A fitness based scanning multi-parent crossover operator using a probabilistic selection

Surapong Auwatanamongkol

บทคัดย่อ

ในงานวิจัยที่ผ่านมาได้มีผู้เสนอการผสมหลายพันธุ์และแม่พันธุ์ในขั้นตอนวิธีเชิงพันธุกรรม ทั้งนี้เพื่อเพิ่มประสิทธิภาพของการค้นหาคำตอบ การใช้หลายพันธุ์แม่พันธุ์ในเวลาเดียวกันจะต้องคำนึงถึงเป้าหมาย 2 อย่าง คือประสิทธิภาพในด้านความเร็วในการค้นหาคำตอบ และคุณภาพของคำตอบที่ได้ ถ้าพยายามมุ่งที่เป้าหมายแรกมากเกินไปจะนำไปสู่การค้นหาที่เร็วกว่าก็ได้ไม่ได้ และโดยทั่วไปถ้าพยายามมุ่งที่เป้าหมายที่สองมากเกินไปจะนำไปสู่การค้นหาที่ช้า และใช้เวลานาน ในบทความนี้จะนำเอาวิธีการผสมหลายพันธุ์แม่พันธุ์ ที่ใช้คำความเหมาะสมของพันธุ์แม่พันธุ์ในการเลือกตัวผู้พันธุ์แม่พันธุ์ที่จะกลายทอดไปสู่ลูก เพื่อให้มีคำความเหมาะสมที่ดียิ่งขึ้น วิธีการใหม่จะมีการเข้าชมที่หลากหลายกว่า 2 เป้าหมาย ด้วยการเลือกอินเซอร์ที่ตั้งอยู่ด้วยความน่าจะเป็นที่มีค่าพันธุ์ขึ้นเรื่อย ๆ วิธีการนี้จะช่วยค้นหาคำตอบที่มีความหลากหลายในช่วงต้นของการค้นหาและปรับให้การค้นหาความรวดเร็วขึ้นในช่วงปลายของการค้นหา จากการทดลองใช้วิธีการใหม่กับการค้นหาค่าแปรที่นำไปให้พันธุ์ขึ้นทดแทนยีนที่สุด พบว่าวิธีการใหม่นี้จะทำให้ได้ค่าคำตอบของค่าแปรที่ดีขึ้น

Surapong เฮ่อวัฒนามงคล เป็นอาจารย์ประจำคณะสถิติประยุกต์ สถาบันบัณฑิตพัฒนบริหารศาสตร์ (e-mail: surapong@as.nida.ac.th).
Abstract

Several multi-parent crossover operators had been proposed to increase performance of genetic algorithms. The operators allow several parents to simultaneously take part in creating offspring. The operators need to balance between the two conflicting goals, exploitation and exploration. Strong exploitation allows fast convergence to succeed but can lead to premature convergence while strong exploration can lead to better solution quality but slower convergence. This paper proposes a new fitness based scanning multi-parent crossover operator for genetic algorithms. The new operator looks for the optimal setting for the two goals to achieve the highest benefits from both. The operator uses a probabilistic selection with an incremental threshold value to allow strong exploration at early stages of the algorithms and strong exploitation at late stages of the algorithms. The experiments conducted on some testing functions show that the operator can give better solution quality and faster convergence when compared with some well-known multi-parent crossover operators.

Keywords: Multi-Parent Crossover, Fitness Based Scanning Crossover, Probabilistic Selection

I. INTRODUCTION

Multi-parent crossovers had shown to improve the effectiveness of genetic algorithms on many problems [A.E. Eiben, C.A. Schippers, 1996]. Multi-parent crossover allows more than two parents to take part in crossover. However, it had been shown that an increased number of parents can lead to stronger exploitation or exploration or both [Chaun-Kang Ting, 2005][ A.E. Eiben, 2002]. The strong exploitation can result in a rapid convergence but can also cause a premature convergence. On the other hand, the strong exploration can slow down the convergence but also helps search into more regions and so increases the chance of reaching the optimal solution. These two conflicting goals need to be balanced.
and taken into consideration for the design of the multi-parent crossover.

