boldsymbol{p}_{k:\mu},$$
$$\boldsymbol{a} := \boldsymbol{m}_{\mathrm{b}} - \boldsymbol{m}_{\mathrm{w}}, \quad (3)$$

where $\mu_{\mathrm{h}} := \lfloor \mu/2 \rfloor$; μ is the number of individuals in the population, called the *population size*; and $\boldsymbol{p}_{k:\mu}$ is the k-th best individual in the current population with respect to the evaluation value. AEGA then generates λ feasible offspring, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\lambda$, according to:

$$\boldsymbol{x}_i = \boldsymbol{m}_{\mathrm{b}} + \sum_{j=1}^{n+1} \varepsilon_{i,j}\mathbf{B}(\boldsymbol{y}_j - \boldsymbol{m}_{\mathrm{b}}), \quad (4)$$

where $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n+1}$ are the parents chosen randomly from the worst half individuals in the population; $\varepsilon_{i,j}$ is a random number from the normal distribution with a mean of zero and a user-defined variance of σ^2; and $\mathbf{B}$ is the expansion matrix for the orthogonal directions. $\mathbf{B}$ is defined as $\mathbf{B} := ((1-\beta)/\|\boldsymbol{a}\|^2)\boldsymbol{a}\boldsymbol{a}^{\mathrm{T}} + \beta\mathbf{I}_n$.

After generating offspring, AEGA updates the population by replacing the $n+1$ parents, $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n+1}$, with the best $n+1$ individuals selected from the λ offspring, $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{n+1}$:

$$\{\boldsymbol{p}'_1, \ldots, \boldsymbol{p}'_\mu\} = \left(\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_\mu\} \setminus \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n+1}\}\right) \cup \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{n+1}\}. \quad (5)$$

3.3. Adaptation 1

Adaptation 1 adapts β according to

$$\beta \leftarrow (1-c_\beta)\beta + c_\beta \cdot \max\left\{\tilde{\delta}, 1\right\}, \ \tilde{\delta} := \sqrt{1+d_\beta\tau}, \quad (6)$$

where τ represents the degree of movement or expansion of two successive populations, and $c_\beta \in [0,1]$ and $d_\beta \in [0,1]$ are user parameters that determine the learning speed of β and adjust the scale of τ, respectively. The max operator is introduced to guarantee $\beta \geq 1$. The reminder of this section explains the definition of τ.

To quantify the degree of the movement or the expansion of two successive populations in an arbitrary orthogonal direction, $\boldsymbol{q}$, *Adaptation 1* employs the ratio of the second moment of the current population that is projected onto one-dimensional space spanned by $\boldsymbol{q}$ to the second moment of the previous population about $\boldsymbol{m}$, where $\boldsymbol{m}$ is the mean of the previous projected population. However, the value of the ratio depends on the population size parameter. Thus, to remove the dependency, *Adaptation 1* uses the parents, $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n+1}$, and the best individuals selected from the offspring, $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{n+1}$, instead of the two successive populations. Noting that the projection of a vector, $\boldsymbol{x}$, onto the one-dimensional space spanned by a unit vector, $\boldsymbol{q}$, is defined as $\boldsymbol{q}^{\mathrm{T}}\boldsymbol{x}$, we can calculate the second moments as:

$$\frac{1}{n+1}\sum_{j=1}^{n+1}\left(\boldsymbol{q}^{\mathrm{T}}\boldsymbol{y}_j - \boldsymbol{q}^{\mathrm{T}}\boldsymbol{m}\right)\left(\boldsymbol{q}^{\mathrm{T}}\boldsymbol{y}_j - \boldsymbol{q}^{\mathrm{T}}\boldsymbol{m}\right)^{\mathrm{T}} =: \boldsymbol{q}^{\mathrm{T}}\mathbf{M}_{\mathrm{y}}\boldsymbol{q}, \quad (7)$$

$$\frac{1}{n+1}\sum_{i=1}^{n+1}\left(\boldsymbol{q}^{\mathrm{T}}\boldsymbol{s}_i - \boldsymbol{q}^{\mathrm{T}}\boldsymbol{m}\right)\left(\boldsymbol{q}^{\mathrm{T}}\boldsymbol{s}_i - \boldsymbol{q}^{\mathrm{T}}\boldsymbol{m}\right)^{\mathrm{T}} =: \boldsymbol{q}^{\mathrm{T}}\mathbf{M}_{\mathrm{s}}\boldsymbol{q}, \quad (8)$$

