Introducing Semantic-Clustering Selection in Grammatical Evolution

Stefan Forstenlechner
Natural Computing Research & Applications Group
Complex & Adaptive Systems Laboratory
School of Business
University College Dublin
stefan.forstenlechner@ucdconnect.ie

Miguel Nicolau
Natural Computing Research & Applications Group
Complex & Adaptive Systems Laboratory
School of Business
University College Dublin
miguel.nicolau@ucd.ie

David Fagan
Natural Computing Research & Applications Group
Complex & Adaptive Systems Laboratory
School of Business
University College Dublin
david.fagan@ucd.ie

Michael O'Neill
Natural Computing Research & Applications Group
Complex & Adaptive Systems Laboratory
School of Business
University College Dublin
m.oneill@ucd.ie

ABSTRACT

Semantics has gained much attention in the last few years and new advanced crossover and mutation operations have been created which use semantic information to improve the quality and generalisability of individuals in genetic programming. In this paper we present a new selection operator in grammatical evolution which uses semantic information of individuals instead of just the fitness value. The semantic traits of an individual are stored in a vector. An unsupervised learning technique is used to cluster individuals based on their semantic vector. Individuals are only allowed to reproduce with individuals from the same cluster to preserve semantic locality and intensify the search in a certain semantic area. At the same time, multiple semantic areas are covered by the search as there exist multiple clusters which cover different areas and therefore preserve semantic diversity. This new selection operator is tested on several symbolic regression benchmark problems and compared to grammatical evolution with tournament selection to analyse its performance.

Keywords

Grammatical Evolution; Semantic; Selection

GECCO '15, July 11 - 15, 2015, Madrid, Spain

ISBN 978-1-4503-3488-4/15/07...$15.00

DOI: http://dx.doi.org/10.1145/2739482.2768502

1. INTRODUCTION

Evolutionary algorithms are inspired by nature. Especially the phrase "survival of the fittest" is used in many papers and reflects how most selection methods work. Fitter individuals are selected more often. This is mainly done, by using a single fitness value, which defines how fit an individual is.

Another important aspect for some natural computing algorithms, like *Particale Swarm Optimization* (PSO) [14] or *Ant Colony Optimization* [7], is social learning. In PSO, a real-valued vector is used as representation for a possible solution, where each value in the vector corresponds to a parameter of the problem. The algorithm is inspired by the flocking behaviour of birds. The movement of an individual influences the behaviour of the other individuals. Ant-colony inspired algorithms construct solutions by placing pheromones in the environment which attract individuals to possible better regions. These interactions in swarm and ant-colony algorithms are used to exchange information to influence the actions taken in the next iterations to move closer to an optimal solution. In nature phenotypic traits influence the social interactions between individuals and therefore the transmission of genetic material.

In this paper, we introduce a social selection mechanism named *Semantic-Clustering Selection* (SCS), which uses semantic information in form of a (semantic) vector as a surrogate for phenotypic traits. In contrary to most existing selection methods, SCS does not solely focus on the fitness value but also considers the semantic similarity between individuals. Individuals which have more phenotypic traits in common are considered to be more attracted to each other and therefore more likely to mate. As traits are inherited by their children, the children may be similar to their parents, thus in the same area of the search space, but hopefully better. This is inspired also by earlier work by Uy et al. [23] which highlighted the importance of semantic locality

and diversity. Locality is preserved by restricting selection to individuals in the same cluster and semantic diversity is preserved by evolving multiple clusters in the population. The benefits of this operator are that it can be integrated within any evolutionary algorithm which uses selection. Furthermore, in contrast to geometric semantic operators which leads to bigger individuals than standard operators, SCS does not influence the size of individuals. As SCS uses a semantic vector for clustering, it does not have to be changed for a specific problem, while geometric semantic operators have to be defined differently for every type of problem.

In this paper, we provide some background in section 2 and a description of SCS in section 3. SCS is tested on several symbolic regression benchmark problems. The experimental settings are given in section 4 and the results in section 5. At the end of the paper, we provide a conclusion of the new operator in section 6 and a prospect of future work in section 7.

2. BACKGROUND

2.1 Grammatical Evolution

Grammatical Evolution (GE) like other evolutionary algorithms uses selection, crossover and mutation to evolve solutions for arbitrary problems [20, 6]. GE in general only evolves an integer string [12]. A mapping process uses these integers with a grammar to generate a corresponding solution. This mapping process makes GE flexible, because GE can generate solutions/programs for a problem, as long as a grammar can be defined for it. Whereas, for other evolutionary algorithms you may have to implement a new solution representation instead of simply exchanging a grammar. Another aspect of GE is that its mapping process is inspired by molecular biology. The integer string can be seen as the genotype of an individual, while the mapped program can be seen as the phenotype that is created by mapping the genotype according to rules defined in the grammar. During the evolution new integer strings (genotypes) are generated by crossover and may be mutated, which has an effect on the phenotype produced in the mapping.

