Outline

Recap

- Search and optimization
- Applications of optimization
- Properties of optimization problems in practice
- Mathematical problem formulation or modeling
- Remarks on the numerical optimization techniques
- Introduction to evolutionary computation (EC)
- Principles of EC: Genetics, evolution and survival of the fittest
- Generalized framework
- Advantages and limitations of EC techniques
- Behavior of EC run
- No Free Lunch (NLP) Theorem in optimization

Generalized Framework of EC Techniques

Algorithm 1 Generalized Framework

Solution representation %Genetics
Input: $t := 1$ (Generation counter), Maximum allowed generation $= T$
Initialize random population $(P(t))$; %Parent population
Evaluate $(P(t))$; %Evaluate objective, constraints and assign fitness
while $t \leq T$ **do**
    $M(t) :=$ Selection$(P(t))$; %Survival of the fittest
    $Q(t) :=$ Variation$(M(t))$; %Crossover and mutation
    Evaluate $Q(t)$; %Offspring population
    $P(t+1) :=$ Survivor$(P(t), Q(t))$; %Survival of the fittest
    $t := t + 1$;
end while

Solution Representation

Decision variables for an optimization problem are represented using Boolean variables in binary-coded genetic algorithm.

- Binary variable: {0,1}
- Real variable
- Discrete and Integer variable

Real Variable (x_i)

- Suppose string length $(l) = 5$ is chosen for x_i
- Maximum value of binary string of $(l) = 5$, that is, $DV(s)\ of\ \{11111\} = b = 31$ and minimum value is $a = 0$
- Suppose lower bound is $x_i^{(L)} = 3$ and upper bound is $x_i^{(U)} = 10$

Real Variable Representation

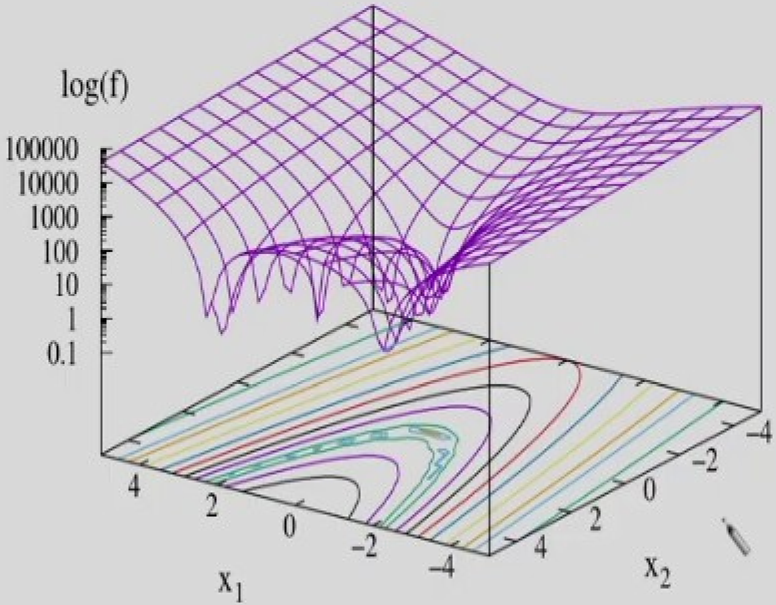
- Value of decision variable, $x_i = x_i^{(L)} + \frac{x_i^{(U)} - x_i^{(L)}}{2^l - 1} DV(s)$, $DV(s)$ is a decoded value of binary string.
- $x_i = 3 + \frac{7}{31} DV(s)$
- String is $= \{11011\}$
- Decoded value $DV(s) = (1 * 2^4) + (1 * 2^3) + (0 * 2^2) + (1 * 2^1) + (1 * 2^0) = 27$
- $x_i = 3 + \frac{7}{31} * 27 = 9.096$
- Precision $= 7/31 = 0.226$
  - The neighbor of 9.096 is $9.096 + 0.226 = 9.322$.
  - We cannot get any value of x_i between 9.096 and 9.322.
- If we want a precision of 0.01, then $7/(2^l - 1) = 0.01$
- $l = \log_2(1 + 7/0.01) = 9.453$ or 10.

