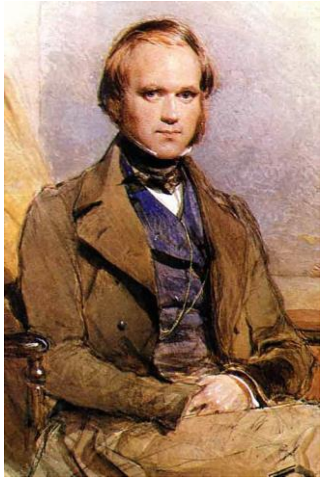
Genetic Algorithm (1)

Winter 2024

Background

- Genetic algorithms (GA) is an optimization technique that simulates the phenomenon of natural evolution.
- Species search for increasingly beneficial adaption for survival within the complicated and changing environment.
- Search takes place in species' chromosomes where changes are graded by the survival and reproduction.

Survival of the Fittest



Charles Darwin
(1809-1882)

- Survival and passing on of the beneficial characteristics to future generations
- Natural selection:
  - favorable heritable traits become more common in successive generations of a population of reproducing organisms
  - Faster rabbits are less likely been eaten by foxes; yet, slower rabbits may still survive.
  - Foxes also evolve. Otherwise, rabbits might become too fast for the foxes to catch.

Natural Selection

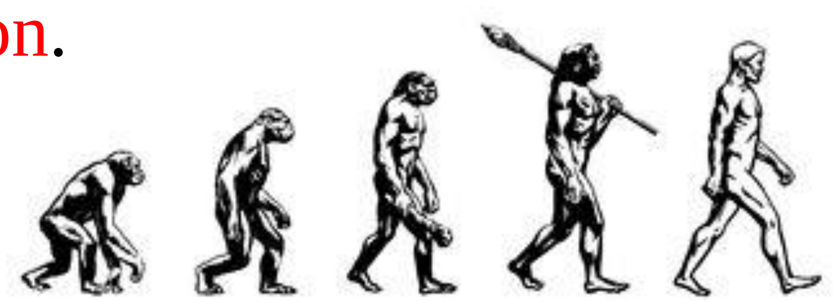
Moltrecht's tree frog

Arctic fox

Peacock

Evolution

- Most organisms evolve by means of two primary processes: natural selection and sexual reproduction.
- Natural Selection determines which members of population survive and reproduce
- Sexual Reproduction ensures mixing and recombination among the genes of their offspring, with infrequent Mutation.



GA History

- John Holland (1975) “Adaptation in Natural and Artificial Systems”

Computer programs that "evolve" in ways that resemble natural selection can solve complex problems even their creators do not fully understand

- David Goldberg (1989) “Genetic Algorithms in Search, Optimization, and Machine Learning”

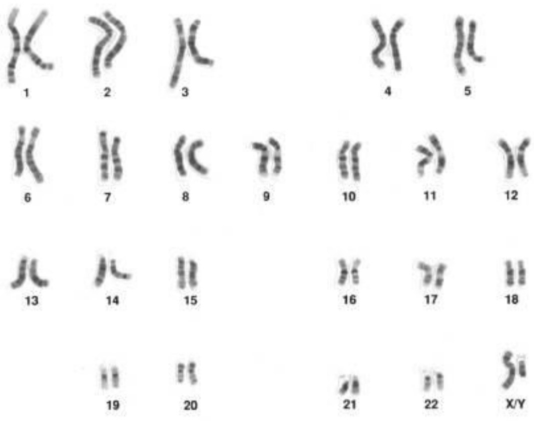
Computer-aided gas pipeline operation using genetic algorithms and rule learning, PhD thesis. University of Michigan. Ann Arbor, MI.


M
NATIONAL
CHAMPIONS

GA software

- MATLAB "Global optimization toolbox" http://www.mathworks.com/products/global-optimization/
- EXCEL Solver: Evolutionary https://bit.ly/336EpMJ
- Lumivero Evolver, an EXCEL add-in https://lumivero.com/products/decision-tools/evolver/
- PyGAD in Python https://pygad.readthedocs.io/en/latest/

Terminology



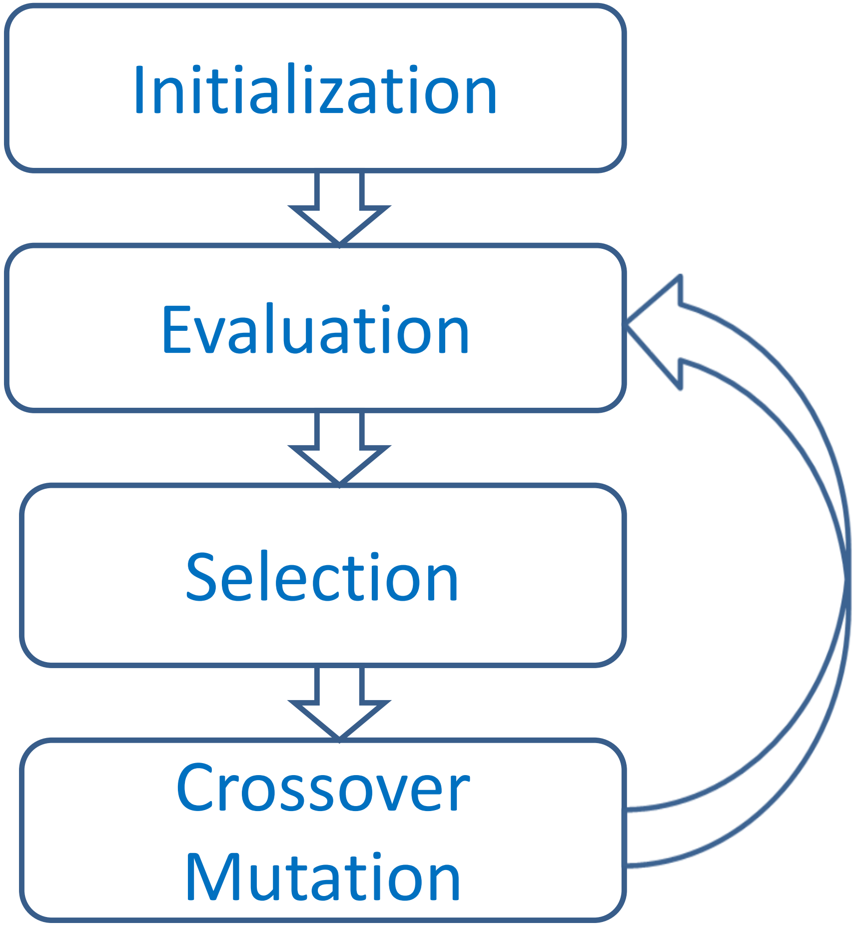
X/Y
Nettie Stevens
1861-1912

- Individual (***chromosome***, *genotype*, *string*) in population for different *generations*
- Chromosome is composed of ***genes*** (*features*, *characters*, *decoders*)
- Genes are located at certain places of the chromosome, which is called ***loci*** (string positions; plural of *locus*).
- Genes may be in different states, called ***alleles*** (feature values)
- The fittest survives (***fitness value*** = *objective value*)

Underlying concepts

- Each chromosome represents a potential solution to a problem
- An evolution process run on a population of chromosomes corresponds to a search through potential solutions
- Balancing two goals:
  - **Explore** the search space
  - **Exploit** the best solutions

GA high-level flowchart



GA structure

Program begins

t ← 0

initialize P(*t*)

evaluate P(*t*)

while (**not** termination-condition) **do**

t ← *t*+1

select P(*t*) from P(*t*-1)

reproduce P(*t*)

evaluate P(*t*)

