New Encoding/Converting Methods of Binary GA/Real-Coded GA

Jong-Wook KIM† and Sang Woo KIM††, Members

SUMMARY This paper presents new encoding methods for the binary genetic algorithm (BGA) and new converting methods for the real-coded genetic algorithm (RCGA). These methods are developed for the specific case in which some parameters have to be searched in wide ranges since their actual values are not known. The oversampling effect which occurs at large values in the wide range search are reduced by adjustment of resolutions in mantissa and exponent of real numbers mapped by BGA. Owing to an intrinsic similarity in chromosomal operations, the proposed encoding methods are also applied to RCGA with remapping (converting as named above) from real numbers generated in RCGA. A simple probabilistic analysis and benchmark with two ill-scaled test functions are carried out. System identification of a simple electrical circuit is also undertaken to testify effectiveness of the proposed methods to real world problems. All the optimization results show that the proposed encoding/converting methods are more suitable for problems with ill-scaled parameters or wide parameter ranges for searching.

key words: binary genetic algorithm, real-coded genetic algorithm, ill-scaled function, system identification, encoding

1. Introduction

System identification is the field of mathematical modelling of systems from experimental data, and is categorized into nonparametric and parametric methods according to model structures [1]. In the case the model structure is considerably complex or largely scaled like power transformers, nonparametric identification is deemed to be suitable. Nonparametric identification is characterized by the property that the resulting models are curves or functions, and its good example is the spectral analysis for the diagnosis of a power transformer [2].

On the other hand, parametric identification is favorable to a system whose model structure is simply described with mathematical static and/or dynamic equations. Most researches in parametric identification area focus on discrete-time systems due to the easiness and simplicity of required mathematics. However, the performance of discrete-time identification depends on the sampling time, and the obtained parameters give little physical insight into the system. Therefore, in the case of diagnosis or modeling, transformation of parameters in a discrete-time system into those in a continuous-time system is required, which is quite cumbersome or even impossible as system dimension is raised. To this end, continuous-time prediction error method free from parameter transformation was provided, which estimates the system parameters by minimizing prediction errors with exact derivatives of the objective function with respect to the adjustable parameters [3]. However, the prediction error method requires a good guess of initial parameters, and update rule design is system dependent.

The genetic algorithm whose chromosome is formed with model parameters and initial states is applied to the parametric identification of a simple electrical circuit [4] and an induction motor [5]. These systems contain physical electrical elements such as resistors, capacitors, and inductors whose values range from micro to kilo units. If actual values of each element are known a priori, which is unrealistic, parameter bounds for search with GA can be set up moderately. For the real systems however it is difficult to determine proper bounds owing to time variance of actual values or inaccuracy of name plate values. Therefore, the parameter bounds have to be selected wide enough not to exclude global minima prior to executing GA, while it is widely acknowledged that as parameter space grows wider optimization performance degenerates as much. The delta coding is an efficient alternative for this problem, but uses a multistage strategy where the global minima can be discarded at an initial search phase [6].

This paper proposes new encoding methods of BGA specifically designed to overcome the wide bound search problem. As search space grows wider, the general linear sampling (or linear encoding) of BGA suffers from inevitable unbalance between small and large numbers owing to the limited chromosome length, i.e., resolution of real numbers. In this paper logarithmic and hybrid encoding methods for search space are developed by modifying the conventional encoding (or decoding, equivalently) function of BGA. These changes in phenotypes enlarge the small numbers often ignored and reduce large numbers overly estimated both in linear sampling. Furthermore, the proposed encoding methods include the linear encoding method such that for narrow search space the effects of all the encoding methods are almost identical.

Since the electrical circuit elements addressed in this paper are assigned positive real values, RCGA is more efficient than BGA in terms of search space resolution, run time, simplicity, convenience of program coding, and so on. RCGA has strings of real numbers instead of binary chro-
The general decoding function which linearly transforms an
l-bit-long binary string \((a_1 \cdots a_l)\) into a bounded real number \(x \in [x^L, x^U]\) is written as

\[
x = x^U - x^L \sum_{i=0}^{l-1} a_{l-i} 2^i + x^L
\]

(2)

where \(a_1\) is a most significant bit (MSB) and \(a_l\) is a least significant bit (LSB). Specifically, \((x^U - x^L)/(2^l - 1)\) plays a role as search resolution or precision. To avoid terminological ambiguity hereafter, definitions are stated for each type of sampled space. The first is linear space.

**Definition 1 (Linear Space):**

The \(n\)-dimensional discretized linear space with finite resolution \(\delta_L\) is defined as:

\[
L : \{ x | x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n \}
\]

(3)

\[
x_i = x_i^L + k_\delta L, \quad i = 1, \ldots, n
\]

\[
k \in \{0, 1, \ldots, \left\lfloor \frac{x_i^U - x_i^L}{\delta_L} \right\rfloor \}
\]

where \(x_i^U\) and \(x_i^L\) are predetermined upper and lower bounds of \(x_i\), respectively.

