On the Locality of Standard Search Operators in Grammatical Evolution

Ann Thorhauer and Franz Rothlauf

University of Mainz, Germany
{thorhauer,rothlauf}@uni-mainz.de
http://www.uni-mainz.de

Abstract. Offspring should be similar to their parents and inherit their relevant properties. This general design principle of search operators in evolutionary algorithms is either known as locality or geometry of search operators, respectively. It takes a geometric perspective on search operators and suggests that the distance between an offspring and its parents should be less than or equal to the distance between both parents. This paper examines the locality of standard search operators used in grammatical evolution (GE) and genetic programming (GP) for binary tree problems. Both standard GE and GP search operators suffer from low locality since a substantial number of search steps result in an offspring whose distance to one of its parents is greater than the distance between both of its parents. Furthermore, the locality of standard GE search operators is higher than that of standard GP search operators, which allows more focused search in GE.

Keywords: Grammatical evolution, genetic programming, locality, geometric crossover, random walk.

1 Introduction

Recombination operators in evolutionary algorithms (EA) aim to construct offspring solutions in such a way that the offspring inherit the properties of their parents. Thus, offspring are constructed using genetic material of their parents, in order to ensure that they are "similar" to them. Analogously, the mutation operators in EA modify the offspring solutions so that the new "mutated" solution is only slightly different from the original solution. Recombination and mutation operators thus both follow the principle of creating solutions that are similar to the original solution. Violating this principle would result in random search since, in this case, the offspring would be highly dissimilar to their parents, and the evolutionary search process would not be able to focus on promising areas of the search space [4].

This basic principle of genetic search operators was first formulated by Liepins and Vose [13] and Radcliffe [17,22], who recognized that search operators cannot be designed independently of the search space. On the contrary, their design must be based on the metric defined in the search space. Indeed, mutation should

T. Bartz-Beielstein et al. (Eds.): PPSN XIII 2014, LNCS 8672, pp. 465–475, 2014.

create an offspring x^o from a parent x^p in such a way that the distance $d(x^p, x^o)$ between parent and offspring is low. Analogously, given two parental solutions x_1^p and x_2^p and one offspring solution x^o, recombination operators should be designed in such a way that $\max(d(x_1^p, x^o), d(x_2^p, x^o)) \leq d(x_1^p, x_2^p)$ [17][18, p. 62], which means that the distances between offspring and parents should be less than or equal to the distance between the parents. By viewing the distance between two solutions as a measurement of dissimilarity, this design principle ensures that the offspring solutions are similar to the original (parent) solution.

This general design principle of search operators introduced by Liepins and Radcliffe [13,17] was later denoted by Rothlauf as the "locality of search operators" [18,19] and by Moraglio as the "geometry of search operators" [15,14]. Mutation operators have high locality, i.e., are geometric if offspring solutions are similar to their parents; analogously, recombination operators have high locality, i.e., are geometric if the distances between offspring and parents are less than or equal to the distance between the parents. Crossover and mutation are defined representation-independent using the notion of distance associated with the search space. The geometric terms use the notions of line segment and ball, which are well defined once a notion of distance in the search space is defined [15].

This paper studies the locality of standard search operators used in GE (crossover, mutation, and duplication) and GP (crossover and reproduction). We examined whether the GE and GP search operators have high locality, i.e., are geometric. In the experiments, we focused on binary trees and performed random walks through the binary tree search space by measuring distances between both parents x_1^p and x_2^p as well as between an offspring x^o and its parents. The locality of search operators is high if $\max(d(x_1^p, x^o), d(x_2^p, x^o)) \leq d(x_1^p, x_2^p)$.

In Sect. 2, we define the locality of search operators and provide a brief overview of the literature on locality. In Sect. 3, we present the experiments and results. The paper ends with some concluding remarks.

2 Locality of Search Operators

Each search space can be defined as a *topological space*, which describes similarities between solutions by defining the relationships between sets of solutions. Formally, a topological space is an ordered pair (X, T), where X is a set of solutions and T is a collection of subsets of X called open sets. We can define different topologies (search spaces) by combining X with different T. For example, *metric search spaces* are a specialized form of topological spaces where similarities between solutions are measured by a distance. In metric search spaces, we have a set X of solutions and a real-valued distance function (also called a metric) $d : X \times X \to \mathbb{R}$ that assigns a real-valued distance to any combination of two elements $x, y \in X$.