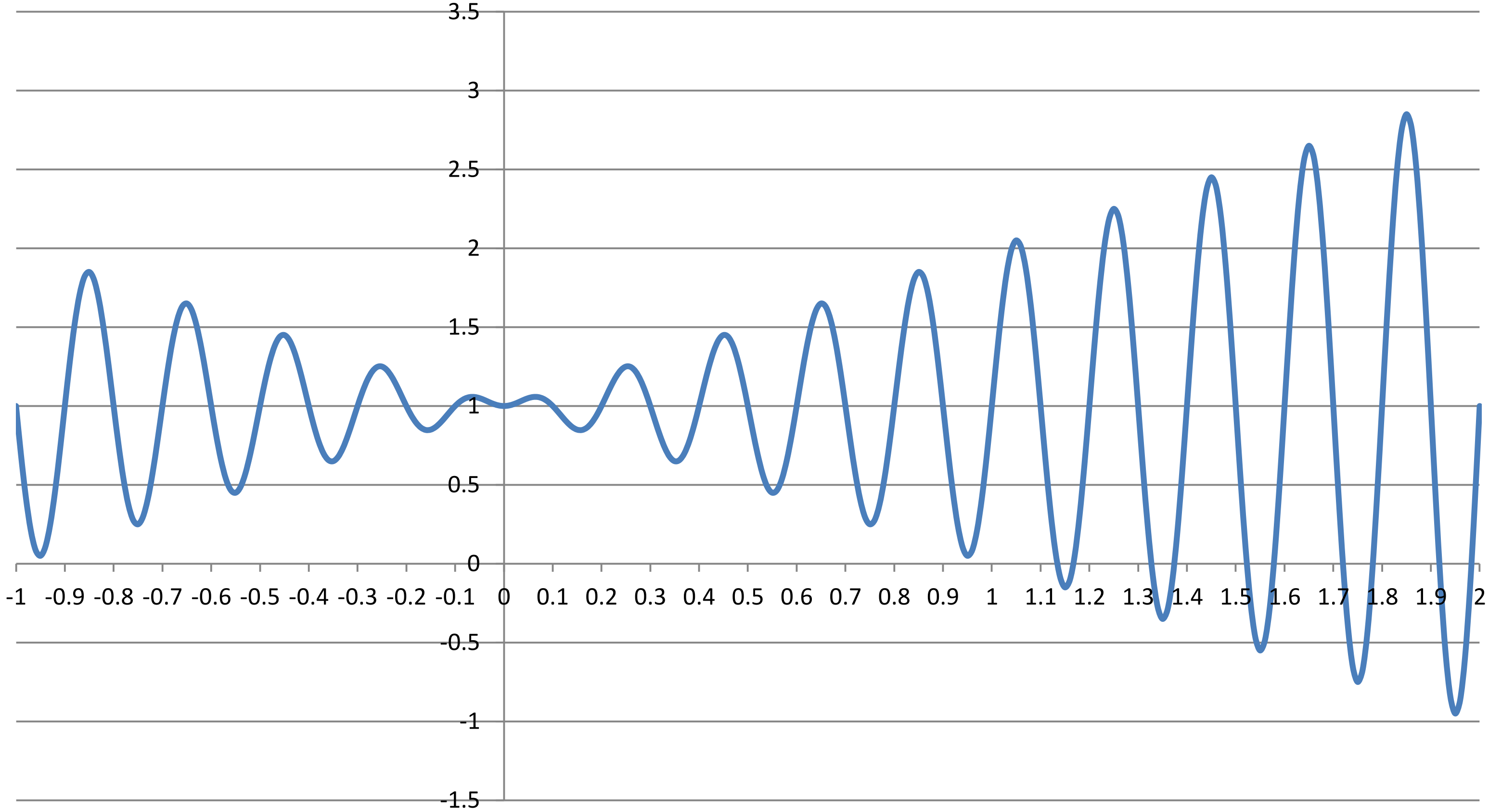
end

end

Crossover
Mutation

Numerical example

Maximize $f(x) = x \sin(10\pi x)+1; \ -1 \leqq x \leqq 2$



Analytical solution

- Find the zeros of the first derivative f'

  f'(x) = sin(10 πx)+ 10 πx*cos(10 πx) =0

- Multiple solutions

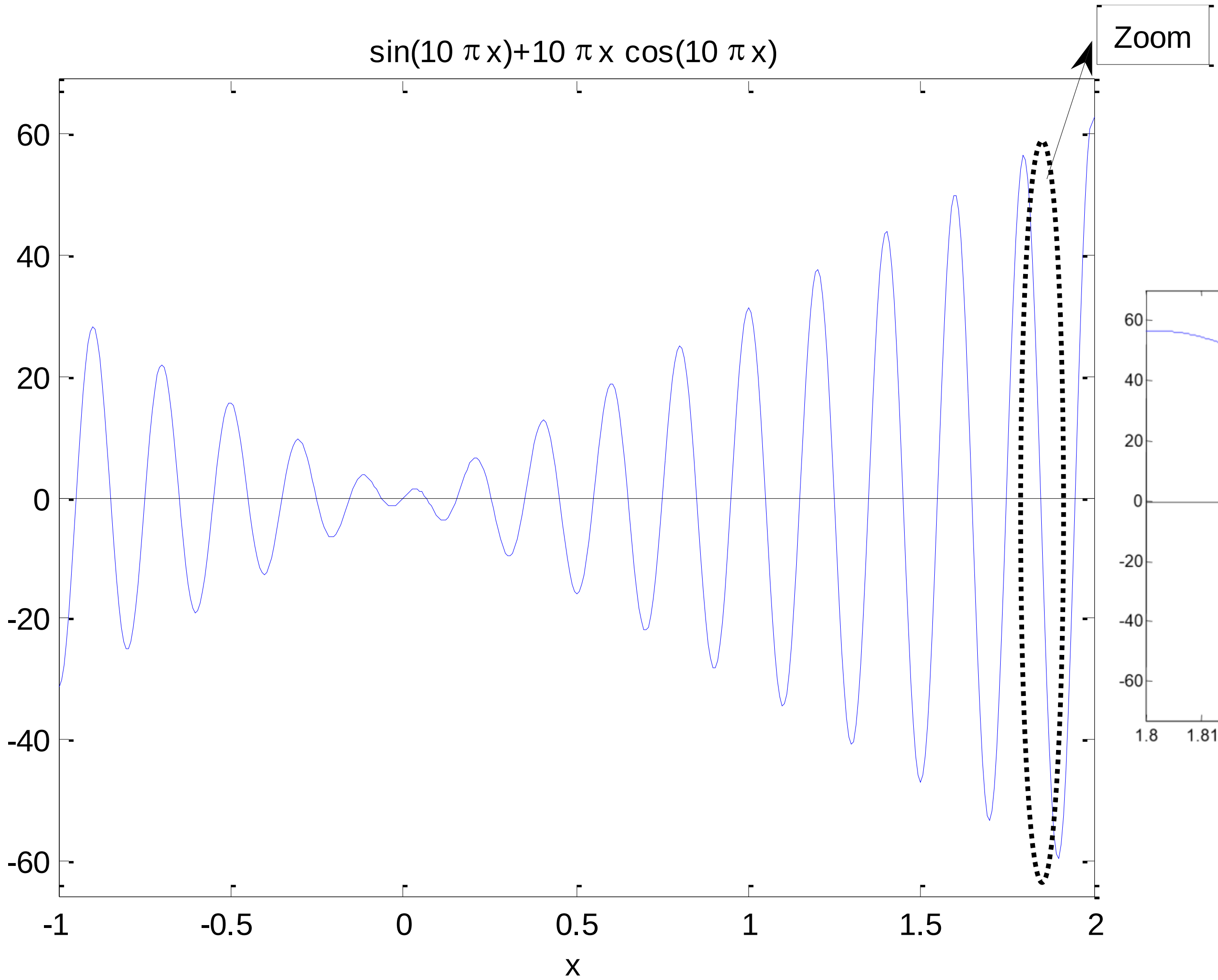
$$x_i = (2i-1)/20 + \varepsilon_i; \text{ for } i = 1,2,\ldots$$