GE has already been used to evolve programs for many different kinds of problems, such as creating truss design [8], optimising pylon structures [4], evolving aircraft models [3], controlling femtocell network coverage [11] and has shown competitive results to other evolutionary algorithms like genetic programming [20]. In this paper, GE is used to compare a new selection method *Semantic-Clustering Selection* (SCS) against tournament selection, which is currently the most common used selection method in the literature for GE and genetic programming (GP).

2.2 Semantics

Semantics has gained much interest over the last years in the field of GP [22], [18]. Many operators and methods based on semantics have been introduced. Semantics has been used to provide additional information of an individual to guide and improve the search process of the evolutionary algorithm. In contrary to the syntax which is the structure of an individual, semantic defines "the behavior of a program, once it is executed on a set of data" [25]. Lets assume that two individuals are given for a regression problem with two input variables x and y:

$$f_1(x, y) = x * y * 2 + x \tag{1}$$

$$f_2(x, y) = (x + x) * y + x \tag{2}$$

These formulae are syntactically different, but semantically identical as they produce the same output if the same input is given. This differentiation between syntax and semantics is important. Individuals with the same semantics have the same fitness, but can have different syntax. Therefore their genotype is different and so different genetic code can be contributed to the next generation by semantically identical individuals. This is not the case for syntactically identical individuals.

Previously, semantics have been used with GE through the use of attributed grammar to solve 01 multi-constrained knapsack problems [5]. Also several operators, mainly crossover and mutation operators, based on semantics have already been introduced, which produce better results than more common genetic operators. Nguyen et al. presented a semantic aware crossover operation [19] to exchange semantically similar subtrees of individuals. This operator has been slightly improved several times (e.g. [22], [24]). An example of a semantic mutation operator is Beadle and Johnson's semantic driven mutation [2], which replaces a subtree with a semantically different one. Another approach to semantics is to use the convexity of the semantic search space. This has been done with geometric semantic genetic programming (GSGP) [18]. GSGP uses the whole structure of the parents plus some random trees and combines them to let the children outperform their parents. The disadvantage of this approach is that the size of individuals increases drastically per generation. The huge size of the individuals makes this approach unusable in practice and the operators are restricted to specific domains (boolean, regression and simple program induction [i.e. if-then-else statements]). Therefore, it cannot be applied to the more general form of GP for program induction.

2.3 Selection methods

Tournament selection is the typical selection method in the literature for GP and GE. It randomly selects a certain number of individuals which participate in a tournament. The fittest individual of a tournament is selected for reproduction. An advantage of tournament selection is that the selection pressure is directly controlled by the tournament size. A high tournament size leads to high selection pressure, whereas a low tournament size indicates low selection pressure. As many other selection mechanisms, like linear rank or proportional selection, tournament selection uses only the fitness value to determine which individual gets selected.

Other selection methods like gender specific selection [1] or sexual selection [10] differentiate between male and female individuals. Different selection mechanisms are used to select a female individual and an appropriate male for reproduction.

An example of an improved tournament selection is, correlative tournament selection [17]. It selects the first individual with a normal tournament selection. But for the second individual, n individuals are randomly selected and the syntactic similarity is calculated between the first and the n individuals. The most similar one is then used for reproduction with the first one.

clusters ← cluster(*previous population*)
while *next population* is not full **do**
  first parent ← tournament selection(*previous population*)
  add *first parent* to *next population*
  cluster ← get cluster of parent(*clusters*, *first parent*)
  second parent ← tournament selection(*cluster*)
  add *second parent* to *next population*
end while

Figure 1: Semantic-Clustering Selection pseudocode

SCS, which is presented in section 3, is similar to correlative tournament selection, but there are two main differences. First, SCS calculates similarity based on the semantics of an individual, rather than on the genotype. Secondly, SCS does not randomly select n individuals, but uses clustering to create a pool of possible mates.

