
ENOPPY

Release 0.1.1

Thieu

Dec 18, 2023

INTRODUCTION:

1	Features	3
1.1	Quick Start	3
1.1.1	Installation	3
1.1.2	Lib's structure	3
1.1.3	Usage	4
1.1.4	Examples	4
1.2	ENOPPY Library	5
1.2.1	enoppy.paper_based	5
1.2.1.1	enoppy.paper_based.ihaoavoa_2022	5
1.2.1.2	enoppy.paper_based.moeosma_2023	7
1.2.1.3	enoppy.paper_based.pdo_2022	10
1.2.1.4	enoppy.paper_based.rwco_2020	16
1.2.2	enoppy.utils	29
1.2.2.1	enoppy.utils.encoder	29
1.2.2.2	enoppy.utils.validator	29
1.2.3	enoppy.engineer	30
1.3	Cite Us	32
1.4	Important links	32
1.5	License	33
2	Indices and tables	35
	Python Module Index	37
	Index	39

ENOPPY (ENgineering Optimization Problems in PYthon) is the largest python library for real-world engineering optimization problems. Contains all real-world engineering problems from CEC competitions and research papers.

- **Free software:** GNU General Public License (GPL) V3 license
- **Total problems:** > 50 problems
- **Documentation:** <https://enoppy.readthedocs.io/en/latest/>
- **Python versions:** 3.7.x, 3.8.x, 3.9.x, 3.10.x, 3.11.x
- **Dependencies:** numpy, scipy

FEATURES

- Our library provides all state-of-the-art engineering optimization problems.
- We have implemented all problems using Numpy to increase the speed of the algorithms.

1.1 Quick Start

1.1.1 Installation

- Install the [current PyPI release](#):

```
$ pip install enoppy==0.1.1
```

- Install directly from source code:

```
$ git clone https://github.com/thieu1995/enoppy.git
$ cd enoppy
$ python setup.py install
```

1.1.2 Lib's structure

Current's structure:

```
docs
examples
enoppy
  paper_based
    pdo_2022.py
    rwco_2020.py
  problem_based
    chemical.py
    mechanism.py
  utils
    validator.py
    visualize.py
  __init__.py
  engineer.py
README.md
setup.py
```

1.1.3 Usage

After installation, you can import ENOPPY as any other Python module:

```
$ python
>>> import enoppy
>>> enoppy.__version__
```

Let's go through some examples.

1.1.4 Examples

How to get the problem and use it:

```
from enoppy.paper_based.moeosma_2023 import SpeedReducerProblem
# SRP = SpeedReducerProblem
# SP = SpringProblem
# HTBP = HydrostaticThrustBearingProblem
# VPP = VibratingPlatformProblem
# CSP = CarSideImpactProblem
# WRMP = WaterResourceManagementProblem
# BCP = BulkCarriersProblem
# MPBPP = MultiProductBatchPlantProblem

srp_prob = SpeedReducerProblem()
print("Lower bound for this problem: ", srp_prob.lb)
print("Upper bound for this problem: ", srp_prob.ub)
x0 = srp_prob.create_solution()
print("Get the objective values of x0: ", srp_prob.get_objs(x0))
print("Get the constraint values of x0: ", srp_prob.get_cons(x0))
print("Evaluate with default penalty function: ", srp_prob.evaluate(x0))
```

Design my own penalty function:

```
import numpy as np
from enoppy.paper_based.moeosma_2023 import HTBP
# HTBP = HydrostaticThrustBearingProblem

def penalty_func(list_objectives, list_constraints):
    list_constraints[list_constraints < 0] = 0
    return np.sum(list_objectives) + 1e5 * np.sum(list_constraints**2)

htbp_prob = HTBP(f_penalty=penalty_func)
print("Lower bound for this problem: ", htbp_prob.lb)
print("Upper bound for this problem: ", htbp_prob.ub)
x0 = htbp_prob.create_solution()
print("Get the objective values of x0: ", htbp_prob.get_objs(x0))
print("Get the constraint values of x0: ", htbp_prob.get_cons(x0))
print("Evaluate with default penalty function: ", htbp_prob.evaluate(x0))
```

For more usage examples please look at [examples]/(examples) folder.

1.2 ENOPPY Library

1.2.1 enoppy.paper_based

1.2.1.1 enoppy.paper_based.ihaoavoa_2022

enoppy.paper_based.ihaoavoa_2022.CBP

alias of *enoppy.paper_based.ihaoavoa_2022.CantileverBeamProblem*

class enoppy.paper_based.ihaoavoa_2022.CantileverBeamProblem(*f_penalty=None*)

Bases: *enoppy.engineer.Engineer*

x = [*x*₁, *x*₂, *x*₃, *x*₄, *x*₅]

evaluate(*x*)

Evaluation of the benchmark function.

Parameters *x* (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Cantilever beam design problem'

enoppy.paper_based.ihaoavoa_2022.REBP

alias of *enoppy.paper_based.ihaoavoa_2022.RollingElementBearingProblem*

class enoppy.paper_based.ihaoavoa_2022.RollingElementBearingProblem(*f_penalty=None*)

Bases: *enoppy.engineer.Engineer*

x = [*x*₁, *x*₂, *x*₃, *x*₄, *x*₅, *x*₆, *x*₇, *x*₈, *x*₉, *x*₁₀] = [*D*_m, *D*_b, *Z*, *f*_i, *f*₀, *K*_{dmin}, *K*_{dmax}, *theta*, *e*, *C*]

amend_position(*x*, *lb=None*, *ub=None*)

Amend position to fit the format of the problem

Parameters *x* (*np.ndarray*) – The current position (solution)

evaluate(*x*)

Evaluation of the benchmark function.

Parameters *x* (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Rolling element bearing design problem'

enoppy.paper_based.ihaoavoa_2022.SRP

alias of *enoppy.paper_based.ihaoavoa_2022.SpeedReducerProblem*

class enoppy.paper_based.ihaoavoa_2022.SpeedReducerProblem(*f_penalty=None*)

Bases: *enoppy.engineer.Engineer*

x = [*x*₁, *x*₂, *x*₃, *x*₄, *x*₅, *x*₆, *x*₇]

Ref: <https://www.hindawi.com/journals/mpe/2013/419043/>

evaluate(*x*)

Evaluation of the benchmark function.

Parameters *x* (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Speed reducer design problem'

enoppy.paper_based.ihaoavoa_2022.TCSP

alias of *enoppy.paper_based.ihaoavoa_2022.TensionCompressionSpringProblem*

class enoppy.paper_based.ihaoavoa_2022.TensionCompressionSpringProblem(*f_penalty=None*)

Bases: *enoppy.engineer.Engineer*

x = [*x*₁, *x*₂, *x*₃] = [*d*, *D*, *N*]

evaluate(*x*)

Evaluation of the benchmark function.

Parameters *x* (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Tension/compression spring design problem'

enoppy.paper_based.ihaoavoa_2022.WBP

alias of *enoppy.paper_based.ihaoavoa_2022.WeldedBeamProblem*

class enoppy.paper_based.ihaoavoa_2022.WeldedBeamProblem(*f_penalty=None*)

Bases: *enoppy.engineer.Engineer*

x = [*x*₁, *x*₂, *x*₃, *x*₄] = [*h*, *l*, *t*, *b*]

evaluate(*x*)

Evaluation of the benchmark function.

Parameters **x** (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Welded beam design problem'

1.2.1.2 enoppy.paper_based.moeosma_2023

`enoppy.paper_based.moeosma_2023.BCP`

alias of `enoppy.paper_based.moeosma_2023.BulkCarriersProblem`

class `enoppy.paper_based.moeosma_2023.BulkCarriersProblem`(*f_penalty=None*)

Bases: `enoppy.engineer.Engineer`

`x` = [L, B, D, T, Vk, CB] = [x1, x2, x3, x4, x5, x6]

Original Ref:

evaluate(*x*)

Evaluation of the benchmark function.

Parameters **x** (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Bulk carriers design problem'

`enoppy.paper_based.moeosma_2023.CSP`

alias of `enoppy.paper_based.moeosma_2023.CarSideImpactProblem`

class `enoppy.paper_based.moeosma_2023.CarSideImpactProblem`(*f_penalty=None*)

Bases: `enoppy.engineer.Engineer`

`x` = [x1, x2, x3, x4, x5, x6, x7]

Original Ref:

evaluate(*x*)

Evaluation of the benchmark function.

Parameters **x** (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Car side impact design problem'

`enoppy.paper_based.moeosma_2023.HTBP`

alias of `enoppy.paper_based.moeosma_2023.HydrostaticThrustBearingProblem`

class `enoppy.paper_based.moeosma_2023.HydrostaticThrustBearingProblem(f_penalty=None)`

Bases: `enoppy.engineer.Engineer`

$x = [x_1, x_2, x_3, x_4] = [R, R_0, \mu, Q]$

evaluate(x)

Evaluation of the benchmark function.

Parameters \mathbf{x} (*np.ndarray, list, tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Hydrostatic thrust bearing design problem'

`enoppy.paper_based.moeosma_2023.MPBPP`

alias of `enoppy.paper_based.moeosma_2023.MultiProductBatchPlantProblem`

class `enoppy.paper_based.moeosma_2023.MultiProductBatchPlantProblem(f_penalty=None)`

Bases: `enoppy.engineer.Engineer`

$x = [N_1, N_2, N_3, V_1, V_2, V_3, TL_1, TL_2, B_1, B_2] = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}]$

Original Ref:

amend_position(x, lb=None, ub=None)

Amend position to fit the format of the problem

Parameters \mathbf{x} (*np.ndarray*) – The current position (solution)

evaluate(x)

Evaluation of the benchmark function.

