

Fitness landscape



from biological science (Wright 1930 [30]) :
Modification of species evolution



Used to model dynamical systems :

- statistical physics,
- molecular evolution,
- ecology, etc



Fitness Landscapes and Problem Hardness in Evolutionary Computation

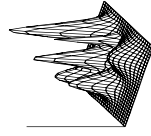
Sebastien Verel
Leonardo Vanneschi

IS laboratory
University of the Saarland, CRIS (France)
University of Turin, CRIS (Italy)
<http://www.is.uni-saarland.fr/~verel/>
<http://personal.csis.uinimbu.br/Vanneschi>

July, 7 2007

Copyright is held by the author/owner(s).
GECCO'07, July 7–11, 2007, London, England, United Kingdom.
ACM 978-1-59593-698-1/07/0007.

In combinatorial optimization

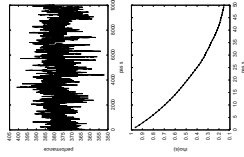


Fitness landscape (S, \mathcal{N}, f) :

- S : set of potential solutions,
- $\mathcal{N} : S \rightarrow 2^S$: neighborhood function,
- $f : S \rightarrow \mathbb{R}$: fitness function.

Outline

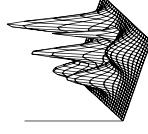
- Part 1 : Fitness landscapes in EA
 - 1 Introduction
 - 2 Multimodal and rugged fitness landscapes
 - 3 Neutral fitness landscapes
- Part 2 : Fitness landscapes in Genetic Programming



Autocorrelation of $(f(s_1), f(s_2), \dots)$ along a random walk (s_1, s_2, \dots) (Wienberger 1990 [28]):

$$\rho(n) = \frac{E[(f(s_i) - \bar{f})(f(s_{i+n}) - \bar{f})]}{\text{var}(f(s))}$$

- autocorrelation length $\tau = \frac{1}{\rho(1)}$
- small τ : rugged landscape
 - long τ : smooth landscape



$$\mathcal{V}: S \rightarrow 2^S \text{ : neighborhood function}$$

$$\forall x \in S, \mathcal{V}(x) = \{y \in S \mid d(y, x) \leq 1\}$$

Problem	$\rho(1)$
symmetric TSP	$1 - \frac{1}{n}$
antisymmetric TSP	$1 - \frac{1}{n^2}$
Graph Coloring Problem	$1 - \frac{1}{(n-1)^2}$
NK landscapes	$1 - \frac{1}{K^{n-1}}$

Local optima : no neighbor solution with higher fitness value
Adaptive walk : (s_0, s_1, \dots) where $s_{i+1} \in \mathcal{V}(s_i)$ and $f(s_i) < f(s_{i+1})$
Attraction basin of s_{opt} : set of solutions of a adaptive walk to s_{opt}



Optimisation difficulty : number and size of attractive basins (Garnier et al [8])

Multimodal and rugged fitness landscapes
 Introduction
 Fitness landscapes and prob. James
 Sebastian Wedi, Leonardo Vannuchi

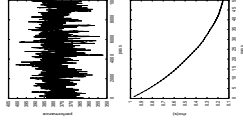
Fitness landscape and crossover ?

- Crossover of complementary strings (Culberson 94 [6]) : comparison with mutation-landscape
- Hypergraph of Cichoff [9] :
- P-structure of Stadler [25] : generalisation of graph theory
- crossover with a random solution
- Space of pair of solutions (Jones, Delfin [19] : study of linear GP crossover)

Maybe the schemata theorem give better results ?...

Multimodal and rugged fitness landscapes
 Introduction
 Fitness landscapes and prob. James
 Sebastian Wedi, Leonardo Vannuchi

Multimodality, ruggedness, epistasis ?



- multimodality/ruggedness : conjecture (Stadler 92 [24], Garcia 97 [7]) :
- I local optimum per boule of rayon τ
- epistasis/ruggedness :
- NK fitness landscapes (Kauffman [14])
 $\tau = \frac{1}{\ln(1 - \frac{1}{2^N})}$ et $d = \frac{N \log_2(K+1)}{2(K+1)}$
- But some counterexamples...
 $\sum_{j=1}^N \exp(-j)/x_j$ et $\prod_{j=1}^N x_j$
- ... open question ?

Multimodal and rugged fitness landscapes
 Introduction
 Fitness landscapes and prob. James
 Sebastian Wedi, Leonardo Vannuchi

Synthesis

- Metaphor from the biology
- Study of multimodal fitness landscapes :
 → optimization algorithms (SA, Tabu Search, Island Model...)

Goal of study of fitness landscapes :

- Links with problem hardness : to make better choices (mapping, fitness function, operators), design of algorithms, etc.
- Proof of convergence (with speed of convergence)
- Autoadaptation of parameters of research

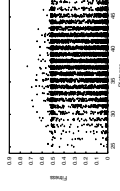
Limits :

- 1 operator = 1 landscape ?
- Crossover ? and link with population ?
- Edges : useful information ?

Multimodal and rugged fitness landscapes
 Introduction
 Fitness landscapes and prob. James
 Sebastian Wedi, Leonardo Vannuchi

Fitness Distance correlation (FDC) (Jones 95 [13])

Correlation between distance to global optimum and fitness



Experimental scale :

- $\rho < -0.15$, easy optimization
- $\rho > 0.15$, hard optimization
- $-0.15 < \rho < 0.15$, undecided zone

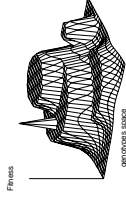
What optimizers do with neutrality?

- Three possibilities:
- Decrease the neutrality
 - Use a specific metaheuristic
 - Increase the neutrality with redundant genotype/phenotype mapping

Neutral Fitness Landscapes

Neutral theory (Kimura \approx 1960 [15])
Theory of mutation and random drift

A considerable number of mutations have no effects on fitness values



- plateaus
- neutral degree
- neutral networks [Schuster 1994, [2]], RNA folding

Decreasing the neutrality (minLA : E. Rodriguez, PPSN05 [20])

Redundant encoding is a drawback, lack of information
graph $G = (V, E)$: labeled each nodes

$$LA(G, \varphi) = \sum_{(u,v) \in E} |\varphi(u) - \varphi(v)| \in \mathbb{N}$$

LA represents a potential drawback, because different linear arrangements can result in the same total edge length. This incomplete information can prevent the search process from finding better solution.

$$\phi(G, \varphi) = LA(G, \varphi) + f_{norm}(G, \varphi)$$

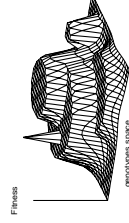
with $f_{norm}(G, \varphi) \in [0, 1]$

- f_{norm} is higher when the differents could be optimized
- f_{norm} makes different between equivalent labelling

Neutral fitness landscapes

Combinatorial optimization

- Problem "not well" defined or changeable environment (Torres 04 [12])
- Redundant problem (symetries, ...) (Goldberg 87 [10])



- Applicative problems :
- Robot controller
 - Circuit design
 - genetic programming
 - Protein Folding
 - learning problems

Neutrality of the problem can not be changed

- Metrowler (L. Barnett [2])

```

step ← 0
Choose initial solution  $s \in S$ 
repeat
  choose  $s' \in \mathcal{N}(s)$  randomly
  if  $f(s') \leq f(s)$  then
     $s \leftarrow s'$ 
  end if
step ← step + 1
until step/max ≤ step
    
```

Good results on ρ -correlated landscapes :
 low probability to find a better solution
 high probability to find a solution with same fitness

Neutrality of the problem can not be changed

- Metrowler (L. Barnett [2])

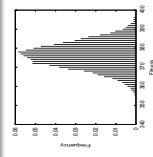
```

step ← 0
Choose initial solution  $s \in S$ 
repeat
  choose  $s' \in \mathcal{N}(s)$  randomly
  if  $f(s') \leq f(s)$  then
     $s \leftarrow s'$ 
  end if
step ← step + 1
until step/max ≤ step
    
```

Good results on ρ -correlated landscapes :
 low probability to find a better solution
 high probability to find a solution with same fitness



Set of solution with fitness value

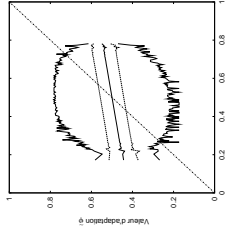


- Introduce in physic (Rose 1996 [21])
- Optimization (Saldoum, J.K. Hoo 2000 [4])

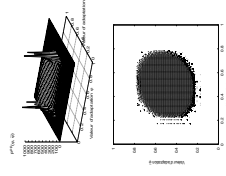
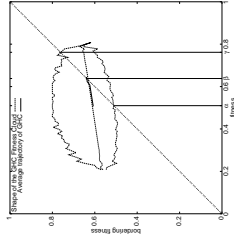
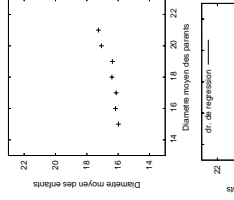
Neutrality of the problem can not be changed

- Extrema selection (Stewart [26]) :

It is sufficient to recognise that the neutrality of a fitness function may be a significant issue when solving solutions. With this in mind, the remainder a novel modification to the standard GA, which is specifically designed to take in advantage of Neutral Network.
 When the solutions are in the same plateaus (0.9 from best solution)
 → selection according to the distance from the centroid of the population



- Probability of finding better solutions
- Average fitness of better neighbor solutions
- Average and standard deviation of fitnesses



$$X(s) = f(s)$$

$$Y(s) = f(op(s))$$

Fitness Cloud de op :
 conditional probability density
 function of Y given X

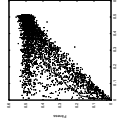
Definitions

- A test of *neutrality* is a predicate $isNeutral : S \times S \rightarrow \{true, false\}$
- For example, $isNeutral(s_1, s_2)$ is true if :
 - $f(s_1) = f(s_2)$,
 - $|f(s_1) - f(s_2)| \leq 1/M$ with M is the population size,
 - $|f(s_1) - f(s_2)|$ is under the evaluation error.
- The *neutral neighborhood* of s is the set of neighbors which have the same fitness $f(s)$

$$V_{near}(s) = \{s' \in V(s) \mid isNeutral(s, s')\}$$
- The *neutral degree* of a solution is the number of its neutral neighbors

$$nDeg(s) = \#(V_{near}(s) - \{s\}).$$

Negative slope coefficient



- Fitness cloud with another sample techniques :
- Metropolis-Hasting sampling
 - Operator : tournament of size t
 - Partition by m segments
- Negative slope coefficient :

$$NSC = \sum_{i=1}^{m-1} \min(P_i, 0)$$

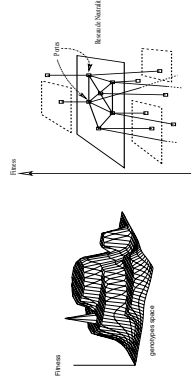
- GP problem :
- Problème even Parity
 - Problème binomial-3
 - Fourmi Sante-Fe

Definitions

- A *neutral walk* : $W_{near} = (s_0, s_1, \dots, s_m)$ for all $i \in \{0, m-1\}$, $s_{i+1} \in V(s_i)$ for all $(i, j) \in \{0, m\}^2$, $isNeutral(s_i, s_j)$ is true.
- A *Neutral Network* : graph $G = (V, E)$ $V \subset S$: for all s and s' from V , there is a neutral walk W_{near} belonging to V from s to s' . Two vertices are connected by an edge of E if they are neutral neighbors.

A fitness landscape is neutral if there are many solutions with high neutral degree.

