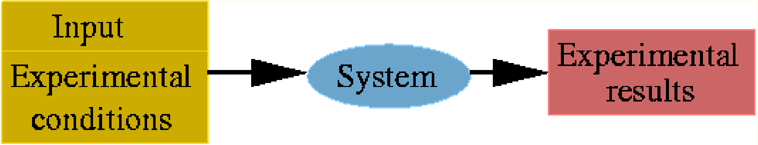
Représentations et connaissances du domaine pour l'identification évolutionnaire

Marc Schoenauer*
Projet Fractales – INRIA Rocquencourt – France
http://www-rocq.inria.fr/fractales/

9 avril 2003

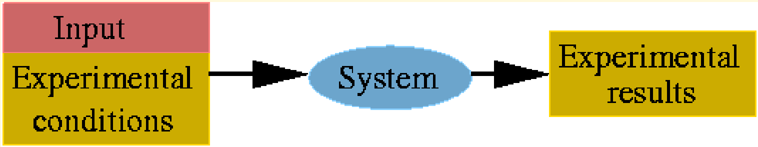
* Au CMAP, École Polytechnique (UMR CNRS 7641) avant sept. 2001

Identification de systèmes



Problème direct:
Phénomène physique - ou sa simulation numérique

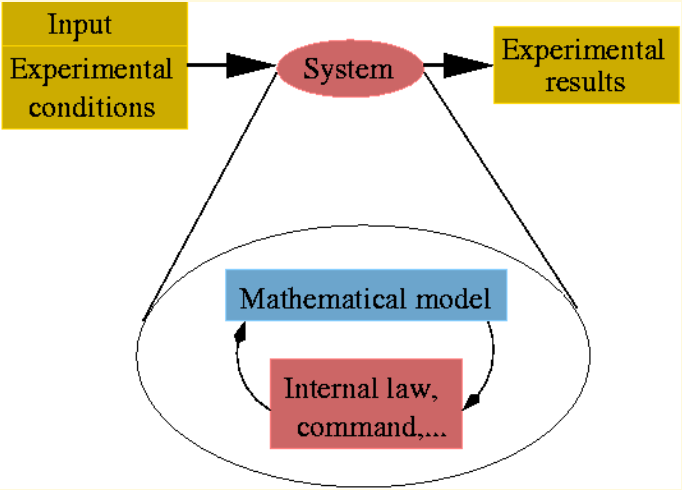
Précise et robuste



Problème inverse:
Touver l'entrée qui va donner la sortie désirée

Identification de systèmes (2)

Reconstruire le système à partir de couples entrée/sortie



- Approche boîte noire : régression — Data fitting
- Modèle mathématique interne : identification de fonction

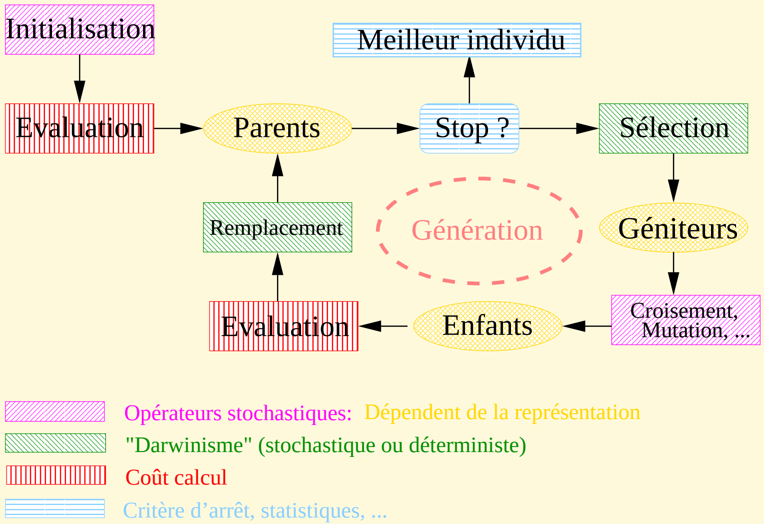
Problèmes souvent mal posés ⇒ optimisation évolutionnaire

Survol

- **Contexte**
  - Algorithmes évolutionnaires
  - Choix de la **représentation** — Programmation génétique
  - Connaissances du domaine — oú ?
- **Identification de lois** — Mécanique des structures
  - Modèles rhéologiques — dans la représentation par arbres
  - Loi d'indentation — Analyse dimensionelle
- **Optimisation de "formes"** — Géophysique, Structures

  Représentation adaptative — dans la fonction performance
- **Conclusions et Perpectives**

Un algorithme “évolutionnaire”



- Paradigme Darwinien grossier
- Optimisation stochastique d’ordre 0

La représentation

- Le Darwinisme ne dépend que de la **performance**
- L'initialisation et les opérateurs de variation ne dépendent que de la **représentation**.