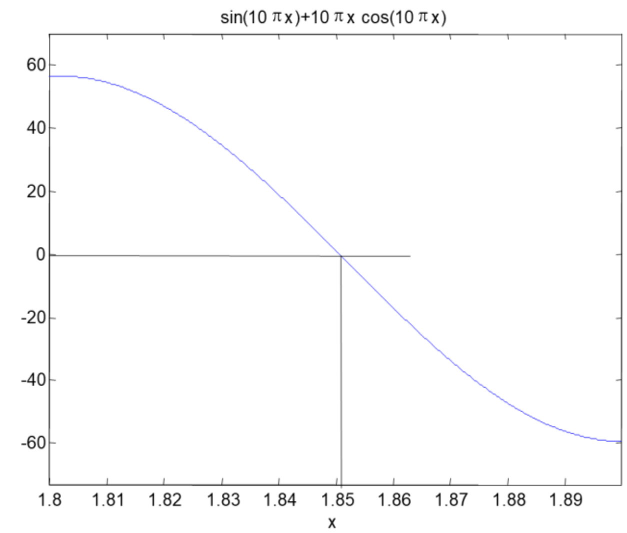
$$x_0 = 0$$

$$x_i = (2i+1)/20 - \varepsilon_i; \text{ for } i = -1,-2,\ldots$$

Local maxima: *i* is odd integer
Local minima: *i* is even integer

Closer look

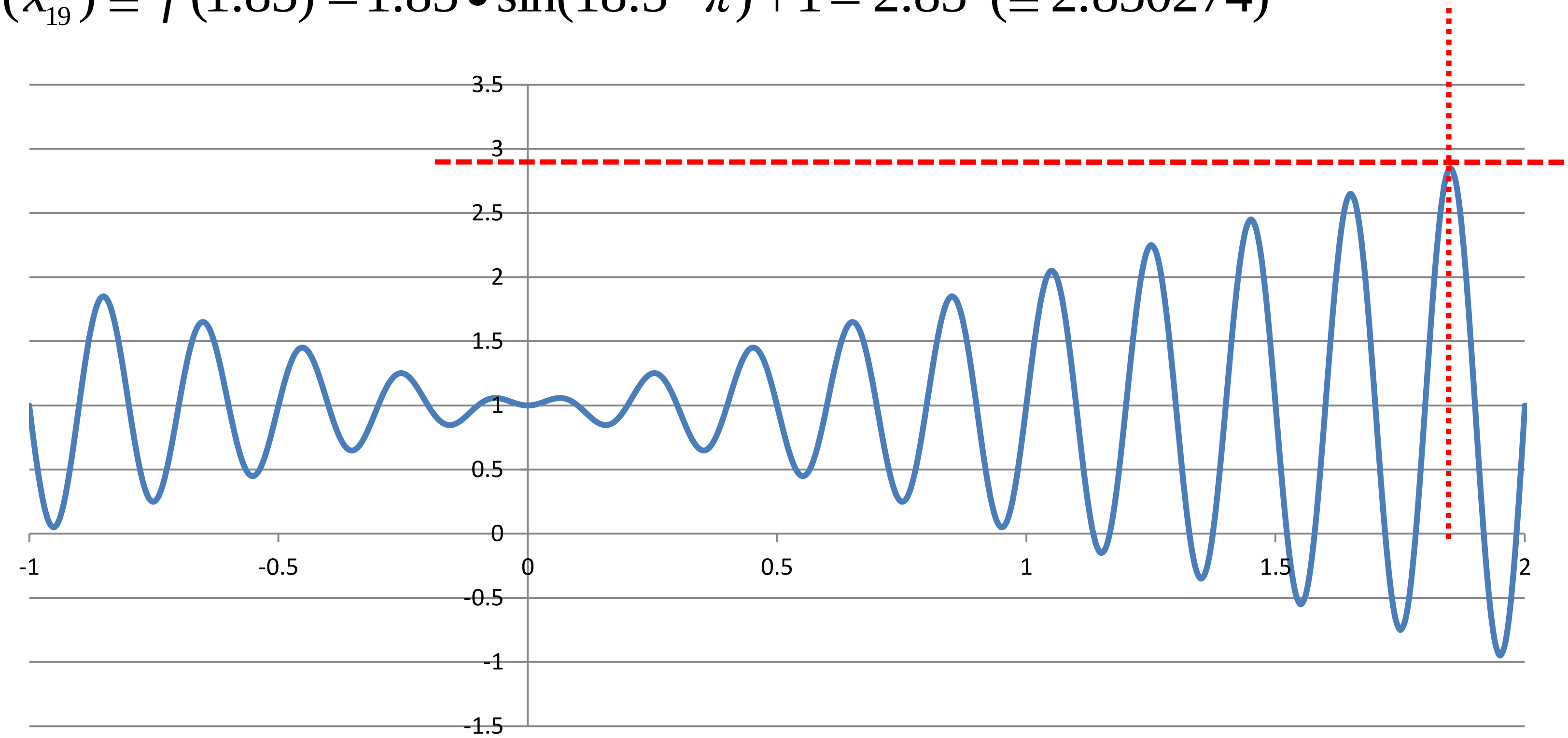

sin(10 π x)+10 π x cos(10 π x)
Zoom
60
40
20
0
-20
-40
-60
-1
-0.5
0
0.5
1
1.5
2
x


sin(10 π x)+10 π x cos(10 π x)
60
40
20
0
-20
-40
-60
1.8
1.81
1.82
1.83
1.84
1.85
1.86
1.87
1.88
1.89
x

Global maxima

$$x_{19} = (2*19-1)/20 + \varepsilon_{19} = 1.85 + \varepsilon_{19} \quad (\cong 1.850542)$$

$$f(x_{19}) \cong f(1.85) = 1.85 \bullet \sin(18.5*\pi) + 1 = 2.85 \quad (\cong 2.850274)$$



Chromosome representation

- Use of a binary vector to represent variable *x*
- The length of the vector depends on the required precision, say 6 places after the decimal point.
- The domain length is 3 (-1.0~2.0); thus, divide the range [-1,2] to $3 \cdot 10^{6}$ equal size ranges.
- 22 bits are needed because

$$2097152 = 2^{21} < 3000000 < 2^{22} = 4194304$$

Mapping from binary to real-value

Maximize f(x) = x sin(10 πx)+1; -1≦x≦2

- Convert the binary string from base 2 to 10

$$< b_{21}b_{20\ldots}b_0 >_2 = (\sum_{i=0}^{21} b_i \bullet 2^i)_{10} = x'$$

- Find a corresponding real number *x*

$$x = -1.0 + x' \frac{\boxed{3}}{2^{\boxed{22}} - 1}$$

domain length is 3

22 bits

$$\frac{x' - 0}{(2^{22} - 1) - 0} = \frac{x - (-1)}{2 - (-1)}$$

x’ range: (0 ~ 2^{22}-1)

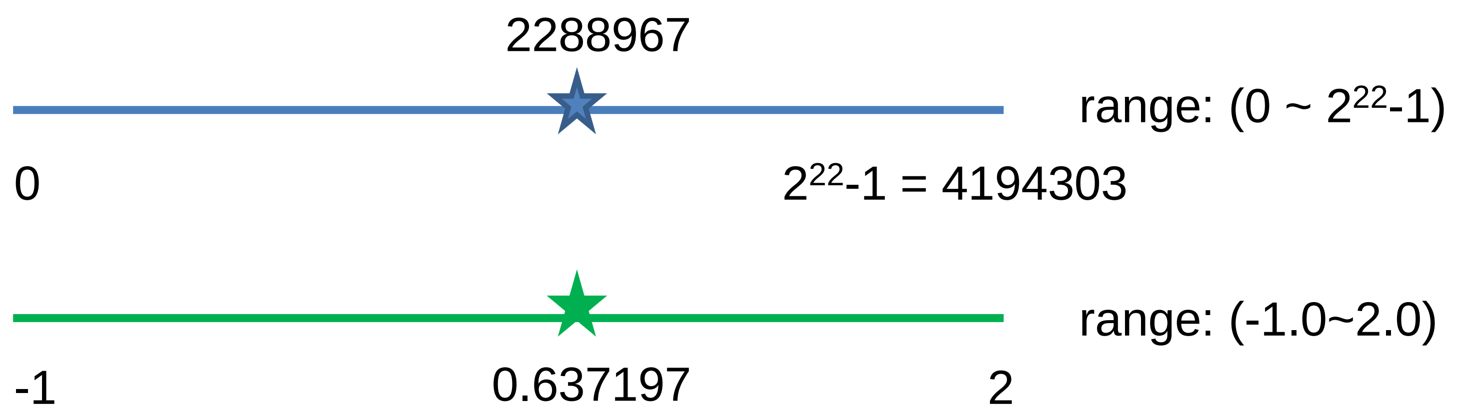
x range: (-1.0~2.0)

Mapping example

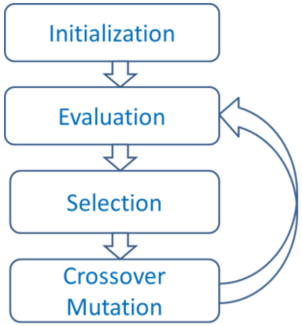
<1000101110110101000111>

x'=2288967 range: (0 ~ 2^{22}-1)

x=0.637197 range: (-1.0~2.0)



GA: Initialization



- Create a population of chromosomes, binary vectors of 22 bits
- Initiate the vectors randomly
- Example

for i=1:popluation_size

  for k=1:length_chromosome

    $v_i(k)$=round(U(0,1))

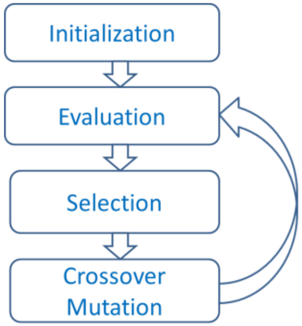
  end

end

k (columns), i (rows):