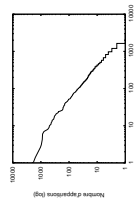
Neutral networks (Schuster 1994 [22])



Intra network Measures
 Size

Classical measures of graph to describe NN :

- 1 The size : number of nodes of NN,
 rank-size of NN in log-log :



Frequency of apparition of a word in a text (Ziff law 1960)

Neutral NK fitness landscapes

$$NK \quad f(s) = \frac{1}{2^k} (0.02 + 0.31 + 0.91 + \dots + 0.20)$$

$$NK_q \text{ (Newmann et al 1998) [17]}$$

$$f(s) = \frac{1}{2^{kq-1}} (1 + 3 + 3 + \dots + 0)$$

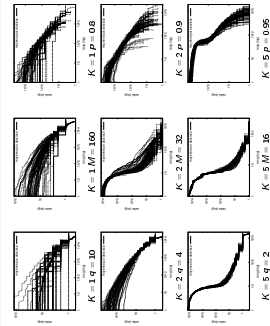
$$NK_M \text{ (Lobo 2004 [16])}$$

$$f(s) = \frac{1}{2^{kq}} E[M] (0.02 + 0.31 + 0.91 + \dots + 0.20)$$

$$NK_p \text{ (Barnett 1998 [1])}$$

$$f(s) = \frac{1}{2^k} (0.02 + 0.31 + 0 + \dots + 0.20)$$

Size of neutral networks



Intra network Measures

Classical measures of graph to describe NN :

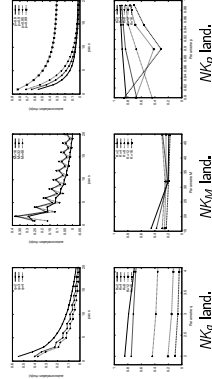
- 1 the size : number of nodes of NN,
- 2 neutral degree distribution :
 measure of the quantity of "neutrality"
- 3 Autocorrelation of neutral degree during neutral walk (Bastolla [3]) :
 comparison with random graph,
 measure of the correlation structure of NN

Classical measures of graph to describe NN :

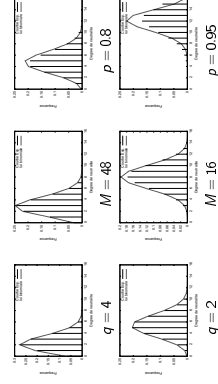
1. the size : number of nodes of NN,
 2. neutral degree distribution : measure of the quantity of "neutrality"
 3. Autocorrelation of neutral degree during neutral walk (Bastolla [3]) :
- comparison with random graph,
 measure of the correlation structure of NW

Classical measures of graph to describe NN :

1. the size : number of nodes of NN,
 2. neutral degree distribution : measure of the quantity of "neutrality"
 3. Autocorrelation of neutral degree during neutral walk (Bastolla [3]) :
- comparison with random graph,
 measure of the correlation structure of NW

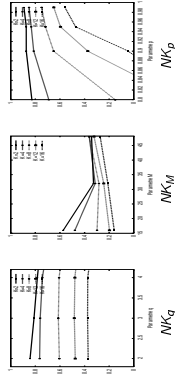


Autocorrelation coefficient of order 1



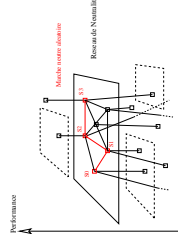
Experimental distribution (impluse), binomial distribution (line).

Autocorrelation of maximal evolvability



autocorrelation coefficient of order 1 for $N = 64$

Inter networks measure



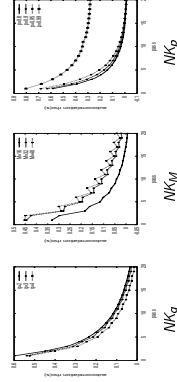
1. rate of innovation (Huynen [11]) :
The number of new accessible structures (fitness) per mutation
2. Autocorrelation of evolvability [28] : autocorrelation of the serie $(evol(s_1), evol(s_2), \dots)$.

Synthesis Measures

- neutral degrees distribution : "How much neutrality in the landscape?"
 - Autocorrelation of neutral degrees : network "structure"
-
- rank-size of NN in log-log : well adapted representation (complex systems)
 - rate of innovation : low information for combinatorial optimization
 - Autocorrelation of maximal evolvability : information on the links between NN

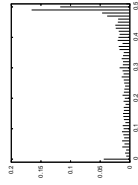
Autocorrelation of maximal evolvability [28]

maximal evolvability autocorrelation function for $N = 16, K = 2$



Density of States

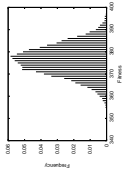
Sampling	Random	Metropolis-Hastings
Fitness = 0	3979	176
Total	4000	4000



Metropolis-Hastings sampling :

- a considerable number of solutions sampled with a fitness ≈ 0.5
- No solution with a fitness value superior to 0.55

Statistical Measures of neutrality

- Density Of States (DOS) [H. Rosé 96] :
- 
- Diameter of NN : diameter of the graph

Study of Neutral Networks

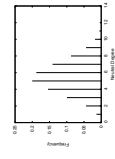
Study of two important large neutral networks :

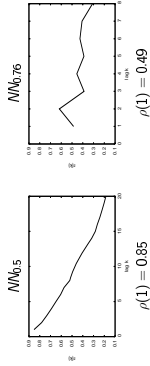
- MM_5 : fitness around 0.5
 Automata that solve the problem on only half of ICs,
 5 neutral walks.
- MM_{76} : fitness around 0.765
 Solutions near a CA found by Mitchell (GA),
 19 neutral walks.

Neutral walks :

- Same starting point on each NN
- Strictly increasing the Hamming distance from the starting solution,
- Stops when there is no neutral step that increases distance.

Statistical Measures of neutrality

- Neutral degree distribution : degree distribution of the vertices
- 
- Autocorrelation of neutral degree along a neutral random walk [Bastolla 03]



- Correlation is not null
 - Correlation for $MM_{6.5} >$ Correlation for $MM_{6.76}$
 - Graphs of Neutral Networks are not random graphs
 - Variation of neutral degree is smooth on MW
- ⇒ important consequence on metaheuristic design

Average length of neutral walks (max 128) :

$MM_{6.5}$	108.2
$MM_{6.76}$	33.1

Result on diameter :

Diameter of $MM_{6.5} >$ Diameter of $MM_{6.76}$.

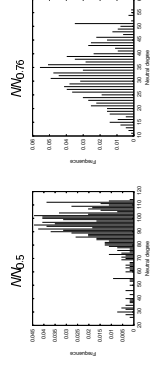
Two symmetries that do not change performance :
 0/1 symmetry and *right / left* symmetry.

Symmetries of *block* which maximize the number of joint bits
 $GK^L = GK^R$, $Das^L = Das^R$, $Davis^L = S_{21}(Davis^R)$,
 $ABK^L = S_{21}(ABK^R)$, $Coel^L = Coel$, $Coel^R = S_{21}(Coel^L)$.

Olympus Landscape, subspace of dimension 77 :

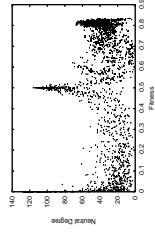
```
0000000 0000000 0000000 0000000 0000000 0000000 0000000 0000000
0000000 0000000 0000000 0000000 0000000 0000000 0000000 0000000
0000000 0000000 0000000 0000000 0000000 0000000 0000000 0000000
```

Distribution of neutral degree collected along all neutral walks.



- $MM_{6.76}$: close to normal,
- $MM_{6.5}$: skewed and approximately bimodal

Neutral Degree

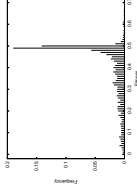


- Two large neutral networks at fitnesses 0 and 0.5 : neutral degree > 70.
- Over fitness 0.5 : average of neutral degree is 37.6.

Density Of States

Null fitness value :

Sampling	Random	MH
Whole search space	99.9%	4.4%
Olympus	28.6%	0.3%



Metropolis-Hastings sampling :

- No solution with a fitness value superior to 0.68
- Advantageous to concentrate the search in the Olympus landscape.

Main Results on Fitness landscape

- Considerable number of CAs of performance 0 or 0.5
- High neutrality of the landscape
- Neutral networks studies are not random graphs
- Fitness landscape of Majority Problem is very hard !
- Defined the Olympus landscape : exploring similarities between the six best rules.
- Less solutions with performance 0
- Easy to find solutions over 0.80 with a simple GA.
- Over performance 0.5 : neutrality of landscape is still high

Neutral Degree : Sampling Method

- Solutions < 0.5 : randomly chosen in Olympus.
- Solutions > 0.5 : from 2 runs of a GA during 10⁷ generations.

AG used :

- Based on GA defined by Mitchell
- Operators : restricted to Olympus subspace
- Selection : tournament selection taking into account the neutrality.

Justified the usefulness of Olympus and neutrality :

- Discover a lot of solutions between 0.80 and 0.835
- Over 50 runs, average performances 0.832,006 (evolutionary AG of Page 02 : 0.80,02).

- 2 Ricardo García-Péjayo and Peter F. Stadler.
Correlation length, isotropy, and meta-stable states.
Physica D, 107 :240–254, 1997.
Santa Fe Institute Preprint 95-05-034.
- 2 Josselin Garnier and Leila Kallal.
Efficiency of local search with multiple local optima.
SIAM Journal on Discrete Mathematics, 15(1) :122–141, 2002.
- 2 P. Gächter and G. Wagner.
Recombination induced hypergraphs : A new approach to
mutation-recombination isomorphism, 1996.
- 2 David E. Goldberg and Philip Segrest.
Finite markov chain analysis of genetic algorithms
In *CGA*, pages 1–8, 1987.
- 2 M. Huynen.
Exploring phenotype space through neutral evolution.
Sébastien Verel, Loonard Vannest | Fitness landscapes and prob. games

- 2 L. Barnett.
Ruggedness and neutrality – the NKp family of fitness
landscapes.
In C. Adami, R. K. Belew, H. Kitano, and C. Taylor, editors,
*ALIFE VI, Proceedings of the Sixth International Conference
on Artificial Life*, pages 18–27. ALIFE, The MIT Press, 1998.
- 2 Lionel Barnett.
Netraveling – optimal evolutionary search with neutral
networks.
In *Proceedings of the 2001 Congress on Evolutionary
Computation CEC2001*, pages 30–37. COEX, World Trade
Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30
2001. IEEE Press.
- 2 U. Bastolla, M. Porto, H. E. Roman, and M. Vendruscolo.
Statistical properties of neutral evolution.
Journal Molecular Evolution, 57(S) :103–119, August, 2003.
Sébastien Verel, Loonard Vannest | Fitness landscapes and prob. games

- 2 E. Iquiedo-Torres.
The role of nearly neutral mutations in the evolution of
dynamical neural networks.
In J. Pollack and al, editors, *Ninth International Conference of
the Simulation and Synthesis of Living Systems (Alife 9)*, pages
322–327. MIT Press, 2004.
- 2 T. Jones.
Evolutionary Algorithms, Fitness Landscapes and Search.
PhD thesis, University of New Mexico, Albuquerque, 1995.
- 2 S. A. Kauffman.
The Origins of Order.
Oxford University Press, New York, 1993.
- 2 M. Kimura.
The Neutral Theory of Molecular Evolution.
Sébastien Verel, Loonard Vannest | Fitness landscapes and prob. games

- 2 Meriem Balabani and Jin-Kao Hao.
An analysis of the configuration space of the maximal
constraint satisfaction problem.
In *PPSN VI : Proceedings of the 6th International Conference
on Parallel Problem Solving from Nature*, pages 49–58,
London, UK, 2000. Springer-Verlag.
- 2 P. Collard, M. Clergie, and M. Defoin Platel.
Synthetic neutrality for artificial evolution.
In *Artificial Evolution : Fourth European Conference AE'00*,
pages 294–296. Springer-Verlag, 2000.
Selected papers in Lecture Notes in Computer Sciences 1829.
- 2 J. C. Culiherson.
Mutation-crossover isomorphisms and the construction of
discrimination function.
Evolutionary Computation, 2 :279–311, 1994.
Sébastien Verel, Loonard Vannest | Fitness landscapes and prob. games

- 2 Peter F. Stadler.
Landscapes and their correlation functions.
J. Math. Chem., 20 :1-45, 1996.
- 2 Peter F. Stadler and W. Schmalz.
The landscape of the travelling salesman problem.
Phys. Letters, A(161) :337-344, 1992.
- 2 Peter F. Stadler and Gunter P. Wagner.
Algebraic theory of recombination spaces.
Evolutionary Computation, 5(3) :241-275, 1997.
- 2 Terry Stewart.
Extrema selection : Accelerated evolution on neutral networks.
In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 25-29. COEX, World Trade Center, 159 Samsong-dong, Gangnam-gu, Seoul, Korea, 27-30 May 2001. IEEE Press.

- 2 Cambridge University Press, Cambridge, UK, 1983.
- 2 J. Lobo, J. H. Miller, and W. Fontana.
Neutrality in technology landscapes, 2004.
- 2 M. Newman and R. Engelhardt.
Effect of neutral selection on the evolution of molecular species.
In *Proc. R. Soc. London B.*, volume 256, pages 1333-1338, 1999.
- 2 Erik Van Nimwegen, James P. Crutchfield, and Martijn Huinen.
Metastable evolutionary dynamics : Crossing fitness barriers or escaping via neutral paths ?
Technical Report 99-07-04.1, SameFe institute, 1999.
- 2 M. Delfino Platani.