Parameters \mathbf{x} (*np.ndarray, list, tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Multi-product batch plant problem'

```

enoppy.paper_based.moeosma_2023.SP
    alias of enoppy.paper_based.moeosma_2023.SpringProblem

enoppy.paper_based.moeosma_2023.SRP
    alias of enoppy.paper_based.moeosma_2023.SpeedReducerProblem

class enoppy.paper_based.moeosma_2023.SpeedReducerProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    x = [x1, x2, x3, x4, x5, x6, x7] = [b, m, z, l1, l2, d1, d2]

    amend_position(x, lb=None, ub=None)
        Amend position to fit the format of the problem

        Parameters x (np.ndarray) – The current position (solution)

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the benchmark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Speed Reducer Design Problem'

class enoppy.paper_based.moeosma_2023.SpringProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    x = [x1, x2, x3] = [d, D, N]

    amend_position(x, lb=None, ub=None)
        Amend position to fit the format of the problem

        Parameters x (np.ndarray) – The current position (solution)

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the benchmark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Spring Design Problem'

enoppy.paper_based.moeosma_2023.VPP
    alias of enoppy.paper_based.moeosma_2023.VibratingPlatformProblem

```

```
class enoppy.paper_based.moeosma_2023.VibratingPlatformProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer
    x = [d1, d2, d3, b, L] = [x0, x1, x2, x3, x4]
    Original Ref: On improving multiobjective genetic algorithms for design optimization

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the benchmark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Vibrating platform design problem'

enoppy.paper_based.moeosma_2023.WRMP
    alias of enoppy.paper\_based.moeosma\_2023.WaterResourceManagementProblem

class enoppy.paper_based.moeosma_2023.WaterResourceManagementProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer
    x = [x1, x2, x3]
    Original Ref:

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the benchmark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Water resource management problem'
```

1.2.1.3 enoppy.paper_based.pdo_2022

```
enoppy.paper_based.pdo_2022.CBD
    alias of enoppy.paper\_based.pdo\_2022.CantileverBeamProblem

enoppy.paper_based.pdo_2022.CBHD
    alias of enoppy.paper\_based.pdo\_2022.CorrugatedBulkheadProblem

enoppy.paper_based.pdo_2022.CSP
    alias of enoppy.paper\_based.pdo\_2022.CompressionSpringProblem
```

```

class enoppy.paper_based.pdo_2022.CantileverBeamProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    Minimize a cantilever beam's weight. [x1, x2, x3, x4, x5]

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the bench-
            mark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Cantilever Beam Design Problem'

class enoppy.paper_based.pdo_2022.CompressionSpringProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    x = [x1, x2, x3, x4]

    CSD aims to minimize the weight of a tension/compression spring given the values of 3 parameters: the
        wire diameter (d=x1), number of active coils (P=x3), and mean coil diameter (D=x2).

    https://sci-hub.se/10.1016/s0166-3615\(99\)00046-9

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the bench-
            mark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Compression Spring Design Problem'

class enoppy.paper_based.pdo_2022.CorrugatedBulkheadProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    [x1, x2, x3, x4] = [width, depth, length, thickness]

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the bench-
            mark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

```

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Corrugated Bulkhead Design Problem'

`enoppy.paper_based.pdo_2022.GTD`

alias of `enoppy.paper_based.pdo_2022.GearTrainProblem`

class `enoppy.paper_based.pdo_2022.GearTrainProblem(f_penalty=None)`

Bases: `enoppy.engineer.Engineer`

Unconstrained discrete design optimization problem [x1, x2, x3, x4] = [n_A, n_B, n_C, n_D]

amend_position(x, lb=None, ub=None)

Amend position to fit the format of the problem

Parameters **x** (`np.ndarray`) – The current position (solution)

evaluate(x)

Evaluation of the benchmark function.

Parameters **x** (`np.ndarray`, `list`, `tuple`) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Gear Train Design Problem'

`enoppy.paper_based.pdo_2022.IBD`

alias of `enoppy.paper_based.pdo_2022.IBeamProblem`

class `enoppy.paper_based.pdo_2022.IBeamProblem(f_penalty=None)`

Bases: `enoppy.engineer.Engineer`

Minimizes the vertical deflection of a beam [x1, x2, x3, x4] = [b, h, t_w, t_f]

evaluate(x)

Evaluation of the benchmark function.

Parameters **x** (`np.ndarray`, `list`, `tuple`) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'I Beam Design Problem'

```

enoppy.paper_based.pdo_2022.PLD
    alias of enoppy.paper_based.pdo_2022.PistonLeverProblem

enoppy.paper_based.pdo_2022.PVP
    alias of enoppy.paper_based.pdo_2022.PressureVesselProblem

class enoppy.paper_based.pdo_2022.PistonLeverProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    [x1, x2, x3, x4] = [H, B, D, X]

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the benchmark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Piston Lever Design Problem'

class enoppy.paper_based.pdo_2022.PressureVesselProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    x = [x1, x2, x3, x4]

    Variables: the inner radius (R=x3), the thickness of the head (Th=x2), the length of the cylindrical section of the vessel (L=x4), and the thickness of the shell (Ts=x1)

    https://sci-hub.se/10.1115/1.2912596

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the benchmark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Pressure Vessel Design Problem'

enoppy.paper_based.pdo_2022.RCB
    alias of enoppy.paper_based.pdo_2022.ReinforcedConcreteBeamProblem

class enoppy.paper_based.pdo_2022.ReinforcedConcreteBeamProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    [x1, x2, x3]

```

amend_position(*x*, *lb=None*, *ub=None*)

Amend position to fit the format of the problem

Parameters *x* (*np.ndarray*) – The current position (solution)

evaluate(*x*)

Evaluation of the benchmark function.

Parameters *x* (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Reinforced Concrete Beam Design Problem'

`enoppy.paper_based.pdo_2022.SRD`

alias of `enoppy.paper_based.pdo_2022.SpeedReducerProblem`

class `enoppy.paper_based.pdo_2022.SpeedReducerProblem`(*f_penalty=None*)

Bases: `enoppy.engineer.Engineer`

Depicts a gearbox that sits between the propeller and engine of an aeroplane [*x*₁, *x*₂, *x*₃, *x*₄, *x*₅, *x*₆, *x*₇] = [*b*, *m*, *z*, *l*₁, *l*₂, *d*₁, *d*₂]

amend_position(*x*, *lb=None*, *ub=None*)

Amend position to fit the format of the problem

Parameters *x* (*np.ndarray*) – The current position (solution)

evaluate(*x*)

Evaluation of the benchmark function.

Parameters *x* (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Speed Reducer Design Problem'

`enoppy.paper_based.pdo_2022.TBTD`

alias of `enoppy.paper_based.pdo_2022.ThreeBarTrussProblem`

`enoppy.paper_based.pdo_2022.TCD`

alias of `enoppy.paper_based.pdo_2022.TubularColumnProblem`

class `enoppy.paper_based.pdo_2022.ThreeBarTrussProblem`(*f_penalty=None*)

Bases: `enoppy.engineer.Engineer`

Minimize three-bar structure weight subject to supporting a total load *P* acting vertically downwards

[x1, x2]

evaluate(x)

Evaluation of the benchmark function.

Parameters *x* (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Three Bar Truss Design Problem'

class `enoppy.paper_based.pdo_2022.TubularColumnProblem`(*f_penalty=None*)

Bases: `enoppy.engineer.Engineer`

[x1, x2] = [d, t]

<https://apmonitor.com/me575/index.php/Main/TubularColumn>

evaluate(x)

Evaluation of the benchmark function.

Parameters *x* (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Tubular Column Design Problem'

`enoppy.paper_based.pdo_2022.WBP`

alias of `enoppy.paper_based.pdo_2022.WeldedBeamProblem`

class `enoppy.paper_based.pdo_2022.WeldedBeamProblem`(*f_penalty=None*)

Bases: `enoppy.engineer.Engineer`

x = [x1, x2, x3, x4]

WBD is subjected to 4 design constraints: shear, beam blending stress, bar buckling load beam, and deflection variables: *h*=x1, *l*=x2, *t*=x3, *b*=x4 *l*: length, *h*: height, *t*: thickness, *b*: weld thickness of the bar

[https://sci-hub.se/10.1016/s0166-3615\(99\)00046-9](https://sci-hub.se/10.1016/s0166-3615(99)00046-9)

evaluate(x)

Evaluation of the benchmark function.

Parameters *x* (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Welded Beam Design Problem'

1.2.1.4 enoppy.paper_based.rwco_2020

enoppy.paper_based.rwco_2020.BPSP

alias of *enoppy.paper_based.rwco_2020.BlendingPoolingSeparationProblem*

class enoppy.paper_based.rwco_2020.BlendingPoolingSeparationProblem(f_penalty=None)

Bases: *enoppy.engineer.Engineer*

Industrial Chemical Processes [x1, x2, x3, x4,..., x37, x38] Blending-Pooling-Separation problem

evaluate(x)

Evaluation of the benchmark function.

Parameters x (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns val – the evaluated benchmark function

Return type float

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_eq_cons(x)

Compute the values of the equality constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Blending-Pooling-Separation problem (Industrial Chemical Processes)'

enoppy.paper_based.rwco_2020.CCSDP

alias of *enoppy.paper_based.rwco_2020.TensionCompressionSpringDesignProblem*

enoppy.paper_based.rwco_2020.HENDC1P

alias of *enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignCase1Problem*

enoppy.paper_based.rwco_2020.HENDC2P

alias of *enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignCase2Problem*

enoppy.paper_based.rwco_2020.HPP

alias of *enoppy.paper_based.rwco_2020.HaverlyPoolingProblem*

class enoppy.paper_based.rwco_2020.HaverlyPoolingProblem(f_penalty=None)

Bases: *enoppy.engineer.Engineer*

Industrial Chemical Processes [x1, x2, x3, x4,..., x9] Haverly's Pooling Problem

evaluate(x)

Evaluation of the benchmark function.

Parameters x (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns val – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_eq_cons(*x*)

Compute the values of the equality constraint functions for a given set of input values.

get_ineq_cons(*x*)

Compute the values of the inequality constraint functions for a given set of input values.

get_objs(*x*)

Maximum to minimum by using negative sign

name = "Haverly's Pooling Problem (Industrial Chemical Processes)"

class enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignCase1Problem(*f_penalty=None*)

Bases: [enoppy.engineer.Engineer](#)

Industrial Chemical Processes [*x*₁, *x*₂, *x*₃, *x*₄,..., *x*₉] Heat Exchanger Network Design (case 1)

evaluate(*x*)

Evaluation of the benchmark function.

Parameters *x* (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_eq_cons(*x*)

Compute the values of the equality constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Heat Exchanger Network Design Case 1 (Industrial Chemical Processes)'

class enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignCase2Problem(*f_penalty=None*)

Bases: [enoppy.engineer.Engineer](#)

Industrial Chemical Processes [*x*₁, *x*₂, *x*₃, *x*₄,..., *x*₁₀, *x*₁₁] Heat Exchanger Network Design (case 2)

evaluate(*x*)

Evaluation of the benchmark function.

