# An Efficient Implementation of Geometric Semantic Genetic Programming for Anticoagulation Level Prediction in Pharmacogenetics

Mauro Castelli[1,2], Davide Castaldi[3], Ilaria Giordani[5], Sara Silva[2,4], Leonardo Vanneschi[1,2,3], Francesco Archetti[3,5], Daniele Maccagnola[3]

[1] ISEGI, Universidade Nova de Lisboa, 1070-312 Lisboa, Portugal
[2] INESC-ID, IST / Universidade Técnica de Lisboa, 1000-029 Lisboa, Portugal
[3] D.I.S.Co., Università degli Studi di Milano-Bicocca, 20126 Milano, Italy
[4] CISUC, Universidade de Coimbra, 3030-290 Coimbra, Portugal
[5] Consorzio Milano Ricerche, 20126 Milano, Italy

**Abstract.** The purpose of this study is to develop an innovative system for Coumarin-derived drug dosing, suitable for elderly patients. Recent research highlights that the pharmacological response of the patient is often affected by many exogenous factors other than the dosage prescribed and these factors could form a very complex relationship with the drug dosage. For this reason, new powerful computational tools are needed for approaching this problem. The system we propose is called Geometric Semantic Genetic Programming, and it is based on the use of recently defined geometric semantic genetic operators. In this paper, we present a new implementation of this Genetic Programming system, that allow us to use it for real-life applications in an efficient way, something that was impossible using the original definition. Experimental results show the suitability of the proposed system for managing anticoagulation therapy. In particular, results obtained with Geometric Semantic Genetic Programming are significantly better than the ones produced by standard Genetic Programming both on training and on out-of-sample test data.

## 1 Introduction

In the last few years researchers have dedicated several efforts to the definition of Genetic Programming (GP) [8] systems based on the semantics of the solutions, where by semantics we generally intend the behavior of a program once it is executed on a set of inputs, or more particularly the set of its output values on input training data [13]. In particular, very recently new genetic operators, called geometric semantic operators, have been proposed by Moraglio et al. [15]. As Moraglio et al. demonstrate in [15], these operators have the interesting property of inducing a unimodal fitness landscape on any problem consisting in finding the match between a set of input data and a set of known outputs (like for instance regression and classification), which facilitates GP evolvability. In this paper the objective is to evaluate the regression performance of this new GP system on a field of pharmacogenetics of oral anticoagulation therapy, comparing the results with the ones obtained by standard GP. The indication for the use of oral

L. Correia, L.P. Reis, and J. Cascalho (Eds.): EPIA 2013, LNAI 8154, pp. 78–89, 2013.

anticoagulant in many patients is to reduce the embolic risk associated with diseases such as atrial fibrillation, left ventricular dysfunction, deep vein thrombosis and mechanical aortic valve replacement and could be useful for patients who had undergone orthopedic surgery. The trial-error basis of the methods currently in use to fine tune the dosage for a given patient along with the responses' variability due to genetic and behavioral factors can result in out of range periods and, therefore, in a non negligible risk of thromboembolic and bleeding events. Therefore, the problem addressed is the prediction of appropriate oral anticoagulant level of medical drugs. A difficulty with the use of oral anticoagulants is that prescription needs to be individually determined for each patient, usually by following a standard initial dosing protocol, measuring the coagulation rate regularly (using the international normalized ratio, INR, which is a measure of prothrombin time. A high INR value indicates overcoagulation) and then adjusting the dose until the required rate of coagulation is obtained. Relevant help could come from computer support. Mathematical models, able to predict the maintenance dose, were already elaborated more than 20 years ago [16]. These models have widely been applied only recently [7]. The use of computer-based techniques has been shown to have a favorable impact in this field [6] and computational techniques capable of producing reliable predictive models are needed.

Geometric Semantic GP could definitely be a promising approach for this issue, given its ability of inducing unimodal fitness landscapes on problems, independently of how complex they are. Nevertheless, as stated by Moraglio et al. [15], these operators have a serious limitation: by construction, they always produce offspring that are larger than their parents (expressed as the total number of tree nodes) and, as demonstrated in [15], this makes the size of the individuals in the population grow exponentially with generations. Thus, after a few generations the population is composed by individuals so big that the computational cost of evaluating their fitness is unmanageable. This limitation makes these operators impossible to use in practice, in particular on complex real-life applications.

