Chapter 6: Optimization Algorithms

Outline

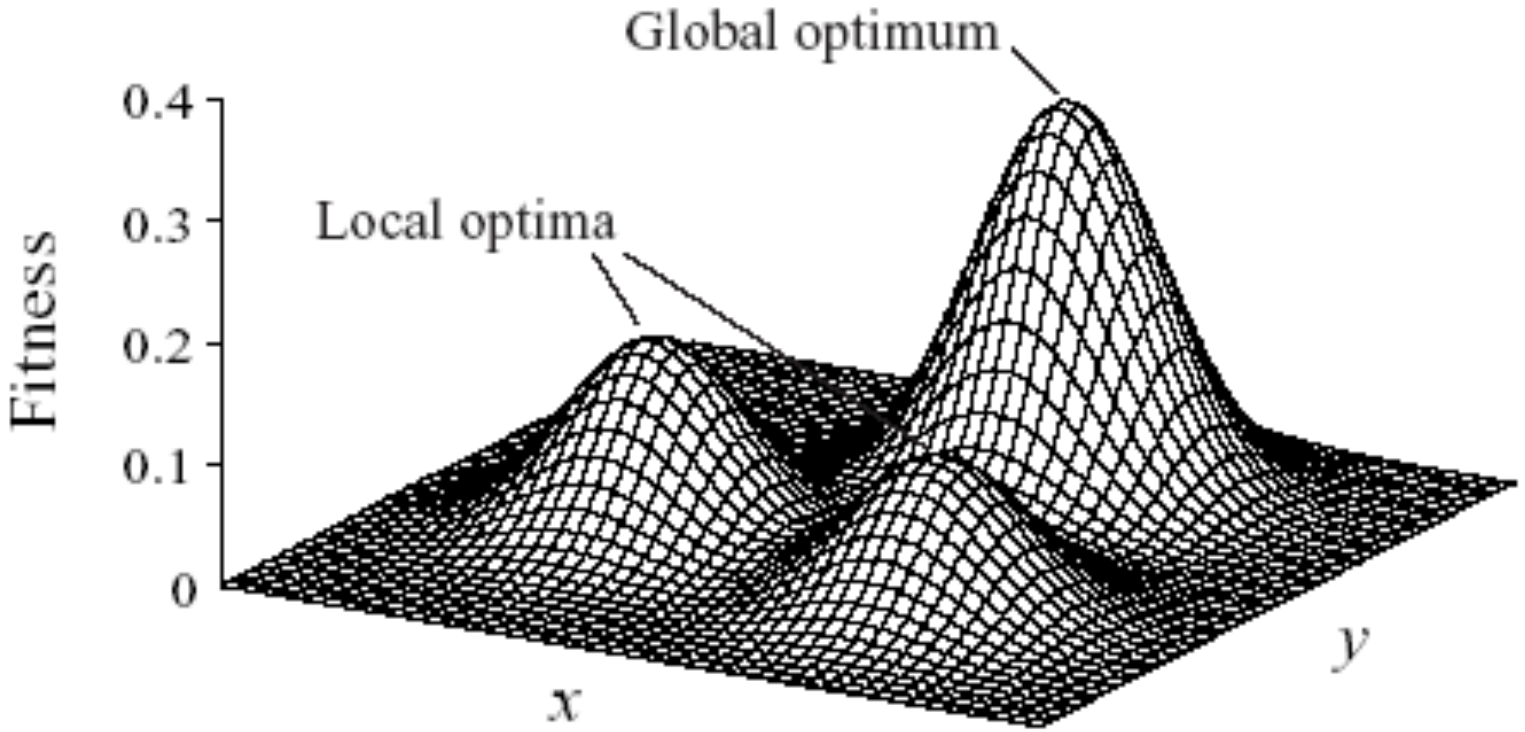
- Cost – fitness – objective function
- Search space
- Hill-climbing
- Simulated annealing
- Genetic Algorithm

Learning problem

- Symbolic learning
  - Steering rules by examples
- Numerical learning ➔ optimization
  - Minimizing cost function
  - Maximizing fitness function
  - Dynamically optimizing objective function

Search space

- Search space = parameter space
  - Combinatorial problem
  - Permutation problem
- Global optimum vs. local optimum


