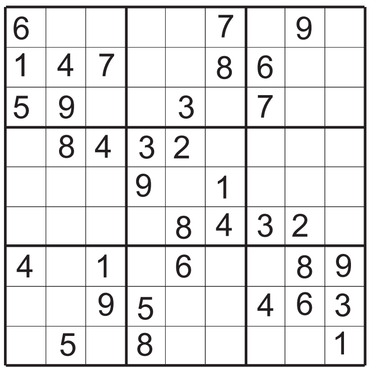
On the Effects of Locality in a Permutation Problem: The Sudoku Puzzle

Edgar Galván-López and Michael O'Neill

Abstract— **We present an analysis of an application of Evolutionary Computation to the Sudoku Puzzle. In particular, we are interested in understanding the locality of the search operators employed, and the difficulty of the problem landscape. Treating the Sudoku puzzle as a permutation problem we analyse the locality of four permutation-based crossover operators, named One Cycle Crossover, Multi-Cycle Crossover, Partially Matched Crossover (PMX) and Uniform Swap Crossover. These were analysed using different crossover rates. Experimental evidence is found to support the fact that PMX and Uniform Swap Crossover operators have better properties of locality relative to the other operators examined regardless of the crossover rates used. Fitness distance correlation, a well-known measure of hardness, is used to analyse problem difficulty and the results are consistent with the difficulty levels associated with the benchmark Sudoku puzzles analysed.**



6					7		9	
1	4	7			8	6		
5	9			3		7		
	8	4	3	2				
			9		1			
				8	4	3	2	
4		1		6			8	9
		9	5			4	6	3
	5		8					1

Fig. 1. Example of Sudoku problem.

I. Introduction

The concept of a fitness landscape was first introduced in biology by Wright [17]. This concept has dominated the way geneticists think about biological evolution and has been adopted within the Evolutionary Computation (EC) community. In simple terms, a fitness landscape can be seen as a plot where each point on the horizontal axis represents all the genes in an individual corresponding to that point. The fitness of that individual is plotted as the height against the vertical axis. Thus, a fitness landscape is a representation of a search space which may contain peaks, valleys, hills and plateaus. How an evolutionary algorithm explores and exploits a fitness landscape is a key element.

In [13], [14], [15] the authors expressed that the locality of a representation describes how well neighbouring genotypes corresponds to neighbouring phenotypes. The authors pointed out that a representation presents high locality if all neighbouring genotypes correspond to neighbouring phenotypes. On the other hand, a representation presents low locality if some neighbouring genotypes do not correspond to neighbouring phenotypes. The authors also mentioned that if a representation has high locality, then we should expect to see a good performance in evolutionary search whereas the opposite is true when a representation has low locality.

In his studies, Rothlauf considered the relationship between genotype-phenotype mapping and the phenotype-fitness mapping to study locality. In this paper we study locality by considering the fitness of individuals. To do so, we will use many Sudoku puzzles with different levels of difficulty (i.e., classified from the source taken[1]).

The paper is organised as follows. In the following section, the Sudoku puzzle is presented. Section III presents a literature review on Locality. In Section IV we describe the operators used on the Sudoku problem to study locality. Section V describes fitness distance correlation, a well-known measure of hardness. Section VI shows the experimental setup used to conduct our research. In Section VII we present and discuss our findings and Section VIII draws some conclusions.

II. Sudoku problem

Sudoku is an immensely popular, originally Japanese number placement puzzle, the subject of hundreds of books and featuring in countless newspapers all over the world. The game is played on a 9 by 9 grid, where each of the 81 squares can be either blank or filled with an integer between 1 and 9. In Figure 1, we present a typical example of the Sudoku problem. (Other sizes are possible, as are variations on the rules but none of this will be discussed here.) The grid starts mostly empty, and is filled with numbers by the player. The objective is to complete a Latin square, so that all rows and columns contain exactly one each of the integers 1 to 9, and that each of the 9 sub-squares also contains one of each integer. In other words, when a number is placed in a position, the same number cannot occur in the same row, column, or sub-square. The problem would be simple enough if it was not for the presence of a number of fixed positions which cannot be altered by the player. These fixed numbers define a particular grid and determine the difficulty of it but not in a straightforward way: the number of fixed positions

Edgar Galván-López and Michael O'Neill are with the University College Dublin, Natural Computing Research & Applications Group, UCD CASL, 8 Belfield Office Park, Beaver Row, Clonskeagh, Dublin 4, email: `edgar.galvan, m.oneill@ucd.ie`).

[1]http://www.websudoku.com/.

have little or no relation to the difficulty, instead certain fixed positions are more difficult than others. Importantly, each valid Sudoku grid has a unique solution. So, we are in the presence of multimodal landscape with only one global optimum.

All 9 by 9 Sudoku grids that are intended to be solvable by humans can be solved within seconds by computers, using standard constraint satisfaction methods (e.g. Knuth's "Dancing links" algorithm). Thus, while the Sudoku problem in general can potentially be very challenging, we are not aiming to compete with established non-evolutionary methods, at least not for the standard sizes of Sudoku.

