Gnowee

Generated by Doxygen 1.8.6

Wed May 10 2017 20:42:18

Contents

1	Main	Page													1
2	Mod	ule Inde	ex												3
	2.1	Module	es				 	 	 	 	 		 		3
3	Nam	espace	Index												5
	3.1	Packag	ges				 	 	 	 	 		 		5
4	Hiera	archical	Index												7
	4.1	Class I	Hierarchy				 	 	 	 	 		 		7
5	Clas	s Index													9
	5.1	Class I	₋ist				 	 	 	 	 		 		9
6	File	Index													11
	6.1	File Lis	st				 	 	 	 	 		 		11
7	Mod	ule Doc	umentatio	on											13
	7.1	Constr	aints				 	 	 	 	 		 		13
		7.1.1	Detailed I	Descripti	ion .		 		13						
	7.2	Gnowe	e				 		14						
		7.2.1	Detailed I	Descripti	ion .		 		14						
		7.2.2	Function	Docume	ntation	ı	 		14						
			7.2.2.1	main .			 	 	 	 	 		 		14
	7.3	Gnowe	eHeuristic	s			 	 	 	 	 		 		16
		7.3.1	Detailed I	Descripti	ion .		 	 	 	 	 		 		16
		7.3.2	Function	Docume	ntation	ı	 	 	 	 	 		 		16
			7.3.2.1	rejectio	n_bou	nds .	 	 	 	 	 		 		16
			7.3.2.2	simple_	_bounc	ds	 		17						
	7.4	Gnowe	eUtilities				 	 	 	 	 		 		18
		7.4.1	Detailed I												18
	7.5	Objecti	veFunction												19
		7.5.1	Detailed I												19
		752													19

iv CONTENTS

			7.5.2.1 prod	 . 19
	7.6	OptiPlo	ot	 . 20
		7.6.1	Detailed Description	 . 20
		7.6.2	Function Documentation	 . 20
			7.6.2.1 plot_feval_hist	 . 20
			7.6.2.2 plot_hist	 . 21
			7.6.2.3 plot_hist_comp	 . 21
			7.6.2.4 plot_optimization	 . 21
			7.6.2.5 plot_tlf	 . 22
			7.6.2.6 plot_vars	 . 22
	7.7	Sampl	ing	 . 23
		7.7.1	Detailed Description	 . 23
		7.7.2	Function Documentation	 . 23
			7.7.2.1 get_cdr_permutations	 . 23
			7.7.2.2 initial_samples	 . 24
			7.7.2.3 levy	 . 24
			7.7.2.4 NOLH	 . 25
			7.7.2.5 params	 . 25
			7.7.2.6 plot_samples	 . 25
			7.7.2.7 tlf	 . 25
8	Nam	nespace	• Documentation	27
8	Nam 8.1	_	Documentation ee Namespace Reference	
8		_		. 27
	8.1	Gnowe 8.1.1	Detailed Description	. 27 . 27
8	8.1	Gnower 8.1.1	ee Namespace Reference	 . 27 . 27
	8.1	Gnowe 8.1.1 ss Docu Constr	pee Namespace Reference	 . 27 . 27 . 29
	8.1	Gnowe 8.1.1 ss Docu Constr 9.1.1	pee Namespace Reference	 . 27 . 27 . 29 . 29
	8.1	Gnowe 8.1.1 ss Docu Constr	pee Namespace Reference Detailed Description mentation raints.Constraint Class Reference Detailed Description Constructor & Destructor Documentation	 . 27 . 27 . 29 . 29 . 30 . 30
	8.1	Gnowe 8.1.1 ss Docu Constr 9.1.1	pee Namespace Reference	 . 27 . 27 . 29 . 30 . 30
	8.1	Gnowe 8.1.1 SS Docu Constr 9.1.1 9.1.2	Detailed Description mentation aints.Constraint Class Reference Detailed Description Constructor & Destructor Documentation 9.1.2.1init Member Function Documentation	. 27 . 29 . 29 . 30 . 30 . 30
	8.1	Gnowe 8.1.1 SS Docu Constr 9.1.1 9.1.2	Detailed Description mentation raints.Constraint Class Reference Detailed Description Constructor & Destructor Documentation 9.1.2.1init Member Function Documentation 9.1.3.1repr	. 27 . 29 . 30 . 30 . 30 . 30 . 30
	8.1	Gnowe 8.1.1 SS Docu Constr 9.1.1 9.1.2	Detailed Description mentation raints.Constraint Class Reference Detailed Description Constructor & Destructor Documentation 9.1.2.1init Member Function Documentation 9.1.3.1repr 9.1.3.2str	. 27 . 29 . 29 . 30 . 30 . 30 . 30 . 30 . 30
	8.1	Gnowe 8.1.1 SS Docu Constr 9.1.1 9.1.2	Detailed Description mentation raints.Constraint Class Reference Detailed Description Constructor & Destructor Documentation 9.1.2.1init Member Function Documentation 9.1.3.1repr 9.1.3.2str	. 27 . 29 . 29 . 30 . 30 . 30 . 30 . 30 . 30 . 30
	8.1	Gnowe 8.1.1 SS Docu Constr 9.1.1 9.1.2	Detailed Description mentation raints.Constraint Class Reference Detailed Description Constructor & Destructor Documentation 9.1.2.1init Member Function Documentation 9.1.3.1repr 9.1.3.2str 9.1.3.3 get_penalty	. 27 . 29 . 30 . 30 . 30 . 30 . 30 . 30 . 31
	8.1	Gnowe 8.1.1 SS Docu Constr 9.1.1 9.1.2	Detailed Description mentation aints.Constraint Class Reference Detailed Description Constructor & Destructor Documentation 9.1.2.1init Member Function Documentation 9.1.3.1repr 9.1.3.2str 9.1.3.3 get_penalty 9.1.3.4 greater_than	. 27 . 29 . 29 . 30 . 30 . 30 . 30 . 30 . 31 . 31
	8.1	Gnowe 8.1.1 SS Docu Constr 9.1.1 9.1.2	Detailed Description mentation raints.Constraint Class Reference Detailed Description Constructor & Destructor Documentation 9.1.2.1init Member Function Documentation 9.1.3.1repr 9.1.3.2str 9.1.3.3 get_penalty 9.1.3.4 greater_than 9.1.3.5 less_or_equal	. 27 . 29 . 29 . 30 . 30 . 30 . 30 . 31 . 31
	8.1	Gnowe 8.1.1 SS Docu Constr 9.1.1 9.1.2	Detailed Description mentation maints.Constraint Class Reference Detailed Description Constructor & Destructor Documentation 9.1.2.1init Member Function Documentation 9.1.3.1repr 9.1.3.2str 9.1.3.2 get_penalty 9.1.3.4 greater_than 9.1.3.5 less_or_equal 9.1.3.6 less_than	. 27 . 29 . 29 . 30 . 30 . 30 . 30 . 31 . 31 . 31
	8.1	Gnowe 8.1.1 SS Docu Constr 9.1.1 9.1.2	Detailed Description mentation aints.Constraint Class Reference Detailed Description Constructor & Destructor Documentation 9.1.2.1init Member Function Documentation 9.1.3.1repr 9.1.3.2str 9.1.3.3 get_penalty 9.1.3.4 greater_than 9.1.3.5 less_or_equal 9.1.3.6 less_than 9.1.3.7 set_constraint_func	. 27 . 29 . 29 . 30 . 30 . 30 . 30 . 31 . 31 . 31 . 32 . 32
	8.1	Gnowe 8.1.1 SS Docu Constr 9.1.1 9.1.2	Detailed Description mentation raints.Constraint Class Reference Detailed Description Constructor & Destructor Documentation 9.1.2.1init Member Function Documentation 9.1.3.1repr 9.1.3.2str 9.1.3.3 get_penalty 9.1.3.4 greater_than 9.1.3.5 less_or_equal 9.1.3.6 less_than 9.1.3.7 set_constraint_func Member Data Documentation	. 27 . 29 . 29 . 30 . 30 . 30 . 30 . 31 . 31 . 31 . 32 . 32

CONTENTS

9.2	Gnowe	eeUtilities.Event Class Reference
	9.2.1	Detailed Description
	9.2.2	Constructor & Destructor Documentation
		9.2.2.1init
	9.2.3	Member Function Documentation
		9.2.3.1repr
		9.2.3.2str
	9.2.4	Member Data Documentation
		9.2.4.1 design
		9.2.4.2 evaluations
		9.2.4.3 fitness
		9.2.4.4 generation
9.3	Gnowe	eeHeuristics.GnoweeHeuristics Class Reference
	9.3.1	Detailed Description
	9.3.2	Constructor & Destructor Documentation
		9.3.2.1init
	9.3.3	Member Function Documentation
		9.3.3.1repr
		9.3.3.2str
		9.3.3.3 cont_levy_flight
		9.3.3.4 crossover
		9.3.3.5 disc_levy_flight
		9.3.3.6 elite_crossover
		9.3.3.7 initialize
		9.3.3.8 mutate
		9.3.3.9 scatter_search
	9.3.4	Member Data Documentation
		9.3.4.1 alpha
		9.3.4.2 convTol
		9.3.4.3 fracDiscovered
		9.3.4.4 fracElite
		9.3.4.5 fracLevy
		9.3.4.6 gamma
		9.3.4.7 initSampling
		9.3.4.8 maxFevals
		9.3.4.9 maxGens
		9.3.4.10 n
		9.3.4.11 optConvTol
		9.3.4.12 optimalFitness
		9.3.4.13 penalty

vi CONTENTS

		9.3.4.14 population
		9.3.4.15 scalingFactor
		9.3.4.16 stallLimit
9.4	Objecti	iveFunction.ObjectiveFunction Class Reference
	9.4.1	Detailed Description
	9.4.2	Constructor & Destructor Documentation
		9.4.2.1init
	9.4.3	Member Function Documentation
		9.4.3.1repr
		9.4.3.2str
		9.4.3.3 ackley
		9.4.3.4 dejong
		9.4.3.5 easom
		9.4.3.6 griewank
		9.4.3.7 mi_chemical_process
		9.4.3.8 mi_pressure_vessel
		9.4.3.9 mi_spring
		9.4.3.10 pressure_vessel
		9.4.3.11 rastrigin
		9.4.3.12 rosenbrock
		9.4.3.13 set_obj_func
		9.4.3.14 shifted_ackley
		9.4.3.15 shifted_dejong
		9.4.3.16 shifted_easom
		9.4.3.17 shifted_griewank
		9.4.3.18 shifted_rastrigin
		9.4.3.19 shifted_rosenbrock
		9.4.3.20 speed_reducer
		9.4.3.21 spring
		9.4.3.22 tsp
		9.4.3.23 welded_beam
	9.4.4	Member Data Documentation
		9.4.4.1 func
		9.4.4.2 objective
9.5	Gnowe	eeUtilities.Parent Class Reference
	9.5.1	Detailed Description
	9.5.2	Constructor & Destructor Documentation
		9.5.2.1init
	9.5.3	Member Function Documentation
		9.5.3.1 <u>repr</u>