- 2 Vesselin K. Vassilev and Julian F. Miller.
The advantages of landscape neutrality in digital circuit evolution.
In *ICES*, pages 252-263, 2000.
- 2 Sebastien Verel, Philippe Collard, and Manuel Clergue.
Measuring the evolvability landscape to study neutrality.
In M. Keizer and et al., editors, *Poster at Genetic and Evolutionary Computation - GECCO-2006*, pages 613-614, Seattle, 8-12 July 2006. ACM Press.
- 2 E. D. Weinberger.
Correlated and uncorrelated fitness landscapes and how to tell the difference.
In *Biological Cybernetics*, pages 63 :325-336, 1990.
- 2 S. Wright.

- 2 *Homologie en Programmation Génétique - Application à la résolution d'un problème inverse*.
PHD thesis, Université de Nice Sophia Antipolis, France, 2004.
- 2 Eduardo Rodriguez-Tello, Jin-Kao Hao, and Jose Torres-Jimenez.
A new evaluation function for the minia problem.
In *Proceedings of the MIC 2005*, pages 796-801, Vienna Austria, 2005.
- 2 Helge Rosé, Werner Ebeling, and Torsten Asselmeier.
The density of states - a measure of the difficulty of optimisation problems.
In *Parallel Problem Solving from Nature*, pages 200-217, 1996.
- 2 P. Schuster, W. Fontana, P. F. Stadler, and I. L. Hofacker.
From sequences to shapes and back : a case study in RNA secondary structures.
In *Proc. R. Soc. London B.*, volume 265, pages 279-284, 1994.

Difficulty of problems in GP

- The ability of GP to find *optimal* solutions.

By analogy with GA hardness theory:

- **Performance:** the proportion of executions for which the *global optimum* has been found in less than **500** generations over **100** runs.

Fitness Distance Correlation (fdc) [T. Jones, 1995]

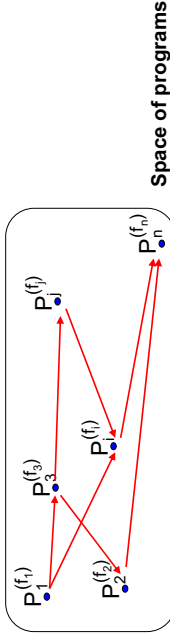
Given a sample of n individuals, let's suppose to know:

- the set $F = \{f_1, f_2, \dots, f_n\}$ of the individual fitnesses
- the genotype of the global optimum (individual with the best fitness)
- a measure to express the genotypic distance between individuals

Let $D = \{d_1, d_2, \dots, d_n\}$ be the n distances to the global optimum, then

fdc is the **correlation between sets F and D**

Main idea



- Notion of **distance**.
- Relationship between **fitness** and **distance to the goal**.

Normal text
The roles of mutation, inbreeding, crossbreeding, and selection in evolution.
In *Proceedings of the Sixth International Congress of Genetics* 7, pages 366–366, 1932.

Fitness Landscapes and Problem Difficulty in Genetic Programming

Leonardo Vanneschi

Dipartimento di Informatica, Sistemistica e Comunicazione (DISCo)
University of Milano-Bicocca, Italy



Audentes fortuna iuvat

A language to code individuals

Inspired by B. Punch (The Royal Tree Problem)

$$F = \{A, B, C, D, \dots\}$$
$$T = \{X\}$$

where:

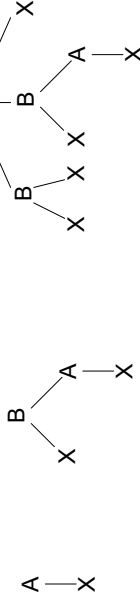
A is a function of arity 1

B is a function of arity 2

C is a function of arity 3, ...

e X is a terminal symbol

Examples



1. Tree Distance (first attempt)

A distance based on *depth* and *recursion* has been defined. It has been called *Depth Recursive Distance (drd)*.

$drd(T_1, T_2)$:

- If the root of T_1 is different from the root of T_2 , the distance is proportional to the difference of their depths
- If the root of T_1 is equal to the root of T_2 and the depth of T_1 or the depth of T_2 are different from zero, then *drd* is applied recursively to the children of the root of the two trees, and all these results are summed.
- If the root of T_1 is equal to the root of T_2 and the depth of T_1 and the depth of T_2 are both equal to zero, then the result returned is zero.

fdc as tool for problem hardness [T. Jones, 1995]

For GAs, problems can be classified in three classes:

- **Misleading** ($fdc \geq 0.15$) in which fitness increases with distance.
- **Difficult** ($-0.15 < fdc < 0.15$) in which there is no correlation between fitness and distance.
- **Straightforward** ($fdc \leq -0.15$) in which fitness increases as the global optimum approaches.

To (experimentally) verify if the same property is also valid for GP:

First step: to choose a distance between genotypes (trees!)

Our approach

- To chose a distance between genotypes to calculate *fdc*
- To define some genetic operators **consistent** with this distance
- To test *fdc* on a set of "**synthetic**" functions:
 - **Trap Functions** (unimodal et multimodal) (*Deb, Goldberg*)
 - **Royal Trees** (*B. Punch*)
 - **MAX Functions** (*Gathercole*)

The k constant

- k is an arbitrary positive constant
- two trees with different roots but with the same depth must have a positive distance
- it must never become negative with the recursive calls



it must be at least as large as the maximum depth allowed for the trees

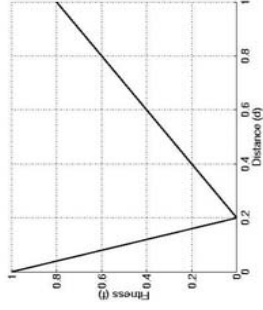
Unimodal Trap Functions

f : distance \rightarrow fitness

$$f(d) = \begin{cases} 1 - \frac{d}{B} & \text{if } d \leq B \\ R \frac{(d-B)}{1-B} & \text{otherwise} \end{cases}$$

where:

- d is the distance between the current individual and the optimum
- B et R are constants $\in [0, 1]$



B = 0.2, R = 0.8

The difficulty of trap functions depends on B and R

Genotypic Distance between Trees A first attempt: Depth Recursive Distance (drd)

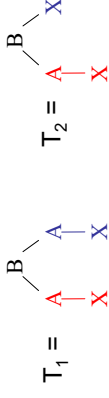
Let T_1 and T_2 be two trees.

$$\text{drd}(T_1, T_2, k) = \begin{cases} k + | \text{td}(T_1) - \text{td}(T_2) | & \text{if } \text{root}(T_1) \neq \text{root}(T_2) \\ 0 & \text{if } \text{td}(T_1) = \text{td}(T_2) = 0 \\ \sum_{i=1}^{n(T_1)} \frac{\text{drd}(s_i(T_1), s_i(T_2), k-1)}{n(T_1)} & \text{otherwise} \end{cases}$$

Annotations: root of T_1 , depth of T_1 , i^{th} subtree of T_1 root, number of sons of T_1 root.

Example of application of distance drd

Let:



Since they have the same root:

$$\begin{aligned} \text{drd}(T_1, T_2, k) &= \frac{\text{drd}(B, A, k-1)}{2} + \frac{\text{drd}(X, X, k-1)}{2} \\ &= 0 + \frac{k-1 + | \text{depth}(\frac{B}{X}) - \text{depth}(X) |}{2} \\ &= \frac{k-1 + |1-0|}{2} = \frac{k}{2} \end{aligned}$$

Royal Trees: definition

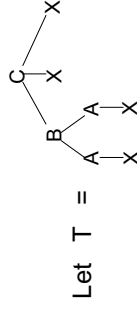
The language is the same, but we change the way of calculating fitness:

- The fitness of a tree is the score of its root
- The score of each node is the sum of the weighted scores of his sons
- If a son is a perfect tree of the appropriate level (i.e. for instance a perfect tree of root C under a node D) the scores of its subtrees, multiplied by a *FullBonus*, is added to the score of the root.
- If the son has the correct root, but it isn't a perfect tree, then the weight is *PartialBonus*.
- If the root of the subtree is incorrect, then the weight is *Penalty*.
- After having calculated the score of the root, if the tree whole is a perfect tree, the score is multiplied by a *CompleteBonus*.

In our experiments we have used the following values:

FullBonus = 2, *PartialBonus* = 1, *Penalty* = 0.0001, *CompleteBonus* = 2

Royal Trees: an example



Let $T =$

We fix:

$$\text{fitness}(X) = 1$$

Then we have:

$$\text{fitness}\left(\begin{matrix} A \\ \diagup \quad \diagdown \\ X \quad X \end{matrix}\right) = (1 * \text{FullBonus}) * \text{CompleteBonus} = (1 * 2) * 2 = 4$$

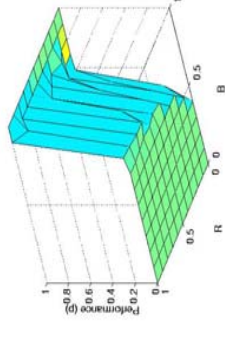
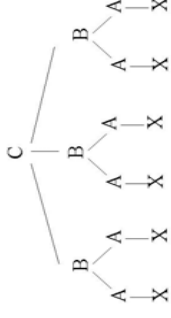
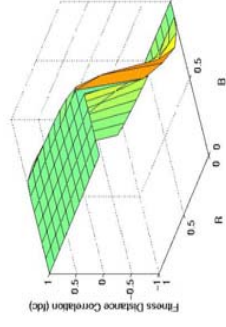
$$\begin{aligned} \text{fitness}\left(\begin{matrix} A \\ \diagup \quad \diagdown \\ \begin{matrix} A \\ \diagup \quad \diagdown \\ X \quad X \end{matrix} \quad X \end{matrix}\right) &= ((4 * \text{FullBonus}) + (4 * \text{FullBonus})) * \text{CompleteBonus} = \\ &= ((4 * 2) + (4 * 2)) * 2 = 16 * 2 = 32 \end{aligned}$$

Finally:

$$\text{fitness}(T) = (32 * \text{FullBonus}) + (1 * \text{Penalty}) + (1 * \text{Penalty})$$

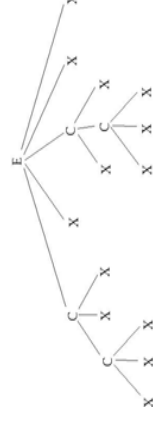
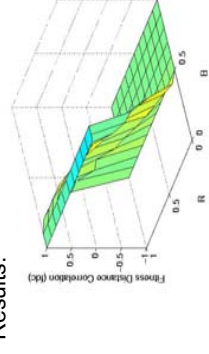
Trap Functions

Tree chosen as optimum :

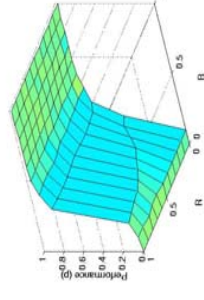


Trap Functions

Tree chosen as optimum :



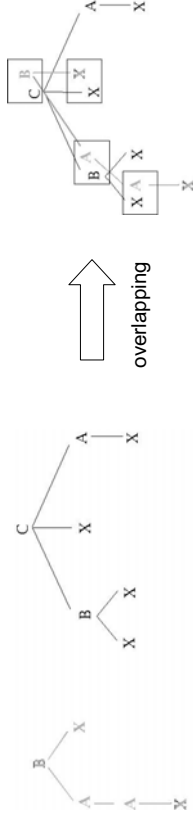
Results:



Similar results hold for different optima

Structural Distance (Intuition)

[Ekárt-Németh 2002]



- We assign a *weight* to each node
- We calculate the difference of the weights of nodes at corresponding positions
- The distance is the **weighted sum** of these differences

Structural Distance (formal definition)

$$\text{dist}(T_1, T_2) = d(R_1, R_2) + k \cdot \sum_{i=1}^m \text{dist}(\text{child}_i(R_1), \text{child}_i(R_2))$$

where:

- R_1 is the root of T_1 and R_2 is the root of T_2 .
- $d(R_1, R_2) = (|c(R_1) - c(R_2)|)^z$, where $c(X)$ is the weight of node X , et $z \in \mathbb{N}$
- $\text{child}_i(Y)$ is the i^{th} of the m possible sons of node Y if $i \leq m$, or the empty tree otherwise.
- $k \in \mathbb{N}$

