Competitive and Co-Operative Analysis of Genetic-Programming

Danish Hamid.1, Hammad Majeed.2, Zulhasni Abdul Rahim.3*, Muhammad Saqib Iqbal.3

1Capital University of Science & Technology (CUST), Pakistan.
2National University of Computer & Emerging Sciences (NUCES), Pakistan.
3Malaysia-Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia (UTM), Malaysia.

*Corresponding author: zulhasni@utm.my

Abstract. Co-operative and competitive building blocks drive the evolution of GP. There exists useful schema which remains promising and persistent in terms of their occurrence over the generations. These schemas are majorly present at levels other than root level, and gradually this genetic material is pushed towards the root level by co-operation, where they compete with each other, few among those increase their occurrence at the root level and rest vanished. These schemas if extended to the deeper depths, still be useful and helps in achieving the goal of the problem. Lastly, some of these schemas are same even in different runs for the same problem. The experiment results show that the evolution is driven by co-operative rooted building blocks as well, along with the competitive building blocks. It shows the count of building blocks in different positions never reaches zero, which indicates the presence of the genetic material required for the construction of the building block at a particular position in the form of co-operation and competition with other schemas, which, at a later stage, may move to the correct position (root level). In our work, these behaviours are also demonstrated by injecting genetic material manually in the population and investigated their prominence throughout the evolution. This work contributes to many dimensions like; by combining these competitive building blocks appear at root level uncovers the problem solution over time. For that reason, by identifying, preserving and exchanging these building blocks explicitly, the system enactment can be improved severely. Additionally, the solutions of the developed versions of the same problem can also be handled through this.

1. Introduction
The Genetic programming resolves one of the significant challenges of computer science, i.e. to get required work done on the computer without advising how to perform it, by a device of automatically making a working computer program from the high-level statement of the problem. The objective of automatic programming is achieved by Genetic programming by reproducing population of computer programs, applying the principles of Darwinian natural selections and operations aspired biologically. These operations and reproduction, mutation, crossover (sexual recombination) and architecture change, shaped after gene duplication and gene omission in nature.
This is an independent device that genetically reproduces a population of computer programs to solve the issue. Genetic programming, mainly, repeatedly transform a population of computer programs into a new generation program, using analogues of naturally occurring genetic operations.

Analytical study in Genetic programming (GP) has always been a challenging task due to exhaustive search of schemata at different levels/depths in the entire population and the complex tree structures which search a problematic task. An only possible solution is to look for ad-hoc methods to achieve this search. This ad-hoc is mostly error-prone and inaccurate. The accuracy of each method varies and depends on its implementation.

Extensive research has been carried out since the GP is introduced to find out carefully how to oversight its search. GP Schema theory [1, 2] has been pulled out through some researchers focusing on extending Genetic Algorithm (GA) [3], while some others worked on defining their own [4-8].

In this paper we presented that evolution is driven by co-operation and competition so following is analyzed 1) There exists schema who have high occurrence at levels other than the root in the earlier generations but are gradually pushed towards root as generations pass, 2) These schemas when we search them in first generations population, they give us valuable occurrence at the root level, and they increase their occurrence at root level as the generation passes till the end of evolution, 3) There exist some particular schemata which are very useful. They increase their occurrence at root level in almost every run with a different population, 4) There exist small schemas (schemas with minimal depth) which give us valuable occurrence at root level initially and increase their occurrence till the end of evolution and if we extend those smaller schemas into larger ones then, search them again, it also gives us valuable occurrence at the root level. It can be treated as a useful material to evolve the specific problem. 5) We validate our results by injecting genetic material in the initial population and investigated co-operation and competition throughout evolution.

Section 2 describes initial literature on GP and crossover and related work on building blocks by other researchers worldwide. Section 3 discusses the analysis of competitive behaviour of (GP) Experimental setup and experiments performed for competitive Analysis is discussed. Also, results for these experiments are presented. Section 4 discusses the analysis of co-operative behaviour of (GP) Experimental setup and experiments performed for competitive Analysis is discussed. Also, results for these experiments are presented. Section 5 discusses the investigation of co-operative and competitive behaviour on the population in which we manually injected genetic material. Section VI discusses the overall conclusion of this work.