Several variations of multi-parent crossover had been proposed [A.E. Eiben, 2002][ A.E. Eiben. 2000]. All proposed multi-parent crossovers rely on the heuristics to select which allele value of the multiple parents to become the value of the corresponding allele of an offspring. Three variations of the heuristics have been proposed, namely uniform scanning, occurrence-based scanning and fitness-based scanning.

For uniform scanning, each allele of the multiple parents has an equal probability to be chosen as the value for the corresponding allele of an offspring. Hence, the allele is chosen randomly with equal probability. For occurrence-based scanning, the allele which has the most number of occurrences among the multiple parents is chosen to be the corresponding allele of an offspring. If a tie occurs on the numbers of occurrences, the allele is chosen randomly from the relevant alleles. For fitness-based scanning, the allele is chosen based on the fitness value of its chromosome. The fitness proportionate strategy can simply be used to select one allele among those of the multiple parents. However, the strategy does not give good performance when compared with the other two scanning methods.

Although several variations of multi-parent crossover operators had been proposed, they sometimes do not show any significant improvement or sometimes degradation in convergence for some testing functions when the number of parents increases [Chaun-Kang Ting, 2005][ A.E. Eiben and T. Back, 1997][A.E.Eiben, P.E. Rau’e, and Zs Ruttkay, 1994][ T. Back, 1994]. The operators still face the problem of strong exploitation that leads to premature convergence on highly multi-modal search spaces.

This paper presents a new fitness based scanning multi-parent crossover operator that can effectively balance the two conflicting goals, exploitation and exploration. Specifically, at any locus the minimum fitness value of parents containing an allele is calculated and then used by the operator to determine which allele to select for an offspring. In general, if the allele with the lowest minimum fitness value is selected, the strongest exploitation is assured. On the other hand, if the allele with the highest minimum fitness value is selected, the strongest exploration is pursued. For binary-coded chromosomes, we use a probabilistic strategy to select between the two allele values. A dynamically adjusted probability parameter is also used to control the selection so the two conflicting goals can be dynamically balanced. According to the experiment results, the new operator can effectively balance the two conflicting goals leading to better solutions and less convergence time.
II. PROPOSED CROSSED OPERATOR

Like any fitness-based scanning crossovers, the proposed crossover operator uses fitness values of parent chromosomes to select which chromosomes to become the donor for an allele of an offspring. However, instead of using a fitness proportionate strategy to select the chromosome, a probabilistic selection is used. Let’s assume that a chromosome is a binary string of length L. The minimum fitness values of the parents with the particular allele value j at locus k, where j is 0 or 1 and k is between 1 and L, is computed as follows:

\[ M_k(j) = \begin{cases} \text{minimum of fitness values of parents with allele value of j at locus k} & \text{if there exists at least such a parent}, \\ \infty & \text{otherwise}. \end{cases} \]

The minimum fitness value \( M_k(j) \) represents the best fitness value among those of the parents containing the allele value of j at locus k. Using fitness-based scanning strategy, which allele value \( C_k \) to be selected for an offspring at locus k can be determined based on these best fitness values. Clearly, an allele value j with less \( M_k(j) \) value, i.e. a fitter allele, should have higher chance to be selected than the one with higher \( M_k(j) \) value. However, the higher the chance we give to the fitter allele, the more exploitation and the less exploration we put on the search. On the other way around the lesser the chance we give the allele, the less exploitation and the higher exploration we pursue. The decision must be made to select which parent allele to become a donor. The decision can be made by a probabilistic selection scheme which can dynamically adjust the probability to select the fitter allele so the amount of the exploitation and exploration pressures can be controlled during the search. In the case that all parents have the same allele value at locus k, the allele value will be selected for the allele value of the offspring. The selection scheme is as follows:

If \( M_k(0) \) is equal to infinity then

\[ C_k = 1 \]

else

If \( M_k(1) \) is equal to infinity then
\[ C_k = \begin{cases} 0 & \text{if } M_k(0) < M_k(1) \\ 1 & \text{if } M_k(0) > M_k(1) \\ 0 \text{ or } 1 \text{(randomly selected)} & \text{otherwise} \end{cases} \]

The threshold \( T \) in the above algorithm represents the probability that we will select the fitter allele for an offspring when we need to make a choice. It also controls the exploitation and exploration pressures for the search. It can be seen that the higher the value of the threshold (closer to one) the higher the exploitation pressure and the lower the exploration pressure, so the fitter allele has a higher chance to be selected for an offspring. On the other hand, the lower value of the threshold, the lower the exploitation pressure and the higher the exploration pressure, so the less fit allele can have a chance to be selected. Notice that the value \( 1 - T \) can represent the upper bound for the percentage of genes of the offspring, that are different from the best-fit parent genes.