To calculate the fitness of each individual in the population, we used the same fitness function described in [5], [9]. The fitness function simply counts whether each integer is present in each row, column and box. That is, for a $9x9$ grid we have a maximum fitness of $9*9+9*9+9*9$ = 243, so a fitness can yield between 0 and 243 (this being the global optimum). Because of the nature of the problem, we decided to treat it as a permutation problem so, we will use only operators based on permutations. This will be explained in detail in Section IV. In the following section we present and discuss the notion of locality.

III. Locality

In [13], [14], [15], Rothlauf mentioned that the understanding of how well neighbouring genotypes corresponds to neighbouring phenotypes is a key element in evolutionary search.

To study locality, it is necessary to define a metric on the search space Φ. In a genotype-phenotype mapping representation, it is clear that we are in the presence of two search spaces, where Φ_g is the genotypic search space and Φ_p is the phenotypic search space. Now, based on a defined metric we can quantify how different or similar two individuals are. In his work, Rothlauf mentioned that for two different search spaces (e.g., genotypic and phenotypic search space) it is necessary to define two different metrics. In our work we will use the same notion, but at the genoytpe-fitness level. In Section IV, we will further discuss this.

The author distinguished two types of locality: low and high locality. The author pointed out that a representation presents high locality if all neighbouring genotypes correspond to neighbouring phenotypes. On the other hand, a representation presents low locality if some neighbouring genotypes do not correspond to neighbouring phenotypes. The author also mentioned that a representation that has high locality is necessary for an efficient evolutionary search.

Rothlauf mentioned that if a representation shows high locality, then any search operator has the same effects in both the genotype and phenotype space. It is clear then that the difficulty of the problem remains unchanged. According to Rothlauf, having high locality is sufficient for an efficient search.

This, however, changes when a representation has low locality. To explain how low locality affects evolution, Rothlauf considered three different categories of hardness (these categories were taken from a well-known measure of hardness presented in [7]. In Section V, we will discuss it in detail). These categories are:

- *easy*, in which fitness increases as the global optimum approaches,
- *difficult*, for which there is no correlation between fitness and distance and
- *misleading*, in which fitness tends to increase with the distance from the global optimum.

If a given problem lies in the first category (i.e., easy), a low locality representation will change this situation by making it more difficult and now, the problem will lie in the second category. According to the author, this is due to low locality randomises the search. To explain this, Rothlauf mentioned that representations with low locality lead to uncorrelated fitness landscapes so, it is difficult for heuristics to extract information.

If a problem lies in the second category, this type of representation does not change the difficulty of the problem. The author pointed out that there are representations that can convert a problem from difficult (class two) to easy (class one). However, according to the author, there are only few representations that have this effect. The same happens for problems lying in class three.

Finally, if the problem lies in the third category, a representation with low locality will transform it and now, the problem will lie in the second category. That is, the problem is less difficult because the search has become more random. As it can be seen, this is a mirror image of a problem lying in the first category and using a representation that has low locality.

In this paper, we will focus our attention on locality of a permutation-based landscape as realised in the Sudoku puzzle, and we will measure locality by paying attention to the effects that changes at genotype level have on the fitness of solutions. Changes at the genotype-level arise due to the crossover operators employed, in this case they are permutation-based operators, named Partially Matched Crossover, One and Multi-cycle Crossover, and Uniform Swap Crossover. These will be introduced and explained in Section IV. When we apply one of these permutation operators we can measure the swapping distance that occurs. Similarly we measure the change in fitness for each swapping distance that can occur. To do both things, we take into account the best individual of the population per generation. We can then determine experimentally the impact that different swap distances have on the change in fitness. In a representation with high-locality we would expect that the smaller genotypic swapping distances correspond with smaller fitness distances, and by extension that larger genotypic swap distances result in proportionately larger fitness distances. In Section VII we will see what kind of locality each of the operators present and how this affects evolutionary search.

IV. Genetic Operators

We will be treating each individual in the population as a sudoku grid (more details about the implementation will be further discussed in Section VI). Figure 1 depicts this idea. We have initialised our individuals by filling the empty squares with the remaining missing numbers, in the range 1 to 9, per row.

