in Artificial Life VIII, *Standish, Abbass, Bedau (eds)(MIT Press) 2002. pp 182–185*

Adaptive Crossover in Genetic Algorithms Using Statistics Mechanism

Shengxiang Yang

Department of Mathematics and Computer Science, University of Leicester
University Road, Leicester LE1 7RH, UK. Email: s.yang@mcs.le.ac.uk

Abstract

Genetic Algorithms (GAs) emulate the natural evolution process and maintain a population of potential solutions to a given problem. Through the population, GAs implicitly maintain the statistics about the search space. This implicit statistics can be used explicitly to enhance GA's performance. Inspired by this idea, a statistics-based adaptive non-uniform crossover (SANUX) has been proposed. SANUX uses the statistics information of the alleles in each locus to adaptively calculate the swapping probability of that locus for crossover operation. A simple triangular function has been used to calculate the swapping probability. In this paper two new functions, the trapezoid and exponential functions, are proposed for SANUX instead of the triangular function. Experiment results show that both functions further improve the performance of SANUX.

Introduction

Genetic Algorithms (GAs) emulate the natural evolution process and maintain a population of potential solutions to a given problem, which are evaluated by a problem-specific fitness function. This population is evolved by randomly selecting relatively fit members and performing genetic operations, especially recombination and mutation, on them to generate a new population (Goldberg 1989). With the progress of the GA, the average fitness of the population increases, hopefully leading to the optimal solution(s) to the problem. Through the population GAs implicitly maintain statistics about the search space. That is, useful materials or building blocks permeate in the population. GAs use the selection, crossover and mutation operators to explicitly extract the implicit statistics from the population to reach the next set of points in the search space.

In fact, this implicit statistics in the population can be used explicitly to enhance GA's performance. Inspired by this idea, a statistics-based adaptive non-uniform crossover, called SANUX, has been proposed by Yang (2002). SANUX explicitly uses the statistics information of the alleles (values of a gene) in each locus to adaptively calculate the swapping probability of that locus for crossover operation. In Yang (2002) a simple triangular function has been used to calculate the swapping probability from the statistics of allele distribution. In this paper, two new functions, a trapezoid function and an exponential function, are proposed for SANUX instead of the simple triangular function. These two functions are proposed with the aim of using the statistics more efficiently with respect to construction and protection of useful building blocks with the progress of the GA.

Related Work on Crossover Operators

Traditionally, GAs have used one-point or two-point crossover (De Jong 1975; Holland 1975). Researchers have also carried out experiments with multi-point crossover: n-point crossover (Eshelman, Caruana, & Schaffer 1989) and uniform crossover (Syswerda 1989). With the n-point crossover, n cut points are randomly chosen within the strings and the $n-1$ segments between the n cut points of the two parents are exchanged. Uniform crossover is the generalization of n-point crossover. It creates offsprings by swapping each bit of two parents with a probability p_s of 0.5. Spears and De Jong (1991) proposed the parameterized uniform crossover where the decision of whether to swap for each locus is made by a biased coin flipping, i.e., the swapping probability p_s could be other than 0.5.

Traditional crossover operators are inherently all based on uniformly randomization mechanism, i.e., generating cut points (n-point crossover) or swapping points (uniform crossover) uniformly randomly across the chromosome. This situation is not very true with natural evolution because it is intrinsically dynamic and adaptive. Recently, researchers have applied adaptation techniques to crossover to enhance GA's capabilities (Eiben, Hinterding, & Michalewicz 1999). According to Yang (2002), adaptation in crossover happens in three levels from top to bottom. In the top level, crossover operators are themself adapted during a run of the GA. In the medium level, the rate or probability of crossover is altered during a run of the GA. In the bottom level, the position of crossing or swapping probability in each locus is adapted during a run of the GA.

Statistics-based Adaptive Non-Uniform Crossover

Description of SANUX

For the convenience of description and analysis, Yang (2002) has introduced the concepts of intrinsic attribute and extrinsic tendency of allele valuing for a gene locus. In the optimal binary-encoded solution(s) of a given problem, a gene locus is called *1-intrinsic* if its allele is 1, *0-intrinsic* if its allele is 0, or *neutral* if its allele can be either 0 or 1. During the running of a GA, for a gene locus, if the frequency of 1's in its alleles over the population tends to increase (to the limit of 1.0) with time, it is called *1-inclined*; if the frequency of 1's tends to decrease (to the limit of 0.0), it is called *0-inclined*; otherwise, if there is no tendency of increasing or decreasing, it is called *non-inclined*.

