

## Project 4

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>Class Documentation</b>	<b>3</b>
2.1	benchmarkfunctions Struct Reference . . . . .	3
2.2	FireflyAlgorithm Class Reference . . . . .	3
2.2.1	Member Function Documentation . . . . .	4
2.2.1.1	fileReader() . . . . .	4
2.2.1.2	fireFlyAlgorithm() . . . . .	4
2.2.1.3	getgBest() . . . . .	4
2.2.1.4	getIntensity() . . . . .	4
2.2.1.5	getrandomdouble() . . . . .	5
2.2.1.6	getrandomInt() . . . . .	5
2.2.1.7	setgBest() . . . . .	6
2.2.1.8	sortByFitness() . . . . .	6
2.3	ParticleSwarm Class Reference . . . . .	6
2.3.1	Constructor & Destructor Documentation . . . . .	7
2.3.1.1	ParticleSwarm() . . . . .	7
2.3.2	Member Function Documentation . . . . .	7
2.3.2.1	fileReader() . . . . .	7
2.3.2.2	getrandomdouble() . . . . .	7
2.3.2.3	getrandomInt() . . . . .	8
2.3.2.4	PSOAlgorithm() . . . . .	8
2.4	Population Class Reference . . . . .	8

2.4.1	Constructor & Destructor Documentation . . . . .	9
2.4.1.1	Population() [1/2] . . . . .	9
2.4.1.2	Population() [2/2] . . . . .	10
2.4.2	Member Function Documentation . . . . .	10
2.4.2.1	getDimensions() . . . . .	10
2.4.2.2	getFitness() . . . . .	10
2.4.2.3	getIterations() . . . . .	10
2.4.2.4	getMax() . . . . .	11
2.4.2.5	getMin() . . . . .	11
2.4.2.6	getpBest() . . . . .	11
2.4.2.7	getPopSize() . . . . .	11
2.4.2.8	getrandomdouble() . . . . .	12
2.4.2.9	getrandomInt() . . . . .	12
2.4.2.10	getSwarm() . . . . .	12
2.4.2.11	getVelocity() . . . . .	13
2.4.2.12	setDimensions() . . . . .	13
2.4.2.13	setFitness() . . . . .	13
2.4.2.14	setIterations() . . . . .	13
2.4.2.15	setMax() . . . . .	15
2.4.2.16	setMin() . . . . .	15
2.4.2.17	setpBest() . . . . .	15
2.4.2.18	setPopSize() . . . . .	16
2.4.2.19	setSwarm() . . . . .	16
2.4.2.20	setVelocity() . . . . .	16
2.4.2.21	solveFitness() . . . . .	17
2.5	run Class Reference . . . . .	17
2.5.1	Member Function Documentation . . . . .	17
2.5.1.1	runProject4() . . . . .	17
2.6	strs Struct Reference . . . . .	18

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">benchmarkfunctions</a>	3
<a href="#">FireflyAlgorithm</a>	3
<a href="#">ParticleSwarm</a>	6
<a href="#">Population</a>	8
<a href="#">run</a>	17
<a href="#">strs</a>	18



## Chapter 2

# Class Documentation

### 2.1 benchmarkfunctions Struct Reference

#### Public Attributes

- string **name**
- double **min**
- double **max**
- double(\* **foo** )(int dim, double myArray[ ])

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

- FunctionStructs.h

### 2.2 FireflyAlgorithm Class Reference

#### Public Member Functions

- void [fireFlyAlgorithm](#) (double min, double max, double(\*foo)(int dim, double myArray[ ]))
- void [fileReader](#) ()
- double \* [getgBest](#) ()
- void [setgBest](#) (double globalBest[ ], int dim)
- int \* [sortByFitness](#) ([Population](#) &population, double(\*foo)(int dim, double myArray[ ]))
- int [getrandomInt](#) (int min, int max)
- double [getrandomdouble](#) (double min, double max)
- double [getIntensity](#) (double firefly[ ], double v, int dim, double(\*foo)(int dim, double myArray[ ]))

#### Public Attributes

- int **dim**
- int **numParticles**
- int **iter**
- double **gama**
- double **alpha**
- double **betamin**

## 2.2.1 Member Function Documentation

### 2.2.1.1 fileReader()

```
void FireflyAlgorithm::fileReader ( )
```

Read in data from config file

### 2.2.1.2 fireFlyAlgorithm()

```
void FireflyAlgorithm::fireFlyAlgorithm (
    double min,
    double max,
    double(*) (int dim, double myArray[]) foo )
```

Runs the firefly algorithm for a fitness function

#### Parameters

<i>min</i>	a double representing the minimum of the range
<i>max</i>	a double representing the maximum of the range
<i>foo</i>	a pointer to the current fitness function

