Industrial Evolutionary Computing

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

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

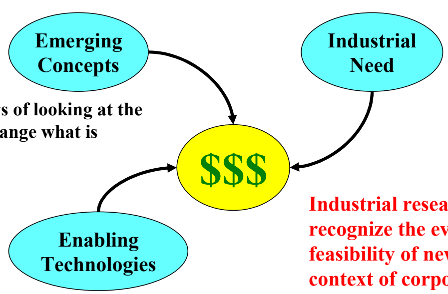
GECCO 2006

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
- Key Technologies
  - Analytic Neural Networks + Support Vector Machines + Genetic Programming + Particle Swarms + …
- Implementation Guidelines
- Integrate & Conquer
- Key Application Areas
- Open Issues & Research areas

Data Modeling
At the Intersection of Opportunity & Need



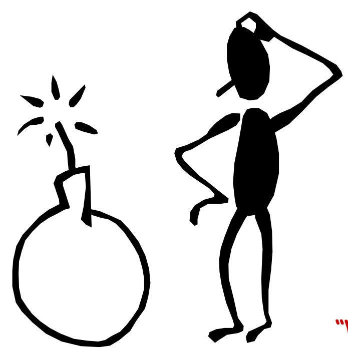
The economic context prioritizes the possibilities.

New ways of looking at the world change what is possible.

Industrial research needs to recognize the evolving potential and feasibility of new ideas within the context of corporate needs.

Technology & Price-Performance shifts enable implementing new concepts and implementing old concepts better.

Motivation



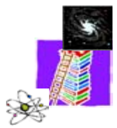
- Industry is great at collecting data … and then performing records retention
- Extracting insight from multivariate data is hard
- Time and money is being wasted

"We are drowning in information and starving for knowledge" – R.D. Roger

Academic vs. industrial data analysis

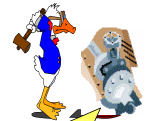
Transfer data into knowledge

Transfer data into value



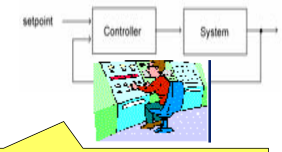
Special Features of Industrial Data Analysis

Operators intervention



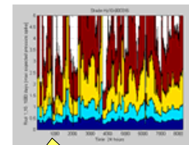
Operators manually modify the process

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

Models need to be developed & updated rapidly

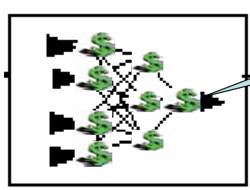
Most of models operate in real time

Intelligent Systems in Industrial Data Analysis: Lessons From the Past

inside
pentium
PROCESSOR

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

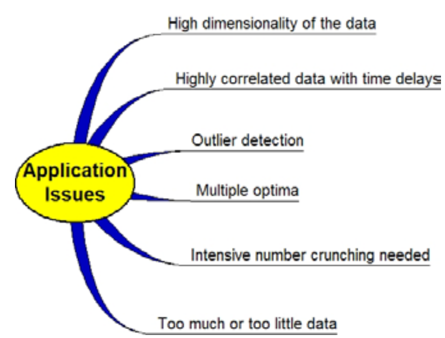
- **static rule-based models not linked to numerical world**
- **the politics of knowledge acquisition**
- **the efforts of knowledge acquisition**



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

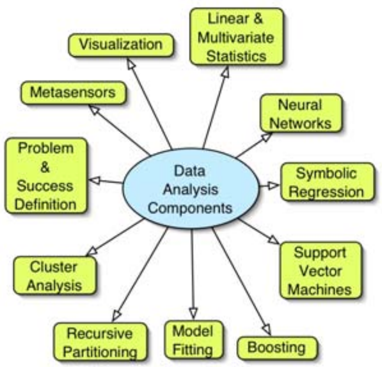
- **black-box models with inefficient structure**
- **fragile models and model validation**
- **maintenance nightmare**

Industrial Data Modeling Issues



"The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' (I found it!) but 'That's funny …'" — Isaac Asimov (1920 - 1992)

Industrial data analysis components



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

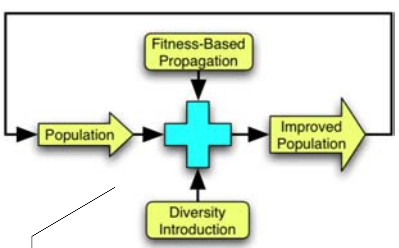
- Facilitate physical/mechanism insight and **understanding**
- **Summarize** data behavior
- Identify data **transforms** and metasensors
- Perform **variable selection**
- Enable response surface **exploration and optimization**
- **Visualize** behavior in the form of a symbolic expression

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

Competing/Complementary Technologies

- Linear Models
  - Linear in coefficients, not necessarily linear in model
  - Often "good enough" and simple
  - Well developed criteria and foundations in linear statistical analysis
  - Typically easy and fast to develop (unless subtleties are involved)
- Neural networks
  - Often good performance but lots of “trust me”
  - A good reference for nonlinear modeling potential
- Support Vector Machines
  - Useful for data compression to match information content
  - Computationally demanding
  - Unique nonlinear outlier detection capability
- Fuzzy Rules/Recursive Partitioning
  - Human interpretability — if simple
  - Can handle categorical data

Evolutionary Computing Theory



It is this simple!

Variants:

- Genetic Algorithms (GA)
- Evolutionary Strategies (ES)
- Evolutionary Programming (EP)
- Genetic Programming (GP)
- Particle Swarm Optimization (PSO)
- Gene Expression Programming (GEP)
- etc.

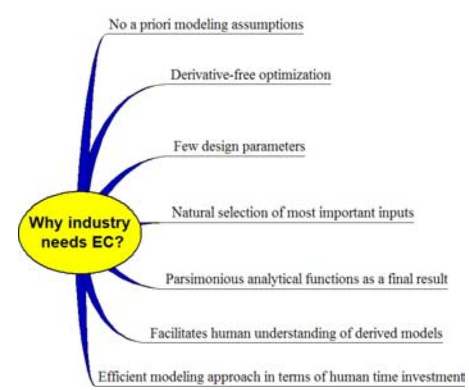
Genetic Programming

- Genome (genetic code) evolves
- Phenotype (realization) judged for fitness
- Goal is to evolve *programs* which solve problems
- The search space is *infinite*!
- Symbolic regression is one application of genetic programming

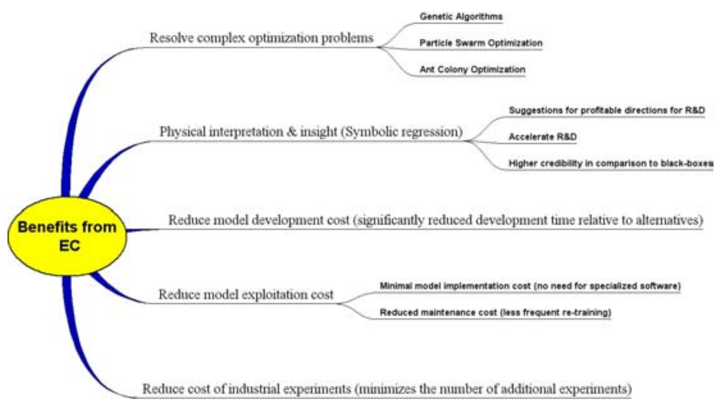
Symbolic Regression

- Goal is to identify expressions which summarize data
- NOT parameter fitting — discovery of both structure and parameters
- The search space is infinite!
- In practice, symbolic regression is part of an integrated methodology

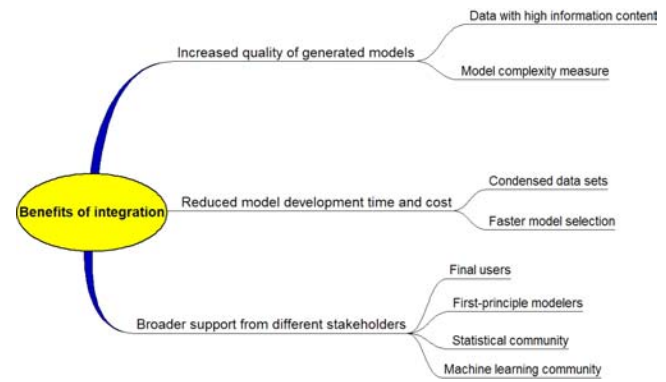
Why industry needs Evolutionary Computing?



