

Project 3

Generated by Doxygen 1.8.17

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 DEInputParameter< Tinput >::Bounds Struct Reference	7
4.1.1 Detailed Description	7
4.1.2 Member Data Documentation	7
4.1.2.1 l	7
4.1.2.2 u	8
4.2 GAInputParameter< Tinput >::Bounds Struct Reference	8
4.2.1 Detailed Description	8
4.2.2 Member Data Documentation	8
4.2.2.1 l	8
4.2.2.2 u	9
4.3 Clock Class Reference	9
4.3.1 Member Function Documentation	9
4.3.1.1 tac()	9
4.3.1.2 tic()	9
4.3.2 Member Data Documentation	9
4.3.2.1 start_c	10
4.3.2.2 stop_c	10
4.4 DEInputParameter< Tinput > Struct Template Reference	10
4.4.1 Detailed Description	11
4.4.2 Member Data Documentation	11
4.4.2.1 bounds	11
4.4.2.2 cr	11
4.4.2.3 dim	11
4.4.2.4 pop_size	12
4.4.2.5 scale_f	12
4.4.2.6 scale_lambda	12
4.4.2.7 t_max	12
4.5 DifferentialEvolution< Tinput, Toutput > Class Template Reference	12
4.5.1 Detailed Description	13
4.5.2 Constructor & Destructor Documentation	14
4.5.2.1 DifferentialEvolution()	14
4.5.2.2 ~DifferentialEvolution()	14
4.5.3 Member Function Documentation	14

4.5.3.1 keepInRange()	14
4.5.3.2 randomR()	14
4.5.3.3 runS10_DE_rand_2_bin()	15
4.5.3.4 runS1_DE_best_1_exp()	15
4.5.3.5 runS2_DE_rand_1_exp()	15
4.5.3.6 runS3_DE_randbest_1_exp()	15
4.5.3.7 runS4_DE_best_2_exp()	15
4.5.3.8 runS5_DE_rand_2_exp()	16
4.5.3.9 runS6_DE_best_1_bin()	16
4.5.3.10 runS7_DE_rand_1_bin()	16
4.5.3.11 runS8_DE_randbest_1_bin()	16
4.5.3.12 runS9_DE_best_2_bin()	16
4.5.3.13 saveResult()	16
4.5.4 Member Data Documentation	17
4.5.4.1 actual_pop	17
4.5.4.2 best_cost	17
4.5.4.3 ms_random	17
4.5.4.4 new_pop	17
4.5.4.5 param	18
4.6 GeneticAlgorithm< Tinput, Toutput >::doCompare Struct Reference	18
4.6.1 Constructor & Destructor Documentation	18
4.6.1.1 doCompare()	18
4.6.2 Member Function Documentation	18
4.6.2.1 operator()()	18
4.6.3 Member Data Documentation	19
4.6.3.1 m_info	19
4.7 PopulationBenchmark< Tinput, Toutput >::doCompare< Tinput, Toutput > Struct Template Reference	19
4.7.1 Constructor & Destructor Documentation	19
4.7.1.1 doCompare()	19
4.7.2 Member Function Documentation	19
4.7.2.1 operator()()	20
4.7.3 Member Data Documentation	20
4.7.3.1 m_info	20
4.8 GeneticAlgorithm< Tinput, Toutput >::doCompareNewPop Struct Reference	20
4.8.1 Constructor & Destructor Documentation	20
4.8.1.1 doCompareNewPop()	20
4.8.2 Member Function Documentation	21
4.8.2.1 operator()()	21
4.8.3 Member Data Documentation	21
4.8.3.1 m_info	21
4.9 Functions1< Tinput, Toutput > Class Template Reference	21
4.9.1 Detailed Description	22

4.9.2 Member Typedef Documentation	23
4.9.2.1 function_pointer	23
4.9.3 Member Function Documentation	23
4.9.3.1 f10EggHolder()	23
4.9.3.2 f11Rana()	23
4.9.3.3 f12Pathological()	24
4.9.3.4 f13Michalewicz()	24
4.9.3.5 f14MasterCosineWave()	25
4.9.3.6 f15Quartic()	25
4.9.3.7 f16Levy()	25
4.9.3.8 f17Step()	26
4.9.3.9 f18Alphine()	26
4.9.3.10 f1Schwefels()	27
4.9.3.11 f2DeJongs1()	27
4.9.3.12 f3Rosenbrock()	27
4.9.3.13 f4Rastrigin()	28
4.9.3.14 f5Griewankgk()	28
4.9.3.15 f6SineEnvelopeSineWave()	29
4.9.3.16 f7StretchedVSineWave()	29
4.9.3.17 f8AckleysOne()	29
4.9.3.18 f9AckleysTwo()	30
4.9.3.19 getFunctionById()	30
4.10 GAInputParameter< Tinput > Struct Template Reference	31
4.10.1 Detailed Description	31
4.10.2 Member Data Documentation	32
4.10.2.1 bounds	32
4.10.2.2 cr	32
4.10.2.3 dim	32
4.10.2.4 er	32
4.10.2.5 m	32
4.10.2.6 ns	33
4.10.2.7 t_max	33
4.11 GeneticAlgorithm< Tinput, Toutput > Class Template Reference	33
4.11.1 Detailed Description	34
4.11.2 Constructor & Destructor Documentation	35
4.11.2.1 GeneticAlgorithm()	35
4.11.2.2 ~GeneticAlgorithm()	35
4.11.3 Member Function Documentation	35
4.11.3.1 crossover()	35
4.11.3.2 evaluateCost()	35
4.11.3.3 findBestSolution()	36
4.11.3.4 getFitness()	36

4.11.3.5 keepInRange()	36
4.11.3.6 mutate()	36
4.11.3.7 printCost()	36
4.11.3.8 printFitness()	37
4.11.3.9 printInputPopulation()	37
4.11.3.10 printNewPopulation()	37
4.11.3.11 printPopulation()	37
4.11.3.12 randomInit()	37
4.11.3.13 reduce()	37
4.11.3.14 saveResult()	38
4.11.3.15 select()	38
4.11.3.16 selectParent()	38
4.11.3.17 sortNewPopulationByIndexAsc()	38
4.11.3.18 sortPopulationByIndexAsc()	38
4.11.4 Member Data Documentation	38
4.11.4.1 cost	39
4.11.4.2 fitness	39
4.11.4.3 ms_random_generator	39
4.11.4.4 new_population	39
4.11.4.5 new_population_asc_index	39
4.11.4.6 new_population_cost	39
4.11.4.7 parameters	40
4.11.4.8 population	40
4.11.4.9 population_asc_index	40
4.11.4.10 total_fitness	40
4.12 GAInputParameter< Tinput >::Mutation Struct Reference	40
4.12.1 Detailed Description	41
4.12.2 Member Data Documentation	41
4.12.2.1 precision	41
4.12.2.2 range	41
4.12.2.3 rate	41
4.13 Population< Tinput, Toutput > Class Template Reference	41
4.13.1 Constructor & Destructor Documentation	42
4.13.1.1 Population()	42
4.13.1.2 ~Population()	42
4.13.2 Member Function Documentation	42
4.13.2.1 fillWithRandom()	43
4.13.2.2 getData() [1/2]	43
4.13.2.3 getData() [2/2]	43
4.13.2.4 printPopulation()	43
4.13.2.5 setData()	43
4.13.2.6 swap()	43

4.13.3 Member Data Documentation	44
4.13.3.1 data	44
4.13.3.2 dimension	44
4.13.3.3 n_items	44
4.13.3.4 random_ms	44
4.14 PopulationBenchmark< Tinput, Toutput > Class Template Reference	44
4.14.1 Constructor & Destructor Documentation	45
4.14.1.1 PopulationBenchmark()	45
4.14.1.2 ~PopulationBenchmark()	46
4.14.2 Member Function Documentation	46
4.14.2.1 calcCost1Item()	46
4.14.2.2 calcCostExt()	46
4.14.2.3 calculateMinCost()	46
4.14.2.4 evaluateCost()	46
4.14.2.5 getMinCost()	46
4.14.2.6 getMinCostData()	47
4.14.2.7 printCost()	47
4.14.2.8 printIndex()	47
4.14.2.9 saveBest()	47
4.14.2.10 setDataAndCost()	47
4.14.2.11 sortIndexByCostAsc()	47
4.14.2.12 swap()	48
4.14.3 Member Data Documentation	48
4.14.3.1 asc_index	48
4.14.3.2 cost	48
4.14.3.3 min_cost	48
4.14.3.4 min_cost_i	48
4.15 Runner< Tinput, Toutput > Class Template Reference	49
4.15.1 Detailed Description	50
4.15.2 Constructor & Destructor Documentation	50
4.15.2.1 Runner()	50
4.15.2.2 ~Runner()	50
4.15.3 Member Function Documentation	50
4.15.3.1 computeStatistic()	51
4.15.3.2 fillDEParameterFromFile()	51
4.15.3.3 fillGAPParameterFromFile()	51
4.15.3.4 printSolutions()	51
4.15.3.5 runOptimization()	51
4.15.3.6 saveStatistic()	52
4.15.4 Member Data Documentation	52
4.15.4.1 algorithm_id	52
4.15.4.2 dimensions	52

4.15.4.3 function_id	52
4.15.4.4 n_runs	52
4.15.4.5 range_high	53
4.15.4.6 range_low	53
4.15.4.7 solutions	53
4.15.4.8 stat_analysis	53
4.16 Runner< Tinput, Toutput >::statistics Struct Reference	53
4.16.1 Detailed Description	54
4.16.2 Member Data Documentation	54
4.16.2.1 mean	54
4.16.2.2 median	54
4.16.2.3 range	54
4.16.2.4 range_max	54
4.16.2.5 range_min	55
4.16.2.6 std_dev	55
4.16.2.7 time_ms	55
5 File Documentation	57
5.1 debug.h File Reference	57
5.1.1 Macro Definition Documentation	57
5.1.1.1 debug	57
5.1.1.2 debug1	57
5.1.1.3 debug_var	58
5.1.1.4 debugfile	58
5.2 differential_evolution.cpp File Reference	58
5.3 differential_evolution.h File Reference	58
5.4 functions1.cpp File Reference	58
5.5 functions1.h File Reference	59
5.6 genetic_algorithm.cpp File Reference	59
5.7 genetic_algorithm.h File Reference	59
5.8 main.cpp File Reference	60
5.8.1 Function Documentation	60
5.8.1.1 main()	60
5.9 population.cpp File Reference	60
5.10 population.h File Reference	61
5.11 population_benchmark.cpp File Reference	61
5.12 population_benchmark.h File Reference	61
5.13 runner.cpp File Reference	61
5.14 runner.h File Reference	62
5.15 util.h File Reference	62
5.15.1 Function Documentation	62
5.15.1.1 printArray()	62