Given these conditions, it is clear that the sudoku puzzle can be treated as a permutation/swap problem. So, for comparison purposes and to study the type of locality present in this problem, we have used four different types of crossover for permutations, named one cycle crossover, multi cycle crossover [10], uniform swap crossover and finally, Partially Matched Crossover (PMX) [6]. These work as follows:

- One cycle crossover. This operator works as follows:
  1) Determine the cycle that is defined by the corresponding positions of elements between two parents.
  2) Copy the elements defined in the cycle to an offspring with the positions corresponding to those of the parent.
  3) Determine the remaining elements for the offspring by deleting those symbols that are already in the cycle from the other parent.
  4) Finally, fulfill the offspring with the remaining elements. Figure 3 illustrates the fours steps involved in Cycle crossover.
- Multi cycle crossover. This type of crossover is very similar to one cycle crossover. The main difference is that multi cycle crossover considers all possible cycles.
- Uniform swap crossover. Uniformly (50%) swaps elements, ensuring that the resulting offspring does not have an element repeated.
- PMX crossover. This operator works as follows:
  1) Select a substring uniformly in two parents at random. These substrings are called mapping sections.
  2) Exchange these two substrings to obtain a partial offspring.
  3) Determine the mapping relationship between two mapping sections.
  4) Finally, legalise the offspring with the mapping relationship. Figure 2 shows the four steps involved in PMX crossover.

In his studies, Rothlauf defined locality within the frame of fitness distance correlation. This measure of hardness will be presented in the following section.

V. Fitness distance correlation

In [7], [8], Jones proposed an heuristic called *fitness distance correlation* (fdc) using the typical GA representation (i.e., the bitstring representation) and successfully tested it in several problems.

The idea of using *fdc* as an heuristic method, as stated in [7], [8], was to create an algebraic metric that can give enough information to determine the difficulty (for a GA) of a given problem when the global optimum is known in advance. To achieve this, Jones explained that it is necessary to consider two main elements:

1) To determine the distance between a potential solution and the global optimum (when using a bitstring representation, this is accomplished by using the Hamming distance) and
2) To calculate the fitness of the potential solution.

With these elements in hand, one can easily compute the *fdc* coefficient using Jones' calculation [7] thereby, in principle, being able to determine in advance the hardness of a problem.

The idea behind *fdc* was to consider fitness functions as heuristic functions and to interpret their results as indicators of the distance to the nearest optimum of the search space. *fdc* is an algebraic measure to express the degree to which the fitness function conveys information about distance to the searcher.

The definition of *fdc* is quite simple: given a set $F = \{f_1, f_2, ..., f_n\}$ of fitness values of n individuals and the corresponding set $D = \{d_1, d_2, ..., d_n\}$ of distances of such individuals from the nearest optimum, *fdc* is given by the following correlation coefficient:

$$fdc = \frac{C_{FD}}{\sigma_F \sigma_D}, \tag{1}$$

where:

$$C_{FD} = \frac{1}{n}\sum_{i=1}^{n}(f_i - \overline{f})(d_i - \overline{d})$$

is the covariance of F and D, and σ_F, σ_D, $\overline{f}$ and $\overline{d}$ are the standard deviations and means of F and D, respectively. The n individuals used to compute *fdc* are obtained via some form of random sampling.

According to [7] a problem can be classified in one of three classes, depending on the value of fdc:

1) *misleading* ($fdc \geq 0.15$), in which fitness tends to increase with the distance from the global optimum;
2) *difficult* ($-0.15 < fdc < 0.15$), for which there is no correlation between fitness and distance; and
3) *easy* ($fdc \leq -0.15$), in which fitness increases as the global optimum approaches.

The threshold interval [-0.15, 0.15] was empirical determined by Jones. In [7], Jones also proposed to use scatter plots (distance versus fitness) when *fdc* does not give enough information about the hardness of a problem.

There are some known weakness in the *fdc* as a measure of problem hardness [1],[12]. However, it is fair to say that the method has been generally very successful, including the use of tree-like structure representation [2], [3], [4], [11], [16].

This measure of hardness will help us to determine the difficulty of solving a given sudoku puzzle. We will further discuss this in Section VII.

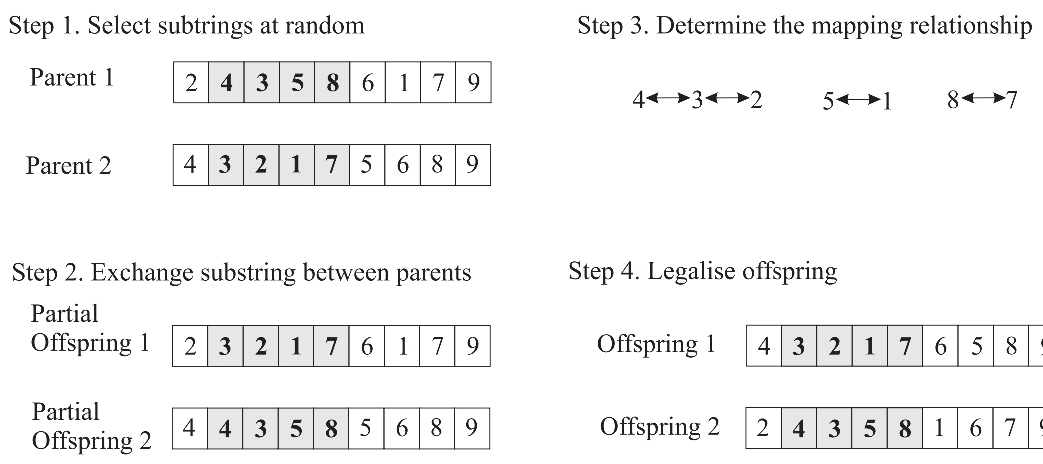


Fig. 2. PMX Crossover, step by step.