Results

Racine	fdc	prediction	p	p ₁	p ₂	p ₃
B	-0.45	straightf.	1	1	1	1
C	-0.33	straightf.	1	1	1	1
D	-0.26	straightf.	0.77	0.81	0.81	0.81
E	-0.22	straightf.	0.42	0.62	0.74	0.79
F	0.035	unknown	0	0	0	0
G	0.26	misleading	0	0	0	0

p (respectively p₁, p₂, p₃) = number of runs in which the optimum has been found in less than 200 (respectively 300, 400, 500) generations divided by the total number of runs (100)

Results

fdc correctly predicts the difficulty of Trap Functions and Royal Trees, but:

no relationship has been observed between distance measure (fdc) and genetic operator (standard GP crossover [Koza, 1992])

- "In order for fdc to be more effective, the distance metric should be defined with regard to the actual neighborhood produced by the genetic operators, so to assure the conservation of the genetic material between neighbors" [Jones, 1995]
- "...either Hamming distance is connected to the way GAs work, or this relation exists in a fortuitous way among the test set chosen by Jones" [Altenberg, 1997]

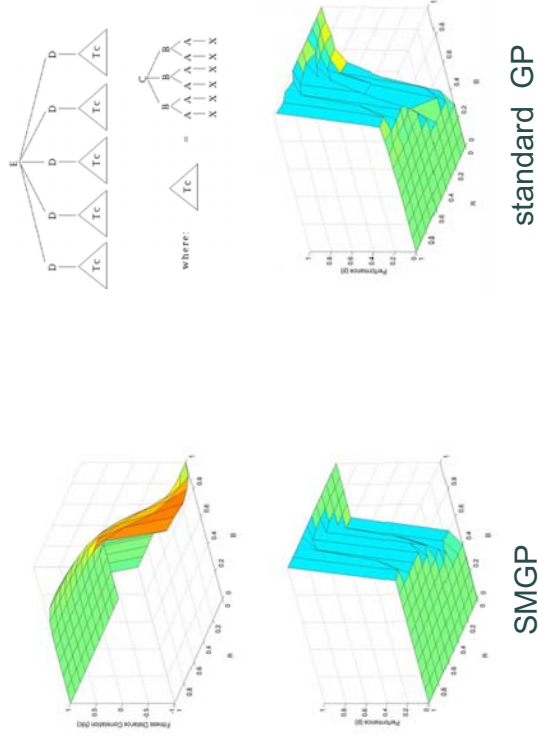
Next Step

1. To choose another distance between genotypes to calculate fdc
2. To define some genetic operators **consistent** with this distance

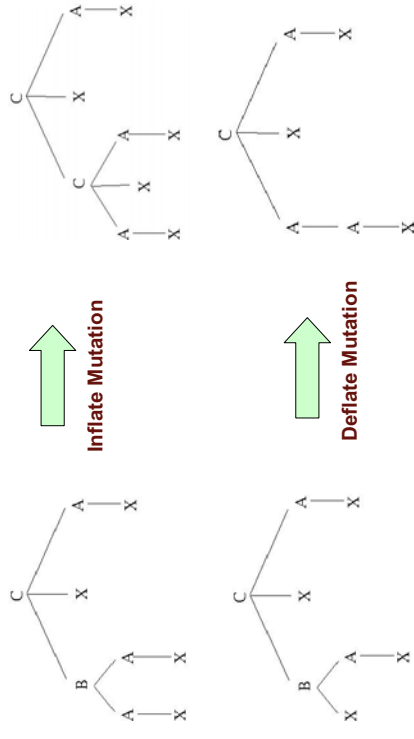
Parameters used in the experiments

- Population size = 200 individuals
- Ramped Half and a Half initialization (Koza)
- Tournament selection (tournament size = 10 individuals)
- Node with maximum arity = F
- If we use standard GP:
 - Crossover rate: 95%
 - Mutation rate: 0%
- If we use SMGP:
 - Crossover rate: 0%
 - Mutation rate: 95%
- Stop criterion: either the optimum has been found, or 500 generations have been executed.
- 100 independent runs.
- For the calculus of fdc: sampling of 40000 individuals without repetitions.

Results (trap functions 1)



Operators of Structural Mutation



GP based only on these operators:
Structural Mutation Genetic Programming (SMGP).

Property (Distance/Operators Consistency)

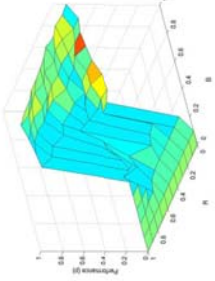
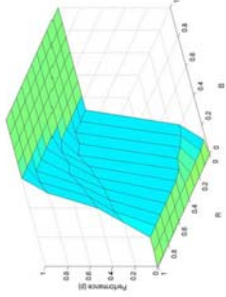
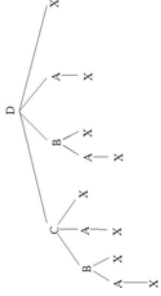
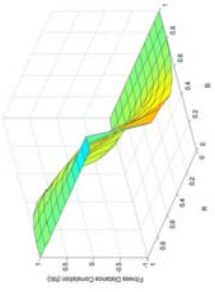
Let:

- $F = \{A, B, C, \dots\}$ $T = \{X\}$
- $\forall s$ t.q. $s \in \{F \cup T\} : c(s) = \text{arity}(s) + 1$
- T_1 et T_2 two trees composed by symbols $\in \{F \cup T\}$
- $k = 1, z = 1$

$$\text{If } \underline{T_1}, T_2 = D$$

then
 T_2 can be obtained from T_1 with a sequence of $D/2$ operations of structural mutation

Results (trap functions 4)



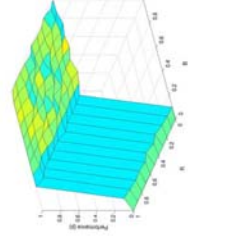
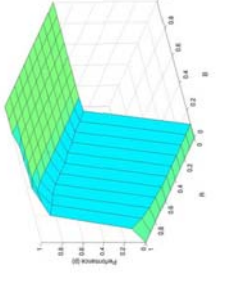
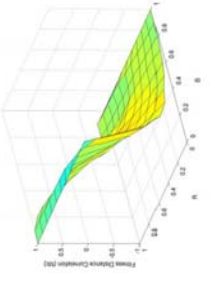
SMGP

standard GP

Leonardo Vanneschi - University of Milano-Bicocca, Italy

26

Results (trap functions 2)



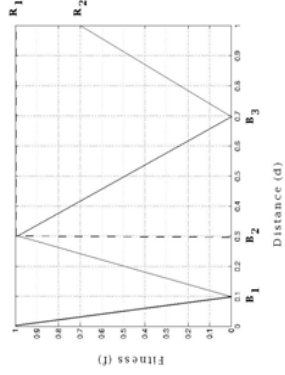
SMGP

standard GP

Leonardo Vanneschi - University of Milano-Bicocca, Italy

24

"W" Trap Functions (Multimodals)

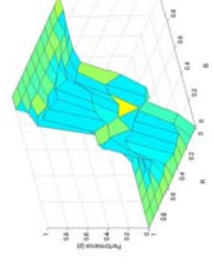
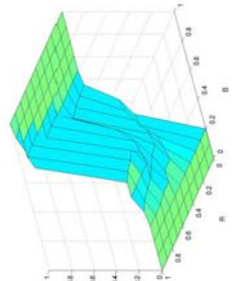
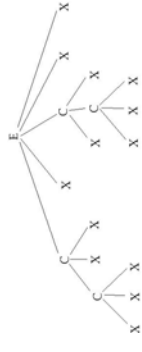
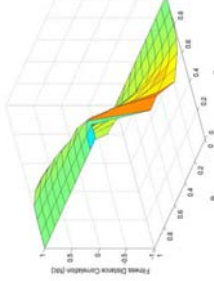


- We chose a tree T_o (origin);
- We set $R_1 = 1$;
- Between the trees with distance = B_2 to T_o we chose one tree (T_1);
- $\forall T$ s.t. $\text{dist}(T, T_o) = B_2$ et $T \neq T_1$, $\text{fitness}(T) = \text{rand}() \in [0, 1)$;
(only two *global optima*)
- To calculate fdc , $\forall T \in \text{sampling}$, we consider $\min(\text{dist}(T, T_o), \text{dist}(T, T_1))$

Leonardo Vanneschi - University of Milano-Bicocca, Italy

27

Results (trap functions 3)



SMGP

standard GP

Leonardo Vanneschi - University of Milano-Bicocca, Italy

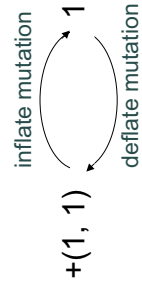
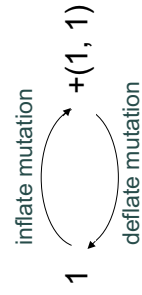
25

Royal Trees Results

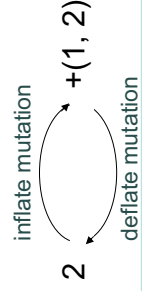
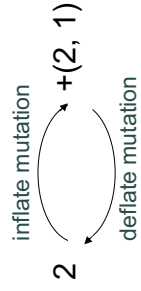
root	fdc	fdc prediction	p (SMGP)	p (stGP)
B	-0.31	straightf.	1	1
C	-0.25	straightf.	1	1
D	-0.20	straightf.	0.76	0.70
E	0.059	unknown	0	0.12
F	0.44	misleading	0	0
G	0.73	misleading	0	0

"MAX" Functions

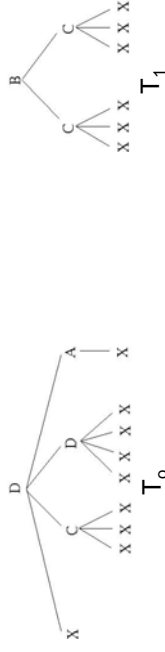
$F = \{+\}, T = \{1\}$
 $c(1) = 1, c(+) = 2$



$F = \{+\}, T = \{1, 2\}$
 $c(1) = 1, c(2) = 2, c(+) = 3$



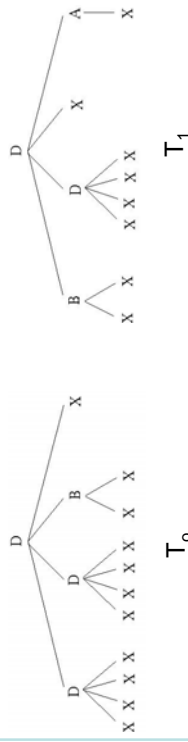
Results ("W" Trap Functions 1)



$\text{dist}(T_0, T_1) = 0.5$ (normalised)

	fdc	fdc prediction	p (SMGP)	p (stGP)
$B_f = 0.4, B_{\frac{1}{3}} = 0.9, R_{\frac{1}{2}} = 0.5$	-0.62	straightf.	0.75	0.81
$B_f = 0.5, B_{\frac{1}{3}} = 0.8, R_{\frac{1}{2}} = 0.4$	-0.88	straightf.	0.98	0.94
$B_f = 0.3, B_{\frac{1}{3}} = 0.9, R_{\frac{1}{2}} = 0.7$	-0.61	straightf.	0.80	0.77
$B_f = 0.2, B_{\frac{1}{3}} = 0.9, R_{\frac{1}{2}} = 0.1$	-0.69	straightf.	0.72	0.91
$B_f = 0.1, B_{\frac{1}{3}} = 0.9, R_{\frac{1}{2}} = 0.3$	-0.72	straightf.	0.85	0.98
$B_f = 0.5, B_{\frac{1}{3}} = 0.6, R_{\frac{1}{2}} = 0.9$	0.34	misleading.	0.33	0.20
$B_f = 0.4, B_{\frac{1}{3}} = 0.6, R_{\frac{1}{2}} = 0.9$	0.36	misleading.	0.14	0.30
$B_f = 0.3, B_{\frac{1}{3}} = 0.6, R_{\frac{1}{2}} = 0.9$	0.33	misleading.	0.31	0.13

Results ("W" Trap Functions 2)



$\text{dist}(T_0, T_1) = 0.1$ (normalised)

	fdc	fdc prediction	p (SMGP)	p (stGP)
$B_f = 0.1, B_{\frac{1}{3}} = 0.1, R_{\frac{1}{2}} = 0.1$	-0.76	straightf.	0.91	0.88
$B_f = 0.1, B_{\frac{1}{3}} = 0.9, R_{\frac{1}{2}} = 0.1$	-0.93	straightf.	0.97	0.99
$B_f = 0.05, B_{\frac{1}{3}} = 0.7, R_{\frac{1}{2}} = 0.1$	-0.90	straightf.	0.98	0.92
$B_f = 0.1, B_{\frac{1}{3}} = 0.8, R_{\frac{1}{2}} = 0.5$	-0.81	straightf.	0.90	0.83
$B_f = 0.1, B_{\frac{1}{3}} = 0.5, R_{\frac{1}{2}} = 0.1$	-0.71	straightf.	0.91	0.90
$B_f = 0.05, B_{\frac{1}{3}} = 0.2, R_{\frac{1}{2}} = 0.9$	0.89	misleading.	0.02	0.13
$B_f = 0.05, B_{\frac{1}{3}} = 0.4, R_{\frac{1}{2}} = 0.9$	0.74	misleading.	0.35	0.17
$B_f = 0.05, B_{\frac{1}{3}} = 0.1, R_{\frac{1}{2}} = 0.6$	0.78	misleading.	0.25	0.23

fdc drawbacks

- Existence of counterexamples
Here: Ridged Royal Trees
(inspired by the counterexample for GAs of [Quick et al., 1998])
- Not a predictive measure
Optima must be known "a priori"
(this drawback makes *fdc* "almost" unusable in practical cases)

A new measure is needed to quantify the difficulty of "real" problems.

