

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

Eric Medvet¹, Tea Tušar²

¹: DIA, University of Trieste, Italy

²: DIS, Jožef Stefan Institute, Ljubljana, Slovenia



VizGEC (@GECCO), 15/7/2017, Berlin (Germany)

Table of Contents

- 1 Motivation
- 2 The DU map
- 3 “Validation”

Grammar-based GP

- grammar-based GP approaches are popular
 - user just needs to provide the grammar and the fitness function
 - ensure valid individuals

Grammar-based GP

- grammar-based GP approaches are popular
 - user just needs to provide the grammar and the fitness function
 - ensure valid individuals
- among them, Grammatical Evolution (GE) widely used

Semantic Scholar search results for "grammatical evolution".

Search query: grammatical evolution

Filter Results:

- Full text PDF available (2,807)

Publication Year

1964 2017

This year (68)
Last five years (987)

Publication Type

Author

Publication Venue

About 3,220 results

Grammatical evolution

[Conor Ryan](#) • GECCO • 2001

Grammatical Evolution is an automatic programming system that is a form of Genetic Programming structures. These structures can be in any form that can be specified using a grammar, in computer networks. When evolving computer languages, multiple types can be handled in a complete

[View PDF](#) [Related Publications](#) [More...](#)

Grammatical Evolution Evolutionary Automatic Programming Edition

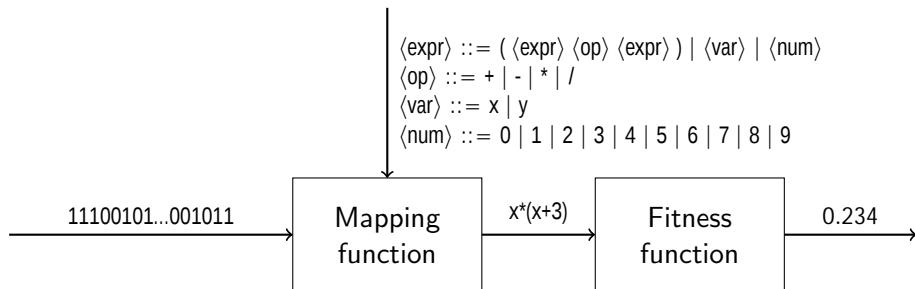
[Maria Adler](#) • 2016

Number of pages: 197 pages Thank you for reading **grammatical evolution** evolutionary automatic programming in an arbitrary language 1st edition. Maybe you have knowledge that, people have look numerous times **evolution** evolutionary automatic programming in an arbitrary language 1st edition, but en

GE

Three representations for the individual:

- genotype: a bit string
- phenotype: a string of the language defined by the CFG
- fitness



Genotype-phenotype mapping

Salient trait:

- allows using (standard) bit string genetic operators
- quite indirect mapping, poor variational inheritance
 - many variants proposed (GE, SGE, WHGE, ...)
 - large/hot debate: “GE mapping is bad!”¹, “No! It’s good!”²³

¹Whigham, Dick, and Maclaurin, “On the mapping of genotype to phenotype in evolutionary algorithms” .

²Squillero and Tonda, “(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” .

³Ryan, “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” .

Mapping and properties

Mapping:

- invalidity: how often the mapping “fails”?
- redundancy⁴: is part of the genotype not used?
- degeneracy⁴: how often different genotypes are mapped to the same phenotype?

⁴Often, redundancy is used for degeneracy

Mapping and properties

Mapping:

- invalidity: how often the mapping “fails”?
- redundancy⁴: is part of the genotype not used?
- degeneracy⁴: how often different genotypes are mapped to the same phenotype?

Mapping + genetic operators:

- locality: are genotypic neighbors also phenotypic neighbors?
- neutrality: how often the child and parent with different genotypes share the same phenotype?

⁴Often, redundancy is used for degeneracy

Mapping and properties

Mapping:

- invalidity: how often the mapping “fails”?
- redundancy⁴: is part of the genotype not used?
- degeneracy⁴: how often different genotypes are mapped to the same phenotype?