0	1	0	…	0
1	1	0		1
1	0	0		1
0	1	1		0
1	0	0		0

GA: Evaluation



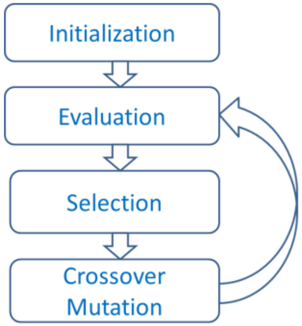
- Evaluation, <u>in this case</u>, is equivalent to the function f(*x*)

$$v_1 = (1000101110110101000111),$$
$$v_2 = (0000001110000000010000),$$
$$v_3 = (1110000000111111000101),$$

$$x_1 = 0.637197, \; x_2 = -0.958973, \text{ and } x_3 = 1.627888,$$

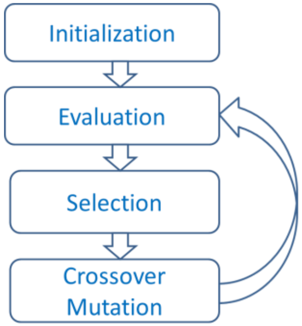
$$f(x_1) = 1.586345,$$
$$f(x_2) = 0.078878,$$
$$f(x_3) = 2.250650.$$

GA: Selection



- Select parent chromosomes for genetic operations
- Original approach:
  - Parents are selected stochastically (survival of the fittest)
  - New generation replaces their parents

GA: Crossover



- Crossover: mate two chromosomes to produce the next generation

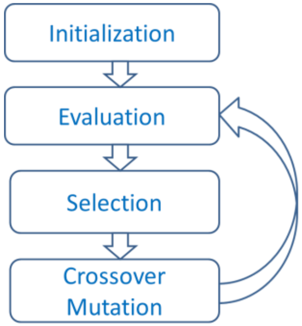
Randomly select the crossover point after the 5th gene

$$\boldsymbol{v}_2 = (00000|01110000000010000), \qquad \boldsymbol{v}_2' = (00000|00000111111000101),$$
$$\boldsymbol{v}_3 = (11100|00000111111000101). \qquad \boldsymbol{v}_3' = (11100|01110000000010000).$$

<u>Swap the second parts of chromosomes</u>

$$f(x_2) = 0.078878, \quad f(x_3) = 2.250650. \quad \Rightarrow \quad f(\boldsymbol{v}_2') = f(-0.998113) = 0.940865, \quad f(\boldsymbol{v}_3') = f(1.666028) = \underline{2.459245}.$$

GA: Mutation



- Mutation: alter one or more genes with the probability equal to the <u>**mutation rate**</u>
- For example

$\boldsymbol{v}_3 = (1110000000111111000101)$, $x_3 = 1.627888$; $f(x_3) = 2.250650$

$\boldsymbol{v}_3{}^* = (1110\cancel{0}00000111111000101)$, $x_3{}^* = 1.721638$; $f(x_3{}^*) = -0.082257$

1

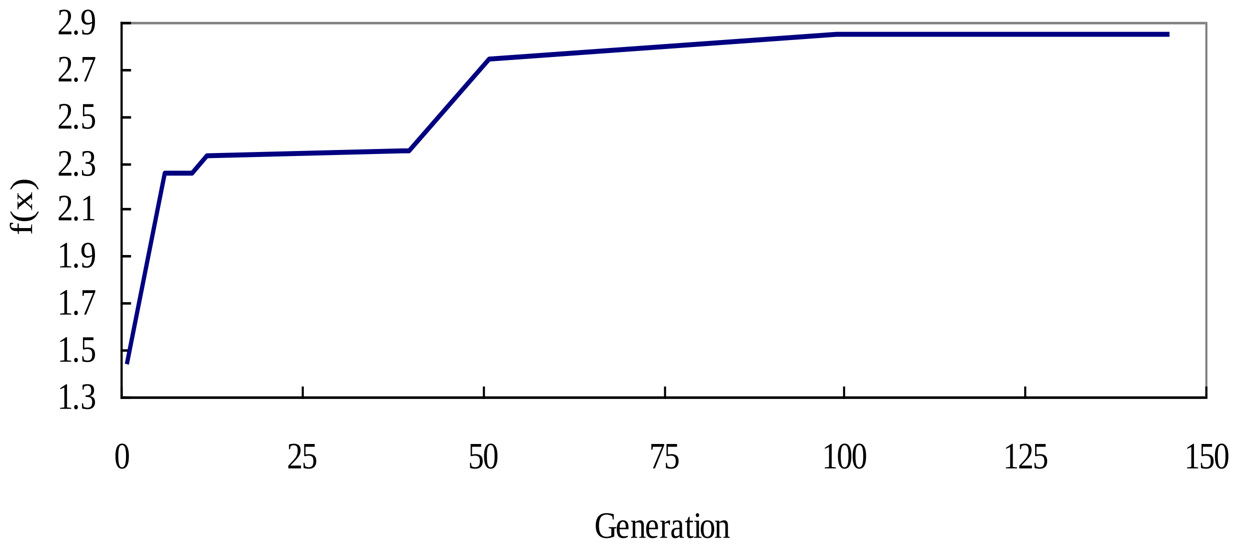
- <u>Mutation may or may not lead to a better solution</u>

GA: Parameters

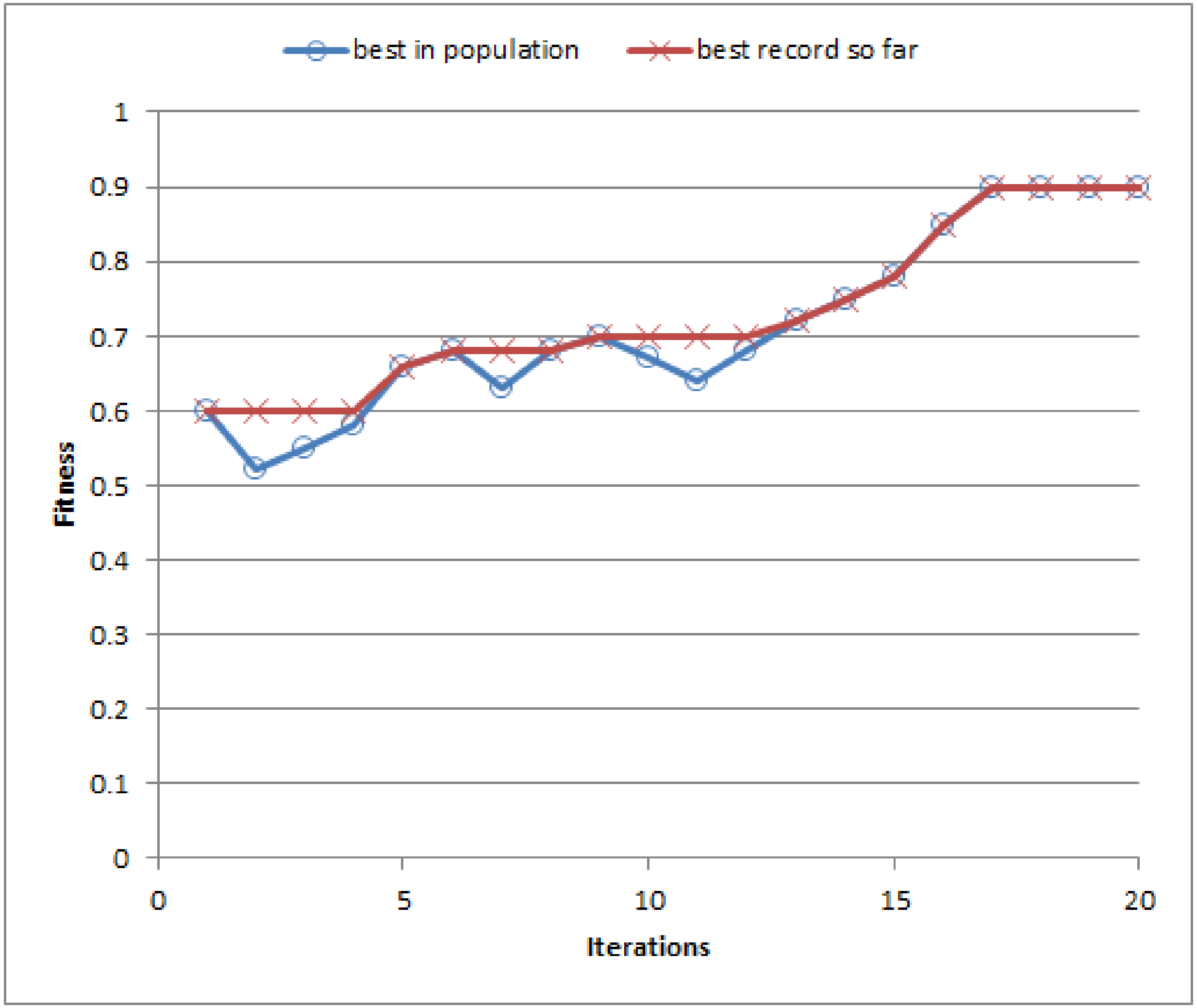
- Population size
  - Usually more than 20, e.g., 100
- Probability of crossover (crossover rate)
  - Usually high, e.g., 0.8
- Probability of mutation (mutation rate)
  - Usually low, e.g., 0.1

