

# Applying Genetic Programming to Bytecode and Assembly

Eric Collom

Division of Science and Mathematics  
University of Minnesota, Morris  
Morris, Minnesota, USA

29 April '14,  
UMM Senior Seminar

# The big picture

- 
- 
- 
- 
-

# Outline

- 1 Background
- 2 Results
- 3 Conclusions

# Outline

- 1 Background
  - background
  - Developmental plasticity in EC

- 2 Results

- 3 Conclusions

# Developmental plasticity in biology

Same genome can lead to different physical structures or behavior depending on environmental factors.

E.g., color of hornworm caterpillars depends on temperature during development.

Plasticity allows development to respond to environmental changes, adding flexibility.



Sam Fraser-Smith  
<http://tr.im/pq71>

# Developmental plasticity in EC

Most EC systems have no (or trivial) developmental processes.

- Therefore can't have developmental plasticity

There are important exceptions. In GP, e.g.:

- Cellular encoding
- Many grammar-based systems
- DTAG3P

These remain, however, the exception rather than the rule.

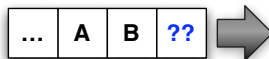
N-gram GP has natural developmental process, so a good candidate for adding developmental plasticity.

# Generating programs with N-gram GP

N-gram GP uses a *probability table* to store likelihood of a triple of instructions appearing in a program:

$$Pr\{x_i \rightarrow x_{i+1} \rightarrow x_{i+2}\}.$$

Given pair of instructions  $(x_i, x_{i+1})$ , this table gives us the probability distribution for the subsequent instruction  $x_{i+2}$ .



...	A	B	C	p = 0.2
-----	---	---	---	---------

...	A	B	D	p = 0.5
-----	---	---	---	---------

...	A	B	E	p = 0.3
-----	---	---	---	---------

# Outline

## 1 Background

## 2 Results

- Empirical comparison of IFD, N-gram GP, and standard GP
- Modularity and repeated structures in IFD

## 3 Conclusions



# Empirical comparison of IFD, N-gram GP, & TinyGP

Compare IFD, regular N-gram GP, and standard sub-tree XO GP (TinyGP)

- 11 different symbolic regression problems
- 100 independent runs for each system + problem + parameter set
- Various parameter settings (e.g., different block sizes)

2 register machine with  $+$ ,  $-$ ,  $\times$ , protected division, and swap

Normalize the clock:

- Count instruction executions
- Allow 50M instruction evaluations per run
- Store machine state so only new block has to be executed in IFD

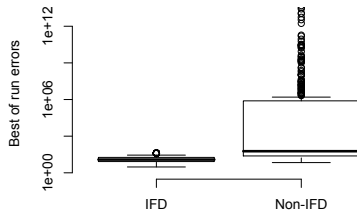
# Success rates on 11 test problems

Label	Function	<i>Successes out of 100 runs</i>		
		TinyGP	N-gram	IFD
P1	$x + x^2 + x^3 + x^4 + x^5$	100	100	100
P2	$-x - 2x^2 + x^3$	100	100	100
P3	$1.009 + 1.419x + x^2$	100	61	100
P4	$6 + x^2 + 3x^3 + 8x^5$	0	0	0
P5	6	100	100	100
P6	$6 + x^2$	100	10	94
P7	$6 + x^2 + 3x^3$	85	0	1
P8	$8x^5$	100	100	100
P9	$3x^3 + 8x^5$	22	55	100
P10	$x^2 + 3x^3 + 8x^5$	100	7	80
Sine	$\sin(x)$	0	1	63

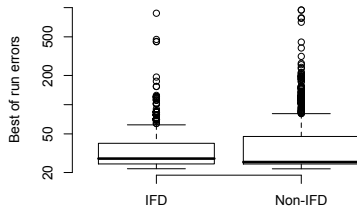
# IFD wins either way

- IFD generates low-error individuals from tables evolved with IFD **and without IFD**.
- IFD's local search is valuable in all phases of the process, even if it wasn't used previously.
- N-gram GP isn't able to work effectively with the more complex probability tables that IFD generates.

Errors for programs generated from IFD matrix using both methods

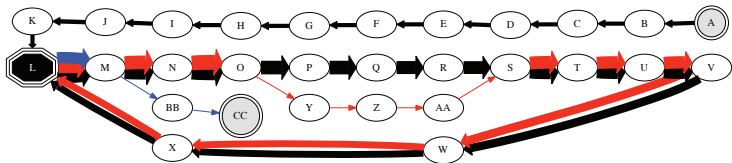


Errors for programs generated from Non-IFD matrix using both methods

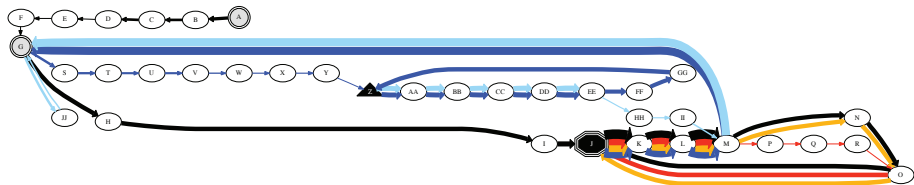


# Structural differences and modularity

Standard N-gram GP tends to converge to a small set of loops with high probability edges.



With IFD there is less convergence, more variety and complexity in the modular structure, & greater use of low probability edges.



# Outline

- 1 Background
- 2 Results
- 3 Conclusions

# Conclusions

- Added developmental plasticity to N-gram GP using Incremental Fitness-based Development (IFD).
- IFD consistently improved N-gram GP performance on suite of test problems.
- “Knocking out” IFD shows it’s valuable in all phases, even if it wasn’t used earlier in a run.
- IFD generates more complex, less converged probability tables.
- IFD generates more modules/loops & uses more low-probability paths.
- Currently exploring applications to dynamic environments.

# Thanks!

Thank you for your time and attention!

Contact:

- `mcphee@morris.umn.edu`
- `http://www.morris.umn.edu/~mcphee/`

## Questions?

# References



N. F. McPhee, E. Crane, S. Lahr, and R. Poli.  
Developmental Plasticity in Linear Genetic Programming.  
In Günther Raidl, *et al*, editors, *GECCO '09*, pages 1019–1026,  
Montréal, Québec, Canada, 2009.



R. Poli and N. McPhee.  
A linear estimation-of-distribution GP system.  
In M. O'Neill, *et al*, editors, *EuroGP 2008*, volume 4971 of *LNCS*,  
pages 206–217, Naples, 26-28 Mar. 2008. Springer.

See the GECCO '09 paper for additional references.