

The DU Map: A Visualization to Gain Insights into Genotype-Phenotype Mapping and Diversity

Eric Medvet
DIA - Università di Trieste
Trieste, Italy
emedvet@units.it

Tea Tušar
Department of Intelligent Systems - Jožef Stefan Institute
Ljubljana, Slovenia
tea.tusar@ijs.si

ABSTRACT

The relation between diversity and genotype to phenotype mapping has been the focus of several studies. In those Evolutionary Algorithms (EAs) where the genotype is a sequence of symbols, the contribution of each of those symbols in determining the phenotype may vary greatly, possibly being null. In the latter case, the unused portions of the genotype may host a large amount of the population diversity. However, reasoning on coarse-grained measures makes it hard to validate such a claim and, more in general, to gain insights into the interactions between genotype-phenotype mapping and diversity. In this paper, we propose a novel visualization which summarizes in a single, compact heat map (the DU map), three kinds of information: (a) how diverse are the genotypes in the population at the level of single symbols; (b) if and to what degree each individual symbol in the genotype contributes to the phenotype; (c) how the two previous measures vary during the evolution. We experimentally verify the usefulness of the DU map w.r.t. its primary goal and, more broadly, when used to analyze different EA design options. We apply it to Grammatical Evolution (GE) as it constitutes an ideal testbed for the DU map, due to the availability of different mapping functions.

CCS CONCEPTS

•Computing methodologies → Genetic programming; •Human-centered computing → Heat maps;

KEYWORDS

Genotype-Phenotype Mapping, Diversity, Redundancy, Grammatical Evolution, Visualization, Heat maps

ACM Reference format:

Eric Medvet and Tea Tušar. 2017. The DU Map: A Visualization to Gain Insights into Genotype-Phenotype Mapping and Diversity. In *Proceedings of The Genetic and Evolutionary Computation Conference, Berlin, Germany, July 15–19, 2017 (GECCO’17)*, 8 pages.

DOI: <http://dx.doi.org/10.1145/3067695.3082554>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO’17, Berlin, Germany

© 2017 ACM. 978-1-4503-4939-0/17/07...\$15.00

DOI: <http://dx.doi.org/10.1145/3067695.3082554>

1 INTRODUCTION

Many evolutionary algorithms (EAs) are based on a twofold representation of individuals, which are described by means of a *genotype* and a *phenotype*. In those algorithms, a genotype-phenotype mapping function maps any genotype to a phenotype, possibly in a many-to-one fashion. Grammatical Evolution (GE) [28] is one such EA in which the mapping function exploits a user-provided context-free grammar that is tailored to the specific problem being tackled. In practice, in order to use GE, the user is only required to provide: (a) the grammar corresponding to a language suitable to describe candidate solutions and (b) a fitness function able to assess them. Both things only require the knowledge of the specific domain, whereas the internals of the EA can be ignored by the user: this feature enabled a wide adoption of GE. Practitioners and researchers extensively relied on GE to tackle a diverse set of problems such as, e.g., generation of road traffic rules [23], identification of taxonomies in Wikipedia [3], and development of artificial neural networks [1].

Despite having favored its widespread adoption, the genotype-phenotype mapping of GE has also been largely discussed among scholars who debated and still debate about the analogies with nature [26, 27, 32, 38], the deviation from its initial motivations [39], and its properties [20, 24, 33, 34]. Among the several arguments raised in those discussions, one is particularly significant: the relation between the search effectiveness and the tendency of the mapping to map different genotypes to the same phenotype. The latter property is called *redundancy* and has been recently the subject of some experimental studies [20, 33]: both works better characterized the redundancy in GE, respectively in terms of its (non) uniformity and its relation with *diversity*; still, the cited works do not deliver a clear view on if and how redundancy affects the search effectiveness.

On the other hand, redundancy is closely related to *neutrality*, i.e., the ability of a genetic operator to introduce different genetic materials without changing the fitness of an individual [41]. In a twofold representation EA, such as GE, neutrality can be achieved also by means of the mapping function which, by favoring many-to-one mapping, allows the progressive modification of a genotype without negatively affect the corresponding (good) phenotype. Indeed, high neutrality has been identified as one of the motivations for GE [8, 9] and stated to be beneficial to—again—*diversity* and eventually to GE effectiveness.

In this paper, we propose a novel visualization aimed at easing the investigation of the relation between genotype-phenotype mapping and diversity. We tailored our proposal to GE, because of its widespread usage and the availability of different variants for the mapping functions [16, 21, 25], which facilitates the experimental

validation of the usefulness of this visualization. However, this visualization is potentially applicable to any EA with a twofold representation and the genotype in the form of a sequence of symbols. Our proposed visualization summarizes in a single, compact heat map three kinds of information: (a) how diverse are the genotypes in the population at the level of single genes (*diversity*); (b) if and to what degree each individual gene in the genotype contributes to the phenotype (*usage*); (c) how the two previous measures vary during the evolution. The rationale for showing together a and b (which form the name of our visualization: Diversity and Usage map, *DU map*) is to allow the expert to, on the one hand, easily spot which portions of the genotype do not or scarcely contribute in determining the phenotype—hence being those places where redundancy/neutrality nestle—and, on the other hand, to verify whether those portions tend to host the largest part of the population diversity.

In addition to pursuing the aforementioned primary goal, the DU map may be useful to EA researchers for: (1) understanding and possibly validating how different mapping functions work; (2) providing the ground for the design of new genetic operators which possibly exploit the knowledge about actual contribution of genes to the mapping; (3) comprehending how the (lack of) diversity is distributed along the genotype.

The remainder of this paper is organized as follows. Section 2 briefly surveys the state-of-the-art in the visualization of evolution internals. Section 3 introduces our visualization by presenting its components and how they are assembled. Section 4 shows how we applied the DU map to analyze different aspects of GE. Finally, Section 5 draws the conclusions.