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

DEInputParameter< Tinput >::Bounds	7
GAInputParameter< Tinput >::Bounds	8
Clock	9
DEInputParameter< Tinput >	10
DifferentialEvolution< Tinput, Toutput >	12
GeneticAlgorithm< Tinput, Toutput >::doCompare	18
PopulationBenchmark< Tinput, Toutput >::doCompare< Tinput, Toutput >	19
GeneticAlgorithm< Tinput, Toutput >::doCompareNewPop	20
Functions1< Tinput, Toutput >	21
GAInputParameter< Tinput >	31
GeneticAlgorithm< Tinput, Toutput >	33
GAInputParameter< Tinput >::Mutation	40
Population< Tinput, Toutput >	41
PopulationBenchmark< Tinput, Toutput >	44
Runner< Tinput, Toutput >	49
Runner< Tinput, Toutput >::statistics	53

Chapter 2

Class Index

2.1 Class List

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

DEInputParameter< Tinput >::Bounds	
Problem bounds	7
GAInputParameter< Tinput >::Bounds	
Problem bounds	8
Clock	9
DEInputParameter< Tinput >	
Template Differential Evolution Input parameter Struct	10
DifferentialEvolution< Tinput, Toutput >	
Template Differential Evolution class	12
GeneticAlgorithm< Tinput, Toutput >::doCompare	18
PopulationBenchmark< Tinput, Toutput >::doCompare< Tinput, Toutput >	19
GeneticAlgorithm< Tinput, Toutput >::doCompareNewPop	20
Functions1< Tinput, Toutput >	
Template class with 18 standard benchmark functions	21
GAInputParameter< Tinput >	
Template Genetic Algorithm Input parameter Struct	31
GeneticAlgorithm< Tinput, Toutput >	
Template Genetic Algorithm (GA) class	33
GAInputParameter< Tinput >::Mutation	
Mutation parameters	40
Population< Tinput, Toutput >	41
PopulationBenchmark< Tinput, Toutput >	44
Runner< Tinput, Toutput >	
Template class which generates random sample of different dimensions and run a specific bench- mark function	49
Runner< Tinput, Toutput >::statistics	
Struct of statistical analysis	53

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

debug.h	57
differential_evolution.cpp	58
differential_evolution.h	58
functions1.cpp	58
functions1.h	59
genetic_algorithm.cpp	59
genetic_algorithm.h	59
main.cpp	60
population.cpp	60
population.h	61
population_benchmark.cpp	61
population_benchmark.h	61
runner.cpp	61
runner.h	62
util.h	62

Chapter 4

Class Documentation

4.1 DEInputParameter< Tinput >::Bounds Struct Reference

problem bounds

```
#include <differential_evolution.h>
```

Collaboration diagram for DEInputParameter< Tinput >::Bounds:

Public Attributes

- Tinput [u](#)
upper bound
- Tinput [l](#)
lower bound

4.1.1 Detailed Description

```
template<class Tinput>  
struct DEInputParameter< Tinput >::Bounds
```

problem bounds

4.1.2 Member Data Documentation

4.1.2.1 l

```
template<class Tinput >  
Tinput DEInputParameter< Tinput >::Bounds::l
```

lower bound

4.1.2.2 u

```
template<class Tinput >
Tinput DEInputParameter< Tinput >::Bounds::u
```

upper bound

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

- [differential_evolution.h](#)

4.2 GAInputParameter< Tinput >::Bounds Struct Reference

problem bounds

```
#include <genetic_algorithm.h>
```

Collaboration diagram for GAInputParameter< Tinput >::Bounds:

Public Attributes

- Tinput [u](#)
upper bound
- Tinput [l](#)
lower bound

4.2.1 Detailed Description

```
template<class Tinput>
struct GAInputParameter< Tinput >::Bounds
```

problem bounds

4.2.2 Member Data Documentation

4.2.2.1 l

```
template<class Tinput >
Tinput GAInputParameter< Tinput >::Bounds::l
```

lower bound

4.2.2.2 u

```
template<class Tinput >  
Tinput GAInputParameter< Tinput >::Bounds::u
```

upper bound

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

- [genetic_algorithm.h](#)

4.3 Clock Class Reference

```
#include <util.h>
```

Collaboration diagram for Clock:

Public Member Functions

- void [tic](#) ()
- double [tac](#) ()

Private Attributes

- clock_t [start_c](#)
- clock_t [stop_c](#)

4.3.1 Member Function Documentation

4.3.1.1 tac()

```
double Clock::tac ( ) [inline]
```

4.3.1.2 tic()

```
void Clock::tic ( ) [inline]
```

4.3.2 Member Data Documentation

4.3.2.1 start_c

```
clock_t Clock::start_c [private]
```

4.3.2.2 stop_c

```
clock_t Clock::stop_c [private]
```

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

- [util.h](#)

4.4 DEInputParameter< Tinput > Struct Template Reference

Template Differential Evolution Input parameter Struct.

```
#include <differential_evolution.h>
```

Collaboration diagram for DEInputParameter< Tinput >:

Classes

- struct [Bounds](#)
problem bounds

Public Attributes

- int [pop_size](#)
population size
- int [dim](#)
problem dimension
- [Bounds bounds](#)
instance of struct #Bounds
- int [t_max](#)
maximum number of iterations
- double [cr](#)
crossover rate
- double [scale_f](#)
scale factor F
- double [scale_lambda](#)
scale factor lambda

4.4.1 Detailed Description

```
template<class Tinput>
struct DEInputParameter< Tinput >
```

Template Differential Evolution Input parameter Struct.

Struct to where all the DE parameters are stored

Author

Chao Huang Lin (chao.huanglin@cwu.edu)

Date

2020-02-08

4.4.2 Member Data Documentation

4.4.2.1 bounds

```
template<class Tinput >
Bounds DEInputParameter< Tinput >::bounds
```

instance of struct #Bounds

4.4.2.2 cr

```
template<class Tinput >
double DEInputParameter< Tinput >::cr
```

crossover rate

4.4.2.3 dim

```
template<class Tinput >
int DEInputParameter< Tinput >::dim
```

problem dimension

4.4.2.4 pop_size

```
template<class Tinput >
int DEInputParameter< Tinput >::pop_size
```

population size

4.4.2.5 scale_f

```
template<class Tinput >
double DEInputParameter< Tinput >::scale_f
```

scale factor F

4.4.2.6 scale_lambda

```
template<class Tinput >
double DEInputParameter< Tinput >::scale_lambda
```

scale factor lambda

4.4.2.7 t_max

```
template<class Tinput >
int DEInputParameter< Tinput >::t_max
```

maximum number of iterations

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

- [differential_evolution.h](#)

4.5 DifferentialEvolution< Tinput, Toutput > Class Template Reference

Template Differential Evolution class.

```
#include <differential_evolution.h>
```

Collaboration diagram for DifferentialEvolution< Tinput, Toutput >:

Public Member Functions

- [DifferentialEvolution](#) ([DEInputParameter](#)< Tinput > [param](#))
- [~DifferentialEvolution](#) ()
- Toutput [runS7_DE_rand_1_bin](#) (int function_id)
- Toutput [runS10_DE_rand_2_bin](#) (int function_id)
- Toutput [runS6_DE_best_1_bin](#) (int function_id)
- Toutput [runS9_DE_best_2_bin](#) (int function_id)
- Toutput [runS8_DE_randbest_1_bin](#) (int function_id)
- Toutput [runS1_DE_best_1_exp](#) (int function_id)
- Toutput [runS2_DE_rand_1_exp](#) (int function_id)
- Toutput [runS3_DE_randbest_1_exp](#) (int function_id)
- Toutput [runS4_DE_best_2_exp](#) (int function_id)
- Toutput [runS5_DE_rand_2_exp](#) (int function_id)

Private Member Functions

- void [randomR](#) (int *r, int n, int current_i)
generate random vector position r used to create trail vectors
- void [saveResult](#) (Toutput [best_cost](#), Tinput *best_vector, std::string result_file)
save the indivuo with best cost in a file
- void [keepInRange](#) (Tinput &element)
keep the element of the vector inside the search range

Private Attributes

- [PopulationBenchmark](#)< Tinput, Toutput > * [actual_pop](#)
actual population
- [PopulationBenchmark](#)< Tinput, Toutput > * [new_pop](#)
new population
- [DEInputParameter](#)< Tinput > [param](#)
DE parameters.
- MersenneTwister [ms_random](#)
mersenne twister random generator
- Toutput * [best_cost](#)
array with history of best cost during the generations

4.5.1 Detailed Description

```
template<class Tinput, class Toutput>
class DifferentialEvolution< Tinput, Toutput >
```

Template Differential Evolution class.

Differential Evolution (DE) is based on the strategy that employs the difference of tow randomly selected parameter vectors as the source of random variations for a third parameter vector.

Author

Chao Huang Lin (chao.huanglin@cwu.edu)

Date

2020-02-08

4.5.2 Constructor & Destructor Documentation

4.5.2.1 DifferentialEvolution()

```
template<class Tinput , class Toutput >
DifferentialEvolution< Tinput, Toutput >::DifferentialEvolution (
    DEInputParameter< Tinput > param )
```

4.5.2.2 ~DifferentialEvolution()

```
template<class Tinput , class Toutput >
DifferentialEvolution< Tinput, Toutput >::~~DifferentialEvolution
```

4.5.3 Member Function Documentation

4.5.3.1 keepInRange()

```
template<class Tinput , class Toutput >
void DifferentialEvolution< Tinput, Toutput >::keepInRange (
    Tinput & element ) [private]
```

keep the element of the vector inside the search range

the method truncate the element to the upper and lower bound of the range, when the element exceeds these limits.