The solution suggested in [15] to overcome this drawback is to integrate in the GP algorithm a "simplification" phase, aimed at transforming each individual in the population into an equivalent (i.e. with the same semantics) but possibly smaller one. Even though this is an interesting and challenging study, depending on the language used to code individuals simplification can be very difficult, and it is often a very time consuming task. For this reason, in this paper we propose a different strategy to solve the problem: we develop a GP system incorporating a new implementation of geometric semantic genetic operators that makes them usable in practice, and does so very efficiently, without requiring any simplification of the individuals during the GP run.

The paper is organized as follows: Section 2 describes the geometric semantic operators introduced by Moraglio et al., while Section 3 presents our new implementation that overcomes the current limitations of these operators, making them usable and efficient. Section 4 presents the medical problem addressed in this paper and highlights its importance for clinicians. Section 5 presents the experimental settings and discusses the obtained results. Finally, Section 6 concludes the paper and provides hints for future research.

## 2 Geometric Semantic Operators

While semantically aware methods [1,9,10] often produced superior performances with respect to traditional methods, many of them are indirect: search operators act on the syntax of the parents to produce offspring, which are successively accepted only if some semantic criterion is satisfied. As reported in [15], this has at least two drawbacks: (i) these implementations are very wasteful as heavily based on trial-and-error; (ii) they do not provide insights on how syntactic and semantic searches relate to each other. To overcome these drawbacks, new operators were introduced in [15] that directly search the semantic space.

To explain the idea behind these operators, let us first give an example using Genetic Algorithms (GAs). Let us consider a GA problem in which the target solution is known and the fitness of each individual corresponds to its distance to the target (our reasoning holds for any distance measure used). This problem is characterized by a very good evolvability and it is in general easy to solve for GAs. In fact, for instance, if we use point mutation, any possible individual different from the global optimum has at least one neighbor (individual resulting from its mutation) that is closer than itself to the target, and thus fitter. So, there are no local optima. In other words, the fitness landscape is unimodal, which usually indicates a good evolvability. Similar considerations hold for many types of crossover, including various kinds of geometric crossover as the ones defined in [10].

Now, let us consider the typical GP problem of finding a function that maps sets of input data into known target ones (regression and classification are particular cases). The fitness of an individual for this problem is typically a distance between its calculated values and the target ones (error measure). Now, let us assume that we are able to find a transformation on the syntax of the individuals, whose effect is just a random perturbation of one of their calculated values. In other words, let us assume that we are able to transform an individual $G$ into an individual $H$ whose output is like the output of $G$, except for one value, that is randomly perturbed. Under this hypothesis, we are able to map the considered GP problem into the GA problem discussed above, in which point mutation is used. So, this transformation, if known, would induce a unimodal fitness landscape on every problem like the considered one (e.g. regressions and classifications), making those problems easily evolvable by GP, at least on training data. The same also holds for transformations on pairs of solutions that correspond to GAs semantic crossovers.

This idea of looking for such operators is very ambitious and extremely challenging: finding those operators would allow us to directly search the space of semantics, at the same time working on unimodal fitness landscapes. Although not without limitations, contribution [15] accomplishes this task, defining new operators that have exactly these characteristics.

Here we report the definition of geometric semantic operators given in [15] for real functions domains, since these are the operators we will use in the experimental phase. For applications that consider other kinds of data, the reader is referred to [15].

**Definition 1. (Geometric Semantic Crossover).** *Given two parent functions* $T_1, T_2 : \mathbb{R}^n \rightarrow \mathbb{R}$*, the geometric semantic crossover returns the real function* $T_{XO} = (T_1 \cdot T_R) + ((1 - T_R) \cdot T_2)$*, where* $T_R$ *is a random real function whose output values range in the interval* $[0, 1]$.

The interested reader is referred to [15] for a formal proof of the fact that this operator corresponds to a geometric crossover on the semantic space. Nevertheless, even without formally proving it, we can have an intuition of it by considering that the (unique) offspring generated by this crossover has a semantic vector that is a linear combination of the semantics of the parents with random coefficients included in $[0, 1]$ and whose sum is equal to 1. To constrain $T_R$ in producing values in $[0, 1]$ we use the sigmoid function: $T_R = \frac{1}{1+e^{-T_{rand}}}$ where $T_{rand}$ is a random tree with no constraints on the output values.

