Semantic Genetic Programming

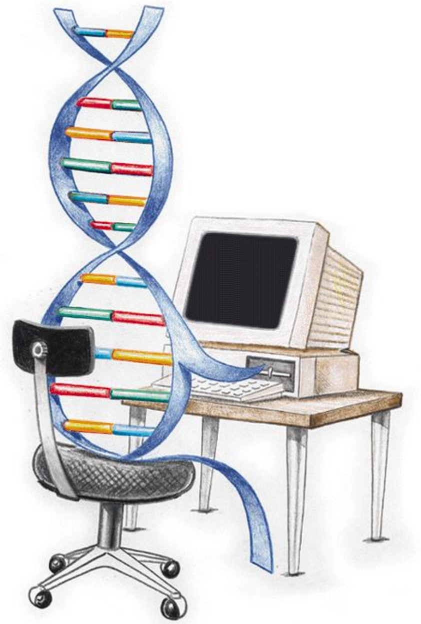
TOMASZ PAWLAK

INSTITUTE OF COMPUTING SCIENCE, POZNAN UNIVERSITY OF TECHNOLOGY

27.11.2012

Outline

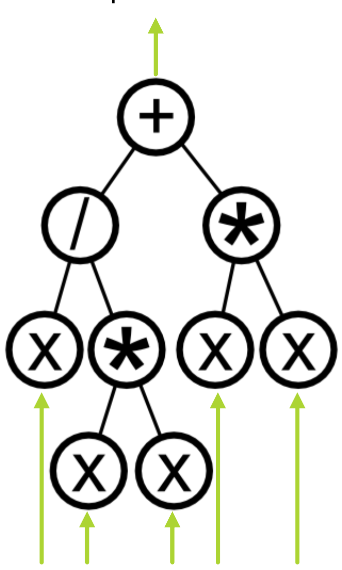
- Genetic programming
- Genetic operators
- Semantics
- Geometric genetic operators
- Results



Genetic Programming

- Automatic induction of computer programs from samples
- Sample (pair of):
  - Set of arguments
  - Desired output value
- Program representation
  - Syntax tree
  - Linear (like assembler)
  - Graph
  - and more…

Genetic Programming



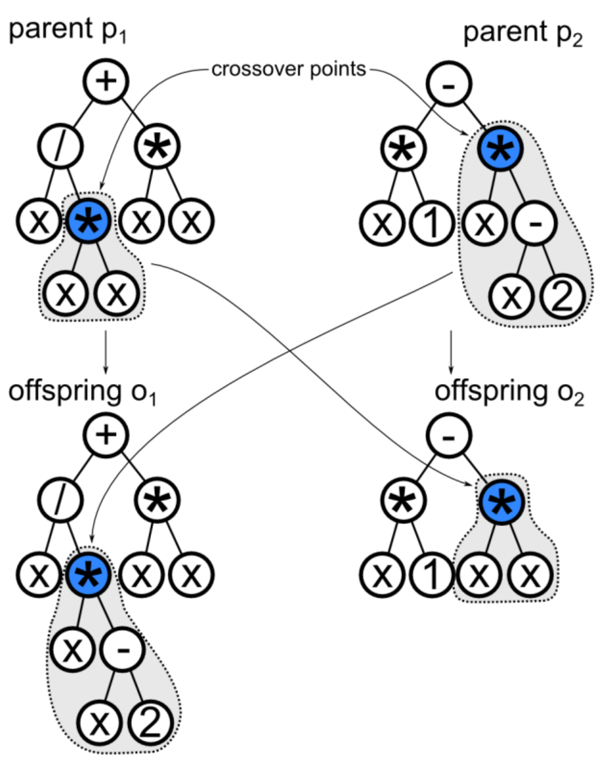
- $\frac{x}{x^2} + x^2 = \frac{1}{x} + x^2$
- Prefix notation:
  - (+ (/ x (* x x)) (* x x))
- No explicit memory storage

GP typical tasks

- Symbolic regression
- Classification
- Planning and control
- Logic circuit synthesis
- Evolvable hardware