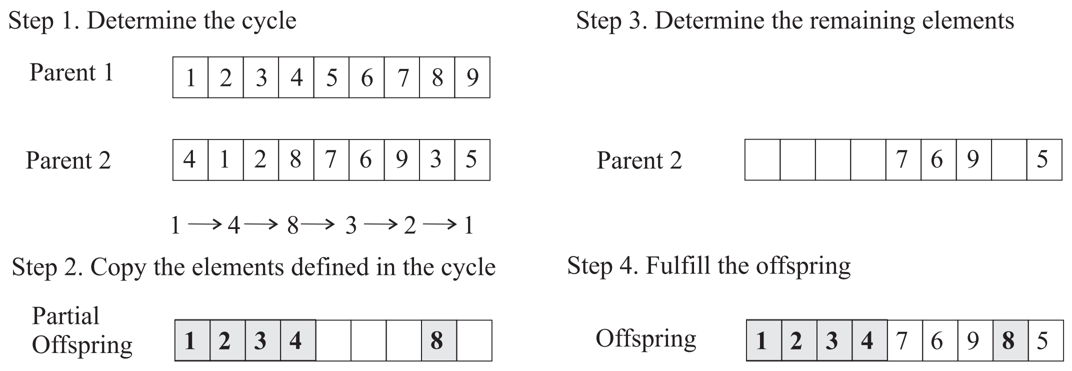

Step 1. Determine the cycle
Parent 1
1 2 3 4 5 6 7 8 9
Parent 2
4 1 2 8 7 6 9 3 5
1 → 4 → 8 → 3 → 2 → 1
Step 3. Determine the remaining elements
Parent 2
7 6 9 5
Step 2. Copy the elements defined in the cycle
Partial Offspring
1 2 3 4 8
Step 4. Fulfill the offspring
Offspring
1 2 3 4 7 6 9 8 5

Fig. 3. Cycle Crossover, step by step.

TABLE I
PARAMETERS USED TO CARRY OUT OUR EXPERIMENTS.

Parameter	*Value*
Length of the genome	81
Population Size	1000
Generations	100
Crossover Rates	1, 0.5
Tournament Size	7
Independent Runs	100

VI. Experimental Setup

We have used six different sudoku puzzles of different level of difficulty. These were taken from *http://www.websudoku.com/*. Two of them were classified as easy, two as medium and finally, two as difficult.

The experiments were conducted using a GA with tournament selection, run for 100 generations. To obtain meaninful results, we performed 100 independent runs for each of operators described in Section IV. Runs were stopped when the maximum number of generations was reached. The parameters we have used are summarised in Table I.

Each individual in the population was treated as a "sudoku shape". That is, an individual was treated as a two-dimensional array (Figure 1 depicts this idea). Each operator described in Section IV was applied per row. For each row, we filled them with the missing numbers in the range [1,9] at random. To do so, we took in consideration the fixed numbers.

VII. Results and Discussion

In Tables III, IV, V and VI we show the results for the one cycle crossover, multi cycle crossover, uniform swap crossover and PMX and setting the crossover rate at 0.5 (second column) and crossover rate at 1 (third column), respectively. In these tables we report the number of runs out of 100 independent runs that were able to solve the problem (i.e., fitness = 243) and those runs that got stuck in local optima (e.g., fitness = {239,241})[2].

Interestingly, *fdc* more less predicted the difficulty of the problem. That is, for the easy and medium problems the

[2]Note that the sum of these can be different of 100 because there are few runs stuck in other local optima, indicating that the sudoku problem is multimodal.

coefficient of *fdc* lies between -0.31 and -0.34 so, according to this measure of hardness, these categories should be equally difficult to be solved by the GA. However, this is not true as can be seen in Tables III, IV, V and VI, where it can be checked that for the medium difficulty level, the operators descrided in Section IV find those problems more difficult compared to the easy problems. Now, for the difficult problems, the coefficient of *fdc* varies from the easy and medium problem. In these problems, *fdc* increases indicating that the problem is in fact more difficult compared to easy and medium levels. This is in fact reflected in the results found by the GA as shown in the last two rows of Tables III, IV, V and VI .

From the results found by the operators, it is quite clear that the operators with best performance in terms of finding the solution are uniform swap crossover (see Table V) and PMX (see Table VI). The opposite happens when one cycle crossover (see Table III) and multi cycle crossover (see Table IV) are used.

We are interested in seeing how these results are obtained by the genetic operators within the frame of locality. From the results reported in Tables III, IV, V and VI , it is clear that the Sudoku problem is an interesting benchmark due to its features of multimodality. Also, it is interesting to see how most the runs end up at fitnesses 239, 241 (these being local optima) or at fitness 243 (global optimum). So, to analyse how locality is present in this problem and to do fair comparisons, we recorded 100 runs that end up at these values for the four crossover types presented and explained in Section IV and using an easy problem (i.e., Easy 1) and setting two crossover rates = $\{0.5, 1.0\}$.