The locality of search operators [18] is equivalent to the concept of geometry of search operators [15]. Both define search operators based on the metric of the search space. Given a metric, we are able to define distances between solutions.

In particular, a mutation operator has high locality if the distance between the resulting offspring and its parent is small. From a geometric perspective, the offspring are in the space-specific ball of a small radius centered in the parent. Analogously, a crossover operator has high locality if the distance between offspring x^o and its parents is less than or equal to the distance between both parents x_1^p and x_2^p ($\max(d(x_1^p, x^o), d(x_2^p, x^o)) \leq d(x_1^p, x_2^p)$); using the notion of geometry, a crossover is geometric if all offspring are in the space-specific segment between their parents [15].

We want to emphasize that the locality of search operators is different from the locality of representations [18]. Representations are genotype-phenotype mappings that assign genotypes to phenotypes. In both search spaces (genotype and phenotype space), a metric defines distances between solutions. However, the metric used in the genotype space and the phenotype space can be different. The locality of a representation describes how well the distances between genotypes fit the distances between the corresponding phenotypes. Thus, a representation has high locality if distances between genotypes are similar to the corresponding phenotype distances, for example if neighboring genotypes correspond to neighboring phenotypes. Analogously, a representation has low locality if genotype and phenotype distances do not fit together, for example if neighboring genotypes are not neighbors in the phenotype search space. Although both concepts, the locality of search operators and the locality of a representation, are based on the notion of distance between solutions, they are quite different and should not be confused with one another. The locality of a representation is relevant for the design of representations, whereas the locality of search operators for the design of meaningful search operators.

2.1 Locality in Genetic Programming

There are a number of studies on the locality of GP [5,6,7], however they do not examine the locality of search operators nor the locality of representations; rather, they focus on the locality of the genotype-fitness mapping. They are mainly interested in how the choice of mutation operators affects the changes in the corresponding fitness values. For example, Galván-López et al. [7] study genotype-fitness mappings and find "that the mutation operators examined are inconsistent with respect to the quality of locality as measured by fitness and structural changes". In a similar paper, Galván-López et al. [5] find that "when the original fitness is low, large genotype jumps can lead to fitness improvements" [5]. In contrast, "when the original fitness is high, large [genotype mutations] tend to be quite detrimental" [5]. Galván-López et al. [6] study whether small genotype changes correspond to small fitness changes. The authors introduce different neighborhood functions on the fitness space and observe that their ability to serve as good predictor of GP performance is limited.

Another study by Uy and his co-workers distinguishes between syntactic and semantic locality of crossover in GP [23,24,25]. They notice that "most GP genetic operators have been designed based on syntax alone; but small changes in syntax can lead to large changes in semantics." [24]. The authors introduce a new semantic similarity based crossover (SSC), which ensures that exchanged subtrees between individuals are semantically similar concerning fitness values. SSC leads to higher GP performance since the resulting genotype step size is smaller than in standard GP operators. Uy et al. [23] compare the syntactic crossover in GP with the semantic crossover in GP. With locality measuring the differences between two subtrees, they find that syntactic locality (measured using Levenshtein tree distance) is less important than semantic locality (measured using fitness differences).

In summary, it is still unclear whether standard GP search operators have high locality, i.e., are geometric. There is evidence that small genotype changes can also lead to large fitness changes, which is detrimental for guided search in GP.

2.2 Locality in Grammatical Evolution

GE [16] is a variant of GP that can evolve complete programs in an arbitrary language using a variable-length binary string. In GE, phenotype expressions are created from binary genotypes by using a complex genotype-phenotype mapping. A genotype consists of groups of eight bits (denoted as codons) which encode an integer value that selects production rules from a grammar in BNF. These rules are used in the mapping process to create a phenotype. The mapping process is deterministic since the same genotype always results in the same phenotype.

The standard GE recombination operator [16] is similar to the cut and splice operator introduced in [8]. After selecting two parents, a crossover point is randomly selected for each parent. Then, the genetic material beyond these points is exchanged between the parents. As a result, rather than remain constant, the length of the genotype changes during the search. The standard GE mutation operator [16] randomly changes the integer value of a codon. The third standard GE operator, duplication, increases the number of available genetic material. It copies a random number of codons starting at a randomly selected start codon and inserts them between the second last and last codon in the genotype.