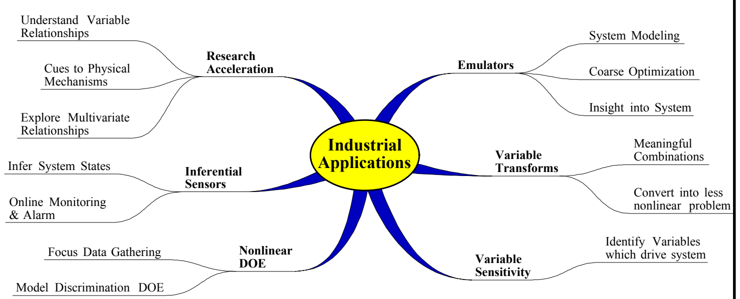
Economic benefits from Evolutionary Computing



Benefits of integrating Evolutionary Computing with other approaches



Application areas with impact

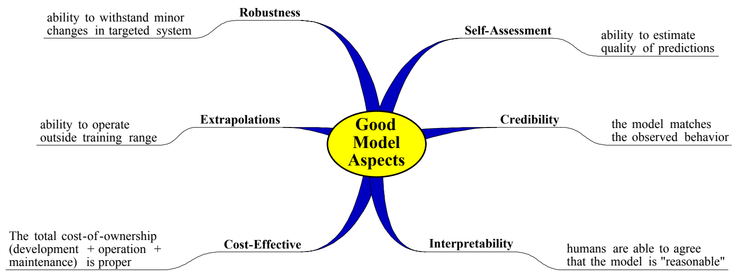


Implementation guidelines

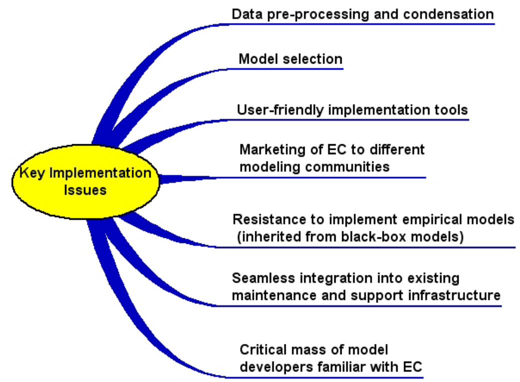
- Requirements for successful empirical modeling
- Key issues to be overcome
- Implementation strategy
- Implementation tools

Requirements for successful data-driven modeling

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

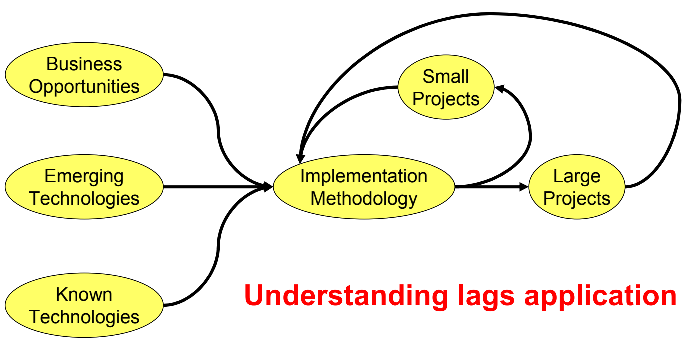


Key issues to overcome



“Good enough is the worst enemy of better”

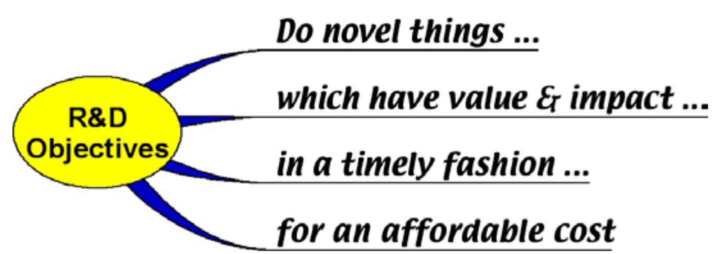
Implementation Strategy



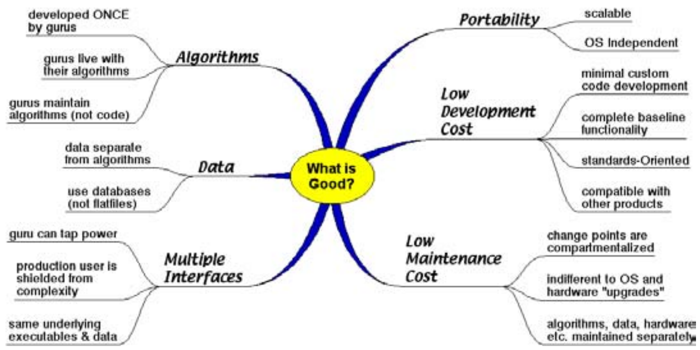
Understanding lags application

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

Corporate Research Objectives



Characteristics of a “Good” Analysis System

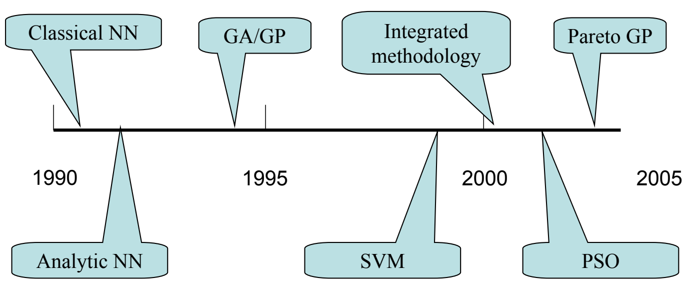


Implementation tools

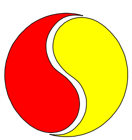
- Mathematica (Dow & Evolved Analytics developed)
  - Symbolic regression package
  - AutoAnalysisTools
  - Analytic neural networks
  - Particle Swarm Optimization (PSO)
  - Group Methods of Data Handling (GMDH)
- MATLAB (Dow developed)
  - Genetic Algorithms (GA)
  - Genetic Programmimg (GP)
  - PSO (single objective and multi-objective)
  - Analytic neural networks
  - Support vector machines
- Tools for model deployment
  - Delphi
  - WebMathematica
  - Excel
  - Process control systems

Using a commercial framework allows us to bring new concepts and technologies to bear while mitigating the development and long-term maintenance costs of exploiting those technologies.

Exploitation/Implementation Sequence of Computational Intelligence Approaches in Dow Chemical

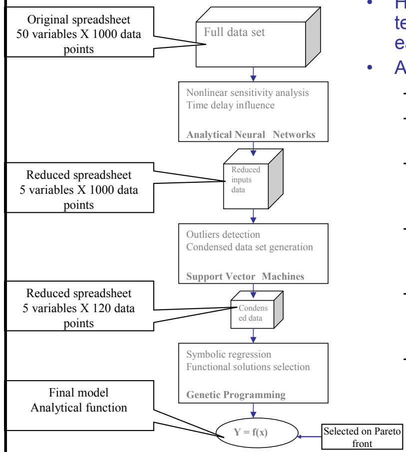


Integrate & Conquer



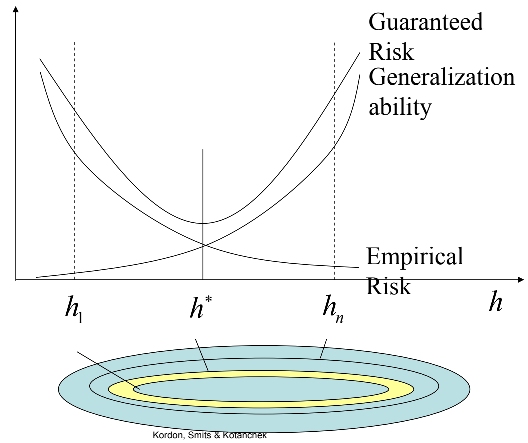
- Integrated methodology for successful EC implementation
- Related approaches
- A case study

Integrated Methodology 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)

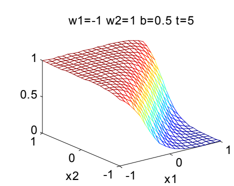
Structural Risk Minimization

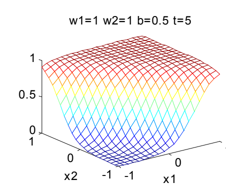


VC-dimension

- In general, VC-dimension does not coincide with the number of parameters (can be larger or smaller)
- VC-dimension of the set of functions is responsible for the generalization ability of learning machines
- Opens remarkable opportunities to overcome the “curse of dimensionality” (large number of parameters, but low VC-dimension)

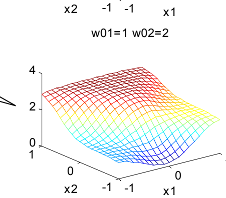
Two hidden nodes



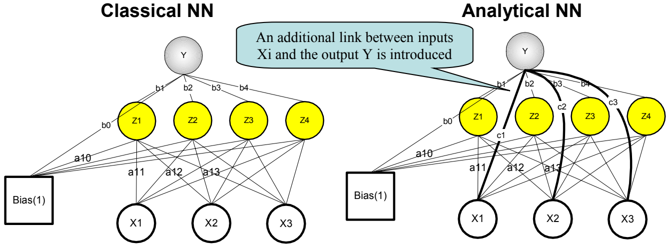


Any complex surface can be approximated by combining simple surfaces corresponding to a single hidden node

