Particle Swarm Optimization

0.0.1

Generated by Doxygen 1.8.17

Chapter 1

Particle Swarm Optimization

Implementation of the Particle Swarm Optimization algorithm in bounded spaces without additional constraints. Suitable for optimization over continuous subspaces. For a review of the algorithm, please refer to Ab Wahab *et al.* 2015.

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 Particle Class Reference

Represents an individual particle in the swarm.

Public Member Functions

- Particle (std::vector< std::pair< double, double >> bounds, double(*fitnessFunction)(std::vector< double >))

  Constructor that initialized a particle with random position and velocity within the allowed bounds.
- void updateVelocity (double inertia, double c1, double c2, std::vector< double > &allTimeBest)

  Update the velocity of the particle according to the properties of the swarm.
- void updatePosition ()

  Update the current position by adding the velocity to it.
- void updateFitness (double(*fitnessFunction)(std::vector< double >))

  Update the fitness (optimality) of the particle according to the current position.
- void updateBest ()

  Update the best position and fitness with the current ones if they exceed the old optimal.
- void printParticleData ()

  Output particle data to console (for debugging purposes).
- double getBestFitness ()

  Retrieve the best optimal the particle has encountered so far.
- std::vector< double > getBestPosition ()

  Retrieve the best position the particle has encountered so far.

Private Attributes

- std::vector< double > **currentPosition**
- std::vector< double > **bestPosition**
- double **currentFitness**
- double **bestFitness**
- std::vector< double > **velocity**

3.1.1 Detailed Description

Represents an individual particle in the swarm.

All methods are public since they need to be accessed from the Swarm class.

Author

Nieves Montes Gómez

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Particle()

```
Particle::Particle (
            std::vector< std::pair< double, double >> bounds,
            double(*)(std::vector< double >) fitnessFunction )
```

Constructor that initialized a particle with random position and velocity within the allowed bounds.

Parameters

bounds	Vector of pairs, indicating the allowed ranges in each dimensions of the search space.
fitnessFunction	Pointer to the function to be optimized. It must take in a vector of doubles and returns a double.

3.1.3 Member Function Documentation

3.1.3.1 getBestFitness()

```
double Particle::getBestFitness ( )
```

Retrieve the best optimal the particle has encountered so far.

Returns

Best optimal found by the particle.

3.1.3.2 getBestPosition()

```
std::vector< double > Particle::getBestPosition ( )
```

Retrieve the best position the particle has encountered so far.

Returns

The vector of the most optimal point found by the particle.

3.1.3.3 updateFitness()

```
void Particle::updateFitness (
            double(*)(std::vector< double >) fitnessFunction )
```

Update the fitness (*optimality*) of the particle according to the current position.

Parameters

fitnessFunction	Function to be optimized.

3.1.3.4 updateVelocity()

```
void Particle::updateVelocity (
            double inertia,
            double c1,
            double c2,
            std::vector< double > & allTimeBestPosition )
```

Update the velocity of the particle according to the properties of the swarm.

Parameters

inertia	Inertia weight hyperparameter.
c1	Weight of the individual's particle best position so far in the update.
c2	Weight of the best position so far across the whole swarm in the update.
allTimeBestPosition	Best position found so far across the whole swarm.

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

- /home/nmontes/OneDrive/Documentos/particle-swarm/particle.cpp

3.2 Swarm Class Reference

Represents a swarm of particles moving accross a bounded space and trying to optimize some given function.

Public Member Functions

- Swarm (std::vector< std::pair< double, double >> bounds, int n, double(*fitness)(std::vector< double >), double omega, double d1, double d2)

  Construct the swarm by randomly initializing the particles, finding and storing the best so far.
- std::tuple< std::vector< double >, double, int > optimizationSearch (int maxTotalIter, int maxPartialIter)

  Perform Particle Swarm Optimization.

Private Member Functions

- Particle * findBestCurrentParticle ()

  Find the particle that is currently in the best position, across the whole swarm.
- void updateBestAllTime ()

  Update the best optimal position and optimal function value.
- int getSpaceDimensions ()

  Get the dimensionality of the search space.
- int getPopulationSize ()

  Get the number of particles in the swarm.

Private Attributes

- std::vector< std::pair< double, double > > **spaceBounds**
- std::vector< Particle > **population**
- double **inertia**
- double **c1**
- double **c2**
- double(* **fitnessFunction**)(std::vector< double >)
- std::vector< double > **bestPositionAllTime**
- double **bestFitnessAllTime**

3.2.1 Detailed Description

Represents a swarm of particles moving accross a bounded space and trying to optimize some given function.

Author

Nieves Montes Gómez

3.2.2 Constructor & Destructor Documentation

3.2.2.1 Swarm()

```
Swarm::Swarm (
            std::vector< std::pair< double, double >> bounds,
            int popSize,
            double(*)(std::vector< double >) fitness,
            double omega,
            double d1,
            double d2 )
```

Construct the swarm by randomly initializing the particles, finding and storing the best so far.

Parameters

bounds	Vector of pairs, indicating the allowed ranges in each dimensions of the search space.
popSize	Number of particles in the swarm.
fitness	Pointer to the function to be optimized. It must take in a vector of doubles and returns a double.
omega	Inertia weight hyperparameter.
d1	Weight of the individual's particle best position so far in the update.
d2	Weight of the best position so far across the whole swarm in the update.

3.2.3 Member Function Documentation

3.2.3.1 findBestCurrentParticle()

```
Particle * Swarm::findBestCurrentParticle ( )  [private]
```

Find the particle that is currently in the best position, across the whole swarm.

Returns

A pointer to the partile in the best position, currently.

3.2.3.2 getPopulationSize()

```
int Swarm::getPopulationSize ( )  [private]
```

Get the number of particles in the swarm.

Returns

The size of the swarm.

3.2.3.3 getSpaceDimensions()

```
int Swarm::getSpaceDimensions ( )  [private]
```

Get the dimensionality of the search space.

Returns

The number of dimensions of the search space.

3.2.3.4 optimizationSearch()

```
std::tuple< std::vector< double >, double, int > Swarm::optimizationSearch (
            int maxTotalIter,
            int maxPartialIter )
```

Perform Particle Swarm Optimization.

Parameters

maxTotalIter	Maximum iterations before the search is halted.
maxPartialIter	Maximum iterations *without an update of the provisional solution* before the search is halted.

Returns

A tuple with the optimal point found, its optimal function value and the number of iterations before halting.

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

- /home/nmontes/OneDrive/Documentos/particle-swarm/swarm.cpp

Index