Evolutionary Computation
A robut optimisation framework for ML

Michèle Sebag

TAO Group – Equipe Inférence et Apprentissage

Laboratoire de Recherche en Informatique

Université Paris-Sud, CNRS-UMR 8623 – Orsay

Michele.Sebag@lri.fr

IRIT — 9 mars 2005

Learning as search

The data $\{\mathbf{x_i} \in \textit{Input space}, y_i \in \textit{Output space}\}$

The search find hypothesis $\hat{y}$: *Input space* $\mapsto$ *Output space*

Input space	$\mathbb{R}^n$, series, graphs, logical formulas
Output space	$\mathbb{R}$, $\{0, 1\}$
Search space	parametric spaces, others, programs

An optimization problem

$$\text{Minimize } \mathcal{F} = \text{generalization error}$$

but $\mathcal{F}$ is unknown.

Learning / Optimizing

A multi-objective optimization problem

Achieve a good trade-off between:

bias (minimize empirical error)

variance (maximize confidence in the empirical error).

Statistical learning

From multi-objective optimization to

constrained optimization (SVM)

well-posed optimization problem (SRM).

Learning and the search space

- Parametric space
- Finite subspace in infinite space (e.g. SVM)
- The expert space/language (e.g. programs)

In complex real-world situations, optimization becomes approximate optimization
since the description of the real world is
radically simplified until reduced to a degree of complication
that the decision maker can handle.

Satisficing *seeks simplification in a somewhat different direction,*
retaining more of the detail of the real-world situation,
but settling for a satisfactory, rather than approximate best, decision.

Herbert Simon, 1982

Part I Evolutionary Computation

$$\mathcal{F} : \Omega \mapsto \mathbb{R}, \quad \text{Find ArgMax}(\mathcal{F})$$

Aimed at ill-posed problems:

- Chaotic $\mathcal{F}$
- Mixed search spaces
- Infinite search spaces
- Non computable $\mathcal{F}$

"Chaotic" $\mathcal{F}$

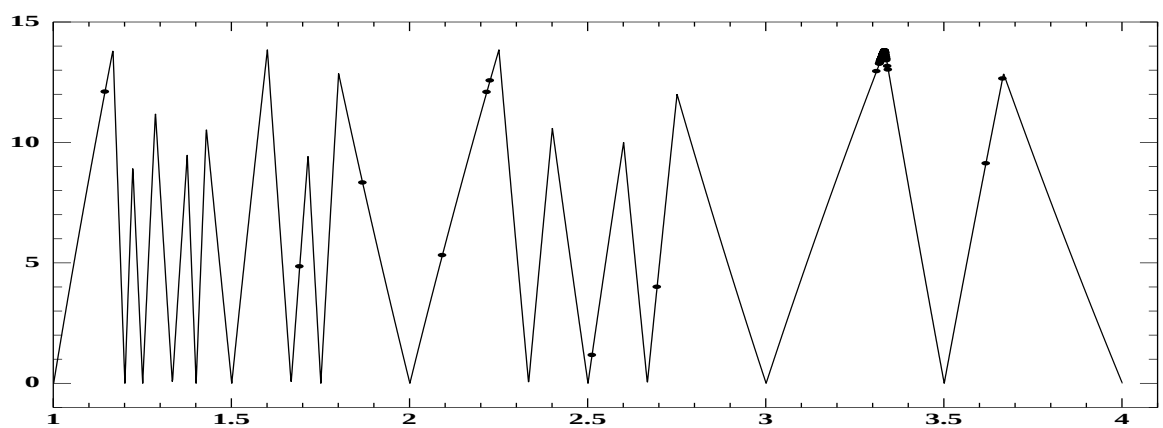
Search space : Direction finders

Place the antennas

But : Maximize tolerance to noise while preserving precision.

