An Analysis of Geometric Semantic Crossover: A Computational Geometry Approach

Mauro Castelli[1], Luca Manzoni[2], Ivo Gonçalves[1,5], Leonardo Vanneschi[1], Leonardo Trujillo[3] and Sara Silva[4,5]

[1]*NOVA IMS, Universidade Nova de Lisboa, 1070-312 Lisboa, Portugal*

[2]*DISCo, Universitá degli Studi di Milano Bicocca, 20126 Milano, Italy*

[3]*Posgrado en Ciencias de la Ingeniería, Instituto Tecnológico de Tijuana, Tijuana, Mexico*

[4]*BioISI, Faculty of Sciences, University of Lisbon, Campo Grande, 1749-016 Lisbon, Portugal.*

[5]*CISUC, Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra, Portugal*

Keywords: Genetic Programming, Semantics, Convex Hull.

Abstract: Geometric semantic operators have recently shown their ability to outperform standard genetic operators on different complex real world problems. Nonetheless, they are affected by drawbacks. In this paper, we focus on one of these drawbacks, i.e. the fact that geometric semantic crossover has often a poor impact on the evolution. Geometric semantic crossover creates an offspring whose semantics stands in the segment joining the parents (in the semantic space). So, it is intuitive that it is not able to find, nor reasonably approximate, a globally optimal solution, unless the semantics of the individuals in the population "contains" the target. In this paper, we introduce the concept of convex hull of a genetic programming population and we present a method to calculate the distance from the target point to the convex hull. Then, we give experimental evidence of the fact that, in four different real-life test cases, the target is always outside the convex hull. As a consequence, we show that geometric semantic crossover is not helpful in those cases, and it is not even able to approximate the population to the target. Finally, in the last part of the paper, we propose ideas for future work on how to improve geometric semantic crossover.

1 INTRODUCTION

Methods to integrate semantic awareness gained a vast popularity in the Genetic Programming (GP) (Koza, 1992) community in the last few years (Vanneschi et al., 2014a). In particular, in the last three years, a noteworthy attention was dedicated to Geometric Semantic GP (GSGP), a version of GP introduced by Moraglio and coauthors in 2012 (Moraglio et al., 2012), that uses so called Geometric Semantic Operators (GSOs), instead of the traditional crossover and mutation.

Even though the term semantics can have several different interpretations, it is a common trend in the GP community (and this is also the definition we adopt here) to identify the semantics of a solution with the vector of its output values on the training data (Vanneschi et al., 2014a; Moraglio et al., 2012). Under this perspective, a GP individual can be identified with a point (its semantics) in a multidimensional space that we call semantic space, which has a number of dimensions equal to the number of training instances. The objective of GSOs is to create transformations on the syntax of individuals that correspond to precise operators of Genetic Algorithms (GAs) in the semantic space. More in particular, GSOs are Geometric Semantic Crossover (GSXO) and Geometric Semantic Mutation (GSM). GSXO corresponds to geometric crossover in the semantic space, in the sense that it generates an offspring whose semantics stands in the segment joining the semantics of the parents. GSM corresponds to geometric mutation (also called ball or box mutation (Vanneschi et al., 2013)) in the semantic space, in the sense that if we mutate an individual x, we obtain an individual y such that the semantics of y is a weak perturbation of the semantics of x.

One of the motivations for the success of GSGP probably lies in the fact that GSOs induce a unimodal fitness landscape on any supervised learning

Castelli, M., Manzoni, L., Gonçalves, I., Vanneschi, L., Trujillo, L. and Silva, S.
An Analysis of Geometric Semantic Crossover: A Computational Geometry Approach.
DOI: 10.5220/0006056402010208
In *Proceedings of the 8th International Joint Conference on Computational Intelligence (IJCCI 2016) - Volume 1: ECTA*, pages 201-208
ISBN: 978-989-758-201-1

problem (like for instance classification or regression), thus favoring GP evolvability. Also thanks to an efficient implementation of GSGP that was defined in 2013 (Vanneschi et al., 2013; Castelli et al., 2015a), it was possible to successfully apply GSGP to several different complex real-life applications (see for instance (Castelli et al., 2014, 2015b, 2013a)). However, GSGP has at least the following recognized drawbacks: (1) GSOs generate individuals that are larger than their parents, and this causes a rapid growth in the size of the individuals in the population; (2) GSXO was shown to be quite ineffective on a large set of applications.