Trois exemples de base:

- Représentation "binaire" $\Omega = \{0,1\}^N$
- Représentation "réelle" $\Omega = [0,1]^N$ or $\mathbb{R}^N \ldots$
- Représentation par arbres GP

Le choix de la représentation est **crucial**

Espaces d'arbres

J. Koza

$$\Omega = \text{Arbres}(\mathcal{N}, \mathcal{T})$$

$\mathcal{N}$ ensemble de noeuds (ou opérateurs)
$\mathcal{T}$ ensemble de feuilles (ou opérandes)

Exemples :

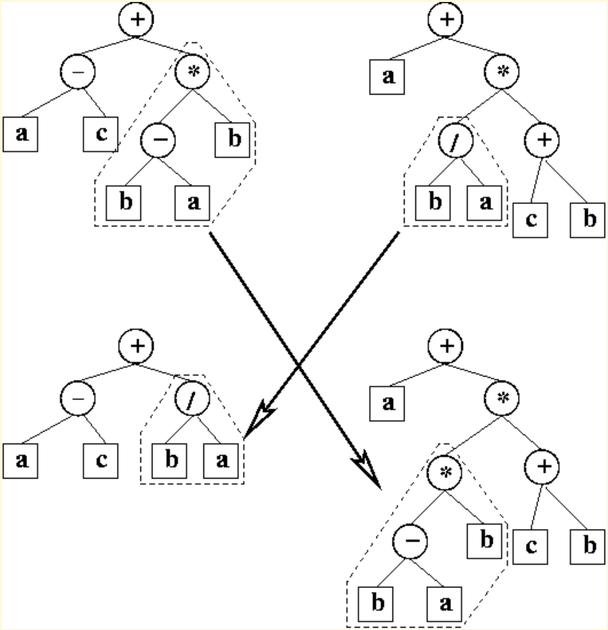
- $\begin{cases} \mathcal{N} = \{+, \times\} \\ \mathcal{T} = \{X, \mathcal{R}\} \\ \Omega = \text{ Polynomes de } X. \end{cases}$

- $\begin{cases} \mathcal{N} = \{ \text{ if-then-else, while-do, repeat-until,..}\} \\ \mathcal{T} = \{\text{expressions, instructions}\} \\ \Omega = \text{ Programmes} \end{cases}$

Croisement des arbres

Principe :

Deux points sont choisis dans les deux parents
Les sous-arbres sous ces points sont échangés.



Croisement des arbres (2)

Idée-Force : fermeture syntaxique

Points délicats :

- Vérifier que la longueur max des arbres est respectée.
- Tout croisement est-il possible ?

Remarque :

1. Opérateur historiquement privilégié de GP.
2. "Le croisement suffit à produire des mutations"...
3. Le croisement tend à produire un enfant long et un enfant court.

Mutation des arbres

Mutation traditionnelle

- Remplacement de sous-arbre

Choisir un point dans le parent
Remplacer le sous arbre par un arbre aléatoire

- Remplacement de noeud

Choisir un noeud/feuille dans le parent
Remplacer ce noeud/feuille par un noeud/feuille de même arité

Mutation “promotionnelle”

- Insertion de noeud

Choisir un point dans le parent
Faire du fils le petit fils

- Promotion de noeud

Choisir un noeud/feuille dans le parent
Remplacer le noeud pere par le noeud fils

Mutation des arbres (2)

Mutation: Terminaux numériques

Mutation “numérique”

- Muter toutes les constantes dans l’arbre. très destructeur
- Mise à niveau : tirer n jeux de constantes, garder le meilleur. coupler optim. non paramétrique/paramétrique
- Optimisation de type montée, ou . . . évolutinnaire

Points délicats :

- Mutation structurelles: Remet en cause la structure de l’arbre $\Rightarrow$ Ajustement des constantes nécessaire
- Très peu de “petites” modifications possibles

Espaces d'arbres
Applications

- "Toutes" fonctions analytiques — ensemble de primitives non limité
- Expressions booléennes — Multiplexeur, classifieur, . . .
- Contrôleurs — robots, usines, . . .
- Règles de développement — Embryogénèse artificielle

Pros : Richesse en fonction des primitives
Fermeture du croisement

Cons : aucun argument théorique !

Représentation :

Paramétrique ou non-paramétrique ?

- Représentations paramétriques : souvent utilisées pour de mauvaises raisons !

Limitations des méthodes d'optimisation traditionelles

- AEs : savent manipuler des représentations quelconques

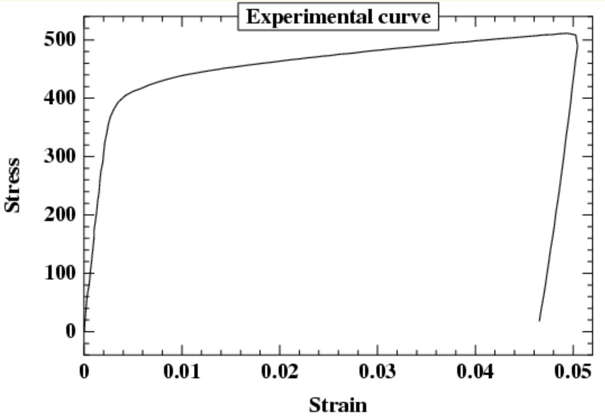
avec des opérateurs "sémantiquement" corrects

Revenir au problème initial sans *a priori*

I – 1D Elasto-visco plastic materials

Input: Experimental curves

- observed strain $\epsilon(t)$ for applied stress $\sigma(t)$;
- observed stress $\sigma(t)$ for applied strain $\epsilon(t)$;



