

My Project

Generated by Doxygen 1.8.15

1 Class Index	1
1.1 Class List	1
2 Class Documentation	3
2.1 FA_Params Struct Reference	3
2.2 functions Class Reference	3
2.2.1 Member Function Documentation	4
2.2.1.1 Ackley_One()	4
2.2.1.2 Ackley_Two()	4
2.2.1.3 Alpine()	5
2.2.1.4 Egg_Holder()	5
2.2.1.5 first_De_Jong()	5
2.2.1.6 Greiwangk()	6
2.2.1.7 Levy()	6
2.2.1.8 Masters_Cosine_Wave()	7
2.2.1.9 Michalewicz()	7
2.2.1.10 Pathological()	7
2.2.1.11 Quartic()	8
2.2.1.12 Rana()	8
2.2.1.13 Rastrigin()	8
2.2.1.14 Rosenbrock()	9
2.2.1.15 Schwefel()	9
2.2.1.16 Sine_Envelope_Sine_Wave()	10
2.2.1.17 Step()	10
2.2.1.18 Stretched_V_Sine_Wave()	10
2.3 HS_Params Struct Reference	11
2.4 matrix Class Reference	11
2.4.1 Constructor & Destructor Documentation	12
2.4.1.1 matrix() [1/2]	12
2.4.1.2 matrix() [2/2]	12
2.5 PSO_Params Struct Reference	13
2.6 utilities Class Reference	13
2.6.1 Member Function Documentation	13
2.6.1.1 simulate()	14
2.6.1.2 str_to_tok()	14
2.6.1.3 write_to_file()	15
Index	17

Chapter 1

Class Index

1.1 Class List

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

FA_Params	3
functions	3
HS_Params	11
matrix	11
PSO_Params	13
utilities	13

Chapter 2

Class Documentation

2.1 FA_Params Struct Reference

Public Attributes

- [matrix](#) * **std_devs**
- double * **gBests**
- double **gamma**
- double **B0**
- double **alpha**
- double **l_b**
- double **u_b**
- int **func_id**
- int **nf**
- int **dim**
- int **num_iters**
- int **std_devs_row**

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

- FA.h

2.2 functions Class Reference

Public Member Functions

- double [Schwefel](#) (double *X, int dimension)
- double [first_De_Jong](#) (double *X, int dimension)
- double [Rosenbrock](#) (double *X, int dimension)
- double [Rastrigin](#) (double *X, int dimension)
- double [Greiwangk](#) (double *X, int dimension)
- double [Sine_Envelope_Sine_Wave](#) (double *X, int dimension)
- double [Stretched_V_Sine_Wave](#) (double *X, int dimension)
- double [Ackley_One](#) (double *X, int dimension)
- double [Ackley_Two](#) (double *X, int dimension)

- double [Egg_Holder](#) (double *X, int dimension)
- double [Rana](#) (double *X, int dimension)
- double [Pathological](#) (double *X, int dimension)
- double [Michalewicz](#) (double *X, int dimension)
- double [Masters_Cosine_Wave](#) (double *X, int dimension)
- double [Quartic](#) (double *X, int dimension)
- double [Levy](#) (double *X, int dimension)
- double [Step](#) (double *X, int dimension)
- double [Alpine](#) (double *X, int dimension)

2.2.1 Member Function Documentation

2.2.1.1 Ackley_One()

```
double functions::Ackley_One (
    double * X,
    int dimension )
```

Ackley's One function

Parameters

<i>X</i>	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Ackley's One function

2.2.1.2 Ackley_Two()

```
double functions::Ackley_Two (
    double * X,
    int dimension )
```

Ackley's Two function

Parameters

<i>X</i>	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Ackley's Twofunction

2.2.1.3 Alpine()

```
double functions::Alpine (
    double * X,
    int dimension )
```

Alpine's function

Parameters

<i>X</i>	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Alpine's function

2.2.1.4 Egg_Holder()

```
double functions::Egg_Holder (
    double * X,
    int dimension )
```

Egg Holder's function

Parameters

<i>X</i>	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Egg Holder's function

2.2.1.5 first_De_Jong()

```
double functions::first_De_Jong (
    double * X,
    int dimension )
```