The followings are the proposed encoding methods of BGA with corresponding definitions. One is exponential encoding, which transforms a binary string \((b_1 \cdots b_l)\) into an exponent of 10 as

\[
y = 10^{\frac{1}{p-1} \left( \sum_{i=0}^{p-1} b_i 2^i \right)}
\]

(4)

where \(e_l = \log_{10} y^l\) and \(e^U = \log_{10} y^U\). In (4), \(y\) is regarded as equidistant in terms of logarithmic scale. Thus, the corresponding space is denominated logarithmic space rather than exponential space.

**Definition 2 (Logarithmic Space):**

An \(n\)-dimensional discretized logarithmic space with finite resolution \(\delta_G\) is defined as:

\[
G : \{ y | y = (y_1, \ldots, y_n)^T \in \mathbb{R}^n \}
\]

(5)

\[
y_i = 10^{e^{i, U} + k_\delta G}, \quad i = 1, \ldots, n
\]

\[
k \in \{0, 1, \ldots, \left\lfloor \frac{e^{i, U} - e^{i, L}}{\delta_G} \right\rfloor \}
\]

where \(e^{i, U} = \log_{10} y^{i, U}\) and \(e^{i, L} = \log_{10} y^{i, L}\) are the exponents of upper and lower bounds of \(y_i\), respectively.

The other is hybrid encoding, which is also a common way to represent floating-point numbers by a fixed-length memory. To implement the hybrid encoding method, a binary string must be partitioned into two parts; mantissa and exponent. If a majority of bits are assigned to a mantissa, this encoding turns closely into a linear encoding. On the other hand, if almost all bits are referred to as those for an exponent, this is approximately exponential encoding. Therefore, we conjecture that the hybrid encoding lies between linear and logarithmic encodings depending on the bit ratio of mantissa and exponent.

Suppose we assign \(p\) and \(q\) bits for the exponent and the mantissa of a real number, respectively, and a binary string is described as \((c_1 \cdots c_p d_1 \cdots d_q)\), whose total length is \(l = p + q\). The hybrid encoding is described as

\[
z = \text{mant} \times 10^{\text{expo}},
\]

(5)

\[
\text{expo} = \delta_e \sum_{i=0}^{p-1} c_{(p-i)} 2^i + \text{expo}, \quad \delta_e = \frac{e^U - e^L}{2^p}
\]

(6)

\[
\text{mant} = \delta_m \sum_{j=0}^{q-1} d_{(q-j)} 2^j + \text{mant}, \quad \delta_m = \frac{10^k - 1}{2^q}
\]

(7)

where \(\delta_e\) and \(\delta_m\) are the resolutions of exponent and mantissa, respectively, that are devised to avoid overlapping of decoding. The details of how \(\delta_e\) and \(\delta_m\) are calculated are provided in Appendix. The hybrid space is defined in accordance with (5)–(7).

**Definition 3 (Hybrid Space):**

An \(n\)-dimensional discretized hybrid space with finite resolution vector \(\delta_H = (\delta_m, \delta_e)^T\) is defined as:

\[
H : \{ z | z = (z_1, \ldots, z_n)^T \in \mathbb{R}^n \}
\]

\[
z_i = (1 + l \delta_m) \times 10^{e^{i, U} + k_\delta_G}, \quad i = 1, \ldots, n
\]

\[
k \in \{0, 1, \ldots, \left\lfloor \frac{e^{i, U} - e^{i, L}}{\delta_e} \right\rfloor \}
\]

\[
l \in \{0, 1, \ldots, \left\lfloor \frac{\delta_m^l - 1}{\delta_m} \right\rfloor \}
\]

(8)

where \(e^{i, U} = \log_{10} z^{i, U}\) and \(e^{i, L} = \log_{10} z^{i, L}\) are the exponents of upper and lower bounds of \(z_i\), respectively.

3.2 Converting Methods of RCGA

Since individuals of RCGA consist of real numbers unlike BGA, the conventional RCGAs require no encoding methods. However, we apply in parallel to RCGA the concepts of the linear, exponential, and hybrid encoding methods based on the intrinsic similarities between BGA and RCGA. A new term for ‘encoding’ is denominated ‘conversion’ for RCGA. Then, except the linear conversion, exponential and hybrid conversion are accomplished using converting functions imitating (4)–(7).

The exponential conversion of RCGA is described as

\[
y_i = 10^{e_i}, \quad r_i \in [\log_{10} y_i^L, \log_{10} y_i^U]
\]

(8)

where \(r_i\) is an \(i\)-th gene of RCGA, and the hybrid conversion for \(r_i\) is written as

\[
z_i = (9 h_i + 1) 10^n, \quad h_i = r_i - n_i, \quad n_i = \lfloor r_i \rfloor, \quad 0 \leq h_i < 1.
\]

(9)

(10)

As shown in (8)–(10), converting functions of RCGA are considerably simple compared with the encoding functions of BGA.
expanded by factors of $F$ constant irrespective of the size of search space. divided by a finite number of grids, whose amount remains mutation. Therefore, it can be regarded that search space is on the resolution of random numbers during crossover and number of bits per parameter, while in RCGA it depends space resolution. In BGA, the resolution depends on the searching for a whole solution space whose dimension equals the number of parameters, both the GAs have finite space resolution. In BGA, the resolution depends on the number of bits per parameter, while in RCGA it depends on the resolution of random numbers during crossover and mutation. Therefore, it can be regarded that search space is divided by a finite number of grids, whose amount remains constant irrespective of the size of search space.

