Crossover in GA : Some Ideas from Marriages in Human Society

3G−4

Goutam Chakraborty
Iwate Prefectural University

ABSTRACT

While searching for optimum solution using genetic algorithm, it is always critical to properly balance between adequate exploration of the search space during earlier generations, and putting required selective pressure to find the optimum during later generations. When to initiate this change from more explorative search to close selective search should again depend on the problem and preferably done adaptively. Common approaches to solve this are fitness scaling, ranking of the chromosomes, tournament selection etc. They balance exploration vs. exploitation during the process of genetic selection. Other proposals are to adaptively change the probabilities of crossover and mutation operations as genetic search progresses. Our proposal is to tune this by restricting the choice of partners for crossover over generations. In real life marriages(crossovers) occur between two individuals of similar status in society, only when they are mature in age and usually from neighboring localities. Similar principle is extended in selecting cross-over partners in GA. In the proposed strategy, the probability of cross-over is more when their rank in the whole population is close and their age matured. At early stage restricting crossover to chromosomes of similar rank would lead to bad exploration. The probability function for selecting partners depending on their ranks changes with advancing generations, so that the effect is negligible in the beginning. At later stage the effect is accentuated so as to be able to fine tune good chromosomes to achieve fast convergence and reach optimum values. The scheme is not centralized like elitist approach. The case of effecting crossover only to partners of matured age were separately studied. This is to disrupt chromosomes which are copied to successive generations. The effectiveness of this new method is tried on problems of maximizing complex multimodal functions. The results are compared with standard genetic algorithm (SGA) and SGA with "linear fitness scaling". Results show that our strategy is superior in terms of probability of hitting the maximum value as well as speed of finding the maximum.

1. INTRODUCTION

Genetic Algorithm (GA) is a search algorithm based on the mechanics of natural selection [1]. Compared to other approaches, they are superior because, (1) of wide applicability and make few assumptions from the problem domain, (2) and are not biased towards local minimums.

The steps of standard genetic algorithm (SGA) are as shown in Fig. 1. We need to encode the solution of the problem in binary bit string and generate a pool of solutions of the problem to form the initial population. A fitness function has to be defined to measure the goodness of these encoded solutions. Then genetic operators *selection*, *crossover* and *mutation* operate on the population to generate new population from the old ones. Good solutions are *selected* with greater probability to next generation, in line with the idea of *survival of the fittest*. *Crossover* operation recombines selected solutions, by swapping part of them, producing divergent solutions to explore the search space. An occasional *mutation* is done on a chromosome by flipping a bit at random position of the encoded chromosome, to facilitate jumping of solutions to new unexplored regions of the search space. As newer and newer generations evolve, the quality of solutions improve.

Algorithm **SGA** $(g, G, \Pi(g), P)$
begin
  $g = 0$; /* g: generation number */
  Create P members of the initial population $\Pi(0)$;
    /* P: population size,*/
    /* $\Pi(g)$: set of members at generation g */
  Evaluate members of $\Pi(0)$;
    /* Calculate fitnesses f_is of all members m_is */
  while ($g \leq G$) /* G: maximum generation */
    $g := g + 1$;
    $\Pi''(g) \xleftarrow{selection} \Pi(g-1)$;
    $\Pi'(g) \xleftarrow{crossover} \Pi''(g)$;
    $\Pi(g) \xleftarrow{mutation} \Pi'(g)$;
  endwhile
end

Figure 1: Algorithm for Standard Genetic Algorithm

The success of genetic search depends on balancing the two aspects of (1) population diversity i.e. exploring the different regions of the search space, and (2) selective pressure i.e. to get to the exact optimum point. The existing approaches are:

1. To control selection by controlling the differences of fitnesses e.g. fitness scaling, ranking.
2. To control selection by novel selection method e.g. tournament selection.
3. By adaptively changing the crossover and mutation probability, and sometimes positions where mutation should occur.

Our proposed strategy is to make some reservation while effecting crossover between two chromosomes. We here propose two new crossover strategies, (1) rank based

Faculty of Software & Information Science, Iwate Prefectural University
152-52, Sugo, Takizawa, Iwate, 020-0193, Japan
E-mail: goutam@soft.iwate-pu.ac.jp

crossover(RBC) and (2) age based crossover (ABC). In the proposed RBC strategy, though two chromosomes are randomly chosen for crossover, the probability that the crossover actually takes place depends on how near they are in the fitness scale. The nearer they are, more is the probability that they will be crossed over. Initially this proximity range covers all the chromosomes (the whole of the fitness scale) so that all randomly chosen pairs are crossed over. This is to allow exploration of different regions of the search space. Slowly when good regions are discovered, we allow crossover between chromosomes of similar rank only. Then good solutions recombine fast to find the optimum location, somewhat like elitism. In elitism all along only the top 10-20% performing individuals are permitted to mate. This obviously leads to premature convergence, whereas in our approach this is the case only at the later generations. This allows fast convergence to optimum location. The probability function changes with generations.

