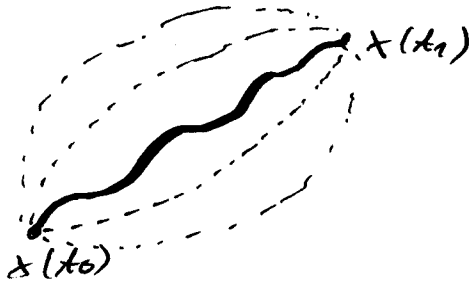


Optimization



Prelude: Variational principles and optimal processes

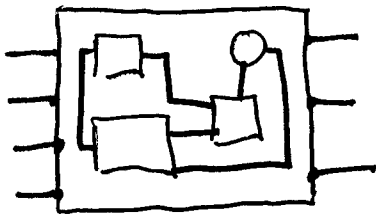
• Least action principles:



$$\delta \int_{t_0}^{t_1} L(x, \dot{x}) dt = 0$$

(Similarly: Fermat's principle, WKB approximation.....)

• Engineering:

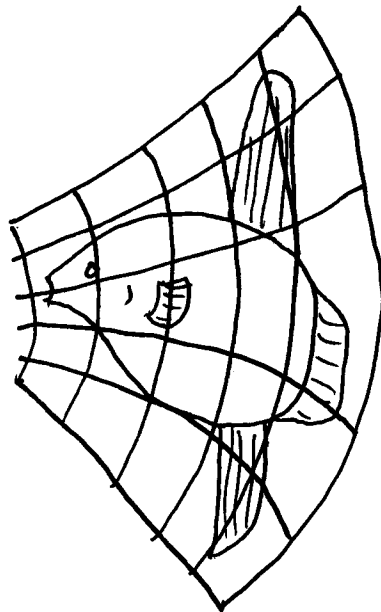
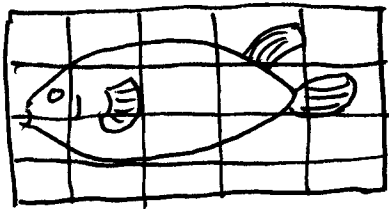


micro chip

Optimize design
(Wires, Devices)

• Biology:

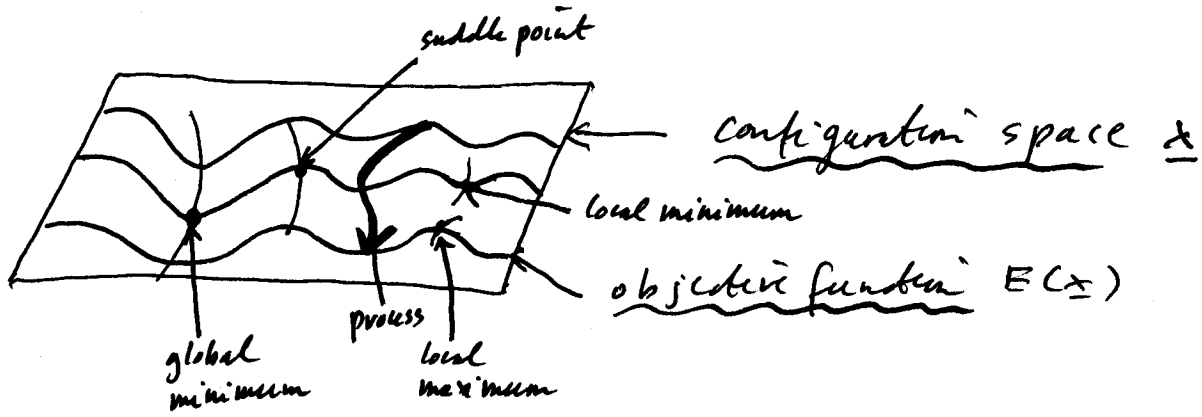
Evolution (Darwin)