Parameters

<i>&element</i>	/// element of the vector
---------------------	------------------------------

4.5.3.2 randomR()

```
template<class Tinput , class Toutput >
void DifferentialEvolution< Tinput, Toutput >::randomR (
    int * r,
    int n,
    int current_i ) [private]
```

generate random vector position r used to create trial vectors

the random vector positions r should satisfy the condition that they cannot repeat and they cannot be equal to the current vector i

Parameters

<i>*r</i>	/// pointer of array r
<i>n</i>	/// size of array r
<i>current↔ _i</i>	/// current position of the vector i

4.5.3.3 runS10_DE_rand_2_bin()

```
template<class Tinput , class Toutput >
Toutput DifferentialEvolution< Tinput, Toutput >::runS10_DE_rand_2_bin (
    int function_id )
```

4.5.3.4 runS1_DE_best_1_exp()

```
template<class Tinput , class Toutput >
Toutput DifferentialEvolution< Tinput, Toutput >::runS1_DE_best_1_exp (
    int function_id )
```

4.5.3.5 runS2_DE_rand_1_exp()

```
template<class Tinput , class Toutput >
Toutput DifferentialEvolution< Tinput, Toutput >::runS2_DE_rand_1_exp (
    int function_id )
```

4.5.3.6 runS3_DE_randbest_1_exp()

```
template<class Tinput , class Toutput >
Toutput DifferentialEvolution< Tinput, Toutput >::runS3_DE_randbest_1_exp (
    int function_id )
```

4.5.3.7 runS4_DE_best_2_exp()

```
template<class Tinput , class Toutput >
Toutput DifferentialEvolution< Tinput, Toutput >::runS4_DE_best_2_exp (
    int function_id )
```

4.5.3.8 runS5_DE_rand_2_exp()

```
template<class Tinput , class Toutput >
Toutput DifferentialEvolution< Tinput, Toutput >::runS5_DE_rand_2_exp (
    int function_id )
```

4.5.3.9 runS6_DE_best_1_bin()

```
template<class Tinput , class Toutput >
Toutput DifferentialEvolution< Tinput, Toutput >::runS6_DE_best_1_bin (
    int function_id )
```

4.5.3.10 runS7_DE_rand_1_bin()

```
template<class Tinput , class Toutput >
Toutput DifferentialEvolution< Tinput, Toutput >::runS7_DE_rand_1_bin (
    int function_id )
```

4.5.3.11 runS8_DE_randbest_1_bin()

```
template<class Tinput , class Toutput >
Toutput DifferentialEvolution< Tinput, Toutput >::runS8_DE_randbest_1_bin (
    int function_id )
```

4.5.3.12 runS9_DE_best_2_bin()

```
template<class Tinput , class Toutput >
Toutput DifferentialEvolution< Tinput, Toutput >::runS9_DE_best_2_bin (
    int function_id )
```

4.5.3.13 saveResult()

```
template<class Tinput , class Toutput >
void DifferentialEvolution< Tinput, Toutput >::saveResult (
    Toutput best_cost,
    Tinput * best_vector,
    std::string result_file ) [private]
```

save the indivuo with best cost in a file

this method is used to save the best cost of each iteration, the saved data are: the cost and the n dimensions elements of the indivuo.

Parameters

<i>best_cost</i>	/// best cost
<i>*best_vector</i>	/// pointer to the best individuo
<i>result_file</i>	/// name of the file to be saved

4.5.4 Member Data Documentation

4.5.4.1 actual_pop

```
template<class Tinput , class Toutput >
PopulationBenchmark<Tinput, Toutput>* DifferentialEvolution< Tinput, Toutput >::actual_pop
[private]
```

actual population

4.5.4.2 best_cost

```
template<class Tinput , class Toutput >
Toutput* DifferentialEvolution< Tinput, Toutput >::best_cost [private]
```

array with history of best cost during the generations

4.5.4.3 ms_random

```
template<class Tinput , class Toutput >
MersenneTwister DifferentialEvolution< Tinput, Toutput >::ms_random [private]
```

mersenne twister random generator

4.5.4.4 new_pop

```
template<class Tinput , class Toutput >
PopulationBenchmark<Tinput, Toutput>* DifferentialEvolution< Tinput, Toutput >::new_pop [private]
```

new population

4.5.4.5 param

```
template<class Tinput , class Toutput >
DEInputParameter<Tinput> DifferentialEvolution< Tinput, Toutput >::param [private]
```

DE parameters.

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

- [differential_evolution.h](#)
- [differential_evolution.cpp](#)

4.6 GeneticAlgorithm< Tinput, Toutput >::doCompare Struct Reference

Collaboration diagram for GeneticAlgorithm< Tinput, Toutput >::doCompare:

Public Member Functions

- [doCompare](#) (const [GeneticAlgorithm](#) &info)
- bool [operator\(\)](#) (const int &i1, const int &i2)

Public Attributes

- const [GeneticAlgorithm](#) & [m_info](#)

4.6.1 Constructor & Destructor Documentation

4.6.1.1 doCompare()

```
template<class Tinput , class Toutput >
GeneticAlgorithm< Tinput, Toutput >::doCompare::doCompare (
    const GeneticAlgorithm & info ) [inline]
```

4.6.2 Member Function Documentation

4.6.2.1 operator>()

```
template<class Tinput , class Toutput >
bool GeneticAlgorithm< Tinput, Toutput >::doCompare::operator() (
    const int & i1,
    const int & i2 ) [inline]
```

4.6.3 Member Data Documentation

4.6.3.1 m_info

```
template<class Tinput , class Toutput >
const GeneticAlgorithm& GeneticAlgorithm< Tinput, Toutput >::doCompare::m_info
```

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

- [genetic_algorithm.h](#)

4.7 PopulationBenchmark< Tinput, Toutput >::doCompare< Tinput, Toutput > Struct Template Reference

Collaboration diagram for PopulationBenchmark< Tinput, Toutput >::doCompare< Tinput, Toutput >:

Public Member Functions

- [doCompare](#) (const [PopulationBenchmark](#) &info)
- bool [operator\(\)](#) (const int &i1, const int &i2)

Public Attributes

- const [PopulationBenchmark](#) & [m_info](#)

4.7.1 Constructor & Destructor Documentation

4.7.1.1 doCompare()

```
template<class Tinput , class Toutput >
template<class Tinput , class Toutput >
PopulationBenchmark< Tinput, Toutput >::doCompare< Tinput, Toutput >::doCompare (
    const PopulationBenchmark & info ) [inline]
```

4.7.2 Member Function Documentation

4.7.2.1 operator()

```
template<class Tinput , class Toutput >
template<class Tinput , class Toutput >
bool PopulationBenchmark< Tinput, Toutput >::doCompare< Tinput, Toutput >::operator() (
    const int & i1,
    const int & i2 ) [inline]
```

4.7.3 Member Data Documentation

4.7.3.1 m_info

```
template<class Tinput , class Toutput >
template<class Tinput , class Toutput >
const PopulationBenchmark& PopulationBenchmark< Tinput, Toutput >::doCompare< Tinput, Toutput >::m_info
```

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

- [population_benchmark.cpp](#)

4.8 GeneticAlgorithm< Tinput, Toutput >::doCompareNewPop Struct Reference

Collaboration diagram for GeneticAlgorithm< Tinput, Toutput >::doCompareNewPop:

Public Member Functions

- [doCompareNewPop](#) (const [GeneticAlgorithm](#) &info)
- bool [operator\(\)](#) (const int &i1, const int &i2)

Public Attributes

- const [GeneticAlgorithm](#) & [m_info](#)

4.8.1 Constructor & Destructor Documentation

4.8.1.1 doCompareNewPop()

```
template<class Tinput , class Toutput >
GeneticAlgorithm< Tinput, Toutput >::doCompareNewPop::doCompareNewPop (
    const GeneticAlgorithm & info ) [inline]
```

4.8.2 Member Function Documentation

4.8.2.1 operator()

```
template<class Tinput , class Toutput >
bool GeneticAlgorithm< Tinput, Toutput >::doCompareNewPop::operator() (
    const int & i1,
    const int & i2 ) [inline]
```

4.8.3 Member Data Documentation

4.8.3.1 m_info

```
template<class Tinput , class Toutput >
const GeneticAlgorithm& GeneticAlgorithm< Tinput, Toutput >::doCompareNewPop::m_info
```

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

- [genetic_algorithm.h](#)

4.9 Functions1< Tinput, Toutput > Class Template Reference

Template class with 18 standard benchmark functions.

```
#include <functions1.h>
```

Collaboration diagram for Functions1< Tinput, Toutput >:

Public Types

- typedef Toutput(Functions1::* [function_pointer](#)) (Tinput *, int)
Type definition of a function pointer.

Public Member Functions

- [function_pointer](#) [getFunctionById](#) (int id)
Get the function pointer by id of the benchmark function.

Private Member Functions

- Toutput [f1Schwefels](#) (Tinput *data, int dimension)
f1: Schwefel's function
- Toutput [f2DeJongs1](#) (Tinput *data, int dimension)
f2: 1st De Jong's function
- Toutput [f3Rosenbrock](#) (Tinput *data, int dimension)
f3: Rosenbrock function
- Toutput [f4Rastrigin](#) (Tinput *data, int dimension)
f4: Rastrigin function
- Toutput [f5Griewangk](#) (Tinput *data, int dimension)
f5: Griewangk function
- Toutput [f6SineEnvelopeSineWave](#) (Tinput *data, int dimension)
f6: Sine Envelope Sine Wave function
- Toutput [f7StretchedVSineWave](#) (Tinput *data, int dimension)
f7: Stretched V Sine Wave function
- Toutput [f8AckleysOne](#) (Tinput *data, int dimension)
f8: Ackley's One function
- Toutput [f9AckleysTwo](#) (Tinput *data, int dimension)
f9: Ackley's Two function
- Toutput [f10EggHolder](#) (Tinput *data, int dimension)
f10: Egg Holder function
- Toutput [f11Rana](#) (Tinput *data, int dimension)
f11: Rana function
- Toutput [f12Pathological](#) (Tinput *data, int dimension)
f12: Pathological function
- Toutput [f13Michalewicz](#) (Tinput *data, int dimension)
f13: Michalewicz function
- Toutput [f14MasterCosineWave](#) (Tinput *data, int dimension)
f14: Masters Cosine Wave function
- Toutput [f15Quartic](#) (Tinput *data, int dimension)
f15: Quartic function
- Toutput [f16Levy](#) (Tinput *data, int dimension)
f16: Levy function
- Toutput [f17Step](#) (Tinput *data, int dimension)
f17: Step function
- Toutput [f18Alphine](#) (Tinput *data, int dimension)
f18: Alpine function

4.9.1 Detailed Description

```
template<class Tinput, class Toutput>
class Functions1< Tinput, Toutput >
```

Template class with 18 standard benchmark functions.

The main function of the class is [getFunctionById\(int id\)](#) which receive a function id, then it returns the function pointer of the selected function.

Author

Chao Huang Lin (chao.huanglin@cwu.edu)

Date

2020-01-31

4.9.2 Member Typedef Documentation

4.9.2.1 function_pointer

```
template<class Tinput , class Toutput >
typedef Toutput (Functions1::* Functions1< Tinput, Toutput >::function_pointer) (Tinput *, int)
```

Type defenition of a function pointer.

