Solving Deceptive Problems Using A Genetic Algorithm with Reserve Selection

Conference Paper · July 2008

DOI: 10.1109/CEC.2008.4630900 · Source: IEEE Xplore

4 authors, including:

Yang Chen
Waseda University

Solving Deceptive Problems Using A Genetic Algorithm with Reserve Selection

Yang Chen, Jinglu Hu, Kotaro Hirasawa, and Songnian Yu

***Abstract*— Deceptive problems are a class of challenging problems for conventional genetic algorithms (GAs), which usually mislead the search to some local optima rather than the global optimum. This paper presents an improved genetic algorithm with reserve selection to solve deceptive problems. The concept "potential" of individuals is introduced as a new criterion for selecting individuals for reproduction, where some individuals with low fitness are also let survive only if they have high potentials. An operator called adaptation is further employed to release the potentials for approaching the global optimum. Case studies are done in two deceptive problems, demonstrating the effectiveness of the proposed algorithm.**

I. Introduction

Genetic algorithms (GAs) are search and optimization techniques based on the theory of biological evolution [1]. They have found successful applications in a wide range of domains such as science, engineering, economics and so on, where GAs usually work more efficiently than conventional methods. However, there still exists problems that are hard for standard GAs.

In standard GAs, as the building block hypothesis suggests, some short, low-order and highly-fit schemata (called building blocks) are used to sample the search space efficiently, thereby directing the evolution to better solutions and even the global optimum []. But in fact, there are some problems where (i) high-order building blocks cannot be easily synthesized by low-order ones, or more seriously (ii) the recombination of low-order schemata guides the search towards solutions that are not globally competitive. These problems are GA-hard due to schema deceptiveness, thus being called deceptive problems. It was claimed that they are the only challenging problems for GAs [2].

The concept of "schema deceptiveness" was first introduced by Goldberg following the early work of Bethke [3]. He designed a GA-deceptive function called the Minimum Deception Problem (MDP) in order to understand what causes difficulty for GAs [4]. Whitley introduced the concepts of fully and consistently deceptive problems, and further analyzed the relationship between deceptive attractors and the global optimum of problems [5]. Homaifar and Deb described the the conditions forming deceptive problems, and further indicated that it is difficult to find all conditions resulting in GA's deception [6], [7]. Richard analyzed three algorithms including the multi-objective fast messy GA, multi-objective Bayesian optimization algorithm, and the non-dominated sorting GA, and compared their performance in solving deception problems [8]. Yang introduced an adaptive group mutation to tackle deception problems in genetic search [9].

As a matter of fact, standard GAs usually suffer from premature convergence in solving deceptive problems, where most individuals are trapped into local optima. The lack of population diversity degenerates the exploration ability of GAs. In the previous work, the authors proposed a genetic algorithm with a reserve selection mechanism (GARS) to maintain population diversity [10]. GARS prevents the population of GAs from prematurely converging on local optima so that it becomes possible to discover better solutions and even the global optimum for deceptive problems. However, it is found that GARS converges more slowly than standard GAs. In this paper, we therefore improve it by introducing the concept of "potential" of individuals as well as an operator called "adaptation" in order to solve deceptive problems more efficiently.

The rest of this paper is organized as follows. Section II gives a brief review of GARS and explains its shortcomings in dealing with deceptive problems. In Section III, we introduce the concept of "potential" into the evaluation of individuals, and employ an operator called "adaptation" to release the "potential" as soon as possible. Empirical studies are done on two deceptive problems to examine the effectiveness of proposed method in Section IV, from which we draw the conclusions in Section V.

II. Genetic Algorithms with Reserve Selection

A. Reserve Selection

As was stated before, standard GAs find difficulties in solving deceptive problems since only highly-fit individuals are favored being selected for reproduction. This inevitably misleads the population-based search to a small part of "promising" areas which in fact contain no global optimum. A new selection scheme known as *reserve selection* is therefore introduced in order to get over "schema deceptiveness". It originates from the idea that some less-fit individuals should also be reserved since they may contain potential building blocks contributing to the formation of global optimum.

