Industrial Evolutionary Computing

Arthur Kordon[#], Guido Smits[#], and Mark Kotanchek[+]

The Dow Chemical Company [#]
Evolved Analytics [+]

GECCO 2005

Overview

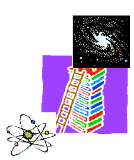
In theory, there is no difference between theory and practice. In practice, there is.
- Jan L.A. van de Snepscheut

- Evolutionary Computing and the business model
- Implementation guidelines
- Integrate & Conquer
- Key application areas
- Open issues

Academic vs. industrial data analysis

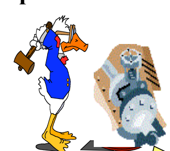
Transfer data into knowledge

Transfer data into value



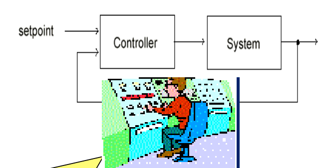
Special Features of Industrial Data Analysis

Operators intervention



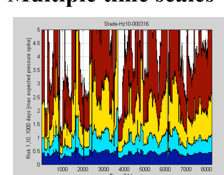
Operators manually modify the process

Curse of closed loops



The majority of process variables are in closed loops and depend on controller adjustments

Multiple time scales



Time scales vary from milliseconds to months

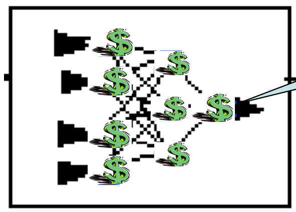
Real-time pressure

Most of models operate in real time

Intelligent Systems in Industrial Data Analysis: Lessons From the Past

The Expert Systems campaign (late 80s)
"We'll put engineers in the box"

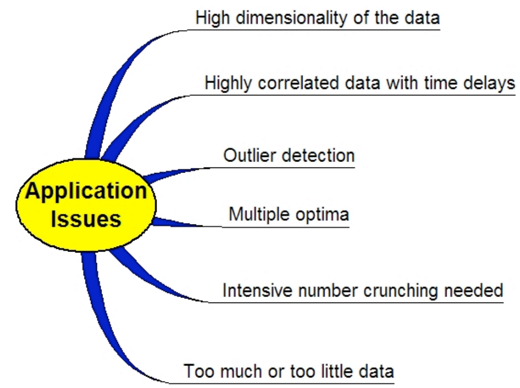
- **static rule-based models not linked to numerical world**
- **the politics of knowledge acquisition**
- **the efforts of knowledge acquisition**



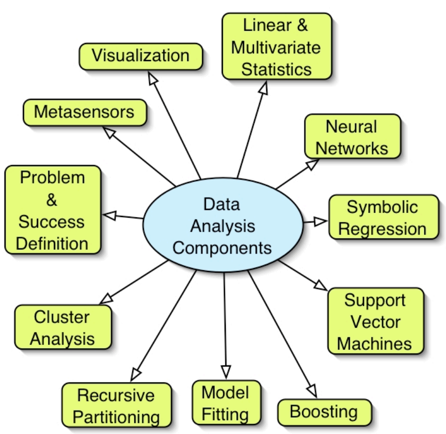
The Neural Networks campaign (early 90s)
"We'll turn data into gold"

- **black-box models with inefficient structure**
- **fragile models and model validation**
- **maintenance nightmare**

Application Issues in the Chemical Industry



Industrial data analysis components

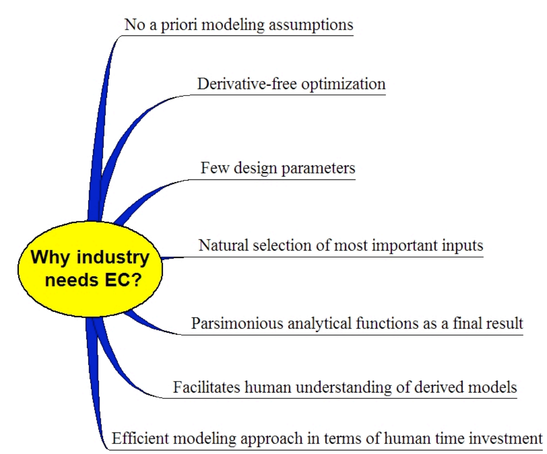


The role of evolutionary computing (symbolic regression) is to …

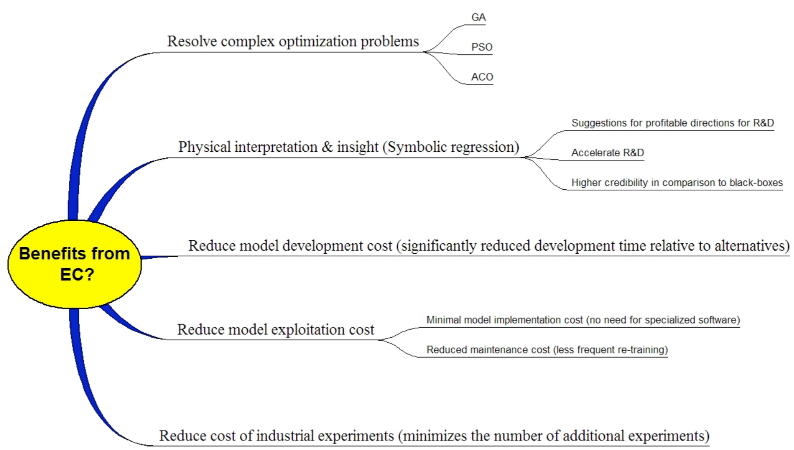
- Facilitate physical/mechanism insight and **understanding**
- **Summarize** data behavior
- Identify data **transforms** and metasensors
- Perform **variable selection**
- Enable response surface **exploration and optimization**
- **Visualize** behavior in the form of a symbolic expression

The overall goal is to achieve speed, accuracy & efficiency. Symbolic regression is part of an integrated methodology.

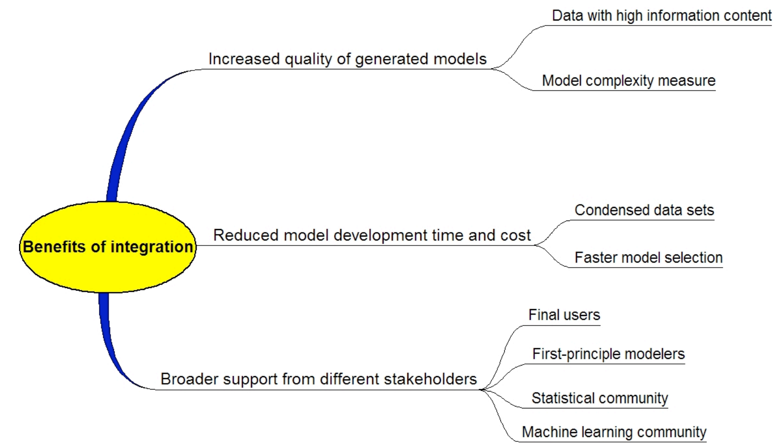
Why industry needs Evolutionary Computing?



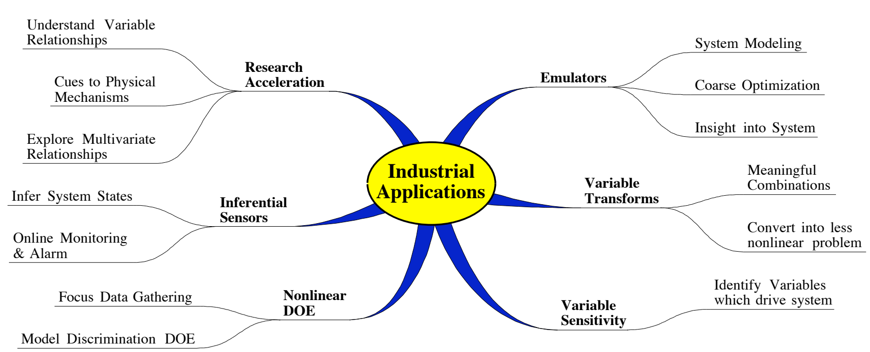
Economic benefits from Evolutionary Computing



Benefits of integrating Evolutionary Computing with other approaches



Application areas with impact

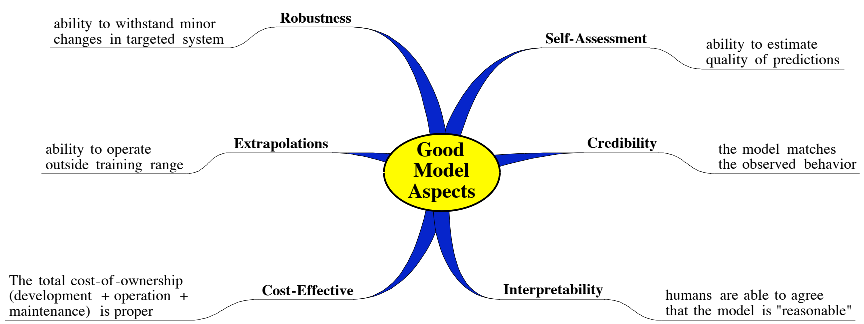


Implementation guidelines

- Requirements for successful empirical modeling
- Key issues to be overcome
- Implementation strategy
- Implementation tools

Requirements for successful data-driven modeling

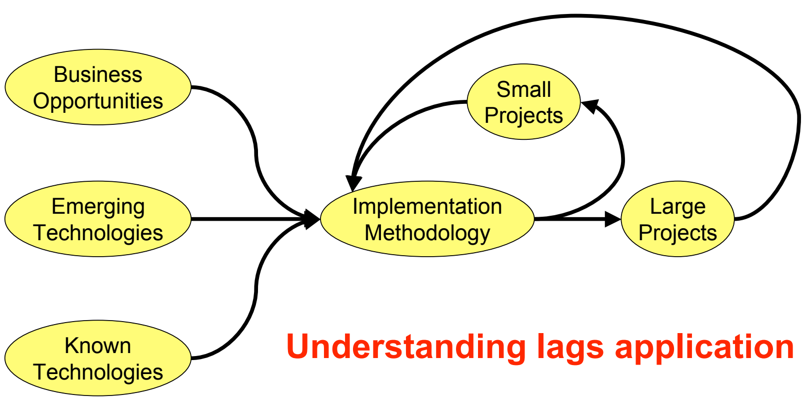
Objective function:
Minimizing modeling cost and maximizing data analysis efficiency under broad range of operating conditions