Output: Behavioral law

Differential equations linking $\epsilon(t)$, $\sigma(t)$ and their derivatives, e.g.

$$\begin{array}{ll} \text{if} & \sigma(t) < \sigma_1 \text{ then } \sigma(t) = a.\epsilon(t) + b.\dot{\epsilon}(t) \\ \text{else if} & \sigma(t) < \sigma_2 \text{ then } \sigma(t) = c.\epsilon(t) + d.\dot{\epsilon}(t) \end{array}$$

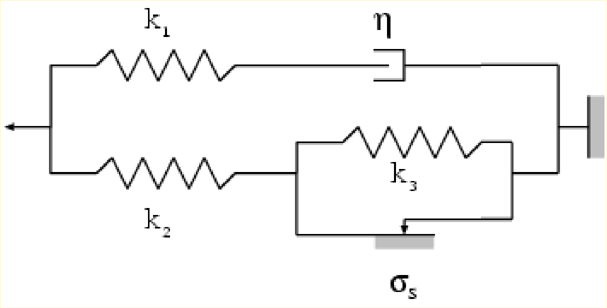
Criteria: the law must fit the experiments **and** be comprehensible.

Search space: Rheological models

Dynamic 1-D laws.

Assembly in series or parallel of

- springs (elastic behavior)
- sliders (plastic behavior)
- dashpots (viscous behavior)



Search space: Rheological models (2)

Identification Goals:

- For a given model, adjust the parameters
  $\Longrightarrow$ **Parametric optimization**
- Optimize both the model and the parameters
  $\Longrightarrow$ **Non-parametric optimization**

Simulation of a Rheological model

Elementary equations:

- Spring(k) $\sigma(t) = k \, . \, \epsilon(t)$
- Slider(η) $\sigma(t) = \eta \, . \, \dot{\epsilon}(t)$
- Dashpot(σ_S) $(\dot{\epsilon}(t) = 0) \; OR \; (|\sigma(t)| = \sigma_S)$

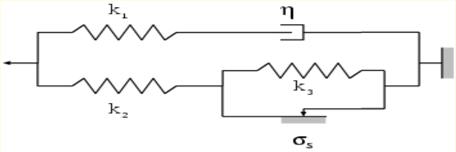
Connection equations:

- Series
  $\epsilon_{parent}(t) = \epsilon_{child_1}(t) + .. + \epsilon_{child_m}(t)$ $\sigma_{parent}(t) = \sigma_{child_1}(t) = .. = \sigma_{child_m}(t)$
- Parallel $\epsilon_{parent}(t) = \epsilon_{child_1}(t) = .. = \epsilon_{child_n}(t)$ $\sigma_{parent}(t) = \sigma_{child_1}(t) + .. + \sigma_{child_n}(t)$

Solve $\rightarrow \sigma_{sim}(t) = Fn(\epsilon_{exp}(t), \dot{\epsilon}_{exp}(t))$

Parametric identification

A given model (e.g. that of the polyethylene)



$$\sigma_{sim}(t) = \mathcal{F}(\epsilon(t), \dot{\epsilon}(t); k_1, k_2, k_3, \eta, \sigma_S)$$

The unknown are $k_1, k_2, k_3, \eta, \sigma_S$, and the goal is to minimize $\sum_i |\sigma_{exp}(t_i) - \sigma_{sim}(t_i)|^2$

Measures are available at discrete times t_i only ($\approx$ 30-300 values).

Ill-posed optimization problem

Methodology

- Write the program computing σ_{sim} (by finite differences approximation of the equations). The parameters of that program are $k_1, k_2, k_3, \eta, \sigma_S$.
- Use trial-and-errors, or iterated hill-climbing to adjust the parameters.

Evolutionary Algorithms are a better choice!

Standard $(10 + 30) - ES$ was used.

Evaluation

Compilation

$$H \quad \longrightarrow \quad \textit{Système d'équations } \mathcal{S}_H$$

- Ressort(k) $\qquad \sigma(t) = k \,.\, \varepsilon(t)$
- Amortisseur(η) $\qquad \sigma(t) = \eta \,.\, \dot{\varepsilon}(t)$
- Patin(σ_S) $\qquad (\dot{\varepsilon}(t) = 0)\ OR\ (|\sigma(t)| = \sigma_S)$
- Série $\qquad \varepsilon_{parent}(t) = \varepsilon_{fils_1}(t) + \varepsilon_{fils_2}(t)$

  $\sigma_{parent}(t) = \sigma_{fils_1}(t) = \sigma_{fils_2}(t)$
- Parallèle $\qquad \varepsilon_{parent}(t) = \varepsilon_{fils_1}(t) = \varepsilon_{fils_2}(t)$

  $\sigma_{parent}(t) = \sigma_{fils_1}(t) + \sigma_{fils_2}(t)$

Simulation

$$\mathcal{S}_H \cup \ (\varepsilon_H(t) = \varepsilon_{exp}(t)) \ \longrightarrow \ \sigma_H(t)$$

Evaluation

$$f(H) = Distance(\sigma_H, \sigma_{exp})$$

Critère d'arrêt

Sources d'erreur

- ED $\longrightarrow$ Différences finies
- Erreurs expérimentales
- Bruit de résolution