Figure 1 illustrates the mechanism of reserve selection. During the evolution, the offspring population is generated in two parts: *non-reserved area* (NRA) and *reserved area* (RA). Similar to the population of standard GAs, NRA

Y. Chen, J. Hu and K. Hirasawa are with the Graduate School of Information, Production and Systems, Waseda University, 2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, Fukuoka 808-0135, Japan (e-mail: yangchen@ruri.waseda.jp, {jinglu,hirasawa}@waseda.jp).

S. Yu is with the School of Computer Engineering and Science, Shanghai University, 149 Yanchang Road, Zhabei District, Shanghai 200072, China (e-mail: snyu@staff.shu.edu.cn).

978-1-4244-1823-7/08/$25.00©2008 IEEE

is produced via the exploitation of highly-fit individuals, and therefore works as an intensified searcher approaching local optima. On the contrary, RA is set up to reserve those less-fit individuals who may contain potential building blocks, thus maintaining a diversified search to explore the global optimum. A *selected table* is employed to label the individuals selected in producing NRA so that they will not be reused to generate RA. The size of RA is called *reserve size*, which plays an important role in balancing exploitation and exploration [11].

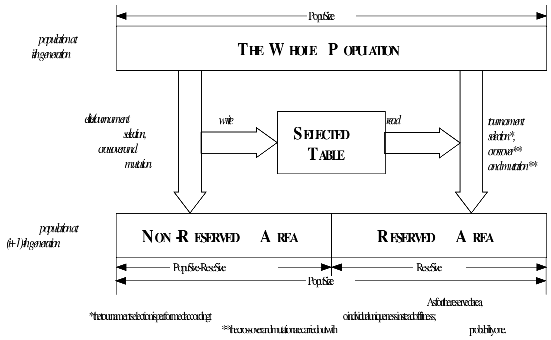

Fig. 1. Reserve selection

The procedure of a genetic algorithm with reserve selection (GARS) is clearly outlined in Algorithm 1, where P denotes the whole population, and X ←select(a,$Y \subseteq Z$) means "to produce X, Y is selected from Z according to a". The only difference from standard GAs lies in the evolution of RA. In reproducing RA, we first select less-fit individuals to be reserved with respect to their *uniqueness* u rather than their fitness f which is typically referred in natural selection. Here, the uniqueness of an individual is defined by its *crowding distance* of fitness, whose computation procedure was described in detail in [10]. The uniqueness-based selection prefers the individuals with distinctive fitness values, consequently helping maintain population diversity in terms of fitness. Besides, genetic operators such as crossover and mutation are applied to RA with probability one (*), by which the potential building blocks buried in reserved individuals are expected to migrate into new host individuals where their powers could be released.

B. The Problems

GARS shows a better performance than standard GAs especially on some complex problems where premature convergence frequently occurs due to deceptive fitness landscapes. However, it usually takes a longer time to converge. Generally speaking, this results from a higher population diversity being maintained so that GARS has to explore a wider range of search space. We argue that it may be rooted in two problems as follows.

- (i) Some less-fit individuals are actually not worth reserving since they embody few building blocks that could be used to synthesize the global optimum. Although they help to maintain population diversity, it may not do any good to evolution;
- (ii) Even if some reserved individuals do contain potential building blocks, the genetic operators formerly employed may not be vigorous enough to bring them into full play. Crossover combines two individuals, while mutation alters an individual only once. Neither of them searches the vicinity of reserved individuals intensively, thereby losing the building blocks inevitably.

Accordingly, it is necessary first to reserve "useful" individuals, and then to make good use of the reserved individuals via effective operators. Next, we present a solution to the issues above mentioned, aiming to speed up the convergence of GARS in solving deceptive problems.

