# Wave: A Genetic Programming Approach to Divide and Conquer


David Medernach
BDS Group University of Limerick
david.medernach@ul.ie

Jeannie Fitzgerald
BDS Group
University of Limerick
jeannie.fitzgerald@ul.ie

R. Muhammad Atif Azad
BDS Group
University of Limerick
atif.azad@ul.ie

Conor Ryan
BDS Group
University of Limerick
conor.ryan@ul.ie



## ABSTRACT

This work introduces *Wave*, a divide and conquer approach to GP whereby a sequence of short, and dependent but potentially *heterogeneous* GP runs provides a collective solution; the sequence akins a *wave* such that each short GP run is a *period* of the wave. Heterogeneity across periods results from varying settings of system parameters, such as population size or number of generations, and also by alternating use of the popular GP technique known as *linear scaling.*



## Categories and Subject Descriptors

D.1.2 [**Programming Techniques**]: Automatic Programming; I.2.6 [**Artificial Intelligence**]: Learning—*Induction*

## Keywords

Genetic algorithms; Genetic programming; Semantic GP


## 1. BACKGROUND

*Sequential Symbolic Regression* [3] (SSR) spreads the task of approximating training data across a number of GP runs, where each such run is termed an *iteration.* At the end of each iteration, outputs of the original problem are modified based on the use of a geometric semantic crossover [2] on the output of the best evolved solution in the current iteration. In the next iteration SSR evolves the best match to the current solution; however, each iteration is homogeneous and typically uses a large number of generations.

Outside the field of GP a similar method is *cascade correlation (CasCor)* [5], where hidden layers are added and trained to reduce the residual error of a previously trained artificial neural network.




*GECCO '15 July 11-15, 2015, Madrid, Spain*





ACM ISBN 978-1-4503-3488-4/15/07.
DOI: http://dx.doi.org/10.1145/2739482.2764659


## 2. WAVE

Instead of the traditional GP approach where each GP run consists of a potentially large number of generations, Wave (see Figure 1) uses a collection of heterogenous periods, and simply sums their best results. At the end of each period, $t$ being the target value at the start of this period, and $f_i$ being the best evolved function for this period, if $(f_i - t)^2 < (0-t)^2$ we add $f_i$ to the joint solution; and $t'$ is the new data-set for the next period such that $t' = t - f_i$; else we launch the next period to again optimise over the same target set $t$. The next period starts afresh with a new population. Each period stops when it ceases to produce a significant improvement and therefore match the following condition: $g_c > g_m$ and $(B^F(g_c) - B^F(g_c - 2)) \leq (B^F(g_c - 2) - B^F(g_c - 5))/200$ where $g_c$ is the current generation, $g_m$ is a minimum number of generations before a period stops and $B^F(g_c)$ is the best training fitness at generation $g_c$. We end a wave when 25 periods have been processed.

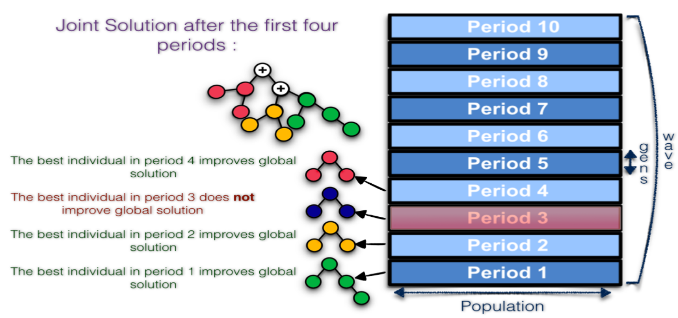


Figure 1: A simple Wave setup is depicted.

Different *periods* of Wave can use different GP settings. Thus, we observe the effect of increasing the population size and the minimum number of generations before stopping a period. We also report experiments where periods alternate between use and non use of linear scaling.

## 3. EXPERIMENTS

For this study we have used two multi-dimensional data-sets from the UCI Machine learning repository [1] (**Concrete**

**Strength** and **Yacht**) and two mathematical functions (**Poly-10** [**4**] $y = x1 * x2 + x3 * x4 + x5 * x6 + x1 * x7 * x9 + x3 * x6 * x10$ and **Div-5** [**6**] $y = \frac{10}{5+\sum_{i=1}^{5}(x_i-3)^2}$). For each mathematical function, we randomly generate 500 data points in the range [0; 1] and for each wave we randomly split the given data-set into two subsets of equal size for training and testing purposes.

We compare Wave with standard GP (both with and without linear scaling; with a population size of 500; each run spans 100 generations) and a non-EC method: multiple linear regression (MLR). We run MLR 100 times, splitting the data randomly into equal partitions for training and testing. At each generation, each individual's fitness is computed both on testing and training data. We adopt for the following naming convention: $Wave : PeriodsNumber : Setting - P : PopulationSize$.

## 4. RESULTS AND DISCUSSION

We measure the various run statistics at the end of each period for Wave experiments and every ten generations for SGP; we call those reporting times *moments*. For each moment, of each data-set, of each setting, median testing and training values are computed over 100 independent runs.

