UNIVERSITY OF LEEDS

# Enhancements to a hybrid genetic programming technique applied to symbolic regression

U. Armani, V.V. Toropov, A. Polynkin, O. M. Querin,
L. Alvarez

8th July 2010

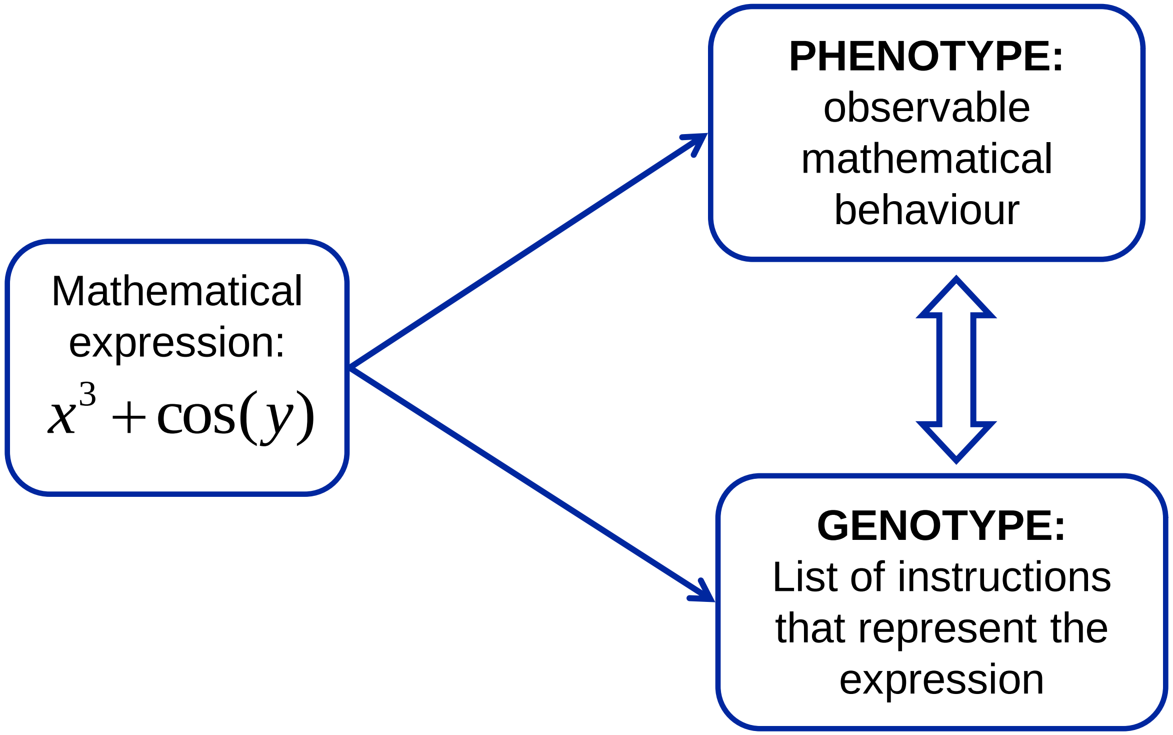
# A brief reminder: tree-based hybrid GP applied to symbolic regression

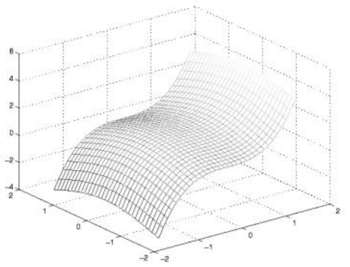
## Symbolic Regression:

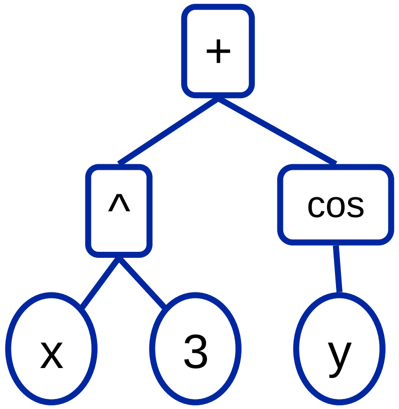
**INPUT DATA** $\longrightarrow$ **MATHEMATICAL EXPRESSION (function)**

## Tree-based GP:

- *individuals are represented by the expressions' syntax trees*
- *genetic modifications are performed on syntax trees*


Mathematical expression:
$x^3 + \cos(y)$
**PHENOTYPE:** observable mathematical behaviour
**GENOTYPE:** List of instructions that represent the expression



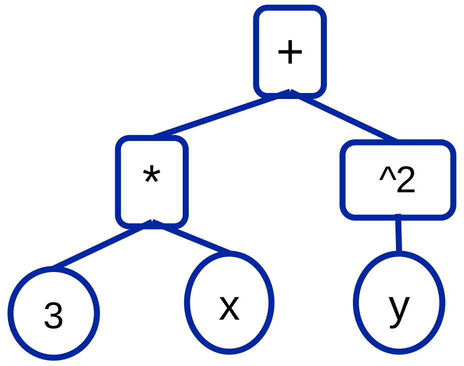


# Conventional GP and Hybrid GP : Genotype

Conventional GP:

• constants are treated like common nodes and do participate in genetic operations.

Genotype = individual

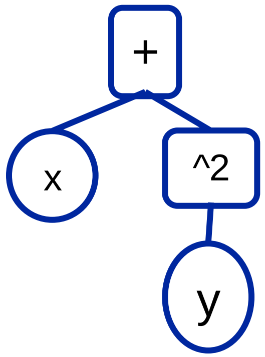


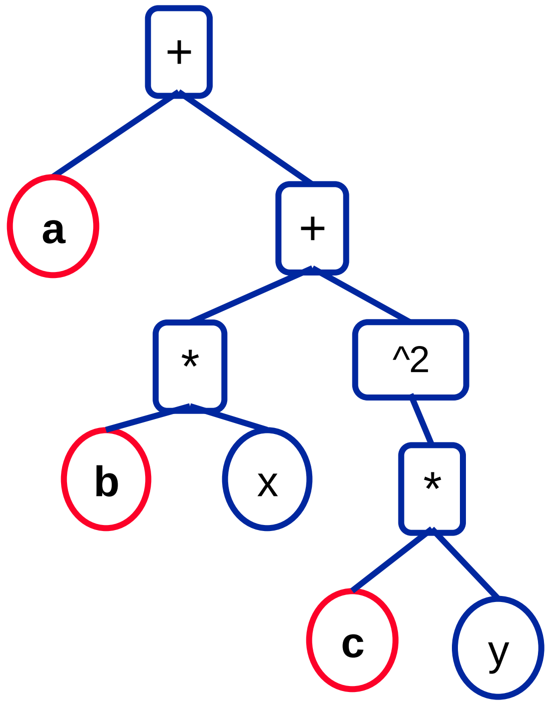
Hybrid GP:

• constants do not participate in genetic operations. They are added afterwards, their location and their values being optimised (see **a**, **b**, **c** below).