**Definition 2. (Geometric Semantic Mutation).** *Given a parent function* $T : \mathbb{R}^n \rightarrow \mathbb{R}$*, the geometric semantic mutation with mutation step* $ms$ *returns the real function* $T_M = T + ms \cdot (T_{R1} - T_{R2})$*, where* $T_{R1}$ *and* $T_{R2}$ *are random real functions.*

Reference [15] formally proves that this operator corresponds to a box mutation on the semantic space, and induces a unimodal fitness landscape. However, even though without a formal proof, it is not difficult to have an intuition of it, considering that each element of the semantic vector of the offspring is a "weak" perturbation of the corresponding element in the parent's semantics. We informally define this perturbation as "weak" because it is given by a random expression centered in zero (the difference between two random trees). Nevertheless, by changing parameter $ms$, we are able to tune the "step" of the mutation, and thus the importance of this perturbation.

We also point out that at every step of one of these operators, offspring contain the complete structure of the parents, plus one or more random trees as its subtrees and some arithmetic operators: the size of each offspring is thus clearly much larger than the one of their parents. The exponential growth of the individuals in the population (demonstrated in [15]) makes these operators unusable in practice: after a few generations the population becomes unmanageable because the fitness evaluation process becomes unbearably slow. The solution that is suggested in [15] consists in performing an automatic simplification step after every generation in which the programs are replaced by (hopefully smaller) semantically equivalent ones. However, this additional step adds to the computational cost of GP and is only a partial solution to the progressive program size growth. Last but not least, according to the particular language used to code individuals and the used primitives, automatic simplification can be a very hard task.

In the next section, we present a new implementation of these operators that overcomes this limitation, making them efficient without performing any simplification step and without imposing any particular representation for the individuals (for example the traditional tree-based representation of GP individuals can be used, as well as a linear representation, or any other one).

For simplicity, from now on, our implementation of GP using the geometric semantic crossover and mutation presented in [15] will be indicated as GS-GP (Geometric-Semantic GP).

## 3 Implementation of Geometric Semantic GP

The implementation we propose can be described as follows. Note that, although we describe the algorithm assuming the representation of the individuals is tree based, the implementation fits any other type of representation.

In a first step, we create an initial population of (typically random) individuals, exactly as in standard GP. We store these individuals in a table (that we call $P$ from now on) as shown in Figure 1(a), and we evaluate them. To store the evaluations we create a table (that we call $V$ from now on) containing, for each individual in $P$, the values resulting from its evaluation on each fitness case (in other words, it contains the semantics of that individual). Hence, with a population of $n$ individuals and a training set of $k$ fitness cases, table $V$ will be made of $n$ rows and $k$ columns.

Then, for every generation, a new empty table $V'$ is created. Whenever a new individual $T$ must be generated by crossover between selected parents $T_1$ and $T_2$, $T$ is represented by a triplet $T = \langle \&(T_1), \&(T_2), \&(R) \rangle$, where $R$ is a random tree and for any tree $\tau$, $\&(\tau)$ is a *reference* (or memory pointer) to $\tau$ (using a C-like notation). This triplet is stored in an appropriate structure (that we call $\mathcal{M}$ from now on) that also contains the name of the operator used, as shown in Figure 1c. The random tree $R$ is created, stored in $P$, and evaluated in each fitness case to reveal its semantics. The values of the semantics of $T$ are also easily obtained, by calculating $(T_1 \cdot R) + ((1-R) \cdot T_2)$ for each fitness case, according to the definition of geometric semantic crossover, and stored in $V'$. Analogously, whenever a new individual $T$ must be obtained by applying mutation to an individual $T_1$, $T$ is represented by a triplet $T = \langle \&(T_1), \&(R_1), \&(R_2) \rangle$ (stored in $\mathcal{M}$), where $R_1$ and $R_2$ are two random trees (newly created, stored in $P$ and evaluated for their semantics). The semantics of $T$ is calculated as $T_1 + ms \cdot (R_1 - R_2)$ for each fitness case, according to the definition of geometric semantic mutation, and stored in $V'$. In the end of each generation, table $V'$ is copied into $V$ and erased. At this point, all the rows of $P$ and $\mathcal{M}$ referring to individuals that are not ancestors[1] of the new population can also be erased, because they will not be used anymore.

In synthesis, this algorithm is based on the idea that, when semantic operators are used, an individual can be fully described by its semantics (which makes the syntactic component much less important than in standard GP), a concept discussed in depth in [15]. Therefore, at every generation we update table $V$ with the semantics of the new individuals, and save the information needed to build their syntactic structures without explicitly building them. In terms of computational time, we emphasize that the process of updating table $V$ is very efficient as it does not require the evaluation of the entire trees. Indeed, evaluating each individual requires (except for the initial generation) a constant time, which is independent from the size of the individual itself. In terms of memory, tables $P$ and $\mathcal{M}$ grow during the run. However, table $P$ adds a maximum of $2 \times n$ rows per generation (if all new individuals are created by mutation) and table $\mathcal{M}$ (which contains only memory pointers) adds a maximum of $n$ rows per generation. Even if we never erase the "ex-ancestors" from these tables (and never reuse random trees, which is also possible), we can manage them efficiently for several thousands of generations.

