Sindh Univ. Res. Jour. (Sci. Ser.) Vol.45 (2) 333-340 (2013)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

An Adaptive Crossover Operator for Genetic Algorithms to Solve the Optimization Problems

I.A. KOREJO++, Z.U.A. KHUHRO, F. A. JOKHIO, N. CHANNA*, H. A. NIZAMANI

Institute of Mathematics and Computer Science, University of Sindh, Jamshoro

Received 26th March 2013 and Revised 2nd June 2013

Abstract: Traditional genetic algorithms have been determined as an active approach for solving the optimization problems. Still now, most of GAs use a single crossover operator for all individuals. It means that all individuals in a population use the same mechanism due to this reason may be lack of intelligence for a particular individual; it is difficult to deal with complex situations. In adaptive scheme, single individual has four crossover operators to cope with different situations in the landscape. The adaptive algorithm can enable an individual to select the best operator according to its own local fitness landscape. The experimental results determine that our proposed technique is able to choose automatically the suitable crossover operator for different optimization problems.

Keywords: Genetic Algorithms, adaptive crossover operator and optimization problems.

1. INTRODUCTION

Based on the principle of natural evolution of species approaches researcher have developed various heuristic and optimization techniques, which are the Evolutionary Algorithms (EAs). There are various classes of evolutionary algorithms, Genetic algorithms are most popular and have been successfully applied to different optimization problems in many fields (Goldberg 1989). According to the Darwin's theory "Survival of the fittest", the best candidate solutions will be more likely survive and form a new population. There are three basic genetic operations of the GAs (Park 2009), called crossover, mutation and selection. The crossover operator is to swap the genetic information to certain probability between a pair of parents to generate a new offsprings. The mutation operator randomly alters gene value(s) in the solution to certain probability to create a new solution. Selection mechanism takes better chromosomes for the next population.

The traditional GAs uses only single crossover operator to produce offspring to form a next generation. Later on, Hong has proposed four different crossover operators (Hong 2000). Some times It is difficult to choose which crossover operator should be suitable for the performance of GAs. Few techniques have already been suggested in the literature (Hong 2000, Li 2008, Gong 2010, korejo 2012) to automatically select the most suitable operator from four available crossover operators. These approaches are employing four genetic operators for the better performance. Various crossover operators have different properties on different stages on evolutionary process of GAs. Therefore, it can be analysed that the work out of adaptive schemes in GAs is still limited.

In order to analyse the above drawback, this paper, proposes adaptive method in GAs, in which simultaneously apply more than one crossover operators in the genetic process. Operator is introduced for GAs. The objective of this method is to adjust probabilities of these operators according to the global behaviour of the whole population for each generation and achieve better individuals in the search space.

Experiments have been carried out on 15 most popular optimization benchmark problems. The experiments results validate that the suggested adaptive technique may effectively choose which operator should be mostly used at each level of the landscape. Proposed algorithm gives better performance of different crossover operators on different benchmark problems and it shows the best result on few problems.

The rest of this paper is summaried as follows. Section 2 provides a basic concept of GAs and adaptation respectively. In section 3, we discuss the design adaptive crossover operator for GAs. An experimental result is carried out to compare the performance of the analysed crossover operator for GAs on benchmark test problems in section 3. Finally, section 4 describes a conclusion.

2. MATERIAL AND METHODS

A. Genetic Algorithms

Genetic Algorithms (GAs) were first time introduced by John Holland in 1960s (Holland 19975),

++Corresponding author: email: cskorejo@gmail.com Cell. No. +92-3323605603

*Institute of Business Administration, University of Sindh, Jamshoro.

which were motivated from the Darwin's theory of survival of the fittest. In GAs, three genetic operations (selection, crossover and mutation) are used to create next generation. Conventional genetic algorithms apply one crossover and one mutation operator during the entire evolutionary process. The process is repeated until termination criterion is satisfied or a constant number of generations expires. The framework of simple genetic algorithms is given in algorithm 1. It uses various parameters like a population size, crossover probability, and mutation probability.

