International Journal of Computational Intelligence Systems, Vol. 8, No. 1 (2015) 62-74

Selection of Encoding Cardinality for a Class of Fitness Functions to Obtain Order-1 Building Blocks

Hongqiang Mo [1] *, Zhong Li [2] , Lianfang Tian [1] , and Xiang Tian [3]

[1] *College of Automation Science and Engineering, South China University of Technology, Guangzhou, 510641, P.R. China.*

E-mail: hqiangmo, chlftian@scut.edu.cn.

[2] *Faculty of Mathematics and Computer Science, FernUniversitaet in Hagen, Hagen, 58084, Germany.*

E-mail: zhong.li@fernuni-hagen.de.

[3] *Department of Mathematics, College of Sciences, South China University of Technology, Guangzhou, 510641, P.R. China.*

E-mail: xtian@scut.edu.cn.

Received 8 March 2014

Accepted 10 June 2014

Abstract

Encoding plays a key role in determining the optimization efficiency of a genetic algorithm. In the optimization of a continuous function, binary encodings are normally used due to their low coding-alphabet cardinalities. Nevertheless, from the viewpoint of building-block supply, it is remarked that a binary encoding is not necessarily the best choice to express a fitness function which is linearly combined of sinusoidal functions with frequencies exponential to a positive integer m when m is not equal to 2. It is proved that, if the frequencies are exponential to m, an encoding of cardinality m can provide a better supply of order-1 building blocks than the encodings of other cardinalities. Taking the advantage of building-block supplies, a genetic algorithm with an encoding of cardinality m has higher chance to find fitter solutions. This assumption is verified via a number of randomly generated fitness functions, and encodings with different cardinalities are compared according to the optimization performance of corresponding genetic algorithms on these fitness functions. The simulation results support the assumption, and show in the statistical sense that the genetic algorithm with an encoding of cardinality m outperforms those of the other cardinalities when the frequencies of the sinusoidal functions are exponential to m.

Keywords: Building block, encoding, genetic algorithm, continuous optimization, schema processing.

*Corresponding author: College of Automation Science and Engineering, South China University of Technology, Guangzhou, 510641, P.R. China. Tel: 86-20-87110719, E-mail: hqiangmo@scut.edu.cn.

1. Introduction

As a crucial part of a genetic algorithm (GA), encoding determines the optimization performance of the algorithm. However, it is still controversial on what kinds of encodings should be used for an optimization problem. Usually, the feasible solutions of a continuous optimization problem are represented with a binary encoding. This practice can be traced back to the principle of minimal alphabets proposed decades ago, which suggests expressing a problem with the smallest alphabet, say, binary encodings, to make full use of implicit parallelism. But the principle has not been unanimous accepted [1,2,3]. Antonisse [1] stated that, for the same purpose of maximizing implicit parallelism, the opposite conclusion, i.e., a large alphabet should be used, could be drawn by looking at the function of don't care character from another viewpoint.Fogel and Ghozeil [3] presented theorems to establish that, under some general assumptions, no choice of cardinality of a representation offered any intrinsic advantage over another. Even if merely the encodings with the smallest alphabet were considered, agreements still could not be reached on whether to use a binary encoding or a Gray one [4,5,6]. The presence of these divergences seems to support the known fact that no representation should be superior for all classes of problems [3,5,7], and evaluation criteria usually depend on what a designer concerns.

When one represents the feasible solutions of a continuous optimization problem with strings, and views the optimal or near-optimal strings as the combinations of building blocks (BBs), the supply of BBs can serve as a reasonable evaluation criterion of encoding performance. The notion of building blocks is frequently used in the literature but rarely defined. In general, a building block can be described as a highly fit solution to a sub-problem that can be expressed as a schema [8,9]. A schema is a template that identifies a subset of strings with similarities at certain string positions, and a single string belongs to all the schemata in which any of its fixed positions appear. For example, the string 1011 is a member of schemata 10** (where the *'s stand for unspecified positions), 1**1, *0**, and so forth. The order of a schema refers to the number of its fixed positions, and if the fixed position of an order-1 schema is at the h-th position of the string counted from the rightmost, we call it an order-1 schema at locus h, where h is a positive integer no larger than string length. For example, 1***, **0* are order-1 schemata at the 4th and 2nd loci, respectively.

Since a schema is a template that identifies a subset of strings, it can be regarded as a particular region in the solution space, and the schemata containing many unspecified positions — the low-order schemata — will typically be sampled by a large fraction of all the strings in a population of a GA. And by manipulating a limited population of strings, a GA actually samples a vastly larger number of regions [8,9,10]. As stated by the schema theorem, successive generations of reproduction produce increasing numbers of trials that lie in the regions represented by highly fit, low-order schemata, i.e., low-order BBs [8,10]. The building block hypothesis further assumes that, when BBs recombine to form even more highly fit, higher-order schemata, a GA rapidly focuses its attention on the most promising parts of the solution space [8]. In this sense, a schema-processing GA performs well when its encoding achieves a sufficient supply of low-order BBs for a given fitness function [9,11,12]. And a practical question intimately associated with the design and applications of GAs regards, given a special class of fitness functions, what kinds of encoding s can provide relatively good supplies of low-order building blocks, and whether or not a binary encoding is always the best choice.

In the GA literature related to BBs, attentions have been paid on the design of operators to find and recombine low-order BBs [12,13,14], or on the features of fitness functions or genetic searching that hamper the low-order BBs

from recombining with each other to form overall good solutions, e.g., multi-modality [15], GA deception [16,17], sampling errors [18] and hyperplane inconsistency [19], etc. But the topics on building-block supply have received relatively few concerns. Before discussing how to make use of the low-order BBs to lead the search to the desired regions, it seems necessary to clarify, in the first place, whether or not a representation can really result in low-order BBs in the initial population. There were models to estimate the population size required to guarantee the presence of all raw BBs in a GA [11] or a genetic-programming [20] population, but the models only established necessary conditions for building-block supply, and did not tell about whether or not an encoding in deed provided low-order BBs, i.e., whether or not there were fitness differences among the low-order schemata.

In this paper, we make a tentative attempt to study this issue by comparing encodings of different cardinalities for a fitness function which is linearly combined of sinusoidal functions with frequencies exponential to a positive integer m. The choice of such fitness functions is mainly inspired by Fourier transformation. Linear expansion of a function into sine functions conforms to the common practices in functional analysis. By doing so, we hope that the future analysis of encoding design can get more supports from functional theory.

The representations discussed in this paper are limited to base-m encodings, whose cardinalities are positive integers m larger than 1. When a solution x, also called string or individual, is represented with the base-m encoding of string length l, $x = \sum_{h=1}^{l} x_h m^{h-l-1}$, where $x_h \in \{0, \cdots, m-1\}$. The commonly used binary representations are base-2 encodings. And for the base-3 encoding of string length 12, $x = \sum_{h=1}^{12} x_h 3^{h-13}$, and $x_h \in \{0,1,2\}$.

