Módulo Python de Algoritmos Genéticos

ORIGEN DE LA IDEA

Ausencia de una Librería

Facilidad de uso

Adaptable a diferentes problemas

Objetivos:

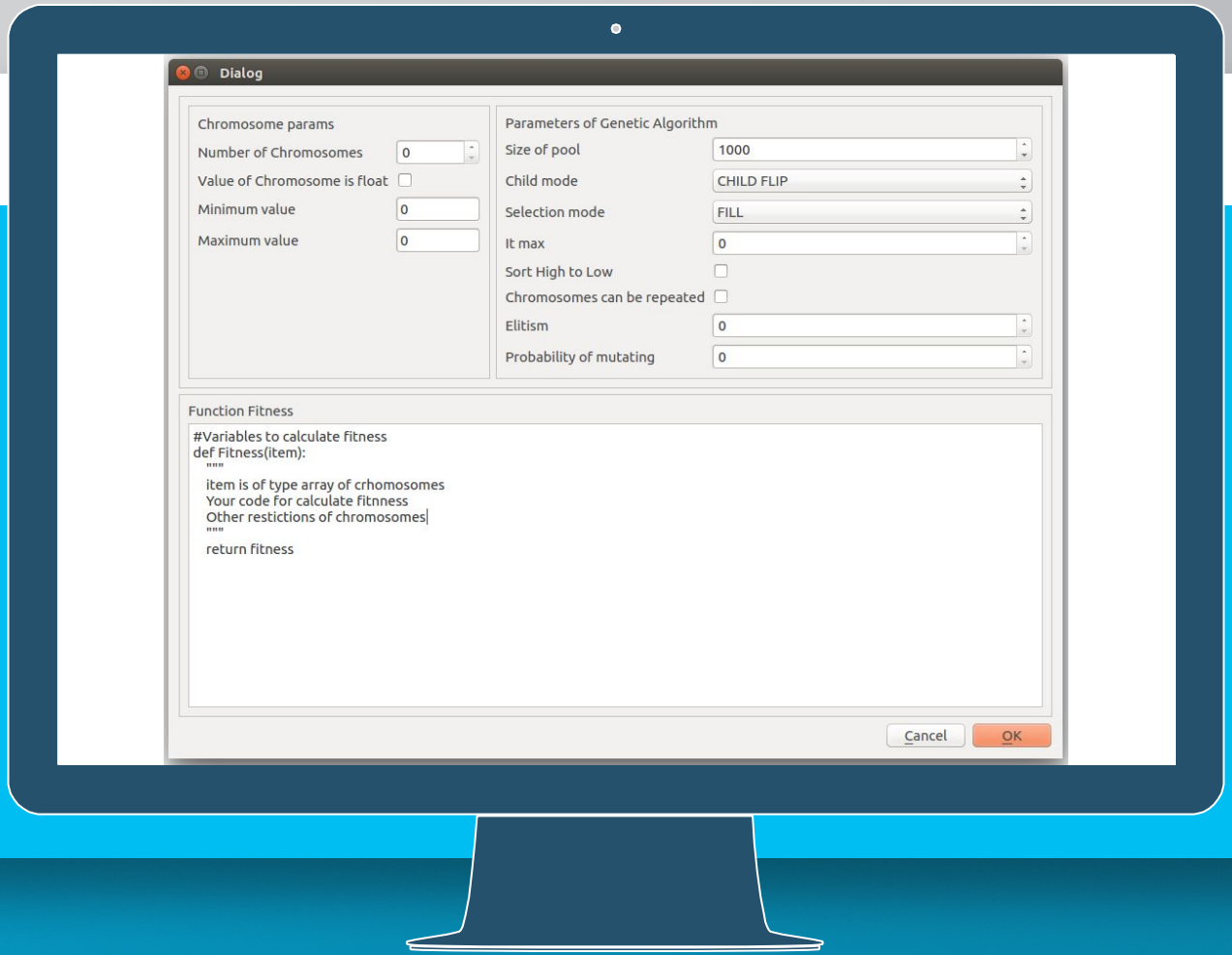
1.-Configurable

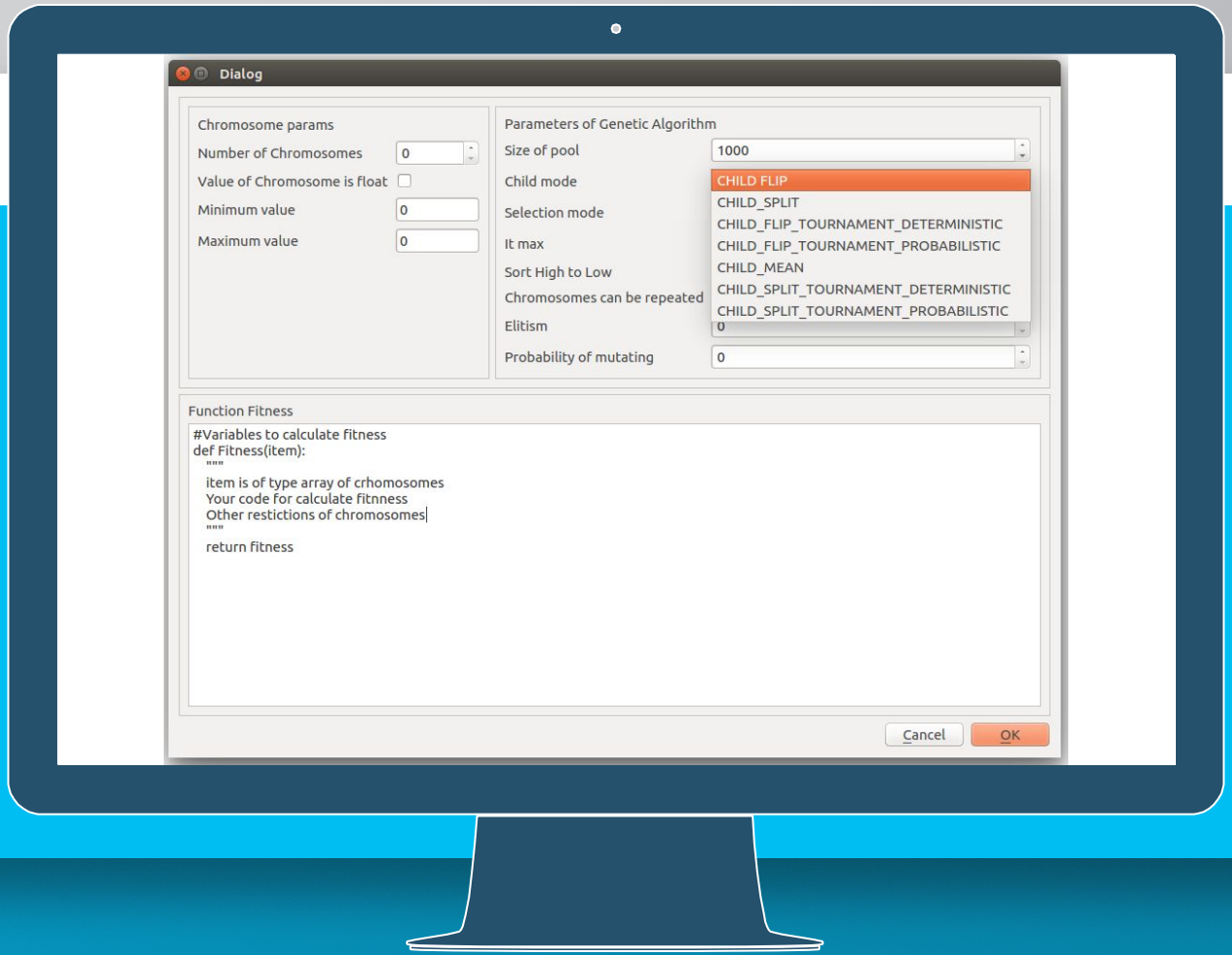
2.-Código compacto

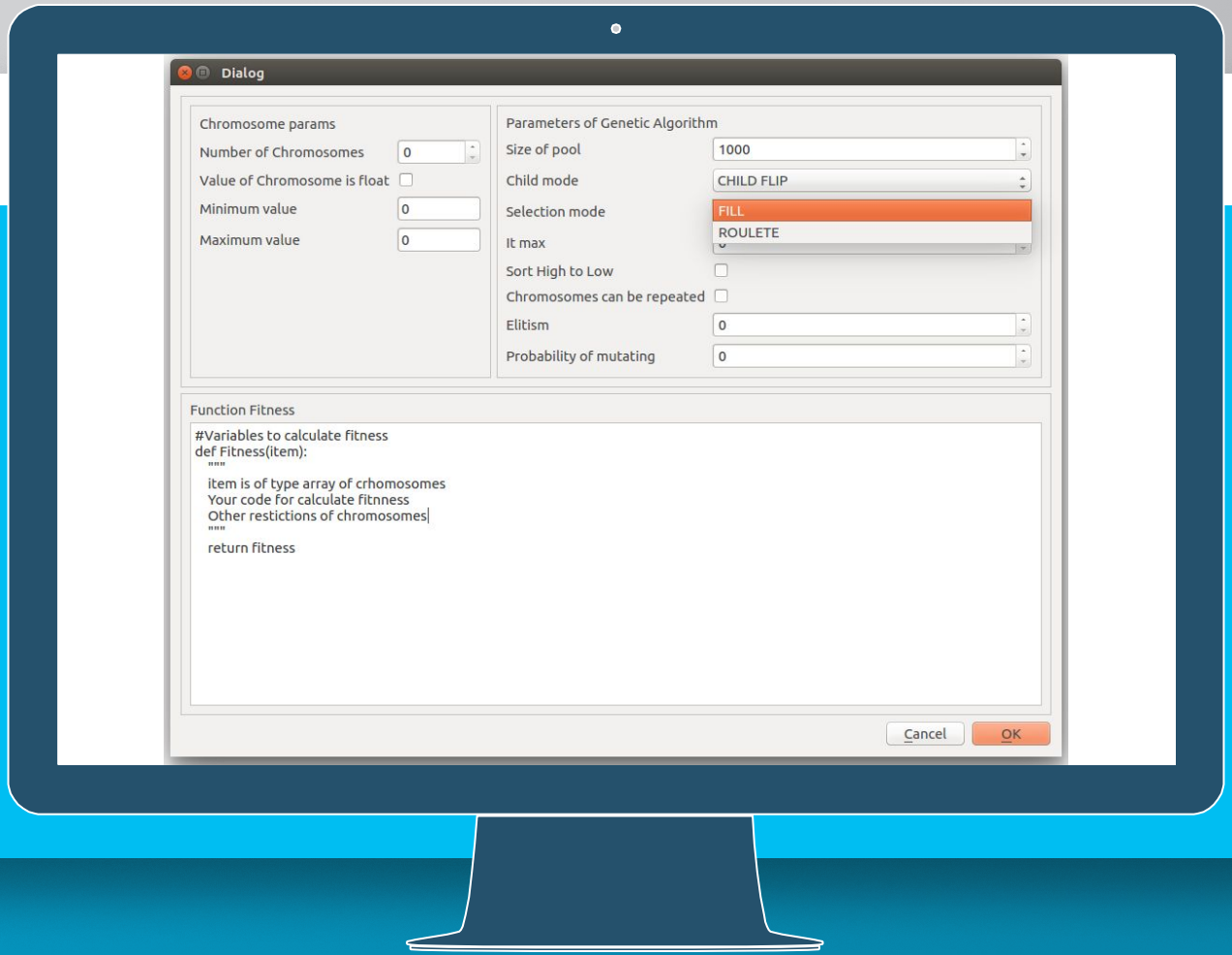
3.-Fácil instalar y usar

4.-Aplicable a diferentes problemas

5.-Funcional


Dialog
Chromosome params
Number of Chromosomes
0
Value of Chromosome is float
Minimum value
0
Maximum value
0
Parameters of Genetic Algorithm
Size of pool
1000
Child mode
CHILD FLIP
Selection mode
FILL
It max
0
Sort High to Low
Chromosomes can be repeated
Elitism
0
Probability of mutating
0