Mapping + genetic operators:

- locality: are genotypic neighbors also phenotypic neighbors?
- neutrality: how often the child and parent with different genotypes share the same phenotype?

Mapping + genetic operators + fitness:

- evolvability: is the child fitter than its parents?

⁴Often, redundancy is used for degeneracy

Properties: the big picture?

Several studies, some (conflicting) claims:

- degeneracy improves diversity⁵
- degeneracy causes lack of phenotypic diversity⁶
- neutrality (negatively) correlated with evolvability⁷
- degeneracy biased towards short individuals⁸
- diversity promotion leads to larger, better individuals⁹

⁵ O'Neill and Ryan, "Genetic code degeneracy: Implications for grammatical evolution and beyond".

⁶ Medvet, "A Comparative Analysis of Dynamic Locality and Redundancy in Grammatical Evolution".

⁷ Medvet, Daolio, and Tagliapietra, "Evolvability in grammatical evolution".

⁸ Thorhauer, "On the non-uniform redundancy in grammatical evolution".

⁹ Medvet, Bartoli, and Squillero, "An effective diversity promotion mechanism in grammatical evolution".

Properties: the big picture?

Several studies, some (conflicting) claims:

- **degeneracy** improves **diversity**⁵
- **degeneracy** causes lack of phenotypic **diversity**⁶
- neutrality (negatively) correlated with evolvability⁷
- **degeneracy** biased towards short individuals⁸
- **diversity** promotion leads to larger, better individuals⁹

⁵ O'Neill and Ryan, "Genetic code degeneracy: Implications for grammatical evolution and beyond".

⁶ Medvet, "A Comparative Analysis of Dynamic Locality and Redundancy in Grammatical Evolution".

⁷ Medvet, Daolio, and Tagliapietra, "Evolvability in grammatical evolution".

⁸ Thorhauer, "On the non-uniform redundancy in grammatical evolution".

⁹ Medvet, Bartoli, and Squillero, "An effective diversity promotion mechanism in grammatical evolution".

Properties: the big picture?

Several studies, some (conflicting) claims:

- degeneracy improves diversity⁵
- degeneracy causes lack of phenotypic diversity⁶
- neutrality (negatively) correlated with evolvability⁷
- degeneracy biased towards short individuals⁸
- diversity promotion leads to larger, better individuals⁹

And some reasonable intuitions:

- evolvability is good
- redundancy) degeneracy) neutrality

⁵ O'Neill and Ryan, "Genetic code degeneracy: Implications for grammatical evolution and beyond".

⁶ Medvet, "A Comparative Analysis of Dynamic Locality and Redundancy in Grammatical Evolution".

⁷ Medvet, Daolio, and Tagliapietra, "Evolvability in grammatical evolution".

⁸ Thorhauer, "On the non-uniform redundancy in grammatical evolution".

⁹ Medvet, Bartoli, and Squillero, "An effective diversity promotion mechanism in grammatical evolution".

Studying GE and its properties

Often observed through coarse-grained measures:

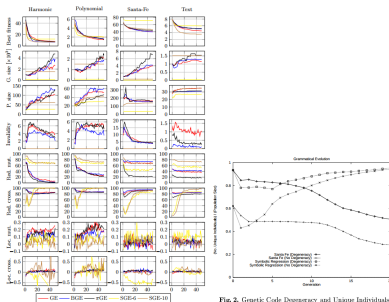


Fig. 2. Genetic Code Degeneracy and Unique Individuals

Studying GE and its properties

Often observed through coarse-grained measures:

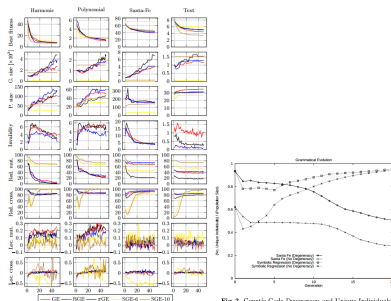


Fig. 2. Genetic Code Degeneracy and Unique Individuals

Can we look closer (and see better)?