Parameters *x* (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_eq_cons(*x*)

Compute the values of the equality constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

```

    name = 'Heat Exchanger Network Design Case 2 (Industrial Chemical Processes)'
enoppy.paper_based.rwco_2020.MDCBDP
    alias of enoppy.paper_based.rwco_2020.MultipleDiskClutchBrakeDesignProblem
enoppy.paper_based.rwco_2020.MPPBP
    alias of enoppy.paper_based.rwco_2020.MultiProductBatchPlantProblem
class enoppy.paper_based.rwco_2020.MultiProductBatchPlantProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer
    Process design and synthesis problems [x1, x2,..., x10] Multi-product batch plant
    amend_position(x, lb=None, ub=None)
        Amend position to fit the format of the problem
        Parameters x (np.ndarray) – The current position (solution)
    evaluate(x)
        Evaluation of the benchmark function.
        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the benchmark problem. Must have len(x) == self.n_dims.
        Returns val – the evaluated benchmark function
        Return type float
    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.
    get_ineq_cons(x)
        Compute the values of the inequality constraint functions for a given set of input values.
    get_objs(x)
        Compute the values of the objective functions for a given set of input values.
    name = 'Multi-product batch plant (Process design and synthesis problems)'
class enoppy.paper_based.rwco_2020.MultipleDiskClutchBrakeDesignProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer
    Mechanical design problems [x1, x2, x3, x4, x5] Multiple disk clutch brake design problem
    evaluate(x)
        Evaluation of the benchmark function.
        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the benchmark problem. Must have len(x) == self.n_dims.
        Returns val – the evaluated benchmark function
        Return type float
    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.
    get_ineq_cons(x)
        Compute the values of the inequality constraint functions for a given set of input values.
    get_objs(x)
        Compute the values of the objective functions for a given set of input values.
    name = 'Multiple disk clutch brake design problem (Mechanical design problems)'
enoppy.paper_based.rwco_2020.OBJ11(x, n)

```

```

enoppy.paper_based.rwco_2020.ODIRSP
    alias of enoppy.paper_based.rwco_2020.OptimalDesignIndustrialRefrigerationSystemProblem

enoppy.paper_based.rwco_2020.OOAUP
    alias of enoppy.paper_based.rwco_2020.OptimalOperationAlkylationUnitProblem

class enoppy.paper_based.rwco_2020.OptimalDesignIndustrialRefrigerationSystemProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    Mechanical design problems [x1, x2,..., x14] Optimal design of industrial refrigeration system

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the benchmark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_ineq_cons(x)
        Compute the values of the inequality constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Optimal design of industrial refrigeration system (Mechanical design problems)'

class enoppy.paper_based.rwco_2020.OptimalOperationAlkylationUnitProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    Industrial Chemical Processes [x1, x2, x3, x4,..., x7] Optimal Operation of Alkylation Unit

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the benchmark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_ineq_cons(x)
        Compute the values of the inequality constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Optimal Operation of Alkylation Unit (Industrial Chemical Processes)'

enoppy.paper_based.rwco_2020.PDP
    alias of enoppy.paper_based.rwco_2020.ProcessDesignProblem

enoppy.paper_based.rwco_2020.PFSP
    alias of enoppy.paper_based.rwco_2020.ProcessFlowSheetingProblem

```

```

enoppy.paper_based.rwco_2020.PGTDOP
    alias of enoppy.paper_based.rwco_2020.PlanetaryGearTrainDesignOptimizationProblem

enoppy.paper_based.rwco_2020.PINBNSP
    alias of enoppy.paper_based.rwco_2020.PropaneIsobutaneNButaneNonsharpSeparationProblem

enoppy.paper_based.rwco_2020.PS01P
    alias of enoppy.paper_based.rwco_2020.ProcessSynthesis01Problem

enoppy.paper_based.rwco_2020.PS02P
    alias of enoppy.paper_based.rwco_2020.ProcessSynthesis02Problem

enoppy.paper_based.rwco_2020.PSADP
    alias of enoppy.paper_based.rwco_2020.ProcessSynthesisAndDesignProblem

enoppy.paper_based.rwco_2020.PVDP
    alias of enoppy.paper_based.rwco_2020.PressureVesselDesignProblem

class enoppy.paper_based.rwco_2020.PlanetaryGearTrainDesignOptimizationProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    Mechanical design problems [x1, x2,...,x9] Planetary gear train design optimization problem

    amend_position(x, lb=None, ub=None)
        Amend position to fit the format of the problem

        Parameters x (np.ndarray) – The current position (solution)

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the benchmark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_eq_cons(x)
        Compute the values of the equality constraint functions for a given set of input values.

    get_ineq_cons(x)
        Compute the values of the inequality constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Planetary gear train design optimization problem (Mechanical design problems)'

class enoppy.paper_based.rwco_2020.PressureVesselDesignProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    Mechanical design problems [x1, x2, x3, x4] Pressure vessel design

    amend_position(x, lb=None, ub=None)
        Amend position to fit the format of the problem

        Parameters x (np.ndarray) – The current position (solution)

    evaluate(x)
        Evaluation of the benchmark function.

```


Parameters **x** (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_ineq_cons(*x*)

Compute the values of the inequality constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Pressure vessel design (Mechanical design problems)'

class `enoppy.paper_based.rwco_2020.ProcessDesignProblem`(*f_penalty=None*)

Bases: [`enoppy.engineer.Engineer`](#)

Process design and synthesis problems [*x*₁, *x*₂, ..., *x*₅] Process design Problem

amend_position(*x*, *lb=None*, *ub=None*)

Amend position to fit the format of the problem

Parameters **x** (*np.ndarray*) – The current position (solution)

evaluate(*x*)

Evaluation of the benchmark function.

Parameters **x** (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_ineq_cons(*x*)

Compute the values of the inequality constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Process design Problem (Process design and synthesis problems)'

class `enoppy.paper_based.rwco_2020.ProcessFlowSheetingProblem`(*f_penalty=None*)

Bases: [`enoppy.engineer.Engineer`](#)

Process design and synthesis problems [*x*₁, *x*₂, *x*₃] Process flow sheeting problem

amend_position(*x*, *lb=None*, *ub=None*)

Amend position to fit the format of the problem

Parameters **x** (*np.ndarray*) – The current position (solution)

evaluate(*x*)

Evaluation of the benchmark function.

Parameters **x** (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_ineq_cons(*x*)

Compute the values of the inequality constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Process flow sheeting problem (Process design and synthesis problems)'

class enoppy.paper_based.rwco_2020.ProcessSynthesis01Problem(*f_penalty=None*)

Bases: [enoppy.engineer.Engineer](#)

Process design and synthesis problems [*x*₁, *x*₂] Process synthesis problem 01

amend_position(*x*, *lb=None*, *ub=None*)

Amend position to fit the format of the problem

Parameters *x* (*np.ndarray*) – The current position (solution)

evaluate(*x*)

Evaluation of the benchmark function.

Parameters *x* (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_ineq_cons(*x*)

Compute the values of the inequality constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Process synthesis 01 problem (Process design and synthesis problems)'

class enoppy.paper_based.rwco_2020.ProcessSynthesis02Problem(*f_penalty=None*)

Bases: [enoppy.engineer.Engineer](#)

Process design and synthesis problems [*x*₁, *x*₂, ..., *x*₉] Process synthesis problem 02

amend_position(*x*, *lb=None*, *ub=None*)

Amend position to fit the format of the problem

Parameters *x* (*np.ndarray*) – The current position (solution)

evaluate(*x*)

Evaluation of the benchmark function.

Parameters *x* (*np.ndarray*, *list*, *tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

```

get_ineq_cons(x)
    Compute the values of the inequality constraint functions for a given set of input values.

get_objs(x)
    Compute the values of the objective functions for a given set of input values.

name = 'Process synthesis 02 problem (Process design and synthesis problems)'

class enoppy.paper_based.rwco_2020.ProcessSynthesisAndDesignProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    Process design and synthesis problems [x1, x2, x3] Process synthesis and design problem

    amend_position(x, lb=None, ub=None)
        Amend position to fit the format of the problem

        Parameters x (np.ndarray) – The current position (solution)

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the benchmark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_eq_cons(x)
        Compute the values of the equality constraint functions for a given set of input values.

    get_ineq_cons(x)
        Compute the values of the inequality constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Process synthesis and design problem (Process design and synthesis problems)'

class enoppy.paper_based.rwco_2020.PropaneIsobutaneNButaneNonsharpSeparationProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    Industrial Chemical Processes [x1, x2, x3, x4, ..., x47, x48] Propane, Isobutane, n-Butane Nonsharp Separation

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the benchmark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_eq_cons(x)
        Compute the values of the equality constraint functions for a given set of input values.

```

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Propane, Isobutane, n-Butane Nonsharp Separation (Industrial Chemical Processes)'

`enoppy.paper_based.rwco_2020.RNDP`

alias of `enoppy.paper_based.rwco_2020.ReactorNetworkDesignProblem`

class `enoppy.paper_based.rwco_2020.ReactorNetworkDesignProblem(f_penalty=None)`

Bases: `enoppy.engineer.Engineer`

Industrial Chemical Processes [x1, x2, x3, x4,..., x6] Reactor Network Design Problem

evaluate(x)

Evaluation of the benchmark function.

Parameters **x** (*np.ndarray, list, tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_eq_cons(x)

Compute the values of the equality constraint functions for a given set of input values.

get_ineq_cons(x)

Compute the values of the inequality constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Reactor Network Design (Industrial Chemical Processes)'

class `enoppy.paper_based.rwco_2020.RobotGripperProblem(f_penalty=None)`

Bases: `enoppy.engineer.Engineer`

Mechanical design problems [x1, x2, x3, x4, x5] Robot gripper problem

evaluate(x)

Evaluation of the benchmark function.

Parameters **x** (*np.ndarray, list, tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_eq_cons(x)

Compute the values of the equality constraint functions for a given set of input values.

get_ineq_cons(x)

Compute the values of the inequality constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Robot gripper problem (Mechanical design problems)'

`enoppy.paper_based.rwco_2020.SCPP`

alias of `enoppy.paper_based.rwco_2020.StepConePulleyProblem`

class `enoppy.paper_based.rwco_2020.StepConePulleyProblem(f_penalty=None)`

Bases: `enoppy.engineer.Engineer`

Mechanical design problems [x1, x2, x3, x4, x5] Step-cone pulley problem

evaluate(*x*)

Evaluation of the benchmark function.

