

## Differential evolution algorithm

Generated by Doxygen 1.8.17



<b>1 Namespace Index</b>	<b>1</b>
1.1 Namespace List	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Namespace Documentation</b>	<b>7</b>
4.1 func Namespace Reference	7
4.1.1 Detailed Description	7
4.1.2 Function Documentation	7
4.1.2.1 ackleysOne()	7
4.1.2.2 ackleysTwo()	8
4.1.2.3 alpine()	8
4.1.2.4 eggHolder()	8
4.1.2.5 firstDeJong()	9
4.1.2.6 griewangk()	9
4.1.2.7 levy()	9
4.1.2.8 mastersCosineWave()	9
4.1.2.9 michalewicz()	10
4.1.2.10 pathological()	10
4.1.2.11 quartic()	10
4.1.2.12 rana()	11
4.1.2.13 rastrigin()	11
4.1.2.14 rosenbrock()	11
4.1.2.15 schwefel()	11
4.1.2.16 sineEnvelopeSineWave()	12
4.1.2.17 step()	12
4.1.2.18 stretchedVSineWave()	12
<b>5 Class Documentation</b>	<b>13</b>
5.1 DataStats Class Reference	13
5.2 DE Class Reference	13
5.2.1 Constructor & Destructor Documentation	13
5.2.1.1 DE()	14
5.2.2 Member Function Documentation	15
5.2.2.1 run()	15
5.3 DEStrategy Struct Reference	15
5.4 Individual Class Reference	16
5.4.1 Detailed Description	16
5.5 Population Class Reference	16
5.5.1 Detailed Description	17

---

5.5.2 Member Function Documentation . . . . .	17
5.5.2.1 init() . . . . .	17
<b>6 File Documentation</b>	<b>19</b>
6.1 src/data_stats.h File Reference . . . . .	19
6.1.1 Detailed Description . . . . .	19
6.2 src/differential_evolution.h File Reference . . . . .	19
6.2.1 Detailed Description . . . . .	20
6.3 src/func.h File Reference . . . . .	20
6.3.1 Detailed Description . . . . .	21
6.4 src/population.h File Reference . . . . .	21
6.4.1 Detailed Description . . . . .	21
6.5 src/run.h File Reference . . . . .	22
6.5.1 Detailed Description . . . . .	22
6.5.2 Function Documentation . . . . .	22
6.5.2.1 output_all() . . . . .	22
6.5.2.2 output_func() . . . . .	23
6.5.2.3 runFunc() . . . . .	23
6.5.2.4 runStrategy() . . . . .	24
<b>Index</b>	<b>25</b>

# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

<a href="#">func</a>	18 functions . . . . .	<a href="#">7</a>
----------------------	------------------------	-------------------



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">DataStats</a> . . . . .	13
<a href="#">DE</a> . . . . .	13
<a href="#">DEStrategy</a> . . . . .	15
<a href="#">Individual</a>	
<a href="#">Individual</a> for population . . . . .	16
<a href="#">Population</a>	
<a href="#">Population</a> for genetic algorithm . . . . .	16





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

src/ <a href="#">data_stats.h</a>	
Data analysis . . . . .	19
src/ <a href="#">differential_evolution.h</a>	
Differential evolution algorithm . . . . .	19
src/ <a href="#">func.h</a>	
Math functions . . . . .	20
src/ <a href="#">population.h</a>	
Population generation . . . . .	21
src/ <a href="#">run.h</a>	
Run . . . . .	22