where $\mathbf{M}_{\mathrm{y}}$ and $\mathbf{M}_{\mathrm{s}}$ are the n-dimensional second moment matrices about $\boldsymbol{m}$ of the parents and the selected individuals, respectively. The quantity of the change of the population in $\boldsymbol{q}$ is then defined as:

$$\tau_{\boldsymbol{q}} := \frac{\boldsymbol{q}^{\mathrm{T}}\mathbf{M}_{\mathrm{s}}\boldsymbol{q}}{\boldsymbol{q}^{\mathrm{T}}\mathbf{M}_{\mathrm{y}}\boldsymbol{q}} - 1. \quad (9)$$

The value of $\tau_{\boldsymbol{q}}$ tends to exceed 0 when the population moves or expands in $\boldsymbol{q}$.

However, the selection of $\boldsymbol{q}$ is not unique in three or more dimensional space. To detect the change even when the population moves or expands in only one direction, AEGA employs the maximum value, $\tau := \max_{\boldsymbol{q}} \tau_{\boldsymbol{q}}$. By solving this maximization problem,[2] we obtain

$$\tau = \mathrm{maxeig}\left\{\left(\mathbf{I}_n - \frac{1}{\|\boldsymbol{b}\|^2}\boldsymbol{b}\boldsymbol{b}^{\mathrm{T}}\right)\mathbf{L}^{-1}\mathbf{M}_{\mathrm{s}}\mathbf{L}^{-\mathrm{T}}\right\} - 1, \quad (10)$$

where $\mathrm{maxeig}\{\cdot\}$ is the maximum eigenvalue, $\mathbf{L}$ is the Cholesky decomposition of $\mathbf{M}_{\mathrm{y}}$, and $\boldsymbol{b} := \mathbf{L}^{-1}\boldsymbol{a}$.

Note that, ignoring the two user parameters, c_β and d_β, by setting them to 1 for simplicity, Eq. (6) enables β to approach the square root of the ratio of the second moments.

3.4. Adaptation 2

Adaptation 2 adapts β according to

$$\beta \leftarrow (\beta - 1)r + 1, \quad (11)$$

where r represents the degree of variation of the promising directions in the recent generations. The reminder of this section explains the definition of r.

To quantify the degree of variation of the promising directions, $\boldsymbol{a}$, in the recent generations, *Adaptation 2* employs the squared ratio of the norm of the exponential moving average vector of $\boldsymbol{a}$ to the exponential moving average value of the norm of $\boldsymbol{a}$. The ratio is defined as:

$$r := \frac{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}}{\ell^2}, \quad (12)$$

where $\boldsymbol{v}$ and ℓ are updated before the adaptation of β according to:

$$\boldsymbol{v} \leftarrow (1-c_{\mathrm{r}})\boldsymbol{v} + c_{\mathrm{r}}\check{\boldsymbol{a}}, \ \ \ell \leftarrow (1-c_{\mathrm{r}})\ell + c_{\mathrm{r}}\|\check{\boldsymbol{a}}\|, \quad (13)$$

where $c_{\mathrm{r}} \in [0,1]$ is a user parameter that determines the learning speed and $\check{\boldsymbol{a}}$ is a normalized promising direction. AEGA uses the promising direction normalized by the population distribution instead of $\boldsymbol{a}$ itself so that r becomes independent of the scale and shape of the popula-

2. See Appendix for the derivation.

tion distribution. The normalized promising direction, $\check{\boldsymbol{a}}$, is defined as:

$$\check{\boldsymbol{a}} := \frac{\sqrt{\boldsymbol{a}^{\mathrm{T}}\mathbf{C}^{-1}\boldsymbol{a}}}{\|\boldsymbol{a}\|}\boldsymbol{a}, \quad (14)$$

where $\mathbf{C}$ is the covariance matrix of the population before replacing the parents with the best individuals selected from the offspring, $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_\mu\}$, and is calculated as:

$$\mathbf{C} := \frac{1}{\mu-1}\sum_{k=1}^{\mu}(\boldsymbol{p}_k-\boldsymbol{m})(\boldsymbol{p}_k-\boldsymbol{m})^{\mathrm{T}},\ \boldsymbol{m} := \frac{1}{\mu}\sum_{k=1}^{\mu}\boldsymbol{p}_k. \quad (15)$$

Note that *Adaptation 2* does not increase β because r is guaranteed to be in $[0,1]$ under appropriate[3] initializations of $\boldsymbol{v}$ and ℓ.