Algorithm 1 Simple Genetic Algorithm (SGA)

1. Initialize population with random individuals
2. Evaluate each individual
3. **While** (Termination criteria) is satisfied **do**
4. Select parents
5. Perform variation operations (Crossover and Mutation) to create offspring
6. Select the best solution according to their fitness function to form the next iteration.
7. **End While**

Adaptation in Crossover Operators.

Adaptation is increasing research area in the genetic algorithms. Several researchers have employed adaptation schemes to improve the GAs performance (Eiben 1999, Eiben 2007). Based on the parameter tuning, adaptation in GAs can be divided into three subclasses: deterministic adaptation where the value of parameter is modified according to predefined rules; adaptive adaptation where statistics information from the whole is applied to alter a parameter value, and self-adaptive adaptation where the parameter value is encoded into and co-evolves with individuals. Three levels of adaptation in crossover was suggested in (Yang 2002).

3. THE ADAPTIVE CROSSOVER

Few years ago, most of the researchers have used single crossover operator to produce offspring. Different crossover operators have different properties, behaviour and change search directions in the search space; due to this reason, GAs are enhancing the performance. The proposed approach to automatically select the most suitable crossover is required. There are two main objectives in this approach. First one is assigning progress values to operators according to the fitness improvements of the individuals, and second one is adjusting the operator's probabilities according to population feedback.

A. Credit Assignment

The relative fitness improvement δ_i is calculated as follows:

$$\delta_i = \frac{best}{cf_i}|pf_i - cf_i| \qquad (1)$$

where pf_i and cf_i represent a parent and its child produced by crossover operator, $best$ is the best fitness so for in the population, and i is the index of individual in the population. The offspring is worst or equal to its parent(in other words there is no any improvement between parent and child), in this situation a null credit is assigned. The formula of relative fitness improvement was proposed by (Ong 2004).

The reward value R_i of operator i is defined as follows:

$$R_i = exp\left(\frac{P_i}{\sum_{j=1}^{N} P_j}\alpha + \frac{s_i}{M_i}(1-\alpha)\right) + c_i p_i - 1, \quad (2)$$

where s_i is the total number of individuals which have better fitness value than themselves after performed crossover operator i. P_i is the selection ratio of operator i, α is random number between 0 and 1, N denotes total number of operators, and c_i represent a penalty factor for the crossover operator i, that is mentioned below:

$$c_i = \begin{cases} 0.9, & if\ s_i = 0\ and\ p_i = max_{j=1}^{N}(p_i) \\ 1, & otherwise \end{cases} \quad ,(3)$$

If there is no any contribution of the previous best crossover operator in the current iteration, then selection ratio of the current best operator will decline. The probabilities of crossover operator i is updated according to the following equation:

$$P_i(t+1) = \frac{R_i}{\sum_{j=1}^{N} R_j}(1 - N * \gamma) + \gamma \qquad (4)$$

where γ is the minimum probability value of each crossover operator, applied to confirm that no operator becomes zero.

B. Learning Strategy

Several researchers have introduced many schemes in genetic algorithms (Hong 2002, Otman

2011); in which four and five crossover operators are used. In order to organise the learning strategy employed in this work. Four crossover operators are chosen from (Herrera 2003), listed as follows:

1) Arithmetical Crossover (Xarith):

It is a linear combination of two solutions. Pair of individuals, chosen randomly for crossover, c_i^1 and c_j^2 may create two children,

$$G_k = (g_1^k, \dots g_i^k, \dots, g_n^k), k = 1, 2,$$
$$g_i^1 = \alpha . c_i^1 + (1 - \alpha) . c_j^2$$
$$g_j^2 = (1 - \alpha) c_i^1 + \alpha . c_j^2 \quad \text{where } \alpha \in [0,1].$$

2) Wright's Heuristic Crossover (ESH):

Suppose that the fitness of c_1 is improved than the one of c_2. Then, two children are formed,

$$G_k = (g_1^k, \dots g_i^k, \dots, g_n^k), k = 1, 2,$$

where $g_i^k = \omega . (c_i^1 - c_i^2) + c_i^1$ and ω is a random number between 0 and 1.