Combination:



Structural difference between classical and analytic neural networks



Hidden nodes calculation

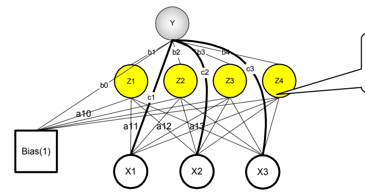
Classical NN:

$Z_1 = F_h(a_{10} + a_{11}X_1 + a_{12}X_2 + a_{13}X_3)$
$Z_2 = F_h(a_{20} + a_{21}X_1 + a_{22}X_2 + a_{23}X_3)$
$Z_3 = F_h(a_{30} + a_{31}X_1 + a_{32}X_2 + a_{33}X_3)$
$Z_4 = F_h(a_{40} + a_{41}X_1 + a_{42}X_2 + a_{43}X_3)$
$Y = F_o(b_0 + b_1Z_1 + b_2Z_2 + b_3Z_3 + b_4Z_4)$

Analytical NN:

$Z_1 = F_h(a_{10} + a_{11}X_1 + a_{12}X_2 + a_{13}X_3)$
$Z_2 = F_h(a_{20} + a_{21}X_1 + a_{22}X_2 + a_{23}X_3)$
$Z_3 = F_h(a_{30} + a_{31}X_1 + a_{32}X_2 + a_{33}X_3)$
$Z_4 = F_h(a_{40} + a_{41}X_1 + a_{42}X_2 + a_{43}X_3)$
$Y = F_o(b_0 + b_1Z_1 + b_2Z_2 + b_3Z_3 + b_4Z_4 + c_1X_1 + c_2X_2 + c_3X_3)$

Analytic neural networks have a fixed Capacity



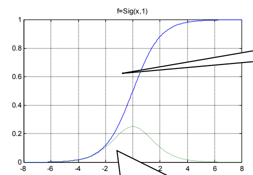
If input-to-hidden layer weights aij are fixed, there is an analytical solution for the weights bi and ci

$$F_o^{-1}(Y) = [1\ X\ Z]^* \begin{bmatrix} b_0 \\ c_i \\ b_j \end{bmatrix}$$

$Z_1 = F_h(a_{10} + a_{11}X_1 + a_{12}X_2 + a_{13}X_3)$
$Z_2 = F_h(a_{20} + a_{21}X_1 + a_{22}X_2 + a_{23}X_3)$
$Z_3 = F_h(a_{30} + a_{31}X_1 + a_{32}X_2 + a_{33}X_3)$
$Z_4 = F_h(a_{40} + a_{41}X_1 + a_{42}X_2 + a_{43}X_3)$
$Y = F_o(b_0 + b_1Z_1 + b_2Z_2 + b_3Z_3 + b_4Z_4 + c_1X_1 + c_2X_2 + c_3X_3)$

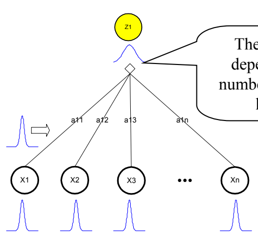
Standard linear regression problem
X – inputs data matrix **(known)**
Z – hidden layer values vector **(known)**
Unique least-squares solutions for bi and ci

Input-to-hidden layer initialization



Hidden nodes have to be within the active region of the nonlinear function

The width of the active zone is defined by the steepness of the function or the"temperature"



The "temperature" depends also on the number of inputs to the hidden node

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

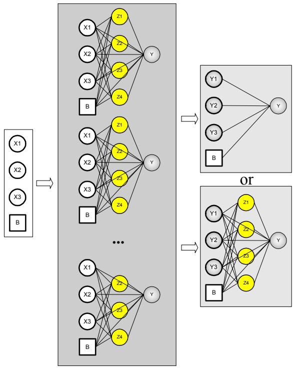
$$Tn = \eta.\frac{\log(2+\sqrt{3})}{\sqrt{ni - 0.5}}$$

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)

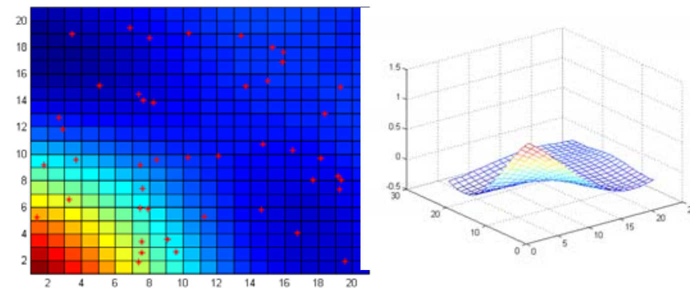
Stacked Analytic Neural Nets (SANN)



- 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

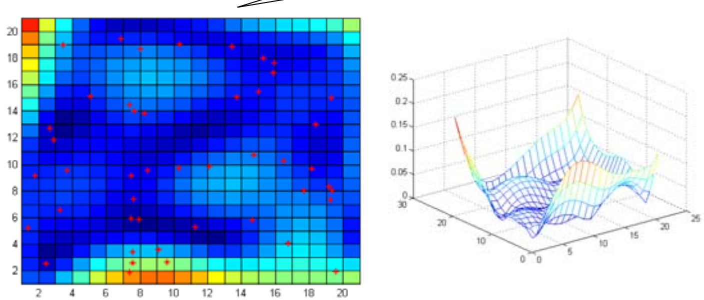
Internally developed in Dow Chemical

Model Mismatch Indicator - 2D



Model Mismatch Indicator - 2D

Models tend to agree where there is data points and tend to disagree where there is no data.



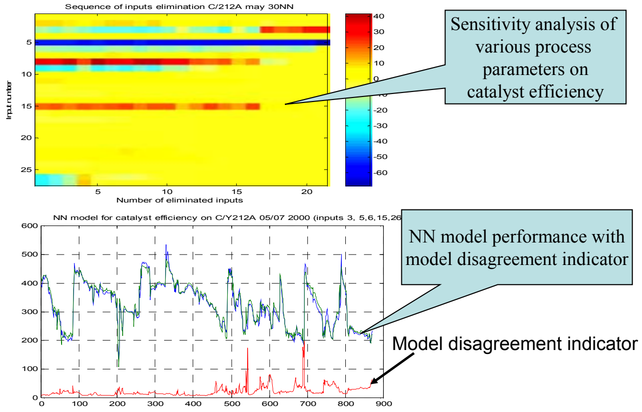
Reduction of the number of input dimensions using Neural Networks

$$SI_j = \frac{\frac{1}{Np}\sum_{p=1}^{Np}\left|\left[\frac{\partial Y}{\partial X_j}\right]_p\right|}{\sqrt{(\mathbf{X}'\mathbf{X})^{-1}_{jj}}}$$

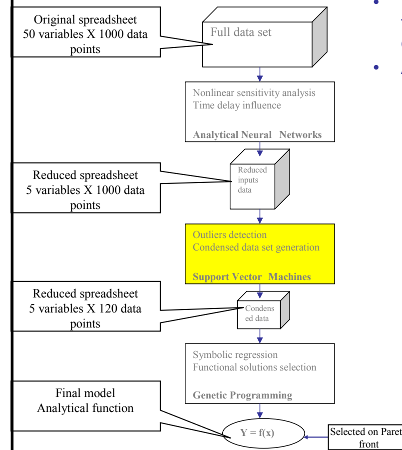
$$\frac{\partial NN_n(\mathbf{X})}{\partial X_i} = w_i^n + \sum_{h=1}^{N_h^n} w_h^n a_h^n (1-a_h^n).t_h^n.w_{ih}^n \quad \text{where} \quad a_h^n = \mathrm{Sig}\left(\sum_{i=0}^{N_i^n} w_{ih}^n X_i^n, \mathrm{t}_\mathrm{h}^\mathrm{n}\right)$$

$$\frac{\partial \mathrm{CM}(\mathbf{X})}{\partial X_i} = \sum_{1}^{N} w_n \frac{\partial NN_n(\mathbf{X})}{\partial X_i}$$

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

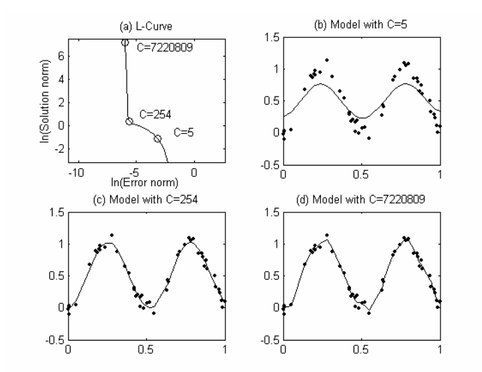


Integrated Methodology for Empirical Models Development

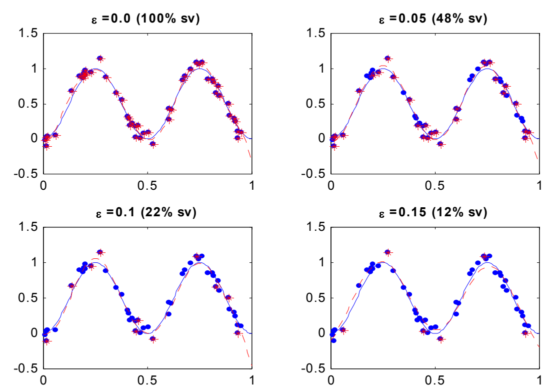


- Hybrid approach integrating multiple technologies exploits the strengths of each
- Advantages:
  - Fast development (days)
  - Robust performance (compact models)
  - Direct implementation in any Distributed Control System (no need for specialized software)
  - Very low capital cost (only if hardware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

