# A NEW CROSSOVER OPERATOR BASED ON THE ROUGH SET THEORY FOR GENETIC ALGORITHMS

FAN LI, QI-HE LIU, FAN MIN, GUO-WEI YANG

College of Computer Science and Engineering, University of Electronic Science and Technology of China,
Chengdu 610051, P.R.China
E-MAIL: lifan@uestc.edu.cn, gwyang@uestc.edu.cn

**Abstract:**

**The performance of genetic algorithms (GAs) is dependent on many factors. In this paper we have isolated one factor: the crossover operator. Commonly used crossover operators such as one-point, two-point and uniform crossover operator are likely to destroy the information obtained previously because of their random choices of crossover points. To overcome this defect, RSO, a new adaptive crossover operator based on the Rough Set theory, is proposed. By using RSO, useful schemata can be found and have a higher probability of surviving recombination regardless of their defining length. In this paper, the mechanism of RSO is discussed and its performance is compared with two-point crossover operator on several typical function optimization problems. The experimental results show that the proposed operator is more efficient.**

**Keywords:**

**Genetic algorithms (GAs); crossover operator; Rough Set theory; attribute reduction; reduct**

## 1. Introduction

Genetic algorithms (GAs) have been validated by their outstanding performance in optimization and machine learning for poorly understood, irregular and complex spaces [1]. The basic idea of GAs is to simulate the mechanisms of natural evolution such as selection, Recombination, and mutation. In canonical GAs [2], population is composed of individuals represented as fixed length binary vectors, and the population is generational. Recombination is implemented as a crossover operator, and mutation is an additional operator to provide diversity in a population. Recombination is one of the most salient features in GAs. Especially many researchers have more interest in crossover operator than other operators, because it is the important element that performs the exchanging and recombining genetic information from parents chosen through the selection mechanism [3]. In essence, through recombination, those distributed schemata are collected to form a better solution. The mechanism of this process can be explained by the schema theory [1]. Unfortunately, due to choosing crossover points randomly, commonly used crossover operators such as one-point, two-point and uniform crossover operator are likely to destroy useful schemata with high defining length. If such destruction happens, especially at the early state of the convergence process, it doesn't give any significant effects on the performance of GAs. Therefore, such crossover operators both to enable recombination and to keep the information obtained previously are desired to make GAs converging to the optimal solution effectively.

For this purpose, several crossover operators have been developed. HRO [4] considers only restricted region which shows homology over a specified threshold value when it selects crossover points. In the Puzzle Algorithm [5], there are two coevolving populations: candidate solutions and candidate building blocks. The fitness of an individual in the building-blocks population depends on individuals from the solutions population. The choice of recombination loci in the solutions population is affected by individuals from the building-blocks population.

In this paper, RSO, a new crossover based on the Rough Set theory, is presented. The main idea is to place constrains on the choice of crossover points. Firstly, candidates for useful schemata are found by using the attribute reduction, a basic notion in the Rough Set theory. Then, they are evaluated through recombination, and useful ones are preserved in this process. Thus, a useful schema has a higher probability of surviving recombination even if its defining length is very high.

The rest of this paper is organized as follows. In section 2, basic notions of the Rough Set theory are briefly reviewed. In section 3, the mechanism of the proposed crossover operator is discussed. In section 4, the efficiency of the proposed crossover operator is tested by computation experiments. We conclude the paper with a summary in Section 5.

0-7803-9091-1/05/$20.00 ©2005 IEEE

## 2. Preliminaries

In this Section, basic concepts of the Rough Set theory are briefly reviewed. More details can be found in [6-9].

Definition 1. An Information system (IS) is a triplet $T$=<$U$, $AT$, $f$>, where $U$ is a non-empty finite set of objects and $AT$ is a non-empty finite set of attributes, $f_a : U \to V_a$ for any $a \in$ AT, where $V_a$ is called domain of an attribute $a$. Here, we assume that an object $x \in U$ possesses only one value for an attribute $a$, $a \in AT$. In an IS, if $AT$ is composed of two disjoint subsets of attributes, called condition and decision attributes set respectively, then the IS is called a decision table, denoted by $DT$=<$U$, $C \cup D$, $f$> , where $C \cup D = AT$, $C \cap D = \varnothing$.