Table of Contents

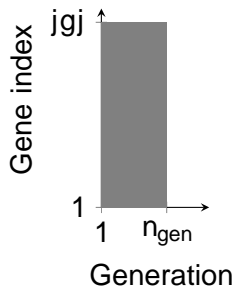
- 1 Motivation
- 2 The DU map
- 3 "Validation"

Our proposal in a nutshell

A novel, compact visualization:

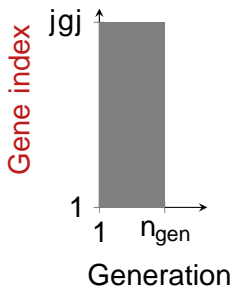
- genotypic **diversity** at the level of the single bit
- contribution (*usage*) of the single bit to the mapping
(: redundancy,) **degeneracy**)
- how they vary (**interact**) during the evolution

DU map: how to read it



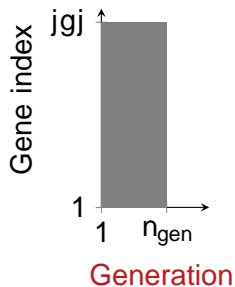
position:

DU map: how to read it



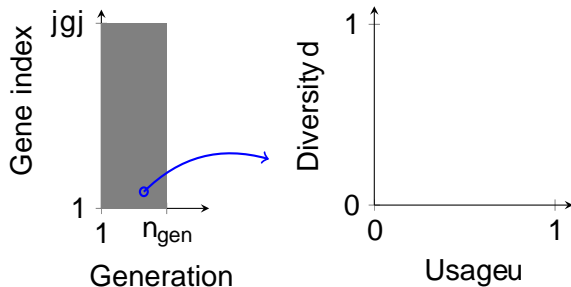
position: wherein the genotype?

DU map: how to read it



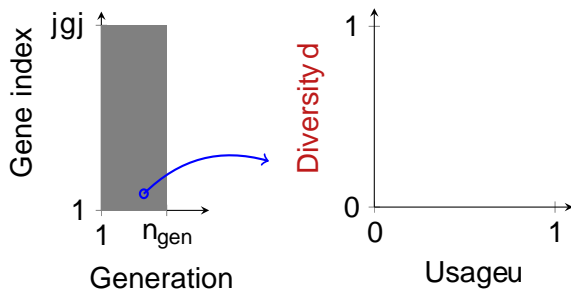
position: wherein the genotype? when during the evolution?

DU map: how to read it



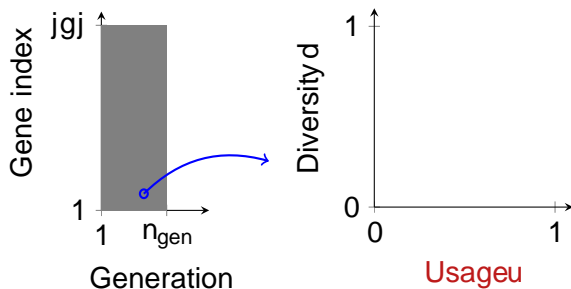
position: wherein the genotype?when during the evolution?
 color: for a given bit, at a given generation,

DU map: how to read it



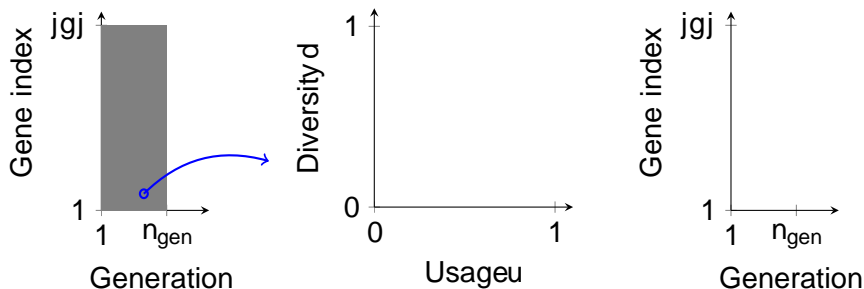
position: wherein the genotype? when during the evolution?
 color: for a given bit, at a given generation, how diverse is that bit now?