Global optimum
Local optima
Fitness
0.4
0.3
0.2
0.1
0
x
y

Global optimum-oriented searching

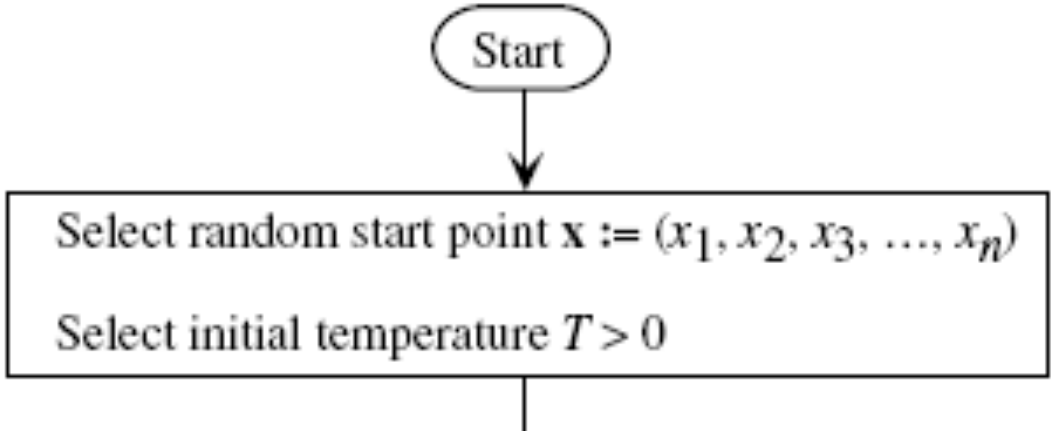
- Hill-climbing
- Simulated annealing
- Genetic algorithm