3.5. Combination of Two Adaptations

AEGA adapts β based on *Adaptations 1* and *2* by:

$$\beta \leftarrow (1-c_\beta)\beta + c_\beta\delta,$$
$$\delta := \max\left\{\left(\sqrt{1+d_\beta\tau}-1\right)r+1,\ 1\right\}. \quad (16)$$

Note that δ in Eq. (16) is equivalent to $\tilde{\delta}$ in Eq. (6) adjusted by r in Eq. (11).

3.6. Algorithm

The algorithm of AEGA is as follows:

1. Initialize the population with μ $(> 2(n+1))$ feasible individuals according to a given distribution.
2. Choose $n+1$ parents, $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n+1}$, randomly from the worst half individuals in the population.
3. Generate λ feasible offspring, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\lambda$, according to Eq. (4) with the current expansion rate, β.
4. Update the population by replacing the parents with the best $n+1$ offspring, $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{n+1}$.
5. Adapt the expansion rate, β, according to *Adaptations 1* and *2* by means of Eq. (16).
6. Repeat Steps 2–5 until termination conditions are met.

4. Experiments with Benchmark Problems

4.1. Purpose and Performance Index

To show the effectiveness of AEGA, we compared the performance of AEGA and that of mAREX/JGG on benchmark problems with implicit active constraints or ridge structures.

As a performance index, we used the minimum number of evaluations when the population size and number of offspring per generation were tuned so that each method can find the optimum in all 10 trials. We determined that the method with the fewer number of evaluations outperforms the other in terms of cost for finding the optimum. We included the number of infeasible individuals that are rejected by the resampling technique in the total number of evaluations.

4.2. Benchmark Problems

We used CSF [16], G06, G07, G09, and G10 [17] as benchmark problems with implicit active constraints. Note that the boundary between feasible and infeasible regions is unknown in advance on implicit constrained black-box problems. We set the dimension, n, and the number of active constraints, m, on CSF to $n = 10$, 20, and $m = 1$, $n/2$, n, respectively. G06, G07, G09, and G10 are widely used active constrained benchmark problems with $n = m = 2$, $n = 10$, $m = 8$, $n = 7$, $m = 4$, and $n = 8$, $m = 6$, respectively. G10 is a difficult problem where convergence on wrong constraint boundaries without the optimum is likely to occur. Although constraints are given explicitly on these problems, we used them as implicit constrained.

We used Rosenbrock (star) and Rosenbrock (chain) [12] as benchmark problems with ridge structures. We set the dimension to $n = 20$, 40 on each problem.

See each study for the details of the problems (i.e., definitions, search spaces, optimal evaluation values, etc.).

4.3. Settings

We initialized a population with feasible individuals that are sampled randomly in the search space of each problem. We tuned the population size, μ, and the number of offspring per generation, λ, of AEGA and mAREX/JGG on each problem. We employed the setting that achieved the minimum average number of evaluations among the all combinations of $\mu = 3n, 4n, \ldots, 10n$ and $\lambda = 2n, 3n, \ldots, 5n$. We determined that a method found the optimum successfully if $f(\boldsymbol{x}^*) - f^* < 10^{-8}$ on CSF and Rosenbrock(star, chain) and $f(\boldsymbol{x}^*) = f^*_{\mathrm{known}}$ on G06, G07, G09, and G10 before the total number of evaluations reached $n \cdot 10^5$, where $f(\boldsymbol{x}^*)$ is the best evaluation value in the population, f^* is the optimal evaluation value, and f^*_{known} is the known best evaluation value. We set other user parameters of mAREX/JGG to those recommended [12, 13]. We decided the recommended parameters of AEGA by preliminary experiments: $c_\beta = 1/(5(n-1)), d_\beta = 0.14, c_{\mathrm{r}} = 1/n, \sigma^2 = 1/n$. We initialized the internal parameters of AEGA as $\beta = 1, \boldsymbol{v} = \mathbf{0}, \ell = 0$.