Optimized shape:
environment, prey,

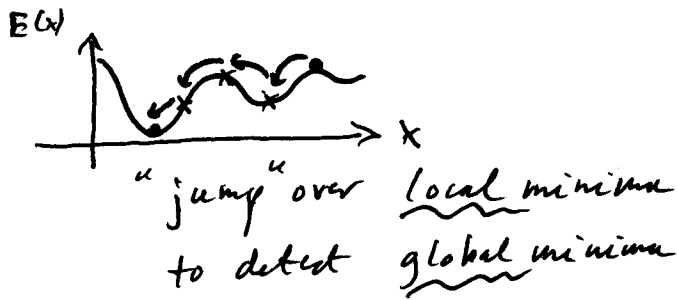
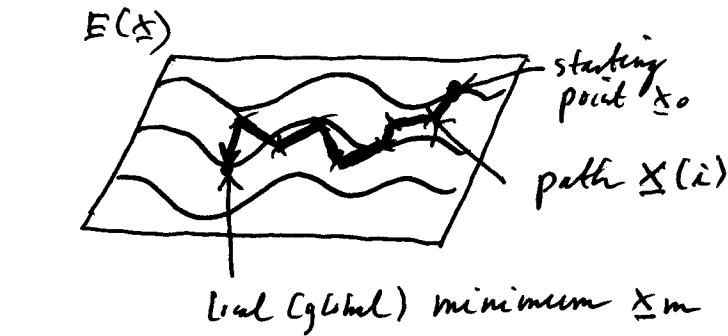
=> Engineering (Bionics)

• The problem:



- => Determine extremal points (local, global)
- => Determine optimal process ("time"-dependent \rightarrow path)

• Simulated annealing:



Physical process:

annealing: liquid freezes and crystallizes

slow cooling: stable phase

fast cooling: metastable phase ('quenching')

"Optimic" phase: tempering of metastable phase

=> introduce temperature T: probabilistic theory (Boltzmann)

$$\text{Prob}(E) \sim \exp(-E/kT)$$

Changes:

$$\text{Prob}(E_1 \rightarrow E_2) \sim \exp(-(E_2 - E_1)/kT)$$

Metropolis algorithm (1953):

Ingredients:

1. Description of possible system configurations.
2. Generator of random changes in the configuration.
3. Suitable objective function E (analog of energy)
4. Control parameter T (analog of temperature);
annealing schedule (variations of T ...)

thermodynamical process à la Metropolis:

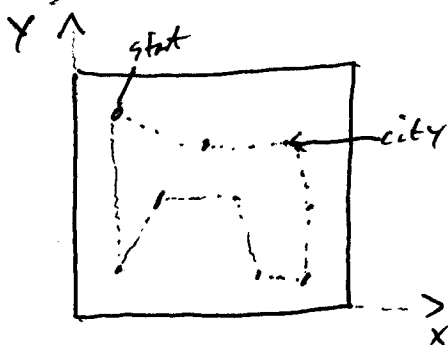
$$\left. \begin{array}{l} E_1 = E(x_1) \\ E_2 = E(x_2) \end{array} \right\} \text{Prob}(E_1 \rightarrow E_2) = \exp[-(E_2 - E_1)/kT]$$

if $E_2 < E_1$: $\text{Prob}(E_1 \rightarrow E_2) > 1$ (accept that step)

if $E_2 > E_1$: generate random number $r \in [0, 1]$
if $\text{Prob}(E_1 \rightarrow E_2) > r$ (accept that step)

if $\text{Prob}(E_1 \rightarrow E_2) < r$ (discard that step)

Classical problem:



Path: - visit all cities
- return to start
=> optimize path (shortest path)