The NASA ST5 spacecraft antenna evolved by GP

Genetic operators: subtree crossover



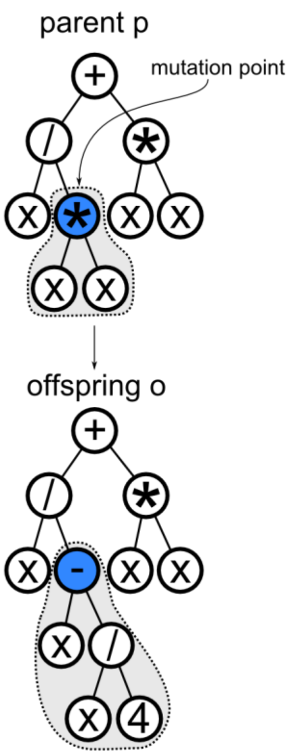
- Is the result predictable?
  - Yes, but…
- Crossover is supposed to produce offspring between parents
  - Average in common sense
- Are $\frac{x}{x\times(x-2)} + x^2$ or $x - x^2$ between $\frac{x}{x^2} + x^2$ and $x - x(x-2)$?

What does `between` mean for programs?

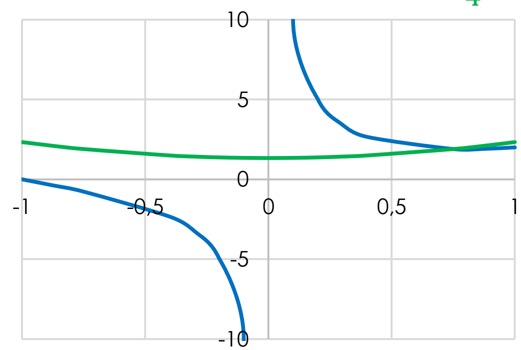
- Point may be between some other points only in a metric space
- We need a metric $d: P \times P \to [0, +\infty)$ defined on program space P:
  - $d(a,b) = 0 \Leftrightarrow a = b$,
  - $d(a,b) = d(b,a)$,
  - $d(a,b) \leq d(a,c) + d(b,c)$.
- But… how to define a metric on pair of programs?

We address this later.

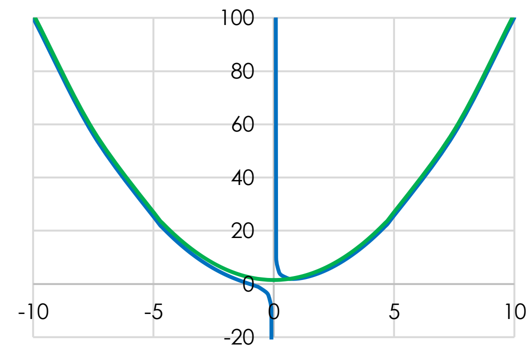
Genetic operators: mutation



- Mutation is supposed to make an elementary change to the given solution
- Is replacement of whole subtree an elementary change?
- Is $\frac{x}{x^2} + x^2$ similar to $\frac{x}{x-\frac{x}{4}} + x^2$?



No?



Yes?

What does `similar` mean for programs?

- How similar is + to –?
  - What about + and /?
- Again:
  - We need a metric
  - How to define a metric on instructions?

Semantics

- We induce programs from samples
- The samples are sets of numbers (in symbolic regression)
  - Set of function arguments
  - The desired 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 it 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

Semantics in context of GP

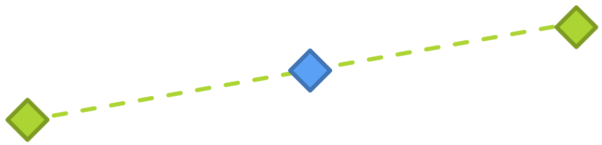
- Computed every time a program is evaluated
  - The fitness function is some kind of distance measure
  - It is essentially free to obtain
- A part of program is also a program, that can be executed
  - Semantics can be calculated in (almost) every node of the tree

Sampled semantics: properties

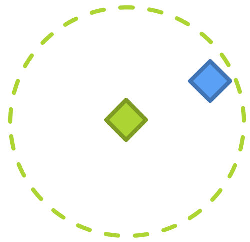
- Advantages
  - Similar representation to the way, how problem is posed
  - Many distance metrics (any Minkowski distance L_p)
  - Low computational costs (in context of GP)
  - Extendable to any precision and any number of values (e.g. complex numbers)
- Disadvantages
  - Does not contain whole information about subject (it's only a sample)
  - Problem-dependent (arguments)

Geometric genetic operators

- In a metric space
  - The object may be between some other objects



  - The object may be in a given perimeter of other object

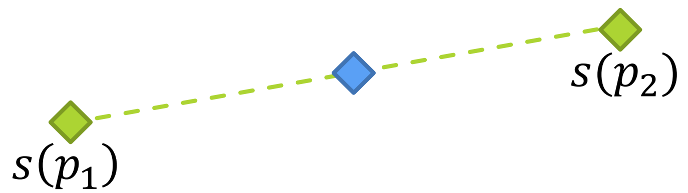


"A recombination operator is a **geometric crossover** under the metric *d* if all offspring are in the *d*-metric segment between its parents."