4.4. Results

Tables 1 and **2** show the results on the problems with the implicit active constraint and the ridge structure, respectively. We found that AEGA reaches the optimum with a fewer number of evaluations than mAREX/JGG on all problems. The reduction ratio is at most 55% on implicit active constrained problems and 35% on problems with ridge structures. These results indicate the effectiveness of AEGA.

3. It follows that $0 \le \|\boldsymbol{v}_0\| \le \ell_0 \Rightarrow 0 \le r \le 1$ by the mathematical induction.

Table 1. Results on CSF, G06, G07, G09 and G10. Each value indicates (the number of evaluations $\pm$ the standard deviation) and tuned values of $[\mu, \lambda]$; and n and m are the dimension and number of active constraints, respectively. The better results are written in bold.

CSF ($n=10$)	$m=1$	$m=n/2=5$	$m=n=10$
AEGA	$(\mathbf{7.20} \pm 0.68) \times 10^3$ $[4n, 3n]$	$(\mathbf{1.85} \pm 0.05) \times 10^4$ $[6n, 2n]$	$(\mathbf{3.27} \pm 0.12) \times 10^4$ $[6n, 2n]$
mAREX/JGG	$(1.47 \pm 0.13) \times 10^4$ $[5n, 4n]$	$(3.58 \pm 0.38) \times 10^4$ $[5n, 3n]$	$(6.48 \pm 0.48) \times 10^4$ $[6n, 2n]$
CSF ($n=20$)	$m=1$	$m=n/2=10$	$m=n=20$
AEGA	$(\mathbf{2.44} \pm 0.33) \times 10^4$ $[5n, 3n]$	$(\mathbf{8.08} \pm 0.25) \times 10^4$ $[7n, 2n]$	$(\mathbf{1.49} \pm 0.27) \times 10^5$ $[6n, 2n]$
mAREX/JGG	$(3.68 \pm 0.20) \times 10^4$ $[6n, 3n]$	$(1.27 \pm 0.05) \times 10^5$ $[7n, 2n]$	$(2.65 \pm 0.24) \times 10^5$ $[8n, 2n]$

	G06 ($n=2, m=2$)	G07 ($n=10, m=8$)
AEGA	$(\mathbf{1.15} \pm 0.12) \times 10^3$ $[4n, 4n]$	$(\mathbf{1.82} \pm 0.08) \times 10^4$ $[6n, 2n]$
mAREX/JGG	$(2.66 \pm 0.33) \times 10^3$ $[5n, 4n]$	$(3.45 \pm 0.43) \times 10^4$ $[6n, 2n]$
	G09 ($n=7, m=4$)	G10 ($n=8, m=6$)
AEGA	$(\mathbf{5.43} \pm 0.52) \times 10^3$ $[5n, 3n]$	$(\mathbf{2.29} \pm 0.14) \times 10^4$ $[8n, 3n]$
mAREX/JGG	$(9.58 \pm 1.13) \times 10^3$ $[5n, 3n]$	$(2.90 \pm 0.47) \times 10^4$ $[5n, 2n]$

Table 2. Results on Rosenbrock(star) and Rosenbrock(chain). Each value indicates (the number of evaluations $\pm$ the standard deviation) and tuned values of $[\mu, \lambda]$ and n is the dimension. The better results are written in bold.

Rosenbrock(star)	$n=20$	$n=40$
AEGA	$(\mathbf{3.92} \pm 0.14) \times 10^4$ $[6n, 4n]$	$(\mathbf{1.58} \pm 0.12) \times 10^5$ $[7n, 4n]$
mAREX/JGG	$(6.17 \pm 0.23) \times 10^4$ $[9n, 3n]$	$(2.07 \pm 0.08) \times 10^5$ $[8n, 4n]$
Rosenbrock(chain)	$n=20$	$n=40$
AEGA	$(\mathbf{8.65} \pm 0.49) \times 10^4$ $[7n, 3n]$	$(\mathbf{4.99} \pm 0.10) \times 10^5$ $[8n, 4n]$
mAREX/JGG	$(1.09 \pm 0.05) \times 10^5$ $[5n, 3n]$	$(5.98 \pm 0.16) \times 10^5$ $[7n, 3n]$

Table 3. Comparison of the performance of AEGA and that of AEGA with no expansion (AEGA$_{\mathrm{noexp}}$). Variables n and m are the dimension and the number of active constraints, respectively. The results of AEGA are reproduced in **Tables 1** and **2**. Note that "–" indicates that no parameter settings exist to find the optimum in all 10 trials. The better results are written in bold.