Traveling sales man

Configuration: cities at (x_i, y_i)
 $i = 1, \dots, N$

Random changes: permutations of
 $1, \dots, N$

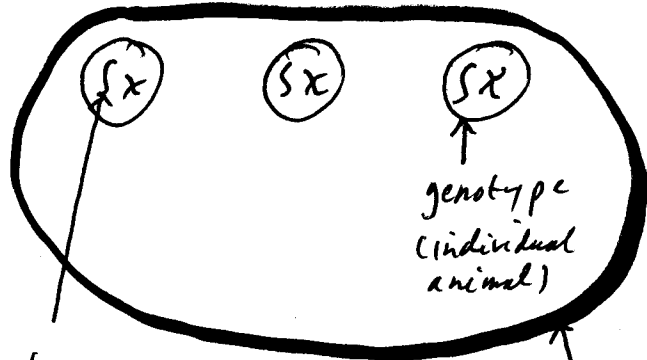
Objective function: length of journey

$$E = L \equiv \sum_{i=1}^N \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$

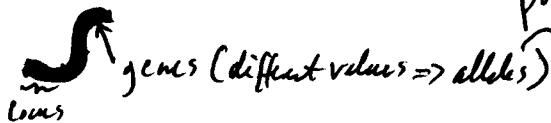
where: $(x_{i+1}, y_{i+1}) = (x_1, y_1)$
(see Numerical recipes)

• Genetic algorithms:

"Poor man's genetics":



Chromosomes:



population

- every individual may be characterized by a certain fitness due to its phenotype

- evolutionary processes:

individuals are subject to natural selection, depending on their fitness

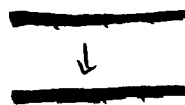
- biological operations:

1. reproduction

2. cross over



3. mutation

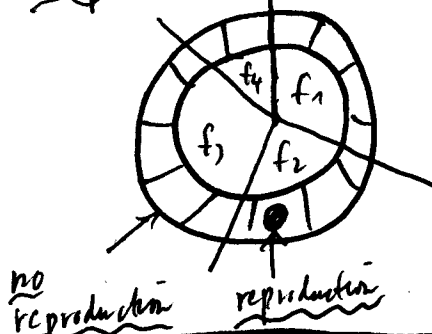


Optimization procedure:

1. Translate configuration space into chromosomes (genotype)
2. Generate initial population.
3. Evaluate fitness function for each genotype.
4. Subject population to biological processes:

- crossover (rate P_c)
 - mutation (rate P_m)
- usually $P_m \ll P_c$

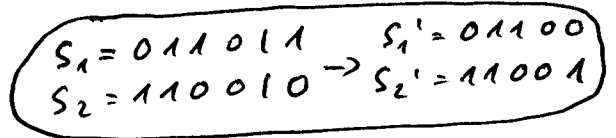
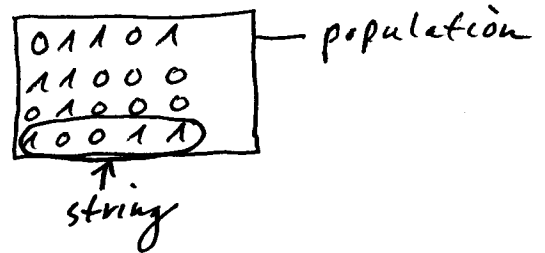
- reproduction:



Roulette wheel

(partitioning due to individual fitness)

=> generate new generation



Final generation: mainly composed of optimal individuals.

Line minimizations:

Goal: $E(\underline{x}) \rightarrow \min$

iterative methods: $\{\underline{x}^k\}$; $\underline{x}^{k+1} = \underline{x}^k + \alpha_k \underline{d}^k$
direction

steps (line minimization):

- given \underline{x}^k and direction \underline{d}^k , find α_k which minimizes $E(\underline{x}^k + \alpha_k \underline{d}^k)$.
- replace: $\underline{x}^{k+1} = \underline{x}^k + \alpha_k \underline{d}^k$

Practical procedure: Find α_k , such that $E(\underline{x}^k + \alpha_k \underline{d}^k) \ll E(\underline{x}^k)$
"sufficiently"

Powell's method:

Conjugate directions \Leftrightarrow "non-interfering" directions

- initial set of directions: $\underline{u}_i = \underline{e}_i$ (basis vectors)