Algorithm 1 A genetic algorithm with reserve selection

begin
  initialize(P);
  (f,u)← evaluate(P);
  while (termination condition) **do**
    $NRA \leftarrow$ select(f,$Q \subseteq P$);
    crossover-mutate(NRA);
    $RA \leftarrow$ select(u,$R \subseteq P \setminus Q$);
    crossover-mutate*(RA);
    $P \leftarrow NRA \cup RA$;
    (f,u)← evaluate(P);
  endwhile
end

III. Potential-based Reserve Selection

In this section, we present a modified version of reserve selection. Different from the previous one which considers uniqueness as the criterion of reserve selection, we are now reserving individuals by taking their *potentials* into account. Even if some individuals are of low fitness, they will be let survive as long as they are of great potentials. In addition, we manage to improve less-fit individuals by releasing their potentials.

A. The Basic Idea

We think that deceptive problems are hard for standard GAs due to their complex fitness landscapes, where

- (i) A spike-like global peak is located within a tiny area of search space. Since it is extremely hard to be sampled by standard GAs without a large population, we expect that it will be discovered by a heuristic search.
- (ii) The global peak is isolated from the outside world by numerous deep valleys, as depicted in Figure 2. These valleys are not attractive to the heuristic search especially when standard GAs are stuck in local peaks or plateaus.

The first issue could be settled by enlarging the population size, but it is sometimes infeasible for large-scale problems. As for the second issue, it may be better if we reserve the

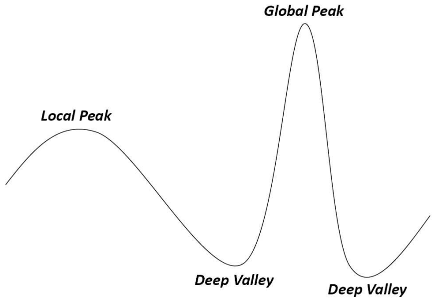

Fig. 2. A global peak isolated by two deep valleys

individuals located in deep valleys rather than those on local peaks or plateaus, since the former may serve as good starting points for climbing global peaks.

B. Evaluate the Potential for An Individual

Hence, we need to learn where an individual is located. This could be achieved by observing its neighborhood. Here, we introduce a new concept, the *potential* of an individual, to describe the capability of its further growth. In fact, it is decided by the class of terrains (deep valleys, local peaks or plateaus) where the individual is located. Technically, the potential of an individual i, denoted by $p(i)$, is measured by the difference in fitness between individual i and its sampling neighbors N_i, as defined in Eq. (1).

$$p(i) = \bar{f}(N_i) - f(i), \tag{1}$$

where $f(i)$ denotes the fitness of individual i, N_i denotes the sampling neighbors of individual i, i.e. a set of individuals that are located in the vicinity of individual i in the search space, and $\bar{f}(N_i)$ denotes the average fitness of N_i, as defined in Eq. (2).

$$\bar{f}(N_i) = \frac{1}{B \times D} \sum_{j=1}^{B} \sum_{k=1}^{D} f(N_i(j,k)), \tag{2}$$

where B denotes the *sampling breadth* and D denotes the *sampling depth*. $N_i(j,k)$ denotes the j-th sampling neighbor on the k-th layer around individual i. An example of the sampling neighbors of individual i is illustrated in Figure 3.

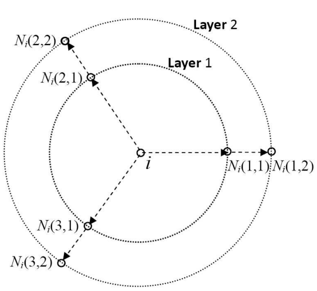

Fig. 3. The sampling neighbors of individual i ($B = 3, D = 2$)

- If $p(i) > 0$, i.e. individual i is of an positive potential, then we call i a *PP individual*. PP individuals are likely to be located in deep valleys, and deserve to be reserved as they may be driving forces in approaching the global optimum as stated before.
- If $p(i) < 0$ (*NP individual*), then individual i is likely to be located on a local peak. Note that it is less probably on the global peak which is too narrow to be sampled. NP individuals are not worth reserving, since it is usually hard to escape from local peaks, let alone their contributions to the global optimum.
- If $p(i) = 0$, then i is likely to be located on either a plateau (*ZP1 individual*) or a slope (*ZP2 individual*). Both of them are less useful than PP individuals, because (i) for ZP1 individuals, it is almost impossible to climb any more, and (ii) for ZP2 individuals, most of them are located on slopes towards local peaks rather than the global one.