For a brief geometric analysis, we define the level set $C$ as

$$C : \{\theta \mid F(\theta) \geq F_0, \theta \in \mathbb{R}^n\}$$

(11)

where $\theta$ is the $n$-dimensional parameter vector, $F(\cdot)$ is the fitness function, and $F_0$ is the constant fitness value. Note that a sufficiently large $F_0$ makes $C$ a single and closed set even though a multimodal optimization problem is addressed.

For an explanation of space expansion, Fig. 1(a) illustrates an optimization problem, whose parameters are $p^1_1 \in [p^1_{1\text{max}}, p^1_{1\text{min}}]$ and $p^2_1 \in [p^2_{1\text{min}}, p^2_{1\text{max}}]$ in a feasible space $S^1$. Let the only upper bounds of each parameter be expanded by factors of $k_1$ and $k_2$, respectively. Then new bounds are described as

$$
\begin{align*}
    p^1_{1\text{max}} &= k_1 p^1_{1\text{max}}, \\
    p^2_{1\text{max}} &= k_2 p^2_{1\text{max}}, \\
    p^1_{1\text{min}} &= p^1_{1\text{min}}, \\
    p^2_{1\text{min}} &= p^2_{1\text{min}}
\end{align*}
$$

(12)

where superscripts 1 and 2 represent ‘before expansion’ and ‘after expansion,’ respectively. Fig. 1(b) shows an expanded solution space $S^2$, where it is notable that despite the expansion of the feasible space the crest set $C$ remains the same. Thus, the relative portion of $C$ decreases for $S^2$, and this contraction may cause difficulty in exploring solutions inside $C$ in terms of probability. A relative probability between the two cases, i.e. before and after expansion, is computed for a 2-dimensional problem as

$$
\frac{P_2(C)}{P_1(C)} = \frac{A(C)/A(S^2)}{A(C)/A(S^1)} = \frac{A(S^1)}{A(S^2)}
$$

(13)

where $P_1(C)$ and $P_2(C)$ represent the probabilities of selecting solutions inside $C$ for corresponding search spaces $S^1$ and $S^2$, respectively, and $A(\cdot)$ stands for area. In the case of $p^1_{1\text{max}} \gg p^1_{1\text{min}}$ and $p^2_{1\text{max}} \gg p^2_{1\text{min}}$, (13) is simplified as

$$
\frac{P_2(C)}{P_1(C)} = \frac{(p^1_{1\text{max}} - p^1_{1\text{min}})k_1k_2}{k_1k_2(p^1_{1\text{max}} - \frac{1}{k_1}p^1_{1\text{min}})k_1k_2(p^2_{1\text{max}} - \frac{1}{k_2}p^2_{1\text{min}})}
\approx \frac{1}{k_1k_2}.
$$

(14)

Thus in an $n$-dimensional problem (13) can be generalized as

$$
\frac{P_2(C)}{P_1(C)} \approx \frac{1}{k_1k_2 \cdots k_n}.
$$

(15)

This result indicates that the linear scale expansion of upper bounds in each parameter can cause performance degradation of the random search during the initialization and reproduction operations of GA. However, owing to the intrinsic mechanism of survival of the fittest, the overall performance of GA is not deteriorated with as much ratio as (15).

To analyze the relative probability (13) for the logarithmic space, the lower and upper bounds in (12) are interpreted on a logarithmic scale as

$$
\begin{align*}
    p^1_{1\text{max}} &= \log k_1 + \log p^1_{1\text{max}} = \log k_1 + p^1_{1\text{max}}, \\
    p^2_{1\text{max}} &= \log k_2 + \log p^2_{1\text{max}} = \log k_2 + p^2_{1\text{max}}, \\
    p^1_{1\text{min}} &= \log p^1_{1\text{min}} = p^1_{1\text{min}}, \\
    p^2_{1\text{min}} &= \log p^2_{1\text{min}} = p^2_{1\text{min}}
\end{align*}
$$

(16)

where the superscript $L$ stands for a logarithmic operation. The relative probability on logarithmic space is computed in a similar manner as

$$
\frac{P_2(C^L)}{P_1(C^L)} = \frac{(p^1_{1\text{max}}^L - p^1_{1\text{min}}^L)(p^2_{1\text{max}}^L - p^2_{1\text{min}}^L)}{(p^1_{1\text{max}}^L - p^1_{1\text{min}}^L)(p^2_{2\text{max}}^L - p^2_{2\text{min}}^L)}
\approx \frac{(\log k_1 + p^1_{1\text{max}}^L)(\log k_2 + p^2_{1\text{max}}^L)}{(\log k_1 + p^1_{1\text{min}}^L)(\log k_2 + p^2_{1\text{min}}^L)}.
$$

(17)
For a more intuitive simplification, if log-scaled expansion ratios are defined as
\[
m_1 = \frac{\log k_1}{p_{1\text{max}}^{1L} - p_{1\text{min}}^{1L}}, \quad m_2 = \frac{\log k_2}{p_{2\text{max}}^{1L} - p_{2\text{min}}^{1L}}
\]
then (17) is simplified as
\[
P_{2}(CL) = \frac{1}{(m_1 + 1)(m_2 + 1)}.
\]

For a simple comparison of (15) and (19) suppose that \(p_1\) and \(p_2\) were presumed to exist between 1 and 100 and 0.001 and 0.1, respectively, but search results show that those ranges fail to include acceptable solutions. As the next step the upper bounds are expanded simultaneously by a factor of 10, which leads the relative probability to enormous decrease of about 1/10000 for linear space, while only a slight decrease of 1/4 for logarithmic space. Thus, logarithmic space is far more reasonable for searching a wide solution space.

