

Página de Ajuda / Help

Função Principal

genAlgo package:geneticAlgorithm R Documentation

Generic genetic function.

Description:

Genetic Algorithms are search and optimization algorithms based on genetic concepts, such as "selection", "crossover" and "mutation".

genAlgo function is a generic genetic function which can receive a fitness function based on a given problem.

Usage:

```
genAlgo = function(pop = FALSE, length = 3, popSize = 5, pCross = 0.7,
pMut = 0.001, generations = 20, fitness = fitSum, showAll = FALSE, report =
FALSE)
```

Arguments:

pop	logical.
length	numerical. Chromosome length.
popSize	numerical. Population size.
pCross	numerical. Crossover rate.
pMut	numerical. Mutation rate.
generations	numerical. Number of iterations.
fitness	function. Fitness function.
showAll	logical. If TRUE, at the end of each generation, prints the new population.
report	logical. If TRUE, at the end of all generations, plots the fitness average and variance through the generations.

Warning:

This function uses a simple fitness function default. The user must define its own fitness function based in its own problem to solve.

Value:

This function has no return value.

Author:

Ana C. M. Ciconelle
ciconella@gmail.com
ana.ciconelle@usp.br

References:

Mitchell, Melanie. An Introduction to Genetic Algorithms. First MIT Press paperback edition, 1998.

See Also:

fitBinary, fitFindSeq, fitY

Examples:

For more example details, see the fitness function help.

```
# example1
genAlgo(length = 5, generations = 3, showAll = TRUE, report = TRUE)
```

```
[1] "Initial Population"
      [,1] [,2] [,3] [,4] [,5]
[1,]    0    1    1    1    1
[2,]    1    0    1    1    0
[3,]    0    1    1    1    1
[4,]    0    1    1    1    1
[5,]    1    1    1    0    0
[6,]    0    0    0    0    1
[7,]    1    1    0    0    0
[8,]    0    0    0    1    1
[9,]    1    1    1    0    0
[10,]   1    1    0    0    0
```

```
[1] "Initial Fitness"
[1] 4 3 4 4 3 1 2 2 3 2
[1] "Mean = "          "2.8"          "Variance ="
"1.066666666666667"
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]    0    1    1    1    0
[2,]    1    1    1    1    1
[3,]    0    0    0    0    0
[4,]    0    1    1    1    1
[5,]    0    1    1    0    0
[6,]    1    0    1    1    1
[7,]    0    0    0    0    0
[8,]    0    0    0    0    0
[9,]    0    1    1    1    1
[10,]   1    1    0    0    0
[1] "mean = "          "2.8"          "var ="
"1.066666666666667"
```

```
      [,1] [,2] [,3] [,4] [,5]
```

```
[1,] 1 1 1 1 1
[2,] 1 1 0 1 0
[3,] 1 1 1 1 1
[4,] 1 1 1 1 1
[5,] 0 1 1 1 1
[6,] 0 1 1 1 1
[7,] 0 1 1 0 0
[8,] 1 1 1 1 1
[9,] 1 1 1 1 1
[10,] 1 0 1 1 1
[1] "mean = " "2.4" "var =" "3.6"
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,] 1 1 1 1 0
[2,] 1 1 1 1 1
[3,] 1 1 1 1 1
[4,] 1 1 1 1 1
[5,] 1 1 1 1 1
[6,] 1 1 0 1 1
[7,] 1 0 1 1 1
[8,] 1 1 1 1 1
[9,] 1 1 1 1 1
[10,] 1 1 1 1 1
[1] "mean = " "4.2" "var ="
"1.066666666666667"
```

```
[1] "Final Population"
      [,1] [,2] [,3] [,4] [,5]
[1,] 1 1 1 1 0
[2,] 1 1 1 1 1
[3,] 1 1 1 1 1
[4,] 1 1 1 1 1
[5,] 1 1 1 1 1
[6,] 1 1 0 1 1
[7,] 1 0 1 1 1
[8,] 1 1 1 1 1
[9,] 1 1 1 1 1
[10,] 1 1 1 1 1
```

```
[1] "Final Fitness"
[1] 5 3 5 5 4 4 2 5 5 4
[1] "Mean = " "4.2" "Variance ="
"1.066666666666667"
```

```
[1] "Best chromosome:"
[1] 1 1 1 1 1
[1] "Fitness:"
[1] 5
[1] "First Appereance in generation:"
[1] 2
```

```
# example2
genAlgo(fitness = fitBinary, report = TRUE)

[1] "Initial Fitness"
[1] "Mean = "          "14710"          "Variance ="
"57490918.4444444"

[1] "Final Fitness"
[1] "Mean = "          "32767"          "Variance =" "0"

[1] "Best chromossome:"
[1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[1] "Fitness:"
[1] 32767
[1] "First Appereance in generation:"
[1] 3

# example3
genAlgo(length = 5, fitness = fitFindSeq, report = TRUE)

[1] "Initial Fitness"
[1] "Mean = "          "2.4"            "Variance ="
"1.822222222222222"

[1] "Final Fitness"
[1] "Mean = "          "3"              "Variance =" "0"

[1] "Best chromossome:"
[1] 0 1 1 1 0

[1] "Fitness:"
[1] 5

[1] "First Appereance in generation:"
[1] 1

# example4
genAlgo(length = 10, fitness = fitY, report = TRUE)

> genAlgo(length = 10, fitness = fitY, report = TRUE)
[1] "Initial Fitness"
[1] "Mean = "          "0.779282794386835" "Variance ="
[4] "0.477544654918165"

[1] "Final Fitness"
[1] "Mean = "          "1.30466061226754" "Variance ="
[4] "0.670012892850927"
```

```
[1] "Best chromossome:"  
[1] 1 0 0 1 0 0 0 0 0  
[1] "Fitness:"  
[1] 1.81966  
  
[1] "First Appereance in generation:"  
[1] 11
```