The analyses will be carried out in two steps: Firstly, we will discuss whether there are order-1 BBs when a base-m encoding is used for a sinusoidal function with frequency exponential to a positive integer m. Secondly, we will study the effect of linear transformation to building-block supply. In Ref. 21, it was pointed out that, for functions linearly combined of n_B basis functions, when the basis functions satisfied certain conditions, a base-m encoding could lead to n_B order-1 BBs with fixed position at different loci. We will apply the results herein, and find out what will happen when the basis functions are sinusoidal functions with frequencies exponential to m. The analysis results will then be tested with simulations to see if a GA with a base-m encoding outplays those of other encodings on the optimization of these fitness functions.

The rest of this paper is organized as follows: Section 2 introduces an index to the existence of order-1 BB. Section 3 explains why there will be order-1 BBs with fixed positions simultaneously at multiple loci when a base-m encoding is used for a fitness function which is linearly combined of sinusoidal functions with frequencies exponential to m, and why the encodings with other cardinalities can not achieve comparable building-block supply. Section 4 provides simulation results in which the optimization performance of encodings with different bases are compared on a number of fitness functions randomly generated. Finally, Section 5 summarizes the paper.

2. Preliminaries

Without loss of generality, suppose the fitness functions $G(x)$ discussed in this paper are defined on $[0,1)$, and satisfy $G(x) \geqslant 0$.

A schema is said to match an individual if they are identical at the fixed positions of the former. The fitness of a schema can be defined as the average fitness of all the individuals matched by the schema in a certain population or in the whole search space. Let's take the base-3 encoding of length 2 as an example: If the latter definition is employed, the fitness of the schema *1 is equal to $(G(01) + G(11) + G(21))/3$. If the former definition is used, giv-

en a population consisting of $01, 01, 22$, and 10, the fitness of *1 is equal to $(G(01)+G(01))/2$. With the former definition, schema fitness depends not only on encoding, but also on the formation of initial population, genetic operators, and selection strategy. The latter, usually used to determine the static fitness distributions of a schema [8,19], is especially suitable to study the sole effect of encoding on schema fitness. Therefore, here and throughout, the latter definition is adopted, and the fitness of the order-1 schema $*\cdots*x_h*\cdots*$ is

$$f_G(x_h) = \frac{\sum_{k=0}^{m^{l-h}-1}\sum_{o=0}^{m^{h-1}-1} G(km^{h-l}+x_h m^{h-l-1}+om^{-l})}{m^{l-1}}, \quad (1)$$

where the symbols f and x_h stand for average fitness and position value (allele value) at the h-th locus counted from the rightmost, respectively, and the subscript G indicates the name of the fitness function. The maximal and minimal fitness of the order-1 schemata at locus h are denoted as $\max_h(f_G(x_h))$ and $\min_h(f_G(x_h))$, respectively. Similar to the definition given in Ref. 8, order-1 BBs are defined as highly fit, order-1 schemata in this paper.

The average fitness of all the order-1 schemata at locus h is

$$f_G(*) = \frac{\sum_{x_h=0}^{m-1} f_G(x_h)}{m}. \quad (2)$$

If an order-1 schema at locus h is much fitter than the other order-1 schemata at the same locus, the ratio of the maximal fitness to the average fitness of all the order-1 schemata at this locus, $\max_h(f_G(x_h))/f_G(*)$, FRMax2Avg at locus h for short, will be significantly larger than 1. In other words, a large FRMax2Avg at a certain locus indicates the existence of an order-1 BB at this locus. Correspondingly, the ratio of the minimal fitness to the average fitness of all the order-1 schemata at locus h, i.e., $\min_h(f_G(x_h))/f_G(*)$, is abbreviated as FRMin2Avg at locus h.

Note that $\min_h(f_G(x_h)) \leqslant f_G(*) \leqslant \max_h(f_G(x_h)) \leqslant \sum_{x_h=0}^{m-1} f_G(x_h) = m f_G(*)$, therefore,

$$\frac{\min_h(f_G(x_h))}{f_G(*)} \leqslant 1 \leqslant \frac{\max_h(f_G(x_h))}{f_G(*)} \leqslant m. \quad (3)$$

3. Supply of Order-1 Building Blocks for $\sum_{i=1}^{n_B}(a_i \sin(2\pi m^{i-1}x+\varphi_i))+c$

Specially, the fitness function considered herein is $\sum_{i=1}^{n_B}(a_i \sin(2\pi m^{i-1}x+\varphi_i))+c$, where m is an integer larger than 1, n_B is a positive integer smaller than string length l, a_i and φ_i are arbitrary real constants, and c is a real number large enough to ensure nonnegative fitness. As has been mentioned in Introduction, the discussions are carried out first on $\sin(2\pi m^{i-1}x+\varphi_i)+1$, and then on $\sum_{i=1}^{n_B}(a_i \sin(2\pi m^{i-1}x+\varphi_i))+c$.

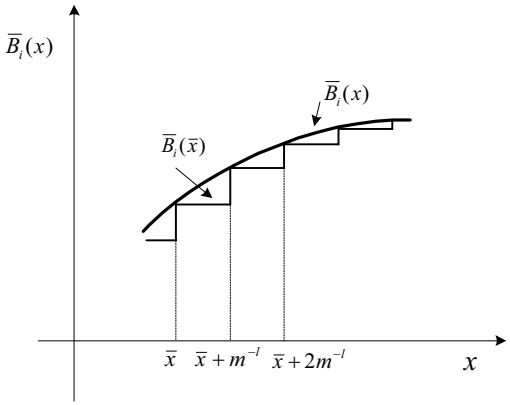

Fig. 1. The illustration of an approximation of $\int_{\bar{x}}^{\bar{x}+m^{-l}} \overline{B}_i(x)dx$ with $\overline{B}_i(\bar{x})$. When $m^{-l} \to 0$, $\int_{\bar{x}}^{\bar{x}+m^{-l}} \overline{B}_i(x)dx \to m^{-l}\overline{B}_i(\bar{x})$.