In Figures 4 and 5, we report the swap distance from the best individual to the optimum solution (y axis) vs. fitnesses values (x axis), setting crossover rate at 0.5 and at 1, respectively. We show this relationship using one cycle crossover (top left of Figures 4 and 5) and multi cycle crossover (top right of Figures 4 and 5), where both operators showed poor performance for all the used sudoku problems. When we compare these plots with those using PMX crossover (bottom left of Figures 4 and 5) and uniform swap crossover (bottom right of Figures 4 and 5), we can see how there is a clear indication that the last two operators explore more points in the search space until finding the global solution. In other words, these operators perform a smoother search. This is even more evident if we focus our attention on the bottom right corner of each plot in both Figures 4 and 5. For instance, for one cycle crossover and multi cycle crossover, there are more points close to the global optimum compared to PMX and uniform crossover. What is also interesting to see is that by using different crossover rates for each of the operators used, the search performance remains almost the same. We can also see that the locality of the crossover operators remains the same by altering the crossover rates used in our experiments. For instance, we can see that high locality is present when PMX and uniform swap mutation (bottom of Figures 4 and 5) irrespective of the crossover rate used (i.e., 0.5, 1.0).

TABLE II
FITNESS DISTANCE CORRELATION.

Difficulty	*fdc*
Easy 1	-0.33
Easy 2	-0.34
Medium 1	-0.31
Medium 2	-0.34
Hard 1	-0.27
Hard 2	-0.23

VIII. CONCLUSIONS

How an evolutionary algorithm explores and exploits the search space is a key element in any EC paradigm. Rothlauf put forward the concept of locality: how well neighbouring genotypes corresponds to neighbouring phenotypes. The same principle can be applied at the genotype-fitness level. Rothlauf also mentioned that there are two forms of locality: high and low locality. It is believed that high locality is sufficient for an algorithm to efficiently explore the search space.

In this work, we have studied four different crossover operators, named One Cycle Crossover, Multi Cycle crossover, Partially Matched Crossover (PMX) and Uniform Swap Crossover and used two different crossover rates (i.e., 0.5 and 1.0). PMX and Uniform Swap Crossover, both showing high locality irrespective of the crossover rate used, shown to be efficient in finding the global solution on a multimodal landscape (many sudoku puzzles of different levels of difficulty).

ACKNOWLEDGMENTS

The authors would like to thank Julian Togelius for his previous collaboration. The authors would like to thank the NCRA group for their valuable comments on the paper. This publication has emanated from research conducted with the financial support of Science Foundation Ireland.

REFERENCES

[1] M. Clergue and P. Collard. GA-Hard Functions Built by Combination of Trap Functions. In D. B. Fogel, M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow, and M. Schackleton, editors, *CEC 2002: Proceedings of the 2002 Congress on Evolutionary Computation*, pages 249–254. IEEE Press, 2002.

[2] E. Galván-López. *An Analysis of the Effects of Neutrality on Problem Hardness for Evolutionary Algorithms*. PhD thesis, Department of Computing and Electronic Systems, University of Essex, United Kingdom, 2008.

[3] E. Galván-López, S. Dignum, and R. Poli. The Effects of Constant Neutrality on Performance and Problem Hardness in GP. In M. ONeill, L. Vanneschi, S. Gustafson, A. I. E. Alcazar, I. D. Falco, A. D. Cioppa, and E. Tarantino, editors, *EuroGP 2008 - 11th European Conference on Genetic Programming*, volume 4971 of *LNCS*, pages 312–324, Napoli, Italy, 26–28 Mar. 2008. Springer.

[4] E. Galván-López and R. Poli. Some Steps Towards Understanding How Neutrality Affects Evolutionary Search. In T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, editors, *Parallel Problem Solving from Nature (PPSN IX). 9th International Conference*, volume 4193 of *LNCS*, pages 778–787, Reykjavik, Iceland, 9-13 Sept. 2006. Springer-Verlag.

TABLE III

ONE CYCLE CROSSOVER. NUMBER OF RUNS ENDING UP AT FITNESSES 239, 241 (LOCAL OPTIMA) AND 243 (GLOBAL OPTIMUM). AVERAGE NUMBER OF GENERATIONS REQUIRED TO SOLVE THE PROBLEM ARE SHOWN WITHIN PARENTHESIS.