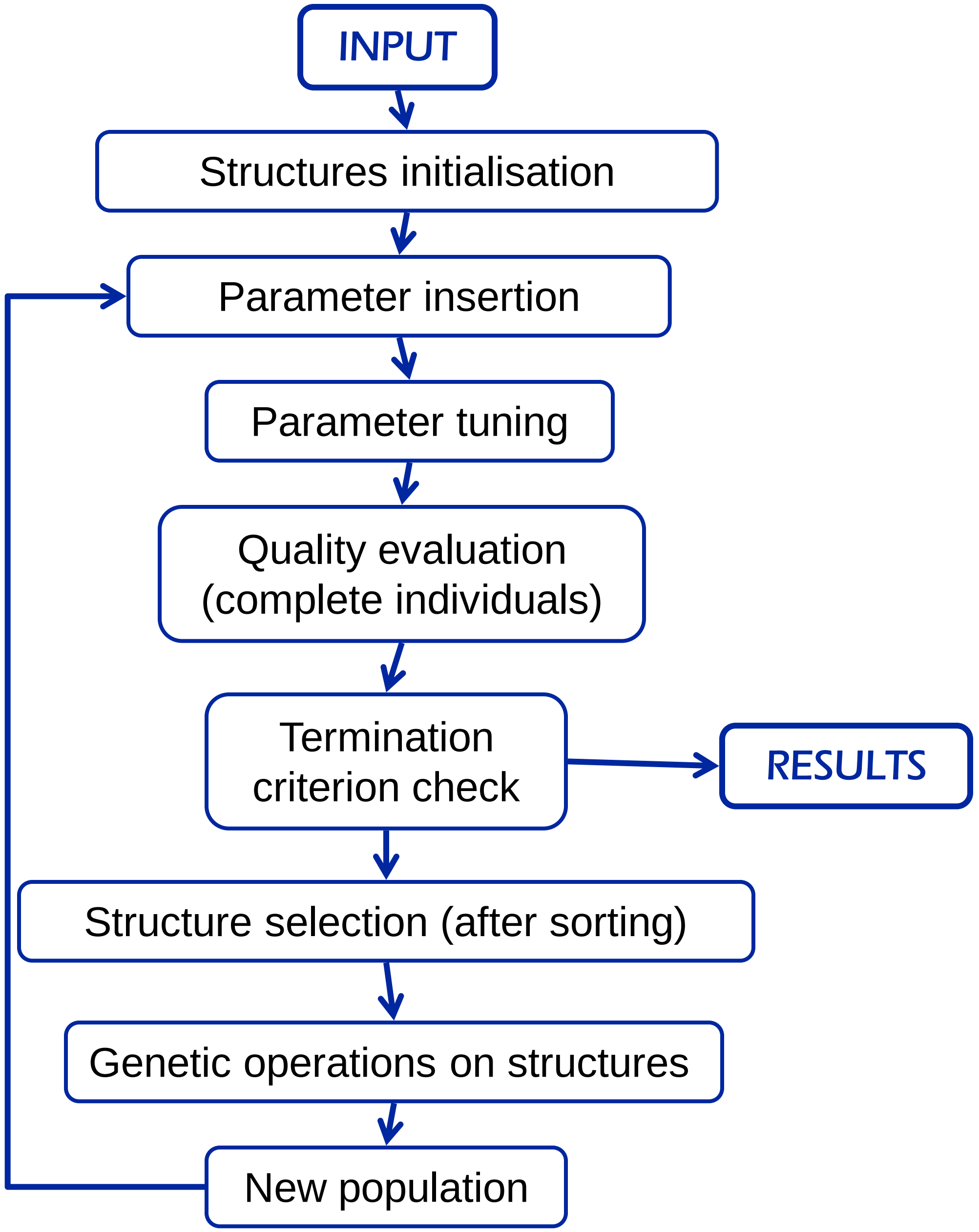
• a "**structure**", or individual without parameters, represents a family of expressions

Genotype ≠ individual





# Hybrid Genetic Programming Algorithm


INPUT
Structures initialisation
Parameter insertion
Parameter tuning
Quality evaluation (complete individuals)
Termination criterion check
RESULTS
Structure selection (after sorting)
Genetic operations on structures
New population

Functions (cos(), ^, +,…), Variables (x,y, ...)
training data set (and validation set)

“Structures”: trees without parameters (ramped half and half method is used)

Deterministic algorithm: the number of parameters is reduced to the minimum

Deterministic algorithm (SQP)

Fitness value: weighted sum of different objectives (RMSE, no of singularities, no of tuning parameters, no of nodes)

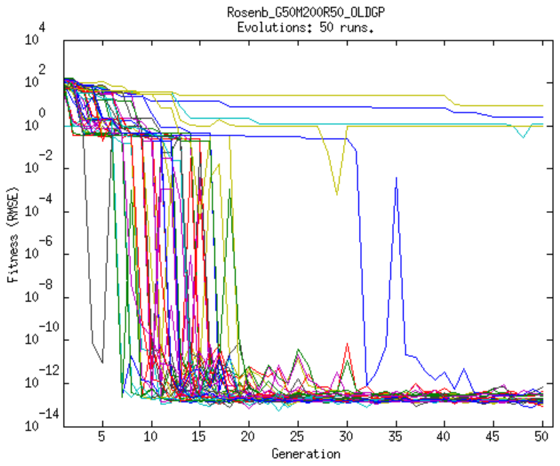
The fitness value is inherited from the corresponding tree with parameters

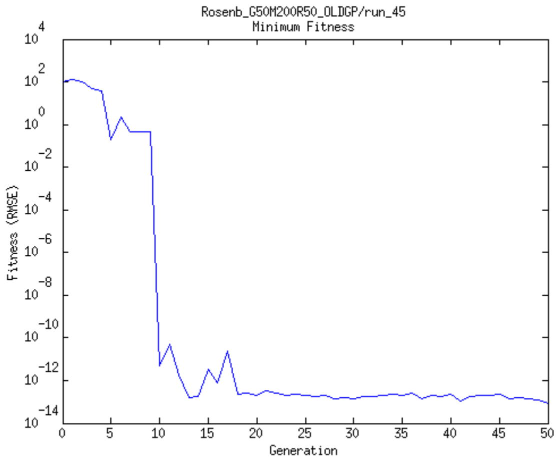
Variation of the structure of the syntax trees (no parameters)

