A Proposed Genetic Algorithm Selection Method.

Dr. Elgasim Elamin Elnima Ali
King Saud University- ccis
elgasime@ccis.ksu.edu.sa

Abstract

Genetic algorithms (GAs) are stochastic search methods that mimic natural biological evolution. Genetic algorithms are broadly used in optimisation problems. They facilitate a good alternative in problem areas where the number of constraints is too large for humans to efficiently evaluate.

This paper presents a new selection method for genetic algorithms. The new method is tested and compared with the Geometric selection method. Simulation studies show remarkable performance of the proposed GA selection method .The proposed selection method is simple to implement, and it has notable ability to reduce the effected of premature convergence compared to other method.

The proposed GAs selection method is expected to help in solving hard problems quickly, reliably, and accurately

1. Introduction

Genetic algorithms were developed by John Holland at the University of Michigan in the early 1970's [1]. Genetic algorithms are theoretically and empirically proven to provide robust search in complex spaces (Goldberg, 1989)[2].

Genetic algorithms are stochastic search methods that mimic natural biological evolution. Genetic algorithms operate on a population (a group of individuals) of potential solutions applying the principle of survival of the fittest to generate improved estimations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness and breeding them together using genetic operators inspired by natural genetics. This process leads to the evolution of better populations than the previous populations [3].

Fitness function is the measure of the quality of an individual. The fitness function should be designed to provide assessment of the performance of an individual in the current population.

In selection the individuals producing offspring are chosen. The selection step is preceded by the fitness assignment which is based on the objective value. This fitness is used for the actual selection process.

There are many types of selection methods used in genetic algorithms, including:

- Rank-based fitness assignment
- Roulette wheel selection
- Stochastic universal sampling
- Local selection
- Truncation selection
- Tournament selection

A decision about the method of selection to be applied is one of the most important decisions to be made. Selection is responsible for the speed of evolution and is often cited as the main reason in cases where premature convergence halts the success of a genetic algorithm.

Crossover. This is a version of artificial mating. Individuals with high fitness should have high probability of mating. Crossover represents a way of moving through the space of possible solutions based on the information gained from the existing solutions.

Mutation. Mutation represents innovation. Mutation is important for boosting the search; some of evolutionary algorithms rely on this operator as the only form of search.

Many researchers have argued that GAs should be modeled more closely to natural genetics (Luke, 2003)[4], and many of them have already done so (Burke, De Jong, Grefenstette, Ramsey, & Wu, 1998)[5]. The proposed method followed the same tradition and developed a breeding method which more closely simulates natural mating.

It is known that genetic algorithms are inspirations from natural genetics. This does not mean that genetic algorithms should to be identical to their natural counterpart.

Genetic algorithms run on benchmarking functions to evaluate their performance. In this experiment we adopt

14 functions proposed in a paper by Jason G. Digalakis, Konstantinos G. Margaritis [6].

Figure 1: Pseudo-code of the standard genetic algorithm.

```
BEGIN  GA
  gen:=0  { generation counter }
  Initialize population P(g)
  Evaluate population P(g)
  done:=false
  WHILE not done DO
    gen:=gen+1
    Select P(gen) from P(gen -1)
    Crossover P(gen)
    Mutate P(gen)
    Evaluate P(gen)
 done:=Optimization criteria met?
 END WHILE
  Oputput best solution
END GA
```

This paper is organized under six sections. The first deals with some basic concepts from genetics algorithms. In section two we describe the benchmarking functions. Section three presents the proposed selection method. Section four is devoted for displaying the results, while the fifth section deals with the discussions. We end with a summary and conclusions.

2. Method

Genetic algorithms start with a population of elements representing a number of potential solutions to the problem at hand. By applying crossover and mutation to some of the genotypes an element or group of elements is hoped to emerge as the optimal (best) solution to the problem at hand. There are many types of crossover and mutation methods used in genetic algorithms and some of them are specific to the problem at hand (Goldberg, 1989)[2]. In this research we implement the most basic type of crossover (arithmetic crossover).