There are no studies on the locality of GE search operators. Instead, existing work focuses on the locality of the genotype-phenotype mapping or on how search performance depends on the type of search operator. Rothlauf and Oetzel [20] study the locality of the genotype-phenotype mapping and find that "the representation used in GE has problems with locality as many neighboring genotypes do not correspond to neighboring phenotypes." Byrne et al. [2,1] distinguish between two types of mutation operators in GE: the structural mutation (that changes the shape of the derivation tree) and the high-locality nodal mutation (that changes the value of a node). They examine the impact of these operators on search performance and find out that both have different goals in a GE search process: exploration and exploitation. Castle and Johnson [3] study the mutation and crossover points in GE and find "that events occurring at the first positions of a genotype are indeed more destructive, but also indicate that they may be the most constructive crossover and mutation points"[3]. Finally, Hugosson et al. [9] examine different binary-integer representations (Gray versus binary code) in GE. They find that the choice of the binary-integer mapping has no influence on GE search performance.

3 Experiments and Results

We studied the locality of standard search operators used in GE and GP. Standard GE search operators are (one-point) crossover, (integer) mutation, and duplication [21]. Standard GP search operators are crossover and reproduction [11]. The locality of recombination operators is high if the distances between offspring and parents are less than or equal to the distance between both parents. For mutation and duplication, high locality implies a low distance between offspring and parent.

3.1 Experimental Design

To study the locality of search operators, we performed random walks using different types of GE and GP search operators. We did not use a selection operator. The search operators created two offspring x_1^o and x_2^o from two parents x_1^p and x_2^p, which replaced their parents. If the variation operators included recombination, we measured the distance $d(x_1^p, x_2^p)$ between both parents as well as the distances $d(x_i^o, x_j^p)$ $(i, j \in \{1, 2\})$ between each of the two offspring and their two parents. If we applied only mutation or duplication (and no crossover), we created one offspring x^o from each of the two parents. Each offspring replaced the corresponding parent. To evaluate the locality of mutation and duplication, we measured the distance $d(x^o, x^p)$ between each offspring and its corresponding parent.

For both GE and GP, the definition of the terminal and function set is relevant for the distances between solutions. In the current study, we focused on problems where solutions are binary trees. Thus, the number of terminals $|T| = 1$ equals the number of functions $|F| = 1$. For GE, we used two production rules: the first one chose between a binary function and a terminal (e.g. $< \mathit{expr} >$::=$< \mathit{expr} > + < \mathit{expr} > \mid < \mathit{var} >$) and the second one defined a terminal (e.g. $< \mathit{var} >$::= X). The fitness of binary trees using associative binary functions (like +,-,*,/) is determined only by its size l (number of terminals plus number of functions) since the order of traversing such a tree is irrelevant (due to the associativity of the function). For example, for $T = \{x\}$ and $F = \{+\}$, all feasible trees of size l encode the expression $(l+1) * \frac{x}{2}$. Consequently, we measured the distance between two solutions by using the Levenshtein distance as metric, that is, the minimal number of operations that are needed to transform one expression into another [12] between the two encoded expressions. For the example ($T = \{x\}$ and $F = \{+\}$), the Levenshtein distance between two valid binary trees x_i and x_j of length l_i and l_j is equal to $|l_i - l_j|$. We should be aware that the size l of a valid tree is always odd. Furthermore, for two valid binary trees, all possible distances are even. In contrast to GP, GE search operators can also create invalid solutions, where the genotype-phenotype mapping process cannot be finished.

In this case, the Levenshtein distance between the two corresponding expressions can be odd since the size of invalid expressions can be even.

In the GP experiments, we applied no constraint on the maximum allowable tree depth. For GE, we did not use a wrapping operator since for all the solutions (binary trees) where wrapping would be necessary the mapping process would never terminate and thus the corresponding individuals would be invalid anyway. Each random walk started with two random, yet identical, solutions. For GP, the initial solutions were created using the grow method with a maximum depth $d_{max} = 6$. For GE, the initial solutions were randomly created with a maximum length of 10 codons (80 binary alleles).