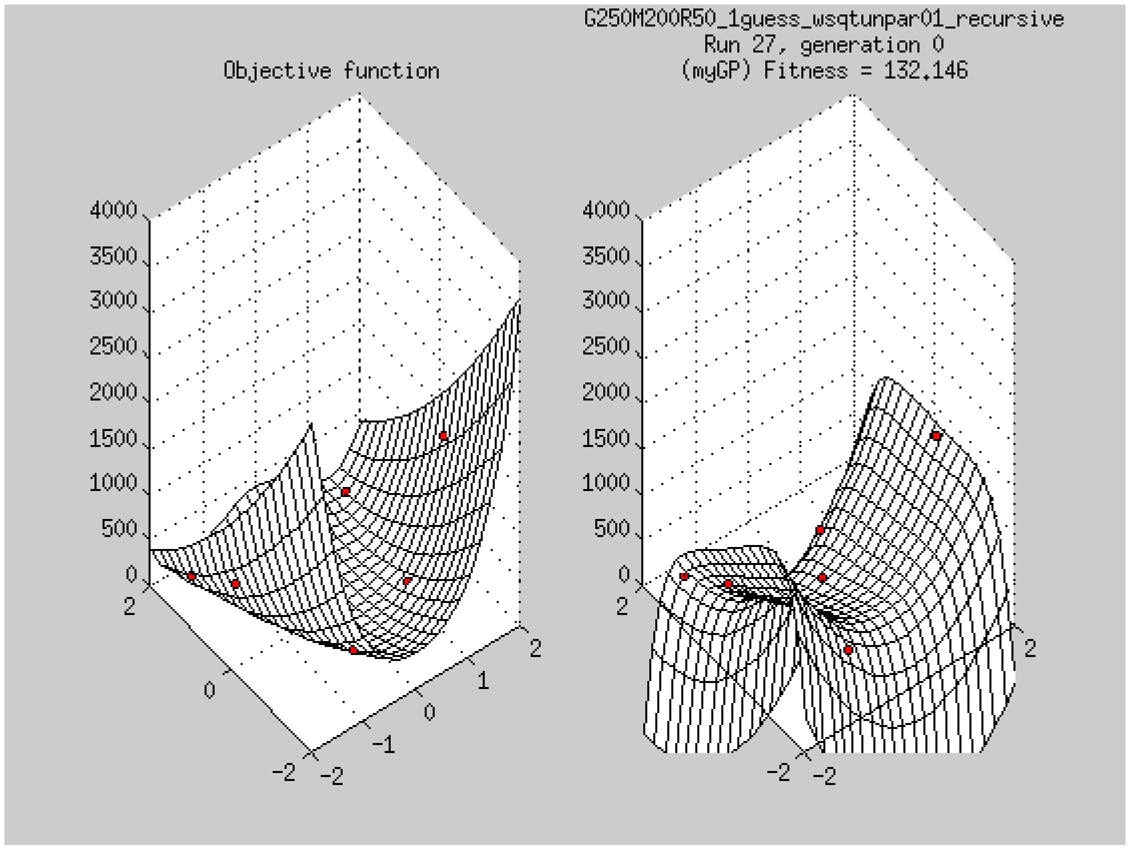
# A few inspiring graphical results

Rosenbrock function: $f(z_1, z_2) = 100\,(z_2 - z_1^2)^2 + (1 - z_1)^2$

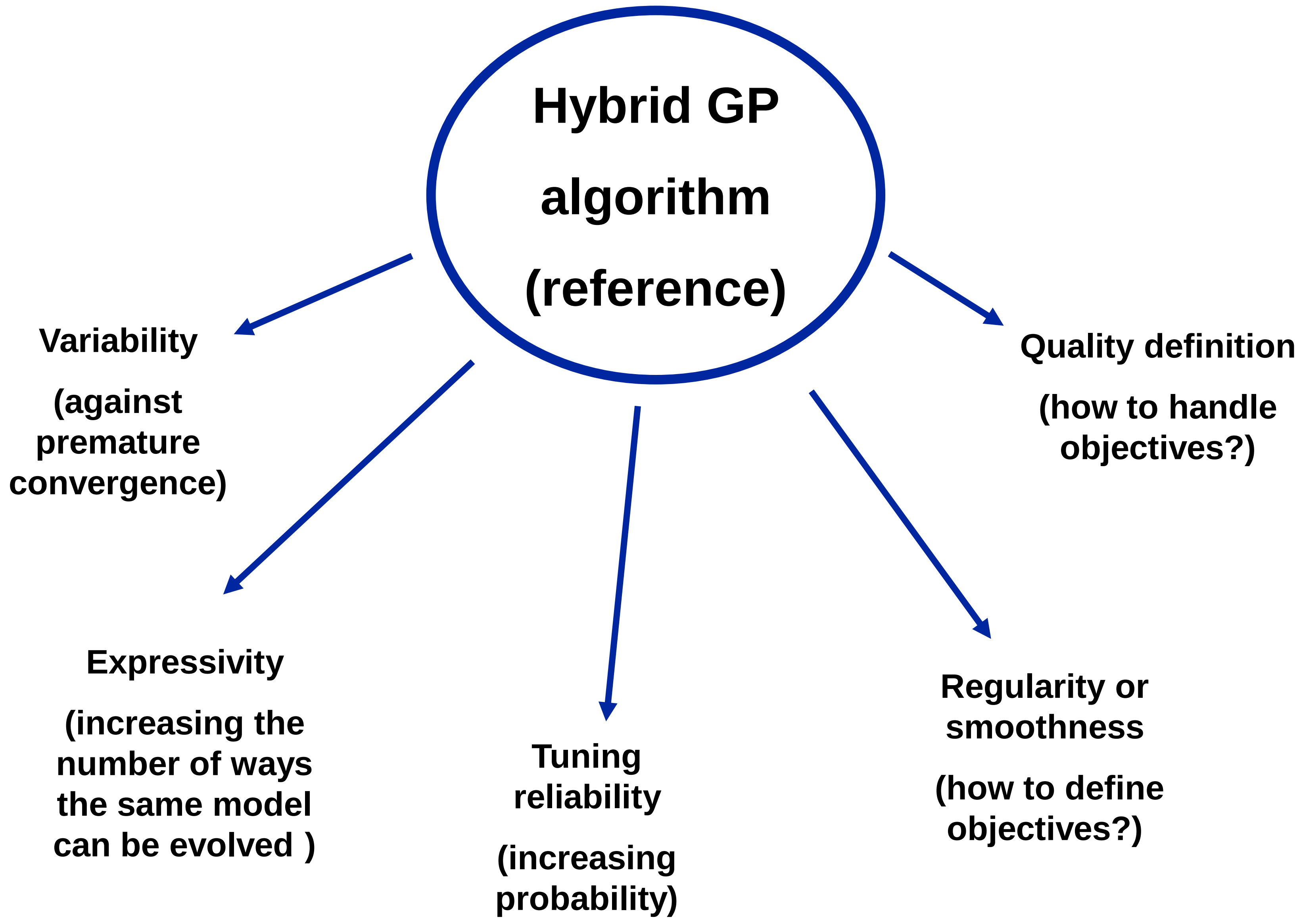
Expression returned by hybrid GP: $1.000000 + 1.000000Z1^2 - 2.000000Z1 + \left(-10.000000Z1^2 + 10.000000Z2\right)^2$







# Can hybrid GP be further improved?


Hybrid GP algorithm (reference)
Variability (against premature convergence)
Expressivity (increasing the number of ways the same model can be evolved )
Tuning reliability (increasing probability)
Regularity or smoothness (how to define objectives?)
Quality definition (how to handle objectives?)

# New hybrid GP implementations – 1

**With_COPIES :** copies are removed from the elite of individuals copied to the new generation

**10guesses :** increased number of initial guesses for SQP algorithm (10 vs. 2 used normally)

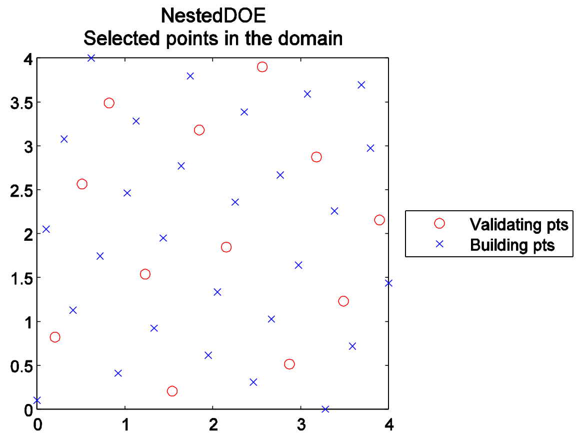
**KILLandFILL :** periodical elimination of 60% of the population. Empty places are filled with newly generated or "joined" individuals

**Shift :** added a new operation that can be selected by GP. It introduces a translation of the argument, increasing the number of ways the same model or surrogate can be built:

$$z_1 * z_2 \Rightarrow a_0 + a_1 z_1 z_2$$

$$z_1 * shift(z_2) \Rightarrow a_0 + z_1(a_1 + a_2 z_2)$$

**NestedDOE :** training data set is split in two subsets: one used for model building, the other for model evaluation. The split is made in order to have optimal latin hypercube distribution in the subsets and in the merged set.