Parameters *x* (`np.ndarray`, `list`, `tuple`) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_eq_cons(*x*)

Compute the values of the equality constraint functions for a given set of input values.

get_ineq_cons(*x*)

Compute the values of the inequality constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Step-cone pulley problem (Mechanical design problems)'

`enoppy.paper_based.rwco_2020.TBTDP`

alias of `enoppy.paper_based.rwco_2020.ThreeBarTrussDesignProblem`

`enoppy.paper_based.rwco_2020.TRP`

alias of `enoppy.paper_based.rwco_2020.TwoReactorProblem`

class `enoppy.paper_based.rwco_2020.TensionCompressionSpringDesignProblem(f_penalty=None)`

Bases: `enoppy.engineer.Engineer`

Mechanical design problems [x1, x2, x3] Tension/compression spring design

evaluate(*x*)

Evaluation of the benchmark function.

Parameters *x* (`np.ndarray`, `list`, `tuple`) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns *val* – the evaluated benchmark function

Return type float

get_cons(*x*)

Compute the values of the constraint functions for a given set of input values.

get_ineq_cons(*x*)

Compute the values of the inequality constraint functions for a given set of input values.

get_objs(*x*)

Compute the values of the objective functions for a given set of input values.

name = 'Tension/compression spring design (Mechanical design problems)'

```
class enoppy.paper_based.rwco_2020.ThreeBarTrussDesignProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    Mechanical design problems [x1, x2] Three-bar truss design problem

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the bench-
            mark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_ineq_cons(x)
        Compute the values of the inequality constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Three-bar truss design problem (Mechanical design problems)'
```

```
class enoppy.paper_based.rwco_2020.TwoReactorProblem(f_penalty=None)
    Bases: enoppy.engineer.Engineer

    Process design and synthesis problems [x1, x2, ..., x8] Two-reactor problem

    amend_position(x, lb=None, ub=None)
        Amend position to fit the format of the problem

        Parameters x (np.ndarray) – The current position (solution)

    evaluate(x)
        Evaluation of the benchmark function.

        Parameters x (np.ndarray, list, tuple) – The candidate vector for evaluating the bench-
            mark problem. Must have len(x) == self.n_dims.

        Returns val – the evaluated benchmark function

        Return type float

    get_cons(x)
        Compute the values of the constraint functions for a given set of input values.

    get_eq_cons(x)
        Compute the values of the equality constraint functions for a given set of input values.

    get_ineq_cons(x)
        Compute the values of the inequality constraint functions for a given set of input values.

    get_objs(x)
        Compute the values of the objective functions for a given set of input values.

    name = 'Two-reactor problem (Process design and synthesis problems)'
```

```
enoppy.paper_based.rwco_2020.WBDP
    alias of enoppy.paper\_based.rwco\_2020.WeldedBeamDesignProblem

enoppy.paper_based.rwco_2020.WMSRP
    alias of enoppy.paper\_based.rwco\_2020.WeightMinimizationSpeedReducerProblem
```

```
class enoppy.paper_based.rwco_2020.WeightMinimizationSpeedReducerProblem(f_penalty=None)
```

Bases: [enoppy.engineer.Engineer](#)

Mechanical design problems [x1, x2,..., x7] Weight minimization of a speed reducer

evaluate(x)

Evaluation of the benchmark function.

Parameters **x** (*np.ndarray, list, tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_ineq_cons(x)

Compute the values of the inequality constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Weight minimization of a speed reducer (Mechanical design problems)'

```
class enoppy.paper_based.rwco_2020.WeldedBeamDesignProblem(f_penalty=None)
```

Bases: [enoppy.engineer.Engineer](#)

Mechanical design problems [x1, x2, x3, x4] Welded beam design

evaluate(x)

Evaluation of the benchmark function.

Parameters **x** (*np.ndarray, list, tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.

Returns **val** – the evaluated benchmark function

Return type float

get_cons(x)

Compute the values of the constraint functions for a given set of input values.

get_ineq_cons(x)

Compute the values of the inequality constraint functions for a given set of input values.

get_objs(x)

Compute the values of the objective functions for a given set of input values.

name = 'Welded beam design (Mechanical design problems)'

```
enoppy.paper_based.rwco_2020.p1
```

alias of [enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignCase1Problem](#)

```
enoppy.paper_based.rwco_2020.p10
```

alias of [enoppy.paper_based.rwco_2020.ProcessFlowSheetingProblem](#)

```
enoppy.paper_based.rwco_2020.p11
```

alias of [enoppy.paper_based.rwco_2020.TwoReactorProblem](#)

```
enoppy.paper_based.rwco_2020.p12
```

alias of [enoppy.paper_based.rwco_2020.ProcessSynthesis02Problem](#)

```
enoppy.paper_based.rwco_2020.p13
```

alias of [enoppy.paper_based.rwco_2020.ProcessDesignProblem](#)

`enoppy.paper_based.rwco_2020.p14`
alias of `enoppy.paper_based.rwco_2020.MultiProductBatchPlantProblem`

`enoppy.paper_based.rwco_2020.p15`
alias of `enoppy.paper_based.rwco_2020.WeightMinimizationSpeedReducerProblem`

`enoppy.paper_based.rwco_2020.p16`
alias of `enoppy.paper_based.rwco_2020.OptimalDesignIndustrialRefrigerationSystemProblem`

`enoppy.paper_based.rwco_2020.p17`
alias of `enoppy.paper_based.rwco_2020.TensionCompressionSpringDesignProblem`

`enoppy.paper_based.rwco_2020.p18`
alias of `enoppy.paper_based.rwco_2020.PressureVesselDesignProblem`

`enoppy.paper_based.rwco_2020.p19`
alias of `enoppy.paper_based.rwco_2020.WeldedBeamDesignProblem`

`enoppy.paper_based.rwco_2020.p2`
alias of `enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignCase2Problem`

`enoppy.paper_based.rwco_2020.p20`
alias of `enoppy.paper_based.rwco_2020.ThreeBarTrussDesignProblem`

`enoppy.paper_based.rwco_2020.p21`
alias of `enoppy.paper_based.rwco_2020.MultipleDiskClutchBrakeDesignProblem`

`enoppy.paper_based.rwco_2020.p22`
alias of `enoppy.paper_based.rwco_2020.PlanetaryGearTrainDesignOptimizationProblem`

`enoppy.paper_based.rwco_2020.p23`
alias of `enoppy.paper_based.rwco_2020.StepConePulleyProblem`

`enoppy.paper_based.rwco_2020.p3`
alias of `enoppy.paper_based.rwco_2020.HaverlyPoolingProblem`

`enoppy.paper_based.rwco_2020.p4`
alias of `enoppy.paper_based.rwco_2020.BlendingPoolingSeparationProblem`

`enoppy.paper_based.rwco_2020.p5`
alias of `enoppy.paper_based.rwco_2020.PropaneIsobutaneNButaneNonsharpSeparationProblem`

`enoppy.paper_based.rwco_2020.p6`
alias of `enoppy.paper_based.rwco_2020.OptimalOperationAlkylationUnitProblem`

`enoppy.paper_based.rwco_2020.p7`
alias of `enoppy.paper_based.rwco_2020.ReactorNetworkDesignProblem`

`enoppy.paper_based.rwco_2020.p8`
alias of `enoppy.paper_based.rwco_2020.ProcessSynthesis01Problem`

`enoppy.paper_based.rwco_2020.p9`
alias of `enoppy.paper_based.rwco_2020.ProcessSynthesisAndDesignProblem`

1.2.2 enoppy.utils

1.2.2.1 enoppy.utils.encoder

class `enoppy.utils.encoder.LabelEncoder`

Bases: `object`

Encode categorical features as integer labels.

fit(*y*)

Fit label encoder to a given set of labels.

y [array-like] Labels to encode.

fit_transform(*y*)

Fit label encoder and return encoded labels.

Parameters *y* (*array-like of shape (n_samples,)*) – Target values.

Returns *y* – Encoded labels.

Return type array-like of shape (*n_samples*,)

inverse_transform(*y*)

Transform integer labels to original labels.

y [array-like] Encoded integer labels.

original_labels [array-like] Original labels.

transform(*y*)

Transform labels to encoded integer labels.

y [array-like] Labels to encode.

encoded_labels [array-like] Encoded integer labels.

1.2.2.2 enoppy.utils.validator

`enoppy.utils.validator.check_bool(name: str, value: bool, bound=(True, False))`

`enoppy.utils.validator.check_float(name: str, value: int, bound=None)`

`enoppy.utils.validator.check_int(name: str, value: int, bound=None)`

`enoppy.utils.validator.check_str(name: str, value: str, bound=None)`

`enoppy.utils.validator.check_tuple_float(name: str, values: tuple, bounds=None)`

`enoppy.utils.validator.check_tuple_int(name: str, values: tuple, bounds=None)`

`enoppy.utils.validator.is_in_bound(value, bound)`

`enoppy.utils.validator.is_str_in_list(value: str, my_list: list)`

1.2.3 enoppy.engineer

class enoppy.engineer.Engineer

Bases: abc.ABC

Defines an abstract class for engineering design problems.

All subclasses should implement the `evaluate` method for a particular problem.

bounds

The lower/upper bounds of the problem. This a 2D-matrix of [lower, upper] array that contain the lower and upper bounds. By default, each problem has its own bounds. But user can try to put different bounds to test the problem.

Type list

n_dims

The dimensionality of the problem. It is calculated from bounds

Type int

lb

The lower bounds for the problem

Type np.ndarray

ub

The upper bounds for the problem

Type np.ndarray

f_global

The global optimum of the evaluated function.

Type float

x_global

A list of vectors that provide the locations of the global minimum. Note that some problems have multiple global minima, not all of which may be listed.

Type np.ndarray

n_fe

The number of function evaluations that the object has been asked to calculate.