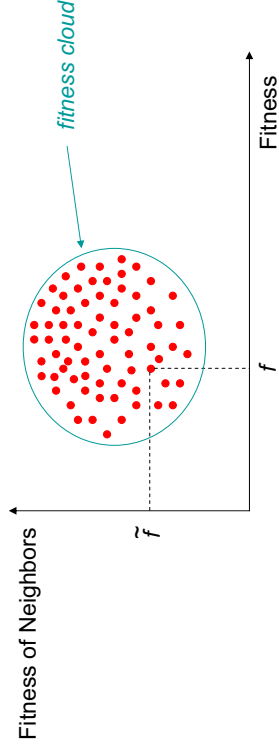
The measure proposed in this thesis is based on the concept of **fitness clouds**.

Fitness Clouds

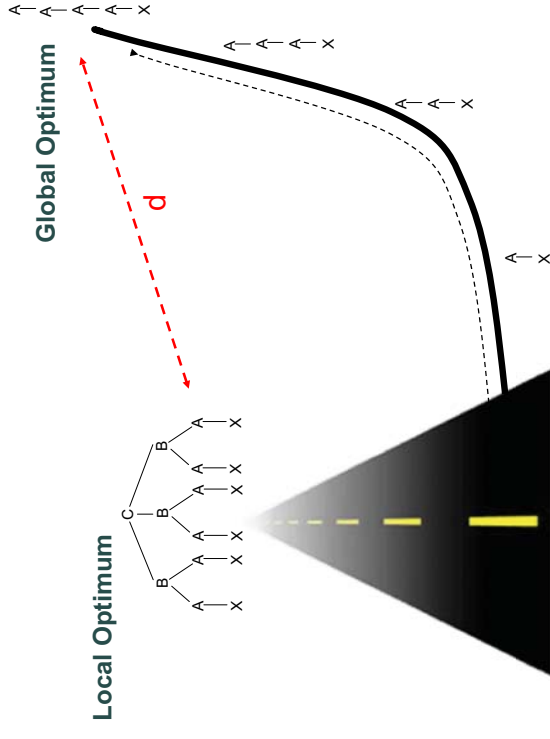
fitness clouds have been first introduced for binary landscapes in [Vérel, et al. 2003]

For each individual γ (in a sample or in the whole search space) a point is plotted:

- **abscissa** = fitness value of γ
- **ordinate** = fitness value of a "particular" neighbor (chosen randomly or by some particular technique).
- **here: neighbor = individual obtained by applying one step of mutation to γ**



Ridge Function (intuition 2)



Ridge Function Results

- Maximum depth for the trees = 10
- **fdc = 0.88**
- Over 100 runs, the optimum has been found 100 times !!!
(**performance = 1**)

1. Sampling the search space

- Individuals with *bad* fitness are likely to be discarded by selection
- It is on *good* individuals that *evolvability* has to be calculated thus, we have used an *importance sampling* (more weight to *good* points)

a sample of GP individuals $\{\gamma_1, \gamma_2, \dots, \gamma_n\}$ can be calculated by the algorithm:

```
begin
   $\gamma_1$  is generated uniformly at random;
  for  $k := 2$  to  $n$  do
    1. an individual  $\delta$  is generated uniformly at random;
    2. a random number  $u$  is generated from a
       uniform  $(0,1)$  distribution;
    3. if  $(u \leq \alpha(f(\gamma_{k-1}), f(\delta)))$ 
       then  $\gamma_k := \delta$ 
       else goto 1.
    endif
    4.  $k := k + 1$ ;
  endfor
end
```

Metropolis-Hastings

where:

$$\alpha(x,y) = \min\left\{1, \frac{y}{x}\right\}$$

2. Sampling the neighborhood

Offspring with bad fitness are probably discarded by selection

thus, the *selection algorithm* itself can be employed to choose which neighbors are to be used

In our experiments, we have used **tournament selection of size 10**, i.e.

- for each neighbor that has to be chosen:
- 10 random neighbors are generated
 - the best is chosen

Utility of Fitness Clouds

They give a visual rendering of **evolvability**

Evolvability

"The ability of an operator/representation scheme to produce offspring that are fitter than their parents"

[Altenberg, *The evolution of evolvability in genetic programming*, 1994]

Evolvability is related to (although not identical to) problem difficulty.

Sampling the search space and the neighborhoods

Main idea:

Evolvability makes sense if it is calculated on "good" individuals ("bad" ones are probably discarded by selection).

Sampling the search space:

Importance sampling (Metropolis-Hastings technique)

Sampling the neighborhoods:

selection (tournament selection of size 10).

Performance

The fraction of executions for which a global optimum has been found before generation 500 over 100 runs

Parameters

Fitness clouds

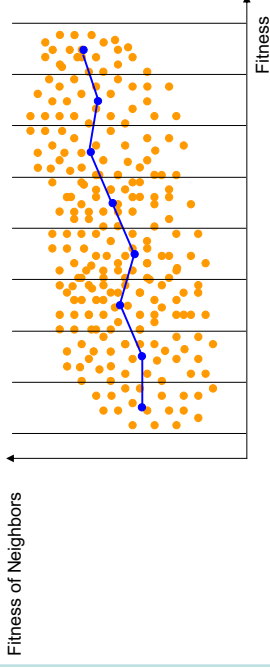
- 40000 points with Metropolis-Hastings
- **Standard subtree mutation** to generate neighborhoods
- For each sampled individual, 1 neighbor chosen by means of **Tournament selection of size 10**

GP runs

- Generational GP
- Population size = 200 individuals
- Ramped half-and-half initialization
- **Tournament selection of size 10**
- **Standard subtree mutation** rate = 95%
- No crossover

Measure of Problem Hardness Based on Fitness Clouds Negative Slope Coefficient (*nsc*)

- The scatterplot is partitioned into n bins
- For each bin, a point is calculated, such that its abscissa is the average of the abscissas and its ordinate is the average of the ordinates.
- All these points are joined by segments $\{S_1, S_2, \dots, S_{n-1}\}$



Binomial-3 Problem [Daida et al. 2001]

Goal: to find a program which matches the following function

$$f(x) = 1 + 3x + 3x^2 + x^3$$

- **Fitness cases** (values for x) = 50 equidistant points over the range $[-1, 0)$
- **Function set** = $\{+, -, *, //\}$ where $//$ is the protected division (it returns 1 if the denominator is 0)
- **Terminal set** = $\{x, R\}$

where R is a set of *Ephemeral Random Constants (ERCs)* uniformly distributed over the range $[-a_R, a_R]$, where a_R is a positive integer constant

Daida et al. have shown that the difficulty of this problem increases as a_R increases

Negative Slope Coefficient Definition

$$nsc = \sum_{i=1}^{n-1} p_i$$

where, $\forall i \in [1, n-1]$

$$p_i = \min \{0, \text{slope}(S_i)\}$$

Hypothesis:

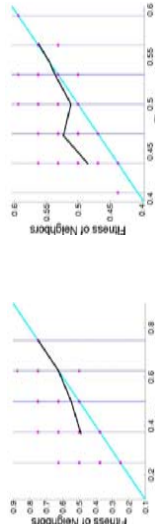
- $nsc = 0 \Rightarrow$ the problem is **easy**
- $nsc < 0 \Rightarrow$ the problem is **difficult** and the magnitude of nsc quantifies the difficulty

Idea:

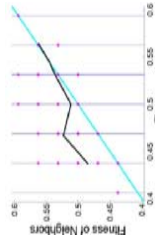
If $nsc < 0$ then there is at least one area of the fitness landscape where evolvability is bad.

Even Parity Problem [Koza, 1992]

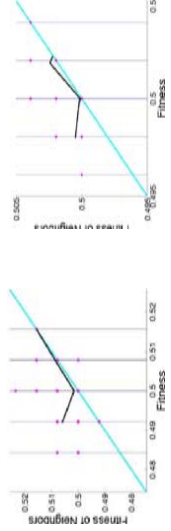
- Primitive sets used: $F = \{\text{NAND, NOR}\}$, $T = \{x_1, x_2, \dots, x_k\}$ where k is the rank of the problem (even parity k problem)
- The difficulty of the problem can be changed by varying k (even parity 5 more difficult than even parity 3...)



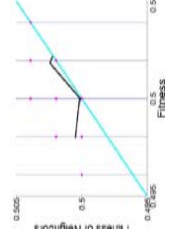
even parity 3



even parity 5



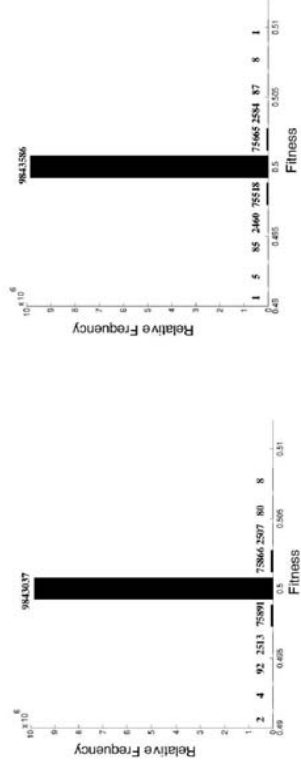
even parity 7



even parity 9

problem	p	nsc	ffc
even parity 3	0.98	0	0.57
even parity 5	0.01	-0.23	0.43
even parity 7	0	-0.42	0.30
even parity 9	0	-1.08	0.25

Even Parity 9 - Fitness Distribution

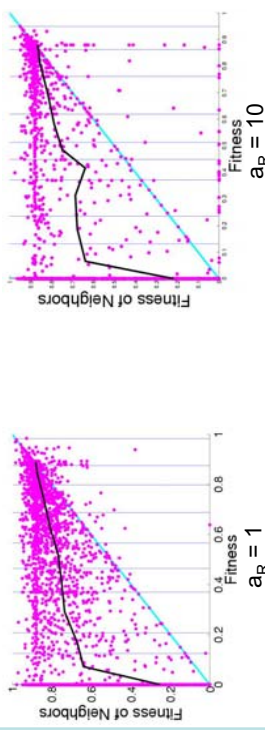


Metropolis-Hastings sampling

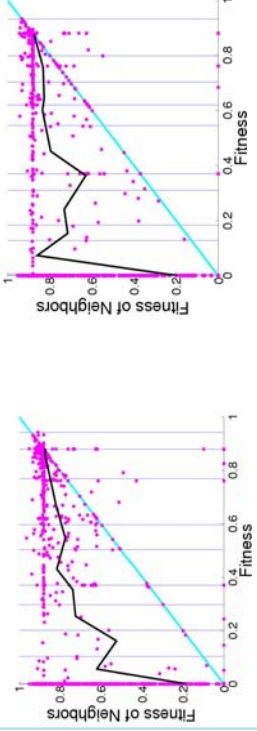
Uniform Random Sampling

- The shape of the scatterplot does not depend on the choice of the Metropolis-Hastings sampling
- These results are coherent with the ones contained in [Langdon and Poli, *Foundations of Genetic Programming*, 2002]