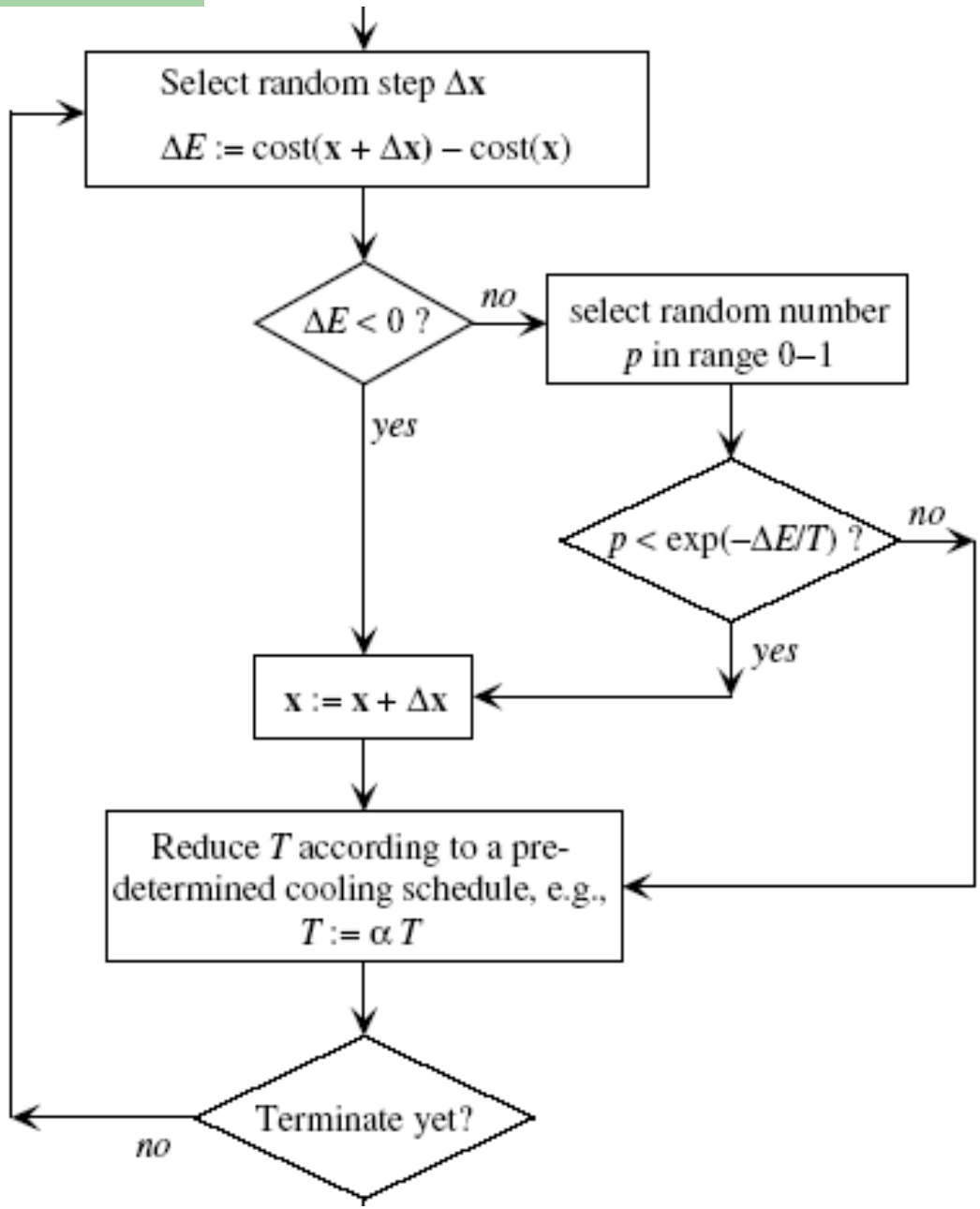
Hill-climbing algorithms

- Hill-climbing
- Steepest gradient
- Gradient-proportional descent

Simulated annealing

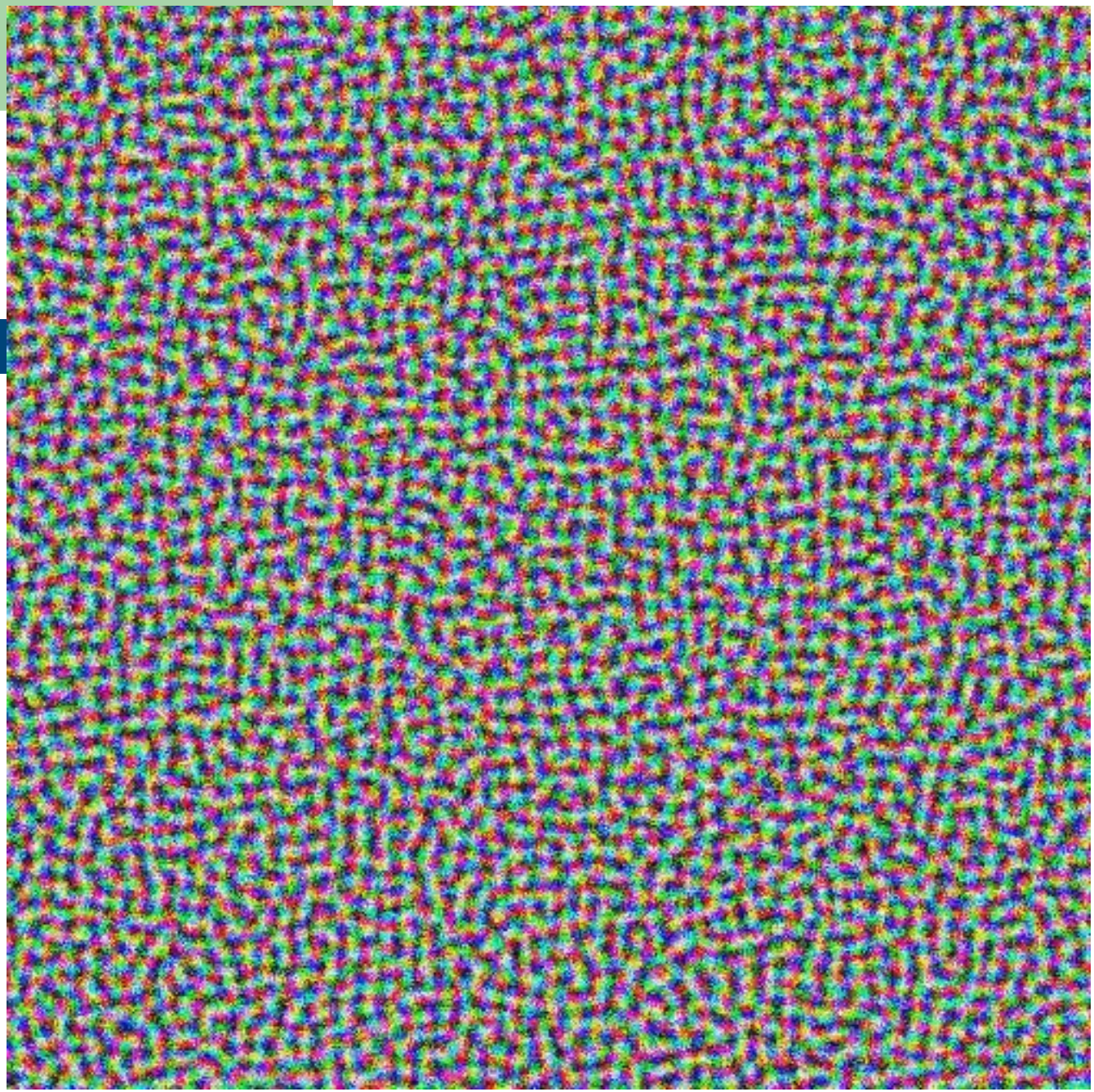
- Simulating the process of annealing metal atoms
- High temperature ➔ high probability for atoms to overcome (break) local energy

