

## Genetic 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 Genetic_algorithm Class Reference	13
5.2.1 Constructor & Destructor Documentation	14
5.2.1.1 Genetic_algorithm()	14
5.2.2 Member Function Documentation	14
5.2.2.1 run()	14
5.3 Population Class Reference	15
5.3.1 Detailed Description	15
5.3.2 Member Function Documentation	15
5.3.2.1 evaluate()	15
5.3.2.2 generation()	15

---

5.3.2.3 init()	15
5.3.2.4 reset()	16
5.3.2.5 sort_by_cost()	16
<b>6 File Documentation</b>	<b>17</b>
6.1 src/data_stats.h File Reference	17
6.1.1 Detailed Description	17
6.2 src/func.h File Reference	17
6.2.1 Detailed Description	18
6.3 src/genetic_algorithm.h File Reference	18
6.3.1 Detailed Description	19
6.4 src/population.h File Reference	19
6.4.1 Detailed Description	19
6.5 src/run.h File Reference	20
6.5.1 Detailed Description	20
6.5.2 Function Documentation	20
6.5.2.1 output()	20
6.5.2.2 output_all()	21
6.5.2.3 runFunc()	21
<b>Index</b>	<b>23</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="#">Genetic_algorithm</a> . . . . .	13
<a href="#">Population</a> . . . . .	15





## 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 . . . . .	17
src/ <a href="#">func.h</a>	
Math functions . . . . .	17
src/ <a href="#">genetic_algorithm.h</a>	
Genetic algorithm . . . . .	18
src/ <a href="#">population.h</a>	
Population generation . . . . .	19
src/ <a href="#">run.h</a>	
Run result . . . . .	20



## 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	description x 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 Genetic\_algorithm Class Reference

#### Public Member Functions

- [Genetic\\_algorithm](#) (float(\*funct)(vector< float > &), float l, float u)  
*Solution sets.*
- void [run](#) ()  
*Initialize genetic algorithm to run for a certain function.*

## Public Attributes

- float **f\_best** = INT\_MAX
- vector< float > **f\_best\_history**
- vector< float > [solution\\_best](#)  
*Minimum f value during execution.*

## 5.2.1 Constructor & Destructor Documentation

### 5.2.1.1 Genetic\_algorithm()

```
Genetic_algorithm::Genetic_algorithm (
    float(*) (vector< float > &) funct,
    float l,
    float u )
```

Solution sets.

Initialize genetic algorithm to run for a certain 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()

```
void Genetic_algorithm::run ( )
```

Initialize genetic algorithm to run for a certain function.

Run genetic algorithm. Status reset before every run.

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

- [src/genetic\\_algorithm.h](#)
- [src/genetic\\_algorithm.cpp](#)

## 5.3 Population Class Reference

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

### Public Member Functions

- void [init](#) (int s, int d, float(\*funct)(vector< float > &), int l, int u)
- void [generation](#) ()  
*Initialize a population.*
- void [evaluate](#) ()  
*Generate random number to fill the population.*
- void [reset](#) ()  
*Evaluate the population the population by first calculating each function's cost, then calculate fitness accordingly.*
- void [sort\\_by\\_cost](#) ()  
*Reset the population for new run.*

### Public Attributes

- vector< vector< float > > **population**
- vector< float > **cost**
- vector< float > **fitness**
- float **fitness\_total**

#### 5.3.1 Detailed Description

[Population](#) for genetic algorithm. Individual's population of x is represented by vector<float>.

#### 5.3.2 Member Function Documentation

##### 5.3.2.1 evaluate()

```
void Population::evaluate ( )
```

Generate random number to fill the population.

Evaluate the population the population by first calculating each function's cost, then calculate fitness accordingly.

##### 5.3.2.2 generation()

```
void Population::generation ( )
```

Initialize a population.

Generate random number to fill the population.