3.1. *Supply of order-1 building block for* $\sin(2\pi m^{i-1}x+\varphi_i)+1$

For convenience, consider a more general form of basis function, $\overline{B}_i(x) = \sin(2\pi p_i x+\varphi_i)+1$, where p_i is a positive integer. Obviously, there are p_i complete cycles of $\overline{B}_i(x)$ within $[0,\ 1)$. To simplify expressions, let's denote m^{h-l-1} as Δ. As illustrated in Fig.1, since $\overline{B}_i(x)$ is integrable, by using the sum-to-product trigono-

metric formulas and Taylor's expansion of sinusoidal function, we have (4).

$$\int_{\overline{x}}^{\overline{x}+m^{-l}} \overline{B_i}(x)dx = \frac{\overline{B_i}(\overline{x}) + \pi p_i m^{-l}\cos(2\pi p_i \overline{x}+\varphi_i) + O(p_i m^{-l})}{m^l}, \quad (4)$$

where $\overline{x} = (mk+x_h)\Delta + om^{-l}$ and $O(p_i m^{-l})$ is the sum of high-order terms of $p_i m^{-l}$. When $m^l \gg p_i$, the terms $\pi p_i m^{-l}\cos(2\pi p_i \overline{x}+\varphi_i) + O(p_i m^{-l})$ can be neglected, and further denote $(mk+x_h)\Delta$ as $\overline{\Delta}$, we have (5).

$$\int_{\overline{\Delta}}^{\overline{\Delta}+\Delta} \overline{B_i}(x)dx = \int_{\overline{\Delta}}^{\overline{\Delta}+m^{-l}} \overline{B_i}(x)dx + \cdots + \int_{\overline{\Delta}+\Delta-m^{-l}}^{\overline{\Delta}+\Delta} \overline{B_i}(x)dx = \sum_{o=0}^{m^{h-1}-1} (m^{-l}\overline{B_i}((km+x_h)\Delta + om^{-l})). \quad (5)$$

Applying the sum-to-product trigonometric formulas to $\int_{\overline{\Delta}}^{\overline{\Delta}+\Delta} \overline{B_i}(x)dx$, one has

$$\int_{\overline{\Delta}}^{\overline{\Delta}+\Delta} \overline{B_i}(x)dx = \Delta + \frac{\sin(\pi p_i \Delta)\sin(2\pi p_i(mk+x_h)\Delta + \pi p_i \Delta + \varphi_i)}{\pi p_i}. \quad (6)$$

According to (1), (5), and (6),

$$f_{\overline{B_i}}(x_h) = m\sum_{k=0}^{m^{l-h}-1} (\Delta + \frac{\sin(\pi p_i \Delta)\sin(2\pi p_i(mk+x_h)\Delta + \pi p_i \Delta + \varphi_i)}{\pi p_i}). \quad (7)$$

Further denote $m\sin(\pi p_i \Delta)/(\pi p_i)$ as A_i and $\pi p_i \Delta(2x_h+1)+\varphi_i$ as $\overline{\varphi}_i$, respectively, and we have

$$f_{\overline{B_i}}(x_h) = 1 + \sum_{k=0}^{m^{l-h}-1} A_i \sin(2kp_i\pi m^{h-l} + \overline{\varphi}_i). \quad (8)$$

The value of $f_{\overline{B_i}}(x_h)$ depends on h, p_i, and x_h. All possible relationships are listed as follows.

(i) If $p_i = m^{l-h}$,

$$f_{\overline{B_i}}(x_h) = 1 + m^{l-h}A_i\sin(\overline{\varphi}_i) = 1 + \frac{m}{\pi}\sin\frac{\pi}{m}\sin(\frac{\pi}{m}(2x_h+1)+\varphi_i). \quad (9)$$

(ii) If $p_i \neq m^{l-h}$,

(a) When $h = l$,

$$f_{\overline{B_i}}(x_h) = 1 + A_i\sin(\overline{\varphi}_i) = 1 + \frac{m}{\pi p_i}\sin\frac{\pi p_i}{m}\sin(\frac{\pi p_i}{m}(2x_l+1)+\varphi_i). \quad (10)$$

(b) When $h \neq l$, the amplitudes of the m^{l-h} sine functions, $A_i\sin(2kp_i\pi m^{h-l} + \overline{\varphi}_i)$, $k \in \{0,\cdots,m^{l-h}-1\}$, are the same, and the phase difference between each adjacent pair of the functions is $2p_i\pi m^{h-l}$. As a result, their sum is equal to 0, and

$$f_{\overline{B_i}}(x_h) = 1. \quad (11)$$

Therefore, if $\overline{B_i}(x) = \sin(2\pi m^{i-1}x + \varphi_i) + 1$, and if a base-$m$ encoding is used for it, according to (9) and (11), respectively, the fitness of the order-1 schema $*\cdots* x_{l-i+1} *\cdots*$ at locus $l-i+1$ is $f_{\overline{B_i}}(x_{l-i+1}) = 1 + m\sin(\pi/m)\sin((2\pi x_{l-i+1}+\pi)/m + \varphi_i)/\pi$, and those at the other loci are all equal to 1. When x_{l-i+1} increases from 0 to $m-1$, we can get the $\max_{l-i+1}(f_{\overline{B_i}}(x_{l-i+1}))$ and $\min_{l-i+1}(f_{\overline{B_i}}(x_{l-i+1}))$, respectively. Obviously, in most cases, $\max_{l-i+1}(f_{\overline{B_i}}(x_{l-i+1}))$ and $\min_{l-i+1}(f_{\overline{B_i}}(x_{l-i+1}))$ are significantly larger and small than $\sum_{x_{l-i+1}=0}^{m-1} f_{\overline{B_i}}(x_{l-i+1})/m$, respectively. As a result, in most cases, the FRMax2Avg and FRMin2Avg at locus $l-i+1$ are significantly larger and smaller, respectively, than 1, with only one exceptional case — when $m=2$ and $\varphi_i = 0.5\pi$, all the FRMax2Avgs and FRMin2Avgs are equal to 1. While at the other loci, FRMax2Avg= 1 =FRMin2Avg. Thus, there will be an order-1 BB at and only at locus $l-i+1$ when a base-m encoding is applied to $\sin(2\pi m^{i-1}x+\varphi_i)+1$.

And if the basis function $\overline{B_i}(x)$ is changed into $\sin(2\pi\overline{m}^{i-1}x+\varphi_i)+1$, where $\overline{m} \neq m$, and

a base-m encoding is used for it, according to (11), $f_{\overline{B_i}}(x_{l-j}) \equiv 1$ for all the $j \in \{1,\cdots,l-1\}$. The only exception occurs at locus l; similar to the discussions in the last paragraph, according to (10), one gets FRMax2Avg$> 1 >$FRMin2Avg at this locus. Therefore, no matter what the value i of $\overline{B_i}(x) = \sin(2\pi\overline{m}^{i-1}x + \varphi_i) + 1$ is, there is one and only one order-1 BB at locus l under this encoding.