The former problem is widely discussed in literature. The implementation proposed in (Vanneschi et al., 2013; Castelli et al., 2015a) is a workaround to this problem, in the sense that, although not limiting the code growth, it makes the system not only usable in practice, but even more efficient than standard GP.

This paper focuses on the latter drawback, already pointed out in the literature for instance in (Moraglio and Mambrini, 2013), where a purely mutation-based GSGP was proposed, after recognizing the uselessness of GSXO. We believe that one of the reasons for the poor performance of GSXO lies in its geometric property. In fact, as we said above, GSXO generates an offspring whose semantics stands in the segment joining the semantics of the parents. In this perspective, if we imagine a GP population as a cloud of points in the semantic space, we could informally say that crossover is only able to generate points that are "inside" the cloud. So, if the target (that is also a known point in the semantic space) is not contained inside the cloud, GSXO will never be able to generate it. Also, if the target is quite far from the cloud, GSXO will not be even able to reasonably approximate it.

The main objective of this paper is to confirm this hypothesis by means of a set of experiments. For achieving this objective, we need a formal tool that allows us to "capture" our idea of cloud of individuals in the semantic space. More specifically, it would be useful to have a formal method to indicate what we could informally call the "border" of a cloud. In this way, we could use this tool both for understanding if a given point is "inside" or "outside" the cloud and for calculating the distance from one point to the cloud. Contributions of this paper are: (1) Introduction of the concept of *convex hull*, as a tool to represent the "border" of a set of points in the semantic space. (2) Introduction of a method to understand if a point is contained in the convex hull or not. (3) Introduction of a method to calculate the distance from a point to the convex hull.

The first contribution has already been considered in (Moraglio, 2011), where authors showed that all the evolutionary algorithms using geometric crossover with no mutation perform the same form of convex search regardless of the underlying representation, the specific selection mechanism, the specific offspring distribution, the specific search space, and the problem at hand.

With the contributions provided in our study, we are able to monitor the convex hull of the points representing the semantics of all the individuals in the population during the GP evolution. In particular, we able to study the evolution of the distance from the target to the convex hull during a GP run.

In this paper, we compare two GSGP systems: the first one uses both GSXO and GSM; the second one uses only GSXO. The different behaviour of the latter, compared to the first, should allow us to shade a light on the limitations of GSXO. As test cases for this experimental study, we have decided to use four real-life symbolic regression problems from the UCI repository (Lichman, 2013).

2 GEOMETRIC SEMANTIC OPERATORS

GSOs are becoming more and more popular in the GP community (Vanneschi et al., 2014a), probably because of their property of inducing a unimodal fitness landscape on any problem consisting in matching sets of input data into known targets (like for instance supervised learning problems, such as regression and classification). To have an intuition of this property (whose proof can be found in (Moraglio et al., 2012)), let us first consider a Genetic Algorithms (GAs) problem in which the unique global optimum is known and the fitness of each individual (to be minimized) corresponds to its distance to the global optimum (our reasoning holds for any employed distance). In this problem, if we use, for instance, *ball mutation* (Krawiec and Lichocki, 2009) (i.e. a variation operator that slightly perturbs some of the coordinates of a solution), then any possible individual different from the global optimum has at least one fitter neighbor (individual resulting from its mutation). So, there are no local optima. In other words, the fitness landscape is unimodal, and consequently the problem is characterized by a good evolvability. Similar considerations hold for many types of crossover, including various kinds of geometric crossover (Krawiec and Lichocki, 2009).

Now, let us consider the typical GP problem of finding a function that maps sets of input data into known target values (as we said, regression and classification are particular cases). The fitness of an individual for this problem is typically a distance between its predicted output values and the target ones (error measure). GSOs simply define transformations on the syntax of the individuals that correspond to geometric crossover and ball mutation in the semantic space, thus allowing us to map the considered GP problem into the previously discussed GA problem.

Geometric semantic crossover (GSXO)[1] generates, as the unique offspring of parents $T_1, T_2 : \mathbb{R}^n \to \mathbb{R}$, the expression:

$$T_{XO} = (T_1 \cdot T_R) + ((1 - T_R) \cdot T_2)$$

where T_R is a random real function whose output values range in the interval $[0,1]$.