4. Experimental Results

4.1 Test Functions and Performance Measure

After developing an optimization method, one has to test it on a suite of benchmark functions at a validation step in accordance with the 'no-free-lunch' theorem [12]. In the literature of evolutionary computation [13]–[15], 14 or 23 test functions, varying from unimodal to multimodal and from low- to high-dimensional cost functions, are provided with their global optima acknowledged. However, the test functions are confined in well-scaled search space, i.e., their bounds are of similar order. Therefore, badly scaled functions are necessary for the specific optimization problem addressed in this paper.

This paper adopts the following functions for test of the proposed encoding/converting methods:

- Brown badly scaled function:
  \[
  f_b(x_1, x_2) = (x_1 - 10^6)^2 + (x_2 - 2 \times 10^{-6})^2 \\
  + (x_1x_2 - 2)^2
  \] (20)
- Powell badly scaled function:
  \[
  f_p(x_1, x_2) = (10^4x_1x_2 - 1)^2 + (\exp[-x_1] \\
  + \exp[-x_2] - 1.0001)^2
  \] (21)

The global minima of (20) and (21) are known to be exactly and almost 0 at the points \((x_1, x_2) = (10^{6}, 2 \times 10^{-6})\) and \((1.098 \cdots 10^{-5}, 9.106 \cdots)\), respectively. Search is undertaken within \([10^{-8}, 10^{8}]\) for (20) and \([10^{-8}, 10^{4}]\) for (21), whose upper bounds are huge values for general optimization problems.

Rudolph [18] has shown that the classical BGA never converges to a global optimum, while the modified versions of maintaining the best solution in the population do. Therefore the adaptive GA (AGA) scheme [19] is adopted for BGA in this paper. GA parameters are set as

- maximum generation = 2000
- number of variables = 2
- population size = 50
- number of restarts = 50
- selection: Roulette wheel selection
- bit length of a substring in BGA = 13
- bit ratio of exponent and mantissa in hybrid encoding of BGA = 4/9
- parameters of adaptive BGA:
  \(k_1 = 0.85, k_2 = 0.5, k_3 = 1.0, k_4 = 0.05\)
- crossover of BGA: one-point crossover
- crossover of RCGA: modified simple crossover
- mutation of BGA and RCGA: uniform mutation
- crossover rate of RCGA = 0.95
- mutation rate of RCGA = 0.01.

For a performance comparison, the following measures are used:

a) MEAN BEST COST: mean of the best costs obtained at each restart.
b) STD DEV: standard deviation of the best costs obtained at each restart.
c) LM BEST COST: logarithmic mean of the best costs.
d) FINAL BEST COST: best of the best costs obtained at each restart.
e) MEAN BEST SOL: mean of the best solutions obtained at each restart.
f) LM BEST SOL: logarithmic mean of each best solution.
g) FINAL BEST SOL: solution for d).

In the above measures, LM (logarithmic mean) represents exponent averaging. That is, the LM of \(x = (x_1, x_2, \cdots, x_m)^T\) is described as
\[
\text{LM}(x) = 10^{\frac{\sum_{i=1}^{m} \log_{10} x_i}{m}}
\] (22)
which is quite reasonable for averaging real numbers whose magnitudes are different in their exponents. For example, the general averaging ignores such a small number as \(10^{-8}\) in presence of a relatively large number of \(10^6\), but the LM balances the two numbers with their exponents. Thus, LM is thought to be a more reliable medium for a set of numbers with large variance.

Table 1 gives performance comparison of the proposed encoding/converting methods on the Brown function for each solution space. Note in the case of RCGA that MEAN BEST COST values are quite different from LM BEST COST values owing to an ill-scaled property of attained cost values. This discrepancy increases as GA finds more numbers of bad solutions of high cost values.

In linear solution space, both GAs fail to find out a
true value \( x_2, 2 \times 10^{-6} \), and just approximate it as 0 because of limited resolution around such a small value. However, in logarithmic and hybrid search spaces, the precise value of \( x_2 \) is obtained, which shows the effectiveness of the proposed encoding/converting methods for normally small and large numbers. Moreover, RCGA searches for better solutions than BGA on logarithmic and hybrid spaces. As is evidenced in the literature, RCGA, which has higher search resolution, is more suitable for nonlinear optimization problems, and the data in Table 1 verifies this result.