Mutation is another recombination technique. It is used to make sure all the elements in a population are not homogeneous and diversity is maintained. Mutation can help genetic algorithms escape local optima in the search for the global optima. Mutation rate is usually ranges not higher than 30% [7]. The MultiNonUniform mutation method is used in experiments at a rate of 5%.

The selection function [8, 9, 10, 11] chooses parents for the next generation based on their fitness. In our experiments we take the traditional Geomtric selection method (Geomtric) which is tested against the proposed method HighLowFit (HLF), which described as follows:

1) Sort the population by the objective value
2) Divide the population to two parts. The upper part is the high fit (HF) and the lower part is the low fit (LF). Name the point of division m.
3) Always select the first parent (P1) form the HF part and the other parent (P2) from the LF part.
4) Compute offsprings from P1 and P2 as usual.
5) Append new offsprings to the end of the current population
6) Sort and cut the tail to maintain the same population size.
7) Repeat the above steps, until termination criteria are met.

The point of division m can be specified as percent. It can generate values form 1 up to 50% of the population size. The actual value selected for m affects the speed of evolution. In our experiment we set m between 7% to 45%.

In the following sections we will present results produced by the two methods.

Variations of the original HighLowFit method are:

a) Change step 6 to: Substitute (Replace) all new offsprings at the tail of the current population maintaining the same population size. Call this variation (HighLowFitR).
b) Change step 6 to: Substitute the higher fit percentage (Partial Replacement) of offsprings at the tail of the current population maintaining the same population size. Call this variation (HighLowFitPr).

3. Benchmarking Functions

3.1 De Jong's function 1

The simplest test function is De Jong's function 1[12]. It is continuos, convex and unimodal.

function definition

$$f(x) = \sum_{i}^{N} x_i^2$$

$-5.12 \leq x_i \leq 5.12.$

global minimum $f(x)=0$; $x_i = 0$, $i=1:N$.

3.2 Rosenbrock's valley

Rosenbrock's valley is a classic optimization problem [12]. The global optimum is inside a long, narrow, parabolic shaped flat valley. Convergence to the global optimum is difficult.

function definition

$$f(x) = \sum_{i}^{N-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$$

$-2.048 \le x_i \le 2.048$.

global minimum f(x)=0; x_i =1, i=1:n.

3.3 De Jong's FUNction 3

Step function [12] is the representative of the problem of flat surfaces. This function is piecewise continuous step function.

$$f(x) = \sum_{i}^{N} |x_i|$$

$-5.12 \le x_i \le 5.12$.

global minimum f(x)=0; x_i =0, i=1:n.

3.4 De Jong's FUNction 4

De Jong's Function 4 [12] is a simple unimodal function with added Gaussian noise.

$$f(x) = \sum_{i}^{N} i x_i^4 + Gasuss\ (0,1)$$

$-1.28 \le x_i \le 1.28$

global minimum f(x)=0; x_i =0, i=1:n.

3.5 Branins's rcos function

The Branin rcos function is a global optimization test function [13]. The function has 3 global optima.

function definition

$$f(x) = (x_2 - (5.1/(4\pi^2))x_1^2 + (5/\pi)x_1 - 6)^2 + 10(1 - (1/(8\pi)))\cos(x_1) + 10$$

$-5 \le x_1 \le 10$, $0 \le x_2 \le 15$.

global minimum $f(x_1,x_2)$=0.397887; (x_1,x_2)=(-π,12.275), (π,2.275), (9.42478,2.475).

3.6 Easom's function

The Easom function is a unimodal test function [14], where the global minimum has a small area relative to the search space.

function definition

$$f(x) = -\cos(x_1)\cos(x_2)\exp(-((x_1 - \pi)^2 + (x_2 - \pi)^2))$$

$-100 \le x_i \le 100$, i=1:2.

global minimum f(x1,x2)=-1; (x1,x2)=(π, π).