- starting configuration \underline{x}_0

- for $i=1, \dots, N$ move \underline{x}^{i-1} to minimum along \underline{u}_i

$\Rightarrow \underline{x}^i$

- for $i=1, \dots, N-1$ set $\underline{u}_i \leftarrow \underline{u}_{i+1}$ } \underline{x}_1 discarded

- Set: $\underline{u}_N \leftarrow \underline{x}_N - \underline{x}_0$ } new direction

- Move \underline{x}_N to minimum along \underline{u}_N , call this point \underline{x}_0

Problem: Linear dependence, induced by throwing away \underline{x}_1
in favor of $\underline{x}_N - \underline{x}_0$

Solution: reinitialize $\underline{u}_i = \underline{e}_i$ after N or $N+1$ iterations
of the basic procedure.

• Conjugate directions:

"non-interfering" directions:

- minimize a function along some direction \underline{u}
- choose new direction \underline{v} ; it should not "spoil" the previous minimization along $\underline{u} \Rightarrow$ change in gradient should be perpendicular to \underline{u} !

\hookrightarrow gradient remains perpendicular to \underline{u} !!!

Technically:

$$f(\underline{x}) = \underbrace{f(\underline{0})}_c + \sum_i \underbrace{\frac{\partial f}{\partial x_i}}_{-b} x_i + \frac{1}{2} \sum_{i,j} \underbrace{\frac{\partial^2 f}{\partial x_i \partial x_j}}_A x_i x_j + \dots$$

$$\Rightarrow \nabla f(\underline{x}) = \underline{A} \underline{x} - \underline{b}$$

changes in ∇f : $\delta(\nabla f) = \underline{A} \cdot (\delta \underline{x})$

\Rightarrow Criteria: $\delta(\nabla f) \perp \underline{u}$ along \underline{v}

thus: $0 = \underline{u} \cdot \delta(\nabla f) = \underline{u} \cdot \underline{A} \cdot \underline{v}$

A-orthogonal basis:

$\underline{A} \underline{x} = \underline{b}$ can be solved with the help of an

A-orthogonal basis of conjugate directions $\{p^j\}_n$:

$$(\underline{A} p^i | p^j) = 0 \quad \text{if } i \neq j$$

$$(\underline{A} p^i | p^i) \neq 0 \quad \forall i=1..n$$

Solution: $\underline{x} = \sum_{j=1}^n \eta_j p^j$ with $\eta_j = \frac{(b, p^j)}{(\underline{A} p^j | p^j)}$
 $\forall j=1..n.$

Construction of suitable basis for minimization:

$f(\underline{x}) \rightarrow \min$ iterative solution: $\underline{x}^1 \rightarrow \underline{x}^2 \rightarrow \dots \underline{x}^k \dots$

assume: $f(\underline{x}) \approx \frac{1}{2} (\underline{x}, \underline{A} \underline{x}) - (\underline{b}, \underline{x}) + c$

"Defect": $\underline{d}^{k+1} = \underline{b} - \underline{A} \underline{x}^{k+1}$ for iteration \underline{x}^{k+1}

$$\Rightarrow \underline{p}^{k+1} = \underline{d}^{k+1} + \sum_{j=1}^k \beta_{kj} \underline{p}^j$$

using: $(\underline{A} \underline{p}^{k+1}, \underline{p}^j) = 0$; $j=1 \dots k$

$$\beta_{kj} = \frac{(\underline{A} \underline{p}^j, \underline{d}^{k+1})}{(\underline{A} \underline{p}^j, \underline{p}^j)} ; j=1 \dots k$$

• Conjugate gradient method:

$$\underline{d}^{k+1} = -\underline{\nabla} f(\underline{x}^k)$$

- start with: $\underline{p}^1 = \underline{d}^1 = -\underline{\nabla} f(\underline{x}^1)$

for $k=1, 2, \dots$ as long as $\underline{d}^k \neq 0$:

- Determine $\alpha_k > 0$ from: $f(\underline{x}^k + \alpha_k \underline{d}^k) \leq f(\underline{x}^k + \alpha \underline{d}^k)$;
 $\forall \alpha \geq 0$

- New iteration point: $\underline{x}^{k+1} = \underline{x}^k + \alpha_k \underline{p}^k$

$$\underline{d}^{k+1} = -\underline{\nabla} f(\underline{x}^{k+1})$$

- New direction: $\underline{p}^{k+1} = \underline{d}^{k+1} + \beta_k \underline{p}^k$

no knowledge
of \underline{A}
necessary !!!

$$\beta^k = \frac{(\underline{\nabla} f(\underline{x}^{k+1}), \underline{\nabla} f(\underline{x}^{k+1}))}{(\underline{\nabla} f(\underline{x}^k), \underline{\nabla} f(\underline{x}^k))}$$

(Fletcher / Reeves)

$$\beta^k = \frac{(\underline{\nabla} f(\underline{x}^{k+1}), \underline{\nabla} f(\underline{x}^{k+1}) - \underline{\nabla} f(\underline{x}^k))}{(\underline{\nabla} f(\underline{x}^k), \underline{\nabla} f(\underline{x}^k))}$$

(Polak / Ribiere)

• Newton-Raphson method:

$$f(\underline{x}) \rightarrow \min \Rightarrow \nabla f(\underline{x}^k) = \underline{0}$$

- Iteration: $\underline{x}^0 \rightarrow \underline{x}^1 \rightarrow \dots \underline{x}^k \dots$

- Local approximation (\underline{x}^k):

$$F_k(\underline{x}) = f(\underline{x}^k) + \nabla f(\underline{x}^k)(\underline{x} - \underline{x}^k) + \frac{1}{2}(\underline{x} - \underline{x}^k) \underbrace{\nabla^2 f(\underline{x}^k)}_{\underline{H} \text{ (Hesse matrix)}}(\underline{x} - \underline{x}^k)$$

- Determine \underline{x}^{k+1} from $F_k(\underline{x}) \rightarrow \min$:

$$\nabla f(\underline{x}^k) + \nabla^2 f(\underline{x}^k)(\underline{x}^{k+1} - \underline{x}^k) = \underline{0}$$

$$\Rightarrow \underline{d} = \underline{x}^{k+1} - \underline{x}^k = -\nabla^2 f(\underline{x}^k)^{-1} \nabla f(\underline{x}^k)$$

Newton-Raphson-step

Problem:
$$\left[\nabla^2 f(\underline{x}^k) \right]_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}(\underline{x}^k)$$

$$\underline{H}_{ij}$$

\underline{H}_{ij} has to be inverted (see above)

Solution: Approximate $\underline{H}_{ij}(\underline{x}^k)^{-1}$ using information from previous $\underline{x}^{k'}$ with $k' < k$, in particular $\nabla f(\underline{x}^{k'})$

\Rightarrow Quasi-Newtonian methods

• Quasi-Newton method:

$$\underline{x}^{k+1} = \underline{x}^k - \lambda_k \underline{s}^k \quad \left(\begin{array}{l} \text{Line minimization:} \\ f(\underline{x}^{k+1}) \approx \min_{\lambda} f(\underline{x}^k - \lambda \underline{s}^k) \end{array} \right)$$

$$\underline{s}^k = \underline{H}(\underline{x}^k)^{-1} \underline{\nabla} f(\underline{x}^k) \quad \left(\begin{array}{l} \text{Newton-Raphson} \\ \text{step} \end{array} \right)$$

Approximation:

$$\underline{s}^k = \underline{H}_k \underline{\nabla} f(\underline{x}^k)$$

approximate Hessian

Define:

$$\underline{p}^k = \underline{x}^{k+1} - \underline{x}^k$$

$$\underline{q}^k = \underline{\nabla} f(\underline{x}^{k+1}) - \underline{\nabla} f(\underline{x}^k)$$