Definition 2. In an IS $T$=<$U$, $AT$, $f$>, each subset of attributes $A \subseteq AT$ determines a binary indiscernibility relation $IND(A)$ of $U$:

$$IND(A)=\{(x, y)\in U\times U | \forall a\in A, f_a(x)=f_a(y)\} \qquad (1)$$

The relation $IND(A)$ is an equivalence relation and constitutes a partition of $U$, denoted by $U/IND(A)$, in short $U/A$. Let $I_A(x)$ denotes the set of objects $\{y\in U | (x,y)\in IND(A)\}$. $I_A(x)$ is the equivalence class that contains the object $x$, and objects from $I_A(x)$ are indiscernible with regard to their description in the IS.

Definition 3. Let $T$=<$U$, $AT$, $f$> be an IS, $A \subseteq AT$ and $X \subseteq U$. The lower approximation $\underline{A}X$ and upper approximation $\overline{A}X$ of the set $X$ with respect to $A$ are defined as:

$$\underline{A}X = \{x \in U \mid I_A(x) \subseteq X\} \qquad (2)$$

$$\overline{A}X = \{x \in U \mid I_A(x) \bigcap X \neq \varnothing\} \qquad (3)$$

$\underline{A}X$ is the set of objects that belongs to $X$ with certainty, whereas $\overline{A}X$ is the set of objects that possibly belongs to $X$.

Definition 4. Let $DT$=<$U$, $C \cup D$, $f$>, the positive region of $D$ with respect to $C$ is denoted by

$$POS_C D = \bigcup_{X \in U/D} \underline{C}X \qquad (4)$$

If $POS_C D = U$, then the decision table $DT$ is consistent, otherwise it is inconsistent.

Definition 5. Let $DT$=<$U$, $C \cup D$, $f$>, a subset $A \subseteq AT$ is a relative reduct of $DT$ iff

$$POS_A D = POS_C D \qquad (5)$$

$$\forall a \in A, \ POS_{A-\{a\}} D \neq POS_C D \qquad (6)$$

Definition 6. Let $RED_D C$ be the set containing all relative reducts of $DT$. The relative core is denoted by

$$CORE_D C = \bigcap_{R \in RED_D C} R \qquad (7)$$

## 3. Crossover operator based on the Rough Set theory (RSO)

In canonical GAs, the schema theory suggests that those schemata with high defining length have lower probability to survive recombination, even if they have higher fitness. This is due to the premise that the crossover operator chooses crossover points at random. But apparently, nature does not do so. Experimental results suggest that human DNA can be partitioned into long blocks, such that recombinants within each block are rare or altogether nonexistent [10].

This paper introduces a new crossover operator that helps to preserve schemata of promising performance. It is clear that we can hardly identify which schema is useful in the search space. However, using the concept of attribute reduction in the Rough Set theory, we can find the key genes on the chromosome that distinguish whether it has promising performance. Thus, we can choose schemata determined by those key genes as candidates and evaluate them. If the evaluation proves that they are useful, we can preserve them during recombination by placing constraints on the choice of crossover points, such that useful schemata have a higher probability of surviving recombination.

### 3.1. Constructing the chromosome decision table

In order to find schemata of promising performance, the chromosome decision table should be constructed firstly.

Let population $\overrightarrow{X}(t) = \{X_1(t), X_2(t), \ldots, X_n(t)\}$, Each individual in population is treated as an object in $DT$, and each bit of the corresponding chromosome string is a condition attribute. Thus, $DT_{chromosome}$=<$U$, $C \cup \{d\}$, $f$>, *where* $C=\{c_1, c_2, \ldots, c_m\}$ ($m$ is the length of the chromosome) and $\forall c_i \in C$, $V_{C_i} = \{0, 1\}$. Also, $V_d = \{0, 1\}$, each object of the decision table is assigned a decision value, which can be defined as:

$$f_d(X_i(t)) = \begin{cases} 1 & J(X_i(t)) \geq J_{av}(\overrightarrow{X}(t)) \\ 0 & J(X_i(t)) < J_{av}(\overrightarrow{X}(t)) \end{cases} \qquad (8)$$

where

$$J_{av}(\overrightarrow{X}(t)) = \frac{1}{n}\sum_{j=1}^{n} J(X_j(t)) \qquad (9)$$

means the average fitness of population $\overrightarrow{X}(t)$.