##### 5.3.2.3 init()

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

**5.3.2.4 reset()**

```
void Population::reset ( )
```

Evaluate the population the population by first calculating each function's cost, then calculate fitness accordingly.

reset the population for new run

**5.3.2.5 sort\_by\_cost()**

```
void Population::sort_by_cost ( )
```

Reset the population for new run.

pair sort to sort the population by cost

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

0.1

##### Date

2020-02-05

### 6.2 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)

### 6.2.1 Detailed Description

Math functions.

#### Author

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

#### Version

0.1

#### Date

2020-02-05

## 6.3 src/genetic\_algorithm.h File Reference

Genetic algorithm.

```
#include <stdio.h>
#include <vector>
#include "genetic_algorithm.h"
#include "population.h"
```

## Classes

- class [Genetic\\_algorithm](#)

### 6.3.1 Detailed Description

Genetic algorithm.

#### 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 [Population](#)

### 6.4.1 Detailed Description

[Population](#) generation.

#### Author

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

#### Version

0.1

#### Date

2020-02-05

## 6.5 src/run.h File Reference

Run result.

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

### Functions

- void [setSeed](#) ()  
*Set seed for [Mersenne Twister](#)*
- [DataStats runFunc](#) (int experiment, string func\_name, float(\*f)(vector< float > &), float max, float min)  
*Run genetic algorithm for a certain function.*
- void [output](#) (string func\_name, [DataStats](#) result, vector< vector< float >> f\_best\_history)  
*Write output file for record.*
- void [output\\_all](#) (vector< [DataStats](#) > result\_best)  
*Write output file for all function results.*

### 6.5.1 Detailed Description

Run result.

Author

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

Version

0.1

Date

2020-02-05

### 6.5.2 Function Documentation

#### 6.5.2.1 output()

```
void output (
    string func_name,
    DataStats result,
    vector< vector< float >> f_best_history )
```

Write output file for record.

Write output file for record



## Parameters

<i>func_name</i>	function name
<i>result</i>	result analysis data
<i>f_best_history</i>	minimum value for fx

### 6.5.2.2 output\_all()

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

Write output file for all function results.

Write output file for all function results

## Parameters

<i>result_best</i>	statistics for results for every function
--------------------	---

### 6.5.2.3 runFunc()

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

Run genetic algorithm for a certain function.

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



# Index

- ackleysOne
  - func, [7](#)
- ackleysTwo
  - func, [8](#)
- alpine
  - func, [8](#)
- DataStats, [13](#)
- eggHolder
  - func, [8](#)
- evaluate
  - Population, [15](#)
- 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](#)
- generation
  - Population, [15](#)
- Genetic\_algorithm, [13](#)
  - Genetic\_algorithm, [14](#)
  - run, [14](#)
- griewangk
  - func, [9](#)
- init
  - Population, [15](#)
- levy
  - func, [9](#)
- mastersCosineWave
  - func, [9](#)
- michalewicz
  - func, [10](#)
- output
  - run.h, [20](#)
- output\_all
  - run.h, [21](#)
- pathological
  - func, [10](#)
- Population, [15](#)
  - evaluate, [15](#)
  - generation, [15](#)
  - init, [15](#)
  - reset, [16](#)
  - sort\_by\_cost, [16](#)
- quartic
  - func, [10](#)
- rana
  - func, [10](#)
- rastrigin
  - func, [11](#)
- reset
  - Population, [16](#)
- rosenbrock
  - func, [11](#)
- run
  - Genetic\_algorithm, [14](#)
- run.h
  - output, [20](#)
  - output\_all, [21](#)
  - runFunc, [21](#)
- runFunc
  - run.h, [21](#)
- schwefel
  - func, [11](#)
- sineEnvelopeSineWave
  - func, [12](#)
- sort\_by\_cost
  - Population, [16](#)
- src/data\_stats.h, [17](#)
- src/func.h, [17](#)
- src/genetic\_algorithm.h, [18](#)
- src/population.h, [19](#)
- src/run.h, [20](#)
- step
  - func, [12](#)

stretchedVSineWave  
func, [12](#)