THE COMPARISON OF THE SPATIAL CROSSOVER VARIANTS

S. L. Sotnyk

Dniprodzerzhynsk State Technical University

ABSTRACT: The article compares the two different crossover variants aimed for genetic algorithm (GA) application connected with problems of encoding of nodes, points, and solution coordinates. The usual GA schemes provide work with chromosomes as with plain strings. The compared methods differ from classical GA by using information of spatial coordinates encoded by chromosomes while applying genetic operators.
KEYWORDS:

1. INTRODUCTION

Most problems emerging in science and technology are the optimization problems. Despite a vast number of theoretical researches in this field, there are still a lot of problems, which mathematical formulation is either very difficult for analytical solution, or is missing. Various heuristic algorithms including genetic algorithms (GA) find their application for these problems.

Genetic algorithms originated from natural reproduction processes of organisms in nature. These algorithms are certainly not the exact copy of natural processes, at least, because the processes themselves have not completely studied yet. For example, the probabilities of the different genetic operators' application for different genes and their different parts are not a constant [EGM05]. However, the author has not been familiar with the developments in GA field that imitate this process precisely. Moreover, in the author's opinion, modeling the methods which precisely copy natural processes is not absolutely necessary. In many cases it's enough to use the basic concepts of the natural process, which the algorithm is based on as far as the execution of GA is performed with another "elemental" base.

Such crossovers' variants imitating the nature in general only use the spatial location of the points (nodes) encoded in genes. The common genetic algorithm spends an essential part of its energy for selecting the suitable location of encoded coordinates (points, nodes, objects) in genes. Usually, it breaks the established ensembles of the points, which leads to a very big number of "rejects". Of course, the GA-researcher wants to "help", to "prompt" the additional information to the algorithm that will let it work without a phase of searching the suitable coding system. This reason impelled the authors of the works [CP06, Sot07] to create the modifications of the basic genetic operator which work with coordinates more carefully.

The crossover variants, that let the genetic algorithm use the additional information about the nature of the problem in case of encoding the spatial coordinates, are described in papers [CP06] and [Sot07]. This article deals with the efficiency comparing of both algorithm variants using a model problem.

The parameters referring to one point (in the first place, the coordinates) are the linked (in genetic meaning) characters in both crossover variants. It means that the partition of the genotype into the elements is not performed within the data of one point, but only on the bounds between them.

2. SPATIAL CROSSOVER 1 (CHERBA AND PUNCH)

The crossover described in work [CP06] is as follows.

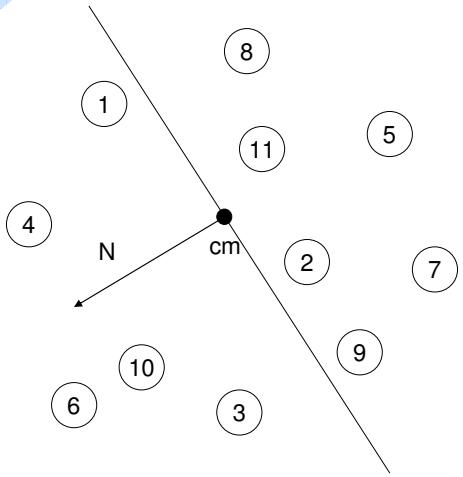

Figure 1: Spatial crossover Cherba and Punch for plane

It's suggested to draw a hyperplane (in two-dimensional case it is a line) through the centre of the points' mass at a random angle. This angle is equal for both parents. Then this plane has to be moved so that it would divide the points in two equinumerous sets (up to a point). The search of such location is performed independently for each of the parents; only the direction of the normal is preserved. After that the combination of the parents' solutions is performed as follows:

$$P = P_p \cup P_n, \tag{1}$$

where P – the set of all points, P_P – the set of points having positive coordinates when projecting to the normal, P_n – the set of points having negative coordinates when projecting to the normal.

$$\forall p_i \in P \left| p_i \circ V_n \geq 0 \quad P_p = P_p \cup p_i \right. \tag{2}$$

$$\forall p_i \in P \left| p_i \circ V_n < 0 \quad P_n = P_n \cup p_i \right. \tag{3}$$