## New hybrid GP implementations – 2

**normFIT :** normalisation of the main objective of the fitness function:

$$F_1(i,t) = RMSNE(i,t) \qquad RMSNE(i,t) = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(\frac{\hat{f}_j(i,t) - f_j}{f_j}\right)^2}$$

**normFITdiv :** as normFIT, by the main objective is divided by the average RMSNE at the previous generation:

$$F_1(i,t) = RMSNE(i,t)/\overline{RMSNE}(t-1)$$

**MinMax :** redefinition of the fitness function:

$$F(i,t) = \max\left\{10a_1F_1(i,t),\ a_2F_2(i,t),\ 10^6F_3(i,t),\ a_4F_4(i,t)\right\}$$
$$a_1 + a_2 + a_3 + a_4 = 1$$

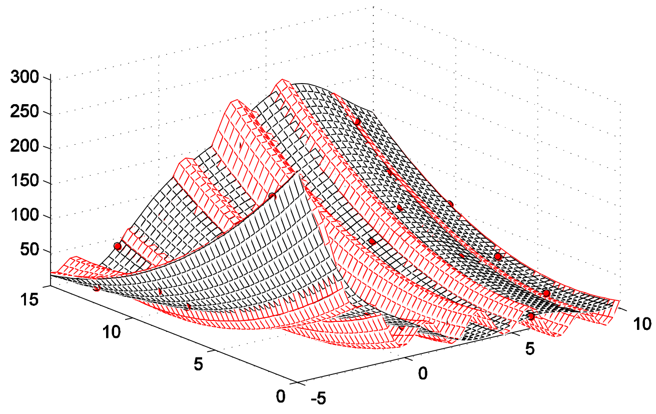
**Omegalim:** an additional algorithm recognises “pulsations” and prevents “high-pulsation” noise:

$$\sin(cz_i) \Rightarrow a_{k,i} = c$$
$$\cos(cz_i) \Rightarrow a_{k,i} = c$$

$$NP_{\max} \longrightarrow \omega_{i,\lim} = \frac{2\pi * NP_{\max}}{r_i}$$

**Penalisation if**

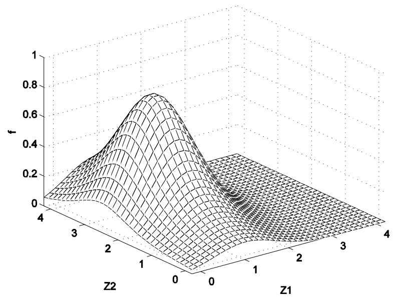
$$a_{i,k} \notin [-\omega_{i,\lim}, \omega_{i,\lim}]$$



## Test problems

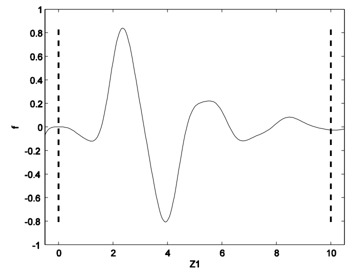
- Kotanchek function:

$$f(x_1, x_2) = e^{-x} x^3 \cos(x)\sin(x)\left[\cos(x)\sin^2(x) - 1\right]$$



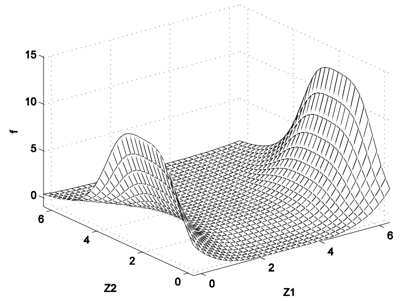
- Salustowicz function:

$$f(x_1, x_2) = e^{-x} x^3 \cos(x)\sin(x)\left[\cos(x)\sin^2(x) - 1\right]$$



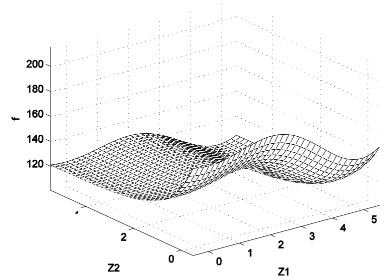
- RatPol2D function:

$$f(x_1, x_2) = \frac{(x_1 - 3)^4 + (x_2 - 3)^3 - (x_2 - 3)}{(x_2 - 2)^4 + 10}$$



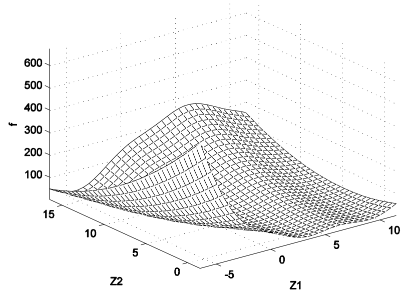
- Hock function:

$$f(x_1, x_2) = \left[30 + x_1 \sin(x_1)\right]\left[4 + \exp(-x_2)\right]$$

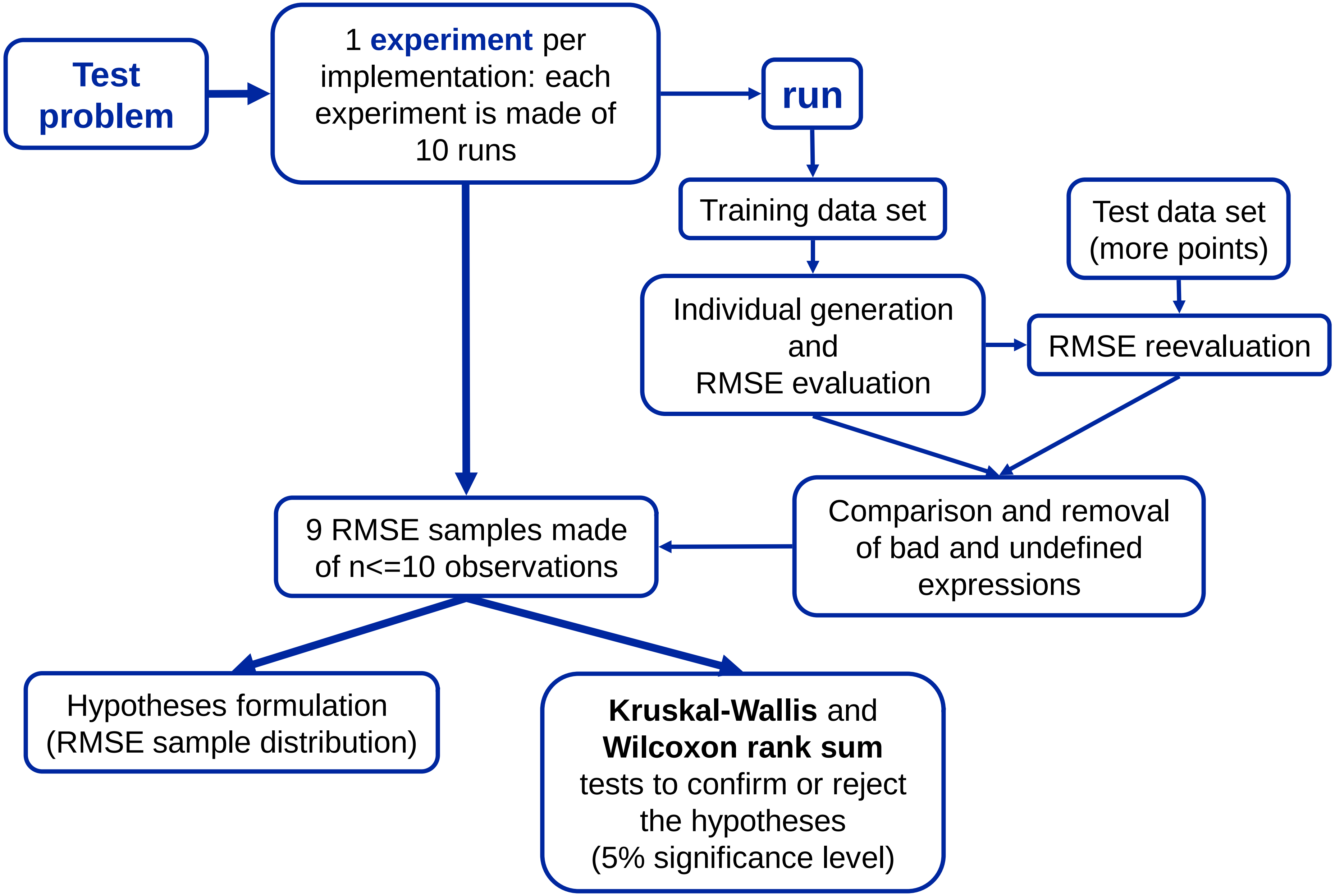


- Branin-Hoo function:

$$f(x_1, x_2) = \left(x_2 - \frac{5.1 x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$$



