Probabilistic Algorithms

Evolutionary Algorithms
Simulated Annealing

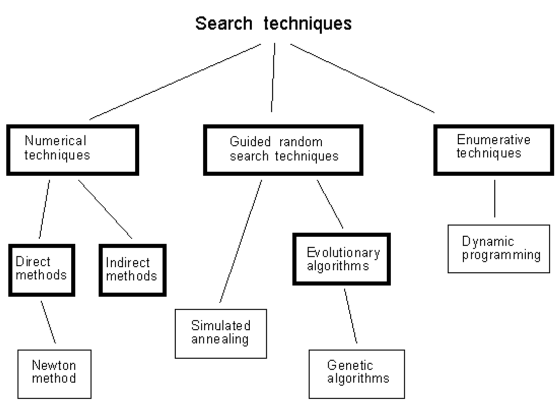


Figure 5.1: Classes of search techniques.

Characteristics

Metaheuristics
"Metaheuristics, although they also optimize through the neighbourhood approach, differ from heuristics in that they can move through neighbours that are worse solutions than the current solution"

Finds global solution – in the limit
But no guarantee of finding global optimum

Large complex search space
high dimensional
multiple local optima

Termination criteria

Allocated time exceeded

Little improvement at iteration

Within threshold of target value

<u>Evolutionary algorithms</u>

Selection by fitness
Reproduction
Inheritance
Mutation
Crossover

Genetic Algorithms – most popular EA

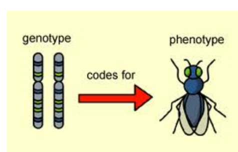
Genetic Programming

Evolutionary Programming

Neuroevolution

etc.