Working Principles Through An Example

Rosenbrock Function

Minimize $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$,

bounds $-5 \leq x_1 \leq 5$ and $-4 \leq x_2 \leq 4$.



- Optimum solution is $x^* = (1, 1)^T$ and $f(x^*) = 0$

Working Principles: Initial Population

Solution Representation

- Let the chromosome string length is $l = 10$.
  - First five bits are used to represent x_1 and rest of them for x_2.

Generate initial random population

- Let the population size is $N = 8$.
- Let the first string is 01100 11010. The first five bits (01100) are used to represent x_1 and the remaining bits (11010) for x_2.

Index	Chromosomes	DV(x_1)	DV(x_2)
1	01100 11010	12	26
2	11000 01011	24	11
3	00110 00110	6	6
4	01000 10111	8	23
5	10100 11101	20	29
6	01101 01000	13	8
7	00101 11011	5	27
8	11100 11000	28	24

Working Principles: Initial Population

- The scaling formula is
$x_1 = x_1^{(L)} + \frac{x_1^{(U)} - x_1^{(L)}}{2^l - 1} DV(s)$
$= -5 + \frac{10}{2^5 - 1} DV(s)$.
- The decoded value of (01100) is
$DV(x_1) = 12$.
- $x_1 = -5 + \frac{10}{2^5 - 1} 12 = -1.129$

- The scaling formula is
$x_2 = x_2^{(L)} + \frac{x_2^{(U)} - x_2^{(L)}}{2^l - 1} DV(s)$
$= -4 + \frac{8}{2^5 - 1} DV(s)$.
- The decoded value of (11010) is
$DV(x_2) = 26$.
- $x_2 = -4 + \frac{8}{2^5 - 1} 26 = 2.710$

Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2
1	01100 11010	12	26	−1.129	2.710
2	11000 01011	24	11	2.742	−1.161
3	00110 00110	6	6	−3.065	−2.452
4	01000 10111	8	23	−2.419	1.935
5	10100 11101	20	29	1.452	3.484
6	01101 01000	13	8	−0.806	−1.935
7	00101 11011	5	27	−3.387	2.968
8	11100 11000	28	24	4.032	2.194

Working Principles: Evaluate Population

Solution 1

- Let solution 1 is represented as $\mathbf{x}^{(1)} = (-1.129, 2.710)^T$.
- Objective function value:

$$f(-1.129, 2.710) = 100(2.710 - (-1.129)^2)^2 + (1 - (-1.129))^2 = 210.445.$$

  - Let us choose the fitness value same as the function value.

Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2	$f(x_1, x_2)$
1	01100 11010	12	26	-1.129	2.710	210.445
2	11000 01011	24	11	2.742	-1.161	7536.407
3	00110 00110	6	6	-3.065	-2.452	14041.882
4	01000 10111	8	23	-2.419	1.935	1546.603
5	10100 11101	20	29	1.452	3.484	189.732
6	01101 01000	13	8	-0.806	-1.935	671.924
7	00101 11011	5	27	-3.387	2.968	7252.209
8	11100 11000	28	24	4.032	2.194	19793.183

Termination Condition

- BGA gets terminated when generation counter is more than the allowed maximum generations, that is, $(t > T)$.
- Some other criterion can also be considered for terminating the algorithm.
- Since it is the first generation, we continue and move to selection operator.

Working Principles: Selection

- The purpose is to identify good (usually above-average) solutions in the population.
- In this process, we eliminate bad solutions in the population.
- We make multiple copies of good solutions.
- Selection can be used either before or after search/variation operators.
  - When selection is used before search/variation operators, it is called as <u>reproduction</u> or <u>selection</u>.
  - After search/variation operators: The process of choosing the next generation population from parent and offspring populations is referred to as <u>survivor</u> or <u>elimination</u>. It is also being referred to as <u>environmental selection</u>.
- Common selection operators are
  - Fitness proportionate selection
  - Tournament selection
  - Ranking selection, etc.

Working Principles: Selection

Binary Tournament Selection Operator

- It is similar to playing a tournament among the teams. Here, binary stands for tournament between two teams.
- Outcome of binary tournament selection operator is: 'win', 'loss' or 'tie'.
- It is performed by picking two solutions randomly and choose the one that has better fitness value.