# How results are analysed


Test problem
1 experiment per implementation: each experiment is made of 10 runs
run
Training data set
Test data set (more points)
Individual generation and RMSE evaluation
RMSE reevaluation
Comparison and removal of bad and undefined expressions
9 RMSE samples made of n<=10 observations
Hypotheses formulation (RMSE sample distribution)
Kruskal-Wallis and Wilcoxon rank sum tests to confirm or reject the hypotheses (5% significance level)

# Are copies beneficial to the evolution?

**P-values from the comparison of Reference** vs. **with_COPIES on the five test problems (Wilcoxon rank sum test):**

| NULL hypothesis: **reference** and **with_COPIES** have the same median | *p*-values | |
|---|---|---|
| | ANOVA | Wilcoxon |
| Kotanchek | 0.9053 | 0.2775 |
| Salustowicz | 0.1806 | 0.6667 |
| RatPol2D | 0.6072 | 0.5490 |
| Hock | 0.1100 | 0.0640 |
| Branin-Hoo | 0.5576 | 0.6064 |

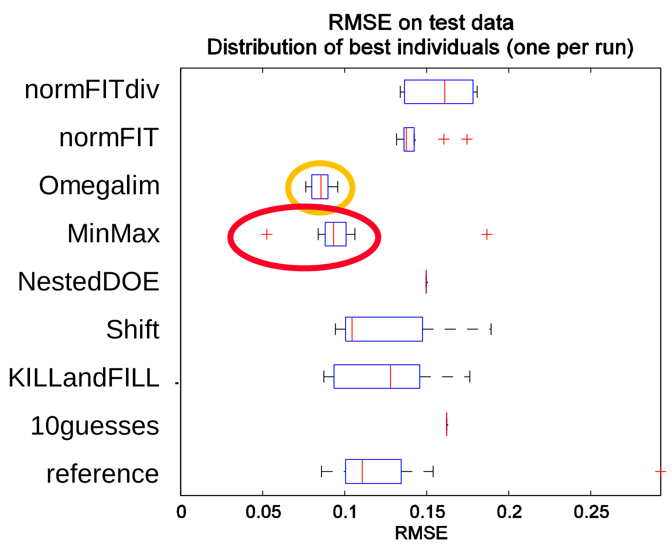
Salustowicz and Branin-Hoo: high percentage (90% and 50%) of individuals generated by **reference** are undefined

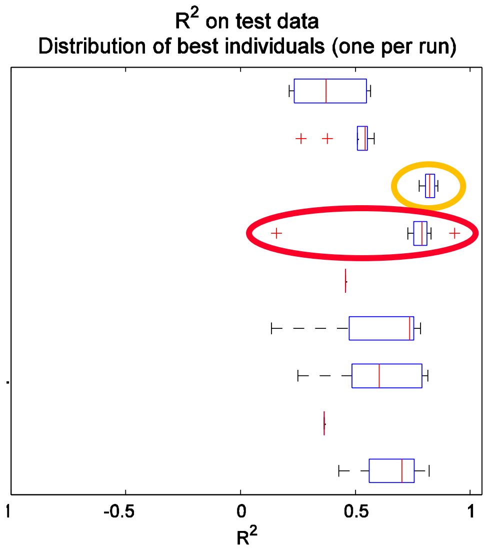
NO significant evidence of a difference in RMSE median according to ANOVA and Wilcoxon rank sum test. Larger samples are needed!