3.7 Six-hump camel back function

The 2-D Six-hump camel back function is a global optimization test function. Within the bounded region are six local minima, two of them are global minima.

function definition

$$f(x) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1 \cdot x_2 + (-4 + 4x_2^2)\, x_2^2$$

$-3 \le x_1 \le 3$, $-2 \le x_2 \le 2$.

global minimum $f(x_1,x_2)$=-1.0316; (x_1,x_2)=(-0.0898,0.7126), (0.0898,-0.7126).

3.8 Rastrigin's function

Rastrigin's function is based on function De Jong 1 [7] with the addition of cosine modulation to produce many local minima. Thus, the test function is highlymultimodal. However, the locations of the minima are regularly distributed.

function definition

$$f(x) = 10N + \sum_{i}^{N} (x_i^2 - 10\cos(2\pi x_i))$$

$-5.12 \le x_i \le 5.12$.

global minimum f(x)=0; x_i =0, i=1:n.

3.9 Schwefel's function

Schwefel's function [7] is deceptive in that the global minimum is geometrically distant from the next best local minima.

function definition

$$f(x) = \sum_{i}^{N} -x_i \sin(\sqrt{|x_i|})$$

$-512 \le x_i \le 512.$

global minimum f(x)=-N·418.9829; x_i =420.9687, i=1:n.

3.10 Michalewicz's function

The Michalewicz function [15] is a multimodal test function (*n*! local optima). The parameter *m* defines the "steepness" of the valleys or edges. Larger *m* leads to more difficult search. Let: (N=10; m=10)

function definition

$$f(x) = -\sum_{i}^{N-1} \sin(x_{i+1})\sin(\frac{2x_{i+1}^2}{\pi})^{2m} + \sin(x_i)\sin(\frac{x_{i+1}^2}{\pi})^{2m}$$

$0 \le x_i \le \pi.$

global minimum = - 4.00391571; x_i = 1.57080 1.57104 1.57104 1.57104 1.57104

3.11 Ackley Function

Ackley function is a widely used multimodal test function.

$$f(x) = \sum_{i=1}^{N-1} \left(3(\cos(2x_i) + \sin(2x_{i+1})) + \frac{\sqrt{x_{i+1}^2 + x_i^2}}{e^{0.2}} \right)$$

$-32.768 \le x_i \le 32.768$

global minimum = -10.4614373402; x_i = -1.51573 -1.11506 -1.10393 -0.74712

3.12 Shekel Function

$$f(x) = -\sum_{i=1}^{M} \frac{1}{(x - A_i)(x - A_i)^T + C_i}$$

$0 \le x_i \le 10$

Where:

$$A = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 5 & 3 & 3 \\ 8 & 1 & 8 & 1 \\ 6 & 2 & 6 & 2 \\ 7 & 3.6 & 7 & 3.6 \end{bmatrix} \qquad C = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \\ 0.7 \\ 0.5 \\ 0.5 \end{bmatrix}$$

global minimum =10.5364098167;
x_i = 4.00075 4.00059 3.99966 3.99951

3.13 Paviani Function

$$f(x) = \sum_{i=1}^{N} \left(Ln^2(x_i - 2) + Ln^2(10 - x_i) \right) - \left(\prod_{i=1}^{N} x_i \right)^{0.2}$$

global minimum = 45.7784697074 ; x_i = 9.350266

3.14 Goldstein-Price's function

The Goldstein-Price function [16] is a global optimization test function.

function definition

$$\begin{aligned} f(x) = {} & [1 + (x_1 + x_2 + 1)^2 \\ & (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 \cdot x_2 + 3x_2^2] \\ & [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 \\ & + 48x_2 - 36x_1 \cdot x_2 + 27x_2^2)] \end{aligned}$$

$-2 \le x_i \le 2$, i=1:2.

global minimum $f(x_1,x_2)=3$; $(x_1,x_2)=(0,-1)$.

