



MUTATIONS

Contents

1.1.	Displacement Mutation	3
1.2.	Flip Bit Mutation	4
1.3.	Gaussian Mutation.....	5
1.4.	Insertion Mutation.....	6
1.5.	Partial Shuffle Mutation.....	7
1.6.	Polynomial Mutation	8
1.7.	Reverse Sequence Mutation	9
1.8.	Twors Mutation	10
1.9.	Uniform Mutation.....	11

Mutation operators may be designed for special type of chromosomes: binary, combinatorial or double (real-variable) ones. If used on unsupported type of a chromosome, the whole process may be terminated instantly:

Mutation operator	Chromosome type		
	Binary	Combinatorial	Double
<i>Displacement Mutation</i>	✓	✓	
<i>Flip Bit Mutation</i>	✓		
<i>Gaussian Mutation</i>			✓
<i>Insertion Mutation</i>	✓	✓	
<i>Partial Shuffle Mutation</i>	✓	✓	
<i>Polynomial Mutation</i>			✓
<i>Reverse Sequence Mutation</i>	✓	✓	
<i>Twors Mutation</i>	✓	✓	
<i>Uniform Mutation</i>	✓		

1.1. Displacement Mutation

Parameters
<i>none</i>

Displacement mutation operates on a sequence of elements, chosen randomly within a chromosome. On implementation, the selected swath is shifted to the left or right edge of the chromosome. Direction in which the sequence is shifted is determined in a random way.

Example

Chromosome:

A	C	D	E	F	B
---	---	---	---	---	---

Mutated chromosome:

At first, a random number of elements to shift is obtained, in this case: **3** (marked on green above). On implementation, three first elements are shifted to the right or three last characters are shifted to the left. At random, the first alternative was enforced, finally producing the mutated individual:

E	F	B	A	C	D
---	---	---	---	---	---

1.2. Flip Bit Mutation

Parameters
<i>none</i>

Flip bit mutation is a mutation operator taking a random element in a binary representation of a floating point chromosome. The selected gene is then inverted, i.e. 1 replaced with 0, or 0 replaced with 1. The flip bit mutation returns exception if used in ordered chromosome-based problems.

Example

Chromosome:

1	0	1	1	0	1
---	---	---	---	---	---

Mutated chromosome:

1	1	1	1	0	1
---	---	---	---	---	---

At random, the element at the second index was chosen to be inverted (marked on green). Since its original value is 0, the operator replaces it with 1:

1.3. Gaussian Mutation

Parameters	
<i>strengthParameter</i>	A strength parameter, taken as 1/30 by default.

Gaussian mutation is used for real-variable (double) chromosomes. A mutated solution candidate c' is obtained from the original one c utilizing the following formula:

$$c' = c + \sqrt{2} \cdot \sigma \cdot (c_{\max} - c_{\min}) \cdot \text{erf}^{-1} u'$$

where c_{\min} and c_{\max} are the minimum (lower bound) and maximum (upper bound) values of a variable defined in a chromosome, respectively. The σ value is the strength parameter given as a node input, taken as 1/30 by default. The rest of symbols is calculated as follows.

At first, a random floating point number u between 0.0 and 1.0 is drawn, used to select one of the two ways the u' is calculated.

If $u \leq 0.50$: $u' = 2 \cdot \delta \cdot (1 - 2 \cdot u)$, where:

$$\delta = \frac{1}{2} \cdot \left[\text{erf} \left[\frac{c_{\min} - c}{\sqrt{2} \cdot \sigma \cdot (c_{\max} - c_{\min})} \right] + 1 \right]$$

Otherwise: $u' = 2 \cdot \delta \cdot (2 \cdot u - 1)$, where:

$$\delta = \frac{1}{2} \cdot \left[\text{erf} \left[\frac{c_{\max} - c}{\sqrt{2} \cdot \sigma \cdot (c_{\max} - c_{\min})} \right] + 1 \right]$$

Erf is the [Gaussian error function](#). Finally an inversed error function of the u' is calculated using the formula:

$$\text{erf}^{-1} u' = \text{sign } u' \cdot \left[\sqrt{\left(\frac{2}{\pi \cdot \alpha} + \frac{\ln(1 - u'^2)}{2} \right)^2 - \frac{\ln(1 - u'^2)}{\alpha}} - \left(\frac{2}{\pi \cdot \alpha} + \frac{\ln(1 - u'^2)}{2} \right) \right]^{\frac{1}{2}}$$

The α is an auxiliary constant variable:

$$\alpha = \frac{8 \cdot (\pi - 3)}{3 \cdot \pi \cdot (4 - \pi)}$$

1.4. Insertion Mutation

Parameters
<i>none</i>

Insertion mutation is a particular variant of the displacement mutation operator (see: chapter 1.1). In this case, only one element is chosen randomly within the whole chromosome. On implementation, the selected gene is shifted to the left or to the right edge of the chromosome. Direction in which the element is shifted is determined in a random way.