Key issues to overcome

- **Key implementation issues**
  - **Data pre-processing and condensation**
  - **Model selection**
  - **User-friendly implementation tools**
  - **Marketing of EC to different modeling communities**
  - **Resistance to implement empirical models (inherited from black-box models)**
  - **Seamless integration into existing maintenance and support infrastructure**
  - **Critical mass of model developers familiar with EC**

Implementation Strategy



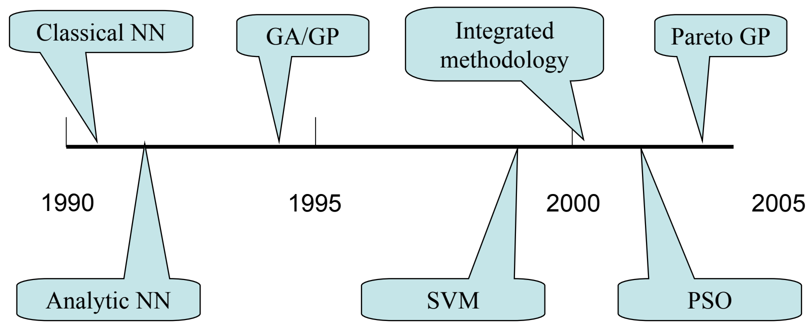
Understanding lags application

(Good judgement comes from experience; experience comes from bad judgement)

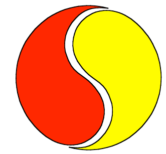
Implementation tools

- MATLAB (Dow developed)
  - GA
  - GP
  - PSO (single objective and multi-objective)
  - Analytic neural networks
  - Support vector machines
- Mathematica (Dow developed)
  - Symbolic regression package
  - AutoAnalysisTools
  - Analytic neural networks
  - PSO
  - Group Methods of Data Handling (GMDH)
- Tools for model distribution
  - Delphi
  - Web Mathematica
  - Excel
  - Process control systems

Exploitation/Implementation Sequence of Computational Intelligence Approaches in Dow Chemical

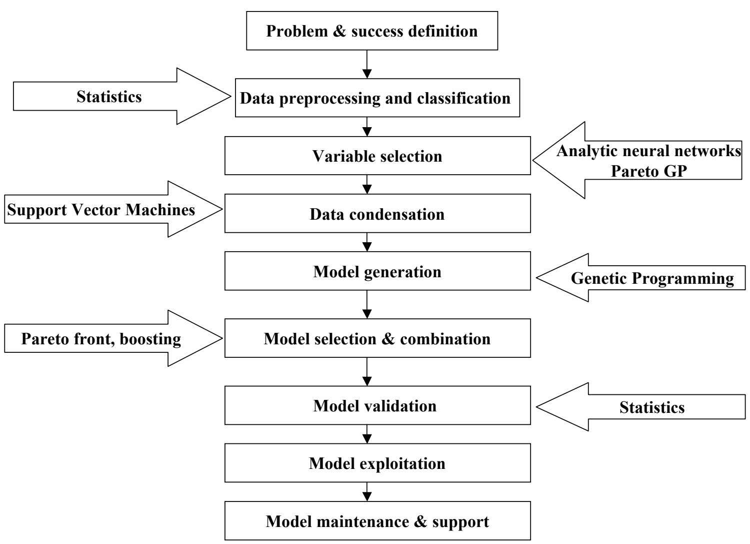


Integrate & Conquer

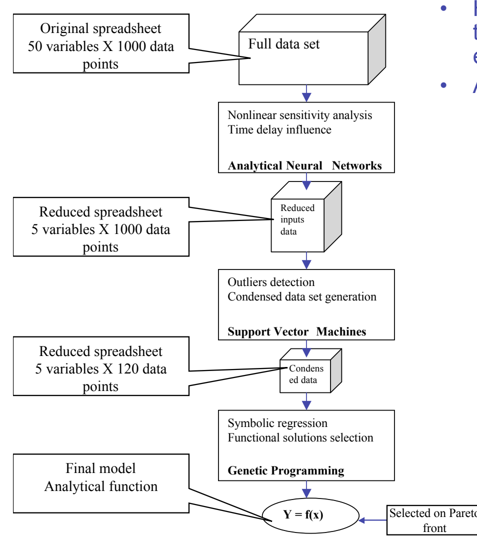


- Integrated methodology for successful EC implementation
- Related approaches
- A case study

Integrated Methodology

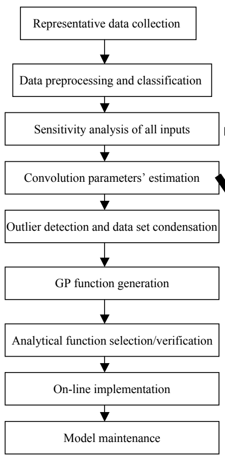


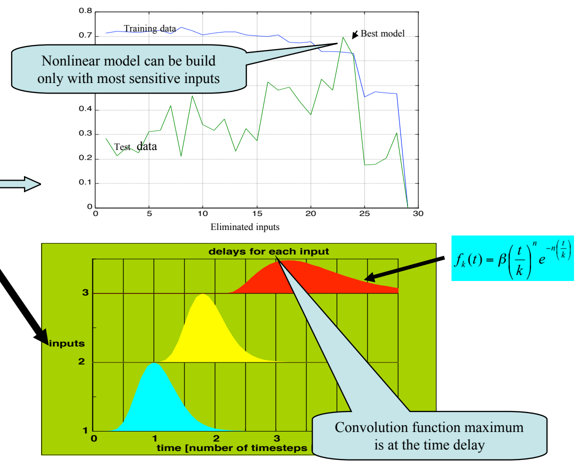
Integrated Methodology for Empirical Models Development



- Hybrid approach integrating multiple technologies exploits the strengths of each
- Advantages:
  - Fast development (days)
  - Robust performance (compact models)
  - Direct implementation in any Distributed Control System (no need for specialized software)
  - Very low capital cost (only if hardware for data collection is unavailable)
  - Low average cost of ownership (reduced development and maintenance cost)
  - Process engineers like it (preferable to black-box models)

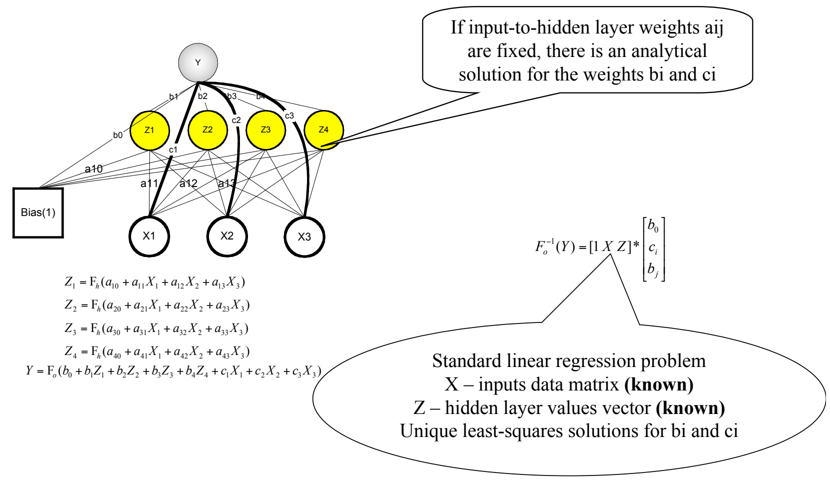
Steps Based on Analytic Neural Nets



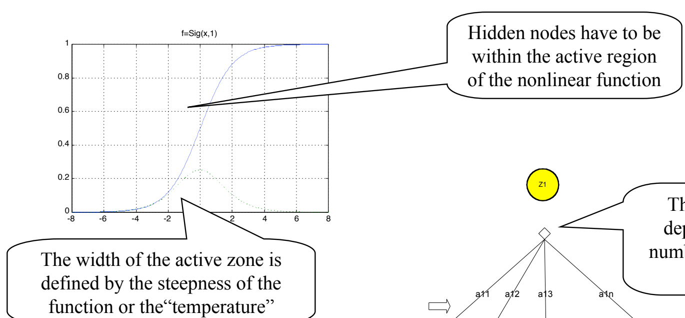


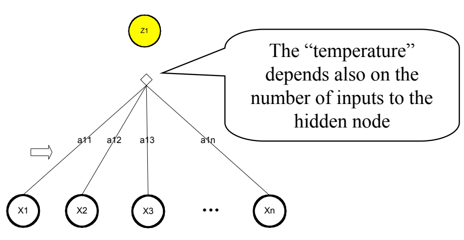
Objective: to supply GP with clean, informative, and parsimonious data set

Key idea behind analytic neural networks



Key technique for input-to-hidden layer initialization





Empirical expression for a normalized "temperature" of a sigmoid function

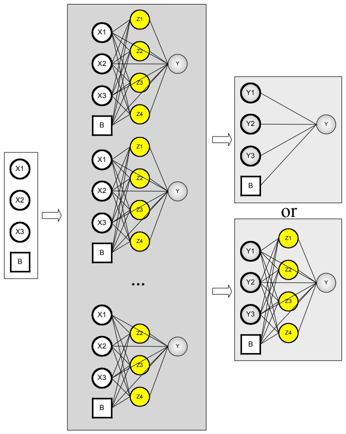
$$Tn = \eta . \frac{\log(2+\sqrt{3})}{\sqrt{ni - 0.5}}$$

Weights from the input-to-hidden layer are Sampled from a normal distribution

Analytic Neural Network Benefits

- **Robust** algorithm
  - No tunable parameters
  - One **global** optimum
- Very **fast**,
  - possible to use a whole range of cross-validation principles from statistics
  - No longer an NP-complete problem
- Strong **theoretical foundation**
  - statistical learning theory
  - Direct measure for the model capacity (VC-dimension)

