Genetic Programming Operators Applied to Genetic Algorithms

Dana Vrajitoru

Department of Mathematics, EPFL, CH-1015 Lausanne
Email: *dana.vrajitoru@epfl.ch*, phone: (+41)(21) 693.55.04

Abstract

Like other learning paradigms, the performance of the genetic algorithms (GAs) is dependent on the parameter choice, on the problem representation, and on the fitness landscape. Accordingly, a GA can show good or weak results even when applied on the same problem. Following this idea, the crossover operator plays an important role, and its study is the object of the present paper. A mathematical analysis has led us to construct a new form of crossover operator inspired from genetic programming (GP) that we have already applied in field of information retrieval. In this paper we extend the previous results and compare the new operator with several known crossover operators under various experimental conditions.

1. INTRODUCTION

A major difficulty encountered when using GAs is parameter setting. There exist many forms and variations of GAs and the best choice is problem dependant (Mitchell *et al.*, 1991; Spears, 1995). An important parameter choice concerns the crossover operator. Various studies have shown that the form of crossover can influence the performance of GAs (Spears, 1995). From the invention of GAs (Holland, 1975), several variations of the crossover have been developed (De Jong, 1975; Syswerda, 1989; Maini *et al.*, 1994; Mohan, 1998). Some particularities of chromosome representation in GP have entailed a crossover operator adapted to them. The present research is based on the same ideas.

In our previous work (Vrajitoru, 1998), we have used the GAs in information retrieval. During the initial experiments, we have encountered difficulties to improve the baseline performance. We have related these difficulties to a particular phenomenon: after a crossover operation, the performances of the children are situated in between the performances of the parents. The analysis of this phenomenon has shown that the reason for it was the fact that the same crossover site is applied to both parents.

In GP, as the individual size is variable, cutting two individuals at the same place has no meaning and each parent has its own cross site. Koza (1992) remarks that, with this operator, the genetic population converge, because an individual crossed with itself can produce offspring different from itself. We also think that the cited phenomenon can be avoided more often by using this operator. Thus, we propose a new crossover operator called *dissociated*, which follows the same idea as the GP crossover.

The results of the new operator have been promising in the difficult problem information retrieval. This success has led us to extend the previous work to various fitness landscapes and this is the main object of the present paper. We have used the set of ten classical test functions and some NP-complete problems to compare the new operator with other known crossover operators. To anticipate the results, the dissociated operator presents some advantages in a great number of situations, but also shows some weak points.

Section 2 introduces the dissociated crossover and presents the theoretical analysis that led us to its conception. Section 3 presents the experimental test conditions and the results. Finally, Section 4 discusses the advantages and disadvantages of the new operator.

2. THEORETICAL PRESENTATION

The present section describes the dissociated crossover as well as the operators that we will to compare with it.

2.1 CROSSOVER OPERATORS

In a previous paper (Vrajitoru, 1998), we introduced a new crossover operator, named *dissociated* crossover, that we will describe first. If $parent_1$ and $parent_2$ are two individuals, or potential solutions to the problem, and $1 \le crossSite_1 \le crossSite_2 \le L$ (the length of the individual) are two crossover sites, the dissociated crossover creates two new individuals, $child_1$ and $child_2$, in the following manner:

$$child_1 = \begin{cases} parent_1(i) & \text{if} \quad i \le crossSite_1 \\ parent_1(i) & \text{if} \quad crossSite_1 < i \le crossSite_2 \\ \text{ or } parent_2(i) & \\ parent_2(i) & \text{if} \quad i > crossSite_2 \end{cases}$$

$$child_2 = \begin{cases} parent_2(i) & \text{if} \quad i \le crossSite_1 \\ 0 & \text{if} \quad crossSite_1 < i \le crossSite_2 \\ parent_1(i) & \text{if} \quad i > crossSite_2 \end{cases} \tag{1}$$

We have chosen to compare this operator with three others: the 1-point, the n-point, and the uniform crossover. We will briefly recall the functionality of these operators.

If $parent_1$ and $parent_2$ are two individuals, and *crossSite* is a random position from 1 through *L*, the *1-point* crossover (Holland, 1975) cuts each parent at the point *crossSite*, and swaps the resulting parts.