$\longrightarrow$

**However, in all the following experiments copies are removed from the elite!**

# Kotanchek test problem



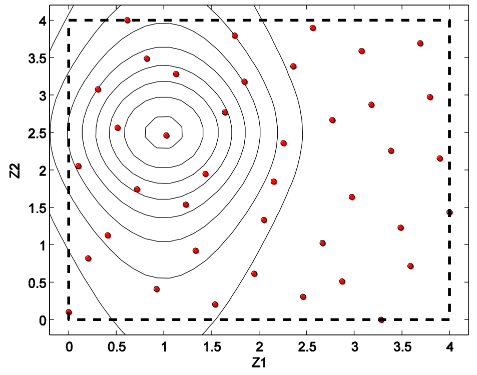


**NestedDOE**: 9 out of 10 expressions not defined on the test data set!

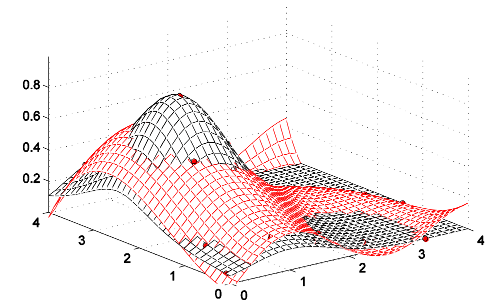
p-value from Kruskal-Wallis = 4.7E-7

Training data set: 40 points

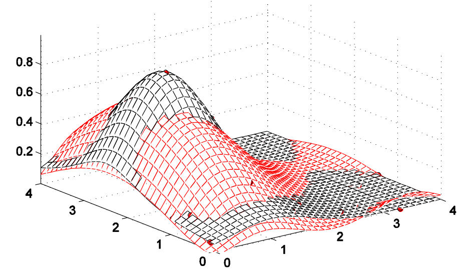
Test data set: 2025 points



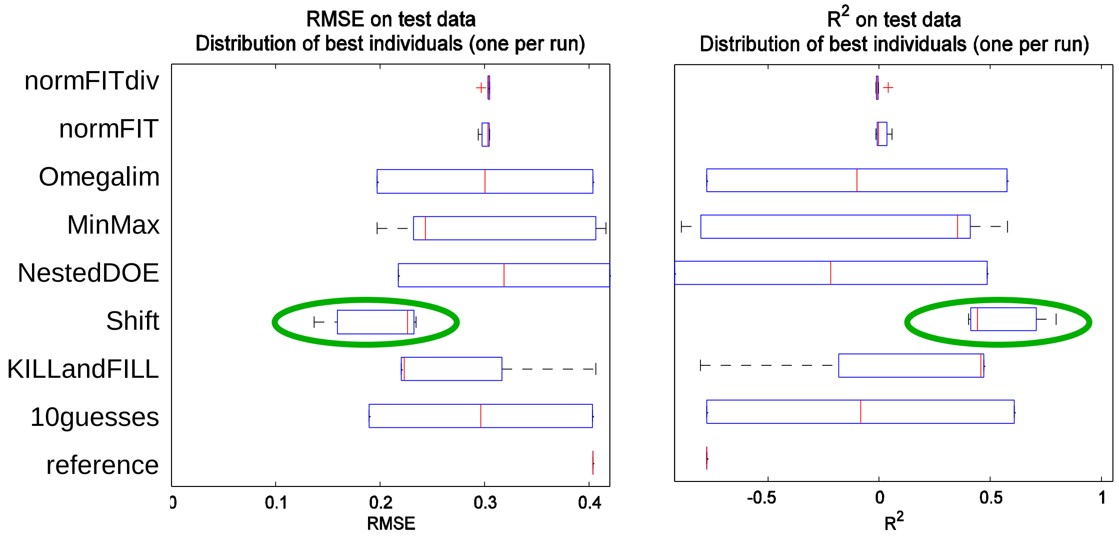
omegalim



MinMax



# Salustowicz problem

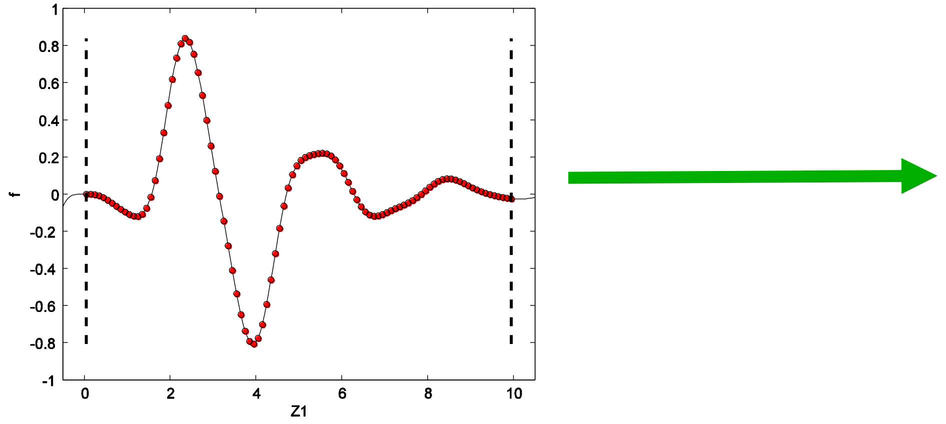


On average 60% of the expressions generated by each implementation are not defined on the test data set!

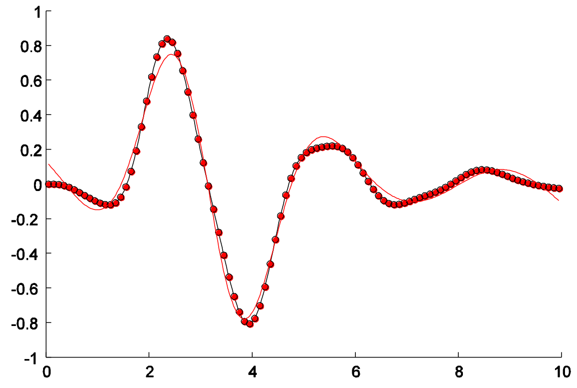
p-value from Kruskal-Wallis = 0.5675

Training data set: 100 points

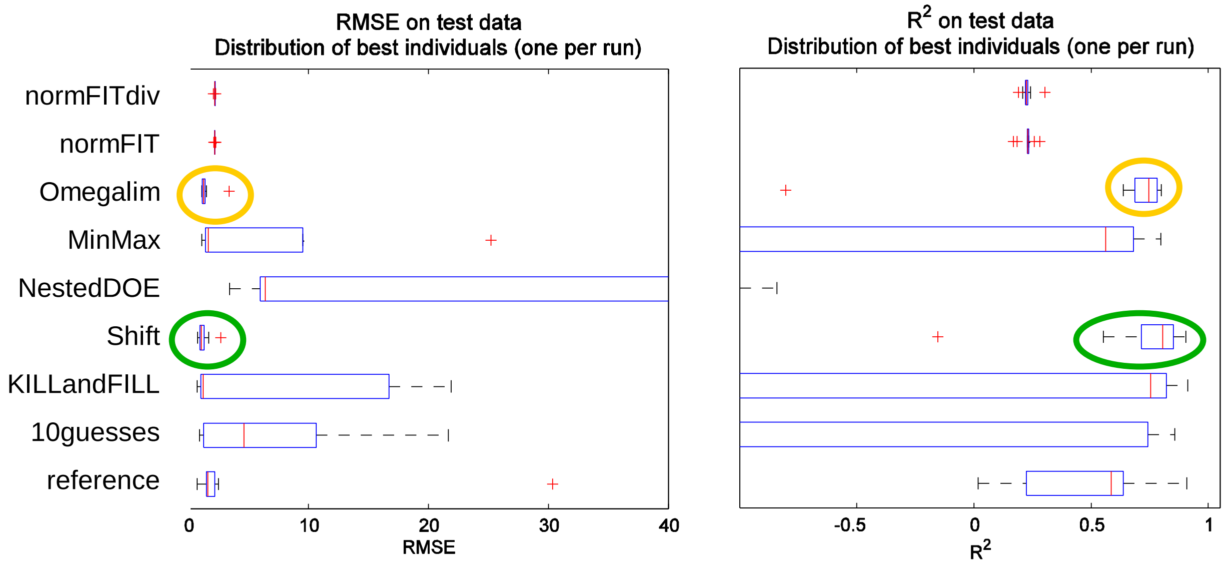
Test data set: 221 points



shift



# RatPol2D problem

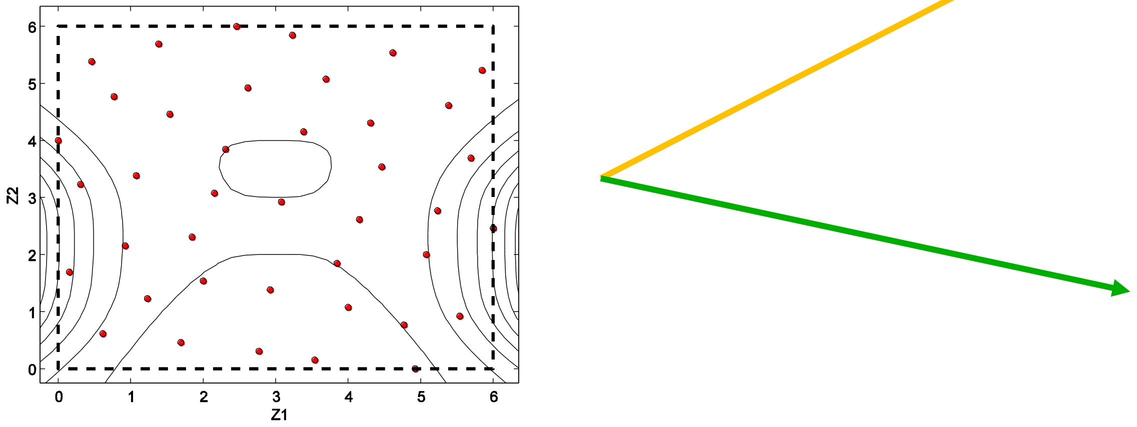


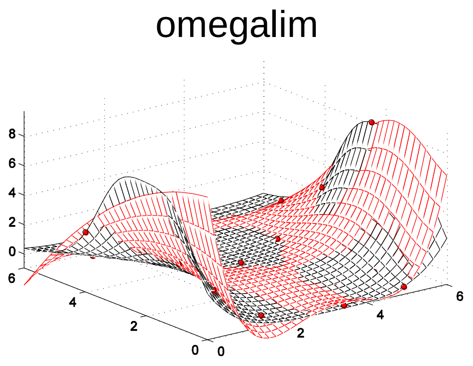
All the expressions generated are defined on the test data set.