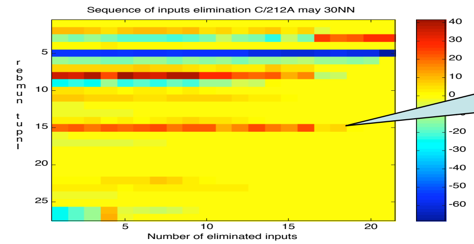
Stacked Analytic Neural Nets (SANN)



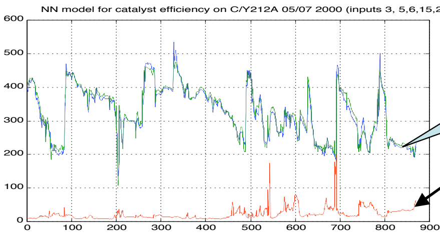
- Fast development
- Diverse subnet consensus indicator of model output quality
- Allows explicit calculations of input/output sensitivity
- Can handle time-delayed inputs by convolution functions
- Gives more reliable estimates based on multiple models statistics

Internally developed in Dow Chemical
by Guido Smits

An example of stacked analytical NN application - a model for catalyst efficiency



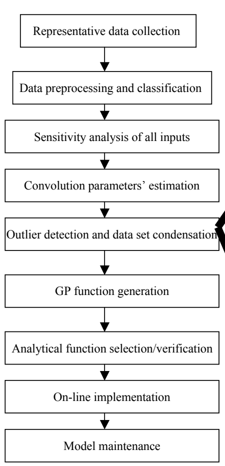
Sensitivity analysis of various process parameters on catalyst efficiency

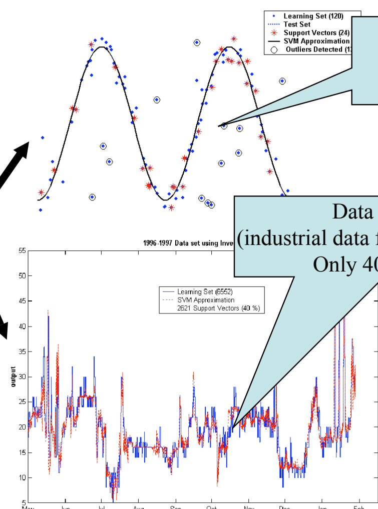


NN model performance with model disagreement indicator

Model disagreement indicator

Steps Based on Support Vector Machines

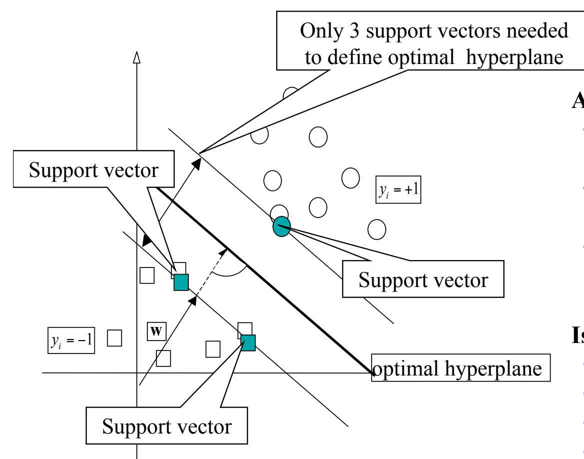




Reliable outlier detection

Data compression
(industrial data from a chemical reactor)
Only 40% of data used

Support Vector Machines for Classification



Advantages

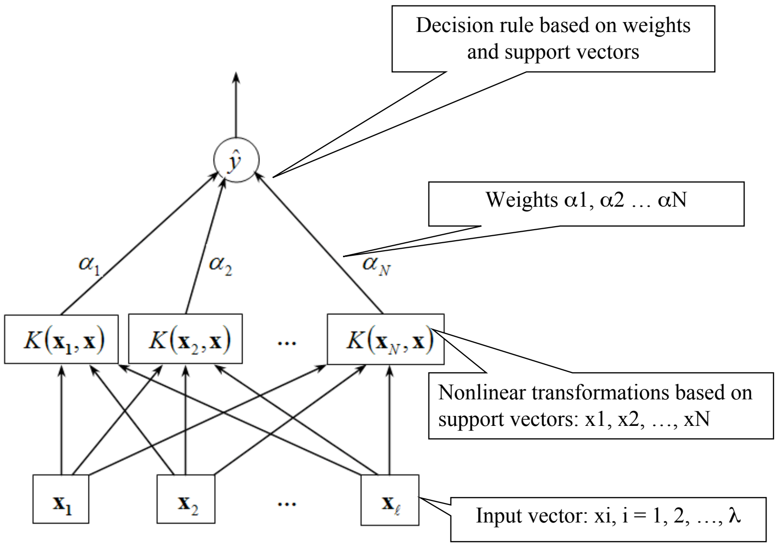
- Solid theoretical basis => Statistical Learning Theory
- Model building is based on global optimum
- Explicit control over model complexity

Issues

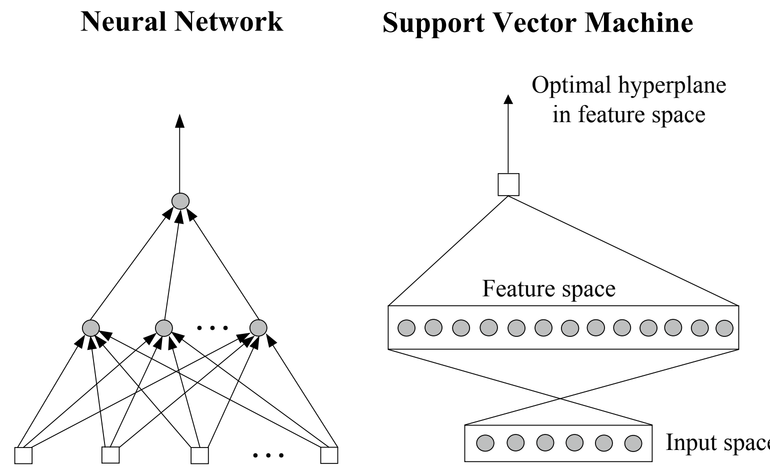
- *ad hoc* Kernel selection
- Complex theory
- No commercial software
- Computationally intensive

Key to robust modeling

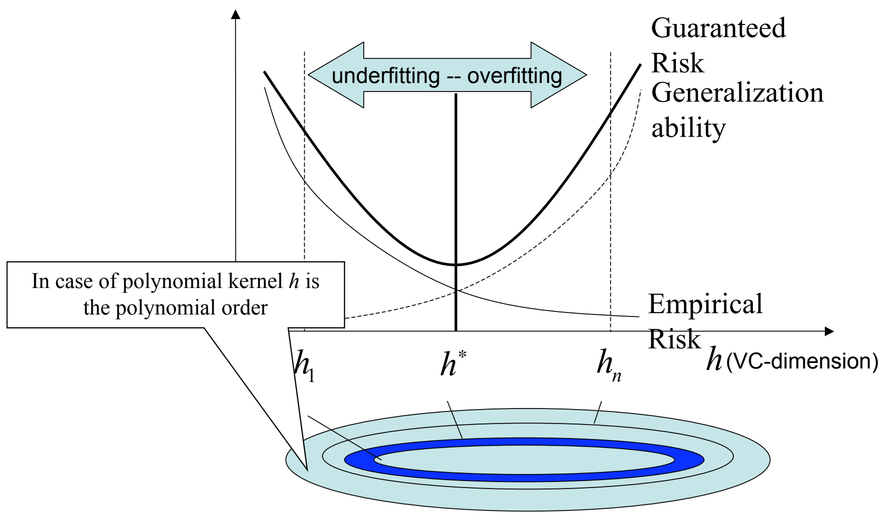
The generic scheme of SVM



Support Vector Machines and Neural Networks



Structural Risk Minimization Principle



Structural Risk Minimization Principle

- Trade-off between quality of approximation of the given data and the complexity of the approximating function.
- The VC-dimension is now a controlling variable
- Chooses the set of functions with the lowest VC-dimension for which minimizing the empirical risk gives the best bound on the actual risk.
- Minimize

$$R(\alpha) \le R_{\text{emp}}(\alpha) + \Phi\left(\tfrac{\ell}{h}\right)$$

Prediction error

Complexity

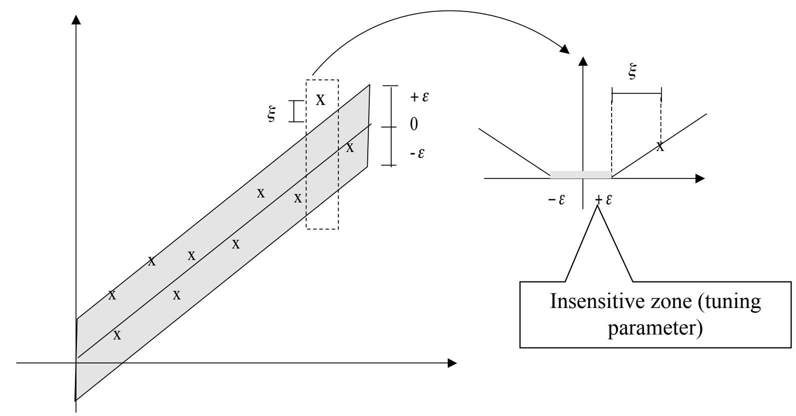
Where _ is the model parameter of interest, *l* is the sample size and *h* is the complexity measure