Here are p_i-points encoded in the genome, V_N – the normal vector to the hyperplane dividing the parent's points into two sets. The result of the sets' union ($\tilde{P}$ – descendant) is built as the union of P_p sets of one of the parents (A) and P_n sets of the other (B).

$$\tilde{P} = P_p(A) \cup P_n(B) \tag{4}$$

It's easy to notice that such a division process of the solution points into equal sets (up to a point) results in the fact that the number of points (the genome length) is kept constant from generation to generation. It's obvious that, the descendents inherit already established complexes of points, which are not mostly divided into parts more frequently than in a common crossover. At the same time the nature of the points' ensembles transferring is rather limited – the central ensembles are broken much more frequently than on the boundaries. Also, the peripheral ensembles are usually broken in radial direction relative to the mass centre. It reduces some searching abilities of this crossover type – for example, the redistribution of the points' density becomes more complicated.

3. SPATIAL CROSSOVER 2

This crossover is described in detail in work [Sot07]. Let's review the main points briefly:
While the simple two-point crossover uses the points' location in genome, such crossover type performs the selection using the points' location in the space of solution. So, the descendant's synthesis process is as follows.
There are two sets of the base points (the parents).

$$\Psi^K = \left(\psi_1^K, \psi_2^K \ldots \psi_n^K\right), \quad k = 1,2 \tag{5}$$

Let's introduce a classifier that allows dividing Ψ^K sets into two non-overlapping sets of the element indexes, which combination gives the original sets:

$$\begin{aligned} &I^K = (1 \ldots n), \\ &I_1^K \in I^K, I_2^K \in I^K, \\ &I_1^K \cap I_2^K = \varnothing, I_1^K \cup I_2^K = I^K, \\ &I_1^K = I_1^K(\Psi), I_2^K = I_2^K(\Psi), \quad k = 1,2 \end{aligned} \tag{6}$$

The following classifier was used in this work (though, of course, it's just one of the possible classifiers):
Let $\left(\psi_1^K \ldots \psi_n^K\right)$ be the data sets that, besides all, also contain the information about the location of the points in space.
Let's attribute the indexes corresponding to the points lying inside the sphere with radius R and with the centre in point P to I_1^K set. R is specified randomly over the range 0 to D, where D is the diameter of the set of possible points for the problem being solved. P is a randomly chosen point lying in the range permissible for the problem (it's usually a random point from the set of admissible solution points).
We'll attribute the indexes corresponding to the points lying outside the hypersphere with radius R and the centre in point P to I_2^K set.
We can get a new set of points (descendant) by joining the points (to be more precise, the parts of the genes containing the coordinates of one point and the information linked with this point) of the parents.

$$Z = \left(\psi_i^1 \in \Psi^1 : i \in I_1^1\right) \cup \left(\psi_i^2 \in \Psi^2 : i \in I_2^2\right) \tag{7}$$

In case of mes Z > n, then remove from Z randomly chosen elements, in number mes Z – n.
If mes Z < n, then replenish Z set with the unused points of the parents.

$$\begin{aligned} Z := Z \cup \{&\left(\psi_i^1 \in \Psi^1 : i \in I_2^1\right) \vee \\ &\vee \left(\psi_i^2 \in \Psi^2 : i \in I_1^2\right)\} \end{aligned} \tag{8}$$

The replenishment procedure is repeated until equality mes Z = n is fulfilled.
The spatial crossover scheme 2 (for two-dimensional case) is graphically displayed in figure 2.
The correction stage takes place after the synthesis stage shown in figure 1. It is necessary, because general number of genes in the parents and in the descendant will not coincide in most cases for such a joining algorithm. That's why if we want the equal number of the points to remain from generation to generation, we remove the unnecessary points or add the unused ones.