The *n-point* crossover is equivalent to *n* independent 1-point crossovers applied in sequence (De Jong, 1975). For our experiments, we have chosen the 2-point crossover operator, because it uses the same number of cross sites as the dissociated crossover.

The *uniform* crossover (Syswerda, 1989) consists of independently choosing, for each locus *i* from 1 to *L*, whether the parents genes will be swapped or not. This choice depends on a swap probability denoted p_{swap}. We have adopted the 0.5 swap probability for our research.

Even if both the 2-point and the dissociated crossover operators use two cross sites, there is an important difference between them (Figure 1), which can be expressed by the following:

- the two-point crossover applies the same two simple crossover operators to each parent,
- the dissociated crossover applies a different 1-point crossover operator to each parent. In this case, the question is not "how do we obtain each child", but "what happens to each parent."

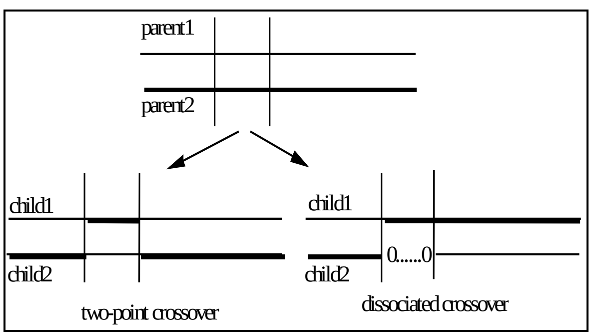

Figure 1: Dissociated Crossover Versus Two-Point Crossover.

2.2 THEORETICAL MOTIVATION

We have initially developed the dissociated crossover in the field of information retrieval (Vrajitoru, 1998) and this subsection presents the theoretical considerations that led us to define this operator.

The main idea of the GAs is to simulate the mechanism of natural selection of living beings, which makes the ecosystems develop and become stable. Within this mechanism, the organisms adapt themselves through generations to their environment and to specifics survival tasks. Through rough competition, the best individuals mate to produce descendants that can inherit their skills, and even increase them.

Inspired by this natural phenomenon, the purpose of the crossover operation is to create new individuals having, hopefully, greater performance than their parents. In our case, we have noticed that the performance of the children can often be expressed by the following:

$$f(parent_1) \le f(child_{1,2}) \le f(parent_2)$$

Intuitively we would say that the children are convex linear combinations of the parents. We will show that it is almost true, and that this phenomenon can slow the performance improvement.

Notation. If *H* is a hyperplane, $o(H)$ denotes its number of fix values, or length, and $E(o(H))$ its expected length.

Let us consider a crossover between $parent_1$ and $parent_2$ and an optimal individual ind_{opt} maximising the length of the intersection with $parent_1 \cup parent_2$. We chose ind_{opt} so that

$$cardinal\left\{ i,\, 1 \le i \le L,\, \begin{pmatrix} ind_{opt}(i) = parent_1(i) or \\ ind_{opt}(i) = parent_2(i) \end{pmatrix} \right\}$$

is maximal.

Let us consider, for each parent, the hyperplane of maximal length containing both the parent and the optimal individual ind_{opt}.:

$$H_i \text{ so that } \begin{cases} parent_i \subset H_i \\ ind_{opt} \subset H_i \\ o(H_i) \text{ is maximal} \end{cases}, \; i = 1,2 \tag{2}$$

Without loss of generality, we can assume that $o(H_1) > o(H_2)$.

Let $child_1$ and $child_2$ be the individuals produced by one crossover between $parent_1$ and $parent_2$. As our goal is to find an optimal individual, at least one of the children should contain a greater part of ind_{opt} than the parents. Let H_3 be the hyperplane of maximal length containing both $child_1$ and ind_{opt}, as we defined it for the parents (Equation 2). We would like to tune the crossover so that $o(H_3) > o(H_1)$.

If *crossSite* is the cut point for the crossover, we denote by I_{left} the interval [1..*crossSite*] and by I_{right} the interval [*crossSite* + 1..*L*]; then each of H_i, i = 1,2, is the intersection of two hyperplanes $H_{i\ left}$ and $H_{i\ right}$, each having fix values only on one side of *crossSite* (see Figure 2).