Recursion (Update):

$$\gamma_k > 0 \quad ; \quad \theta_k \geq 0$$

$$\underline{H}_k = \underline{\Psi}(\gamma_k, \theta_k, \underline{H}_k, \underline{p}^k, \underline{q}^k)$$

update function

where: $\underline{\Psi}(\gamma, \theta, \underline{H}, \underline{p}, \underline{q}) = \gamma \underline{H}$

$$+ \left(1 + \gamma \theta \frac{\underline{q}^T \underline{H} \underline{q}}{\underline{p}^T \underline{q}} \right) \frac{\underline{p} \underline{p}^T}{\underline{p}^T \underline{q}} - \gamma \frac{(1-\theta)}{\underline{q}^T \underline{H} \underline{q}} \underline{H} \underline{q} \cdot \underline{q}^T \underline{H}$$

$$- \frac{\gamma \theta}{\underline{p}^T \underline{q}} (\underline{p} \underline{q}^T \underline{H} + \underline{H} \underline{q} \underline{p}^T)$$

Special cases:

a) $\gamma_k = 1; \theta_k = 0$ Davidson / Fletcher / Powell (DFP) method

b) $\gamma_k = 1; \theta_k = 1$ Broyden / Fletcher / Goldfarb / Shanno (BFGS) method

c) $\gamma_k = 1; \theta_k = \frac{\underline{p}_k^T \underline{q}_k}{(\underline{p}_k^T \underline{q}_k - \underline{q}_k^T \underline{H}_k \underline{q}_k)}$ Broyden method

• Example (Quantum chemistry):

$$E(\underline{R}) = \langle \phi(\underline{x}) | H(\underline{R}, \underline{x}) | \phi(\underline{x}) \rangle \quad \text{Energy}$$

with:

$$\text{Born-Oppenheimer approximation} \left\{ \begin{array}{ll} \underline{R} & \text{nuclear coordinates} \\ \underline{x} & \text{electronic coordinates} \\ H(\underline{R}, \underline{x}) & \text{Hamiltonian} \\ |\phi(\underline{x})\rangle & \text{electronic wave function} \\ & (\langle \phi(\underline{x}) | \phi(\underline{x}) \rangle = 1) \end{array} \right.$$

Hellmann-Feynman forces:

$$\frac{\partial E}{\partial R_i} = \left\langle \frac{\partial \phi}{\partial R_i} | H | \phi \right\rangle + \left\langle \phi | H | \frac{\partial \phi}{\partial R_i} \right\rangle + \left\langle \phi | \frac{\partial H}{\partial R_i} | \phi \right\rangle$$

$$\Rightarrow \frac{\partial E}{\partial R_i} = \left\langle \phi | \frac{\partial H}{\partial R_i} | \phi \right\rangle \quad \text{as } \underbrace{H | \phi \rangle = \epsilon | \phi \rangle}_{\text{Schrodinger-Gl.}}$$

in practice: $|\phi\rangle \approx |\phi(\underline{R})\rangle$ due to approximations

Structure optimizations:

$$E(\underline{R}) = E(\underline{R}_0) + \left[\frac{\partial E}{\partial \underline{R}}(\underline{R}_0) \right] (\underline{R} - \underline{R}_0) + \frac{1}{2} (\underline{R} - \underline{R}_0)^T \underbrace{\left[\frac{\partial^2 E}{\partial R_i \partial R_j}(\underline{R}_0) \right]}_{\underline{H}(\underline{R}_0)} (\underline{R} - \underline{R}_0) + \dots$$

$$\Rightarrow \boxed{(\underline{R} - \underline{R}_0) = - \underline{H}^{-1}(\underline{R}_0) \cdot \frac{\partial E}{\partial \underline{R}}(\underline{R}_0)}$$

(conjugate gradient method, Quasi-Newton methods, ...)