[1] We abuse the term "ancestors" to designate not only the parents but also the random trees used to build an individual by crossover or mutation.

The final step of the algorithm is performed after the end of the last generation: in order to reconstruct the individuals, we need to "unwind" our compact representation and make the syntax of the individuals explicit. Therefore, despite performing the evolutionary search very efficiently, in the end we cannot avoid dealing with the large trees that characterize the standard implementation of geometric semantic operators. However, most probably we will only be interested in the best individual found, so this unwinding (and recommended simplification) process may be required only once, and it is done offline after the run is finished. This greatly contrasts with the solution proposed by Moraglio et al. of building and simplifying every tree in the population at each generation online with the search process.

Let us briefly consider the computational cost of evolving a population of $n$ individuals for $g$ generations. At every generation, we need $O(n)$ space to store the new individuals. Thus, we need $O(ng)$ space in total. Since we need to do only $O(1)$ operations for any new individual (since the fitness can be computed directly from the semantics, which can immediately be obtained from the semantics of the parents), the time complexity is also $O(ng)$. Thus, we have a linear space and time complexity with respect to population size and number of generations.

Excluding the time needed to build and simplify the best individual, the proposed implementation allowed us to evolve populations for thousands of generations with a considerable speed up with respect to standard GP. Next we provide a simple example that illustrates the functioning of the proposed algorithm.

**Example.** Let us consider the simple initial population $P$ shown in table (a) of Figure 1 and the simple pool of random trees that are added to $P$ as needed, shown in table (b). For simplicity, we will generate all the individuals in the new population (that we call $P'$ from now on) using only crossover, which will require only this small amount of random trees. Besides the representation of the individuals in infix notation, these tables contain an identifier (Id) for each individual ($T_1$, ..., $T_5$ and $R_1$, ..., $R_5$). These identifiers will be used to represent the different individuals, and the individuals created for the new population will be represented by the identifiers $T_6$, ..., $T_{10}$.

We now describe the generation of a new population $P'$. Let us assume that the (non-deterministic) selection process imposes that $T_6$ is generated by crossover between $T_1$ and $T_4$. Analogously, we assume that $T_7$ is generated by crossover between $T_4$ and $T_5$, $T_8$ is generated by crossover between $T_3$ and $T_5$, $T_9$ is generated by crossover between $T_1$ and $T_5$, and $T_{10}$ is generated by crossover between $T_3$ and $T_4$. Furthermore, we assume that to perform these five crossovers the random trees $R_1$, $R_2$, $R_3$, $R_4$ and $R_5$ are used, respectively. The individuals in $P'$ are simply represented by the set of entries exhibited in table (c) of Figure 1 (structure $\mathcal{M}$). This table contains, for each new individual, a *reference* to the ancestors that have been used to generate it and the name of the operator used to generate it (either "crossover" or "mutation").

Let us assume that now we want to reconstruct the genotype of one of the individuals in $P'$, for example $T_{10}$. The tables in Figure 1 contain all the information needed to do that. In particular, from table (c) we learn that $T_{10}$ is obtained by crossover between $T_3$ and $T_4$, using random tree $R_5$. Thus, from the definition of geometric semantic crossover, we know that it will have the following structure: $(T_3 \cdot R_5) + ((1 - R_5) \cdot T_4)$. The remaining tables (a) and (b), that contain the syntactic structure of $T_3$, $T_4$, and

| Id | Individual |
|---|---|
| $T_1$ | $x_1 + x_2 x_3$ |
| $T_2$ | $x_3 - x_2 x_4$ |
| $T_3$ | $x_3 + x_4 - 2x_1$ |
| $T_4$ | $x_1 x_3$ |
| $T_5$ | $x_1 - x_3$ |

(a)

| Id | Individual |
|---|---|
| $R_1$ | $x_1 + x_2 - 2x_4$ |
| $R_2$ | $x_2 - x_1$ |
| $R_3$ | $x_1 + x_4 - 3x_3$ |
| $R_4$ | $x_2 - x_3 - x_4$ |
| $R_5$ | $2x_1$ |