4. Results

Using the selection methods and functions described in the previous sections we tried to determine the effect on the performance of a GA. The implementation of our programs was carried out with the help of the GAOT library.

As the purpose of this experiment was to measure the performance of the proposed selection methods, this section contains the results obtained from the genetic selection methods runs. The analysis of selection techniques was based on a comparison of their respective performance estimated as the number of function evaluations. The raw data obtained from the runs comprised the average, minimum and maximum of 10 runs for each function of the set of functions under consecration.

The following parameters are used in experiments:

- Population sizes: 100.
- Population selection methos: a) Geometric method, b) HighLowFit (HLF)
- Crossover Operator: simple arithmetic crossover (X1) or modified arithmetic crossover (X2).
- Mutation: MultiNonUniform with probability 5%
- Algorithm ending criteria: the executions stop on convergence to the optima within 1e-7, or reaching 25000 generations.
- Pseudorandom generators : MATLAB with seed reset to 1 - 10
- Fitness function: Objective value.
- Each function is run 10 times.

We examined the average, minimum and maximum of the evaluations for each function

Table 1 shows summary for the two selection methods with simple arithmetic crossover (X1) method.

Simple arithmetic crossover (X1)

	Geometric		
Function	**Mean**	**Min**	**Max**
De Jong 1-2	5,875	982	23,611
Branin's Rcos	41,219	1,766	76,849
Eesoms	15,548	1,570	41,700
six-hump	16,386	1,178	45,290
Rosenbrocks	519,248	1,540	936,393
Rastrigins	977,250	732,425	1,243,422
Schwefels	681,134	320,132	2,394,518
De Jong 3	266,399	81,441	464,080
De Jong 4	26,597	13,707	46,981
Ackley 4	75,652	23,846	110,765
Michalewicz	152,176	73,984	330,578
Shekel-4	54,102	29,349	96,589
paviani-10	181,581	85,821	334,206
Goldstein-Price	46,199	1,956	90,805
Mean	218,526	97,836	445,413

Table 1 continued:

Simple arithmetic crossover (X1)

	HighLowFit			
Function	**Mean**	**Min**	**Max**	**Mean%**
De Jong 1-2	1,033	308	2,688	17.6%
Branin's Rcos	2,582	644	6,818	6.3%
Eesoms	3,273	910	11,101	21.1%
six-hump	1,260	518	2,562	7.7%
Rosenbrocks	122,859	588	205,647	23.7%
Rastrigins	139,147	107,268	165,340	14.2%
Schwefels	144,379	38,598	343,720	21.2%
De Jong 3	51,657	38,384	80,968	19.4%
De Jong 4	4,589	3,024	5,726	17.3%
Ackley 4	16,801	7,210	27,392	22.2%
Michalewicz	104,920	32,336	338,724	68.9%
Shekel-4	53,946	6,692	164,454	99.7%
paviani-10	29,463	20,412	46,336	16.2%
Goldstein-Price	3,714	588	8,372	8.0%
Mean	48,545	18,391	100,703	22.2%

Table 1. Summary of results of 10 runs (X1).

Table 2. Shows summary for the two selection methods using the modified crossover (X2) method.

Table 2. Summary of results of 10 runs (X2).

Modified arithmetic crossover (X2)

	Geometric		
Function	**Mean**	**Min**	**Max**
De Jong 1-2	3,682	1,350	13,164
Branin's Rcos	23,954	969	89,825
Eesoms	24,459	2,330	52,023
six-hump	4,850	1,174	21,148
Rosenbrocks	216,353	1,376	1,391,803
Rastrigins	776,933	494,494	1,395,963
Schwefels	315,859	196,013	556,822
De Jong 3-5	210,954	69,885	447,029
De Jong 4-8	32,732	20,555	41,802
Ackley 4	85,479	22,155	168,321
Michalewicz	145,522	66,809	295,501
Shekel-4	128,074	41,873	415,960
paviani-10	170,367	118,503	264,158
Goldstein-Price	28,946	1,738	102,283
Mean	154,869	74,230	375,414