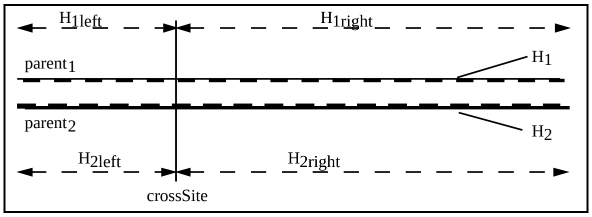

Figure 2: Hyperplanes Created By Crosssite

After the crossover (see Figure 3), it seams clear that H_3 is the intersection between $H_{1\ left}$ and $H_{2\ right}$; then we can compute:

$$o(H_3) = o(H_{1left}) + o(H_{2right})$$

We also know that

$$o(H_1) = o(H_{1left}) + o(H_{1right})$$

The condition $o(H_3) > o(H_1)$ becomes:

$$o(H_3) > o(H_1) \Leftrightarrow o(H_{2right}) > o(H_{1right}) \qquad (3)$$

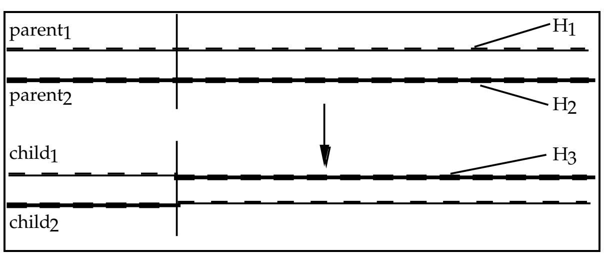

Figure 3: Crossover Result

Equation (3) means that $child_1$ can be closer to the optimal individual if and only if we replace $H_{1\ right}$ with a hyperplane of greater length. We already know that $o(H_1) > o(H_2)$. Under this condition, can H_2 be more dense than H_1 on the interval I_{right}? We will show that the chances are low, especially when starting with a random population and when L is large.

Let us consider the case of the starting population, where each individual has been chosen at random; then we could say that the probability that a gene has a property Q : [1..*L*] • {*true*, *false*} should not depend on the position of the gene (all genes have equal chances). If I is an interval, we could express this property by:

$$p\left(Q\left(parent_j(i)\right)\right) = \text{constant}, \forall i \in I, j = 1,2 \qquad (4)$$

Equation (4) is the key to our analysis and we should check its probability. We already mentioned that it should hold for the starting population. The schemata theorem (Goldberg, 1989) tells us that as the individuals evolve through generations, large connected schemata tend to dominate the population. This also means that Equation (4) becomes more probable, and that the following analysis should hold especially in the convergence situation.

In Equation (4), we will now replace the property Q by the fact that the gene is a fix position in $H_{1,2}$:

$$p\left(parent_j(i) \in H_j \middle| i \in I_{right}\right) = p\left(parent_j(i) \in H_j\right) \Leftrightarrow$$
$$p\left(parent_j(i) \in H_{j\ right}\right) = p\left(parent_j(i) \in H_j\right)$$
$$= \frac{o(H_j)}{L} \qquad j = 1,2$$

By computing the expected length of $H_{1\ right}$ and of $H_{2\ right}$:

$$E\left(o(H_{j\ right})\right) = (L - crossSite) \cdot p\left(parent_j(i) \in H_{j\ right}\right)$$
$$= o(H_j) \cdot \frac{L - crossSite}{L} \qquad j = 1,2$$

We obtain that:

$$o(H_1) > o(H_2) \Rightarrow E\left(o(H_{1right})\right) > E\left(o(H_{2right})\right) \qquad (5)$$

We can see that there is a contradiction between Equations (3) and (5). This contradiction implies that the expected length of the hyperplane containing both $child_1$ and ind_{opt} does not exceed the length of the largest hyperplane containing one of the parents and ind_{opt}. In other words, the child is no closer to the optimal individual than its parents were and the search cannot succeed.

If we note by $\alpha = crossSite/L$, then we could write:

$$E\left(o(H_3)\right) = \alpha \cdot o(H_1) + (1 - \alpha) \cdot o(H_2)$$

This means that the child is expected to be a convex linear combination of the parents, as we have stated it in the beginning of the section.

3. EXPERIMENTAL COMPARISON

This section describes the various experimental conditions and the results of the comparison between the dissociated crossover and the classical operators we presented in Section 2.