Analogously, *geometric semantic mutation* (GSM) returns, as the result of the mutation of an individual $T : \mathbb{R}^n \to \mathbb{R}$, the expression:

$$T_M = T + ms \cdot (T_{R1} - T_{R2})$$

where T_{R1} and T_{R2} are random real functions with codomain in $[0,1]$ and ms is a parameter called mutation step.

Moraglio et al. (Moraglio et al., 2012) show that GSXO corresponds to geometric crossover in the semantic space (i.e. the point representing the offspring stands on the segment joining the points representing the parents) and GSM corresponds to ball mutation on the semantic space (the semantics of the individual generated by mutation is a weak perturbation of the semantics of the individual to which mutation is applied), and thus GSM induces a unimodal fitness landscape on the above mentioned types of problem.

3 CONVEX HULL

This section reports simple computational geometry concepts that will be used in the following sections to analyze the performance of GSXO. The following definitions are taken from (de Berg et al., 2008). A subset S of the plane is called convex if and only if for any pair of points p, $q \in S$ the line segment pq is completely contained in S. The convex hull $\mathbb{CH}(S)$ of a set S is the smallest convex set that contains S. In other terms, it is the intersection of all convex sets that contain S. To simplify, it is possible to visualize what the convex hull looks like by a thought experiment (taken from (de Berg et al., 2008)). Imagine that the points are nails sticking out of the plane. Take an elastic rubber band, hold it around the nails, and let it go. It will snap around the nails, minimizing its length. The area enclosed by the rubber band is the convex hull of the set of points. This leads to an alternative definition of the convex hull of a finite set P of points in the plane: it is the unique convex polygon whose vertices are points from P and that contains all points of P. It is possible to prove that this definition is equivalent to the one given earlier (de Berg et al., 2008).

While several algorithms have been proposed to efficiently determine the convex hull of a set of points, the large majority of them considers only a 2-dimensional or 3-dimensional space. In this work, we need to find the convex hull in an n-dimensional space, where the size of the space n is determined by the independent variables that characterize the particular application at hand.

For this reason, in our work we follow a different approach. Instead of directly building the convex hull, we try to understand if a point (that successively in our experiments will be the target) is inside the convex hull formed by a set of points (i.e., the semantics of the candidate solutions). To do that we follow the method reported in Chapter 11 of (Boyd and Vandenberghe, 2004). Basically the idea is to solve a system of linear equations subjects to some constraints. If a solution to the system exists then it is possible to conclude that a given point is inside the convex hull.

The method, described in detail in (Boyd and Vandenberghe, 2004), re-adapted by us for GP, is the following: let n be the number of individuals in a population and m the number of training samples. Let $x_{i,j}$ be the *signed error* of the i-th individual on the j-th training sample. We can the build the following system of linear equations in the n variables $a_1, \ldots, a_n$:

$$\begin{aligned}
&a_1 x_{1,1} + a_2 x_{2,1} + \ldots + a_n x_{n,1} = 0\\
&a_1 x_{1,2} + a_2 x_{2,2} + \ldots + a_n x_{n,2} = 0\\
&\vdots\\
&a_1 x_{1,m} + a_2 x_{2,m} + \ldots + a_n x_{n,m} = 0\\
&a_1 + a_2 + \ldots + a_n = 1\\
&a_i \geq 0 \qquad\qquad \forall i \in \{1, \ldots, n\}
\end{aligned}$$

The previous system has a solution (i.e., a vector $(a_1, a_2, \ldots, a_n)$ if and only if the optimal individual (i.e., the one that has zero error on the training set) can be expressed as a linear combination of the existing individuals where the coefficients are $a_1, a_2, \ldots, a_n$. This is actually equivalent of saying that the optimum resides in the convex hull given by the points

[1]Here we report the definition of the geometric semantic operators as given by Moraglio et al. for real functions domains, since these are the operators we will use in the experimental phase. For applications that consider other types of data, the reader is referred to (Moraglio et al., 2012).