Here, the neighborhood relation defined on the search space largely depends on the representation or encoding scheme of GAs, and therefore could vary from problem to problem. An example of neighborhood relation is given in Table I. Besides, the evaluation of potential is somewhat influenced by the sampling breadth B and depth D. If they are large, the evaluation will be more precise whereas it is more time-consuming. In practice, we limit the value of $B \times D$ so as to avoid too many fitness evaluations.

TABLE I

AN EXAMPLE OF NEIGHBORHOOD RELATION

Encoding scheme	Neighborhood relation
binary string	flip one bit
integer permutation	swap two digits
floating-point vector	Gaussian mutation

During the evolution, the potential is evaluated for each individual as another criterion of selection (for details, see Algorithm 3). In reserving less-fit individuals, we not only consider their uniqueness in order to explore a wide search space as previously done, but also prefer "useful" ones (i.e. those with high potentials) that are expected to lead the search to better solutions for deceptive problems.

C. Release Potentials by the Adaptation Operator

Although individuals with high potentials are reserved, we still need to make good use of them by releasing their potentials. Since most reserved individuals are PP ones located in deep valleys, we try to pull them up by using a new operator called *adaptation*, approaching better solutions and even the global optimum. Algorithm 2 describes the procedure of adaptation operator, where B, D, $f(i)$ and $N_i(j,k)$ were defined in Section 3.2. The procedure starts from a reserved individual i, and iteratively moves to a better solution i' in its neighborhood. In this way, the adaptation operator releases the potentials of reserved individuals successfully by pulling them from deep valleys, as shown in Figure 4.

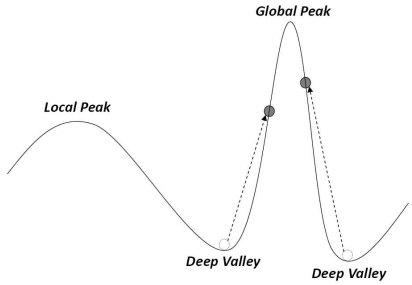

Fig. 4. The adaptation operator pulls PP individuals from deep valleys

Algorithm 2 The adaptation operator

input: A reserved individual i.
output: The improved individual i'.
begin
    $k \leftarrow 1$
    while $k \leq D$ **do**
        $i' \leftarrow i$
        $j \leftarrow 1$
        while $j \leq B$ **do**
            if $f(N_i(j,k)) > f(i')$
              $i' \leftarrow N_i(j,k)$
            endif
            $j \leftarrow j+1$
        endwhile
        $i \leftarrow i'$
        $k \leftarrow k+1$
    endwhile
end

The adaptation operator is applied together with crossover and mutation, playing an important role in searching the global optimum. The algorithm which incorporates the potential-based reserve selection and the adaptation operator is given in Algorithm 3. Different from the previous one (see Algorithm 1), in addition to the mechanism of uniqueness-based selection, we also reserve individuals by referring their potentials and apply adaptation to have the potentials released. A parameter called *adaptation rate* $p_a \in [0,1]$ is checked at each generation so that both mechanisms are carried out alternately.

IV. Solving Deceptive Problems: Case Studies

In this section, we examine the performance of the improved algorithm on two deceptive problems, and compare the results with those obtained by standard GAs.

A. Case Study 1: An Order-3 Deceptive Problem

We first consider Goldberg's order-3 deceptive problem, which has been widely used as a test bed by researchers. The problem is defined in Table II, where a set of binary strings corresponds to a set of function values. The problem is challenging since it has a deceptive attractor 000 which is a local optimum superior to all its neighbors in the search space. The problem is also fully deceptive as the local optimum has an attractive basin that covers most of the search space, while an isolated spike represents the global optimum 111 which is exactly the complement of the local optimum.