3.1 EXPERIMENT DESCRIPTION

The domain of information retrieval has provided us with some special experimental conditions: a very large individual size, a small number of generations, for reasons of computational time, and the fact that all optimal individuals share similar relevant information. On this problem, the dissociated crossover has systematically and significantly performed better than the 1-point crossover (Vrajitoru, 1998).

Nevertheless, from the fact that the new operator is fit for this problem, we cannot conclude that it should perform well in any condition. So, we have run new experiments with completely different fitness landscapes and parameter settings that we will describe in detail.

For each of the chosen functions, we have performed 10 experiences for each operator. We have limited the number of generations to 1000 or until convergence of the population. For each experiment, the population is composed of 20 individuals. As we were only interested in crossover comparison, the mutation probability equals 0, and the crossover probability is equal to 1. We have used the fitness-proportionate selection (Goldberg, 1989), and a variant of the elitist reproduction: the ancient best individual replaces the worst individual in the new generation, if and only if the new generation contains nothing better than it.

We have only considered the performance of the best individual in each generation. To evaluate the experiments, we have chosen two measures: the mean performance over 10 runs, and the best run for each crossover operator. For each measure, we have used the Mann-Whitney test (Saporta, 1990) to decide if the results of the top operator are significantly better than those obtained by the second one.

We were interested in four results for each problem: the top first and second operator for the mean and the best run. In case the top operator is significantly better than the second one according to the Mann-Whitney test, its name is emboldened. For the best run we have denoted in parenthesis the minimal number of generations that the operator has needed in this run to find an optimal individual, assuming the search was successful.

3.2 STANDARD FUNCTIONS SET

The evolution of the Gas for many years, has brought the researchers to build a set of standard test functions that are often used to test a form or to compare two variants of GAs. They seamed like a good start for our experiment, as many researchers have already used them (Kingdom and Dekker, 1995; Belew, 1992). F1 through F5 have been developed by De Jong (1975), F6 by Rastrigin, F7 by Schwefel, F8 by Griewank, F9, and F10 by Schaffer (1989). Their full description can be found in (Whitley *et al.,* 1996).

Recent studies (Davis, 1991; Whitley *et al*., 1996) have criticised these functions as being easy for hill-climbing, scalable, symmetric on their variables, and linear. They still offer a variety of characteristics and test the GA against different search spaces, reasons for which we chose to use them in our experiment.

The results presented in Table 1 show an important advantage of the dissociated crossover over the others. It appears on the top for more than half of the runs, and its results are often significant.

We can also remark that, for these functions, the best operator always found the optimal solution, which suggests that they are not very hard for the GAs. For this reason, we have also tried more difficult test functions.

Table 1: Result Of The Standard Functions Set

FUNCTION	MEAN TOP	MEAN 2ND	BEST RUN TOP	BEST RUN 2ND
F1	**dissociated**	2-points	**dissociated (10)**	2-points (650)
F2	**dissociated**	1-point	**dissociated (10)**	uniform (250)
F3	**dissociated**	uniform	**dissociated (10)**	uniform (450)
F4	**dissociated**	uniform	**dissociated (30)**	2-points
F5	1-point	2-points	dissociated (10)	2-points (30)
F6, N=2	**dissociated**	1-point	uniform (350)	1-point (650)
F6, N=5	dissociated	2-points	1-point (850)	uniform
F7, N=5	**dissociated**	2-points	2-points (1000)	dissociated
F8, N=5	**dissociated**	1-point	**dissociated (10)**	uniform (1000)
F9	**dissociated**	1-point	uniform (30)	dissociated (70)
F10	**dissociated**	1-point	dissociated (20)	2-points (50)

3.3 NP-COMPLETE PROBLEMS

To increase the difficulty of the test, we have chosen the class of NP-complete problems. Several researchers have already applied the GAs to some of these functions (De Jong and Spears, 1989, 1990). Their work was of great help in our experiments, concerning the general concepts and the set of problems they provided.

The first NP-complete problem we have used is the Boolean satisfiability problem (SAT). Given a Boolean expression depending on some variables, does there exist an assignment of those variables such that the value of the expression becomes true?

The genetic representation of this problem is straightforward. Each gene is a variable occurring in the expression, and the values 0 / 1 are interpreted as false / true.