2 RELATED WORK

Visualization is a powerful tool for supporting reasoning and is often used to gain insight into the workings of EAs. The visual representation of single solutions and populations of solutions heavily depends on the solution encoding (in the genotype and phenotype space) and comprises, among others, simple binary “zebra” representations [40], heat maps (also called matrix charts or density plots) for real-valued genotypes [12, 35], graphs and trees for discrete optimization problems [6], and domain-specific representations for some real-world problems [11, 29, 37]. In case of many objectives, the focus of the analysis, and consequently visualization, shifts to the objective space and the challenge of visualizing Pareto front approximations [36].

Static representations of solutions and populations can be enriched by adding information on the changes brought by crossover and mutation operators. Some visualization tools specifically support analysis of the relations between parent and offspring solutions, and exploration of the ancestry of the chosen (usually best) solution [5, 6, 19]. Other research focuses on visualizing the progress of the evolution [13, 18] and the balance between exploration and exploitation during the search [2, 15]. The latter is closely related to the diversity of solutions.

Defining diversity can be intricate in the existence of a genotype-phenotype mapping or in the case of multiple objectives. Nevertheless, if specified as a measure on the population, it can be visualized with the same techniques as the fitness of the best (or median)

individual, for example, with a simple line graph showing how the diversity of the population changes during the evolution [4]. Another option is to define diversity based on the occurrence of symbols in the genes, in which case a heat map can be used to illustrate how diversity evolves for each gene separately [35]. The latter approach is used in our diversity maps.

The proposed DU map employs color to convey two kinds of information (diversity and usage) on a single heat map. To the best of our knowledge, the only other approach that uses a similar idea of combining several values in one color is the visualization with pseudo-color [10]. There, the heat map of a population contains binary genotypes of its individuals—one per row. The color of each cell/gene is determined depending on the gene value and the objective and fitness values¹ of the individuals. Genes with a value of 0 are colored in blue, while genes with a value of 1 are red. Then, these two “basic” colors are modified for the whole row in hue and brightness depending on the individual’s objective and fitness values, respectively.

Our approach differs from [10] in two aspects: (1) the DU maps visualize the whole evolution, not just a single population; and (2) color is assigned for each gene separately and is based on its diversity and usage rather than on its value and the fitness and objective of the individual.

3 VISUALIZATION WITH DU MAPS

As stated in the Introduction, our visualization may be applied to any EA where the genotype is a sequence of symbols. We here formally define the entities on which our proposal is built and then show how those definitions are cast to GE.

3.1 Basic definitions

We consider a generic EA in which a population of solutions (individuals) evolves for a number n_{gen} of generations: we denote with S_i the population at the i th generation—we do not require that the population size remains constant during the evolution. Each individual s is associated with exactly one genotype $g = (g_1, \dots, g_l) \in \mathcal{A}^l$, which is a sequence of l symbols of an alphabet \mathcal{A} . In other words, all individuals have genotypes with the same length, which remains constant during the evolution. This technique could be potentially applied also to genotypes with variable length, but we argue that it would not be particularly valuable. Each individual s is also associated with a phenotype p , which is obtained by means of a mapping $m : \mathcal{A}^l \rightarrow \mathcal{P}$ between the genotype space \mathcal{A}^l and the phenotype space \mathcal{P} .

We assume that a *usage function* $u : \mathcal{A}^l \rightarrow [0, 1]^l$ exists, which measures the degree $u_k(g) \in [0, 1]$ to which each k th symbol g_k of the genotype g concurred in determining the phenotype $m(g)$. The usage function is inherently related to the mapping m ; for a given m , $u(g)$ depends only on g . We also assume that a *diversity function* $d : \mathbb{N}^{\mathcal{A}} \rightarrow [0, 1]$ exists, $\mathbb{N}^{\mathcal{A}}$ being the set of all the multisets built from \mathcal{A} , which measures the symbol diversity of a multiset $A \in \mathbb{N}^{\mathcal{A}}$ of symbols of \mathcal{A} as a number in $[0, 1]$, where 0 means

¹A distinction is made between the objective and fitness values to accommodate for problems where the fitness value contains some other information in addition to the objective value, such as a penalty determined through expert knowledge of the problem.

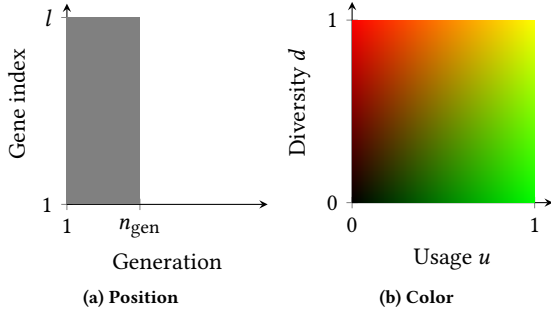


Figure 1: The DU map legend: on the left, the position of a map cell r.r.t. gene index and the generation number; on the right, the color w.r.t. the values of the diversity function d and the usage function u .

minimal diversity (e.g., $d(A) = 0$ if A contains only instances of the same symbol) and 1 means maximal diversity (e.g., $d(A) = 1$ if $A = \mathcal{A}$).

We propose a new visualization, called the DU map, that visualizes information about an evolution $S_1, \dots, S_{n_{\text{gen}}}$ as a rectangular color map of size $n_{\text{gen}} \times l$ —i.e., where the x -axis represents the generation and the y -axis represents the position within the genotype (see Figure 1a). In particular, the color of each (x, y) point in the map is computed as follows (see Figure 1b). The intensity value $i_{\text{red}}(x, y)$ of the red channel is determined by the diversity:

$$i_{\text{red}}(x, y) = d(\{g_y, s \in S_x\}) \quad (1)$$

where g_y is the y th symbol of the genotype of the individual s of the population S_x of the x th generation. The intensity value $i_{\text{green}}(x, y)$ of the green channel is determined by the usage:

$$i_{\text{green}}(x, y) = \frac{1}{|S_x|} \sum_{s \in S_x} u_y(g) \quad (2)$$

where g is the genotype of the individual s of the population S_x of the x th generation. The intensity value of the blue channel $i_{\text{blue}}(x, y)$ is set to 0 for all points (x, y) :

$$i_{\text{blue}}(x, y) = 0 \quad (3)$$

As can be seen from Figure 1b, low usage and low diversity is shown in black, while high usage and high diversity is depicted in yellow. On the other hand, low diversity and high usage results in green hues, while high diversity and low usage in red ones.