It is easy to prove that a chromosome decision table is a consistent decision table.

Here, we employ two examples to illustrate some concepts and computations involved in our proposed method. The aim of both examples is to maximize a function $f(x)=x^2, 0\le x\le 31$.

Example 1: The initial population $\overrightarrow{X}(0)$ = {20, 15, 2, 5, 8}.

Example 2: The initial population $\overrightarrow{X}(0)$ = {18, 16, 9, 7}.

In both examples, the length of the chromosome is 5. The chromosome decision tables are constructed as follows:

Table 1. Chromosome decision table 1

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | d |
|---|---|---|---|---|---|---|
| $X_1(0)$ | 0 | 0 | 1 | 0 | 1 | 1 |
| $X_2(0)$ | 1 | 1 | 1 | 1 | 0 | 1 |
| $X_3(0)$ | 0 | 1 | 0 | 0 | 0 | 0 |
| $X_4(0)$ | 1 | 0 | 1 | 0 | 0 | 0 |
| $X_5(0)$ | 0 | 0 | 0 | 1 | 0 | 0 |

Table 2. Chromosome decision table 2

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | d |
|---|---|---|---|---|---|---|
| $X_1(0)$ | 0 | 0 | 0 | 0 | 1 | 1 |
| $X_2(0)$ | 0 | 1 | 0 | 0 | 1 | 1 |
| $X_3(0)$ | 1 | 0 | 0 | 1 | 0 | 0 |
| $X_4(0)$ | 1 | 1 | 1 | 0 | 0 | 0 |

### 3.2. Finding candidate schemata

By the definition of a reduct, it is a minimal subset of attributes that enables us to classify objects with high fitness (decision values are 1) and those with low fitness (decision values are 0). Thus, we can choose schemata determined by a reduct to be candidate schemata with promising performance. It is clear that finding all reducts then choosing the most suitable one is the best scheme. But it is already proved to be a NP-hard problem [11]. So it is more feasible to use a heuristic algorithm to acquire the optimal or hypo-optimal result, especially when the size of population is large.

Definition 7. Let $DT$=<$U$, $C\cup\{d\}$, $f$> be an decision table and $A\subseteq C$, then the significance for each attribute $a\in C\setminus A$ is defined as:

$$SGF(a,A)=\eta_{A\cup\{a\}}-\eta_A \tag{10}$$

where

$$\eta_A=\frac{card(POS_A D)}{card(U)} \tag{11}$$

Based on the definition above, an attribute reduction algorithm is presented as follows:

Input: $DT_{chromosome}$=<$U$, $C\cup\{d\}$, $f$>.

Output: a reduct of $DT_{chromosome}$.

Step1. Compute $\eta_C$ for $DT_{chromosome}$.

Step2. Compute $CORE_{\{d\}}C$ . Let $RED$ = $CORE_{\{d\}}C$, and compute $\eta_{RED}$.

Setp2.1. Let $B$ = $C\setminus RED$. For each attribute $a\in B$, compute $\eta_{RED\cup\{a\}}$;

Setp2.2. Choose an attribute $b$ such that $\eta_{RED\cup\{b\}}$ is maximal (thus *SGF* (*b, RED*) is maximal), inserting it to the end of *RED*. Then, set $\eta_{RED}=\eta_{RED\cup\{b\}}$;

Setp2.3. If $\eta_{RED}$= $\eta_C$ then go to Setp3. Otherwise, go to Setp2.1。

Setp3. From the end to the head of *RED*, test whether an attribute $c$ is redundant:

- If $\eta_{RED\setminus\{c\}}$= $\eta_C$, then $c$ is redundant, deleting it from *RED*;
- If all the attributes in *RED* have been tested, the algorithm completes.