Table 2 provides search results of the Powell function under the same search condition with Table 1. Unlike the Brown function, the Powell function appears to contain local minima at \((x_1, x_2) = (10.1, 9.88 \times 10^{-6})\) as shown in logarithmic space with RCGA. The diversity of LM BEST SOL values also demonstrates the ruggedness of this function. Despite this difficulty, the proposed methods have found acceptable solutions among which a global minimum is \((x_1, x_2) = (1.1035 \times 10^{-5}, 9.9624)\) with the cost of 2.41 \times 10^{-11} found in hybrid space by RCGA.

### 4.2 System Identification of a Simple Electrical Circuit

The main concern of this paper is to identify a simple electrical circuit whose elements are a resistor with resistance \( R \), an inductor with inductance \( L \), and a capacitor with capacitance \( C \), as shown in Fig. 2. The state space and output equations of this system are given as

\[
\begin{align*}
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{bmatrix} &= 
\begin{bmatrix}
-\frac{R}{L} & -\frac{1}{C} \\
\frac{1}{C} & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
+ 
\begin{bmatrix}
\frac{1}{L} \\
0
\end{bmatrix} u,
\end{align*}
\]

\[
y = 
\begin{bmatrix}
1 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
\]

where \( x_1 \) is input current and \( x_2 \) is capacitor voltage. Because initial state is necessary for determining unique states at each time, parameter vector in (11) is determined as

\[
\theta = (R \ L \ C \ x_1(0) \ x_2(0))^T.
\]

The fitness function in GAs is designed to minimize the sum squared estimation error as

\[
F(\theta) = \frac{1}{N} \sum_{i=1}^{N} (y_i(\theta) - y_i(\theta_0))^2
\]

where \( N \) is the number of data points and the true parameter vector is

\[
\theta_0 = (1.22 \times 10^4 \ 9.60 \times 10^{-2} \ 5.00 \times 10^{-5} \ 7.28 \times 10^{-4} \ 2.98 \times 10^{-1} )^T.
\]

Input signal used for system identification is PRBS (Pseudo-Random Binary Signal) [1], and its characteristic parameters are given in Table 3 where \( N_r \) is the number of shifting registers, \( p \) is the frequency divider, and \( T_s \) is the sampling period in seconds.

<table>
<thead>
<tr>
<th>GA</th>
<th>SPACE</th>
<th>MEAN BEST COST</th>
<th>STD DEV</th>
<th>LM BEST COST</th>
<th>FINAL BEST SOL</th>
<th>MEAN BEST SOL</th>
<th>LM BEST SOL</th>
<th>FINAL BEST SOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGA</td>
<td>LINEAR</td>
<td>3.45 \times 10^{-3}</td>
<td>1.26 \times 10^{-2}</td>
<td>9.27 \times 10^{-4}</td>
<td>1.22 \times 10^{-6}</td>
<td>6.88 \times 10^{-3}</td>
<td>2.92 \times 10^{-3}</td>
<td>2.48</td>
</tr>
<tr>
<td></td>
<td>LOG</td>
<td>4.94 \times 10^{-6}</td>
<td>5.59 \times 10^{-6}</td>
<td>2.13 \times 10^{-7}</td>
<td>2.60 \times 10^{-10}</td>
<td>7.92 \times 10^{-6}</td>
<td>1.18 \times 10^{-6}</td>
<td>5.31</td>
</tr>
<tr>
<td></td>
<td>HYBRID</td>
<td>2.02 \times 10^{-6}</td>
<td>3.58 \times 10^{-6}</td>
<td>3.64 \times 10^{-7}</td>
<td>4.17 \times 10^{-9}</td>
<td>1.20 \times 10^{-6}</td>
<td>7.15 \times 10^{-6}</td>
<td>1.56</td>
</tr>
<tr>
<td>RCGA</td>
<td>LINEAR</td>
<td>8.99 \times 10^{-11}</td>
<td>3.73 \times 10^{-2}</td>
<td>3.73 \times 10^{-4}</td>
<td>1.00 \times 10^{-11}</td>
<td>3.89 \times 10^{-11}</td>
<td>1.29 \times 10^{-11}</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>LOG</td>
<td>5.82 \times 10^{-13}</td>
<td>2.12 \times 10^{-3}</td>
<td>4.49 \times 10^{-7}</td>
<td>4.85 \times 10^{-13}</td>
<td>1.97 \times 10^{-13}</td>
<td>1.26 \times 10^{-13}</td>
<td>1.95</td>
</tr>
<tr>
<td></td>
<td>HYBRID</td>
<td>6.19 \times 10^{-13}</td>
<td>4.68 \times 10^{-3}</td>
<td>1.16 \times 10^{-13}</td>
<td>2.43 \times 10^{-11}</td>
<td>1.43 \times 10^{-13}</td>
<td>6.87</td>
<td>5.91 \times 10^{-14}</td>
</tr>
</tbody>
</table>

Table 3 Characteristics of PRBS.

<table>
<thead>
<tr>
<th>( N_r )</th>
<th>( p )</th>
<th>( T_s )</th>
<th>( N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>17</td>
<td>100 \times 10^{-6}</td>
<td>1500</td>
</tr>
</tbody>
</table>
Table 4  Comparison of cost values between BGA and RCGA on linear, logarithmic, and hybrid search space with respect to narrow, medium, and wide parameter ranges.