The 3-antennas case:

$\mathcal{F}$ = Tolerance (position of 2nd antenna)



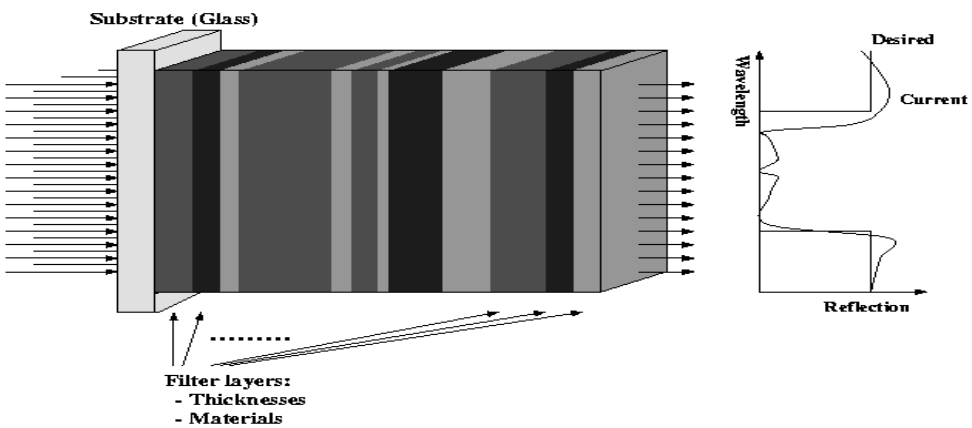
Author : L. Taïeb, CMAP & Thomson.

Mixt search space : discrete $\times$ real-valued

Search space : Optical filters

$$(\text{material, thickness})_1 \ldots (\text{material, thickness})_N$$

Goal : Fit of a prescribed answer w.r.t. wavelength



Authors :

Schutz & Bäck, ICD, Dortmund.

Martin, Rivory & Schoenauer, Paris VI & CMAP.

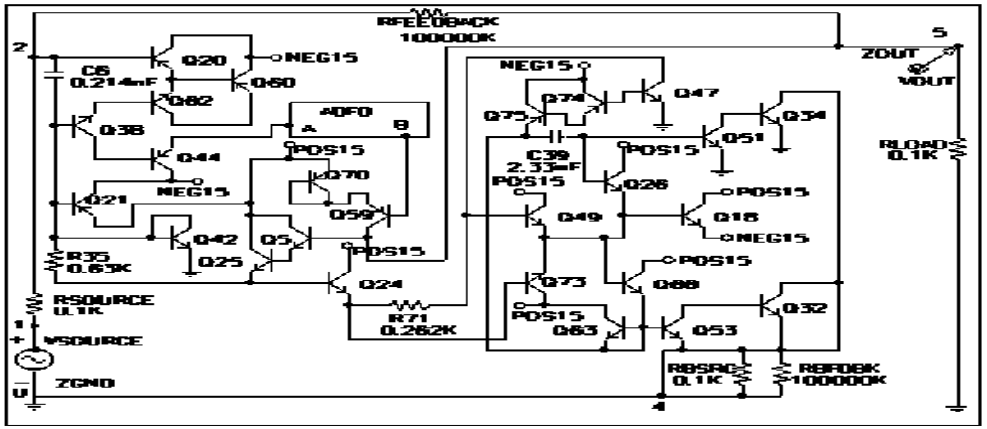
Analog circuits

Search space : Graph + components

Transistors, resistors, capacitors, . . .

Goal : Meet specifications.

e.g. Cubic root extraction



Authors : Koza et al., Stanford.

$\mathcal{F}$ non computable

Search space : Coffee blends

Goal : Give the clients the same aroma

$\mathcal{F}$ = expert advice

Authors : Herdy & al., TU Berlin.

Optimization Algorithms

- Gradient-based algorithms
- Hill-Climbing
- Enumerative methods
- Stochastic methods (meta-heuristics)

Comparison

- Nature of search space
- Regularity of objective function (and/or constraints)
- Local vs global search

Stochastic methods

Select $X_i \in \Omega$

According to some probability distribution

- Monte-Carlo — X_i follows a uniform law
- Simulated annealing — Kirkpatrick, Gelatt and Vecchi, 1983
- Tabu search — F. Glover – 1977 & 1989
- Evolutionary algorithms ... — since 1965

Global methods, VERY costly

But often the last chance :-)

Evolutionary Computation

Contents

1. A biological inspiration
2. Key issues
3. Representation
4. Evolution Scheme (selection, replacement)
5. Variation Operators (crossover, mutation)
6. I tried and it didn't work
7. References