Funções Auxiliares

cummulative package:geneticAlgorithm R
Documentation

Find the interval of chances of chromosomes of being select.

Description:

Given a fitness array, returns a interval of chances of chromosomes of being select.

Usage:

```
cummulative(fit)
```

Arguments:

`fit` numeric. An array in which the numbers represents the fitness of chromosomes.

Author:

```
Ana C. M. Ciconelle  
cicconella@gmail.com  
ana.ciconelle@usp.br
```

References:

Mitchell, Melanie. An Introduction to Genetic Algorithms. First MIT Press paperback edition, 1998.

Example:

```
> b #Fitness  
[1] 1 2 3  
> b <- b/sum(b)  
> b <- cummulative(b)  
> b  
[1] 0.1666667 0.5000000 1.0000000 # Chromosome 1 has 0.16666 chances of be a  
parent, while chrom. 2 has (0.5 - 0.16) chances and chromosome 3 has 0.5.
```

selection package:geneticAlgorithm R

Documentation

Selects chromosomes in the population for reproduction.

Description:

Selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be selected to reproduce.

Usage:

```
selection(normFit)
```

Arguments:

`normFit` numeric. An array in which the numbers represents the intervals to a chromosome become a parent.

Value:

Returns the array of chosen parents of next generation.

Author:

Ana C. M. Ciconelle
ciconella@gmail.com
ana.ciconelle@usp.br

References:

Mitchell, Melanie. An Introduction to Genetic Algorithms. First MIT Press paperback edition, 1998.

See Also:

`cummulative`

Example:

```
> b #Fitness
[1] 1 2 3
> b <- b/sum(b)
> b <- cummulative(b)
> b
[1] 0.1666667 0.5000000 1.0000000 # Chromosome 1 has 0.16666 chances of be a
parent, while chrom. 2 has (0.5 - 0.16) chances and chromosome 3 has 0.5.
> selection(b)
[1] 3 2 2
```

`crossover`

package:geneticAlgorithm

R

Documentation

Do crossover between parents chromosomes.

Description:

Randomly chooses locus and exchanges the subsequences before and after that locus between two chromosomes to create two offsprings. If number of parents are odd, one parent turns a new offspring.

Usage:

```
crossover (pop, parents, pCross)
```

Arguments:

pop	numeric. Current population.
parents	numeric. Chromosome selected to be parents.
pCross	numeric. Probability of a crossover occurs.

Value:

Returns the population of the next generation.

Author:

Ana C. M. Ciconelle
ciconella@gmail.com
ana.ciconelle@usp.br

References:

Mitchell, Melanie. An Introduction to Genetic Algorithms. First MIT Press paperback edition, 1998.

See Also:

cummulative, selection

mutation package:geneticAlgorithm R Documentation

Do mutations in chromosomes of a given population.