<table>
<thead>
<tr>
<th>SPACE</th>
<th>BOUND</th>
<th>BGA</th>
<th></th>
<th></th>
<th>RCGA</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MEAN</td>
<td>STD</td>
<td>LM</td>
<td>FINAL</td>
<td>MEAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BEST</td>
<td>DEV</td>
<td>BEST</td>
<td>COST</td>
<td>BEST</td>
</tr>
<tr>
<td>LINEAR</td>
<td>NARROW</td>
<td>2.81×10⁻⁴</td>
<td>2.45×10⁻⁴</td>
<td>1.44×10⁻⁴</td>
<td>1.98×10⁻⁴</td>
<td>2.86×10⁻⁴</td>
</tr>
<tr>
<td></td>
<td>MEDIUM</td>
<td>2.30×10⁻³</td>
<td>2.05×10⁻³</td>
<td>1.22×10⁻³</td>
<td>4.63×10⁻⁴</td>
<td>7.08×10⁻²</td>
</tr>
<tr>
<td></td>
<td>WIDE</td>
<td>9.37×10⁻³</td>
<td>9.11×10⁻³</td>
<td>4.55×10⁻³</td>
<td>4.69×10⁻³</td>
<td>8.89×10⁻²</td>
</tr>
<tr>
<td>LOG</td>
<td>NARROW</td>
<td>3.11×10⁻⁴</td>
<td>3.55×10⁻⁴</td>
<td>1.05×10⁻⁴</td>
<td>3.19×10⁻⁴</td>
<td>5.51×10⁻²</td>
</tr>
<tr>
<td></td>
<td>MEDIUM</td>
<td>1.86×10⁻³</td>
<td>3.33×10⁻³</td>
<td>3.96×10⁻³</td>
<td>4.84×10⁻⁴</td>
<td>3.87×10⁻⁴</td>
</tr>
<tr>
<td></td>
<td>WIDE</td>
<td>1.98×10⁻³</td>
<td>3.79×10⁻³</td>
<td>4.10×10⁻⁴</td>
<td>6.96×10⁻⁶</td>
<td>2.17×10⁻⁴</td>
</tr>
<tr>
<td>HYBRID</td>
<td>NARROW</td>
<td>1.97×10⁻⁴</td>
<td>2.91×10⁻⁴</td>
<td>4.31×10⁻⁵</td>
<td>4.40×10⁻⁴</td>
<td>5.22×10⁻⁴</td>
</tr>
<tr>
<td></td>
<td>MEDIUM</td>
<td>2.18×10⁻³</td>
<td>2.88×10⁻³</td>
<td>6.36×10⁻⁴</td>
<td>8.35×10⁻⁵</td>
<td>3.07×10⁻⁴</td>
</tr>
<tr>
<td></td>
<td>WIDE</td>
<td>2.49×10⁻³</td>
<td>3.72×10⁻³</td>
<td>4.70×10⁻⁴</td>
<td>1.41×10⁻⁵</td>
<td>4.47×10⁻⁴</td>
</tr>
</tbody>
</table>

Table 5  Comparison of obtained solutions between BGA and RCGA on linear, logarithmic, and hybrid search space with respect to narrow, medium, and wide parameter ranges. The true values are R=12.20 (Ω), L=96.00 (mH), and C=50.00 (μF).

<table>
<thead>
<tr>
<th>SPACE</th>
<th>BOUND</th>
<th>BGA</th>
<th></th>
<th></th>
<th>RCGA</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MEAN</td>
<td>STD</td>
<td>LM</td>
<td>FINAL</td>
<td>MEAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BEST</td>
<td>DEV</td>
<td>BEST</td>
<td>COST</td>
<td>BEST</td>
</tr>
<tr>
<td>LINEAR</td>
<td>NARROW</td>
<td>12.90</td>
<td>12.21</td>
<td>94.90</td>
<td>96.53</td>
<td>50.63</td>
</tr>
<tr>
<td></td>
<td>MEDIUM</td>
<td>12.71</td>
<td>12.22</td>
<td>94.49</td>
<td>98.57</td>
<td>51.38</td>
</tr>
<tr>
<td></td>
<td>WIDE</td>
<td>14.25</td>
<td>11.99</td>
<td>94.91</td>
<td>93.78</td>
<td>53.49</td>
</tr>
<tr>
<td>LOG</td>
<td>NARROW</td>
<td>12.23</td>
<td>12.20</td>
<td>93.08</td>
<td>95.71</td>
<td>51.81</td>
</tr>
<tr>
<td></td>
<td>MEDIUM</td>
<td>12.94</td>
<td>12.20</td>
<td>103.5</td>
<td>95.98</td>
<td>46.11</td>
</tr>
<tr>
<td></td>
<td>WIDE</td>
<td>12.33</td>
<td>12.18</td>
<td>101.83</td>
<td>95.12</td>
<td>47.29</td>
</tr>
<tr>
<td>HYBRID</td>
<td>NARROW</td>
<td>12.16</td>
<td>12.20</td>
<td>93.51</td>
<td>96.07</td>
<td>51.51</td>
</tr>
<tr>
<td></td>
<td>MEDIUM</td>
<td>12.74</td>
<td>12.24</td>
<td>98.63</td>
<td>94.87</td>
<td>48.73</td>
</tr>
<tr>
<td></td>
<td>WIDE</td>
<td>12.57</td>
<td>12.18</td>
<td>105.61</td>
<td>94.66</td>
<td>45.30</td>
</tr>
</tbody>
</table>

Table 6  Parameter bounds.