4.9.3 Member Function Documentation

4.9.3.1 f10EggHolder()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f10EggHolder (
    Tinput * data,
    int dimension ) [private]
```

f10: Egg Holder function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.2 f11Rana()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f11Rana (
    Tinput * data,
    int dimension ) [private]
```

f11: Rana function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.3 f12Pathological()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f12Pathological (
    Tinput * data,
    int dimension ) [private]
```

f12: Pathological function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.4 f13Michalewicz()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f13Michalewicz (
    Tinput * data,
    int dimension ) [private]
```

f13: Michalewicz function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.5 f14MasterCosineWave()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f14MasterCosineWave (
    Tinput * data,
    int dimension ) [private]
```

f14: Masters Cosine Wave function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.6 f15Quartic()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f15Quartic (
    Tinput * data,
    int dimension ) [private]
```

f15: Quartic function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.7 f16Levy()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f16Levy (
    Tinput * data,
    int dimension ) [private]
```

f16: Levy function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.8 f17Step()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f17Step (
    Tinput * data,
    int dimension ) [private]
```

f17: Step function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.9 f18Alphine()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f18Alphine (
    Tinput * data,
    int dimension ) [private]
```

f18: Alpine function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.10 f1Schwefels()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f1Schwefels (
    Tinput * data,
    int dimension ) [private]
```

f1: Schwefel's function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.11 f2DeJongs1()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f2DeJongs1 (
    Tinput * data,
    int dimension ) [private]
```

f2: 1st De Jong's function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.12 f3Rosenbrock()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f3Rosenbrock (
    Tinput * data,
    int dimension ) [private]
```

f3: Rosenbrock function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.13 f4Rastrigin()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f4Rastrigin (
    Tinput * data,
    int dimension ) [private]
```

f4: Rastrigin function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.14 f5Griewangk()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f5Griewangk (
    Tinput * data,
    int dimension ) [private]
```

f5: Griewangk function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.15 f6SineEnvelopeSineWave()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f6SineEnvelopeSineWave (
    Tinput * data,
    int dimension ) [private]
```

f6: Sine Envelope Sine Wave function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.16 f7StretchedVSineWave()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f7StretchedVSineWave (
    Tinput * data,
    int dimension ) [private]
```

f7: Stretched V Sine Wave function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.17 f8AckleysOne()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f8AckleysOne (
    Tinput * data,
    int dimension ) [private]
```

f8: Ackley's One function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.18 f9AckleysTwo()

```
template<class Tinput , class Toutput >
Toutput Functions1< Tinput, Toutput >::f9AckleysTwo (
    Tinput * data,
    int dimension ) [private]
```

f9: Ackley's Two function

Parameters

<i>data</i>	Pointer of a data array (datatype defined by the user)
<i>dimension</i>	Size of the data array, the size of the array also represents the number of dimensions

Returns

The result of the mathematical function

4.9.3.19 getFunctionById()

```
template<class Tinput , class Toutput >
Functions1< Tinput, Toutput >::function_pointer Functions1< Tinput, Toutput >::getFunction↵
ById (
    int id )
```

Get the function pointer by id of the benchmark function.

Parameters

<i>id</i>	identification code of the benchmark function
-----------	---

Returns

Function pointer of the selected benchmark function

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

- [functions1.h](#)
- [functions1.cpp](#)

4.10 GAInputParameter< Tinput > Struct Template Reference

Template Genetic Algorithm Input parameter Struct.

```
#include <genetic_algorithm.h>
```

Collaboration diagram for GAInputParameter< Tinput >:

Classes

- struct [Bounds](#)
problem bounds
- struct [Mutation](#)
Mutation parameters.

Public Attributes

- int [ns](#)
number of solutions
- int [dim](#)
problem dimension
- [Bounds](#) [bounds](#)
instance of struct #Bounds
- int [t_max](#)
maximum number of iterations
- double [cr](#)
crossover rate
- [Mutation](#) [m](#)
instance of struct #Mutation
- double [er](#)
elistism rate

4.10.1 Detailed Description

```
template<class Tinput>  
struct GAInputParameter< Tinput >
```

Template Genetic Algorithm Input parameter Struct.

Struct to where all the GA parameters are stored

Author

Chao Huang Lin (chao.huanglin@cwu.edu)

Date

2020-02-01

4.10.2 Member Data Documentation

4.10.2.1 bounds

```
template<class Tinput >  
Bounds GAInputParameter< Tinput >::bounds
```

instance of struct #Bounds

4.10.2.2 cr

```
template<class Tinput >  
double GAInputParameter< Tinput >::cr
```

crossover rate

4.10.2.3 dim

```
template<class Tinput >  
int GAInputParameter< Tinput >::dim
```

problem dimension

4.10.2.4 er

```
template<class Tinput >  
double GAInputParameter< Tinput >::er
```

elistism rate

4.10.2.5 m

```
template<class Tinput >  
Mutation GAInputParameter< Tinput >::m
```

instance of struct #Mutation

4.10.2.6 ns

```
template<class Tinput >
int GAInputParameter< Tinput >::ns
```

number of solutions

4.10.2.7 t_max

```
template<class Tinput >
int GAInputParameter< Tinput >::t_max
```

maximum number of iterations

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

- [genetic_algorithm.h](#)

4.11 GeneticAlgorithm< Tinput, Toutput > Class Template Reference

Template Genetic Algorithm (GA) class.

```
#include <genetic_algorithm.h>
```

Collaboration diagram for GeneticAlgorithm< Tinput, Toutput >:

Classes

- struct [doCompare](#)
- struct [doCompareNewPop](#)

Public Member Functions

- [GeneticAlgorithm](#) (GAInputParameter< Tinput > parameters)
constructor of the class [GeneticAlgorithm](#)
- [~GeneticAlgorithm](#) ()
- Toutput [findBestSolution](#) (int function_id, Tinput range_low, Tinput range_high)
Find the best individuo of the population.
- void [printInputPopulation](#) ()
- void [printNewPopulation](#) ()
- void [printCost](#) (Toutput *cost)
- void [printFitness](#) ()

Private Member Functions

- void [sortPopulationByIndexAsc](#) ()
- void [sortNewPopulationByIndexAsc](#) ()
- void [saveResult](#) (Toutput best_cost, Tinput *best_individuo, std::string result_file)
- void [randomInit](#) (Tinput range_low, Tinput range_high)
- void [evaluateCost](#) (int function_id, Tinput **pop, Toutput *cost)
 - calculate the cost*
- void [getFitness](#) ()
 - calculate fitness*
- void [select](#) (Tinput *parent1, Tinput *parent2)
 - select 2 parents by roulette wheel selection*
- void [selectParent](#) (Tinput *parent)
 - select 1 parent*
- void [crossover](#) (Tinput *parent1, Tinput *parent2, double cr)
 - crossover*
- void [reduce](#) (int elite_sn, Toutput &best_cost, Tinput *best_individuo)
- void [mutate](#) (Tinput *individuo)
- void [printPopulation](#) (Tinput **pop)
- void [keepInRange](#) (Tinput &element)

Private Attributes

- Tinput ** [population](#)
 - pointer to array of pointers*
- [GAInputParameter](#)< Tinput > [parameters](#)
- Toutput * [cost](#)
 - array that contains the cost of each individuo*
- Toutput * [new_population_cost](#)
 - cost of the new population*
- Toutput * [fitness](#)
 - array that contains the fitness of each individuo*
- Toutput [total_fitness](#)
 - total fitness of the population*
- Tinput ** [new_population](#)
- int * [population_asc_index](#)
- int * [new_population_asc_index](#)
- MersenneTwister [ms_random_generator](#)
 - mersenne twister random generator*

4.11.1 Detailed Description

```
template<class Tinput, class Toutput>
class GeneticAlgorithm< Tinput, Toutput >
```

Template Genetic Algorithm (GA) class.

Genetic Algorithms (GA) are the heuristic search and optimization techniques that mimic the process of natural evolution.

Author

Chao Huang Lin (chao.huanglin@cwu.edu)

Date

2020-02-01

4.11.2 Constructor & Destructor Documentation

4.11.2.1 GeneticAlgorithm()

```
template<class Tinput , class Toutput >
GeneticAlgorithm< Tinput, Toutput >::GeneticAlgorithm (
    GAInputParameter< Tinput > parameters )
```

constructor of the class [GeneticAlgorithm](#)

Parameters

<i>parameters</i>	configuration parameters for GA
-------------------	---------------------------------

4.11.2.2 ~GeneticAlgorithm()

```
template<class Tinput , class Toutput >
GeneticAlgorithm< Tinput, Toutput >::~~GeneticAlgorithm
```

4.11.3 Member Function Documentation

4.11.3.1 crossover()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::crossover (
    Tinput * parent1,
    Tinput * parent2,
    double cr ) [private]
```

crossover

4.11.3.2 evaluateCost()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::evaluateCost (
    int function_id,
    Tinput ** pop,
    Toutput * cost ) [private]
```

calculate the cost

4.11.3.3 findBestSolution()

```
template<class Tinput , class Toutput >
Toutput GeneticAlgorithm< Tinput, Toutput >::findBestSolution (
    int function_id,
    Tinput range_low,
    Tinput range_high )
```

Find the best individuo of the population.

Returns

return the best cost (evaluate the function)

4.11.3.4 getFitness()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::getFitness [private]
```

calculate fitness

4.11.3.5 keepInRange()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::keepInRange (
    Tinput & element ) [private]
```

4.11.3.6 mutate()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::mutate (
    Tinput * individuo ) [private]
```

4.11.3.7 printCost()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::printCost (
    Toutput * cost )
```

4.11.3.8 printFitness()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::printFitness
```

4.11.3.9 printInputPopulation()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::printInputPopulation
```

4.11.3.10 printNewPopulation()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::printNewPopulation
```

4.11.3.11 printPopulation()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::printPopulation (
    Tinput ** pop ) [private]
```

4.11.3.12 randomInit()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::randomInit (
    Tinput range_low,
    Tinput range_high ) [private]
```

4.11.3.13 reduce()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::reduce (
    int elite_sn,
    Toutput & best_cost,
    Tinput * best_individuo ) [private]
```

4.11.3.14 saveResult()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::saveResult (
    Toutput best_cost,
    Tinput * best_individuo,
    std::string result_file ) [private]
```

4.11.3.15 select()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::select (
    Tinput * parent1,
    Tinput * parent2 ) [private]
```

select 2 parents by roulette wheel selection

4.11.3.16 selectParent()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::selectParent (
    Tinput * parent ) [private]
```

select 1 parent

4.11.3.17 sortNewPopulationByIndexAsc()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::sortNewPopulationByIndexAsc [private]
```

4.11.3.18 sortPopulationByIndexAsc()

```
template<class Tinput , class Toutput >
void GeneticAlgorithm< Tinput, Toutput >::sortPopulationByIndexAsc [private]
```

4.11.4 Member Data Documentation

4.11.4.1 cost

```
template<class Tinput , class Toutput >
Toutput* GeneticAlgorithm< Tinput, Toutput >::cost [private]
```

array that contains the cost of each individuo

4.11.4.2 fitness