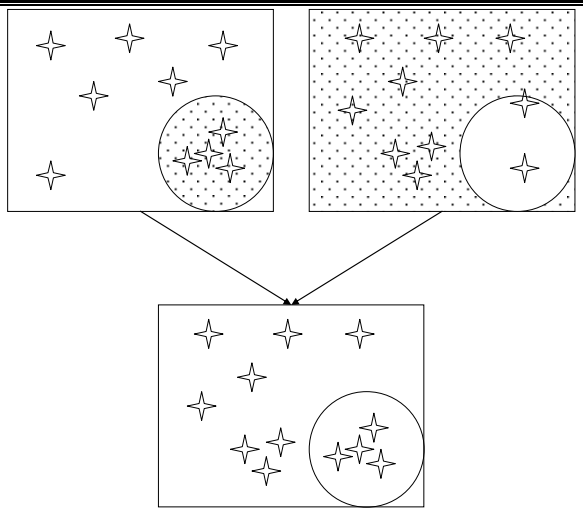

Figure 2: Graphical representation of the spatial crossover introduced in work [Sot07]

4. THE COMPARISON

To check the effectiveness of the introduced algorithms of the genetic material joining, the corresponding utility in C# language (.NET Framework platform) was created. The utility allows comparing the GA working with a two-point and spatial crossover (both variants) on the same problem. While testing, genes encoding was performed in correspondence with the scheme of continuous genetic algorithms [HLV98, HLS05, Wri91, DK95].

The problem was in the following. There is RGB pattern picture. It's necessary to reconstruct it as truly as possible using the limited number of the basis points. The points that are not the basis ones assume the colour of the nearest basis point.

Minkowski distance was chosen as the proximity measure of the points for spatial crossover 2 (9).

$$d = |x_1 - x_2| + |y_1 - y_2| \qquad (9)$$

It's clearly noticeable that for the chosen metric, the cut "window" from figure 2 will be diamond-shaped. In this case L_1 metric was chosen because the preliminary experiments didn't show any substantial differences between the speed and the quality of the algorithm's working when using L_1 and L_2 (Euclidean equation), and L_1 is simpler in calculating respect.

The description given in work [CP06] proved to be inconvenient for practical realization. That's why instead of the described process of the hyperplane shifting along the normal, the normal construction was only performed with the following projection of the points to it by the formula of coordinate system turning (affine transformations' subset). We get formula (10) for two-dimensional case.

$$n = x\cos(\alpha) + y\sin(\alpha), \qquad (10)$$

where x, y are corresponding coordinates of the solution point, α is the angle of the normal vector relative to the axis x, n is a coordinate of the point projection to the normal.

After projecting all the points to the normal (for each parent separately), the points' sorting was made by the projection coordinates, and the joining of half the points of one parent and another half from the other one was performed. Such an algorithm is equal to the one described in [CP06], and at the same time it's more convenient and efficient.

The reconstruction error of the image in space L_2 was taken as the measure of the solution's quality (11).

$$e = \sum_{x=1}^{Width} \sum_{y=1}^{Height} \sum_{c=1}^{3} \left(Y_{x,y,c} - \tilde{Y}_{x,y,c}\right)^2, \qquad (11)$$

where $Y_{x,y,c}$ and $\tilde{Y}_{x,y,c}$ are the brightness of the colour component in the point with coordinates (x, y) in the original and reconstructed images correspondingly.

While reconstructing, GA was started with the same settings. The only difference was the choice of one or another crossover variant. It was found out beforehand that (for evolution of 500 generations and population of 100 individuals) the optimal settings of the crossover probability are within p_C=0.5, and the mutation probability is p_M=0…0.1. These values are equally good as for the usual two-point crossover as for the spatial one.

The reconstruction error on the graphs is recounted by formula (12) for visual perception convenience.

$$e = \sqrt{\sum_{x=1}^{Width} \sum_{y=1}^{Height} \sum_{c=1}^{3} \left(Y_{x,y,c} - \tilde{Y}_{x,y,c}\right)^2}. \qquad (12)$$

The average results (after five experiments made with the same settings) of reconstruction for two different images shown in figure 3 are represented below (figures 4, 5). One test image (Baboon) is the example where the basis points should have comparatively equal distribution on the plane. Another one (Synthetic) has as extensive areas where the brightness of the points is practically the same as the areas with tiny details and halftone transitions, which means uneven distribution of the nodes on the plane for suboptimal solutions.