$x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,m})$ (i.e., the signed error vectors of the individuals). This is a powerful tool that allows us to build the optimal solution by combining existing individuals. However, this is possible only when the above mentioned system of linear equations has a solution (i.e., the optimum is in the convex hull).

Interestingly, it is also possible to change the previous problem, hence achieving more information, by finding the distance from the optimum to the convex hull, instead of simply asking whenever the optimum is inside it. This can be performed by solving the following linear programming problem in the variables $a_1, \ldots, a_n$, $e_1, \ldots, e_m$, and $\bar{e}_1, \ldots, \bar{e}_m$:

$$\begin{aligned}
\textbf{minimize}\ & e_1 + \bar{e}_1 + e_2 + \bar{e}_2 + \ldots + e_m + \bar{e}_m \\
& \text{with constraints:} \\
& a_1 x_{1,1} + a_2 x_{2,1} + \ldots + a_n x_{n,1} + e_1 - \bar{e}_1 = 0 \\
& a_1 x_{1,2} + a_2 x_{2,2} + \ldots + a_n x_{n,2} + e_2 - \bar{e}_2 = 0 \\
& \vdots \\
& a_1 x_{1,m} + a_2 x_{2,m} + \ldots + a_n x_{n,m} + e_m - \bar{e}_m = 0 \\
& a_1 + a_2 + \ldots + a_n = 1 \\
& a_i \geq 0, \forall i \in \{1, \ldots, n\} \\
& e_i \geq 0, \forall i \in \{1, \ldots, m\} \\
& \bar{e}_i \geq 0, \forall i \in \{1, \ldots, m\}
\end{aligned}$$

First of all, notice that the previous system has always a solution and that only one between e_i and $\bar{e}_i$ can be non-zero when $\sum_{i=1}^{m}(e_i + \bar{e}_i)$ is minimized. The term $e_i + \bar{e}_i$ is always positive and represents the distance (along the i-th coordinate) from the convex hull to the global optimum. Recall that since all points are in an m-dimensional space, $\sum_{i=1}^{m}(e_i + \bar{e}_i)$ represents a distance - the commonly called *taxicab distance* - from the convex hull to the global optimum. We have decided not to use the more common euclidean distance since it would have required a non-linear target, making the problem non-linear. When the distance is zero the global optimum is inside the convex hull and, as before, the values of $a_1, \ldots, a_n$ give us a way to combine existing solutions to build an optimal solution. To illustrate these facts, we can observe in Figure 1 the convex hull generated by a population of 10 individuals (represented as points) and what is the distance from the optimum (the vector of all zeros, representing no error on the training set) to the convex hull. The point inside the convex hull and closest to the optimum is (c), which is a linear combination of two other individuals, (a) and (b).

Since linear programming problems can be efficiently solved by the *internal points* method (Bonnans et al., 2006) or by the *simplex* method (even if, contrarily to the former one, the latter can have an exponential runtime), it is feasible to compute the distance from the optimum to the convex hull generated by the current population at each generation.

4 EXPERIMENTAL SETTINGS

As test problems we used four symbolic regression problems from the UCI repository (Lichman, 2013). The problems were chosen to have a low number of features to reduce the number of variables when computing the distance from the convex hull:

- Airfoil Self-Noise (Airfoil), with 1502 instances and 5 features;
- Concrete Compressive Strength (Concrete), with 1029 instances each with 8 features;
- Concrete Slump Test (Slump), with 102 instances and 9 features;
- Yacht Hydrodynamics (Yacht), with 307 instances each with 6 features.

Each dataset was split into 100 pairs of training and test sets, the former containing 70% of the instances (chosen at random with uniform distribution), and the latter the remaining 30% of the instances. For all the considered test problems, a total of 100 runs have been performed with each technique. In each run, a different partition between training and test data has been considered. All the runs used populations of 100 individuals allowed to evolve for 1000 generations. Trees initialization was performed using the Ramped Half-and-Half method (Koza, 1992) with a maximum initial depth equal to 6. The function set contained the four binary arithmetic operators, including protected division as in (Koza, 1992). The terminal set contained a number of variables equal to the number of features in the dataset, plus 100 random constants. These constants were generated randomly with uniform distribution in the range $[-100, 100]$ at the beginning of each run. Survival from one generation to the other was always guaranteed to the best individual of the population (elitism). A random mutation step (generated with uniform distribution in the range $[0, 1]$) has been considered in each mutation event. GSXO and GSM probabilities were equal to 0.9 and 0.5 respectively. These rates have been selected based on the guidelines reported in (Castelli et al., 2015c).