Working Principles: Binary Tournament Selection Operator

Index	1	2	3	4
Fitness	210.445	7536.407	14041.882	1546.603
Index	5	6	7	8
Fitness	189.732	671.924	7252.209	19793.183

Index	$f(x_1, x_2)$	Winner
2	7536.407	Index 4
4	1546.603	
7	7252.209	Index 7
3	14041.882	
8	19793.183	Index 6
6	671.924	
1	210.445	Index 5
5	189.732	

Index	$f(x_1, x_2)$	Winner
5	189.732	Index 5
7	7252.209	
4	1546.603	Index 4
2	7536.407	
3	14041.882	Index 6
6	671.924	
8	19793.183	Index 1
1	210.445	

Working Principles: Crossover Operator

- Crossover operator is responsible for creating new solutions. These new solutions explore the search space.
- Crossover is performed with probability (p_c). Generally, the value of p_c is kept high that supports exploration of search space.
- Types of crossover operators
  - Single-point crossover operator
  - n-point crossover operator
  - Uniform crossover operator

Single-point crossover operator

- For performing single-point crossover, two solutions are picked randomly from the pool at a time.

Working Principles: Mating Pool

- Mating pool is created after performing selection operator.

Old Index	New index	Chromosomes	DV (x_1)	DV (x_2)	x_1	x_2	$f(x_1, x_2)$
4	1	01000 10111	8	23	−2.419	1.935	1546.603
7	2	00101 11011	5	27	−3.387	2.968	7252.209
6	3	01101 01000	13	8	−0.806	−1.935	671.924
5	4	10100 11101	20	29	1.452	3.484	189.732
5	5	10100 11101	20	29	1.452	3.484	189.732
4	6	01000 10111	8	23	−2.419	1.935	1546.603
6	7	01101 01000	13	8	−0.806	−1.935	671.924
1	8	01100 11010	12	26	−1.129	2.710	210.445

- Let solutions with the following new indexes are picked for performing crossover.
  - Solutions {4,7}, {8,2}, {5,1} and {6,3}

Working Principles: Crossover

- The random numbers are generated for each pair of solutions. These random numbers are compared with p_c for performing crossover.
- Let $p_c = 0.9$.

Pair	Random number (r)	Pair	Random number (r)
{4,7}	0.75	{8,2}	0.23
{5,1}	0.93	{6,3}	0.68

- For the first pair, since $r = 0.75 < p_c = 0.9$, we perform crossover operator.
- Randomly, we choose 8th site to perform crossover.

Index	Chromosome	DV(s)	(x_1, x_2)	$f(x_1, x_2)$
P4:	10100111|01	(20,29)	$(1.452, 3.484)$	189.732
P7:	01101010|00	(13,8)	$(-0.806, -1.935)$	671.924
O4:	10100111|00	(20,28)	$(1.452, 3.226)$	125.336
O7:	01101010|01	(13,9)	$(-0.806, -1.677)$	545.121

Working Principles: Crossover

- For the second pair, since $r = 0.23 < p_c = 0.9$, we perform crossover operator.
- Randomly, we choose 3rd site to perform crossover.

Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2	$f(x_1, x_2)$
P8	011|0011010	12	26	−1.129	2.710	210.445
P2	001|0111011	5	27	−3.387	2.968	7252.209
O8	011|0111011	13	27	−0.806	2.968	540.287
O2	001|0011010	4	26	−3.710	2.710	12236.916

- For the third pair, since $r = 0.93 > p_c = 0.9$, we do not perform crossover operator. These solutions are copied directly.

Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2	$f(x_1, x_2)$
O5	1010011101	20	29	1.452	3.484	189.732
O1	0100010111	8	23	−2.419	1.935	1546.603

Working Principles: Crossover

- For the fourth pair, since $r = 0.68 < p_c = 0.9$, we perform crossover operator.
- Randomly, we choose 6th site to perform crossover.

Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2	$f(x_1, x_2)$
P6	010001|0111	8	23	−2.419	1.935	1546.603
P3	011010|1000	13	8	−0.806	−1.935	671.924
O6	010001|1000	8	24	−2.419	2.194	1351.054
O3	011010|0111	13	7	−0.806	−2.194	812.047

Working Principles: Crossover

- We observe that crossover can create good or bad solutions with respect to their parent solutions.
- If a bad solution is created, it will be eliminated during selection in further generations.
- If a good solution is created, it will have multiple copies in further generations.
- As crossover is performed on two parent solutions which survived the tournament selection
  - New solutions/offspring will more likely to preserve good traits of parents and will evolve as better solutions than their parents.