Table 1 to Table 4 give several examples, in which the base-2 encoding of string length 19 and the base-3 encoding of string length 12 are used to express the sinusoidal functions, $\sin(2\pi 2^{i-1}x) + 1$, and $\sin(2\pi 3^{i-1}x) + 1,\ i \in \{1,\cdots,4\}$, respectively. As shown in Table 1 for $\sin(2\pi 2^{i-1}x) + 1$, under the base-2 encoding, the FRMax2Avgs on the diagonal — at locus $20-i$ for $\sin(2\pi 2^{i-1}x) + 1$ — are much larger than the others, which indicates a correlation between the locus position with large FRMax2Avg and the function frequency. Thus, there is an order-1 BB at locus $20-i$ when the base-2 encoding is used for $\sin(2\pi 2^{i-1}x) + 1$. However, when the feasible solutions of $\sin(2\pi 2^{i-1}x) + 1$ are encoded with the base-3 encoding, as shown in Table 2, most of the FRMax2Avgs and FRMin2Avgs are equal to 1.0, except those at locus 12. As a result, when the base-3 encoding is employed, there is one and only one order-1 BB at the highest locus regardless of the value of i. The similar thing happens when the encodings are used for $\sin(2\pi 3^{i-1}x) + 1$, as shown in Table3 and Table 4, in which the base-3 encoding can obtain order-1 BB at different locus when i changes, and the base-2 encoding can get an order-1 BB only at the highest locus no matter what the value of i is.

3.2. *Supply of order-1 building blocks for* $\sum_{i=1}^{n_B}(a_i \sin(2\pi m^{i-1}x + \varphi_i)) + c$

Suppose the FRMax2Avgs for a series of basis functions are significantly larger than 1 at different loci, one may surmise that large FRMax2Avgs can be obtained simultaneously at multiple loci for a fitness function linearly combined of them, since the averaging of fitness is a kind of linear transform. Specifically, consider the fitness functions of the form, $G(x) = \sum_{i=1}^{n_B} a_i B_i(x) + c$, where a_i and c are real constants, and n_B is a positive integer smaller than l. Note that all the basis functions and their combinations should meet the abovementioned demands, i.e., being nonnegative and defined on $[0,1)$. Suppose that the FRMax2Avg for $B_i(x)$ at locus h_i and that for $B_j(x)$ at locus h_j are much larger than 1 under a base-m encoding, where $i,j \in \{1,\cdots,n_B\}$, $h_i,h_j \in \{1,\cdots,l\}$, and $i \neq j$. One may care about whether or not the encoding can get large FRMax2Avgs simultaneously at both loci under transformation $a_iB_i + a_jB_j + c$. Theorem 1 gives a hint for the demands on the basis functions and the forms of their combinations [21].

Theorem 1. *If* $\max_{h_j}(f_{B_i}(x_{h_j})) = \min_{h_j}(f_{B_i}(x_{h_j}))$ *and* $\max_{h_i}(f_{B_j}(x_{h_i})) = \min_{h_i}(f_{B_j}(x_{h_i}))$, *where* $i,j \in \{1,\cdots,n_B\}$, $h_i,h_j \in \{1,\cdots,l\}$, *and* $i \neq j$, *then*

$$\frac{\max_{h_i}(f_T(x_{h_i}))}{f_T(*)} = \frac{\max\left(a_i \max_{h_i}(f_{B_i}(x_{h_i})), a_i \min_{h_i}(f_{B_i}(x_{h_i}))\right) + a_j f_{B_j}(*) + c}{a_i f_{B_i}(*) + a_j f_{B_j}(*) + c}, \tag{12}$$

$$\frac{\max_{h_j}\left(f_T(x_{h_j})\right)}{f_T(*)} = \frac{a_i f_{B_i}(*) + \max\left(a_j \max_{h_j}\left(f_{B_j}(x_{h_j})\right), a_j \min_{h_j}\left(f_{B_j}(x_{h_j})\right)\right) + c}{a_i f_{B_i}(*) + a_j f_{B_j}(*) + c}. \tag{13}$$

where T stands for the transformation $a_iB_i + a_jB_j + c$.

Table 1. The FRMax2Avgs and FRMin2Avgs for $\sin(2\pi 2^{i-1}x)+1, i \in \{1,\cdots,4\}$ and the FRMax2Avgs for some of their combinations under the base-2 encoding of string length 19.

Fitness functions		Fixed positions				
		19^{th}	18^{th}	17^{th}	16^{th}	other loci
$\sin 2\pi x+1$	FRMax2Avgs	**1.64**	1.00	1.00	1.00	1.00
	FRMin2Avgs	**0.36**	1.00	1.00	1.00	1.00
$\sin 4\pi x+1$	FRMax2Avgs	1.00	**1.64**	1.00	1.00	1.00
	FRMin2Avgs	1.00	**0.36**	1.00	1.00	1.00
$\sin 8\pi x+1$	FRMax2Avgs	1.00	1.00	**1.64**	1.00	1.00
	FRMin2Avgs	1.00	1.00	**0.36**	1.00	1.00
$\sin 16\pi x+1$	FRMax2Avgs	1.00	1.00	1.00	**1.64**	1.00
	FRMin2Avgs	1.00	1.00	1.00	**0.36**	1.00
$\sum_{i=1}^{4}(\sin(2\pi 2^{i-1}x)+1)$	FRMax2Avgs	**1.16**	**1.16**	**1.16**	**1.16**	1.00
$\sum_{i=1}^{4}((-1)^{i-1}\sin(2\pi 2^{i-1}x)+1)$	FRMax2Avgs	**1.16**	**1.16**	**1.16**	**1.16**	1.00
$0.31\sin(2\pi x+0.71\pi)+0.72\sin(4\pi x-0.23\pi)$ $-0.45\sin(8\pi x+0.64\pi)+0.62\sin(16\pi x-0.95\pi)+2.1$	FRMax2Avgs	**1.06**	**1.16**	**1.06**	**1.19**	1.00

Table 2. The FRMax2Avgs and FRMin2Avgs for $\sin(2\pi 2^{i-1}x)+1, i \in \{1,\cdots,4\}$ and the FRMax2Avgs for some of their combinations under the base-3 encoding of string length 12.

Fitness functions		Fixed positions				
		12^{th}	11^{th}	10^{th}	9^{th}	other loci
$\sin 2\pi x+1$	FRMax2Avgs	**1.72**	1.00	1.00	1.00	1.00
	FRMin2Avgs	**0.28**	1.00	1.00	1.00	1.00
$\sin 4\pi x+1$	FRMax2Avgs	**1.36**	1.00	1.00	1.00	1.00
	FRMin2Avgs	**0.64**	1.00	1.00	1.00	1.00
$\sin 8\pi x+1$	FRMax2Avgs	**1.18**	1.00	1.00	1.00	1.00
	FRMin2Avgs	**0.82**	1.00	1.00	1.00	1.00
$\sin 16\pi x+1$	FRMax2Avgs	**1.09**	1.00	1.00	1.00	1.00
	FRMin2Avgs	**0.91**	1.00	1.00	1.00	1.00
$\sum_{i=1}^{4}(\sin(2\pi 2^{i-1}x)+1)$	FRMax2Avgs	**1.34**	1.00	1.00	1.00	1.00
$\sum_{i=1}^{4}((-1)^{i-1}\sin(2\pi 2^{i-1}x)+1)$	FRMax2Avgs	**1.11**	1.00	1.00	1.00	1.00
$0.31\sin(2\pi x+0.71\pi)+0.72\sin(4\pi x-0.23\pi)$ $-0.45\sin(8\pi x+0.64\pi)+0.62\sin(16\pi x-0.95\pi)+2.1$	FRMax2Avgs	**1.10**	1.00	1.00	1.00	1.00

Table 3. The FRMax2Avgs and FRMin2Avgs for $\sin(2\pi 3^{i-1}x)+1, i \in \{1,\cdots,4\}$ and the FRMax2Avgs for some of their combinations under the base-3 encoding of string length 12.