	CSF($n=20, m=1$)	CSF($n=20, m=10$)	CSF($n=20, m=20$)
AEGA	$(2.44 \pm 0.33) \times 10^4$ $[5n, 3n]$	$(\mathbf{8.08} \pm 0.25) \times 10^4$ $[7n, 2n]$	$(\mathbf{1.49} \pm 0.27) \times 10^5$ $[6n, 2n]$
AEGA$_{\mathrm{noexp}}$	$(\mathbf{2.35} \pm 0.09) \times 10^4$ $[6n, 2n]$	$(9.58 \pm 0.66) \times 10^4$ $[10n, 2n]$	$(1.71 \pm 0.12) \times 10^5$ $[9n, 3n]$
	G07($n=10, m=7$)	G10($n=8, m=6$)	Rosenbrock(chain, $n=20$)
AEGA	$(\mathbf{1.15} \pm 0.12) \times 10^3$ $[4n, 4n]$	$(\mathbf{2.29} \pm 0.14) \times 10^4$ $[8n, 3n]$	$(\mathbf{8.65} \pm 0.49) \times 10^4$ $[7n, 3n]$
AEGA$_{\mathrm{noexp}}$	$(2.07 \pm 0.12) \times 10^4$ $[8n, 2n]$	–	–

4.5. Discussions

4.5.1. Effectiveness of the Adaptive Expansion with β

To evaluate the effectiveness of the expansion along orthogonal directions, we conducted additional experiments with AEGA that uses $\beta = 1$ throughout the search and was named AEGA$_{\mathrm{noexp}}$. The settings followed those of the aforementioned experiments.

Table 3 shows the results of AEGA$_{\mathrm{noexp}}$. We found that the performance of AEGA$_{\mathrm{noexp}}$ deteriorated as the number of active constraints increased. Moreover, AEGA$_{\mathrm{noexp}}$ failed in finding the optimum on G10 and Rosenbrock(chain). These results suggest that the adaptive expansion with β works effectively.

4.5.2. Behavior of the Adaptation of β

To verify that the adaptation of β works as expected, in **Fig. 7** we show the transitions of the coordinate values of the mean of the population and β in a typical trial on 20-dimensional Rosenbrock(chain).

As shown in **Fig. 7**, AEGA increases β while the population moves along a ridge in the middle of the search (i.e., from nearly 100 to 1100 generations) and decreases β when the population converges to the optimum (i.e., after nearly 1100 generations). This result suggests that β is adapted successfully as we designed in Section 3.1.2.

5. Application to Lens Design Problem

To verify the effectiveness of AEGA in real-world problems, we applied AEGA and mAREX/JGG to a four fixed-focus lens system design problem [15].

The purpose in solving the problem is to identify the lens system that satisfies given design specifications (i.e., the focal length, the F-number and the angle of view) and minimizes the resolution and the distortion of the image. **Fig. 8** shows an example of the lens system. The system consists of four lenses. The decision variables

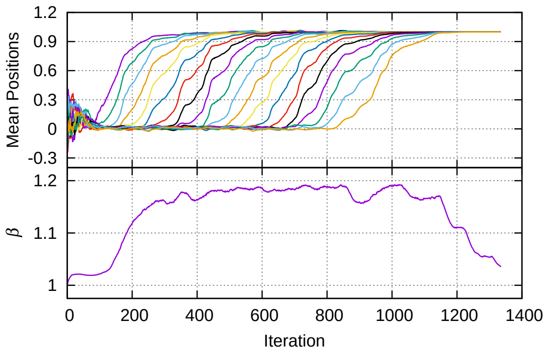

Fig. 7. Transitions of the mean coordinates of the population and β in one typical trial on 20-dimensional Rosenbrock(chain). The horizontal axis shows the generation.