<table>
<thead>
<tr>
<th>PARAMETERS</th>
<th>NARROW</th>
<th>MEDIUM</th>
<th>WIDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>1 × 10⁰</td>
<td>1 × 10⁻⁷</td>
<td>1 × 10⁻⁴</td>
</tr>
<tr>
<td>L</td>
<td>10⁻⁲ × 10⁻¹</td>
<td>10⁻³</td>
<td>10⁻⁶ × 10⁻³</td>
</tr>
<tr>
<td>C</td>
<td>10⁻⁶ × 10⁻²</td>
<td>10⁻³ × 10⁻¹</td>
<td>10⁻⁶ × 10⁻³</td>
</tr>
</tbody>
</table>

- bit ratio of exponent and mantissa particularly in hybrid encoding of BGA = 3/10.

Experiment results are summarized in Tables 4 and 5. The terms NARROW, MEDIUM, and WIDE represent that the order differences of upper and lower bounds in physical parameters including R, L, and C are 2, 3, and 4, respectively, as shown in Table 6.

In linear space, as search bounds are expanded to medium and wide bounds, LM BEST COST values increases about ten times as large as those in BGA, and several hundred times in RCGA. On the contrary, the LM BEST COST values in logarithmic and hybrid space are constant or even decreased under the same condition. Moreover, as to the FINAL BEST COST values, as search bounds are expanded, average exponent number in linear space decreases from −5.53 to −7 (LOG) and −6.33 (HYBRID) in BGA, and from −3 to −6.33 (LOG) and −5.67 (HYBRID) for RCGA. In other words, as search space grows wider by factor of 10ⁿ where n is the dimension of a parameter vector, BGA and RCGA in linear search space tend to give poor performances, while in logarithmic and hybrid search spaces their performances are either unaffected or improved. Table 5 shows that as cost is minimized each parameter approaches to its true value.

Figure 3 and Fig. 4 are provided to illustrate behavior of the cost values during search processes at each search space and encoding/converting method. Figure 3 shows mean values of average cost values (for each restart) versus generation axis, which supplies a general inspection to GA practitioners. The performance of BGA, as shown in Figs. 3(a), 3(c), and 3(e), is raised in logarithmic and hybrid spaces, which can be confirmed by noting that their cost differences between narrow, medium, and wide ranges are reduced. The superiority of the proposed methods is more prominent for RCGA as shown in Figs. 3(b), 3(d), and 3(f). Unlike BGA, the hybrid converting of RCGA shown in Fig. 3(f) outperforms other methods in that search performance is unaffected by assigned searching ranges. Despite this superiority, overall average cost profiles represent that the proposed encoding methods for BGA is more reliable than those for RCGA.

Figure 4 shows mean values of best-so-far costs versus generation axis, which illustrate a similar minimization pattern in a smoothed fashion. Owing to the perturbations of average cost values in narrow range shown in Fig. 3, the mean best-so-far costs of BGA in logarithmic and hybrid space shown in Figs. 4(c) and 4(e) are almost equal and better than the costs of RCGA shown in Figs. 4(d) and 4(f). Overall comparison of BGA and RCGA ensures that the performance of BGA is slightly better than RCGA owing to the chromosomal structures and the adaptive scheme used for crossover and mutation in BGA. However, development of
improved crossover and mutation operators in RCGA would overcome this weakness.

It should be noted that the best identification result is attained at the MEDIUM bounds in logarithmic search space for both GAs. This may be not general but problem specific, since the benchmark results indicate that hybrid search
space is superior. This observation however implies that if logarithmic or hybrid solution space is adopted for GA, parameter bounds can be set a little larger than those in linear search space with conventional GAs.

This paper specifically concerns real positive parameters to be sought by GA. This constraint is due to the fact that a target of the proposed optimization methods is identification of an electrical system composed of real positive elements such as resistors, inductors, and capacitors. However, to expand the application field of the proposed methods search range should contain negative real values. Among several alternatives, adding a polarity bit to each parameter
is recommended for BGA referring to the microprocessor technique. That is, if the polarity bit is assigned 0 (1) for a string, the decoding processes refer to the negative (positive) upper and lower bounds for the corresponding parameter and multiply a decoded value by \(-1(1)\). This additional operation can be simplified for the case of symmetrical search range, e.g. \(-10^5 \sim 10^5\). This principle of division can be also applied to RCGA in a different manner. Considering that the proposed converting methods of RCGA can be understood as mapping, some modification of definition in (8)–(10) will allow a mapping to negative real values.