Working Principles: Crossover

Offspring population after crossover

Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2	$f(x_1, x_2)$
4	1010011100	20	28	1.452	3.226	125.336
7	0110101001	13	9	−0.806	−1.677	545.121
8	0110111011	13	27	−0.806	2.968	540.287
2	0010011010	4	26	−3.710	2.710	12236.916
5	1010011101	20	29	1.452	3.484	189.732
1	0100010111	8	23	−2.419	1.935	1546.603
6	0100011000	8	24	−2.419	2.194	1351.054
3	0110100111	13	7	−0.806	−2.194	812.047

Working Principles: Mutation

- Mutation operator is also responsible for search aspect of GA.
- The purpose of mutation is to keep diversity in the population.
- Mutation is generally performed with a small probability (p_m).

Bit-wise mutation operator

- A solution is chosen with the probability ($p_m = 0.10$) and a random bit is chosen for mutation.
- Following are the random numbers for performing mutation

Index	Random number (r)	Index	Random number (r)
1	0.05	2	0.32
3	0.15	4	0.01
5	0.06	6	0.24
7	0.5	8	0.54

Working Principles: Mutation

- For solution 1, since $r = 0.05 < p_m = 0.1$, we perform mutation.
- Randomly, we pick 4th bit-position to mutate 0 to 1, or 1 to 0.

Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2	$f(x_1, x_2)$
1	010 0 010111	8	23	−2.419	1.935	1546.603
1	010 1 010111	10	23	−1.774	1.935	154.658

- For solution 2, since $r = 0.32 > p_m = 0.1$, we do not perform mutation.
- Copy the solution.

Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2	$f(x_1, x_2)$
2	0010011010	4	26	−3.710	2.710	12236.916

Working Principles: Mutation

- For solution 3, since $r = 0.15 > p_m = 0.1$, we do not perform mutation.
- Copy the solution.

Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2	$f(x_1, x_2)$
3	0110100111	13	7	−0.806	−2.194	812.047

- For solution 4, since $r = 0.01 < p_m = 0.1$, we perform mutation.
- Randomly, we pick 5th bit-position for mutating the bit.

Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2	$f(x_1, x_2)$
4	1010 0 11100	20	28	1.452	3.226	125.336
4	1010 1 11100	21	28	1.774	3.226	1.208

Working Principles: Mutation

- For solution 5, since $r = 0.06 < p_m = 0.1$, we perform mutation.
- Randomly, we pick 1st bit-position for mutating the bit.

Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2	$f(x_1, x_2)$
5	$\boxed{1}$ 010011101	20	29	1.452	3.484	189.732
5	$\boxed{0}$ 010011101	20	29	−3.710	3.484	10585.571

- For other solutions, since $r > p_m$, we do not perform mutation.
- Copy them.

Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2	$f(x_1, x_2)$
6	0100011000	8	24	−2.419	2.194	1351.054
7	0110101001	13	9	−0.806	−1.677	545.121
8	0110111011	13	27	−0.806	2.968	540.287

Working Principles: Mutation

Offspring population after mutation

Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2	$f(x_1, x_2)$
1	0101010111	10	23	-1.774	1.935	154.658
2	0010011010	4	26	-3.710	2.710	12236.916
3	0110100111	13	7	-0.806	-2.194	812.047
4	1010111100	21	28	1.774	3.226	1.208
5	0010011101	20	29	-3.710	3.484	10585.571
6	0100011000	8	24	-2.419	2.194	1351.054
7	0110101001	13	9	-0.806	-1.677	545.121
8	0110111011	13	27	-0.806	2.968	540.287

- Similar to crossover operator, mutation operator can create better or worse solution than parent solution.
- However, good solution will be emphasized and bad solution may get deleted by selection operator in further generations.

Working Principles: Survivor

- It is used to preserve good solutions for the next generation.
- Survivor or elimination stage is also referred to as environmental selection.