2.3.1 Semantics in selection

Galvan-Lopez et al. have presented a selection method based on semantics which is called *semantics in selection* (SiS) [9]. It improves tournament selection by selecting two *semantically different* parents for crossover. So the first parent is selected with a normal tournament which only uses the fitness of the individuals. The second tournament checks that the individual is semantically different and then it selects the individual with the best fitness of the semantically different individuals of the current tournament. Galvan-Lopez et al. argue that GP with SiS promotes genetic diversity and present impressive advantages over standard GP. The experimental settings used in that paper do not use mutation. GP with SiS may not need mutation, due to the fact that SiS promotes diversity, but standard GP may prematurely converge if mutation is not used. Additionally, the tournament size was set to 7 throughout all experiments, independent of the population size. Three different population sizes have been used for the experiments, 126, 250 and 500. SiS promotes semantic diversity, therefore GP with SiS may not be affected as much as GP without SiS by the high selection pressure due to the big tournament size and that no mutation is used. Thus, the experimental settings might have been favourable for GP with SiS. Additionally, Galvan-Lopez et al. did not consider semantic locality, which has been shown to provide a performance advantage [23].

3. SEMANTIC-CLUSTERING SELECTION

In this section we present a new selection method, named *Semantic-Clustering Selection* (SCS) which uses semantic information to explore different areas of the search space while intensifying the search in these areas. It aims to balance semantic locality and diversity. The idea is to create a vector which contains semantic information for every individual. Afterwards, individuals are clustered based on this vector by a clustering algorithm. These clusters define the different areas of the search space which are currently explored. Then tournament selection is used to select individuals for reproduction. The first parent is selected from the whole population, while the second parent has to be an individual which is semantically similar to the first one. Therefore, it is selected from the same cluster as the first parent. The pseudocode of SCS is depicted in algorithm 1.

The goal is to only mate semantically similar parents with another, to improve the fitness of a specific area of the search space. At the same time, because several clusters are created within a population, the population does not converge towards a single best individual or area, but rather towards several areas. This should improve the robustness of the search process as it should be more likely to find a good solution when covering several promising areas.

The semantic output of an individual depends on the problem that is tackled. In this paper, we analyse the performance of SCS on symbolic regression problems. Therefore, the semantic output can be defined as the output of the functions produced by the individuals and we use the semantic distance measure, proposed by Uy et al [22], for calculating the distance between two individuals. But instead of randomly sampled values for the distance measure, the fitness cases of the problems were used. The distance measure is used for clustering.

In contrary to geometric semantic operators which create huge models which are not practical, SCS only uses semantic information for selection. It can be easily be integrated in any evolutionary algorithm which uses a selection operator.

4. EXPERIMENTAL SETTING

For the experiments with SCS, we used k-means [16] for clustering. K-means initializes k points, called centroids, within given observations, which are going to be clustered. It then iteratively executes two steps. The assignment step assigns the observations to the closest centroid. Every observation which is assigned to the same centroid belongs to the same cluster. The update step sets every centroid to the center of its cluster by calculating the mean over all the observations, which are assigned to the centroid. The steps are repeated until k-means converges, which means that the centroids have not been moved any more, or until a specified number of iterations is reached.

As you need to specify the number of clusters for k-means as well as initialize the centroids, several different settings are being tested. For the number of clusters, the values 2, 5, 10 and 20 are being analysed. The reason for also using a large amount of clusters like 20 is that initial experiments showed that outliers influence the clustering and some clusters may then only contain 1 individual or a very small number of individuals. By using a large number of clusters we expect to at least get some appropriate sized clusters. As for initializing the centroids, we use three different approaches. The first approach is to randomly initialize the centroids with the values of an individual. The second is to use the semantic vector of the k best individuals. For the last approach, the semantic distance to the target is calculated and the individuals are sorted accordingly. The initial centroids are the k individuals furthest apart in this sorting. For example, if $k = 2$, then the semantic vectors of the individual closest and furthest from the target are used as the initial centroids.

We tested SCS on the following well known regression problems, Keijzer-6 [13], Nguyen-7 [22], Pagie-1 [21] and Vladislavleva-4 [26], proposed as good benchmarks by White et al. [27]. Also only the training set was used for evaluation. The grammar contained all operators suggested for the specific problem, but only 1.0 as constant. Single point crossover and int flip mutation have been used as operators. A summary of the parameter settings can be found in

Table 1: Experimental parameter settings

Runs	30
Generations	200 (400 for Pagie-1)
Population size	500
Tournament size	5 (adapted)
Crossover probability	0.9
Mutation probability	0.05
Elite size	1
Maximum clustering iterations	50
Number of clusters	2, 5, 10, 20

table 1. The number of generations was set to 200 for all problems except for Pagie-1, because neither standard GE nor GE with SCS have converged after 200 generations for this problem. Therefore, the number of generations was increased to 400 for this problem. The general tournament size was set to 5, which is 1% of the whole population. When an individual from a cluster was selected, the tournament size was adapted to the size of the cluster. E.g. for a cluster of size 270, the tournament size would be 2.7, which is rounded to 3. A minimum tournament size was set to 2, in case of small clusters, to avoid random search and maintain a certain amount of selection pressure.