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

Geometric crossover

- So, we can calculate (range of) semantics between semantics of parents $s(p_1)$ and $s(p_2)$



- But… how to obtain a program having desired semantics?
  - If it were easy, we would not need an optimization algorithm

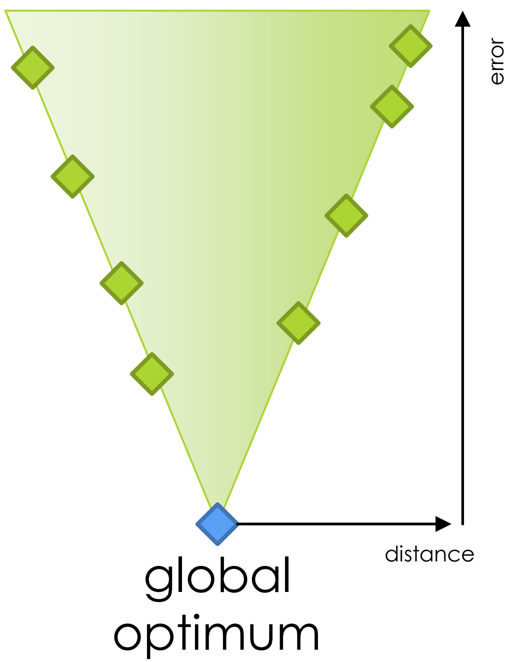
How do we obtain a program having semantics intermediate between two other programs?

- We can build a library of programs
  - How big should this library be?
  - Too few programs:
    - We may be not able to find the desired one
  - Too many programs:
    - We could not store the library in memory (slow access)
  - Infinite number of programs…

Not possible for many real-world problems.

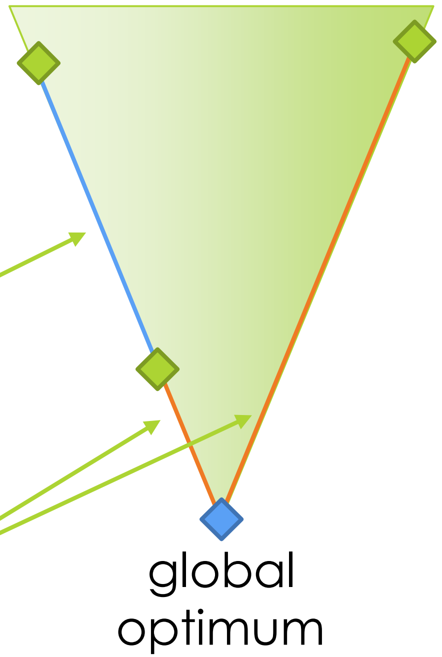
Why do we need the geometric crossover?

- 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 vertex is the global optimum
  - Programs lie on the edges of cone

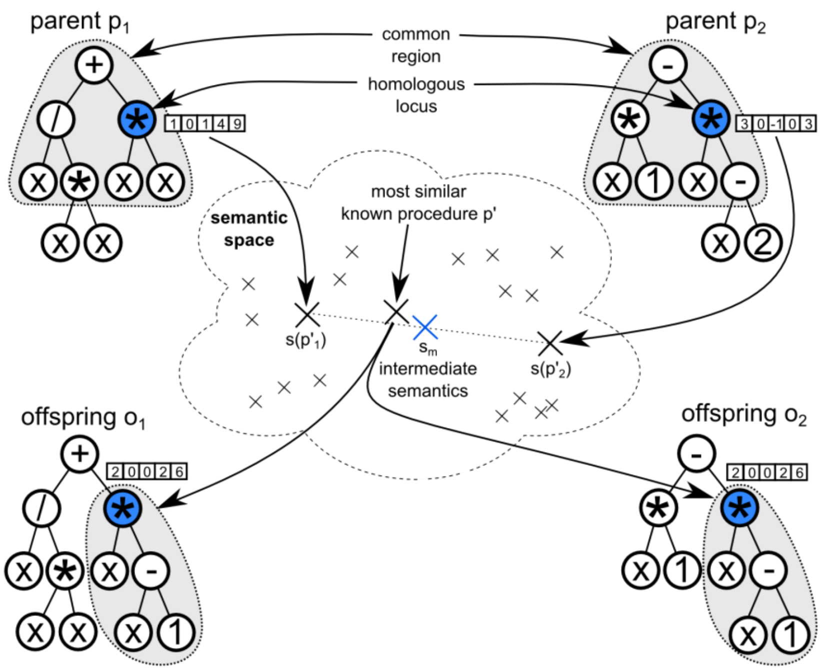


Why do we need the geometric crossover?