Function Fitness
#Variables to calculate fitness
def Fitness(item):
"""
item is of type array of crhomosomes
Your code for calculate fitnness
Other restictions of chromosomes
"""
return fitness
Cancel
OK


Dialog
Chromosome params
Number of Chromosomes
0
Value of Chromosome is float
Minimum value
0
Maximum value
0
Parameters of Genetic Algorithm
Size of pool
1000
Child mode
CHILD FLIP
CHILD_SPLIT
CHILD_FLIP_TOURNAMENT_DETERMINISTIC
CHILD_FLIP_TOURNAMENT_PROBABILISTIC
CHILD_MEAN
CHILD_SPLIT_TOURNAMENT_DETERMINISTIC
CHILD_SPLIT_TOURNAMENT_PROBABILISTIC
Selection mode
It max
Sort High to Low
Chromosomes can be repeated
Elitism
0
Probability of mutating
0
Function Fitness
#Variables to calculate fitness
def Fitness(item):
"""
item is of type array of crhomosomes
Your code for calculate fitnness
Other restictions of chromosomes
"""
return fitness
Cancel
OK


Dialog
Chromosome params
Number of Chromosomes
0
Value of Chromosome is float
Minimum value
0
Maximum value
0
Parameters of Genetic Algorithm
Size of pool
1000
Child mode
CHILD FLIP
Selection mode
FILL
ROULETE
It max
Sort High to Low
Chromosomes can be repeated
Elitism
0
Probability of mutating
0
Function Fitness
#Variables to calculate fitness
def Fitness(item):
"""
item is of type array of crhomosomes
Your code for calculate fitnness
Other restictions of chromosomes
"""
return fitness
Cancel
OK