Binomial-3 Problem (experimental results 1)



$a_R = 1$



$a_R = 1000$

Binomial-3 Problem (experimental results 2)

a_R	p	nsc	ffc
1	0.89	0	0.68
10	0.42	-0.15	0.64
100	0.35	-0.98	0.65
1000	0.29	-2.03	0.65

- nsc gets more negative as the problem gets more difficult
- nsc is zero for the "easy" instance of the problem ($a_R = 1$)
- for comparison: ffc is the fitness-fitness correlation [Manderick et al., 1991] clearly it does not give any indication about problem difficulty

Crossover Distance

Other Operator Based Distances

For instance: Structural Mutation is consistent with Structural Distance⁽¹⁾

Formal Proof in:

[L. Vanneschi et al., *Fitness Distance Correlation in Structural Mutation Genetic Programming*, EUROGP 2003]

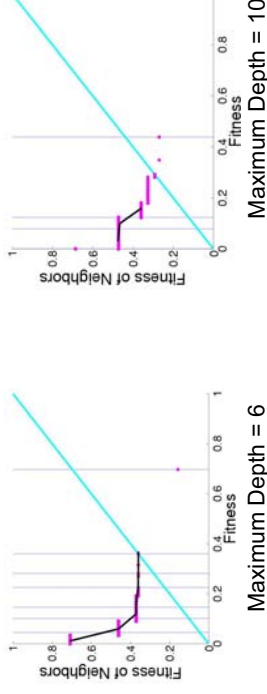
⁽¹⁾ A. Ekárt, S. Z. Németh, *Maintaining the Diversity of Genetic Programs*, EUROGP 2002

Subtree Crossover Distance is more difficult

- subtree crossover is binary,
- subtree crossover is "destructive",
- trees are dynamic structures, etc.

Artificial Ant on the Santa Fe Trail [Koza, 1992]

- Difficulty tuning is obtained by changing the maximum tree depth difficulty decreases as individuals' size increases [Langdon and Poli, *Foundations of Genetic Programming*, 2002]



max. tree depth	p	nsc	ffc
10	0.05	-3.21	-0.86
6	0	-9.84	-0.84

Summary of the results obtained with the nsc

- Good hardness indicator for:
 - Trap Functions
 - Royal Trees
 - Binomial-3 Problem [Daida et al., 2001]
 - Even Parity Problem [Koza, 1992]
 - Artificial Ant on the Santa Fe Trail [Koza, 1992]
- Many ways of calculating the nsc have been used:
 - Number of neighbors for each sampled individual
 - Number of mutations to generate neighbors
 - Different types of mutations to generate neighbors
 - Different techniques to partition the fitness clouds into bins
- nsc is predictive \Rightarrow it can be used on *any* problem
- nsc has not been normalized yet into a given range (classification of different problems by their difficulty)

The (Basic) Idea

We don't have to count *how many crossovers* it takes to transform a tree T_1 into another tree T_2 , but *how probable* it is to obtain T_2 by applying crossover to T_1 (in just one step!).

[S. Gustafson, L. Vanneschi, *Operator based distance for Genetic Programming: Subtree Crossover Distance*, EUROGP 2005]

Subtree Crossover Distance (SCD)
between two trees T_1 and T_2

=

Probability of:

- Selecting a subtree S_{T_1} from T_1 , and
- Finding a subtree S_{T_2} in the population P

Such that:

Replacing S_{T_1} with S_{T_2} in T_1 we get T_2

Why a Distance Based on Subtree Crossover?

- In Genetic Programming (GP), genetic operators define a neighborhood structure (topological space) over trees.
- To analyse various dynamics of GP (diversity, etc.), it is often useful to quantify the distance between two trees in this topological space.
- Subtree crossover is the most common genetic operator in GP → subtree crossover distance.

A difficult task:

- subtree crossover is binary,
- subtree crossover is "destructive",
- trees are dynamic structures, etc.

Terminology

- SCD is a probability!
- SCD between two trees T_1 and T_2 is a function of T_1 , T_2 and the population (P) in which T_1 and T_2 are!

Thus

SCD is NOT a distance (metric) !!

We need a *similarity / dissimilarity measure (for subtree crossover)*, not necessarily an (Euclidean) distance metric.

The term *pseudo-distance* would be more appropriate.

What do we mean by Distance Based on Subtree Crossover?

"Topological crossovers [...] are well defined once a notion of distance over the solution set is defined. Simply stated, the offspring they produce are *between their parents*"

[A. Moraglio, R. Poli, *Topological Crossover for the Permutation Representation*, GSICE 2005]

In this paper, we take up a different point view:

the distance of a solution T_1 to a solution T_2 should *not* express "betweenness" of T_1 and T_2 , but "how easy" (or "how difficult") it is to transform T_1 into T_2 (by means of a genetic operator, crossover in our case).

(If it is "easy" to transform T_1 into T_2 , the distance between T_1 and T_2 must be "small". Otherwise it must be "large").

The "New" SCD Definition

```
func SCD( $T_1, T_2, P$ ) {  
   $S = \text{diff}(T_1, T_2)$   
   $res = 1$   
   $\forall (s_{T_1}^i, s_{T_2}^i) \in S:$   
     $ps1 = \text{probSelecting}(s_{T_1}^i, T_1)$   
     $ps2 = \text{probCreating}(s_{T_2}^i, P)$   
     $res = res * (1 - ps1 * ps2)$   
   $\text{return}(res)$   
}
```

the complexity is still "reasonable"!

The new operator $\text{diff}(T_1, T_2)$ returns the set

$$S = \{(s_{T_1}^1, s_{T_2}^1), (s_{T_1}^2, s_{T_2}^2), \dots, (s_{T_1}^n, s_{T_2}^n)\}$$

such that:

if we replace $s_{T_1}^i$ with $s_{T_2}^i$ ($\forall i \in [1, n]$) in T_2 we obtain T_1

The "Old" SCD Definition (EUROGP 2005)

```
func distance( $T_1, T_2, V, P$ ) {  
   $(s_{T_1}, s_{T_2}) = T_1 / T_2$   
   $ps1 = \text{probSelecting}(s_{T_1}, T_1)$   
   $ps2 = \text{probCreating}(s_{T_2}, P)$   
   $\text{return}(1 - ps1 * ps2)$   
}
```

where: T_1 / T_2 is the "difference" between T_1 and T_2

This operator returns a pair of subtrees (s_{T_1}, s_{T_2}) such that $s_{T_2} \in T_2$ must replace $s_{T_1} \in T_1$ to make $T_1 = T_2$.

Theory meets Practice: Is The "New" SCD Definition Useful ?

We have done some experiments:

- Fitness Distance Correlation (FDC)
- Fitness Sharing
- Diversity

Test Problems:

- Syntactic Trees
- Trap Functions

Advantage of the "old" SCD definition

In [S. Gustafson, L. Vanneschi, *Operator based distance for Genetic Programming: Subtree Crossover Distance*, EUROGP 2005] we have shown that the complexity of SCD is "reasonable"

Drawback of the "old" SCD definition

If T_1 and T_2 differed in more than one subtree, we approximated the distance between them by setting it equal to one (very large).

In some cases, this approximation may be too coarse, thus compromising the accuracy of the measure (for instance when using the SCD for calculating the FDC).

Fitness-Distance Correlation (Definition)

(Jones, 1995)

Let $\{i_1, i_2, \dots, i_n\}$ be a set of n sampled individuals,

$$F = \{f_1, f_2, \dots, f_n\}$$

their n fitness values and

$$D = \{d_1, d_2, \dots, d_n\}$$

the n distances of each individual to a global optimum, we define

FDC = statistical correlation between sets F and D .

According to Jones:

- Positive FDC → very hard problem
- FDC approximately equal to zero → hard problem
- Negative FDC → easy problem

We have studied FDC for GP on large samples of individuals.

Here we study it dynamically (how it changes during the evolution) on the individuals in the population.

Syntactic Trees

(Gustafson, Vanneschi, 2005)

Trees are represented using the sets:

$$F = \{N\}, T = \{L\}$$

where

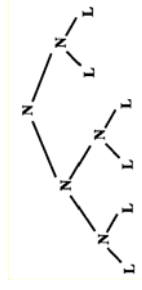
- N is a binary operator (N stands for “Non-terminal”)
- L stands for “Leaf”.

No “content” is associated with the nodes and fitness is equal to the *edit distance* (ED) to a fixed global optimum.

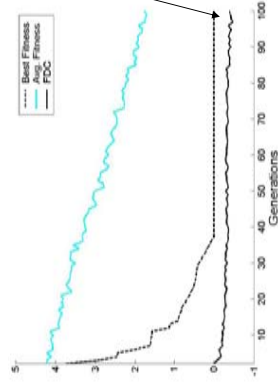
Experimental Results (FDC - Syntactic Trees)

Successful Runs

Optimal Tree =



FDC is negative and decreases during evolution



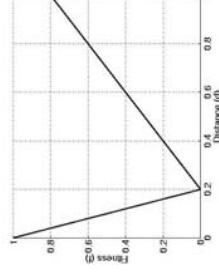
(standard deviations on the paper)

Trap Functions

(Deb, Goldberg, 1993)

$F = \{A, B, C, \dots\}$, $T = \{X\}$
 arity(A) = 1, arity(B) = 2, arity(C) = 3, ...

f : distance → fitness



$b = 0.2$, $r = 0.8$

The difficulty of trap functions depends on b and r

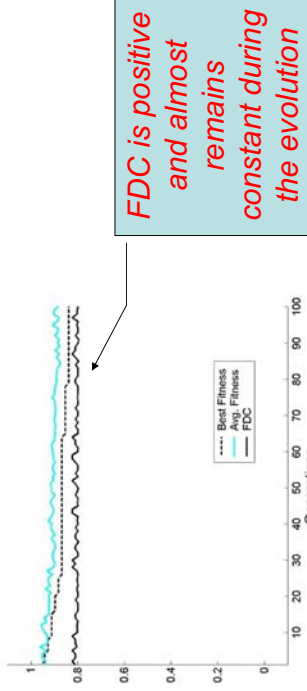
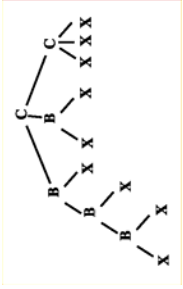
where:

- d is the distance between the current individual and the optimum
- b and r are constants $\in [0, 1]$

Experimental Results (FDC - Royal Trees)

Unsuccessful Runs

Optimal Tree =



Leonardo Vanneschi - University of Milano-Bicocca, Italy

70

FDC Results - Discussion

For *successful runs*, FDC is negative → as individuals get **better** (in fitness) they also get **closer** (in terms of SCD) to the global optimum (*until it is reached by GP!*).

For *unsuccessful runs*, FDC is positive or approximately equal to zero → as individuals get **better** (in fitness) they **don't get closer** (in terms of SCD) to the global optimum (and thus GP does not reach it!).

FDC + SCD = good indicator for GP dynamics !

SCD = a good distance to calculate the FDC !

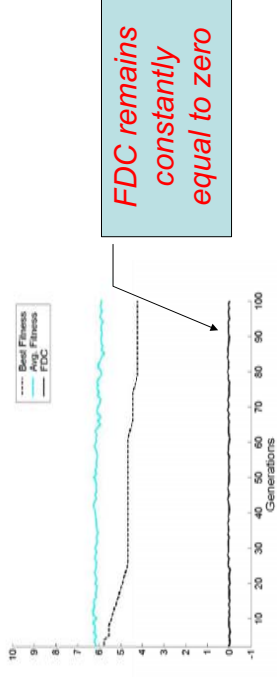
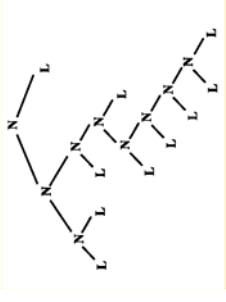
Leonardo Vanneschi - University of Milano-Bicocca, Italy

71

Experimental Results (FDC - Syntactic Trees)

Unsuccessful Runs

Optimal Tree =



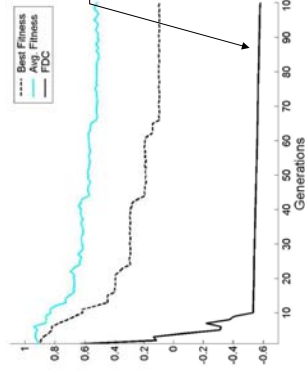
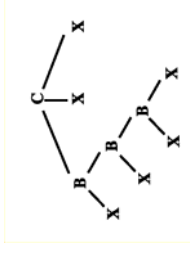
Leonardo Vanneschi - University of Milano-Bicocca, Italy

68

Experimental Results (FDC - Royal Trees)

Successful Runs

Optimal Tree =



Leonardo Vanneschi - University of Milano-Bicocca, Italy

69

Fitness Sharing Results - Discussion

GP systems with fitness sharing using SCD "perform better" than standard GP and fitness sharing systems using ED (for the problems studied here!)