CONTENTS vii

		9.5.3.2str	59
	9.5.4	Member Data Documentation	59
		9.5.4.1 changeCount	59
		9.5.4.2 fitness	59
		9.5.4.3 stallCount	59
		9.5.4.4 variables	59
9.6	Gnowe	eUtilities.ProblemParameters Class Reference	60
	9.6.1	Detailed Description	60
	9.6.2	Constructor & Destructor Documentation	61
		9.6.2.1init	61
	9.6.3	Member Function Documentation	62
		9.6.3.1repr	62
		9.6.3.2str	63
		9.6.3.3 get_preset_params	63
	9.6.4	Member Data Documentation	63
		9.6.4.1 discreteVals	63
		9.6.4.2 histTitle	63
		9.6.4.3 lb	63
		9.6.4.4 optimum	63
		9.6.4.5 pltTitle	64
		9.6.4.6 ub	64
		9.6.4.7 varNames	64
		9.6.4.8 varType	64
9.7	Gnowe	eUtilities.Switch Class Reference	64
	9.7.1	Detailed Description	64
	9.7.2	Constructor & Destructor Documentation	65
		9.7.2.1init	65
	9.7.3	Member Function Documentation	65
		9.7.3.1iter	65
		9.7.3.2 match	65
	9.7.4	Member Data Documentation	65
		9.7.4.1 fall	65
		9.7.4.2 value	65
9.8	Gnowe	eUtilities.WeightedRandomGenerator Class Reference	66
	9.8.1	Detailed Description	66
10 File	Docum	entation	67
10.1	/home/	pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/Constraints.py File Reference	67
10.2	/home/	pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/Gnowee.py File Reference	67

viii CONTENTS

10.7	/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/Sampling.py File Reference .	69
10.6	/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/OptiPlot.py File Reference	69
10.5	/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/ObjectiveFunction.py File Reference	68
10.4	/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/GnoweeUtilities.py File Reference	68
10.3	/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/GnoweeHeuristics.py File Reference	67

Main Page

Gnowee

Version

1.0

Gnowee is a general nearly-global metaheuristic optimization algorithm. It uses a blend of common heuristics to solve difficult gradient free constrained MINLP problems with mixed variables. It is capable of solving simpler problems, but may not be the algorithm of choice.

Running Gnowee

For examples on how to run Gnowee, please refer to the runGnowee notebook included in the src directory.

Building Documentation

To build the documentation, in the docs/src directory run the command:

>> doxygen Doxyfile

This will build the html and latex version of the documentation. The symlink in the docs directory for the html index should automatically update. T

The up-to-date latex documentation is included in a pdf form in the repo under the docs directory. If an update of the latex documentation is desired, go to the docs/latex directory and run the command:

>> make

This will build the latex documentation. The file will be named refman.pdf and be placed in this directory.

Citation Information

To cite Gnowee, use the following reference:

Contact information

Bugs and suggestions for improvement can be submitted via the GitHub page: https://github.com/-SlaybaughLab/Gnowee

Alternatively, questions or comments on Gnowee can be directed to:

James Bevins

2 Main Page

james.e.bevins@gmail.com

Licensing Information

Acknowledements

AF, advisor, NSF

Module Index

2.1 Modules

Hara		lint a	of all	modules
пеге	is a	IISt (n all	modules

Constraints												 					 					13
Gnowee												 					 					14
GnoweeHeuristics												 					 					16
GnoweeUtilities .												 					 					18
ObjectiveFunction												 					 					19
OptiPlot												 					 					20
Sampling												 					 					23

Module Index

Namespace Index

3.1	Packag	nes
J. I	i achai	463

Here are the packages with brief descriptions (if available):	
Gnowee	
Contains the Growee ontimization program and associated utilities	27

6 Namespace Index

Hierarchical Index

4.1 Class Hierarchy

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

object	
Constraints.Constraint	29
GnoweeHeuristics.GnoweeHeuristics	. 34
GnoweeUtilities.Event	. 32
GnoweeUtilities.Parent	. 58
GnoweeUtilities.ProblemParameters	60
GnoweeUtilities.Switch	64
GnoweeUtilities.WeightedRandomGenerator	. 66

8 **Hierarchical Index**

Class Index

5.1 Class List

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

Constraints.Constraint	
The class creates a Constraints object that can be used in optimization algorithms	29
GnoweeUtilities.Event	
Represents a snapshot in the optimization process to be used for debugging, benchmarking, and	
user feedback	32
GnoweeHeuristics.GnoweeHeuristics	
The class is the foundation of the Gnowee optimization algorithm	34
ObjectiveFunction. ObjectiveFunction	
The class creates a ObjectiveFunction object that can be used in optimization algorithms	43
GnoweeUtilities.Parent	
The class contains all of the parameters pertinent to a member of the population	58
GnoweeUtilities.ProblemParameters	
Creates an object containing key features of the chosen optimization problem	60
GnoweeUtilities.Switch	
Creates a switch class object to switch between cases	64
GnoweeUtilities.WeightedRandomGenerator	66

10 Class Index

File Index

6.1 File List

Here is a list of all documented files with brief descriptions:

/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/Constraints.py	67
/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/Gnowee.py	67
/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/GnoweeHeuristics.py	67
/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/GnoweeUtilities.py	68
/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/ObjectiveFunction.py	68
/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/OptiPlot.py	69
/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/Sampling.py	69

12 File Index

Module Documentation

7.1 Constraints

Defines a class to perform constraint calculations.

Classes

• class Constraints.Constraint

The class creates a Constraints object that can be used in optimization algorithms.

7.1.1 Detailed Description

Defines a class to perform constraint calculations.

Author

James Bevins,

Date

9May17

7.	2	Gr	າດເ	NE	סנ
		чı	IV	A A C	,

Main program for the **Gnowee** metaheuristic algorithm.

Functions

· def Gnowee.main

Main controller program for the Gnowee optimization.

7.2.1 Detailed Description

Main program for the Gnowee metaheuristic algorithm. General nearly-global metaheuristic optimization algorithm. Uses a blend of common heuristics to solve difficult gradient free constrained MINLP problems with categorical variables. It is capable of solving simpler problems, but may not be the algorithm of choice.

For examples on how to run Gnowee, please refer to the runGnowee notebook included in the src directory.

Author

James Bevins

Date

9May17

7.2.2 Function Documentation

7.2.2.1 def Gnowee.main (func, lb, ub, varType, gh, discreteVals = [])

Main controller program for the **Gnowee** optimization.

7.2 Gnowee 15

Parameters

func	function The objective function to be minimized.
lb	list or array The lower bounds of the design variable(s). Only enter the bounds for continuous and integer/binary variables. The order must match the order specified in varType and ub.
ub	list or array The upper bounds of the design variable(s). Only enter the bounds for continuous and integer/binary variables. The order must match the order specified in varType and lb.
varType	list or array The type of variable for each position in the upper and lower bounds array. Discrete variables are to be included last as they are specified separatly from the lb/ub throught the discreteVals optional input. A variable can have two types (for example, 'dx' could denote a layer that can take multiple materials and be placed at multiple design locations) Allowed values: 'c' = continuous over a given range (range specified in lb & ub). 'i' = integer/binary (difference denoted by ub/lb). 'd' = discrete where the allowed values are given by the option discreteVals nxm arrary with n=# of discrete variables and m=# of values that can be taken for each variable. 'x' = combinatorial. All of the variables denoted by x are assumed to be "swappable" in combinatorial permutations. There must be at least two variables denoted as combinatorial. 'f' = fixed design variable. Will not be considered of any permutation.
gh	GnoweeHeuristic object An object constaining the settings and methods required for the Gnowee optimization algorithm.
discreteVals	list of list(s) nxm with n=# of discrete variables and m=# of values that can be taken for each variable. For example, if you had two variables representing the tickness and diameter of a cylinder that take standard values, the discreteVals could be specified as: discreteVals = [[0.125, 0.25, 0.375], [0.25, 0.5, 075]] Gnowee will then map the optimization results to these allowed values.

Returns

list: List for design event objects for the current top solution vs generation. Only stores the information when new optimal designs are found.

7.3 GnoweeHeuristics

Heuristics and settings supporting the Gnowee metaheuristic optimization algorithm.

Classes

· class GnoweeHeuristics.GnoweeHeuristics

The class is the foundation of the Gnowee optimization algorithm.

Functions

• def GnoweeHeuristics.simple_bounds

Application of problem boundaries to generated solutions.

• def GnoweeHeuristics.rejection_bounds

Application of problem boundaries to generated solutions.

7.3.1 Detailed Description

Heuristics and settings supporting the Gnowee metaheuristic optimization algorithm. This instantiates the class and methods necessary to perform an optimization using the Gnowee algorithm. Each of the heuristics can also be used independently of the Gnowee algorithm by instantiating this class and choosing the desired heuristic.

The default settings are those found to be best for a suite of benchmark problems but one may find alternative settings are useful for the problem of interest based on the fitness landscape and type of variables.

Author

James Bevins

Date

8May17

7.3.2 Function Documentation

7.3.2.1 def GnoweeHeuristics.rejection_bounds (parent, child, stepSize, lb, ub)

Application of problem boundaries to generated solutions.

Adjusts step size for all rejected solutions until within the boundaries.

parent	array
	The current system designs.
child	array
	The proposed new system designs.
stepSize	float
	The stepsize for the permutation.

7.3 GnoweeHeuristics 17

lb	array
	The lower bounds of the design variable(s).
ub	array
	The upper bounds of the design variable(s).

Returns

array: The new system design that is within problem boundaries.

7.3.2.2 def GnoweeHeuristics.simple_bounds ($\it child, lb, ub$)

Application of problem boundaries to generated solutions.

If outside of the boundaries, the variable defaults to the boundary.

Parameters

child	array
	The proposed new system designs.
lb	array
	The lower bounds of the design variable(s).
ub	array
	The upper bounds of the design variable(s).

Returns

array: The new system design that is within problem boundaries.

7.4 GnoweeUtilities

Classes and methods to support the **Gnowee** optimization algorithm.

Classes

• class GnoweeUtilities.Parent

The class contains all of the parameters pertinent to a member of the population.

• class GnoweeUtilities.Event

Represents a snapshot in the optimization process to be used for debugging, benchmarking, and user feedback.

• class GnoweeUtilities.ProblemParameters

Creates an object containing key features of the chosen optimization problem.

· class GnoweeUtilities.Switch

Creates a switch class object to switch between cases.

7.4.1 Detailed Description

Classes and methods to support the **Gnowee** optimization algorithm.

Author

James Bevins

Date

9May17

7.5 ObjectiveFunction 19

7.5 ObjectiveFunction

Defines a class to perform objective function calculations.

Classes

· class ObjectiveFunction.ObjectiveFunction

The class creates a ObjectiveFunction object that can be used in optimization algorithms.

Functions

· def ObjectiveFunction.prod

Computes the product of a set of numbers (ie big PI, mulitplicative equivalent to sum).

7.5.1 Detailed Description

Defines a class to perform objective function calculations. This class contains the necessary functions and methods to create objective functions and initialize the necessary parameters. The class is pre-stocked with common benchmark functions for easy fishing.

Users can modify the this class to add additional functions following the format of the functions currently in the class.

Author

James Bevins

Date

10May17

7.5.2 Function Documentation

7.5.2.1 def ObjectiveFunction.prod (iterable)

Computes the product of a set of numbers (ie big PI, mulitplicative equivalent to sum).

Parameters

iterable list or array or generator Iterable set to multiply.

Returns

float: The product of all of the items in iterable

7.6 OptiPlot

Plotting functions developed to help visualize and quantify the metaheuristic optimization process.