## Chapter 4

# Namespace Documentation

### 4.1 func Namespace Reference

18 functions

#### Functions

- float [schwefel](#) (vector< float > &x)
- float [firstDeJong](#) (vector< float > &x)
- float [rosenbrock](#) (vector< float > &x)
- float [rastrigin](#) (vector< float > &x)
- float [griewangk](#) (vector< float > &x)
- float [sineEnvelopeSineWave](#) (vector< float > &x)
- float [stretchedVSineWave](#) (vector< float > &x)
- float [ackleysOne](#) (vector< float > &x)
- float [ackleysTwo](#) (vector< float > &x)
- float [eggHolder](#) (vector< float > &x)
- float [rana](#) (vector< float > &x)
- float [pathological](#) (vector< float > &x)
- float [michalewicz](#) (vector< float > &x)
- float [mastersCosineWave](#) (vector< float > &x)
- float [quartic](#) (vector< float > &x)
- float [levy](#) (vector< float > &x)
- float [step](#) (vector< float > &x)
- float [alpine](#) (vector< float > &x)

#### 4.1.1 Detailed Description

18 functions

#### 4.1.2 Function Documentation

##### 4.1.2.1 ackleysOne()

```
float func::ackleysOne (
    vector< float > & x )
```

Function 8, Implementation of Ackley's One function

**Parameters**

<i>x</i>	descriptionx Vector of float
----------	------------------------------

**4.1.2.2 ackleysTwo()**

```
float func::ackleysTwo (
    vector< float > & x )
```

Function 9, Implementation of Ackley's Two function

**Parameters**

<i>x</i>	descriptionx Vector of float
----------	------------------------------

**4.1.2.3 alpine()**

```
float func::alpine (
    vector< float > & x )
```

Function 18, Implementation of Alpine function

**Parameters**

<i>x</i>	descriptionx Vector of float
----------	------------------------------

**4.1.2.4 eggHolder()**

```
float func::eggHolder (
    vector< float > & x )
```

Function 10, Implementation of Egg Holder function

**Parameters**

<i>x</i>	descriptionx Vector of float
----------	------------------------------

#### 4.1.2.5 firstDeJong()

```
float func::firstDeJong (
    vector< float > & x )
```

Function 2, Implementation of 1st De Jong's function

##### Parameters

x	descriptionx Vector of float
---	------------------------------

#### 4.1.2.6 griewangk()

```
float func::griewangk (
    vector< float > & x )
```

Function 5, Implementation of Griewangk function

##### Parameters

x	descriptionx Vector of float
---	------------------------------

#### 4.1.2.7 levy()

```
float func::levy (
    vector< float > & x )
```

Function 16, Implementation of Levy function

##### Parameters

x	descriptionx Vector of float
---	------------------------------

#### 4.1.2.8 mastersCosineWave()

```
float func::mastersCosineWave (
    vector< float > & x )
```

Function 14, Implementation of Masters Cosine Wave function

**Parameters**

<i>x</i>	descriptionx Vector of float
----------	------------------------------

**4.1.2.9 michalewicz()**

```
float func::michalewicz (  
    vector< float > & x )
```

Function 13, Implementation of Michalewicz function

**Parameters**

<i>x</i>	descriptionx Vector of float
----------	------------------------------

**4.1.2.10 pathological()**

```
float func::pathological (  
    vector< float > & x )
```

Function 12, Implementation of Pathological function

**Parameters**

<i>x</i>	descriptionx Vector of float
----------	------------------------------

**4.1.2.11 quartic()**

```
float func::quartic (  
    vector< float > & x )
```

Function 15, Implementation of Quartic function

**Parameters**

<i>x</i>	descriptionx Vector of float
----------	------------------------------

#### 4.1.2.12 rana()

```
float func::rana (
    vector< float > & x )
```

Function 11, Implementation of Rana function

##### Parameters

x	descriptionx Vector of float
---	------------------------------

#### 4.1.2.13 rastrigin()

```
float func::rastrigin (
    vector< float > & x )
```

Function 4, Implementation of Rastrigin's function

##### Parameters

x	descriptionx Vector of float
---	------------------------------