Example

Chromosome:

A	C	D	E	F	B
---	---	---	---	---	---

Mutated chromosome:

At random, the element at the second index was chosen within the chromosome to be shifted (marked on green above). On implementation, the element will be moved to the left or to the right edge of the chromosome. At random, the first alternative was enforced, finally producing the mutated individual:

A	D	E	F	B	C
---	---	---	---	---	---

1.5. Partial Shuffle Mutation

Parameters
<i>none</i>

Partial shuffle mutation (**PSM**) operator obtains a random swath within a chromosome. The idea of the operator is to shuffle elements included in the swath. Several shuffles may be used if previous one did not cause any change in the sequence.

Example

Chromosome:

A	C	D	E	F	B
---	---	---	---	---	---

Mutated chromosome:

It is assumed that second, third and fourth elements are chosen to form a swath (marked on green above). The implemented randomizer is responsible for shuffling the elements within the swath. Finally, a new sequence is produced:

A	C	E	D	F	B
---	---	---	---	---	---

1.6. Polynomial Mutation

Parameters	
<i>distributionIndex</i>	<i>A polynomial probability distribution (values of 20 – 100 are recommended).</i>

Polynomial mutation is used for real-variable (double) chromosomes. According to Deb (Deb, 2014), in this operator, a polynomial probability distribution is used to perturb a solution in a candidate's vicinity. Mathematical formulas are used as follows.

At first, a random floating point number u between 0.0 and 1.0 is drawn, used to select one of the two ways the candidate c is mutated to c' :

If $u \leq 0.50$: $c' = c + \delta \cdot (c - c_{\min})$, where:

$$\delta = (2 \cdot u)^{\frac{1}{\eta_m + 1}} - 1$$

Otherwise: $c' = c + \delta \cdot (c_{\max} - c)$, where:

$$\delta = 1 - [2 \cdot (1 - u)]^{\frac{1}{\eta_m + 1}}$$

In the formulas above c_{\min} and c_{\max} are the minimum (lower bound) and maximum (upper bound) values of a variable defined in a chromosome, respectively. The η_m value used in the formulas above is the distribution index given as a node input.

It is found that η_m values in range between 20.0 and 100.0 are adequate for most optimization problems. Its default value is 20.0.

1.7. Reverse Sequence Mutation

Parameters
<i>none</i>

Reverse sequence mutation (**RSM**) is an operator considering a randomly chosen swath within a chromosome. The selected sequence is then reversed.

Example

Chromosome:

A	C	D	E	F	B
---	---	---	---	---	---

Mutated chromosome:

At random, second, third and fourth elements were chosen and will be modified by the operator. The initial order of the elements is: C, D and E (marked on green above). After reversion, the new order is: E, D and C, respectively:

A	E	D	C	F	B
---	---	---	---	---	---

1.8. Twors Mutation

Parameters
<i>none</i>

Twors mutation operator obtains a random pair of elements within a chromosome. The elements are then shuffled, trading their locations.

Example

Chromosome:

A	C	D	E	F	B
---	---	---	---	---	---

Mutated chromosome:

First and third elements were randomly chosen to be shuffled (marked on green above). First element is then moved to the third index and third element is moved to the first index, forming a new order in the chromosome structure:

D	C	A	E	F	B
---	---	---	---	---	---

1.9. Uniform Mutation

Parameters
<i>none</i>

The uniform mutation iterates through the whole length of a chromosome. At each element, a random number is obtained, deciding if the element should be replaced with a new one or skipped. Decision is made basing on mutation probability specified by a user.

The operator cannot be used on chromosomes in ordered chromosome-based problems as it may cause repetition of genes.

Example

Chromosome:

1	0	1	1	0	1
---	---	---	---	---	---

Mutated chromosome:

The algorithm iterates through the chromosome. At random, two elements – at first and third index were chosen for mutation (marked on green below). The new values of the elements are uniformly drawn – note that the element chosen for mutation may change its initial value or not, if drawn so:

1	0	0	1	0	1
---	---	---	---	---	---

References

Deb, K.: *An Efficient Constraint Handling Method for Genetic Algorithms*. Computer Methods in Applied Mechanics and Engineering, vol. 186, issue 2, 4, June 2000, pp. 311 – 338.

Deb, K., Deb, D.: *Analyzing Mutation Schemes for Real-Parameter Genetic Algorithms*. International Journal of Artificial Intelligence and Soft Computing, vol. 4, issue 1, February 2014, pp. 1 – 28.

Larrañaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., Dizdarevic, S.: *Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators*. Artificial Intelligence Review, vol. 13, issue 2, April 1999, pp. 129 – 170.