Symbolic Regression via Genetic Programming

Katya Vladislavleva and Guido Smits

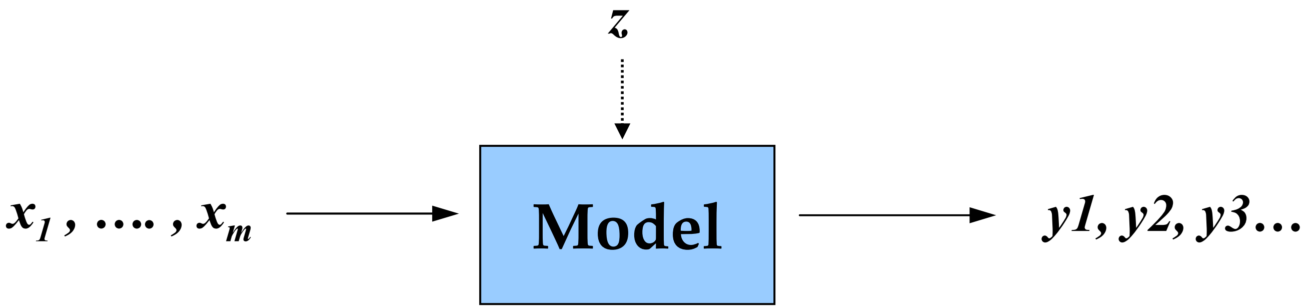
Tilburg University and Dow Benelux B.V.

Outline: Provoking the discussion

- The world is nonlinear – exploiting the problem structure is important
- The world is full of noise – uncertainty is everywhere
- The world is complex – we almost never have all the information
- The world is multi-objective – the interaction between objectives makes search difficult
- The world is deceptive – navigation is difficult and exploration is essential

- When we succeed in finding the minimum space where the minimum complexity model ‘lives’, optimization and design becomes easy

Problem: System design and Optimization



$$y = f(x_{i1} , …. , x_{ik})$$

Applications in chemical industry:

- Improved product consistency
- Minimum operator interference
- Automatic calibration
- Enhanced reactor control
- Reduced off-grade
- Faster type changes
- Fast response time

Real-life problems are hard

- Calculation of the response can be computationally expensive
- Feasibility of the response calculation is domain dependent (there may be infeasible regions of the design space)
- Response can be corrupted by noise
- Possible design variables may not be obvious and are not necessarily independent

Inspired by a (wonderful) presentation of Vassili Toropov
on multidisciplinary design optimization

What do we do?

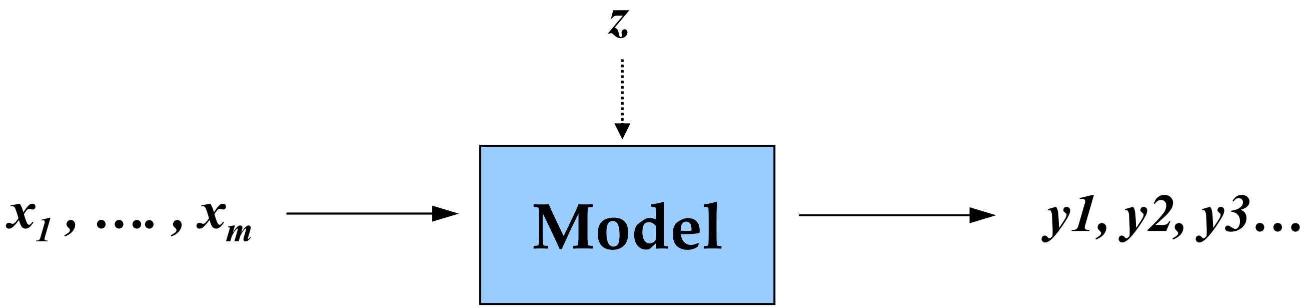
> *If something's hard to do, then it's not worth doing.*
>
> *Homer Simpson*

The picture of Homer and the quote are taken from the presentation of V.V. Toropov on multidisciplinary design optimization

Or we can use metamodels

- If the problem is too hard, we can use an approximation (a *metamodel*) of the ‘true’ underlying functional relationship by a model with required/desired properties (smooth, cheaper to compute, etc.).

Problem: System design and Optimization



$$y = f(x_{i1}, \ldots, x_{ik})$$

Applications in chemical industry:

- Improved product consistency
- Minimum operator interference
- Automatic calibration
- Enhanced reactor control
- Reduced off-grade
- Faster type changes
- Fast response time

Requirements to metamodels

You are never given a wish without being given a power to make it come true.

Richard Bach

- Credibility
- Interpretability
- Cost effectiveness
- Extrapolative capabilities
- Robustness
- Self-Assessment

Some Possible Solutions

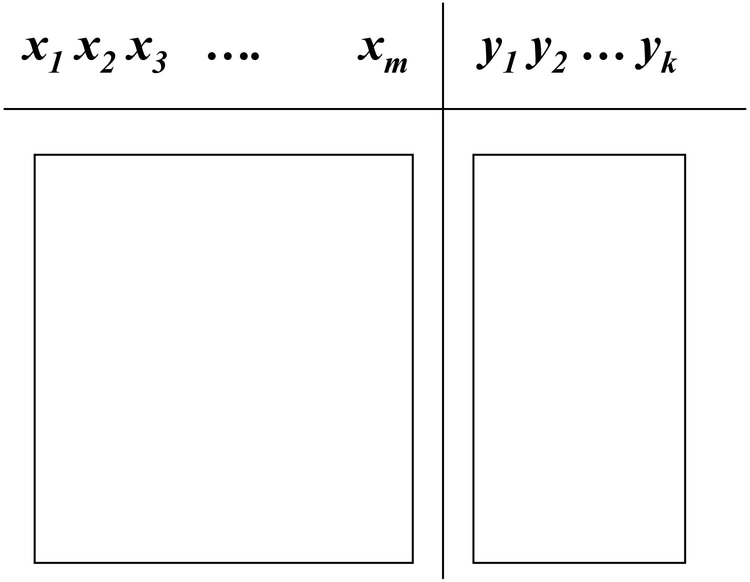
- Linear Regression
- Nonlinear Regression
- Rational Polynomials
- Kriging
- Neural Networks
- Support Vector Machines
- Symbolic Regression via Genetic Programming

Why we like Genetic Programming

Nothing astonishes men so much as common sense and plain dealing.
Ralph Waldo Emerson

- No a priory modeling assumptions
- Multiple solutions generated simultaneously with a few design parameters
- Natural selection of key variables
- No assumptions on independence of variables
- Human understanding of evolved models is possible

How does it work



$x_1\, x_2\, x_3 \quad \ldots. \qquad x_m$

$y_1\, y_2 \ldots y_k$

$$y = f(x_{i1}, \ldots., x_{ik})$$

Stochastic Iterative Search

- Initialization
- Selection
- Optimization
- Evaluation

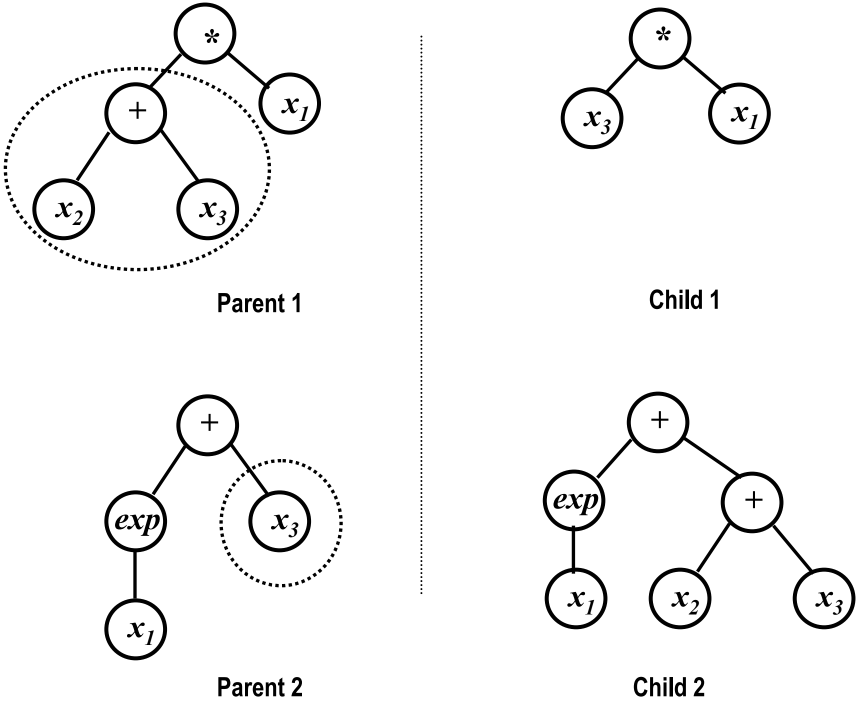
Simple EA with archiving

```
t=0;                        % Iteration step count
Archive(0)= [];             % Initialize Archive
Population(0)= [];          % Initialize population

while search is not terminated
    t=t+1;
    % Generate new population from archive models on
    % previous step of iteration
    Population(t)= Generate(Archive(t-1),Population(t-1));
    %Update Archive
    Archive(t) = UpdateArchive(Archive(t-1), Population(t));
end

%Return solutions of a GP run
Archive(t)
```

Model Representation and Meaning



Basic functions:

$\{+, -, *, /, sqrt, x^{real}, \ln x, e^{x}, e^{-x}, x^{y}\}$

Input variables:

$\{x_1, \ldots, x_n\}$ + Constants

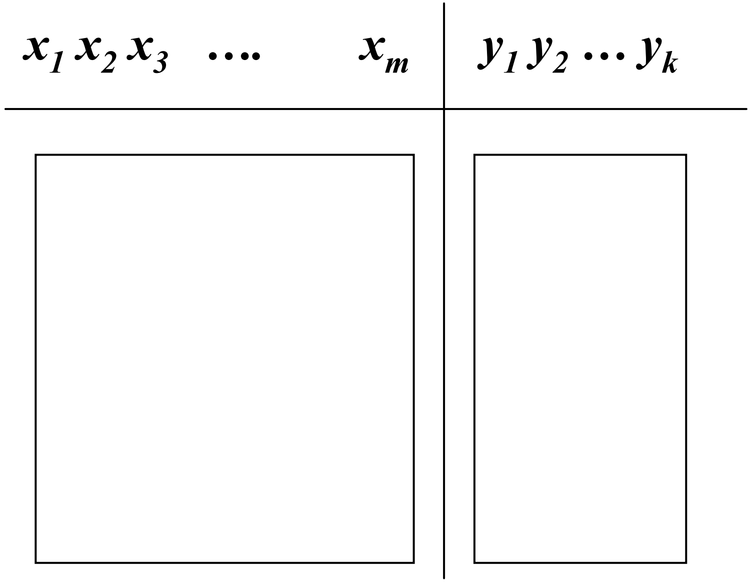
Operations on models:

crossover

mutation

cloning

Real Data has its issues



- High dimensional
- Not designed
- Correlated
- Noisy
- Redundant or sparse
- Multiple time scales

They may cause over-fitting…

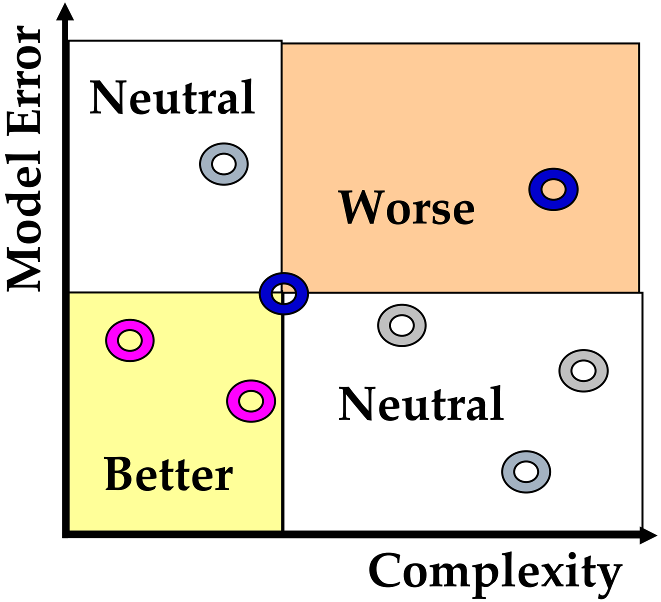
Complexity control

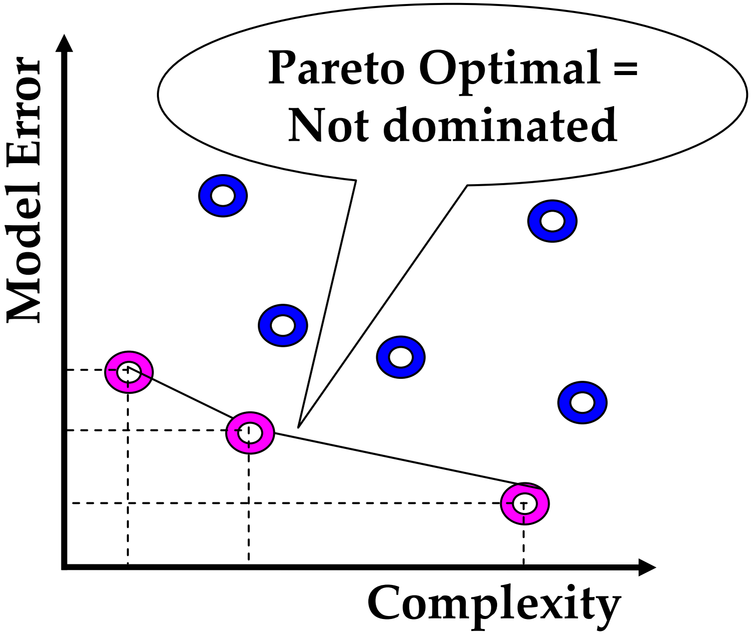
No more things should be presumed to exist than are absolutely necessary.
William of Okkam

- Numerical accuracy
- Structural complexity (e.g., size)
- Smoothness of the response surface (e.g., order of nonlinearity) and graceful degradation
- Dimensionality
- Consistency in units
- …

Multiple criteria need to be satisfied at the same time

Dominance and Pareto Optimality

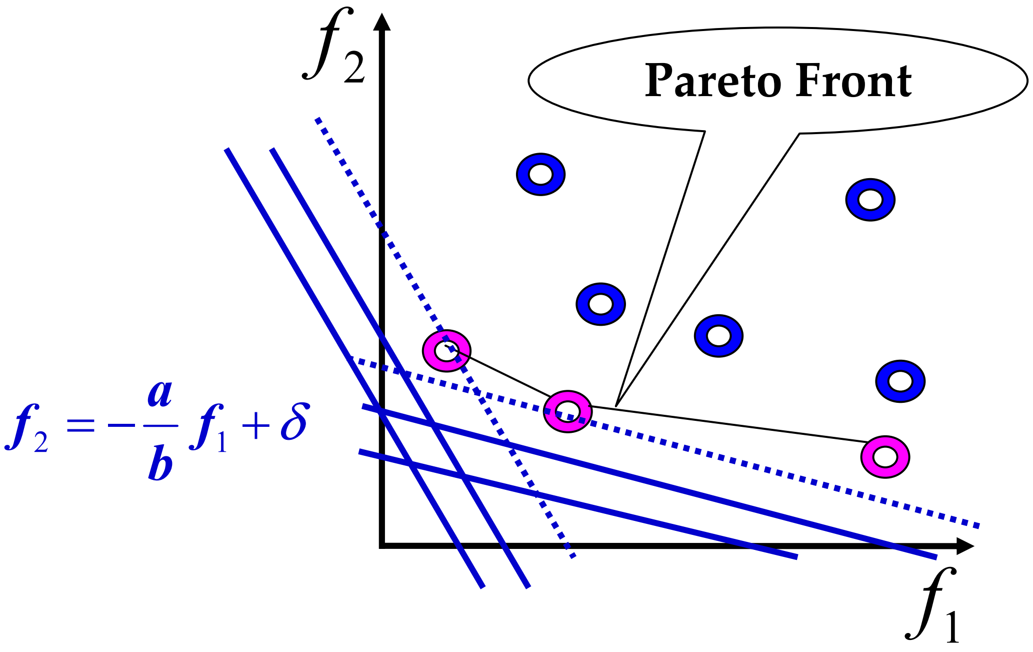




Composite fitness function is not good enough

$$\min(f_1, f_2)$$

$$\min(a \cdot f_1 + b \cdot f_2)$$



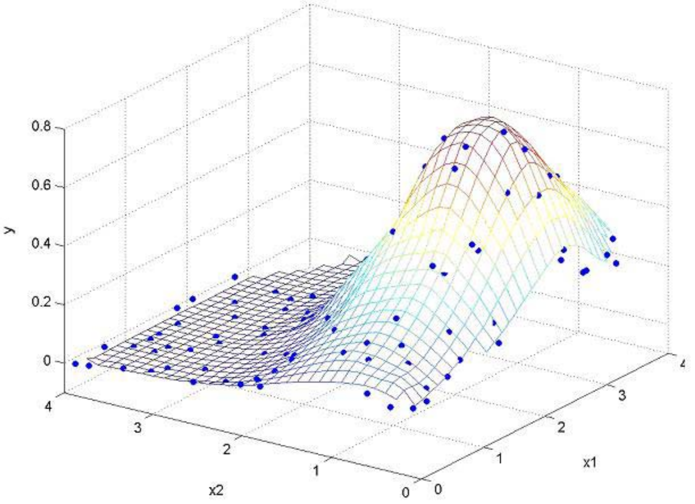
When the 'right' *a* and *b* are unknown, one can try different combinations of them. This may require too many GP runs.

Two-objective optimization aims at finding the entire Pareto front in one GP run.

Complexity is not only the size..

