Bloat Control in Genetic Programming with a Histogram-based Accept-Reject Method

Marc-André Gardner Christian Gagné Marc Parizeau
Laboratoire de vision et systèmes numériques
Département de génie électrique et de génie informatique
Université Laval, Québec (Québec), Canada G1V 0A6
marc-andre.gardner.1@ulaval.ca, {christian.gagne, marc.parizeau}@gel.ulaval.ca

ABSTRACT

Recent bloat control methods such as dynamic depth limit (DynLimit) and Dynamic Operator Equalization (DynOpEq) aim at modifying the tree size distribution in a population of genetic programs. Although they are quite efficient for that purpose, these techniques have the disadvantage of evaluating the fitness of many bloated Genetic Programming (GP) trees, and then rejecting most of them, leading to an important waste of computational resources. We are proposing a method that makes a histogram-based model of current GP tree size distribution, and uses the so-called accept-reject method for generating a population with the desired target size distribution, in order to make a stochastic control of bloat in the course of the evolution. Experimental results show that the method is able to control bloat as well as other state-of-the-art methods, with minimal additionnal computational efforts compared to standard tree-based GP.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; G.3 [**Probability and Statistics**]: Probabilistic algorithms (including Monte Carlo)

Keywords

Genetic Programming, Bloat Control, Acceptance Sampling

General Terms

Performance

The Histogram-based Accept-Reject Method for Genetic Programming (HARM-GP) is inspired from distribution-based bloat control techniques such as DynOpEq [4], but takes a different approach for defining the target distribution and evaluating fitnesses. Indeed, the target distribution with HARM-GP is defined from the size distribution of individuals at the previous generation, with a cut-off point determined from the best-so-far individual size, for reducing the frequency of large bloated individuals in the population. The cut-off point is evaluated at each generation as being half-way the current size of the best-so-far individual and the cut-off value of the previous generation. The target distribution (w_i) at the right of the cut-off point is given by an exponential decay function parameterized by a τ parameter corresponding to the size increase ($x_i = x_0 + \tau$) with half the frequency of the cut-off (w_0):

$$w_i = w_0 \exp\left[-\ln(2) \cdot \frac{x_i - x_0}{\tau}\right]. \quad (1)$$

Moreover, the histogram of the source distribution is obtained with kernel density estimation, using a triangular kernel. Figure 1 presents an example of a source (boxes) and target histograms (shaded area) obtained with this method.

In HARM-GP, a new population is produced by the classical accept-reject method, a well-known Monte Carlo approach commonly used for generating random numbers for arbitrary distributions [2]. An accept probability is computed for each individual using the ratio between the target and source distribution values for the size of the individuals. Individuals are generated by crossover, mutation, and reproduction from the previous population and are tested for acceptation with this probability until the requested population size is obtained, without evaluating the candidates fitness. Therefore, in opposition to DynLimit [3] and DynOpEq, HARM-GP induces no supplementary fitness evaluations compared to standard GP.

Experiments are made over six bloat control approaches:

NoLimit: standard GP without any bloat control;

DepthLimit: static depth limit [1] of 17;

DynLimit: dynamic depth limit [3], with initial limit of 6;

DynOpEq: with bin width of 5 [4], except for Symbolic Regression where a bin width of 1 is used;

HARM5: HARM-GP with a small half-life of $\tau = 5$, for strong bloat control;

HARM40: HARM-GP with a larger half-life of $\tau = 40$, leading to softer bloat control.

Experiments were performed using DEAP[1] on runs with population size of 1000 individuals, crossover probability of 0.8, subtree mutation probability of 0.1 on the Symbolic Regression, Artificial Ant, and Parity-6 problems [1]. Tournament selection with 5 participants is used for the Symbolic Regression and Parity-6 problems, with runs stopped after a budget of 80k fitness evaluations. For the Artificial Ant, 100k evaluations are conducted, with 7 participants to tournaments. Figure 2 plots three graphs for the Artificial Ant problem. Table 1 presents a set of detailed measurements carried out on all three problems tested with the six bloat methods we compared. Results show that HARM-GP is able

[1]Freely available at `http://deap.googlecode.com`.

GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

Table 1: Experimental results. Third to fifth columns present results computed on successful runs only. Mean overall accumulated size corresponds to the total number of nodes processed in the population when the maximum number of fitness evaluation is reached.