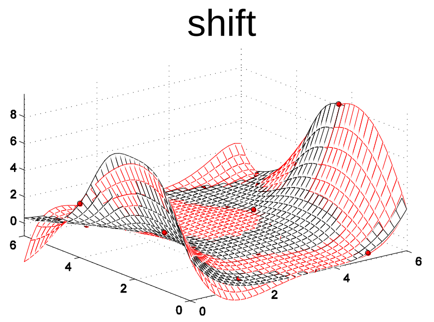
p-value from Kruskal-Wallis = 5.97E-5

Training data set: 40 points

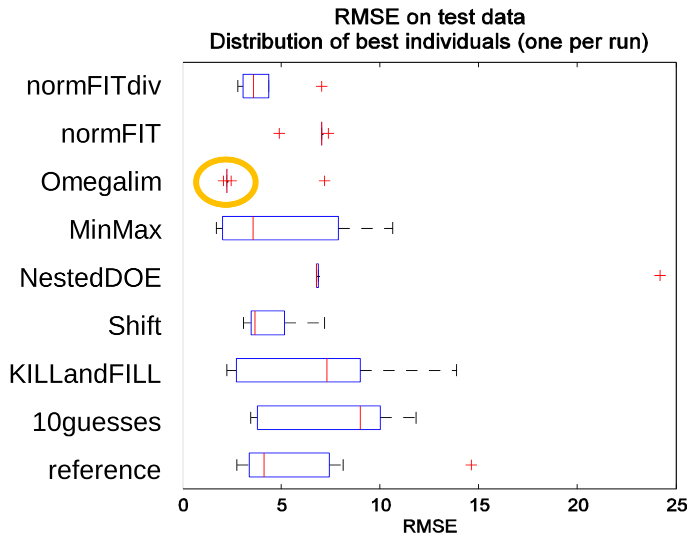
Test data set: 1156 points

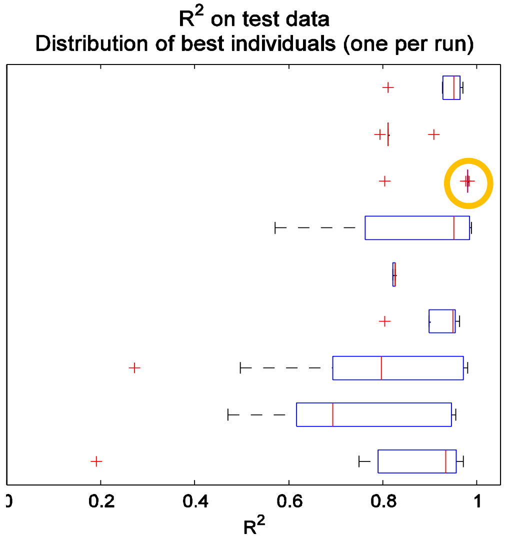






# Hock problem



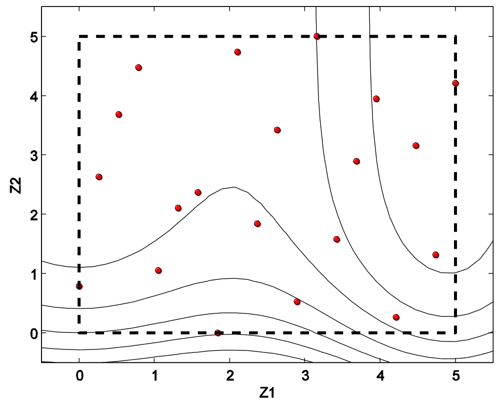


All the expressions generated are defined on the test data set.

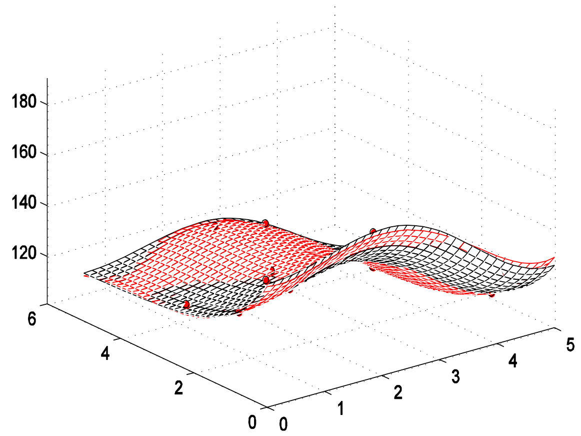
p-value from Kruskal-Wallis = 4.58E-4

Training data set: 20 points

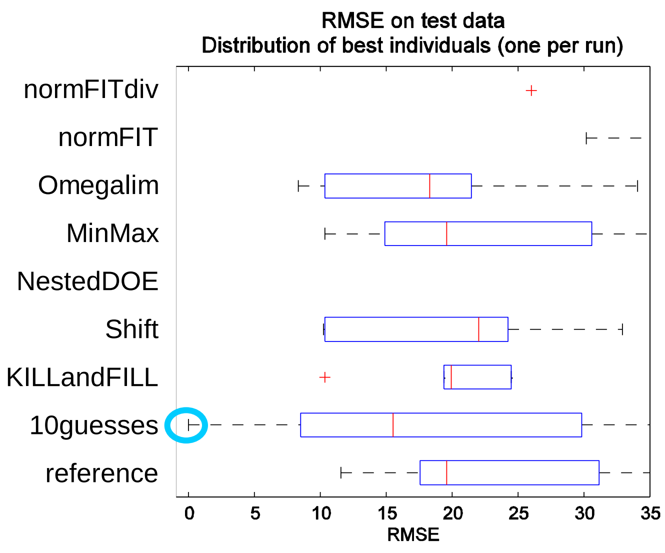
Test data set: 441 points

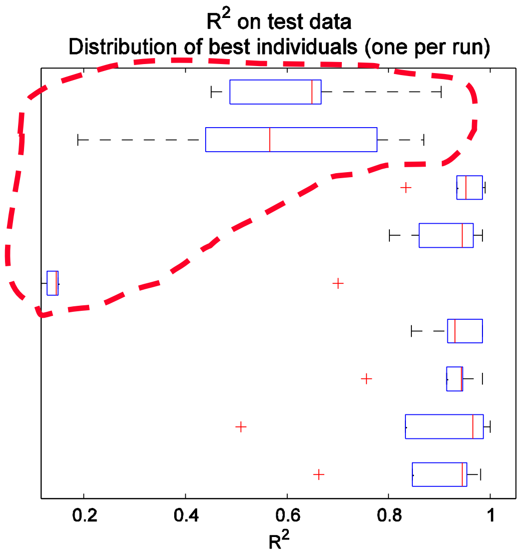


**omegalim**



# Branin-Hoo problem





On average 36% of the expressions generated by each implementation are not defined on the test data set!

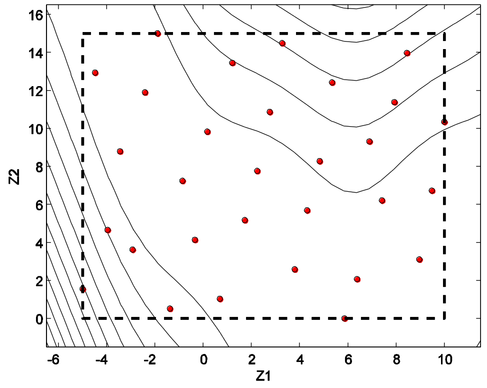
p-value from Kruskal-Wallis = 2.3E-5

↓

**NestedDOE, normFIT** and **normFITdiv** return the worst expressions.

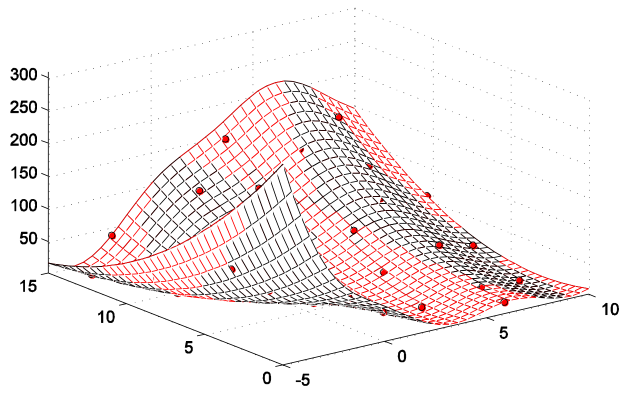
Training data set: 30 points

Test data set: 1369 points



→

**10guesses**



## Conclusions

- For **reference** implementation, there is not a definitive evidence that deleting copies improves the search.
- **NestedDOE**, **normFIT** and **normFITdiv** show poor performances.
- Increasing the number of initial guesses for SQP does not pay off: a huge increase in computational cost does not result in better regression quality.
- **Omegalim** performs consistently better than the other implementations in all the test problems