1st De Jong's function

Parameters

X	the input space
<i>dimension</i>	the size of the input space

Returns

: result of 1st De Jong's function

2.2.1.6 Greiwangk()

```
double functions::Grewangk (
    double * X,
    int dimension )
```

Grewangk's function

Parameters

X	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Greiwangk's function

2.2.1.7 Levy()

```
double functions::Levy (
    double * X,
    int dimension )
```

Levy's function

Parameters

X	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Levy's function

2.2.1.8 Masters_Cosine_Wave()

```
double functions::Masters_Cosine_Wave (
    double * X,
    int dimension )
```

Masters Cosine Wave's function

Parameters

<i>X</i>	the input space
<i>dimension</i>	the size of the input space

Returns

: Masters Cosine Wave's function

2.2.1.9 Michalewicz()

```
double functions::Michalewicz (
    double * X,
    int dimension )
```

Michalewicz's function

Parameters

<i>X</i>	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Michalewicz's function

2.2.1.10 Pathological()

```
double functions::Pathological (
    double * X,
    int dimension )
```

Pathological's function

Parameters

<i>X</i>	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Pathological's function

2.2.1.11 Quartic()

```
double functions::Quartic (
    double * X,
    int dimension )
```

Quartic's function

Parameters

<i>X</i>	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Quartic's function

2.2.1.12 Rana()

```
double functions::Rana (
    double * X,
    int dimension )
```

Rana's function

Parameters

<i>X</i>	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Rana's function

2.2.1.13 Rastrigin()

```
double functions::Rastrigin (
    double * X,
    int dimension )
```

Rastrigin's function

Parameters

X	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Rastrigin's function

2.2.1.14 Rosenbrock()

```
double functions::Rosenbrock (
    double * X,
    int dimension )
```

Rosenbrock's function

Parameters

X	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Rosenbrock's function

2.2.1.15 Schwefel()

```
double functions::Schwefel (
    double * X,
    int dimension )
```

Schwefel's function

Parameters

X	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Schwefel's function

2.2.1.16 Sine_Envelope_Sine_Wave()

```
double functions::Sine_Envelope_Sine_Wave (
    double * X,
    int dimension )
```

Sine Envelope Sine Wave's function

Parameters

<i>X</i>	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Sine Envelope Sine Wave's function

2.2.1.17 Step()

```
double functions::Step (
    double * X,
    int dimension )
```

Step's function

Parameters

<i>X</i>	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Step's function

2.2.1.18 Stretched_V_Sine_Wave()

```
double functions::Stretched_V_Sine_Wave (
    double * X,
    int dimension )
```

Stretched V Sine Wave's function

Parameters

<i>X</i>	the input space
<i>dimension</i>	the size of the input space

Returns

: result of Stretched V Since Wave's function

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

- functions.h
- functions.cpp

2.3 HS_Params Struct Reference

Public Attributes

- [matrix](#) * **std_devs**
- double * **gBests**
- double * **gWorsts**
- double **HMCR**
- double **PAR**
- double **HMS**
- double **EOR**
- double **l_b**
- double **u_b**
- double **bw**
- int **func_id**
- int **nh**
- int **dim**
- int **num_iters**
- int **std_devs_row**

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

- HS.h

2.4 matrix Class Reference

Public Member Functions

- [matrix](#) (int num_rows, int num_columns, int l_b, int h_b, mt19937 &mt_rand)
- [matrix](#) (int num_rows, int num_columns)

Public Attributes

- const int **num_rows**
- const int **num_columns**
- const int **l_b**
- const int **h_b**
- mt19937 **mt_rand**
- double ** **mat**

2.4.1 Constructor & Destructor Documentation

2.4.1.1 `matrix()` [1/2]

```
matrix::matrix (
    int num_rows,
    int num_columns,
    int l_b,
    int h_b,
    mt19937 & mt_rand )
```

generate an empty matrix and fill it up with randomly generated numbers within some range

Parameters

<i>num_rows</i>	integer representing the number of rows in the matrix
<i>dim</i>	integer representing the dimension or number of columns in the matrix
<i>l_b</i>	double representing the lowest bound for the random generator
<i>h_b</i>	double representing the highest bound for the random generator

Returns

: a matrix of randomly generated numbers

2.4.1.2 `matrix()` [2/2]

```
matrix::matrix (
    int num_rows,
    int num_columns )
```

generate an empty matrix

Parameters

<i>num_rows</i>	integer representing the number of rows in the matrix
<i>dim</i>	integer representing the dimension or number of columns in the matrix

Returns

: an empty matrix

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

- `matrix.h`
- `matrix.cpp`

2.5 PSO_Params Struct Reference

Public Attributes

- [matrix](#) * **std_devs**
- double * **gBests**
- double **c1**
- double **c2**
- double **k**
- double **l_b**
- double **u_b**
- int **func_id**
- int **np**
- int **dim**
- int **num_iters**
- int **std_devs_row**

