

# PBILAlgorithm

## Algorithm

eda, binary

`Gevol.evolution.algorithm.eda.binary`

### Description

Population-Based Incremental Learning (PBIL) algorithm was invented by Shumeet Baluja in 1994. It is very easy, almost self-learning algorithm from Estimation of Distribution Algorithms.

Result of this algorithm is a vector with probabilities to get 1 for each gene. Genes are separate in the chromosome. In every iteration the best individual is being taken and the vector is modified based on its values. Probabilities are changed according to the formula:

$$P_i = ((1 - LR)P_i) + x_iLR$$

Where

- $P_i$  - probability to get 1 on i-th gene
- LR - learning rate, how fast the algorithm learns
- $x_i$  - actual value on i-th gene of the best individual

On each gene mutation may occur or not. If it occurs, the probability is changed.

$$P_i = P_i(1 - MUT) + rand(1,0)MUT$$

Where

- $P_i$  - probability to get 1 on i-th gene
- MUT - mutation size
- $rand(1,0)$  - random value of 1 or 0

### Parameters

1. LR - learning rate, how strong the probability for each gene will be modified based on current the best individual
2. MUT - mutation size, how strong probability will be affected by mutation
3. MUT\_PROB - probability that mutation occurs
4. Population size - how big population will be generated

### Pseudocode

```
P = generatePopulation(V)
I = theBestIndividual(P)
V = updateProbabilityVector(V, I, LR)
V = mutation(V, MUT, MUT_PROB)
```

### Implementation details

It generates new population based on the current model. So it doesn't require any selection operator. Population size must be set also in algorithm class, because it is used to generate the first population during `Init()`. The constructor:

```
this.PopulationSize = newPopulationSize;
```

```
Operators.Add(new PBILOperator(newPopulationSize, chromosomeLength, learningRate,  
mutationSize, mutationProbability));
```

Mutation for each probability is calculated in the same loop just after the probability is calculated. It improves performance as the loop doesn't have to be repeated.

## References

1. Population-Based Incremental Learning, Shumeet Baluja, School of Computer Science Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, CMU-CS-94-163, June 2, 1994