Estimation de l'erreur

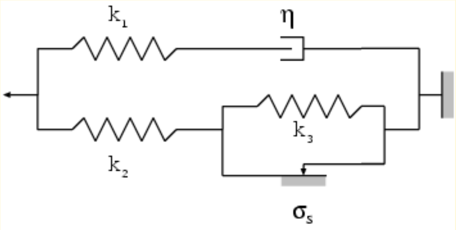
$$Err = ||\sigma_H(t_{exp} = t_1, t_2, t_3, ...) - \sigma_H(t_{exp} = t_1, t_3, t_5, ...)||$$

Critère de succès

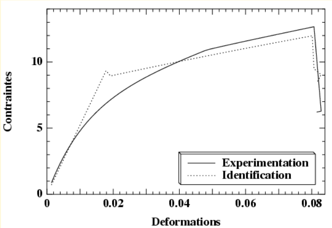
$$f(H) \approx Err$$

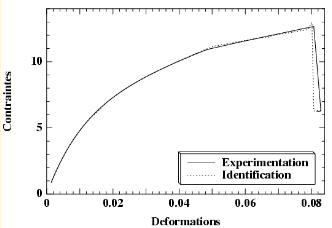
Results

Model for the polyethylene



Responses of the best model in the population





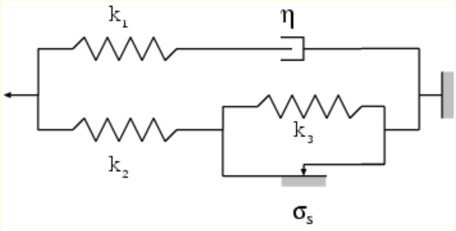
after 20 generations after 200 generations

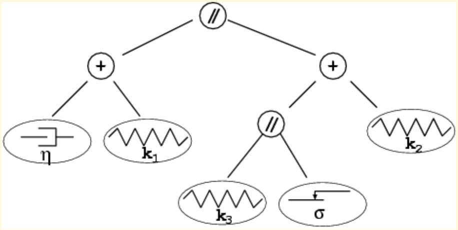
	k_1	η	k_2	k_3	σ_S
Identification	627.2	4748.	98.3	73.8	4.94
"Experimental"	587.9	4914.	93.1	116	4.49

Rheological GP

Rheological models $\equiv$ Trees built from

- $\mathcal{N} = \{$ series $+$, parallel // $\}$
- $\mathcal{T} = \{$ Spring(k), Slider(σ_S), Dashpot(η)$\}$

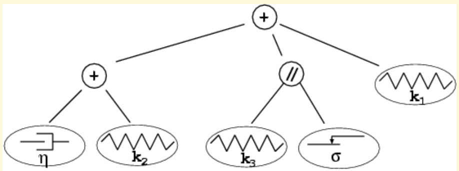


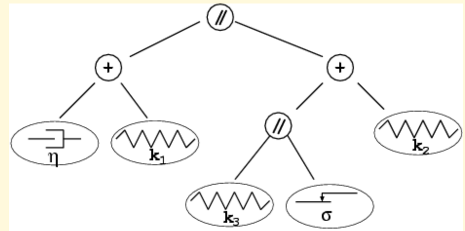


Fitness computation

- Need for an **interpreter** of rheological model
- The sliders raise many difficulties (2 modes depending on σ w.r.t. the threshold.
- Complexity: $T \times 2.(3N)^3/3$, where T is the number of time steps of the loading history and N the size of the model
- Much slower than the compiled program used for parametric identification.

Results

- 20% of successful runs (w.r.t. error criterion)
- 



Repeatedly found (wrong!) structure   Compare to actual one

$\longrightarrow$ due to the absence of *creep* in the experiments.

- Best values of the parameters:

	k_1	η	k_2	k_3	σ_S
"Exp."	790.45	6248.60	150.20	41.60	7.25
Res.	998.89	8698.78	133.08	39.66	19.04

Rheological models identificationl: conclusion

Parametric identification

- gives more accurate results more rapidly
- ... if the guess of the model is good.
- otherwise, the bias can be misleading.

Non-parametric identification

- looks for solution in a much larger space
- ... but can easily get lost
- and may require heavier computational skills.

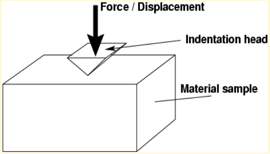
In both cases, the experimental data are crucial:
Use EC to discover discriminant experiments for similar models.

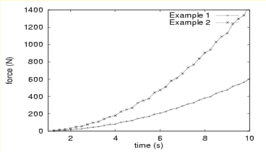
e.g. creep in the polyethylene case above

II – Dimension aware GP

The mechanical problem

Indentation experiments on unknown material





Goal: Find expression $\mathcal{F}$ s.t.

$$\text{Force} = \mathcal{F}(\text{displacement, time, material parameters})$$

Ratle, Sebag – EEAAX and LMS, Ecole Polytechnique, 2000

Representations

- Parametric: Look for scalar A and P s.t.

$$\mathcal{F} := Au^2 e^{Pt}$$

Smallest possible search space – no need for EC

- Non-parametric:
  Need for understandable law $\longrightarrow$ **Genetic Programming**

Largest possible search space, best possible solutions?