Usually and hopefully as the GA progresses, those gene loci that are 1-intrinsic (or 0-intrinsic) will appear to be 1-inclined (or 0-inclined), i.e., the frequency of 1's in the alleles of these loci will eventually converge to 1 (or 0). This convergent information is implicit in the population. Traditional crossover operators make no use of this implicit information and consistently generate cut points or swapping points randomly but uniformly across the chromosome. Obviously it will be beneficial to link these partially or fully converged genes into groups or building blocks and co-evolve them during crossover operations. This is realized in SANUX. SANUX explicitly uses the convergence information as feedback to adaptively adjust the swapping probability for each locus based on the statistics of alleles in that locus.

We use the frequency of 1's in the alleles in a locus over the population (equivalently we can also use the frequency of 0's as the argument) to calculate corresponding swapping probability of that locus. The frequency of 1's in the alleles of a locus can be looked as the degree of convergence to "1" for that locus. Let L be the length of binary strings, $f_1(i,t)$ $(i = 1,\ldots,L)$ denote the frequency of 1's in the alleles in locus i over the population at time (generation) t and $p_s(i,t)$ $(i = 1,\ldots,L)$ denote the swapping probability of locus i at time t. In Yang (2002), a simple triangular function is used to calculate $p_s(i,t)$ from $f_1(i,t)$, defined as follows:

$$p_s(i,t) = P_{\max} - 2|f_1(i,t) - 0.5|(P_{\max} - P_{\min}) \qquad (1)$$

where $|.|$ is an absolute function, $P_{\max}$ and $P_{\min}$ are the maximum and minimum allowable swapping probabilities for a locus respectively.

Now during the evolution of the GA, after a new population has been created, we first calculate $f_1(i,t)$ for each locus i over population t and from this obtain $p_s(i,t)$ of locus i. Then we can perform SANUX operations similarly as traditional parameterized uniform crossover, as illustrated in Figure 1.

1's Freq. in loci:	0.9	0.2	0.6	0.9	0.4	0.2
Calculating:	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$
Swapping Prob.:	0.1	0.2	0.4	0.1	0.4	0.2
Biased Flipping:	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$
Created Mask:	0	0	1	0	1	0
Applying Mask:			$\Downarrow$		$\Downarrow$	
Parent P_1:	1	1	0	1	0	1
Parent P_2:	1	1	1	1	1	0
Swapping:			$\Downarrow$		$\Downarrow$	
Child C_1:	1	1	1	1	1	1
Child C_2:	1	1	0	1	0	0

Figure 1: An example operation of SANUX with triangular calculation function where $P_{\max} = 0.5$ and $P_{\min} = 0$.

From above descriptions, it can be seen that SANUX is very simple. With SANUX, what we add to traditional uniform crossover are spatially only one real vector that records the swapping probability for each locus, and computationally only one statistics per generation that calculates the frequency of ones (hence the swapping probability) for each locus. Another more important point of SANUX is its property of implicit *gene linkage*. Gene linkage is the property of grouping interactive genes to evolve them together. For example, in Figure 1 loci 1 and 4 are more convergent and implicitly linked though they are not adjacent because the probability for them to co-evolve via crossover is $0.1 * 0.1 + 0.9 * 0.9 = 0.82$, which is very high.

New Calculating Functions

From above discussions, we can see that there is a close loop control in SANUX in the sense of constructing and preserving building blocks. The GA is first used to construct useful building blocks. Then after some time when some useful building blocks have been built up the statistics information is used to protect these building blocks from being destroyed by the crossover and hence divert the crossover to search unconverged loci for more unknown useful building blocks because they have relatively higher swapping probabilities. With this understanding, we can further enhance this close loop by introducing new functions instead of the triangular function to calculate the swapping probability of each locus from its allele distribution statistics.

The first function is a trapezoid function, as shown in Figure 2, defined by the following equation.

$$p_s(i,t) = \begin{cases} P_{\max}, & \text{if } |f_1(i,t) - 0.5| \le a \\ \frac{b-|f_1(i,t)-0.5|}{b-a} * (P_{\max} - P_{\min}) + P_{\min}, & \\ \qquad \text{if } a < |f_1(i,t) - 0.5| < b & \\ P_{\min}, & \text{if } |f_1(i,t) - 0.5| \ge b \end{cases} \qquad (2)$$

where a and b are parameters that satisfy the relation of $0 \le a \le b \le 0.5$. As shown in Figure 2, the parameters a and b split the whole range of $f_1(i,t)$ into three zones: I) $|f_1(i,t) - 0.5| \le a$; II) $a < |f_1(i,t) - 0.5| < b$; III) $|f_1(i,t) - 0.5| \ge b$. Zone I is called *constructive zone* where useful building blocks are mainly constructed by SANUX. Zone II is called *transient zone* where building blocks are constructed as well as destroyed. And Zone III is called *protective zone* where building blocks found so far are protected from being destroyed by SANUX while exploring unknown useful building blocks is still carried out by SANUX on unconverged loci. From Figure 2 it can be seen that the larger the value of a, the wider the region that allows the GA to construct building blocks, and that the smaller the value of b, the stronger the protection to converged genes or found building blocks from being destroyed by SANUX.