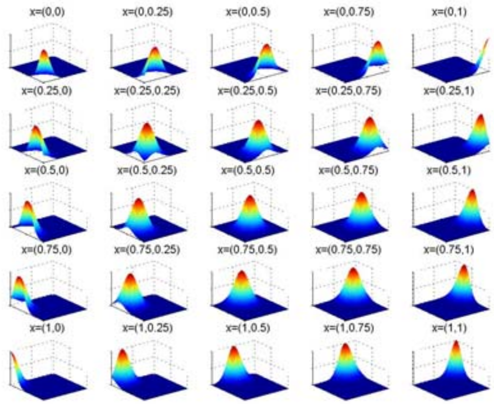
Explicit Complexity Control in Support Vector Machines (SVM)



Controlled Data Compression



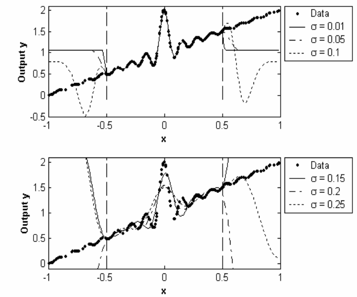
Local Kernel



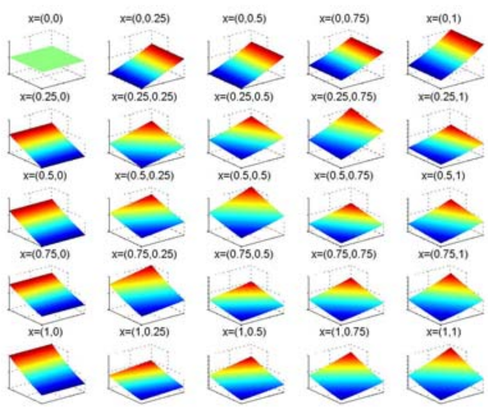
RBF Kernel with σ=0.2

Interpolation/Extrapolation of Local Kernel

- Small widths of kernel interpolate better
- Outside input range, no local information is available and the kernel levels off – no extrapolation
- No single choice of width achieves both



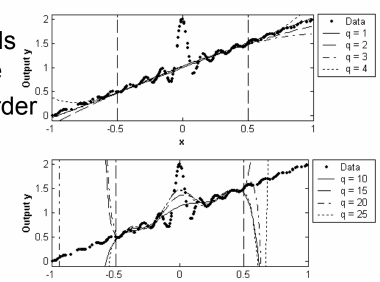
Global Kernel



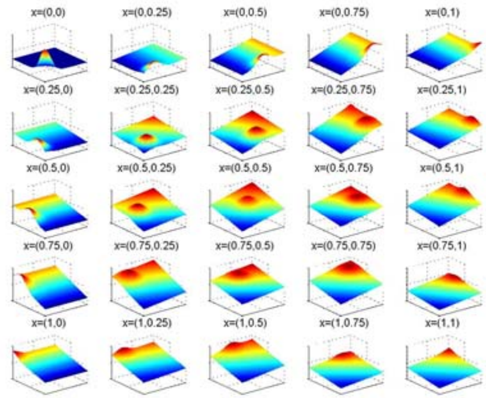
Polynomial Kernel with degree=2

Interpolation/Extrapolation of Global Kernel

- Lower order polynomials extrapolate better
- High order polynomials needed to interpolate
- No single choice of order achieves both



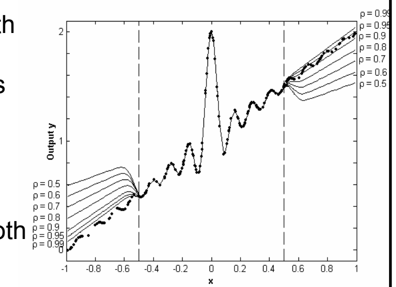
Mix of Local and Global Kernel



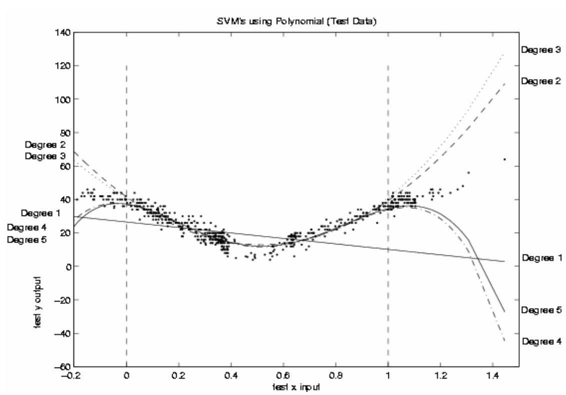
Mixed Kernel with degree=1, σ=0.2, ρ=0.7

Interpolation/Extrapolation with Mixed kernels

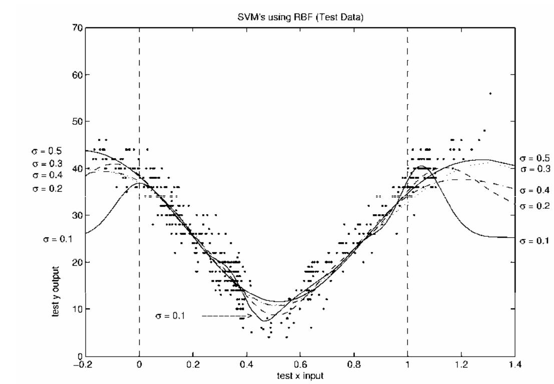
- Mixture of first degree polynomial and RBF with σ=0.01
- RBF contribution makes interpolation possible
- Polynomial makes extrapolation possible
- Single choice of parameters achieves both



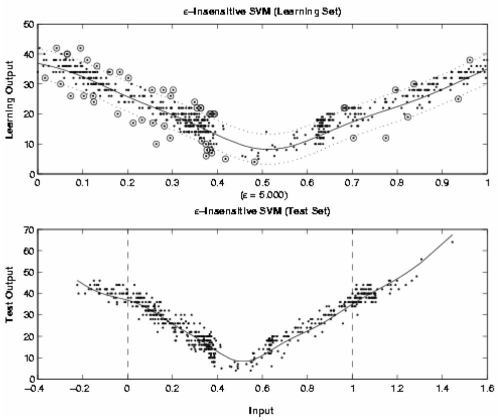
Industrial Example: Polynomial Kernel



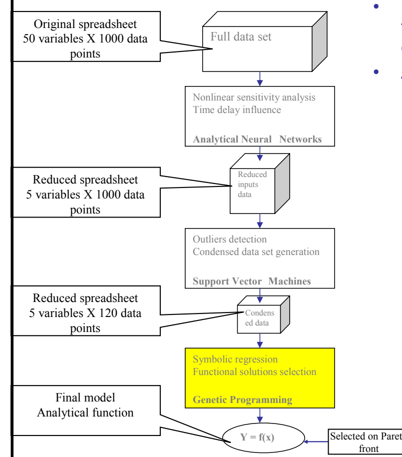
Industrial Example: RBF Kernel



Industrial Example: Mixed Kernel



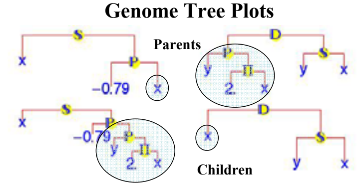
Integrated Methodology for Empirical Models Development



- 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)

Genetic Programming

Genome Tree Plots



Example of Crossover Operation

Phenotypes (Expressions)

Parents

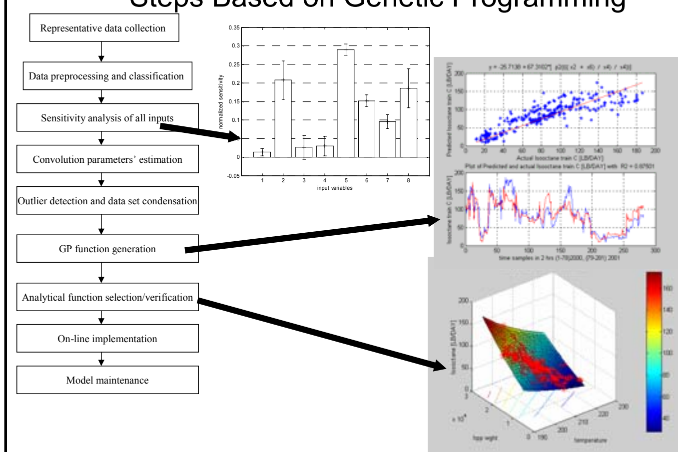
$-(-0.787701)^{x} + x$ $\quad \frac{y^{2x}}{-x+y}$