	Crossover Rate = 0.5				*Crossover Rate = 1.0*			
	Fit. 239	Fit. 241	Fit. 243	Avg. Fit.	Fit. 239	Fit. 241	Fit. 243	Avg. Fit.
Easy 1	28	27	5 (39.40)	238.58	24	26	4 (42.19)	238.62
Easy 2	18	16	4 (61.50)	237.61	19	18	3 (66.83)	237.72
Medium 1	22	11	0 (NA)	237.28	20	12	0 (NA)	229.43
Medium 2	25	13	1 (69.00)	237.75	21	10	0 (NA)	227.53
Hard 1	26	6	0 (NA)	237.37	25	4	0 (NA)	228.75
Hard 2	17	10	0 (NA)	237.15	16	8	0 (NA)	229.07

TABLE IV

MULTI CYCLE CROSSOVER. NUMBER OF RUNS ENDING UP AT FITNESSES 239, 241 (LOCAL OPTIMA) AND 243 (GLOBAL OPTIMUM). AVERAGE NUMBER OF GENERATIONS REQUIRED TO SOLVE THE PROBLEM ARE SHOWN WITHIN PARENTHESIS.

	Crossover Rate = 0.5				*Crossover Rate = 1.0*			
	Fit. 239	Fit. 241	Fit. 243	Avg. Fit.	Fit. 239	Fit. 241	Fit. 243	Avg. Fit.
Easy 1	42	25	6 (17.16)	239.23	26	34	7 (12.42)	239.11
Easy 2	37	29	7 (21.42)	239.91	40	27	7 (12.57)	239.37
Medium 1	31	9	1 (18.00)	237.91	28	14	1 (14.00)	238.12
Medium 2	28	42	2 (23.00)	239.73	34	32	1 (11.00)	239.22
Hard 1	42	4	0 (NA)	237.95	37	8	1 (19.00)	238.13
Hard 2	22	3	1 (19.00)	237.29	25	6	0 (NA)	237.33

TABLE V

UNIFORM SWAP CROSSOVER. NUMBER OF RUNS ENDING UP AT FITNESSES 239, 241 (LOCAL OPTIMA) AND 243 (GLOBAL OPTIMUM). AVERAGE NUMBER OF GENERATIONS REQUIRED TO SOLVE THE PROBLEM ARE SHOWN WITHIN PARENTHESIS.

	Crossover Rate = 0.5				*Crossover Rate = 1.0*			
	Fit. 239	Fit. 241	Fit. 243	Avg. Fit.	Fit. 239	Fit. 241	Fit. 243	Avg. Fit.
Easy 1	17	55	21 (26.09)	240.84	19	51	29 (19.44)	240.44
Easy 2	22	34	36 (25.30)	241.03	16	45	30 (18.26)	241.00
Medium 1	32	23	2 (29.50)	238.73	36	25	6 (19.00)	239.26
Medium 2	31	55	1 (20.00)	240.04	40	40	8 (20.25)	240.03
Hard 1	49	19	2 (31.00)	239.03	55	19	4 (23.25)	239.35
Hard 2	51	12	3 (30.33)	238.85	41	26	8 (25.75)	239.62

TABLE VI

PMX CROSSOVER. NUMBER OF RUNS ENDING UP AT FITNESSES 239, 241 (LOCAL OPTIMA) AND 243 (GLOBAL OPTIMUM). AVERAGE NUMBER OF GENERATIONS REQUIRED TO SOLVE THE PROBLEM ARE SHOWN WITHIN PARENTHESIS.

	Crossover Rate = 0.5				*Crossover Rate = 1.0*			
	Fit. 239	Fit. 241	Fit. 243	Avg. Fit.	Fit. 239	Fit. 241	Fit. 243	Avg. Fit.
Easy 1	17	57	19 (21.31)	240.81	17	56	22 (15.22)	240.97
Easy 2	31	35	23 (21.78)	240.44	22	52	17 (15.29)	240.60
Medium 1	34	25	6 (26.66)	239.16	31	26	7 (17.71)	239.30
Medium 2	29	58	6 (21.17)	240.33	35	53	2 (16.50)	240.09
Hard 1	63	19	1 (24.00)	239.16	53	23	3 (20.66)	239.21
Hard 2	42	21	4 (26.00)	238.98	37	25	3 (19.33)	239.22

[5] E. Galván-López, J. Togelius, and S. Lucas. Towards Understanding the Effects of Neutrality on the Sudoku Problem. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1509–1509, New York, NY, USA, 2007. ACM.

[6] D. E. Goldberg and R. Lingle. Alleles, Loci and the Traveling Salesman Problem. In *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, pages 154–159, Carnegie-Mellon University, 1985.

[7] T. Jones. *Evolutionary Algorithms, Fitness Landscapes and Search*. PhD thesis, University of New Mexico, Albuquerque, 1995.