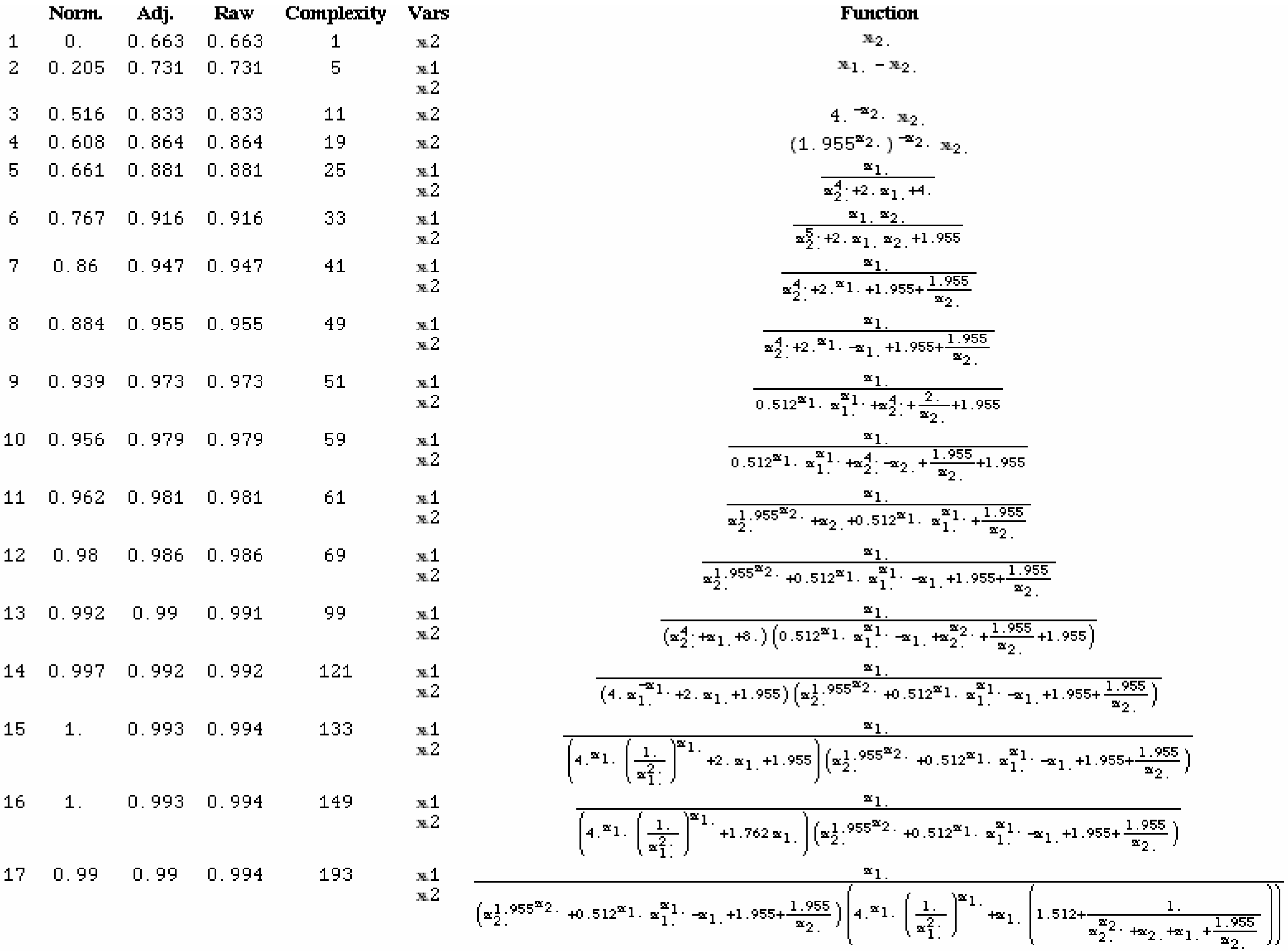
Model $e\mathcal{T}$:

$$\frac{e^{-(x_2-1)^2}}{1.2+(x_1-2.5)^2}$$

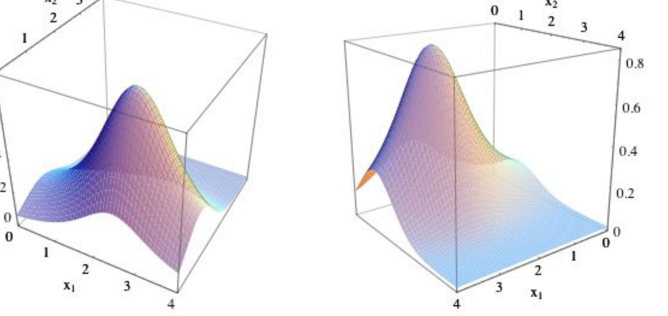


$$\theta \mapsto \widetilde{\theta} \mapsto (C_1(\widetilde{\theta}), C_2(\widetilde{\theta}), \ldots, C_k(\widetilde{\theta})) \in R^k$$

Example

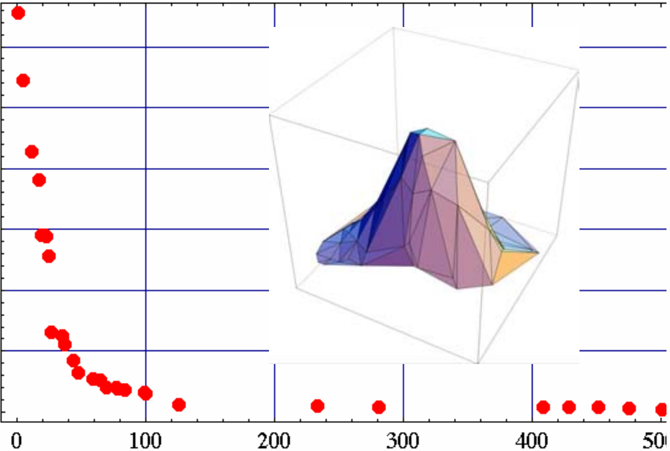


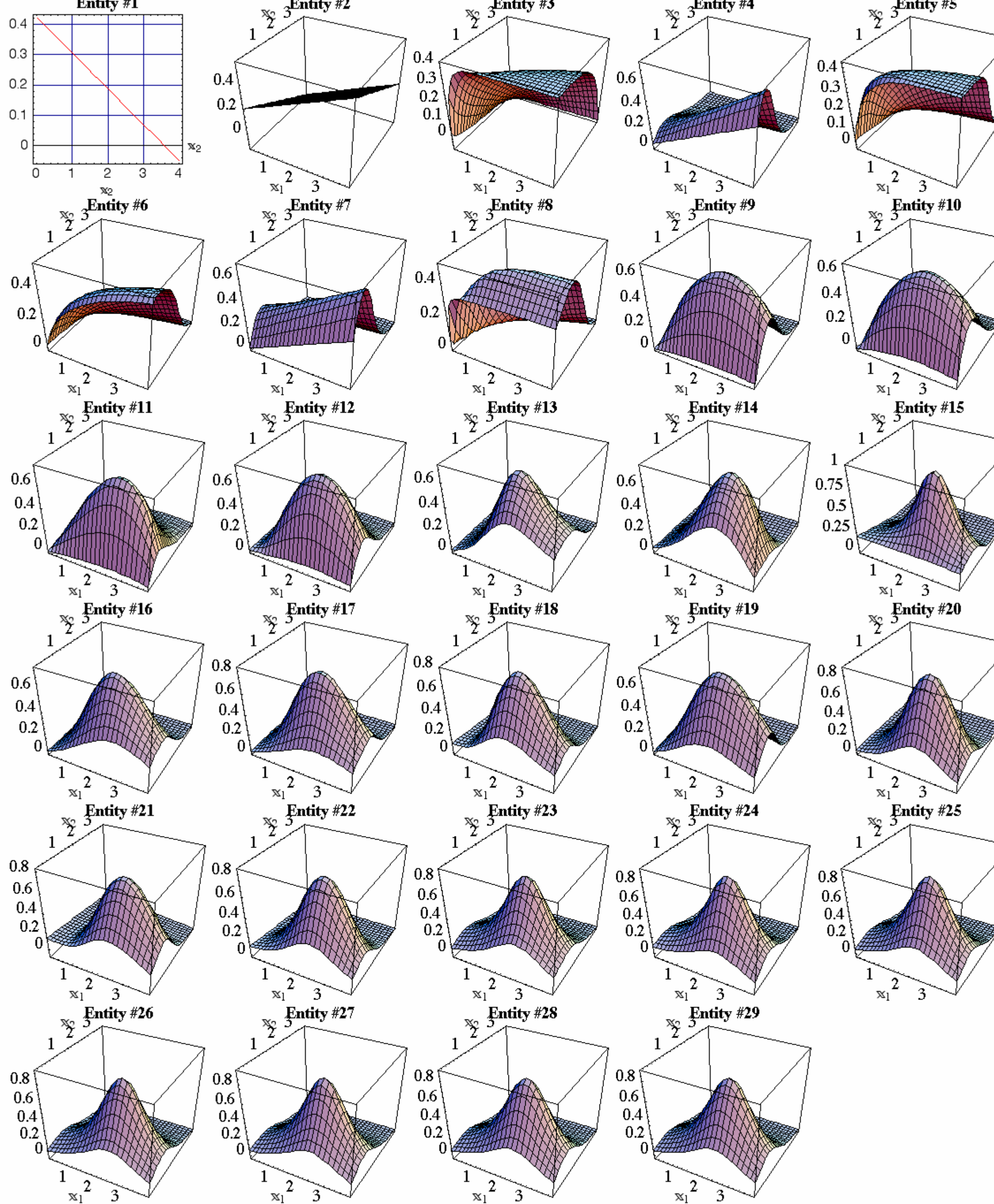
	Norm.	Adj.	Raw	Complexity	Vars	Function
1	0.	0.663	0.663	1	x2	x_2
2	0.205	0.731	0.731	5	x1 x2	$x_1 - x_2$
3	0.516	0.833	0.833	11	x2	$4.^{-x_2}\, x_2$
4	0.608	0.864	0.864	19	x2	$(1.955^{x_2})^{-x_2}\, x_2$
5	0.661	0.881	0.881	25	x1 x2	$\frac{x_1}{x_2^4 + 2.\, x_1 + 4.}$
6	0.767	0.916	0.916	33	x1 x2	$\frac{x_1 x_2}{x_2^5 + 2.\, x_1 x_2 + 1.955}$
7	0.86	0.947	0.947	41	x1 x2	$\frac{x_1}{x_2^4 + 2.^{x_1} + 1.955 + \frac{1.955}{x_2}}$
8	0.884	0.955	0.955	49	x1 x2	$\frac{x_1}{x_2^4 + 2.^{x_1} - x_1 + 1.955 + \frac{1.955}{x_2}}$
9	0.939	0.973	0.973	51	x1 x2	$\frac{x_1}{0.512^{x_1} x_1^{x_1} + x_2^4 + \frac{2.}{x_2} + 1.955}$
10	0.956	0.979	0.979	59	x1 x2	$\frac{x_1}{0.512^{x_1} x_1^{x_1} + x_2^4 - x_2 + \frac{1.955}{x_2} + 1.955}$
11	0.962	0.981	0.981	61	x1 x2	$\frac{x_1}{x_2^{1.955^{x_2}} + x_2 + 0.512^{x_1} x_1^{x_1} + \frac{1.955}{x_2}}$
12	0.98	0.986	0.986	69	x1 x2	$\frac{x_1}{x_2^{1.955^{x_2}} + 0.512^{x_1} x_1^{x_1} - x_1 + 1.955 + \frac{1.955}{x_2}}$
13	0.992	0.99	0.991	99	x1 x2	$\frac{x_1}{(x_2^4 + x_1 + 8.)\left(0.512^{x_1} x_1^{x_1} - x_1 + x_2^{x_2} + \frac{1.955}{x_2} + 1.955\right)}$
14	0.997	0.992	0.992	121	x1 x2	$\frac{x_1}{(4.\, x_1^{-x_1} + 2.\, x_1 + 1.955)\left(x_2^{1.955^{x_2}} + 0.512^{x_1} x_1^{x_1} - x_1 + 1.955 + \frac{1.955}{x_2}\right)}$
15	1.	0.993	0.994	133	x1 x2	$\frac{x_1}{\left(4.^{x_1} \left(\frac{1.}{x_1^2}\right)^{x_1} + 2.\, x_1 + 1.955\right)\left(x_2^{1.955^{x_2}} + 0.512^{x_1} x_1^{x_1} - x_1 + 1.955 + \frac{1.955}{x_2}\right)}$
16	1.	0.993	0.994	149	x1 x2	$\frac{x_1}{\left(4.^{x_1} \left(\frac{1.}{x_1^2}\right)^{x_1} + 1.762\, x_1\right)\left(x_2^{1.955^{x_2}} + 0.512^{x_1} x_1^{x_1} - x_1 + 1.955 + \frac{1.955}{x_2}\right)}$
17	0.99	0.99	0.994	193	x1 x2	$\frac{x_1}{\left(x_2^{1.955^{x_2}} + 0.512^{x_1} x_1^{x_1} - x_1 + 1.955 + \frac{1.955}{x_2}\right)\left(4.^{x_1} \left(\frac{1.}{x_1^2}\right)^{x_1} + x_1 \left(1.512 + \frac{1.}{x_2^{x_2} + x_2 + x_1 + \frac{1.955}{x_2}}\right)\right)}$