Since mutation is the only operator that can produce solution *outside* the convex hull generated by the current population, we have also explored the effect of a crossover-only evolution by performing the same tests with a mutation rate of zero (i.e., no mutation). Notice that, in this way, the improvement possible by

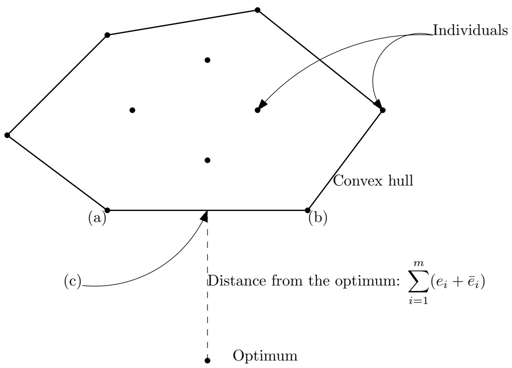

Figure 1: An example of how the convex hull is related to the optimal solution. Point (c), which is the closest point in the convex hull to the target, is a linear combination between (a) and (b), that are points that belong to the border of the convex hull.

GSGP are limited by the best possible solution that can be found inside the convex hull, therefore we expect that the fitness will rapidly reach a plateaux from which improvements are no further possible.

At each generation of each test we computed the distance of the optimum from the convex hull using the *lp_solve* linear programming solver (http://sourceforge.net/projects/lpsolve/). In total, over 8×10^5 linear programming problems have been solved.

5 EXPERIMENTAL RESULTS

In the first part of our experimental study, we analyzed the fitness of the best individual in the population at each generation, both on training and test sets. The two systems that have been compared are the "usual" GSGP, that employs both GSXO and GSM and a GSGP systems that uses only GSXO. Results of this analysis are reported in Figure 2. As it is possible to see, for all the studied test problems a similar pattern appears: when GSXO is the only genetic operator used, the best training and test fitness do not improve during the evolution. In other words, no evolution is taking place and the best individual obtained at the end of the search process has a fitness that is comparable to the one that was found in the very first part of the evolution. The behaviour of training and test fitness is different when the search process uses both GSXO and GSM. In fact, for all the test problems, both training and test fitness keep improving steadily until the end of the run.

For a better understanding of the behaviour just observed, in the second part of the experimental analysis we have taken into account, at each generation, the distance between the global optimum and the convex hull defined by the current population. Results of this analysis (obtained using a GSGP system that uses both GSXO and GSM and a GSGP system that uses GSXO only) are reported in Figure 3.

If we consider GSGP that uses only GSXO, in all the test problems the distance from the convex hull to the target remains practically constant during the whole evolution. Hence, by only using GSXO, GSGP is not able to "push" the search process close to a globally optimal solution. The solutions always remain inside the convex hull defined by the initial population. Looking at Figure 3, it is also possible to see a "jump" in the very first generations (usually the first two or three generations). In fact, the initial population usually contains several highly semantically different solutions. After the selection takes place in the very first generations, several of these solutions (the ones with poor fitness) do not survive and the convex hull, intuitively, covers a smaller area (i.e., hypervolume) of the semantic space. As we can observe, in all the single runs we performed, the global optimum is never enclosed in the convex hull, which clearly makes GSXO practically useless. In fact, all the individuals created by GSXO will lie in the convex hull obtained after the first generation of the search process. The situation is different when a combination of GSXO and GSM is used. In this case, in all the studied test problems, the distance from the convex hull to the global optimum steadily decreases for the whole evolution.

To conclude the experimental analysis, we study the relation between fitness (Figure 2) and distance (Figure 3). Figure 4 reports the scatter plots of the training fitness with respect to the distance. Looking at these plots, it is clear that, as expected, the two quantities are strongly correlated. It is worth pointing out that fitness is not exactly equal to the distance of the convex hull to the target: in principle, the convex hull changes at each iteration in position and size.

