
evopy

Release 0.2.2

Jun 01, 2019

Contents:

1	evopy package	3
1.1	Subpackages	3
1.2	Submodules	3
1.3	evopy.evopy module	3
1.4	evopy.individual module	4
1.5	evopy.progress_report module	5
1.6	evopy.strategy module	5
1.7	Module contents	6
2	Indices and tables	7
	Python Module Index	9
	Index	11

evopy: Evolutionary Strategies made simple!

1.1 Subpackages

1.1.1 evopy.utils package

Submodules

evopy.utils.random module

A utility module for random number generation.

`evopy.utils.random.random_with_seed(seed)`
Return RandomState instances based on given seed.

Parameters `seed` – the seed to use for the random number generator

Module contents

The evopy evolutionary strategy algorithm package utility package.

1.2 Submodules

1.3 evopy.evopy module

Module used for the execution of the evolutionary algorithm.

```
class evopy.evopy.EvoPy (fitness_function, individual_length, warm_start=None, generations=100,
                        population_size=30, num_children=1, mean=0, std=1, maximize=False,
                        strategy=<Strategy.SINGLE_VARIANCE: 1>, random_seed=None, reporter=None,
                        target_fitness_value=None, max_run_time=None)
```

Bases: object

Main class of the EvoPy package.

```
__init__ (fitness_function, individual_length, warm_start=None, generations=100, population_size=30,
          num_children=1, mean=0, std=1, maximize=False, strategy=<Strategy.SINGLE_VARIANCE: 1>,
          random_seed=None, reporter=None, target_fitness_value=None, max_run_time=None)
```

Initializes an EvoPy instance.

Parameters

- **fitness_function** – the fitness function on which the individuals are evaluated
- **individual_length** – the length of each individual
- **warm_start** – the individual to start from
- **generations** – the number of generations to execute
- **population_size** – the population size of each generation
- **num_children** – the number of children generated per parent individual
- **mean** – the mean for sampling the random offsets of the initial population
- **std** – the standard deviation for sampling the random offsets of the initial population
- **maximize** – whether the fitness function should be maximized or minimized
- **strategy** – the strategy used to generate offspring by individuals. For more information, check the Strategy enum
- **random_seed** – the seed to use for the random number generator
- **reporter** – callback to be invoked at each generation with a ProgressReport as argument
- **target_fitness_value** – target fitness value for early stopping
- **max_run_time** – maximum time allowed to run in seconds

run()

Run the evolutionary strategy algorithm.

Returns the best genotype found

1.4 evopy.individual module

Module containing the individuals of the evolutionary strategy algorithm.

```
class evopy.individual.Individual (genotype, strategy, strategy_parameters, random_seed=None)
```

Bases: object

The individual of the evolutionary strategy algorithm.

This class handles the reproduction of the individual, using both the genotype and the specified strategy.

For the full variance reproduction strategy, we adopt the implementation as described in: [1] Schwefel, Hans-Paul. (1995). Evolution Strategies I: Variants and their computational

implementation. G. Winter, J. Perieaux, M. Gala, P. Cuesta (Eds.), Proceedings of Genetic Algorithms in Engineering and Computer Science, John Wiley & Sons.

__init__ (*genotype, strategy, strategy_parameters, random_seed=None*)

Initialize the Individual.

Parameters

- **genotype** – the genotype of the individual
- **strategy** – the strategy chosen to reproduce. See the Strategy enum for more information
- **strategy_parameters** – the parameters required for the given strategy, as a list

evaluate (*fitness_function*)

Evaluate the genotype of the individual using the provided fitness function.

Parameters **fitness_function** – the fitness function to evaluate the individual with

Returns the value of the fitness function using the individuals genotype

1.5 evopy.progress_report module

Module containing the ProgressReport class, used to report on the progress of the optimizer.

class `evopy.progress_report.ProgressReport` (*generation, best_genotype, best_fitness*)

Bases: `object`

Class representing a report on an intermediate state of the learning process.

__init__ (*generation, best_genotype, best_fitness*)

Initializes the report instance.

Parameters

- **generation** – number identifying the reported generation
- **best_genotype** – the genotype of the best individual of that generation
- **best_fitness** – the fitness of the best individual of that generation

1.6 evopy.strategy module

Module containing enum Strategy describing different strategies available to the algorithm.

class `evopy.strategy.Strategy`

Bases: `enum.Enum`

Enum used to distinguish different types of strategies.

These strategies are used to determine the mechanism which each individual can use to control its own mutability. The three strategies which are included are:

- **SINGLE_VARIANCE**: the same variance is used for each allele, no covariances
- **MULTIPLE_VARIANCE**: each allele has its own variance, no covariances
- **FULL_VARIANCE**: each allele has its own variance, complete variances (encoded as rotation angles)

FULL_VARIANCE = 3

```
MULTIPLE_VARIANCE = 2  
SINGLE_VARIANCE = 1
```

1.7 Module contents

The evopy evolutionary strategy algorithm package.

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

e

- `evopy`, 6
- `evopy.evopy`, 3
- `evopy.individual`, 4
- `evopy.progress_report`, 5
- `evopy.strategy`, 5
- `evopy.utils`, 3
- `evopy.utils.random`, 3

Symbols

`__init__()` (*evopy.evopy.EvoPy method*), 4
`__init__()` (*evopy.individual.Individual method*), 5
`__init__()` (*evopy.progress_report.ProgressReport method*), 5

E

`evaluate()` (*evopy.individual.Individual method*), 5
`EvoPy` (*class in evopy.evopy*), 3
`evopy` (*module*), 6
`evopy.evopy` (*module*), 3
`evopy.individual` (*module*), 4
`evopy.progress_report` (*module*), 5
`evopy.strategy` (*module*), 5
`evopy.utils` (*module*), 3
`evopy.utils.random` (*module*), 3

F

`FULL_VARIANCE` (*evopy.strategy.Strategy attribute*), 5

I

`Individual` (*class in evopy.individual*), 4

M

`MULTIPLE_VARIANCE` (*evopy.strategy.Strategy attribute*), 5

P

`ProgressReport` (*class in evopy.progress_report*), 5

R

`random_with_seed()` (*in evopy.utils.random module*), 3
`run()` (*evopy.evopy.EvoPy method*), 4

S

`SINGLE_VARIANCE` (*evopy.strategy.Strategy attribute*), 6
`Strategy` (*class in evopy.strategy*), 5