2. Background and Related Work

Extensive research has been carried out since the GP is introduced to find out carefully how to oversight its search. GP Schema theory [10, 9] has been pulled out through some researchers focusing on extending Genetic Algorithm (GA) [3], while some others worked on defining their own [4, 5].

Mainly all GP schema works focus on the recognition and investigation of the processing sub-trees and tree fragments. Generalize schema are referred to as a don’t-care (#) in GPs. The definition of schema determines this (#) can belong to any sub-tree [5], rooted sub-tree or to a terminal or a function [1, 2]. The identification of valuable schemata is a real confront.

These useful schemata are the building blocks or schema in literature [1, 2, 7]. It is fall out that GP, by combining these building blocks uncovers the problem solution over time. For that reason, by identifying, preserving and exchanging explicitly, the system enactment can be improved severely. Additionally, the solutions of the developed versions of the same problem can also be handled through this.

In 2004 A paper named A Competitive Building Block Hypothesis showed how rooted building blocks could be developed in GP-type systems. According to its hypothesis, the competitive building blocks share the same root structure and throughout the population compete to extend their particular extensions to the conventional structure. It illustrated that they do not only work in a cycle with non-rooted co-operative
building blocks, but competitive building blocks also exist in the structure. This research work is based on this paper [8].

GP problem implementation is simple. Significant schema, dead code (introns) and module acquisition can be identified and manipulated. By counteracting the result of the schema in the container-tree, it is evaluated on the schema contribution principle. After that, the effect is checked on the container trees fitness. They empirically illustrated its pros and cons [9].

A new schema-based approach is proposed to observe the loss of diversity in Genetic Programming populations. This approach is all about genealogy information and is easily integrated with existing GP variants. Research proved that OSGP-S showed comparable or better results than standard OSGP. There are two main practical advantages of this approach that are one it does not change the behaviour of the crossover and selection and secondly, a correctly parameterized diversification strategy will only act on highly similar groups of individuals, where the additional mutation does not damage the algorithm's ability to improve solutions [10] locally.

Schema Theory and Markov Chains Models for Genetic Programming and Variable-length Genetic Algorithms with Homologous Crossover presents an algorithm which applies to the group mentioned above of cross over operators and defines a theoretical approach for Genetic Programming and type of genetic Algorithms where length varies over time. The algorithm presented uses the idea of GP crossover masks where different concepts used in GA theory are generalized, and the population is distributed. Concepts of hyper schema node reference systems are also discussed which are used to represent schema of variable length [10].

In 2008 a paper named Semantic building blocks in genetic programming concluded that a considerable percentage of crossover events (typically over 75% in our experiments) is necessary for useful search in the semantic space. In addition to this, the results also show a robust correlation between lack of progress and high proportions of fixed contexts that is helpful in future researches as well [11].

Will Smart, Peter Andreae and Mengjie Zhang in Empirical Analysis of GP Tree-Fragments have explored the behaviour and use of fragment schemas in tree-based Genetic Programming. Experimentation has been carried out on a population size of 300 programs where the depth of each is seven nodes. Experiments concluded that a considerable amount of changes in each iteration occurred in the population fragments, which can increase or decrease with each generation [12, 13].

The co-operative nature of individuals in a population has been used in many applications to solve problems of a particular domain where the primary function is to optimize the objective function. The use of local and global fitness functions allow devising an efficient schema for optimization where individuals evolve with each generation. The paper presents the analysis of using two-level fitness functions and also uses a mutation operator of classical Genetic Programming [14].

The thesis presented by Lawrence Charles John Beadle in Semantic and Structural Analysis of Genetic Programming gives an empirical analysis of the evolved programs which includes the diversity, schema representation and continuously changing of program structure in each generation. During the run of GP, different operators are applied systematically to increase the diversity, and redundant portions of the program code are eliminated to decrease the size of the computer program. Performance impacts of these operators have been discussed in the thesis in detail. In the final stage, program structure importance is evaluated, which uses diverse evolutionary parameters to find a solution to the problem in optimization domains [15].