The data of the image reconstruction with the help of the usual two-point crossover are also given for comparison.

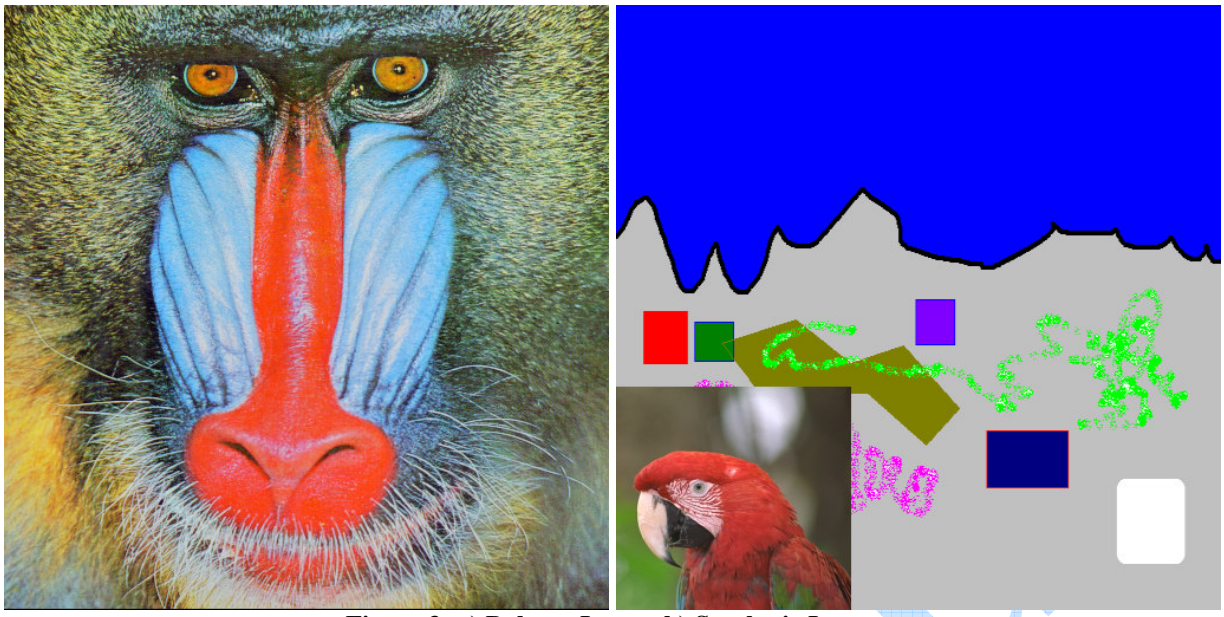

Figure 3: a) Baboon Image, b) Synthetic Image

