Approximating Geometric Crossover by Semantic Backpropagation

KRZYSZTOF KRAWIEC, TOMASZ PAWLAK

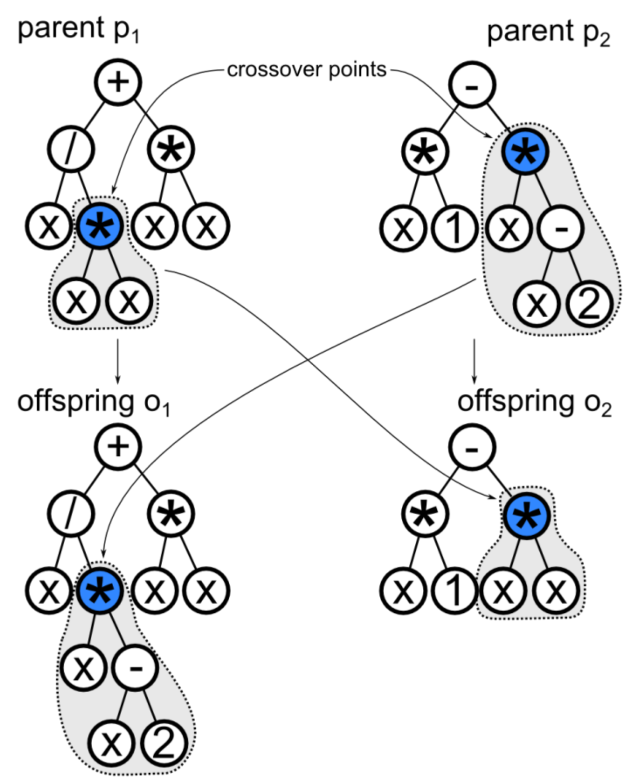
INSTITUTE OF COMPUTING SCIENCE, POZNAN UNIVERSITY OF TECHNOLOGY, POLAND

9.07.2013

Motivations

- Crossover is supposed to produce offspring that lays **in-between** parents
  - Average in common sense

- Canonic tree-swapping crossover
- Is $\frac{x}{x\times(x-2)} + x^2$ or $x - x^2$ in-between $\frac{\mathrm{x}}{\mathrm{x}^2} + x^2$ and $x - x(x-2)$?



Motivations

- Canonic Genetic Programming
  - Purely syntactic manipulations of program code
- Is offspring related to parents?

- How to measure similarity of programs?
- How to tell that an offspring lays between the parents?

What does `between` mean for programs?

- Point may be between some other points only in a metric space
- We need a metric $d: P \times P \to [0, +\infty)$ defined on program space P:
  - $d(a, b) = 0 \Leftrightarrow a = b$,
  - $d(a, b) = d(b, a)$,
  - $d(a, b) \leq d(a, c) + d(b, c)$.
- But… how to define a metric on pair of programs?

Semantics

- We induce programs from samples
- The samples are sets of numbers (in symbolic regression)
  - Set of function arguments
  - The target output value
- Let us use similar representation as semantics
  - Set of function arguments
  - The calculated output value
- Call it sampled semantics

Semantics: example

- Consider functions $f(x) = \frac{x}{x^2} + x^2$ and $g(x) = \frac{x}{x-\frac{x}{4}} + x^2$
- Sample them equidistantly in range $[-1,1]$ using 10 samples

x	f(x)	g(x)
-1,00	0,00	2,33
-0,78	-0,68	1,94
-0,56	-1,49	1,64
-0,33	-2,89	1,44
-0,11	-8,99	1,35
0,11	9,01	1,35
0,33	3,11	1,44
0,56	2,11	1,64
0,78	1,89	1,94
1,00	2,00	2,33