```
template<class Tinput , class Toutput >
Toutput* GeneticAlgorithm< Tinput, Toutput >::fitness [private]
```

array that contains the fitness of each individuo

4.11.4.3 ms_random_generator

```
template<class Tinput , class Toutput >
MersenneTwister GeneticAlgorithm< Tinput, Toutput >::ms_random_generator [private]
```

mersenne twister random generator

4.11.4.4 new_population

```
template<class Tinput , class Toutput >
Tinput** GeneticAlgorithm< Tinput, Toutput >::new_population [private]
```

4.11.4.5 new_population_asc_index

```
template<class Tinput , class Toutput >
int* GeneticAlgorithm< Tinput, Toutput >::new_population_asc_index [private]
```

4.11.4.6 new_population_cost

```
template<class Tinput , class Toutput >
Toutput* GeneticAlgorithm< Tinput, Toutput >::new_population_cost [private]
```

cost of the new population

4.11.4.7 parameters

```
template<class Tinput , class Toutput >
GAInputParameter<Tinput> GeneticAlgorithm< Tinput, Toutput >::parameters [private]
```

4.11.4.8 population

```
template<class Tinput , class Toutput >
Tinput** GeneticAlgorithm< Tinput, Toutput >::population [private]
```

pointer to array of pointers

4.11.4.9 population_asc_index

```
template<class Tinput , class Toutput >
int* GeneticAlgorithm< Tinput, Toutput >::population_asc_index [private]
```

4.11.4.10 total_fitness

```
template<class Tinput , class Toutput >
Toutput GeneticAlgorithm< Tinput, Toutput >::total_fitness [private]
```

total fitness of the population

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

- [genetic_algorithm.h](#)
- [genetic_algorithm.cpp](#)

4.12 GAInputParameter< Tinput >::Mutation Struct Reference

[Mutation](#) parameters.

```
#include <genetic_algorithm.h>
```

Collaboration diagram for GAInputParameter< Tinput >::Mutation:

Public Attributes

- double [rate](#)
- double [range](#)
- double [precision](#)

4.12.1 Detailed Description

```
template<class Tinput>
struct GAInputParameter< Tinput >::Mutation
```

[Mutation](#) parameters.

4.12.2 Member Data Documentation

4.12.2.1 precision

```
template<class Tinput >
double GAInputParameter< Tinput >::Mutation::precision
```

4.12.2.2 range

```
template<class Tinput >
double GAInputParameter< Tinput >::Mutation::range
```

4.12.2.3 rate

```
template<class Tinput >
double GAInputParameter< Tinput >::Mutation::rate
```

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

- [genetic_algorithm.h](#)

4.13 Population< Tinput, Toutput > Class Template Reference

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

Inheritance diagram for Population< Tinput, Toutput >:

Collaboration diagram for Population< Tinput, Toutput >:

Public Member Functions

- [Population](#) (int [n_items](#), int [dimension](#))
- [~Population](#) ()
- void [fillWithRandom](#) (Tinput range_low, Tinput range_high)
- void [printPopulation](#) ()
- Tinput [getData](#) (int i, int j)
- Tinput * [getData](#) (int i)
- void [setData](#) (Tinput *item, int i)

Static Public Member Functions

- static void [swap](#) ([Population](#) &A, [Population](#) &B)

Protected Attributes

- Tinput ** [data](#)
- int [dimension](#)
- int [n_items](#)
total number of items in data

Private Attributes

- MersenneTwister [random_ms](#)
mersenne twister random generator

4.13.1 Constructor & Destructor Documentation

4.13.1.1 Population()

```
template<class Tinput , class Toutput >
Population< Tinput, Toutput >::Population (
    int n_items,
    int dimension )
```

4.13.1.2 ~Population()

```
template<class Tinput , class Toutput >
Population< Tinput, Toutput >::~~Population
```

4.13.2 Member Function Documentation

4.13.2.1 fillWithRandom()

```
template<class Tinput , class Toutput >
void Population< Tinput, Toutput >::fillWithRandom (
    Tinput range_low,
    Tinput range_high )
```

4.13.2.2 getData() [1/2]

```
template<class Tinput , class Toutput >
Tinput * Population< Tinput, Toutput >::getData (
    int i )
```

4.13.2.3 getData() [2/2]

```
template<class Tinput , class Toutput >
Tinput Population< Tinput, Toutput >::getData (
    int i,
    int j )
```

4.13.2.4 printPopulation()

```
template<class Tinput , class Toutput >
void Population< Tinput, Toutput >::printPopulation
```

4.13.2.5 setData()

```
template<class Tinput , class Toutput >
void Population< Tinput, Toutput >::setData (
    Tinput * item,
    int i )
```

4.13.2.6 swap()

```
template<class Tinput , class Toutput >
void Population< Tinput, Toutput >::swap (
    Population< Tinput, Toutput > & A,
    Population< Tinput, Toutput > & B ) [static]
```

4.13.3 Member Data Documentation

4.13.3.1 data

```
template<class Tinput , class Toutput >
Tinput** Population< Tinput, Toutput >::data [protected]
```

4.13.3.2 dimension

```
template<class Tinput , class Toutput >
int Population< Tinput, Toutput >::dimension [protected]
```

4.13.3.3 n_items

```
template<class Tinput , class Toutput >
int Population< Tinput, Toutput >::n_items [protected]
```

total number of items in data

4.13.3.4 random_ms

```
template<class Tinput , class Toutput >
MersenneTwister Population< Tinput, Toutput >::random_ms [private]
```

mersenne twister random generator

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

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

4.14 PopulationBenchmark< Tinput, Toutput > Class Template Reference

```
#include <population_benchmark.h>
```

Inheritance diagram for PopulationBenchmark< Tinput, Toutput >:

Collaboration diagram for PopulationBenchmark< Tinput, Toutput >:

Classes

- struct [doCompare](#)

Public Member Functions

- [PopulationBenchmark](#) (int *n_items*, int *dimension*)
- [~PopulationBenchmark](#) ()
- void [evaluateCost](#) (int *function_id*)
- Toutput [calcCost1Item](#) (int *function_id*, int *i*)
- void [sortIndexByCostAsc](#) ()
- void [saveBest](#) (Toutput *best_cost*, Tinput **best_individuo*)
- void [printCost](#) ()
- void [printIndex](#) ()
- void [setDataAndCost](#) (Tinput **item*, int *i*, Toutput *c*)
- void [calculateMinCost](#) ()
- Toutput [getMinCost](#) ()
- Tinput * [getMinCostData](#) ()

Static Public Member Functions

- static Toutput [calcCostExt](#) (int *function_id*, Tinput **item*, int *dim*)
- static void [swap](#) ([PopulationBenchmark](#) &*A*, [PopulationBenchmark](#) &*B*)

Private Attributes

- Toutput * [cost](#)
array that contains the cost of each item
- Toutput [min_cost](#)
- int [min_cost_i](#)
- int * [asc_index](#)

Additional Inherited Members

4.14.1 Constructor & Destructor Documentation

4.14.1.1 PopulationBenchmark()

```
template<class Tinput , class Toutput >
PopulationBenchmark< Tinput, Toutput >::PopulationBenchmark (
    int n_items,
    int dimension )
```

4.14.1.2 ~PopulationBenchmark()

```
template<class Tinput , class Toutput >
PopulationBenchmark< Tinput, Toutput >::~~PopulationBenchmark
```

4.14.2 Member Function Documentation

4.14.2.1 calcCost1Item()

```
template<class Tinput , class Toutput >
Toutput PopulationBenchmark< Tinput, Toutput >::calcCost1Item (
    int function_id,
    int i )
```

4.14.2.2 calcCostExt()

```
template<class Tinput , class Toutput >
Toutput PopulationBenchmark< Tinput, Toutput >::calcCostExt (
    int function_id,
    Tinput * item,
    int dim ) [static]
```

4.14.2.3 calculateMinCost()

```
template<class Tinput , class Toutput >
void PopulationBenchmark< Tinput, Toutput >::calculateMinCost
```

4.14.2.4 evaluateCost()

```
template<class Tinput , class Toutput >
void PopulationBenchmark< Tinput, Toutput >::evaluateCost (
    int function_id )
```

4.14.2.5 getMinCost()

```
template<class Tinput , class Toutput >
Toutput PopulationBenchmark< Tinput, Toutput >::getMinCost
```


4.14.2.6 getMinCostData()

```
template<class Tinput , class Toutput >
Tinput * PopulationBenchmark< Tinput, Toutput >::getMinCostData
```

4.14.2.7 printCost()

```
template<class Tinput , class Toutput >
void PopulationBenchmark< Tinput, Toutput >::printCost
```

4.14.2.8 printIndex()

```
template<class Tinput , class Toutput >
void PopulationBenchmark< Tinput, Toutput >::printIndex
```

4.14.2.9 saveBest()

```
template<class Tinput , class Toutput >
void PopulationBenchmark< Tinput, Toutput >::saveBest (
    Toutput best_cost,
    Tinput * best_individuo )
```

4.14.2.10 setDataAndCost()

```
template<class Tinput , class Toutput >
void PopulationBenchmark< Tinput, Toutput >::setDataAndCost (
    Tinput * item,
    int i,
    Toutput c )
```

4.14.2.11 sortIndexByCostAsc()

```
template<class Tinput , class Toutput >
void PopulationBenchmark< Tinput, Toutput >::sortIndexByCostAsc
```

4.14.2.12 swap()

```
template<class Tinput , class Toutput >
void PopulationBenchmark< Tinput, Toutput >::swap (
    PopulationBenchmark< Tinput, Toutput > & A,
    PopulationBenchmark< Tinput, Toutput > & B ) [static]
```

< array that contains the cost of each item

4.14.3 Member Data Documentation

4.14.3.1 asc_index

```
template<class Tinput , class Toutput >
int* PopulationBenchmark< Tinput, Toutput >::asc_index [private]
```

4.14.3.2 cost

```
template<class Tinput , class Toutput >
Toutput* PopulationBenchmark< Tinput, Toutput >::cost [private]
```

array that contains the cost of each item

4.14.3.3 min_cost

```
template<class Tinput , class Toutput >
Toutput PopulationBenchmark< Tinput, Toutput >::min_cost [private]
```

4.14.3.4 min_cost_i

```
template<class Tinput , class Toutput >
int PopulationBenchmark< Tinput, Toutput >::min_cost_i [private]
```

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

- [population_benchmark.h](#)
- [population_benchmark.cpp](#)

4.15 Runner< Tinput, Toutput > Class Template Reference

Template class which generates random sample of different dimensions and run a specific benchmark function.