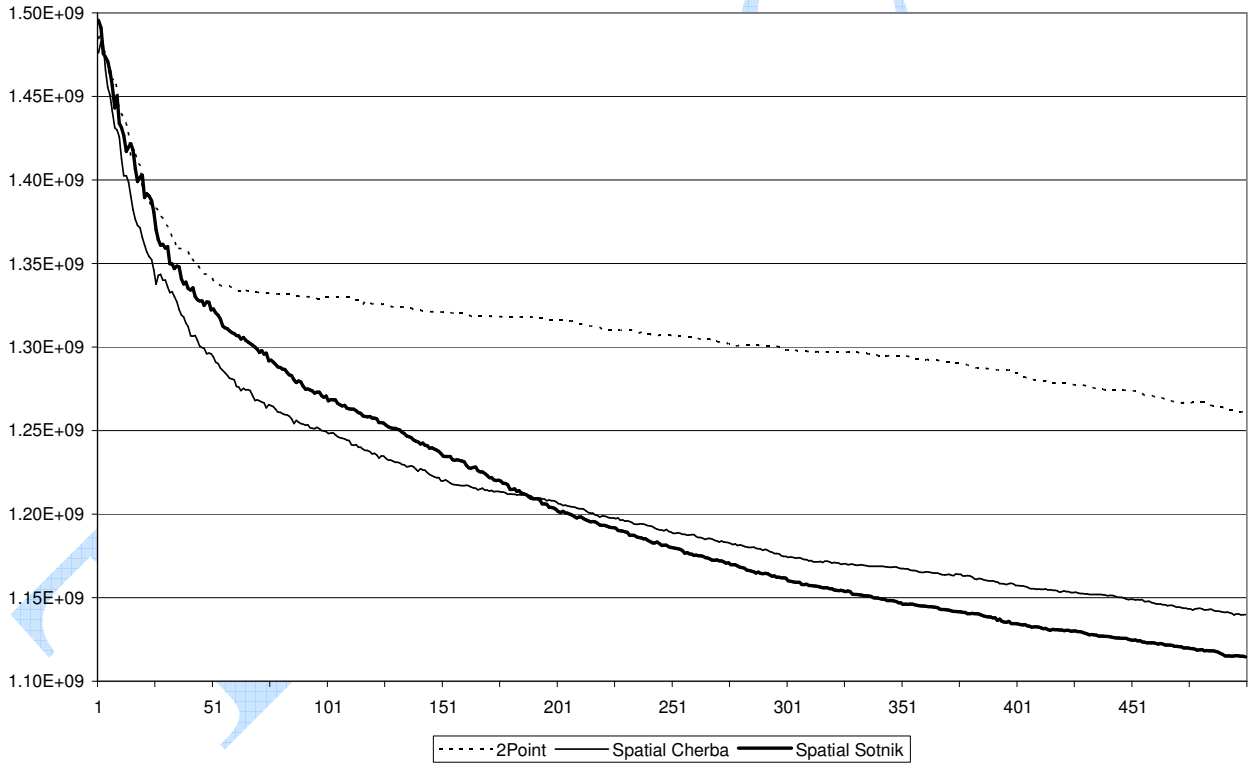

Figure 4: Baboon Image Reconstruction. X axis is a generation, Y axis is an error. The lower it is, the less the error is (better results)