### 2.2.1.3 getgBest()

```
double * FireflyAlgorithm::getgBest ( )
```

Returns the global best value

#### Returns

an array containing the global best value

### 2.2.1.4 getIntensity()

```
double FireflyAlgorithm::getIntensity (
    double firefly[],
    double v,
    int dim,
    double(*) (int dim, double myArray[]) foo )
```

Returns the light intensity level of a firefly



## Parameters

<i>firefly</i>	an array containing the firefly
<i>v</i>	the distance between two fireflies
<i>dim</i>	the number of dimensions
<i>foo</i>	a pointer to the current fitness function

## Returns

the intensity value of the firefly as a double

## 2.2.1.5 getrandomdouble()

```
double FireflyAlgorithm::getrandomdouble (
    double min,
    double max )
```

returns a randomly generated double

## Parameters

<i>min</i>	the minimum value to be returned
<i>max</i>	the maximum value to be returned

## Returns

A random double between min and max

## 2.2.1.6 getrandomInt()

```
int FireflyAlgorithm::getrandomInt (
    int min,
    int max )
```

returns a randomly generated integer

## Parameters

<i>min</i>	the minimum value to be returned
<i>max</i>	the maximum value to be returned

## Returns

A random integer between min and max

### 2.2.1.7 setgBest()

```
void FireflyAlgorithm::setgBest (
    double globalBest[],
    int dim )
```

Sets the global best value of a particle in the swarm

#### Parameters

<i>globalBest</i>	an array containing the new best value
<i>dim</i>	the number of dimensions

### 2.2.1.8 sortByFitness()

```
int * FireflyAlgorithm::sortByFitness (
    Population & population,
    double(*) (int dim, double myArray[]) foo )
```

Returns an array contain the sorted order of the population based on fitness

#### Parameters

<i>population</i>	a reference to the population to be sorted
<i>foo</i>	a pointer to the current fitness function

#### Returns

an integer array containing the order of the population if it was sorted

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

- FireflyAlgorithm.h
- FireflyAlgorithm.cpp

## 2.3 ParticleSwarm Class Reference

### Public Member Functions

- [ParticleSwarm](#) ()
- void [PSOAlgorithm](#) (double min, double max, double(\*foo)(int dim, double myArray[]))
- void [fileReader](#) ()
- int [getrandomInt](#) (int min, int max)
- double [getrandomdouble](#) (double min, double max)

## Public Attributes

- int **numParticles**
- int **dim**
- int **iter**
- double **c1**
- double **c2**
- double \* **pBest**
- double \* **gBest**

## 2.3.1 Constructor & Destructor Documentation

### 2.3.1.1 ParticleSwarm()

```
ParticleSwarm::ParticleSwarm ( )
```

Constructor for Particle Swarm calls file reader to read in parameters.

## 2.3.2 Member Function Documentation

### 2.3.2.1 fileReader()

```
void ParticleSwarm::fileReader ( )
```

Read in data from config file

### 2.3.2.2 getRandomdouble()

```
double ParticleSwarm::getRandomdouble (
    double min,
    double max )
```

returns a randomly generated double

#### Parameters

<i>min</i>	the minimum value to be returned
<i>max</i>	the maximum value to be returned

#### Returns

A random double between min and max

### 2.3.2.3 getRandomInt()

```
int ParticleSwarm::getRandomInt (
    int min,
    int max )
```

returns a randomly generated integer

#### Parameters

<i>min</i>	the minimum value to be returned
<i>max</i>	the maximum value to be returned

#### Returns

A random integer between min and max

### 2.3.2.4 PSOAlgorithm()

```
void ParticleSwarm::PSOAlgorithm (
    double min,
    double max,
    double(*) (int dim, double myArray[]) foo )
```

Runs Partical Swarm algorithm for a given fitness function

#### Parameters

<i>min</i>	a double representing the minimum of the range
<i>max</i>	a double representing the maximum of the range
<i>foo</i>	a pointer to the current fitness function

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

- ParticleSwarm.h
- ParticleSwarm.cpp

## 2.4 Population Class Reference

### Public Member Functions

- [Population](#) (int popSize, int dim, double min, double max)
- [Population](#) (const [Population](#) &pop)

- void [solveFitness](#) (int dim, int popSize, double min, double max, double(\*foo)(int dim, double myArray[ ]))
- double \* [getFitness](#) ()
- int [getrandomInt](#) (int min, int max)
- double [getrandomdouble](#) (double min, double max)
- double [getMin](#) ()
- double [getMax](#) ()
- int [getPopSize](#) ()
- int [getDimensions](#) ()
- int [getIterations](#) ()
- void [setDimensions](#) (int dimensions1)
- void [setPopSize](#) (int popsize)
- void [setIterations](#) (int iter)
- void [setMin](#) (double minimum)
- void [setMax](#) (double maximum)
- void [setFitness](#) (double fit[ ], int size)
- double \*\* [getSwarm](#) ()
- void [setSwarm](#) (double particle[ ], int index, int dim)
- double \* [getVelocity](#) (int index)
- void [setVelocity](#) (double vel[ ], int index, int dim)
- double \* [getpBest](#) (int index)
- void [setpBest](#) (double pbest[ ], int index, int dim)

## 2.4.1 Constructor & Destructor Documentation

### 2.4.1.1 [Population\(\)](#) [1/2]

```
Population::Population (
    int popsize,
    int dim,
    double min,
    double max )
```

Returns a two dimensional array of type double containing popsize arrays of dim elements, between the range of min to max.