(b)

| Id | Operator | Entry |
|---|---|---|
| $T_6$ | crossover | $\langle \&(T_1), \&(T_4), \&(R_1) \rangle$ |
| $T_7$ | crossover | $\langle \&(T_4), \&(T_5), \&(R_2) \rangle$ |
| $T_8$ | crossover | $\langle \&(T_3), \&(T_5), \&(R_3) \rangle$ |
| $T_9$ | crossover | $\langle \&(T_1), \&(T_5), \&(R_4) \rangle$ |
| $T_{10}$ | crossover | $\langle \&(T_3), \&(T_4), \&(R_5) \rangle$ |

(c)

**Fig. 1.** Illustration of the example described in Section 3. (a) The initial population $P$; (b) The random trees used by crossover; (c) The representation in memory of the new population $P'$.

$R_5$, provide us with the rest of the information we need to completely reconstruct the syntactic structure of $T_{10}$, which is $((x_3 + x_4 - 2x_1) \cdot (2x_1)) + ((1 - (2x_1)) \cdot (x_1 x_3))$ and upon simplification can become $-x_1(4x_1 - 3x_3 - 2x_4 + 2x_1 x_3)$.

## 4 Oral Anticoagulant Therapy

Coumarins-derived Oral Anticoagulant therapy (OAT), prescribed to more than 2.5 million new patients per year, is commonly used as life-long therapy in the prevention of systemic embolism in patients with atrial fibrillation, valvular heart disease, and in the primary and secondary prevention of venous and pulmonary thromboembolism. It is also used for the prevention of thromboembolic events in patients with acute myocardial infarction and with angina pectoris, in patients with heart valves, and after some types of orthopedic surgery.

Due to the increased prevalence of atrial fibrillation and thromboembolic disorders in elderly people [3] oral anticoagulation is one of the most frequently prescribed therapy in elderly patients.

Aging is a complex process which is accompanied by a potential multitude of issues that include numerous health problems associated to a multiple administration of medications, often coupled with reduced mobility and greater frailty, with a tendency to fall. Despite its validated efficiency, all these conditions are often cited as reasons to preclude the elderly from being anticoagulated [4].

In all subjects a combination of personal, genetic and non-genetic factors are responsible for about 20-fold variation in the coumarins dose required to achieve the therapeutic range of drug action, evaluated by the prothrombin international normalized ratio (INR) measurement. In case of elderly patients, this variability is highlight by clinically significant interaction due to coadministration of different drugs [14], and by liver and renal impairment which can further emphasize this interaction or directly modify the anticoagulant action [2]. For this reasons, oral anticoagulant therapy Initiation in elderly is more challenging than other patients.

The safety and efficacy of warfarin therapy are dependent on maintaining the INR within the target range for the indication. Due to above-cited inter patient variability in drug dose requirements, empiric dosing results in frequent dose changes as the therapeutic international normalized ratio (INR) gets too high or low, leaving patients at risk for bleeding (over-coagulation) and thromboembolism (under-coagulation). This means

that there is a need to carry on the research to develop predictive models that are able to account for strong intraindividual variability in elderly patients' response to coumarins treatment.

Most of the computational approaches in the literature for the definition of mathematical models to support management decisions for OAT, provide the use of regression methods. The widely applied technique is Multiple Linear Regression, especially used to predict the value of the maintenance dose [7]. Other linear and non linear approaches enclose Generalized Linear Models [5] and polynomial regression [11]. More complex machine learning techniques were also employed to support clinical decisions on therapy management. A review of these methods is proposed in [12] for a review).

## 5 Experimental Study

In this section the experimental phase is outlined. In particular, section 5.1 briefly described the data used in the experimental phase; section 5.2 presents the experimental settings for the considered systems, while section 5.3 contains a detailed analysis of the obtained results.