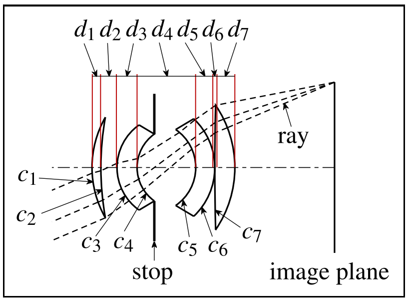

Fig. 8. Example of four fixed-focus lens systems. Variable d_i $(i = 1, \ldots, 7)$ is the distance between the i-th and $(i+1)$-th lens surfaces, and c_i $(i = 1, \ldots, 7)$ is the curvature of the i-th lens surface. The figure depicts the known best solution.

are the distance between two successive lens surfaces, $d_1, d_2, \ldots, d_7$, and the radii of curvature of each lens surface, $c_1, c_2, \ldots, c_7$, which results in a 14-dimensional optimization problem. The distance between the last lens and image plane, and the radius of the last lens are both calculated automatically to satisfy the given focal length.

The problem is formulated as an implicit constrained black-box function optimization problem. The objective function is black-box because the evaluation value is the sum of the resolution and distortion that are obtained by optical simulations. The constraint is implicit because an individual is regarded as infeasible when all rays do not reach the image plane, which is determined only through simulations. The problem is known to have ridge structures and active constraint as well as a multimodal landscape.

We used the design specifications of the lens system in [15]. We set the population size and number of offspring per generation to $[\mu, \lambda] = [25n, 15n]$, respectively, and followed the previous experiments for the other settings. We conducted 20 trials for each method.

Figure 9 shows the convergence curves of successful trials of each method. We found that AEGA approaches the known best solution with approximately 25% fewer number of evaluations than mAREX/JGG on average. In addition, whereas mAREX/JGG reaches the known best solution in seven trials, AEGA reaches it in 11 trials.

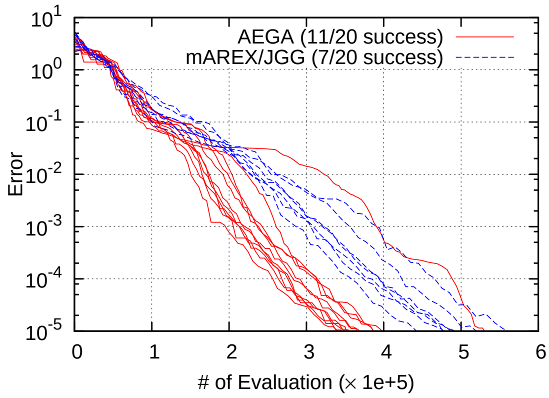

Fig. 9. Transitions of the errors between the best evaluation value in the population and known best in the successful trials of the 20 total. The horizontal axis shows the total number of evaluations.

These results suggest the effectiveness of AEGA.

6. Conclusion

In this study, we proposed a new RCGA called AEGA for black-box function optimization problems with implicit active constraints and ridge structures. This algorithm overcomes the problem of mAREX/JGG. AEGA generates offspring with parents that are chosen randomly from the worst half individuals in the current population and the mean of the best half to extrapolate the population in a more stable manner than in mAREX/JGG. Moreover, AEGA controls the width of the offspring distribution separately in two directions to improve search efficiency. We showed that AEGA found the optimum with fewer evaluations than mAREX/JGG on benchmark problems. The maximum reducing ratios were 55% on the problems with implicit active constraints and 35% on those with ridge structures. In addition, we showed that AEGA can find the known best solution with 25% fewer evaluations on a four fixed-focus lens design problem, which is recognized as a difficult real-world problem.

In a future study, we plan to establish analytical recommended values of user parameters of AEGA to improve the usability of practitioners. Applying AEGA to various classes of benchmark problems including multimodal problems and more challenging real-world problems, while analyzing the behavior of AEGA in the process, are also crucial.

Acknowledgements
This work is partially supported by JSPS KAKENHI.

References:

[1] D. Whitley, M. Lunacek, and J. Knight, "Ruffled by Ridges: How Evolutionary Algorithms Can Fail," Proc. of the 6th Annual Conf. on Genetic and Evolutionary Computation (GECCO'04), pp. 294-306, 2004.

[2] L. Davis, "Handbook of Genetic Algorithms," Van Nostrand Rein-

hold, 1991.

[3] A. H. Wright, "Genetic Algorithms for Real Parameter Optimization," Foundations of Genetic Algorithms, pp. 205-218. Morgan Kaufmann, 1991.

[4] L. Eshelman, "Real-Coded Genetic Algorithms and Interval-Schemata," Foundations of Genetic Algorithms, pp. 187-202, 1993.

[5] I. Ono and S. Kobayashi, "A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover," Proc. of the 7th Int. Conf. on Genetic Algorithms, pp. 246-253, 1997.