3) BLX-α-β (Xblx):

Two chromosomes are produced,

$$G_k = (g_1^k, \dots g_i^k, \dots, g_n^k), k = 1, 2,$$

where g_i^k is a randomly selected number within the interval $[C_{min} - I\gamma, C_{max} + I\gamma]$, where

$C_{max} = max\{c_i^1, c_i^2\}$,

$C_{min} = min\{c_i^1, c_i^2\}$, and

$I = C_{max} - C_{min}$

4) Simulated binary crossover (Xsb):

Two children are created,

$$G_k = (g_1^k, \dots g_i^k, \dots, g_n^k), k = 1, 2,$$

where

$$g_i^1 = \frac{1}{2}[(1 - \beta_k) . c_i^1 + (1 - \beta_k) . c_i^2]$$
$$g_i^2 = \frac{1}{2}[(1 - \beta_k) . c_i^1 + (1 - \beta_k) . c_i^2]$$

$\beta_k (\geq 0)$ is a sample form a random number generator having the density

$$\rho(\beta) = \begin{cases} \frac{1}{2} . (\eta + 1)\beta^{\eta}, & if\ 0 \leq \beta \leq 1 \\ \frac{1}{2} . (\eta + 1)\frac{1}{\beta^{\eta+2}}, & if\ \beta > 1 \end{cases}$$

This distribution can be obtained easily from a uniform random number source by the transformation

$$\beta(\mathcal{N}) = \begin{cases} (2\mathcal{N})^{\frac{1}{\eta+1}} & if\ \mathcal{N}(0,1) \leq \frac{1}{2} \\ [2(1 - \mathcal{N})]^{1/(\eta+1)} & if \mathcal{N}(0,1) > \frac{1}{2} \end{cases}$$

C. GA with Adaptive crossover Operator

For simplicity, we are designing adaptive crossover method that enhances GAs performance by adapting the probabilities of each operator according to the population feedback instead of just depend on a priori knowledge. The proposed algorithm uses four crossover operators, which are mentioned earlier. The initial ratio of each crossover operator is ¼. Each crossover operator is applied to its selection ratio. Ratio of these operators is adaptively adjusted regarding to the evaluation results of the respective children in the next population. However, different operators having different properties on different functions (Hong 2000). It is difficult to choose best crossover operator for the particular problem in hand. The most suitable crossover operator will be chosen automatically during the evolutionary process.

The framework of GA with adaptive crossover operator is illustrated in algorithm 2. Some steps are modified with respect to the simple GA algorithm, these steps are noticeable with arrow.

Algorithm 2 Genetic algorithm with Adaptive Crossover Operator.

1. Initialize population with random individuals
2. Evaluate each individual
   Set N = 4, and γ=0.01;
   For each crossover operator i, set R_i=0,
   P_i = 1/N
3. **While** (Termination criteria) is satisfied **do**
4. **for** (j:= 1 to popsize) **do**
5 choose individual j by the roulette wheel scheme.
6. select crossover operator i based on its probability.
7. Crossover individual i with solution j applying selected crossover operator i
8. Mutate the individual j by using Gaussian mutation operator
9. **endfor**
10. Calculate progress value δ_i using Eq: 1 for each operator
11. Compute the reward value R_i using Eq: 2 for each operator
12. Update the probability P_i using Eq: 4 for each operator
13. **End While**

4. EXPERIMENTAL STUDIES

In order to evaluate the performance of the adaptive algorithm as compared to that of four GAs algorithms applying crossover operators described in section 3.B i.e Xarith, ESH, Xblx,and Xsb. These five approaches are conducted on a standard set benchmark problem. The number of 15 test optimization problems are selected from the test suit (Yao 1999) and description of these problems are given in the **(Table 1).** In this work, selected test problems are widely used for the experiments (Dong 2007, Li 2008). Functions $f_{01} - f_{05}$ are uni-modal functions, single peak exist in their functions. Function f_{06} is the step function and their behaviour is discontinuous with one minimum. Function f_{07} is a noisy quartic function along with uniformly distributed random number between zero and one. The rest of the problems are having more than one peaks, where the number of local minimum increases exponentially with the problem dimension.