Fitness functions		Fixed positions				
		12^{th}	11^{th}	10^{th}	9^{th}	other loci
$\sin 2\pi x+1$	FRMax2Avgs	**1.72**	1.00	1.00	1.00	1.00
	FRMin2Avgs	**0.28**	1.00	1.00	1.00	1.00
$\sin 6\pi x+1$	FRMax2Avgs	1.00	**1.72**	1.00	1.00	1.00
	FRMin2Avgs	1.00	**0.28**	1.00	1.00	1.00
$\sin 18\pi x+1$	FRMax2Avgs	1.00	1.00	**1.72**	1.00	1.00
	FRMin2Avgs	1.00	1.00	**0.28**	1.00	1.00
$\sin 54\pi x+1$	FRMax2Avgs	1.00	1.00	1.00	**1.72**	1.00
	FRMin2Avgs	1.00	1.00	1.00	**0.28**	1.00
$\sum_{i=1}^{4}(\sin(2\pi 3^{i-1}x)+1)$	FRMax2Avgs	**1.18**	**1.18**	**1.18**	**1.18**	1.00
$\sum_{i=1}^{4}((-1)^{i-1}\sin(2\pi 3^{i-1}x)+1)$	FRMax2Avgs	**1.18**	**1.18**	**1.18**	**1.18**	1.00
$-0.25\sin(2\pi x-0.57\pi)-0.36\sin(6\pi x-0.43\pi)$ $-0.77\sin(18\pi x+0.95\pi)+0.69\sin(54\pi x-0.15\pi)+2.07$	FRMax2Avgs	**1.07**	**1.10**	**1.24**	**1.15**	1.00

Table 4. The FRMax2Avgs and FRMin2Avgs for $\sin(2\pi 3^{i-1}x)+1, i \in \{1,\cdots,4\}$ and the FRMax2Avgs for some of their combinations under the base-2 encoding of string length 19.

Fitness functions		Fixed positions			
		19^{th}	18^{th}	17^{th}	other loci
$\sin 2\pi x+1$	FRMax2Avgs	**1.64**	1.00	1.00	1.00
	FRMin2Avgs	**0.36**	1.00	1.00	1.00
$\sin 6\pi x+1$	FRMax2Avgs	**1.21**	1.00	1.00	1.00
	FRMin2Avgs	**0.79**	1.00	1.00	1.00
$\sin 18\pi x+1$	FRMax2Avgs	**1.07**	1.00	1.00	1.00
	FRMin2Avgs	**0.93**	1.00	1.00	1.00
$\sin 54\pi x+1$	FRMax2Avgs	**1.02**	1.00	1.00	1.00
	FRMin2Avgs	**0.98**	1.00	1.00	1.00
$\sum_{i=1}^{4}(\sin(2\pi 3^{i-1}x)+1)$	FRMax2Avgs	**1.24**	1.00	1.00	1.00
$\sum_{i=1}^{4}((-1)^{i-1}\sin(2\pi 3^{i-1}x)+1)$	FRMax2Avgs	**1.12**	1.00	1.00	1.00
$-0.25\sin(2\pi x-0.57\pi)-0.36\sin(6\pi x-0.43\pi)$ $-0.77\sin(18\pi x+0.95\pi)+0.69\sin(54\pi x-0.15\pi)+2.07$	FRMax2Avgs	**1.04**	1.00	1.00	1.00

Proof. By definition (1) and (2), respectively,

$$f_T(x_{h_i}) = a_i f_{B_i}(x_{h_i}) + a_j f_{B_j}(x_{h_i}) + c, \qquad (14)$$

and

$$f_T(*) = a_i f_{B_i}(*) + a_j f_{B_j}(*) + c. \qquad (15)$$

With the known condition, $\max_{h_i}\left(f_{B_j}(x_{h_i})\right) = \min_{h_i}\left(f_{B_j}(x_{h_i})\right)$, we get

$$\max_{h_i}\left(f_{B_j}(x_{h_i})\right) = \min_{h_i}\left(f_{B_j}(x_{h_i})\right) = f_{B_j}(*). \qquad (16)$$

Depending on the sign of a_i,

$$\max_{h_i}(a_i f_{B_i}(x_{h_i})) = \max(a_i \max_{h_i}(f_{B_i}(x_{h_i})), a_i \min_{h_i}(f_{B_i}(x_{h_i}))). \qquad (17)$$

For the same reason,

$$\max_{h_i}(a_j f_{B_j}(x_{h_i})) = \max(a_j \max_{h_i}(f_{B_j}(x_{h_i})), a_j \min_{h_i}(f_{B_j}(x_{h_i}))). \qquad (18)$$

Substituting (16) into (18) results in

$$\max_{h_i}(a_j f_{B_j}(x_{h_i})) = a_j f_{B_j}(*). \qquad (19)$$

From (14), (15), (17) and (19), it is straightforward to obtain (12). And similarly, we have (13). □

Remark: Besides the conditions given in Theorem 1, if we further demand $a_i f_{B_i}(*) \approx a_j f_{B_j}(*) \gg c$, then from (12), we have (20). And if $\max_{h_i}(f_{B_i}(x_{h_i})) \gg f_{B_i}(*) \gg \min_{h_i}(f_{B_i}(x_{h_i}))$ and $\max_{h_j}\left(f_{B_j}(x_{h_j})\right) \gg f_{B_j}(*) \gg \min_{h_j}\left(f_{B_j}(x_{h_j})\right)$, FRMax2Avgs at locus h_i and locus h_j for $T = a_i B_i + a_j B_j + c$ will be much larger than 1. The requirement $a_i f_{B_i}(*) \approx a_j f_{B_j}(*) \gg c$ may seem restrictive in practices, but actually, the FRMax2Avgs are still acceptably large provided c is small enough.