Numerical example
Experimental results

Generation number	Evaluation function
1	1.441942
6	2.250003
8	2.250283
9	2.250284
10	2.250363
12	2.328077
39	2.344251
40	2.345087
51	2.738930
99	2.849246
137	2.850217
145	2.850227



Convergence history



Other types of chromosomes

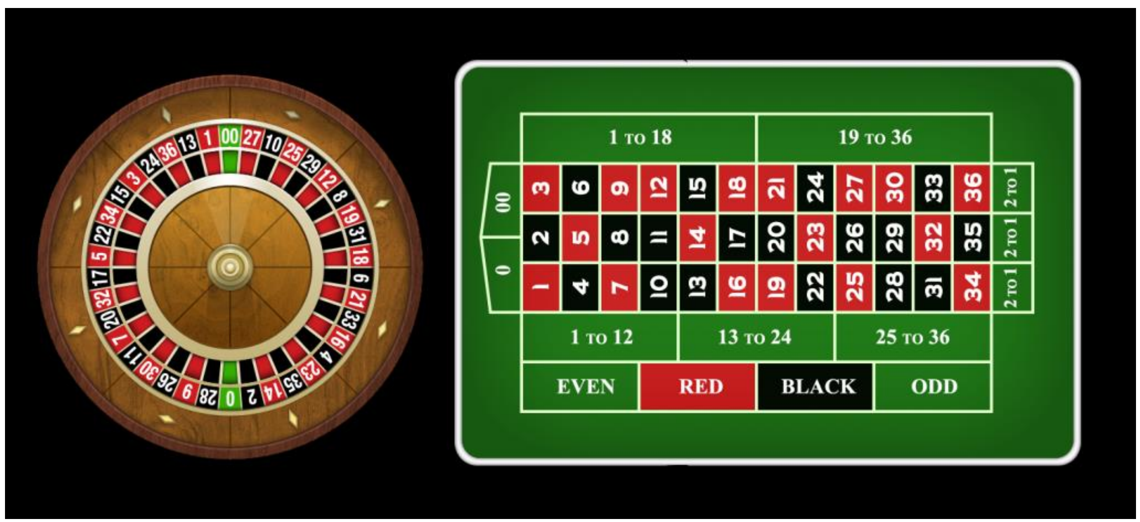
- TSP: [1 3 5 6 7 2 4]
- Knapsack: [0 1 0 0 1]
- Assignment: [1 0 0 | 0 0 1 | 0 1 0]
- VRP: [2 5 | 4 1 3 6]

Discussion: Population size

- Small population: may undercover the solution space
- Large population: incur severe computational penalties, i.e., longer time

Discussion: Selection (1)

- Roulette wheel
  - Selection of parents in accordance with the probability distribution of the fitness values



Discussion: Selection (2)

- Selection procedure (maximization problem)
  1. Calculate the fitness value $eval(v_i)$, i=1, 2,… *pop_size*
  2. Find the total fitness of the population
     $F = \Sigma eval(v_i)$
  3. Calculate the probability of a selection p_i for each chromosome; $p_i = eval(v_i) / F$
  4. Calculate a cumulative probability q_i for each chromosome; $q_i = \Sigma p_i$
  5. Generate a random number $r \sim U(0,1)$
  6. Select the ith chromosome such that $q_{i-1} < r \le q_i$

Selection example

Chrom. #	Fit.	Portion (Normalization)	Cumulative probability
# 1	8	8/50=0.16	0~0.16
# 2	6	6/50=0.12	0.16~0.28
# 3	12	12/50=0.24	0.28~0.52
# 4	20	20/50=0.4	0.52~0.92
# 5	4	4/50=0.08	0.92~1.0

Round 1
Random number=0.712
Pick #4 as Parent 1
Random number=0.328
Pick #3 as Parent 2

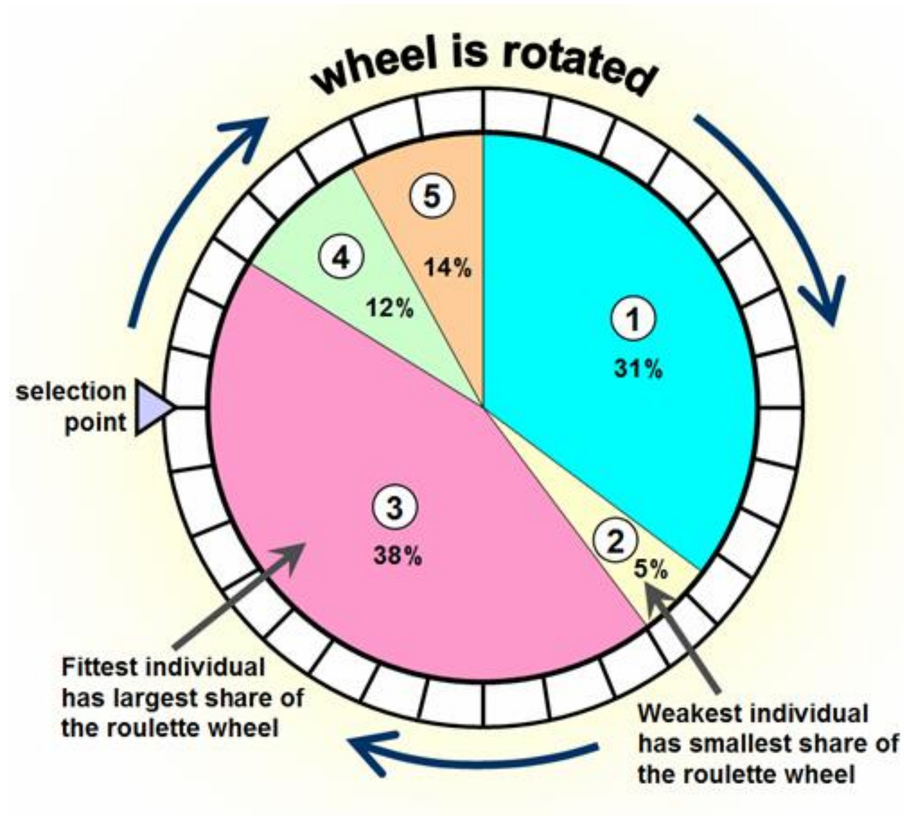
Go on to crossover

Do not pick identical parents
need to check after selection
may use "pick without replacement"

What to do if it is a minimization problem?

Discussion: Selection (3)

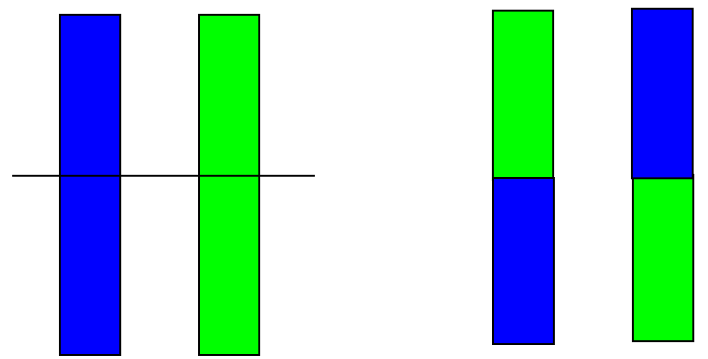
- Schema Theorem
  - The best chromosomes get more copies
  - The average stay even
  - The worst die off