[6] S. Tsutsui, M. Yamamura, and T. Higuchi, "Multi-parent recombination with simplex crossover in real coded genetic algorithms," Proc. of the Genetic and Evolutionary Computation Conf. 1999 (GECCO'99), pp. 657-664, 1999.

[7] H. Kita, I. Ono, and S. Kobayashi, "Multi-parental extension of the unimodal normal distribution crossover for real-coded genetic algorithms," Proc. of 1999 Congress on Evolutionary Computation (CEC'99), pp. 1581-1587, 1999.

[8] K. Deb, A. Anand, and D. Joshi, "A computationally efficient evolutionary algorithm for real-parameter optimization," Evolutionary Computation, Vol.10, No.4, pp. 371-395, 2002.

[9] S. Kobayashi, "The frontiers of real-coded genetic algorithms," Trans. of the Japanese Society for Artificial intelligence, Vol.24, No.1, pp. 147-162, 2009 (in Japanese).

[10] T. Bäck, F. Hoffmeister, and H. P. Schwefel, "A Survey of Evolution Strategies," Proc. of the 4th Int. Conf. on Genetic Algorithms, pp. 2-9, 1991.

[11] H. P. Schwefel, "Numerical Optimization of Computer Models," John Wiley & Sons, Inc., 1981.

[12] Y. Akimoto, Y. Nagata, J. Sakuma, I. Ono, and S. Kobayashi, "Proposal and evaluation of adaptive real-coded crossover AREX," Trans. of the Japanese Society for Artificial Intelligence, Vol.24, No.6, pp. 446-458, 2009 (in Japanese).

[13] Y. Akimoto, J. Sakuma, I. Ono, and S. Kobayashi, "Adaptation of expansion rate for real-coded crossovers," Proc. of the 11th Annual Conf. on Genetic and Evolutionary Computation (GECCO'09), pp. 739-746, 2009.

[14] K. Uemura, N. Nakashima, Y. Nagata, and I. Ono, "A New Real-coded Genetic Algorithm for Implicit Constrained Black-box Function Optimization," Proc. of 2013 IEEE Congress on Evolutionary Computation (CEC'13), pp. 2887-2894, 2013.

[15] I. Ono, S. Kobayashi, and K. Yoshida, "Optimal lens design by real-coded genetic algorithms using UNDX," Computer Methods in Applied Mechanics and Engineering, Vol.186, No.2-4, pp. 483-497, 2000.

[16] D. V. Arnold and N. Hansen, "A $(1+1)$-CMA-ES for Constrained Optimisation," Proc. of the 8th Annual Conf. on Genetic and Evolutionary Computation (GECCO'12), pp. 297-304, 2012.

[17] Z. Michalewicz and M. Schoenauer, "Evolutionary Algorithms for Constrained Parameter Optimization Problems," Evolutionary Computation, Vol.4, pp. 1-32, 1996.

[18] D. G. Luenberger and Y. Ye, "Linear and Nonlinear Programming," Int. Series in Operations Research & Management Science, Vol.116, Springer, 3 edition, 2008.

Appendix A. Derivation of τ

We derive τ in Eq. (10), which is defined as the maximum value of $\tau_{\boldsymbol{q}}$ in Eq. (9).
Proof. We define the maximum problem as:

$$\tau := \max_{q \in Q} \tau_{\boldsymbol{q}} = \max_{q \in Q} \frac{\boldsymbol{q}^{\mathrm{T}} \mathbf{M}_{\mathrm{s}} \boldsymbol{q}}{\boldsymbol{q}^{\mathrm{T}} \mathbf{M}_{\mathrm{y}} \boldsymbol{q}} - 1, \quad \ldots\ldots \quad (17)$$

where Q is the set of all orthogonal directions. We can ignore the scalar multiplication of $\boldsymbol{q}$ because $\tau_{\boldsymbol{q}}$ is invariant against the scalar multiplication of $\boldsymbol{q}$. Here, we choose:

$$Q := \{\boldsymbol{q} \mid \boldsymbol{a}^{\mathrm{T}} \boldsymbol{q} = 0,\ \boldsymbol{q}^{\mathrm{T}} \mathbf{M}_{\mathrm{y}} \boldsymbol{q} = 1\}$$