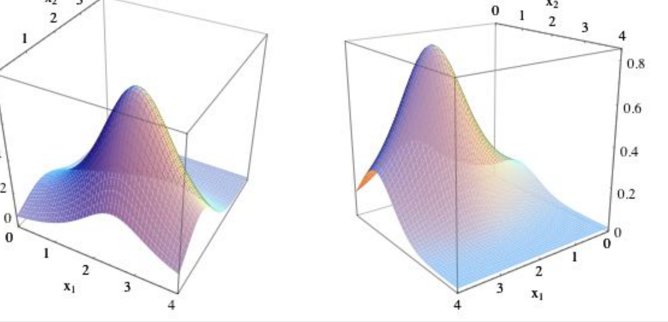


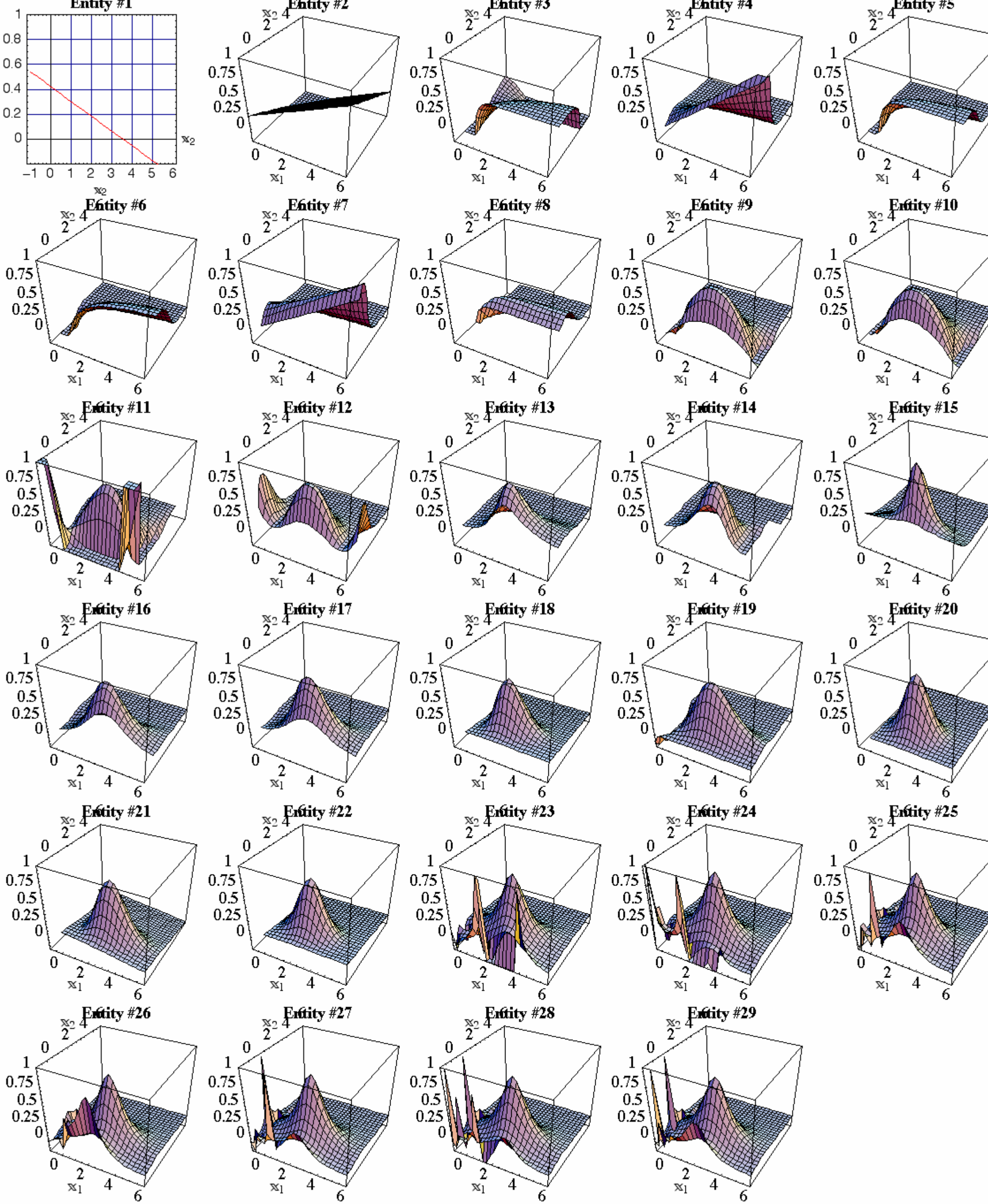
Truth

$$\frac{e^{-(x_2-1)^2}}{1.2+(x_1-2.5)^2}$$



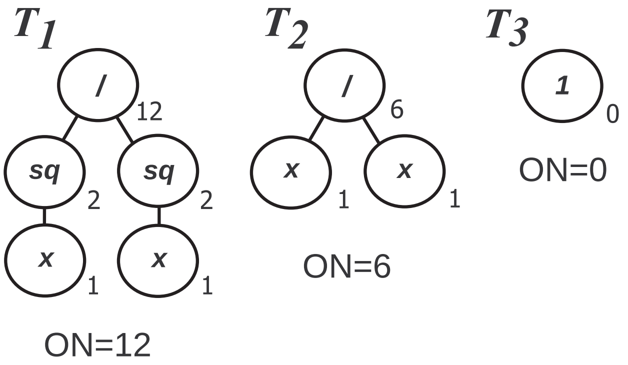




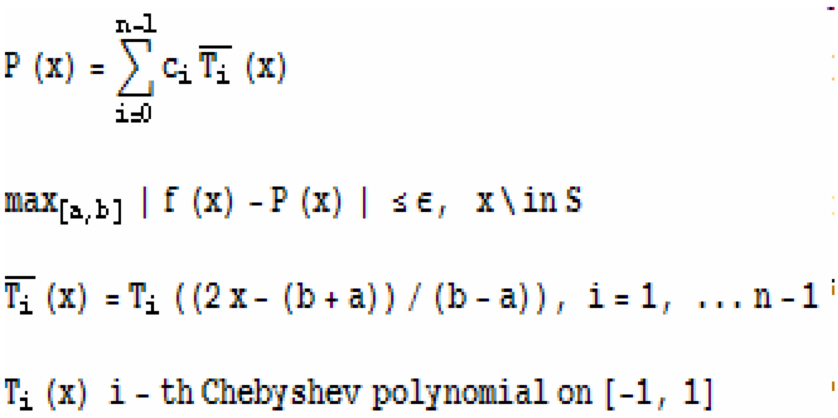


Note the pathologies at high complexity when extrapolating

Nonlinearity Control produces 'Smoother' Solutions



- All functions are treated as univariate
- Interval arithmetic is taken into account
- Chebyshev Approximation of certain accuracy is exploited:



$$P(x) = \sum_{i=0}^{n-1} c_i \overline{T_i}(x)$$

$$\max_{[a,b]} | f(x) - P(x) | \leq \epsilon, \quad x \in S$$

$$\overline{T_i}(x) = T_i((2x - (b+a))/(b-a)), \; i = 1, \ldots n-1$$

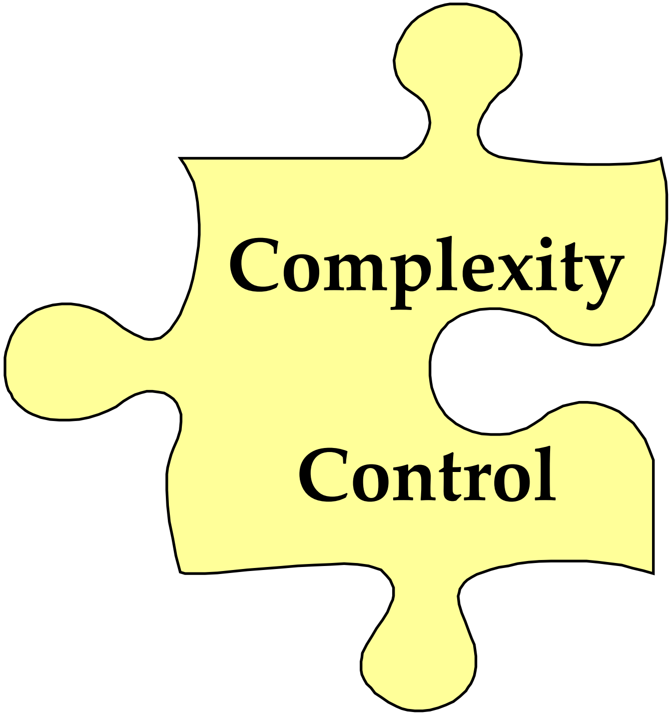
$T_i(x)$ i-th Chebyshev polynomial on $[-1, 1]$

Do not get too scared by the ‘Classical’ Problems in the GP field

- Over-fitting and weak generalization capabilities
- Bloat
- Premature convergence
- Efficiency / Scale-up

There are ways to solve them, or, at least, to heavily reduce their negative effects

To reduce over-fitting:



Several optimization measures are explicitly taken into account and are optimized simultaneously to:

- control bloat
- prevent over-fitting
- improve interpretability of solutions
- promote generalization capabilities
- generate solution ensembles with trust metrics

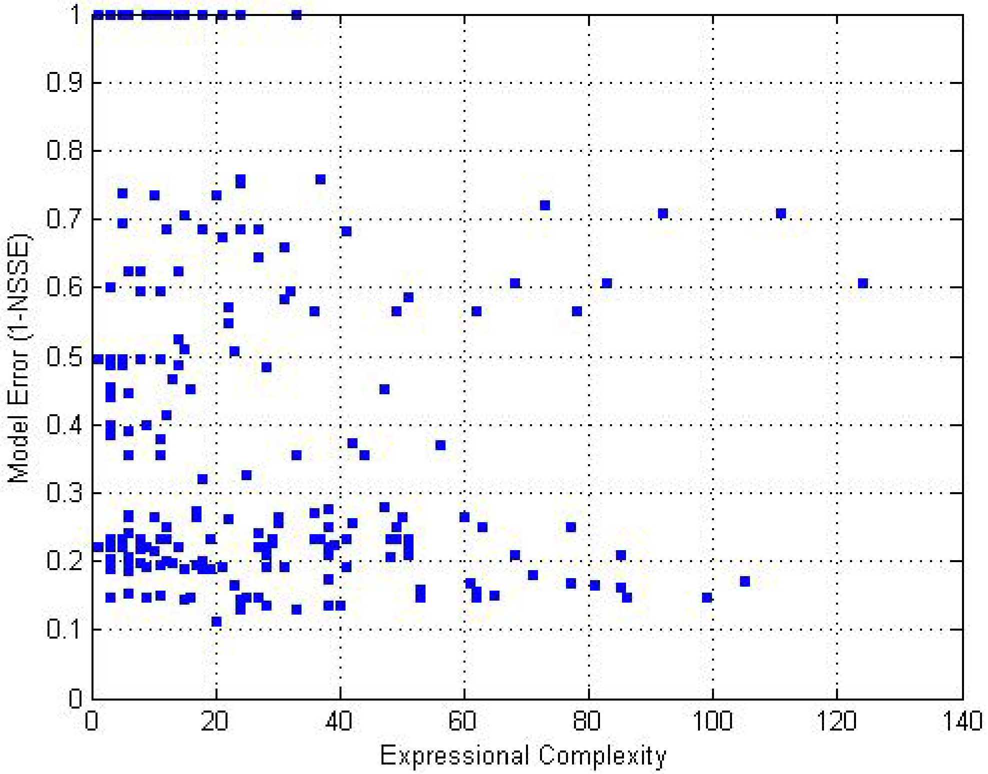
To get effective evolutions that scale well

An archive of the ‘best’ solutions is maintained (separately from the population) throughout the run.

- allows abundant **exploration** without a risk of loss in **exploitation**
- enables the GP system to discover good solutions with considerably **smaller population sizes**
- introduces natural **ranking** of solutions **w.r.t. optimization objectives**
- makes the introduction of ‘species’ easy

Archiving in Pareto GP

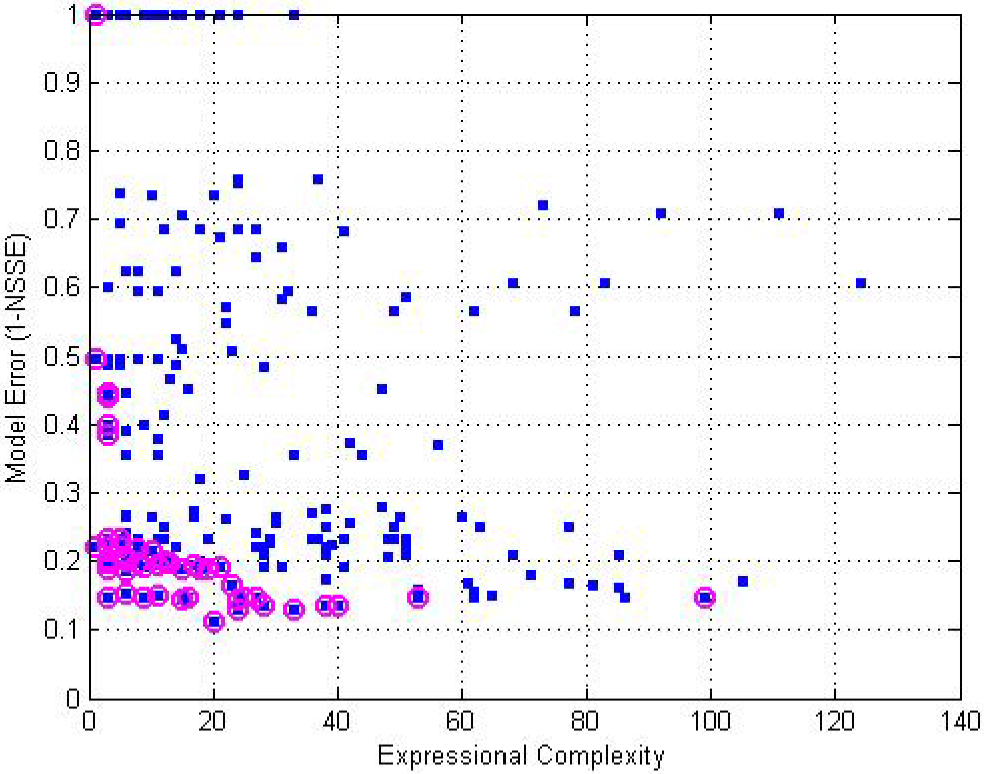
Population
generation 1

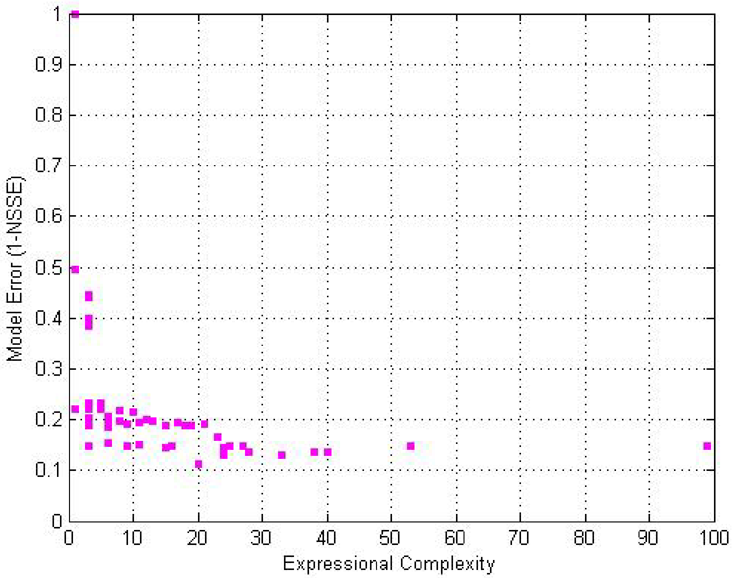


Archiving in Pareto GP

Population(1)