Will Smart in Empirical Analysis of GP tree fragments presents an approach for empirical analysis in a scenario where the population of individuals scale with a given schema where previous global analysis is not feasible. The method proposed in the thesis can perform well for analysis on the schema, which is large and uses the notion of maximal schema, maximal program subset, representative set of schemata, and
representative program subset. These concepts maximize the efficiency of the analysis and reduce the complexity without affecting the results [16].

MOIRCGA (A multi-offspring improved real-coded genetic algorithm) is proposed to solve constrained optimization problems, using the heuristically normal distribution and direction-based crossover (HNDDBX). On the one hand, HNDDBX operator guarantees that cross-generated offsprings are located near the better individuals in the population to ensure that there is an excellent chance of generating better offspring’s. Moreover, as iterations increase, the same individuals are likely to appear in the population that increases the possibility that the two parents of participation crossover are the same. So, the crossover operation does not generate new individuals and therefore, does not work. To avoid the problem, the substitution operation is added after the crossover so that there is no duplication of the same individuals in the population [17].

Even though many methods use this simplification of operators, some of the theorems have a significant impact in GP schema. More different the beginning building blocks are, more efficient is the search in GP. Large populations of individuals result in convergence to better solutions. The drawback of using large building blocks is that they take more time to evaluate the fitness function of individuals. The analysis done in the paper confirms that if the required building block is not found in initial iterations, the tree takes longer to converge to a solution [16].

3. Competitive Analysis of Genetic Programming

In GP there exist building blocks which in the start of evolution compete with other schema and retain their occurrence at root level till the end of evolution. Moreover, they extend their selves in deeper depths and remain useful, promising and persistent. More specifically there is promising schema and their occurrence at the root level is high (or we can say building blocks) in earlier generations increase their occurrence at the root level till the end of evolution by competing with other schemas. This behaviour of Genetic programming helps it to solve the problem. For example, if we are solving regression problem (x^4 + x^3 + x^2 + x) then the schema containing nodes x, +, * are useful and can be the most promising and persistent schema or simply says building blocks as shown in figure 1.

![Figure 1. Competition among a group of schemas](image)

Keeping in view the competitive behaviour of genetic programming, we started performing a competitive analysis of GP. For this purpose, we first formulated hypothesis for competitive analysis. Which tells 1) There exist some schemas when we search them in first generations population, they give us valuable occurrence at the root level and they increase their occurrence at root level as the generation passes till the end of evolution. 2) There exist small schemas (schemas with minimal depth) which give us valuable occurrence at root level initially and increase their occurrence till the end of evolution and if we extend those smaller schemas into larger ones then, search them again, it also gives us valuable occurrence at the root level. It can be treated as a useful material to evolve the specific problem.3) There are groups of
schematics whose occurrence at root level in first generations is valuable, but only a few of those increase their occurrence at root level till the end of evolution and remaining schematics in those groups decrease their occurrence at root level as the generations pass. It clarifies the fact that each schema in such a group competes with rest of schematics in the group, only a few are successful in the competition, and most of them gradually vanish as the generations pass. 4) There exist some particular schematics which are very useful. They increase their occurrence at root level in almost every run with a different population.

Once the hypothesis is made we searched different schematics of different depths in the population to analyze competitive behaviour in Genetic programming (GP), we selected regression problem as a benchmark and evolved it ten times with different populations. The Schematics searched for a single run are compared with the population of different generations occurring randomly until the end. For this study, we set a generation threshold = 5. After that, we compared each schematic with the population of 5th, 10th, 15th generation till the end of evolution.

By performing this experiment on each run, we get our results in tabular form and are ready for analyzing the competitive behaviour of Genetic programming (GP). Once we got our desired results in a tabular form, we started our analysis on our hypothesis, which is mainly based on competition.