Algorithm 3 An improved genetic algorithm with reserve selection

begin
    initialize(P);
    (f,u,p)$\leftarrow$ evaluate(P);
    while (termination condition) **do**
        $NRA \leftarrow$ select(f,$Q \subseteq P$);
        crossover-mutate(NRA);
        $r \leftarrow$ rand(0,1)
        if $r > p_a$
          $RA \leftarrow$ select(u,$R \subseteq P \setminus Q$);
          crossover-mutate*(RA);
        else
          $RA \leftarrow$ select(p,$R \subseteq P \setminus Q$);
          adaptation(RA);
        endif
        $P \leftarrow NRA \cup RA$;
        (f,u,p)$\leftarrow$ evaluate(P);
    endwhile
end

TABLE II
An order-3 deceptive problem

Binary String	Function Value
000	28
001	26
010	22
011	0
100	14
101	0
110	0
111	30

Since a three-bit binary function itself is too small to demonstrate any search algorithm, Goldberg constructed a bigger one [12]. It is composed of 10 three-bit deceptive subfunctions, each of which is order-3 deceptive. The value of a 30-bit function is thus defined as the sum of the value of its 10 subfunctions. For this problem, there are totally $2^{30} = 1.07 \times 10^9$ points in the search space, where $2^{10} = 1024$ points are optima (each subfunction has 2 optima), among which only one is the global optimum (i.e. the string with all 1s).

The problem becomes more difficult when the bits of subfunctions are reordered so that the three bits of each subfunction are maximally separated. Hence, the three bits of the i-th subfunction are located at position i, $i+10$ and $i+20$ respectively ($i = 1 \ldots 10$). The combination of deception and ordering makes the problem difficult for standard GAs to discover the global optimum. It was shown that standard GAs consistently converge to a suboptimal solution to the problem, with each subfunction converging to 000 instead of 111 [12].

In this test, we check the effectiveness of proposed method (GARS) by comparing it with standard GAs (GA). Table III lists the parameter settings for the experiment, which are tuned for standard GAs before runs to optimize the performance. The genetic operators adopted here are tournament selection with elitist strategy, one-point crossover and bit mutation. Besides, we set *sampling breadth* $B = 30$ and *sampling depth* $D = 1$ for GARS. The results are shown in Table IV-VI, where three groups of tests with different number of deceptive subfunctions 10, 100 and 200 are conducted respectively. We measure the best ($max.$), the mean ($avg.$) and the standard deviation (SD) of the optimal fitness obtained in 100 independent runs as yardsticks of performance evaluation.

TABLE III

PARAMETER SETTINGS FOR THE ORDER-3 DECEPTIVE PROBLEM

Parameter	Value
population size	10
generation number	100
tournament size	2
crossover rate	0.5
mutation rate	1.0
adaptation rate	1.0

TABLE IV

COMPARE GARS WITH GA IN THE ORDER-3 DECEPTIVE PROBLEM WITH 10 DECEPTIVE SUBFUNCTIONS (THE GLOBAL MAXIMUM IS 300)

Method	$max.$	$avg.$	SD
GA	296	287.52	**3.27561**
GARS	**300**	**292.2**	3.58329

As can be learned from Table IV, standard GAs are unable to find the global optimum in 100 runs since they are prone to premature convergence. However, the proposed method solves this deceptive problem successfully. As the number of deceptive subfunctions increases to 100 and 200 (see Table V and VI, the problem becomes more and more difficult for standard GAs. Within the same number of generations, however, GARS performs better than GA since it releases the potentials of reserved individuals by adaptation which could easily improve poor solutions for this problem.