```
#include <runner.h>
```

Collaboration diagram for Runner< Tinput, Toutput >:

Classes

- struct [statistics](#)
struct of statistical analysis

Public Member Functions

- [Runner](#) (int [n_runs](#))
constructor of the class [Runner](#) allocate memory for vectors (array of array)
- [~Runner](#) ()
destructor of the class [Runner](#) free memory for vectors (array of array)
- void [runOptimization](#) (int [algorithm_id](#), std::string [config_file](#), int [function_id](#), int [dimensions](#), Tinput [range_low](#), Tinput [range_high](#))
- void [printSolutions](#) ()

Private Member Functions

- void [computeStatistic](#) (double [time_ms](#))
compute the statistical result of [solutions](#) and save it in [stat_analysis](#)
- void [saveStatistic](#) ()
save statistical result in csv file
- void [fillGAPParameterFromFile](#) (std::string [config_filename](#), [GAInputParameter](#)< Tinput > ¶meters)
- void [fillDEParameterFromFile](#) (std::string [config_filename](#), [DEInputParameter](#)< Tinput > ¶meters)

Private Attributes

- [statistics](#) [stat_analysis](#)
instance of struct #statistics [result of [computeStatistic\(\)](#)]
- int [dimensions](#)
size of the sample or size of the array pointed by each element of vectors [parameter from constructor [Runner\(\)](#)]
- int [n_runs](#)
size of the array vectors [parameter from constructor [Runner\(\)](#)]
- Tinput [range_low](#)
lowest value that the random generator should produce for each element of the sample [parameter from [run\(\)](#)]
- Tinput [range_high](#)
highest value that the random generator should produce for each element of the sample [parameter from [run\(\)](#)]
- Toutput * [solutions](#)
pointer to the array of the result of the benchmark function
- int [function_id](#)
id of the function that will be run [parameter from [run\(\)](#)]
- int [algorithm_id](#)

4.15.1 Detailed Description

```
template<class Tinput, class Toutput>
class Runner< Tinput, Toutput >
```

Template class which generates random sample of different dimensions and run a specific benchmark function.

Author

Chao Huang Lin (chao.huanglin@cwu.edu)

Date

2020-01-31

4.15.2 Constructor & Destructor Documentation

4.15.2.1 Runner()

```
template<class Tinput , class Toutput >
Runner< Tinput, Toutput >::Runner (
    int n_runs )
```

constructor of the class [Runner](#) allocate memory for vectors (array of array)

Parameters

<i>dimensions</i>	/// size of the sample or size of the array pointed by each element of vectors
<i>n_runs</i>	/// number of runs

4.15.2.2 ~Runner()

```
template<class Tinput , class Toutput >
Runner< Tinput, Toutput >::~~Runner
```

destructor of the class [Runner](#) free memory for vectors (array of array)

4.15.3 Member Function Documentation

4.15.3.1 computeStatistic()

```
template<class Tinput , class Toutput >
void Runner< Tinput, Toutput >::computeStatistic (
    double time_ms ) [private]
```

compute the statistical result of [solutions](#) and save it in [stat_analysis](#)

Parameters

<i>time_ms</i>	///< running time in ms
----------------	-------------------------

4.15.3.2 fillDEParameterFromFile()

```
template<class Tinput , class Toutput >
void Runner< Tinput, Toutput >::fillDEParameterFromFile (
    std::string config_filename,
    DEInputParameter< Tinput > & parameters ) [private]
```

4.15.3.3 fillGAPParameterFromFile()

```
template<class Tinput , class Toutput >
void Runner< Tinput, Toutput >::fillGAPParameterFromFile (
    std::string config_filename,
    GAInputParameter< Tinput > & parameters ) [private]
```

4.15.3.4 printSolutions()

```
template<class Tinput , class Toutput >
void Runner< Tinput, Toutput >::printSolutions
```

4.15.3.5 runOptimization()

```
template<class Tinput , class Toutput >
void Runner< Tinput, Toutput >::runOptimization (
    int algorithm_id,
    std::string config_file,
    int function_id,
    int dimensions,
    Tinput range_low,
    Tinput range_high )
```

4.15.3.6 saveStatistic()

```
template<class Tinput , class Toutput >
void Runner< Tinput, Toutput >::saveStatistic [private]
```

save statistical result in csv file

4.15.4 Member Data Documentation

4.15.4.1 algorithm_id

```
template<class Tinput , class Toutput >
int Runner< Tinput, Toutput >::algorithm_id [private]
```

4.15.4.2 dimensions

```
template<class Tinput , class Toutput >
int Runner< Tinput, Toutput >::dimensions [private]
```

size of the sample or size of the array pointed by each element of vectors [parameter from constructor [Runner\(\)](#)]

4.15.4.3 function_id

```
template<class Tinput , class Toutput >
int Runner< Tinput, Toutput >::function_id [private]
```

id of the function that will be run [parameter from [run\(\)](#)]

4.15.4.4 n_runs

```
template<class Tinput , class Toutput >
int Runner< Tinput, Toutput >::n_runs [private]
```

size of the array vectors [parameter from constructor [Runner\(\)](#)]

4.15.4.5 range_high

```
template<class Tinput , class Toutput >
Tinput Runner< Tinput, Toutput >::range_high [private]
```

highest value that the random generator should produce for each element of the sample [parameter from run()]

4.15.4.6 range_low

```
template<class Tinput , class Toutput >
Tinput Runner< Tinput, Toutput >::range_low [private]
```

lowest value that the random generator should produce for each element of the sample [parameter from run()]

4.15.4.7 solutions

```
template<class Tinput , class Toutput >
Toutput* Runner< Tinput, Toutput >::solutions [private]
```

pointer to the array of the result of the benchmark function

4.15.4.8 stat_analysis

```
template<class Tinput , class Toutput >
statistics Runner< Tinput, Toutput >::stat_analysis [private]
```

instance of struct #statistics [result of [computeStatistic\(\)](#)]

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

- [runner.h](#)
- [runner.cpp](#)

4.16 Runner< Tinput, Toutput >::statistics Struct Reference

struct of statistical analysis

Collaboration diagram for Runner< Tinput, Toutput >::statistics:

Public Attributes

- double [mean](#)
- double [std_dev](#)
- double [range](#)
- double [median](#)
- double [time_ms](#)
- double [range_min](#)
- double [range_max](#)

4.16.1 Detailed Description

```
template<class Tinput, class Toutput>
struct Runner< Tinput, Toutput >::statistics
```

struct of statistical analysis

4.16.2 Member Data Documentation

4.16.2.1 mean

```
template<class Tinput , class Toutput >
double Runner< Tinput, Toutput >::statistics::mean
```

4.16.2.2 median

```
template<class Tinput , class Toutput >
double Runner< Tinput, Toutput >::statistics::median
```

4.16.2.3 range

```
template<class Tinput , class Toutput >
double Runner< Tinput, Toutput >::statistics::range
```

4.16.2.4 range_max

```
template<class Tinput , class Toutput >
double Runner< Tinput, Toutput >::statistics::range_max
```


4.16.2.5 range_min

```
template<class Tinput , class Toutput >  
double Runner< Tinput, Toutput >::statistics::range_min
```

4.16.2.6 std_dev

```
template<class Tinput , class Toutput >  
double Runner< Tinput, Toutput >::statistics::std_dev
```

4.16.2.7 time_ms

```
template<class Tinput , class Toutput >  
double Runner< Tinput, Toutput >::statistics::time_ms
```

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

- [runner.h](#)

Chapter 5

File Documentation

5.1 debug.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define `debug`(x)
- #define `debug1`(x)
- #define `debug_var`(x)
- #define `debugfile`(x)

5.1.1 Macro Definition Documentation

5.1.1.1 debug

```
#define debug(  
    x )
```

5.1.1.2 debug1

```
#define debug1(  
    x )
```

5.1.1.3 debug_var

```
#define debug_var(
    x )
```

5.1.1.4 debugfile

```
#define debugfile(
    x )
```

5.2 differential_evolution.cpp File Reference

```
#include "differential_evolution.h"
```

Include dependency graph for differential_evolution.cpp: This graph shows which files directly or indirectly include this file:

5.3 differential_evolution.h File Reference

```
#include "functions1.h"
#include "population_benchmark.cpp"
#include "debug.h"
#include "../lib/mt19937ar_class.h"
#include "util.h"
#include <time.h>
#include <fstream>
```

Include dependency graph for differential_evolution.h: This graph shows which files directly or indirectly include this file:

Classes

- struct [DEInputParameter< Tinput >](#)
Template Differential Evolution Input parameter Struct.
- struct [DEInputParameter< Tinput >::Bounds](#)
problem bounds
- class [DifferentialEvolution< Tinput, Toutput >](#)
Template Differential Evolution class.

5.4 functions1.cpp File Reference

```
#include "functions1.h"
```

Include dependency graph for functions1.cpp: This graph shows which files directly or indirectly include this file:

5.5 functions1.h File Reference

```
#include <math.h>
#include <cstdlib>
```

Include dependency graph for functions1.h: This graph shows which files directly or indirectly include this file:

Classes

- class [Functions1< Tinput, Toutput >](#)
Template class with 18 standard benchmark functions.

5.6 genetic_algorithm.cpp File Reference

```
#include "genetic_algorithm.h"
```

Include dependency graph for genetic_algorithm.cpp: This graph shows which files directly or indirectly include this file:

5.7 genetic_algorithm.h File Reference

```
#include "../lib/mt19937ar_class.h"
#include "debug.h"
#include <iostream>
#include "functions1.h"
#include <time.h>
#include <algorithm>
#include <fstream>
#include "util.h"
```

Include dependency graph for genetic_algorithm.h: This graph shows which files directly or indirectly include this file:

Classes

- struct [GAInputParameter< Tinput >](#)
Template Genetic Algorithm Input parameter Struct.
- struct [GAInputParameter< Tinput >::Bounds](#)
problem bounds
- struct [GAInputParameter< Tinput >::Mutation](#)
Mutation parameters.
- class [GeneticAlgorithm< Tinput, Toutput >](#)
Template Genetic Algorithm (GA) class.
- struct [GeneticAlgorithm< Tinput, Toutput >::doCompare](#)
- struct [GeneticAlgorithm< Tinput, Toutput >::doCompareNewPop](#)

5.8 main.cpp File Reference

```
#include <iostream>
#include <string>
#include "functions1.cpp"
#include "runner.cpp"
#include "genetic_algorithm.cpp"
#include "population.h"
#include "debug.h"
#include "differential_evolution.cpp"
Include dependency graph for main.cpp:
```

Functions

- `int main (int argc, char **argv)`
Project 3, Genetic Algorithm (GA) and Differential Evolution (DE)

5.8.1 Function Documentation

5.8.1.1 main()