### 5.1 Data Description

We collect data from clinical computerized databases based on $950$ genotyped over $65$ years old patients in anticoagulant therapy. A data preprocessing approach returned 748 $cleaned$ patients (i.e. with complete data, not missing any information) useful for analysis. The features of this dataset can be summarized in four main entities: personal characteristics, anamnestic features, genetic data and therapy's characteristics . Demographic information includes body mass index and smoke habit; the anamnestic data are related to medical evidence leading to OAT (Atrial Fibrillation, Deep Venous Thrombosis/Pulmunary Embolism, other), a comorbidity (yes or not) and polipharmacotherapy evaluations (digitalis, amiodarone, furosemide, nitrates, beta blockers, calcium channel blockers, ACE inhibitors, diuretic tiazidic, sartanic, statins and other) and a renal function parameter (creatinine clearance); genetic data include the information related to the genetic polymorphysms involved in the metabolism of anticoagulant drug (CYP2C9, VKORC1 and CYP4F2); therapy's features describe patient's INR range, the INR range assigned within which patient should remain during therapy (2-3, 2.5-3.5 , 2.5-3), target INR (represented by the average of the values of INR range, respectively 2.5, 3 and 2.75), vitamin k antagonist anticoagulant drug (warfarin 5mg and acenocumarol 4 or 1mg) and their dosage, which is the independent variable of the study. All data used in the study were checked by clinicians of the anticoagulation clinical center. Dataset includes two subsets of patients: $403$ stable patients which reached a stable response to therapy (stay in assigned INR range without significant modification of drug dose) and $345$ unstable patients which did not reach stability. Descriptive statistic table relative to all features of the dataset is available as supplementary material on the authors' website (<anonymized>).

### 5.2 Experimental Settings

We have tested our implementation of GP with geometric semantic operators (GS-GP) against a standard GP system (STD-GP). A total of 30 runs were performed with each technique using different randomly generated partitions of the dataset into training (70%) and test (30%) sets. All the runs used populations of 200 individuals allowed to evolve for 500 generations. Tree initialization was performed with the Ramped Half-and-Half method [8] with a maximum initial depth equal to 6. The function set contained the four binary arithmetic operators $+$, $-$, $*$, and $/$ protected as in [8]. Fitness was calculated as the root mean squared error (RMSE) between predicted and expected outputs. The terminal set contained the number of variables corresponding to the number of features in each dataset. Tournaments of size 4 were used to select the parents of the new generation. To create new individuals, STD-GP used standard (subtree swapping) crossover [8] and (subtree) mutation [8] with probabilities 0.9 and 0.1, respectively. In our system this means that crossover is applied 90% of the times (while 10% of the times a parent is copied into the new generation) and 10% of the offspring are mutated. For GS-GP, crossover rate was 0.7, while mutation rate was 0.3, since preliminary tests have shown that the geometric semantic operators require a relatively high mutation rate in order to be able to effectively explore the search space. Survival was elitist as it always copied the best individual into the next generation. No maximum tree depth limit has been imposed during the evolution.

### 5.3 Experimental Results

The experimental results are reported using curves of the fitness (RMSE) on the training and test sets and boxplots obtained in the following way. For each generation the training fitness of the best individual, as well as its fitness in the test set (that we call test fitness) were recorded. The curves in the plots report the median of these values for the 30 runs. The median was preferred over the mean because of its higher robustness to outliers. The boxplots refer to the fitness values in generation 500. In the following text we may use the terms fitness, error and RMSE interchangeably.

Figure 2(a) and Figure 2(b) report training and test error for STD-GP and GS-GP while generations elapse. These figures clearly show that GS-GP outperforms STD-GP on both training and test sets. Figure 2(c) and Figure 2(d) report a statistical study of the training and test fitness of the best individual, both for GS-GP and STD-GP, for each of the 30 performed runs. Denoting by IQR the interquartile range, the ends of the whiskers represent the lowest datum still within 1.5 IQR of the lower quartile, and the highest datum still within 1.5 IQR of the upper quartile. As it is possible to see, GS-GP produces solutions with a lower dispersion with respect to the ones produced by STD-GP. To analyze the statistical significance of these results, a set of tests has been performed on the median errors. As a first step, the Kolmogorov-Smirnov test has shown that the data are not normally distributed and hence a rank-based statistic has been used. Successively, the Wilcoxon rank-sum test for pairwise data comparison has been used under the alternative hypothesis that the samples do not have equal medians. The p-values obtained are $3.4783 \times 10^{-4}$ when test fitness of STD-GP is compared to test fitness of GS-GP and $4.6890 \times 10^{-7}$ when training fitness of STD-GP is compared to training fitness of GS-GP. Therefore, when employing the usual significance level $\alpha = 0.05$ (or even a smaller