In tables 1-4 we report the best moment (lowest test fitness) for each setting on each data-set. Additionally, we report the **Fastest Good Wave**; this is a Wave set up which outperforms the best SGP moment on test fitness and consumes fewer nodes than any other Wave. We use the Mann-Whitney U test at p = 0.05 to test the statistical significance.

| Method | Moment | Train | Test | Nodes |
|---|---|---|---|---|
| SGP:LS-P:500 | 71g | 5.16411 | 6.21251 | 2468926 |
| SGP:NS-P:500 | 81g | 4.09004 | **4.48998** | 3410695 |
| Wave:25:LS-P:100 | 23p | 7.42025 | 7.87332 | 2056251 |
| Wave:25:NS-P:100 | 22p | 4.01469 | 5.41762 | 999693 |
| Wave:200:LS-P:25 | 200p | 9.02013 | 9.30088 | 450465 |
| Wave:25:LS-P:500 | 25p | 6.48936 | 7.08987 | 3229500 |
| Wave:25:NS-P:500 | 9p | 3.38198 | 4.73403 | 2464113 |
| Wave:25:LS:NS-P:500 | 16p | 3.35759 | **4.50582** | 4342475 |
| MLR | NA | 8.86019 | 9.11015 | NA |
| **Fastest Good Wave :** | | | | |
| NA | | | | |

Table 1: Experimental results on Yacht dataset.

| Method | Moment | Train | Test | Nodes |
|---|---|---|---|---|
| SGP:LS-P:500 | 51g | 14.1436 | 16.1199 | 671274 |
| SGP:NS-P:500 | 81g | 14.6654 | **14.8268** | 3080889 |
| Wave:25:LS-P:100 | 25p | 14.3886 | 14.5569 | 3139598 |
| Wave:25:NS-P:100 | 25p | 10.3821 | 11.3939 | 970429 |
| Wave:200:LS-P:25 | 12p | 16.2683 | 16.3699 | 402010 |
| Wave:25:LS-P:500 | 21p | 13.8497 | 13.9208 | 1753484 |
| Wave:25:NS-P:500 | 14p | 8.72476 | 10.1065 | 4418211 |
| Wave:25:LS:NS-P:500 | 24p | 7.71554 | **8.98308** | 7463434 |
| MLR | NA | 10.3123 | 10.5693 | NA |
| **Fastest Good Wave :** | | | | |
| Wave:25:LS;p:500 | 2 p | 14.4465 | 14.5430 | 133466 |

Table 2: Experimental results on Concrete dataset.

The Fastest Good Wave results show that Wave not only achieves significantly better training fitness, but also produces at least equivalent testing fitness in three of the four data-sets, with significantly fewer node evaluations.
The setting which emerges as the most efficient is that which alternates use of linear scaling (Wave:25:LS:NS-P:500). This outperforms all chosen benchmarks on three out of the four data-sets, and among Wave setups, is only outperformed by Wave:25:LS-P:500 on the poly-10 problem. The heterogenity within the wave proves particularly useful because linear scaling does not always produce the best results. A mixed approach using the Wave paradigm appears to be a more flexible strategy that produces consistently good results.

| Method | Moment | Train | Test | Nodes |
|---|---|---|---|---|
| SGP:LS-P:500 | 1g | 0.47447 | 0.47591 | 11515 |
| SGP:NS-P:500 | 71g | 0.37882 | **0.38868** | 383535 |
| Wave:25:LS-P:100 | 7p | 0.35781 | 0.45138 | 32134 |
| Wave:25:NS-P:100 | 25p | 0.20110 | 0.24104 | 2653574 |
| Wave:200:LS-P:25 | 9p | 0.20627 | 0.24712 | 220448 |
| Wave:25:LS-P:500 | 6p | 0.16587 | **0.18736** | 1097015 |
| Wave:25:NS-P:500 | 24p | 0.17971 | 0.22286 | 2824953 |
| Wave:25:LS:NS-P:500 | 10p | 0.16333 | 0.19930 | 1777974 |
| MLR | NA | 0.76773 | 0.77197 | NA |
| **Fastest Good Wave :** | | | | |
| Wave:25:Norm;p:100 | 3 p | 0.35231 | 0.37657 | 51693 |

Table 3: Experimental results on Poly-10 dataset.

| Method | Moment | Train | Test | Nodes |
|---|---|---|---|---|
| SGP:LS-P:500 | 71g | 0.02377 | **0.02395** | 2234774 |
| SGP:NS-P:500 | 81g | 0.05738 | 0.05897 | 2731481 |
| Wave:25:LS-P:100 | 7p | 0.00698 | 0.00999 | 258080 |
| Wave:25:NS-P:100 | 25p | 0.10738 | 0.11704 | 3267500 |
| Wave:200:LS-P:25 | 6p | 0.02829 | 0.03099 | 15358 |
| Wave:25:LS-P:500 | 6p | 0.00442 | 0.00484 | 1050492 |
| Wave:25:NS-P:500 | 23p | 0.07700 | 0.08441 | 2143142 |
| Wave:25:LS:NS-P:500 | 9p | 0.00424 | **0.00480** | 1508355 |
| MLR | NA | 0.71604 | 0.71736 | NA |
| **Fastest Good Wave :** | | | | |
| Wave:25:LS;p:100 | 3 p | 0.01808 | 0.02098 | 78061 |

Table 4: Experimental results on Div-5 dataset.

## 5. REFERENCES


[1] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[2] A Moraglio, K Krawiec, C G. Johnson Geometric Semantic Genetic Programming. *Parallel Problem Solving from Nature, Springer Berlin Heidelberg*, 2012.

[3] LOVB Oliveira, FEB Otero, GL Pappa, J Albinati Sequential Symbolic Regression with Genetic Programming. *Genetic and Evolutionary Computation, Springer*, 2014.

[4] R. Poli. A simple but theoretically-motivated method to control bloat in genetic programming. In *Genetic Programming*, pages 204–217. Springer, 2003.

[5] Prechelt, Lutz Investigation of the CasCor family of learning algorithms. *Neural Networks 10, Elsevier*, pages 885–896, 1997.

[6] E J. Vladislavleva, G F. Smits, and D. Den Hertog. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. *Evolutionary Computation, IEEE Transactions on*, 13(2):333–349, 2009.