- It is guaranteed that:
  - An intermediate semantics between any pair of semantics must be not worse than the worst of the pair
- A sketch of proof:
- If the pair lies on a single side of cone
  - The fitness of intermediate solution must be between fitness values defined by the pair
- If the pair lies on opposite sides of cone
  - The fitness of intermediate solution must be not worse than fitness of the worst of pair



Locally Geometric Semantic Crossover (LGX)



- Choose a homologous crossover point (syntactically)
- Calculate average semantics between subtrees rooted at chosen point
- Use library to find the closest procedure to the calculated semantics
- Place the found procedure at crossover point in both parents

"**Geometric mutation** is defined geometrically requiring that offspring are in a *d*-ball of a certain radius centered in the parent."

ALBERTO MORAGLIO, *ABSTRACT CONVEX EVOLUTIONARY SEARCH*, FOGA'11

Locally Geometric Semantic Mutation (LGM)

- Similar to LGX
- Randomly choose mutation point
- Choose a procedure from library according to the Poisson distribution (with given λ)
- Replace the subtree rooted at mutation point with the chosen procedure
- Rationale:
  - The change cannot be too little
  - The change cannot be too big

Competition

- Semantic-Aware Crossover (SAC)
- Semantic Similarity-based Crossover (SSC)
- Semantic-Aware Mutation (SAM)
- Semantic Similarity-based Mutation (SSM)

Control methods

- Tree Swapping Crossover (GPX)
- One Point Crossover (GPH)
- Nonhomologous Geometric Crossover (NHX)
- Random Crossover (RX)