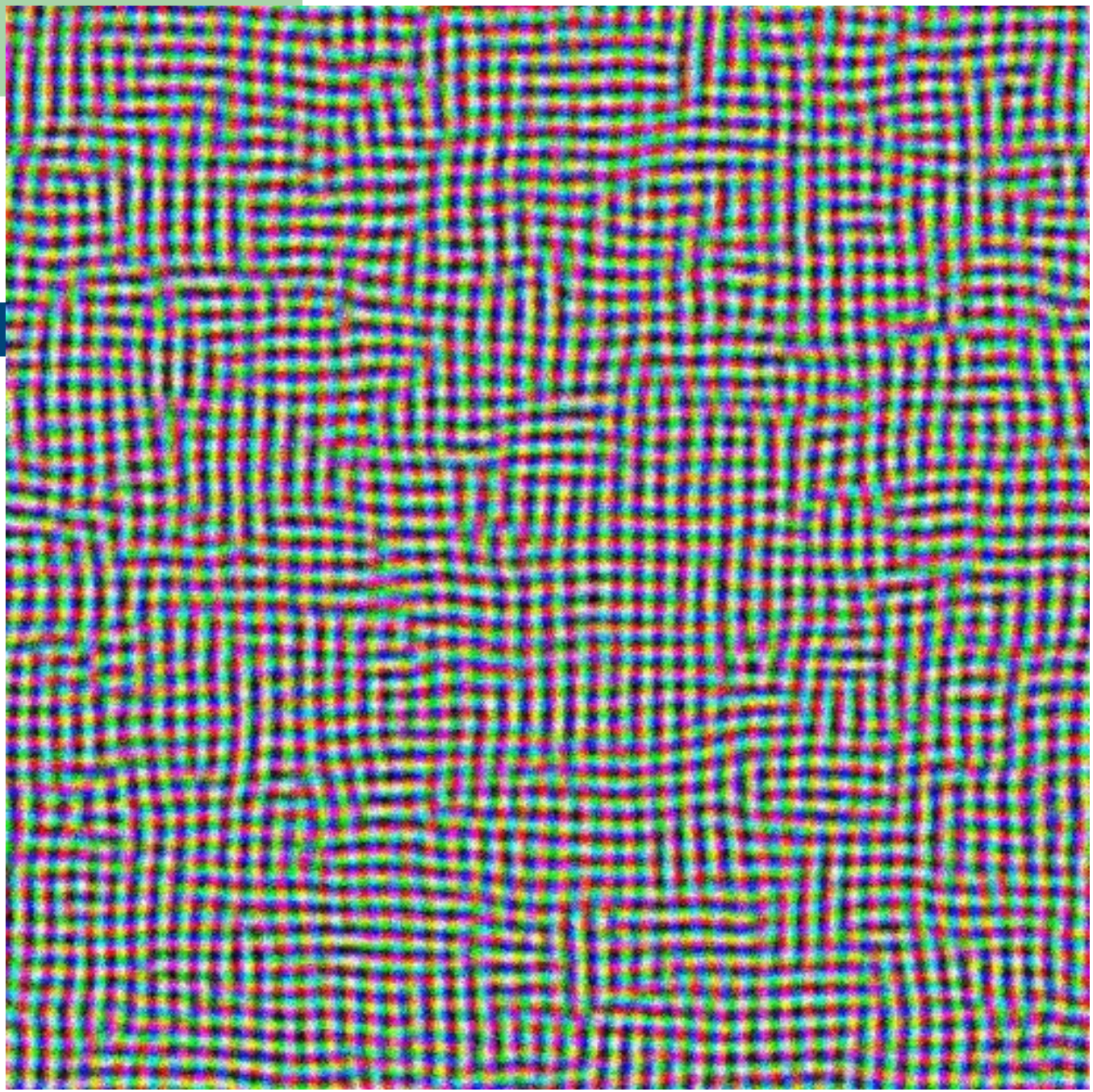

Start
Select random start point $\mathbf{x} := (x_1, x_2, x_3, \ldots, x_n)$
Select initial temperature $T > 0$