Table 2 continued:

Modified arithmetic crossover (X2)

	HighLowFit			
Function	**Mean**	**Min**	**Max**	**Mean%**
De Jong 1-2	482	378	630	13.1%
Branin's Rcos	580	364	684	2.4%
Eesoms	949	798	1,090	3.9%
six-hump	559	462	714	11.5%
Rosenbrocks	689	574	1,064	0.3%
Rastrigins	72,085	47,880	98,126	9.3%
Schwefels	37,546	26,208	48,006	11.9%
De Jong 3	23,296	8,722	35,868	11.0%
De Jong 4	4,025	2,772	5,348	12.3%
Ackley 4	7,759	3,948	12,698	9.1%
Michalewicz	27,692	4,214	71,653	19.0%
Shekel-4	130,156	7,980	348,304	101.6%
paviani-10	20,719	15,092	29,288	12.2%
Goldstein-Price	739	560	1,232	2.6%
Mean	23,377	8,568	46,765	15.1%

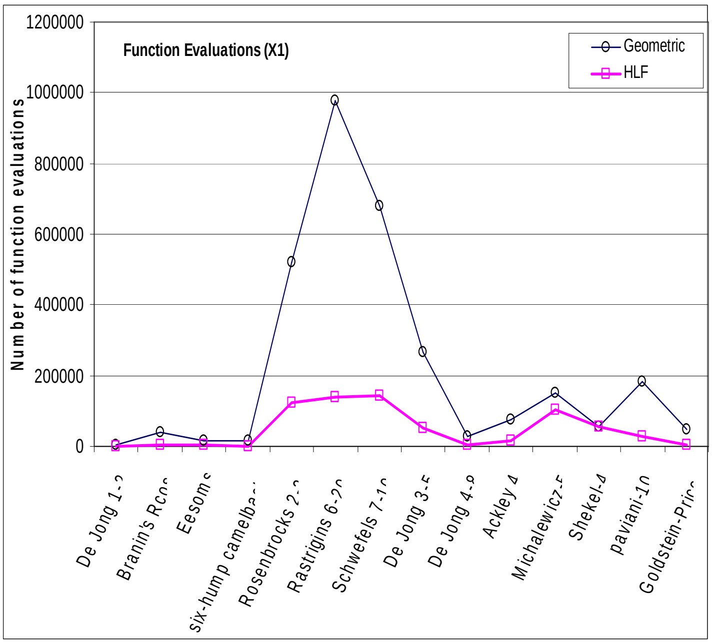

Figure 1. Number of function evaluations of the two methods (X1).

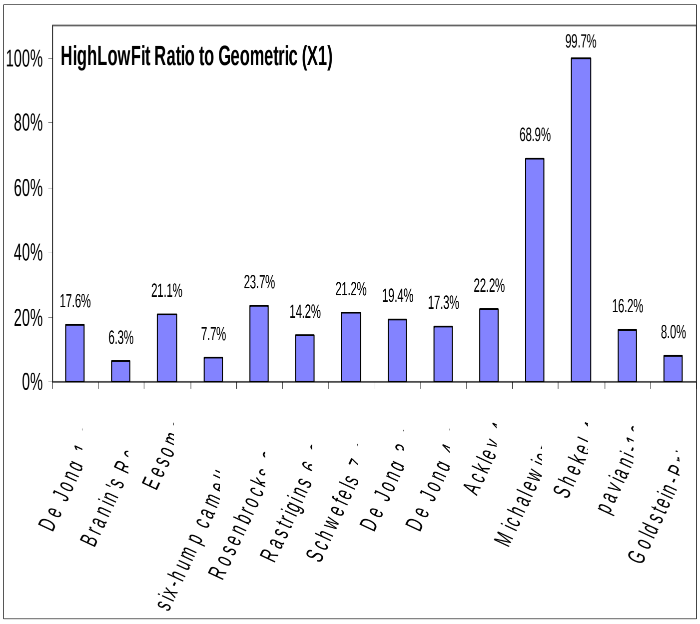

Figure 2. Percentage of HighLowFit method to the Geometric method (X1).