Structural Risk Minimization in learning algorithms

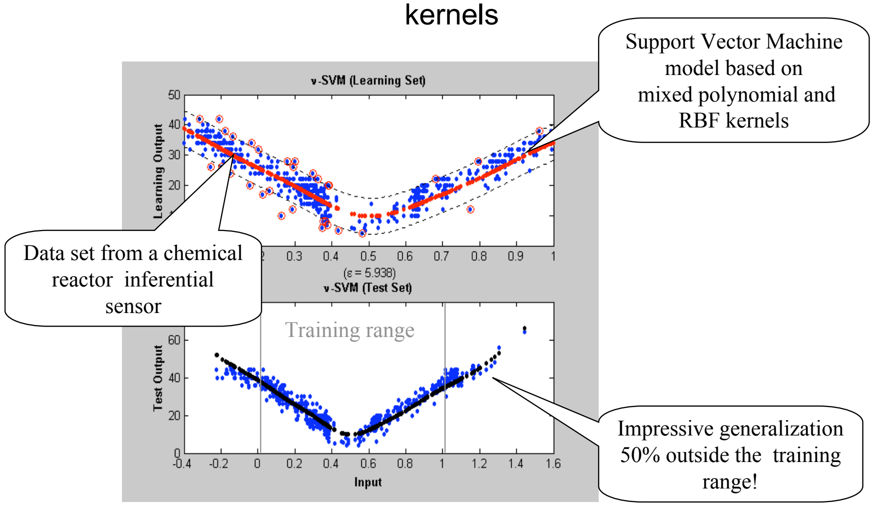
- Keep $\Phi(\frac{\ell}{h})$ fixed, minimize $R_{\text{emp}}(\alpha)$
  - Neural Networks
- Keep $R_{\text{emp}}(\alpha)$ fixed, minimize $\Phi(\frac{\ell}{h})$
  - Support Vector Machines

Neural Networks and Support Vector Machines are two sides of the same coin

SVM for Regression: Constructing a tube

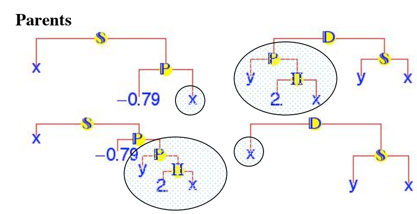


Generalization capabilities of SVM based on mixed kernels



Genetic Programming

Genome Tree Plots



Example of Crossover Operation

Phenotypes (Expressions)

Parents

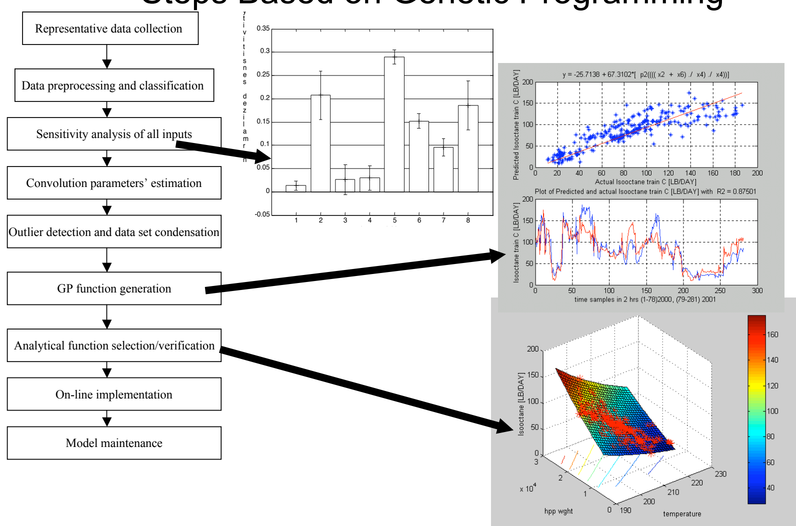
$-(-0.787701)^{x} + x \qquad \frac{y^{2x}}{-x+y}$

Children

$-(-0.787701)^{y^{2x}} + x \qquad \frac{x}{-x+y}$

- Based on artificial evolution of millions of potential nonlinear functions => survival of the fittest
- Many possible solutions with different levels of complexity
- The final result is an explicit nonlinear function
- Better generalization capabilities than neural nets
- Low implementation requirements
- Time delays
- Sensitivity analysis of large data sets
- Relatively slow (several hours of computational time)

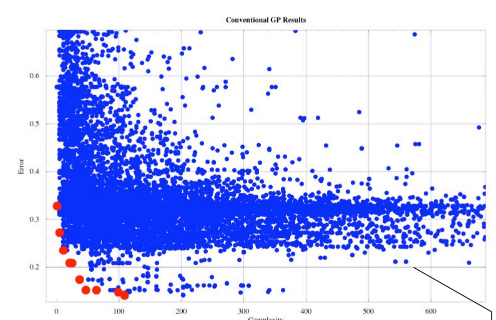
Steps Based on Genetic Programming



Problems with Genetic Programming

- Relatively Slow Discovery
  – Computational demands are intense
- Selection of "Quality" Solutions
  – Trade-off of Complexity vs. Performance
- Good-but-not-Great Solutions
  – Other nonlinear techniques (e.g., neural nets) outperform in raw performance
- Bloat
  – Parsimony control requires user intervention and is problem dependent

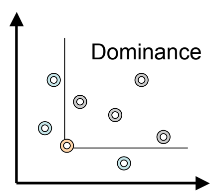
The Pareto Front

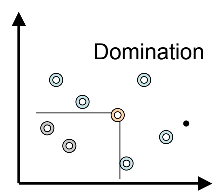


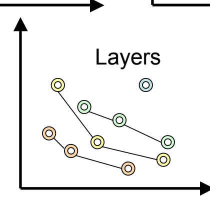
Note that much evolutionary effort is spent exploring high complexity & high fitness regions

- Identifies trade-off surface between competing objectives
  – e.g., performance vs. complexity
- Pareto front solutions are the best "bang-for-the-buck"
- Introns are punished automatically
- How can we exploit?

Pareto Performance

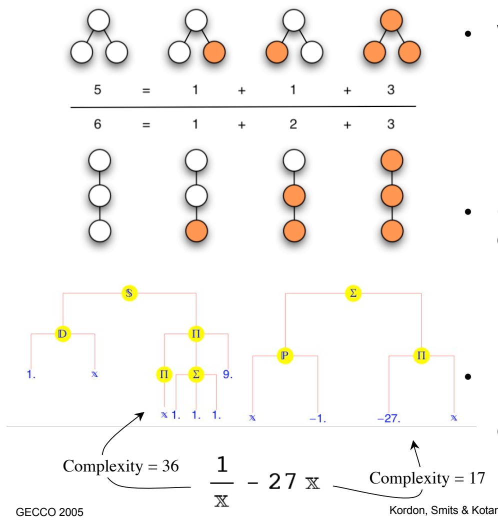






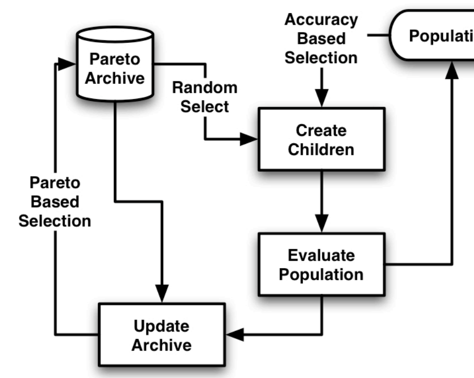
- Characterizing Pareto Performance
  – Dominance
  – Domination
  – Layer
  – Combinations …
- Computational Issues
  – Brute force is $M N^2$
  – Can do $M N \log_{M-1}(N)$ or $M N \log_{M-2}(N)$ if clever
    - M = # of objectives
    - N = population size
  – Computation demands need to be considered in algorithm design

Genome Complexity



- What is complexity?
  - # of nodes?
  - Tree depth?
  - Included functions?
  - Number of variables?
  - Combinations?
- Chosen function is sum of sum of node counts
  - Provides more resolution at low end of complexity than simply using node count
  - Rewards fewer layers
- Real goal is to characterize the (relative) "smoothness" of the evolved function

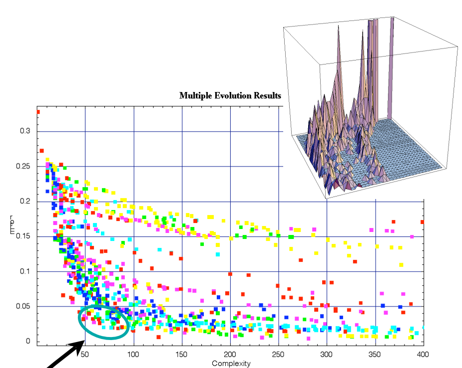
Pareto GP Algorithm



- Select from population based upon model accuracy
- Select randomly from Pareto archive
- Cascades …
  - Pareto archive maintained
  - Population wiped out (fresh genes!)
- Independent runs with independent archives for diversity
- There are other variants along these lines

Pareto GP Features

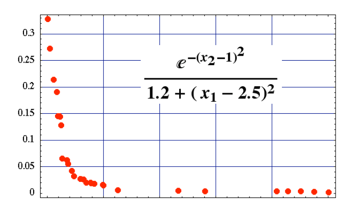
- 60-100x speedup relative to prior symbolic regression implementation
- Improved solutions
- Implicit intron suppression
- Facilitates …
  - Generation of diverse ensembles
  - Metavariable identification
  - Faster post-run analysis



Solutions of similar performance but diverse structure

Pareto Front

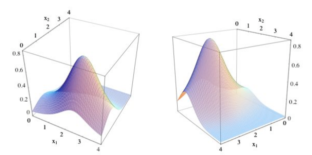
- A run tends to fully explore a foundation structure
- Independent evolutions will result in different (but still fit) structures
- Cascading results from independent evolutions seems to be beneficial
- Note that we are not strictly restricted to the Pareto front in selecting models -- many models may be "good enough" and have the benefit of being structurally different and diverse