The general parameters are selected as follows: The population size is 100 and 300 generations are allowed. The initial selection ratio was 1/4 for each crossover operator and the minimum selection ratio was set to 0.001. The roulette-wheel selection is used as the selection method. Probabilities of crossover and mutation were set as suggested by (Korejo 2012). The initial selection ratio was 1/4 for each adaptive mutation operator and the minimum selection ratio was set to 0.001 for each adaptive mutation operator. Tournament selection with the tournament size 2, elitism size 1 and 2 point crossover were allowed to generate next population.

4.1 Experimental Results and Analysis

This section presents the average best fitness result over the 50 independent trails for each algorithm on the test problems. The results of **(Table 2)** show that average best fitness of each algorithm. Table 2 is a summary of the results achieved by five algorithms on the 15 optimization problems. The performance of these five algorithms is different on the various benchmark problems during the evolutionary process. Any bold average fitness value determines that the related approach gives the best result among the five algorithms. The adaptive crossover operator obtains the equal performance on few functions, which are indicated by the (*) symbol and gets a closer result to the ESH, Xsb, and Xarith on different optimization problems. It is interesting to note that the performance of adaptX algorithm is better than Xblx on all optimization problems. On function f_4, f_{10}, f_{11},f_{12}, f_{13}, and f_{14}, the efficiency of Xadapt is the second best among the other algorithms. Generally, the performance of ESH crossover operator is better than some algorithms on different problems.

Table 1: The detailed description of the benchmark functions, where n is the number of variables of a function, D $\in R_n$, f_{min} is the minimum value of function.

Test problems	n	D	f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	10	[-100,100]	0
$f_2(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	10	[-5.12,5.12]	0
$f_3 = \sum_{i=1}^{n} \left(\sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - n \sum_{k=0}^{k_{max}} [a^k \cos(\pi b^k)], a = 0.5, b = 3, k_{max} = 20$	10	[-0.5,0.5]	0
$f_4(x) = \frac{1}{4000} \sum_{i=1}^{n} (x_i - 100)2 - \prod_{i=1}^{n} \cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$	10	[-600, 600]	0
$f_5(x) = -20\, exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{2} \sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$	10	[-32, 32]	0
$f_6(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)2$	10	[-100,100]	0
$f_7(x) = \sum_{i=1}^{n} i x_i^4 + \cup(0,1)$	10	[-1.28, 1.28]	0
$f_8(x) = \sum_{i=1}^{n} 100(x_{i+1}^2 - x_i)2 + (x_i - 1)2$	10	[-30, 30]	0

$f_9(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	10	[-500, 500]	-4189
$f_{10}(x) = 418.9829.n + \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	10	[-500, 500]	0
$f_{11}(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	10	[-10, 10]	0
$f_{12}(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right) 2$	10	[-100,100]	0
$f_{13}(x) = max_{i=1}^{n} \lvert x_i \rvert$	10	[-100, 100]	0
$f_{14}(x) = \frac{\pi}{30}\left\{10\, sin^2(y_1 - 1) + \sum_{i=1}^{n-1} (y_i - 1)^2.[1 + 10 sin^2(y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^{n} \mu(x_i, 5, 100, 4), y_i = 1 + (x_i + 1)/4$	10	[-50, 50]	0
$f_{15}(x) = 0.1\left\{10 sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2.[1 + sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + sin^2(2\pi x_n)] \right\} + \sum_{i=1}^{n} \mu(x_i, 5, 100, 4)$	10	[-50, 50]	0

Table 2: Comparison on average best fitness value of different crossover operators on some functions.

