Genetic Algorithm

Tournament selection

A method of selecting an individual from a population of individuals. Tournament selection involves running several "tournaments" among a few individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover.

Pseudo code:

choose n (the tournament size) individuals from the population at random
choose the best individual from pool/tournament with probability p
choose the second best individual with probability p*(1-p)
choose the third best individual with probability p*((1-p)^2)
and so on...

Genetic Algorithm

Tournament selection

```
def gen_parents(nnum):
    Ipar = np.zeros((2,nnum))
    Ipar[0][:] = np.argsort(np.random.rand(nnum))
    Ipar[1][:] = np.argsort(np.random.rand(nnum))
    Ipar = Ipar.astype(int)
    return Ipar
def tournament_selection(Ppop, PI, ptou):
    npop  = Ppop.shape[0]
    npvar = Ppop.shape[1]
    Pnew = np.zeros((npop,npvar))
    Ipar = gen_parents(npop)
    for i in range(npop):
        Pnew[i][:] = Ppop[Ipar[0][i]][:]
        ptran = np.random.random(1)
        if ptran < ptou:
            if PI[Ipar[1][i]] < PI[Ipar[0][i]]:
                Pnew[i][:] = Ppop[Ipar[1][i]][:]
    return Pnew
```

Genetic Algorithm

Crossover

- Two parents (randomly selected) produce two offspring

- Chromosomes of two parents can be copied unmodified as offspring

- The probability of crossover is between 0.6 to 1.0

Crossover: 1. point (binary)

One point crossover (binary)									
Parent 1	0	0	1	0	1	0	1	0	1
Parent 2	0	1	0	1	0	0	0	1	0
Offspring 1	0	0	1	1	0	0	0	1	0
Offspring 2	0	1	0	0	1	0	1	0	1

Crossover: 2. point (binary)

Two point crossover (binary)									
Parent 1	0	0	1	0	1	0	1	0	1
Parent 2	0	1	0	1	0	0	0	1	0
Offspring 1	0	0	1	1	0	0	0	0	1
Offspring 2	0	1	0	0	1	0	1	1	0

Crossover: Uniform crossover (binary)

Crossing based on random numbers 0 or 1

Uniform crossover (binary)	0	0	1	1	0	1	0	0	1
Parent 1	0	0	1	0	1	0	1	0	1
Parent 2	0	1	0	1	0	0	0	1	0
Offspring 1	0	0	0	1	1	0	1	0	0
Offspring 2	0	1	1	0	0	0	0	1	1

Crossover: HX (Heuristic - crossover)

From a pair of parents, $x^1 = (x_1^1, x_2^1, \dots \dots, x_n^1)$

$x^2 = (x_1^2, x_2^2, \dots \dots, x_n^2)$

an offspring $y_1 = (y_1, y_2, \dots \dots, y_n)$

is generated in the following manner

real $y_i = u(x_i^2 - x_i^1) + x_i^2$

integer $y_i = int(u(x_i^2 - x_i^1) + x_i^2)$

where u is a uniformly distributed random number [0,1]
and parent (2) has fittness value not wors than parent (1)

GA: LX crossover (Laplace – real – int)

Laplace crossover is defined for real and integer variables.

A parameter in the Laplace crossover operator take care of integer decision variables in the optimization problem.

Two offsprings $y^1 = \ (y_1^1, y_2^1, \ldots\ldots, y_n^1)$ and

$$y^2 = \ (y_1^2, y_2^2, \ldots\ldots, y_n^2)$$

are generated from a pair of parents

$$x^1 = (x_1^1, x_2^1, \ldots\ldots x_n^1) \text{ and}$$

$$x^2 = (x_1^2, x_2^2, \ldots\ldots x_n^2)$$

in following way:

GA: LX crossover (Laplace – real – int) cont.

First, a uniformly distributed random numbers $u_i, r_i = [0,1]$ are generated.

Then, a random number β_i is generated which follows the Laplace distribution by simply inverting the distribution function of Laplace distribution as follows:

$$\beta_i = \begin{cases} a - b \log_e u_i & r_i \leq \frac{1}{2} \\ a + b \log_e u_i & r_i > \frac{1}{2} \end{cases}$$

GA: LX crossover (Laplace – real – int) cont.

where a is location parameter and b > 0 is scaling parameter. If the decision variables have a restriction to be integer then $b = b_{int}$, otherwise $b = b_{real}$, i.e., for integer and real decision variables, scaling parameter is different.