The information visualized by the DU map can be decomposed in two gray-scale heat maps by plotting separately the red and green channels: we called those maps Diversity and Usage, respectively.

3.2 Application to GE

GE [28] has been introduced by Ryan et al. two decades ago as a variant of Genetic Programming [14] able to generate programs in any language described by a context-free grammar (CFG). GE evolves a population of binary strings which are mapped to strings of the language defined by the CFG by means of a mapping function. The latter consumes the genes in the genotype in groups of 8 (each

called codon) in order to choose one of the options in the production rule for the leftmost non-terminal still present in the phenotype.

After this seminal work, several other variants of GE has been introduced, most of them consisting only or mainly of a different mapping function. In this work, we considered the standard original GE [28], Position-independent GE (π GE) [25], a modification of the Structural GE (SGE) [16], and Weighted Hierarchical GE (WHGE) [21], all of them operating on binary strings as genotypes. We refer the reader to the cited papers for more details about these variants.

As opposed to the other three variants, SGE works on integer strings as genotypes: the length of the genotype and the domain of each integer gene is determined based on the grammar [16]; moreover, SGE adopts an ad hoc crossover operator which takes into account the structure of the genotype. In order to allow for a more meaningful comparison among the variants, we modified SGE to use binary strings. Specifically, we designed a grammar-dependant procedure for transforming any binary string in a integer string in which the length of the string and the domains of the integers are consistent with the grammar. We also modified the peculiar SGE crossover operator to handle appropriately binary string genotypes while roughly preserving the semantic of the original SGE crossover. The resulting variant, which we call BitSGE, vaguely resembles SGERed, a redundant variant of SGE which the authors of the latter introduced in [17] in order to analyze more in detail the redundancy and locality of their proposal.

As we consider binary strings, the alphabet is given by $\mathcal{A} = \{0, 1\}$. The diversity function d is defined with:

$$d(A) = 1 - 2 \left| \frac{1}{2} - \frac{|\{a \in A : a = 0\}|}{|A|} \right| \quad (4)$$

which results in $d(A) = 0$ if and only if all bits in A are 0 (or 1) and in $d(A) = 1$ if and only if exactly half of the bits are 0.

Finally, the usage function u is defined with:

$$u((g_1, \dots, g_l)) = \frac{1}{\max_{i \in \{1, \dots, l\}} c_i} (c_1, \dots, c_l) \quad (5)$$

where c_i is the number of times the i th bit has been used during the mapping. Figure 2 shows an example of the computation of the color for two specific cells on the DU map using the above defined functions d and u . A prototype implementation of the machinery needed to obtain the DU map from a GE run is publicly available at <https://github.com/ericmedvet/evolved-ge>.

Since the way each bit is used during the mapping is tightly related to the specific mapping function m being adopted by the GE variant, the proposed usage function u exhibits different properties for the considered variants. In particular, in GE and π GE, groups of 8 or 16 adjacent bits in the genotype have the same value for c_i (and hence $u_i(g)$); moreover, each c_i is limited to n_{wrapping} , i.e., a parameter of both variants representing the maximum number of times the genotype may be reused, if needed, during the mapping— n_{wrapping} is usually set to 10, as in our experiments. In BitSGE, $c_i \in \{0, 1\}$ (and hence $u_i(g)$ can be written simply as $u_i(g) = c_i$) since each bit is used at most once—this is one of the salient features of BitSGE. In WHGE, no further characterization of u may be made, since bits can be reused several times and are not grouped in fixed-length slices.

Population S_{10} at generation 10:

$g_0 = (0, 1, 1, 0, 1, 0, 1, 1)$	$c_0 = (1, 1, 1, 1, 0, 0, 0, 0)$
$g_1 = (0, 1, 1, 0, 1, 1, 0, 1)$	$c_1 = (1, 1, 1, 1, 0, 0, 0, 0)$
$g_2 = (0, 0, 1, 0, 1, 1, 1, 0)$	$c_2 = (1, 1, 1, 1, 1, 1, 0, 0)$
$g_3 = (0, 1, 1, 0, 0, 0, 0, 0)$	$c_3 = (1, 1, 1, 1, 1, 0, 0, 0)$

Color at position $x = 10, y = 2$:

$i_{\text{red}}(10, 2) = d(\{1, 1, 0, 1\}) = 0.5$

$i_{\text{green}}(10, 2) = \frac{1}{4}(1 + 1 + 1 + 1) = 1$

$i_{\text{blue}}(10, 2) = 0$

Color at position $x = 10, y = 6$:

$i_{\text{red}}(10, 6) = d(\{0, 1, 1, 0\}) = 1$

$i_{\text{green}}(10, 6) = \frac{1}{4}(0 + 0 + 1 + 0) = 0.25$

$i_{\text{blue}}(10, 6) = 0$

Figure 2: A color computation example for two cells on the DU map with genotype length $l = 8$. At the top, the genotypes of the four individuals in the population and the corresponding counts of bit usages during the mapping. At the center and the bottom, the computations of the RGB channels.

4 EXPERIMENTS AND DISCUSSION

4.1 General analysis

We performed one run of each of the four GE variants (GE, π GE, SGE, and WHGE) on four benchmark problems (Harmonic, Poly4, Santa-Fe, and Text—the same ones as in [20]) using the parameters shown in Table 1. Concerning the crossover operator for all but BitSGE variants, we used a two-points crossover in which the two cut points are at the same position in the two parents, hence ensuring that the length of the children is the same as the length of the parents.

Figure 3 shows several different visualizations of the results obtained on the Poly4 problem with the genotype length $l = 256$ bit. In the following, we will discuss them in more detail.