Description:

Randomly flips some of the bits in a chromosome.

Usage:

```
mutation(newPop, pMut)
```

Arguments:

newPop	numeric. Current population.
pCross	numeric. Probability of a mutation occurs.

Value:

Returns the population of the next generation.

Author:

Ana C. M. Ciconelle
ciconella@gmail.com
ana.ciconelle@usp.br

References:

Mitchell, Melanie. An Introduction to Genetic Algorithms. First MIT Press paperback edition, 1998.

See Also:

cummulative, selection

Example:

```
> pop
      [,1] [,2] [,3]
[1,]    1    1    0
[2,]    0    1    1
[3,]    0    1    0

> mutation(pop, 0.5)
      [,1] [,2] [,3]
[1,]    0    0    0
[2,]    1    0    0
[3,]    0    0    1
```

printReport package:geneticAlgorithm R
Documentation

Plot graphs with the fitness mean and variance in every generation.

Description:

Plot graphs with the fitness mean and variance in every generation.

Usage:

```
printReport(generations, fitMean, fitVar)
```

Arguments:

generations numeric. Number of iterations.
fitMean numeric. Array with the mean fitness of every generation.
fitVar numeric. Array with the variance fitness of every generation.

Value:

This function has no return value.

Author:

Ana C. M. Ciconelle
ciconella@gmail.com
ana.ciconelle@usp.br

Funções Fitness/Exemplos

Exemplo 1

fitSum() package:none R Documentation

Calculates fitness.

Description:

Sum the number of ones in a chromosome. Given a population of chromosomes of 0's and 1's, the GA search for the generation in which a chromosome of all ones is discovered and the generation in which all chromosomes are a string of all ones.

Arguments:

pop numeric. A matrix where each line is chromosome(string of 0's and 1's).

Usage:

fitSum(pop)

Author:

Ana C. M. Ciconelle
ciconella@gmail.com
ana.ciconelle@usp.br

References:

Mitchell, Melanie. An Introduction to Genetic Algorithms. First MIT Press paperback edition, 1998.

Example:

See in genAlgo() help.

Exemplo 2

fitBinary() package:none R Documentation

Calculates fitness.

Description:

Fitness is the integer represented by the binary number of the chromosome. Given a population of chromosomes of 0's and 1's, the GA search for the generation in which a chromosome of all ones is discovered and the generation in which all chromosomes are a string of all ones.

Arguments:

pop numeric. A matrix where each line is chromosome(string of 0's and 1's).

Usage:

fitBinary(pop)

Author:

Ana C. M. Ciconelle
cicconella@gmail.com
ana.ciconelle@usp.br

References:

Mitchell, Melanie. An Introduction to Genetic Algorithms. First MIT Press paperback edition, 1998.

Example:

See in genAlgo() help.

Exemplo 3

fitFindSeq() package:none R Documentation

Calculates fitness.

Description:

Given a population of chromosomes of 0's and 1's, in this case: c(0,1,1,1,0), the GA search for the generation in which a chromosome with this given sequence is discovered and the generation in which all chromosomes are this string.

Fitness is calculated based on the difference between the chromosomes of the population and the given one.

Arguments:

pop numeric. A matrix where each line is chromosome(string of 0's and 1's).

Usage:

```
fitFindSeq(pop)
```

Author:

Ana C. M. Ciconelle
ciconella@gmail.com
ana.ciconelle@usp.br

References:

Mitchell, Melanie. An Introduction to Genetic Algorithms. First MIT Press paperback edition, 1998.

Example:

See in `genAlgo()` help.

Exemplo 4

fitY() package:none R Documentation

Calculates fitness.

Description:

Given a population of chromosomes of 0's and 1's, the GA search for the number that maximizes $f(y) = y * \sin(y)$, where $0 < y < \pi$.

Arguments:

pop numeric. A matrix where each line is chromosome(string of 0's and 1's).

Usage:

```
fitBinary(pop)
```

Author:

Ana C. M. Ciconelle
ciconella@gmail.com
ana.ciconelle@usp.br

References:

Mitchell, Melanie. An Introduction to Genetic Algorithms. First MIT Press paperback edition, 1998.

Example:

See in `genAlgo()` help.

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2013:alunos:trabalho_final:ana.ciconelle:help 

Last update: **2020/08/12 06:04**