Functions

· def OptiPlot.plot_vars

Plot the variables as they change in the optimization process.

def OptiPlot.plot_hist

Plots the histogram of function evaluation results from multiple runs of an optimization algorithm.

def OptiPlot.plot_hist_comp

Histograms and plots the comparison of two sets of function evaluation data.

· def OptiPlot.plot_feval_hist

Plots the fitness vs function evaluation results of an optimization algorithm run.

def OptiPlot.plot_tlf

Plots a comparison of the TLF to the Levy distribution.

• def OptiPlot.plot_optimization

Plots the results of optimization process for a given algorithm and parameter.

7.6.1 Detailed Description

Plotting functions developed to help visualize and quantify the metaheuristic optimization process.

Author

James Bevins

Date

10May17

7.6.2 Function Documentation

```
7.6.2.1 def OptiPlot.plot_feval_hist ( data = [], listData = [], label = [] )
```

Plots the fitness vs function evaluation results of an optimization algorithm run.

Can plot a single run or multiple to compare results. To plot multiple data sets, use the listData argument; otherwise, use the data argument.

data	list or array			
	Contains the function eval history. Columns are: [function evals, fitness, number of datapoints].			
listData	listData list of lists or arrays Contains a list of function eval histories. Columns are: [function evals, fitness, number of datapoints].			

7.6 OptiPlot 21

label	list
	List of names corresponding to the data sets provided.

7.6.2.2 def OptiPlot.plot_hist (data, title = ", xLabel = ")

Plots the histogram of function evaluation results from multiple runs of an optimization algorithm.

Can be used to understand the convergence of the algorithm.

Parameters

data	list
	Contains the number of function evals for each optimization run.
title	string
	Title for plot.
xLabel	string
	Label for independent variable.

7.6.2.3 def OptiPlot.plot_hist_comp (data, data2, dataLabels, title = ", xLabel = ")

Histograms and plots the comparison of two sets of function evaluation data.

Parameters

data	list		
	Contains the number of function evals for each optimization run.		
data2 list			
Contains the number of function evals for each optimization run for a second set of runs.			
dataLabels	list		
Contains the legend label names for each data set.			
title	string		
	Title for plot.		
xLabel	string		
	Label for independent variable.		

7.6.2.4 def OptiPlot.plot_optimization (data, label, title = ")

Plots the results of optimization process for a given algorithm and parameter.

data	array					
	Contains the function eval history.	Columns are:	[function evals,	fitness,	number	of
	datapoints]					

label	list
	List of names of the problem types ran.
title	string
	Title for plot.

```
7.6.2.5 def OptiPlot.plot_tlf ( alpha = 1.5, gamma = 1., numSamp = 1E7, cutPoint = 10. )
```

Plots a comparison of the TLF to the Levy distribution.

Parameters

alpha	float			
	Levy exponent - defines the index of the distribution and controls scale properties of the			
	stochastic process.			
gamma	float			
	Gamma - Scale unit of process for Levy flights.			
numSamp	integer			
·	Number of Levy flights to sample.			
cutPoint	float			
	Point at which to cut sampled Levy values and resample.			

```
7.6.2.6 def OptiPlot.plot_vars ( data, lowBounds = [], upBounds = [], title = [], label = [] )
```

Plot the variables as they change in the optimization process.

Currently only functions in post-processing, not real time.

data	list of event objects	
	Contain the optimization history in event objects within the data list.	
IowBounds	array	
	The lower bounds of the design variable(s).	
upBounds	array	
	The upper bounds of the design variable(s).	
title	string	
litio	Title for plot.	
	Title for piot.	
label	list	
	List of names of design variables.	

7.7 Sampling 23

7.7 Sampling

Different methods to perform phase space sampling and random walks.

Functions

· def Sampling.initial_samples

Generate a set of samples in a given phase space.

• def Sampling.plot_samples

Plot the first 2 and 3 dimensions on the sample distribution.

· def Sampling.levy

Sample the Levy distribution given by.

· def Sampling.tlf

Samples from a truncated Levy flight distribution (TLF) according to Manegna, "Stochastic Process with Ultraslow Convergence to a Gaussian: The Truncated Levy Flight" to map a levy distribution onto the interval [0,1].

def Sampling.NOLH

This library allows to generate Nearly Orthogonal Latin Hypercubes (NOLH) according to Cioppa (2007) and De Rainville et al.

· def Sampling.params

Returns the NOLH order \$m\$, the required configuration length \$q\$ and the number of columns to remove to obtain the desired dimensionality.

def Sampling.get_cdr_permutations

Generate a set of CDR permulations for NOLH.

7.7.1 Detailed Description

Different methods to perform phase space sampling and random walks. Design of experiment and phase space sampling methods. Includes some vizualization tools.

Dependencies on pyDOE.

Author

James Bevins

Date

8May17

7.7.2 Function Documentation

7.7.2.1 def Sampling.get_cdr_permutations (dim)

Generate a set of CDR permulations for NOLH.

Parameters

dim	integer
	The dimension of the space.

Returns

array: A configuration vector.

array: Array containing the indexes of the colummns to be removed from conf vector.

7.7.2.2 def Sampling.initial_samples (lb, ub, method, numSamp)

Generate a set of samples in a given phase space.

The current methods available are 'random', 'nolh', 'nolh-rp', 'nolh-cdr', or 'lhc'.

Parameters

lb	array			
	The lower bounds of the design variable(s).			
ub	array			
	The upper bounds of the design variable(s).			
method	string			
	String representing the chosen sampling method. Valid options are: 'random', 'nolh', 'nolh-rp', 'nolh-cdr', or 'lhc'.			
numSamp	integer			
	The number of samples to be generated. Ignored for nolh algorithms.			

Returns

array: The list of coordinates for the sampled phase space.

7.7.2.3 def Sampling.levy (nc, nr = 0, alpha = 1.5, gam = 1, n = 1)

Sample the Levy distribution given by.

$$L_{\alpha,\gamma}(z) = \frac{1}{\pi} \int\limits_{0}^{+\infty} e^{-\gamma q^{\alpha}} \cos(qz) dq$$

using the Mantegna algoritm outlined in "Fast, Accurate Algorithm for Numerical Simulation of Levy Stable Stochastic Processes."

Parameters

nc	integer The number of columns of Levy values for the return array.		
nr	integer The number of rows of Levy values for the return array. The number of rows of Levy values for the return array.		
alpha	float Levy exponent - defines the index of the distribution and controls scale properties of the stochastic process.		
gam	float Gamma - Scale unit of process for Levy flights.		
n	integer Number of independent variables - can be used to reduce Levy flight sampling variance.		

Returns

array: Array representing the levy flights for each nest.

7.7 Sampling 25

7.7.2.4 def Sampling.NOLH (conf, remove = None)

This library allows to generate Nearly Orthogonal Latin Hypercubes (NOLH) according to Cioppa (2007) and De Rainville et al.

(2012) and reference therein.

```
https://pypi.python.org/pypi/pynolh
```

Constructs a Nearly Orthogonal Latin Hypercube (NOLH) of order m from a configuration vector conf. The configuration vector may contain either the numbers in [0 q-1] vector may contain either the numbers

$$d = m + \{m-1\}\{2\}$$

is the NOLH dimensionality.

The whole library is incorporated here with minimal modification for commonality and consolidation of methods.

Parameters

conf	array Configuration vector.
remove	array Array containing the indexes of the colummns to be removed from conf vector.

Returns

array: Array containing nearly orthogonal latin hypercube sampling.

7.7.2.5 def Sampling.params (dim)

Returns the NOLH order \$m\$, the required configuration length \$q\$ and the number of columns to remove to obtain the desired dimensionality.

Parameters

dim	integer
	The dimension of the space.

7.7.2.6 def Sampling.plot_samples (s)

Plot the first 2 and 3 dimensions on the sample distribution.

Can't plot the full hyperspace yet. Produces a very simple plot for visualizing the difference in the sampling methods.

Parameters

S	array
	The list of coordinates for the sampled phase space.

```
7.7.2.7 def Sampling.tlf ( numRow = 1, numCol = 1, alpha = 1.5, gam = 1., cutPoint = 10.)
```

Samples from a truncated Levy flight distribution (TLF) according to Manegna, "Stochastic Process with Ultraslow Convergence to a Gaussian: The Truncated Levy Flight" to map a levy distribution onto the interval [0,1].

Parameters

numRow	integer Number of rows of Levy flights to sample.			
	Number of rows of Levy highes to sample.			
numCol	integer			
	Number of columns of Levy flights to sample.			
alpha	float			
	Levy exponent - defines the index of the distribution and controls scale properties of the			
	stochastic process.			
gam	float			
	Gamma - Scale unit of process for Levy flights.			
cutPoint	float			
	Point at which to cut sampled Levy values and resample.			

Returns

array: Array representing the levy flights on the interval (0,1).

Namespace Documentation

8.1 Gnowee Namespace Reference

Contains the **Gnowee** optimization program and associated utilities.

Functions

def main

Main controller program for the Gnowee optimization.

8.1.1 Detailed Description

Contains the Gnowee optimization program and associated utilities.

Version

1.0

General nearly-global metaheuristic optimization algorithm. Uses a blend of common heuristics to solve difficult gradient free constrained MINLP problems with categorical variables. It is capable of solving simpler problems, but may not be the algorithm of choice.

For examples on how to run Gnowee, please refer to the runGnowee notebook included in the src directory.

Author

James Bevins

Date

9May17

See Also

Gnowee
GnoweeHeuristics
GnoweeUtilities
ObjectiveFunction
Constraints
OptiPlot
Sampling

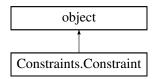
Namespace	Docume	entation

Chapter 9

Class Documentation

9.1 Constraints.Constraint Class Reference

The class creates a Constraints object that can be used in optimization algorithms. Inheritance diagram for Constraints.Constraint:



Public Member Functions

def __init__

Constructor to build the ObjectiveFunction class.

def __repr__

Constraint class param print function.

def __str__

Human readable Constraint print function.

· def set constraint func

Converts an input string name for a function to a function handle.

def get_penalty

Calculate the constraint violation penalty, if any.

· def less_or_equal

Compares a previously calculated value to a user specifed maximum including that maximum.

• def less_than

Compares a previously calculated value to a user specifed maximum excluding that maximum.

def greater_than

Compares the calculated value to the minimum specified by the user.

Public Attributes

• func

function handle: The function handle for the constraint function to be used for the optimization.

· constraint

float: The constraint to be enforced.

penalty

float: The penalty to be applied if the constraint is violated

9.1.1 Detailed Description

The class creates a Constraints object that can be used in optimization algorithms.

9.1.2 Constructor & Destructor Documentation

9.1.2.1 def Constraints.Constraint__init__(self, method = None, constraint = None, penalty = 1E15)

Constructor to build the ObjectiveFunction class.

Parameters

self	object pointer
	The object pointer.
method	string
	The name of the constraint function to evaluate.
constraint	float
	The constraint to be compared against.
penalty	float
	The penalty to be applied if a constraint is violated. 1E15 is recommended.

9.1.3 Member Function Documentation

9.1.3.1 def Constraints.Constraint.__repr__ (self)

Constraint class param print function.

Parameters

self	pointer
	The Constraint pointer.

9.1.3.2 def Constraints.Constraint.__str__ (self)

Human readable Constraint print function.

Parameters

self	pointer
	The Constraint pointer.

9.1.3.3 def Constraints.Constraint.get_penalty (self, violation)

Calculate the constraint violation penalty, if any.