The first two rows of visualizations in Figure 3 are simple line graphs presenting some measures over the course of the evolution. The first row shows the best and average fitness of the population in blue and red, respectively. The second row shows two different measures of diversity—the ratio of unique genotypes in the population in blue and the ratio of unique phenotypes in red. We can immediately observe that the phenotype diversity is much lower than the genotype diversity in all GE variants. Also, while the middle two variants increase the genotype diversity in the second half of the evolution, this is hardly reflected in the phenotype diversity.

Next, the third row in Figure 3 contains “zebra” representations of the genotype of the best individual for each generation (black for zeros and white for ones). While these representations allow to spot some portions of the genotype in which more “noise” reflects the occurrence of frequent bit changes (hence suggesting they host

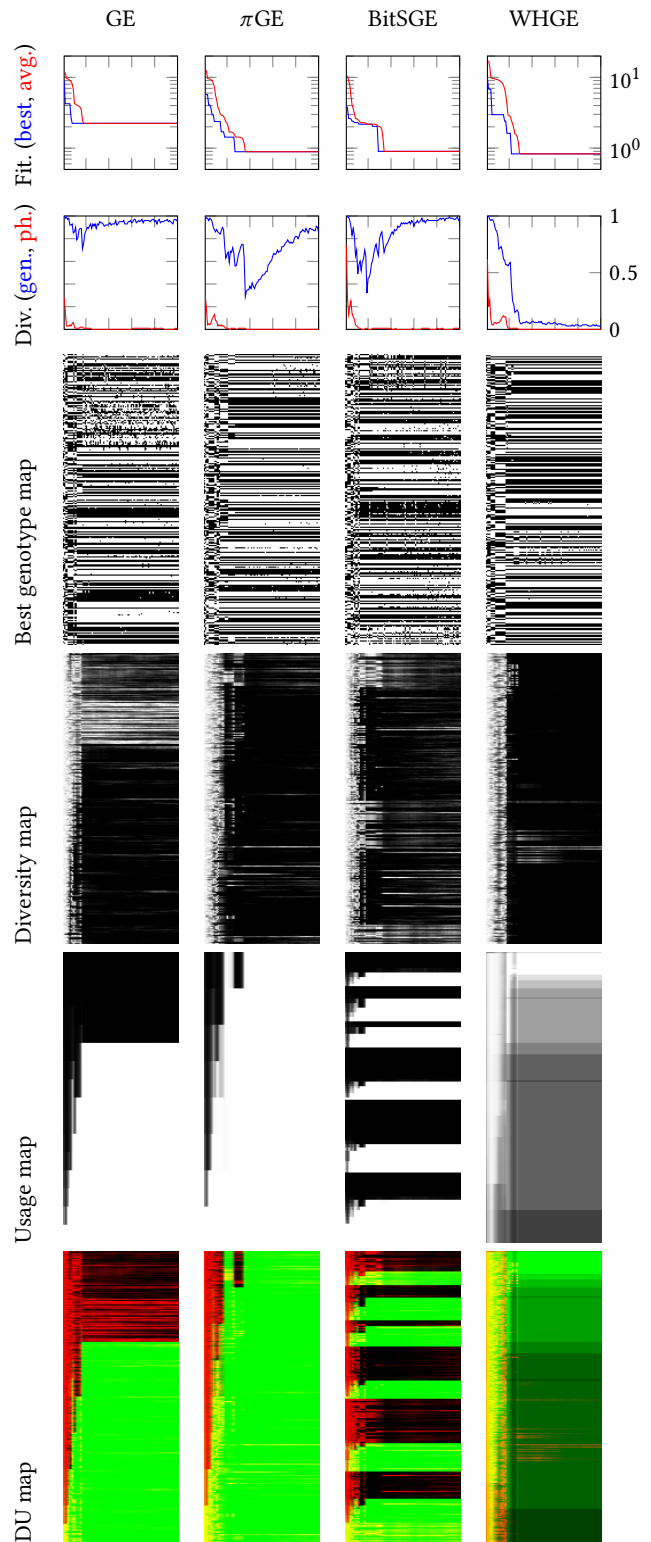


Figure 3: Different plots (rows, see text) for the Poly4 problem tackled with the four variants (columns) with a genotype of 256 bit.

Table 1: Evolution parameters for the four GE variants.

Population size (n_{pop})	500
Population initialization	random
Genotype length l	256 bit
Generations (n_{gen})	100
Crossover op. (all but BitSGE)	two-points
Crossover op. (for BitSGE)	SGE-like
Crossover probability	0.8
Mutation operator	Bit flip with $p_{\text{mut}} = 0.01$
Mutation probability	0.2
Parent selection	tournament with size 3
Survival selection	best fitness
Replacement	$m + n$ with $m = n$
Codon length (for GE, π GE)	8 bit
n_{wrapping} (for GE, π GE)	10
Max depth (for BitSGE)	6
Max depth (for WHGE)	3

some diversity), it does not give any hint about which bits are actually used during the mapping.

The bottom three rows in Figure 3 present gray-scale Diversity and Usage maps and the combined DU maps, respectively. On both gray-scale maps, low values of diversity and usage are shown in black and a high values in white, while the meaning of the colors of the DU maps is presented in Figure 1b.

Three considerations may be done by observing the DU maps of Figure 3. The most important is that the DU map allows to appreciate at a glance that there is a rather clear relation between the location of the diversity and the actual usage of the genotype during the mapping. For GE, π GE, and BitSGE in particular, it can be seen that the portions of the genotype which are not used (not green or yellow) tend to host most of the diversity (red stripes). Moreover, whenever a portion begins to be used, the corresponding bits mostly tend to lose diversity: i.e., the red stripes along the x -axis stop when the green begins (they are rarely continued as yellow lines). The same information is shown separately by the Diversity and Usage maps, but the interaction between the two properties can hardly be appreciated by observing those maps side-by-side.

Second, the green and yellow parts of the DU show how the different mapping functions of the four considered GE variants consume the genotype in order to produce the phenotype. In particular, it can be seen that GE and π GE consume the genotype starting from the lower indexes (i.e., the bottom in our maps); note that the wrapping never occurred in these two examples—instead, wrapping is visible in Figure 4 for π GE on all but one problem and in Figure 5 for GE with $l = 512$ bit on the Santa-Fe problem. In BitSGE, adjacent portions of the genotype are related to different non-terminals (see [16]): within each portion, bits are consumed starting from lower indexes. In WHGE, every bit is used at least once, and the more times they are used, the greater the number of non-terminals they concurred to map (see [21]).