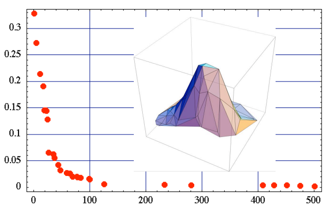
Out[93]//TableForm=

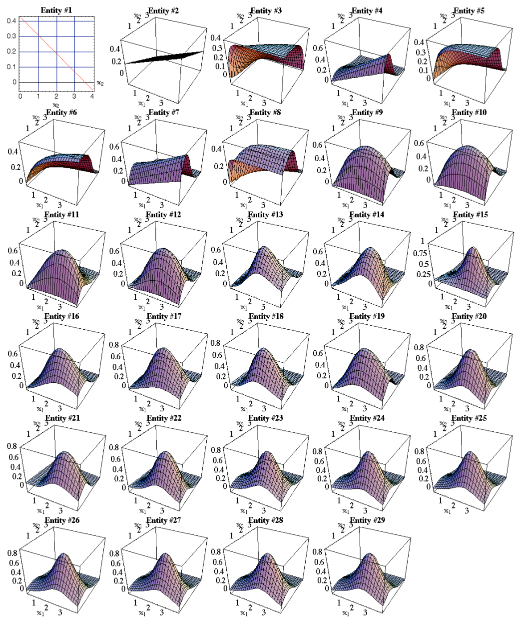
	Norm.	Adj.	Raw	Complexity	Vars	Function
1	0.673	0.672	0.672	1	x2	x_2
2	0.729	0.728	0.728	5	x1 x2	$x_1 - x_2$
3	0.787	0.786	0.786	11	x1 x2	$4.^{-x_1} x_2^{x_1}$
4	0.811	0.81	0.81	17	x1 x2	$x_2^{-x_2}(x_1+x_2)$
5	0.857	0.855	0.855	19	x1 x2	$0.225^{x_1} x_2^{-x_2}$
6	0.857	0.855	0.855	22	x1 x2	$\frac{x_1}{x_2^{x_2}+x_1+0.335}$
7	0.857	0.856	0.856	23	x1 x2	$x_2^{-x_2}(x_1+3.\,x_2)$
8	0.874	0.872	0.872	25	x1 x2	$(x_1 x_2^{-x_2})^{\frac{x_2}{x_1}}$
9	0.936	0.934	0.934	27	x1 x2	$-(x_1-5.)\,x_1\,x_2^{0.753-1.\,x_2}$
10	0.939	0.937	0.937	35	x1 x2	$-(x_1-5.)\,x_1\,x_2^{0.568-0.753\,x_2}$
11	0.947	0.945	0.945	37	x1 x2	$\left((5.281-x_1)^{4.^{-x_2}\,x_1}\right)^{x_2}$
12	0.96	0.958	0.958	44	x1 x2	$(x_1-5.)^{2.}\,x_1^{2.}\,x_2^{0.753-1.\,x_2}$
13	0.97	0.968	0.968	48	x1 x2	$\frac{x_1}{x_2^{x_2}-x_2+(x_1-2.)^{2.}+1.219}$
14	0.975	0.973	0.973	59	x1 x2	$\left((5.281-x_1)^{1.281^{x_2}-x_2^{x_2}}\right)^{x_1}$
15	0.976	0.974	0.974	65	x1 x2	$4.^{x_2}\left(\frac{1.}{x_2^{x_2}+x_1^{2.}-5.\,x_1+6.}\right)^{x_2}$
16	0.982	0.98	0.98	69	x1 x2	$4.099^{\frac{x_1}{(1.099^{x_1})^{x_1}+x_2^{x_2}-x_2}}$
17	0.982	0.98	0.98	77	x1 x2	$(x_2+4.099)^{\frac{x_1}{(1.099^{x_1})^{x_1}+x_2^{x_2}-x_2}}$
18	0.983	0.981	0.981	78	x1 x2	$2.\,(3.138^{x_2})^{2.-x_2}\,x_1^{4.-x_1}-x_2$
19	0.984	0.981	0.981	84	x1 x2	$2.^{x_1(2.^{1.-x_1}\,x_1-1.)}\,(3.^{x_2})^{2.-x_2}\,x_1$
20	0.986	0.984	0.984	99	x1 x2	$2.\,(3.138^{x_2})^{2.-x_2}\,x_1^{2.^{1.-x_1}\,x_1^{2.}-x_1+2.}$
21	0.987	0.985	0.985	100	x1 x2	$(3.^{x_2})^{2.-x_2}\,x_1^{5.-x_1}-x_1-x_2$
22	0.996	0.994	0.994	126	x1 x2	$2.\,(3.138^{x_2})^{2.-x_2}\,x_1^{2.^{1.-x_1}\,x_1^{2.}-x_1+2.}-x_2$
23	0.997	0.995	0.995	233	x1 x2	$\frac{2.\,x_1+2.\,x_2+3.}{(x_2-1.)\,x_2^{2.}+\frac{1.}{3.}\,x_1\,(x_2-1.\,(x_1-3.261)\,x_1)\left(\frac{1.}{x_1+3.\,x_2}-1.\right)+3.261}$
24	0.998	0.996	0.996	281	x1 x2	$\frac{x_2+x_1\left(\frac{2.\,x_2}{x_2^{2.}+1.}+1.\right)+3.}{(x_2-1.)\,x_2^{2.}+\frac{1.}{3.}\,x_1\,(x_2-1.\,(x_1-3.261)\,x_1)\left(\frac{1.}{x_1+3.\,x_2}-1.\right)+3.261}$
25	0.999	0.996	0.996	409	x1 x2	[illegible]
26	0.999	0.996	0.996	429	x1 x2	[illegible]
27	0.999	0.996	0.996	451	x1 x2	[illegible]
28	1.	0.997	0.997	475	x1 x2	[illegible]
29	1.	0.998	0.998	501	x1 x2	[illegible]

Pareto Front

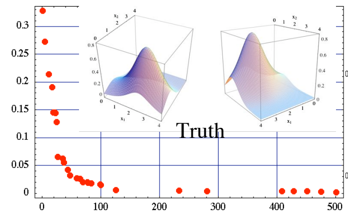


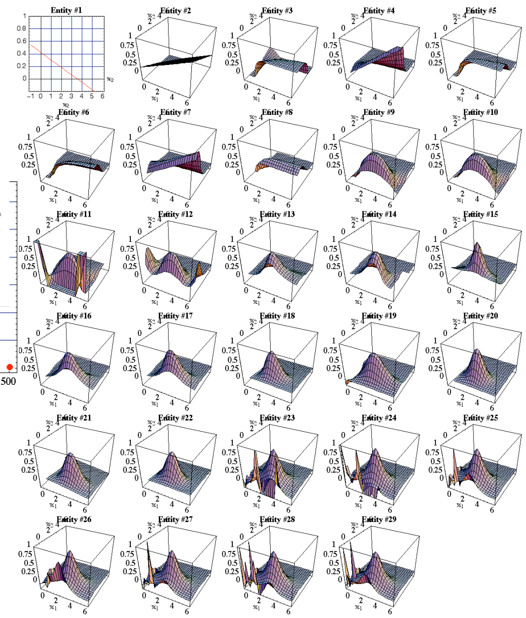
Truth $\frac{e^{-(x_2-1)^2}}{1.2+(x_1-2.5)^2}$





Extra-polation

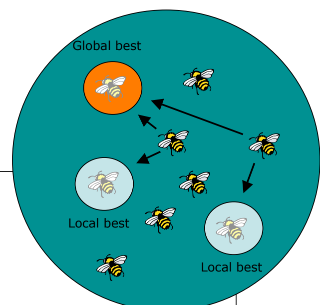




- Note the pathologies at high complexity when extrapolating

Particle swarm optimization

An efficient technique to find the global optimum
for model inversion and non-linear parameter estimation



At each time step *t*

For each particle *i*

Update the position change (velocity)

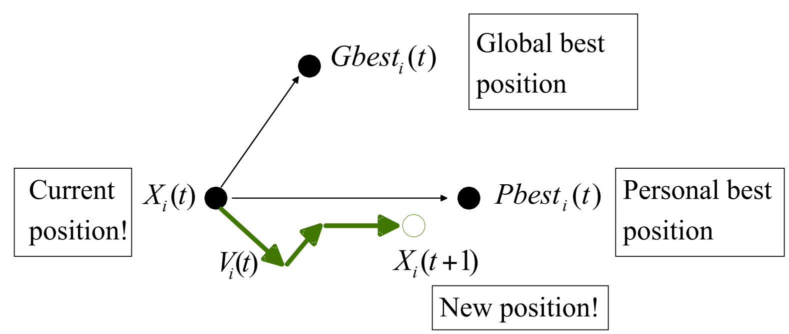
$$V_i(t+1) = \chi \cdot (V_i(t) + c_1 \cdot rand(0,1) \cdot (P_i(t) - X_i(t)) + c_2 \cdot rand(0,1) \cdot (P_g(t) - X_i(t))$$

Then move $X_i(t+1) = X_i(t) + V_i(t+1)$

Note: - stochastic component

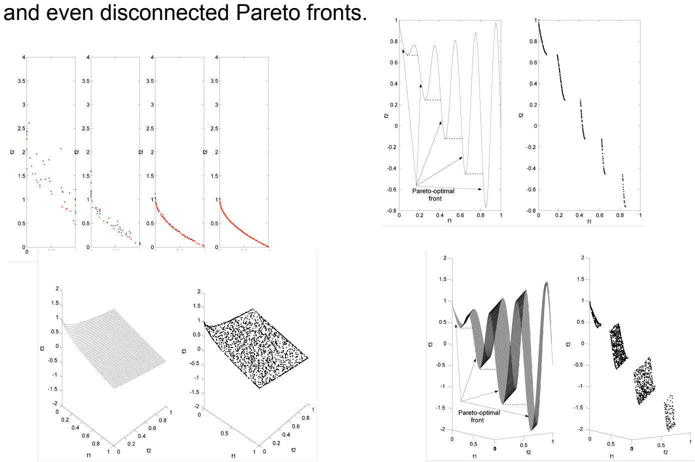
- parameters c_1, c_2, χ default values (2.05, 2.05, 0.73)