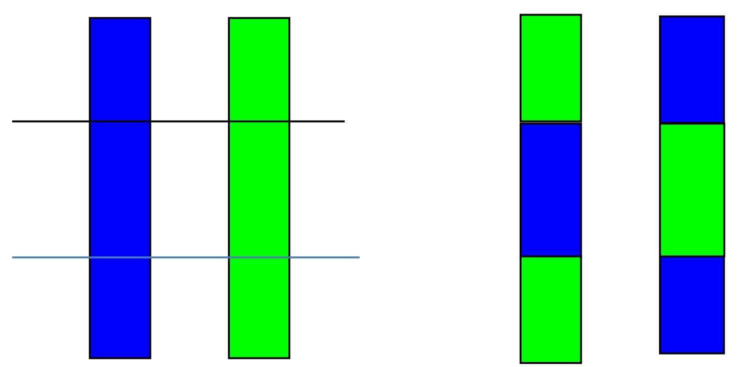
Source: Newcastle University

Discussion: Crossover

- When crossover rate is met, do crossover
- Various crossover mechanisms
  1. One-point



  2. Two-point

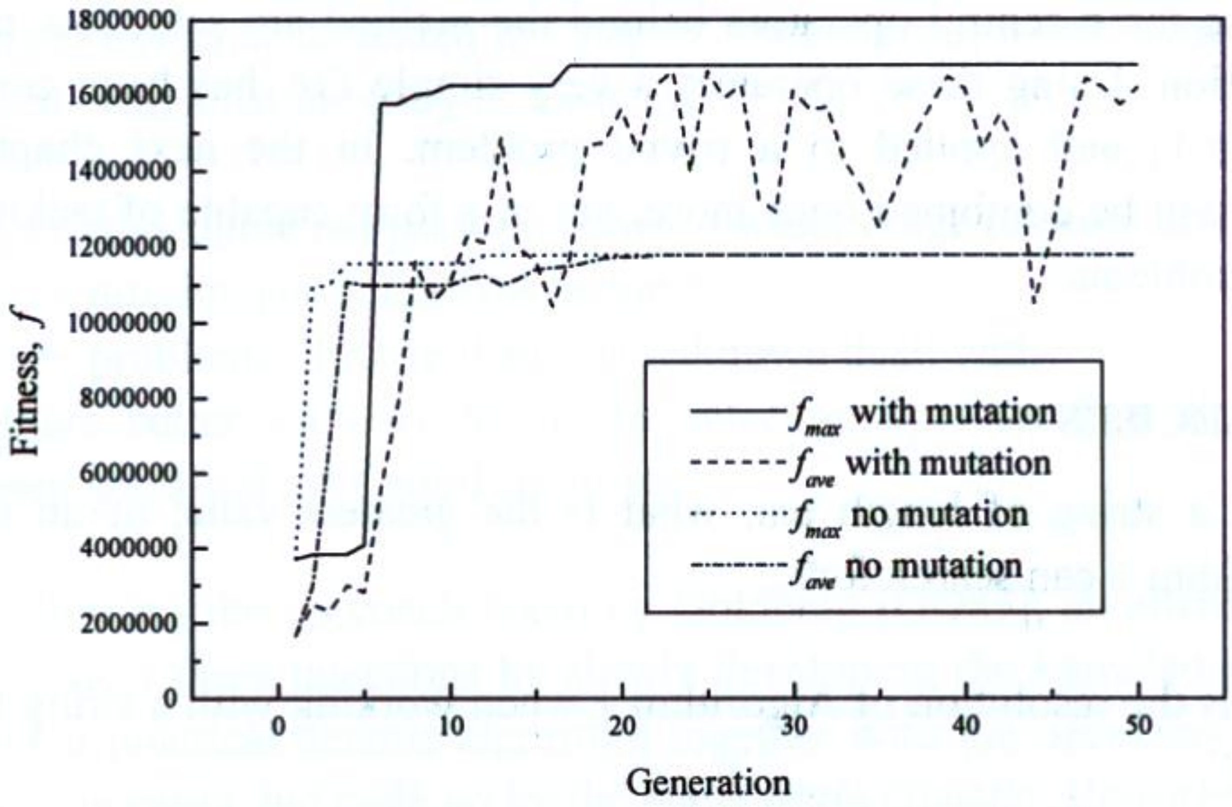


Discussion: Mutation

- Bit-by-bit base
- Probability of mutation, p_m
- Mutation procedure
  1. Generate a random number $r \sim U(0,1)$
  2. If $r < p_m$, mutate the bit

Effect of Mutation

Maxmize $f(x) = x^2; x \in \mathrm{I}; 0 \le x \le 4095$



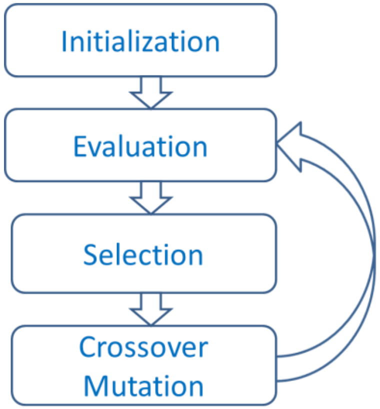
Note:
We are usually interested in the best (max in this case) and average fitness values

Complete Illustration

- We will illustrate GA through initiation, and then combine crossover and mutation

Numerical Example (1)

Max $f(x)=1/(x+1)$ $0 \le x \le 15$ $x \in \mathbf{Z}$



Step 1:

4-bit chromosomes

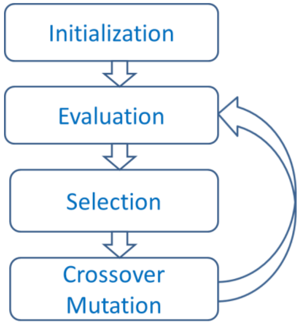
Population size=4 (control parameters)

CR=0.9; MR=0.2 (control parameters)

Step 2: random population; evaluation

	Chromosomes	x	f(x)
#1	[1 0 1 0]	x=10	1/11
#2	[0 0 1 1]	x=3	1/4
#3	[1 1 0 0]	x=12	1/13
#4	[0 1 1 0]	x=6	1/7

Numerical Example (2)



Step 3-1: choose first pair of parents

	Chromosomes	x	f(x)		Normalization	Cumulation
#1	[1 0 1 0]	x=10	1/11	0.09091	0.16214	0.1621
#2	[0 0 1 1]	x=3	1/4	0.25000	0.44588	0.6080
#3	[1 1 0 0]	x=12	1/13	0.07692	0.13719	0.7452
#4	[0 1 1 0]	x=6	1/7	0.14286	0.25479	1.0000
				0.56069	1.00000	

Generate random number z1=0.3867; **pick #2**

"pick without replacement"

	Chromosomes	x	f(x)		Normalization	Cumulation
#1	[1 0 1 0]	x=10	1/11	0.09091	0.29260	0.2926
#3	[1 1 0 0]	x=12	1/13	0.07692	0.24759	0.5402
#4	[0 1 1 0]	x=6	1/7	0.14286	0.45981	1.0000
				0.31069	1.00000	

Generate random number z2=0.8219; **pick #4**

Numerical Example (3)

Step 4-1: use single point crossover

#2 [0 0 1 1]

#4 [0 1 1 0]

Generate random number z3=0.3981

Because 0.3981 < CR=0.9; **Do crossover**

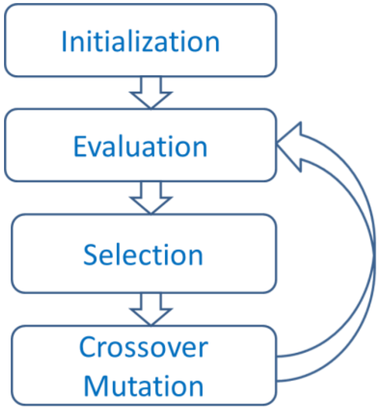
Generate random number z4=0.4517

Crossover location: ceiling[0.4517*3]=2

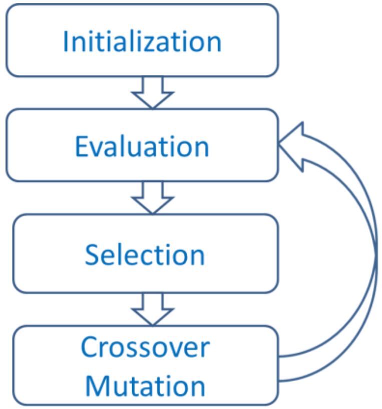
Generate two children

#1 [0 1 1 1]

#2 [0 0 1 0]



Numerical Example (4)



Step 5-1: mutation

#1 [0 1 1 1]

#2 [0 0 1 0]