Parameters

self	pointer
	The Constraint pointer.
violation	float
	The magnitude of the constraint violation used for scaling the penalty.

Returns

float: The scaled penalty.

9.1.3.4 def Constraints.Constraint.greater_than (self, candidate)

Compares the calculated value to the minimum specified by the user.

Parameters

self	pointer
	The Constraint pointer.
candidate	float
	The calculated value corresponding to a candidate design.

Returns

float: The penalty associated with the candidate design.

9.1.3.5 def Constraints.Constraint.less_or_equal (self, candidate)

Compares a previously calculated value to a user specifed maximum including that maximum.

Parameters

self	pointer
	The Constraint pointer.
candidate	float
	The calculated value corresponding to a candidate design.
	· - · · · · · · · · · · · · · · · · · ·

Returns

float: The penalty associated with the candidate design.

9.1.3.6 def Constraints.Constraint.less_than (self, candidate)

Compares a previously calculated value to a user specifed maximum excluding that maximum.

Parameters

Generated on Wed May 10 2017 20:42:18 for Gnowee by Doxygen

self	pointer
	The Constraint pointer.
candidate	float
	The calculated value corresponding to a candidate design.

Returns

float: The penalty associated with the candidate design.

9.1.3.7 def Constraints.Constraint.set_constraint_func (self, funcName)

Converts an input string name for a function to a function handle.

Parameters

self	pointer The Constraint pointer.
funcName	string A string identifying the constraint function to be used.

9.1.4 Member Data Documentation

9.1.4.1 Constraints.Constraint.constraint

float: The constraint to be enforced.

9.1.4.2 Constraints.Constraint.func

function handle: The function handle for the constraint function to be used for the optimization.

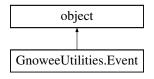
The function must be specified as a method of the class.

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

• /home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/Constraints.py

9.2 GnoweeUtilities.Event Class Reference

Represents a snapshot in the optimization process to be used for debugging, benchmarking, and user feedback. Inheritance diagram for GnoweeUtilities.Event:



Public Member Functions

def __init__

Constructor to build the Event class.

def repr

Event print function.

def __str__

Human readable Event print function.

Public Attributes

· generation

integer: The generation the design was arrived at.

· evaluations

integer: The number of fitness evaluations done to obtain this design.

fitness

float: The assessed fitness for the current set of variables.

design

array: The set of variables representing a design solution.

9.2.1 Detailed Description

Represents a snapshot in the optimization process to be used for debugging, benchmarking, and user feedback.

9.2.2 Constructor & Destructor Documentation

9.2.2.1 def GnoweeUtilities.Event.__init__ (self, generation, evaluations, fitness, design)

Constructor to build the Event class.

Parameters

self	Event pointer
	The Event pointer.
generation	integer
	The generation the design was arrived at.
evaluations	integer
	The number of fitness evaluations done to obtain this design.
fitness	float
	The assessed fitness for the current set of variables.
design	array
	The set of variables representing a design solution.

9.2.3 Member Function Documentation

9.2.3.1 def GnoweeUtilities.Event.__repr__ (self)

Event print function.

Parameters

self	Event pointer
	The Event pointer.

9.2.3.2 def GnoweeUtilities.Event.__str__(self)

Human readable Event print function.

Parameters

self	Event pointer
	The Event pointer.

9.2.4 Member Data Documentation

9.2.4.1 GnoweeUtilities.Event.design

array: The set of variables representing a design solution.

9.2.4.2 GnoweeUtilities.Event.evaluations

integer: The number of fitness evaluations done to obtain this design.

9.2.4.3 GnoweeUtilities.Event.fitness

float: The assessed fitness for the current set of variables.

9.2.4.4 GnoweeUtilities.Event.generation

integer: The generation the design was arrived at.

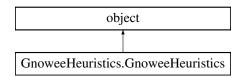
The documentation for this class was generated from the following file:

• /home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/GnoweeUtilities.py

9.3 GnoweeHeuristics.GnoweeHeuristics Class Reference

The class is the foundation of the Gnowee optimization algorithm.

Inheritance diagram for GnoweeHeuristics:



Public Member Functions

def init

Constructor to build the GnoweeHeuristics class.

def __repr__

GnoweeHeuristics print function.

def __str__

Human readable GnoweeHeuristics print function.

· def initialize

Initialize the population according to the sampling method chosen.

· def disc_levy_flight

Generate new children using truncated Levy flights permutation of current generation design parameters according to:

· def cont_levy_flight

Generate new children using Levy flights permutation of current generation design parameters according to:

· def scatter search

Generate new designs using the scatter search heuristic according to:

· def elite crossover

Generate new designs by using inver-over on combinatorial variables.

· def crossover

Generate new children using distance based crossover strategies on the top parent.

· def mutate

Generate new children by adding a weighted difference between two population vectors to a third vector.

Public Attributes

· population

integer The number of members in each generation.

initSampling

string The method used to sample the phase space and create the initial population.

fracDiscovered

float Discovery probability used for the mutate() heuristic.

fracElite

float Elite fraction probability used for the scatter_search(), crossover(), and cont_crossover() heuristics.

fracLevy

 ${\it float Levy flight probability used for the } {\it disc_levy_flight()} \ {\it and cont_levy_flight()} \ {\it heuristics}.$

alpha

float Levy exponent - defines the index of the distribution and controls scale properties of the stochastic process.

• gamma

float Gamma - scale unit of process for Levy flights.

• n

integer Number of independent variables - can be used to reduce Levy flight sampling variance.

scalingFactor

float Step size scaling factor used to adjust Levy flights to length scale of system.

penalty

float Individual constraint violation penalty to add to objective function.

maxGens

integer The maximum number of generations to search.

· maxFevals

integer The maximum number of objective function evaluations.

convTol

float The minimum change of the best objective value before the search terminates.

· stallLimit

integer The maximum number of gen3rations to search without a descrease exceeding convTol.

optimalFitness

float The best know fitness value for the problem considered used to test for convergence.

optConvTol

float The maximum deviation from the best know fitness value before the search terminates.

9.3.1 Detailed Description

The class is the foundation of the **Gnowee** optimization algorithm.

It sets the settings required for the algorithm and defines the heurstics.

9.3.2 Constructor & Destructor Documentation

```
9.3.2.1 def GnoweeHeuristics.GnoweeHeuristics.__init__ ( self, population = 25, initSampling = 'lhc', fracDiscovered = 0.2, fracElite = 0.2, fracLevy = 0.2, alpha = 1.5, gamma = 1, n = 1, scalingFactor = 10.0, penalty = 0.0, maxGens = 200000, maxFevals = 200000, convTol = 1e-6, stallLimit = 225, optimalFitness = 0, optConvTol = 1e-2)
```

Constructor to build the GnoweeHeuristics class.

The default settings are found to be optimized for a wide range of problems, but can be changed to optimize performance for a particular problem type or class. For more details, refer to the benchmark code in the development branch of the repo or <insert link="" to="" paper>="">.

If the optimizal fitness is unknown, as it often is, this can be left as zero or some reasonable guess based on the understanding of the problem. If the opimtimal fitness is set below what is actually obatinable, the only impact is the removal of this convergence criteria, and the program will still run.

self	GnoweeHeuristic pointer
	The GnoweeHeuristics pointer.
population	integer
	The number of members in each generation.
initSampling	string
	The method used to sample the phase space and create the initial population. Valid options are 'random', 'nolh', 'nolh-rp', 'nolh-cdr', and 'lhc' as specified in init_samples().
fracDiscovered	: float
	Discovery probability used for the mutate() heuristic.
fracElite	float
	Elite fraction probability used for the scatter_search(), crossover(), and cont_crossover() heuristics.

fracLevy	float Levy flight probability used for the disc_levy_flight() and cont_levy_flight() heuristics.
alpha	float Levy exponent - defines the index of the distribution and controls scale properties of the stochastic process.
gamma	float Gamma - scale unit of process for Levy flights.
n	integer Number of independent variables - can be used to reduce Levy flight sampling variance.
penalty	float Individual constraint violation penalty to add to objective function.
scalingFactor	float Step size scaling factor used to adjust Levy flights to length scale of system. The implementation of the Levy flight sampling makes this largely arbitrary.
maxGens	integer The maximum number of generations to search.
maxFevals	integer The maximum number of objective function evaluations.
convTol	float The minimum change of the best objective value before the search terminates.
stallLimit	integer The maximum number of generations to search without a descrease exceeding convTol.
optimalFitness	float The best know fitness value for the problem considered used to test for convergence.
optConvTol	float The maximum deviation from the best know fitness value before the search terminates.

9.3.3 Member Function Documentation

9.3.3.1 def GnoweeHeuristics.GnoweeHeuristics.__repr__ (self)

GnoweeHeuristics print function.

Parameters

self	GnoweeHeuristics pointer
	The GnoweeHeuristics pointer.

9.3.3.2 def GnoweeHeuristics.GnoweeHeuristics. $_$ str $_$ (self)

Human readable GnoweeHeuristics print function.

Parameters

self	GnoweeHeuristics pointer
	The GnoweeHeuristics pointer.

9.3.3.3 def GnoweeHeuristics.GnoweeHeuristics.cont_levy_flight (self, pop, lb, ub, varID)

Generate new children using Levy flights permutation of current generation design parameters according to:

$$x_r^{g+1} = x_r^g + \frac{1}{\beta} L_{\alpha,\gamma},$$

where $L_{\alpha,\gamma}$ is calculated in levy() according to the Mantegna algorithm. Applies rejection_bounds() to ensure all solutions lie within the design space by adapting the step size to the size of the design space.

Parameters

self	GnoweeHeuristic pointer
	The GnoweeHeuristics pointer.
рор	list of arrays
	The current parent sets of design variables representing system designs for the population.
lb	array
	The lower bounds of the design variable(s).
ub	array
	The upper bounds of the design variable(s).
varID	array
	A truth array indicating the location of the variables to be permuted. If the variable is to be permuted, a 1 is inserted at the variable location; otherwise a 0.

Returns

list of arrays: The proposed children sets of design variables representing the updated design parameters. *list:* A list of the identities of the chosen index for each child.

9.3.3.4 def GnoweeHeuristics.GnoweeHeuristics.crossover (self, pop, lb, ub, varID, intDiscID = None)

Generate new children using distance based crossover strategies on the top parent.

Ideas adapted from Walton "Modified Cuckoo Search: A New Gradient Free Optimisation Algorithm" and Storn "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces"

self	GnoweeHeuristic pointer The GnoweeHeuristics pointer.
рор	list of arrays The current parent sets of design variables representing system designs for the population.

lb	array The lower bounds of the design variable(s).
ub	array The upper bounds of the design variable(s).
varID	array A truth array indicating the location of the variables to be permuted. If the variable is to be permuted, a 1 is inserted at the variable location; otherwise a 0.
intDiscID	A truth array indicating the location of the discrete variable. A 1 is inserted at the discrete variable location; otherwise a 0. If no discrete variables, the array will be set to 0 automatically.

Returns

list of arrays: The proposed children sets of design variables representing the updated design parameters. *list:* A list of the identities of the chosen index for each child.

$9.3.3.5 \quad \text{def GnoweeHeuristics.GnoweeHeuristics.disc_levy_flight (} \quad \textit{self, pop, lb, ub, varID} \text{)}$

Generate new children using truncated Levy flights permutation of current generation design parameters according to:

$$L_{\alpha,\gamma} = FLOOR(TLF_{\alpha,\gamma} * D(x)),$$

where $TLF_{\alpha,\gamma}$ is calculated in tlf(). Applies rejection_bounds() to ensure all solutions lie within the design space by adapting the step size to the size of the design space.