Throughout the random walk, the variation operators were applied with standard probabilities. For GP, the crossover probability was $p_c = 0.9$ (biased towards selecting internal nodes with a probability of 0.9) and the reproduction probability was $p_r = 0.1$ [10]. For GE, the crossover probability was $p_c = 0.9$, the mutation probability was $p_m = 0.01$, and the duplication probability was $p_d = 0.01$ [21]. In each search step, two new offspring that replaced their parents were generated. Each random walk terminated after 50 search steps, generating overall 100 offspring. For each experimental setting, we performed 100,000 random walks resulting in a total of 10 million offspring.

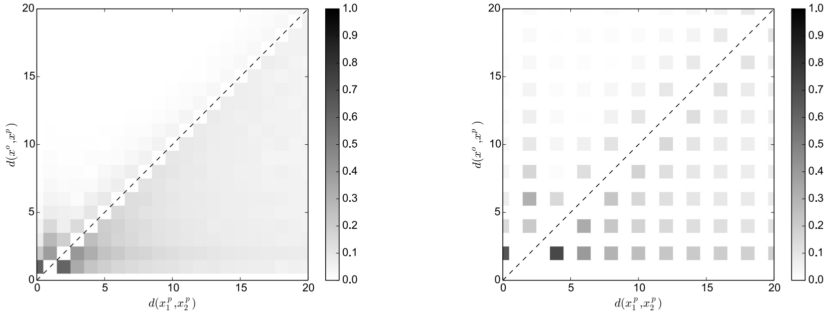
3.2 Results

For GE and GP, we studied the locality of the combined standard search operators as well as the locality of recombination alone. Furthermore, we examined the locality of the mutation and duplication operator used in GE only.

Locality of Standard Search Operators. We studied whether the standard search operators used in GE (crossover, mutation, duplication) and GP (crossover, reproduction) have high locality. In each step of the random walk, the GE operators were applied with probabilities $p_c = 0.9$, $p_m = 0.01$, and $p_d = 0.01$ and the GP operators were applied with $p_c = 0.9$ and $p_r = 0.1$.

For GE, in 75.4% of all cases, the minimal distance $min(d(x_1^p, x^o), d(x_2^p, x^o))$ between an offspring and its parents is equal to 0. For GP, 54.5% of all offspring are identical to at least one of its parents ($min(d(x_1^p, x^o), d(x_2^p, x^o)) = 0$). In the following plots, we will ignore all such applications of search operators.

Figure 1 plots the distribution of $d(x^o, x_j^p)$ ($j \in \{1, 2\}$) over $d(x_1^p, x_2^p)$. For increased clarity, we only show the results for $d(x_1^p, x_2^p), d(x^o, x^p) \leq 20$. For a given distance $d(x_1^p, x_2^p)$, the gray-coded squares indicate the percentage of offspring whose distance to one of their parents is equal to $d(x^o, x^p)$ (darker squares indicate a higher percentage of offspring). For example, for GE and parental distance $d(x_1^p, x_2^p) = 4$, about 22.8% of all offspring have $d(x^o, x_j^p) = 1$, 31.8% have $d(x^o, x_j^p) = 2$, 25% have $d(x^o, x_j^p) = 3$, 0% have $d(x^o, x_j^p) = 4$ (they are duplicates of one parent and thus excluded from analysis), 9.8% have $d(x^o, x_j^p) = 5$, and so on. Search operators have high locality if they produce only offspring with $d(x^o, x_i^p) \leq d(x^{p1}, x^{p2})$ ($i \in \{1, 2\}$), which are located in the lower right triangle of the plot below the line through origin. All offspring located in the upper left triangle are the result of a low-locality operator. The plots indicate that standard search operators of GE as well as GP suffer from low locality since a substantial number of random walk steps resulted in offspring whose distance to one of their parents is greater than the distance between both parents. We should be aware that there are only even-numbered distances between GP solutions (leading to "holes" in the GP plot), whereas for GE odd-numbered distances also exist (either the parent or the offspring is invalid).