Third, by observing the DU map together with the Fitness and Diversity evolution line graphs, it can be seen that the DU maps capture key events of the evolution. In particular, when the fitness is improving, the green channel (usage) changes along the x -axis.

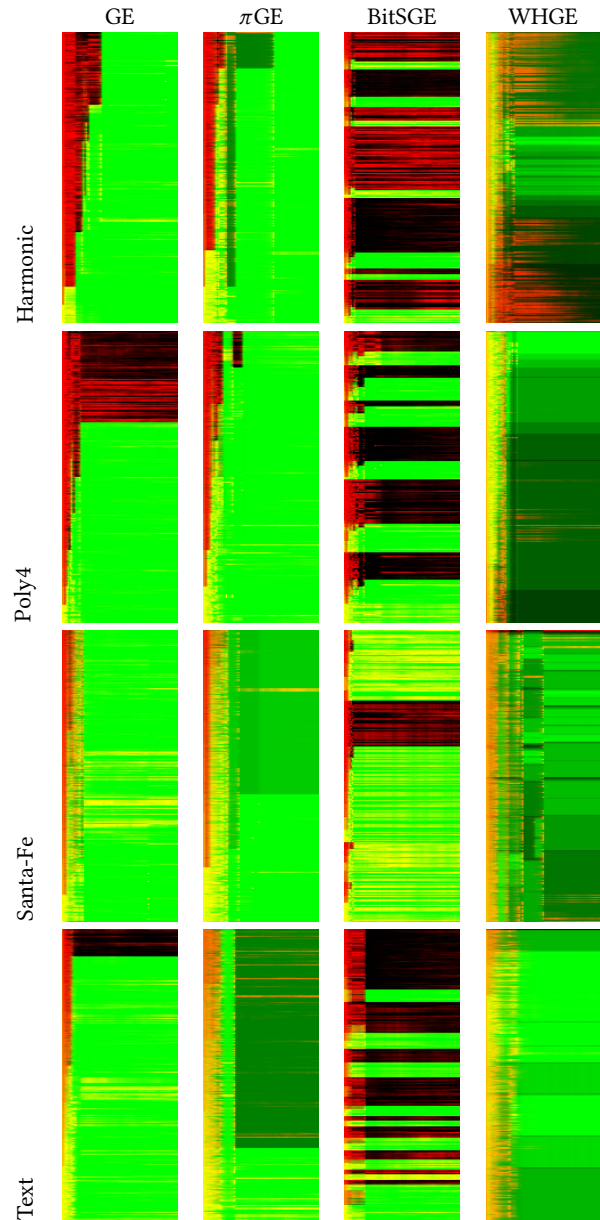
**Figure 4: The DU maps for the four problems (rows) tackled with the four variants (columns) with a genotype of 256 bit.**

Figure 4 shows the DU maps for all the problems and all the GE variants. The figure confirms, in general, the conclusions drawn from Figure 3. Moreover, maps in Figure 4 give some hints about the differences between problems when tackled with the same GE variant. For example, it can be seen how the problem grammars affect the structure of the genotype in BitSGE, reflected by the number and, to some degree, the width of the green and yellow stripes.

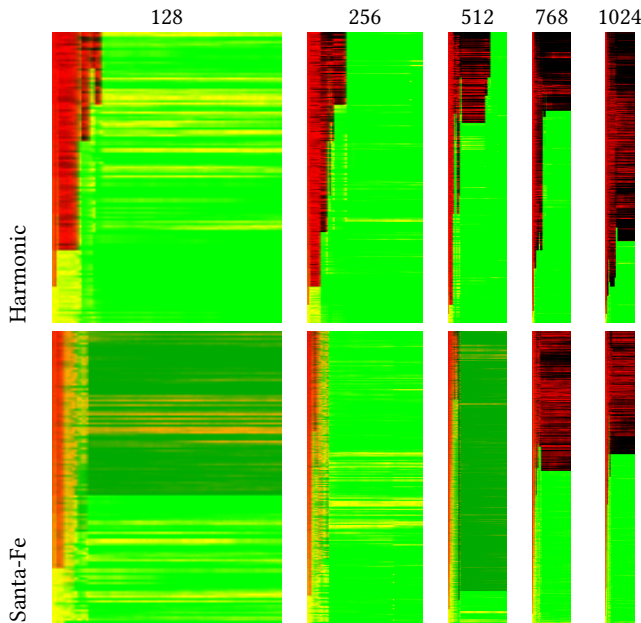


Figure 5: Comparison of five different genotype lengths (columns) for two problems (rows) using the standard GE variant.

4.2 Visualizing the impact of design choices

We performed three additional suites of experiments aimed at verifying if the DU map can be used to investigate the impact of EA design choices or parameters on the evolution. In particular, we focused on: (1) the genotype length l , (2) the introduction of a diversity promotion strategy, (3) the selection pressure.

4.2.1 Genotype length. The length of the genotype is a parameter that can be particularly hard to set: when the genotype is too short, the optimal solution cannot be included in the set of mappable phenotypes, whereas, when it is too long, the search space can be too large to be effectively explored.

We performed different runs by changing the GE variant, the problem, and the value for the genotype length l (128, 256, 512, 768, and 1024 bit). Figure 5 shows the DU maps for the Harmonic and Santa-Fe problems with the standard GE variant: to ease visualization, the maps are rescaled to have constant height.

The DU maps highlight the fact that, for a given problem, only a given number of bits is actually used to generate the solution. From another point of view, the larger the genotype, the smaller the portion of the genotype which is used (green and yellow hues). A large unused portion is an opportunity for redundancy: this could itself constitute an explanation for the interaction between individual size, diversity, and redundancy, which have been recently observed in [20].

4.2.2 Diversity promotion. The lack of diversity has been one of the most debated aspects of many EAs and argued to be a motivation for the premature convergence to sub-optimal solutions [30, 31].