Select random step Δx
ΔE := cost(x + Δx) – cost(x)
ΔE < 0 ?
no
select random number
p in range 0–1
yes
p < exp(–ΔE/T) ?
no
yes
x := x + Δx
Reduce T according to a pre-
determined cooling schedule, e.g.,
T := α T
Terminate yet?
no

Acceptance Probability

- P = exp(-E_a/kT)
- P = exp(-$\Delta E/T$)
- P = $\dfrac{1}{1+\exp(\Delta E/T)}$





Genetic Algorithm

- Inspired by natural evolution
- Claimed to reach global optimum

Genetic Algorithm

- Using chromosomes to represent candidate solutions
- Generating a population of candidates
- Fitness evaluation
- Selection for reproduction
- Reproduction
  - Mating
  - Cross-over
  - Mutation
  - Selection

Basic GA

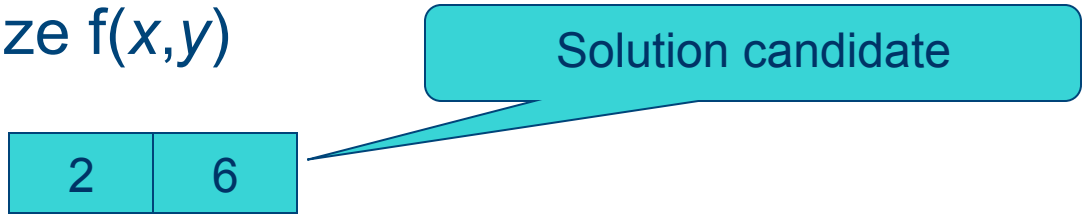
- Make thing “genetic”
  - Maximize f(x,y)

Basic GA

- Make thing “genetic”
  - Maximize f(*x*,*y*) (log(x^2+x^y)-cos(y^x))

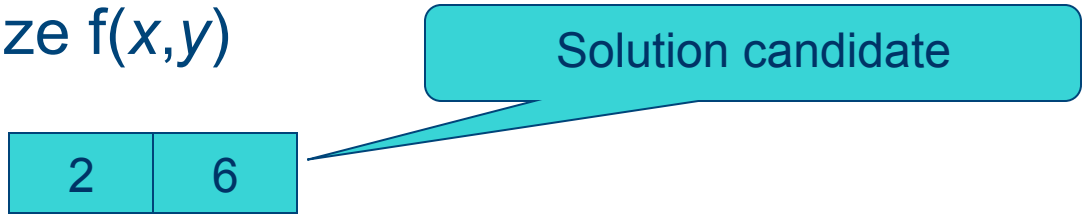
Basic GA

- Make thing “genetic”
  - Maximize f(*x*,*y*)



Basic GA

- Make thing “genetic”
  - Maximize f(*x*,*y*)



  - Allele
  - Locus

Basic GA

- Make thing “genetic”
  - Maximize f(*x*,*y*)