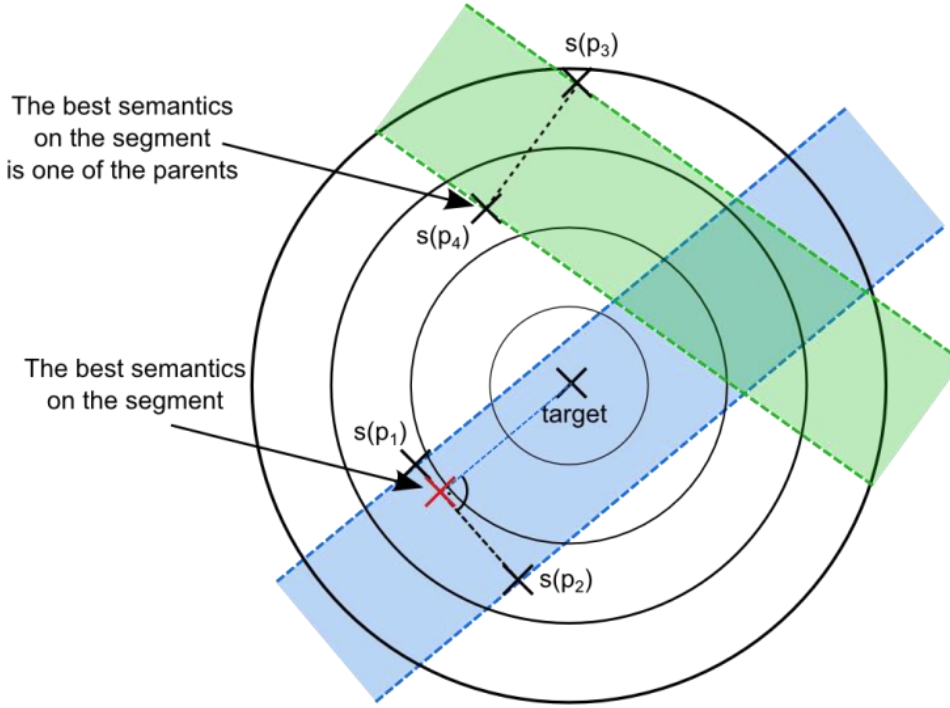
- Again: How (dis)similar is $f(x)$ to $g(x)$? Just chose a metric:
  - Manhattan: 32,93
  - Euclidean: 14,48
  - Chebyshev: 10,33

"A recombination operator is a **geometric crossover** under the metric *d* if all offspring are in the *d*-metric segment between its parents."

Alberto Moraglio, *Abstract Convex Evolutionary Search*, FOGA'11

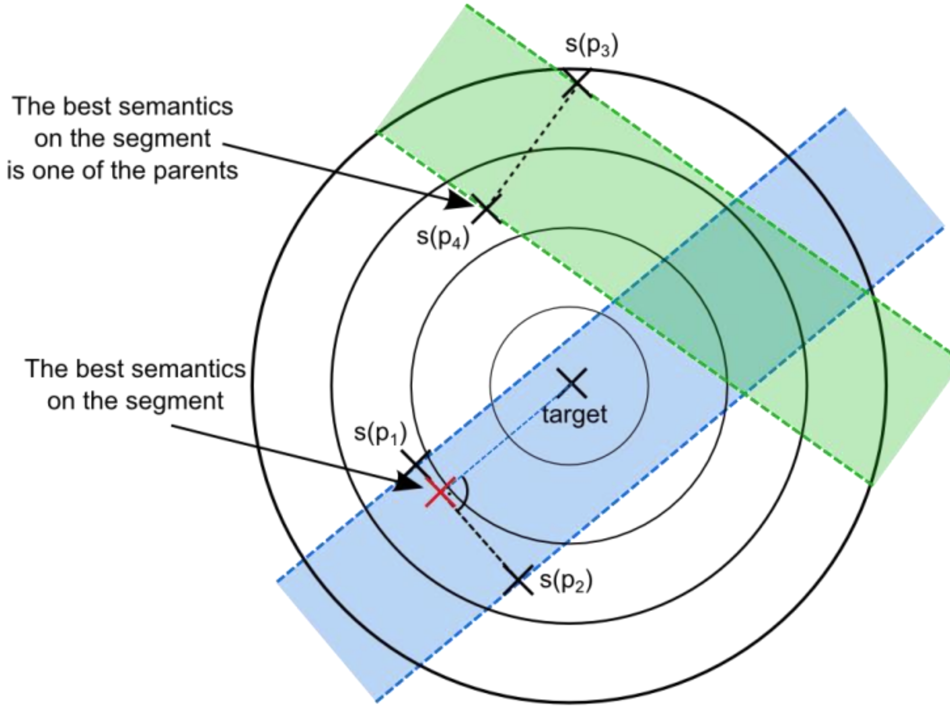
Why do we need the geometric crossover?