Particle's Movement – A Compromise

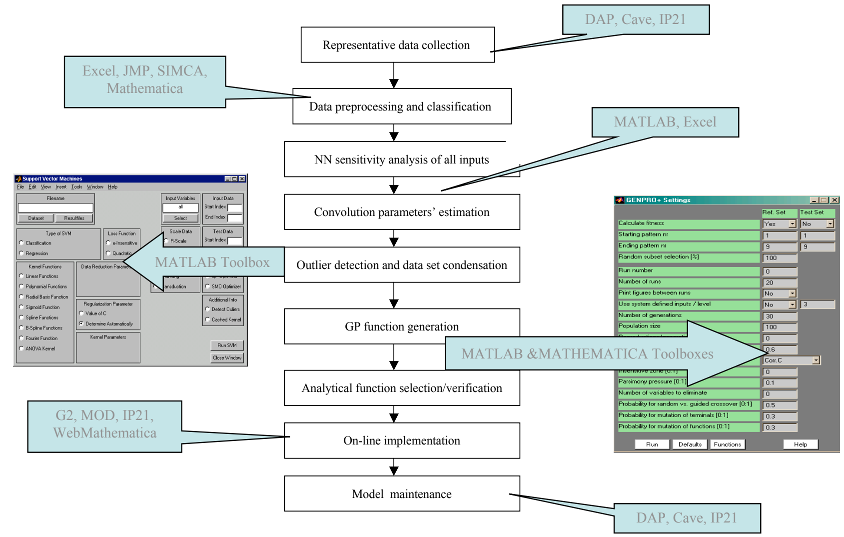


Multi-Objective PSO

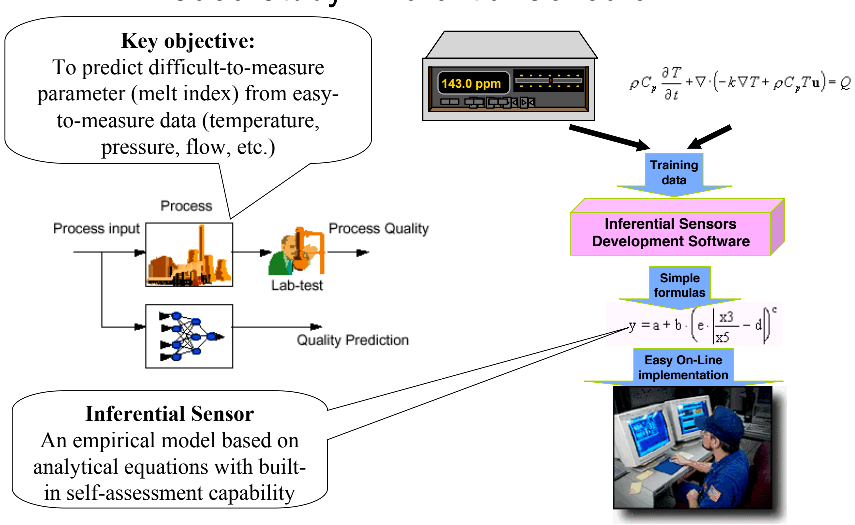
Efficient technique to determine the Pareto front for problems with convex, non-convex and even disconnected Pareto fronts.



Software tools

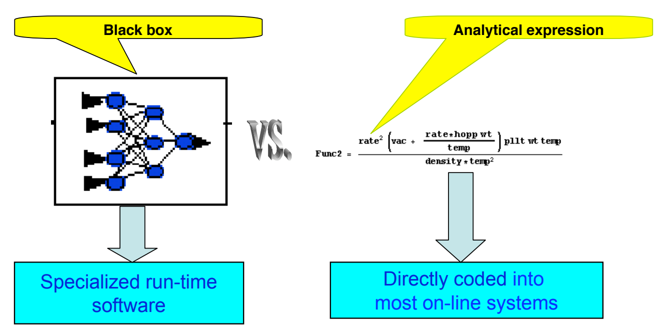


Case Study: Inferential Sensors

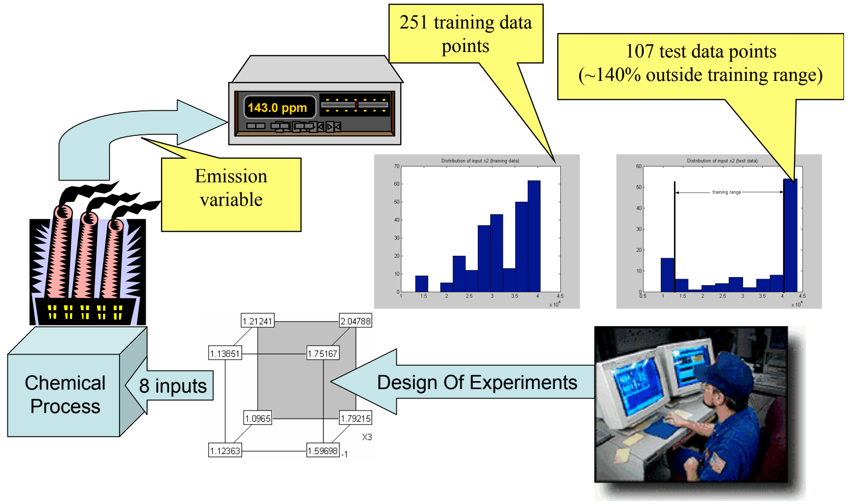


Issues with neural net-based inferential sensors

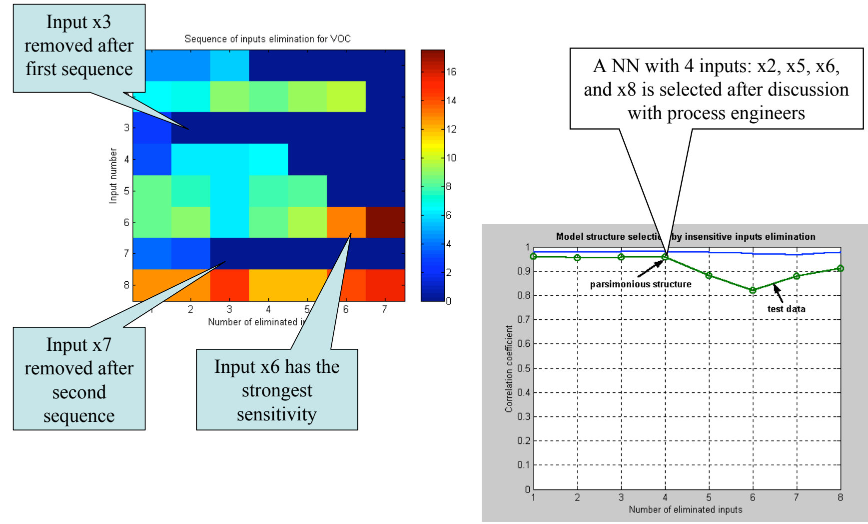
Issues with existing neural net-based inferential sensors:
- High sensitivity to process changes
- Frequent re-training
- Complicated development & maintenance
- Low survival rate after 3 years in operation
- Engineers hate black-boxes



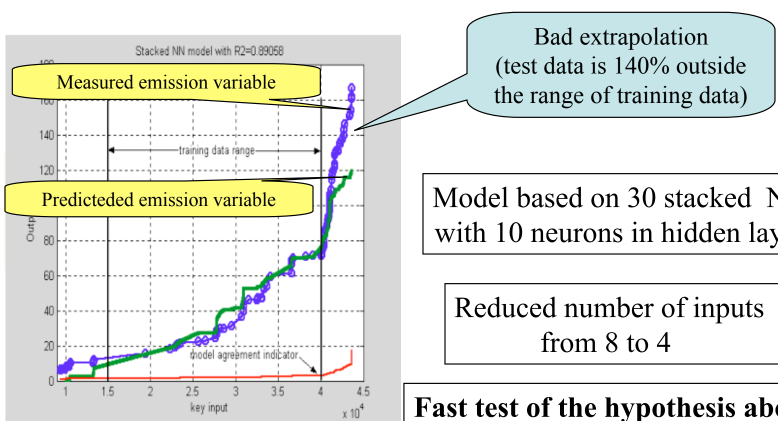
Inferential sensor for emission monitoring: A case study Data Collection



Inferential sensor for emission monitoring: A case study Sensitivity analysis by SANN



Inferential sensor for emission monitoring: A case study (SANN model performance)

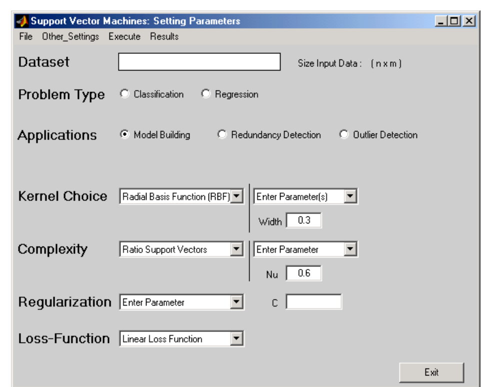


Model based on 30 stacked NN with 10 neurons in hidden layer

Reduced number of inputs from 8 to 4

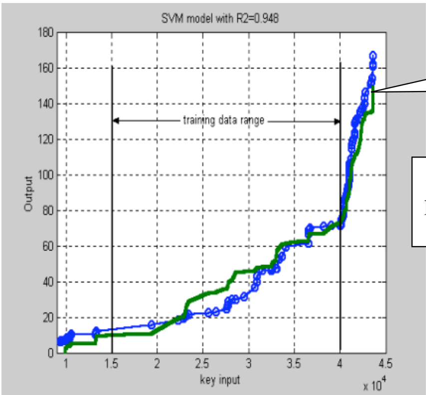
Fast test of the hypothesis about potential nonlinear relationship (in 20-30 min)

Inferential sensor for emission monitoring: A case study (SVM parameters)



Parameters:
% support vectors: 10
C = 10^6
Mixed Kernels: Polynomial and RBF
Range of Polynomial kernels: 1-3
Range of RBF kernel: 0.25-0.75
Range of ratio 0.5 – 0.99

Inferential sensor for emission monitoring: A case study (SVM model performance)

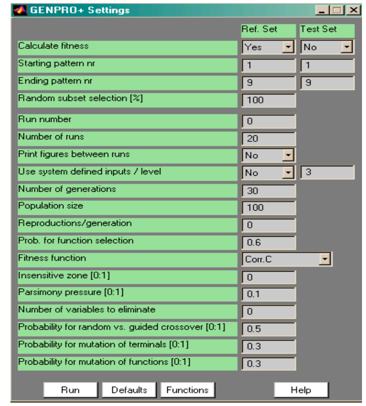


Impressive extrapolation (test data is 140% outside the range of training data)

Model based on a mixture of 2nd order polynomial global kernel and RBF local kernel with width of 0.5 and ratio of 0.95

Reduced number of training data points from 251 to 34 (based on support vectors)

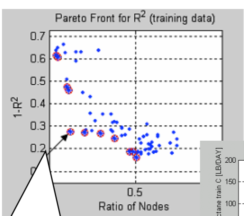
Inferential sensor for emission monitoring: A case study (GP parameters)