The results of SCS with the different initialization strategies and cluster sizes are compared to standard GE with tournament selection, as well as to SiS based on [9]. For all the experiments the same settings were used.

5. RESULTS

In this section we present the results of our experiments in table 2 and table 3. These tables show the average best fitness as well as the standard deviation of the average best fitness of 30 runs. Fitness is measured as the mean squared error to the target function. Therefore, a value closer to 0 is better. Table 2 shows the results of standard GE with tournament selection and the results of GE with SiS. Table 3 summarizes the results of the experiments with SCS with the three different initialization strategies and the different number of clusters.

5.1 Semantics in selection

At first, we take a look at the results of SiS shown in table 2. The problem Nguyen-7 has also been used in the paper that presented SiS, but was named $F6$ [9]. In general the results of SiS are very similar compared to standard GE. SiS is in most cases slightly worse than standard GE. In case of Keijzer-6 there is even a significant difference. This contradicts the findings in [9]. The reason might be that the parameter settings differ from the original settings for SiS. Thus, we also run experiments with SiS and standard GE with the settings from [9]. On the one hand, SiS is always better than standard GE when using the original settings. On the other hand, as expected and described in section 2, the results for both SiS and standard GE are worse than the results presented here, due to the high selection pressure and not using mutation. In the end, GE and GP are different systems. Maybe SiS would have performed better than tournament selection in GP.

5.2 Semantic-Clustering Selection

Three different initialization strategies and four different number of clusters are used for SCS. In total, 12 different settings are compared to standard GE. We have used all these different settings, not to give SCS an unfair advantage, but to see the effects of different settings possible in SCS.

As can be seen in table 3, not a single setting is better on all problems than standard GE. Actually, standard GE performs better on Keijzer-6 than SiS and any SCS setting. If we take a look at the other problems, many settings perform slightly better than standard GE, but in general all results are pretty similar. No statistical significant results have been achieved with SCS. Further, no general best setting for SCS has been found on this problems. Different settings perform better on different problems. Even if we only compare the initialization strategies or number of clusters, no clear best setting can be found. Nevertheless, SCS was able to achieve better results on 3 of the 4 problems, which are most of the times even more robust.

To analyse the behaviour of SCS, we measured several attributes of the clustering and the selection. One measure was the average size of the biggest cluster over the generations. The results mainly suggested that one cluster is dominating the others, especially with a small number of clusters. Additionally, the more training points a problem has the bigger the dominating cluster gets. Clustering in such an high dimensional space is rather difficult, which may lead to these results. A higher number of clusters might be beneficial to get a meaningful clustering. We do not provide plots of the cluster sizes because they only show that a higher number of clusters reduces the size of the biggest cluster.

Another measure we used, which shows more interesting data, is the distance between two parents that have been selected for reproduction. The semantic vectors of the parents that are selected to create children together are used with the Canberra distance [15] to calculate how similar the parents are. The Canberra distance is shown in equation 3.

$$d(p,q) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i| + |q_i|} \tag{3}$$

We adapted the distance measure to be normalized by the length of the semantic vector to receive a number in the interval $[0, 1]$, see equation 4. p and q are the semantic vectors of the two parents. *Note that when two vectors are similar, the result will be closer to* 0.

$$d(p,q) = \frac{1}{n} \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i| + |q_i|} \tag{4}$$

The measurements of pairwise similarity of parents for the 'random' initialization strategy is shown in figure 2. As expected, in most cases it shows that the more clusters we use the more similar the parents will be that are selected to reproduce together. Another observation that can be made is that for the easier problems Keijzer-6 and Nguyen-7, the similarity stays rather constant, except from an initial spike. In case of Pagie-1 and Vladislavleva-4, where the fitness is not as good as for the other two problems, the similarity seems to decrease slowly after the 20th generation. Maybe the number of semantic diverse individuals which are able to achieve similar fitness increases over time. Further investigation is necessary to fully understand this phenom-

Table 2: Results of standard GE and GE with SiS. Columns denoted with a *-sign have used the settings from [9].