Parameters

self	GnoweeHeuristic pointer
	The GnoweeHeuristics pointer.
рор	list of arrays
	The current parent sets of design variables representing system designs for the population.
lb	array
	The lower bounds of the design variable(s).
ub	array
	The upper bounds of the design variable(s).
varID	array
	A truth array indicating the location of the variables to be permuted. If the variable is to be permuted, a 1 is inserted at the variable location; otherwise a 0.

Returns

list of arrays: The proposed children sets of design variables representing the updated design parameters. *list:* A list of the identities of the chosen index for each child.

9.3.3.6 def GnoweeHeuristics.GnoweeHeuristics.elite_crossover (self, pop)

Generate new designs by using inver-over on combinatorial variables.

Adapted from ideas in Tao, "Iver-over Operator for the TSP."

Parameters

self	GnoweeHeuristic pointer
	The GnoweeHeuristics pointer.
рор	list of arrays
	The current parent sets of design variables representing system designs for the population.

Returns

list of arrays: The proposed children sets of design variables representing the updated design parameters.

9.3.3.7 def GnoweeHeuristics.GnoweeHeuristics.initialize (self, numSamples, sampleMethod, lb, ub, varType)

Initialize the population according to the sampling method chosen.

Parameters

self	GnoweeHeuristic pointer
	The GnoweeHeuristics pointer.
numSamples	integer
	The number of samples to be generated.
sampleMethod	string
	The method used to sample the phase space and create the initial population. Valid options
	are 'random', 'nolh', 'nolh-rp', 'nolh-cdr', and 'lhc' as specified in init_samples().
lb	array
	The lower bounds of the design variable(s).
ub	array
	The upper bounds of the design variable(s).
varType	array
	The type of variable for each design parameter. Allowed values: 'c' = continuous
	i' = integer/binary (difference denoted by ub/lb)
	'd' = discrete where the allowed values are given by the option discrete Vals nxm arrary with n=# of discrete variables and m=# of values that can be taken for each variable
	x' = combinatorial. All of the variables denoted by x are assumed to be "swappable" in
	combinatorial permutations. There must be at least two variables denoted as combinatorial. 'f' = fixed design variable
	j

Returns

list of arrays: The initialized set of samples.

9.3.3.8 def GnoweeHeuristics.GnoweeHeuristics.mutate (self, pop, lb, ub, varlD, intDisclD = None)

Generate new children by adding a weighted difference between two population vectors to a third vector.

Ideas adapted from Storn, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces" and Yang, "Nature Inspired Optimmization Algorithms"

Parameters

self	GnoweeHeuristic pointer
	The GnoweeHeuristics pointer.
рор	list of arrays
	The current parent sets of design variables representing system designs for the population.
lb	array
	The lower bounds of the design variable(s).
ub	array
	The upper bounds of the design variable(s).
varID	array
	A truth array indicating the location of the variables to be permuted. If the variable is to be
	permuted, a 1 is inserted at the variable location; otherwise a 0.
intDiscID	array
	A truth array indicating the location of the discrete variable. A 1 is inserted at the discrete variable location; otherwise a 0. If no discrete variables, the array will be set to 0 automatically.

Returns

list of arrays: The proposed children sets of design variables representing the updated design parameters.

9.3.3.9 def GnoweeHeuristics.GnoweeHeuristics.scatter_search (self, pop, lb, ub, varlD, intDisclD = None)

Generate new designs using the scatter search heuristic according to:

$$x^{g+1} = c_1 + (c_2 - c_1)r$$

where

$$c_1 = x^e - d(1 + \alpha \beta)$$

$$c_2 = x^e - d(1 - \alpha\beta)$$

$$d = \frac{x^r - x^e}{2}$$

and

$$\alpha=$$
 1 if i $<$ j & -1 if i $>$ j

$$\beta = \frac{|j-i|-1}{b-2}$$

where b is the size of the population.

Adapted from ideas in Egea, "An evolutionary method for complex- process optimization."

Applies simple_bounds() to ensure all solutions lie within the design space by adapting the step size to the size of the design space.

self	GnoweeHeuristic pointer
	The GnoweeHeuristics pointer.

рор	list of arrays
	The current parent sets of design variables representing system designs for the population.
lb	array
	The lower bounds of the design variable(s).
ub	array
	The upper bounds of the design variable(s).
varID	array
	A truth array indicating the location of the variables to be permuted. If the variable is to be permuted, a 1 is inserted at the variable location; otherwise a 0.
intDiscID	array
	A truth array indicating the location of the discrete variable. A 1 is inserted at the discrete variable location; otherwise a 0. If no discrete variables, the array will be set to 0 automatically.

Returns

list of arrays: The proposed children sets of design variables representing the updated design parameters. *list:* A list of the identities of the chosen index for each child.

9.3.4 Member Data Documentation

9.3.4.1 GnoweeHeuristics.GnoweeHeuristics.alpha

float Levy exponent - defines the index of the distribution and controls scale properties of the stochastic process.

9.3.4.2 GnoweeHeuristics.GnoweeHeuristics.convTol

float The minimum change of the best objective value before the search terminates.

9.3.4.3 GnoweeHeuristics.GnoweeHeuristics.fracDiscovered

float Discovery probability used for the mutate() heuristic.

9.3.4.4 GnoweeHeuristics.GnoweeHeuristics.fracElite

float Elite fraction probability used for the scatter_search(), crossover(), and cont_crossover() heuristics.

9.3.4.5 GnoweeHeuristics.GnoweeHeuristics.fracLevy

float Levy flight probability used for the disc_levy_flight() and cont_levy_flight() heuristics.

9.3.4.6 GnoweeHeuristics.GnoweeHeuristics.gamma

float Gamma - scale unit of process for Levy flights.

9.3.4.7 GnoweeHeuristics.GnoweeHeuristics.initSampling

string The method used to sample the phase space and create the initial population.

Valid options are 'random', 'nolh', 'nolh-rp', 'nolh-cdr', and 'lhc' as specified in init_samples().

9.3.4.8 GnoweeHeuristics.GnoweeHeuristics.maxFevals

integer The maximum number of objective function evaluations.

9.3.4.9 GnoweeHeuristics.GnoweeHeuristics.maxGens

integer The maximum number of generations to search.

9.3.4.10 GnoweeHeuristics.GnoweeHeuristics.n

integer Number of independent variables - can be used to reduce Levy flight sampling variance.

9.3.4.11 GnoweeHeuristics.GnoweeHeuristics.optConvTol

float The maximum deviation from the best know fitness value before the search terminates.

9.3.4.12 GnoweeHeuristics.GnoweeHeuristics.optimalFitness

float The best know fitness value for the problem considered used to test for convergence.

9.3.4.13 GnoweeHeuristics.GnoweeHeuristics.penalty

float Individual constraint violation penalty to add to objective function.

9.3.4.14 GnoweeHeuristics.GnoweeHeuristics.population

integer The number of members in each generation.

9.3.4.15 GnoweeHeuristics.GnoweeHeuristics.scalingFactor

float Step size scaling factor used to adjust Levy flights to length scale of system.

The implementation of the Levy flight sampling makes this largely arbitrary.

9.3.4.16 GnoweeHeuristics.GnoweeHeuristics.stallLimit

integer The maximum number of gen3rations to search without a descrease exceeding convTol.

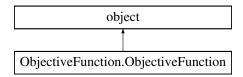
The documentation for this class was generated from the following file:

• /home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/GnoweeHeuristics.py

9.4 ObjectiveFunction.ObjectiveFunction Class Reference

The class creates a ObjectiveFunction object that can be used in optimization algorithms.

Inheritance diagram for ObjectiveFunction. ObjectiveFunction:



Public Member Functions

def init

Constructor to build the ObjectiveFunction class.

def __repr__

ObjectiveFunction class param print function.

def __str__

Human readable ObjectiveFunction print function.

· def set_obj_func

Converts an input string name for a function to a function handle.

def spring

Spring objective function with penalty method of constraint enforcement.

· def mi_spring

Spring objective function with penalty method of constraint enforcement.

· def welded beam

Welded Beam objective function with penalty method of constraint enforcement.

· def pressure_vessel

Pressure vessel objective function with penalty method of constraint enforcement.

• def mi_pressure_vessel

Mixed Integer Pressure vessel objective function with penalty method of constraint enforcement.

· def speed_reducer

Speed reducer objective function with penalty method of constraint enforcement.

• def mi_chemical_process

Chemical process design mixed integer problem.

· def ackley

Ackley Function: Mulitmodal, n dimensional.

def shifted_ackley

Ackley Function: Mulitmodal, n dimensional Ackley Function that is shifted from the symmetric 0, 0, 0, ..., 0 optimimum.

· def dejong

De Jong Function: Unimodal, n-dimensional.

def shifted_dejong

De Jong Function: Unimodal, n-dimensional De Jong Function that is shifted from the symmetric 0, 0, 0, ..., 0 optimimum.

• def easom

Easom Function: Multimodal, n-dimensional.

· def shifted_easom

Easom Function: Multimodal, n-dimensional Easom Function that is shifted from the symmetric pi, pi optimimum.

· def griewank

Griewank Function: Multimodal, n-dimensional.

def shifted_griewank

Griewank Function: Multimodal, n-dimensional Griewank Function that is shifted from the symmetric 0, 0, 0, ..., 0 optimimum.

· def rastrigin

Rastrigin Function: Multimodal, n-dimensional.

· def shifted_rastrigin

Rastrigin Function: Multimodal, n-dimensional Rastrigin Function that is shifted from the symmetric 0, 0, 0, ..., 0 optimimum.

· def rosenbrock

Rosenbrock Function: uni-modal, n-dimensional.

· def shifted rosenbrock

Rosenbrock Function: uni-modal, n-dimensional Rosenbrock Function that is shifted from the symmetric 0,0,0...0 optimimum.

· def tsp

Generic objective funtion to evaluate the TSP optimization by calculating total distance traveled.

Public Attributes

• func

function handle The function handle for the objective function to be used for the optimization.

· objective

integer, float, or numpy array The desired outcome of the optimization.

9.4.1 Detailed Description

The class creates a ObjectiveFunction object that can be used in optimization algorithms.

9.4.2 Constructor & Destructor Documentation

9.4.2.1 def ObjectiveFunction.ObjectiveFunction.__init__(self, method = None, objective = None)

Constructor to build the ObjectiveFunction class.

Parameters

self	pointer
	The ObjectiveFunction pointer.
method	string
	The name of the objective function to evaluate.
objective	integer, float, or numpy array
	The desired objective associated with the optimization. The chosen value and type must be compatible with the optiization function chosen.

9.4.3 Member Function Documentation

9.4.3.1 def ObjectiveFunction.ObjectiveFunction.__repr__ (self)

ObjectiveFunction class param print function.

self	pointer
	The ObjectiveFunction pointer.

9.4.3.2 def ObjectiveFunction.ObjectiveFunction.__str__ (self)

Human readable ObjectiveFunction print function.

Parameters

self	pointer
	The ObjectiveFunction pointer.

9.4.3.3 def ObjectiveFunction.ObjectiveFunction.ackley (self, u, penalty = 0.0)

Ackley Function: Mulitmodal, n dimensional.