$(\mu + \lambda)$-strategy

- μ stands for parent population and λ stands for offspring population.
- We combine both the population and choose the best solutions for the next generation population.

Working Principles: Survivor

Parent population

Index	$f(x_1, x_2)$
P1	210.445
P2	7536.407
P3	14041.882
P4	1546.603
P5	189.732
P6	671.924
P7	7252.209
P8	19793.183

Offspring population

Index	$f(x_1, x_2)$
O1	154.658
O2	12236.916
O3	812.047
O4	1.208
O5	10585.571
O6	1351.054
O7	545.121
O8	540.287

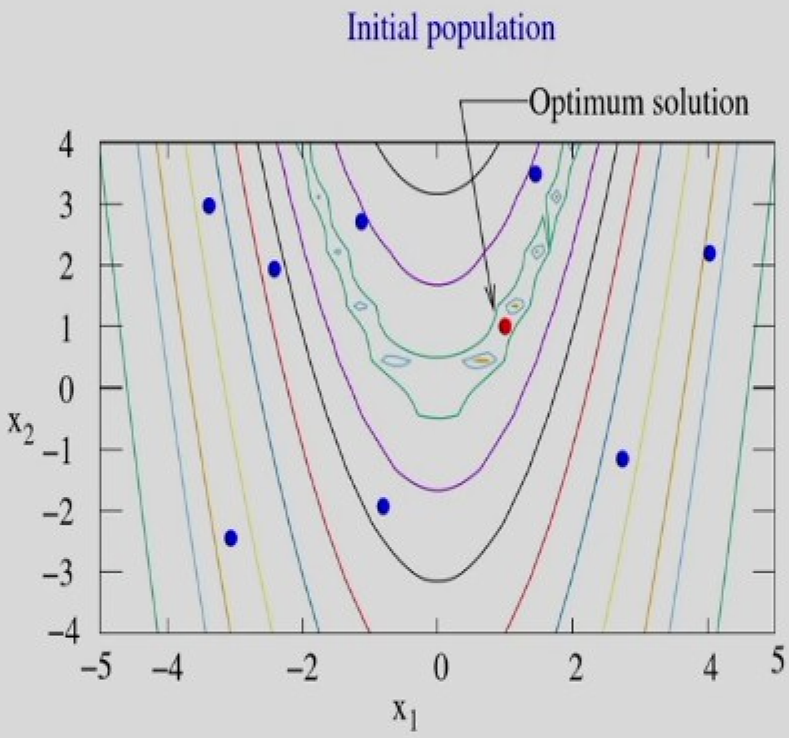
- Since it is the minimization problem, we select solutions in an ascending order of their fitness values.
- Select O4, O1, P5, P1, O8, O7, P6 and O3.