generation 1

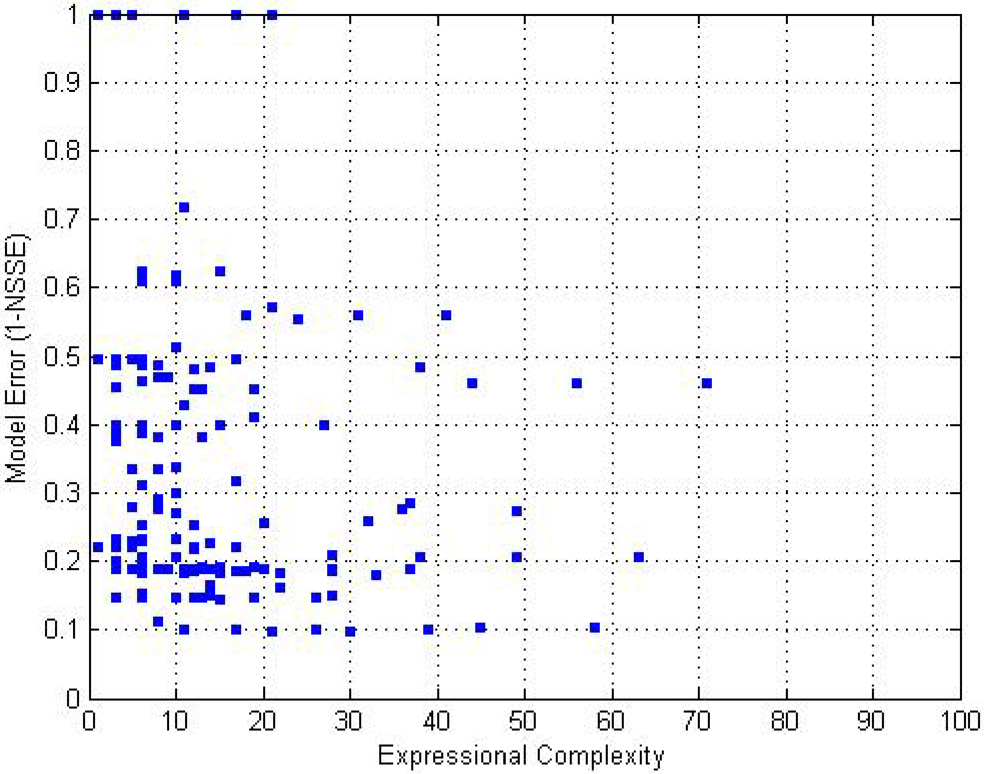


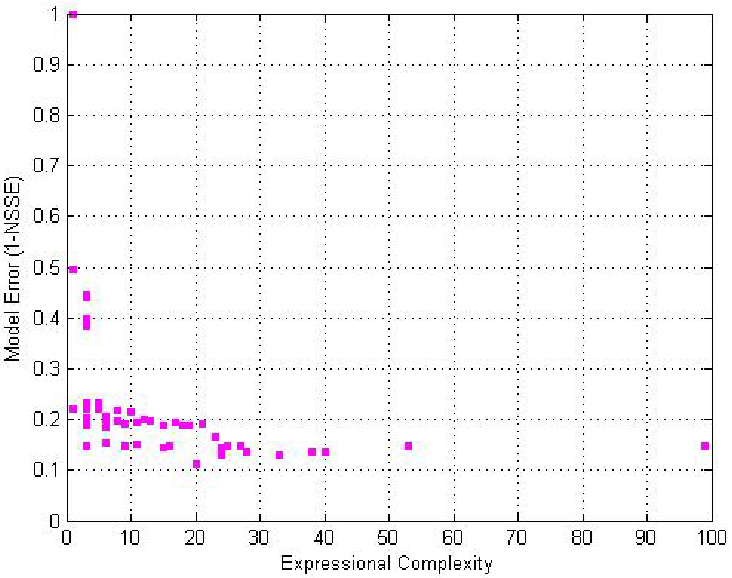


Archive(1)

Archiving in Pareto GP

Population(2)
generation 2



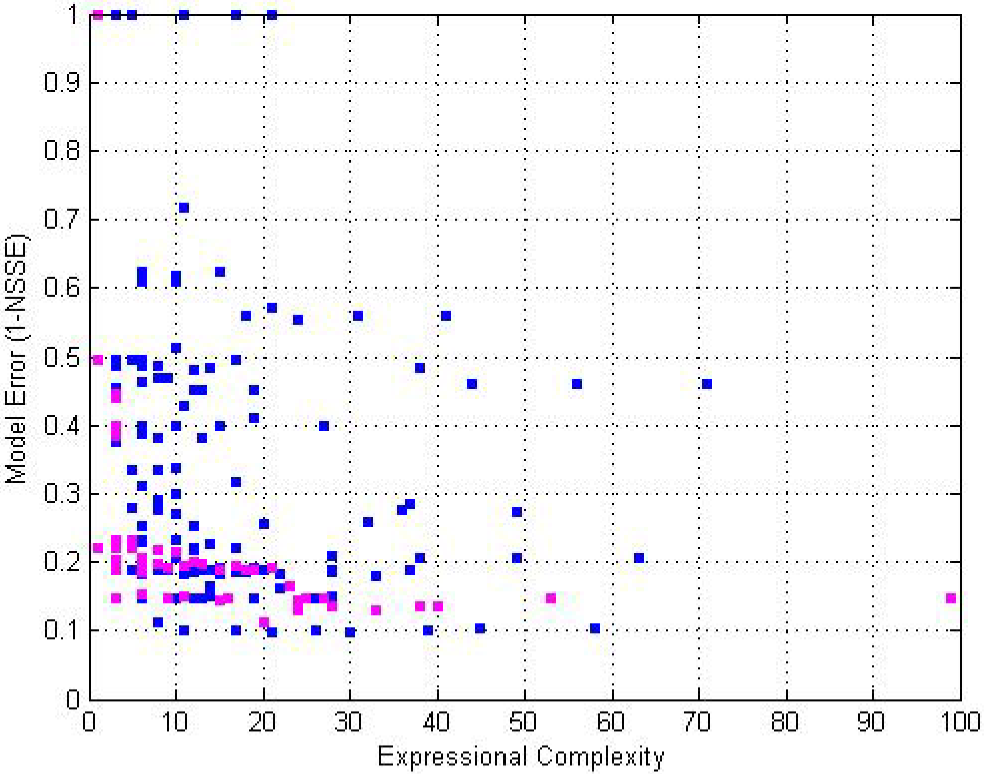


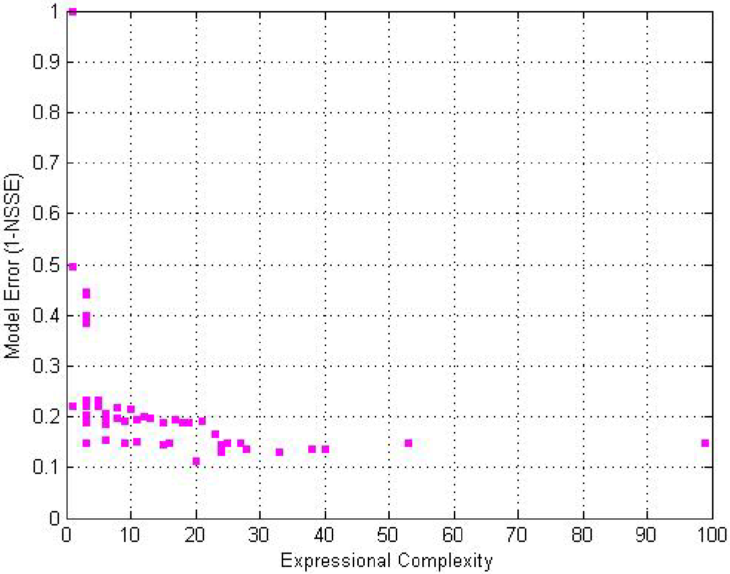
Archive(1)