DU map: how to read it



position: wherein the genotype? when during the evolution?
 color: for a given bit, at a given generation, how diverse is that bit now? **to which degree is it used** now?

DU map: how to read it



position: wherein the genotype? when during the evolution?

color: for a given bit, at a given generation, how diverse is that bit now? to which degree is it used now?

all together: view of one run

How diverse is that bit? red channel i_{red}

$$i_{red}(x; y) = d(f g_y; s \in S_x g) \quad (1)$$

$d(A) \in [0; 1]$: a measure of diversity of a set of symbols

g_y : y th symbol of the genotype of the individuals

S_x : population at the x th generation

How diverse is that bit? red channel_{red}

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

$d(A) \in [0; 1]$: a measure of diversity of a set of symbols

g_y : y th symbol of the genotype of the individuals

S_x : population at the x th generation

For bit string genotypes:

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

$d(A) = 0$ if all the same, $d(A) = 1$ if half 0 and half 1

Degree of usage of that bit! green channel i_{green}

$$i_{\text{green}}(x; y) = \frac{1}{|S_x|} \sum_{s \in S_x} u_y(g) \quad (3)$$

$u(g) \in [0; 1]^{|S_x|}$: a (vector) measure of usage of each symbol in S_x
 $u_y(g)$: y th component of the vector $u(g)$

Degree of usage of that bit! green channel i_{green}

$$i_{\text{green}}(x; y) = \frac{1}{|S_x|} \sum_{s \in S_x} u_y(g) \quad (3)$$

$u(g) \in [0; 1]^{|S_x|}$: a (vector) measure of usage of each symbol in

$u_y(g)$: y th component of the vector $u(g)$

For bit string genotypes:

$$u((g_1; g_2; \dots)) = \frac{1}{\max_{i \in \{1, 2, \dots, n\}} c_i} (c_1; c_2; \dots) \quad (4)$$

c_i : number of times i th bit been used during the mapping
depends on the mapping

$c_i \in \{0; 1; \dots; n_{\text{wrapping}}\}$ for GE, GE

$c_i \in \{0; 1\}$ for SGE

$c_i = 1$ for HGE/WHGE

Table of Contents

- 1 Motivation
- 2 The DU map
- 3 \Validation"

How to validate?

is the DU map better than its components?

is it helpful to understand GE and its properties?

can it be used to compare design options?

DU Map vs. its components

Poly4 benchmark, GE mapping w/ $w_{\text{wrapping}} = 10$, $|g_j| = 256$ bit, 100 generations

Best bits Diversity Usage DU map

DU Map vs. its components

Poly4 benchmark, GE mapping $w_{\text{wrapping}} = 10$, $j_{\text{g}} = 256$ bit, 100 generations

Best bits Diversity Usage DU map



initially, best evolves "everywhere", then only on a portion

DU Map vs. its components

Poly4 benchmark, GE mapping w/ $w_{\text{wrapping}} = 10$, $|j| = 256$ bit, 100 generations

Best bits Diversity Usage DU map

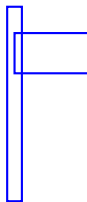


initially, best evolves "everywhere", then only on a portion

DU Map vs. its components

Poly4 benchmark, GE mapping w/ $n_{\text{wrapping}} = 10$, $|jg| = 256$ bit, 100 generations

Best bits Diversity Usage DU map



- initially, best evolves “everywhere”, then only on a portion
- initially, diversity “everywhere”, then only on a portion