Problems	Xadapt	ESH	Xsb	Xblx	Xarith
f_1	7.30631e-029	**7.31539e-083**	2.56073e-029	7.94705e-006	2.93677e-007
f_2	0*	0*	0	0.00389068	0.000899863
f_3	0*	0*	0.00069259	0.029218	0.126311
f_4	0.0924502	**0.0320687**	0.132714	0.645036	0.102135
f_5	4.44089e-016*	4.44089e-016*	3.87838e-015	0.644755	0.00192361
f_6	0*	0*	0*	0*	0*
f_7	0.000466229	0.000386919	0.00237767	0.0110921	**0.000373741**
f_8	**0.00518182**	0.00550568	0.00550568	0.01162	0.00777275
f_9	5.67029	5.24156	**2.99011**	96.5953	5.29677
f_{10}	-3195.4	**-3616.21**	-3026.2	-2529.78	-2665.83
f_{11}	995.796	**582.378**	1225.58	1625.04	1512.85
f_{12}	7.45732e-043	**9.41406e-104**	8.69141e-040	2.1549e-005	0.00245718
f_{13}	2.23248e-024	2.19426e-010	**1.87708e-033**	0.000346266	0.266158
f_{14}	4.54648e-006	0.000644446	**1.48627e-013**	1.52896	0.0108182
f_{15}	1.57054e-032*	1.57054e-032*	0.00345563	0.014466	2.36846e-008

The experimental results are shown in the **(Fig. 1)**. Xadapt crossover operator become a second or third best operator in the rest of crossover operators on different optimization functions. From Fig. 1, it can be seen that the performance of various crossover operators are different on benchmark optimization problems. It can also be shown that the convergence speed of Xadapt is equal on first row of the graph and second fast convergence rate of Xadapt on second row of graph.

In summary, from the above analysis it is to be concluded that the performance of five mutation operators is reasonably fine except SANUM. The proposed mutation algorithm is better than the other mutation algorithms on few problems.

4. CONCLUSIONS

Genetic Algorithms(GAs) have become important and promising filed for the researchers in solving the optimization problems. Generally, the conventional GAs solve these optimization problems but GAs have some drawbacks such as premature convergence and poor stability.

We have proposed an adaptive crossover for genetic algorithms to automatically apply suitable crossover operator during the evolutionary process. However; Different crossover operators provide different efficiency on the global optimization problems. The main objective of the proposed algorithm is to modify the crossover operator's probabilities based on the global behaviour of the population for the next generation. The experimental results show that adaptX gives the balance performance on the all test functions. Hence, the incorporation of adaptx operator is an encouraging work for enhancing the performance of GAs.