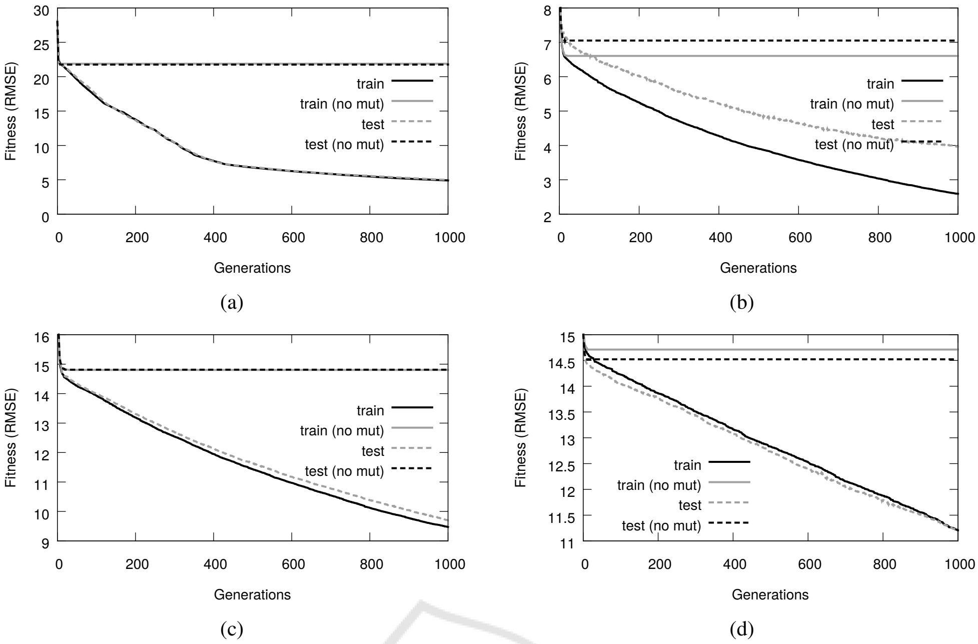

Figure 2: Training and test fitness for the considered test problems. Median calculated over 100 runs. (a) Airfoil dataset, (b) Slump, (c) Concrete, (d) Yacht.

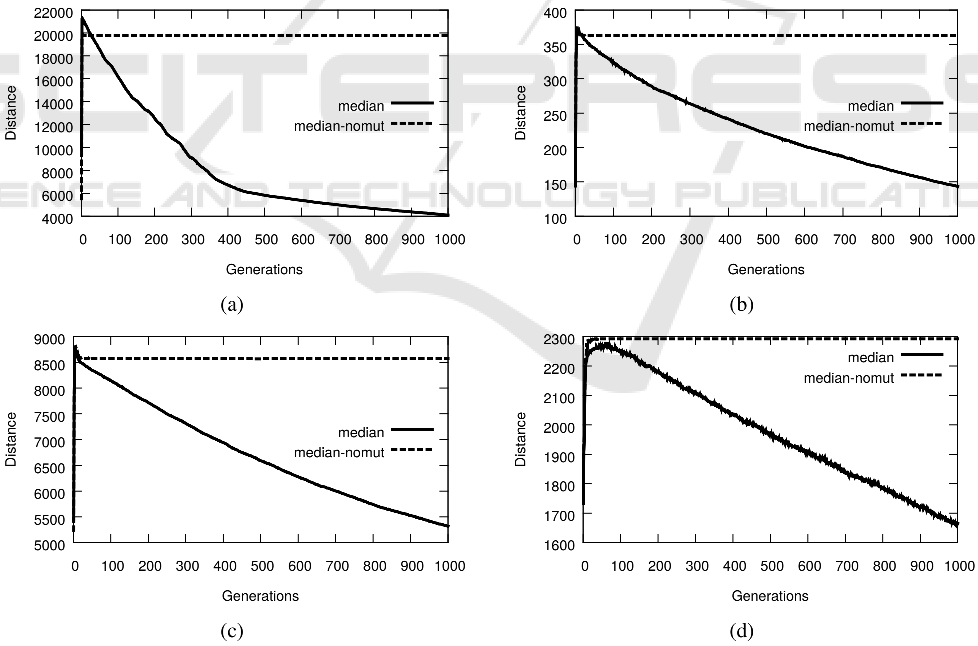

Figure 3: Distance between the convex hull and the global optimum. Median calculated over 100 runs. (a) Airfoil dataset, (b) Slump, (c) Concrete, (d) Yacht.