$$\frac{\max_{h_i}(f_T(x_{h_i}))}{f_T(*)} = \frac{\max(a_i \max_{h_i}(f_{B_i}(x_{h_i})), a_i \min_{h_i}(f_{B_i}(x_{h_i})))}{2a_i f_{B_i}(*)} + \frac{1}{2}. \qquad (20)$$

From Theorem 1 and the Remark, we can figure out requirements for a certain basis function $B_i(x)$, where $i \in \{1,\cdots,n_B\}$, as follows: the maximal and minimal fitness of the order-1 schemata at locus h_i should be remarkably greater and smaller, respectively, than the average fitness of all the order-1 schemata at this locus, whereas the maximal, minimal, and average fitness of the order-1 schemata at the other loci should be almost identical, as shown in (21) and (22).

$$\max_{h_i}\left(f_{B_i}(x_{h_i})\right) \gg f_{B_i(*)} \gg \min_{h_i}\left(f_{B_i}(x_{h_i})\right) \qquad (21)$$

$$\forall j \neq i,\ \max_{h_j}\left(f_{B_i}(x_{h_j})\right) = f_{B_i}(*) = \min_{h_j}\left(f_{B_i}(x_{h_j})\right) \tag{22}$$

According to the results in Section 3.1, when a base-m encoding is applied to $\sin(2\pi m^{i-1}x+\varphi_i)+1$, the FRMax2Avg and FRMin2Avg at locus $l-i+1$ will be significantly larger and smaller, respectively, than 1, and the FRMax2Avgs and FRMin2Avgs at the other loci are all equal to 1. That is to say, $\sin(2\pi m^{i-1}x+\varphi_i)+1$ satisfies (21) and (22) under a base-m encoding. As a result, there will be n_B order-1 BBs at the leftmost n_B loci when a base-m encoding with long enough string length is used for $\sum_{i=1}^{n_B}(a_i\sin(2\pi m^{i-1}x+\varphi_i))+c$. An encoding of another cardinality does not have such advantage; when it is used for $\sum_{i=1}^{n_B}(a_i\sin(2\pi m^{i-1}x+\varphi_i))+c$, an order-1 BB can usually be obtained only at the highest locus. In a word, a base-m encoding outplays an encoding of another cardinality on the supply of order-1 BBs when applied to $\sum_{i=1}^{n_B}(a_i\sin(2\pi m^{i-1}x+\varphi_i))+c$.

The last lines of the tables, from Table 1 to Table 4, give several examples. As shown in Table 1, when the base-2 encoding of string length 19 is applied to $\sum_{i=1}^{4}(\sin(2\pi 2^{i-1}x)+1)$, there are order-1 BBs at the 4 highest loci. The same thing happens when the base-3 encoding of string length 12 is employed for $\sum_{i=1}^{4}(\sin(2\pi 3^{i-1}x)+1)$, as shown in Table 3. However, when the base-2 encoding is used for $\sum_{i=1}^{4}(\sin(2\pi 3^{i-1}x)+1)$ or the base-3 one for $\sum_{i=1}^{4}(\sin(2\pi 2^{i-1}x)+1)$, as shown in Table 2 and Table 4, respectively, only one BB at the highest locus can be obtained. The situations of the other combinations of the sinusoidal basis functions given in the tables are similar to those of $\sum_{i=1}^{4}(\sin(2\pi 2^{i-1}x)+1)$ and $\sum_{i=1}^{4}(\sin(2\pi 3^{i-1}x)+1)$, in which the base-2 encoding and the base-3 one achieve order-1 BBs at 4 loci for the fitness functions with frequencies of the basis functions exponential to 2 and 3, respectively.

4. Simulation Results

Taking the advantage of building-block supplies, a GA with an encoding of cardinality m is likely to outdo those of other cardinalities in the optimization of $\sum_{i=1}^{n_B}(a_i\sin(2\pi m^{i-1}x+\varphi_i))+c$, and has higher chance to find fitter solutions. Therefore, the encodings are further evaluated according to the optimization performance of corresponding GAs.

```
do{
        if ( j < ( PopulationSize/3 ) )
        {       mate1= RouletteWheelSelection();
                mate2= RouletteWheelSelection();
        }
        else if ( j < ( 2* PopulationSize /3 ) )
        {       mate1= RouletteWheelSelection ();
                mate2= RandomSelection();
        }
        else
        {       mate1= RandomSelection();
                mate2= RandomSelection();
        }
} while (j < PopulationSize );
ElitismSelection();

// PopulationSize is the size of population; mate1 and mate2 are the
        // individuals selected from the parent population.
//RouletteWheelSelection() implements roulette-wheel selection.
//RandomSelection() randomly selects an individual from the
        // parent population with a uniform rate.
// ElitismSelection() replaces the worst individual of the offspring
        //population with the fittest one of the parent population.
```

Fig. 2. The pseudo codes of the selection strategies.

4.1. Algorithm descriptions

The introduction of a properly designed hybrid operator can often significantly enhance the performance of a GA [12,13,22,23], and therefore can facilitate encoding evaluation. However, there are too many hybrid GAs, and it is almost impossible to compare encodings based on all of them one by one. In this paper, we limit the simulations within the frame of pure GAs, and only employ general selection strategies, single-point crossover, and bitwise mutation. The selection implemented herein is a compromise between diversity maintenance and the

realization of "survival of the fittest", whose pseudo codes are shown in Fig. 2. In the single-point crossover, a crossover point is selected, and string from the rightmost of chromosome to the crossover point is copied from one parent , i.e., mate1 in Fig.2, and the rest is copied from the second parent , i.e., mate2 in Fig.2. For a base-m encoding, the bitwise mutation is to replace the original allele value with another integer randomly selected from $\{1,\cdots,m-1\}$. For functions $\sum_{i=1}^{n_B}(a_i\sin(2\pi m^{i-1}x+\varphi_i))+c$, a base-$m$ encoded GA can result in order-1 BBs at n_B loci, so that the GA can make use of this advantage to search for fit allele values at multiple loci in parallel. Therefore, a high mutation rate is favorable. In the simulation, a mutation rate up to 0.3 is used, and its results are much better than those of a mutation rate much smaller than 0.3(Pls. see Fig. 6).

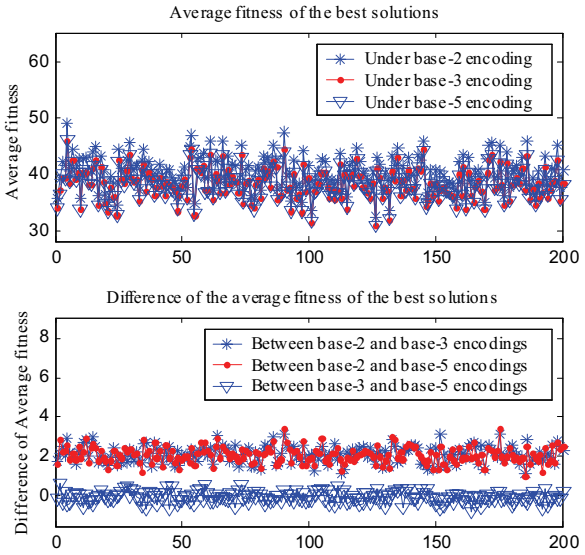

Fig. 3. The average fitness of the best solutions found in the 20 rounds of calculation for the fitness functions $\sum_{i=1}^{51}(a_i\sin(2\pi 2^{i-1}x+\varphi_i)+|a_i|)$ under the three encodings. The GA with the base-2 encoding significantly outperform the other GAs.