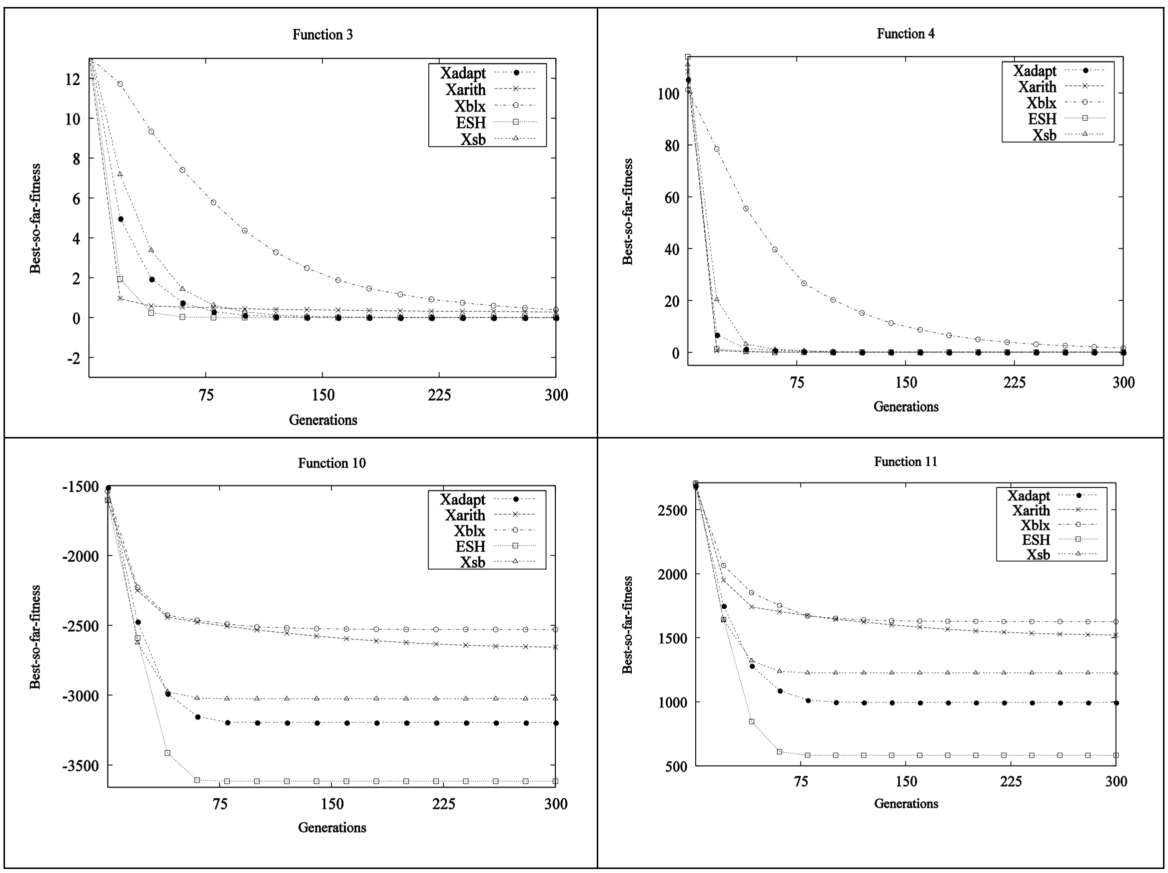

Fig. 1: Comparison of Gas with different crossover operator on selected benchmark problems.

REFERENCES:

Eiben, A. E., Z. Michalewicz,, M. Schoenauer, and J. E. Smith (2007) Parameter control in evolutionary algorithms. In Lobo, F.G. editor, *Parameter Setting in Evolutionary algorithms,* 19-46. Springer.

Eiben, A. E., R. Hinterding and Z. Michalewicz (1999) Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 3 **(2):** 124–141.

Goldberg, D. E. (1989) Genetic Algorithms in earch Optimization and Machine Learning. Addition-Wesley longman Publishing Co., Inc. Boston, MA, USA.

Gong. W, (2010) Adaptive strategy selection in differential evolution. *Proc: of the 12th conference on genetic and evolutionary computation* 409-416. Portland, Oregan, USA.

Herrera, F., M. Lozano, A. M. Sanchez, (2003) Taxonomy for the Crossover Operator for Real-Coded Genetic Algorithms: An Experimental Study. *International Journal of Intelligent Systems*, **(18)**: 309-338.

Hong, T. P., H. S. Wang, and W.C. Chen (2000) Simultaneously applying multiple mutation operators in genetic algorithms. *Journal of Heuristics*, **(6):** 439-455.

Korejo, I. A., K. Brohi, and M. S. Vighio, (2012) An adaptive mutation operator for global optimization problems. *(Sindh University Research Journal (Science Serices))* 44 **(2):** 291-294.

Li, C., S. Yang, and I. Korejo (2008) An adaptive mutation operator for particle swarm optimization. (*In Proceedings of the (2008) UK Workshop on Computational Intelligence.)* 165-170. Leicestor, UK.

Ong, Y. S., and A. J. Keane, (2004) Meta-Lamarckian learning in memetic algorithms. (*IEEE Trans. On Evol. Comput).* 8 **(2):** 99-110.

Otman, A. and A. Jaafar, (2011). A comparative study of Adaptive Crossover Operators for Genetic Algorithms to Resolve the Traveling Salesman Problem, *(International Journal of Computer Applications)*, 3, **(11)**:49-57.

Park. S. J., P. Hwang. and P. Kim. (2009) A new evotutionary particle filter for the prevention of sample improvement. (IEEE Trans. Evolu., Comput), 13, **(4)**:801-809.

Yao, X., Y. Liu, and G. Lin, (1999) Evolutionary programming made faster, IEEE Trans. Evolutionary Computation 3 **(2)**: 82–102.

Yang, S. (2002) Adaptive non-uniform crossover based on statistics for genetic algorithms. In W. B. Langdon, *et al,* editors, Proc. Of the Genetic and Evolutionary Computation Conference, GECCO, 650-657. San Francisco, CA 94104, USA.