	SiS	SiS* Pop 126 Gen 200	SiS* Pop 250 Gen 100	SiS* Pop 500 Gen 50	standard GE
Keijzer-6	0.00434 ± 0.00303	0.02551 ± 0.02756	0.01510 ± 0.02119	0.01234 ± 0.00785	$\mathbf{0.00248 \pm 0.00164}$
Nguyen-7	0.00039 ± 0.00037	0.00587 ± 0.00797	0.00387 ± 0.00626	0.00243 ± 0.00279	0.00036 ± 0.00044
Pagie-1	0.07493 ± 0.03859	0.27599 ± 0.10906	0.27499 ± 0.12715	0.30191 ± 0.11503	0.06538 ± 0.04562
Vladislavleva-4	0.03759 ± 0.00356	0.04835 ± 0.03594	0.04188 ± 0.00509	0.04028 ± 0.00413	0.03789 ± 0.00354

Table 3: Results of Semantic-Clustering Selection (SCS). "random", "best", "distance" means the initialization strategy of k-means, as explained in the first paragraph of section 4. The number indicates the number of clusters used. The results of standard GE are always in rightmost column for an easier comparison.

	SCS random 2	SCS best 2	SCS distance 2	standard GE
Keijzer-6	0.00311 ± 0.00208	0.00364 ± 0.00245	0.00331 ± 0.00287	$\mathbf{0.00248 \pm 0.00164}$
Nguyen-7	0.00035 ± 0.00039	0.00031 ± 0.00036	0.00040 ± 0.00041	0.00036 ± 0.00044
Pagie-1	0.05928 ± 0.03895	0.07893 ± 0.05429	0.05991 ± 0.03210	0.06538 ± 0.04562
Vladislavleva-4	0.03761 ± 0.00329	0.03707 ± 0.00384	0.03802 ± 0.00281	0.03789 ± 0.00354

	SCS random 5	SCS best 5	SCS distance 5	standard GE
Keijzer-6	0.00379 ± 0.00262	0.00417 ± 0.00265	0.00327 ± 0.00259	$\mathbf{0.00248 \pm 0.00164}$
Nguyen-7	0.00046 ± 0.00062	0.00033 ± 0.00039	$0.00032 \pm \mathbf{0.00031}$	0.00036 ± 0.00044
Pagie-1	$\mathbf{0.05538} \pm 0.02933$	0.06609 ± 0.03373	$0.05851 \pm \mathbf{0.02758}$	0.06538 ± 0.04562
Vladislavleva-4	0.03800 ± 0.00298	0.03750 ± 0.00329	0.03830 ± 0.00399	0.03789 ± 0.00354

	SCS random 10	SCS best 10	SCS distance 10	standard GE
Keijzer-6	0.00347 ± 0.00270	0.00376 ± 0.00242	0.00340 ± 0.00257	$\mathbf{0.00248 \pm 0.00164}$
Nguyen-7	0.00051 ± 0.00046	0.00035 ± 0.00038	0.00038 ± 0.00037	0.00036 ± 0.00044
Pagie-1	0.05795 ± 0.03528	0.06634 ± 0.03873	0.07575 ± 0.04538	0.06538 ± 0.04562
Vladislavleva-4	0.03811 ± 0.00412	$\mathbf{0.03675} \pm 0.00333$	0.03720 ± 0.00360	0.03789 ± 0.00354

	SCS random 20	SCS best 20	SCS distance 20	standard GE
Keijzer-6	0.00358 ± 0.00297	0.00309 ± 0.00181	0.00284 ± 0.00207	$\mathbf{0.00248 \pm 0.00164}$
Nguyen-7	$\mathbf{0.00030} \pm 0.00032$	0.00036 ± 0.00038	0.00053 ± 0.00047	0.00036 ± 0.00044
Pagie-1	0.06669 ± 0.03332	0.05996 ± 0.03441	0.07231 ± 0.04177	0.06538 ± 0.04562
Vladislavleva-4	0.03770 ± 0.00352	$0.03718 \pm \mathbf{0.00263}$	0.03792 ± 0.00445	0.03789 ± 0.00354

ena. Lastly, the initial generation is generated randomly by ramped half-and-half, which explains why selection in the first generation selects rather dissimilar parents and why the similarity increases afterwards. But it is interesting that in all 4 problems in the first 10 generations right after the increase, there is a spike where similarity rapidly decreases and right afterwards increases again.