#### Parameters

<i>popSize</i>	an integer representing the size of the population
<i>dim</i>	an integer representing the number of dimensions
<i>min</i>	a double representing the minimum of the range
<i>max</i>	a double representing the maximum of the range

#### Returns

an 2D array of doubles of size dim by popSize representing a new population

### 2.4.1.2 Population() [2/2]

```
Population::Population (
    const Population & pop )
```

Creates a copy of a population

#### Parameters

<i>pop</i>	a <a href="#">Population</a> to be copied
------------	---

## 2.4.2 Member Function Documentation

### 2.4.2.1 getDimensions()

```
int Population::getDimensions ( )
```

Returns the number of dimensions

#### Returns

The current number of dimensions

### 2.4.2.2 getFitness()

```
double * Population::getFitness ( )
```

Returns the fitness array

#### Returns

The current fitness array

### 2.4.2.3 getIterations()

```
int Population::getIterations ( )
```

Returns the number of iterations

#### Returns

The current number of iterations

#### 2.4.2.4 getMax()

```
double Population::getMax ( )
```

Returns the maximum

##### Returns

The current maximum

#### 2.4.2.5 getMin()

```
double Population::getMin ( )
```

Returns the minimum

##### Returns

The current minimum

#### 2.4.2.6 getpBest()

```
double * Population::getpBest (
    int index )
```

Returns the particles best value

##### Parameters

<i>index</i>	the index of the particle
--------------	---------------------------

##### Returns

an array containing the particles' best value

#### 2.4.2.7 getPopSize()

```
int Population::getPopSize ( )
```

Returns the [Population](#) Size

##### Returns

The current population size

#### 2.4.2.8 getRandomdouble()

```
double Population::getRandomdouble (
    double min,
    double max )
```

returns a randomly generated double

##### Parameters

<i>min</i>	the minimum value to be returned
<i>max</i>	the maximum value to be returned

##### Returns

A random double between min and max

#### 2.4.2.9 getRandomInt()

```
int Population::getRandomInt (
    int min,
    int max )
```

returns a randomly generated integer

##### Parameters

<i>min</i>	the minimum value to be returned
<i>max</i>	the maximum value to be returned

##### Returns

A random integer between min and max

#### 2.4.2.10 getSwarm()

```
double ** Population::getSwarm ( )
```

Returns the swarm matrix

##### Returns

A Matrix containing the current swarm



#### 2.4.2.11 `getVelocity()`

```
double * Population::getVelocity (
    int index )
```

Returns the velocity of a given particle

##### Parameters

<i>dim</i>	the number of dimensions
------------	--------------------------

##### Returns

an array containing the particles' velocity( at a given index)

#### 2.4.2.12 `setDimensions()`

```
void Population::setDimensions (
    int dimensions1 )
```

Sets the number of dimensions

##### Parameters

<i>dimensions1</i>	a value of type integer
--------------------	-------------------------

#### 2.4.2.13 `setFitness()`

```
void Population::setFitness (
    double fit[],
    int size )
```

Sets the fitness array

##### Parameters

<i>fit</i>	an array of type double
------------	-------------------------

#### 2.4.2.14 `setIterations()`

```
void Population::setIterations (
    int iter )
```

Sets the number of iterations

## Parameters

<i>iter</i>	a value of type integer
-------------	-------------------------

## 2.4.2.15 setMax()

```
void Population::setMax (
    double maximum )
```

Sets the maximum of the current fitness function

## Parameters

<i>maximum</i>	a value of type double
----------------	------------------------

## 2.4.2.16 setMin()

```
void Population::setMin (
    double minimum )
```

Sets the minimum of the current fitness function

## Parameters

<i>minimum</i>	a value of type double
----------------	------------------------

## 2.4.2.17 setpBest()

```
void Population::setpBest (
    double pbest[],
    int index,
    int dim )
```

Sets the best value of a particle in the swarm

## Parameters

<i>pbest</i>	an array containing the new best value
<i>index</i>	an int representing the index of the corresponding particle
<i>dim</i>	the number of dimensions

#### 2.4.2.18 setPopSize()

```
void Population::setPopSize (
    int popsize )
```

Sets the [Population](#) Size

##### Parameters

<i>popsize</i>	a value of type integer
----------------	-------------------------