4.2. *Simulation settings and results*

The settings and results of the simulation are given as follows.

- **Fitness functions:** Totally 600 samples of fitness functions, $\sum_{i=1}^{n_B}(a_i\sin(2\pi m^{i-1}x+\varphi_i)+|a_i|)$, are randomly generated, with each setting of $(m=2,\ n_B=51)$, $(m=3,\ n_B=32)$, and $(m=5,\ n_B=22)$ applied to 200 samples. For these functions, $c=\sum_{i=1}^{n_B}|a_i|$ so as to ensure nonnegative fitness.
- **Encoding schemes:** The base-2 encoding of string length 62, the base-3 encoding of string length 39, and the base-5 encoding of string length 27 are used.
- **Parameter settings:** The population size, crossover rate, and mutation rate are 300, 0.8 and 0.3, respectively. For each pair of fitness function and encoding, 20 repetitive rounds of calculation are performed, with each round running for 200 generations.
- **Performance index:** The search spaces of the functions are too large for an exhaustive algorithm to find their global optima in limited time. In this regard, the encodings are compared according to the fittest solutions that the corresponding GAs find, and the fitter the best solutions a GA finds, the better the corresponding encoding performs. To each of the fitness functions, the three encodings are applied in turn, and in view of the probabilistic development of the solutions, the fitness values of the best solutions found in the 20 repetitive rounds are averaged, and the averages are compared.
- **Simulation Results:** The average fitness of the best solutions and the differences of the average fitness between each pair of the GAs with the three encodings are shown in Fig. 3, Fig. 4 and Fig. 5, respectively.

As shown in Fig. 3, for all of the 200 samples in which the frequencies of the basis functions are integral powers of 2, the average fitness of the best solutions found by the base-2 encoded GA are point-to-point larger than those of the best solutions found by the other GAs. Similar situations occur when the frequencies are integral powers of 3 and 5, respectively. As shown in Fig. 4 and Fig. 5, for each of the situations,

the base-3 encoded GA and the base-5 encoded one, respectively, play the best. The simulation results indirectly confirm the analyses in Section 3.

Performance of GAs with different mutation rates is also compared, with results given in Fig. 6. As shown in Fig. 6, the performance of the GAs with the higher mutation rates, 0.1 and 0.3, is significantly better than that of the GAs with the low mutation rate of 0.01.

5. Conclusion

In this paper, it was suggested to encode the feasible solutions of $\sum_{i=1}^{n_B}(a_i \sin(2\pi m^{i-1}x+\varphi_i))+c$ with a base-m encoding so as to obtain a better supply of order-1 BBs. Both the analysis and simulation results showed that, for such fitness functions, a base-m encoding outperformed the encodings with the other cardinalities on building-block supply and optimization performance.

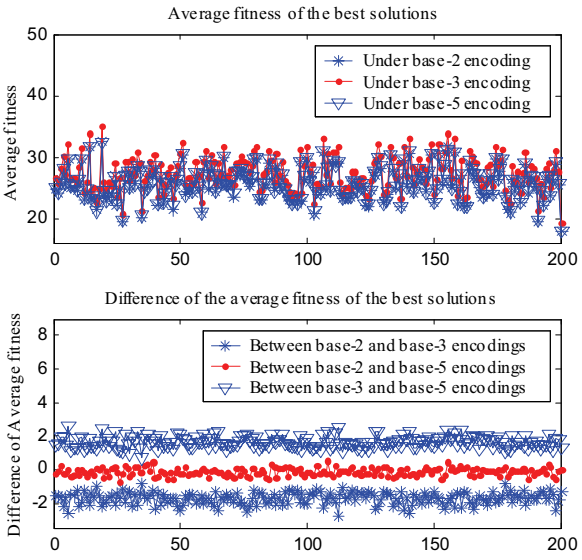

Fig. 4. The average fitness of the best solutions found in the 20 rounds of calculation for the fitness functions $\sum_{i=1}^{32}(a_i \sin(2\pi 3^{i-1}x+\varphi_i)+|a_i|)$ under the three encodings.The GA with the base-3 encoding significantly outperform the other GAs.

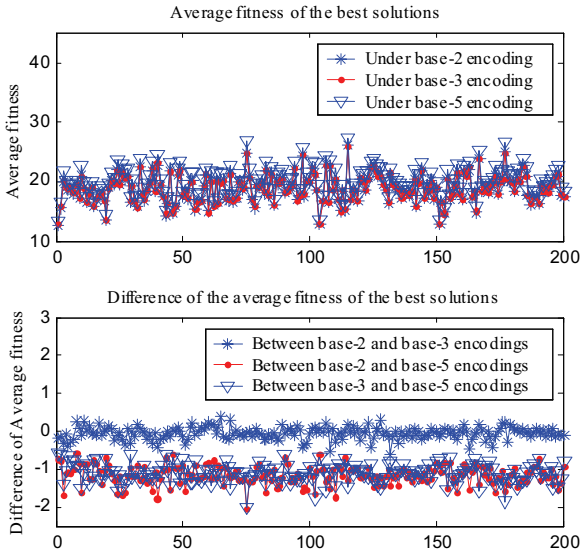

Fig. 5. The average fitness of the best solutions found in the 20 rounds of calculation for the fitness functions $\sum_{i=1}^{22}(a_i \sin(2\pi 5^{i-1}x+\varphi_i)+|a_i|)$ under the three encodings. The GA with the base-5 encoding significantly outperform the other GAs.

The main point of this paper was about the selection of encoding cardinality, and not on genetic operators. The simulation has been implemented with pure GAs. Yet, it does not hamper one from establishing more skillful memetics algorithms [22,23] to facilitate encoding evaluation, since a skill applies to an encoding of a certain cardinality can usually be used for those of other cardinalities.

It should be noted that the idea that GAs search by schema sampling has received many different criticisms, and to evaluate an encoding according to building-block supply is not always reliable. That we based the analyses in this paper upon schema theory stemmed from the following considerations: There is not a perfect explanation for the mechanisms of genetic search yet; the schema theorem itself is correct [24], and the improved versions of the theorem [25,26,27] have helped to describe genetic behaviors more exactly. Under such circumstances, it seems advisable to seek supports from schema theory, which can guide the design of an algo-

rithm, even if it is imperfect. And we will seek for better criteria and their rationales in future study.