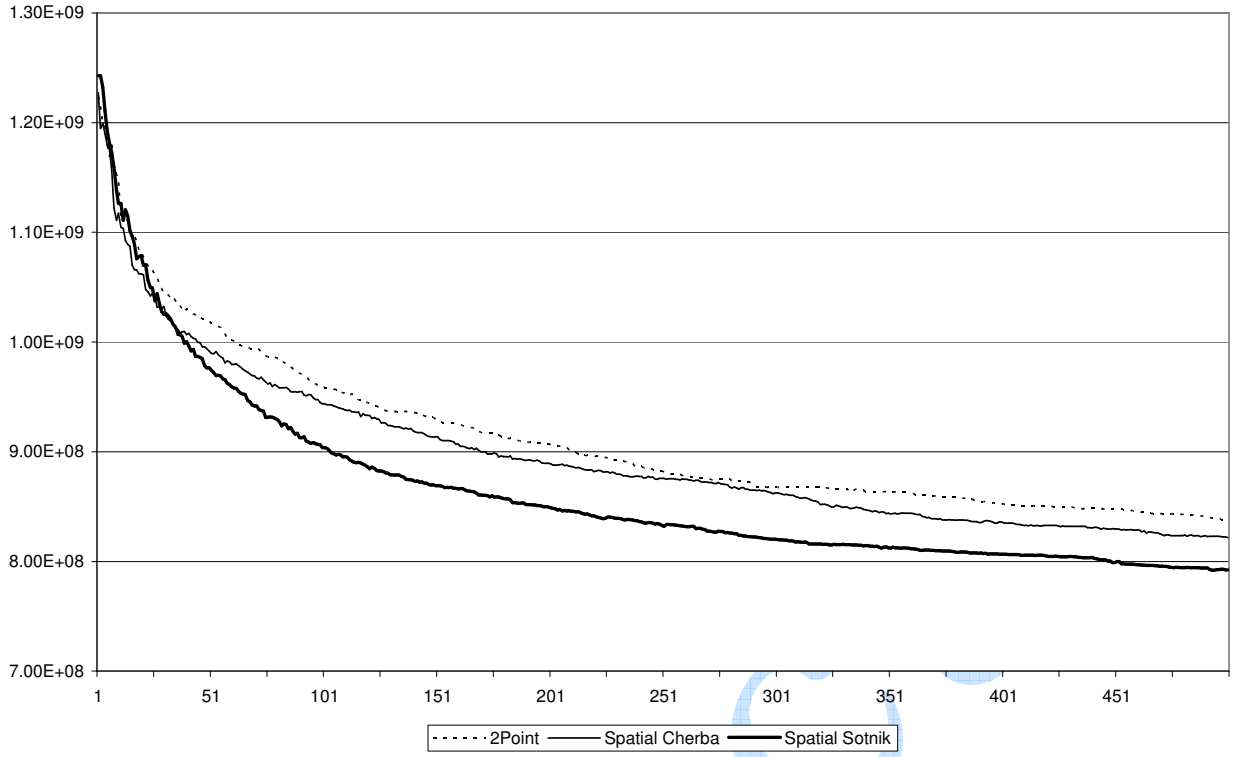

Figure 5: Synthetic Image Reconstruction

5. CONCLUSION

Both spatial crossover modifications provide better effectiveness in comparison with the two-point crossover in the problems where the nodes' spatial location is searched for. Higher effectiveness here means the searching of solution with the same quality for less number of generations, or getting a more successful solution in the same generation. For most problems being solved with help of GA the most labour-consuming (difficult) part is the procedure of quality estimation of the obtained solution, which lets not pay attention (within reasonable limits, of course) to complication of the descendants' obtaining procedure.

When comparing the results obtained with the help of the different spatial crossover variants, the following regularity occurs – a spatial crossover [CP06], as a rule, starts faster, but then variant [Sot07] is gradually catching up with it and begins to show better results. It is especially noticeable for the images requiring substantial redistribution of the nodes' density.

The explanation for this result is the following. Crossover [CP06] practically always combines genetic material of both parents in large blocks. The division hyperplane practically always goes through the centre of the solution region. Such an approach lets quickly and roughly fit the first approximations. But later, it's necessary to fit the solutions by smaller groups. Algorithm [Sot07] makes it better.

Such results let us make the following assumption. To get more efficient algorithm we can select the sizes of cut sections in parents adaptively in corresponding with the generation's number. This hypothesis will be checked by the author in the nearest future.

REFERENCES

[EGM05] **P. D. Evans, S. L. Gilbert, N. Mekel-Bobrov et al.** - *Microcephalin, a Gene Regulating Brain Size, Continues to Evolve Adaptively in Humans*. Science v. 309, 9 September 2005.

[CP06] **D. M. Cherba, W. F. Punch** - *Crossover gene selection by spatial location*. GECCO 2006: 1111-1116

[Sot07] **S. L. Sotnyk** - *The spatial crossover for the problems of nodes' choice optimization using the genetic algorithm*, "Mathematical modeling" journal - 1(16) 2007 – in Russian.

[HLV98] **F. Herrera, M. Lozano, J. L. Verdegay** - *Tackling real-coded genetic algorithms: operators and tools for the be-*

haviour analysis, Artificial Intelligence Review, Vol. 12, No. 4, 1998. – P. 265-319.

[HLS05] **F. Herrera, M. Lozano, A. M. Sanchez** - *Hybrid Crossover Operators for Real-Coded Genetic Algorithms: An Experimental Study*, Soft Comput. 9(4): 280-298 (2005).

[Wri91] **A. Wright** - *Genetic algorithms for real parameter optimization,* Foundations of Genetic Algorithms, V. 1. – 1991. – P. 205-218.

[DK95] **K. Deb, A. Kumar** - *Real-coded genetic algorithms with simulated binary crossover: Studies on multi-modal and multi-objective problems*. Complex Systems, 9(6), 431—454, 1995.