Evolutionary Computation — The roots

- Genetic Algorithms — J. Holland, 75, D.E. Goldberg 89
  AI and biology – US, east coast
- Evolution Strategies — I. Rechenberg 65,73, H.P. Schwefel 65,81
  Engineers – Germany
- Evolutionary Programming — L. Fogel, 66, D. Fogel, 91,95
  Automata and time series – US, west coast
- Genetic Programming — J. Koza, 92 – a late root!
  GAs on parse-trees

Darwinian Paradigm

- Natural Selection survival of species adapted to the environment
- Variability parents → (seemingly?) non directed small deviations
- "Objective" survive and reproduce
- Resulting Adaptation : new species emerge

  e.g. antibiotic-resistant germs

Darwin is

- Source of inspiration
- Didactic help
- **Not** a justification — **Not** a limitation

Evolutionary Algorithms
The metaphor

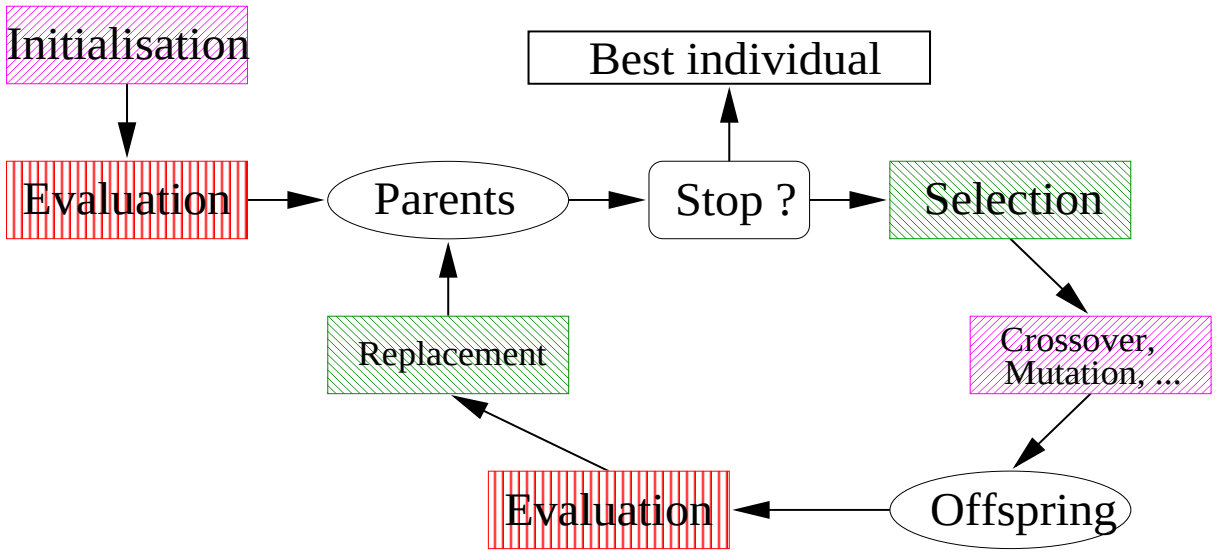
Model : The Darwinian evolution of natural populations.
Survival of the fittest

Wording :

INDIVIDUAL	Element X of Ω
FITNESS OF X	$\mathcal{F}(X)$
POPULATION	$\Pi = \{X_1, .., X_P\}$
GENERATION	From population Π_t to population Π_{t+1}

Process

1) Under environmental pressure,
2) individuals reproduce, recombine, and mutate.
3) After some time, emergence of fit individuals close to the optima of $\mathcal{F}$...



Stochastic operators

"Darwinism" (stochastic or determinist)

Main CPU cost

Evolutionary Computation
Conditions of use

A 0th order method

only requires $\mathcal{F}$ to be computable on Ω

Reserved to ill-posed problems

consistently slower than other approaches

when the latter apply

Requisite

Strong causality principle — Rechenberg, 73

Else : Monte-Carlo methods

But EC can be degraded to Monte-Carlo...

Key issues

- One population, not one individual

  Preserve genetic diversity

- A game with incomplete information

  Exploration vs Exploitation

- Relationship of the evolution engine and the fitness to optimize

  Black box – or tailored optimization ?