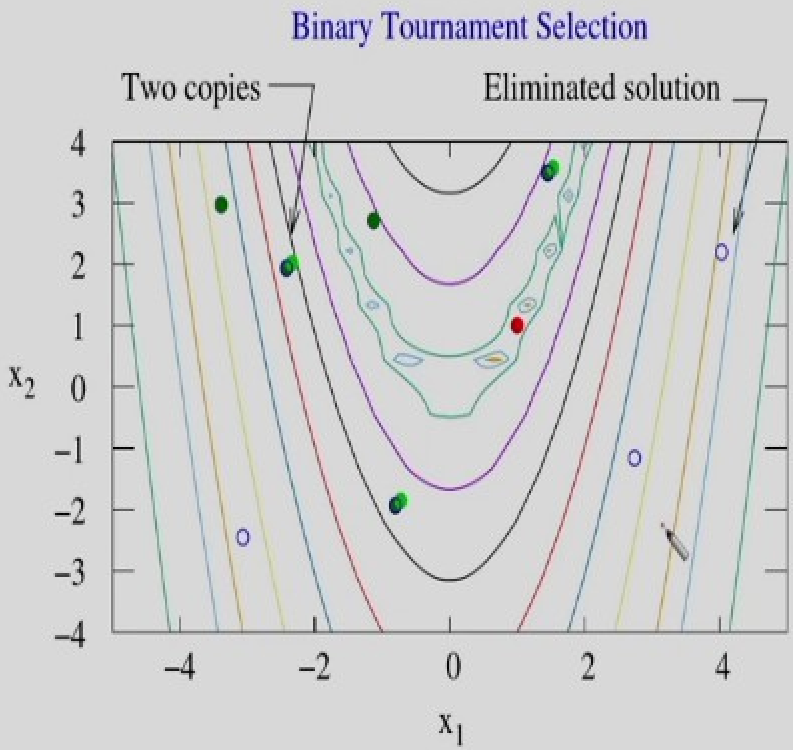
Working Principles: Survivor

Next generation population

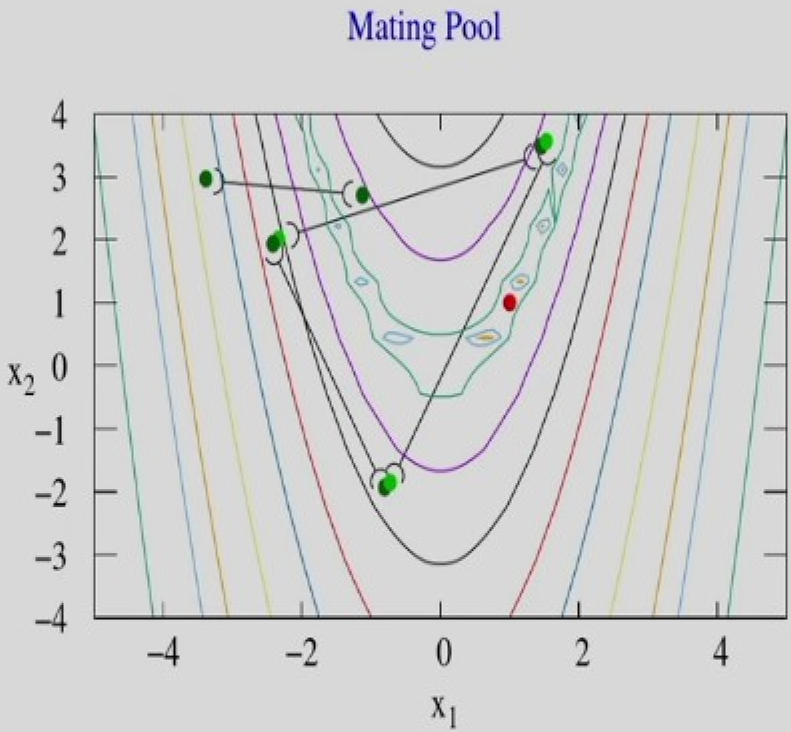
Index	Chromosomes	DV(x_1)	DV(x_2)	x_1	x_2	$f(x_1, x_2)$
1	1010111100	21	28	1.774	3.226	1.208
2	0101010111	10	23	−1.774	1.935	154.658
3	10100 11101	20	29	1.452	3.484	189.732
4	01100 11010	12	26	−1.129	2.710	210.445
5	0110111011	13	27	−0.806	2.968	540.287
6	0110101001	13	9	−0.806	−1.677	545.121
7	01101 01000	13	8	−0.806	−1.935	671.924
8	0110100111	13	7	−0.806	−2.194	812.047