Optimal example:

$$u = [0, 0, 0, 0, ... n-1]$$

fitness = 0.0

Taken from: "Nature-Inspired Optimization Algorithms"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array The design parameters to be evaluated.
penalty	float Per constraint violation penalty. Not used for this function; included for method input consistency.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.4 def ObjectiveFunction.ObjectiveFunction.dejong (self, u, penalty = 0.0)

De Jong Function: Unimodal, n-dimensional.

Optimal example:

$$u = [0, 0, 0, 0, ... n-1]$$

fitness = 0.0

Taken from: "Nature-Inspired Optimization Algorithms"

self	pointer The ObjectiveFunction pointer.
и	array The design parameters to be evaluated.

penalty	float
	Per constraint violation penalty. Not used for this function; included for method input consis-
	tency.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.5 def ObjectiveFunction.ObjectiveFunction.easom (self, u, penalty = 0.0)

Easom Function: Multimodal, n-dimensional.

Optimal example:

u = [pi, pi]

fitness = 1.0

Taken from: "Nature-Inspired Optimization Algorithms"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array The design parameters to be evaluated.
penalty	float Per constraint violation penalty. Not used for this function; included for method input consis-
	tency.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.6 def ObjectiveFunction.ObjectiveFunction.griewank (self, u, penalty = 0.0)

Griewank Function: Multimodal, n-dimensional.

Optimal example:

u = [0, 0, 0, ..., 0]

fitness = 0.0

Taken from: "Nature-Inspired Optimization Algorithms"

self	pointer
	The ObjectiveFunction pointer.

и	array
	The design parameters to be evaluated.
penalty	float
	Per constraint violation penalty. Not used for this function; included for method input consis-
	tency.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.7 def ObjectiveFunction.ObjectiveFunction.mi_chemical_process (self, u, penalty = 1E15)

Chemical process design mixed integer problem.

Optimal example:

u = [(0.2, 0.8, 1.907878, 1, 1, 0, 1]

fitness = 4.579582

Taken from: "An Improved PSO Algorithm for Solving Non-convex NLP/MINLP Problems with Equality Constraints"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array
	The design parameters to be evaluated. [x1, x2, x3, y1, y2, y3, y4]
penalty	float
	Per constraint violation penalty.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.8 def ObjectiveFunction.ObjectiveFunction.mi_pressure_vessel (self, u, penalty = 1 E 1 5)

Mixed Integer Pressure vessel objective function with penalty method of constraint enforcement.

Near optimal example:

u = [58.2298, 44.0291, 17, 9]

fitness = 7203.24

Optimal example obtained with Gnowee:

u = [38.819876, 221.985576, 0.750000, 0.375000]

fitness = 5855.893191

Taken from: "Nonlinear Integer and Discrete Programming in Mechanical Design Optimization"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array
	The design parameters to be evaluated.
penalty	float
	Per constraint violation penalty.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.9 def ObjectiveFunction.ObjectiveFunction.mi_spring (self, u, $penalty = 1 \times 15$)

Spring objective function with penalty method of constraint enforcement.

Optimal Example:

u = [1.22304104, 9, 36] = [1.22304104, 9, 0.307]

fitness = 2.65856

Taken from Lampinen, "Mixed Integer-Discrete-Continuous Optimization by Differential Evolution"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array
	The design parameters to be evaluated.
penalty	float
	Per constraint violation penalty.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.10 def ObjectiveFunction.ObjectiveFunction.pressure_vessel (self, u, penalty = 1E15)

Pressure vessel objective function with penalty method of constraint enforcement.

Near Optimal Example:

u = [0.81250000001, 0.4375, 42.098445595854923, 176.6365958424394]

fitness = 6059.714335

Optimal obtained using Gnowee:

 $u = [0.7781686880924992, \, 0.3846491857203429, \, 40.319621144688995, \, 199.99996630362293]$

fitness = 5885.33285347

Taken from: "Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array
	The design parameters to be evaluated.
penalty	float
	Per constraint violation penalty.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.11 def ObjectiveFunction.ObjectiveFunction.rastrigin (self, u, penalty = 0.0)

Rastrigin Function: Multimodal, n-dimensional.

Optimal example:

u = [0, 0, 0, ..., 0]

fitness = 0.0

Taken from: "Nature-Inspired Optimization Algorithms"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array The design parameters to be evaluated.
penalty	float Per constraint violation penalty. Not used for this function; included for method input consistency.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.12 def ObjectiveFunction.ObjectiveFunction.rosenbrock (self, u, penalty = 0.0)

Rosenbrock Function: uni-modal, n-dimensional.

Optimal example:

$$u = [1, 1, 1, ..., 1]$$

fitness = 0.0

Taken from: "Nature-Inspired Optimization Algorithms"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array
	The design parameters to be evaluated.
penalty	float
	Per constraint violation penalty. Not used for this function; included for method input consis-
	tency.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.13 def ObjectiveFunction.ObjectiveFunction.set_obj_func (self, funcName)

Converts an input string name for a function to a function handle.

Parameters

self	pointer The ObjectiveFunction pointer.
	· ·
funcName	string
	A string identifying the objective function to be used.

9.4.3.14 def ObjectiveFunction.ObjectiveFunction.shifted_ackley (self, u, penalty = 0.0)

Ackley Function: Mulitmodal, n dimensional Ackley Function that is shifted from the symmetric 0, 0, 0, ..., 0 optimimum.

Optimal example:

$$u = [0, 1, 2, 3, ... n-1]$$

fitness = 0.0

Taken from: "Nature-Inspired Optimization Algorithms"

self	pointer
	The ObjectiveFunction pointer.
и	array
	The design parameters to be evaluated.
penalty	float
	Per constraint violation penalty. Not used for this function; included for method input consis-
	tency.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.15 def ObjectiveFunction.ObjectiveFunction.shifted_dejong (self, u, penalty = 0.0)

De Jong Function: Unimodal, n-dimensional De Jong Function that is shifted from the symmetric 0, 0, 0, ..., 0 optimimum.

Optimal example:

$$u = [0, 1, 2, 3, ... n-1]$$

fitness = 0.0

Taken from: "Nature-Inspired Optimization Algorithms"

Taken from: "Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array
	The design parameters to be evaluated.
penalty	float
	Per constraint violation penalty. Not used for this function; included for method input consis-
	tency.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.16 def ObjectiveFunction.ObjectiveFunction.shifted_easom (self, u, penalty = 0.0)

Easom Function: Multimodal, n-dimensional Easom Function that is shifted from the symmetric pi, pi optimimum.

Optimal example:

$$u = [pi, pi+1]$$

fitness = 1.0

Taken from: "Nature-Inspired Optimization Algorithms"

self	pointer
	The ObjectiveFunction pointer.
и	array
	The design parameters to be evaluated.

penalty	float
	Per constraint violation penalty. Not used for this function; included for method input consis-
	tency.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.17 def ObjectiveFunction.ObjectiveFunction.shifted_griewank (self, u, penalty = 0.0)

Griewank Function: Multimodal, n-dimensional Griewank Function that is shifted from the symmetric 0, 0, 0, ..., 0 optimimum.

Optimal example:

$$u = [0, 1, 2, ..., n-1]$$

fitness = 0.0

Taken from: "Nature-Inspired Optimization Algorithms"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array
	The design parameters to be evaluated.
penalty	float
	Per constraint violation penalty. Not used for this function; included for method input consis-
	tency.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.18 def ObjectiveFunction.ObjectiveFunction.shifted_rastrigin (self, u, penalty = 0.0)

Rastrigin Function: Multimodal, n-dimensional Rastrigin Function that is shifted from the symmetric 0, 0, 0, ..., 0 optimimum.

Optimal example:

$$u = [0, 1, 2, ..., n-1]$$

fitness = 0.0

Taken from: "Nature-Inspired Optimization Algorithms"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array
	The design parameters to be evaluated.
penalty	float
	Per constraint violation penalty. Not used for this function; included for method input consis-
	tency.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.19 def ObjectiveFunction.ObjectiveFunction.shifted_rosenbrock (self, u, penalty = 0.0)

Rosenbrock Function: uni-modal, n-dimensional Rosenbrock Function that is shifted from the symmetric 0,0,0...0 optimimum.

Optimal example:

u = [1, 2, 3, ...n]

fitness = 0.0

Taken from: "Nature-Inspired Optimization Algorithms"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array The design parameters to be evaluated.
penalty	float Per constraint violation penalty. Not used for this function; included for method input consistency.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.20 def ObjectiveFunction.ObjectiveFunction.speed_reducer (self, u, penalty = 1 E 1 5)

Speed reducer objective function with penalty method of constraint enforcement.

Optimal example:

u = [58.2298, 44.0291, 17, 9]

fitness = 2996.34784914

Optimal example obtained with Gnowee:

u = [3.500000, 0.7, 17, 7.300000, 7.800000, 3.350214, 5.286683]

fitness = 5855.893191

Taken from: "Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array
	The design parameters to be evaluated.
penalty	float
	Per constraint violation penalty.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.21 def ObjectiveFunction.ObjectiveFunction.spring (self, u, penalty = 1E15)

Spring objective function with penalty method of constraint enforcement.

Optimal Example:

u = [0.05169046, 0.356750, 11.287126]

fitness = 0.0126653101469

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array
	The design parameters to be evaluated.
penalty	float
	Per constraint violation penalty.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.22 def ObjectiveFunction.ObjectiveFunction.tsp (self, u, penalty = 0.0)

Generic objective funtion to evaluate the TSP optimization by calculating total distance traveled.

self	pointer
	The ObjectiveFunction pointer.

и	array
	The city pairs to be evaluated.
penalty	float
	Per constraint violation penalty. Not used for this function; included for method input consis-
	tency.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.3.23 def ObjectiveFunction.ObjectiveFunction.welded_beam (self, u, penalty = 1E15)

Welded Beam objective function with penalty method of constraint enforcement.

Optimal Example:

u = [0.20572965, 3.47048857, 9.0366249, 0.20572965]

fitness = 1.7248525603892848

Taken from: "Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer"

Parameters

self	pointer
	The ObjectiveFunction pointer.
и	array
	The design parameters to be evaluated.
penalty	float
	Per constraint violation penalty.

Returns

array: The fitness associated with the specified input.

array: The assessed value for each constraint for the specified input.

9.4.4 Member Data Documentation

9.4.4.1 ObjectiveFunction.ObjectiveFunction.func

function handle The function handle for the objective function to be used for the optimization.

The function must be specified as a method of the class.

9.4.4.2 ObjectiveFunction.ObjectiveFunction.objective

integer, float, or numpy array The desired outcome of the optimization.

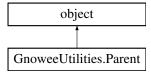
The documentation for this class was generated from the following file:

/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/ObjectiveFunction.py

9.5 GnoweeUtilities.Parent Class Reference

The class contains all of the parameters pertinent to a member of the population.

Inheritance diagram for GnoweeUtilities.Parent:



Public Member Functions

def __init__

Constructor to build the Parent class.

def repr

Parent print function.

def __str__

Human readable Parent print function.

Public Attributes

· variables

array: The set of variables representing a design solution.

· fitness

float: The assessed fitness for the current set of variables.

changeCount

integer: The number of improvements to the current population member.

stallCount

integer: he number of evaluations since the last improvement.

9.5.1 Detailed Description

The class contains all of the parameters pertinent to a member of the population.