#### 4.1.2.14 rosenbrock()

```
float func::rosenbrock (
    vector< float > & x )
```

Function 3, Implementation of Rosenbrock's function

##### Parameters

x	descriptionx Vector of float
---	------------------------------

#### 4.1.2.15 schwefel()

```
float func::schwefel (
    vector< float > & x )
```

Function 1, Implementation of Schwefel's function

**Parameters**

<i>x</i>	descriptionx Vector of float
----------	------------------------------

**4.1.2.16 sineEnvelopeSineWave()**

```
float func::sineEnvelopeSineWave (  
    vector< float > & x )
```

Function 6, Implementation of Sine Envelope Sine Wave function

**Parameters**

<i>x</i>	descriptionx Vector of float
----------	------------------------------

**4.1.2.17 step()**

```
float func::step (  
    vector< float > & x )
```

Function 147 Implementation of Step function

**Parameters**

<i>x</i>	descriptionx Vector of float
----------	------------------------------

**4.1.2.18 stretchedVSineWave()**

```
float func::stretchedVSineWave (  
    vector< float > & x )
```

Function 7, Implementation of Stretched V Sine Wave function

**Parameters**

<i>x</i>	descriptionx Vector of float
----------	------------------------------



## Chapter 5

# Class Documentation

### 5.1 DataStats Class Reference

#### Public Member Functions

- void [run](#) ()  
*Generate analytical data.*

#### Public Attributes

- float **mean** = INT\_MAX
- float **median** = INT\_MAX
- float **stand** = INT\_MAX
- float **range** [2]
- float **time** = 0
- vector< float > **data**

The documentation for this class was generated from the following files:

- src/[data\\_stats.h](#)
- src/data\_stats.cpp

### 5.2 DE Class Reference

#### Public Member Functions

- [DE](#) ([DEStrategy](#) s, float(\*f)(vector< float > &), float l, float u)
- vector< float > [run](#) ()

#### 5.2.1 Constructor & Destructor Documentation

### 5.2.1.1 DE()

```
DE::DE (
    DEStrategy s,
    float(*) (vector< float > &) f,
    float l,
    float u )
```

Initialize differential evolution algorithm to run for one function

## Parameters

<i>f</i>	function
<i>l</i>	low bound for x
<i>u</i>	high bound for x

## 5.2.2 Member Function Documentation

### 5.2.2.1 run()

```
vector< float > DE::run ( )
```

run differential evolution algorithm

## Parameters

<i>i</i>	ith individual
<i>j</i>	jth element in individual

## Returns

history values of best function value achieved

The documentation for this class was generated from the following files:

- [src/differential\\_evolution.h](#)
- [src/differential\\_evolution.cpp](#)

## 5.3 DEStrategy Struct Reference

### Public Attributes

- string **perturbed\_vector**
- int [difference\\_vector](#)  
*x: string denoting the vector to be perturbed, choosing from best, rand or rand-to-best*
- string [crossover\\_type](#)  
*y: number of difference vector considered for perturbation of x, 1 or 2*

The documentation for this struct was generated from the following file:

- [src/differential\\_evolution.h](#)

## 5.4 Individual Class Reference

[Individual](#) for population.

```
#include <population.h>
```

### Public Member Functions

- **Individual** (float(\*f)(vector< float > &), vector< float > numbers)

### Public Attributes

- vector< float > **data**
- float(\* **function** )(vector< float > &)
- float **cost**

#### 5.4.1 Detailed Description

[Individual](#) for population.

The documentation for this class was generated from the following files:

- src/[population.h](#)
- src/population.cpp

## 5.5 Population Class Reference

[Population](#) for genetic algorithm.

```
#include <population.h>
```

### Public Member Functions

- void [init](#) (int s, int d, float(\*f)(vector< float > &), float l, float u)
- void [reset](#) ()

*reset the population for new run*

### Public Attributes

- vector< [Individual](#) > **data**  
*Initialize a population.*
- int **individual\_best**

### 5.5.1 Detailed Description

[Population](#) for genetic algorithm.