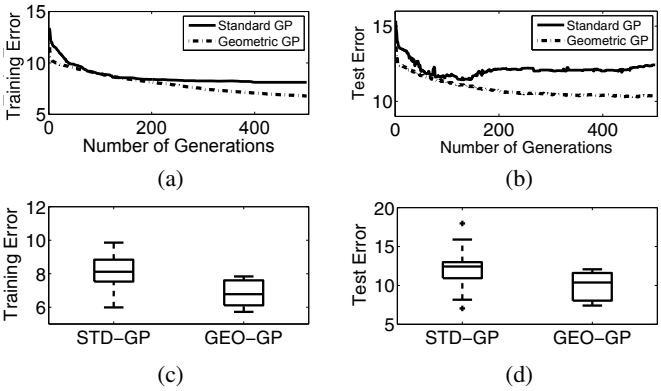


**Fig. 2.** Median of train (a) and test (b) fitness for the considered techniques at each generation calculated over 30 independent runs. Train (c) and test (c) fitness of the best individual produced in each of the 30 runs at the last performed generation.

one), we can clearly state that GS-GP produces fitness values that are significantly lower (i.e., better) than the STD-GP both on training and test data.

### 5.4 Generalization Ability

Given the very promising results obtained by GS-GP, we have performed a further experimental analysis to investigate the generalization ability of the models produced by the new technique.

A first indication about the behavior of GS-GP and STD-GP on unseen data comes from the Figure 2(b) of the previous section. From this figure, it seems that, differently from $ST-GP$, GS-GP is able to produce a model that does not overfit the unseen data.

To confirm this hypothesis, in this section we report the results obtained running GS-GP for $10000$ generations. Given the fact that geometric semantic genetic operators induce a unimodal fitness landscape, we expected that the fitness on the training set will improve for GS-GP, but the main concern regards its generalization ability when the number of generations increases. In particular, in this section we want to answer the following question: do the good performances of GS-GP on training set result in an overfitted model for unseen data?

Figure 3(a) and Figure 3(b) allow to answer to this question. In particular, the good results that GS-GP has obtained on training data were expected: the geometric semantic

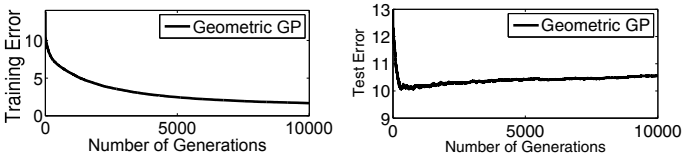


**Fig. 3.** Median of train (left) and test (right) fitness for GS-GP at each generation calculated over 30 independent runs

operators induce an unimodal fitness landscape, which facilitates evolvability. On the other hand, these excellent results on the training set do not affect the generalization ability of the model on unseen data. As it is possible to note from Figure 3(b), GS-GP produces a model that is able to handle unseen instances with a test fitness comparable to the one obtained in Figure 2(b). Furthermore, from Figure 3(b) we see that after generation 500 the error on the training set is slightly increasing, but not in a comparable way to the irregular behavior of the curve of STD-GP reported in Figure 2(b). This is a very promising result, in particular with respect to the considered applicative domain. Moreover, this results seems to indicate an important difference between $STD-GP$ and GS-GP: while $STD-GP$ overfits the data after a few generations, GS-GP seems to be much more robust to unseen data, at least for the studied application.

## 6 Conclusions

New genetic operators, called geometric semantic operators, have been proposed for genetic programming. They have the extremely interesting property of inducing a unimodal fitness landscape for any problem consisting in matching input data into known target outputs. This should make all the problems of this kind easily evolvable by genetic programming.

Nevertheless, in their first definition these new operators have a strong limitation that makes them unusable in practice: they produce offspring that are larger than their parents, and this results in an exponential growth of the size of the individuals in the population. This paper proposed an implementation of GP that uses the geometric semantic operators efficiently. This new GP system evolves the semantics of the individuals without explicitly building their syntax. It does so by keeping a set of trees (of the initial population and the random ones used by geometric semantic crossover and mutation) in memory and a set of pointers to them, representing the "instructions" on how to build the new individuals. Thanks to this compact representation, it is possible to make use of the great potential of geometric semantic GP to solve complex real-life problems.

The proposed GP system was used to address an important application in the field of pharmacogenetics. In particular, the problem addressed is the prediction of appropriate oral anticoagulant level of medical drugs. A difficulty with oral anticoagulants use is that prescription needs to be individually determined for each patient, usually by following a standard initial dosing protocol, measuring the coagulation rate regularly and then adjusting the dose until the required rate of coagulation is obtained.

The experimental results demonstrate that the new system outperforms standard genetic programming. Besides the fact that the new genetic programming system has excellent results on training data, we were positively surprised by its excellent generalization ability on the studied applications, testified by the good results obtained on test data.