Typical values: $a = 0$ $b_{int} = 0.35$ $b_{real} = 0.15$

The offsprings are given by the equations:

$$y_i^1 = x_i^1 + \beta\left|x_i^1 - x_i^2\right|$$

$$y_i^2 = x_i^2 + \beta\left|x_i^1 - x_i^2\right|$$

GA: Non-Uniform Mutation

Michalewichz's Non-Uniform Mutation is one of the widely used mutation operators in real coded GAs.

From the point $x^t = (x_1^t, x_2^t, \ldots\ldots x_n^t)$ the muted point $x^{t+1} = (x_1^{t+1}, x_2^{t+1}, \ldots\ldots x_n^{t+1})$ is created as follows:

$$x_i{}^{t+1} = \begin{cases} x_i^t + \Delta\left(t, x_i^u - x_i^t\right) & if \quad r \leq \frac{1}{2} \\ x_i^t - \Delta\left(t, x_i^t - x_i^l\right) & if \quad r > \frac{1}{2} \end{cases}$$

GA: Non-Uniform Mutation

where t is the current generation number and r is a uniformly distributed random number in interval [0,1]

x_i^l and x_i^u are lower an duper bounds of the ith component of the decision vector respectively.

The function $\Delta(t, y) = y\left(1 - u^{\left(1-\frac{t}{T}\right)}\right)^b$ where u is a uniformly distributed random number in the interval [0,1], T is the maximum number of generations and b is a parameter, determining the strength of the mutation operator.

Genetic Algorithm

Permutation Crossover

- Ordinal representation (One point crossover)
- Partially – mapped crossover
- Cycle crossover
- Modified crossover
- Order crossover
- Order based crossover
- Position based crossover
- Edge recombination

Genetic Algorithm

NP combinatorial Problems: Crossover

The ordinal representation (One point crossover)

Current Tour	Canonic Tour	Ordinal Representation
<u>1</u> 2 5 6 4 3 8 7	<u>1</u> 2 3 4 5 6 7 8	1
1 <u>2</u> 5 6 4 3 8 7	<u>2</u> 3 4 5 6 7 8	1 1
1 2 <u>5</u> 6 4 3 8 7	3 4 <u>5</u> 6 7 8	1 1 3
1 2 5 <u>6</u> 4 3 8 7	3 4 <u>6</u> 7 8	1 1 3 3
1 2 5 6 <u>4</u> 3 8 7	3 <u>4</u> 7 8	1 1 3 3 2
1 2 5 6 4 <u>3</u> 8 7	<u>3</u> 7 8	1 1 3 3 2 1
1 2 5 6 4 3 <u>8</u> 7	7 <u>8</u>	1 1 3 3 2 1 2
1 2 5 6 4 3 8 <u>7</u>	<u>7</u>	1 1 3 3 2 1 2 1

The ordinal representation.

parent 1 (12564387):	**1**	**1** |	**3**	**3**	**2**	**1**	**2**	**1**
parent 2 (14236578):	1	3 |	1	1	2	1	1	1
offspring (12346578):	**1**	**1**	1	1	2	1	1	1

NP combinatorial Problems: Crossover

Crossover operators preserving the absolute position

The partially-mapped crossover

parent 1	:	1	2 |	5	6	4 |	3	8	7
parent 2	:	1	4 |	2	3	6 |	5	7	8
offspring (step 1)	:	1	4	5	6	4	5	7	8
(step 2)	:	1	3	5	6	4	2	7	8

The partially-mapped crossover.

NP combinatorial Problems: Crossover

The cycle crossover

parent 1 :	1	3	5	6	4	2	8	7
parent 2 :	1	4	2	3	6	5	7	8
offspring :	1	3	2	6	4	5	7	8

NP combinatorial Problems: Crossover

Crossover operators preserving the relative order

The modified crossover

parent 1:	1	2	|	5	6	4	3	8	7
parent 2:	1	4		2	3	6	5	7	8
offspring:	1	2		4	3	6	5	7	8

The modified crossover.

NP combinatorial Problems: Crossover

The order crossover

parent 1	:	1	2	|	5	6	4	|	3	8	7
parent 2	:	1	4	|	2	3	6	|	5	7	8
offspring											
(step 1)	:	-	-		5	6	4		-	-	-
(step 2)	:	2	3		5	6	4		7	8	1

The order crossover.