Type int

amend_position(*x*, *lb=None*, *ub=None*)

Amend position to fit the format of the problem

Parameters *x* (*np.ndarray*) – The current position (solution)

property bounds

The lower/upper bounds to be used for optimization problem. This a 2D-matrix of [lower, upper] array that contain the lower and upper bounds for the problem. The problem should not be asked for evaluation outside these bounds. `len(bounds) == n_dims`.

check_penalty_func(*func=None*)

check_solution(*x*)

Raise the error if the problem size is not equal to the solution length

Parameters *x* (*np.ndarray*) – The solution

convex = True

create_solution()
 Create a random solution for the current problem
Returns **solution** – The random solution
Return type np.ndarray

default_penalty(*list_objs=None, list_cons=None*)

differentiable = **True**

evaluate(*x*)
 Evaluation of the benchmark function.
Parameters **x** (*np.ndarray, list, tuple*) – The candidate vector for evaluating the benchmark problem. Must have `len(x) == self.n_dims`.
Returns **val** – the evaluated benchmark function
Return type float

get_cons(*x*)
 Compute the values of the constraint functions for a given set of input values.

get_eq_cons(*x*)
 Compute the values of the equality constraint functions for a given set of input values.

get_ineq_cons(*x*)
 Compute the values of the inequality constraint functions for a given set of input values.

get_objs(*x*)
 Compute the values of the objective functions for a given set of input values.

get_paras()
 Return the parameters of the problem. Depended on function

property lb
 The lower bounds for the problem
Returns **lb** – The lower bounds for the problem
Return type 1D-vector

linear = **False**

property n_cons
 The number of constraint functions of the problem.

property n_dims
 The dimensionality of the problem.

property n_eq_cons
 The number of equality constraint functions of the problem.

property n_ineq_cons
 The number of inequality constraint functions of the problem.

property n_objs
 The number of objective functions of the problem.

name = **'Benchmark name'**

parametric = **True**

property ub
 The upper bounds for the problem

Returns `ub` – The upper bounds for the problem

Return type 1D-vector

1.3 Cite Us

If you are using enoppy in your project, we would appreciate citations:

```
@software{nguyen_van_thieu_2023_7953207,
  author      = {Nguyen Van Thieu},
  title       = {ENOPPY: A Python Library for Engineering Optimization Problems},
  month       = may,
  year        = 2023,
  publisher    = {Zenodo},
  doi         = {10.5281/zenodo.7953206},
  url         = {https://github.com/thieu1995/enoppy}
}

@article{van2023mealpy,
  title={MEALPY: An open-source library for latest meta-heuristic algorithms in Python},
  author={Van Thieu, Nguyen and Mirjalili, Seyedali},
  journal={Journal of Systems Architecture},
  year={2023},
  publisher={Elsevier},
  doi={10.1016/j.sysarc.2023.102871}
}
```

If you have an open-ended or a research question, you can contact me via nguyenthieu2102@gmail.com

1.4 Important links

- Official source code repo: <https://github.com/thieu1995/enoppy>
- Official document: <https://enoppy.readthedocs.io/>
- Download releases: <https://pypi.org/project/enoppy/>
- Issue tracker: <https://github.com/thieu1995/enoppy/issues>
- Notable changes log: <https://github.com/thieu1995/enoppy/blob/master/ChangeLog.md>
- Examples with different mealpy version: <https://github.com/thieu1995/enoppy/blob/master/EXAMPLES.md>
- Join our telegram community: [link](<https://t.me/+fRVCJGuGJg1mNDg1>)
- **This project also related to my another projects which are “meta-heuristics” and “neural-network”, check it here**
 - <https://github.com/thieu1995/mealpy>
 - <https://github.com/thieu1995/permetrics>
 - <https://github.com/thieu1995/opfunu>
 - <https://github.com/thieu1995/metaheuristics>
 - <https://github.com/thieu1995/MetaCluster>

- <https://github.com/thieu1995/pfevaluator>
- <https://github.com/thieu1995/IntelELM>
- <https://github.com/thieu1995/MetaPerceptron>
- <https://github.com/thieu1995/GrafoRVFL>
- <https://github.com/thieu1995/reflame>
- <https://github.com/aiir-team>

1.5 License

The project is licensed under GNU General Public License (GPL) V3 license.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

e

- `enoppy.engineer`, [30](#)
- `enoppy.paper_based.ihaoavoa_2022`, [5](#)
- `enoppy.paper_based.moeosma_2023`, [7](#)
- `enoppy.paper_based.pdo_2022`, [10](#)
- `enoppy.paper_based.rwco_2020`, [16](#)
- `enoppy.utils.encoder`, [29](#)
- `enoppy.utils.validator`, [29](#)

INDEX

A

amend_position() (enoppo.engineer.Engineer method), 30

amend_position() (enoppo.paper_based.ihaoavoa_2022.RollingElementBearingProblem method), 5

amend_position() (enoppo.paper_based.moeosma_2023.CantileverBeamProblem method), 8

amend_position() (enoppo.paper_based.moeosma_2023.SpatialCantileverBeamProblem method), 9

amend_position() (enoppo.paper_based.moeosma_2023.SpringImpactProblem method), 9

amend_position() (enoppo.paper_based.pdo_2022.GearTrainProblem method), 12

amend_position() (enoppo.paper_based.pdo_2022.ReinforcedConcreteBeamProblem method), 13

amend_position() (enoppo.paper_based.pdo_2022.SpeedReductionProblem method), 14

amend_position() (enoppo.paper_based.rwco_2020.MultiBodyCrashProblem method), 18

amend_position() (enoppo.paper_based.rwco_2020.PlanetaryGearTrainDesignOptimizationProblem method), 20

amend_position() (enoppo.paper_based.rwco_2020.PressureVesselDesignProblem method), 20

amend_position() (enoppo.paper_based.rwco_2020.ProcessSynthesisProblem method), 21

amend_position() (enoppo.paper_based.rwco_2020.ProcessUnitSizingProblem method), 21

amend_position() (enoppo.paper_based.rwco_2020.ProcessSynthesisProblem method), 22

amend_position() (enoppo.paper_based.rwco_2020.ProcessSynthesisProblem method), 22

amend_position() (enoppo.paper_based.rwco_2020.ProcessSynthesisProblem method), 23

amend_position() (enoppo.paper_based.rwco_2020.TwoReactorProblem method), 26

BPSP (in module enoppo.paper_based.rwco_2020), 16

BulkCarriersProblem (class in enoppo.paper_based.moeosma_2023), 7

C

CantileverBeamProblem (class in enoppo.paper_based.ihaoavoa_2022), 5

SpatialCantileverBeamProblem (class in enoppo.paper_based.pdo_2022), 10

SpringImpactProblem (class in enoppo.paper_based.moeosma_2023), 7

GearTrainProblem (in module enoppo.paper_based.pdo_2022), 10

CBHD (in module enoppo.paper_based.pdo_2022), 10

ReinforcedConcreteBeamProblem (in module enoppo.paper_based.ihaoavoa_2022), 5

CCSDP (in module enoppo.paper_based.rwco_2020), 16

SpeedReductionProblem (in module enoppo.utils.validator), 29

check_float() (in module enoppo.utils.validator), 29

check_int() (in module enoppo.utils.validator), 29

check_penalty_func() (enoppo.engineer.Engineer method), 30

check_solution() (enoppo.engineer.Engineer method), 30

check_str() (in module enoppo.utils.validator), 29

check_tuple_float() (in module enoppo.utils.validator), 29

check_tuple_int() (in module enoppo.utils.validator), 29

check_tuple_str() (in module enoppo.utils.validator), 29

CompressionSpringProblem (class in enoppo.paper_based.pdo_2022), 11

CorrugatedBulkheadProblem (class in enoppo.paper_based.pdo_2022), 11

create_solution() (enoppo.engineer.Engineer method), 30

CSP (in module enoppo.paper_based.moeosma_2023), 7

CSP (in module enoppo.paper_based.pdo_2022), 10

B

BCP (in module enoppo.paper_based.moeosma_2023), 7

BlendingPoolingSeparationProblem (class in enoppo.paper_based.rwco_2020), 16

bounds (enoppo.engineer.Engineer attribute), 30

bounds (enoppo.engineer.Engineer property), 30

D

default_penalty() (enoppo.engineer.Engineer method), 31

differentiable (enoppo.engineer.Engineer attribute), 31

E

Engineer (class in *enoppy.engineer*), 30

enoppy.engineer
module, 30

enoppy.paper_based.ihaoavao_2022
module, 5

enoppy.paper_based.moeosma_2023
module, 7

enoppy.paper_based.pdo_2022
module, 10

enoppy.paper_based.rwco_2020
module, 16

enoppy.utils.encoder
module, 29

enoppy.utils.validator
module, 29

evaluate() (*enoppy.engineer.Engineer* method), 31

evaluate() (*enoppy.paper_based.ihaoavao_2022.CantileverBeamProblem*
method), 5

evaluate() (*enoppy.paper_based.ihaoavao_2022.RollingElementBearingProblem*
method), 5

evaluate() (*enoppy.paper_based.ihaoavao_2022.SpeedReducerProblem*
method), 6

evaluate() (*enoppy.paper_based.ihaoavao_2022.TensionCompressionSpringProblem*
method), 6

evaluate() (*enoppy.paper_based.ihaoavao_2022.WeldedBeamProblem*
method), 6

evaluate() (*enoppy.paper_based.moeosma_2023.BulkCarriersProblem*
method), 7

evaluate() (*enoppy.paper_based.moeosma_2023.CarSideImpactProblem*
method), 7

evaluate() (*enoppy.paper_based.moeosma_2023.HydrostaticThrustBearingProblem*
method), 8

evaluate() (*enoppy.paper_based.moeosma_2023.MultiProductBatchPlantProblem*
method), 8

evaluate() (*enoppy.paper_based.moeosma_2023.SpeedReducerProblem*
method), 9

evaluate() (*enoppy.paper_based.moeosma_2023.SpringProblem*
method), 9

evaluate() (*enoppy.paper_based.moeosma_2023.VibratingPlatformProblem*
method), 10

evaluate() (*enoppy.paper_based.moeosma_2023.WaterResourceManagementProblem*
method), 10

evaluate() (*enoppy.paper_based.pdo_2022.CantileverBeamProblem*
method), 11

evaluate() (*enoppy.paper_based.pdo_2022.CompressionSpringProblem*
method), 11

evaluate() (*enoppy.paper_based.pdo_2022.CorrugatedBulkheadProblem*
method), 11

evaluate() (*enoppy.paper_based.pdo_2022.GearTrainProblem*
method), 12

evaluate() (*enoppy.paper_based.pdo_2022.IBeamProblem*
method), 12

evaluate() (*enoppy.paper_based.pdo_2022.PistonLeverProblem*
method), 13

evaluate() (*enoppy.paper_based.pdo_2022.PressureVesselProblem*
method), 13

evaluate() (*enoppy.paper_based.pdo_2022.ReinforcedConcreteBeamProblem*
method), 14