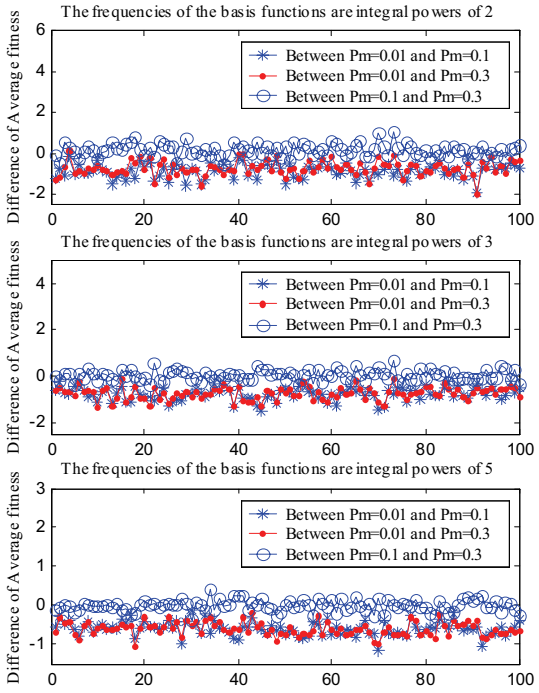

Fig. 6. The differences of the average fitness of the best solutions found in the 20 rounds of calculation between each pair of the GAs with different mutation rates P_m for the fitness functions $\sum_{i=1}^{n_B}(a_i \sin(2\pi m^{i-1}x+\varphi_i)+|a_i|)$, where the settings are $(m=2,\ n_B=51)$ under the base-2 encoding, $(m=3,\ n_B=32)$ under the base-3 encoding, and $(m=5,\ n_B=22)$ under the base-5 encoding, respectively. For each setting of $(m,\ n_B)$, there is no difference between the performance with $Pm=0.1$ and that with $Pm=0.3$ in the statistical sense, but they both significantly better than that with $Pm=0.01$.

Acknowledgments

This work was supported by "National Natural Science Foundation of China, 61105062", "Fundamental Research Funds for the Central Universities, SCUT, 2012ZZ0106", and in part by "National Natural Science Foundation of China, 61273249", "Natural Science Foundation of Guangdong Province, China, S2012010009886", and "the Key Laboratory of Autonomous Systems and Network Control, Ministry of Education".

References

1. J. Antonisse, "A new interpretation of schema notation that overturns the binary encoding constraint,"*Proceedings of the 3rd International Conference on Genetic Algorithms*, **1**, 86–91(1989).
2. K. Chellapilla and D. B. Fogel, "Fitness distributions in evolutionary computation: Motivation and examples in the continuous domain," *BioSystems*, **54(1)**, 15–29 (1999).
3. D. B. Fogel and A. Ghozeil, "A note on representations and variation operators," *IEEE Transactions on Evolutionary Computation*, **1(2)**, 159–161 (1997).
4. J. Rowe, D. Whitley, L. Barbulescu, and J.-P. Watson, "Properties of gray and binary representations," *Evolutionary Computation*, **12(1)**,47–76(2004).
5. D. Whitley, "A free lunch proof for gray versus binary encodings," *Proceedings of the Genetic and Evolutionary Computation Conference*, **1**, 726–733 (1999).
6. U. K. Chakraborty and C. Z. Janikow, "An analysis of gray versus binary encoding in genetic search," *Information Sciences*, **156(3)**, 253–269(2003).
7. D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, **1(1)**, 67–82(1997).
8. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
9. F. Rothlauf, *Representations for genetic and evolutionary algorithms*, Springer, 2006.
10. J. H. Holland, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, U Michigan Press, 1975.
11. D. E. Goldberg, K. Sastry, and T. Latoza, " On the supply of building blocks," *Proc. of the 2001 Genetic and Evolutionary Computation Conference*, 336–342 (2001).
12. D. E. Goldberg, *Design of Innovation: Lessons from and for Competent Genetic Algorithms*, Kluwer Academic Publishers, 2002.
13. D. E. Goldberg, K. Sastry, and Y. Ohsawa, "Discovering deep building blocks for competent ge-

netic algorithms using chance discovery via keygraphs," *Chance Discovery*, 276–301(2003).
14. Y.P. Chen and M.H. Lim, *Linkage in Evolutionary Computation (Studies in Computational Intelligence)*, Springer Verlag, 2009.
15. J. Horn and D. E. Goldberg, "Genetic algorithm difficulty and the modality of fitness landscapes," *Foundations of Genetic Algorithms*, **3**, 243–269 (1995).
16. K. Deb and D. E. Goldberg, "Sufficient conditions for deceptive and easy binary functions," *Annals of Mathematics and Artificial Intelligence*, **10(4)**, 385C408(1994).
17. D. E. Goldberg, "Construction of high-order deceptive functions using low-order walsh coefficients," *Annals of Mathematics and Artificial Intelligence*, **5(1)**, 35–48 (1992).
18. S. Forrest and M. Mitchell, "What makes a problem hard for a genetic algorithm? some anomalous results and their explanation," *Machine Learning*, **13(2-3)**, 285–319(1993).
19. D. Whitley, R. B. Heckendorn, and S. Stevens, "Hyperplane ranking, nonlinearity and the simple genetic algorithm," *Information Sciences*, **156(3)**, 123–145(2003).
20. K. Sastry, U.-M. OReilly, D. E. Goldberg, and D. Hill, "Building block supply in genetic programming," *Genetic Programming Theory and Practice*, **9**, 137–154(2003).
21. —, "Construction of fitness functions for which order-1 building blocks can be obtained at multiple loci," *2013 Asia-Pacific Computational Intelligence and Information Technology Conference*, 149–155(2013).
22. N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: Model, taxonomy, and design issues," *IEEE Transactions on Evolutionary Computation*, **9(5)**,474–488(2005).
23. X.S. Chen, Y.S. Ong, M.H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Transactions on Evolutionary Computation*, **15(5)**, 591–607(2011).
24. D. Whitley, "An overview of evolutionary algorithms: Practical issues and common pitfalls," *Information and software technology*, **43(14)**, 817–831(2001).
25. C. Stephens and H. Waelbroeck, "Schemata evolution and building blocks," *Evolutionary computation*, **7(2)**, 109–124(1999).
26. C. R. Stephens and R. Poli, "Coarse-grained dynamics for generalized recombination," *IEEE Transactions on Evolutionary Computation*, **11(4)**, 541–557(2007).
27. L. Altenberg, "The schema theorem and prices theorem," *Foundations of Genetic Algorithms*, **3**, 23–49(1995).