Generate random number z5=0.8247

Because 0.8247 > MR=0.2; No mutation for #1

Generate random number z6=0.1296

Because 0.1296 < MR=0.2; **Do mutation for #2**

Generate random number z7=0.6183

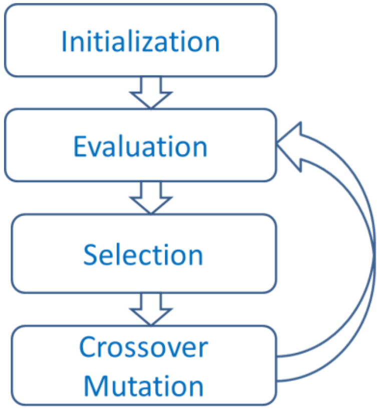
Mutation location: ceiling[0.6183*4]=3; **#2** becomes [0 0 0 0]

Numerical Example (5)

Step 3-2: choose second pair of parents

Step 4-2: crossover; produce #3 and #4

Step 5-2: mutation



Children chromosomes replace their parents

Return to Step 2

Test Functions

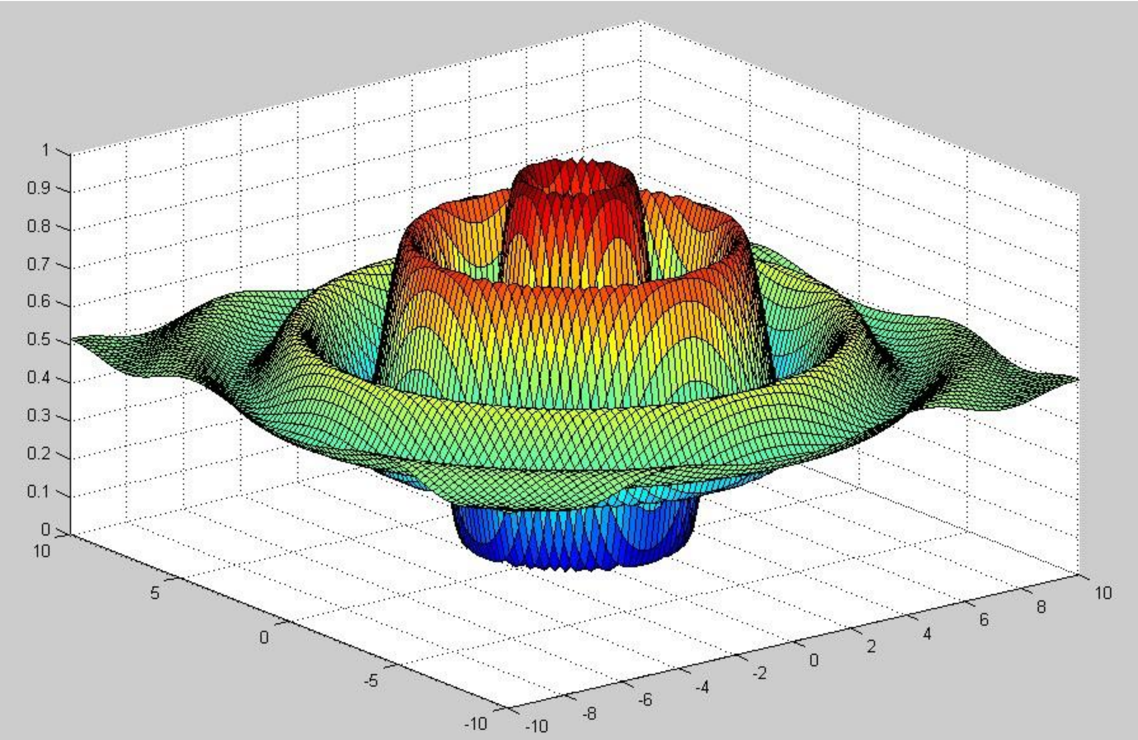
- Benchmark cases and datasets have been proposed before
- These multi-dimensional cases involve nonlinearity and oscillation around the optimal solutions
- So, there exists a high probability for each optimization technique to trap into local optima.
- Direct comparisons can be made among different algorithms

Benchmark Problems

- Benchmark instances can be found at
  - https://www.sfu.ca/~ssurjano/optimization.html
  - http://people.brunel.ac.uk/~mastjjb/jeb/info.html
  - TSP: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/
  - Project scheduling: http://www.om-db.wi.tum.de/psplib/getdata_mm.html

Case (1): Schaffer

$$f(x,y) = 0.5 + (\sin^2(\sqrt{(x^2+y^2)}) - 0.5)/((1+0.001(x^2+y^2))^2)$$



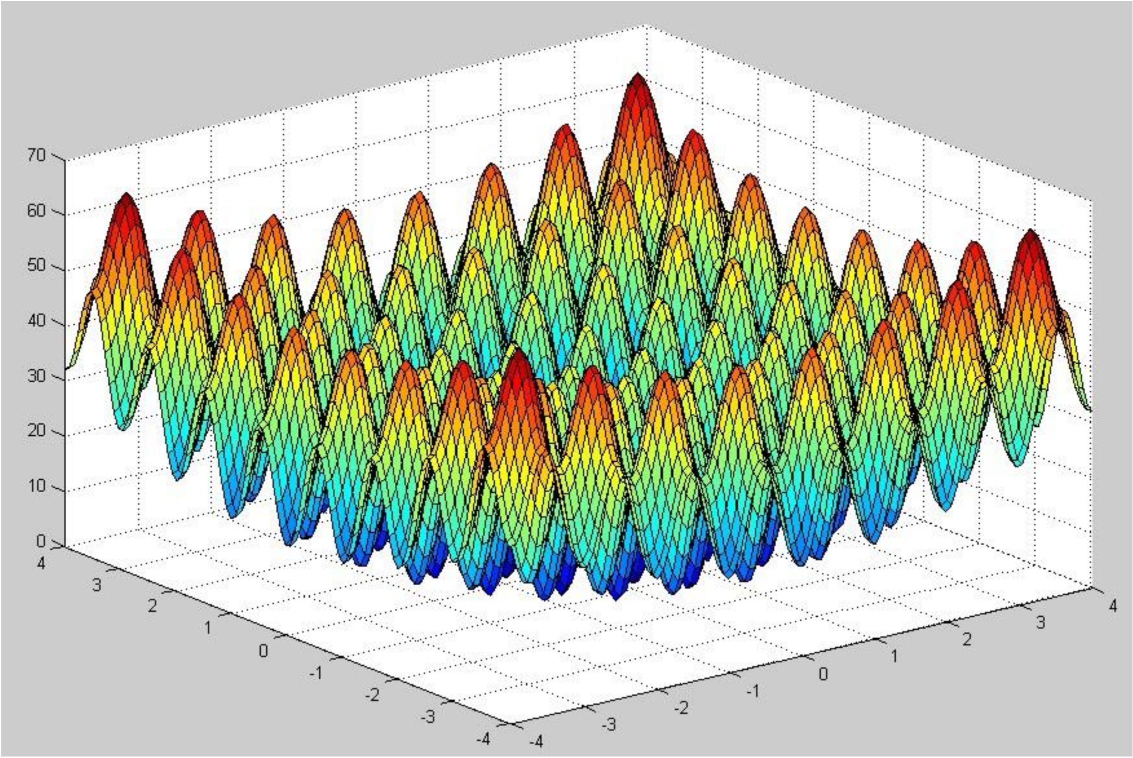
$-10 \leq x_i \leq 10$

Minimize f(x,y)=0

(x,y) = (0, 0)

Case (2): Rastrigin

$$f(x) = 10 \cdot D + \sum_{i=1}^{D} \left(x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i) \right),$$