3.1. Competitive Analysis (Results)

Table 1 shows the competition between the different schematic of Depth 1 from this particular run. For discussion sake, we take some schematic of depth one into account showed in table 1 given below. The entire schematic with a higher count in 10th generation competes with each other. In the 10th Generation, the occurrence of majority schematic lies between 2 and 1 at the root level. Therefore, we analyzed that there is a competition between them. According to our hypothesis, only a few schematics are persistent and gradually increase their occurrence at root level till the end of evolution. They are the most useful material concerning our problem, whereas remaining schematics are persistent and gradually vanished. By looking at table 1, we can see that (+ X *) remains persistent, and it gradually increased its occurrence over generations. In fifth generation (+ X *) occurred for zero time, in 10th, 15th, 20th, 25th and 28th generation its occurrence was 2, 3, 1, 19 and 56 respectively. As we are solving regression problem we have to evolve (x4 + x3 + x2 + x), Therefore (+ X *) is increasing its occurrence at root level as the generations pass so its useful material for our desired evolution. However, others vanish with time.

3.2. Results Depth 2 Schema

Table 2 shows the competition between the different schematic of Depth 2 from this particular run. For discussion sake, we take some schematic of depth two into account as shown in table 2.

Table 1. Results with minimal depth Schema

| Searched Schema | Occurrence At root Level |
|-----------------|--------------------------|
|                 | Generation 5 | Generation 10 | Generation 15 | Generation 20 | Generation 25 | Generation 28 |
| + X *           | 0            | 2             | 3             | 1             | 19            | 56            |
| * C sinX        | 0            | 2             | 2             | 0             | 0             | 0             |
| * X rlog        | 0            | 2             | 0             | 0             | 0             | 0             |
| * C +           | 0            | 1             | 2             | 0             | 2             | 0             |
| ( + - exp )     | 0            | 1             | 2             | 0             | 0             | 0             |
| * C rlogC       | 0            | 1             | 0             | 0             | 0             | 0             |
| + rlogC -       | 0            | 1             | 0             | 0             | 0             | 0             |
| rlog +          | 0            | 1             | 0             | 0             | 0             | 0             |

The entire schematic with a higher count in 25th generation competes with each other. In 25th Generation, the occurrence of majority schematic lies between 6 and 1 at the root level. Therefore we analyzed that there...
is a competition between them. According to our hypothesis, only a few schemas are persistent and gradually increase their occurrence at root level till the end of evolution. They are the most useful material concerning our problem, whereas remaining schemas are persistent and gradually vanished. By looking at table 2, we can see that (+ $X * + X$) remains persistent, and it gradually increased its occurrence over generations. In 5th, 10th, 15th, 20th its occurred for zero time, in 25th and 28th its occurrence was 6 and 44 respectively. Same with (* $X + X *$), (* $\sin X \sin X$), (+ $X * \sin X$). As we are solving regression problem we have to evolve ($x^4 + x^3 + x^2 + x$). Therefore (+ $X * + X$), ( * $X * X$ *) and ( * $X + X \sin X$) is increasing its occurrence at root level as the generations passes so its useful material for our desired evolution. However (+ $X \sin X$) maintains its occurrence up to some level and others Vanish with time.

Table 2. Results with depth 2 Schema

| Searched Schema | Occurrence At Root Level |
|-----------------|--------------------------|
|                 | Generation 5 | Generation 10 | Generation 15 | Generation 20 | Generation 25 | Generation 28 |
| (+ $X * + X$)   | 0            | 0              | 0              | 0              | 6              | 44             |
| + $X \sin X$    | 0            | 0              | 0              | 0              | 5              | 5              |
| + $C X$         | 0            | 0              | 0              | 0              | 5              | 0              |
| X               | 0            | 0              | 0              | 0              | 3              | 3              |
| *$X X$          | 0            | 0              | 0              | 0              | 3              | 1              |
| +$X*X*$         | 0            | 0              | 0              | 0              | 3              | 1              |
| $\sin X \sin X$| 0            | 0              | 0              | 0              | 2              | 2              |
| *+$X*X$         | 0            | 0              | 0              | 0              | 2              | 1              |
| +$X*CX$         | 0            | 0              | 0              | 0              | 2              | 0              |