Then, schemata determined by the output of the attribute reduction algorithm are chosen to be candidate schemata with promising performance. The set of those schemata is denoted as $S_{can}$. In example 1, supposing that we have *RED*= $\{c_1, c_4, c_5\}$, $S_{can}$ = {(0, *, *, 0, 1), (1, *, *, 1, 0)}. In example 2, supposing that we have *RED*= $\{c_3, c_4\}$, $S_{can}$ = {(*, *, 0, 0, *)}. But schemata in $S_{can}$ are not always useful, so the recombination strategy is proposed as follows to evaluate them as well as exchanging genes on the chromosome.

### 3.3. Recombination strategy

In the method proposed in this paper, each recombination uses two parents, say $X_1(t)$ and $X_2(t)$, to create two children. During recombination two parents are selected and their fitness is checked by their decision values.

- If $f_d(X_1(t))=1\wedge f_d(X_2(t))=1$ or $f_d(X_1(t))=0\wedge f_d(X_2(t))=0$, crossover points are chosen randomly, then recombination is executed like a standard GA.
- Otherwise, genes on the position determined by schemata in $S_{can}$ are exchanged, with probability $p_c$.

After recombination the two new children and two parents are evaluated and the replacement selection is employed.

The main idea of this strategy is as follows:

- If the fitness of either parent is greater than or less than $J_{av}(\overrightarrow{X}(t))$ , then crossover points are chosen randomly. If a single child's fitness is less than the fitness of its parent, it is eliminated through the replacement selection. Otherwise, the parent is replaced by its child. This is done to reflect the genes' contribution to the fitness increase.

- If the fitness of a parent is greater than $J_{av}(\overrightarrow{X}(t))$ and the fitness of the other is less than $J_{av}(\overrightarrow{X}(t))$, say $f(X_1(t))=1$ and $f(X_2(t))=0$, we intend to preserve schemata in $S_{can}$ through recombination. At the same time, they are checked by recombination. If a schema in $S_{can}$ is useful, through the recombination, a child whose fitness is greater than $X_2(t)$ and a child whose fitness is less than $X_1(t)$ will be produced. After the replacement selection, this schema exists in both survivals. Otherwise a child whose fitness is greater than $X_1(t)$ and a child whose fitness is less than $X_2(t)$ will be produce. Thus, if a schema in $S_{can}$ is not useful, it does not exist in both survivals after the replacement selection.

Fig. 1 and Fig. 2 illustrate the mechanism for the above examples.