Children

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

- 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**
- *Can* have better **generalization capabilities** than neural nets
- Low **implementation** requirements
- Issues include …
  - Time delays
  - Sensitivity analysis of large data sets
  - Relatively slow development (hours of computation time)

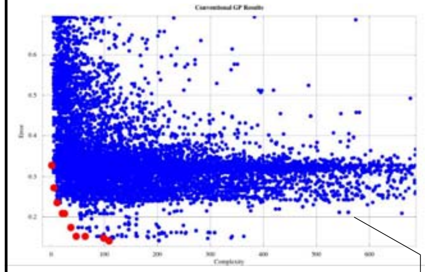
Steps Based on Genetic Programming



Classic 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

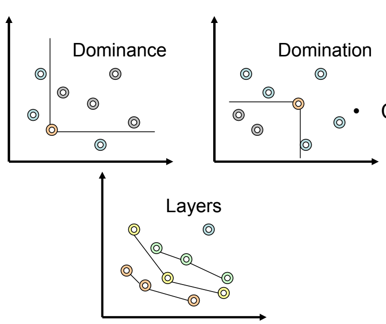
The Pareto Front



- 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?

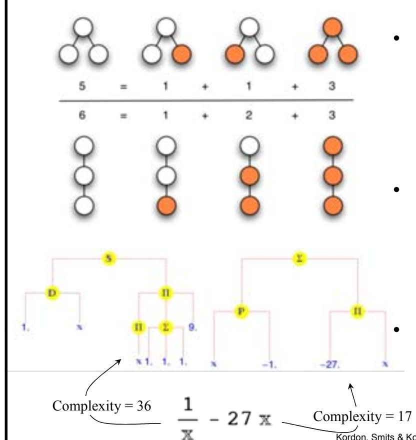
Note that much evolutionary effort is spent exploring high complexity & high fitness regions

Pareto Performance



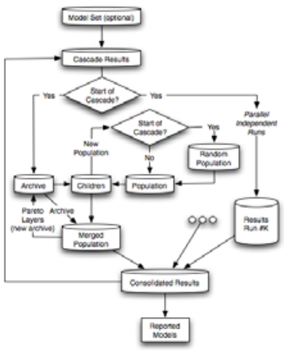
- Characterizing Pareto Performance
  - Dominance
  - Domination
  - Layer
  - Combinations …
- Computational Issues
  - Brute force is $M N^2$
  - Can do $M N \log_{M-1}(N)$ or $M N \log_{M-2}(N)$ if clever
    - M = # of objectives
    - N = population size
  - Computation demands need to be considered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

Genome Complexity



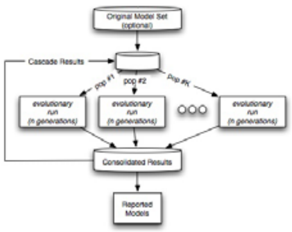
- What is complexity?
  - # of nodes?
  - Tree depth?
  - Included functions?
  - Number of variables?
  - Combinations?
- Chosen function is sum of sum of node counts
  - Provides more resolution at low end of complexity than simply using node count
  - Rewards fewer layers
- Real goal is to characterize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

ParetoGP Algorithm



- Maintain archive based upon Pareto layers
- Each child results from one archive and one population parent
- Cascades …
  - Pareto archive maintained
  - Population wiped out (fresh genes!)
- Independent runs with independent archives for diversity
- This approach is intrinsically Pareto-aware

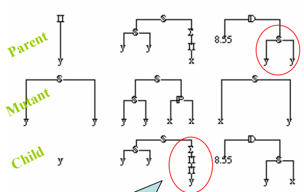
ClassicGP Algorithm



- ClassicGP can be Pareto-aware if a Pareto-aware selection scheme is used
- Most Pareto selection schemes are slow
- Finding the Pareto front can be relatively efficient
- Pareto Elite or Pareto Tourney may be viable selection schemes
  - Pareto tourney: select Pareto fronts from random subpopulations until desired number of models is reached
  - Pareto elite: select randomly from elite (defined using Pareto layers)

Symbolic Regression via GP

```
GenomeTreePlot[{parents,
  MutateSubtree[parents,
   MaximumTreeDepth → 3,
   MaximumArity → 2,
   DataVariables → {x, y}],
  Crossover[parents]}];
```



Introns are either overly complex or non-functional

Nuances…

choice of operators
- functional building blocks

parsimony pressure
- preference for simpler/smaller solutions

diversity operators
- modify fit solutions and the relative presence of each mechanism

fitness-based breeding rights
- proportional, ranking, elitist, tournament, random, etc.

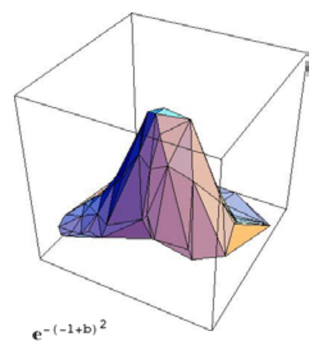
evolution environment
- population size, number of generations, population interaction, fitness criteria, etc.

genetic modifications
- coefficient & structure optimization

automatically defined functions
- dynamically determined building blocks

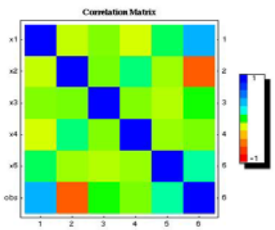
metasensor definitions
- dynamically determined transforms and variable combinations

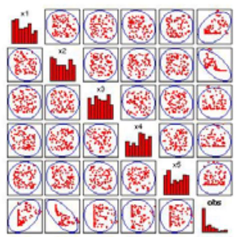
A Toy Problem for Illustration



$$\frac{e^{-(-1+b)^2}}{1.2+(-2.5+a)^2}$$

- We sampled a function of two variables at 100 random points in the range [0,4]
- The data matrix has three random spurious variables in the range [0,4]
- Notice that the entire parameter space is not covered



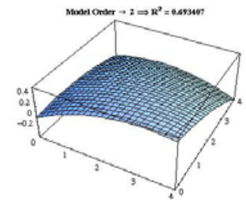


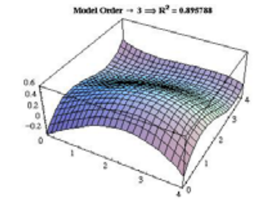
Getting the Zen of the Data

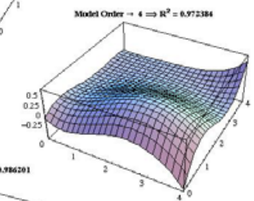
- In this simple example, we could probably guess that only two variables were important for model building
- Correlated inputs can be a problem for some other modeling techniques
- However, lack of correlation to the response does not necessarily correspond to lack of importance

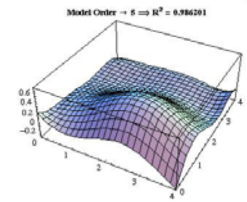
Context-free analysis leads to confidently wrong answers!

Linear Models





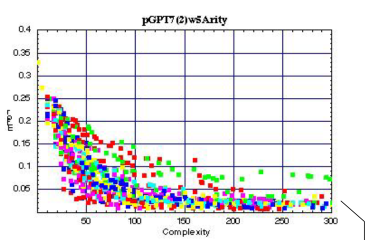




- Here we look at 2nd through 5th order models of the two driving variables (a 3rd order model with all five variables has 56 terms)
- Notice the edges -- these models would likely not extrapolate well!
- However, not much time was required to achieve a poor model!

The Pareto Front: Handling Competing Objectives

No more things should be presumed to exist than are absolutely necessary — W. Occam [1280–1349]



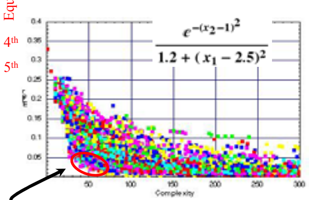
These are the error vs. complexity results of multiple independent symbolic regressions. Note that there is variability from run to run due to the random nature of the evolutionary process.