evaluate() (*enoppy.paper_based.pdo_2022.SpeedReducerProblem*
method), 14

evaluate() (*enoppy.paper_based.pdo_2022.ThreeBarTrussProblem*
method), 15

evaluate() (*enoppy.paper_based.pdo_2022.TubularColumnProblem*
method), 15

evaluate() (*enoppy.paper_based.pdo_2022.WeldedBeamProblem*
method), 15

evaluate() (*enoppy.paper_based.rwco_2020.BlendingPoolingSeparationProblem*
method), 16

evaluate() (*enoppy.paper_based.rwco_2020.HaverlyPoolingProblem*
method), 16

evaluate() (*enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignProblem*
method), 17

evaluate() (*enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignProblem*
method), 17

evaluate() (*enoppy.paper_based.rwco_2020.MultipleDiskClutchBrakeDesignProblem*
method), 18

evaluate() (*enoppy.paper_based.rwco_2020.MultiProductBatchPlantProblem*
method), 18

evaluate() (*enoppy.paper_based.rwco_2020.OptimalDesignIndustrialReactorProblem*
method), 19

evaluate() (*enoppy.paper_based.rwco_2020.OptimalOperationAlkylationProblem*
method), 19

evaluate() (*enoppy.paper_based.rwco_2020.PlanetaryGearTrainDesignProblem*
method), 20

evaluate() (*enoppy.paper_based.rwco_2020.PressureVesselDesignProblem*
method), 20

evaluate() (*enoppy.paper_based.rwco_2020.ProcessDesignProblem*
method), 21

evaluate() (*enoppy.paper_based.rwco_2020.ProcessFlowSheetingProblem*
method), 21

evaluate() (*enoppy.paper_based.rwco_2020.ProcessSynthesis01Problem*
method), 22

evaluate() (*enoppy.paper_based.rwco_2020.ProcessSynthesis02Problem*
method), 22

evaluate() (*enoppy.paper_based.rwco_2020.ProcessSynthesisAndDesignProblem*
method), 23

evaluate() (*enoppy.paper_based.rwco_2020.PropaneIsobutaneNButaneSeparationProblem*
method), 23

evaluate() (*enoppy.paper_based.rwco_2020.ReactorNetworkDesignProblem*
method), 24

evaluate() (*enoppy.paper_based.rwco_2020.RobotGripperProblem*
method), 24

evaluate() (*enoppy.paper_based.rwco_2020.StepConePulleyProblem*
method), 25

evaluate() (*enoppy.paper_based.rwco_2020.TensionCompressionSpringProblem*
method), 25

evaluate() (enoppy.paper_based.rwco_2020.ThreeBarTrussProblem method), 26
 evaluate() (enoppy.paper_based.rwco_2020.TwoReactorProblem method), 26
 evaluate() (enoppy.paper_based.rwco_2020.WeightMiningProblem method), 27
 evaluate() (enoppy.paper_based.rwco_2020.WeldedBeamDesignProblem method), 27
F
 f_global (enoppy.engineer.Engineer attribute), 30
 fit() (enoppy.utils.encoder.LabelEncoder method), 29
 fit_transform() (enoppy.utils.encoder.LabelEncoder method), 29
G
 GearTrainProblem (class in enoppy.paper_based.pdo_2022), 12
 get_cons() (enoppy.engineer.Engineer method), 31
 get_cons() (enoppy.paper_based.ihaoavoa_2022.CantileverBeamProblem method), 5
 get_cons() (enoppy.paper_based.ihaoavoa_2022.RollingElementBearingProblem method), 5
 get_cons() (enoppy.paper_based.ihaoavoa_2022.SpeedReducerProblem method), 6
 get_cons() (enoppy.paper_based.ihaoavoa_2022.TensionCompressionSpringProblem method), 6
 get_cons() (enoppy.paper_based.ihaoavoa_2022.WeldedBeamProblem method), 7
 get_cons() (enoppy.paper_based.moeosma_2023.BulkCarriersProblem method), 7
 get_cons() (enoppy.paper_based.moeosma_2023.CarSideImpactProblem method), 7
 get_cons() (enoppy.paper_based.moeosma_2023.HydrostaticThrustBearingProblem method), 8
 get_cons() (enoppy.paper_based.moeosma_2023.MultiProductBatchPlantProblem method), 8
 get_cons() (enoppy.paper_based.moeosma_2023.SpeedReducerProblem method), 9
 get_cons() (enoppy.paper_based.moeosma_2023.SpringProblem method), 9
 get_cons() (enoppy.paper_based.moeosma_2023.VibratingPlatformProblem method), 10
 get_cons() (enoppy.paper_based.moeosma_2023.WaterResourceManagementProblem method), 10
 get_cons() (enoppy.paper_based.pdo_2022.CantileverBeamProblem method), 11
 get_cons() (enoppy.paper_based.pdo_2022.CompressionSpringProblem method), 11
 get_cons() (enoppy.paper_based.pdo_2022.CorrugatedBulkheadProblem method), 11
 get_cons() (enoppy.paper_based.pdo_2022.GearTrainProblem method), 12
 get_cons() (enoppy.paper_based.pdo_2022.IBeamProblem method), 12
 get_cons() (enoppy.paper_based.pdo_2022.PistonLeverProblem method), 13
 get_cons() (enoppy.paper_based.pdo_2022.PressureVesselProblem method), 13
 get_cons() (enoppy.paper_based.pdo_2022.ReinforcedConcreteBeamProblem method), 14
 get_cons() (enoppy.paper_based.pdo_2022.SpeedReducerProblem method), 14
 get_cons() (enoppy.paper_based.pdo_2022.ThreeBarTrussProblem method), 15
 get_cons() (enoppy.paper_based.pdo_2022.TubularColumnProblem method), 15
 get_cons() (enoppy.paper_based.pdo_2022.WeldedBeamProblem method), 15
 get_cons() (enoppy.paper_based.rwco_2020.BlendingPoolingSeparationProblem method), 16
 get_cons() (enoppy.paper_based.rwco_2020.HaverlyPoolingProblem method), 17
 get_cons() (enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignProblem method), 17
 get_cons() (enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignProblem method), 17
 get_cons() (enoppy.paper_based.rwco_2020.MultipleDiskClutchBrakeDesignProblem method), 18
 get_cons() (enoppy.paper_based.rwco_2020.MultiProductBatchPlantProblem method), 18
 get_cons() (enoppy.paper_based.rwco_2020.OptimalDesignIndustrialReactorProblem method), 19
 get_cons() (enoppy.paper_based.rwco_2020.OptimalOperationAlkylationProblem method), 19
 get_cons() (enoppy.paper_based.rwco_2020.PlanetaryGearTrainDesignProblem method), 20
 get_cons() (enoppy.paper_based.rwco_2020.PressureVesselDesignProblem method), 21
 get_cons() (enoppy.paper_based.rwco_2020.ProcessDesignProblem method), 21
 get_cons() (enoppy.paper_based.rwco_2020.ProcessFlowSheetingProblem method), 22
 get_cons() (enoppy.paper_based.rwco_2020.ProcessSynthesis01Problem method), 22
 get_cons() (enoppy.paper_based.rwco_2020.ProcessSynthesis02Problem method), 22
 get_cons() (enoppy.paper_based.rwco_2020.ProcessSynthesisAndDesignProblem method), 23
 get_cons() (enoppy.paper_based.rwco_2020.PropaneIsobutaneNButaneSeparationProblem method), 23
 get_cons() (enoppy.paper_based.rwco_2020.ReactorNetworkDesignProblem method), 24
 get_cons() (enoppy.paper_based.rwco_2020.RobotGripperProblem method), 24
 get_cons() (enoppy.paper_based.rwco_2020.StepConePulleyProblem method), 25