Archiving in Pareto GP

Population(2) and Archive(1)

generation 2

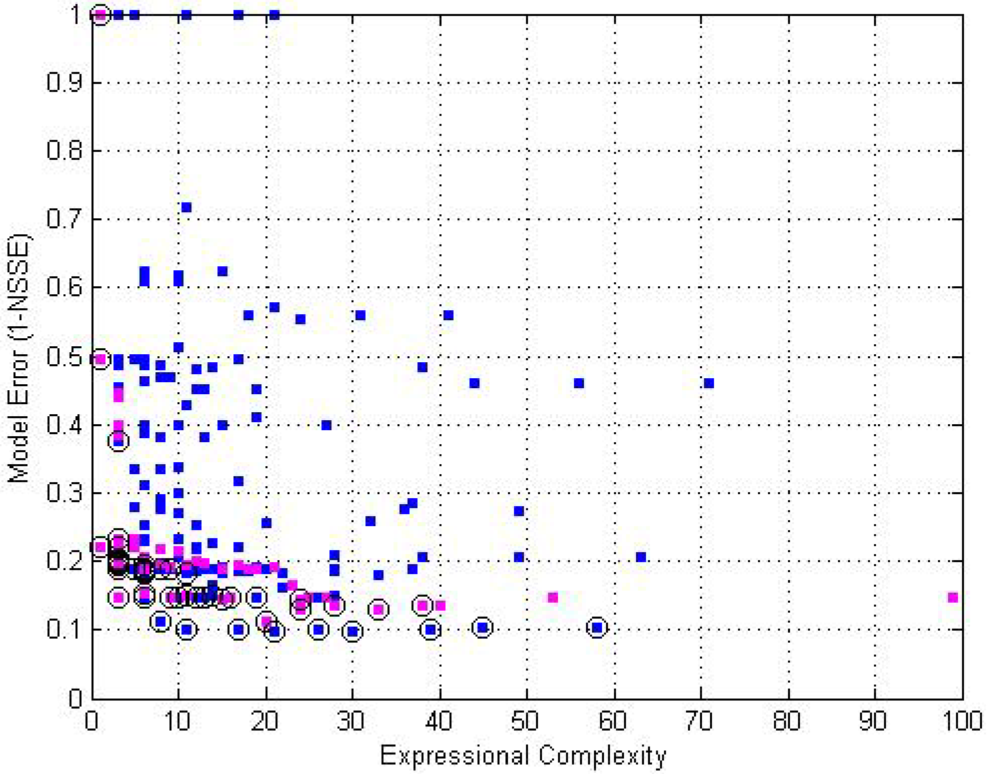


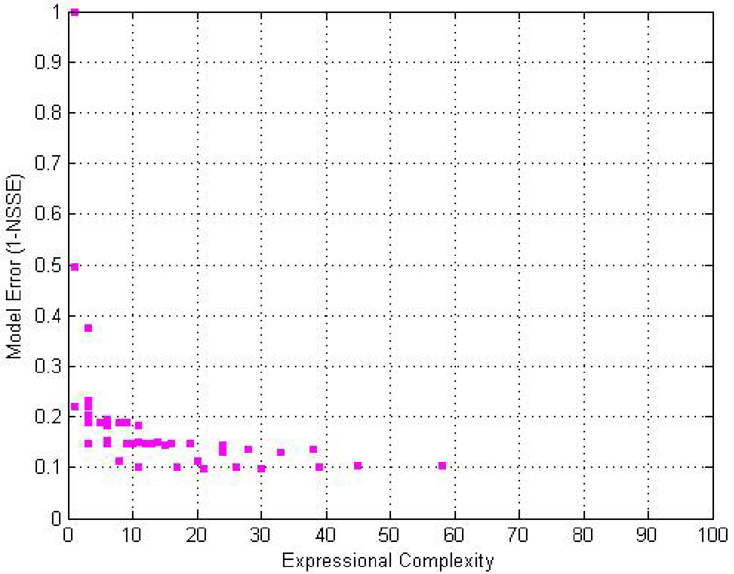


Archive(1)

Archiving in Pareto GP

Population(2) and Archive(1)
generation 2

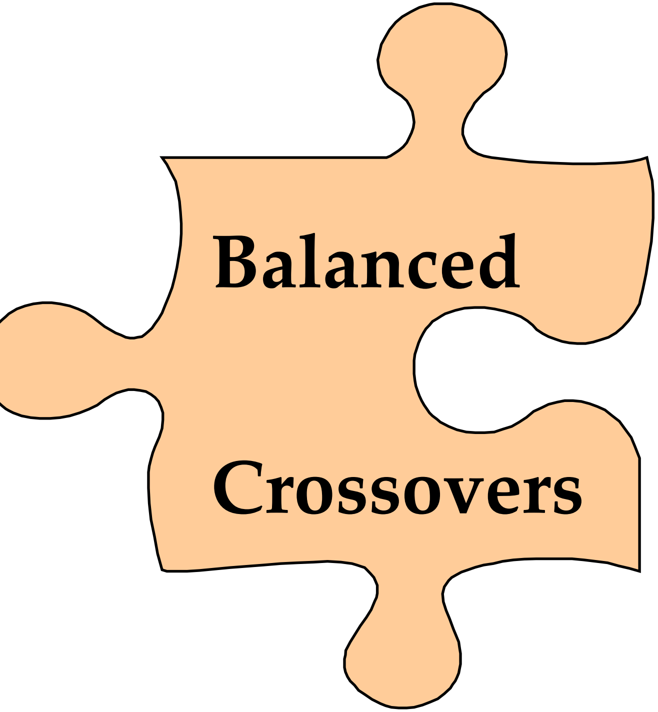




Archive(2)

Area under Pareto Front is a more robust measure for estimating performance of the GP run

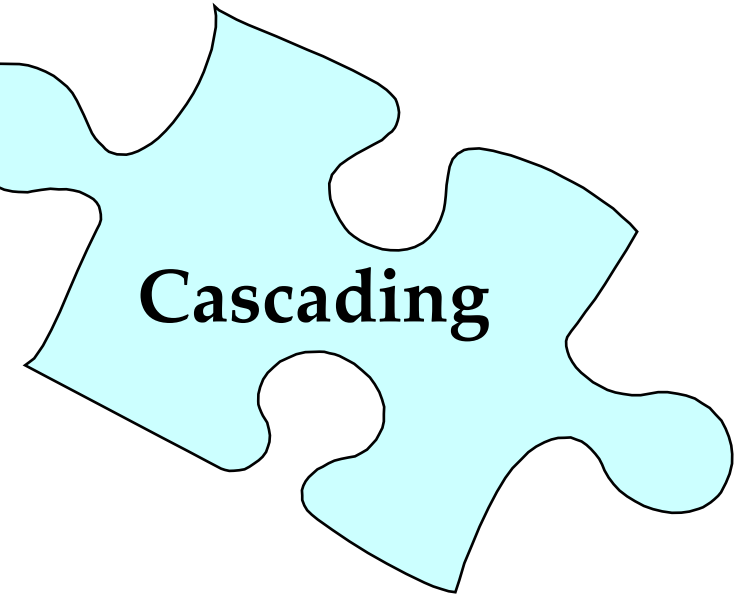
To reduce bloat:



With tree-based genotypes the crossover point of the second parent is chosen NOT randomly, but wisely, such that the parents exchange sub-trees of a similar size.

This helps to control **bloat**.

To avoid premature convergence:

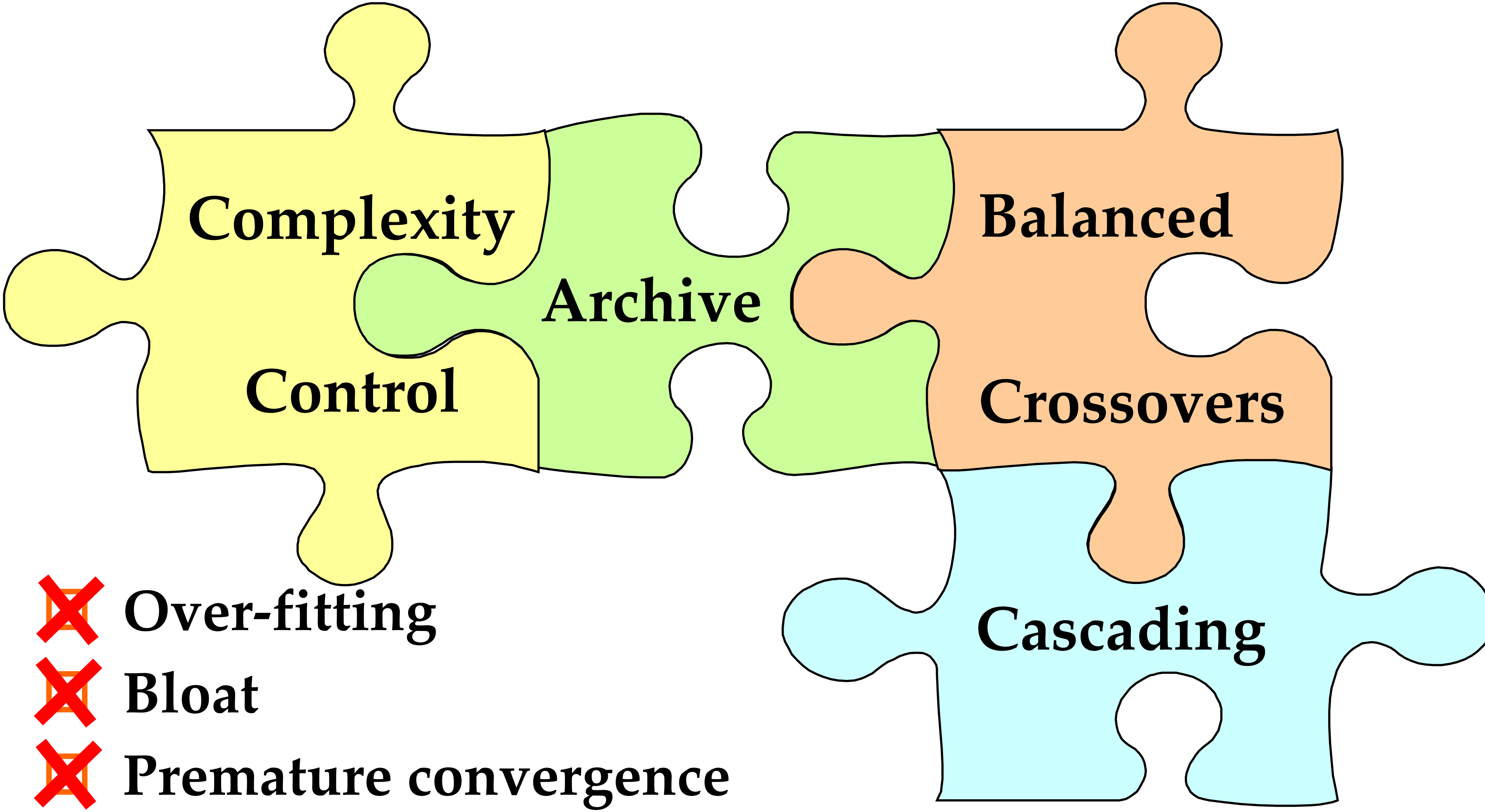


Cascading

Every *k* generations (a.k.a. ‘cascade’) the **population is re-initialized** (but the archive survives).