- Identifies trade-off surface between competing objectives
  - e.g., performance vs. complexity
- Pareto front solutions are the best "bang-for-the-buck"
- Accuracy and simplicity are automatically rewarded
- Pareto Front Benefits
  - Avoids need for *a priori* combination of objectives into a single metric
  - The shape of the front gives us insight into the problem
  - Identifies multiple candidate solutions simultaneously

Evolved Models

	model	complexity	vars	abs corr	R^2	
1	x_2	1	x2	0.672146	0.45178	
2	$x_1 - x_2$	5	x1 x2	0.72785	0.529765	
3	$0.293716^{x_1} x_2^{x_1}$	11	x1 x2	0.805609	0.649005	
4	$0.301214^{0.612354 x_1} x_2^{0.612354 x_1}$	17	x1 x2	0.818587	0.670084	2nd
5	$0.344236^{x_1} x_1^{[illegible]}$	19	x1 x2	0.858864	0.737648	
6	$\left(\frac{x_1}{x_2}\right)^{(5-x_1) x_1}$	25	x1 x2	0.895316	0.801592	3rd
7	$2.23888^{x_1} (-5+x_1) x_1 \left(\frac{1}{x_2}\right)^{x_1}$	27	x1 x2	0.949914	0.902336	Equivalent linear model
8	$2.15727^{x_1} (-4.89307+x_1) x_1 \left(\frac{1}{x_2}\right)^{x_1}$	33	x1 x2	0.950524	0.903495	
9	$1.78744^{[illegible]} (-5+x_1) x_1 \left(\frac{1}{x_2}\right)^{[illegible]}$	35	x1 x2	0.958384	0.9185	
10	$2.23888^{2 x_1} (-5+x_1)^2 x_1^2 \left(\frac{1}{x_2}\right)^{2 x_1}$	40	x1 x2	0.972973	0.946676	
11	$0.415404^{-2 x_1} (5-x_1)^2 x_1^2 x_2^{-2 x_1}$	49	x1 x2	0.976215	0.952995	
12	$1.78744^{[illegible]} (-5+x_1)^2 x_1^2 \left(\frac{1}{x_2}\right)^{[illegible]}$	52	x1 x2	0.980611	0.961599	
13	$(5-x_1)^2 x_1^2 \left(\frac{4 x_1}{x_2}\right)^{x_1}$	54	x1 x2	0.980731	0.961833	
14	[illegible]	62	x1 x2	0.985068	0.970359	4th
15	$(9.56047+(9-2 x_1)^{x_1})^{[illegible]}$	65	x1 x2	0.989806	0.979716	5th
16	[illegible]	72	x1 x2	0.994007	0.988051	
17	$(9+(9-2 x_1)^{x_1}+x_1)^{[illegible]}$	73	x1 x2	0.99426	0.988554	
18	$(8.18505+(9-2 x_1)^{x_1}+x_1)^{[illegible]}$	78	x1 x2	0.994281	0.988596	
19	$(9+(9-2 x_1)^{x_1}+2 x_1)^{[illegible]}$	83	x1 x2	0.994852	0.98973	
20	$(9+(9-2 x_1)^{x_1}+2 x_1-x_2)^{[illegible]}$	95	x1 x2	0.995965	0.991947	
21	$(8.18505+(9-2 x_1)^{x_1}+2 x_1-x_2)^{[illegible]}$	100	x1 x2	0.99604	0.992095	

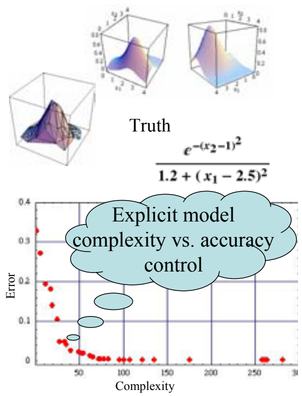
- A run tends to fully explore a foundation structure
- Independent evolutions will result in different (but still fit) structures
- Cascading results from independent evolutions seems to be beneficial
- Note that we are not strictly restricted to the Pareto front in selecting models -- many models may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

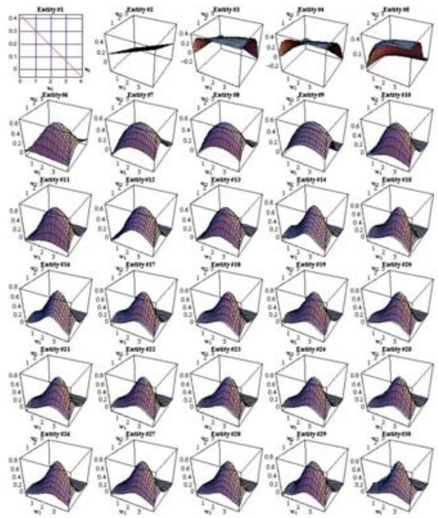


$$\frac{e^{-(x_2-1)^2}}{1.2+(x_1-2.5)^2}$$

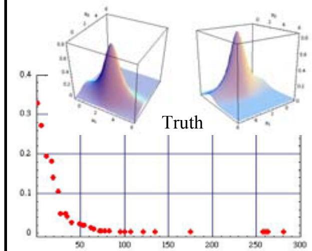
similar performance but diverse structure

Pareto Front Models

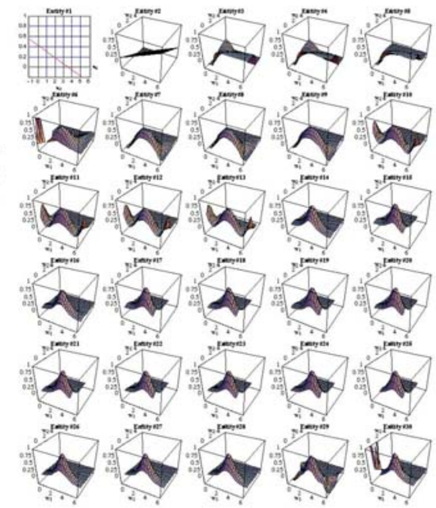




Parsimony & Extrapolation



- Note the pathologies at high complexity when extrapolating
- In general, we want to avoid over-modeling!



Symbolic Regression: Summary Benefits

Compact Nonlinear Models
- Compact empirical models can be suitable for **online implementation**
- Model(s) can be used as an **emulator** for coarse system optimization

Driving Variable Selection & Identification
- Appropriate models may be developed from **poorly structured data sets** (too many variables & not enough measurements)
- Identified driving variables may be used as **inputs into other modeling tools**

Metasensor (Variable Transform) Identification
- Identifying **variable couplings** can give insight into underlying physical mechanisms
- Identified metavariables can enable **linearizing transforms** to meld symbolic regression and more traditional statistical analysis
- Metavariables can also be used as **inputs into other modeling tools**

Diverse Model Ensembles
- The independent evolutions will produce **independent models**. Independent (but comparable) models may be stacked into ensembles whose divergence in prediction may be an indicator of extrapolation & model **trustworthiness**. This is an issue in high dimensional parameter spaces.

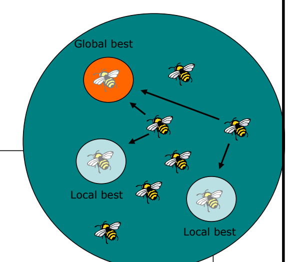
Human Insight
- The **transparency** of the evolved models as well as the explicit identification of the model **complexity-accuracy trade-off** is very compelling
- Examining an expression can be viewed as a **visualization** technique for high-dimensional data

Rapid Modeling
- Exploitation of the Pareto front has resulted in several orders-of-magnitude in the symbolic regression **performance** relative to more traditional GP. This greatly increases the range of possible applications.

There are many benefits to symbolic regression. These are enhanced when coupled with other analysis tools and techniques.

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)

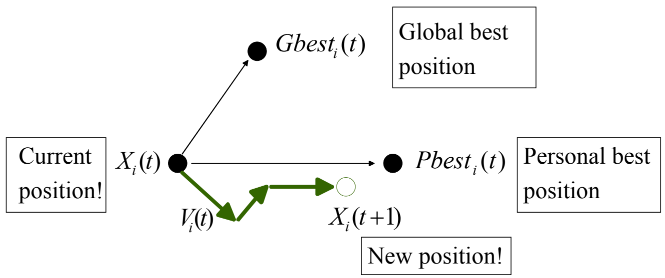
$$V_i(t+1) = \chi \cdot (V_i(t) + c_1 \cdot rand(0,1) \cdot (P_i(t) - X_i(t)) + c_2 \cdot rand(0,1) \cdot (P_g(t) - X_i(t)))$$

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

Note: - stochastic component

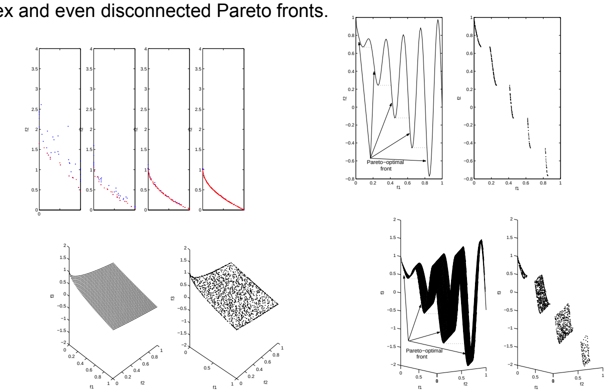
- parameters c_1, c_2, χ default values (2.05, 2.05, 0.73)