(a) GE (b) GP

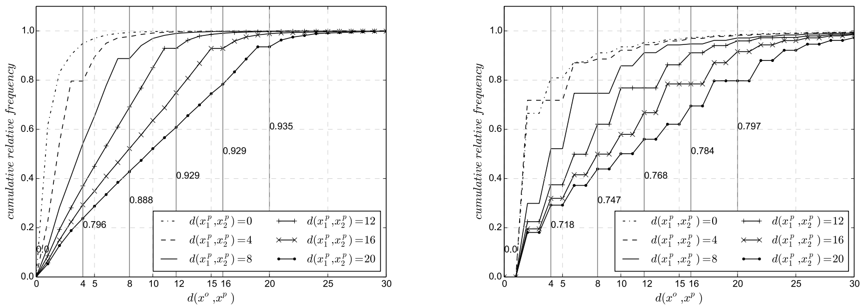
Fig. 1. Distribution of $d(x^o, x_j^p)$ ($j \in \{1, 2\}$) over $d(x_1^p, x_2^p)$

To get deeper insights into the locality of standard search operators, we will now focus on all offspring that are generated from parents with a given distance $d(x^{p1}, x^{p2})$. Figure 2 plots the number of offspring x^o (cumulative relative frequency) over $d(x^o, x_j^p)$ ($j \in \{1, 2\}$) for fixed distances $d(x^{p1}, x^{p2}) \in \{0, 4, 8, 12, 16, 20\}$. Each line represents a vertical cut through Fig. 1, summing up all offspring whose distance to its parents is less than or equal to $d(x^o, x^p)$. For example, for GE and parental distance $d(x_1^p, x_2^p) = 4$, 79.6% of all offspring have a distance to their parents that is less than or equal to 4 ($d(x^o, x_j^p) \leq 4$). Thus, in 79.6% cases the standard GE search operators have high locality. Search operators would have perfect locality if the cumulative frequency was 1 for $d(x^o, x^p) \leq d(x_1^p, x_2^p)$. Table 1 summarizes the percentage of applications of standard GE and GP search operators resulting in an offspring where $d(x^o, x^p) > d(x_1^p, x_2^p)$.

We see that standard GE and GP operators suffer from low locality since a substantial number of offspring display a distance to one of their parents that is greater than the distance between their parents. In general, the locality of standard GE search operators is higher than that of standard GP operators. For example, for $d(x_1^p, x_2^p)$=8, about 90% of all GE offspring but only about 75% of all GP offspring have less or equal distances to their parents than their parents do to each other.

Table 1. Percentage of offspring, where $d(x^o, x^p) > d(x_1^p, x_2^p)$

$d(x_1^p, x_2^p)$	4	8	12	16	20
GE	20.4	11.2	7.1	7.1	6.5
GP	28.2	25.3	23.2	21.6	20.3



(a) GE (b) GP

Fig. 2. Number of offspring x^o (cumulative frequency) over $d(x^o, x^p)$ for fixed distances $d(x_1^p, x_2^p) \in \{0, 4, 8, 12, 16, 20\}$

Locality of GE Crossover. We studied the locality of the GE and GP crossover operator. When using the same experimental design as above, we were faced with the problem that performing a GE random walk with only crossover would lead to a non-representative sample of offspring. Since crossover alone cannot increase the genetic material of the genotypes (this is the aim of duplication), it would only reshuffle genetic material between the two random walk solutions and not create representative GE solutions obtained in a standard GE run. Thus, to ensure representative GE solutions and to also be able to generate also longer GE genotypes, we slightly modified our experimental setting. We considered all 10 million offspring created in the random walks using the combined standard search operators as described above, but only applied crossover with $p_c = 1$ to each pair of offspring. By only applying crossover to the solutions generated by standard search operators, we were able study the locality of crossover in detail.

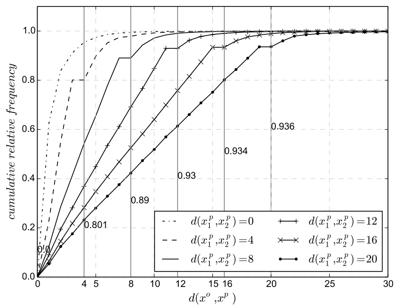


Fig. 3. GE with only crossover: number of offspring x^o (cumulative frequency) over $d(x^o, x^p)$ for fixed distances $d(x_1^p, x_2^p)$

We will only present results for GE, since the results for GP are identical to Figs. 1(b) and 2(b). Since the GP reproduction operator just copies a parent to its offspring, the locality of crossover plus reproduction is equal to crossover alone. Figure 3 shows the results for GE using only crossover ($p_c = 1$, no mutation or duplication). The comparison of these results to the previous results of the combined standard search operators (Fig. 2(a)) reveals no larger differences.