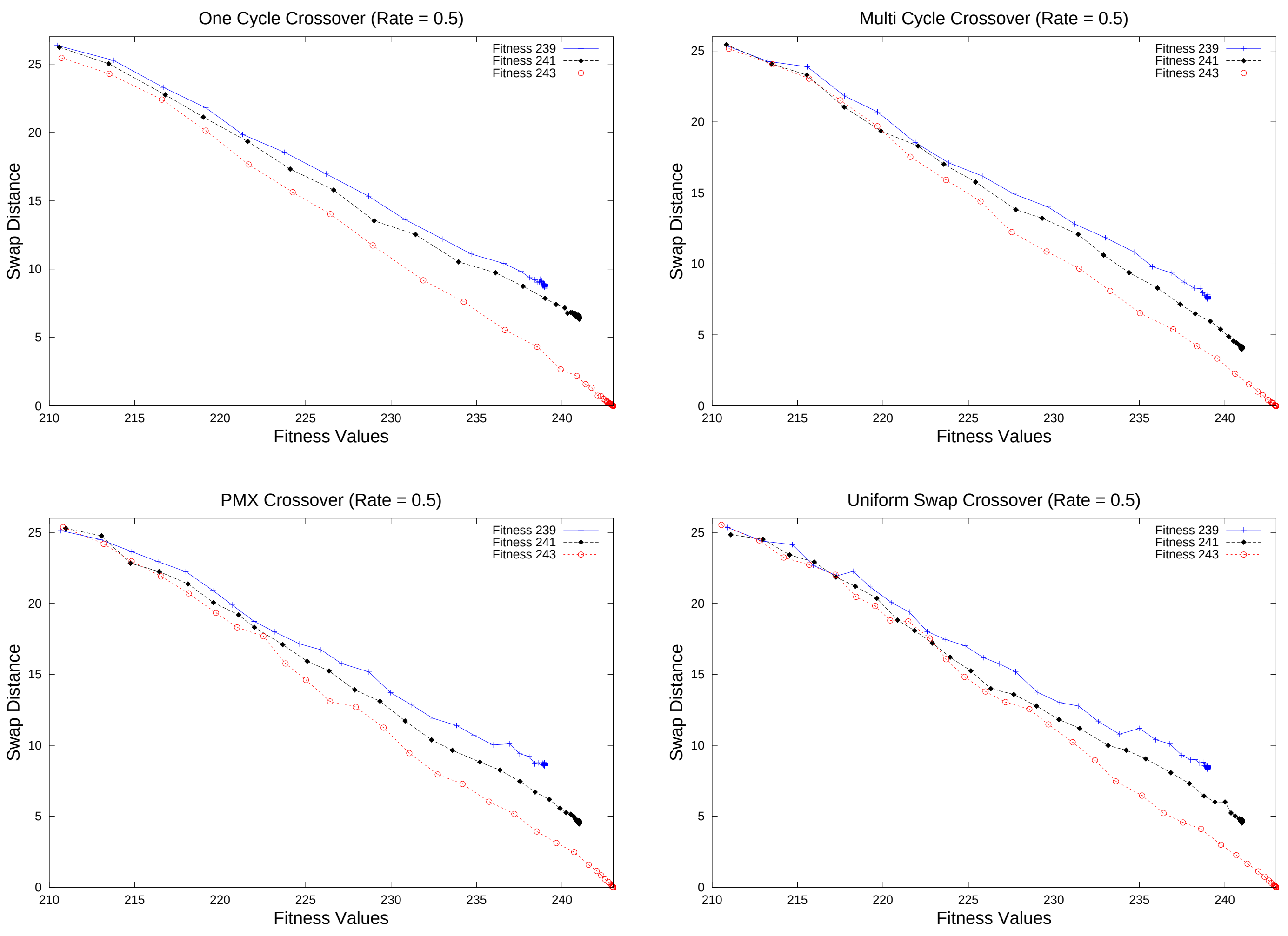


Fig. 4. Swap distance between the best individual per generation and the global solution versus fitnesses using an easy Sudoky puzzle (i.e., easy 1) and setting crossover rate at 0.5.

[8] T. Jones and S. Forrest. Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms. In L. J. Eshelman, editor, *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 184–192, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers.

[9] A. Moraglio, J. Togelius, and S. M. Lucas. Product Geometric Crossover for the Sudoku Puzzle. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, 2006. *Togelius is working at IDSIA on SNF grant 21-113364 to J. Schmidhuber.*

[10] I. M. Oliver, D. J. Smith, and J. R. C. Holland. A study of permutation crossover operators on the traveling salesman problem. *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application table of contents*, pages 224–230, 1987.

[11] R. Poli and E. Galván-López. On The Effects of Bit-Wise Neutrality on Fitness Distance Correlation, Phenotypic Mutation Rates and Problem Hardness. In C. R. Stephens, M. Toussaint, D. Whitley, and P. Stadler, editors, *Foundations of Genetic Algorithms IX*, Lecture Notes in Computer Science, pages 138–164, Mexico city, Mexico, 8-11 Jan. 2007. Springer-Verlag.

[12] R. J. Quick, V. J. Rayward-Smith, and G. D. Smith. Fitness Distance Correlation and Ridge Functions. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, pages 77–86, London, UK, 1998. Springer-Verlag.

[13] F. Rothlauf. *Representations for Genetic and Evolutionary Algorithms*. Springer, 2nd edition, 2006.