get_cons() (enoppy.paper_based.rwco_2020.TensionCompressionSpringDesignProblem method), 25
 get_cons() (enoppy.paper_based.rwco_2020.ThreeBarTrussDesignProblem method), 26
 get_cons() (enoppy.paper_based.rwco_2020.TwoReactorProblem method), 26
 get_cons() (enoppy.paper_based.rwco_2020.WeightMinimizationSpeedReducerProblem method), 27
 get_cons() (enoppy.paper_based.rwco_2020.WeldedBeamDesignProblem method), 27
 get_eq_cons() (enoppy.engineer.Engineer method), 31
 get_eq_cons() (enoppy.paper_based.rwco_2020.BlendingProcessDesignProblem method), 16
 get_eq_cons() (enoppy.paper_based.rwco_2020.HaverlyPoolingProblem method), 17
 get_eq_cons() (enoppy.paper_based.rwco_2020.HeatExchangerDesignProblem method), 17
 get_eq_cons() (enoppy.paper_based.rwco_2020.HeatExchangerDesignProblem method), 17
 get_eq_cons() (enoppy.paper_based.rwco_2020.PlanetaryGearTrainDesignProblem method), 20
 get_eq_cons() (enoppy.paper_based.rwco_2020.ProcessDesignProblem method), 23
 get_eq_cons() (enoppy.paper_based.rwco_2020.PropaneIsobutaneSeparationProblem method), 23
 get_eq_cons() (enoppy.paper_based.rwco_2020.ReactorNetworkDesignProblem method), 24
 get_eq_cons() (enoppy.paper_based.rwco_2020.RobotGripperProblem method), 24
 get_eq_cons() (enoppy.paper_based.rwco_2020.StepConePulleyProblem method), 25
 get_eq_cons() (enoppy.paper_based.rwco_2020.TwoReactorProblem method), 26
 get_ineq_cons() (enoppy.engineer.Engineer method), 31
 get_ineq_cons() (enoppy.paper_based.rwco_2020.HaverlyPoolingProblem method), 17
 get_ineq_cons() (enoppy.paper_based.rwco_2020.MultipleDiskClutchBrakeDesignProblem method), 18
 get_ineq_cons() (enoppy.paper_based.rwco_2020.MultiProductBatchPlantProblem method), 18
 get_ineq_cons() (enoppy.paper_based.rwco_2020.OptimalDesignInsulationRefrigerationSystemProblem method), 19
 get_ineq_cons() (enoppy.paper_based.rwco_2020.OptimalOperationAlloyUnitProblem method), 19
 get_ineq_cons() (enoppy.paper_based.rwco_2020.PlanetaryGearTrainDesignProblem method), 20
 get_ineq_cons() (enoppy.paper_based.rwco_2020.PressureVesselDesignProblem method), 21
 get_ineq_cons() (enoppy.paper_based.rwco_2020.ProcessDesignProblem method), 21
 get_ineq_cons() (enoppy.paper_based.rwco_2020.ProcessFlowSheetingProblem method), 22
 get_ineq_cons() (enoppy.paper_based.rwco_2020.ProcessSynthesisProblem method), 22
 get_obj_cons() (enoppy.engineer.Engineer method), 31
 get_obj_cons() (enoppy.paper_based.ihaoavoa_2022.CantileverBeamProblem method), 6
 get_obj_cons() (enoppy.paper_based.ihaoavoa_2022.RollingElementBearingProblem method), 6
 get_obj_cons() (enoppy.paper_based.ihaoavoa_2022.SpeedReducerProblem method), 6
 get_obj_cons() (enoppy.paper_based.ihaoavoa_2022.TensionCompressionSpringProblem method), 6
 get_obj_cons() (enoppy.paper_based.ihaoavoa_2022.WeldedBeamProblem method), 7
 get_obj_cons() (enoppy.paper_based.moeosma_2023.BulkCarriersProblem method), 7
 get_obj_cons() (enoppy.paper_based.moeosma_2023.CarSideImpactProblem method), 8
 get_obj_cons() (enoppy.paper_based.moeosma_2023.HydrostaticThrustBearingProblem method), 8
 get_obj_cons() (enoppy.paper_based.moeosma_2023.MultiProductBatchPlantProblem method), 8
 get_obj_cons() (enoppy.paper_based.moeosma_2023.SpeedReducerProblem method), 8
 get_obj_cons() (enoppy.paper_based.moeosma_2023.SpringProblem method), 9
 get_obj_cons() (enoppy.paper_based.moeosma_2023.VibratingPlatformProblem method), 9
 get_obj_cons() (enoppy.paper_based.moeosma_2023.WaterResourceManagementProblem method), 9
 get_obj_cons() (enoppy.paper_based.pdo_2022.CantileverBeamProblem method), 11
 get_obj_cons() (enoppy.paper_based.pdo_2022.CompressionSpringProblem method), 11
 get_obj_cons() (enoppy.paper_based.pdo_2022.CorrugatedBulkheadProblem method), 11

[get_objs\(\) \(enoppy.paper_based.pdo_2022.GearTrainProblem method\), 12](#)
[get_objs\(\) \(enoppy.paper_based.pdo_2022.IBeamProblem method\), 12](#)
[get_objs\(\) \(enoppy.paper_based.pdo_2022.PistonLeverProblem method\), 13](#)
[get_objs\(\) \(enoppy.paper_based.pdo_2022.PressureVesselDesignProblem method\), 13](#)
[get_objs\(\) \(enoppy.paper_based.pdo_2022.ReinforcedConcreteProblem method\), 14](#)
[get_objs\(\) \(enoppy.paper_based.pdo_2022.SpeedReducerProblem method\), 14](#)
[get_objs\(\) \(enoppy.paper_based.pdo_2022.ThreeBarTrussProblem method\), 15](#)
[get_objs\(\) \(enoppy.paper_based.pdo_2022.TubularColumnProblem method\), 15](#)
[get_objs\(\) \(enoppy.paper_based.pdo_2022.WeldedBeamDesignProblem method\), 16](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.BlendingPoolingSeparationProblem method\), 16](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.HaverlyPoolingProblem method\), 17](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignCase1Problem method\), 17](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignCase2Problem method\), 17](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.MultipleDiskClutchBrakeDesignProblem method\), 18](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.MultiProductBatchPlantProblem method\), 18](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.OptimalDesignIndustrialRefrigerationSystemProblem method\), 19](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.OptimalOperationAlkylationUnitProblem method\), 19](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.PlanetaryGearTrainDesignOptimizationProblem method\), 20](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.PressureVesselDesignProblem method\), 21](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.ProcessDesignProblem method\), 21](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.ProcessFlowSheetingProblem method\), 22](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.ProcessSynthesis01Problem method\), 22](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.ProcessSynthesis02Problem method\), 23](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.ProcessSynthesisAndDesignProblem method\), 23](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.PropaneIsobutaneNButaneNonsharpSeparationProblem method\), 23](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.ReactorNetworkDesignProblem method\), 24](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.RobotGripperProblem method\), 24](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.StepConePulleyProblem method\), 25](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.TensionCompressionSpringProblem method\), 25](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.ThreeBarTrussDesignProblem method\), 26](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.TwoReactorProblem method\), 26](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.WeightMinimizationSpeedReducerProblem method\), 27](#)
[get_objs\(\) \(enoppy.paper_based.rwco_2020.WeldedBeamDesignProblem method\), 27](#)
[get_obj_classes\(\) \(enoppy.engineer.Engineer method\), 31](#)
[GTD \(in module enoppy.paper_based.pdo_2022\), 12](#)

H

[HaverlyPoolingProblem \(class in enoppy.paper_based.rwco_2020\), 16](#)
[HeatExchangerNetworkDesignCase1Problem \(class in enoppy.paper_based.rwco_2020\), 17](#)
[HeatExchangerNetworkDesignCase2Problem \(class in enoppy.paper_based.rwco_2020\), 17](#)
[HMDCP \(in module enoppy.paper_based.rwco_2020\), 16](#)
[HMDCP2 \(in module enoppy.paper_based.rwco_2020\), 16](#)
[HMDCP3 \(in module enoppy.paper_based.rwco_2020\), 16](#)
[HTBP \(in module enoppy.paper_based.moesma_2023\), 8](#)
[HydrostaticThrustBearingProblem \(class in enoppy.paper_based.moesma_2023\), 8](#)

I

[IBD \(in module enoppy.paper_based.pdo_2022\), 12](#)
[IBeamProblem \(class in enoppy.paper_based.pdo_2022\), 12](#)
[inverse_transform\(\) \(enoppy.utils.encoder.LabelEncoder method\), 29](#)
[is_in_bound\(\) \(in module enoppy.utils.validator\), 29](#)
[is_str_in_list\(\) \(in module enoppy.utils.validator\), 29](#)

L

[LabelEncoder \(class in enoppy.utils.encoder\), 29](#)
[lb \(enoppy.engineer.Engineer attribute\), 30](#)
[lb \(enoppy.engineer.Engineer property\), 31](#)
[linear \(enoppy.engineer.Engineer attribute\), 31](#)

M

[MDCBDP \(in module enoppy.paper_based.rwco_2020\), 18](#)
[module enoppy.engineer, 30](#)
[enoppy.paper_based.ihaoavoa_2022, 5](#)

enoppy.paper_based.moeosma_2023, 7
 enoppy.paper_based.pdo_2022, 10
 enoppy.paper_based.rwco_2020, 16
 enoppy.utils.encoder, 29
 enoppy.utils.validator, 29
 MPBP (in module enoppy.paper_based.rwco_2020), 18
 MPBPP (in module enoppy.paper_based.moeosma_2023), 8
 MultipleDiskClutchBrakeDesignProblem (class in enoppy.paper_based.rwco_2020), 18
 MultiProductBatchPlantProblem (class in enoppy.paper_based.moeosma_2023), 8
 MultiProductBatchPlantProblem (class in enoppy.paper_based.rwco_2020), 18

N

n_cons (enoppy.engineer.Engineer property), 31
 n_dims (enoppy.engineer.Engineer attribute), 30
 n_dims (enoppy.engineer.Engineer property), 31
 n_eq_cons (enoppy.engineer.Engineer property), 31
 n_fe (enoppy.engineer.Engineer attribute), 30
 n_ineq_cons (enoppy.engineer.Engineer property), 31
 n_objs (enoppy.engineer.Engineer property), 31
 name (enoppy.engineer.Engineer attribute), 31
 name (enoppy.paper_based.ihaoavao_2022.CantileverBeamProblem attribute), 5
 name (enoppy.paper_based.ihaoavao_2022.RollingElementBearingProblem attribute), 5
 name (enoppy.paper_based.ihaoavao_2022.SpeedReducerProblem attribute), 6
 name (enoppy.paper_based.ihaoavao_2022.TensionCompressionSpringProblem attribute), 6
 name (enoppy.paper_based.ihaoavao_2022.WeldedBeamProblem attribute), 7
 name (enoppy.paper_based.moeosma_2023.BulkCarriersProblem attribute), 7
 name (enoppy.paper_based.moeosma_2023.CarSideImpactProblem attribute), 8
 name (enoppy.paper_based.moeosma_2023.HydrostaticThrustBearingProblem attribute), 8
 name (enoppy.paper_based.moeosma_2023.MultiProductBatchPlantProblem attribute), 8
 name (enoppy.paper_based.moeosma_2023.SpeedReducerProblem attribute), 9
 name (enoppy.paper_based.moeosma_2023.SpringProblem attribute), 9
 name (enoppy.paper_based.moeosma_2023.VibratingPlatformProblem attribute), 10
 name (enoppy.paper_based.moeosma_2023.WaterResourceManagementProblem attribute), 10
 name (enoppy.paper_based.pdo_2022.CantileverBeamProblem attribute), 11
 name (enoppy.paper_based.pdo_2022.CompressionSpringProblem attribute), 11
 name (enoppy.paper_based.pdo_2022.CorrugatedBulkheadProblem attribute), 12
 name (enoppy.paper_based.pdo_2022.GearTrainProblem attribute), 12
 name (enoppy.paper_based.pdo_2022.IBeamProblem attribute), 12
 name (enoppy.paper_based.pdo_2022.PistonLeverProblem attribute), 13
 name (enoppy.paper_based.pdo_2022.PressureVesselProblem attribute), 13
 name (enoppy.paper_based.pdo_2022.ReinforcedConcreteBeamProblem attribute), 14
 name (enoppy.paper_based.pdo_2022.SpeedReducerProblem attribute), 14
 name (enoppy.paper_based.pdo_2022.ThreeBarTrussProblem attribute), 15
 name (enoppy.paper_based.pdo_2022.TubularColumnProblem attribute), 15
 name (enoppy.paper_based.pdo_2022.WeldedBeamProblem attribute), 16
 name (enoppy.paper_based.rwco_2020.BlendingPoolingSeparationProblem attribute), 16
 name (enoppy.paper_based.rwco_2020.HaverlyPoolingProblem attribute), 17
 name (enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignCase attribute), 17
 name (enoppy.paper_based.rwco_2020.HeatExchangerNetworkDesignCase attribute), 17
 name (enoppy.paper_based.rwco_2020.MultipleDiskClutchBrakeDesignProblem attribute), 18
 name (enoppy.paper_based.rwco_2020.MultiProductBatchPlantProblem attribute), 18
 name (enoppy.paper_based.rwco_2020.OptimalDesignIndustrialRefrigerationProblem attribute), 19
 name (enoppy.paper_based.rwco_2020.OptimalOperationAlkylolationUnitProblem attribute), 19
 name (enoppy.paper_based.rwco_2020.PlanetaryGearTrainDesignOptimizationProblem attribute), 20
 name (enoppy.paper_based.rwco_2020.PressureVesselDesignProblem attribute), 21
 name (enoppy.paper_based.rwco_2020.ProcessDesignProblem attribute), 21
 name (enoppy.paper_based.rwco_2020.ProcessFlowSheetingProblem attribute), 22
 name (enoppy.paper_based.rwco_2020.ProcessSynthesis01Problem attribute), 22
 name (enoppy.paper_based.rwco_2020.ProcessSynthesis02Problem attribute), 23
 name (enoppy.paper_based.rwco_2020.ProcessSynthesisAndDesignProblem attribute), 23
 name (enoppy.paper_based.rwco_2020.PropaneIsobutaneNButaneNonshareProblem attribute), 24
 name (enoppy.paper_based.rwco_2020.ReactorNetworkDesignProblem attribute), 24