The difficulty is the choice of the fitness function. The classical evaluation of a Boolean expression returns only the true or false values. Within this evaluation, as long as an individual does not represent an exact solution for the expression, it is evaluated to 0. This makes it difficult for the GA to improve the individual performance, because it cannot decide whether an individual is far from or close to the researched solution.

De Jong and Spears (1989), propose an improvement of this fitness function by the following:

$$v_1 \; and \; v_2 \; and \ldots v_n = \left(\frac{v_1 + v_2 + \cdots v_n}{n} \right)^p$$

$$v_1 \; or \; v_2 \; or \ldots v_n = \max(v_1, v_2, \ldots v_n)$$

We have performed our experiences with p = 1. Greater values of p have not shown significantly different results.

We have used the set of problems provided by De Jong and Spears (1989). All problems are noted according to their number of optimal and suboptimal peaks. They all have a single optimal solution, and several suboptimal ones, which makes them hard for the GAs.

From the results presented in Table 2, we can see that the other operators perform better than the dissociated crossover on these problems. This result shows that this problem is especially difficult for the dissociated crossover for reasons we will explain later on.

Table 2: Results On The SAT Problems

PROBLEM	MEAN TOP	MEAN 2ND	BEST RUN TOP	BEST RUN 2ND
Peak1	dissociated	2-points	**dissociated (10)**	uniform (150)
Peak2	2-points	1-point	uniform (50)	1-point (150)
Peak3	2-points	uniform	1-point (90)	uniform (110)
Peak4	Uniform	2-points	uniform (90)	2-points (100)
Peak5	2-points	1-point	1-point (90)	uniform (110)
Peak6	1-point	2-points	**uniform (100)**	1-point (250)

Kingdon and Dekker (1995) have suggested that when an evolutionary algorithm has difficulties in finding the best solution, reshaping the search space can help recover these difficulties. We have followed this idea and have reshaped the problems by a permutation of the genes. The new results, presented in Table 3, show again an advantage of the dissociated crossover over the others.

The second NP-complete problem we have tested is the hamiltonian circuit (HC). Given an oriented graph, does there exist a circuit that passes once and only once by each node? This well-known NP-complete problem is equivalent to a travelling salesman (TS) problem where all arc costs would be equal to 1. Several researchers have already used Gas to solve the TS problem (Maini *et al.*, 1994; Sushil and Gong, 1997; Ross *et al.*, 1998).

Table 3: Reshaped SAT Problems

PROBLEM	MEAN TOP	MEAN 2ND	BEST RUN TOP	BEST RUN 2ND
Peak2'	**dissociated**	2-points	uniform (60)	dissociated (90)
Peak3'	dissociated	2-points	dissociated (10)	uniform (50)
Peak4'	dissociated	1-point	**dissociated (20)**	uniform (110)
Peak5'	dissociated	2-points	**dissociated (20)**	uniform (170)
Peak6'	dissociated	2-points	**dissociated (20)**	2-points

We could represent the HC problem directly in a genetic form, but it is easier to transform the HC instances into SAT instances, whose genetic representation we already know, as suggested by De Jong & Spears (1989, 1990).

Given a graph with the nodes $\{A_1, A_2, \ldots A_n\}$, and some oriented arches between these nodes, the transformation of this HC instance into a SAT instance contains the following steps:

For each node A_i, we construct the following expression:

$$E_{i,in} = xor\left(A_k A_i, \forall \text{ arch } A_k A_i \text{ that enters } A_i\right)$$

$$E_{i,out} = xor\left(A_i A_j, \forall \text{ arch } A_i A_j \text{ that exits from} A_i\right)$$

Then the SAT expression is:

$$\left(And\left(E_{i,in}, i = 1..n\right)\right) \; and \; \left(And\left(E_{i,out}, i = 1..n\right)\right)$$

We have used graphs with the number of nodes from 4 to 15, and some of them were provided by De Jong and Spears (1989). We have noted the problems according to the number of nodes in the graph. The results, presented in Table 4, show that, even if the dissociated crossover performs quite well on the whole, almost none of the top operators is significantly better than the others. We think that this fact is due to the difficulty of the HC problems.

