Behavioral Programming: A Broader and More Detailed Take on Semantic GP

Krzysztof Krawiec[1] Una-May O'Reilly[2]

[1]Institute of Computing Science
Poznan University of Technology, Poland

[2]CSAIL, Massachusetts Institute of Technology, Cambridge, MA

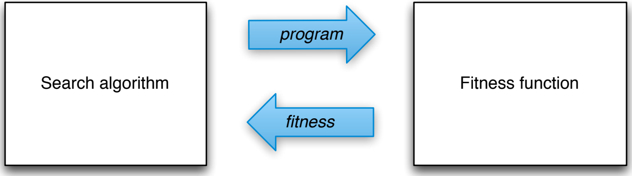
GECCO'14, Vancouver, July 15, 2014

Problem statement and key insights

1. **Problem:** Evaluation bottleneck in Genetic Programming
2. **Hypothesis:** Richer evaluation feedback can make search more efficient
3. **Proposed solution:** Behavioral programming
4. **Result:** Better performance and implicit problem decomposition

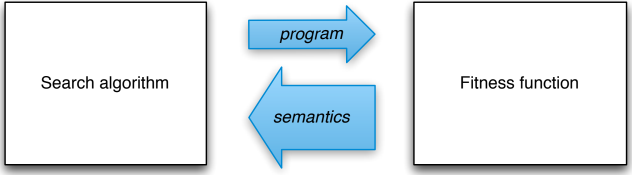
Conventional evaluation in GP

- GP = iterative search in program space, driven by fitness function



- Complex evaluation process
- Yet: scant feedback for the search algorithm (scalar fitness)

Semantic-aware evaluation

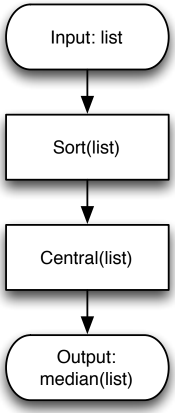


- Richer feedback: semantics (program output for every example)
- Specialized search operators exploit program semantics
  - E.g., using the geometric properties of semantic space (Moraglio *et al.* 2011)

Observations

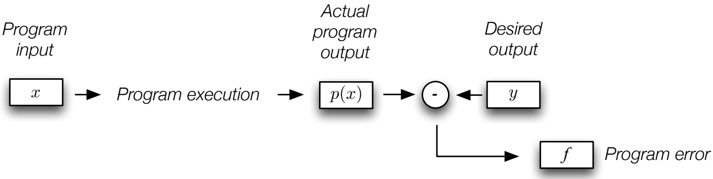
- Semantics does not capture the entirety of program activity.
    - Only the output of a program matters.
- Program output is a combined effects of multiple execution steps (instructions).
- A program can produce incorrect output, even if it is partially correct,
    - i.e., features subprograms that produce useful/correct partial results.

Example: Calculating the median

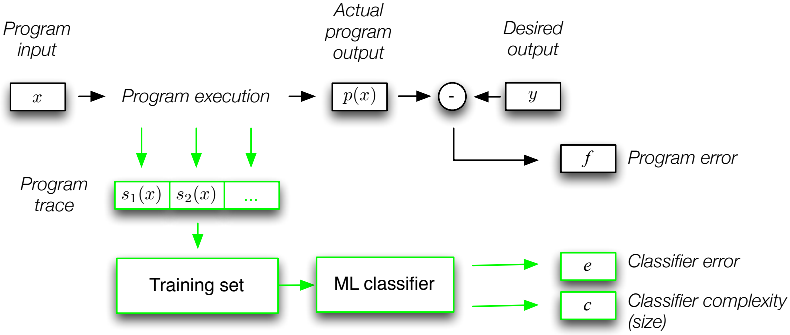


- The sorted list is a *desired intermediate computation state*.
- Stage-wise problem decomposition.

PANGEA: Pattern-Guided Evolutionary Algorithm



PANGEA: Pattern-Guided Evolutionary Algorithm



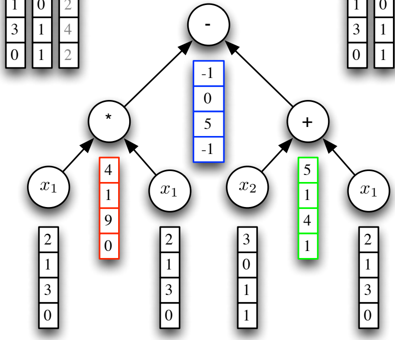
- **Black:** Conventional GP
- **Green:** PANGEA (Krawiec & Swan, GECCO'13)

Example (nominal domain, tree-based GP)