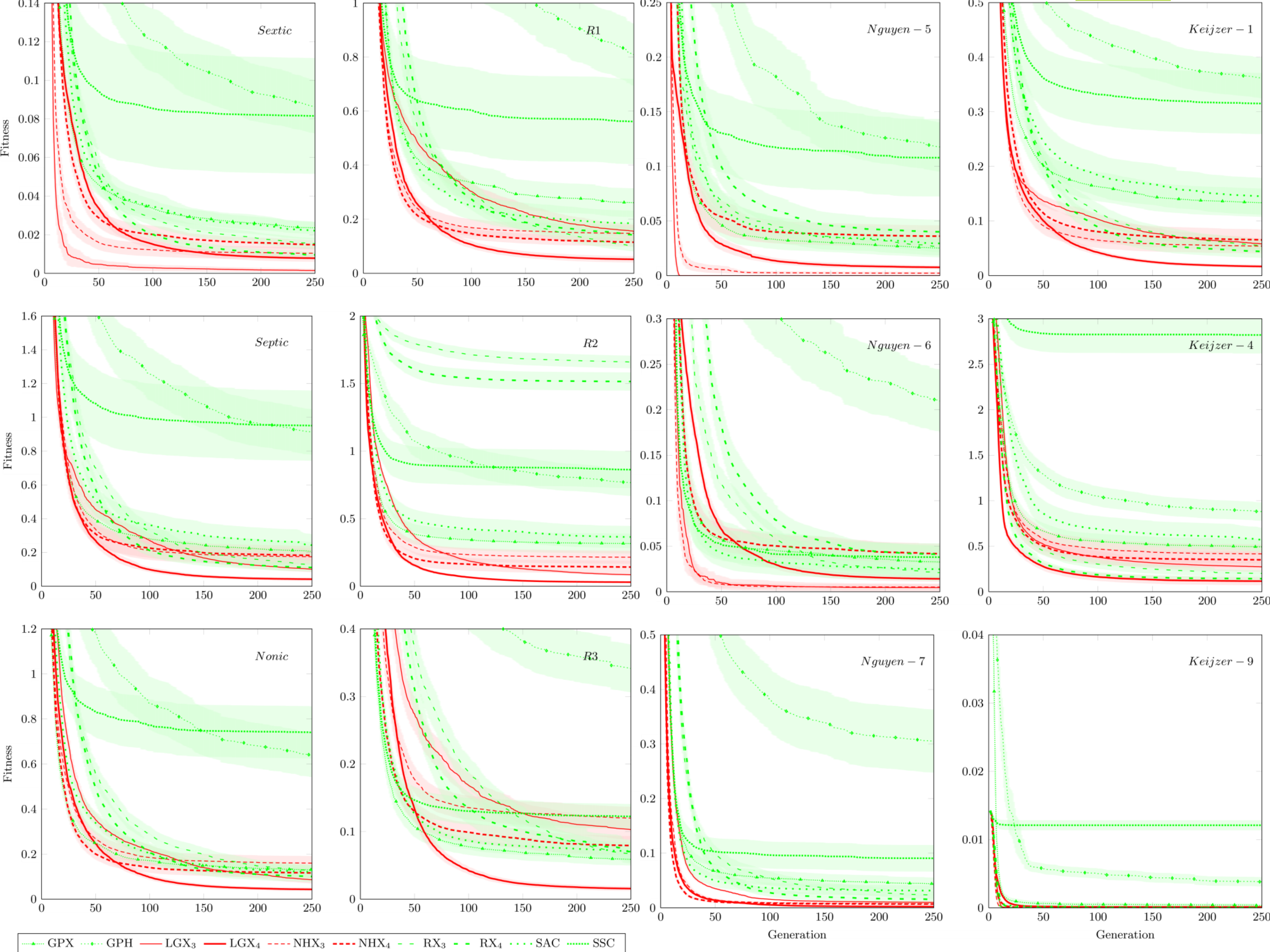
The experiment

Parameter	LGX, NHX, RX	SAC, SSC	GPX, GPH
Instruction set	{+,-,*,/,sin,cos,exp,log,x}		
Population size	1024		
Initial max tree depth	6		
Max tree depth	17		
Selection	Tournament selection		
Trials per experiment	100 independent runs		
Termination condition	250 generations and at least 200s of total time		
Crossover probability	0.9		
Mutation probability	0.0	0.0	0.1
Reproduction probability	0.1	0.1	0.0
Max tree depth in library	{3,4}	–	–
Neighborhood size	8	–	–
Semantic sensitivity	–	0.5	–
Lower bound semantic sensitivity	–	0.0001	–
Upper bound semantic sensitivity	–	0.4	–