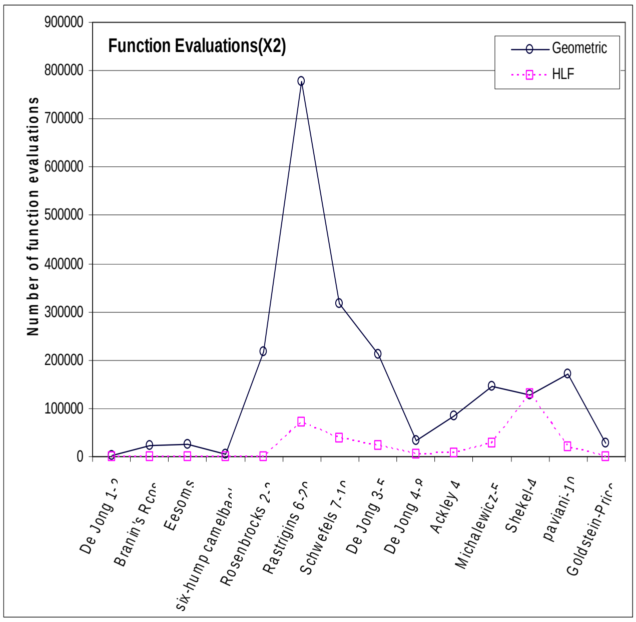

Figure 3. Number of function evaluations of the two methods (X2).

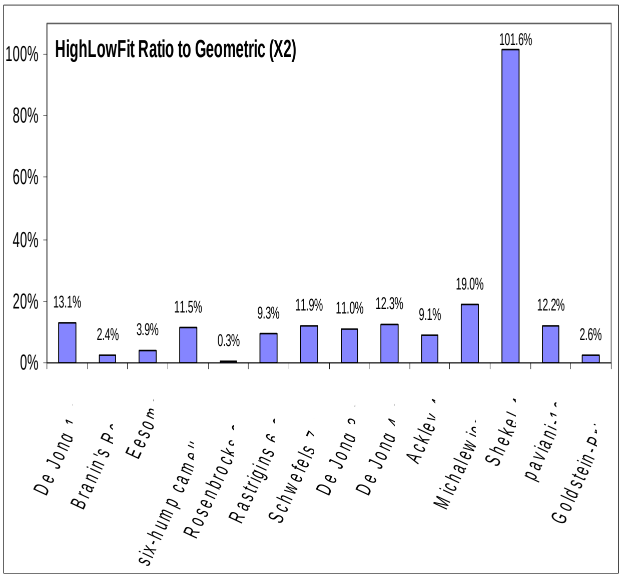

Figure 4. Percentage of HighLowFit method to the Geometric method (X2).

5. Discussion

Numerous research efforts have been made to establish the relative suitability of various selection techniques that can be applied in genetic algorithms. A major problem in any attempt at ranking performance has been the apparent bias of different operators to different tasks. Hence, we run the experiments on two crossover operators. One of them is well known simple arithmetic crossover method (X1), and the other one is a modified version of it (X2).

The detailed results show that the Geometric method do not converged only once (Schwefels function) in all the 140 runs under the simple arithmetic crossover method and two time (Shekel function) under the crossover method. On the other hand HighLowFit selection method do not converged only once (Shekel function).

From table 1, the ratio of number of function evaluations for HLF method compared with Geometric method ranges between 6.3% and 99.7%, and the average is 22.2%, when using X1. The corresponding scores for X2 are, 0.3%, 101.6%, and 15.1% respectively.