for the definition of Q. This then leads to

$$\begin{aligned} \tau + 1 &= \max_{q \in Q} \frac{\boldsymbol{q}^{\mathrm{T}} \mathbf{M}_{\mathrm{s}} \boldsymbol{q}}{\boldsymbol{q}^{\mathrm{T}} \mathbf{M}_{\mathrm{y}} \boldsymbol{q}} \\ &= \max_{q \in Q} \boldsymbol{q}^{\mathrm{T}} \mathbf{M}_{\mathrm{s}} \boldsymbol{q} \\ &= \max_{\boldsymbol{b}^{\mathrm{T}} \tilde{\boldsymbol{q}} = 0, \tilde{\boldsymbol{q}}^{\mathrm{T}} \tilde{\boldsymbol{q}} = 1} \tilde{\boldsymbol{q}}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_{\mathrm{s}} \mathbf{L}^{-\mathrm{T}} \tilde{\boldsymbol{q}}, \quad \ldots\ldots \quad (18) \end{aligned}$$

where $\mathbf{L}$ is the Cholesky decomposition of $\mathbf{M}_{\mathrm{y}}$ (i.e., $\mathbf{M}_{\mathrm{y}} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$) and $\boldsymbol{b}$ and $\tilde{\boldsymbol{q}}$ are defined as $\boldsymbol{b} := \mathbf{L}^{-1} \boldsymbol{a}$ and $\tilde{\boldsymbol{q}} := \mathbf{L}^{\mathrm{T}} \boldsymbol{q}$, respectively. We can derive Eq. (10) by means of the method of Lagrange multipliers [18] by defining the Lagrangian as:

$$L(\tilde{\boldsymbol{q}}, \lambda_1, \lambda_2) := \tilde{\boldsymbol{q}}^{\mathrm{T}} \mathbf{L}^{-1} \mathbf{M}_{\mathrm{s}} \mathbf{L}^{-\mathrm{T}} \tilde{\boldsymbol{q}} - \lambda_1 \boldsymbol{b}^{\mathrm{T}} \tilde{\boldsymbol{q}} - \lambda_2 (\tilde{\boldsymbol{q}}^{\mathrm{T}} \tilde{\boldsymbol{q}} - 1),$$

where λ_1 and λ_2 are the Lagrange multipliers. □

Name:
Kento Uemura

Affiliation:
Doctoral Student, Department of Computational Intelligence and Systems Science, Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology (Present: Fujitsu Laboratories Ltd.)

Address:
4259 Nagatsuta-cho, Midori-ku, Yokohama 226-8502, Japan

Brief Biographical History:
2010-2012 Master's Student, Tokyo Institute of Technology
2012-2015 Doctoral Student, Tokyo Institute of Technology
2012-2015 Research Fellowship for Young Scientists (DC1) of the Japan Society for the Promotion of Science
2015- Fujitsu Laboratories Ltd.

Main Works:
• K. Uemura, S. Kinoshita, Y. Nagata, S. Kobayashi, and I. Ono, "Big-valley Explorer: A Framework of Real-coded Genetic Algorithms for Multi-funnel Function Optimization," Trans. of the Japanese Society for Evolutionary Computation, Vol.4, No.1, pp. 1-12, 2013.

Membership in Academic Societies:
• The Japanese Society for Evolutionary Computation (JPNSEC)

Name:
Isao Ono

Affiliation:
Associate Professor, Department of Computational Intelligence and Systems Science, Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology

Address:
4259 Nagatsuta-cho, Midori-ku, Yokohama 226-8502, Japan

Brief Biographical History:
1998 Assistant Professor, University of Tokushima
1998-2001 Lecturer, University of Tokushima
2001-2005 Associate Professor, University of Tokushima
2005- Associate Professor, Tokyo Institute of Technology

Main Works:
• T. Shioda and I. Ono, "Adaptive Weighted Aggregation with Step Size Control Weight Adaptation for Multiobjective Continuous Optimization," SICE J. of Control, Measurement, and System Information, Vol.8, No.5, 2015.

Membership in Academic Societies:
• The Japanese Society for Evolutionary Computation (JPNSEC)
• The Society of Instrument and Control and Engineering (SICE)
• The Institute of Electrical Engineering of Japan (IEEJ)
• The Iron and Steel Institute of Japan (ISIJ)
• The Japanese Society for Artificial Intelligence (JSAI)
• The Institute of Systems, Control and Information Engineering (ISCIE)