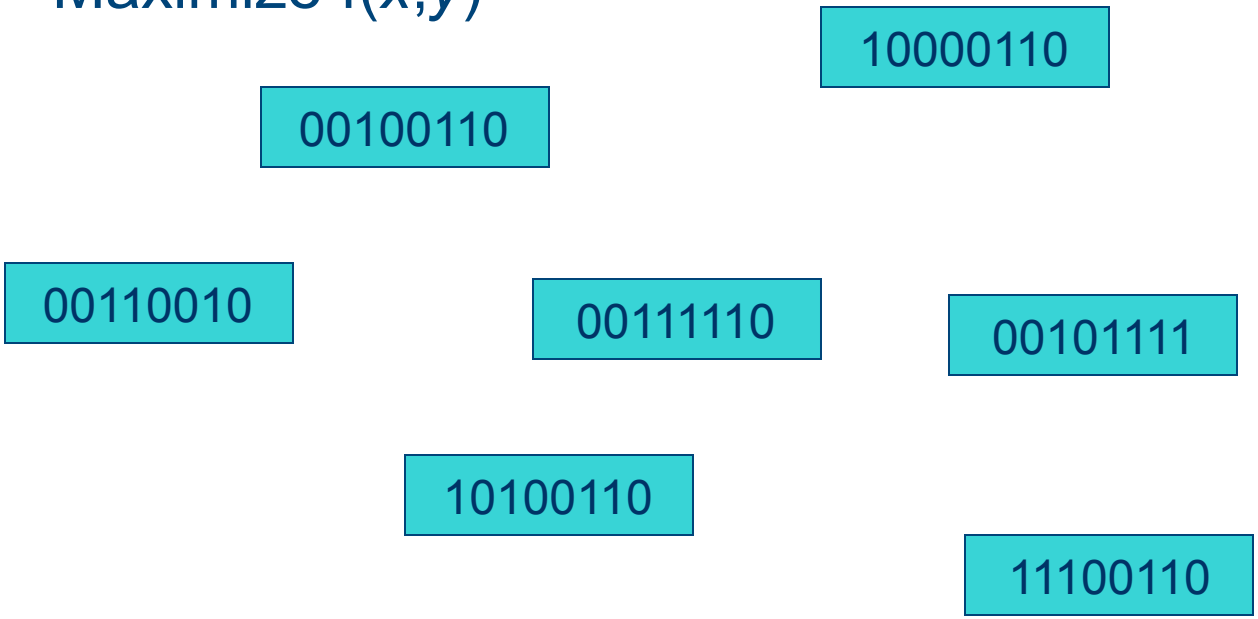
Basic GA

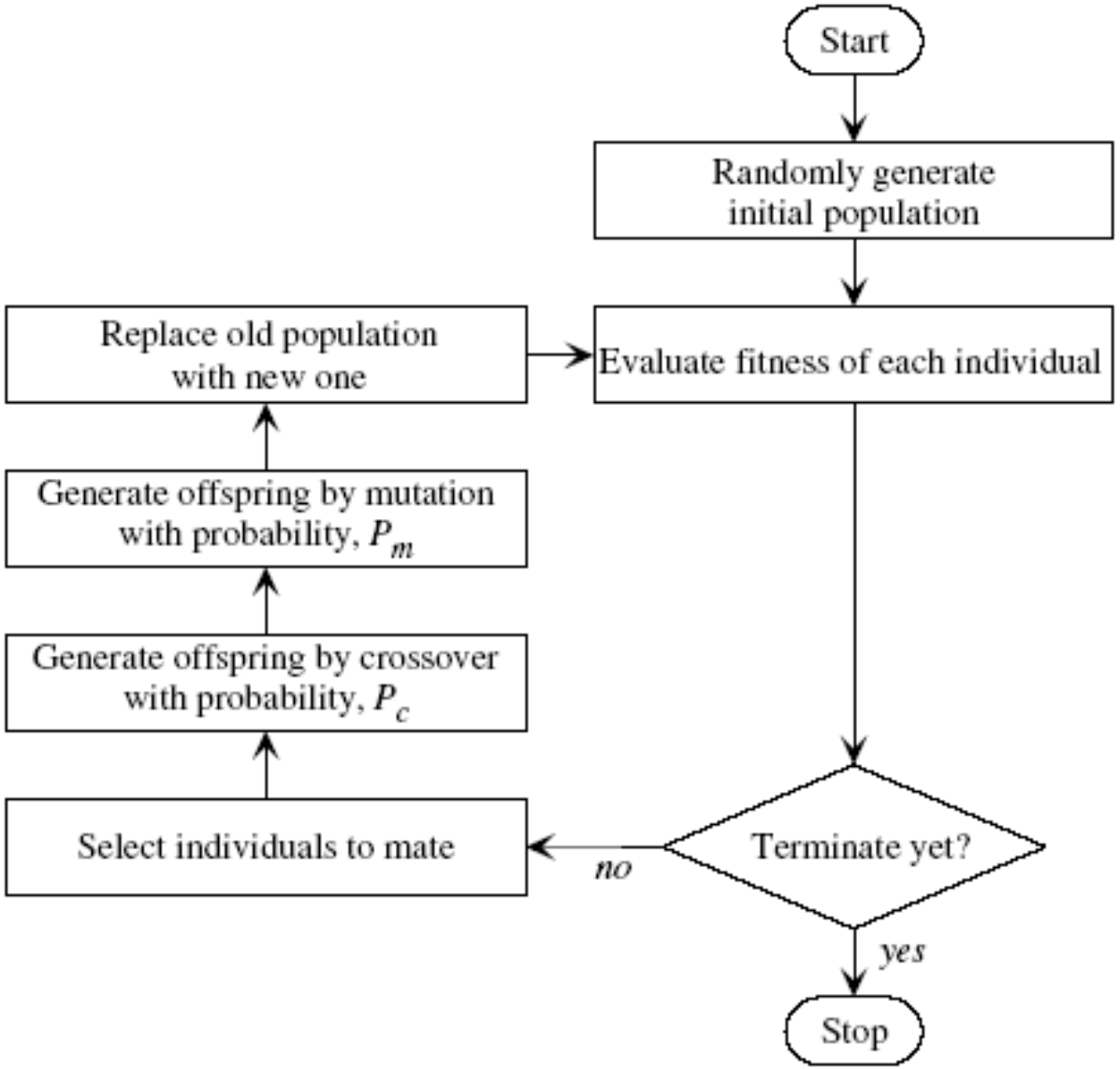
- Make thing “genetic”
  - Maximize f(*x*,*y*)