### 5.5.2 Member Function Documentation

#### 5.5.2.1 init()

```
void Population::init (
    int s,
    int d,
    float (*) (vector< float > &) f,
    float l,
    float u )
```

Initialize a population

##### Parameters

<i>s</i>	population size
<i>d</i>	population dimension
<i>low</i>	x low bound
<i>high</i>	x high bound

The documentation for this class was generated from the following files:

- [src/population.h](#)
- [src/population.cpp](#)



## Chapter 6

# File Documentation

### 6.1 src/data\_stats.h File Reference

data analysis

```
#include <vector>
```

#### Classes

- class [DataStats](#)

#### 6.1.1 Detailed Description

data analysis

##### Author

Cheng Su ( [csu@cwu.edu](mailto:csu@cwu.edu) )

##### Version

1.0

##### Date

2020-02-11

### 6.2 src/differential\_evolution.h File Reference

differential evolution algorithm

```
#include <string>
#include <vector>
#include "population.h"
```

## Classes

- struct [DEStrategy](#)
- class [DE](#)

### 6.2.1 Detailed Description

differential evolution algorithm

#### Author

Cheng Su ( [csu@cwu.edu](mailto:csu@cwu.edu) )

#### Version

1.0

#### Date

2020-02-11

## 6.3 src/func.h File Reference

Math functions.

```
#include <stdio.h>
#include <vector>
```

## Namespaces

- [func](#)  
*18 functions*

## Functions

- float [func::schwefel](#) (vector< float > &x)
- float [func::firstDeJong](#) (vector< float > &x)
- float [func::rosenbrock](#) (vector< float > &x)
- float [func::rastrigin](#) (vector< float > &x)
- float [func::griewangk](#) (vector< float > &x)
- float [func::sineEnvelopeSineWave](#) (vector< float > &x)
- float [func::stretchedVSineWave](#) (vector< float > &x)
- float [func::ackleysOne](#) (vector< float > &x)
- float [func::ackleysTwo](#) (vector< float > &x)
- float [func::eggHolder](#) (vector< float > &x)
- float [func::rana](#) (vector< float > &x)
- float [func::pathological](#) (vector< float > &x)
- float [func::michalewicz](#) (vector< float > &x)
- float [func::mastersCosineWave](#) (vector< float > &x)
- float [func::quartic](#) (vector< float > &x)
- float [func::levy](#) (vector< float > &x)
- float [func::step](#) (vector< float > &x)
- float [func::alpine](#) (vector< float > &x)



### 6.3.1 Detailed Description

Math functions.

Author

Cheng Su ( [csu@cwu.edu](mailto:csu@cwu.edu) )

Version

0.1

Date

2020-02-05

## 6.4 src/population.h File Reference

[Population](#) generation.

```
#include <stdio.h>
#include <vector>
```

### Classes

- class [Individual](#)  
*Individual for population.*
- class [Population](#)  
*Population for genetic algorithm.*

### 6.4.1 Detailed Description

[Population](#) generation.

Author

Cheng Su ( [csu@cwu.edu](mailto:csu@cwu.edu) )

Version

1.0

Date

2020-02-1

## 6.5 src/run.h File Reference

run

```
#include <vector>
#include "data_stats.h"
#include "differential_evolution.h"
```

### Functions

- void [setSeed](#) ()  
*Set seed for [Mersenne Twister](#)*
- [DataStats runFunc](#) (int experiment, string func\_name, float(\*f)(vector< float > &), float l, float u)
- [DataStats runStrategy](#) ([DEStrategy](#) strat, int experiment, float(\*f)(vector< float > &), float l, float u)  
*Run genetic algorithm for a certain function.*
- void [output\\_func](#) (string func\_name, vector< [DataStats](#) > result)
- void [output\\_all](#) (vector< [DataStats](#) > result\_best)

### 6.5.1 Detailed Description

run

Author

Cheng Su ( [csu@cwu.edu](mailto:csu@cwu.edu) )