As we can notice from Table 4, on the problems where the graphs have 9 to 15 nodes, the GA was not able to find the solution under any of the crossover operators. We have performed new experiments where the population size went up to 200 individuals. The graphs with more than 12 nodes were still too hard, but for the others, the optimal solution was found several times. With the new experiments, we have classified the operators by the number of times they found the optimal solution. We have considered that the best run for each operator was the one where the optimal solution was found with the least effort, computed as the population size multiplied by the number of generations, measure inspired from GP (Koza, 1992).

Table 4: HC Results

PROBLEM	MEAN TOP	MEAN 2ND	BEST RUN TOP	BEST RUN 2ND
HC4	dissociated	uniform	all (10)	
HC5	2-points	dissociated	all	
HC6	uniform	2-points	uniform (10)	dissociated (50)
HC7	dissociated	2-points	1-point (50)	2-points (100)
HC8	dissociated	uniform	**uniform (50)**	dissociated
HC9	dissociated	2-points	1-point	uniform
HC10	dissociated	1-point	dissociated	uniform
HC11	dissociated	uniform	uniform	2-points
HC12	1-point	dissociated	uniform	dissociated
HC13	2-points	1-point	2-points	1-point
HC14	uniform	dissociated	uniform	dissociated
HC15	uniform	2-points	2-points	1-point

Table 5 presents these new results in the same form. For the mean run, the parentheses contain the number of times that the operator has found the optimal solution. For the best run, the parentheses contain the minimal computational effort (population size multiplied by the number of generations) needed by the operator to find an optimal solution. The dissociated crossover still performs well according to these new measures, but is surpassed by the 2-points crossover.

Table 5 Results on Hard HC Problems

PROBLEM	MEAN TOP	MEAN 2ND	BEST RUN TOP	BEST RUN 2ND
hc9	2-points (13)	1-point (9)	2-points (176*30)	dissociated (100*98)
hc10	uniform (3)	2-points (3)	uniform (176*163)	2-points (176*109)
hc11	dissociated (1)	-	dissociated (90*738)	-

4. DISCUSSION

The results presented in Tables 1 to 5 suggest that the new crossover operator is an efficient one. We will first discuss the reasons for our optimism, and then some of its disadvantages we have found.

4.1 ADVANTAGES OF THE DISSOCIATED CROSSOVER

The crossover operator presented in this article, seams to perform quite well on the whole. To justify this affirmation, Table 6 presents the percentage of occurrence of each operator on each of the four positions we have considered in Tables 1 to 5.

Table 6. General Classification of the Operators

OPERATOR	MEAN TOP	MEAN 2ND	BEST RUN TOP	BEST RUN 2ND
dissociated	66.67%	9.09%	40.62%	21.88%
1-point	6.06%	45.45%	15.62%	15.62%
2-points	15.15%	30.31%	9.38%	25.00%
uniform	12.12%	15.15%	34.38%	37.50%

We can see that the dissociated crossover has the best percentage of occurrences in the top both for the mean run and for the best run. It shows a particularly good mean performance, where it occupies the first place two thirds of the time.

The 1-point crossover, that is a conservative one, seams to be the worst choice. The 2-point crossover performs quite well for the mean runs, but shows poor results for the best runs. The uniform crossover seams good both for the mean runs and for the best runs. The uniform crossover, which is more exploratory than the 2-point crossover, presents less performance on the average than it, but has more chances for a good best-case performance.

A second advantage of the dissociated crossover is the fast fitness evolution. This feature is obvious even when our operator did not perform better than the others. To illustrate this statement, Figures 4 to 7 show the performance evolution during the first 100 generations for two functions where the dissociated crossover was on top (F8 and Peak6'), and two others where it was not (Peak6 and HC14). All graphics concern the mean runs. This can be particularly interesting when the problem size is large and the genetic algorithm has difficulties in increasing the performance.