Particle's Movement – A Compromise

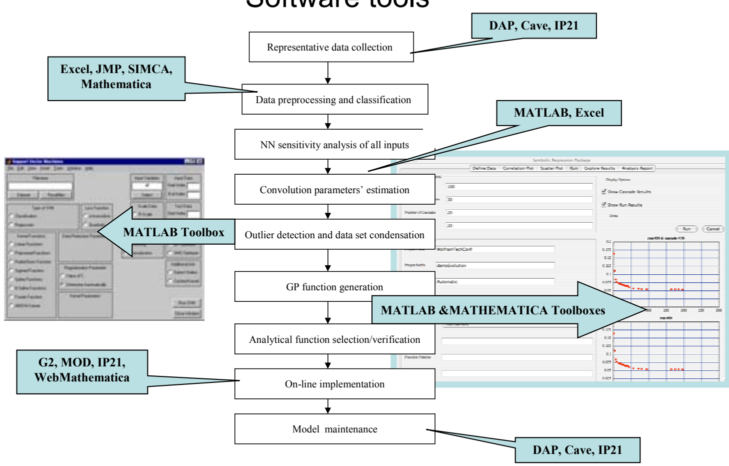


Multi-Objective PSO

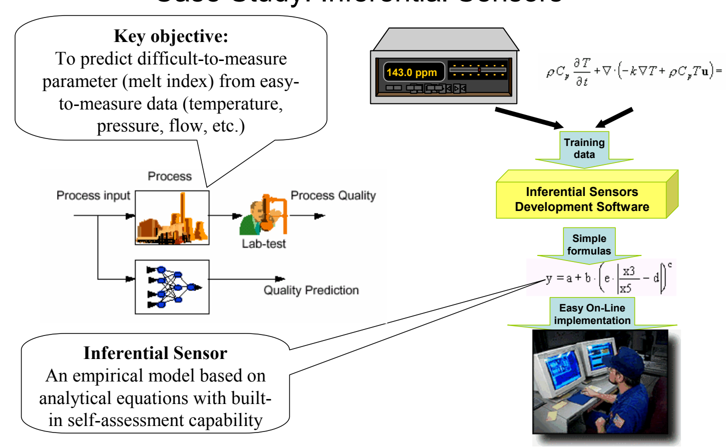
Efficient technique to determine the Pareto front for problems with convex, non-convex and even disconnected Pareto fronts.



Software tools



Case Study: Inferential Sensors

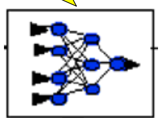


Issues with neural net-based inferential sensors

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

Black box



VS.

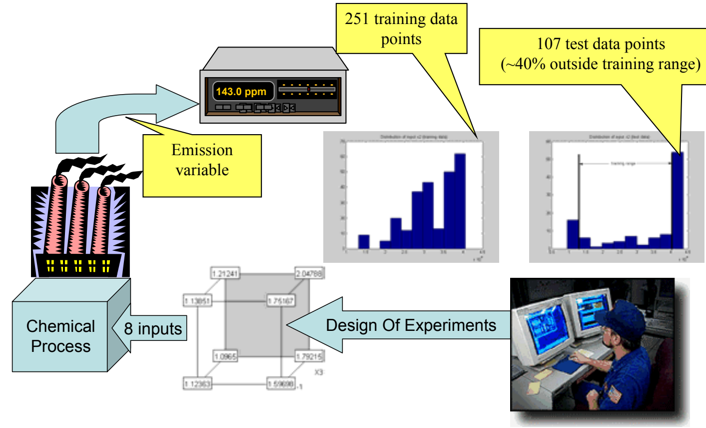
Analytical expression

$$Func2 = \frac{rate^2\left(vac + \frac{rate * hopp\ wt}{temp}\right) pllt\ wt\ temp}{density * temp^2}$$

Specialized run-time software

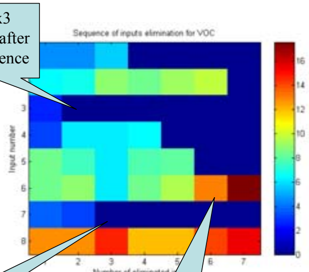
Directly coded into most on-line systems

Inferential sensor for emission monitoring: A case study Data Collection



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

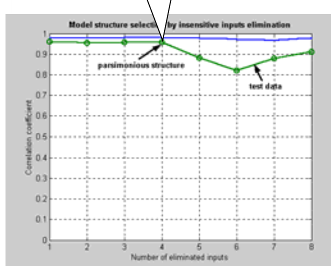
Input x3 removed after first sequence



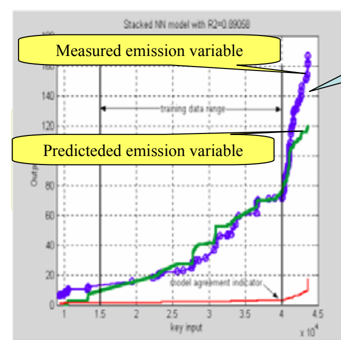
Input x7 removed after second sequence

Input x6 has the strongest sensitivity

A NN with 4 inputs: x2, x5, x6, and x8 is selected after discussion with process engineers



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



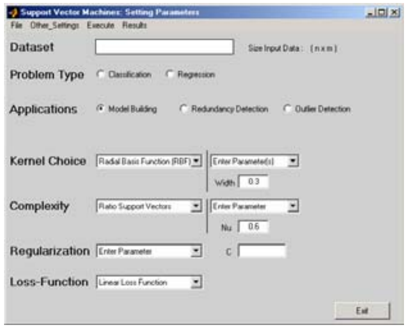
Bad extrapolation (test data is 40% outside the range of training data)

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

Reduced number of inputs from 8 to 4

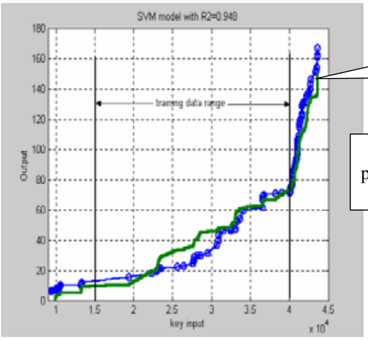
Fast test of the hypothesis about potential nonlinear relationship (in 20-30 min)

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

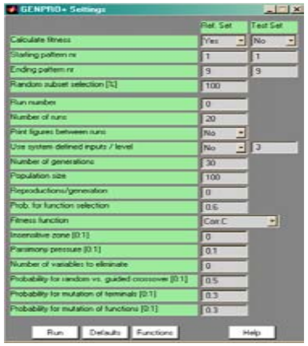


Parameters:
% support vectors: 10
C = 10^6
Mixed Kernels: Polynomial and RBF
Range of Polynomial kernels: 1-3
Range of RBF kernel: 0.25-0.75
Range of ratio 0.5 – 0.99

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



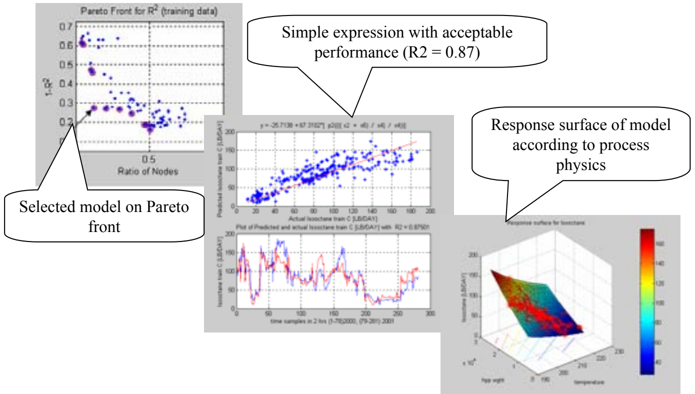
Inferential sensor for emission monitoring: A case study (GP parameters)



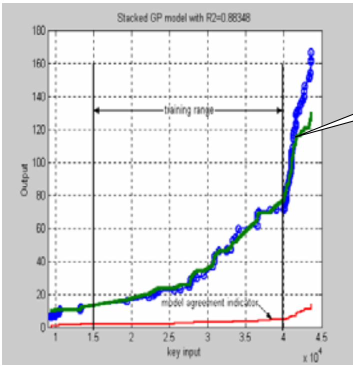
Parameters for a GP simulated evolution

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)



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

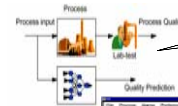


In operation since August 2001

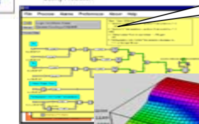
Model based on 8 Stacked Symbolic Predictors

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

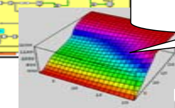
Key application areas



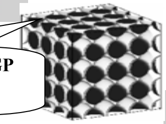
Robust Inferential Sensors
Mass-scale on-line empirical models