Thus, the locality of GE standard search operators is mainly determined by the locality of the crossover operator.

Locality of GE Mutation and Duplication. We will now focus on the GE mutation and duplication operator. Both operators have high locality if $d(x^o, x^p)$ is low. We chose the same experimental setting as in our GE crossover study and applied either mutation (p_m=0.01) or duplication ($p_d = 1$) to all offspring that were generated during the random walks using all GE search operators. Just as crossover alone cannot increase the length of GE individuals, mutation alone cannot either; only duplication can increase the amount of genetic material (but not the diversity of the material).

Figure 4 plots the number of offspring x^o (cumulative relative frequency) over the distance $d(x^o, x^p)$ between offspring and corresponding parent. We omitted all cases where $d(x^o, x^p) = 0$ and plotted the results for $d(x^o, x^p) \leq 10$. Since p_m is low and many mutations and duplications have no effect on the encoded expression, many offspring are identical to their parents. For mutation, 98.8% of the 10 million offspring expressions are identical to their parents ($d(x^o, x^p)$=0). Only about 30% of the remaining offspring have a distance of 1 to their parents ($d(x^o, x^p) = 1$). For duplication, 53.3% of all offspring are identical to their parents. About 50% of the remaining offspring have $d(x^0, x^p) = 1$. Both local search operators suffer from low locality since they create offspring whose distances to their parents are large.

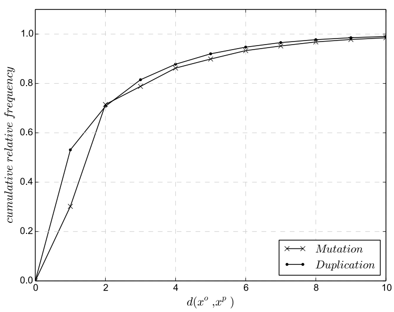


Fig. 4. GE with only mutation or duplication: number of offspring x^o (cumulative frequency) over $d(x^o, x^p)$

4 Conclusions

This work studies the locality of standard search operators for GE (crossover, mutation, and duplication) and GP (crossover and reproduction) by performing random walks through the search space of binary trees and measuring the distances between offspring and parents. The locality of standard search operators is high if the distances between the offspring and their parents is less than or equal to the distance between both parents. This concept is also known as the geometry of search operators. For binary trees we found out, that both GE and GP standard search operators have problems with low locality since a substantial number of offspring are not similar to their parents. Comparing GE and GP reveals that standard GE operators have higher locality than standard GP operators. The locality of the standard search operators in GE is mainly determined by the crossover operator; mutation and duplication are less important. They are necessary to obtain a high diversity within the genetic material, but have low impact on the overall locality of the GE variation operators.

In the future we will extend this analysis to non-binary trees with more complex terminal and function sets. Although the results of the current study only hold for binary trees, we expect to see similar results for other tree structures. Moreover, we are going to use other distance metrics to measure similarities between individuals.