- promotes new genetic material
- prevents inbreeding
- avoids premature convergence

The puzzle evolves…



Complexity Control

Archive

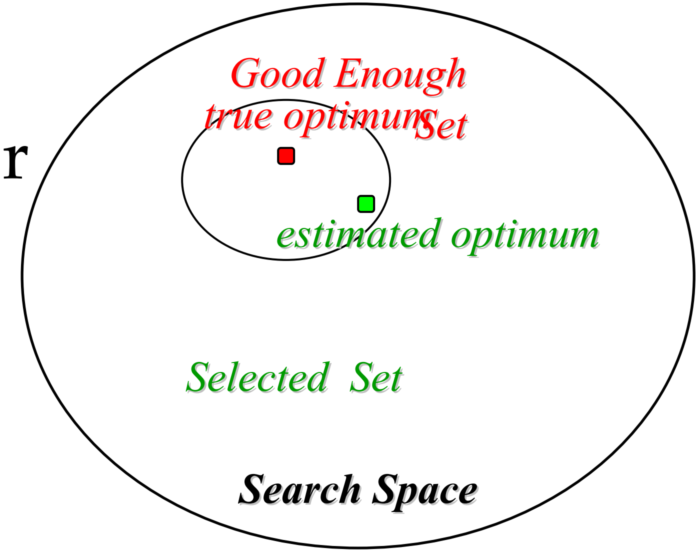
Balanced Crossovers

Cascading

- ☒ Over-fitting
- ☒ Bloat
- ☒ Premature convergence
- ☒ Efficiency / Scale-up

Goal Softening

- **Good enough** rather than **Optimal**
- **With high probability** rather than **For sure**
- **Approximate solution** rather than **Exact solution**
- **Heuristics** rather than **Rational decision**
- Estimate **order** rather than **value**



Efficiency and Scale Up

- ☐ Fitness evaluations eat most of the CPU time
- ☐ Computational budget needs to be spent wisely

Population of **popSize** individuals is evaluated on

nRecords records over

nGen generations. Which gives

popSize x nRecords x nGen function evaluations.

Ranking in GP

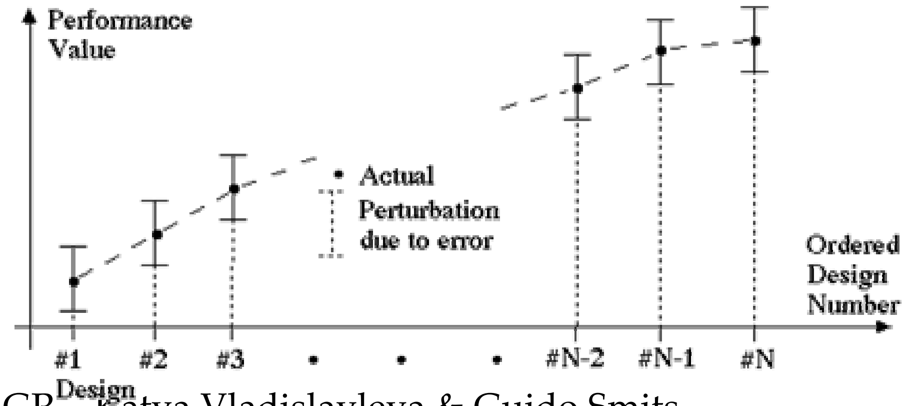
- The main driving force of evolution lies in endowing **better** individuals with **more propagation rights**
- The **order** of individuals **is sufficient**, exact performance **values** are not required

Soften the requirements?

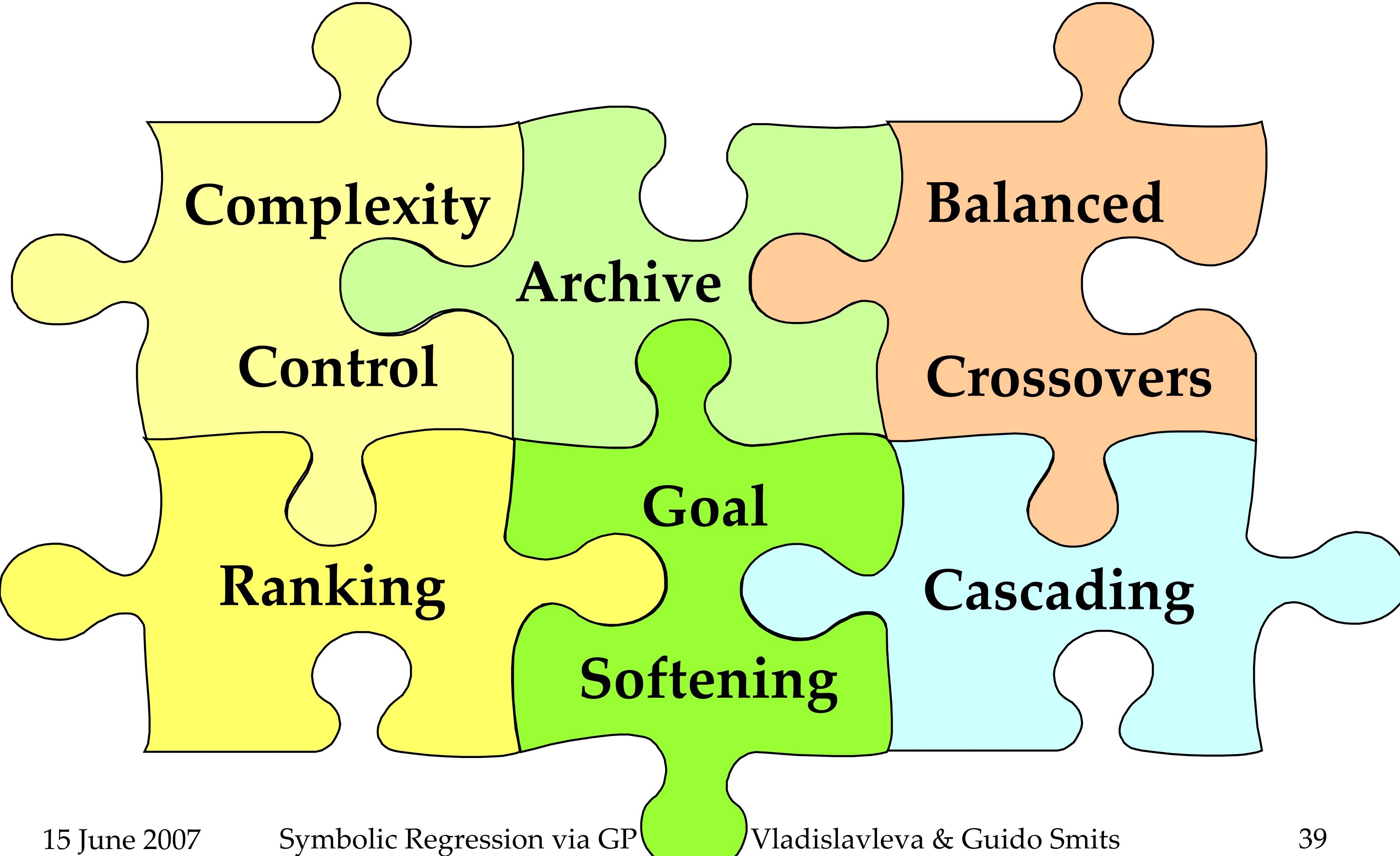
- Ranking is all we need

- Computational effort is spent mostly on fitness evaluation

Can we work with an approximate ranking based on partial fitness evaluations only?



Focus on Ranking and goal softening improves things greatly…



Complexity Control

Archive

Balanced Crossovers

Ranking

Goal Softening

Cascading

Yes, we can reduce computation time and get more robust solutions

- Goal softening works. It comes from
  - The use of subsets for fitness evaluation instead of the entire set
  - The use of **random** subsets of a given size (regularizes solutions, reduces over-fitting)
  - The decision to not re-evaluate archive models at every generation
  - The decision to spend more budget at the beginning of the run, and less at the end

Summary on Soft Evolutions

- It does make sense to explore intensively before we start exploiting
- Approximate ranking of gp individuals improves their robustness
- It is better to start with a very coarse ranking by evaluating individuals on random subsets of only 10% or 20% of the original size

Conclusions

- Symbolic Regression via GP helps to extract information (value) from the data
- Symbolic regression via GP has evolved considerably since the early 90’s

Summary

- SR via GP facilitates physical insight & understanding
- Summarizes multivariate data behavior
- Identifies key variables and data transforms
- Enables response surface exploration and optimization
- Visualizes the data behavior in the form of an analytical expression
- Suggests future experiments

- Physical and Chemical laws are not taken into account yet
- Data is assumed to be reasonably accurate
- Experts are still required, which is a good thing…