3.3. Results for the same schema across the run

Results Depth 1 Schema: We also analyzed that there exist some particular schemas which are very useful. They increase their occurrence at root level in almost every run with a different population. Following are the analysis of the schema which are same, useful, promising and persistent in each run. We analyzed ten different runs of a symbolic regression problem, which shows that some same schema across the runs are useful and increase their occurrence at root level gradually till the last population and rest vanished with time. Following results validates it. We run regression problem ten times we are discussing only two runs here. We analyzed after this experiment that there exist schemas across the runs which are useful and promising based on their persistent occurrence on root level with time. There is some schema of depth one which remains the same across the runs. In the following tables, 3 and 4 the highlighted rows show the particular + $X *$ and + $X \sin X$ schema remained persistent and promising throughout the evolution in all the three runs. As discussed earlier, this is the most useful material to get good results in the existing regression problem.
### Results Depth 2 Schema:
We are discussing only two runs for depth two as well. We analyzed after this experiment that there exist schemas across the runs, which are useful and promising based on their persistent occurrence on root level with time. In the following tables 5 and 6, the highlighted rows show the particular (+ X * +X)schema remained persistent and promising throughout the evolution in both runs. As discussed earlier, this is the most useful material to get good results in the existing regression problem. Similar is the case with depth three and more schema.

### 4. Co-Operative Analysis Of GP
In GP there exists building blocks which in the start of evolution present in the levels other than root and gradually move themselves to the root level and retain their occurrence at root level till the end of evolution. Also, these schemas are useful, promising and persistent throughout the evolution. More specifically there is a schema which is present in the levels other than root (or we can say building blocks) in earlier generations increase their occurrence at the root level as the generations pass till the end of evolution. This behaviour of Genetic programming helps it to solve the problem. For example, if we are solving regression problem (x₄ + x₃ + x₂ + x) then the schema containing nodes x, +, * are useful and can be the most promising and persistent schema or simply says building blocks as shown in figure 2.

### Table 3. Results with depth 1 Schema (Run 1)

| Searched Schema | Occurrence At root Level |
|-----------------|--------------------------|
|                 | Generation 5 | Generation 10 | Generation 15 | Generation 20 | Generation 25 | Generation 30 |
| + X *           | 0            | 0             | 3             | 4             | 18            | 21            |
| sinC            | 0            | 0             | 3             | 0             | 0             | 0             |
| cos -           | 0            | 0             | 3             | 0             | 0             | 0             |
| * X sin         | 0            | 0             | 2             | 1             | 1             | 1             |
| / sin X +       | 0            | 0             | 2             | 0             | 1             | 0             |
| + sinX *        | 0            | 0             | 1             | 1             | 1             | 1             |
| sin cosX        | 0            | 0             | 1             | 0             | 0             | 0             |
| sin cosC        | 0            | 0             | 1             | 0             | 0             | 0             |
| * + X           | 0            | 0             | 1             | 0             | 0             | 0             |
| (- X sinX)      | 0            | 0             | 1             | 0             | 0             | 0             |
| (+ X sin)       | 0            | 0             | 1             | 5             | 7             | 7             |

### Table 4. Results with depth 1 Schema (Run 2)

| Searched Schema | Occurrence At root Level |
|-----------------|--------------------------|
|                 | Generation 5 | Generation 10 | Generation 13 |
| X               | 13           | 13            | 1             |
| sinX            | 7            | 5             | 0             |
| / sin exp       | 7            | 0             | 0             |
| + X *           | 6            | 35            | 61            |
| (- X sinX)      | 6            | 0             | 0             |
| (X C)           | 5            | 0             | 0             |
| * X X           | 4            | 4             | 2             |
| expX            | 3            | 3             | 2             |
| sin +           | 3            | 0             | 0             |
### Table 5. Results with depth 2 Schema (Run 1)

| Searched Schema | Occurrence At root Level |
|-----------------|--------------------------|
|                 | Generation 5 | Generation 10 | Generation 13 |
| (+ X * + X)     | 0            | 2              | 38            |
| rlog - C X      | 0            | 2              | 0             |
| exp + X *       | 0            | 2              | 0             |
| + X C           | 0            | 2              | 0             |
| (+ X sin sin)   | 0            | 2              | 0             |
| + * - X + * X   | 0            | 1              | 5             |
| * - exp X cos X X | 0           | 1              | 2             |
| (+ X exp X)     | 0            | 1              | 1             |