Results of table 1 and 2 are graphed in figures 1-4.

The HLF selection method achieved better performance (Number of evaluations) compared with Geometric selection method in all cases except for Shekel function (101.6%) where it failed to converge at on trial of it is runs.

The choice of population size 100 is consistent with the experiments conducted by many researchers, but it can be lowered or raised. We found populations of 60-150 to be acceptable for our task, but in general we state that the optimum population size depends on the complexity of the domain.

The other two variants of HLF (HighLowFit-R, HighLowFit-Pr) produced (51.4.2% , 67.9%) for the X1 and (30.2% , 40.9%) for X1 compared to the Geometric selection method.

6. Conclution

Restricting mating to take place between higher fit and lower fit parents only keeps good diversity in population for many generations. In all 14 functions listed in section 3, we found good results in favor of the HLF selection method and it is variants. A through investigation of the HLF method would be needed to reveal it is behavior under a Varity of genetic algorithm controls.

References

[1] Holland, J. H., *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press, 1975.

[2] Goldberg, D. E., *Genetic Algorithms in Search, Optimization*, and Machine Learning. Reading, Mass.: Addison-Wesley, 1989.

[3] E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter Control in Evolutionary Algorithms", *IEEE Transactions on Evolutionary Computation*, IEEE, 3(2), 1999, pp. 124-141.

[4] Luke, S., "Modification point depth and genome growth in genetic programming". *Evolutionary Computation*, 1998, 11(1),67-106.

[5] Burke, D. S., De Jong, K. A., Grefenstette, J. J., & C. L. Ramsey, "Putting more Genetics into Genetic Algorithms". *Evolutionary Computation* 6(4), 387-410.

[6] Digalakis Jason, Margaritis Konstantinos, "An Experimental study of Benchmarking Functions for Evolutionary Algorithms". *International Journal of Computer Mathemathics*, April 2002, Vol. 79, pages 403-416.
[7] Lima, C., Sastry, K., Goldberg, D. E., Lobo, F., "Combining competent crossover and mutation operators: A probabilistic model building approach". *Proceedings of the 2005 Genetic and Evolutionary Computation Conference*. 2005, 735—742.
[8] Bäck, T. and Hoffmeister, *F*., "Extended Selection Mechanisms in Genetic Algorithms". *in [ICGA4]*, 1991, pp. 92-99.
[9] Baker, J. E., "Adaptive Selection Methods for Genetic Algorithms". in *[ICGA1]*, pp. 101-111, 1985.
[10] Blickle, T. and Thiele, L., *A Comparison of Selection Schemes used in Genetic Algorithms*. TIK Report Nr. 11, December 1995,
[11] Goldberg, D. E. and Deb, K., "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms". *in [FGA1]*, pp. 69-93, 1991.
[12] De Jong, K., *An analysis of the behavior of a class of genetic adaptive systems, Doctoral dissertation*, University of Michigan, Dissertation Abstracts International, 36(10), 5140B, University Microfilms No. 76-9381, 1975.
[13] Branin, F. K., "A widely convergent method for finding multiple solutions of simultaneous nonlinear equations". *IBM J. Res. Develop.*, pp. 504-522, Sept., 1972.
[14] Easom, E. E., *A survey of global optimization techniques*. M. Eng. thesis, Univ. Louisville, Louisville, KY, 1990.
[15] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs, Second, Extended Edition*. Berlin, Heidelberg, New York: Springer-Verlag, 1994.
[16] Goldstein, A. A. and Price, I. F., "On descent from local minima". *Math. Comput.*, 1971, Vol. 25, No. 115.
[17] Syswerda, G., "Uniform crossover in genetic algorithms". *in [ICGA3]*, pp. 2-9, 1989.
[18] Pohlheim, H., *Genetic and Evolutionary Algorithm Toolbox for use with Matlab - Documentation*. Technical Report, Technical University Ilmenau, 1996.