6 CONCLUSIONS AND FUTURE WORK

This paper contains a study aimed at motivating the poor performance of geometric semantic crossover in geometric semantic genetic programming. We could informally explain our intuition as follows: since it creates offspring that stand on the segment joining the parents in the semantic space, geometric semantic crossover is only able to create individuals that stand

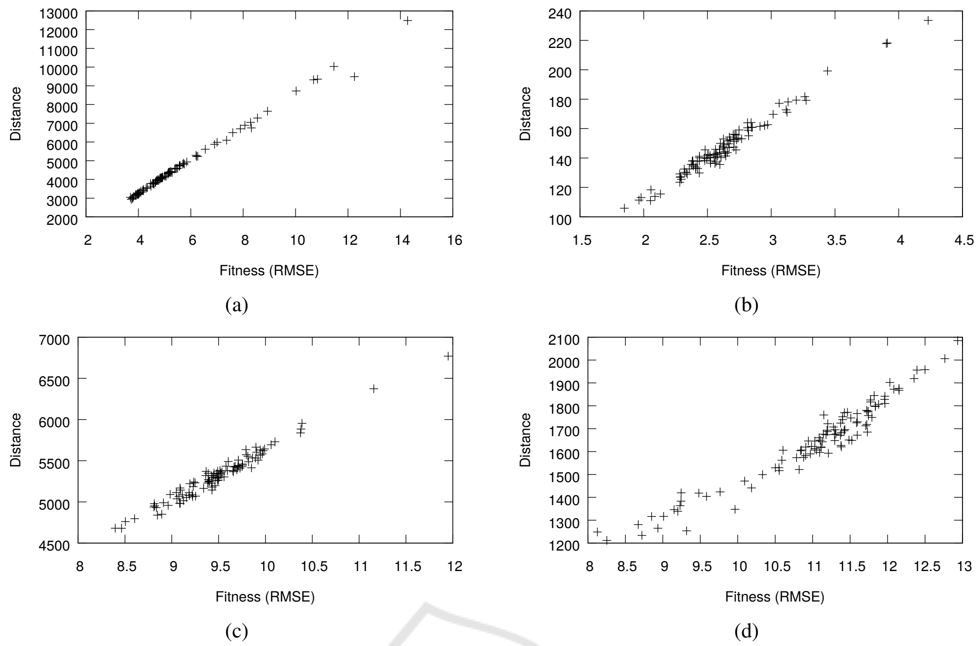

Figure 4: Scatter plot showing the correlation between training fitness and the distance from the convex hull to the global optimum. (a) Airfoil dataset, (b) Slump, (c) Concrete, (d) Yacht.

inside the area defined by the individuals already existing in the population. If the target is very far from that area, geometric semantic crossover is not be able to find it. To corroborate our hypothesis, we introduced a method to check whether a given point is contained in the convex hull or not, and a method to calculate the distance of a point to the convex hull. Using these notions, we have been able to experimentally demonstrate the appropriateness of our interpretation about the poor performance of geometric semantic crossover. In the first part of our experimental study, we have considered four real-life symbolic regression applications and we have shown that a system of geometric semantic genetic programming that uses only geometric semantic crossover was not able to evolve at all on those problems. In other words, the best fitness at the end of a run is comparable to the one that was found in the very first generations, both on the training and on the test set. At the same time, a system of geometric semantic genetic programming that uses both geometric semantic crossover and geometric semantic mutation is able to evolve, steadily improving fitness until the end of the run, both on the training and test sets. This confirms a behaviour that had already been observed several times in the literature: geometric semantic crossover gives a practically null contribution to the evolution, while the most useful genetic operator of geometric semantic genetic programming is geometric semantic mutation. As a second step of our experimental analysis, we have studied the evolution of the distance from the convex hull defined by the current population to the target, both when geometric semantic crossover is the only used genetic operator and when it is used with geometric semantic mutation. The presented results clearly show that when geometric semantic crossover is used in isolation, the distance from the convex hull to the target remains practically constant during the whole evolution, instead of the case when both operators are used, in which it steadily decreases for the whole duration of the run. This motivates the poor usefulness of crossover, corroborating our intuition: if (as in the studied test cases) the convex hull is rather far from the target, crossover is virtually useless, since it will never be able to generate a global optimum.