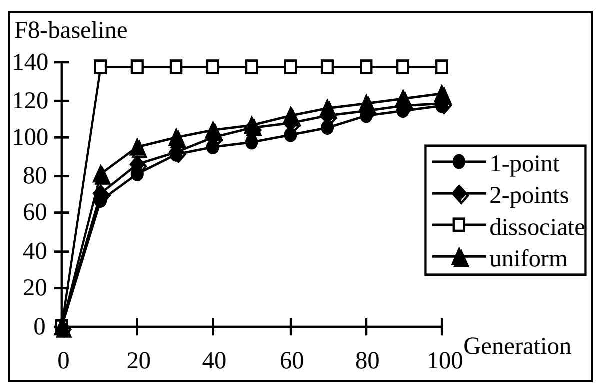

Figure 4. Mean Run for F8

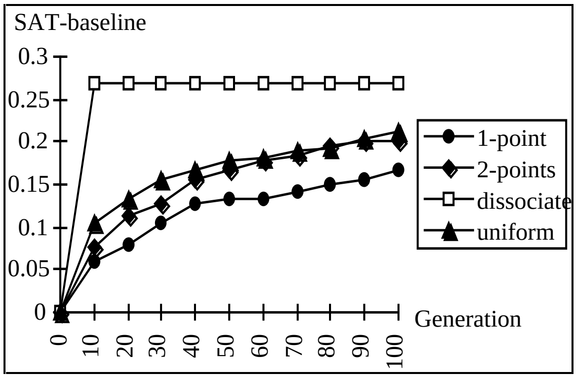

Figure 5. Mean Run for Peak6

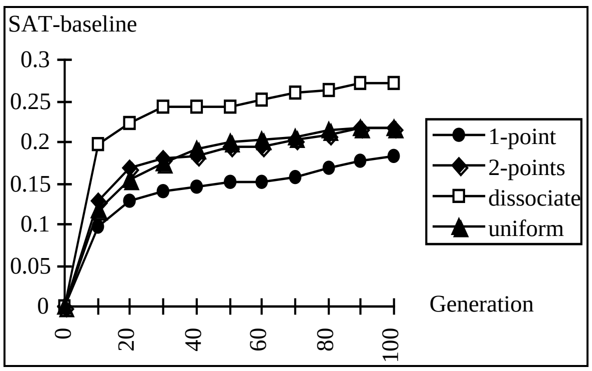

Figure 6. Mean Run for Peak6'

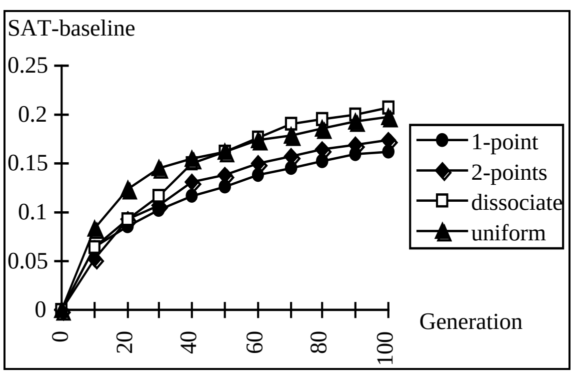

Figure 7: Mean Run for HC14

Last, but not least, with this operator the genetic population can only converge to the individual having only 0 gene values. Thus, the diversity of the population is conserved even after an important number of generations.

4.2 DISADVANTAGES

The last advantage of the dissociated crossover can also be a possible source of problems. Whenever an individual containing large intervals of 0 values is a suboptimal solution and local peak, the GA may fail to find the optimal solution.

We have encountered this situation for the SAT fitness landscape where the problems present suboptimal solutions with 15 to 30 adjacent genes having the 0 value. These problems are hard for all genetic operators, but they seam to disadvantage the dissociated operator more than the others.

The reshaped SAT problems (see Table 3) have up to 6 adjacent genes with the 0 value in the suboptimal peaks and do not present difficulties for the dissociated crossover. This also seams to be the threshold after which continuous 0 intervals, in a suboptimal solution are difficult for the dissociated crossover.

This inconvenience can be corrected by increasing the mutation rate or by changing Equation (1) to avoid filling large intervals with the '0' value. For example, we could replace in Equation (1) the 0 value for the genes between $crossSite_1$ and $crossSite_2$ with either a logical operation between $parent_1(i)$ and $parent_2(i)$, like 'and' or 'xor', or just a random choice between 0 and 1. The experiments can also be run with a convergence-avoiding clause, like the one suggested by Eshelmann (1991).

5. CONCLUSIONS

Through this paper, we have compared the dissociated crossover with other classical crossover operators. A theoretical analysis motivates the choice of this operator. For the experiments, we have used several fitness functions that test the operators under various conditions. The results presented in Tables 1 to 5 show that the new crossover operator performs better than the others in many cases. From Table 4 we can also see that when the problem is difficult, there is no wonder solution, and the differences between the operators are not significant.