- Refinement: dimension aware GP

Assumption:

finite set of units	$\{m, s, kg\}$	length, time, mass
compound units	$U_{ijk} : m^i s^j kg^k$	$-2 \leq i, j, k \leq 2$

GP with prior knowledge on the search space

Historical GP:

Closure hypothesis

All expressions are admissible

- PRO: Simple variation operators
- CON: Huge search space, many irrelevant individuals

Physical identification → Dimensionally consistent laws

$$Oranges \neq Apples + Bananas$$

Constraints on the admissible expressions

syntactic constraints = grammar

Context-Free Grammars: $\{S, N, T, P\}$

S : start symbol
N : set of non-terminal symbols — rewritten using production rules
T : set of terminal symbols
P : set of production rules

Example: Universal grammar

$$
\begin{aligned}
N &= \{<E>,\ <O>,\ <V>\} \\
T &= \{+,\ \times,\ x,\ \mathcal{R}\}
\end{aligned}
$$

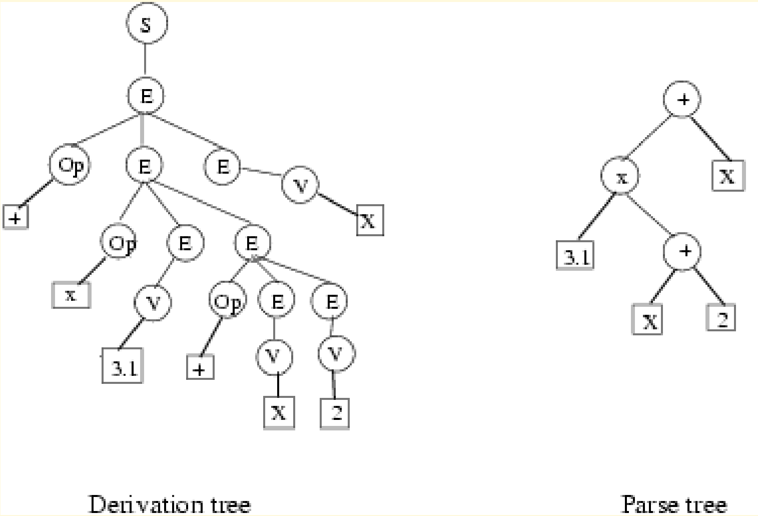
$\mathcal{R}$ = any real-valued constant

$$
P = \left\{ \begin{array}{rcl}
S & := & <E> \ ; \\
<E> & := & <O>\ <E>\ <E> \ \mid \ <V> \ ; \\
<O> & := & + \mid \times \ ; \\
<V> & := & x \mid \mathcal{R} \ ;
\end{array} \right\}
$$

$\longrightarrow$ “Standard” GP trees ($\mathcal{N} = \{+, \times\},\ \mathcal{T} = \{x, \mathcal{R}\}$)

Enforcing constraints through grammars

Derivation trees ⟶ Parse trees



Beware !	**CFG**	**GP**
Terminals	variables, constants, operators	variables and constants
Non-Terminals	typed expressions	operators

GP on derivation trees – Gruau 96

- Initialization: uniform selection among derivations in a production rule

filter out trees with depth $> D_{max}$

- Crossover: swap nodes with same non-terminal symbol

$\equiv$ Strongly Type Genetic Programming

Montana 1995, Haynes et al. 1996

- Mutation: select another derivation

Dimension grammar

Physical units			
Quantity	mass	length	time
Variables			
K (Elastic element)	+1	0	–1
n (Viscous element)	+1	0	–1
t (time)	0	0	+1
u (displacement)	0	1	0
Solution			
F (Force)	1	1	-2

Automatic generation of the grammar

each compound unit → a non-terminal symbol
admissible combinations → production rules

N	non-terminals	$\{U_{ijk}\}$
T	terminals	$\{Vars, \mathcal{R}, +, -, *, /, exp\}$
P	production rules	

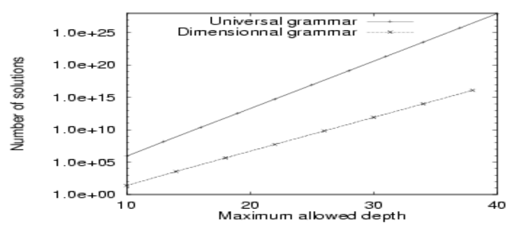
$$U_{ijk} := U_{ijk} + U_{ijk} \mid U_{ijk} - U_{ijk} \mid U_{ijk}\; exp^{\,U_{000}}$$
$$\big|_{abc+def=ijk}\, U_{abc} \;*\; U_{def}$$
$$\big|_{abc-def=ijk}\, U_{abc} \,/\, U_{def}$$
$$\big|_{unit(var)=ijk}\, Var$$

$$\mathcal{F} := \quad mass \times length \times time^{-2}$$

Automatically generated

First Results

Reduction of the search space



Poor performances ...blamed on Initialization

Uniform initialization: $\mathcal{P}$(non-terminal) $\gg$ $\mathcal{P}$ (terminal)

deep trees, most are filtered out

Note : Similar to constrained optimization with sparse feasible region

Ryan et al, 1998

Poor initial population $\longrightarrow$ poor performances

Initialization in Grammar Guided GP

Biased initialization fails

- Set $\mathcal{P}$(terminals) $\gg$ $\mathcal{P}$(non-terminals)
- Population poorly diversified, premature convergence