Genetics



Chromosomes, genotype of genome

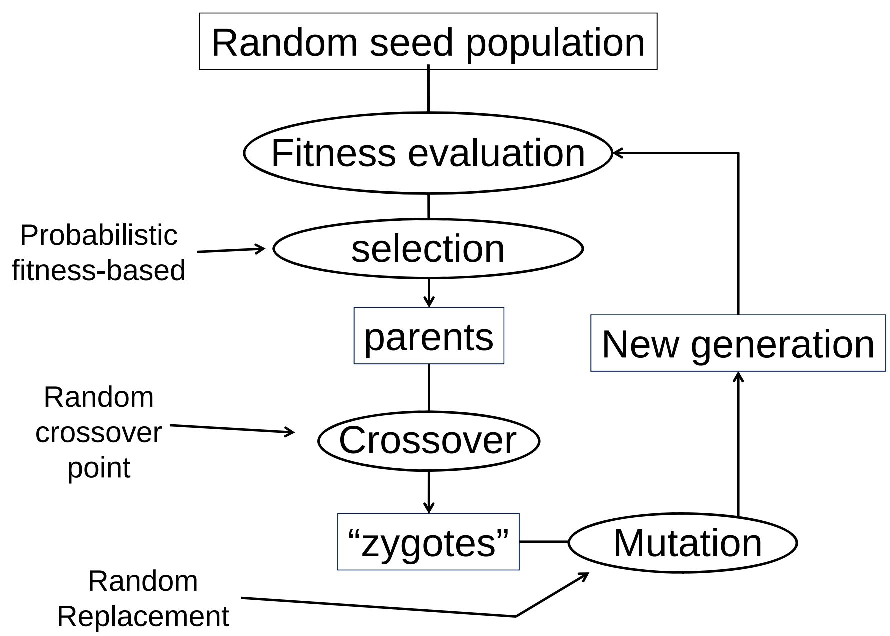
Candidate solutions - phenotype

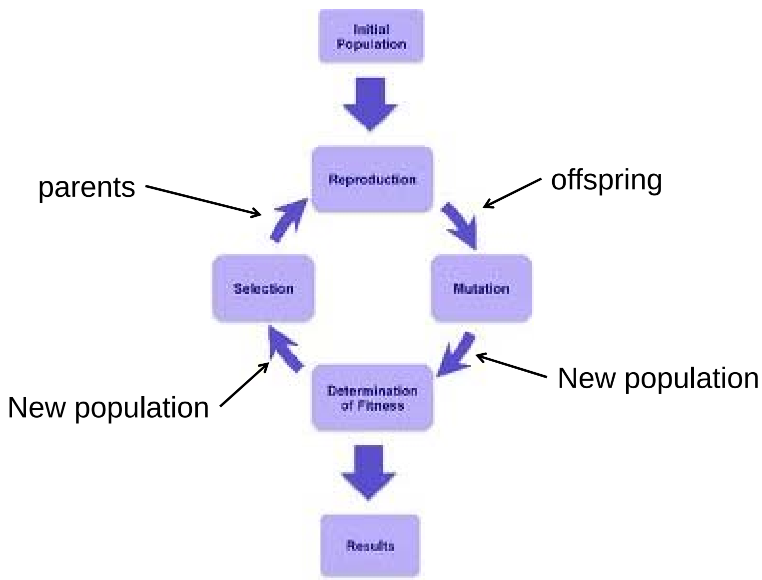
Encoding

1. Binary vector
2. Continuous variables – finite representation

<u>Robustness desirable</u>

All changes result in viable individual



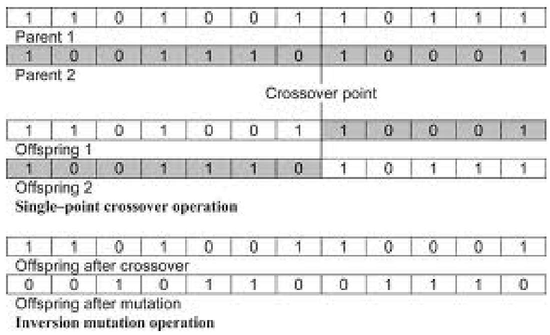


Fitness evaluation & selection

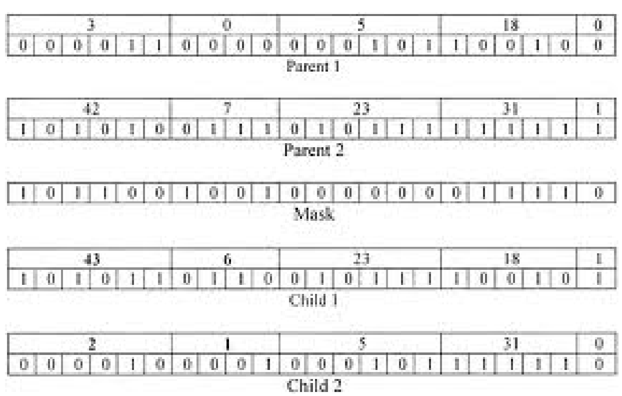
Fitness function
evaluates “goodness” of individual

Select fit and some not-so-fit

Crossover



Masking



Mutation

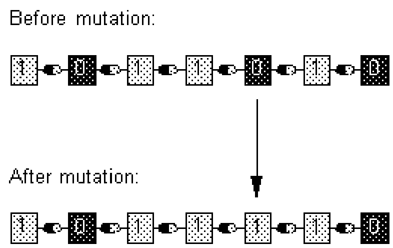
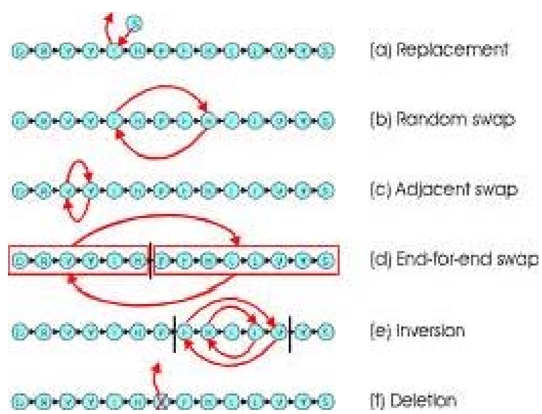


Figure 5.4: Mutation.