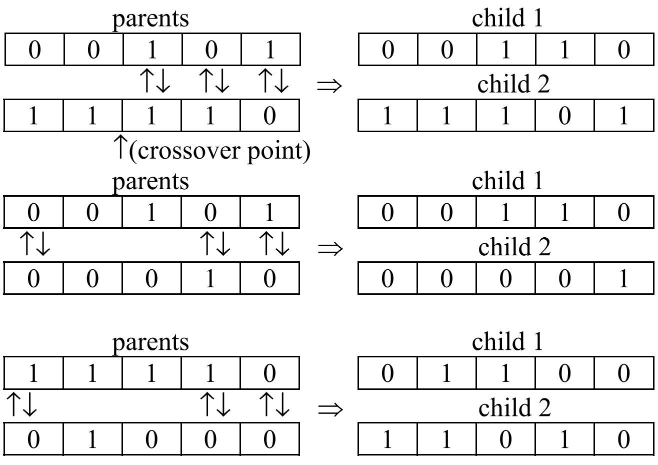

Figure 1. Recombinations in example 1

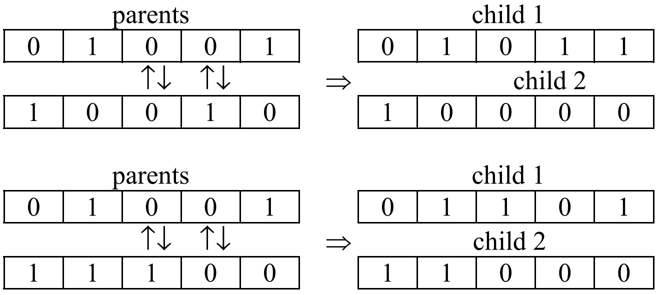

Figure 2. Recombinations in example 2

In the first recombination of Fig. 1, the fitness of either parent is greater than $J_{av}(\overrightarrow{X}(t))$, so the crossover point is chosen randomly. Child 2 has an increase in fitness hence parent 1 and child 2 come into the next generation. In the second and the third recombination, "bad" parents (parents whose fitness is less than $J_{av}(\overrightarrow{X}(t))$) have an increase in fitness after exchanging genes determined by schemata in $S_{can}$. Meanwhile, "good" parents (parents whose fitness is greater than $J_{av}(\overrightarrow{X}(t))$) have a decrease in fitness. Thus, schemata in $S_{can}$ can be regarded as useful. After the replacement selection, schemata in $S_{can}$ survive in spite of their high defining length.

But in either recombination of Fig. 2, after exchanging genes determined by schemata in $S_{can}$, the fitness of the "bad" parent decrease while the fitness of the "good" parent increase. That means schemata in $S_{can}$ are not those schemata with promising performance. After the replacement selection, they do not exist in survivals and the corresponding schemata in "bad" parents, (*, *, 1, 0, *) and (*, *, 0, 1, *), are not eliminated.

In a word, recombination strategy proposed here is responsible for not only exchanging and recombining genetic information from parents but evaluating schemata in $S_{can}$ to decide if they are those schemata with promising performance.

## 4. Performance evaluation

We have implemented the GA using RSO on function optimization problems and its performance has been compared with the simple GA. We used the "mean best" of all function evaluations to measure the performance (averaged over 10 runs). In this paper, we performed comparative experiment for three typical functions. They were:

(1) Schwefel's sphere function

$$f_1(x)=\sum_{i=1}^{30} x_i^2 \quad -5.12\le x_i\le 5.12 \qquad (12)$$

(2) DeJong's function

$$f_2(x)=\sum_{i=1}^{29}(100(x_i^2-x_{i+1})^2+(x_i-1)^2) \qquad -5.12\le x_i\le 5.12 \qquad (13)$$

(3)Griewangk's function

$$f_3(x)=1+\sum_{i=1}^{20}\frac{x_i^2}{800}-\prod_{i=1}^{20}\cos\frac{x_i}{\sqrt{i}}, \quad -512\le x_i\le 512 \qquad (14)$$

Functions $f_1(x)$ and $f_2(x)$ are unimodal functions, and $f_3(x)$ is a multimodal function where the number of local minima increases exponentially with the dimension.

In all experiments, the crossover rate is 0.85; the mutation rate is 0.001; the population size is 100; maximum number of generations is 10000. The simple GA uses two-point crossover operator. We use linear ranking selection with $\eta$=1.05. During the whole optimization process we kept the best value found in all generations, beginning from the initial population. Experimental results show that for all test functions the GA using RSO performs better than the simple GA.

Table 3. Experimental results

| Algorithm | simple GA | GA using RSO |
|---|---|---|
| $f_1^{avg}$ | 32.853 | 10.602 |
| $f_2^{avg}$ | 13078.212 | 3642.138 |
| $f_3^{avg}$ | 9.622 | 3.237 |

## 5. Conclusions and discussions

In this paper, we have described a new crossover operator based on the Rough Set theory, called RSO, for GAs. Unlike conventional crossover operator such as one-point, two-point and uniform crossover operator, it allows exploitation of good schemata regardless of their defining length.

RSO uses the attribute reduction, a concept in the Rough Set theory, to find candidate schemata with promising performance. Then, a recombination strategy is used to evaluate if these schemata are useful as well as to exchange genetic information from parents.

Experiments indicate that GA using RSO performs better than traditional GA, which uses two-point crossover, on a set of test functions.