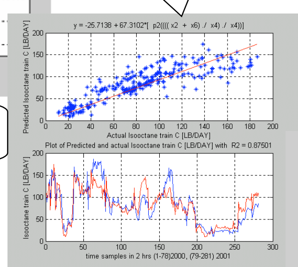
Parameters for a GP simulated evolution

Reference data	:34
Random subset selection [%]	:100
Number of runs	:20
Population size	:500
Number of generations	:100
Probability for function as next node	:0.6
Optimization function	:Corr.
Parsimony pressure	:0.1
Prob. for random vs guided crossover	:0.5
Probability for mutation of terminals	:0.3
Probability for mutation of functions	:0.3

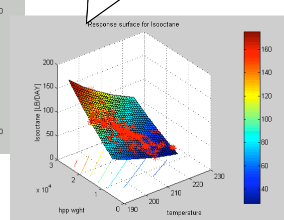
Inferential sensor for emission monitoring: A case study (Selected symbolic regression model)



Simple expression with acceptable performance (R2 = 0.87)

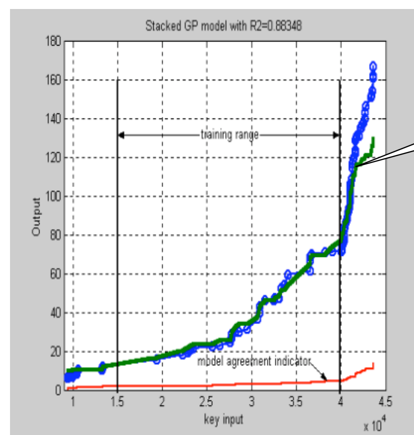


Response surface of model according to process physics



Selected model on Pareto front

Inferential sensor for emission monitoring: A case study (Final solution: Stacked Symbolic Regression model)

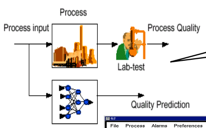


In operation since August 2001

Model based on 8 Stacked Symbolic Predictors

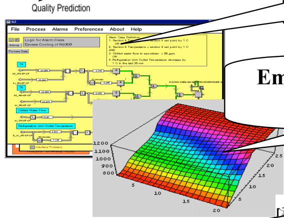
Shorter evolutionary process based on 8.44% of the original training data set

Key application areas

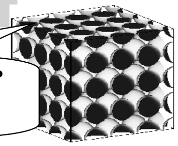


Robust Inferential Sensors
Mass-scale on-line empirical models

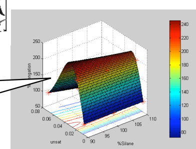
Automated Operating Discipline
Consistent intelligent on-line supervision



Empirical Emulators of Fundamental Models
Effective on-line process optimization



Fundamental model building based on GP
Accelerated new product development

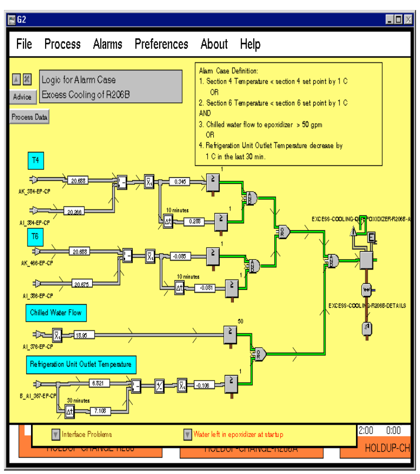


Nonlinear DOE based on GP
Minimizing expensive process experiments

EC Applications in Dow Chemical

Application Domains	Examples
Material Design	• Color Matching • Appearance Engineering • Polymer Design • Synthetic Leather
Materials Research	• Diverse Chemical Library Selection • Fundamental Model Building • Reaction Kinetics Modeling • Combi-Chem Catalyst Exploration • Combi-Chem Data Analysis
Production Design	• Acicular Mullite Emulator • EDC/VCM Nonlinear DOE • Bioreactor Optimization
Production Monitoring & Analysis	• Epoxy Holdup Monitoring • Isocyanate Level Estimation • FTIR Calibration Variable Selection • Poly-3 Volatile Emission Monitoring • Epoxy Intelligent Alarm Processing • PerTet Emulator for Online Optimization • Emissions Monitoring
Business Modeling	• Diffusion of Innovation • Hydrocarbon Trading & Energy Systems Optimization • Scheduling Heuristics • Plant Capacity Drivers

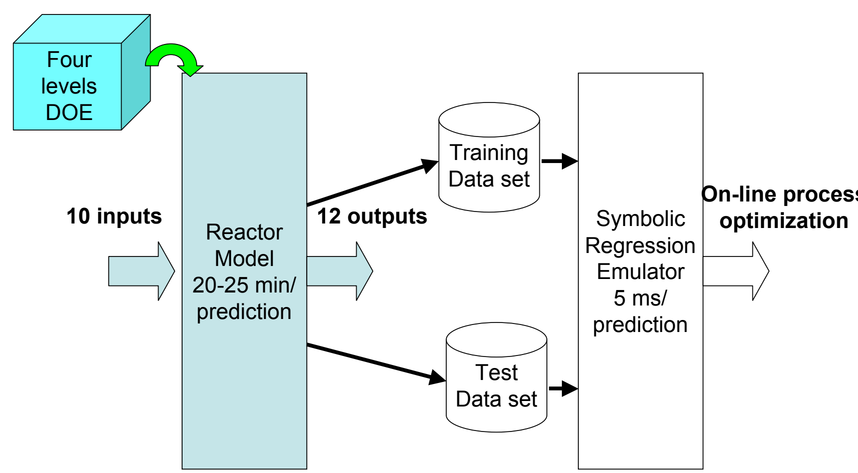
Automating Operating Discipline



- Heuristic rules defined verbally by process engineers/operators
- holdup predictor designed by stacked analytic NN and GP
- all decision blocks have fuzzy thresholds defined by membership functions
- simple empirical models and mass balances
- fundamental model predictions are used in the heuristic rules

- reduced major shutdowns
- reduced lab sampling

Emulator for optimization of an industrial chemical process

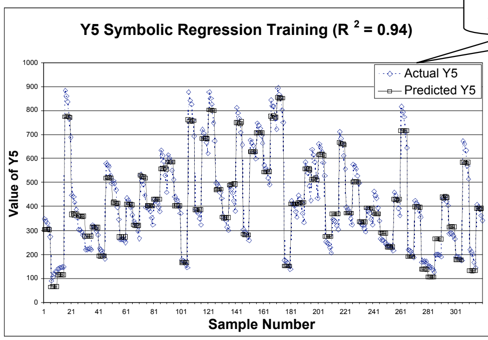


Symbolic regression-based emulator's performance

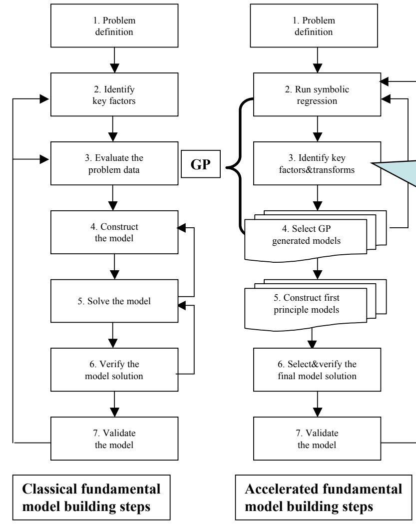
Simple expression for on-line implementation

$$Y5 = 3x_9 + \frac{6x_3 + x_4 + x_5 + 2x_6 + x_2 x_9 - 3x_{10} - \frac{x_2 - 3x_3 - x_5 + \sqrt{x_6} - 2x_7 - x_9 + x_{10}}{\mathrm{Log}[x_2]^2}}{x_2}$$

Acceptable performance

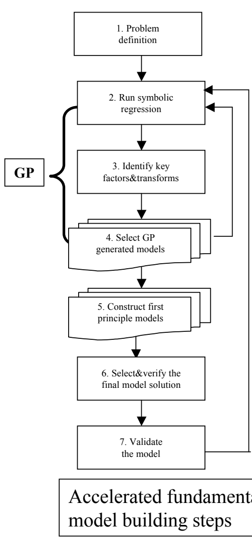


Accelerated Fundamental Model Building Based on GP

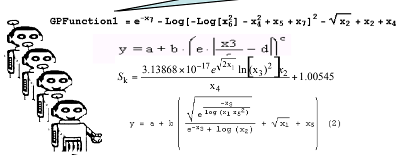


Can we accelerate hypothesis search by simulated evolution?

Fundamental Model Building Based on GP



Run simulated evolution before beginning fundamental modeling



Virtual modelers

The evolutionary process identifies the key input variables as well as natural groupings & relationships. Combining this with a domain knowledge and first-principles insights is very powerful.

Approaches to accelerate fundamental model building process

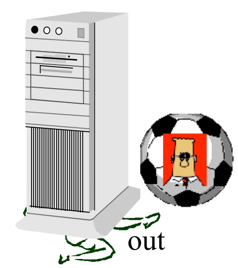
AI approach

Mimic the expert

Reduce hypothesis search by GP

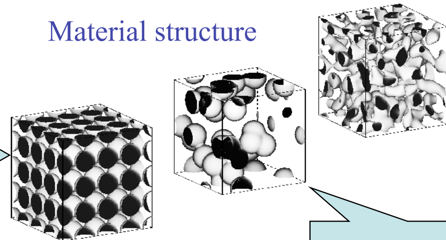
Maximize creativity of the expert

GP as automated invention machine



Eliminate the expert

The problem of structure-properties in fundamental modeling

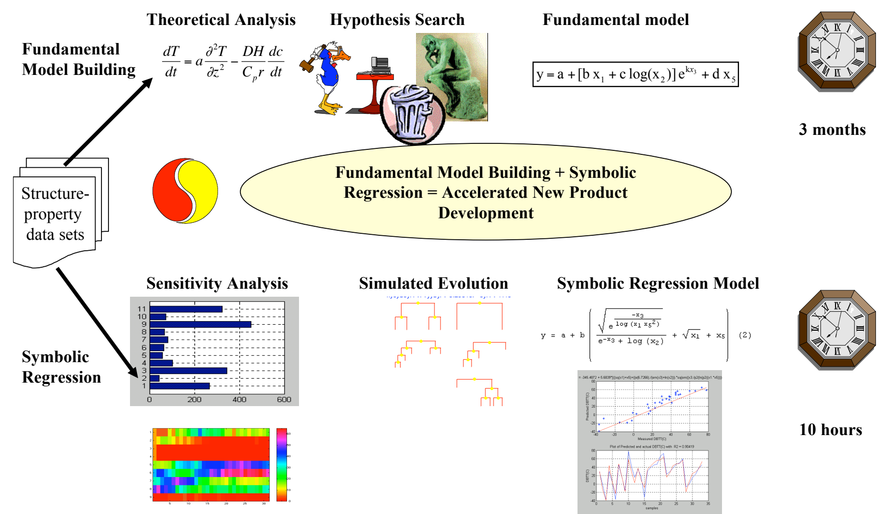


Properties:
- molecular weight
- particle size
- crystallinity
- volume fraction
- material morphology
- etc.