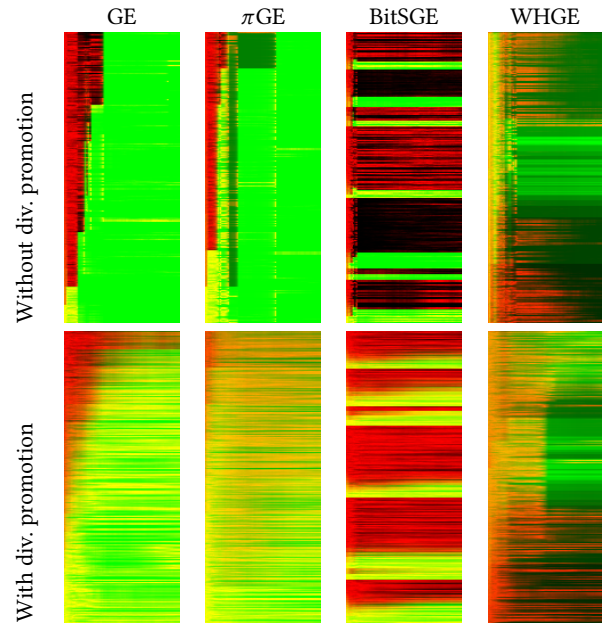


Figure 6: Impact of the phenotype diversity promotion (rows) with the four variants (columns) on the Harmonic problem.

Since our proposed map is designed specifically to visualize diversity, we performed some experiments in order to verify if and how it can visually capture the modification induced by a diversity promotion strategy. To this end, we considered a very simple mechanism consisting in enforcing the diversity at the level of the phenotype. Specifically, whenever a new individual is generated, if another individual with the same phenotype already exists in the population, the old individual is discarded and the new one is inserted in the population. We chose to impose the diversity at the level of the phenotype because it has been showed that, in GE, this results in better effectiveness with respect to the genotype and fitness levels [22].

We performed different runs by changing the GE variant, the problem and, for each combination, turning the diversity promotion mechanism off and on. Figure 6 shows the DU maps for the Harmonic problem with all the variants without diversity promotion (top) and with it (bottom). Some observations can be made.

First, the maps make apparent the fact that enforcing the diversity at the phenotype level also increases the diversity at the genotype level: the maps on the bottom row tend to be less green and black (a sign of low diversity) and more yellow and red (which means high diversity) than the ones on the top row. Moreover, it is easy to appreciate the increase in genotype diversity on the entire length of the genotype: we remark that simply considering the diversity index as the ratio of unique genotypes in the population (see the second row in Figure 3) hardly suggests a similar finding: in facts, the degree to which two genotypes differ is not captured by that index.

Second, besides capturing, by design, the diversity at the genotype level, our map is able to represent also the diversity at the

phenotype level. It can be seen (in particular for GE and BitSGE) that the passage between unused (black and red) and used (green and yellow) areas is sharper in the DU maps obtained without diversity promotion than in the ones with diversity promotion, where it is more blurred. This difference is caused by the fact that i_{green} is the average of the usage on the entire population: if the phenotypes are all equal, the usage of each bit is the same and i_{green} tends to be either 0 or 1, otherwise it takes on different values within $[0, 1]$, resulting in blurriness. In WHGE the difference between using and not using promotion is less apparent, since usage values for WHGE spread more uniformly in the interval $[0, 1]$. Interestingly, it can be seen by comparing maps corresponding to the same variants that the rough shape and position of the green and yellow areas are the same with and without diversity promotion: this suggests that the population is composed of diverse phenotypes which tend to resemble the best one.

4.2.3 Selection pressure. A common cause for the low diversity in Evolutionary Computation (EC) approaches, with respect to the counterpart natural evolution, has been deemed to be in the usage of a fitness function instead of an environment [30]—i.e., a context which can strongly affect (in a stochastic way) the ability of the individual to perform its task. In such a scenario, even a relatively low selection pressure can result in few individuals, possibly sub-optimal, dominating all the others, eventually leading to low diversity in the population. Not surprisingly, the tuning of the selection pressure has been, since the early days of EC, one of the weapons used by EA practitioners to fight the low diversity and, hence, the premature convergence [7]. We hence performed some experiments to verify how our DU map visualizes selection pressure.

We performed different runs by changing the GE variant, the problem, and the survival selection criterion between two options: best fitness (as in all other experiments) and random, respectively corresponding to high and low selection pressure. Figure 7 shows the DU maps for the Poly4 problem for all the GE variants with the worst fitness selection (top) and random selection (bottom).

It can be seen that the maps reflect the lower selection pressure in three ways. First, the “slope” of the leftmost side of the green and yellow areas is less steep with low pressure than with high pressure. This is particularly apparent for the GE and π GE variants, whereas it is, due to its nature, difficult to see for the WHGE variant. Second, the genotype diversity (red and yellow hues) is higher, but still tends to decrease when the evolution seems to have reached an equilibrium (BitSGE and WHGE). Finally, the phenotype diversity is also higher with low selection pressure than with the high one. This is reflected again in the blurriness of the edges between the black/red and green/yellow areas—the edges are blurred for low selection pressure (a sign of high phenotype diversity) and sharp for high selection pressure (low phenotype diversity).

