Project 4

Generated by Doxygen 1.8.14

Contents

Chapter 1

Class Index

1.1 Class List

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

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

min	a double representing the minimum of the range
max	a double representing the maximum of the range
foo	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

firefly	an array containing the firefly
v	the distance between two fireflies
dim	the number of dimensions
foo	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

min	the minimum value to be returned
max	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

min	the minimum value to be returned
max	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

globalBest	an array containing the new best value
dim	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

population	a reference to the population to be sorted
foo	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

min	the minimum value to be returned
max	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

min	the minimum value to be returned
max	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

min	a double representing the minimum of the range
max	a double representing the maximum of the range
foo	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

popSize	an integer representing the size of the population
dim	an integer representing the number of dimensions
min	a double representing the minimum of the range
max	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

pop	a Population 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

index	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

min	the minimum value to be returned
max	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

min	the minimum value to be returned
max	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 martix

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

dim	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

dimensions1	a value of type integer

2.4.2.13 setFitness()

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

Sets the fitness array

Parameters

fit	an array of type double

2.4.2.14 setIterations()

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

Sets the number of iterations

Parameters

iter	a value of type integer

2.4.2.15 setMax()

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

Sets the maximum of the current fitness function

Parameters

maximum	a value of type double

2.4.2.16 setMin()

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

Sets the minimum of the current fitness function

Parameters

minimum	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

pbest	an array containing the new best value
index	an int representing the index of the corisponding paticle
dim	the number of dimensions

2.4.2.18 setPopSize()

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

Sets the Population Size

Parameters

popsize	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

particle	an array containing a particle
index	an int representing the index of the swarm to be replaced with the new particle
dim	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

vel	an array containing velocity
index	an int representing the index of the corisponding paticle
dim	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

dim	an integer representing the number of dimensions
population	an 2D double array containing the current population
popSize	an integer representing the size of the population
min	a double representing the minimum of the range
max	a double representing the maximum of the range
foo	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