Because SCD is consistent with crossover (and thus appropriately models its behavior) ?

Fitness Sharing (Goldberg, 1991)

$$f_s(i) = \frac{f(i)}{\sum_{j \in P \wedge j \neq i} s(d(i, j))}$$

$$s(x) = 1 - x$$

Test Problem: Syntactic Trees

Global Optima: randomly generated trees (with the grow method) with a maximum depth equal to 8

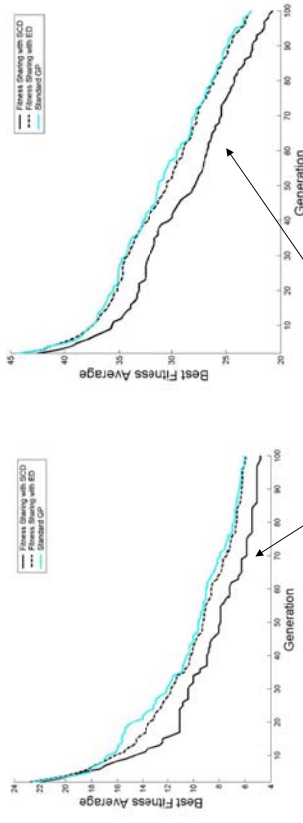
Diversity

Standard deviations of the distances of the individuals in the population from a pre-fixed tree (= "origin").

Test Problem: Syntactic Trees

Global optima = origin = randomly generated trees (with the grow initialisation method) with a maximum depth equal to 8

Experimental Results (Fitness Sharing)



Fitness sharing systems with SCD perform better than fitness sharing with ED and standard GP

Conclusions

SCD appropriate for:

- Measuring the FDC *dynamically* (during evolution)
- Fitness Sharing

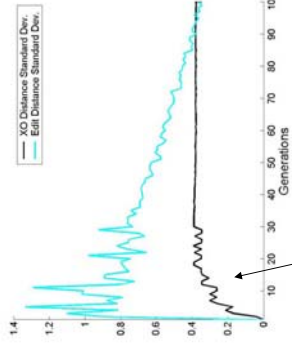
Our hypothesis: SCD appropriately models subtree crossover

- SCD diversity behave differently than ED diversity (slightly increasing and larger than zero for successful runs, approximately zero for unsuccessful runs)

Can it be used to predict the behavior of GP runs?

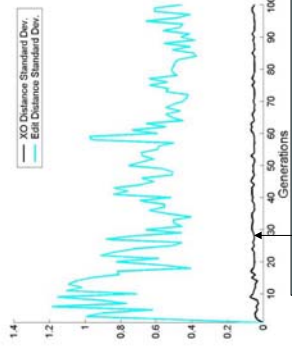
Experimental Results (Diversity)

Successful Runs



It grows !

Unsuccessful Runs



"more or less" constant and "small" !

Diversity Results - Discussion

For the problems studied here, the population is, most of the time, composed by individuals which are far (in the sense of SCD) to the global optimum.

For *successful runs*, SCD diversity grows up → some individuals which can easily be transformed into the optimum are generated.

For *unsuccessful runs*, SCD remains very small → the population is always composed by individuals which are far to the optimum. This thing does not change during evolution!

SCD distance can be used to predict GP behavior?

Neutrality

What is "new" in this work?

- We study even parity fitness landscapes for standard tree-based GP (instead of Cartesian GP or linear GP).
- We don't artificially introduce neutrality into the landscapes (in order to study its effects on GP performance), but we study them "as they are".
- We introduce some new **neutrality measures**
- We introduce new genetic operators to define neighborhood between trees.
- We don't use a fixed set of functions to build solutions, but we compare the landscapes induced by two different sets of boolean operators: {NAND} and {XOR, NOT}
- We define a new **sampling methodology**

Goals of this study

1. Defining some indicators of the difficulty of GP even parity landscapes based on the concept of neutrality
 - no "stand-alone" infallible hardness measures, but each one of them should give us the picture of one characteristic of the fitness landscape and all together they should give us a rather complete picture that should help us to draw some conclusions about problem hardness*
2. Defining a new sampling strategy for GP in general (and particularly suitable for boolean landscapes).
 - Sampling s fitness landscape is an important task independently from the hardness indicator we want to employ: if we don't sample the important parts of the landscape or if we don't give a sufficiently faithful picture of it, we will fail!*

Our Approach

1. Defining some *neutrality measures*
2. Using them to characterize two "*small*" landscapes (which we are able to generate exhaustively)

Partial goal: to test if the measures defined in point 1 are suitable indicators of fitness landscapes difficulty

3. Presenting a new *sampling technique*

4. Applying the same measures as in point 1 to two "*large*" landscapes, but "*similar*" to the ones studied at point 2 (which we study by means of the sampling technique presented in point 3).

Partial goal: to test if the sampling technique presented in point 3 gives a faithful "picture" of the landscape (at least from the point of view of neutrality) and to have a confirmation of the fact that the measures defined in point 1 are suitable indicators of fitness landscapes difficulty

GP and Neutrality - State of The Art

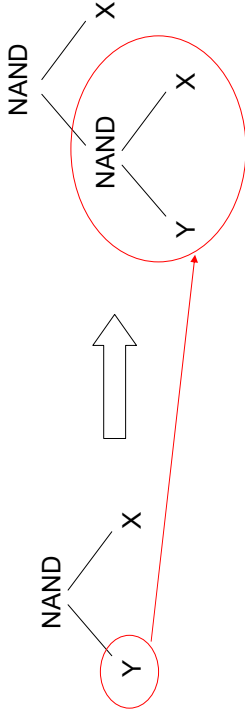
- [Yu, Miller, 2001-2002]: artificially adding neutrality can help Cartesian GP to navigate some fitness landscapes.
- [Collins, 2005]: criticism towards the previous results
- Other contributions: [Reidys, Stadler, 2001], [Toussaint, Igel, 2002], [Geard, 2002], [Collard et al., 1999], ...: neutrality plays an important role in EAs.

GP Boolean Landscapes - State of The Art

- [Langdon, Poli, 2001]
- [Koza, 1992]
- [Collins, 2004]
- ...

2. **Strict inflate mutation**: transforms leaf into a tree of depth 1, rooted in a random operator and whose children are a random list of variables containing also the original leaf in a random position

Example



3. **Point Terminal Mutation**: replaces a leaf with another random leaf.

Some Definitions

- **Fitness Landscape**: $L = (S, V, f)$, where
 - S is the search space,
 - $V: S \rightarrow 2^S$ is a neighborhood function specifying for each $s \in S$ the set of its neighbors $V(s)$
 - f is the fitness function
- **Neutral neighborhood**: $V_N(s) = \{s' \in V(s) \mid f(s') = f(s)\}$
- **Neural Network** [Schuster et al., 1994]: a graph connected component (S, E) , where: $E = \{(s, s') \in S^2 \mid s' \in V_N(s)\}$

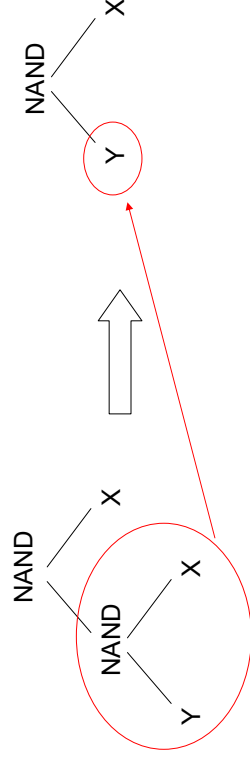
But before...

Genetic operators that induce the neighborhood

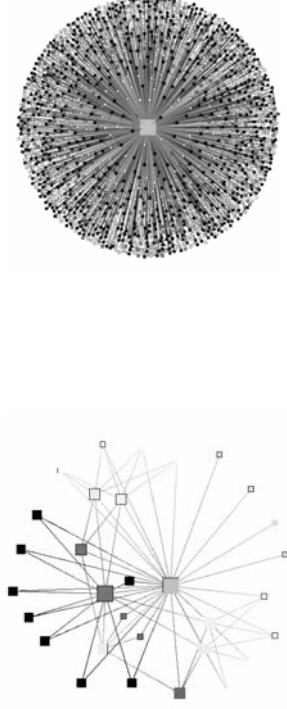
Inspired by the structural mutation operators defined in [Vanneschi et al., 2003]

1. **Strict deflate mutation**: transforms a subtree of depth 1 into a randomly selected leaf chosen among its children.

Example



Graphical Representation of the two landscapes



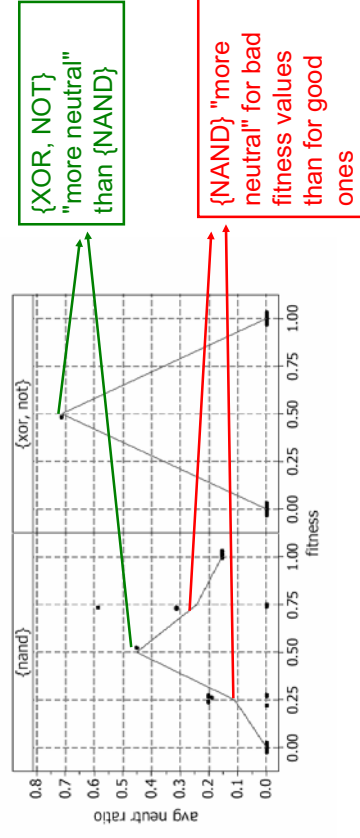
{NAND}

{XOR, NOT}

Average Neutrality Ratio

Neutrality ratio of an individual s = the number of neutral neighbors of s divided by the total number of neighbors of s ($= \#V_N(s) / V(s)$)

Average neutrality ratio of a neutral network = the average of the neutrality ratios of all the individuals belonging to that network.



"Small" Landscapes

- Even Parity 2 Problem
- Maximum Tree Depth = 3
- Sets of Operators: {NAND} and {XOR, NOT}

Some statistics:

	$\mathcal{L}_{(2,3)}^{\{\text{XOR}, \text{NOT}\}}$	$\mathcal{L}_{(2,3)}^{\{\text{NAND}\}}$
No. of individuals	5552	1446
No. of optimal solutions	660	8
No. of neutral networks	1389	31
Average network size	3.99	46.64

can be misleading, we'll see it soon...

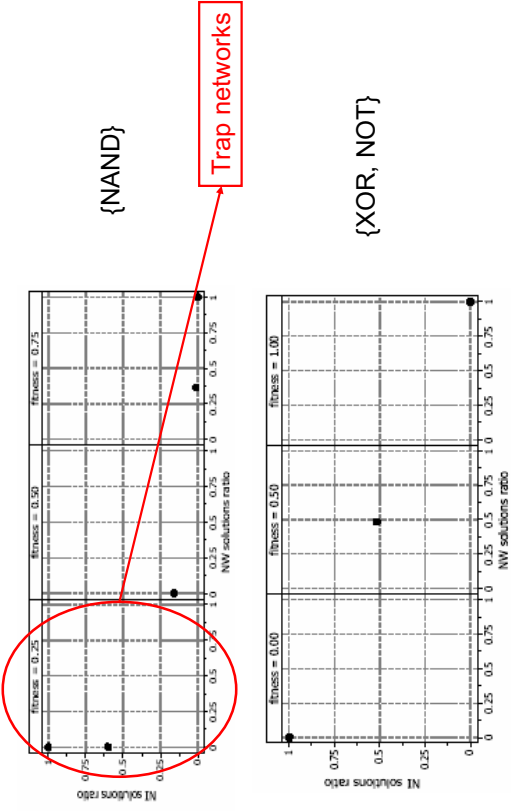
Parity Landscapes Properties

- Supposing that all fitness values have been normalized into the range $[0, 1]$, if an expression does not contain at least an occurrence of each variable, then its fitness is $= 0.5$ [Langdon, Poli, 2002]
The wide majority of individuals have fitness = 0.5 !!!!