TABLE V

COMPARE GARS WITH GA IN THE ORDER-3 DECEPTIVE PROBLEM WITH 100 DECEPTIVE SUBFUNCTIONS (THE GLOBAL MAXIMUM IS 3000)

Method	$max.$	$avg.$	SD
GA	2642	2545.24	44.5554
GARS	**2904**	**2886.82**	**8.49986**

TABLE VI

COMPARE GARS WITH GA IN THE ORDER-3 DECEPTIVE PROBLEM WITH 200 DECEPTIVE SUBFUNCTIONS (THE GLOBAL MAXIMUM IS 6000)

Method	$max.$	$avg.$	SD
GA	4686	4503.42	89.0525
GARS	**5688**	**5607.88**	**32.7827**

B. Case Study 2: A Highly Deceptive 2D Problem

The second problem we examine is a simple but highly deceptive 2D problem [13]. The domain space is a unit square $[0,1] \times [0,1]$, where two narrow regions $I_1 : [a, a+\delta] \times [0,1]$ and $I_2 : [0,1] \times [b, b+\delta]$ for some $a, b, \delta \in [0,1]$ are defined. Typically δ is chosen as a small value so that I_1 and I_2 do not occupy much of the domain space. The fitness function to be maximized is defined as follows, which is depicted in Figure 5.

$$f(x,y) = \begin{cases} 1 & \text{if } (x,y) \in I_1 \setminus I_2, \\ 2 & \text{if } (x,y) \in I_2 \setminus I_1, \\ 3 & \text{if } (x,y) \notin I_1 \cup I_2, \\ 4 & \text{if } (x,y) \in I_1 \cap I_2. \end{cases}$$

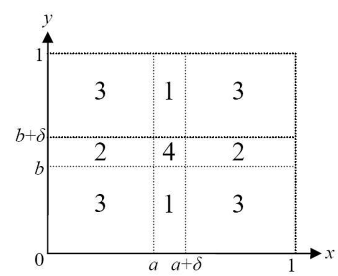

Fig. 5. A highly deceptive 2D problem

For this problem, the mutation operator is set to randomly modify either the x or y coordinate of an individual, while the crossover operator is set to combine the x coordinate from one individual with the y coordinate from another to produce a new offspring. Under these genetic operators, this is a highly deceptive optimization problem that is very difficult for standard GAs. The size of the domain where the function is maximized to 4 is only δ^2, which is tiny for small values of δ. Moreover, the local maxima whose fitness is 3 cover most of the domain space, and the only way to the global maximum is by leaving the local maxima and exploring the space of individuals with lower fitness of 1 or 2.

The parameter settings for the experiment are summarized in Table VII. We apply the same genetic operators as in the previous case, and $B = 4, D = 2$ is set for GARS. The results are given in Table VIII-X, where three groups of tests with different setting of $\delta = 0.1$, 0.05 and 0.01 are conducted respectively. Similarly, all the results are the statistical outcome of the optimal fitness obtained in 100 independent runs.

TABLE VII

PARAMETER SETTINGS FOR THE HIGHLY DECEPTIVE 2D PROBLEM

Parameter	Value
population size	10
generation number	100
tournament size	2
crossover rate	0.5
mutation rate	1.0
adaptation rate	0.0

TABLE VIII

COMPARE GARS WITH GA IN THE HIGHLY DECEPTIVE 2D PROBLEM WITH $\delta = 0.1$

Method	$max.$	$avg.$	SD
GA	4	3.5	0.5
GARS	4	**4**	**0**

Evidently, the deceptiveness increases while the value of δ decreases. For a large value of $\delta = 0.1$, the performance is not so bad even if standard GAs are used, as shown in Table VIII. Nevertheless, one observes that GARS could always find the global optimum every time in 100 runs. Table IX and X indicate that the performance of standard GAs degenerates a lot when $\delta = 0.05$ and 0.01, whereas GARS is still effective since the crossover of two reserved individuals with fitness 1 and 2 respectively results in the global maximum with fitness 4.