5 CONCLUDING REMARKS

We have proposed the DU map, a novel compact visualization that facilitates reasoning about diversity and its relation to genotype-phenotype mapping. Our technique may be applied to any EA with a twofold representation (genotype and phenotype) and the genotype in the form of a sequence of symbols, as GE. The DU map

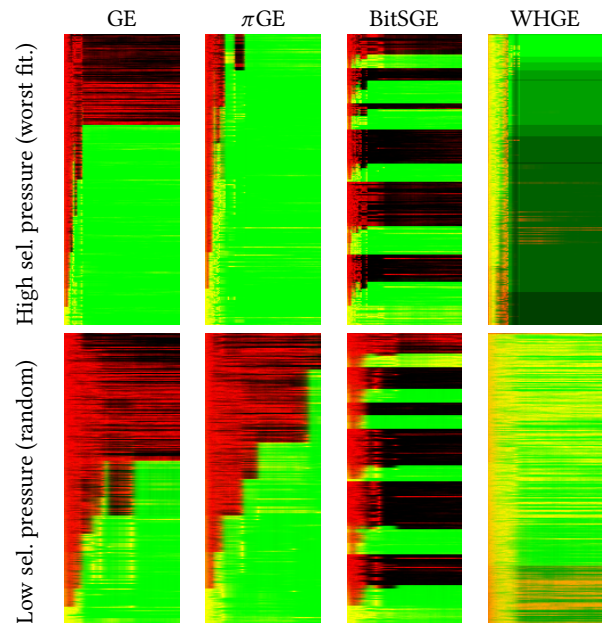


Figure 7: Impact of different selection pressure (rows) with the four variants (columns) on the Poly4 problem.

is a heat map encoding three kinds of information: (a) genotype diversity at the level of the single genotype symbol; (b) degree of genotype symbol contribution to the phenotype (usage); (c) variation of the two previous measures during the evolution.

We performed several experiments aimed at verifying if the DU map is indeed helpful in gaining insights about diversity and its relation to genotype-phenotype mapping. We also explored the usage of the DU map as a tool for taking more informed decisions about different EA design options, such as enabling or disabling a diversity promotion mechanism or varying the selection pressure. The outcome of our experimental validation was promising: we think that the DU map may be a valuable tool for EA, in general, and GE, in particular, researchers and practitioners interested in better understanding the evolution dynamics.

ACKNOWLEDGEMENTS

The authors are grateful to Danny Tagliapietra who contributed to the implementation of the experimental evaluation machinery.

This work is part of a project that has received funding from the *European Union’s Horizon 2020 research and innovation program* under grant agreement No. 692286. This work was partially funded also by the Slovenian Research Agency under research program P2-0209.

REFERENCES

- [1] Fardin Ahmadizar, Khabat Soltanian, Fardin AkhlaghianTab, and Ioannis Tsoulos. 2015. Artificial neural network development by means of a novel combination of grammatical evolution and genetic algorithm. *Engineering Applications of Artificial Intelligence* 39 (2015), 1–13.
- [2] Heni Ben Amor and Achim Rettinger. 2005. Intelligent exploration for genetic algorithms: Using self-organizing maps in evolutionary computation. In *Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2005*. ACM, 1531–1538.