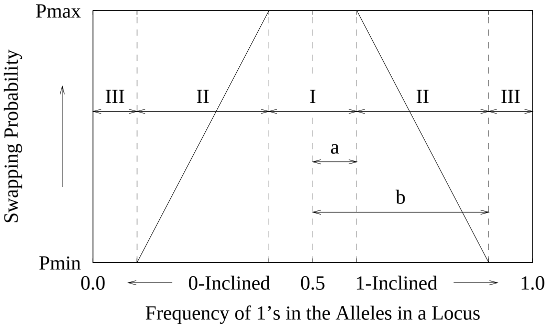

Figure 2: Trapezoid function for calculating the swapping probability of a locus.

The second function we introduce here is an exponential function, as shown in Figure 3, defined as follows:

$$p_s(i,t) = \alpha * \exp(-(f_1(i,t) - 0.5)^2/\beta) \tag{3}$$

where α is the parameter with the same functionality as $P_{\max}$ in Equations (1) and (2), and β is the parameter that combines the functionalities of parameters $P_{\min}$, a and b in Equation (2) and controls the falling speed of $p_s(i,t)$ when $f_1(i,t)$ diverts away from 0.5. The smaller the value of β, the faster the falling speed of $p_s(i,t)$, and the stronger the protection to converged genes or building blocks found so far. As shown in Figure 3, the solid curve shows the function with $(\alpha, \beta) = (0.5, 0.04)$ while the dashed curve is when $(\alpha, \beta) = (0.5, 0.02)$.

Both the two functions enlarge the region for the GA to construct building blocks and enhance the protection to found building blocks at the same time. Both sides are expected to be beneficial for SANUX's performance.

Experimental Study

The Test Problems

The Max Ones Problem The Max Ones problem is simply to maximize 1's in a binary string. The fitness of a string is the number of 1's it contains. A string length of 100 bits is used for this study.

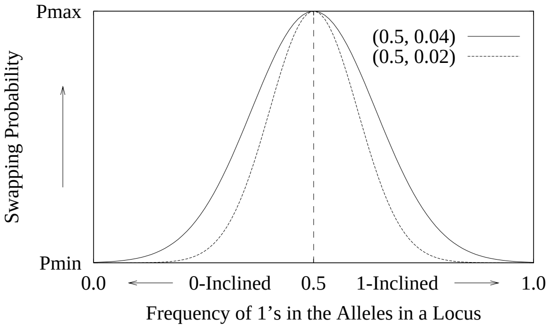

Figure 3: Exponential function for calculating the swapping probability of a locus.

Royal Road Functions The Royal Road functions R_1 and R_2 are devised by Forrest and Mitchell (1993) to investigate GA's performance with respect to schema processing and recombination in an idealized form. The optimal fitness for R_1 and R_2 is 64 and 192 respectively.

Experiment Results

To test the performance of proposed two calculating functions for SANUX, in our experiment study we compared the SANUX with the trapezoid and exponential functions over SANUX with the triangular function and traditional 2-point, 0.5 uniform and 0.2 uniform crossover on the test problems. The parameters for different functions of SANUX were set as follows: $P_{\max} = 0.5$ and $P_{\min} = 0.0$ for triangular and trapezoid functions, $(a, b) = (0.1, 0.4)$ for trapezoid function, and $(\alpha, \beta) = (0.5, 0.04)$ for exponential function.

For each experiment of combining different crossover and test problems, 100 independent runs were executed under the same 100 different random seeds. In all the experiments, the GA used the fitness proportionate selection with the stochastic universal sampling (SUS) and elitist model, and bit flip mutation. And typically the probabilities of crossover and mutation were fixed to 0.6 and 0.001 respectively and the population size was set to 100 for each run. For each run, we recorded the best-so-far fitness every 100 evaluations. Here, only those chromosomes changed by crossover and mutation operations were evaluated and counted into the number of evaluations. Each experiment result was averaged over the 100 independent runs. The experiment results are shown in Figure 4 and Figure 5 respectively.