Problem

x_1	x_2	y
2	3	3
1	0	2
3	1	4
0	1	2

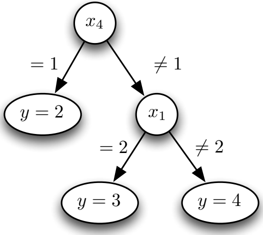
GP Individual



ML dataset

x_1	x_2	x_3	x_4	y
2	3	4	5	3
1	0	1	1	2
3	1	9	4	4
0	1	0	1	2

Decision tree



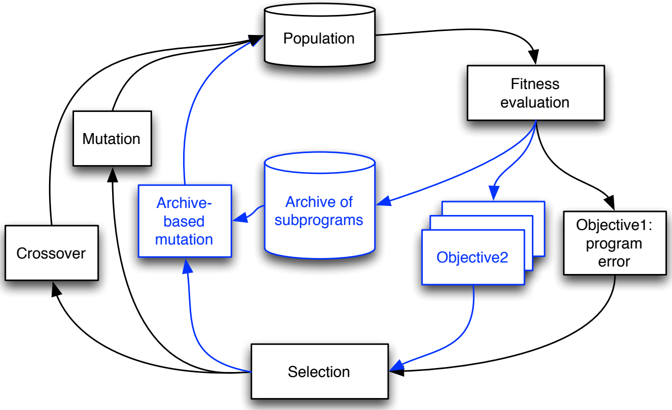
Evaluation:

Program error	10
Classifier error	0 examples
Classifier complexity	5 nodes

Behavioral programming

Behavioral programming = PANGEA extended with:

- Multiobjective evaluation and selection (NSGA-II, Deb *et al.* 2002),
- Archiving of promising subprograms,
- Mutation operator supplied by subprograms from the archive.



Experiment: Benchmarks

Domain	Instruction set	Problem	Variables	Fitness cases	Space size
Boolean	and, nand, or, nor	Cmp6	6	64	2^{64}
		Cmp8	8	256	2^{256}
		Maj6	6	64	2^{64}
		Maj8	8	256	2^{256}
		Mux6	6	64	2^{64}
		Mux11	11	$2,048$	2^{2048}
		Par6	6	64	2^{64}
		Par8	8	256	2^{256}
Categorical	$a_i(x, y)$	D-a1..a5	3	27	3^{27}
	$a_i(x, y)$	M-a1..a5	3	15	3^{15}
Regression	$+$, $-$, $*$, %, sin, cos, log, exp, $-x$	Keij1,Keij4	1	20	–
		Nguy3,Nguy4	1		
		Nguy5,Nguy6	1		
		Nguy7,Sext	1		
		Keij5, Keij11	2		
		Keij12, Keij13	2		
		Keij14, Nguy9	2		
		Nguy10, Nguy12	2		
		Keij15	3		

Experiment: Configurations

		GP			Behavioral programming		
		GP1	GP1L	GP2	BPGP4	BP2A	BPGP4A
Objectives	Program error	✓	✓	✓	✓	✓	✓
	Program size			✓	✓	✓	✓
	Classifier complexity				✓		✓
	Classifier error				✓		✓
Operators	Mutation	0.1	0.1	0.1	0.1	0.1	0.1
	Crossover	0.9	0.9	0.9	0.9		
	Archive mutation					0.9	0.9
	Pop. size	100	1000	100	100	100	100

- Decision tree: REPTree (fast, applicable to classification and regression)
- Statistics: Friedman test + Shaffer's post-hoc analysis

Experiment: Results

(Ranks averaged over 35 benchmarks)

Success rate:	**BPGP4A**	GP1L	BP2A	BPGP4	GP1	GP2
	2.43	3.10	3.36	3.43	**3.86**	**4.83**
Program error:	**BPGP4A**	GP1L	BP2A	BPGP4	GP1	GP2
	2.17	2.21	**3.31**	**3.71**	**3.79**	**5.8**
Predictive accuracy:	BPGP4	**BPGP4A**	BP2A	GP1	GP1L	GP2
	2.65	3.35	3.47	3.47	3.59	4.47
Run time:	GP1	GP2	BPGP4	**BPGP4A**	BP2A	GP1L
	1.09	**3.11**	**3.77**	**3.97**	**4.09**	**4.97**
Program size:	GP2	BPGP4	BP2A	**BPGP4A**	GP1	GP1L
	1.26	**2.57**	**2.89**	**3.91**	**4.81**	**5.56**

Essential factors: behavioral evaluation and archive-based mutation

Conclusions

Behavioral programming =

- Better-informed search process
- Better search performance
- Almost parameterless
- Domain-independence
- Implicit problem decomposition
- Applicable to all GP paradigms

Future/ongoing work:

- Alternative ‘pattern detectors’ (PPSN’14)

Acknowledgment: FULBRIGHT Polska