- Consider:
  - the Euclidean distance as a fitness/error function
  - fitness landscape spanned over k-dimensional space of program semantics
- It must be a cone
  - The apex is the global optimum
  - Programs lie on the edges of cone



Why do we need the geometric crossover?

- It is guaranteed that:
- An intermediate semantics between any pair of semantics must be not worse than the worst of the pair



Approximately Geometric Semantic Crossover (AGX)

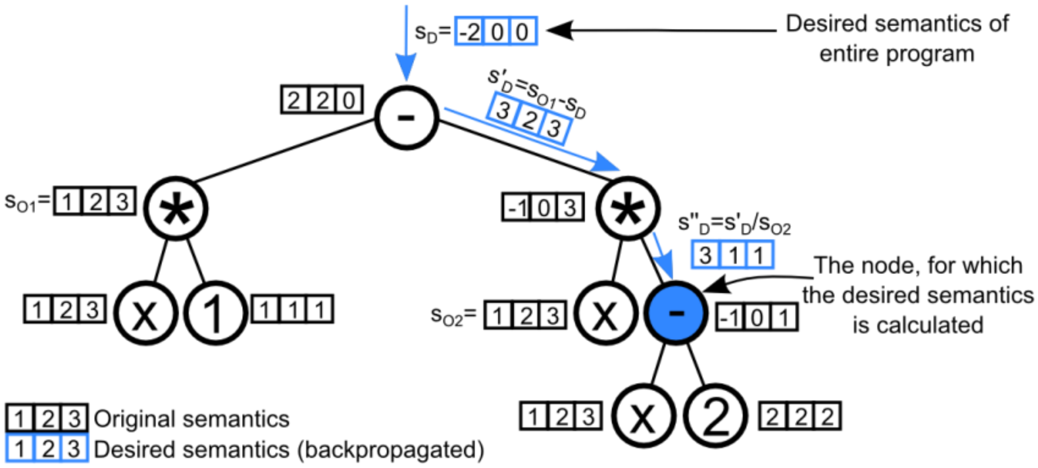
- Given two parents:
  - Calculate their semantics
  - Determine a midpoint between them
- For each parent separately:
  - Randomly choose a crossover point
  - Backpropagate midpoint to the crossover point → desired semantics
  - Replace crossover point by a subprogram having semantics that minimizes error to the desired semantics

Semantic backpropagation

- The objective
  - Propagate the semantic target backwards through the program tree, so that it defines a subgoal for a subproblem
- Input
  - The program p
  - The target semantics s_D
  - The chosen node p'
- Output
  - Desired semantics $s_D(p')$ for p'

Semantic backpropagation

- Starting from the root node, for each node p on the path to p', do recursively:
  - Obtain an inverse instruction p^{-1} to p w.r.t. child node p_c, which is next on the path
  - Execute p^{-1} to compute desired semantics $s_D(p_c)$
  - Stop if recursion reaches the chosen node ($p_c \equiv p'$)