name (enoppy.paper_based.rwco_2020.RobotGripperProblem (class in module enoppy.paper_based.rwco_2020), 24
 attribute), 24
 name (enoppy.paper_based.rwco_2020.StepConePulleyProblem (class in module enoppy.paper_based.pdo_2022), 13
 attribute), 25
 name (enoppy.paper_based.rwco_2020.TensionCompressionPlanetaryGearTrainDesignOptimizationProblem (class in module enoppy.paper_based.rwco_2020), 20
 attribute), 25
 name (enoppy.paper_based.rwco_2020.ThreeBarTrussDesignProblem (class in module enoppy.paper_based.pdo_2022), 12
 attribute), 26
 name (enoppy.paper_based.rwco_2020.TwoReactorProblem (class in module enoppy.paper_based.rwco_2020), 20
 attribute), 26
 name (enoppy.paper_based.rwco_2020.WeightMinimizationSpeedReducerProblem (class in module enoppy.paper_based.pdo_2022), 13
 attribute), 27
 name (enoppy.paper_based.rwco_2020.WeldedBeamDesignProblem (class in module enoppy.paper_based.rwco_2020), 21
 attribute), 27
O
 OBJ11C (in module enoppy.paper_based.rwco_2020), 18
 ODIRSP (in module enoppy.paper_based.rwco_2020), 18
 OOAUP (in module enoppy.paper_based.rwco_2020), 19
 OptimalDesignIndustrialRefrigerationSystemProblem (class in module enoppy.paper_based.rwco_2020), 19
 OptimalOperationAlkylationUnitProblem (class in module enoppy.paper_based.rwco_2020), 19
P
 p1 (in module enoppy.paper_based.rwco_2020), 27
 p10 (in module enoppy.paper_based.rwco_2020), 27
 p11 (in module enoppy.paper_based.rwco_2020), 27
 p12 (in module enoppy.paper_based.rwco_2020), 27
 p13 (in module enoppy.paper_based.rwco_2020), 27
 p14 (in module enoppy.paper_based.rwco_2020), 27
 p15 (in module enoppy.paper_based.rwco_2020), 28
 p16 (in module enoppy.paper_based.rwco_2020), 28
 p17 (in module enoppy.paper_based.rwco_2020), 28
 p18 (in module enoppy.paper_based.rwco_2020), 28
 p19 (in module enoppy.paper_based.rwco_2020), 28
 p2 (in module enoppy.paper_based.rwco_2020), 28
 p20 (in module enoppy.paper_based.rwco_2020), 28
 p21 (in module enoppy.paper_based.rwco_2020), 28
 p22 (in module enoppy.paper_based.rwco_2020), 28
 p23 (in module enoppy.paper_based.rwco_2020), 28
 p3 (in module enoppy.paper_based.rwco_2020), 28
 p4 (in module enoppy.paper_based.rwco_2020), 28
 p5 (in module enoppy.paper_based.rwco_2020), 28
 p6 (in module enoppy.paper_based.rwco_2020), 28
 p7 (in module enoppy.paper_based.rwco_2020), 28
 p8 (in module enoppy.paper_based.rwco_2020), 28
 p9 (in module enoppy.paper_based.rwco_2020), 28
 parametric (enoppy.engineer.Engineer attribute), 31
 PDP (in module enoppy.paper_based.rwco_2020), 19
 PFSP (in module enoppy.paper_based.rwco_2020), 19
 PGTDOP (in module enoppy.paper_based.rwco_2020), 19
 PINNSP (in module enoppy.paper_based.rwco_2020), 20
 PistonLeverProblem (class in module enoppy.paper_based.pdo_2022), 13
 PlanetaryGearTrainDesignOptimizationProblem (class in module enoppy.paper_based.rwco_2020), 20
 PressureVesselDesignProblem (class in module enoppy.paper_based.rwco_2020), 20
 PressureVesselProblem (class in module enoppy.paper_based.pdo_2022), 13
 ProcessDesignProblem (class in module enoppy.paper_based.pdo_2022), 13
 ProcessFlowSheetingProblem (class in module enoppy.paper_based.rwco_2020), 21
 ProcessSynthesis01Problem (class in module enoppy.paper_based.rwco_2020), 22
 ProcessSynthesis02Problem (class in module enoppy.paper_based.rwco_2020), 22
 ProcessSynthesisAndDesignProblem (class in module enoppy.paper_based.rwco_2020), 23
 PropaneIsobutaneNButaneNonsharpSeparationProblem (class in module enoppy.paper_based.rwco_2020), 23
 PS01P (in module enoppy.paper_based.rwco_2020), 20
 PS02P (in module enoppy.paper_based.rwco_2020), 20
 PSADP (in module enoppy.paper_based.rwco_2020), 20
 PVDP (in module enoppy.paper_based.rwco_2020), 20
 PVP (in module enoppy.paper_based.pdo_2022), 13
R
 RCB (in module enoppy.paper_based.pdo_2022), 13
 ReactorNetworkDesignProblem (class in module enoppy.paper_based.rwco_2020), 24
 REBP (in module enoppy.paper_based.ihaoavao_2022), 5
 ReinforcedConcreteBeamProblem (class in module enoppy.paper_based.pdo_2022), 13
 RNDP (in module enoppy.paper_based.rwco_2020), 24
 RobotGripperProblem (class in module enoppy.paper_based.rwco_2020), 24
 RollingElementBearingProblem (class in module enoppy.paper_based.ihaoavao_2022), 5
S
 SCPP (in module enoppy.paper_based.rwco_2020), 24
 SP (in module enoppy.paper_based.moeosma_2023), 8
 SpeedReducerProblem (class in module enoppy.paper_based.ihaoavao_2022), 6
 SpeedReducerProblem (class in module enoppy.paper_based.moeosma_2023), 9
 SpeedReducerProblem (class in module enoppy.paper_based.pdo_2022), 14
 SpringProblem (class in module enoppy.paper_based.moeosma_2023), 9
 SRD (in module enoppy.paper_based.pdo_2022), 14

SRP (in module *enoppy.paper_based.ihaoavao_2022*), 5
 SRP (in module *enoppy.paper_based.moeosma_2023*), 9
 StepConePulleyProblem (class in *enoppy.paper_based.rwco_2020*), 25

X

x_global (*enoppy.engineer.Engineer* attribute), 30

T

TBTD (in module *enoppy.paper_based.pdo_2022*), 14
 TBTD (in module *enoppy.paper_based.rwco_2020*), 25
 TCD (in module *enoppy.paper_based.pdo_2022*), 14
 TCSP (in module *enoppy.paper_based.ihaoavao_2022*), 6
 TensionCompressionSpringDesignProblem (class in *enoppy.paper_based.rwco_2020*), 25
 TensionCompressionSpringProblem (class in *enoppy.paper_based.ihaoavao_2022*), 6
 ThreeBarTrussDesignProblem (class in *enoppy.paper_based.rwco_2020*), 25
 ThreeBarTrussProblem (class in *enoppy.paper_based.pdo_2022*), 14
 transform() (*enoppy.utils.encoder.LabelEncoder* method), 29
 TRP (in module *enoppy.paper_based.rwco_2020*), 25
 TubularColumnProblem (class in *enoppy.paper_based.pdo_2022*), 15
 TwoReactorProblem (class in *enoppy.paper_based.rwco_2020*), 26

U

ub (*enoppy.engineer.Engineer* attribute), 30
 ub (*enoppy.engineer.Engineer* property), 31

V

VibratingPlatformProblem (class in *enoppy.paper_based.moeosma_2023*), 9
 VPP (in module *enoppy.paper_based.moeosma_2023*), 9

W

WaterResourceManagementProblem (class in *enoppy.paper_based.moeosma_2023*), 10
 WBDP (in module *enoppy.paper_based.rwco_2020*), 26
 WBP (in module *enoppy.paper_based.ihaoavao_2022*), 6
 WBP (in module *enoppy.paper_based.pdo_2022*), 15
 WeightMinimizationSpeedReducerProblem (class in *enoppy.paper_based.rwco_2020*), 26
 WeldedBeamDesignProblem (class in *enoppy.paper_based.rwco_2020*), 27
 WeldedBeamProblem (class in *enoppy.paper_based.ihaoavao_2022*), 6
 WeldedBeamProblem (class in *enoppy.paper_based.pdo_2022*), 15
 WMSRP (in module *enoppy.paper_based.rwco_2020*), 26
 WRMP (in module *enoppy.paper_based.moeosma_2023*), 10