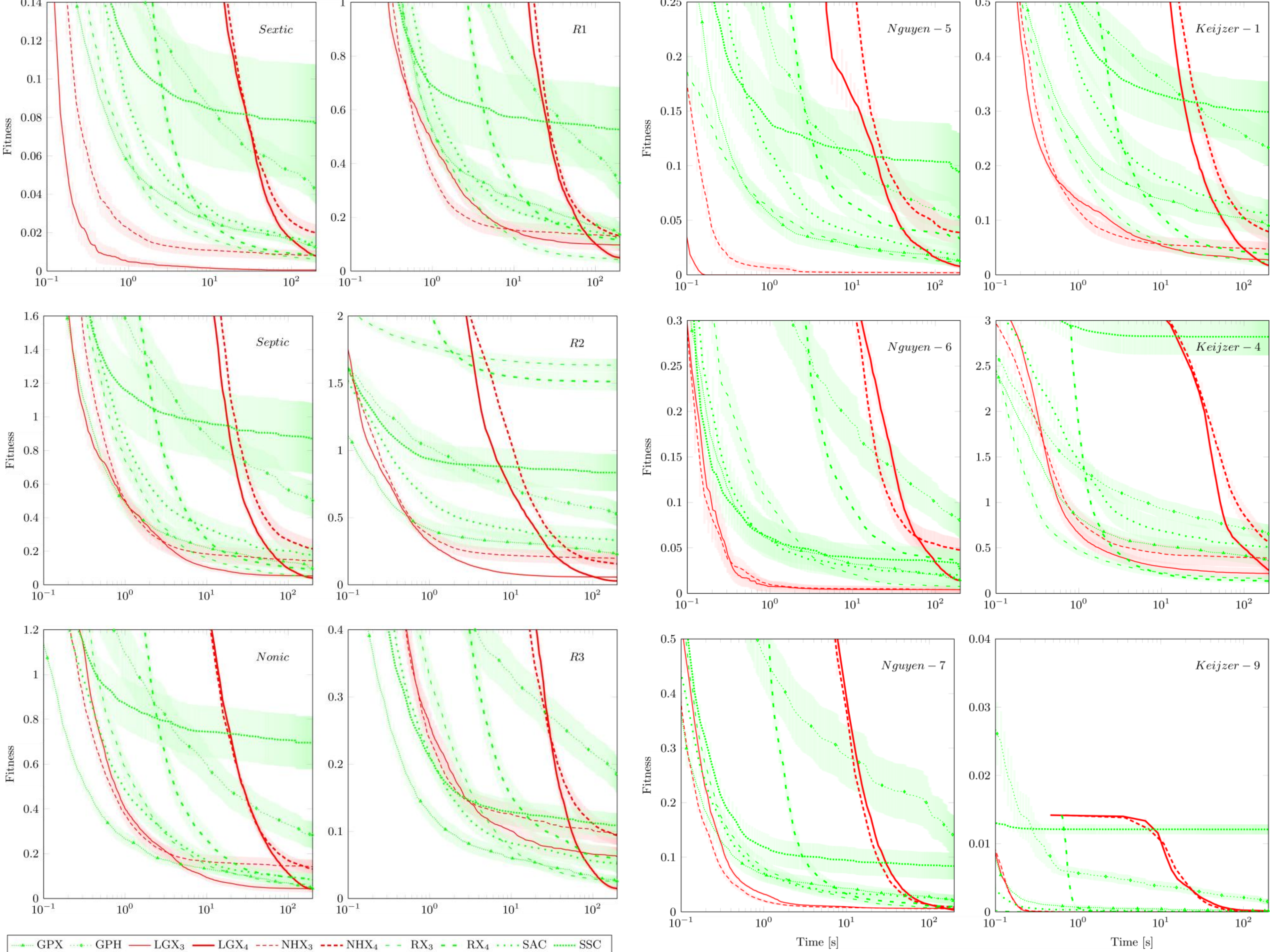
Benchmark problems

Problem	Definition (formula)	Training set	Test set
Sextic	$x^6 - 2x^4 + x^2$	U[-1, 1, 20]	R[-1, 1, 20]
Septic	$x^7 - 2x^6 + x^5 - x^4 + x^3 - 2x^2 + x$	U[-1, 1, 20]	R[-1, 1, 20]
Nonic	$x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x$	U[-1, 1, 20]	R[-1, 1, 20]
R1	$(x+1)^3/(x^2 - x + 1)$	U[-1, 1, 20]	R[-1, 1, 20]
R2	$(x^5 - 3x^3 + 1)/(x^2 + 1)$	U[-1, 1, 20]	R[-1, 1, 20]
R3	$(x^6 + x^5)/(x^4 + x^3 + x^2 + x + 1)$	U[-1, 1, 20]	R[-1, 1, 20]
Nguyen-5	$\sin(x^2)\cos(x) - 1$	U[-1, 1, 20]	R[-1, 1, 20]
Nguyen-6	$\sin(x) + \sin(x + x^2)$	U[-1, 1, 20]	R[-1, 1, 20]
Nguyen-7	$\log(x+1) + (x^2 + 1)$	U[0, 2, 20]	R[0, 2, 20]
Keijzer-1	$0.3x\sin(2\pi x)$	U[-1, 1, 20]	R[-1, 1, 20]
Keijzer-4	$x^3 e^{-x}\cos(x)\sin(x)(\sin^2(x)\cos(x) - 1)$	U[0, 10, 20]	R[0, 10, 20]
Keijzer-9	$\log\left(x + \sqrt{x^2 + 1}\right)$	U[0, 100, 20]	R[0, 100, 20]

U[a, b, c] = c values chosen uniformly from range [a, b]
R[a, b, c] = c values chosen randomly with uniform distribution from range [a, b]



Sextic
R1
Nguyen − 5
Keijzer − 1
Septic
R2
Nguyen − 6
Keijzer − 4
Nonic
R3
Nguyen − 7
Keijzer − 9
Fitness
Generation
GPX
GPH
LGX3
LGX4
NHX3
NHX4
RX3
RX4
SAC
SSC


Sextic
R1
Nguyen − 5
Keijzer − 1
Septic
R2
Nguyen − 6
Keijzer − 4
Nonic
R3
Nguyen − 7
Keijzer − 9
Fitness
Time [s]
GPX
GPH
LGX$_3$
LGX$_4$
NHX$_3$
NHX$_4$
RX$_3$
RX$_4$
SAC
SSC

Success rate (%)