Constraint resolution for initialization

- Minimal tree depth for each non-terminal or derivation
- On-line filtering out of derivations

  incompatible with maximum depth

- GP initialization = constraint solver

$\longrightarrow$ Diversified initial population within depth D_{Max}

Experimental validation

Parameters

Population size	2000
Max. number of generations	1000
Probability of Crossover	0.8
Probability of tree mutation	0.2
Probability of point mutation	0.8
Number of training examples	20
Number of independent runs	10

Experiment Goal

Compare the effects of Background Knowledge

- No background knowledge (Universal Grammar)

  GP search space

- Little BK: ad hoc operators

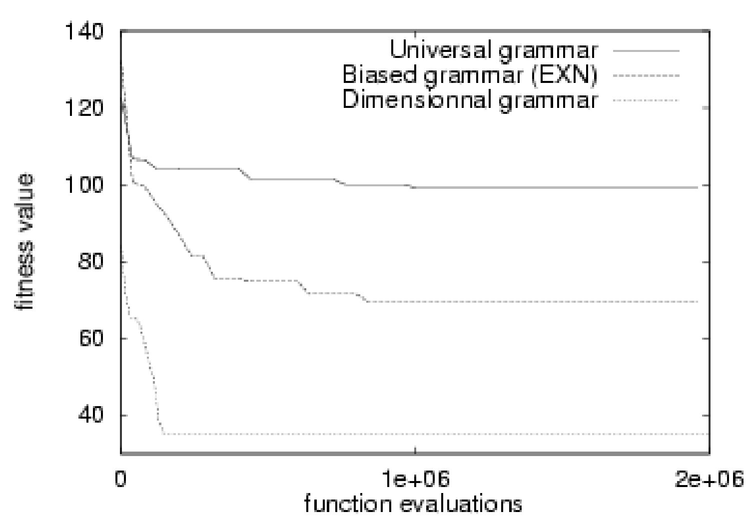
  Exponential-Neg to guarantee stability

- Knowledge on the model shape

  $\mathcal{F} := Au^2 exp(Pt)$, where A and P are GP trees

- On each of the above, dimensional consistency: Grammar-Guided GP

Experimental Results – on-line



Convergence: Impact of grammars

Experimental Results – off-line

Grammar	Average fitness	Std. deviation
universal-untyped	6.2236E+4	0.0E+0
$[A\exp(Pt)]$-untyped	6.5762E+4	2.2E+3
$[Au^2\exp(Pt)]$-untyped	5.1194E+4	1.9E+3
universal-dim	3.1009E+4	5.8E+3
$[A\exp(Pt)$-dim	4.0089E+4	2.7E+3
$[Au^2\exp(Pt)$-dim	3.6357E+4	3.4E+3

Grammar Guided GP: lessons

- If you can bypass EC safely, please do :-)
- "The more the better" not always true
- GP needs help - use background knowledge

But G^{3}P hardly scales up . . . GE, Keijzer & Ryan 2000

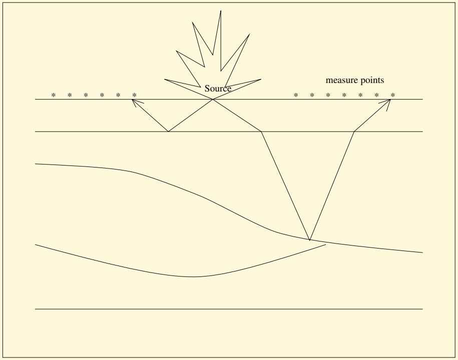
Adaptive complexity

The geophysical problem

- **Question:** Where is the oil?

  Underground layout of velocity?

- **Experiment:**
  A seismic explosion + recordingS of elastic waves at some receptors (*seismograms*).



Hypothesis: Blocky model

Piecewise constant velocity

PhD of F. Mansanne – Dec. 2000 – coll. IFP

Parametric Representation

- Series of layers

  Stoffa and Sen 91 — Unrealistic

- User-defined models of increasing complexity

  Boschetti 95 — A priori information required

- Control points of splines (**not** blocky model)

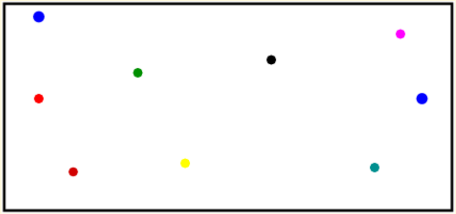
  CMAP-IFP 95 — Uniformly distributed

  Docherty & al. 97 — Cleverly situated

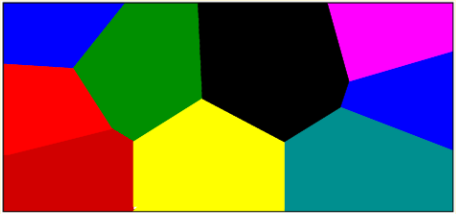
But domain knowledge is not always available