DU Map vs. its components

Poly4 benchmark, GE mapping w/ $n_{\text{wrapping}} = 10$, $|jg| = 256$ bit, 100 generations

Best bits Diversity Usage DU map



- initially, best evolves “everywhere”, then only on a portion
- initially, diversity “everywhere”, then only on a portion
- initially, used portion (and pheno size) increases

DU Map vs. its components

Poly4 benchmark, GE mapping w/ $n_{\text{wrapping}} = 10$, $|jg| = 256$ bit, 100 generations

Best bits Diversity Usage DU map



- initially, best evolves “everywhere”, then only on a portion
- initially, diversity “everywhere”, then only on a portion
- initially, used portion (and pheno size) increases
- **diversity is in unused (redundant) portion**) degeneracy) neutrality
) low phenotypic diversity

Comparing mappings and problems

GE GE BitSGE WHGE

Text

- “structure” of the genotype
- wrapping (GE, GE), no redundancy (WHGE)

Santa-Fe

Comparing mappings and problems

GE GE BitSGE WHGE

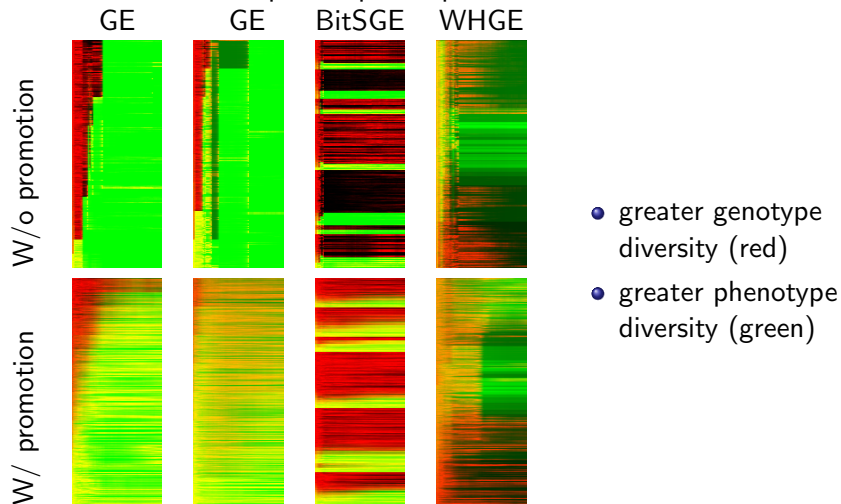
Text

Santa-Fe

- “structure” of the genotype
- wrapping (GE, GE), no redundancy (WHGE)
- phenotype size (estimate)

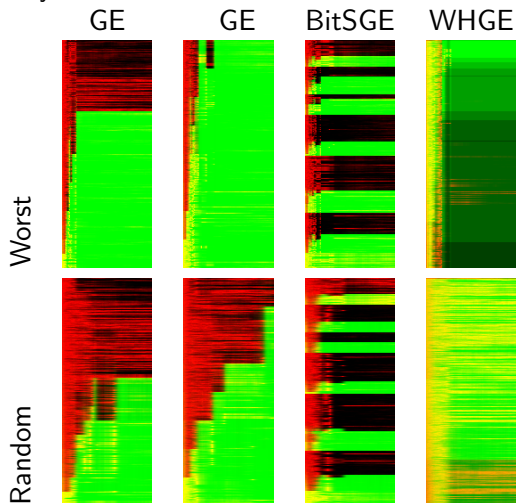
"Looking inside" diversity promotion

Harmonic, vanilla vs. phenotype uniqueness enforcement



“Looking inside” selection pressure

Poly4, remove worst vs. random



- more diversity (both genotype and phenotype)
- slower evolution

Conclusions

GE (and variants):

- used, studied, improved
- mapping properties effectiveness: still unclear

DU map:

- visualizes mapping, redundancy, diversity
- helps in designing improvements

Thanks!