Problem	GPX	GPH	LGX3	LGX4	NHX3	NHX4	RX3	RX4	SAC	SSC
Sextic			**85**	3	31		6	1	3	2
Septic										
Nonic	**1**									
R1										
R2										
R3										
Nguyen-5	8	3	**100**	1	73	1	6		4	3
Nguyen-6	50	9	**91**	3	78	3	22	1	53	48
Nguyen-7	6				**12**	1			5	2
Keijzer-1										
Keijzer-4				**1**	**1**			**1**		
Keijzer-9	41		**91**	24	71	34	63	56	75	2

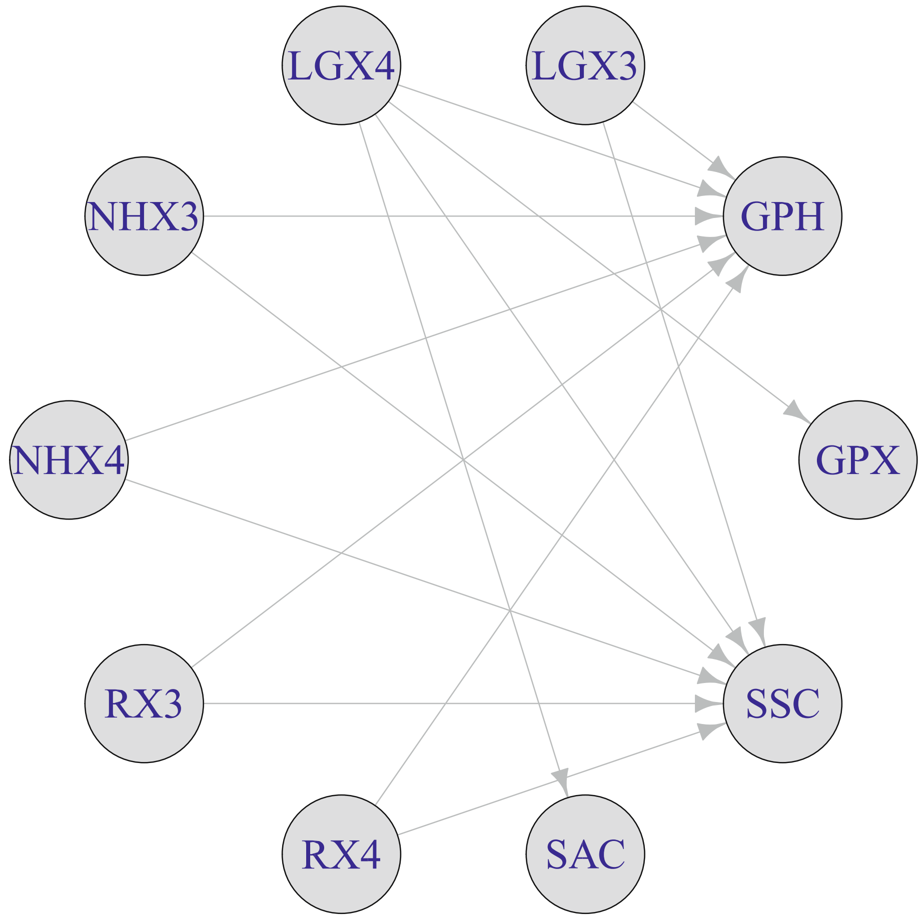
Statistical significance

- Friedman’s test for multiple achievements of a series of subjects on the average of best-of-run fitness
  - $p = 2.589 \times 10^{-8}$
- Post-hoc analysis (symmetry test)

	GPX	GPH	LGX3	LGX4	NHX3	NHX4	RX3	RX4	SAC	SSC
GPX		0.310					0.899	0.899	1.000	0.487
GPH										1.000
LGX3	0.149	**0.000**			0.980	0.804	0.958	0.958	0.125	**0.000**
LGX4	**0.010**	**0.000**	0.997		0.582	0.236	0.486	0.486	**0.008**	**0.000**
NHX3	0.840	**0.002**					1.000	1.000	0.804	**0.006**
NHX4	0.987	**0.017**			1.000		1.000	1.000	0.980	**0.039**
RX3		**0.004**								**0.010**
RX4		**0.004**					1.000			**0.011**
SAC		0.351					0.872	0.871		0.535
SSC										

Probability of erroneously judging the method in row as outranking the method in column. Lower values = stronger confidence. $\alpha = 0.05$

Outranking graph



Generalization abilities

Errors committed on test set by the best-of-run individuals as of 250 generation.