Although in most problems the pairwise similarity of parents selected for reproduction increases if more clusters are used, this is not always the case. Especially for Vladislavleva-4, a difference can be seen, also when changing the initialization strategy for the clusters, as depicted in figure 3 where the y axis has been set to show values between 0 and 0.1. While SCS with 5 clusters selects the most similar parents with 'best', it selects the most dissimilar parents when 'random' is used, compared to the other number of clusters.

6. CONCLUSION

In this paper we presented a new selection method *Semantic-Clustering Selection* based on semantics, which uses unsupervised learning techniques to cluster individuals based on their semantic similarity. Individuals are then limited to only reproduce with another inside these clusters to intensify the search process inside each cluster while several areas of the search space are covered. The selection method SCS was tested on several regression problems and compared to standard GE as well as to SiS presented in [9], which is another selection method which uses semantics. The presented results show that SCS is a promising method which is able to perform better than standard GE and that it makes the search process more robust.

7. FUTURE WORK

In this paper, SCS was only applied to symbolic regression. The high dimensional space makes it difficult to create meaningful clusters. In the future, we will try to use only a subset of the trainings data points to decrease the dimensionality for the clustering. We are also going to analyse the behaviour of SCS on other types of problems like boolean or classification problems. Additionally, SCS was only used

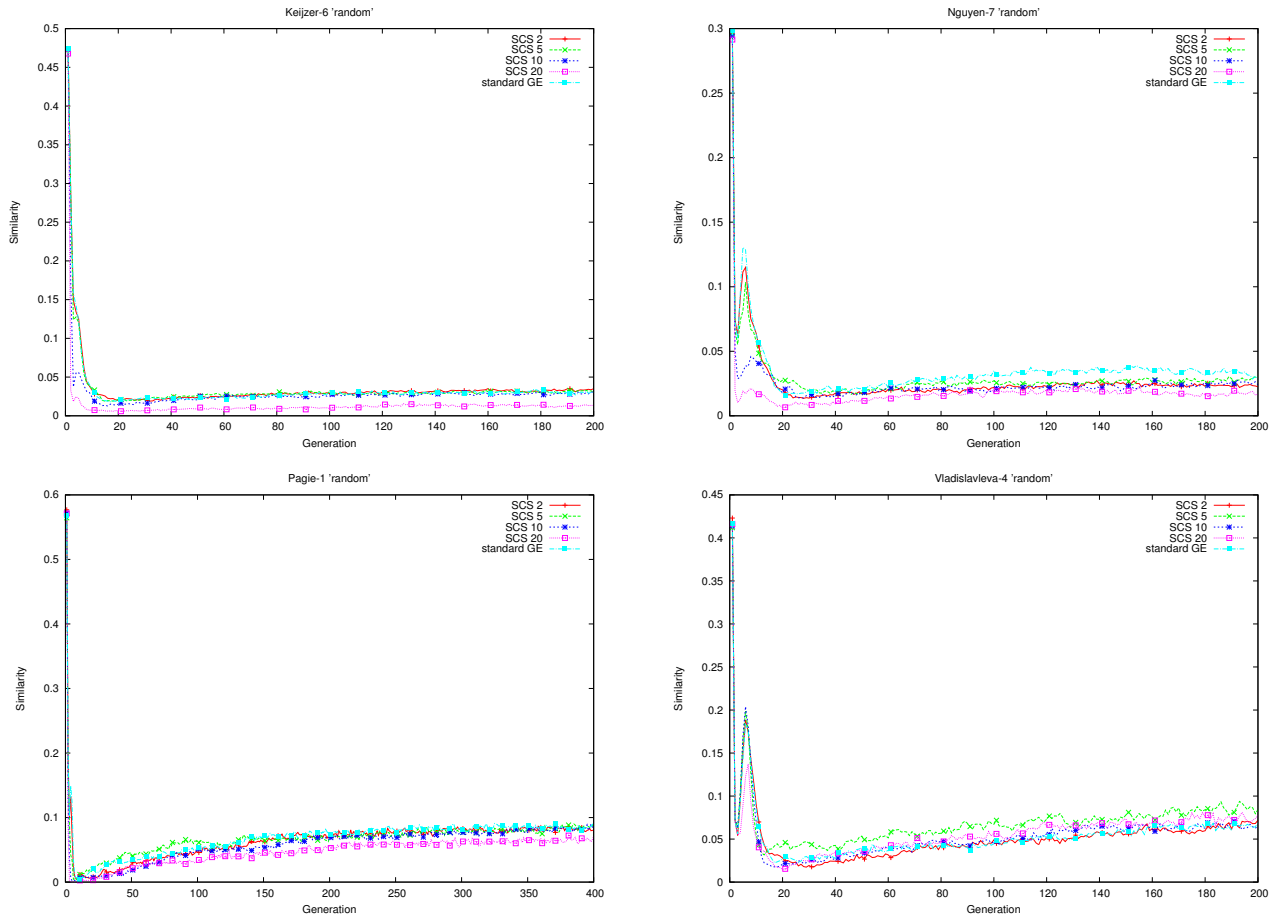

Figure 2: Average Canberra distance of parents over the generations. ***Note the different ranges on the y axis.***