During the earlier rounds of a genetic algorithm, considerable high exploration pressure and considerable low exploitation pressure are desired since the more search space can be explored and premature convergence can be avoided. During the later rounds of the genetic algorithm, the inverse must hold so the search can be accelerated toward the optimal solution. Hence, to achieve such a scenario, the threshold value must be dynamically adjusted. The threshold should be initially set to a value greater than 0.6, i.e. 0.8 is used in our experiments, to allow considerable high exploration pressure and considerable
low exploitation pressure. The threshold should then be gradually incremented by a constant $\Delta T$ for every generation. If the threshold reaches the maximum value of 1.0, the threshold will remain constant until the end of the genetic algorithm. The value of the constant $\Delta T$ represents a linear growth of $T$, which can be computed as follows

$$\Delta T = r \times (1.0 - T_{\text{init}})/\text{MaxGens}$$

Where $r$ is an incremental factor, $T_{\text{init}}$ is the initial value of the threshold $T$ and $\text{MaxGens}$ is the maximum number of generations. Notice that if $r$ is chosen to be zero, the threshold will remain constant and so there is no dynamic adjustment to the value of the threshold in this case.

III. PERFORMANCE EVALUATION

We conducted a series of experiments to evaluate the performance of the proposed operator. We compared the performance of the genetic algorithm using uniform scanning crossover, occurrence-based scanning and our proposed fitness-based crossover, respectively. The fitness-based crossover using proportionate selection scheme is not considered since it gives poor performance when compared with the three methods. The test functions include the extended De Jong’s second function (F2e), the Rastrigin function (RAS), and the Griewangk function (GRI). These test functions and the related parameters are described as follows. The objectives of the experiments are to find the global minimum of these test functions. The setting of the genetic algorithm for the experiments is summarized in Table 1. Each experiment is conducted over 100 independent runs.

Function F2e

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

Where

$N = 10$

Range of $x_i$ is (-2.048, 2.047)

Number bits of $x_i = 12$
Function RAS

\[ RAS = 10N + \sum_{i=1}^{N} \left( x_i^2 - 10 \cos(2\pi x_i) \right) \]

Where

\( N = 10 \)

Range of \( x_i \) is (-5.12,5.11)

Number bits of \( x_i = 10 \)

Function GRI

\[ GRI = 1 + \sum_{i=1}^{N} \frac{x_i^2}{4000} - \prod_{i=1}^{N} \cos \left( \frac{x_i}{\sqrt{i}} \right) \]

Where

\( N = 10 \)

Range of \( x_i \) is (-512,511)

Number bits of \( x_i = 10 \)

Table 1: The setting of the genetic algorithm for the experiments

<table>
<thead>
<tr>
<th>Representation</th>
<th>Bit string</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA model</td>
<td>Simple GA</td>
</tr>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>Parent selection</td>
<td>2-tournament</td>
</tr>
<tr>
<td>Crossover</td>
<td>Occurrence, Uniform and Proposed</td>
</tr>
<tr>
<td>Number of parents</td>
<td>2-10</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>1.0</td>
</tr>
<tr>
<td>Mutation</td>
<td>Bit-flip</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>1/length of a chromosome</td>
</tr>
<tr>
<td>Survivor selection</td>
<td>Generational</td>
</tr>
<tr>
<td>Termination</td>
<td>500 generations</td>
</tr>
<tr>
<td>Number of runs</td>
<td>100 runs per experiment</td>
</tr>
</tbody>
</table>
Figure 1-6 shows the comparisons of the mean best fitness values of the three test functions, averaged over 100 runs. The comparisons are shown between the results of genetic algorithm using the uniform multi-parent crossover operator and our proposed operator and between those of using occurrence-based operator and our proposed operator. Figure 7-11 shows the comparisons of convergence speeds of the genetic algorithm using the three operators with 10 parents on the function F2e. Figure 12 shows the convergence speeds of the genetic algorithm using our proposed operator with 10 parents on the function F2e but different incremental factor values. The results of convergence speeds of the genetic algorithm using the three operators for the other two testing functions will not be presented since they are similar to those of function F2e.