Modos de generar hijos:

- Child_flip
- Child_flip_tournamet_deterministic
- Child_flip_tournamet_random
- Child_split
- Child_split_tournamet_deterministic
- Child_split_tournamet_random
- Child_mean

1. Child_flip

“

Padre[1, 2, 3, 4, 5, 6, 7, 8, 9]

Hijo [1, 2, 6, 5, 4, 3, 7, 8, 9]

2. Child_flip_tournamet_deterministic

“

Padre[1, 2, 3, 4, 5, 6, 7, 8, 9]

Hijo 1 [1, 2, 6, 5, 4, 3, 7, 8, 9]

Hijo 2 [2, 1, 3, 4, 5, 6, 9, 8, 7]

3. Child_flip_tournamet_random

“

Padre[1, 2, 3, 4, 5, 6, 7, 8, 9]

Hijo 1 [1, 2, 6, 5, 4, 3, 7, 8, 9]

Hijo 2 [2, 1, 3, 4, 5, 6, 9, 8, 7]

4. Child_split

“

Padre[1, 2, 3, 4, 5, 6, 7, 8, 9]

Padre[9, 8, 7, 6, 5, 4, 3, 2, 1]

Hijo 1 [1, 2, 3, 6, 5, 4, 3, 2, 1]

Hijo 2 [9, 8, 7, 4, 5, 6, 7, 8, 9]

5. Child_split_tournamet_deterministic

“

Padre[1, 2, 3, 4, 5, 6, 7, 8, 9]

Padre[9, 8, 7, 6, 5, 4, 3, 2, 1]

Hijo 1 [1, 2, 3, 6, 5, 4, 3, 2, 1]

Hijo 2 [9, 8, 7, 4, 5, 6, 7, 8, 9]

6. Child_split_tournamet_random

“

Padre[1, 2, 3, 4, 5, 6, 7, 8, 9]

Padre[9, 8, 7, 6, 5, 4, 3, 2, 1]

Hijo 1 [1, 2, 3, 6, 5, 4, 3, 2, 1]

Hijo 2 [9, 8, 7, 4, 5, 6, 7, 8, 9]

7. Child_mean

“

Padre 1 [1, 5, 7, 6]

Padre 2 [3, 8, 1, 3]

Hijo [2, 6, 4, 4]

Mutaciones

Original [1, 2, 3, 4, 5, 6, 7, 8, 9]

Final [1, 2, 3, 8, 5, 6, 7, 4, 9]

“

Confi. 1 : child flip, fill next gen

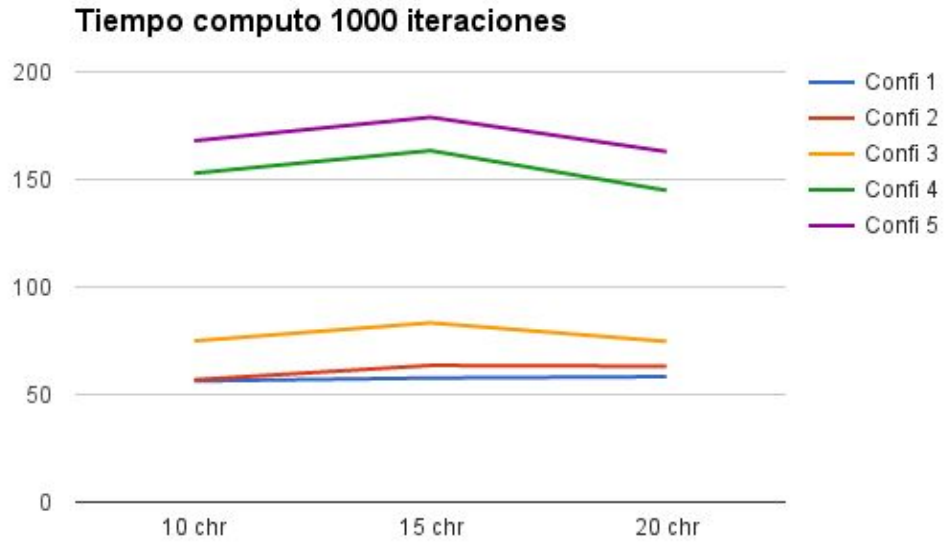
Confi. 2: child split, fill next gen

Confi. 3: child flip torneo det, fill next gen

Confi. 4: child flip, roulette

Confi. 5: child flip torneo det, roulette

Tiempo computo diferentes configuraciones



Resultados

	1000	5000
10 Cromosomas	304	304
15 Cromosomas	263	255
20 Cromosomas	243.3	236

¿Alguna pregunta?

https://github.com/ibarbech/GeneticAlgorithmLibraryPython