- [3] Lourdes Araujo, Juan Martinez-Romo, and Andrés Duque. 2015. Grammatical evolution for identifying Wikipedia taxonomies. In *Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015*. ACM, 1345–1346.
- [4] Michael Barlow, John Galloway, and Hussein A. Abbass. 2002. Mining evolution through visualization. In *Workshop Proceedings of the Eighth International Conference on Artificial Life, Alife VIII*. MIT Press, 103–110.
- [5] Bogdan Burlacu, Michael Affenzeller, Michael Kommenda, Stephan M. Winkler, and Gabriel Kronberger. 2013. Visualization of genetic lineages and inheritance information in genetic programming. In *Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2013*. ACM, 1351–1358.
- [6] António Cruz, Penousal Machado, Filipe Assunção, and António Leitão. 2015. ELICIT: Evolutionary computation visualization. In *Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015*. ACM, 949–956.
- [7] Kenneth A. De Jong. 2006. *Evolutionary Computation: A Unified Approach*. MIT Press.
- [8] David Fagan, Michael O'Neill, Edgar Galván-López, Anthony Brabazon, and Sean McGarraghy. 2010. An analysis of genotype-phenotype maps in grammatical evolution. In *Proceedings of the 13th European Conference on Genetic Programming, EuroGP 2010 (Lecture Notes in Computer Science)*, Vol. 6021. Springer, 62–73.
- [9] Jonatan Hugosson, Erik Hemberg, Anthony Brabazon, and Michael O'Neill. 2010. Genotype representations in grammatical evolution. *Applied Soft Computing* 10, 1 (2010), 36–43.
- [10] Shin-ichi Ito, Yasue Mitsukura, Hiroko Nakamura Miyamura, Takafumi Saito, and Minoru Fukumi. 2007. A visualization of genetic algorithm using the pseudocolor. In *Revised Selected Papers from the 14th International Conference on Neural Information Processing, ICONIP 2007 (Lecture Notes in Computer Science)*, Vol. 4985. Springer, 444–452.
- [11] Edward Keedwell, Matthew Barrie Johns, and Dragan A. Savić. 2015. Spatial and temporal visualisation of evolutionary algorithm decisions in water distribution network optimisation. In *Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015*. ACM, 941–948.
- [12] Namrata Khemka and Christian Jacob. 2008. What hides in dimension X? A quest for visualizing particle swarms. In *Proceedings of the 6th International Conference on Ant Colony Optimization and Swarm Intelligence, ANTS 2008 (Lecture Notes in Computer Science)*, Vol. 5217. Springer, 191–202.
- [13] Yong-Hyuk Kim, Kang Hoon Lee, and Yourim Yoon. 2009. Visualizing the search process of particle swarm optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2009*. ACM, 49–56.
- [14] John R Koza. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Vol. 1. MIT press.
- [15] Shih-Hsi Liu, Matej Črepinšek, and Marjan Mernik. 2012. Analysis of VEGA and SPEA2 using exploration and exploitation measures. In *Proceedings of the 5th International Conference on Bioinspired Optimization Methods and their Applications, BIOMA 2012*. Jozef Stefan Institute, 97–108.
- [16] Nuno Lourenço, Francisco B Pereira, and Ernesto Costa. 2015. SGE: A structured representation for grammatical evolution. In *Revised Selected Papers from the International Conference on Artificial Evolution (Evolution Artificielle), EA 2015 (Lecture Notes in Computer Science)*, Vol. 9554. Springer, 136–148.
- [17] Nuno Lourenço, Francisco B. Pereira, and Ernesto Costa. 2016. Unveiling the properties of structured grammatical evolution. *Genetic Programming and Evolvable Machines* 17, 3 (2016), 251–289.
- [18] Evelyne Lutton, Julie Fouquier, Nathalie Perrot, Jean Louchet, and Jean-Daniel Fekete. 2011. Visual analysis of population scatterplots. In *Revised Selected Papers from the 10th International Conference on Artificial Evolution (Evolution Artificielle), EA 2011 (Lecture Notes in Computer Science)*, Vol. 7401. Springer, 61–72.
- [19] Nicholas Freitag McPhee, Maggie M. Casale, Mitchell Finzel, Thomas Helmut, and Lee Spector. 2016. Visualizing genetic programming ancestries. In *Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2016*. ACM, 1419–1426.
- [20] Eric Medvet. 2017. *A Comparative Analysis of Dynamic Locality and Redundancy in Grammatical Evolution*. Springer International Publishing, Cham, 326–342. DOI : http://dx.doi.org/10.1007/978-3-319-55696-3_21
- [21] Eric Medvet. 2017. Hierarchical grammatical evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017*. ACM, to appear.
- [22] Eric Medvet, Alberto Bartoli, and Giovanni Squillero. 2017. An effective diversity promotion mechanism in grammatical evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017*. ACM, to appear.
- [23] Eric Medvet, Alberto Bartoli, and Jacopo Talamini. 2017. *Road Traffic Rules Synthesis Using Grammatical Evolution*. Springer International Publishing, Cham, 173–188. DOI : http://dx.doi.org/10.1007/978-3-319-55792-2_12
- [24] Eric Medvet, Fabio Daolio, and Danny Tagliapietra. 2017. Evolvability in grammatical evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017*. ACM, to appear.
- [25] Michael O'Neill, Anthony Brabazon, Miguel Nicolau, Sean Mc Garraghy, and Peter Keenan. 2004. π Grammatical evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2004 (Lecture Notes in Computer Science)*, Vol. 3103. Springer, 617–629.
- [26] Michael O'Neill and Miguel Nicolau. 2017. Distilling the salient features of natural systems: Commentary on “On the mapping of genotype to phenotype in evolutionary algorithms” by Whigham, Dick and Maclaurin. *Genetic Programming and Evolvable Machines* (2017), 1–5. DOI : <http://dx.doi.org/10.1007/s10710-017-9293-0>
- [27] Conor Ryan. 2017. A rebuttal to Whigham, Dick, and Maclaurin by one of the inventors of grammatical evolution: Commentary on “On the mapping of genotype to phenotype in evolutionary algorithms” by Peter A. Whigham, Grant Dick, and James Maclaurin. *Genetic Programming and Evolvable Machines* (2017), 1–5. DOI : <http://dx.doi.org/10.1007/s10710-017-9294-z>
- [28] Conor Ryan, J.J. Collins, and Michael O'Neill. 1998. Grammatical evolution: Evolving programs for an arbitrary language. In *Proceedings of the First European Workshop on Genetic Programming, EuroGP'98 (Lecture Notes in Computer Science)*, Vol. 1391. Springer, 83–96.
- [29] Lukáš Sekanina and Vlastimil Kapusta. 2016. Visualisation and analysis of genetic records produced by Cartesian genetic programming. In *Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2016*. ACM, 1411–1418.
- [30] Giovanni Squillero and Alberto Tonda. 2016. Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization. *Information Sciences* 329 (2 2016), 782–799.
- [31] Giovanni Squillero and Alberto Tonda. 2016. Promoting diversity in evolutionary algorithms: An updated bibliography. In *Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2016*. ACM, 943–944.
- [32] Giovanni Squillero and Alberto Tonda. 2017. (Over-)Realism in evolutionary computation: Commentary on “On the mapping of genotype to phenotype in evolutionary algorithms” by Peter A. Whigham, Grant Dick, and James Maclaurin. *Genetic Programming and Evolvable Machines* (2017), 1–3. DOI : <http://dx.doi.org/10.1007/s10710-017-9295-y>
- [33] Ann Thorhauer. 2016. On the non-uniform redundancy in grammatical evolution. In *Proceedings of the International Conference on Parallel Problem Solving from Nature, PPSN XIV (Lecture Notes in Computer Science)*, Vol. 9921. Springer, 292–302.
- [34] Ann Thorhauer and Franz Rothlauf. 2014. On the locality of standard search operators in grammatical evolution. In *Proceedings of the International Conference on Parallel Problem Solving from Nature, PPSN XIII (Lecture Notes in Computer Science)*, Vol. 8672. Springer, 465–475.
- [35] Zoltán Tóth. 2003. A graphical user interface for evolutionary algorithms. *Acta Cybernetica* 16, 2 (2003), 337–365.
- [36] Tea Tušar and Bogdan Filipić. 2015. Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the projection method. *IEEE Transactions on Evolutionary Computation* 19, 2 (2015), 225–245.
- [37] Tamara Ulrich. 2013. Pareto-set analysis: Biobjective clustering in decision and objective spaces. *Journal of Multi-Criteria Decision Analysis* 20, 5-6 (2013), 217–234.
- [38] Peter A. Whigham, Grant Dick, and James Maclaurin. 2017. On the mapping of genotype to phenotype in evolutionary algorithms. *Genetic Programming and Evolvable Machines* (2017), 1–9. DOI : <http://dx.doi.org/10.1007/s10710-017-9288-x>
- [39] Peter A. Whigham, Grant Dick, James Maclaurin, and Caitlin A. Owen. 2015. Examining the best of both worlds of grammatical evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015*. ACM, 1111–1118.
- [40] Annie S. Wu, Kenneth A. De Jong, Donald S. Burke, John J. Grefenstette, and Connie Loggia Ramsey. 1999. Visual analysis of evolutionary algorithms. In *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, Vol. 2. IEEE, 1419–1425.
- [41] Tina Yu and Julian Miller. 2001. Neutrality and the evolvability of boolean function landscape. In *Proceedings of the 4th European Conference on Genetic Programming, EuroGP 2001 (Lecture Notes in Computer Science)*, Vol. 2038. Springer, 204–217.