Evolutionary Computation Components

- The representation
- The evolution scheme
- The variation operators
- Stopping criterion and Evaluation of the results

Representation
Genotypes — Phenotypes

Genotypes :	The code space	DNA
Phenotypes :	The manifestation space	shape, behavior

$$\varphi : \quad Genotype \quad \mapsto \quad Phenotype$$

$$\mathcal{F} : \quad Phenotype \quad \mapsto \quad \mathbb{R}$$

Genotypic level	initialization, crossover, mutation
Milieu	computes $\mathcal{F} o\ \varphi$
Phenotypic level	Selection, replacement

The choice of the representation

- Conditions the success — a general AI feature

- Governs all genotypic operators:
  - Initialization
  - Crossover
  - Mutation

Old days: Supremacy of binary alphabet.

refutations: Antonisse 89, Radcliffe 91

These days: customize the representation

= use background knowledge

Rationale: BK allows for exploring small relevant subspaces in otherwise intractable spaces

Initialization of the population

Draw $\Pi_0 = \{X_1, \ldots X_P\}$

- Uniform sampling of Ω.

  $\Omega = \{0,1\}^N$, $X_i^j = 0$ or 1 with probability 0.5

  $\Omega = [0,1]^N$, $X_i^j = \text{random}()$

- ... but different uniformity criterion can be used
- Using a priori knowledge — *Inoculate* with good solutions
- Result of another evolution — Use successive milieux $\mathcal{F}_1, \mathcal{F}_2, ..$

Evaluation of the current population

BY FAR the most costly step

in all real-world applications

- Do not re-compute the fitness of unmodified individuals
- Use an approximate fitness as much as possible

  build an estimator from the known exact values
- ... but not for too long

  Optima of approximate fitness $\neq$ real optima

Evolution schemes

Selection :	Individuals (t)	$\mapsto$ Parents
Replacement :	Individuals (t) + Offspring	$\mapsto$ Individuals$(t+1)$

- Control the population trajectory

- Control the trade-off between random walk and hill climbing

Selection

Goal : Select reproducers.

- Darwinism: Bias toward the fittest individuals
- Too strong bias: Loss of diversity
- Too small bias: No convergence

Modes

- Deterministic (from fitness comparisons)
- Stochastic: Fitness (order) proportional
- Use prior knowledge

Deterministic Selection/Replacement

Historically: The $(\mu \overset{+}{,} \lambda)$-ES evolution schemes

Scheme : μ parents give λ offspring

Selection / Replacement :

- (μ, λ)-ES : next μ parents = best among the λ offspring.

  Pros: Good convergence theoretical results

  Diversity more likely preserved

  Cons: Might lose the best solution

- $(\mu + \lambda)$-ES : next μ parents = best among the μ parents and the λ offspring

  Pros: robust

  Cons: loss of diversity, premature convergence

Parameters: μ and λ

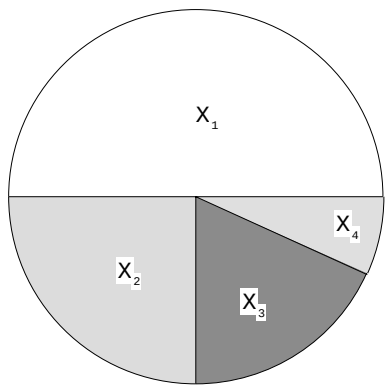
Stochastic selection

(I) Proportional selection: Roulette wheel

What ? Selection of P parents among $\Pi_t = \{X_1, \ldots, X_P\}$

$$Probability(\text{ selecting } X_i) \propto \mathcal{F}(X_i)$$

How ? **A roulette wheel**: Width of sector i : $\dfrac{\mathcal{F}(X_i)}{\sum_j \mathcal{F}(X_j)}$



Selection of one individual : $\mathcal{D}_{\mathcal{F}}$

"Throw the ball" and choose corresponding parent

Roulette wheel (2)

- Roulette wheel: use $\mathcal{D}_{\mathcal{F}}$

- Fitness scaling, tune selective pressure ps $\qquad \mathcal{D}_{\mathcal{F}} \rightarrow \mathcal{D}_{\alpha\mathcal{F}+\beta}$