9.5.2 Constructor & Destructor Documentation

9.5.2.1 def GnoweeUtilities.Parent.__init__ (self, variables = None, fitness = 1E15, changeCount = 0, stallCount = 0)

Constructor to build the Parent class.

self	Parent pointer
	The Parent pointer.

variables	array
	The set of variables representing a design solution.
fitness	float
	The assessed fitness for the current set of variables.
changeCount	integer
	The number of improvements to the current population member.
stallCount	integer
	The number of evaluations since the last improvement.

9.5.3 Member Function Documentation

9.5.3.1 def GnoweeUtilities.Parent.__repr__ (self)

Parent print function.

Parameters

self	Parent pointer
	The Parent pointer.

9.5.3.2 def GnoweeUtilities.Parent.__str__ (self)

Human readable Parent print function.

Parameters

self	Parent pointer
	The Parent pointer.

9.5.4 Member Data Documentation

9.5.4.1 GnoweeUtilities.Parent.changeCount

integer: The number of improvements to the current population member.

9.5.4.2 GnoweeUtilities.Parent.fitness

float: The assessed fitness for the current set of variables.

9.5.4.3 GnoweeUtilities.Parent.stallCount

integer: he number of evaluations since the last improvement.

9.5.4.4 GnoweeUtilities.Parent.variables

array: The set of variables representing a design solution.

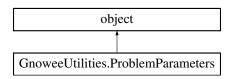
The documentation for this class was generated from the following file:

/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/GnoweeUtilities.py

9.6 GnoweeUtilities.ProblemParameters Class Reference

Creates an object containing key features of the chosen optimization problem.

Inheritance diagram for GnoweeUtilities.ProblemParameters:



Public Member Functions

- def __init__
- def __repr__

ProblemParameters class attribute print function.

def __str__

Human readable ProblemParameters print function.

def get_preset_params

Instantiates a ProblemParameters object and populations member variables from a set of predefined problem types.

Public Attributes

Ib

array: The lower bounds of the design variable(s).

ub

array: The upper bounds of the design variable(s).

varType

array: The type of variable for each position in the upper and lower bounds array.

· discreteVals

array: nxm with n=# of discrete variables and m=# of values that can be taken for each variable.

optimum

float: The global optimal solution.

pltTitle

string: The title used for plotting the results of the optimization.

histTitle

string: The plot title for the histogram of the optimization results.

varNames

list of strings: The names of the variables for the optimization problem.

9.6.1 Detailed Description

Creates an object containing key features of the chosen optimization problem.

The methods provide a way of predefining problems for repeated use.

9.6.2 Constructor & Destructor Documentation

```
9.6.2.1 def GnoweeUtilities.ProblemParameters.__init__ ( self, lowerBounds = [], upperBounds = [], varType = [], discreteVals = [], optimum = 0 . 0, pltTitle = ", histTitle = ", varNames = ")
```

Parameters

self	pointer The ProblemParameters pointer.
lowerBounds	array The lower bounds of the design variable(s). Only enter the bounds for continuous and integer/binary variables. The order must match the order specified in varType and ub.
upperBounds	array The upper bounds of the design variable(s). Only enter the bounds for continuous and integer/binary variables. The order must match the order specified in varType and lb.
varType	list or array The type of variable for each position in the upper and lower bounds array. Discrete variables are to be included last as they are specified separatly from the lb/ub throught the discreteVals optional input. A variable can have two types (for example, 'dx' could denote a layer that can take multiple materials and be placed at multiple design locations) Allowed values: 'c' = continuous over a given range (range specified in lb & ub). 'i' = integer/binary (difference denoted by ub/lb). 'd' = discrete where the allowed values are given by the option discreteVals nxm arrary with n=# of discrete variables and m=# of values that can be taken for each variable. 'x' = combinatorial. All of the variables denoted by x are assumed to be "swappable" in combinatorial permutations. There must be at least two variables denoted as combinatorial. 'f' = fixed design variable. Will not be considered of any permutation.
discreteVals	list of list(s) nxm with n=# of discrete variables and m=# of values that can be taken for each variable. For example, if you had two variables representing the tickness and diameter of a cylinder that take standard values, the discreteVals could be specified as: discreteVals = [[0.125, 0.25, 0.375], [0.25, 0.5, 075]] Gnowee will then map the optimization results to these allowed values.
optimum	float The global optimal solution.
pltTitle	string The title used for plotting the results of the optimization.
histTitle	string The plot title for the histogram of the optimization results.
varNames	list of strings The names of the variables for the optimization problem.

9.6.3 Member Function Documentation

9.6.3.1 def GnoweeUtilities.ProblemParameters.__repr__ (self)

ProblemParameters class attribute print function.

Parameters

Generated on Wed May 10 2017 20:42:18 for Gnowee by Doxygen

self	pointer	7
	The ProblemParameters pointer.	

9.6.3.2 def GnoweeUtilities.ProblemParameters.__str__ (self)

Human readable ProblemParameters print function.

Parameters

self	pointer
	The ProblemParameters pointer.

9.6.3.3 def GnoweeUtilities.ProblemParameters.get_preset_params (self, funct, algorithm = ", dimension = 2)

Instantiates a ProblemParameters object and populations member variables from a set of predefined problem types.

Parameters

self	pointer
	The ProblemParameters pointer.
funct	string
	Name of function being optimized.
algorithm	string
	Name of optimization program used.
dimension	integer
	Used to set the dimension for scalable problems.

Returns

ProblemParameters object: A ProblemParameters object with populated member variables.

9.6.4 Member Data Documentation

9.6.4.1 GnoweeUtilities.ProblemParameters.discreteVals

array: nxm with n=# of discrete variables and m=# of values that can be taken for each variable.

9.6.4.2 GnoweeUtilities.ProblemParameters.histTitle

string: The plot title for the histogram of the optimization results.

9.6.4.3 GnoweeUtilities.ProblemParameters.lb

array: The lower bounds of the design variable(s).

9.6.4.4 GnoweeUtilities.ProblemParameters.optimum

float: The global optimal solution.

9.6.4.5 GnoweeUtilities.ProblemParameters.pltTitle

string: The title used for plotting the results of the optimization.

9.6.4.6 def GnoweeUtilities.ProblemParameters.ub

array: The upper bounds of the design variable(s).

9.6.4.7 GnoweeUtilities.ProblemParameters.varNames

list of strings: The names of the variables for the optimization problem.

9.6.4.8 GnoweeUtilities.ProblemParameters.varType

array: The type of variable for each position in the upper and lower bounds array.

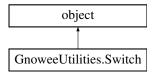
The documentation for this class was generated from the following file:

• /home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/GnoweeUtilities.py

9.7 GnoweeUtilities.Switch Class Reference

Creates a switch class object to switch between cases.

Inheritance diagram for GnoweeUtilities.Switch:



Public Member Functions

def __init__

Case constructor.

def __iter__

Return the match method once, then stop.

· def match

Indicate whether or not to enter a case suite.

Public Attributes

value

string: Case selector value.

fall

boolean: Match indicator.

9.7.1 Detailed Description

Creates a switch class object to switch between cases.

9.7.2 Constructor & Destructor Documentation

9.7.2.1 def GnoweeUtilities.Switch.__init__ (self, value)

Case constructor.

Parameters

self	pointer The Switch pointer.
value	string Case selector value.

9.7.3 Member Function Documentation

9.7.3.1 def GnoweeUtilities.Switch.__iter__ (self)

Return the match method once, then stop.

Parameters

self	pointer
	The Switch pointer.

9.7.3.2 def GnoweeUtilities.Switch.match (self, args)

Indicate whether or not to enter a case suite.

Parameters

self	pointer The Switch pointer.
*args	list List of comparisons.

Returns

boolean: Outcome of comparison match

9.7.4 Member Data Documentation

9.7.4.1 GnoweeUtilities.Switch.fall

boolean: Match indicator.

9.7.4.2 GnoweeUtilities.Switch.value

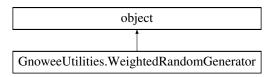
string: Case selector value.

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

/home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/GnoweeUtilities.py

9.8 GnoweeUtilities.WeightedRandomGenerator Class Reference

 $Inheritance\ diagram\ for\ Gnowee Utilities. Weighted Random Generator:$



Public Member Functions

- def __init__
- def next
- def call

Public Attributes

· totals

9.8.1 Detailed Description

Defines a class of weights to be used to select number of instances in array randomly with linear weighting.

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

• /home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/GnoweeUtilities.py

Chapter 10

File Documentation

10.1 /home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/Constraints.py File Reference

Classes

· class Constraints.Constraint

The class creates a Constraints object that can be used in optimization algorithms.

10.2 /home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/Gnowee.py File Reference

Namespaces

Gnowee

Contains the Gnowee optimization program and associated utilities.

Functions

· def Gnowee.main

Main controller program for the Gnowee optimization.

10.3 /home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/GnoweeHeuristics.py File Reference

Classes

· class GnoweeHeuristics.GnoweeHeuristics

The class is the foundation of the Gnowee optimization algorithm.

Namespaces

• Gnowee

Contains the Gnowee optimization program and associated utilities.

68 File Documentation

Functions

· def GnoweeHeuristics.simple bounds

Application of problem boundaries to generated solutions.

def GnoweeHeuristics.rejection_bounds

Application of problem boundaries to generated solutions.

10.4 /home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/GnoweeUtilities.py File Reference

Classes

· class GnoweeUtilities.Parent

The class contains all of the parameters pertinent to a member of the population.

· class GnoweeUtilities.Event

Represents a snapshot in the optimization process to be used for debugging, benchmarking, and user feedback.

• class GnoweeUtilities.ProblemParameters

Creates an object containing key features of the chosen optimization problem.

· class GnoweeUtilities.Switch

Creates a switch class object to switch between cases.

· class GnoweeUtilities.WeightedRandomGenerator

Namespaces

• Gnowee

Contains the Gnowee optimization program and associated utilities.

Functions

• def GnoweeUtilities.Get_Best

10.5 /home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/Objective-Function.py File Reference

Classes

· class ObjectiveFunction.ObjectiveFunction

The class creates a ObjectiveFunction object that can be used in optimization algorithms.

Namespaces

• Gnowee

Contains the Gnowee optimization program and associated utilities.

Functions

def ObjectiveFunction.prod

Computes the product of a set of numbers (ie big PI, mulitplicative equivalent to sum).

10.6 /home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/OptiPlot.py File Reference

Namespaces

• Gnowee

Contains the Gnowee optimization program and associated utilities.

Functions

· def OptiPlot.plot vars

Plot the variables as they change in the optimization process.

· def OptiPlot.plot_hist

Plots the histogram of function evaluation results from multiple runs of an optimization algorithm.

def OptiPlot.plot_hist_comp

Histograms and plots the comparison of two sets of function evaluation data.

· def OptiPlot.plot_feval_hist

Plots the fitness vs function evaluation results of an optimization algorithm run.

· def OptiPlot.plot tlf

Plots a comparison of the TLF to the Levy distribution.

· def OptiPlot.plot_optimization

Plots the results of optimization process for a given algorithm and parameter.

10.7 /home/pyne-user/Dropbox/UCB/Research/ETAs/Design/Gnowee/src/Sampling.py File Reference

Namespaces

Gnowee

Contains the Gnowee optimization program and associated utilities.

Functions

· def Sampling.initial_samples

Generate a set of samples in a given phase space.

• def Sampling.plot_samples

Plot the first 2 and 3 dimensions on the sample distribution.