From the experimental results, we can see that the occurrence-based operator exhibits lower explorations but stronger exploitations as the number of parents participating in the crossover increases. Hence, more rapid convergences are seen but poorer solution qualities are achieved due to premature convergence. The uniform scanning operator can not give any improvement in terms of solution qualities and convergence speed as a higher number of parents are taking part in the crossover. The solution qualities of the genetic algorithm when using the uniform scanning operator are better than those when using the occurrence-based operator. On the other hand, convergence of the genetic algorithm when using uniform scanning operator are slower than those of the occurrence-based operator. The number of parents seems to have no impact on exploitation and exploration of the genetic algorithm in this case of the uniform scanning operator.

Our proposed operator can achieve better solutions than the other two when the number of parents is minimal at two. The operator can also improve the quality of solutions as the number of parents increases. However, there is no significant improvement when the number of parents exceeds a specific value, i.e. 4 to 6. The better solutions achieved by the proposed operator should be due to its dynamic scheme that applies high exploration pressure and low exploitation pressure at the earlier stages of the search and gradually change to the inverse during the late stages of the search. Hence, premature convergence can be avoided while better solutions can be reached.

The convergence speeds of the proposed operator are also slightly better than those of the uniform scanning operator. From the experimental results illustrated in figure 12, it can be seen that the higher is the incremental factor value, the higher is the exploitation pressure and so the higher is the convergence speed. However, when the incremental factor value becomes larger than a specific value, there is a higher
chance that the premature convergence may occur. We can see this from the slightly drop in the mean fitness values of solutions when the incremental factor value increases above a specific value, e.g. 1.0.

![Figure 1](image1.png)

Figure 1: Mean Best Fitness Value Comparison between uniform and proposed operators for function F2e

![Figure 2](image2.png)

Figure 2: Mean Best Fitness Value Comparison between Occurrence and proposed operators for function F2e
Figure 3: Mean Best Fitness Value Comparison between uniform and proposed operators for function RAS

Figure 4: Mean Best Fitness Value Comparison between Occurrence and proposed operators for function RAS
Figure 5: Mean Best fitness Value Comparison between uniform and proposed operators for function GRI.

Figure 6: Mean Best fitness Value Comparison between Occurrence and proposed operators for function GRI.
Figure 7: Convergence Speed Comparison among the three operators in the case of 2 parent recombination for function F2e

Figure 8: Convergence Speed Comparison among the three operators in the case of 4 parent recombination for function F2e
Figure 9: Convergence Speed Comparison among the three operators in the case of 6 parent recombination for function F2e.

Figure 10: Convergence Speed Comparison among the three operators in the case of 8 parent recombination for function F2e.
Figure 11: Convergence Speeds among the three operators in the case of 10 parent recombination for function F2e.

Figure 12: Convergence Speeds for different incremental rates of Threshold T in the case of 10 parent recombination for function F2e.
<table>
<thead>
<tr>
<th>Incremental Factor Value</th>
<th>0.0</th>
<th>0.5</th>
<th>1.0</th>
<th>2.0</th>
<th>4.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Best Fitness Value</td>
<td>80.5</td>
<td>7.3</td>
<td>6.8</td>
<td>7.3</td>
<td>7.2</td>
</tr>
</tbody>
</table>

Table 2: Mean Best Fitness Values for different incremental rates of threshold T in the case of 10 parent recombination for function F2e

IV. CONCLUSION

This paper presents a new multi-parent crossover operator that is based on fitness scanning approach. The proposed operator incorporates a dynamic scheme to maintain high exploration pressure and low exploitation pressure during the earlier stages of the genetic algorithm then gradually change to the opposite at the later stages of the genetic algorithm. This could help the algorithm explore more search space and avoid premature convergence due to strong exploitation. At the same time the operator can help the search to reach better solutions at the later stages as the exploitation pressures are allowed to increase. The experimental results show that the proposed operator not only outperforms the uniform and occurrence based scanning operators in terms of the solution qualities but also the convergence speed.

REFERENCES


