

CGAAlgorithm

Algorithm

eda, binary

Namespace

Description

Compact Genetic Algorithm was introduced to avoid high memory usage. In this algorithm the population is not used. Vector of probabilities is generated based on only two individuals. The population is simulated by parameter used to modify the vector.

At first step two individuals are generated from the model. The better one is used to update probabilities.

$$p_i = \begin{cases} p_i + \frac{1}{n}, & I_{winner_j} \neq I_{loser_j} \text{ and } I_{winner_j} = 1 \\ p_i - \frac{1}{n}, & I_{winner_j} \neq I_{loser_j} \text{ and } I_{winner_j} = 0 \end{cases}$$

Where

- p_i - probability to get 1 on i-th gene
- n - simulated population size
- I_{winner_j} - value of j-th gene in better individual
- I_{loser_j} - value of j-th gene in worse individual

The algorithm runs until all probabilities have value 0 or 1.

Parameters

1. n - simulated population size

Pseudocode

```
M = initModel(0.5)
while not each m in M = 0 or 1
    P = generateTwoIndividuals(M)
    I = getBetterIndividual(P)
    M = updateModel(I, M)
```

Implementation details

Source codes with explanations.

References

1. The Compact Genetic Algorithm, Georges R. Harik, Fernando G. Lobo, David E. Goldberg, University of Illinois at Urbana-Champaign, Urbana, IL 61801, IlliGal Report No. 97006, August 1997