```
int main (
    int argc,
    char ** argv )
```

Project 3, Genetic Algorithm (GA) and Differential Evolution (DE)

the main must receive 7 arguments to be able to run GA or DE algorithms with the 18 benchmarking functions arguments: `opt_alg_id` `opt_conf_file` `n_runs` `function_id` `dimensions` `range_min` `range_max` `opt_alg_id` = optimization algorithm id [0 = GA, 1-10 = DE] `opt_conf_file` = optimization algorithm configuration file `n_runs` = numbers of runs o iterations of a specific optimization algorithm and benchmarking function `function_id` = id of benchmarking function `dimensions` = dimensions of each vector o individual (size of each individual array) `range_min` = range lower bound `range_max` = range upper bound

Author

Chao Huang Lin (chao.huanglin@cwu.edu)

Date

2020-01-31

5.9 population.cpp File Reference

```
#include "population.h"
Include dependency graph for population.cpp: This graph shows which files directly or indirectly include this file:
```

5.10 population.h File Reference

```
#include "../lib/mt19937ar_class.h"
#include <time.h>
#include <iostream>
#include "util.h"
```

Include dependency graph for population.h: This graph shows which files directly or indirectly include this file:

Classes

- class [Population< Tinput, Toutput >](#)

5.11 population_benchmark.cpp File Reference

```
#include "population_benchmark.h"
```

Include dependency graph for population_benchmark.cpp: This graph shows which files directly or indirectly include this file:

Classes

- struct [PopulationBenchmark< Tinput, Toutput >::doCompare< Tinput, Toutput >](#)

5.12 population_benchmark.h File Reference

```
#include "population.cpp"
#include "functions1.h"
#include <algorithm>
```

Include dependency graph for population_benchmark.h: This graph shows which files directly or indirectly include this file:

Classes

- class [PopulationBenchmark< Tinput, Toutput >](#)

5.13 runner.cpp File Reference

```
#include "runner.h"
```

Include dependency graph for runner.cpp: This graph shows which files directly or indirectly include this file:

5.14 runner.h File Reference

```
#include <random>
#include <iostream>
#include <algorithm>
#include "functions1.h"
#include "genetic_algorithm.h"
#include "differential_evolution.h"
#include <string>
#include "../lib/mt19937ar_class.h"
#include "debug.h"
#include <fstream>
#include <vector>
#include <sstream>
#include "util.h"
```

Include dependency graph for runner.h: This graph shows which files directly or indirectly include this file:

Classes

- class [Runner< Tinput, Toutput >](#)
Template class which generates random sample of different dimensions and run a specific benchmark function.
- struct [Runner< Tinput, Toutput >::statistics](#)
struct of statistical analysis

5.15 util.h File Reference

```
#include <time.h>
#include <iostream>
```

Include dependency graph for util.h: This graph shows which files directly or indirectly include this file:

Classes

- class [Clock](#)

Functions

- template<class T >
void [printArray](#) (T *array, int n, char separator)

5.15.1 Function Documentation

5.15.1.1 printArray()