#### 2.4.2.19 setSwarm()

```
void Population::setSwarm (
    double particle[],
    int index,
    int dim )
```

Sets a specific particle in the swarm

##### Parameters

<i>particle</i>	an array containing a particle
<i>index</i>	an int representing the index of the swarm to be replaced with the new particle
<i>dim</i>	the number of dimensions

#### 2.4.2.20 setVelocity()

```
void Population::setVelocity (
    double vel[],
    int index,
    int dim )
```

Sets the velocity of a particle in the swarm

##### Parameters

<i>vel</i>	an array containing velocity
<i>index</i>	an int representing the index of the corresponding particle
<i>dim</i>	the number of dimensions

## 2.4.2.21 solveFitness()

```
void Population::solveFitness (
    int dim,
    int popSize,
    double min,
    double max,
    double(*) (int dim, double myArray[]) foo )
```

Returns a array of type double containing dim elements representing the fitnesses of each member of the population.

## Parameters

<i>dim</i>	an integer representing the number of dimensions
<i>population</i>	an 2D double array containing the current population
<i>popSize</i>	an integer representing the size of the population
<i>min</i>	a double representing the minimum of the range
<i>max</i>	a double representing the maximum of the range
<i>foo</i>	a pointer to the current fitness function

## Returns

an 2D array of doubles of size dim by popSize representing a new population

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

- Population.h
- Population.cpp

## 2.5 run Class Reference

## Public Member Functions

- void [runProject4](#) ()

## 2.5.1 Member Function Documentation

## 2.5.1.1 runProject4()

```
void run::runProject4 ( )
```

Runs the Particle swarm and firefly algorithms and places their results in the output files

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

- run.h
- run.cpp

## 2.6 strs Struct Reference

### Public Attributes

- double **value**
- int **index**

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

- FireflyAlgorithm.h

# Index

benchmarkfunctions, [3](#)

fileReader

    FireflyAlgorithm, [4](#)

    ParticleSwarm, [7](#)

fireFlyAlgorithm

    FireflyAlgorithm, [4](#)

FireflyAlgorithm, [3](#)

    fileReader, [4](#)

    fireFlyAlgorithm, [4](#)

    getIntensity, [4](#)

    getgBest, [4](#)

    getrandomInt, [5](#)

    getrandomdouble, [5](#)

    setgBest, [6](#)

    sortByFitness, [6](#)

getDimensions

    Population, [10](#)

getFitness

    Population, [10](#)

getIntensity

    FireflyAlgorithm, [4](#)

getIterations

    Population, [10](#)

getMax

    Population, [10](#)

getMin

    Population, [11](#)

getPopSize

    Population, [11](#)

getSwarm

    Population, [12](#)

getVelocity

    Population, [12](#)

getgBest

    FireflyAlgorithm, [4](#)

getpBest

    Population, [11](#)

getrandomInt

    FireflyAlgorithm, [5](#)

    ParticleSwarm, [8](#)

    Population, [12](#)

getrandomdouble

    FireflyAlgorithm, [5](#)

    ParticleSwarm, [7](#)

    Population, [11](#)

PSOAlgorithm

    ParticleSwarm, [8](#)

ParticleSwarm, [6](#)

    fileReader, [7](#)

    getrandomInt, [8](#)

    getrandomdouble, [7](#)

    PSOAlgorithm, [8](#)

    ParticleSwarm, [7](#)

Population, [8](#)

    getDimensions, [10](#)

    getFitness, [10](#)

    getIterations, [10](#)

    getMax, [10](#)

    getMin, [11](#)

    getPopSize, [11](#)

    getSwarm, [12](#)

    getVelocity, [12](#)

    getpBest, [11](#)

    getrandomInt, [12](#)

    getrandomdouble, [11](#)

    Population, [9](#)

    setDimensions, [13](#)

    setFitness, [13](#)

    setIterations, [13](#)

    setMax, [15](#)

    setMin, [15](#)

    setPopSize, [16](#)

    setSwarm, [16](#)

    setVelocity, [16](#)

    setpBest, [15](#)

    solveFitness, [16](#)

run, [17](#)

    runProject4, [17](#)

runProject4

    run, [17](#)

setDimensions

    Population, [13](#)

setFitness

    Population, [13](#)

setIterations

    Population, [13](#)

setMax

    Population, [15](#)

setMin

    Population, [15](#)

setPopSize

    Population, [16](#)

setSwarm

    Population, [16](#)

setVelocity

- Population, [16](#)
- setgBest
  - FireflyAlgorithm, [6](#)
- setpBest
  - Population, [15](#)
- solveFitness
  - Population, [16](#)
- sortByFitness
  - FireflyAlgorithm, [6](#)
- strs, [18](#)