Key modeling effort for new product development

Modeling issues:
- nonlinear interaction
- large number of preliminary expensive experiments required
- large number of possible mechanisms
- slow fundamental model building
- insufficient data for training neural nets

Case Study with Structure-Property Relationships



Results from hypothesis search
Selected symbolic regression empirical model

Fundamental model

$$\boxed{y = a + [b\,x_1 + c\log(x_2)]\,e^{kx_3} + d\,x_5}$$

Selected empirical model

GP-generated empirical model captured correctly the functional forms of the fundamental model

Square root form for x1

$$y = a + b\left(\frac{\sqrt{e^{\frac{-x_3}{\log(x_1 x_5^2)}}}}{e^{-x_3} + \log(x_2)} + \sqrt{x_1} + x_5\right)$$

Linear form for x5

Exponential form for x3

Logarithmic form for x2

Comparative Analysis of Symbolic Regression in Fundamental Model Building

Advantages of Symbolic Regression

Model Development Speed
- 10 hours vs. 3 months

Summarize Multivariate Data
- convert data into equations to facilitate human insight
- can explore parameter sensitivity and play what-if games

Accuracy
- achieved > 90% correlation with experimental results

Identify Key Variables and Transforms
- with the exception of x1, symbolic regression captured correctly all other functional forms in the model

Suggest Physical Mechanisms
- evolved expressions and equation "building blocks" may be interpreted from a first-principles viewpoint

Suggest Future Experiments
- optima in evolved expressions may be validated in future experiments

Disadvantages of Symbolic Regression

Blind to Physics and Chemistry
- genetic programming does not currently take into account the physical or chemical laws
- expressions may have no physical meaning - mathematical consistency is how fitness is defined
- inclusion of physical constraints is a research topic

Garbage-In/Garbage Out
- appropriate variables must be supplied
- data is assumed to be accurate
- operational range should be covered

Experts (Scientists) are Still Required

domain expert is an absolute must for interpretation of evolved expressions

The domain expert delivers the final fundamental model

GP and Design Of Experiments (DOE) Models Showing Lack of Fit

Situations of Lack of Fit

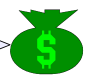
1. Simple factorial DOE
Enough experiments to fit first order model

$$y = \hat{a}_o + \sum_{i=1}^{k} \hat{a}_i x_i + \sum_{i<j}\sum \hat{a}_{ij} x_i x_j$$

Classical approach if LOF add experiments to fit second order model

$$S_k = \beta_o + \sum_{i=1}^{k} \beta_i x_i + \sum \beta_{ii} x_i^2 + \sum_{i<j}\sum \beta_{ij} x_i x_j$$

More costly experiments



2. A response surface DOE
already had all experiments to fit second order model

$$S_k = \beta_o + \sum_{i=1}^{k} \beta_i x_i + \sum \beta_{ii} x_i^2 + \sum_{i<j}\sum \beta_{ij} x_i x_j$$

Classical approach if LOF no alternative (use model as it is)

Suggested approach: Use GP to transform inputs

1. Generate GP models

$$S_k = \frac{3.13868 \times 10^{-17} e^{\sqrt{2x_1}} \ln\left[(x_3)^2\right] x_2}{x_4} + 1.00545 \quad (2)$$

Selected solution

2. Generate input transforms

Variable transformations suggested by GP model

Original Variable	Transformed Variable
x_1	$Z_1 = \exp\left(\sqrt{2x_1}\right)$
x_2	$Z_2 = x_2$
x_3	$Z_3 = \ln[(x_3)^2]$
x_4	$Z_4 = x_4^{-1}$

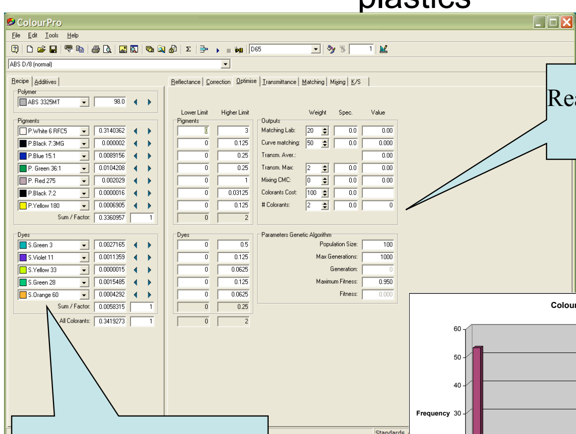
3. Fit response surface model in transformed variables

$$S_k = \beta_o + \sum_{i=1}^{4} \beta_i Z_i + \sum_{i<j}\sum \beta_{ij} Z_i Z_j + \sum_{i=1}^{4} \beta_{ii} Z_i^2$$

Source	DF	Sum of Square	Mean Square	F Ratio
Lack of Fit	2	0.00049190	0.000246	2.2554
Pure Error	2	0.00021810	0.000109	Prob > F
Total Error	2	0.00071000		0.3072
				Max RSq
				0.9999

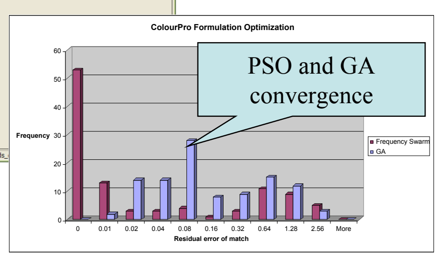
No Lack Of Fit (p=0.3037)

PSO application: Optimizing color spectrum of plastics



Real-time optimization in 2-3 seconds

Multiple-objective PSO with 15 variables



PSO and GA convergence

Other PSO applications

- Drug release predictor
  - 6 parameters
  - population size = 30
  - optimization time: ~ 30 seconds
- Foam acoustics performance predictor
  - 8 parameters
  - population size = 50
  - optimization time: ~ 5 seconds
- Crystallization kinetics predictor
  - 4 parameters
  - population size = 30
  - optimization time: ~ 2 seconds

Open Issues

Open Issues

- **Complexity Control & Smoothness Characterization**
- **Diversity Detection**
- **Identifying Metavariables**
- **Convert Hard GP Problems into Easy Ones**
- **Should we be doing Cultural Programming?**
- **Blending Heuristics & Prior Knowledge**

Summary

- Evolutionary Computing can create significant value to industry by reducing model development time and model exploitation cost
- Integrating EC with Neural Networks, Support Vector Machines, and Statistics is recommended for successful industrial applications
- This strategy works for many real applications in the chemical industry
- The key application areas are:
  - Inferential sensors
  - Improved process monitoring and control
  - Accelerated new product development
  - Effective design of experiments
- And this is only the beginning …

Acknowledgement

We would like to acknowledge the contribution of the following researchers from The Dow Chemical Company:

Alex Kalos

Kip Mercure

Flor Castillo

Elsa Jordaan

Leo Chiang

Irina Graf

References

1. M. Kotanchek, G. Smits, and A. Kordon, *Industrial Strength Genetic Programming*, In *GP Theory and Practice* (R. Riolo and B. Worzel-Eds), Kluwer, 2003.
2. A. Kordon, G. Smits, A. Kalos, and E. Jordaan, *Robust Soft Sensor Development Using Genetic Programming*, In *Nature-Inspired Methods in Chemometrics*, (R. Leardi-Editor), Elsevier, 2003.
3. Kordon A.K, G.F. Smits, E. Jordaan and E. Rightor, *Robust Soft Sensors Based on Integration of Genetic Programming, Analytical Neural Networks, and Support Vector Machines*, Proceedings of WCCI 2002, Honolulu, pp. 896 – 901, 2002.
4. Kotanchek M., A. Kordon, G. Smits, F. Castillo, R. Pell, M.B. Seasholtz, L. Chiang, P. Margl, P.K. Mercure, A. Kalos, *Evolutionary Computing in Dow Chemical*, Proceedings of GECCO'2002, New York, volume Evolutionary Computation in Industry, pp. 101-110., 2002
5. Kordon A. K., H.T. Pham, C.P. Bosnyak, M.E. Kotanchek, and G. F. Smits, *Accelerating Industrial Fundamental Model Building with Symbolic Regression: A Case Study with Structure – Property Relationships*, Proceedings of GECCO'2002, New York, volume Evolutionary Computation in Industry, pp. 111-116, 2002
6. Castillo F., K. Marshall, J. Greens, and A. Kordon, *Symbolic Regression in Design of Experiments: A Case Study with Linearizing Transformations*, Proceedings of GECCO'2002, New York, pp. 1043-1048.
7. Kordon A., E. Jordaan, L. Chew, G. Smits, T. Bruck, K. Haney, and A. Jenings, Biomass Inferential Sensor Based on Ensemble of Models Generated by Genetic Programming, accepted for GECCO 2004, 2004.
8. Smits G. and M. Kotanchek, *Pareto-Front Exploitation in Symbolic Regression*, In *GP Theory and Practice* (R. Riolo and B. Worzel-Eds), Kluwer, 2004.
9. Kordon A., A. Kalos, and B. Adams, *Empirical Emulators for Process Monitoring and Optimization*, Proceedings of the IEEE 11th Conference on Control and Automation MED'2003, Rhodes, Greece, pp.111, 2003.