The experiments have also shown a disadvantage of the new operator. Still, the condition under which the dissociated crossover can deceive does not appear very often, which explains why most of the results are positive. The negative feature can be eliminated by various procedures, and is the main object of our future work.

Our experiments have shown that the best crossover operator can be different according to the fitness landscape. This means that for the evolutionary algorithms, there is no general good solution, and that the parameters and variations of an algorithm should be chosen according to the problem's needs.

Acknowledgements

This research was partially done at the University of Neuchâtel, Computer Science Department, CH-2000 Neuchâtel, Switzerland.

References

R. Belew (1992). *Paradigmatic over-fitting*. Genetic Algorithms - Digest, 6(18).

L. Davis (1991). Bit-Climbing, Representational Bias, and Test Suite Design. In Booker, L. & Belew, R. (eds), *Proceedings of International Conference on Genetic Algorithms*, 18-23.

K.A. De Jong (1975). *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems*. Doctoral dissertation, University of Michigan.

K. De Jong, and W. Spears, (1989). Using Genetic Algorithms to Solve NP-Complete Problems. *International Conference on Genetic Algorithms*, George Mason University, Fairfax (VA), 124-132.

K. De Jong, and W. Spears, (1990). An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms. *International Workshop on Parallel Problem Solving from Nature*, University of Dortmund, 38-47.

L. Eshelman (1991). The CHC Adaptive Search Algorithm. How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In Rawlins, G. (ed), *Foundations of Genetic Algorithms - 1*, Morgan Kaufmann, 265-283.

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

J. H. Holland (1975). *Adaptation in Natural and Artificial Systems*. Univ. of Michigan Press, Ann Arbor.

J. Kingdon, and L. Dekker (1995). The Shape of Space. *Proceedings of Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA '95)*, IEE, London, 543-548.

J. Koza (1992): *Genetic Programming: on the Programming of Computers by Means of Natural Selection*. Cambridge (MA): The MIT Press.

H. Maini, K. Mehrotra, C. Mohan, and S. Ranka (1994). Knowledge-based Nonuniform Crossover. *Complex Systems*, 8, 257-293.

M. Mitchell, S. Forrest, and J. H. Holland (1991). The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance. In *Toward a Practice of Autonomous Systems: Proceeding of the First European Conference on Artificial Life*, Cambridge (MA): The MIT Press.

C. K. Mohan (1998) Selective Crossover: Towards Fitter Offspring. *Symposium on Applied Computing (SAC'98)*, Atlanta (GA).

H. Mühlenbein, and D. Schlierkamp-Voosen (1993). Predictive Models for the Breeder Genetic Algorithm. *Journal of Evolutionary Computation*, 1(1), 25-49.

C. Ross, V. Eisele, J. Denton, J. Bins, C. Guerra, D. Whitley, A. Howe, and J. P. Watson (1998). The Traveling Salesrep Problem, Edge Assembly Crossover, and 2-opt. *PPSN-V*.

G. Saporta (1990). *Probabilités, analyse des données et statistique*. Editions Technip, Paris.

J. D. Schaffer, R. A. Caruana, L. Eshelman, and R. Das (1989). A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimisation. In Schaffer, J.D. (ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, 51-60, Morgan Kauffman.

W. Spears (1995). Adapting Crossover in evolutionary algorithms. *Proceedings of the Fourth Annual Conference on Evolutionary Programming*.

J. L. Sushil, and L. Gong, (1997). Augmenting Genetic Algorithms with Memory to Solve Traveling Salesman Problems. *Proceedings of the Joint Conference on Information Sciences*, Duke University, 108-111.

G. Syswerda (1989). Uniform Crossover in Genetic Algorithms. In Schaffer, J. D. (ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, San Mateo (CA).

D. Vrajitoru (1998). Crossover Improvement for the Genetic Algorithm in Information Retrieval. *Information Processing and Management*, 34(4), 405-415.

D. Whitley, K. Mathias, S. Rana, and J. Dzubera (1996). Evaluating Evolutionary Algorithms. *Artificial Intelligence*, 85, 245-276.