### Table 6. Results with depth 2 Schema (Run 2)

| Searched Schema | Occurrence At Root Level |
|-----------------|--------------------------|
|                 | Generation 5 | Generation 10 | Generation 15 | Generation 20 | Generation 25 | Generation 28 |
| (+ X * + X)     | 0            | 0              | 0              | 0              | 6             | 44            |
| + X sin *       | 0            | 0              | 0              | 0              | 5             | 5             |
| + C X           | 0            | 0              | 0              | 0              | 5             | 0             |
| X               | 0            | 0              | 0              | 0              | 3             | 3             |
| *X X            | 0            | 0              | 0              | 0              | 3             | 1             |
| +X*X*           | 0            | 0              | 0              | 0              | 3             | 1             |
| sin * sin X sin X | 0           | 0              | 0              | 0              | 2             | 2             |
| +X*X*           | 0            | 0              | 0              | 0              | 2             | 1             |
| +X*X X          | 0            | 0              | 0              | 0              | 2             | 0             |
Keeping in view the co-operative behaviour of genetic programming, we started performing co-operative analysis of GP. For this purpose, we first formulated hypothesis for co-operative analysis, Which tells 1) There are schema exists which are persistent and promising till the end of evolution. These schemas are the most valuable and useful for that specific problem. 2) The schema discussed above have high occurrence at levels other than the root in the earlier generations but are gradually pushed towards root as generations pass.

Once the hypothesis is made we searched different schemas of different depths in the population to analyze co-operative behaviour in Genetic programming (GP), we selected regression problem as a benchmark and evolved it ten times with different populations. This searched schema is then investigated in the population from earlier generations to last generations to analyze co-operative behaviour.

4.1. Results for Co-operative Analysis
Results (Run 1) of Schema: + X * Figure 3 shows that the schema (+ X *) in Run 1 is a persistent and promising schema from the start till the end of evolution. In this figure red line shows the occurrence of the schema (+ X *) at levels other than the root level over the generations and the blue line shows the occurrence of the same schema (+ X *) at the root level for Run 1. If we carefully analyze the figure 1 we get to know there exists co-operative behaviour in it, because the occurrence of the schema (+ X *) at other levels is more promising in earlier generations but later on (+ X *) is pushed towards root level as clearly shown in figure 3.

Results (Run 2) of Schema: + X * Figure 4 shows that the schema (+ * X) in Run 2 is a persistent and promising schema from the start till the end of evolution. In this figure red line shows the occurrence of the
schema (+ * X) at levels other than the root level over the generations and the blue line shows the occurrence of the same schema (+ * X) at the root level for Run 2. If we carefully analyze the figure 2 we get to know there exists co-operative behaviour in it, because the occurrence of the schema (+ * X) at other levels is more promising in earlier generations but later on (+ * X) is pushed towards root level as clearly shown in figure 4.

Figure 4. Results for Co-operative behaviour (Run 2)

5. Discussion
The above experimental results clearly show that evolution is driven by competitive and co-operative behaviour. Now we further validated our hypothesis by making the problem statement tougher. Our benchmark problem was Quartic Symbolic regression i-e x4 + x3 + x2 + x. We make the problem harder by increasing the polynomial as x6 + x5 + x4 + x3 + x2 + x, evolve it and then once again investigated the competitive and co-operative behaviour by searching schema.

We took the population of 100, and in the last generation, 74 individuals got (+ x *) at the root level in a particular run which is the primary genetic material for this particular problem.