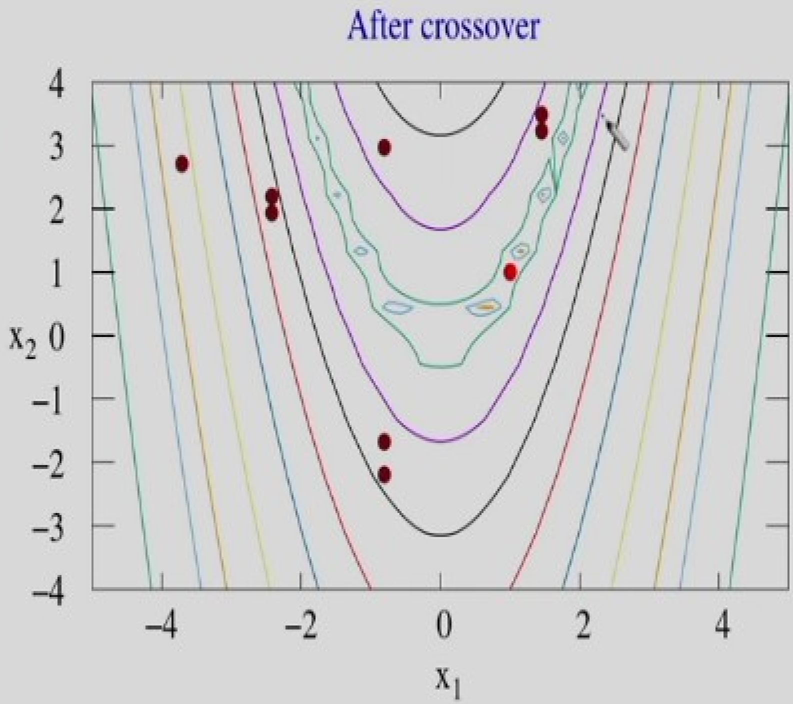
Graphical Example



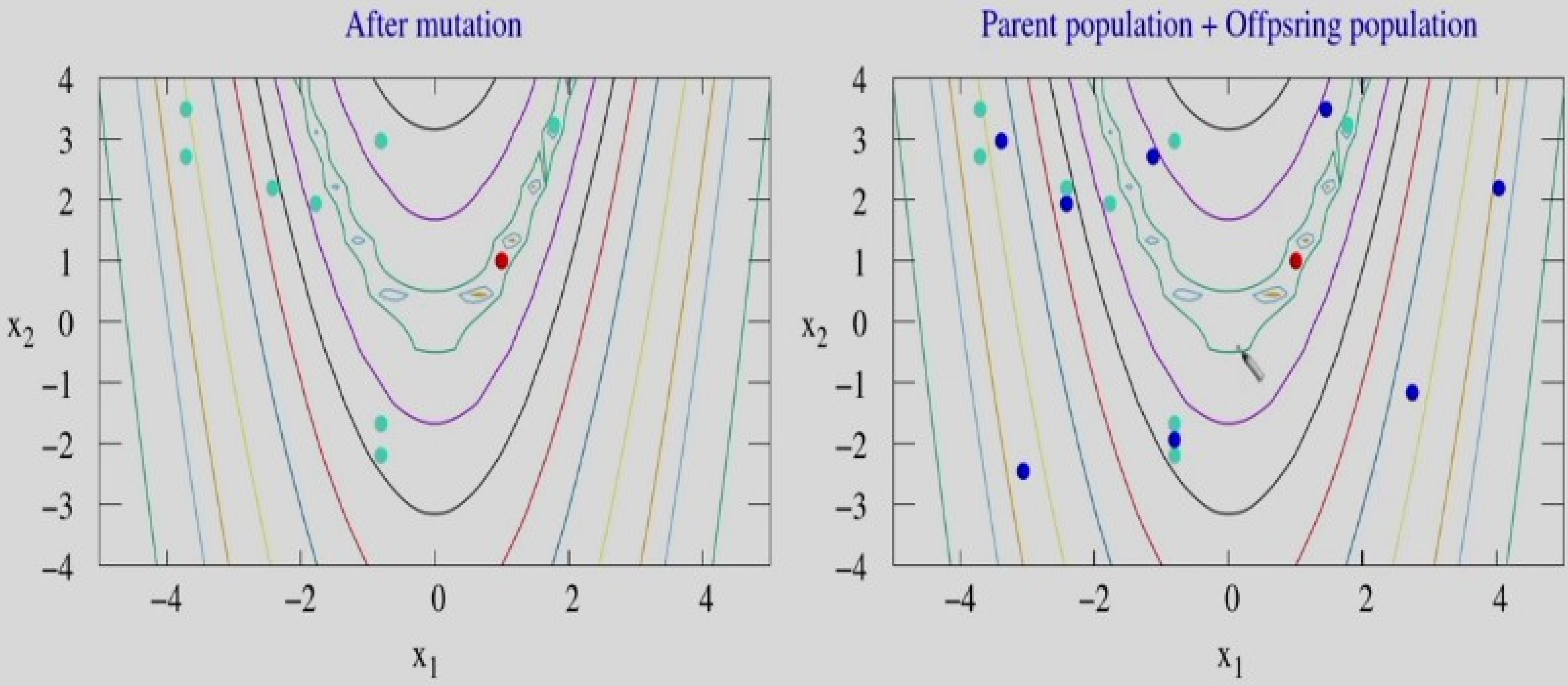


Graphical Example

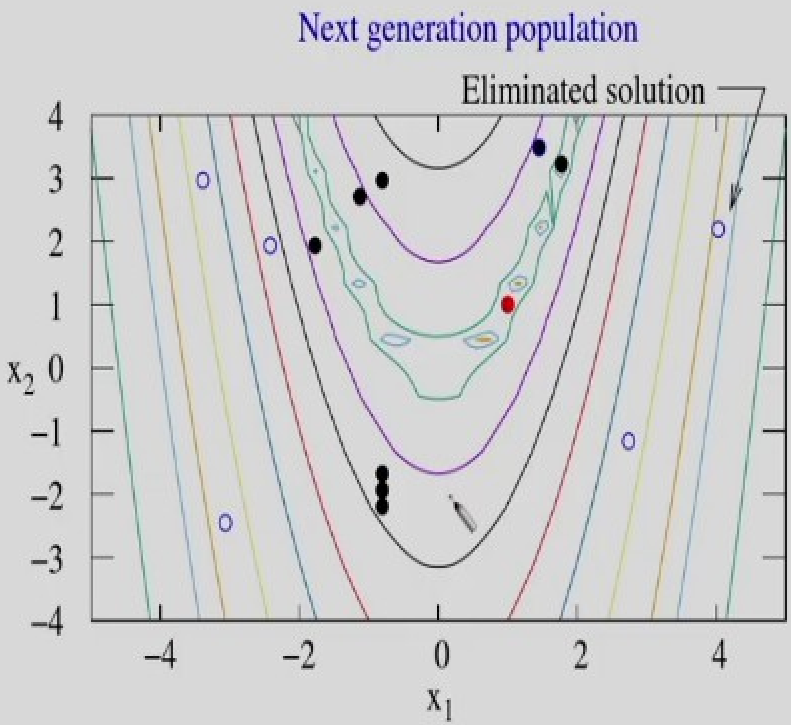


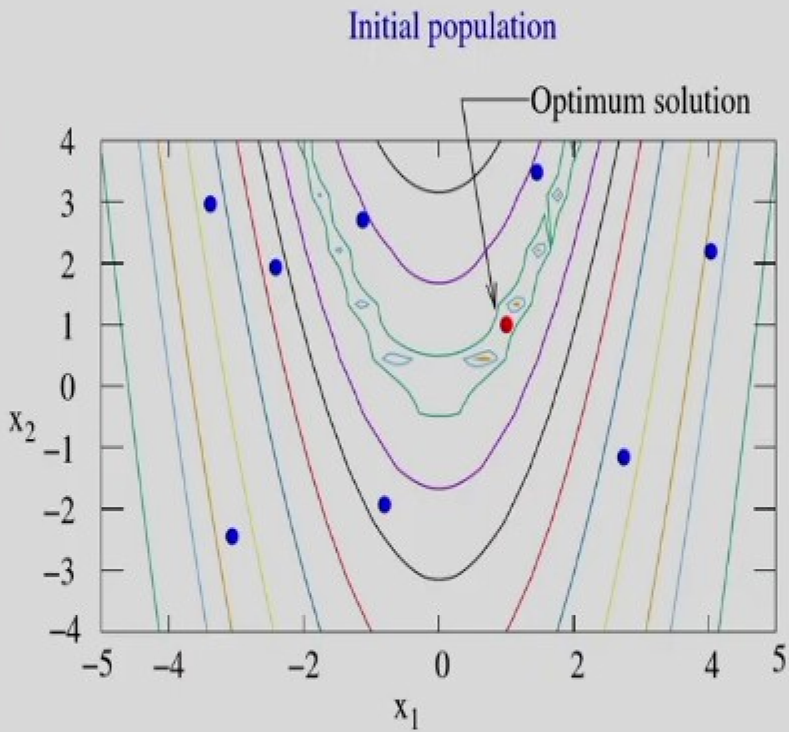


Graphical Example



Graphical Example





Closure

- Binary-coded genetic algorithm
  - Generalized framework
  - Solution representation
  - Working principles through an example
    - selection operator,
    - crossover and mutation operators,
    - survivor operator
  - Graphical example