

Differential evolution algorithm

Generated by Doxygen 1.8.17

1 Namespace Index	1
1.1 Namespace List	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Namespace Documentation	7
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
5 Class Documentation	13
5.1 DataStats Class Reference	13
5.2 DE Class Reference	13
5.2.1 Constructor & Destructor Documentation	14
5.2.1.1 DE()	14
5.2.2 Member Function Documentation	14
5.2.2.1 run()	14
5.3 DEStrategy Struct Reference	15
5.4 Population Class Reference	15
5.4.1 Detailed Description	15
5.4.2 Member Function Documentation	15
5.4.2.1 init()	15

6 File Documentation	17
6.1 src/data_stats.h File Reference	17
6.1.1 Detailed Description	17
6.2 src/differential_evolution.h File Reference	17
6.2.1 Detailed Description	18
6.3 src/func.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_all()	20
6.5.2.2 output_fHistory()	21
6.5.2.3 output_func()	21
6.5.2.4 runFunc()	21
6.5.2.5 runStrategy()	22
Index	25

Chapter 1

Namespace Index

1.1 Namespace List

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

func	18 functions	7
----------------------	------------------------	-------------------

Chapter 2

Class Index

2.1 Class List

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

DataStats	13
DE	13
DEStrategy	15
Population	
Population for genetic algorithm	15

Chapter 3

File Index

3.1 File List

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

src/data_stats.h	
Data analysis	17
src/differential_evolution.h	
Differential evolution algorithm	17
src/func.h	
Math functions	18
src/population.h	
Population generation	19
src/run.h	
Run	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

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

4.1.2.2 ackleysTwo()

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

Function 9, Implementation of Ackley's Two function

Parameters

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

4.1.2.3 alpine()

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

Function 18, Implementation of Alpine function

Parameters

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

4.1.2.4 eggHolder()

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

Function 10, Implementation of Egg Holder function

Parameters

x	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

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

4.1.2.9 michalewicz()

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

Function 13, Implementation of Michalewicz function

Parameters

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

4.1.2.10 pathological()

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

Function 12, Implementation of Pathological function

Parameters

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

4.1.2.11 quartic()

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

Function 15, Implementation of Quartic function

Parameters

x	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

x	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

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

4.1.2.17 step()

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

Function 147 Implementation of Step function

Parameters

x	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

x	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`
- float `median`
- float `stand`
- float `range_low`
- float `range_high`
- float `time_avg`
- vector< float > `time`
- 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

reset population

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 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< vector< float > > **data**
Initialize a population.
- vector< float > **cost**
- int **cost_best_index**
- int **cost_best**

5.4.1 Detailed Description

[Population](#) for genetic algorithm.

5.4.2 Member Function Documentation

5.4.2.1 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)

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)

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)

Version

0.1

Date

2020-02-05

6.4 src/population.h File Reference

[Population](#) generation.

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

Classes

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

6.4.1 Detailed Description

[Population](#) generation.

Author

Cheng Su (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](#) (string func_name, string strategy_number, [DEStrategy](#) strat, int experiment, float(*f)(vector< float > &), float l, float u)
Run genetic algorithm for a certain function.
- void [output_fHistory](#) (string func_name, string strategy_number, vector< vector< float > > f_bests_history)
- 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)

Version

1.0

Date

2020-02-11

6.5.2 Function Documentation

6.5.2.1 output_all()

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

Write best result for every function

Parameters

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

6.5.2.2 output_fHistory()

```
void output_fHistory (
    string func_name,
    string strategy_number,
    vector< vector< float >> f_bests_history )
```

Write best result history for each strategy for each function

Parameters

<i>func_name</i>	function name
<i>strategy_number</i>	strategy number
<i>f_bests_history</i>	history of best function value achieved

6.5.2.3 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

6.5.2.4 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

6.5.2.5 runStrategy()

```
DataStats runStrategy (
    string func_name,
    string strategy_number,
    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, [14](#)
 run, [14](#)

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](#)

init
 Population, [15](#)

levy
 func, [9](#)

mastersCosineWave
 func, [9](#)

michalewicz
 func, [10](#)

output_all
 run.h, [20](#)

output_fHistory
 run.h, [21](#)

output_func
 run.h, [21](#)

pathological
 func, [10](#)

Population, [15](#)
 init, [15](#)

quartic
 func, [10](#)

rana
 func, [10](#)

rastrigin
 func, [11](#)

rosenbrock
 func, [11](#)

run
 DE, [14](#)

run.h
 output_all, [20](#)
 output_fHistory, [21](#)
 output_func, [21](#)
 runFunc, [21](#)
 runStrategy, [22](#)

runFunc
 run.h, [21](#)

runStrategy
 run.h, [22](#)

schwefel
 func, [11](#)

sineEnvelopeSineWave
 func, [12](#)

src/data_stats.h, [17](#)

src/differential_evolution.h, [17](#)

src/func.h, [18](#)

src/population.h, [19](#)

src/run.h, [20](#)

step
 func, [12](#)

stretchedVSineWave
 func, [12](#)