```
template<class T >
void printArray (
    T * array,
    int n,
    char separator )
```


Index

- ~DifferentialEvolution
 - DifferentialEvolution< Tinput, Toutput >, [14](#)
- ~GeneticAlgorithm
 - GeneticAlgorithm< Tinput, Toutput >, [35](#)
- ~Population
 - Population< Tinput, Toutput >, [42](#)
- ~PopulationBenchmark
 - PopulationBenchmark< Tinput, Toutput >, [45](#)
- ~Runner
 - Runner< Tinput, Toutput >, [50](#)
- actual_pop
 - DifferentialEvolution< Tinput, Toutput >, [17](#)
- algorithm_id
 - Runner< Tinput, Toutput >, [52](#)
- asc_index
 - PopulationBenchmark< Tinput, Toutput >, [48](#)
- best_cost
 - DifferentialEvolution< Tinput, Toutput >, [17](#)
- bounds
 - DEInputParameter< Tinput >, [11](#)
 - GAInputParameter< Tinput >, [32](#)
- calcCost1Item
 - PopulationBenchmark< Tinput, Toutput >, [46](#)
- calcCostExt
 - PopulationBenchmark< Tinput, Toutput >, [46](#)
- calculateMinCost
 - PopulationBenchmark< Tinput, Toutput >, [46](#)
- Clock, [9](#)
 - start_c, [9](#)
 - stop_c, [10](#)
 - tac, [9](#)
 - tic, [9](#)
- computeStatistic
 - Runner< Tinput, Toutput >, [50](#)
- cost
 - GeneticAlgorithm< Tinput, Toutput >, [38](#)
 - PopulationBenchmark< Tinput, Toutput >, [48](#)
- cr
 - DEInputParameter< Tinput >, [11](#)
 - GAInputParameter< Tinput >, [32](#)
- crossover
 - GeneticAlgorithm< Tinput, Toutput >, [35](#)
- data
 - Population< Tinput, Toutput >, [44](#)
- debug
 - debug.h, [57](#)
- debug.h, [57](#)
 - debug, [57](#)
 - debug1, [57](#)
 - debug_var, [57](#)
 - debugfile, [58](#)
- debug1
 - debug.h, [57](#)
- debug_var
 - debug.h, [57](#)
- debugfile
 - debug.h, [58](#)
- DEInputParameter< Tinput >, [10](#)
 - bounds, [11](#)
 - cr, [11](#)
 - dim, [11](#)
 - pop_size, [11](#)
 - scale_f, [12](#)
 - scale_lambda, [12](#)
 - t_max, [12](#)
- DEInputParameter< Tinput >::Bounds, [7](#)
 - l, [7](#)
 - u, [7](#)
- differential_evolution.cpp, [58](#)
- differential_evolution.h, [58](#)
- DifferentialEvolution
 - DifferentialEvolution< Tinput, Toutput >, [14](#)
- DifferentialEvolution< Tinput, Toutput >, [12](#)
 - ~DifferentialEvolution, [14](#)
 - actual_pop, [17](#)
 - best_cost, [17](#)
 - DifferentialEvolution, [14](#)
 - keepInRange, [14](#)
 - ms_random, [17](#)
 - new_pop, [17](#)
 - param, [17](#)
 - randomR, [14](#)
 - runS10_DE_rand_2_bin, [15](#)
 - runS1_DE_best_1_exp, [15](#)
 - runS2_DE_rand_1_exp, [15](#)
 - runS3_DE_randbest_1_exp, [15](#)
 - runS4_DE_best_2_exp, [15](#)
 - runS5_DE_rand_2_exp, [15](#)
 - runS6_DE_best_1_bin, [16](#)
 - runS7_DE_rand_1_bin, [16](#)
 - runS8_DE_randbest_1_bin, [16](#)
 - runS9_DE_best_2_bin, [16](#)
 - saveResult, [16](#)
- dim
 - DEInputParameter< Tinput >, [11](#)

- GAInputParameter< Tinput >, 32
- dimension
 - Population< Tinput, Toutput >, 44
- dimensions
 - Runner< Tinput, Toutput >, 52
- doCompare
 - GeneticAlgorithm< Tinput, Toutput >::doCompare, 18
 - PopulationBenchmark< Tinput, Toutput >::doCompare< Tinput, Toutput >, 19
- doCompareNewPop
 - GeneticAlgorithm< Tinput, Toutput >::doCompareNewPop, 20
- er
 - GAInputParameter< Tinput >, 32
- evaluateCost
 - GeneticAlgorithm< Tinput, Toutput >, 35
 - PopulationBenchmark< Tinput, Toutput >, 46
- f10EggHolder
 - Functions1< Tinput, Toutput >, 23
- f11Rana
 - Functions1< Tinput, Toutput >, 23
- f12Pathological
 - Functions1< Tinput, Toutput >, 24
- f13Michalewicz
 - Functions1< Tinput, Toutput >, 24
- f14MasterCosineWave
 - Functions1< Tinput, Toutput >, 24
- f15Quartic
 - Functions1< Tinput, Toutput >, 25
- f16Levy
 - Functions1< Tinput, Toutput >, 25
- f17Step
 - Functions1< Tinput, Toutput >, 26
- f18Alphine
 - Functions1< Tinput, Toutput >, 26
- f1Schwefels
 - Functions1< Tinput, Toutput >, 27
- f2DeJongs1
 - Functions1< Tinput, Toutput >, 27
- f3Rosenbrock
 - Functions1< Tinput, Toutput >, 27
- f4Rastrigin
 - Functions1< Tinput, Toutput >, 28
- f5Griewangk
 - Functions1< Tinput, Toutput >, 28
- f6SineEnvelopeSineWave
 - Functions1< Tinput, Toutput >, 29
- f7StretchedVSineWave
 - Functions1< Tinput, Toutput >, 29
- f8AckleysOne
 - Functions1< Tinput, Toutput >, 29
- f9AckleysTwo
 - Functions1< Tinput, Toutput >, 30
- fillDEParameterFromFile
 - Runner< Tinput, Toutput >, 51
- fillGAPParameterFromFile
 - Runner< Tinput, Toutput >, 51
- fillWithRandom
 - Population< Tinput, Toutput >, 42
- findBestSolution
 - GeneticAlgorithm< Tinput, Toutput >, 35
- fitness
 - GeneticAlgorithm< Tinput, Toutput >, 39
- function_id
 - Runner< Tinput, Toutput >, 52
- function_pointer
 - Functions1< Tinput, Toutput >, 23
 - Options1< Tinput, Toutput >, 21
- f10EggHolder, 23
- f11Rana, 23
- f12Pathological, 24
- f13Michalewicz, 24
- f14MasterCosineWave, 24
- f15Quartic, 25
- f16Levy, 25
- f17Step, 26
- f18Alphine, 26
- f1Schwefels, 27
- f2DeJongs1, 27
- f3Rosenbrock, 27
- f4Rastrigin, 28
- f5Griewangk, 28
- f6SineEnvelopeSineWave, 29
- f7StretchedVSineWave, 29
- f8AckleysOne, 29
- f9AckleysTwo, 30
- function_pointer, 23
- getFunctionByld, 30
- functions1.cpp, 58
- functions1.h, 59
- GAInputParameter< Tinput >, 31
 - bounds, 32
 - cr, 32
 - dim, 32
 - er, 32
 - m, 32
 - ns, 32
 - t_max, 33
- GAInputParameter< Tinput >::Bounds, 8
 - l, 8
 - u, 8
- GAInputParameter< Tinput >::Mutation, 40
 - precision, 41
 - range, 41
 - rate, 41
- genetic_algorithm.cpp, 59
- genetic_algorithm.h, 59
- GeneticAlgorithm
 - GeneticAlgorithm< Tinput, Toutput >, 35
- GeneticAlgorithm< Tinput, Toutput >, 33
 - ~GeneticAlgorithm, 35
 - cost, 38
 - crossover, 35
 - evaluateCost, 35

- findBestSolution, 35
- fitness, 39
- GeneticAlgorithm, 35
- getFitness, 36
- keepInRange, 36
- ms_random_generator, 39
- mutate, 36
- new_population, 39
- new_population_asc_index, 39
- new_population_cost, 39
- parameters, 39
- population, 40
- population_asc_index, 40
- printCost, 36
- printFitness, 36
- printInputPopulation, 37
- printNewPopulation, 37
- printPopulation, 37
- randomInit, 37
- reduce, 37
- saveResult, 37
- select, 38
- selectParent, 38
- sortNewPopulationByIndexAsc, 38
- sortPopulationByIndexAsc, 38
- total_fitness, 40
- GeneticAlgorithm< Tinput, Toutput >::doCompare, 18
 - doCompare, 18
 - m_info, 19
 - operator(), 18
- GeneticAlgorithm< Tinput, Toutput >::doCompareNewPop, 20
 - doCompareNewPop, 20
 - m_info, 21
 - operator(), 21
- getData
 - Population< Tinput, Toutput >, 43
- getFitness
 - GeneticAlgorithm< Tinput, Toutput >, 36
- getFunctionById
 - Functions1< Tinput, Toutput >, 30
- getMinCost
 - PopulationBenchmark< Tinput, Toutput >, 46
- getMinCostData
 - PopulationBenchmark< Tinput, Toutput >, 46
- keepInRange
 - DifferentialEvolution< Tinput, Toutput >, 14
 - GeneticAlgorithm< Tinput, Toutput >, 36
- I
 - DEInputParameter< Tinput >::Bounds, 7
 - GAInputParameter< Tinput >::Bounds, 8
- m
 - GAInputParameter< Tinput >, 32
- m_info
 - GeneticAlgorithm< Tinput, Toutput >::doCompare, 19
- GeneticAlgorithm< Tinput, Toutput >::doCompareNewPop, 21
- PopulationBenchmark< Tinput, Toutput >::doCompare< Tinput, Toutput >, 20
- main
 - main.cpp, 60
- main.cpp, 60
 - main, 60
- mean
 - Runner< Tinput, Toutput >::statistics, 54
- median
 - Runner< Tinput, Toutput >::statistics, 54
- min_cost
 - PopulationBenchmark< Tinput, Toutput >, 48
- min_cost_i
 - PopulationBenchmark< Tinput, Toutput >, 48
- ms_random
 - DifferentialEvolution< Tinput, Toutput >, 17
- ms_random_generator
 - GeneticAlgorithm< Tinput, Toutput >, 39
- mutate
 - GeneticAlgorithm< Tinput, Toutput >, 36
- n_items
 - Population< Tinput, Toutput >, 44
- n_runs
 - Runner< Tinput, Toutput >, 52
- new_pop
 - DifferentialEvolution< Tinput, Toutput >, 17
- new_population
 - GeneticAlgorithm< Tinput, Toutput >, 39
- new_population_asc_index
 - GeneticAlgorithm< Tinput, Toutput >, 39
- new_population_cost
 - GeneticAlgorithm< Tinput, Toutput >, 39
- ns
 - GAInputParameter< Tinput >, 32
- operator()
 - GeneticAlgorithm< Tinput, Toutput >::doCompare, 18
 - GeneticAlgorithm< Tinput, Toutput >::doCompareNewPop, 21
 - PopulationBenchmark< Tinput, Toutput >::doCompare< Tinput, Toutput >, 19
- param
 - DifferentialEvolution< Tinput, Toutput >, 17
- parameters
 - GeneticAlgorithm< Tinput, Toutput >, 39
- pop_size
 - DEInputParameter< Tinput >, 11
- Population
 - Population< Tinput, Toutput >, 42
- population
 - GeneticAlgorithm< Tinput, Toutput >, 40
- Population< Tinput, Toutput >, 41
 - ~Population, 42
 - data, 44

- dimension, 44
- fillWithRandom, 42
- getData, 43
- n_items, 44
- Population, 42
- printPopulation, 43
- random_ms, 44
- setData, 43
- swap, 43
- population.cpp, 60
- population.h, 61
- population_asc_index
 - GeneticAlgorithm< Tinput, Toutput >, 40
- population_benchmark.cpp, 61
- population_benchmark.h, 61
- PopulationBenchmark
 - PopulationBenchmark< Tinput, Toutput >, 45
- PopulationBenchmark< Tinput, Toutput >, 44
 - ~PopulationBenchmark, 45
 - asc_index, 48
 - calcCost1Item, 46
 - calcCostExt, 46
 - calculateMinCost, 46
 - cost, 48
 - evaluateCost, 46
 - getMinCost, 46
 - getMinCostData, 46
 - min_cost, 48
 - min_cost_i, 48
 - PopulationBenchmark, 45
 - printCost, 47
 - printIndex, 47
 - saveBest, 47
 - setDataAndCost, 47
 - sortIndexByCostAsc, 47
 - swap, 47
- PopulationBenchmark< Tinput, Toutput >::doCompare<
 - Tinput, Toutput >, 19
 - doCompare, 19
 - m_info, 20
 - operator(), 19
- precision
 - GAInputParameter< Tinput >::Mutation, 41
- printArray
 - util.h, 62
- printCost
 - GeneticAlgorithm< Tinput, Toutput >, 36
 - PopulationBenchmark< Tinput, Toutput >, 47
- printFitness
 - GeneticAlgorithm< Tinput, Toutput >, 36
- printIndex
 - PopulationBenchmark< Tinput, Toutput >, 47
- printInputPopulation
 - GeneticAlgorithm< Tinput, Toutput >, 37
- printNewPopulation
 - GeneticAlgorithm< Tinput, Toutput >, 37
- printPopulation
 - GeneticAlgorithm< Tinput, Toutput >, 37
- Population< Tinput, Toutput >, 43
- printSolutions
 - Runner< Tinput, Toutput >, 51
- random_ms
 - Population< Tinput, Toutput >, 44
- randomInit
 - GeneticAlgorithm< Tinput, Toutput >, 37
- randomR
 - DifferentialEvolution< Tinput, Toutput >, 14
- range
 - GAInputParameter< Tinput >::Mutation, 41
 - Runner< Tinput, Toutput >::statistics, 54
- range_high
 - Runner< Tinput, Toutput >, 52
- range_low
 - Runner< Tinput, Toutput >, 53
- range_max
 - Runner< Tinput, Toutput >::statistics, 54
- range_min
 - Runner< Tinput, Toutput >::statistics, 54
- rate
 - GAInputParameter< Tinput >::Mutation, 41
- reduce
 - GeneticAlgorithm< Tinput, Toutput >, 37
- Runner
 - Runner< Tinput, Toutput >, 50
- Runner< Tinput, Toutput >, 49
 - ~Runner, 50
 - algorithm_id, 52
 - computeStatistic, 50
 - dimensions, 52
 - fillDEParameterFromFile, 51
 - fillGAPParameterFromFile, 51
 - function_id, 52
 - n_runs, 52
 - printSolutions, 51
 - range_high, 52
 - range_low, 53
 - Runner, 50
 - runOptimization, 51
 - saveStatistic, 51
 - solutions, 53
 - stat_analysis, 53
- Runner< Tinput, Toutput >::statistics, 53
 - mean, 54
 - median, 54
 - range, 54
 - range_max, 54
 - range_min, 54
 - std_dev, 55
 - time_ms, 55
- runner.cpp, 61
- runner.h, 62
- runOptimization
 - Runner< Tinput, Toutput >, 51
- runS10_DE_rand_2_bin
 - DifferentialEvolution< Tinput, Toutput >, 15
- runS1_DE_best_1_exp

- DifferentialEvolution< Tinput, Toutput >, [15](#)
- runS2_DE_rand_1_exp
 - DifferentialEvolution< Tinput, Toutput >, [15](#)
- runS3_DE_randbest_1_exp
 - DifferentialEvolution< Tinput, Toutput >, [15](#)
- runS4_DE_best_2_exp
 - DifferentialEvolution< Tinput, Toutput >, [15](#)
- runS5_DE_rand_2_exp
 - DifferentialEvolution< Tinput, Toutput >, [15](#)
- runS6_DE_best_1_bin
 - DifferentialEvolution< Tinput, Toutput >, [16](#)
- runS7_DE_rand_1_bin
 - DifferentialEvolution< Tinput, Toutput >, [16](#)
- runS8_DE_randbest_1_bin
 - DifferentialEvolution< Tinput, Toutput >, [16](#)
- runS9_DE_best_2_bin
 - DifferentialEvolution< Tinput, Toutput >, [16](#)
- saveBest
 - PopulationBenchmark< Tinput, Toutput >, [47](#)
- saveResult
 - DifferentialEvolution< Tinput, Toutput >, [16](#)
 - GeneticAlgorithm< Tinput, Toutput >, [37](#)
- saveStatistic
 - Runner< Tinput, Toutput >, [51](#)
- scale_f
 - DEInputParameter< Tinput >, [12](#)
- scale_lambda
 - DEInputParameter< Tinput >, [12](#)
- select
 - GeneticAlgorithm< Tinput, Toutput >, [38](#)
- selectParent
 - GeneticAlgorithm< Tinput, Toutput >, [38](#)
- setData
 - Population< Tinput, Toutput >, [43](#)
- setDataAndCost
 - PopulationBenchmark< Tinput, Toutput >, [47](#)
- solutions
 - Runner< Tinput, Toutput >, [53](#)
- sortIndexByCostAsc
 - PopulationBenchmark< Tinput, Toutput >, [47](#)
- sortNewPopulationByIndexAsc
 - GeneticAlgorithm< Tinput, Toutput >, [38](#)
- sortPopulationByIndexAsc
 - GeneticAlgorithm< Tinput, Toutput >, [38](#)
- start_c
 - Clock, [9](#)
- stat_analysis
 - Runner< Tinput, Toutput >, [53](#)
- std_dev
 - Runner< Tinput, Toutput >::statistics, [55](#)
- stop_c
 - Clock, [10](#)
- swap
 - Population< Tinput, Toutput >, [43](#)
 - PopulationBenchmark< Tinput, Toutput >, [47](#)
- t_max
 - DEInputParameter< Tinput >, [12](#)
- GAInputParameter< Tinput >, [33](#)
- tac
 - Clock, [9](#)
- tic
 - Clock, [9](#)
- time_ms
 - Runner< Tinput, Toutput >::statistics, [55](#)
- total_fitness
 - GeneticAlgorithm< Tinput, Toutput >, [40](#)
- u
 - DEInputParameter< Tinput >::Bounds, [7](#)
 - GAInputParameter< Tinput >::Bounds, [8](#)
- util.h, [62](#)
- printArray, [62](#)