$\longrightarrow$ representations of variable complexity

Voronoi representation



Colored Voronoi sites



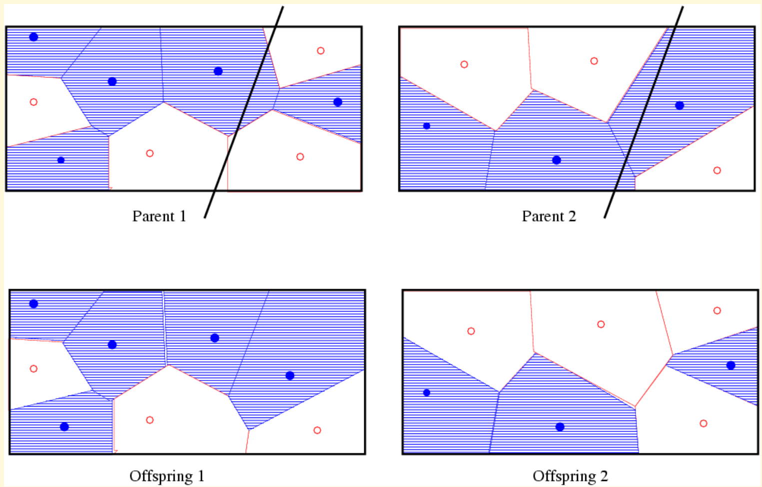
Colored Voronoi cells

Genotype:

Unordered variable length list of labeled Voronoi sites $((S_1, v_1), \ldots, (S_n, v_n))$, $n \in \mathbb{N}$, $v_i \in \{V_{min}, V_{max}\}$

Variation operators
Crossover

Geometrical exchange of Voronoi sites

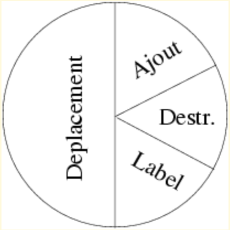


Variation operators
Mutations

- Adaptive Gaussian mutation of sites coordinates
- Adaptive Gaussian mutation of velocity labels
- Site addition - deletion

as smooth as possible

Choice according to user-defined weights



Two fitness functions

"Standard" Least Square fitness

- Solve the wave equation in the whole domain
- Compute LS distance between simulation and experients
- CPU time increases with # shots, mesh size

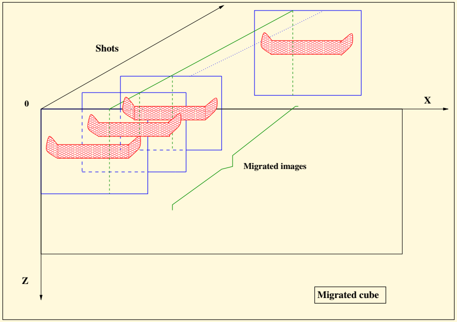
Semblance fitness

- Using geophysic specific image analysis techniques
- CPU time $\propto$ # shots, # receptors, mesh size
- Much faster to compute than the wave equation …

for a single receptor

Image migration

- For each receptor,
- Compute the backward propagation of the recorded wave
- Compare with the direct seismic wave due to the explosion

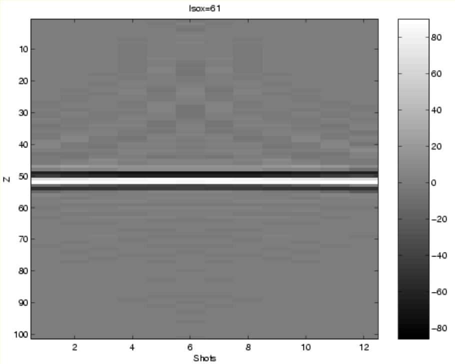


- Diffracting points are the only points where there is time coincidence
- ... if the velocity model used to compute the waves is correct!

The semblance fitness function

Taner & Koehler, 1969

- Measures the horizontal alignment of reflection events at a given X position.
- Sucessful in GA-based (1D) North Sea profile identification (Jin & Madriaga 93).



$$Fitness = \sum_{X \in \xi} \frac{|| \sum_{i=1}^{i=nshots} trace_i(X,Z) ||}{\sum_{i=1}^{i=nshots} ||trace_i^2(X,Z)||}$$

First experiments using Semblance

Geological

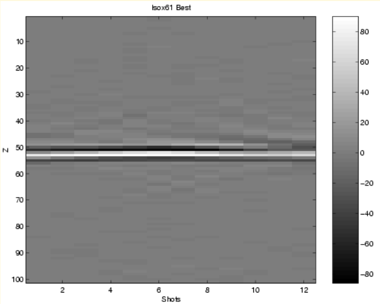
- 3100m $\times$ 1000m domain, 25m discretization step
- 12 shots (every 50m), 101 receptors (every 25m)
- IFP dedicated solver — One direct problem = 70s of Alpha 500
- Artificial test case: — homogeneous underground ($2000ms^{-1}$), one horizontal diffrator at $Z = 500m$

Evolutionary

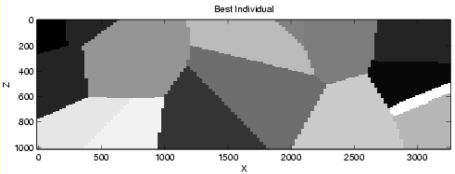
- Pop. size 30, $p_{crois} = 0.7$, $p_{mut} = 0.6$
- Deterministic binary tournament,
  Standard elitist generational replacement
- 2500 evaluations ($\equiv$ 24h)

A bug in the fitness

Robust convergence to very low semblance values ...