· def Sampling.levy

Sample the Levy distribution given by.

def Sampling.tlf

Samples from a truncated Levy flight distribution (TLF) according to Manegna, "Stochastic Process with Ultraslow Convergence to a Gaussian: The Truncated Levy Flight" to map a levy distribution onto the interval [0,1].

· def Sampling.NOLH

This library allows to generate Nearly Orthogonal Latin Hypercubes (NOLH) according to Cioppa (2007) and De Rainville et al.

· def Sampling.params

Returns the NOLH order \$m\$, the required configuration length \$q\$ and the number of columns to remove to obtain the desired dimensionality.

• def Sampling.get_cdr_permutations

Generate a set of CDR permulations for NOLH.

Index

/home/pyne-user/Dropbox/UCB/Research/ETAs/-	Constraints.Constraint, 29
Design/Gnowee/src/Constraints.py, 67	Constraints::Constraint
/home/pyne-user/Dropbox/UCB/Research/ETAs/-	init, 30
Design/Gnowee/src/Gnowee.py, 67	repr, 30
/home/pyne-user/Dropbox/UCB/Research/ETAs/-	str, 30
Design/Gnowee/src/GnoweeHeuristics.py,	constraint, 32
67	func, 32
/home/pyne-user/Dropbox/UCB/Research/ETAs/-	get_penalty, 30
Design/Gnowee/src/GnoweeUtilities.py, 68	greater_than, 31
/home/pyne-user/Dropbox/UCB/Research/ETAs/-	less_or_equal, 31
Design/Gnowee/src/ObjectiveFunction.py, 68	less_than, 31
/home/pyne-user/Dropbox/UCB/Research/ETAs/-	set constraint func, 32
Design/Gnowee/src/OptiPlot.py, 69	cont_levy_flight
/home/pyne-user/Dropbox/UCB/Research/ETAs/-	GnoweeHeuristics::GnoweeHeuristics, 38
•	
Design/Gnowee/src/Sampling.py, 69	convTol
init	GnoweeHeuristics::GnoweeHeuristics, 42
Constraints::Constraint, 30	crossover
GnoweeHeuristics::GnoweeHeuristics, 36	GnoweeHeuristics::GnoweeHeuristics, 38
GnoweeUtilities::Event, 33	dejong
GnoweeUtilities::Parent, 58	ObjectiveFunction::ObjectiveFunction, 47
GnoweeUtilities::ProblemParameters, 61	design
GnoweeUtilities::Switch, 65	_
ObjectiveFunction::ObjectiveFunction, 45	GnoweeUtilities::Event, 34
iter	disc_levy_flight
GnoweeUtilities::Switch, 65	GnoweeHeuristics::GnoweeHeuristics, 39
repr	discrete Vals
Constraints::Constraint, 30	GnoweeUtilities::ProblemParameters, 63
GnoweeHeuristics::GnoweeHeuristics, 37	easom
GnoweeUtilities::Event, 33	ObjectiveFunction::ObjectiveFunction, 48
GnoweeUtilities::Parent, 59	elite crossover
GnoweeUtilities::ProblemParameters, 62	GnoweeHeuristics::GnoweeHeuristics, 39
ObjectiveFunction::ObjectiveFunction, 45	evaluations
str	
Constraints::Constraint, 30	GnoweeUtilities::Event, 34
GnoweeHeuristics::GnoweeHeuristics, 37	fall
GnoweeUtilities::Event, 34	GnoweeUtilities::Switch, 65
GnoweeUtilities::Parent, 59	fitness
GnoweeUtilities::ProblemParameters, 63	GnoweeUtilities::Event, 34
ObjectiveFunction::ObjectiveFunction, 45	GnoweeUtilities::Parent, 59
Objective unctionObjective unction, 40	fracDiscovered
ackley	
ObjectiveFunction::ObjectiveFunction, 47	GnoweeHeuristics::GnoweeHeuristics, 42
alpha	fracElite
GnoweeHeuristics::GnoweeHeuristics, 42	GnoweeHeuristics::GnoweeHeuristics, 42
Gnoweer leuristicsGnoweer leuristics, 42	fracLevy
changeCount	GnoweeHeuristics::GnoweeHeuristics, 42
GnoweeUtilities::Parent, 59	func
constraint	Constraints::Constraint, 32
Constraints::Constraint, 32	ObjectiveFunction::ObjectiveFunction, 57
	aamma
Constraints, 13	gamma

INDEX 71

GnoweeHeuristics::GnoweeHeuristics, 42	repr, 59
generation	str, 59
GnoweeUtilities::Event, 34	changeCount, 59
get_cdr_permutations	fitness, 59
Sampling, 23	stallCount, 59
get_penalty	variables, 59
Constraints::Constraint, 30	GnoweeUtilities::ProblemParameters
get_preset_params	init, 61
GnoweeUtilities::ProblemParameters, 63	repr, 62
Gnowee, 14, 27	str, 63
main, 14	discreteVals, 63
GnoweeHeuristics, 16	get_preset_params, 63
rejection_bounds, 16	histTitle, 63
simple bounds, 17	lb, 63
GnoweeHeuristics.GnoweeHeuristics, 34	optimum, 63
GnoweeHeuristics::GnoweeHeuristics	pltTitle, 63
init, 36	ub, 64
, 37	varNames, 64
, 37	varType, 64
alpha, 42	GnoweeUtilities::Switch
cont levy flight, 38	init, 65
convTol, 42	iter, 65
crossover, 38	fall, 65
disc_levy_flight, 39	match, 65
elite_crossover, 39	value, 65
fracDiscovered, 42	greater_than
fracElite, 42	Constraints::Constraint, 31
	griewank
fracLevy, 42	ObjectiveFunction::ObjectiveFunction, 48
gamma, 42	
initSampling, 42	histTitle
initialize, 40	GnoweeUtilities::ProblemParameters, 63
maxFevals, 43	,
maxGens, 43	initSampling
mutate, 40	GnoweeHeuristics::GnoweeHeuristics, 42
n, 43	initial_samples
optConvTol, 43	Sampling, 23
optimalFitness, 43	initialize
penalty, 43	GnoweeHeuristics::GnoweeHeuristics, 40
population, 43	,
scalingFactor, 43	lb
scatter_search, 41	GnoweeUtilities::ProblemParameters, 63
stallLimit, 43	less_or_equal
GnoweeUtilities, 18	Constraints::Constraint, 31
GnoweeUtilities.Event, 32	less_than
GnoweeUtilities.Parent, 58	Constraints::Constraint, 31
GnoweeUtilities.ProblemParameters, 60	levy
GnoweeUtilities.Switch, 64	Sampling, 24
GnoweeUtilities.WeightedRandomGenerator, 66	Camping, 2
GnoweeUtilities::Event	main
init, 33	Gnowee, 14
repr, 33	match
str, 34	GnoweeUtilities::Switch, 65
design, 34	maxFevals
evaluations, 34	GnoweeHeuristics::GnoweeHeuristics, 43
fitness, 34	maxGens
generation, 34	GnoweeHeuristics::GnoweeHeuristics, 43
GnoweeUtilities::Parent	mi_chemical_process
init, 58	ObjectiveFunction::ObjectiveFunction, 49
	Objectives anothermospectives unettern, 40

72 INDEX

mi_pressure_vessel	Sampling, 25
ObjectiveFunction::ObjectiveFunction, 49	penalty
mi_spring	GnoweeHeuristics::GnoweeHeuristics, 43
ObjectiveFunction::ObjectiveFunction, 50	plot_feval_hist
mutate	OptiPlot, 20
GnoweeHeuristics::GnoweeHeuristics, 40	plot_hist
	OptiPlot, 21
n	plot_hist_comp
GnoweeHeuristics::GnoweeHeuristics, 43	OptiPlot, 21
NOLH	plot_optimization
Sampling, 24	OptiPlot, 21
11. 8	plot_samples
objective	Sampling, 25
ObjectiveFunction::ObjectiveFunction, 57	plot_tlf
ObjectiveFunction, 19	OptiPlot, 22
prod, 19	plot_vars
ObjectiveFunction. ObjectiveFunction, 43	OptiPlot, 22
ObjectiveFunction::ObjectiveFunction	pltTitle
init, 45	GnoweeUtilities::ProblemParameters, 63
repr, 45	population
str, 45	GnoweeHeuristics::GnoweeHeuristics, 43
ackley, 47	pressure_vessel
dejong, 47	ObjectiveFunction::ObjectiveFunction, 50
easom, 48	prod
func, 57	ObjectiveFunction, 19
griewank, 48	
mi_chemical_process, 49	rastrigin
mi_pressure_vessel, 49	ObjectiveFunction::ObjectiveFunction, 51
mi_spring, 50	rejection_bounds
objective, 57	GnoweeHeuristics, 16
pressure_vessel, 50	rosenbrock
rastrigin, 51	ObjectiveFunction::ObjectiveFunction, 51
rosenbrock, 51	Compling 00
set_obj_func, 52	Sampling, 23
shifted_ackley, 52	get_cdr_permutations, 23
shifted_dejong, 53	initial_samples, 23
shifted_easom, 53	levy, 24
shifted_griewank, 54	NOLH, 24
shifted_rastrigin, 54	params, 25
shifted_rosenbrock, 55	plot_samples, 25 tlf, 25
speed_reducer, 55	
spring, 56	scalingFactor
tsp, 56	GnoweeHeuristics::GnoweeHeuristics, 43 scatter search
welded_beam, 57	GnoweeHeuristics::GnoweeHeuristics, 41
optConvTol	
GnoweeHeuristics::GnoweeHeuristics, 43	set_constraint_func
OptiPlot, 20	Constraints::Constraint, 32
plot_feval_hist, 20	set_obj_func
plot_hist, 21	ObjectiveFunction::ObjectiveFunction, 52
plot_hist_comp, 21	shifted_ackley
plot_optimization, 21	ObjectiveFunction::ObjectiveFunction, 52
plot_tlf, 22	shifted_dejong ObjectiveFunction: ObjectiveFunction 53
plot_vars, 22	ObjectiveFunction::ObjectiveFunction, 53
optimalFitness	shifted_easom
GnoweeHeuristics::GnoweeHeuristics, 43	ObjectiveFunction::ObjectiveFunction, 53
optimum	shifted_griewank
GnoweeUtilities::ProblemParameters, 63	ObjectiveFunction::ObjectiveFunction, 54
narama	shifted_rastrigin
params	ObjectiveFunction::ObjectiveFunction, 54

INDEX 73

```
shifted_rosenbrock
     ObjectiveFunction::ObjectiveFunction, 55
simple_bounds
    GnoweeHeuristics, 17
speed_reducer
    ObjectiveFunction::ObjectiveFunction, 55
spring
     ObjectiveFunction::ObjectiveFunction, 56
stallCount
    GnoweeUtilities::Parent, 59
stallLimit
    GnoweeHeuristics::GnoweeHeuristics, 43
tlf
     Sampling, 25
tsp
    ObjectiveFunction::ObjectiveFunction, 56
ub
    GnoweeUtilities::ProblemParameters, 64
value
    GnoweeUtilities::Switch, 65
varNames
    GnoweeUtilities::ProblemParameters, 64
varType
    GnoweeUtilities::ProblemParameters, 64
variables
    GnoweeUtilities::Parent, 59
welded_beam
    ObjectiveFunction::ObjectiveFunction, 57
```