## Future work:

- Extension of the pulsation control (**omegalim**) to recognise entire expressions as pulsations (interval arithmetics)
- Exploitation of derivative values in fitness evaluation and tuning (smoothing technique)
- Further research to understand why GP encourages the generation of linear combination of primitives as opposed to nested functions (that could explain the general poor performances on Salustowicz test problem)

## A few references…

- Standard GP algorithm:

**J. R. Koza,** "*Genetic programming: on the programming of computers by means of natural selection*" MIT press, Cambridge, MA, USA, sixth edition, 1992

**R. Poli, W. B. Langdon, and N. F. McPhee,** "*A field guide to genetic programming*" published via http://lulu.com and freely available at http://www.gp-field-guide.org.uk, 2008. (With contributions by J. R. Koza)

**W. Banzhaf, Frank D. Francone, Robert E. Keller, and Peter Nordin,** "*Genetic programming: an introduction: on the automatic evolution of computer programs and its applications"* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

- Integration of deterministic methods in standard GP algorithm:

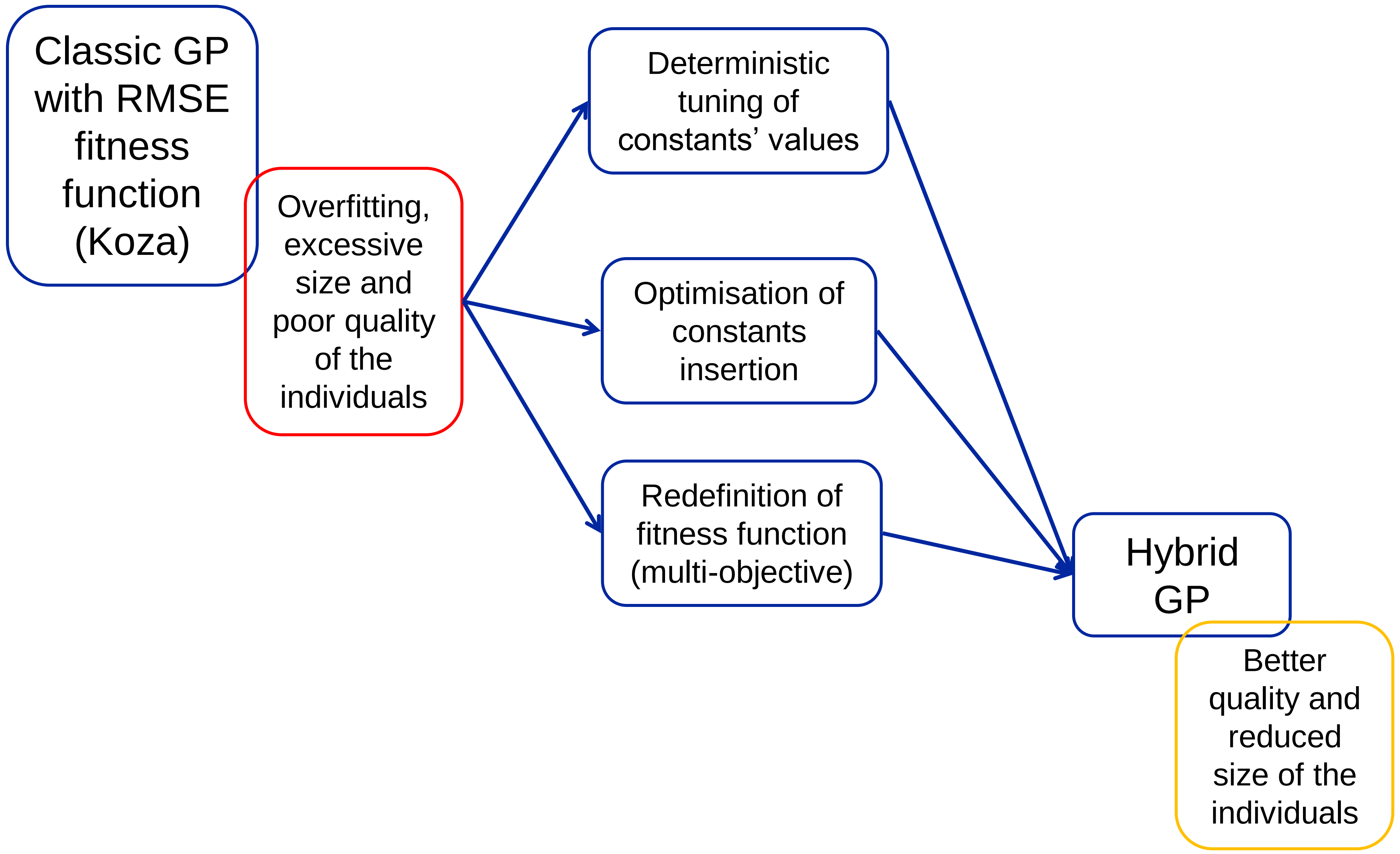
**Topchy and W. F. Punch,** "*Faster genetic programming based on local gradient search of numeric leaf values*" in Lee Spector, Erik D. Goodman, Annie Wu, W. B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), pages 155-162, San Francisco, California, USA, 7-11 July 2001. Morgan Kaufmann.

**Alvarez, L. F.,** Design optimization based on genetic programming, Ph.D. thesis, University of Bradford, Bradford, UK, 2000.

**Vladislavleva, E.,** Model-based Problem Solving through Symbolic Regression via Pareto Genetic Programming, Ph.D. thesis, Tilburg University,Tilburg, the Netherlands, 2008.

**Hornby, G. S.,** "ALPS: the age-layered population structure for reducing the problem of premature convergence," GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM, New York, NY, USA, 2006, pp. 815–822.

# Direction of current research: hybrid GP


Classic GP with RMSE fitness function (Koza)
Overfitting, excessive size and poor quality of the individuals
Deterministic tuning of constants' values
Optimisation of constants insertion
Redefinition of fitness function (multi-objective)
Hybrid GP
Better quality and reduced size of the individuals

# Redefinition of Fitness Function in hybrid GP

- Fitness evaluation : **weighted approach**.

$$F(k,t) = a_1F_1 + a_2F_2 + a_3F_3 + a_4F_4 \qquad a_1 + a_2 + a_3 + a_4 = 1$$

$$F_1 = \frac{RMSE(i,t)}{\overline{\overline{RMSE}}(t-1)} \qquad RMSE(i,t) = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(\hat{f}_j(i,t) - f_j\right)^2}$$

$$F_2 = \text{number of tuning parameters}$$

$$F_3 = \text{number of singularities}$$

$$F_4 = \text{number of nodes}$$

$$a_1 = 0.989 \quad a_2 = 0.01 \quad a_3 = 0.1 \quad a_4 = 0.001 \qquad \text{for example}$$