Cuckoo Search Optimization Algorithm

Generated by Doxygen 1.8.17

Chapter 1

Namespace Index

1.1 Namespace List

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

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

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 CS Class Reference

Public Member Functions

- **CS** (int strat, float(∗f)(vector< float > &), float l, float u)
- vector< float > run ()

5.1.1 Member Function Documentation

5.1.1.1 run()

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

get a cuckoo i randomly, generate a new cuckoo Fi and calculate cost

choose a nest j among n randomly and get Fj's cost

replace j with i if Fj < Fi

abandon pa of n worse nests, keep the best solutions and build new ones

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

- src/cuckoo_search.h
- src/cuckoo_search.cpp

5.2 DataStats Class Reference

Public Member Functions

- void run ()

  Generate analytical data.

Public Attributes

- float **mean**
- float **median**
- float **stand**
- float **range** [2]
- 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

5.3 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
- void sort_cost ()

  Reset the population for new run.

Public Attributes

- vector< vector< float > > data

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

5.3.1 Detailed Description

Population for genetic algorithm.

5.3.2 Member Function Documentation

5.3.2.1 init()

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

Initialize a population

Parameters

s	population size
d	population dimension
low	x low bound
high	x high bound

5.3.2.2 sort_cost()

```
void Population::sort_cost ( )
```

Reset the population for new run.

pair sort to sort the population by cost from low cost to high 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)

Version

1.0

Date

2020-02-11

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)

Version

0.1

Date

2020-02-05

6.3 src/population.h File Reference

Population generation.

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

Classes

- class Population

  Population for genetic algorithm.

6.3.1 Detailed Description

Population generation.

Author

Cheng Su (csu@cwu.edu)

Version

1.0

Date

2020-02-1

6.4 src/run.h File Reference

run

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

Functions

- void **setSeed** ()
- DataStats runFunc (int experiment, string func_name, float(∗f)(vector< float > &), float l, float u)
- void output_fHistory (int strategy, string func_name, vector< vector< float >> f_bests_history)
- void output_func (string func_name, vector< DataStats > result)
- void output_all (vector< DataStats > result_best)

6.4.1 Detailed Description

run

Author

Cheng Su (csu@cwu.edu)

Version

1.0

Date

2020-02-11

6.4.2 Function Documentation

6.4.2.1 output_all()

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

write best result for every function

Parameters

result_best	best result for each function

6.4.2.2 output_fHistory()

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

output f history

6.4.2.3 output_func()

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

write result to a file for one function

Parameters

func_name	function name
result	result stat
f_bests_history	cost history

output result stats

output time history

write result to file

write time history to file

6.4.2.4 runFunc()

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

run pso for a function

Parameters

strat	strategy
experiment	number of experiment
f	function
l	low x bound
u	up x bound

Returns

return result analysis

Index