We conclude that the initial results of this study indicate that RSO may be a good candidate for a crossover operator in which practitioners can have more confidence to use as a starting point for an adaptive GA system. But there are still some issues needs to be discussed in future works.

- Currently, when assigning the decision values for each object in $DT_{chromosome}$, we just consider its fitness value. But maybe this is not an efficient way because the diversity of population is not taken into account. For example, we can consider the fitness value and the Hamming Distance to the best object at the same time.
- In this paper, we focus on the crossover operator only. But the mutation operator is another important factor that influences the performance of GAs. So if we reconstruct mutation operator by the Rough Set theory, the performance of GAs is promising.
- Further experiments should be carried out to compare RSO with other adaptive crossovers. We also intend to analyse the creation, propagation and disruption of a schema to determine exactly why and how a GA benefits from RSO and use those findings towards more reliable and adaptive crossover operators.

## References

[1] J. Holland, Adaptation in Natural and Artificial Systems, Michigan University Press, Ann Arbor, MI, 1975.

[2] D. E. Goldberg, Genetic Algorithms in search, optimization and machine learning, Addison-Wesley, 1989.

[3] L. Davis, Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.

[4] Jong-man Park, Jeon-gue Park *et al.*, “Robust and Efficient Genetic Crossover Operator: Homologous Recombination”, Proceedings of 1993 International Joint Conference on Neural Networks, Nagoya, pp. 2975-2978, October 1993.

[5] Assaf Zaritsky, Moshe Sipper, “The Preservation of Favored Building Blocks in the Struggle for Fitness: The Puzzle Algorithm”, IEEE Transactions on Evolutionary Computation, Vol. 39, No. 5, pp. 443-455, October 2004.

[6] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, W. Ziarko, “Rough sets”, Communication of the ACM, Vol. 38, No. 11, pp. 89-95, 1995.

[7] Z. Pawlak, “Rough set approach to knowledge-based decision support”, European Journal of Operational Research, Vol. 99, No. 1, pp. 46-57, 1997.

[8] Pawlak Z. Rough sets, decision algorithms and Bayes’ theorem. European Journal of Operational Research, Vol. 136, No. 1, pp. 181-189, 2002.

[9] Wen-Xiu Zhang，Wei-Zhi Wu，Ji-Ye Liang，De-Yu Li, Rough Set Theory and Approach, Science Press, Beijng，2001.

[10] S. B. Gabriel *et al.*, “The structure of haplotype blocks in the human genome”, Science, Vol. 296, No. 5576, pp. 2225-2229, 2002.

[11] S. K. M. Wong, W. Ziarko, “Optimal decision rules in decision table”, Bulletin of Polish Academy of Science, Vol. 33, No. 11-12, pp. 693-696, 1985.

[12] C. Emmanouilidis, A. Hunter, J. MacIntyre, “A Multiobjective Evolutionary Setting for Feature Selection and a Commonality-Based Crossover Operator”, Proceedings of the 2000 Congress on Evolutionary Computation, La Jolla, pp. 309-316,

2000.

[13] Peter A. N. Bosman, Dirk Thierens, "The Balance Between Proximity and Diversity in Multiobjective Evolutionary Algorithms", IEEE Transactions on Evolutionary Computation, Vol. 7, No. 2, pp. 443-455, April 2003.

[14] Zong-Ben Xu, Kwong-Sak Leung, Yong Liang *et al.*, "Efficiency speed-up strategies for evolutionary computation: fundamentals and fast-GAs", Applied Mathematics and Computation, Vol. 142 , No. 2-3, pp. 341-388, October 2003.

[15] Fan Li, Qihe Liu, Guowei Yang, "A Heuristic Algorithm for Attribute Reduction in Incomplete Information Systems", Progress in Intelligence Computation and Applications, Wuhan, pp.574-580, April 2005.

[16] Qihe Liu, Fan Min, Hongbin Cai, Guowei Yang, "Equivalent Representation of Knowledge Reduction in a New Information View and an Algebra View", Progress in Intelligence Computation and Applications, Wuhan, pp. 427-431, April 2005.