TABLE IX

COMPARE GARS WITH GA IN THE HIGHLY DECEPTIVE 2D PROBLEM WITH $\delta = 0.05$

Method	$max.$	$avg.$	SD
GA	4	3.16	0.366606
GARS	4	**4**	**0**

TABLE X

COMPARE GARS WITH GA IN THE HIGHLY DECEPTIVE 2D PROBLEM WITH $\delta = 0.01$

Method	$max.$	$avg.$	SD
GA	4	3.01	**0.0994987**
GARS	4	**3.65**	0.47697

V. CONCLUSIONS AND FUTURE WORKS

Generally speaking, standard GAs find difficulties in solving deceptive problems where they usually suffer from premature convergence. In order to attack them efficiently, it is necessary to (1) reserve some less-fit individuals to maintain population diversity; (2) take effective measures to make good use of reserved individuals for global optimization. In this paper, we introduce the concept of "potential" into the evaluation of individuals and reserve those individuals with high potentials. Moreover, a new operator called adaptation is performed on reserved individuals to release their potentials. The improved algorithm could help to speed up the convergence, solving deceptive problems more efficiently than standard GAs. We are now trying our algorithm on other deceptive problems in the literature as well as some real-world problems. Furthermore, the algorithm will be compared with other methods available for deceptive problems.

ACKNOWLEDGMENT

This research was supported by "Ambient SoC Global COE Program of Waseda University" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

REFERENCES

[1] J. H. Holland, *Adaptation in Natural and Artificial System*, Ann Arbor: The University of Michigan, 1975.

[2] R. Das and L. D. Whitley, "The only challenging problems are deceptive: global search by solving order-1 hyperplanes," in *Proceedings of the Fourth International Conference on Genetic Algorithms*, R. Belew and L. Booker, editors, 1991, pages 166–173.

[3] A. D. Bethke, "*Genetic Algorithm as Function Optimizers*," PhD thesis, Department of Computer and Communication Science, University of Michigan, 1980.

[4] D. E. Goldberg, "Simple genetic algorithm and the minimal deceptive problem," in *Genetic Algorithms and Simulated Annealing*, L. Davis, editor, San Mateo, CA: Morgan Kaufmann, 1987, pages 74–88.

[5] L. D. Whitley, "Fundamental principles of deception in genetic search," in *Foundations of Genetic Algorithms 1*, G. Rawlins, editor, San Mateo, CA: Morgan Kaufmann, 1991, pages 221–241.

[6] A. Homaifar, X. Qi, and J. Fost, "Analysis and design of a general GA deceptive problem," in *Proceedings of the Fourth International Conference on Genetic Algorithms*, R. Belew and L. Brooker, editors, 1991, pages 196–203.

[7] K. Deb, and D. E. Goldberg, "Sufficient conditions for deceptive and easy binary functions," *Annals of Mathematics and Artificial Intelligence*, vol. 10, no. 4, pages 385–408, October 1993.

[8] O. D. Richard, P. K. Mark, and B. L. Gary, "Multi-objective fast messy genetic algorithm solving deception problems," in *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, 2004, pages 1502–1509.

[9] S. Yang, "Adaptive group mutation for tackling deception in genetic search," *WSEAS Transactions on Systems*, vol. 3, no. 1, pages 107–112, January 2004.

[10] Y. Chen, J. Hu, K. Hirasawa, and S. Yu, "GARS: an improved genetic algorithm with reserve selection for global optimization," in *Proceedings of the 2007 Genetic and Evolutionary Computation Conference*, pages 1173–1178, London, United Kingdom, July 2007.

[11] Y. Chen, J. Hu, K. Hirasawa, and S. Yu, "Performance tuning of genetic algorithms with reserve selection," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, pages 2202–2209, Singapore, September 2007.

[12] D. E. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: Motivation, analysis and first results," *Complex Systems*, 3:493–530, May 1989.

[13] M. Hutter. "Fitness uniform selection to preserve genetic diversity," in X. Yao, editor, *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, pages 783–788, Washington D.C., USA, May 2002.