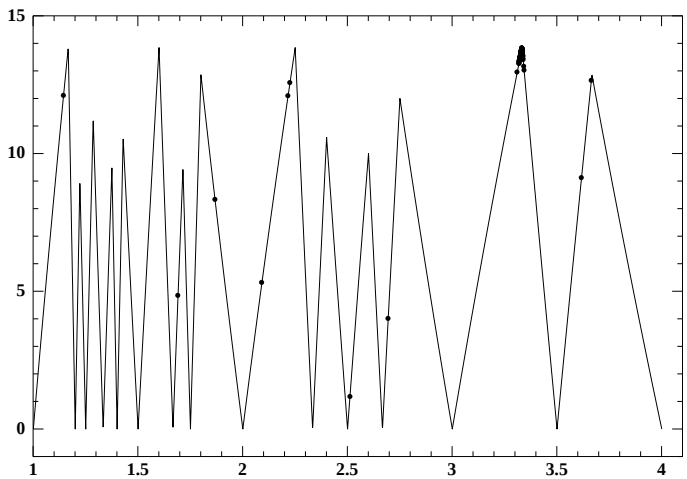
$$ps = \text{expected nb of times } X_{best} \text{ is selected}$$

- Fitness ranking $\qquad$ use $\mathcal{D}_{rank}$

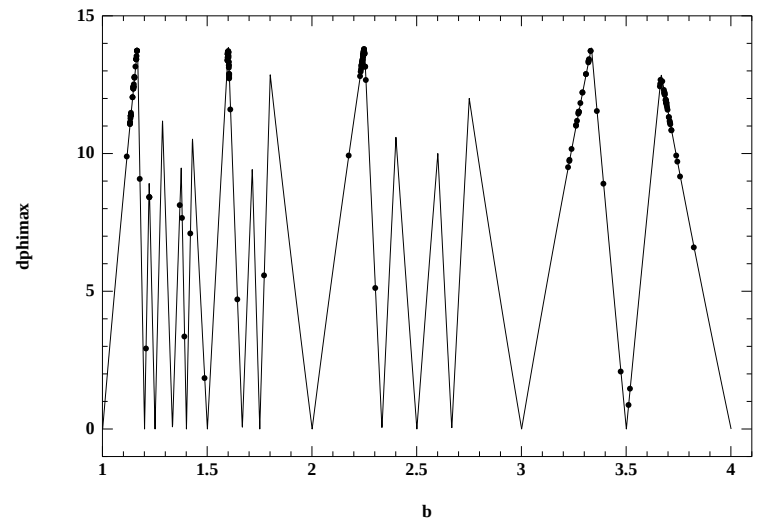
- Fitness sharing, use $\mathcal{D}_{\mathcal{G}}$, with

$$\mathcal{G}(X_i) = \frac{\mathcal{F}(X_i)}{\sum_j similarity(X, X_j)}$$

Example
Direction finders with sharing



Without sharing



With sharing

Stochastic Selection (II) : Tournament selection

$\mathcal{D}_{\mathcal{F},K}$

- Uniform selection $\{X_{i_1}, \ldots, X_{i_K}\}$ in Π_t
- return ArgMax $\mathcal{F}$

Note :

Easy to tune

Easy to parallelize

Variation Operators

Selected Individuals = Parents $\mapsto$ Offspring

MUTATION : $\Omega \mapsto \Omega$

creates a distance

CROSSOVER : $\Omega \times \Omega \mapsto \Omega \times \Omega$

creates an adaptive distance

- Genotypic operators

mutation-based distance should reflect genotypic distance

- Roles: permit exploration AND exploitation

Mutation

$$\Omega \rightarrow \Omega$$

Analogy: asexual reproduction.

Role :

- Exploration — allows evolution to escape

palliates the limitations of crossover

limited by current population diversity

- Exploitation — offspring are on average "close" to the parents.

Historical debate: mutation is

- A background operator for GAs
- The main operator for ES/EP

Bitstring mutations

$$\{0,1\}^N \rightarrow \{0,1\}^N$$

Bit-wise mutation

- For each individual — parent or offspring created by crossover
- and for each bit position l,

$$(b_1, b_2, \ldots, b_N) \xrightarrow{p_m} (b_1, b_2, \ldots, \overline{b_l}, b_{l+1}, \ldots, b_N)$$

- Adjust p_m
  - for being effective — $p_m > \frac{1}{N \times P}$
  - if many successful mutations, — $p_m \nearrow$
  - else — $p_m \searrow$

    (or restart evolution)