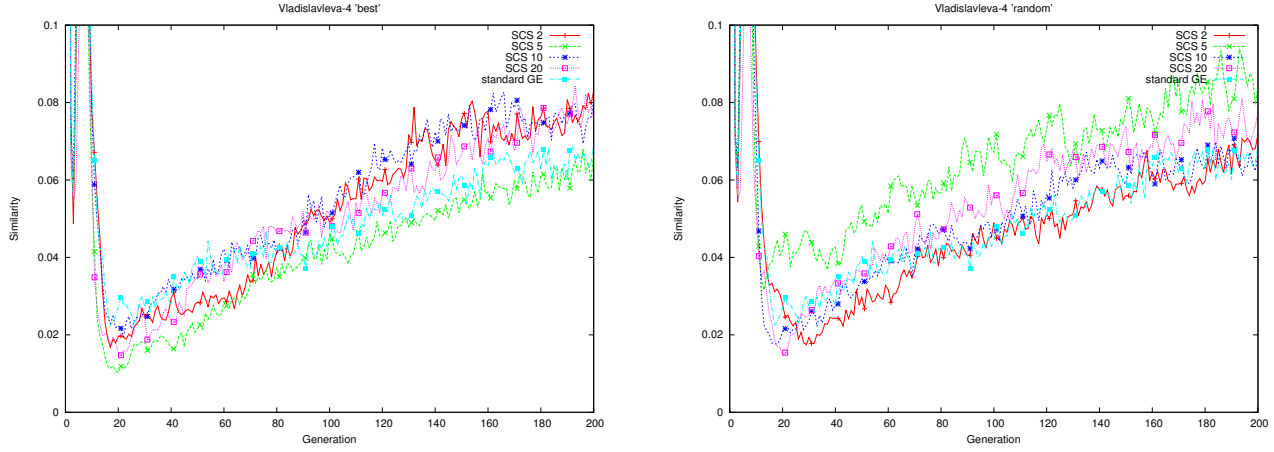

Figure 3: Average Canberra distance of parents over the generations for Valdislavleva-4 with 'best' and 'random' initialization strategy.

with k-means. As already stated, any clustering algorithm can be used. K-means is a rather simple algorithm and the number of clusters has to be predefined. In the future, we will test SCS with more advanced clustering algorithms and even with algorithms which are able to determine an appropriate number of clusters by themselves. The semantic distance measure we used might not be optimal for clustering in such a high dimensional space. We may also have to consider other measures. Furthermore, as a next step to improve SCS, analysing clusters and how they influence the search processes might give better insight in the selection method. How do clusters change over time? How often does migration of individuals to other clusters happen? SCS may not influence the size of the clusters, but it might influence how the centroids of the clusters move through the search space. All the answers of these questions might help us to improve the selection process and the already promising results of SCS.

8. ACKNOWLEDGMENTS

This research is based upon works supported by the Science Foundation Ireland, under Grant No. 13/IA/1850.

9. REFERENCES

[1] M. Affenzeller, S. Winkler, S. Wagner, and A. Beham. *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. Chapman & Hall/CRC, 1st edition, 2009.

[2] L. Beadle and C. Johnson. Semantically driven mutation in genetic programming. In *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pages 1336–1342, May 2009.

[3] J. Byrne, P. Cardiff, A. Brabazon, and M. O'Neill. Evolving parametric aircraft models for design exploration and optimisation. *Neurocomputing*, 142(0):39 – 47, 2014. {SI} Computational Intelligence Techniques for New Product Development.

[4] J. Byrne, M. Fenton, E. Hemberg, J. McDermott, and M. O'Neill. Optimising complex pylon structures with grammatical evolution. *Information Sciences*, (0):–, 2014.

[5] R. Cleary and M. O'Neill. An attribute grammar decoder for the 01 multiconstrained knapsack problem. In G. Raidl and J. Gottlieb, editors, *Evolutionary Computation in Combinatorial Optimization*, volume 3448 of *Lecture Notes in Computer Science*, pages 34–45. Springer Berlin Heidelberg, 2005.