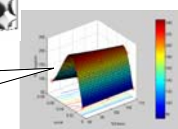
Automated Operating Discipline
Consistent intelligent on-line supervision



Empirical Emulators of Fundamental Models
Effective on-line process optimization



Fundamental model building based on GP
Accelerated new product development

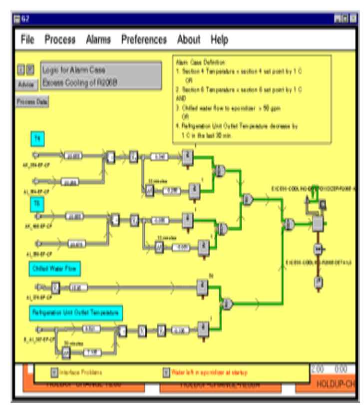


Nonlinear DOE based on GP
Minimizing expensive process experiments

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

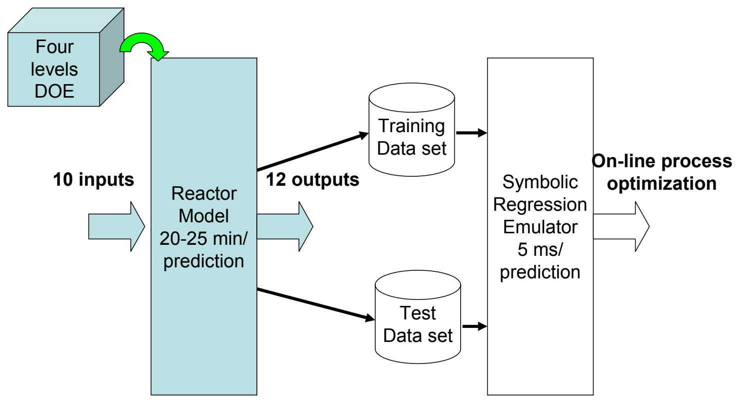
Automating Operating Discipline



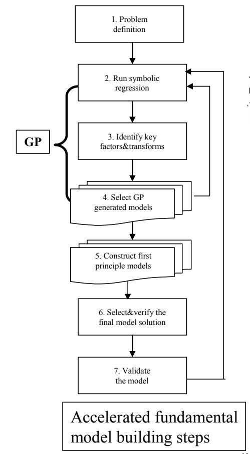
- Heuristic rules defined verbally by process engineers/operators
- holdup predictor designed by stacked analytic NN and GP
- all decision blocks have fuzzy thresholds defined by membership functions
- simple empirical models and mass balances
- fundamental model predictions are used in the heuristic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

- reduced major shutdowns
- reduced lab sampling

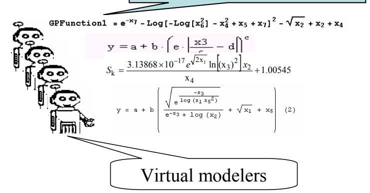
Emulator for optimization of an industrial chemical process



Fundamental Model Building Based on GP



Run simulated evolution before beginning fundamental modeling



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

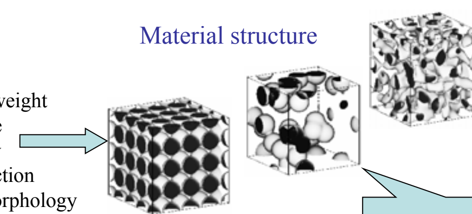
GP as automated invention machine

Eliminate the expert

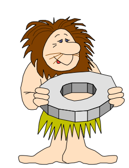
The problem of structure-properties in fundamental modeling

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

Material structure



Key modeling effort for new product development



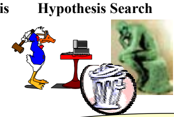
Modeling issues:
- nonlinear interaction
- large number of preliminary expensive experiments required
- large number of possible mechanisms
- slow fundamental model building
- insufficient data for training neural nets

Case Study with Structure-Property Relationships

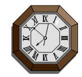
Theoretical Analysis — **Hypothesis Search** — **Fundamental model**

Fundamental Model Building

$$\frac{dT}{dt} = a\frac{\partial^2 T}{\partial z^2} - \frac{DH}{C_p r}\frac{dc}{dt}$$



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



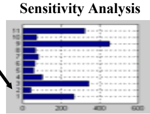
3 months

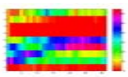
Structure-property data sets

Fundamental Model Building + Symbolic Regression = Accelerated New Product Development

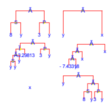
Symbolic Regression

Sensitivity Analysis



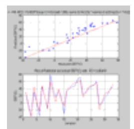


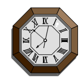
Simulated Evolution



Symbolic Regression Model

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





10 hours

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

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

Classical approach if LOF add experiments to fit second order model

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

More costly experiments

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

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

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

Suggested approach: Use GP to transform inputs

1. Generate GP models

$$S_k = \frac{3.13868\times10^{-17} e^{\sqrt{2x_1}}\ln\left[(x_3)^2\right] x_2}{x_4} + 1.00545 \quad (2)$$

Selected solution

2. Generate input transforms

Variable transformations suggested by GP model

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

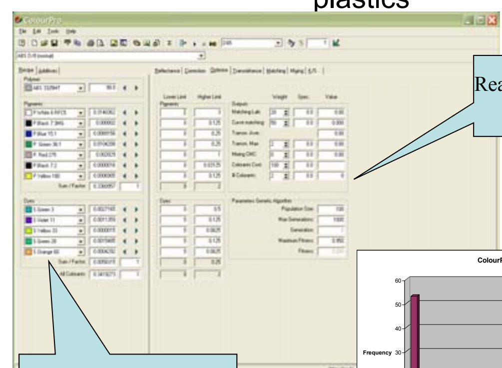
3. Fit response surface model in transformed variables

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

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

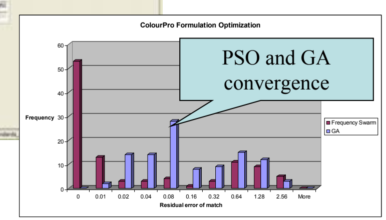
No Lack Of Fit (p=0.3037)

PSO application: Optimizing color spectrum of plastics



Real-time optimization in 2-3 seconds

Multiple-objective PSO with 15 variables

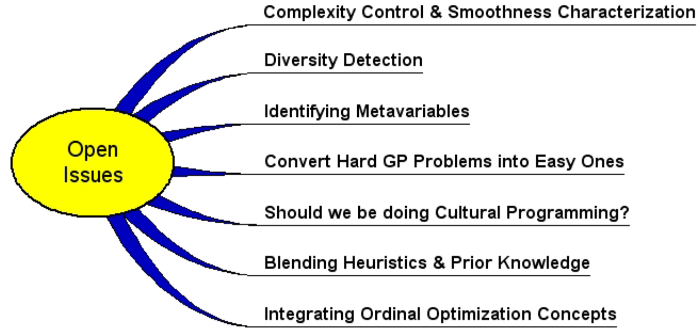


PSO and GA convergence

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 & Current Research



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

Katya Vladislavleva – Tilburg University

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*, Proceedings of GECCO 2004, Seattle, WA, pp. 1078-1089, 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.
10. Kordon A. and CT Lue, *Symbolic Regression Modeling of Blown Film Process Effects*, Proceedings of CEC 2004, Portland, OR, pp. 561-568, 2004.
11. Jordaan, E., A. Kordon, G. Smits, and L. Chiang, *Robust Inferential Sensors based on Ensemble of predictors generated by Genetic Programming*, Proceedings of PPSN 2004, Birmingham, UK, pp. 522-531, 2004

References

12. Kordon A., E. Jordaan, F. Castillo, A. Kalos, G. Smits, and M. Kotanchek, *Competitive Advantages of Evolutionary Computation for Industrial Applications*, Proceedings of CEC 2005, Edinburgh, UK, pp. 166-173, 2005
13. F. Castillo, A. Kordon, J. Sweeney, and W. Zirk, *Using Genetic Programming in Industrial Statistical Model Building*, In: O'Raily U. , Yu T., Riolo, R. and Worzel, B. (eds): *Genetic Programming Theory and Practice II*. Springer, NY, New York, pp. 31 – 48, 2004.
14. A. Kordon, F. Castillo, G. Smits, and M. Kotanchek, *Application Issues of Genetic Programming in Industry*, In: Yu T., Riolo, R. and Worzel, B. (eds): *Genetic Programming Theory and Practice III*. Springer, NY, New York, pp. 241 - 258, 2006.
15. G. Smits, A. Kordon, E. Jordaan . C. Vladislavleva, and M. Kotanchek, *Variable Selection in Industrial Data Sets Using Pareto Genetic Programming*, In: Yu T., Riolo, R. and Worzel, B. (eds): *Genetic Programming Theory and Practice III*. Springer, NY, New York, pp. 79 - 92, 2006