5.1. Manual insertion of Genetic Material
We further validated our hypothesis by manually inserting genetic material or useful schema e-g (+ x *) in the initial population at the leaf level, and observed that particular schema from the very 1st generation to the last generation and we concluded that it gradually moves itself to the root level and retain its occurrence at root level till the end of evolution. Also, statically it gives us the same results.

6. Conclusion
These experiments validate the hypothesis that the evolution is driven by co-operative rooted building blocks as well, along with the competitive building blocks. The results show the count of building blocks in different positions never reaches zero, which indicates the presence of the genetic material required for the construction of the building block at a certain position in the form of co-operation and competition with other schemas, which, at a later stage, may move to the correct position (root level). It is fall out that GP, by combining these competitive building blocks appear at root level uncovers the problem solution over time. For that reason, by identifying, preserving and exchanging these building blocks explicitly, the system enactment can be improved severely. Additionally, the solutions of the developed versions of the same problem can also be handled through this.

6.1. Limitations and Recommendations for Future Research
This work also opens several avenues for future research as well. Because by identifying, preserving and changing explicitly these building blocks, system enactment can be improved easily. So in future research can be done for devising a technique which increases the occurrence of these building blocks through GP operators.
References

[1] R. Poli and W. B. Langdon, "Schema theory for genetic programming with one-point crossover and point mutation," *Evolutionary Computation*, vol. 6, no. 3, pp. 231-252, 1998.

[2] R. Poli and W. B. Langdon, "On the search properties of different crossover operators in genetic programming," *Genetic Programming*, pp. 293-301, 1998.

[3] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press, 1992.

[4] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," *Statistics and Computing*, vol. 4, no. 2, pp. 87-112, 1994.

[5] U.-M. O'Reilly, "An analysis of genetic programming," Carleton University, 1995.

[6] U.-M. O. R. F. Oppacher, "The troubling aspects of a building block hypothesis for genetic programming," *Foundations of Genetic Algorithms 1995 (FOGA 3)*, vol. 3, p. 73, 2014.

[7] J. P. Rosca, "Analysis of complexity drift in genetic programming," *Genetic Programming*, pp. 286-294, 1997.

[8] C. Ryan, H. Majeed, and A. Azad, "A competitive building block hypothesis," in *Genetic and Evolutionary Computation Conference*, 2004, pp. 654-665: Springer.

[9] H. Majeed, "A new approach to evaluate GP schema in context," in *Proceedings of the 7th annual workshop on Genetic and evolutionary computation*, 2005, pp. 378-381.

[10] R. Poli, N. F. McPhee, and J. E. Rowe, "Exact schema theory and markov chain models for genetic programming and variable-length genetic algorithms with homologous crossover," *Genetic Programming and Evolvable Machines*, vol. 5, no. 1, pp. 31-70, 2004.

[11] N. F. McPhee, B. Ohs, and T. Hutchison, "Semantic building blocks in genetic programming," in *European Conference on Genetic Programming*, 2008, pp. 134-145: Springer.

[12] W. Smart, "Exploring and Exploiting Fragments," Citeseer, 2006.

[13] W. Smart, P. Andreae, and M. Zhang, "Empirical analysis of GP tree-fragments," in *European Conference on Genetic Programming*, 2007, pp. 55-67: Springer.

[14] M. Aichour and E. Lutton, "Co-operative co-evolution inspired operators for classical GP schemes," in *Nature Inspired Cooperative Strategies for Optimization (NICOSEO 2007)*: Springer, 2008, pp. 169-178.

[15] L. C. J. Beadle, "Semantic and structural analysis of genetic programming," Citeseer, 2009.

[16] W. Smart and M. Zhang, "Empirical analysis of schemata in genetic programming using maximal schemata and MSG," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 2008, pp. 2983-2990: IEEE.

[17] J. Wang, M. Zhang, O. K. Ersoy, K. Sun, and Y. Bi, "An Improved Real-Coded Genetic Algorithm Using the Heuristical Normal Distribution and Direction-Based Crossover," *Computational Intelligence and Neuroscience*, vol. 2019, 2019.