In Age Based Crossover (ABC) strategy, after two chromosomes are randomly selected for crossover, the actual crossover depends probabilistically on the their ages - the older they are the more is the probability that they will be crossed over. There are already some proposals to restrict the life time of individuals to one or two generations only, regardless of their fitnesses. This is done at the selection stage - old chromosomes are not allowed to reproduce i.e. go to the next generation. We on the other hand indulge in old individuals, which due to high fitness propagates through generations, be allowed to go to the next generation but help in exploration by crossover.

2. RANK BASED CROSSOVER

The algorithm of SGA is described in the Introduction. Now the crossover part is elaborated in Fig. 2.

Algorithm **RBC** ($\Pi''(g), p_c, P$)
01 **while** ($no\,of\,crossovers \leq \lceil (p_c \times P)/2 \rceil$)
02 Select randomly a pair $m_i(g)$, $m_j(g)$ from $\Pi''(g)$;
/* where $\Pi''(g)$ is the population after selection */
03 **if** ($\varphi\,(f_i^{nor}(g) - f_j^{nor}(g)) > rand(0,1)$)
/* $f_i^{nor}(g), f_j^{nor}(g)$ are normalized */
/* fitnesses of $m_i(g), m_j(g)$ */
04 Perform crossover between $m_i(g)$ and $m_j(g)$;
05 **endif**
06 **endwhile**

Figure 2: Algorithm for rank based crossover

The basic steps of RBC strategy for crossover is shown above. First the fitness of all members in $\Pi''(g)$ are normalized. We need to know only the values for $f^{min}(g)$ and $f^{max}(g)$ for finding this normalized fitnesses. As shown in line 01 of Fig. 2, until enough number of crossover is done, two members from $\Pi''(g)$ are selected at random. The φ function mentioned in line 3 is the bell shaped function, and its value is high when the normalized fitness of the two selected chromosomes are close. The variance of φ is initially very high, allowing any two chromosomes to crossover. It is slowly reduced with increasing generation (g). Different functions to control this reduction were experimented.

3. AGE BASED CROSSOVER

The algorithm of SGA is described in the Introduction. Following psudocode, detailing the crossover part, presents the working principle of ABC.

Algorithm **ABC** ($\Pi''(g), p_c, P$)
01 **while** ($no\,of\,crossovers \leq \lceil (p_c \times P)/2 \rceil$)
02 Select randomly a pair $m_i(g)$, $m_j(g)$ from $\Pi''(g)$;
/* where $\Pi''(g)$ is the population after selection */
03 **if** ($(rand(0,1) \times (\alpha_2/\alpha_P)) > 0.5$)
/* α_P is the av. age of whole population */
/* α_2 is the av. age of $m_i(g)$ and $m_j(g)$ */
04 Perform crossover between $m_i(g)$ and $m_j(g)$;
05 **endif**
06 **endwhile**

Figure 3: Algorithm for Age Based Crossover

Here first the average age of all members in $\Pi''(g)$ is found. It is denoted as α_P in Fig. 4. The average age of the two chromosomes, $m_i(g)$ and $m_j(g)$, randomly selected from $\Pi''(g)$ to be crossed over is α_2. $rand(0,1)$ returns a random fraction between 0 and 1 with uniform probability distribution. When the age of $m_i(g)$ and $m_j(g)$ are greater than the average, they crossover with higher probability. Thus on an average old chromosomes are disrupted more often than fresh chromosomes.

5. RESULTS & DISCUSSIONS

New reservation methods while crossing over two chromosomes are proposed to increase selection pressure and improve exploration of the search space. In RBC strategy the the exploration during the initial generation is unaffected, but high selection pressure is enforced during later stage of generations. In ABC, crossover partners randomly selected from population are allowed to perform crossover with greater probability when their average age is more than the population average. This facilitates better and faster exploration of the search space.

It should be emphasized that the proposed RBC and ABC strategies could be used together and in fact in conjunction with other other approaches to achieve similar goals. This is because the existing approaches try to deal with this problem at the selection stage and sometimes by manipulating fitnesses. It would be interesting to perform experiments by using them together. These experiments we are planning to do in near future.

In case of RBC, depending on the nature of the search space, there would be an optimum choice for α. It is evident that for complex search problems, a higher value of $\alpha > 1.0$ is a better choice for better exploration in the beginning. On the other hand, for simple unimodal problems, $\alpha < 1$ should work faster. The best would be to to adaptively set the value of α with growing generations, by observing the mode of fitness change for the whole population. Also it would be an interesting exercise to combine RBC and ABC by a unique probability function.

References

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