NP combinatorial Problems: Crossover

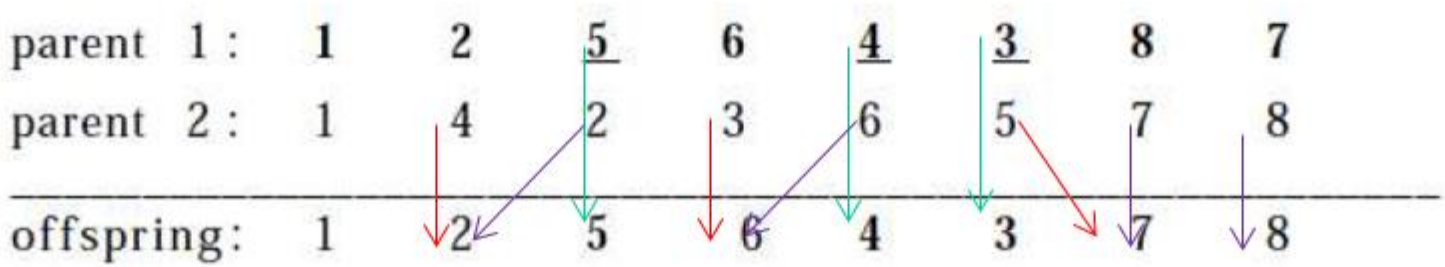
The order based crossover

parent 1 :	1	2	5	6	4	3	8	7
parent 2 :	1	4	2	3	6	5	7	8
offspring :	1	5	2	4	6	3	7	8

The order-based crossover

NP combinatorial Problems: Crossover

The position based crossover



The position-based crossover.

Genetic Algorithm

NP combinatorial problems: Edge recombination

od = 8 ;
xp1 = [1 3 5 6 4 2 8 7] ;
xp2 = [1 4 2 3 6 5 7 8] ;

Edge table:

city	1	has	edges	to	:	3 4 7 8
city	2	has	edges	to	:	3 4 8
city	3	has	edges	to	:	1 2 5 6
city	4	has	edges	to	:	1 2 6
city	5	has	edges	to	:	3 6 7
city	6	has	edges	to	:	3 4 5
city	7	has	edges	to	:	1 5 8
city	8	has	edges	to	:	1 2 7

Genetic Algorithm

NP combinatorial problems: Edge

City 1 is randomly selected as starting city

All edges incident to city 1 are deleted from the edge map

From city 1, we can go to cities 3,4,7 and 8.

City 3 has three active edge, while city 4, 7 and 8 have two edge. Hence, a random choice is made between 4, 7 and 8.

City 8 is randomly selected.

From city 8 we can go to cites 2 and 7.

As indicated in edge map, city 2 has two active edges and city 7 has only one.

City 7 is selected and from there is no choice, but go to city 5.

Genetic Algorithm

NP combinatorial problems: Edge

From city 5 the edge map offers a choice between cities 3 and 6, both with two active edge.

City 6 is randomly selected.

From city 6 we can go to cities 3 and 4, both with one active edge.

City 4 is randomly selected.

Finally from city 4 we can only go to city 2 and from 2 to city 3

The final tour: 18756423

NP combinatorial problems: Edge

The final tour is 18756423 and all edges are inherited from both parents.

City 1 is selected

(a)
city 2 has edges to : 3 4 8
city 3 has edges to : 2 5 6
city 4 has edges to : 2 6
city 5 has edges to : 3 6 7
city 6 has edges to : 3 4 5
city 7 has edges to : 5 8
city 8 has edges to : 2 7

City 8 is selected

(b)
city 2 has edges to : 3 4
city 3 has edges to : 2 5 6
city 4 has edges to : 2 6
city 5 has edges to : 3 6 7
city 6 has edges to : 3 4 5
city 7 has edges to : 5

City 7 is selected

(c)
city 2 has edges to : 3 4
city 3 has edges to : 2 5 6
city 4 has edges to : 2 6
city 5 has edges to : 3 6
city 6 has edges to : 3 4 5

City 5 is selected

(d)
city 2 has edges to : 3 4
city 3 has edges to : 2 6
city 4 has edges to : 2 6
city 6 has edges to : 3 4

City 6 is selected

(e)
city 2 has edges to : 3 4
city 3 has edges to : 2
city 4 has edges to : 2

City 4 is selected

(f)
city 2 has edges to : 3
city 3 has edges to : 2 6

City 2 is selected

(g)
city 3 has edges to :

City 3 is selected