Real mutation

$$\mathbb{R}^N \rightarrow \mathbb{R}^N$$

Add Gaussian perturbations $X_i := X_i + N(0, \sigma)$

All expertise in the choice of σ

- Fitness-based (EP-type)
  - Best individuals mutate small — Exploitation
  - Worse individuals mutate large — Exploration
- History-based (Rechenberg **1/5th rule**)
  - Successful mutation if offspring better than parent
  - $\tau = \%$ successful mutations in the last T generations
  - If $\tau > 0.2 \quad \sigma * = 1.22 \qquad$ else $\sigma \div = 1.22$
- Adaptive mutations (modern ES)

σ is adjusted by evolution itself

Adaptive mutation : change the representation

$$\Omega \mapsto \Omega \times \{\text{mutation parameter}\} \qquad X \mapsto (X, \sigma)$$

A two-step mutation

- Mutate σ
- Mutate X_i using the new σ

σ adjusted "for free"

Why ?

- Successive successful mutations unlikely if σ is totally wrong
- Persistent good offspring result from successive successful mutations
- Persistent offspring must have well-tuned σ

Adaptive mutation (2)

- Scalar Std. Deviation $\mathbb{R}^N \mapsto \mathbb{R}^N \times \mathbb{R}^+$

- Vector Std. Deviation $\mathbb{R}^N \mapsto \mathbb{R}^N \times \mathbb{R}^{+\ N}$

$$(X_i, \sigma_i) \rightarrow \begin{cases} \kappa = \tau N(0,1) \\ \text{for each coordinate } j \\ \quad \sigma_i^j := \sigma_i^j \ exp(\kappa + \tau' N(0,1)) \\ \quad X_i^j := X_i^j + N(0, \sigma_i^j) \end{cases}$$

with

$\tau \propto \frac{1}{\sqrt{2\sqrt{N}}}$

$\tau' \propto \frac{1}{\sqrt{2N}}$

- Matrix Std Deviation $\Omega = \mathbb{R}^N \mapsto \Omega \times \mathbb{R}^{+\ N} \times [0, 2\pi]^{N(N-1)/2}$

Crossover

$$\Omega \times \Omega \rightarrow \Omega \times \Omega$$

Analogy : Sexual reproduction

Role : Fast Exploitation

Offspring retain some qualities of both parents

Historical debate: crossover is

- The main operator for AG fans
- A useful operator for ES though initially absent
- A useless (or worst) operator for EP

Binary crossover

$$\{0,1\}^N \times \{0,1\}^N \rightarrow \{0,1\}^N \times \{0,1\}^N$$

Exchange of bits between both parents

2-points crossover DeJong 75

$$\left.\begin{array}{l}(b_1,\ldots,b_N)\\(c_1,\ldots,c_N)\end{array}\right\} \xrightarrow{p_c} \left\{\begin{array}{l}(b_1,..,b_l,c_{l+1},..,c_m,b_{m+1},..,b_N)\\(c_1,..,c_l,b_{l+1},..,b_m,c_{m+1},..,c_N)\end{array}\right.$$

where $l,m = U(\{1..N-1\}), l \neq m$

Real crossover

$\Omega = \mathbb{R}^N$ or $\prod[a_i, b_i]$

Arithmetical crossover

Linear combination of parents

- Offspring in segment (X, Y) $\qquad \alpha = U([0,1])$

$$Crossover(X, Y) = \alpha X + (1 - \alpha)Y$$

- Offspring in hyperrectangle (X, Y) $\qquad \alpha_i = U([0,1])$

$$Crossover(X, Y) = (\alpha_i X^i + (1 - \alpha_i)Y^i)_{i=1}^N$$

Prefer $\alpha_i = U([-r,\ 1 + r])$

Crossover vs mutation

Crossover

- modifications are population-dependent
- "proportional" to diversity
- Exploitation operator (in general)

Mutation

- **Mandatory**
- Exploration operator (in general)
- Must be able to explore the whole search space
- Destructive effects increase with evolution

Elitism

Survival of the best current individual

- Theory : speeds up convergence
  increases the chance of premature convergence

- Application: hate to lose a good solution
  btw, keep best-so-far individual in memory