These findings pave the way for future work. In particular, we identify the possibility of searching for a set of individuals in the population, to which geometric semantic mutation could be applied, in such a way that the new convex hull, obtained after this mutation, contains the target. This is a very important objective, but still would use geometric semantic mutation as a crucial operator, thus confirming the idea that geometric semantic crossover, in isolation, is not effective.

ACKNOWLEDGEMENT

This research was partially supported by CONACYT Basic Science Research Project No. 178323, DGEST (México) Research Projects No. 5149.13-P and TIJ-ING-2012-110, TecNM (México) Research Projects 5414.14-P and 5621.15-P, as well as by FP7- Marie Curie-IRSES 2013 European Commission program with project ACoBSEC with contract No. 612689 and also by BioISI R&D unit, UID/MULTI/04046/2013 funded by FCT/MCTES/PIDDAC, Portugal.

REFERENCES

Bonnans, J. F., Gilbert, J. C., Lemaréchal, C., and Sagastizábal, C. A. (2006). *Numerical Optimization: Theoretical and Practical Aspects (Universitext)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA.

Castelli, M., Castaldi, D., Giordani, I., Silva, S., Vanneschi, L., Archetti, F., and Maccagnola, D. (2013a). An efficient implementation of geometric semantic genetic programming for anticoagulation level prediction in pharmacogenetics. In *Progress in Artificial Intelligence*, pages 78–89. Springer Berlin Heidelberg.

Castelli, M., Silva, S., and Vanneschi, L. (2015a). A c++ framework for geometric semantic genetic programming. *Genetic Programming and Evolvable Machines*, 16(1):73–81.

Castelli, M., Vanneschi, L., and Felice, M. D. (2015b). Forecasting short-term electricity consumption using a semantics-based genetic programming framework: The south italy case. *Energy Economics*, 47:37 – 41.

Castelli, M., Vanneschi, L., and Popovič, A. (2015c). Parameter evaluation of geometric semantic genetic programming in pharmacokinetics. *International journal of bio-inspired computation*, pages 1 – 10. To appear.

Castelli, M., Vanneschi, L., and Silva, S. (2014). Prediction of the unified parkinson's disease rating scale assessment using a genetic programming system with geometric semantic genetic operators. *Expert Systems with Applications*, 41(10):4608 – 4616.

de Berg, M., Cheong, O., van Kreveld, M., and Overmars, M. (2008). Computational geometry. In *Computational Geometry*, pages 1–17. Springer Berlin Heidelberg.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA.

Krawiec, K. and Lichocki, P. (2009). Approximating geometric crossover in semantic space. In *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 987–994, Montreal. ACM.

Lichman, M. (2013). UCI machine learning repository.

Moraglio, A. (2011). Abstract convex evolutionary search. In *Proceedings of the 11th Workshop Proceedings on Foundations of Genetic Algorithms*, FOGA '11, pages 151–162, New York, NY, USA. ACM.

Moraglio, A., Krawiec, K., and Johnson, C. (2012). Geometric semantic genetic programming. In *Parallel Problem Solving from Nature - PPSN XII*, volume 7491 of *Lecture Notes in Computer Science*, pages 21–31. Springer Berlin Heidelberg.

Moraglio, A. and Mambrini, A. (2013). Runtime analysis of mutation-based geometric semantic genetic programming for basis functions regression. In *Proceedings of the annual international conference on Genetic and Evolutionary Computation*, GECCO '13, pages 989–996, New York, NY, USA. ACM.

Vanneschi, L., Castelli, M., Manzoni, L., and Silva, S. (2013). A new implementation of geometric semantic GP and its application to problems in pharmacokinetics. In *Proceedings of the 16th European Conference on Genetic Programming, EuroGP 2013*, volume 7831 of *LNCS*, pages 205–216, Vienna, Austria. Springer Verlag.

Vanneschi, L., Castelli, M., and Silva, S. (2014a). A survey of semantic methods in genetic programming. *Genetic Programming and Evolvable Machines*, 15(2):195–214.