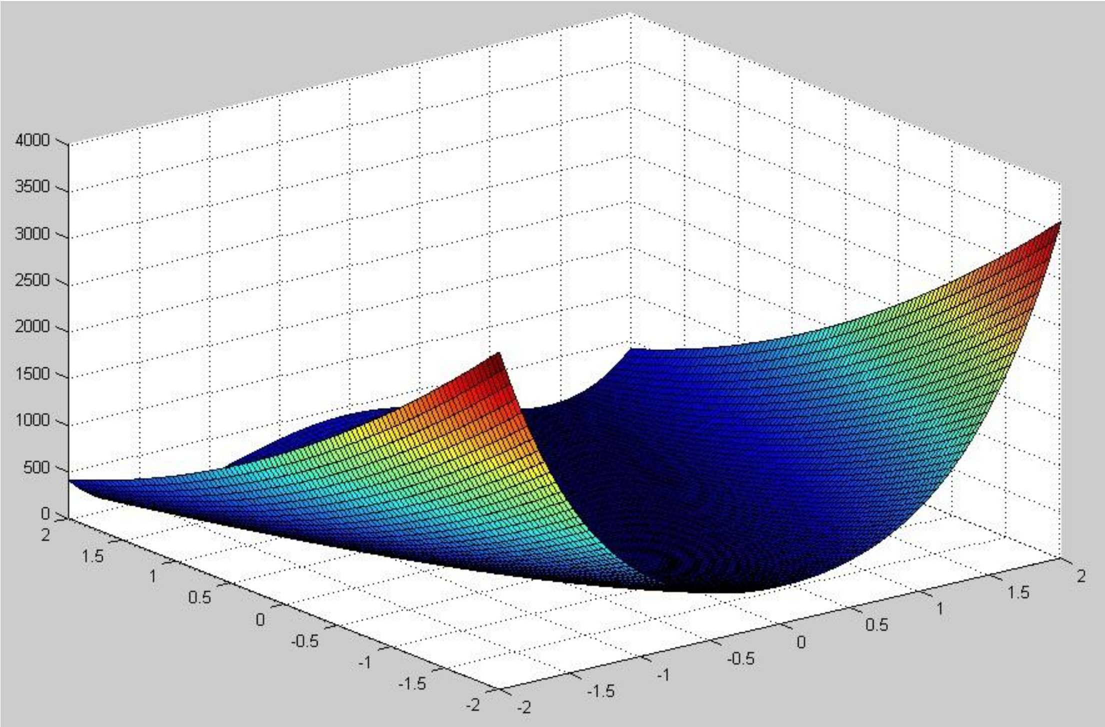
$-4 \le x_i \le 4$

Minimal f(x)=0

$x_{\mathrm{i}} = 0$

Case (3): Rosenbrock

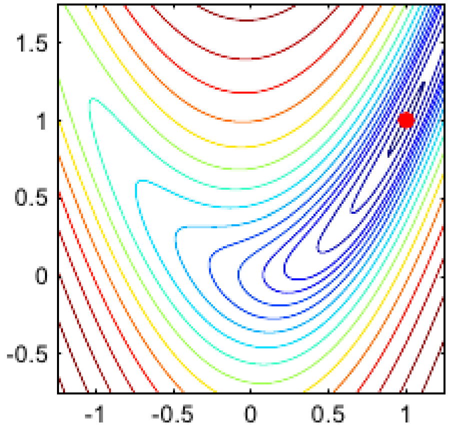
$$f(x)=\sum_{i=1}^{D-1} 100\cdot\left(x_{i+1}-x_i^2\right)^2+\left(1-x_i\right)^2,$$



$-2 \leq x_i \leq 2$

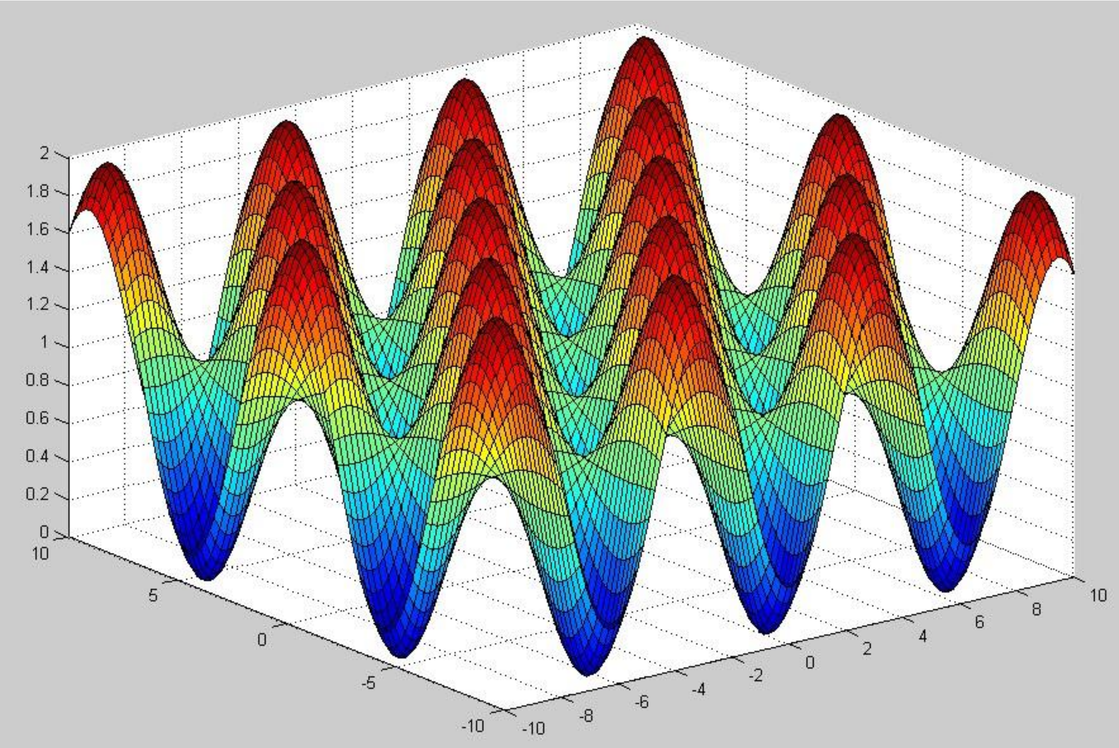
Minimal f(x)=0

$x_{\mathrm{i}} = 1$



Case (4): Griewank

$$f(x)=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1,$$



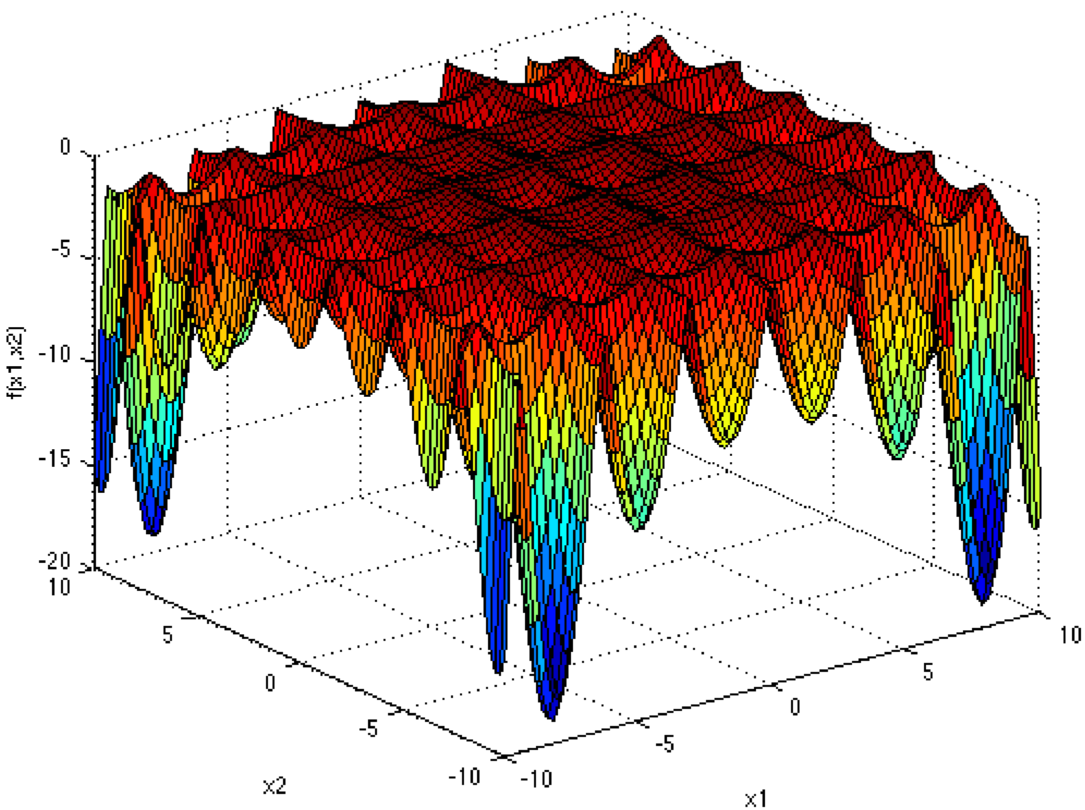
$-10 \le x_i \le 10$

Minimal f(x)=0

$x_{\mathrm{i}} = 0$

Case (5): Holder Table

$$f(\mathbf{x}) = -\left|\sin(x_1)\cos(x_2)\exp\left(\left|1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\right|\right)\right|$$



$-10 \leq x_i \leq 10$

Minimal $f(x)$=-19.21

$x = (\pm 8.055, \pm 9.664)$

Continue for GA (2)