In case of exponential conversion, for example, upper and lower exponent bounds for negative values are specified and added to those of positive values. Assume that an optimization problem contains a parameter whose range is \(-10^5 \sim 10^4\), and one assigns \(-2 \sim 5\) for exponents of negative values and \(-3 \sim 4\) for those of positive values. Then a total exponent range is modified from \(-3 \sim 4\) (positive only) into \(-3 \sim 11(= 4 + 7)\). The added number 7 denotes a difference of exponent ranges of negative values. During RCGA, an exponent number of higher than 4 (upper limit of positive values) is regarded as corresponding to a negative value, which is transformed and decoded to a negative value. For instance, a value of 7 is finally decoded to \(-10^5\), if a negative exponent is computed from \(-2\) (lower limit) and added by the overflow number of 3 (= 7 – 4). A remaining problem of containing zero can be simply resolved by extending lower exponent limit as small as possible for approximation, e.g. \(10^{-10} \approx 0\).

As a summary, the proposed encoding/converting methods for BGA/RCGA are shown to be more effective than the conventional linear encoding/converting methods for the problems of ill-scaled parameters and of wide parameter bounds. The proposed methods will fall under a similar method with the linear encoding in narrow parameter bounds, which means that the proposed methods are to be considered as general.

5. Conclusion

This paper proposes logarithmic and hybrid encodings/converting methods for BGA/RCGA considering identification of electrical systems whose physical parameters are the real numbers of different order. A simple probabilistic analysis is provided to investigate what occurs in terms of random sampling as the search space is expanded. Function optimization and system identification results show that the use of proposed encoding/converting methods guarantee a robust optimization performance irrelevant to the size of entire search space.

In the case of searching for real numbers, RCGA is preferable for its simple coding, fast computation, and various mutation and crossover operators. Our future work will focus on improving crossover and mutation operators in RCGA, and applying to the identification of high-order real-world systems.

References


Appendix: Calculation of \(\delta_e\) and \(\delta_m\) in Hybrid Encoding of BGA

The resolution of an exponent and a mantissa, \(\delta_e\) and \(\delta_m\),
respectively, can be calculated by boundary conditions. Assume $p$ and $q$ bits in GA are assigned, respectively, to the exponent and the mantissa of a real number $x \in [x^L, x^U]$, and $e^U = \log_{10} x^U$ and $e^L = \log_{10} x^L$ are upper and lower bound exponents of $x$. Since we fixed $\delta_e$ and $\delta_m$ to be constant, general expression of transformed real numbers using $p$ bits for the exponent and $q$ bits for the mantissa is given by

$$x = (1 + j\delta_m)10^{e^L+i\delta_e},$$

$$0 \leq i \leq 2^p - 1, \ 0 \leq j \leq 2^q - 1.$$ (A·1)

In (A·1), the upper bound condition is satisfied if

$$(1 + 2^q\delta_m)10^{e^L+(2^p-1)\delta_e} = 10^{e^U}. \quad \text{(A·2)}$$

Note in (A·2) that $2^q$ is $(2^q - 1) + 1$, i.e., larger by 1 than the maximum integer which can be reached by $q$ bits. This setting is specifically devised to avoid the overlap that occurs when the exponent is increased from $i\delta_e$ to $(i+1)\delta_e$:

$$(1 + 2^q\delta_m)10^{e^L+i\delta_e} = 1 \cdot 10^{e^L+(i+1)\delta_e} = 10^0 \cdot 10^{e^L+i\delta_e}. \quad \text{(A·3)}$$

From (A·3), the following relation is obtained:

$$(1 + 2^q\delta_m) = 10^{\delta_e}. \quad \text{(A·4)}$$

Inserting this into (A·2) gives

$$10^{e^L+2^q\delta_e} = 10^{e^U}, \quad \text{(A·5)}$$

thus,

$$\delta_e = \frac{e^U - e^L}{2^p}. \quad \text{(A·6)}$$

Since $\delta_e$ is obtained, $\delta_m$ is easily calculated from (A·4):

$$\delta_m = \frac{10^{\delta_e} - 1}{2^q}. \quad \text{(A·7)}$$

---

**Jong-Wook Kim** was born in Youngpoong, Korea, in 1970. He received the B.S., M.S., and Ph.D. degrees from the Electrical and Computer Engineering Division at Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 1998, 2000, and 2004, respectively. Currently, he is a Researcher with the Electrical Steel Sheet Research Group in POSCO Technical Research Laboratories. His research interests include core design of transformers, diagnosis of electrical systems, intelligent control, optimization algorithm, and system identification.

**Sang Woo Kim** was born in Pyungtaek, Korea, in 1961. He received the B.S., M.S., and Ph.D. degrees from Department of Control and Instrumentation Engineering, Seoul National University, in 1983, 1985, and 1990, respectively. Currently, he is an Associate Professor in the Department of Electronics and Electrical Engineering at Pohang University of Science and Technology (POSTECH), Pohang, Korea. He joined POSTECH in 1992 as an Assistant Professor and was a Visiting Fellow in the Department of Systems Engineering, Australian National University, Canberra, Australia, in 1993. His current research interests are in optimal control, optimization algorithm, intelligent control, wireless communication, and process automation.