Semantic backpropagation: possible cases

- Instruction is invertible
  - $p: y \leftarrow x + c \quad \Rightarrow p^{-1}: x \leftarrow c - y$
- Instruction is ambiguously invertible
  - $p: z \leftarrow x^2 \qquad \Rightarrow p^{-1}: x \in \{-\sqrt{z}, \sqrt{z}\}$
  - $p: \sin(x) \qquad \Rightarrow p^{-1}: x \leftarrow \arcsin(z) + 2k\pi, k \in \mathbb{Z}$
- Instruction is non-invertible
  - $p: z \leftarrow e^x \qquad \Rightarrow p^{-1}: \forall_{z \in \mathbb{R}^-} x \leftarrow X$ (NaN, inconsistent)
- Argument of instruction is ineffective
  - $p: z \leftarrow 0 \times x \quad \Rightarrow p^{-1}: x \leftarrow ?$ (don't care)

Library of procedures

- A static library
  - All possible programs built upon given set of instructions
  - Filtered for semantic uniqueness

- In experiment:
  - Instructions $\{+, -, \times, /, sin, cos, exp, log, x\}$
  - Max tree height $\mathrm{h} \in \{3,4\}$
  - Total number of programs: $212, 108520$

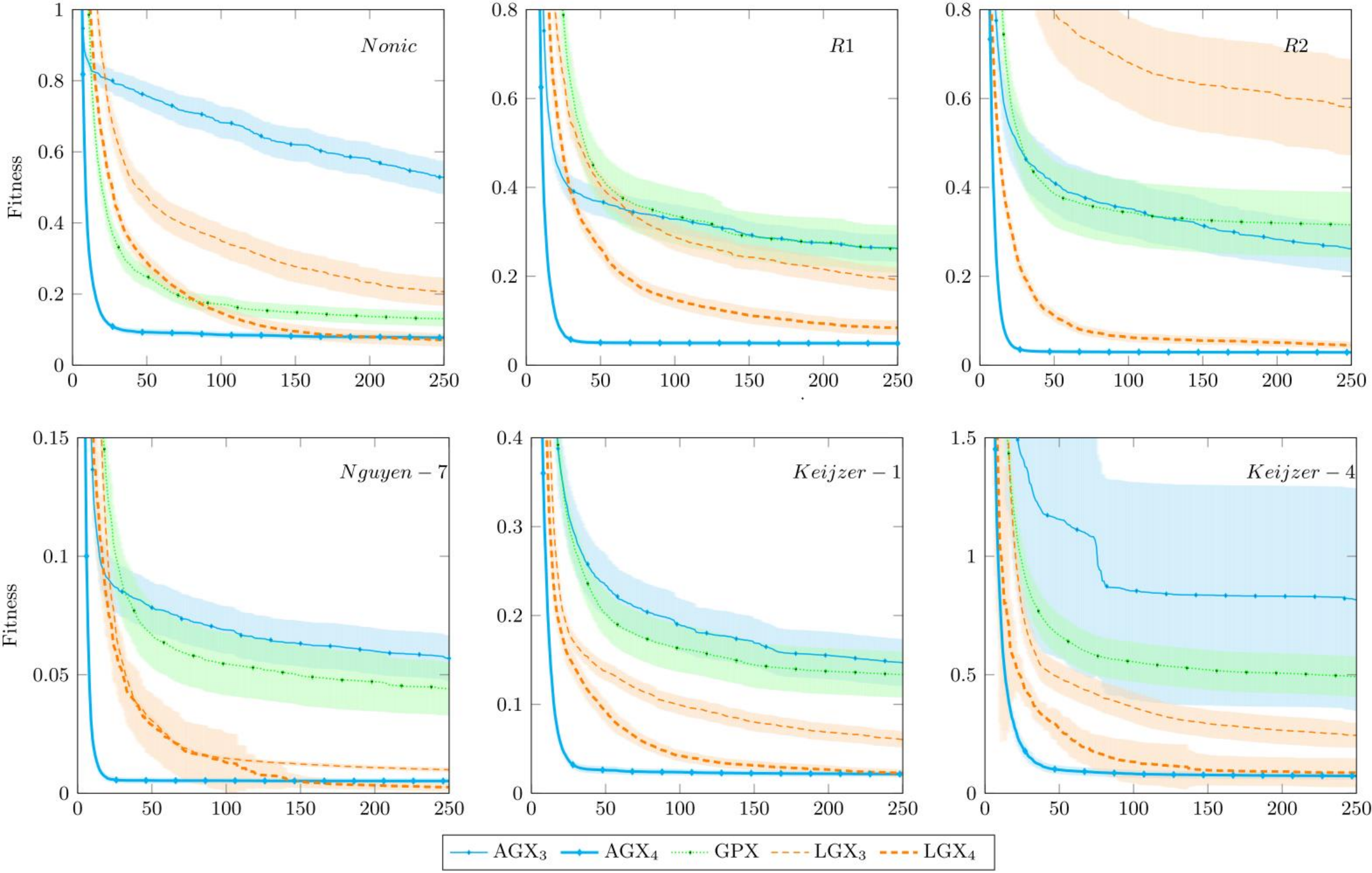
The experiment

- Competition:
  - GPX: standard tree-swapping crossover
  - LGX: locally geometric semantic crossover*

Problem	Definition (formula)	Training set	Test set
Nonic	$x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x$	E[-1, 1, 20]	U[-1, 1, 20]
R1	$(x+1)^3/(x^2 - x + 1)$	E[-1, 1, 20]	U[-1, 1, 20]
R2	$(x^5 - 3x^3 + 1)/(x^2 + 1)$	E[-1, 1, 20]	U[-1, 1, 20]
Nguyen-7	$\log(x+1) + (x^2 + 1)$	E[0, 2, 20]	U[0, 2, 20]
Keijzer-1	$0.3x\sin(2\pi x)$	E[-1, 1, 20]	U[-1, 1, 20]
Keijzer-4	$x^3 e^{-x} \cos(x) \sin(x)(\sin^2(x)\cos(x) - 1)$	E[0, 10, 20]	U[0, 10, 20]