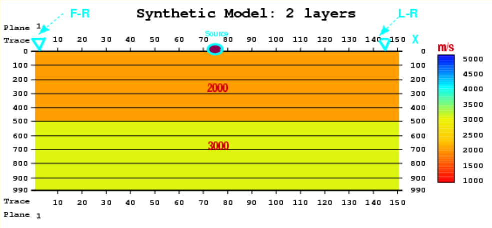
Best Iso-X plot for receptor 61 found by evolution



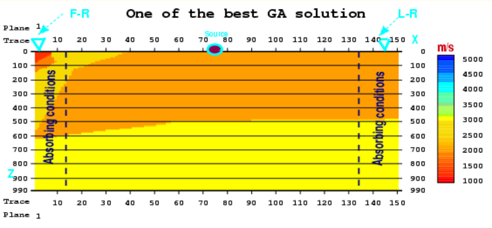
Corresponding absurd velocity layout

LS fitness would avoid such weird solutions

Results using LS fitness

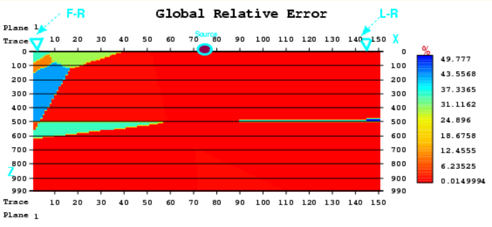


Reference

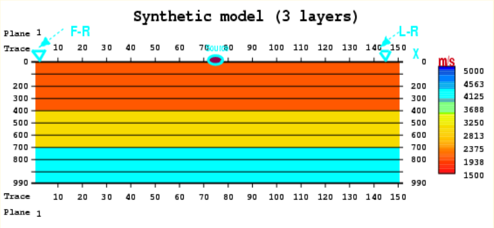


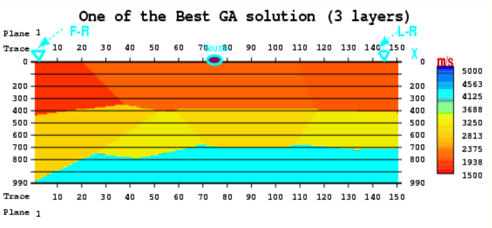
Best results (out of 5 runs)

- 2 homogeneous layers
- 100 × 150 mesh
- 30000 LS evaluations
- 150h on O2 *Silicon*



Relative error

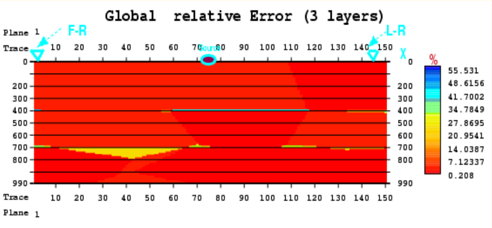




Reference

Best results (out of 5 runs)

- 3 homogeneous layers
- 100×150 mesh
- 30000 LS evaluations
- 150h on O2 *Silicon*



Relative error

But does not work on more complex undergrounds

A mixed fitness

- Use both fitnesses — Semblance and Least Square
- But not at the same time — Swap fitness every N generations

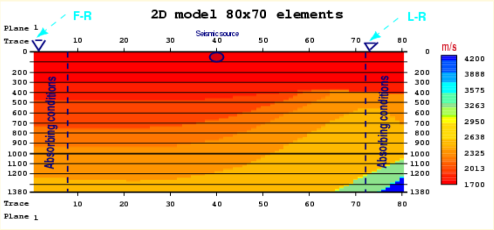
You can also use tune the complexity of the simulation

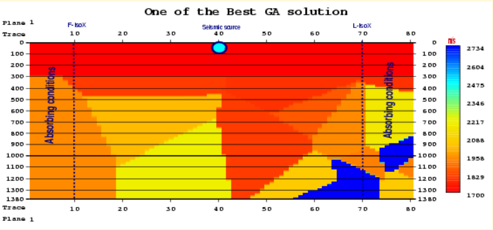
- Number of receptors, of migrates wavelengths — Semblance cost only
- Number of shots — requires as many simulations
- Size of mesh — Not done yet

and eventually gradually increase it along evolution

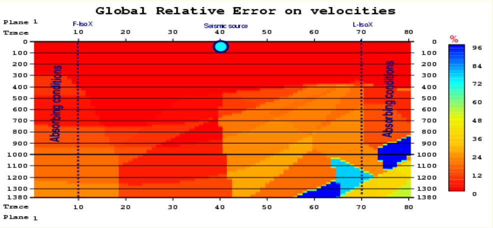
Results using mixed fitness

swap every 5 generations





- Picrocol benchmark
- 80 $\times$ 70 mesh
- 20000 evaluations each
- 200h on O2 *Silicon*



Relative error

Geophysical inversion: lessons

- EAs do not have any common sense
- Beware of . . . exclusive background knowledge
- Use all information
- Go parallel!

Conclusions

- Représentations

Souplesse extraordinaire

Pas de magie

- Connaissance du domaine

Initialisation, opérateurs de variation, fonction objectif

- Hybridation

Techniques d'apprentissage, de recherche locale