[14] F. Rothlauf and D. Goldberg. Redundant Representations in Evolutionary Algorithms. *Evolutionary Computation*, 11(4):381–415, 2003.

[15] F. Rothlauf and M. Oetzel. On the Locality of Grammatical Evolution. In P. Collet, M. Tomassini, M. Ebner, S. Gustafson, and A. Ekart, editors, *EuroGP*, volume 3905 of *Lecture Notes in Computer Science*, pages 320–330. Springer, 2006.

[16] M. Tomassini, L. Vanneschi, P. Collard, and M. Clergue. A Study of Fitness Distance Correlation as a Difficulty Measure in Genetic Programming. *Evolutionary Computation*, 13(2):213–239, Summer 2005.

[17] S. Wright. The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution. In D. F. Jones, editor, *Proceedings of the Sixth International Congress on Genetics*, volume 1, pages 356–366, 1932.

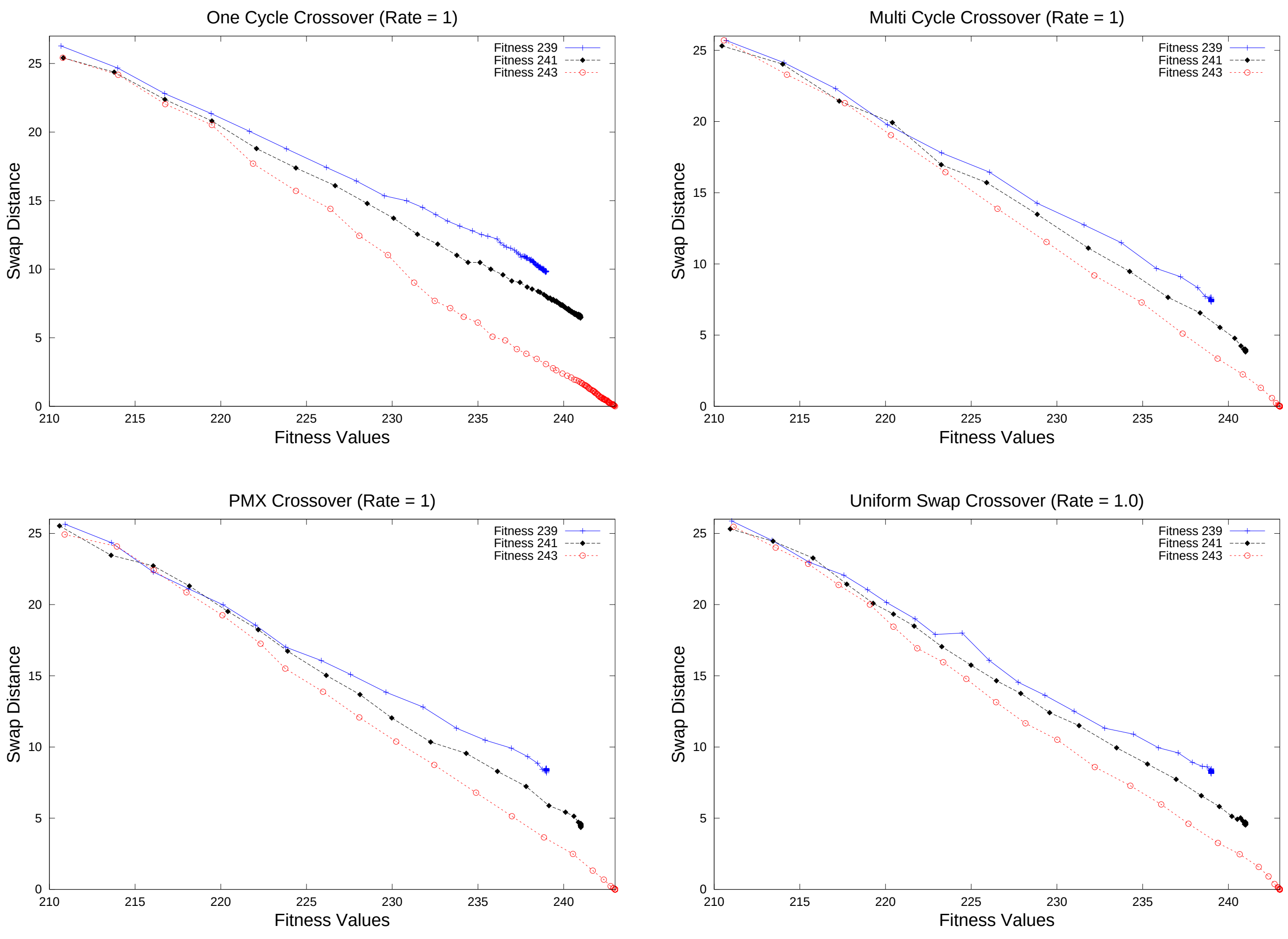


Fig. 5. Swap distance between the best individual per generation and the global solution versus fitnesses using an easy Sudoky puzzle (i.e., easy 1) and setting crossover rate at 1.0.