00100110

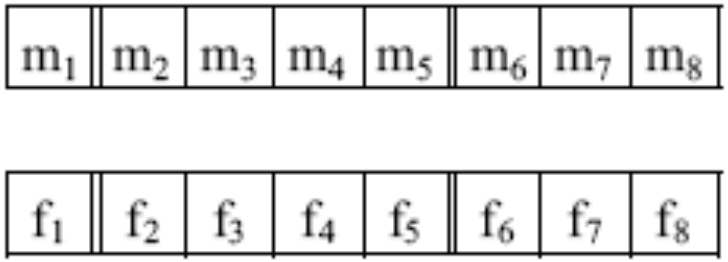
Basic GA

- Make thing “genetic”
  - Maximize f(*x*,*y*)

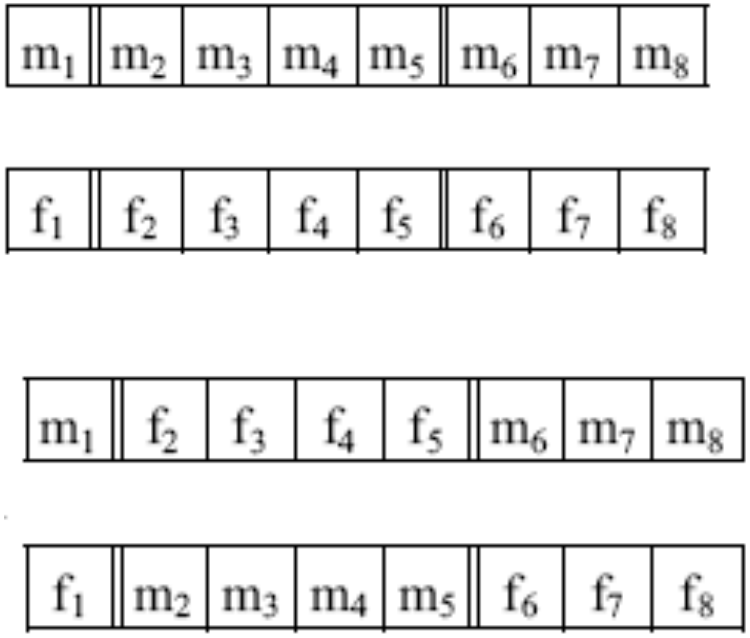



Start
Randomly generate initial population
Evaluate fitness of each individual
Terminate yet?
no
yes
Select individuals to mate
Generate offspring by crossover with probability, P_c
Generate offspring by mutation with probability, P_m
Replace old population with new one
Stop

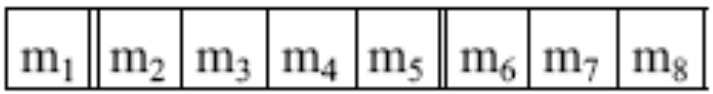
Crossover



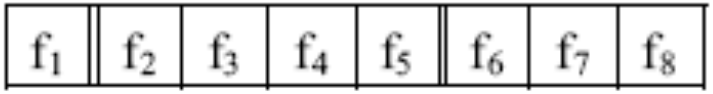
Crossover


m1 m2 m3 m4 m5 m6 m7 m8
f1 f2 f3 f4 f5 f6 f7 f8
m1 f2 f3 f4 f5 m6 m7 m8
f1 m2 m3 m4 m5 f6 f7 f8