References

1. Byrne, J., O'Neill, M., McDermott, J., Brabazon, A.: An analysis of the behaviour of mutation in grammatical evolution. In: Esparcia-Alcázar, A.I., Ekárt, A., Silva, S., Dignum, S., Uyar, A.Ş. (eds.) EuroGP 2010. LNCS, vol. 6021, pp. 14–25. Springer, Heidelberg (2010)
2. Byrne, J., O'Neill, M., Brabazon, A.: Structural and nodal mutation in grammatical evolution. In: GECCO 2009: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 1881–1882. ACM (2009)
3. Castle, T., Johnson, C.G.: Positional effect of crossover and mutation in grammatical evolution. In: Esparcia-Alcázar, A.I., Ekárt, A., Silva, S., Dignum, S., Uyar, A.Ş. (eds.) EuroGP 2010. LNCS, vol. 6021, pp. 26–37. Springer, Heidelberg (2010)
4. Doran, J., Michie, D.: Experiments with the graph traverser program. Proceedings of the Royal Society of London (A) 294, 235–259 (1966)
5. Galván-López, E., McDermott, J., O'Neill, M., Brabazon, A.: Towards an understanding of locality in genetic programming. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO 2010, pp. 901–908. ACM, New York (2010)
6. Galván-López, E., McDermott, J., O'Neill, M., Brabazon, A.: Defining locality as a problem difficulty measure in genetic programming. Genetic Programming and Evolvable Machines 12(4), 365–401 (2011)
7. Galvan-Lopez, E., O'Neill, M., Brabazon, A.: Towards understanding the effects of locality in gp. In: Eighth Mexican International Conference on Artificial Intelligence, MICAI 2009, pp. 9–14 (2009)
8. Goldberg, D.E., Korb, B., Deb, K.: Messy genetic algorithms: Motivation, analysis, and first results. Complex Systems 3(5), 493–530 (1989)
9. Hugosson, J., Hemberg, E., Brabazon, A., O'Neill, M.: An investigation of the mutation operator using different representations in grammatical evolution. In: 2nd International Symposium "Advances in Artificial Intelligence and Applications", Wisla, Poland, October 15-17, vol. 2, pp. 409–419 (2007)
10. Koza, J.R.: Genetic programming: On the programming of computers by natural selection. MIT Press, Cambridge (1992)
11. Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., Lanza, G.: Genetic Programming IV: Routine human-competitive machine intelligence. Springer, New York (2005)
12. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady 10(8), 707–710 (1966); Doklady Akademii Nauk SSSR 163(4), 845–848 (1965)
13. Liepins, G.E., Vose, M.D.: Representational issues in genetic optimization. Journal of Experimental and Theoretical Artificial Intelligence 2, 101–115 (1990)

14. Moraglio, A.: Towards a Geometric Unification of Evolutionary Algorithms. PhD thesis, Department of Computer Science, University of Essex (November 2007)
15. Moraglio, A., Poli, R.: Topological interpretation of crossover. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 1377–1388. Springer, Heidelberg (2004)
16. O'Neill, M., Ryan, C.: Grammatical evolution. IEEE Transactions on Evolutionary Computation 5(4), 349–358 (2001)
17. Radcliffe, N.J.: Equivalence class analysis of genetic algorithms. Complex Systems 5(2), 183–205 (1991)
18. Rothlauf, F.: Distributed Autonomous Robotics Systems, 1st edn. STUDFUZZ, vol. 104. Springer, Heidelberg (2002)
19. Rothlauf, F.: Design of Modern Heuristics. Springer, Heidelberg (2011)
20. Rothlauf, F., Oetzel, M.: On the locality of grammatical evolution. In: Collet, P., Tomassini, M., Ebner, M., Gustafson, S., Ekárt, A. (eds.) EuroGP 2006. LNCS, vol. 3905, pp. 320–330. Springer, Heidelberg (2006)
21. Ryan, C., Collins, J.J., Neill, M.O.: Grammatical evolution: Evolving programs for an arbitrary language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds.) EuroGP 1998. LNCS, vol. 1391, pp. 83–95. Springer, Heidelberg (1998)
22. Surry, P.D., Radcliffe, N.: Formal algorithms + formal representations = search strategies. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 366–375. Springer, Heidelberg (1996)
23. Uy, N.Q., Hoai, N.X., O'Neill, M., McKay, B.: The role of syntactic and semantic locality of crossover in genetic programming. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6239, pp. 533–542. Springer, Heidelberg (2010)
24. Uy, N.Q., O'Neill, M., Hoai, N.X., Mckay, B., Galván-López, E.: Semantic similarity based crossover in gp: The case for real-valued function regression. In: Collet, P., Monmarché, N., Legrand, P., Schoenauer, M., Lutton, E. (eds.) EA 2009. LNCS, vol. 5975, pp. 170–181. Springer, Heidelberg (2010)
25. Uy, N.Q., Hoai, N.X., O'Neill, M., McKay, R.I., Phong, D.N.: On the roles of semantic locality of crossover in genetic programming. Information Sciences 235, 195–213 (2013)