MyPyOpt

Release 0.1

Edwin Lee

Nov 22, 2022

CONTENTS

This project is a very minimal optimization tool in Python. The basis for this work was my own development in an optimization course in grad school. The original implementation was in Fortran, but I converted it to Python, cleaned it up drastically, added tests and CI, docs, etc., and here it is.

The easiest way to see it in action is to check out the demos folder in the repository. It has demonstrations that use a simple Python function call as the simulation, and also one that pretends like it is calling EnergyPlus, both then calibrating given specific decision variables and an objective to minimize.

Contents:

CHAPTER

ONE

DECISION VARIABLE CLASS DOCUMENTATION

class mypyopt.decision_variable.**DecisionVariable**(*variable_name: str*, *minimum: float = -10000*, *maximum: float = 10000*, *initial_value: float = 1*, *initial_step_size: float = 0.1*, *convergence_criterion: float = 0.001*)

Bases: object

A structure for defining a single dimension in the optimization parameter space

to_dictionary() → dict

Converts the meaningful parts of this decision variable into a dictionary for project summary reports

Returns

Dictionary of decision variable information

CHAPTER

TWO

EXCEPTIONS CLASS DOCUMENTATION

exception mypyopt.exceptions.**MyPyOptException**

Bases: Exception

Currently just a simple wrapper around the base Exception class for convenience. It may evolve with additional capabilities eventually

CHAPTER

THREE

IO CLASS DOCUMENTATION

class mypyopt.input_output.**InputOutputManager**

Bases: object

This class defines some input/output-related conveniences

static write_line(*console: bool*, *full_output: Optional[TextIO]*, *string: str*)

A static method in the class used for convenience when printing out information.

Parameters

- **console** – A boolean for whether to report the string to standard output
- **full_output** – A file stream; if not None, it will be written to with the string
- **string** – The string to report; a newline is appended to the end if it doesn't have one already

CHAPTER

FOUR

OBJECTIVE EVALUATION RESULT CLASS DOCUMENTATION

class mypyopt.objective_evaluation.**ObjectiveEvaluation**(*state: int*, *value: Any*, *message: str = ''*)

Bases: object

This class defines the return type from an evaluation of the objective function.

The objective function is generally intended to be minimized by the optimizer search() function, so it is often a sum of squares error between some known quantity and the current outputs

CHAPTER

FIVE

OPTIMIZATION PROJECT STRUCTURE CLASS DOCUMENTATION

class mypyopt.project_structure.**ProjectStructure**(*expansion: float = 1.2*, *contraction: float = 0.85*, *max_iterations: int = 2000*, *project_name: str = 'project_name'*, *output_dir_path: Optional[Path] = None*, *verbose: bool = False*)

Bases: object

This class defines high level project-wide settings

CHAPTER

SIX

OPTIMIZER BASE CLASS DOCUMENTATION

class mypyopt.optimizer.**Optimizer**(*project_settings: ProjectStructure*, *decision_variable_array: List[DecisionVariable]*, *callback_f_of_x: Callable[[Dict[str, float]], Any]*, *callback_objective: Callable[[Any], List[float]]*, *input_output_worker: Optional[InputOutputManager] = None*, *callback_progress: Optional[Callable[[int, float], None]] = None*, *callback_completed: Optional[Callable[[SearchReturnType], None]] = None*)

Bases: object

This is a base class of an Optimizer to define the interface

abstract f_of_x(*parameter_hash: Dict[str, float]*)

This function calls the "f_of_x" callback function, getting outputs for the current parameter space; then passes those outputs into the objective function callback as an array, which usually returns the sum-sq-err between known values and current outputs.

Parameters

parameter_hash – A dictionary of parameters with keys as the variable names, and current variable values

abstract search() → SearchReturnType

This is the main driver function for the optimization. It walks the parameter space finding a minimum objective function. Requirements: call callback_progress and callback_completed as needed Call f(x) with a hash of parameter names and values

CHAPTER

SEVEN

OPTIMIZER (HEURISTIC SEARCH) CLASS DOCUMENTATION

class mypyopt.optimizer_heuristic_search.**HeuristicSearch**(*project_settings: ProjectStructure*, *decision_variable_array: List[DecisionVariable]*, *callback_f_of_x: Callable[[Dict[str, float]], Any]*, *callback_objective: Callable[[Any], List[float]]*, *input_output_worker: Optional[InputOutputManager] = None*, *callback_progress: Optional[Callable[[int, float], None]] = None*, *callback_completed: Optional[Callable[[SearchReturnType], None]] = None*)

Bases: `Optimizer`

This class implements a heuristic, multi-variable, search optimization technique. The process is:

1. Evaluate an objective value at the initial point $j_0 = f(x_0)$
2. Loop over each decision variable, perturb it in the current direction, and evaluate a new objective value with all other variables at their current position $j_i = f(\tilde{x})$
3. If the objective value reduced, which is the goal, move in the current direction and continue looping. If the objective value increased, reverse directions and contract.
4. Continue looping until all decision variables are converged between the current and prior iteration, or maximum iterations is reached.

f_of_x(*parameter_hash: Dict[str, float]*)

This function calls the "f_of_x" callback function, getting outputs for the current parameter space; then passes those outputs into the objective function callback as an array, which usually returns the sum-sq-err between known values and current outputs.

search() → SearchReturnType

This is the main driver function for the optimization. It walks the parameter space finding a minimum objective function.

CHAPTER

EIGHT

RETURN STATE ENUMERATION CLASS DOCUMENTATION

class mypyopt.return_state_enum.**ReturnStateEnum**

Bases: object

This class simply defines some constants for how functions return

InfeasibleDV = -1

Search failed because the decision variable went out of the valid parameter space range

InfeasibleObj = -2

Search failed because the objective function could not be calculated, probably a failed simulation call

InvalidInitialPoint = -4

Search failed because the initial point was invalid

Successful = 0

Search returned successfully

UnsuccessfulOther = -3

Search failed for an unknown reason

UserAborted = -9

Search was stopped because the user forced it to stop

static all_enums() → List[int]

static enum_to_string(*enum*)

This static function converts an enumerated constant integer into a string representation

Parameters

enum – A constant as defined in this class

Returns

A string description of the constant

CHAPTER

NINE

SEARCH RETURN TYPE CLASS DOCUMENTATION

class mypyopt.search_return_type.**SearchReturnType**(*success*, *error_reason*, *values=None*)

Bases: object

This class defines a response structure for a given project search

CHAPTER

TEN

INDEX AND TABLES

- genindex
- modindex
- search