E[a,b,n] – n points chosen *equidistantly* from range [a,b]
U[a,b,n] – n points chosen randomly with *uniform* distribution from range [a,b]

* K. Krawiec, T. Pawlak, Locally geometric semantic crossover: a study on the roles of semantics and homology in recombination operators. Genetic Programming and Evolvable Machines, 14(1):31-63, 2013.


Nonic
R1
R2
Nguyen − 7
Keijzer − 1
Keijzer − 4
Fitness
AGX3
AGX4
GPX
LGX3
LGX4

Test-set performance

Average error committed by best-of-run individual on test set.

Problem	AGX3	AGX4	GPX	LGX3	LGX4
Nonic	0.359	**0.093**	0.130	0.201	0.191
R1	0.224	**0.050**	0.261	0.167	0.103
R2	10^7	**0.028**	0.316	0.621	0.042
Nguyen-7	0.051	0.005	0.044	0.018	**0.004**
Keijzer-1	0.190	**0.039**	0.134	0.091	0.041
Kejzer-4	3.113	10^{13}	**0.492**	2.008	2.854

Geometry of operators

Depth of crossover	Fraction of geometric offspring AGX	LGX	GPX
1	.0155	**.1676**	.0035
2	**.0151**	.0100	.0031
3	**.0136**	.0031	.0018
4	**.0105**	.0016	.0020
5	**.0055**	.0014	.0011
6	**.0028**	.0009	.0007
7	**.0017**	.0006	.0005
8	**.0012**	.0004	.0003
9	**.0010**	.0007	.0003
10	**.0006**	.0005	.0003
11	**.0005**	.0002	.0003
12	**.0004**	.0001	.0003
13	**.0003**	.0002	.0002
14	.0002	.0000	**.0005**
15	.0000	.0000	**.0002**
16	.0000	.0000	**.0005**
17	.0000	.0000	.0000
Overall	**.0057**	.0035	.0008

Future work

- Test other libraries
- Add support for constants
- Compare with Random Desired Operator*

* K. Krawiec, B. Wieloch. Running Programs Backwards, GECCO 2013.

Thank you

Questions?