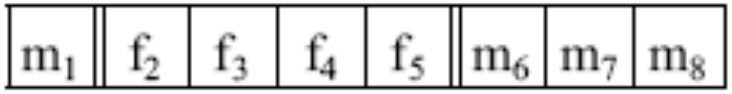
Crossover



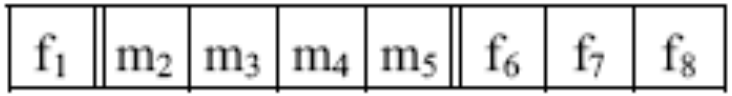
10000110



00100010



00000010



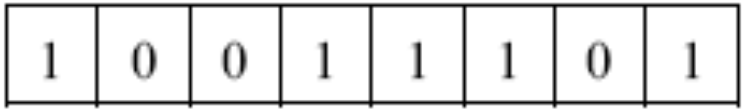
10100110

Mutation

- Altering the value of one or more loci
- Based on a probability P_m

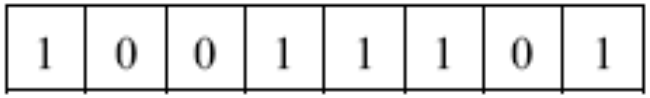
Validity check

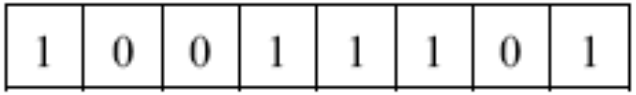
- A = 01, B = 10, C = 11
- BACA



Validity check

- A = 01, B = 10, C = 11
- BACA





Selection

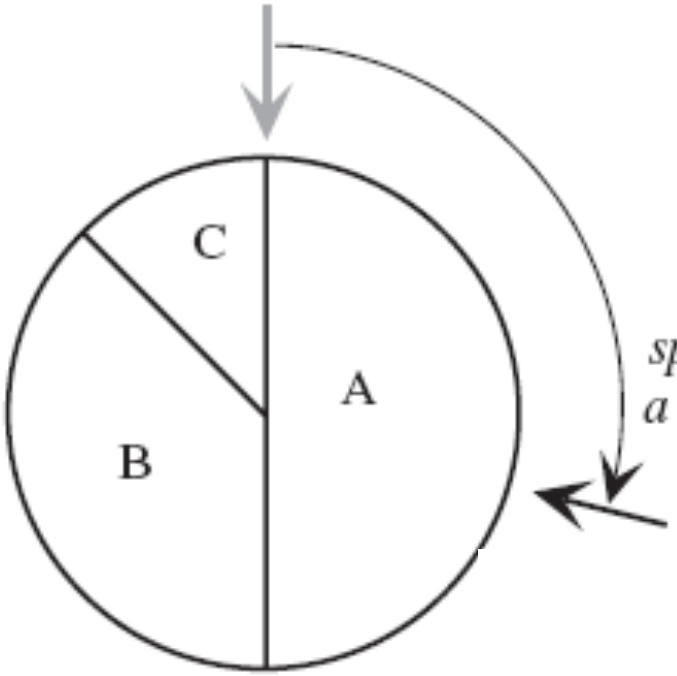
- Selection methods:
  - Strong
  - Week

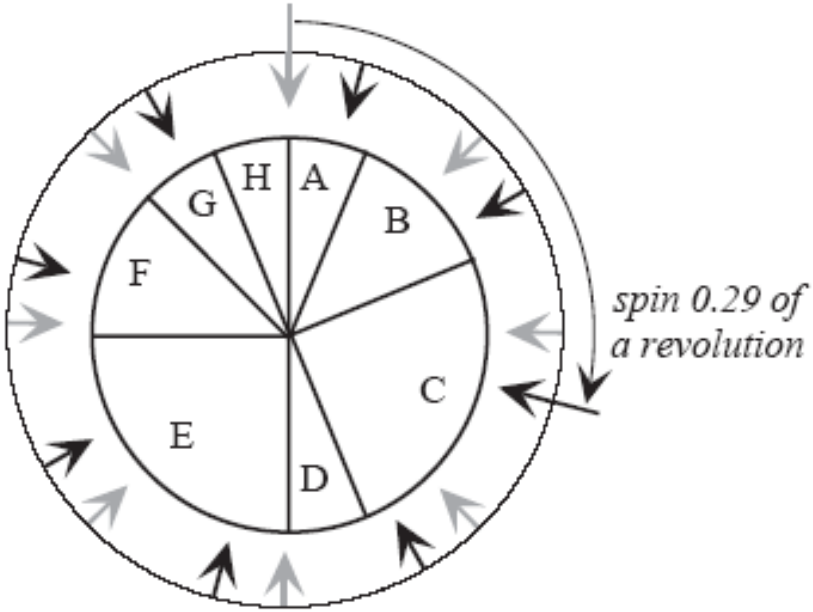
Selection

- Selection methods:
  - Too Strong: premature convergence
  - Too Week: slow evolution

Selection methods

- Roulette wheel selection
- Linear fitting scaling
- Boltzman fitness scaling
- Rank selection


C
A
B
spin 0.29 of
a revolution


H
A
G
B
F
C
E
D
spin 0.29 of
a revolution