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

- PSO.h

2.6 utilities Class Reference

Public Member Functions

- double * [str_to_tok](#) (char *string, char *delim, int num_tokens)
- void [write_to_file](#) ([matrix](#) *mat, string file_name)
- int [get_algorithm_id](#) ()
- int [get_selection_id](#) ()
- double [find_lowest](#) (const double *list, int len)
- void [simulate](#) (int dim, int ns, int num_functions, double *ranges, int algo_id, int num_iters, int num_exp, double c1, double c2, double k, double gamma, double BO, double alpha, double HMCR, double PAR, double bw, mt19937 &mt_rand)

2.6.1 Member Function Documentation

2.6.1.1 simulate()

```
void utilities::simulate (
    int dim,
    int ns,
    int num_functions,
    double * ranges,
    int algo_id,
    int num_iters,
    int num_exp,
    double c1,
    double c2,
    double k,
    double gamma,
    double B0,
    double alpha,
    double HMCR,
    double PAR,
    double bw,
    mt19937 & mt_rand )
```

simulate the Particle Swarm Optimization, the Firefly algorithm and the Harmony Search algorithm

Parameters

<i>dim</i>	: an integer for the dimension of the solutions
<i>ns</i>	: an integer the number of solutions
<i>num_functions</i>	: an integer for the number of objective functions to be simulated (the 18 functions)
<i>ranges</i>	an array of doubles containing the lower and upper bound for each of the objective functions
<i>algo_id</i>	an integer for the evolutionary algorithm to be simulated
<i>num_iters</i>	: an integer for the number of iterations for the swarm algorithms
<i>num_exp</i>	an integer for the number of experimentations to be run
<i>c1,c2</i>	doubles
<i>k</i>	a double
<i>gamma</i>	a double
<i>B0</i>	a double
<i>alpha</i>	a double
<i>HMCR</i>	a double
<i>PAR</i>	a double
<i>bw</i>	a double
<i>mt_rand</i>	a seeded random generator to generate random numbers (seeded once in main.cpp)

Returns

: None

2.6.1.2 str_to_tok()

```
double * utilities::str_to_tok (
    char * string,
```

```
char * delim,  
int num_tokens )
```

split a string into double tokens

Parameters

<i>string</i>	the string to be splitted
<i>delim</i>	the character that separates the tokens in the string
<i>num_tokens</i>	number of tokens to expect

Returns

: an array of doubles

2.6.1.3 write_to_file()

```
void utilities::write_to_file (   
    matrix * mat,   
    string file_name )
```

write a 2d array to a csv file

Parameters

<i>mat</i>	a matrix containing the elements to write to the csv file
<i>file_name</i>	the name of the file where data will be saved

Returns

: None

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

- utilities.h
- utilities.cpp

Index

- Ackley_One
 - functions, [4](#)
- Ackley_Two
 - functions, [4](#)
- Alpine
 - functions, [5](#)
- Egg_Holder
 - functions, [5](#)
- FA_Params, [3](#)
- first_De_Jong
 - functions, [5](#)
- functions, [3](#)
 - Ackley_One, [4](#)
 - Ackley_Two, [4](#)
 - Alpine, [5](#)
 - Egg_Holder, [5](#)
 - first_De_Jong, [5](#)
 - Greiwangk, [6](#)
 - Levy, [6](#)
 - Masters_Cosine_Wave, [6](#)
 - Michalewicz, [7](#)
 - Pathological, [7](#)
 - Quartic, [8](#)
 - Rana, [8](#)
 - Rastrigin, [8](#)
 - Rosenbrock, [9](#)
 - Schwefel, [9](#)
 - Sine_Envelope_Sine_Wave, [9](#)
 - Step, [10](#)
 - Stretched_V_Sine_Wave, [10](#)
- Greiwangk
 - functions, [6](#)
- HS_Params, [11](#)
- Levy
 - functions, [6](#)
- Masters_Cosine_Wave
 - functions, [6](#)
- matrix, [11](#)
 - matrix, [12](#)
- Michalewicz
 - functions, [7](#)
- Pathological
 - functions, [7](#)
- PSO_Params, [13](#)
- Quartic
 - functions, [8](#)
- Rana
 - functions, [8](#)
- Rastrigin
 - functions, [8](#)
- Rosenbrock
 - functions, [9](#)
- Schwefel
 - functions, [9](#)
- simulate
 - utilities, [13](#)
- Sine_Envelope_Sine_Wave
 - functions, [9](#)
- Step
 - functions, [10](#)
- str_to_tok
 - utilities, [14](#)
- Stretched_V_Sine_Wave
 - functions, [10](#)
- utilities, [13](#)
 - simulate, [13](#)
 - str_to_tok, [14](#)
 - write_to_file, [15](#)
- write_to_file
 - utilities, [15](#)