Mutation



Generations

Keep same size population

Follow promising lines by
1. Mating fit parents
2. Crossover

Global search by
1. Mutation
2. Unfit parent selection

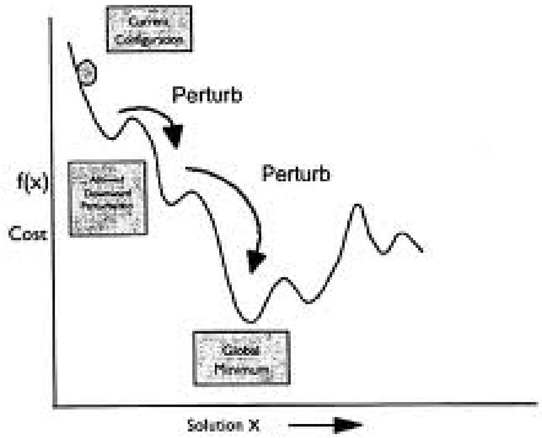
Simulated Annealing

Metalurgy

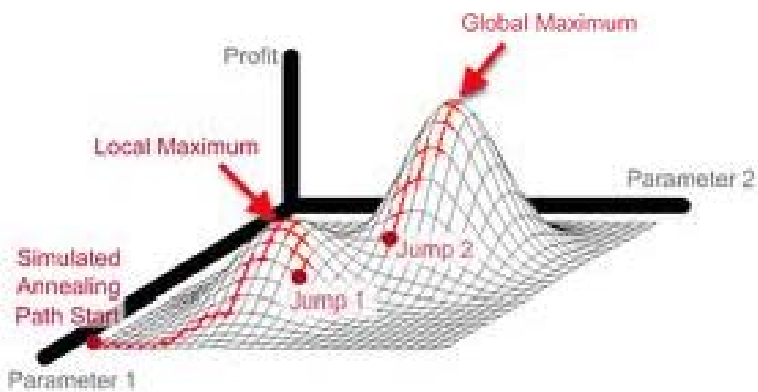
internal energy, heat

Raise temperature to unstick atoms

To find configurations with lower internal energy



Simulated Annealing can escape local minima with chaotic jumps



State Space

State space
{s}

Generate neighbor
Neighbor function
s'=neighbor(s)

Probabilistically move to s'

Evaluate state – probablistic move

Energy function (fitness function)
e = E(s)

Compute energy of neighbor
e'=E(s')

P(e',e,T) — Probability of
going from state with energy e
to state with energy e'
While temperature is T

Acceptance Probability Function P()

Acceptance function: P(e,e',T)

P <> 0 when e'>e
=> not stuck at local minimum

As T->0, P->0 for e'>e => downhill

Originally P(e,e',T)=1 whenever e'>e

Usually P decreases as e'-e increases

T->0 by time or compute expense

$P(e',e,T_{large}) > P(e',e,T_{small})$

State Parameters

State space {s}

Energy function E(s)

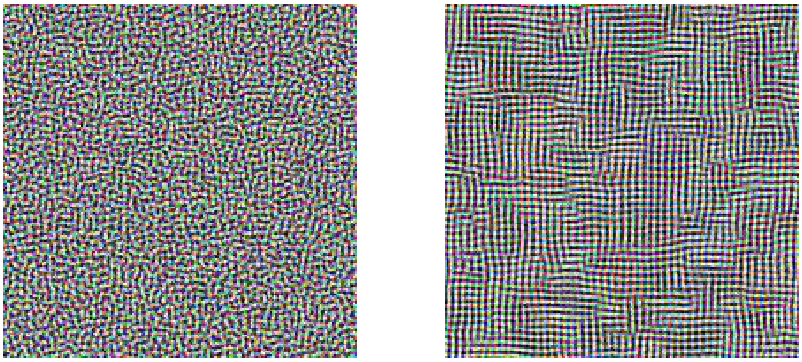
Neighborhood function neighbor(s)

Acceptance probability P(e',e,T) e.g. exp((e-e')/T)

Annealing schedule T(t)

Initial temperature T_{init}

Choose similar solution, not radical one ?



Example illustrating the effect of cooling schedule on the performance of simulated annealing. The problem is to rearrange the pixels of an image so as to minimize a certain potential energy function, which causes similar colours to attract at short range and repel at a slightly larger distance. The elementary moves swap two adjacent pixels. These images were obtained with a fast cooling schedule (left) and a slow cooling schedule (right), producing results similar to amorphous and crystalline solids, respectively.