Considering the good result achieved in this study, we will pursue it: beside new experimental validations on new data and different applications, we plan to orient our future activity towards more theoretical studies of the generalization ability of geometric semantic genetic programming. Moreover, regarding the oral anticoagulant therapy problem, we plan to start a work of interpretation of the models generated by GP, in strict collaboration with a team of expert clinicians.

**Acknowledgments.** This work was supported by national funds through FCT under contract PEst-OE/EEI/LA0021/2013 and projects EnviGP (PTDC/EIA-CCO/103363/2008) and MassGP (PTDC/EEI-CTP/2975/2012). Portugal.

## References

1. Beadle, L., Johnson, C.: Semantically driven crossover in genetic programming. In: Proc. of the IEEE World Congress on Comput. Intelligence, pp. 111–116. IEEE Press (2008)
2. Capodanno, D., Angiolillo, D.J.: Antithrombotic therapy in the elderly. Journal of the American College of Cardiology 56(21), 1683–1692 (2010)
3. Anderson Jr., F.A., Wheeler, H.B., Goldberg, R.J., et al.: A population-based perspective of the hospital incidence and case-fatality rates of deep vein thrombosis and pulmonary embolism: The worcester dvt study. Archives of Internal Medicine 151(5), 933–938 (1991)
4. Fang, M.C., Chen, J., Rich, M.W.: Atrial fibrillation in the elderly. Am. J. Med. 120(6), 481–487 (2007)
5. Fang, M.C., Machtinger, E.L., Wang, F., Schillinger, D.: Health literacy and anticoagulation-related outcomes among patients taking warfarin. J. Gen. Intern. Med. 21(8), 841–846 (2006)
6. Jowett, S., Bryan, S., Poller, L., Van Den Besselaar, A.M.H.P., Van Der Meer, F.J.M., Palareti, G., Shiach, C., Tripodi, A., Keown, M., Ibrahim, S., Lowe, G., Moia, M., Turpie, A.G., Jespersen, J.: The cost-effectiveness of computer-assisted anticoagulant dosage: results from the European action on anticoagulation (eaa) multicentre study. J. Thromb. Haemost. 7(9), 1482–1490 (2009)
7. Klein, T.E., Altman, R.B., Eriksson, N., Gage, B.F., Kimmel, S.E., Lee, M.-T.M., Limdi, N.A., Page, D., Roden, D.M., Wagner, M.J., Caldwell, M.D., Johnson, J.A.: Estimation of the dose with clinical and pharmacogenetic data. New England Journal of Medicine 360(8), 753–764 (2009)
8. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
9. Krawiec, K.: Medial crossovers for genetic programming. In: Moraglio, A., Silva, S., Krawiec, K., Machado, P., Cotta, C. (eds.) EuroGP 2012. LNCS, vol. 7244, pp. 61–72. Springer, Heidelberg (2012)
10. Krawiec, K., Lichocki, P.: Approximating geometric crossover in semantic space. In: GECCO 2009, July 8-12, pp. 987–994. ACM (2009)
11. Leichsenring, I., Plesch, W., Unkrig, V., Kitchen, S., Kitchen, D.P., Maclean, R., Dikkeschei, B., van den Besselaar, A.M.H.P.: Multicentre isi assignment and calibration of the inr measuring range of a new point-of-care system designed for home monitoring of oral anticoagulation therapy. Thromb. Haemost. 97(5), 856–861 (2007)
12. Martin, B., Filipovic, M., Rennie, L., Shaw, D.: Using machine learning to prescribe warfarin. In: Dicheva, D., Dochev, D. (eds.) AIMSA 2010. LNCS, vol. 6304, pp. 151–160. Springer, Heidelberg (2010)
13. McPhee, N.F., Ohs, B., Hutchison, T.: Semantic building blocks in genetic programming. In: O'Neill, M., Vanneschi, L., Gustafson, S., Esparcia Alcázar, A.I., De Falco, I., Della Cioppa, A., Tarantino, E. (eds.) EuroGP 2008. LNCS, vol. 4971, pp. 134–145. Springer, Heidelberg (2008)
14. Miners, J.O., Birkett, D.J.: Cytochrome p4502c9: an enzyme of major importance in human drug metabolism. British Journal of Clinical Pharmacology 45(6), 525–538 (1998)
15. Moraglio, A., Krawiec, K., Johnson, C.G.: Geometric semantic genetic programming. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012, Part I. LNCS, vol. 7491, pp. 21–31. Springer, Heidelberg (2012)
16. Ryan, P.J., Gilbert, M., Rose, P.E.: Computer control of anticoagulant dose for therapeutic management. BMJ 299(6709), 1207–1209 (1989)