[6] I. Dempsey, M. O'Neill, and A. Brabazon. *Foundations in Grammatical Evolution for Dynamic Environments*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[7] M. Dorigo. Ant colony optimization: A new meta-heuristic. In *Proceedings of the Congress on Evolutionary Computation*, pages 1470–1477. IEEE Press, 1999.

[8] M. Fenton, C. McNally, J. Byrne, E. Hemberg, J. McDermott, and M. O'Neill. Automatic innovative truss design using grammatical evolution. *Automation in Construction*, 39(0):59 – 69, 2014.

[9] E. Galván-López, B. Cody-Kenny, L. Trujillo, and A. Kattan. Using semantics in the selection mechanism in genetic programming: A simple method for promoting semantic diversity. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 2972–2979, June 2013.

[10] K. S. Goh, A. Lim, and B. Rodrigues. Sexual selection for genetic algorithms. *Artif. Intell. Rev.*, 19(2):123–152, Apr. 2003.

[11] E. Hemberg, L. Ho, M. O'Neill, and H. Claussen. A comparison of grammatical genetic programming grammars for controlling femtocell network coverage. *Genetic Programming and Evolvable Machines*, 14(1):65–93, 2013.

[12] J. Hugosson, E. Hemberg, A. Brabazon, and M. O'Neill. Genotype representations in grammatical evolution. *Appl. Soft Comput.*, 10(1):36–43, 2010.

[13] M. Keijzer. Improving symbolic regression with interval arithmetic and linear scaling. In C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli, and E. Costa, editors, *Genetic Programming*, volume 2610 of *Lecture Notes in Computer Science*, pages 70–82. Springer Berlin Heidelberg, 2003.

[14] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, Nov 1995.

[15] G. N. Lance and W. T. Williams. Mixed-data classificatory programs i - agglomerative systems. *Australian Computer Journal*, 1(1):15–20, 1967.

[16] J. B. Macqueen. Some methods for classification and analysis of multivariate observations. In *Procedings of the Fifth Berkeley Symposium on Math, Statistics, and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[17] K. Matsui. New selection method to improve the population diversity in genetic algorithms. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, volume 1, pages 625–630 vol.1, 1999.

[18] A. Moraglio, K. Krawiec, and C. Johnson. Geometric semantic genetic programming. In C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, editors, *Parallel Problem Solving from Nature - PPSN XII*, volume 7491 of *Lecture Notes in Computer Science*, pages 21–31. Springer Berlin Heidelberg, 2012.

[19] Q. Nguyen, X. Nguyen, and M. O'Neill. Semantic aware crossover for genetic programming: The case for real-valued function regression. In L. Vanneschi, S. Gustafson, A. Moraglio, I. De Falco, and M. Ebner, editors, *Genetic Programming*, volume 5481 of *Lecture Notes in Computer Science*, pages 292–302. Springer Berlin Heidelberg, 2009.

[20] M. O'Neill and C. Ryan. *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*. Kluwer Academic Publishers, Norwell, MA, USA, 2003.

[21] L. Pagie and P. Hogeweg. Evolutionary consequences of coevolving targets. *Evol. Comput.*, 5(4):401–418, Dec. 1997.

[22] N. Q. Uy, N. X. Hoai, M. O'Neill, R. McKay, and E. Galván-López. Semantically-based crossover in genetic programming: application to real-valued

symbolic regression. *Genetic Programming and Evolvable Machines*, 12(2):91–119, 2011.

[23] N. Q. Uy, N. X. Hoai, M. O'Neill, R. McKay, and D. N. Phong. On the roles of semantic locality of crossover in genetic programming. *Information Sciences*, 235(0):195 – 213, 2013. Data-based Control, Decision, Scheduling and Fault Diagnostics.

[24] N. Q. Uy, E. Murphy, M. O'Neill, and N. X. Hoai. Semantic-based subtree crossover applied to dynamic problems. In *KSE*, pages 78–84. IEEE, 2011.

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

[26] E. Vladislavleva, G. Smits, and D. den Hertog. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. *Evolutionary Computation, IEEE Transactions on*, 13(2):333–349, April 2009.

[27] D. R. White, J. Mcdermott, M. Castelli, L. Manzoni, B. W. Goldman, G. Kronberger, W. Jaśkowski, U.-M. O'Reilly, and S. Luke. Better gp benchmarks: Community survey results and proposals. *Genetic Programming and Evolvable Machines*, 14(1):3–29, Mar. 2013.