Problem	GPX	GPH	LGX3	LGX4	NHX3	NHX4	RX3	RX4	SAC	SSC
Sextic	0.024	0.086	**0.002**	0.091	10^{13}	0.044	0.029	0.106	0.092	0.106
Septic	0.207	0.914	**0.096**	0.214	0.197	0.390	0.220	10^{13}	0.366	0.776
Nonic	0.130	0.639	**0.104**	0.217	0.150	0.226	10^{13}	0.828	0.199	0.577
R1	0.261	0.809	0.159	0.181	0.145	0.185	**0.124**	40.32	0.238	0.515
R2	0.316	0.767	0.092	**0.091**	0.245	0.357	10^{5}	10^{13}	0.451	0.958
R3	**0.059**	0.341	0.090	0.144	0.225	0.139	0.238	0.661	10^{13}	0.179
Nguyen-5	0.025	0.118	**0.000**	0.013	0.003	0.040	0.030	10^{13}	0.046	0.092
Nguyen-6	0.033	0.210	**0.004**	0.033	**0.004**	0.041	0.019	0.129	0.026	10^{13}
Nguyen-7	0.044	0.305	0.008	**0.005**	0.007	0.007	0.043	10.90	0.056	0.085
Keijzer-1	0.134	0.362	**0.092**	0.108	0.106	1.381	0.103	67.36	10^{13}	0.335
Keijzer-4	**0.492**	0.881	1.363	13.27	1.838	10^{13}	1.675	30.24	54.42	10^{13}
Keijzer-9	**0.000**	0.004	0.003	0.592	0.005	0.064	0.159	4.160	0.011	0.192

Future work

- Comparative analysis of performance of LGM
- Analysis of propagation of geometric changes done by LGX
- Geometric and semantically-based initialization of population
- Move the concept of semantically geometric operators outside GP:
  - Local search heuristics

~~Future~~ current work: propagation of geometric changes

Percent of geometric changes propagated to higher level nodes in tree (46080 samples).

Depth of crossover:																	
1																	
2	0,000	0,283															
3	0,000	0,003	0,337														
4		0,005	0,019	0,348													
5	0,000	0,003	0,009	0,000	0,338												
6		0,001	0,005	0,000	0,000	0,415											
7	0,000	0,000		0,000	0,001	0,002	0,420										
8	0,000			0,000	0,000	0,003	0,016	0,360									
9						0,007	0,010	0,001	0,316								
10	0,000			0,000	0,000	0,006	0,009	0,001	0,003	0,215							
11	0,000	0,000	0,000	0,001	0,000	0,005	0,007	0,001	0,005	0,003	0,228						
12	0,000	0,000	0,000	0,000		0,002	0,008		0,004	0,001		0,355					
13	0,000		0,000		0,000	0,003	0,011	0,001	0,005	0,001	0,000		0,348				
14					0,000	0,005	0,020		0,005	0,001			0,000	0,237			
15					0,001	0,002	0,014		0,002		0,001				0,210		
16							0,007									0,274	
17							0,020										0,061
Depth of tree:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

References

- Krzysztof Krawiec, Tomasz Pawlak, *Locally Geometric Semantic Crossover*, GECCO'12, ACM, 2012.
- Krzysztof Krawiec, Tomasz Pawlak, *A quantitative analysis of Locally Geometric Semantic Crossover*, PPSN'12, LNCS, 2012.
- Krzysztof Krawiec, Tomasz Pawlak, *Locally Geometric Semantic Crossover: a Study on the Roles of Semantics and Homology in Recombination Operators*, Genetic Programming and Evolvable Machines, 2012 (IF = 1.000, KBN = 25).

References

- John Koza, *On the programming of computers by means of natural selection*, MIT Press, 1994.
- Hornby, Globus, Linden, Lohn, *Automated antenna design with evolutionary algorithms*, American Institute of Aeronautics and Astronautics, 2006.
- Alberto Moraglio, *Abstract Convex Evolutionary Search, Proceedings of FOGA'11, ACM, 2011.*
- Nguyen, Nguyen, O'Neill, *Semantic aware crossover for genetic programming: The case for real-valued function regression*, LNCS, 2009.
- Nguyen, Nguyen, O'Neill, *Semantic similarity based crossover in GP: The case for real-valued function regression*, LNCS, 2009.
- Nguyen, Nguyen, O'Neill, McKay, Galban-Lopez, Semantically-based crossover in genetic programming: application to real-valued symbolic regression, Genetic Programming and Evolvable Machines, 2011.
- Nguyen, Nguyen, O'Neill, Semantics based mutation in genetic programming: the case for real-valued symbolic regression, MENDEL'09, 2009.

Thank you

Questions?