{XOR, NOT}

- All possible fitness values are only 0, 0.5 and 1 (in our work 0 = good, 1 = bad).
- There is only one neutral network at fitness 0.5 (central network)
- All other nets are composed by one individual (peripheral networks)
- All peripheral networks are connected with the central one by just one mutation

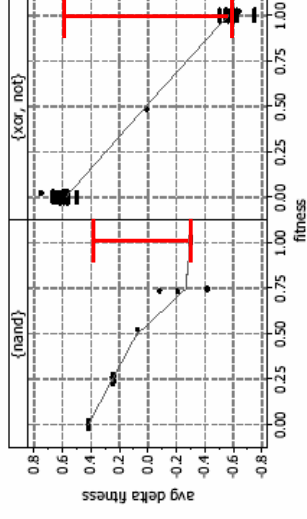
NI Solutions Ratio vs NW Solutions Ratio



Average Δ -fitness

Let N be a neutral network, then the average Δ -fitness of N is:

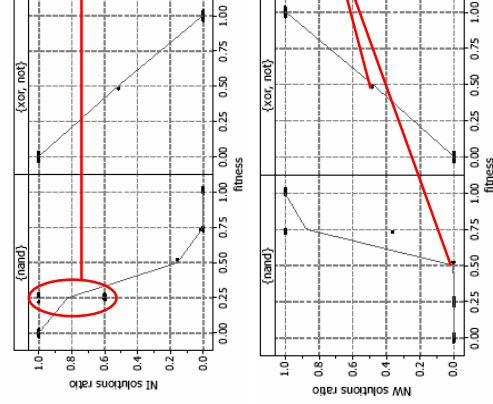
$$\Delta f(N) := \frac{1}{|N|} \cdot \sum_{s \in N} \left[\frac{\sum_{v \in \mathcal{V}(s)} (f(v) - f(s))}{|\mathcal{V}(s)|} \right]$$



Sampling Methodology

- **Modified Metropolis Sampling**
Like Metropolis, but gives a reward to individuals with different fitness values (and thus also to the ones with a different fitness than 0.5).
- **Vertical Expansion**
Takes as input the output of the Modified Metropolis sampling and enriches it with some non-neutral neighbors of its individuals.
- **Horizontal Expansion**
Takes as input the output of the Vertical Expansion phase and enriches it with as many neutral neighbors of its individuals as possible (builds neutral networks)

Non-Improvable (NI) and Non-Worsenable (NW) solutions ratio



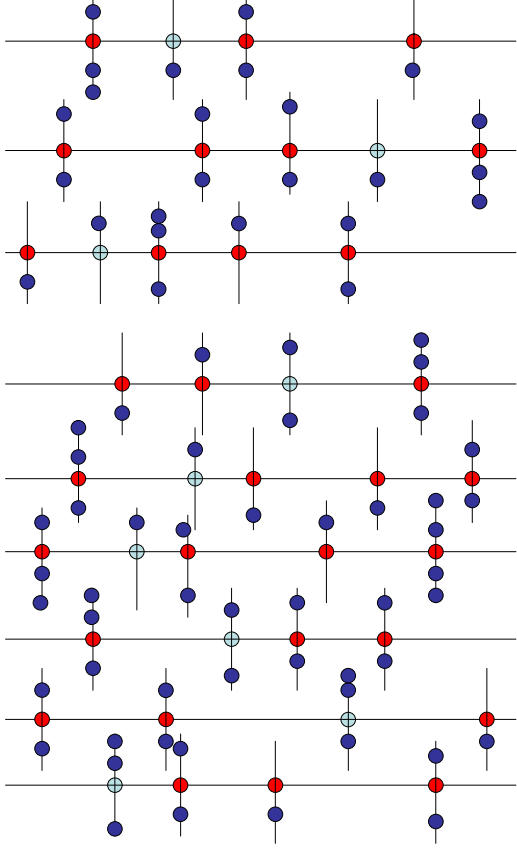
The acceptance probability

$$\alpha(f(P), f(T)) = (1 - p_m) \cdot \log_{10} \left(\frac{9 \cdot |f(P) - f(T)|}{\max(f(P), 1 - f(P)) + 1} + 1 \right) + p_m$$

- If $|f(P) - f(T)| = 0$ then small probability of accepting T ($= p_m$)
- The larger the value of $|f(P) - f(T)|$ the higher the probability of accepting T
- In particular, if T has the most different possible fitness value from P , the $|f(P) - f(T)| = \max(f(P), 1 - f(P))$. In that case, the logarithmic term becomes $\log_{10}(9+1) = \log_{10}(10) = 1$ and thus $\alpha(f(P), f(T)) = 1$.

We have chosen a logarithmic function because it increases very quickly for small differences and thus it rewards solutions T with a slightly different fitness from P .

Modified EM algorithm example!



Vertical Expansion

For each individual $i \in S$, L different neighbors of i are generated by means of L different mutations.

Each one of these neighbors is accepted or rejected according to the same probability as in the Modified Metropolis algorithm.

Remark: since p_m is "small", there is a "small" probability of having neutral neighbors in S at the end of the vertical expansion

Modified Metropolis Sampling

Generate a random individual P ;

$S = \{P\}$;

repeat

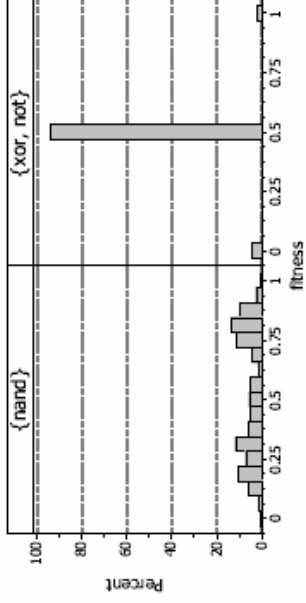
 generate a random individual T ;

$$\alpha(f(P), f(T)) = (1 - p_m) \cdot \log_{10} \left(\frac{9 \cdot |f(P) - f(T)|}{\max(f(P), 1 - f(P)) + 1} + 1 \right) + p_m$$

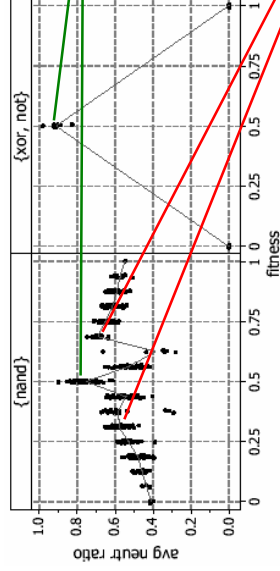
if $(\alpha(f(P), f(T)) > \text{rand}(0,1))$ **then** $P = T$; $S = S \cup \{P\}$ **endif**

until $\#S = N$ (prefixed constant)

Fitness Distributions with our Sampling Technique



Average Neutrality Ratio



{XOR, NOT}
"more neutral"
than {NAND}

{NAND} "more neutral" for bad fitness values than for good ones

Horizontal Expansion

```

iter ← 1;
while (at least one incomplete neutral network exists
in S) and (|S| < Smax) do
  N ← set of incomplete networks in S of size
  less than Imax;
  for all N' ∈ N do
    for all i ∈ N do
      for all j ∈ V(i) do
        if (rnd(0,1) < β(f(i), f(j), iter)) and
           (|S| < Smax) then
          S ← S ∪ {j};
        endif
      endforall
    endforall
  endforall
  iter ← iter + 1;
endwhile
return(S);
  
```

For each individual in the sample

We consider all its neighbors

And we accept them with probability β

where:

$$\beta(f(i), f(j), iter) = \begin{cases} 1 & \text{if } f(i) = f(j), \\ k^{-iter} & \text{otherwise} \end{cases}$$

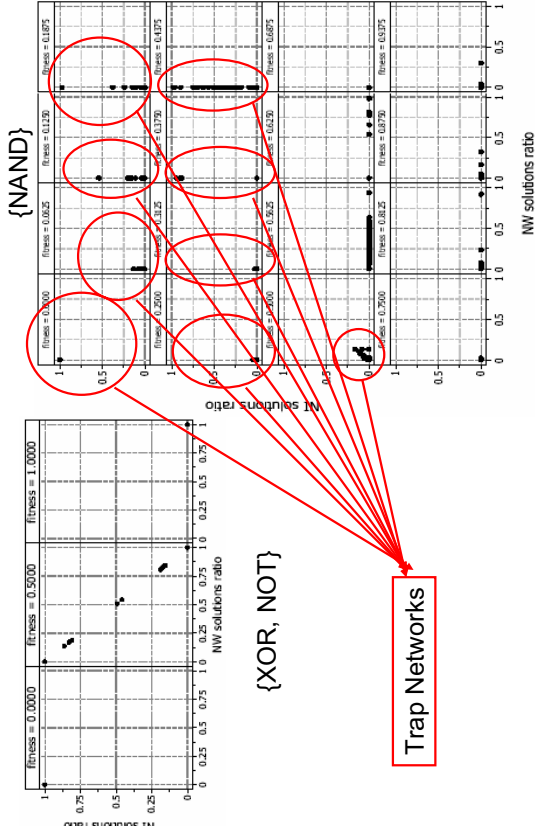
"Large" Fitness Landscapes

- Even Parity 4 problem
- Maximum Tree Depth = 8
- Sets of Operators: {NAND} and {XOR, NOT}

Some statistics:

	{XOR; NOT}	{NAND}
p_m for Modified Metropolis	0.0005	
p_m for vertical expansion	0.00005	
k for horizontal expansion	4	
Minimal size of an incomplete network	2	
Sample size of Modified Metropolis	10	3
L of vertical expansion	100	10
Size of generated sample	14589	30238
No. of networks contained into the sample	898	492

NI Solutions Ratio vs NW Solutions Ratio



Discussion

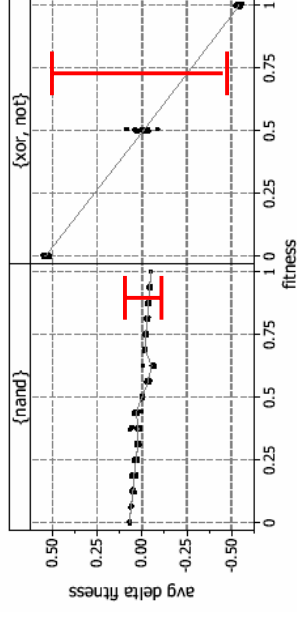
- Networks with bad fitness values are "more neutral" than networks with good fitness values for {NAND}. It is not the case for {XOR, NOT}
- It is unlikely to improve fitness mutating individuals belonging to neutral networks with good fitness for {NAND}. It is not the case for {XOR, NOT}
- Mutations induce largest fitness modifications for {XOR, NOT} than for {NAND}

Conclusions

Average Neutrality Ratio, Average Δ -fitness NI Solutions Ratio and NW solutions ratio (which are all neutrality networks measures) are good hardness indicators for these two particular landscapes

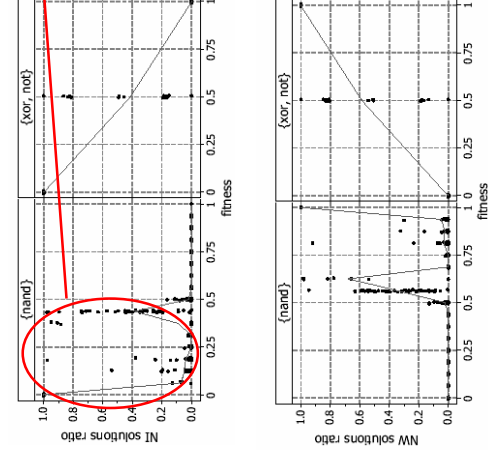
These results hold both for "small" landscapes that we have studied exhaustively and for "large" landscapes that we have studied by means of our (new!) sampling technique \rightarrow it seems a "good" sampling technique!

Average Δ -fitness



For {XOR, NOT} mutations can produce larger fitness improvements

Non-Improvable (NI) and Non-Worsenable (NW) solutions ratio



Many Trap Networks!

Thank you

vanneschi@disco.unimib.it

<http://personal.disco.unimib.it/Vanneschi>