From Figure 4 and Figure 5 it can be seen that SANUX with different calculating functions has outperformed traditional crossover operators on the test problems. On the test problems lowering the swapping probability p_s from 0.5 to 0.2 improves the performance of uniform crossover (0.2 uniform crossover outperforms 2-

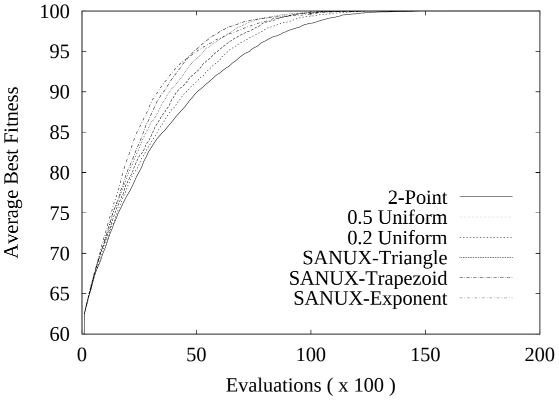

Figure 4: Average best-so-far curves for GAs with different crossover operators on Max Ones problem.

point and 0.5 uniform crossover), however 0.2 uniform crossover is beaten by SANUX. The relative performance of different crossover seems quite consistent on the test problems, approximately from best to worst in the following order: SANUX-Exponent ⇒ SANUX-Trapezoid ⇒ SANUX-Triangle ⇒ 0.2 uniform ⇒ 0.5 uniform ⇒ 2-point crossover. Within SANUX both the trapezoid and exponential functions have outperformed the triangular function. This confirms our prediction.

Conclusions

In this paper, we propose two new functions, the trapezoid and exponential functions, that can be used within SANUX instead of the triangular function (Yang 2002) to calculate the swapping probability for a locus from its allele distribution statistics. Both these two functions widen the region for the GA to construct building blocks and enhance the protection to searched building blocks at the same time, hence are expected to be beneficial for the performance of GAs.

The experiment results show that SANUX with the trapezoid and exponential functions performs better than SANUX with the triangular function and traditional two-point and uniform crossover (parameterized or not) on the test problems. Our experiment results indicate that the exponential function is a good choice to be used in SANUX for its performance and simplicity, and that SANUX with the exponential calculating function is a good candidate crossover operator for GAs.

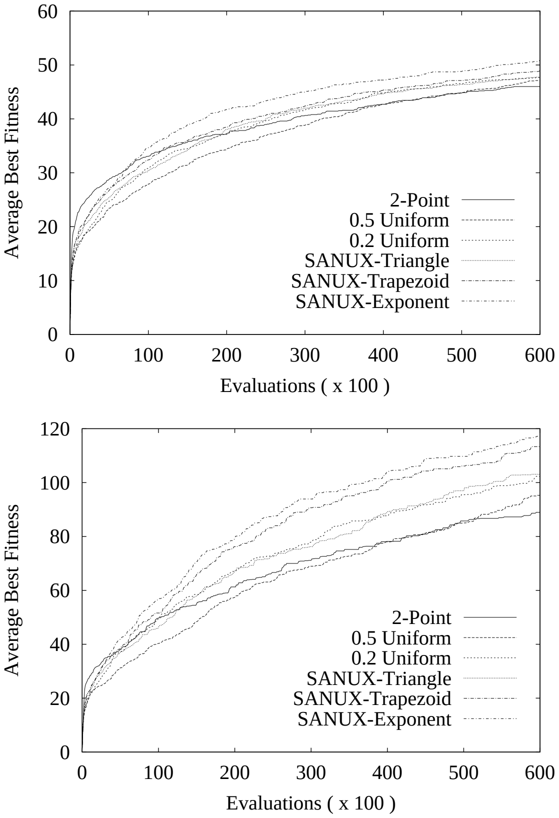

Figure 5: Average best-so-far curves for GAs with different crossover operators on (Top) R_1 and (Bottom) R_2.

References

De Jong, K. A. 1975. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD Thesis, University of Michigan, Ann Abor.

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

Eshelman, L. J.; Caruana, R. A.; and Schaffer J. D. 1989. Biases in the crossover landscape. In Schaffer, D., ed., *Proc. of the 3rd Int. Conf. on Genetic Algorithms*, 10-19. Morgan Kaufmann.

Forrest, S., and Mitchell, M. 1993. Relative building-block fitness and the building-block hypothesis. In Whitley, D., ed., *Foundations of Genetic Algorithms 2.* Morgan Kaufmann.

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

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: University of Michigan Press.

Spears, W. M., and De Jong, K. A. 1991. On the virtues of parameterized uniform crossover. In Belew, R. K., and Booker, L. B., eds., *Proc. 4th Int. Conf. on Genetic Algorithms*, 230-236. Morgan Kaufmann.

Syswerda, G. 1989. Uniform crossover in genetic algorithms. In Schaffer, J. D., ed., *Proc. of the 3rd Int. Conf. on Genetic Algorithms*, 2-9. Morgan Kaufmann.

Yang, S. 2002. Adaptive non-uniform crossover based on statistics for genetic algorithms. In Langdon, W. B., *et al*, eds., *Proc. of the Genetic and Evolutionary Computation Conference, GECCO 2002*, 650-657.

Morgan Kaufmann.