Elitism

If $\mathrm{Max}(\mathcal{F}(\Pi_{t+1})) < \mathrm{Max}(\mathcal{F}(\Pi_t))$

Replace the worst of Π_{t+1}

by the best of Π_t

Stopping Criterion

- When optimum is found
- When no improvement can be expected

  Loss of diversity
- When the ratio $\dfrac{\text{Expected improvement}}{\text{Amount of required resource}}$ is too high

  (bounded rationality)
- When allocated resources have been spent

  Maximum number of fitness evaluation

Heuristic :

STOP after a given number of evaluations without improvement

User-defined

Results evaluation

From the application point of view

Design (conception) context

Find at least once a very good solution

Production context

Always find a reasonably good solution

Other Aspects

- Multi-modal function

  Find multiple optima using **niching** techniques

- Hybridization

  Coupling with local methods

- Constrained Optimization

- Multi-objective Optimization

- Genetic Programming

Empirical conclusions

Failures

- I tried to use XXX algorithm, but it didn't work ...

  No black box usage

- I tried on that quadratic problem, it was dreadfully slow

  Do not compete with deterministic methods whenever these apply

Recommendations

- Yet unsolved problems

  irregular, "chaotic" functions and/or constraints

- Multiple optima — implicit criteria, multi-criteria
- Ill-posed problems — solution validated by the user
- Coupling with local methods — *ma non troppo*

Bibliography

BOOKS: Oldies but goodies

- L.J. Fogel and A.J. Owens and M.J. Walsh, *Artificial Intelligence Through Simulated Evolution*, John Wiley & Sons, 1966
- J. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, 1975
- H.-P. Schwefel, *Numerical Optimization of Computer Models*, John Wiley & Sons, 1981 (Nelle édition 1995).
- D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison Wesley, 1989

Classic, but outdated

Reference books

- Z. Michalewicz, *Genetic Algorithms+Data Structures = Evolution Programs*, Springer Verlag, 1992–1996.

Still young!

- J. Koza, *Genetic Programming*, I, II & III, MIT Press, 1992, 1994 & 1999 (+ videos!).

- D. B. Fogel, *Evolutionary Computation, Toward a New Philosophy of Machine Intelligence*, IEEE Press, 1995.

- Melanie Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1996.
- T. Bäck, *Evolutionary Algorithms in theory and practice*, New-York:Oxford University Press, 1995.
- W. Banzhaf, P. Nordin, R.E. Keller, F.D. Francone, *Genetic Programming — An Introduction On the Automatic Evolution of Computer Programs and Its Applications*, Morgan Kaufmann, 1997.
- Michael D. Vose, *The Simple Genetic Algorithm: foundations and theory*, MIT Press, 1999.

Journals

- Evolutionary Computation.
- Complex systems.
- BioSystems.
- Journal of Heuristics.
- IEEE Transactions on Evolutionary Computation.
- Journal of Global Optimization (electronic).
- Genetic Programming and Evolvable Hardware Journal (forthcoming).

On the Web

- Evonet: European Network of Excellence on Evolutionary Computation
  http://www.dcs.napier.ac.uk/evonet/evotop.html
- Polytechnique:
  http://www.cmap.polytechnique.fr/www.eark/

GA news:

```
- Send administrative (subscribe, unsubscribe, change of address, etc.,)
requests to GA-List-Request@AIC.NRL.NAVY.MIL
*****************************************************
- You can access back issues, GA code, conference announcements, etc., either
through the WWW at URL http://www.aic.nrl.navy.mil/galist/ or through anonymous
ftp at ftp.aic.nrl.navy.mil [132.250.84.25] in /pub/galist.
```

Conferences

GECCO Genetic and Evolutionary Computation Conference, since 1999, USA
merges ICGA, Int. Conf. on Genetic Algorithms, since 1987, USA
and GP, Genetic Programming Int. Conf, since 1996, USA

PPSN Parallel Problems Solving from Nature, since 1990, Europe

CEC IEEE Conf. on Evolutionary Computation, since 1999, USA + Japan
merges EP Annual Conf. on Evolutionary Programming, since 1992, USA
and IEEE Int. Conf. on Evolutionary Computation, since 1994, World-wide

EuroGP, joint with Evonet Workshops
European Conf. on Genetic Programming, since 1998, Europe

EA Artificial Evolution, since 1994, France & Europe