Bloat control method	Successful runs	Mean number of evaluations for success	Mean successful individual size	Mean accumulated size before success	Mean overall accumulated size
Symbolic Regression					
NoLimit	87%	12 320	22.1	221 634	20 203 244
DepthLimit	85%	12 517	22.0	220 897	4 902 013
DynLimit	93%	14 835	18.8	235 510	2 152 695
DynOpEq	74%	50 648	21.8	696 070	1 391 389
HARM5	99%	13 162	14.8	125 011	1 081 575
HARM40	88%	14 991	18.4	280 149	3 019 076
Artificial Ant					
NoLimit	36%	15 265	55.2	1 058 498	38 038 917
DepthLimit	29%	12 103	43.3	671 771	13 445 970
DynLimit	28%	25 213	46.6	1 373 064	7 471 147
DynOpEq	40%	37 368	51.0	3 064 943	18 704 826
HARM5	39%	16 396	18.3	272 691	1 615 314
HARM40	46%	22 278	35.6	816 279	4 057 703
Parity-6					
NoLimit	97%	12 607	52.3	896 412	26 772 717
DepthLimit	89%	13 127	50.6	698 269	8 769 320
DynLimit	92%	12 148	40.3	438 914	3 954 319
DynOpEq	94%	35 339	34.1	955 259	2 853 804
HARM5	79%	25 728	17.6	369 804	1 173 898
HARM40	96%	16 634	29.4	589 192	3 137 083

of the same success rate as other methods (similar or better performances), while keeping tight control over accumulated size (less computation).

Acknowledgements

We acknowledge financial support from NSERC (Canada) and FQRNT (Québec), and access to supercomputing facilities of CLUMEQ/Compute Canada.

1. REFERENCES

[1] J. R. Koza. *Genetic programming: on the programming of computers by means of natural selection*. MIT press, 1992.

[2] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*. Wiley-Interscience, 2008.

[3] S. Silva and E. Costa. Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories. *Genetic Programming and Evolvable Machines*, 10(2):141–179, 2009.

[4] S. Silva and L. Vanneschi. Operator equalisation, bloat and overfitting: a study on human oral bioavailability prediction. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO 2009)*, 2009.

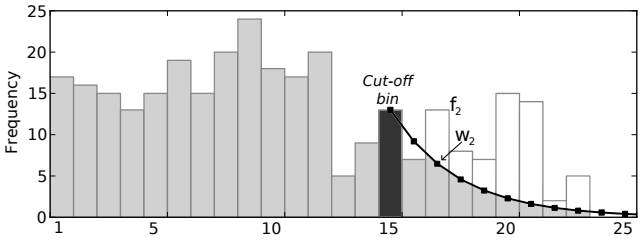

Figure 1: HARM-GP source and target histograms.

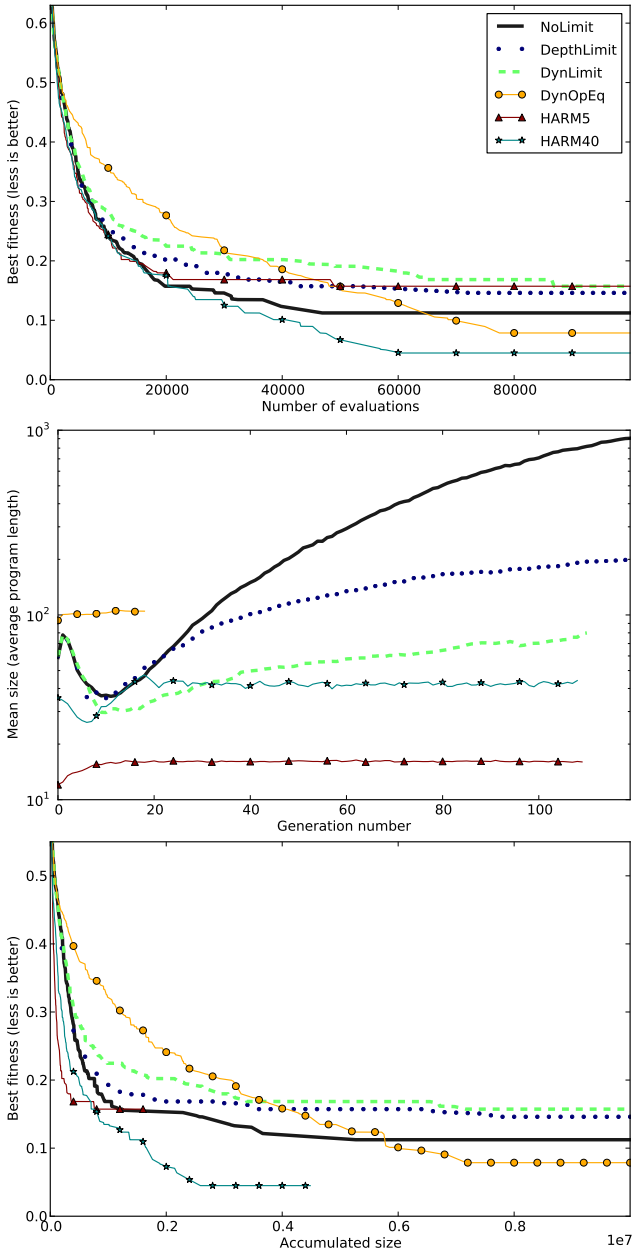

Figure 2: Size and fitness for the Artificial Ant problem: (top) best fitness median (over the 100 runs) according to the number of fitness evaluations; (middle) mean size median at each generation; and (bottom) best fitness median against accumulated size.