Version

1.0

Date

2020-02-11

### 6.5.2 Function Documentation

#### 6.5.2.1 output\_all()

```
void output_all (
    vector< DataStats > result_best )
```

write best result for every function

## Parameters

<i>result_best</i>	best result for each function
--------------------	-------------------------------

## 6.5.2.2 output\_func()

```
void output_func (
    string func_name,
    vector< DataStats > result )
```

write result for one function for all strategies

## Parameters

<i>func_name</i>	function name
<i>result</i>	result of all strategies

## 6.5.2.3 runFunc()

```
DataStats runFunc (
    int experiment,
    string func_name,
    float(*) (vector< float > &) f,
    float l,
    float u )
```

Run genetic algorithm for a certain function

## Parameters

<i>experiment</i>	experiment size
<i>func_name</i>	function name
<i>min</i>	low bound for x
<i>max</i>	high bound for x

## Returns

best result among 10 strategies

strategy 1 best/1/exp

strategy 2 rand/1/exp

strategy 3 rand-to-best/1/exp

strategy 4 best/2/exp

strategy 5 rand/2/exp

strategy 6 best/1/bin

strategy 7 rand/1/bin

strategy 8 rand-to-best/1/bin

strategy 9 best/2/bin

strategy 10 rand/2/bin

#### 6.5.2.4 runStrategy()

```
DataStats runStrategy (
    DEStrategy strat,
    int experiment,
    float(*) (vector< float > &) f,
    float l,
    float u )
```

Run genetic algorithm for a certain function.

run one strategy for a function

##### Parameters

<i>strat</i>	strategy
<i>experiment</i>	number of experiment
<i>f</i>	function
<i>l</i>	low x bound
<i>u</i>	up x bound

##### Returns

return result analysis

# Index

- ackleysOne
  - func, [7](#)
- ackleysTwo
  - func, [8](#)
- alpine
  - func, [8](#)
- DataStats, [13](#)
- DE, [13](#)
  - DE, [13](#)
  - run, [15](#)
- DEStrategy, [15](#)
- eggHolder
  - func, [8](#)
- firstDeJong
  - func, [8](#)
- func, [7](#)
  - ackleysOne, [7](#)
  - ackleysTwo, [8](#)
  - alpine, [8](#)
  - eggHolder, [8](#)
  - firstDeJong, [8](#)
  - griewangk, [9](#)
  - levy, [9](#)
  - mastersCosineWave, [9](#)
  - michalewicz, [10](#)
  - pathological, [10](#)
  - quartic, [10](#)
  - rana, [10](#)
  - rastrigin, [11](#)
  - rosenbrock, [11](#)
  - schwefel, [11](#)
  - sineEnvelopeSineWave, [12](#)
  - step, [12](#)
  - stretchedVSineWave, [12](#)
- griewangk
  - func, [9](#)
- Individual, [16](#)
- init
  - Population, [17](#)
- levy
  - func, [9](#)
- mastersCosineWave
  - func, [9](#)
- michalewicz
  - func, [10](#)
- output\_all
  - run.h, [22](#)
- output\_func
  - run.h, [23](#)
- pathological
  - func, [10](#)
- Population, [16](#)
  - init, [17](#)
- quartic
  - func, [10](#)
- rana
  - func, [10](#)
- rastrigin
  - func, [11](#)
- rosenbrock
  - func, [11](#)
- run
  - DE, [15](#)
- run.h
  - output\_all, [22](#)
  - output\_func, [23](#)
  - runFunc, [23](#)
  - runStrategy, [24](#)
- runFunc
  - run.h, [23](#)
- runStrategy
  - run.h, [24](#)
- schwefel
  - func, [11](#)
- sineEnvelopeSineWave
  - func, [12](#)
- src/data\_stats.h, [19](#)
- src/differential\_evolution.h, [19](#)
- src/func.h, [20](#)
- src/population.h, [21](#)
- src/run.h, [22](#)
- step
  - func, [12](#)
- stretchedVSineWave
  - func, [12](#)