

---

# hetrob Documentation

*Release 0.1.0*

**Jonas Teufel**

**Nov 24, 2020**



## CONTENTS:

<b>1</b>	<b>HetRob - Coordination of Heterogeneous Robotic Teams</b>	<b>1</b>
1.1	Installation . . . . .	1
1.2	Basic Usage . . . . .	1
1.3	Credits . . . . .	1
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Stable release . . . . .	3
2.2	From sources . . . . .	3
<b>3</b>	<b>Getting started</b>	<b>5</b>
3.1	Example scripts . . . . .	5
<b>4</b>	<b>Contributing</b>	<b>7</b>
4.1	Types of Contributions . . . . .	7
4.2	Get Started! . . . . .	8
4.3	Pull Request Guidelines . . . . .	9
4.4	Tips . . . . .	9
4.5	Deploying . . . . .	9
<b>5</b>	<b>Credits</b>	<b>11</b>
5.1	Development Lead . . . . .	11
5.2	Contributors . . . . .	11
<b>6</b>	<b>History</b>	<b>13</b>
6.1	0.1.0 (2020-06-30) . . . . .	13
6.2	0.2.0 (2020-10-02) . . . . .	13
6.3	0.3.0 (2020-11-23) . . . . .	13
6.4	0.4.0 (20202-11-24) . . . . .	13
<b>7</b>	<b>Indices and tables</b>	<b>15</b>



# HETROB - COORDINATION OF HETEROGENEOUS ROBOTIC TEAMS

A python package for the optimization of mission plans for heterogeneous, cooperating robotic teams using genetic algorithms.

- Free software: MIT license
- Documentation: <https://hetrob.readthedocs.io>.

## 1.1 Installation

The package is best installed using the Python package manager PIP.

```
$ pip3 install --user hetrob
```

## 1.2 Basic Usage

For a more detailed description of the usage refer to the [Documentation](#).

## 1.3 Credits

- [Cookiecutter](#): A python boilerplate project templating CLI tool
- [cookiecutter-pypackage](#): A cookiecutter template, which provides the basic boilerplate for creating the folder structure of a publishable python package
- [deap](#): Short for “distributed evolutionary algorithms with python”



## INSTALLATION

### 2.1 Stable release

To install hetrob, run this command in your terminal:

```
$ pip install hetrob
```

This is the preferred method to install hetrob, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

### 2.2 From sources

The sources for hetrob can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/the16thpythonist/hetrob
```

Or download the [tarball](#):

```
$ curl -OJL https://github.com/the16thpythonist/hetrob/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```





## GETTING STARTED

The best way of getting started with the hetrob documentation is to first read the all the example python modules in the intended order. After that, the additional chapters of this documentation can be used to potentially read up on topics which were not mentioned within these examples.

### 3.1 Example scripts

The following listing provides an overview of all the example files, which will serve as a ‘guided tour’ through the functionality of the hetrob library. Each script contains example code as well as extensive comments which explain, as well as the additional underlying principles for most of the operations.

This list of examples is sorted in the order in which the information will make the most sense.

- [basic\\_vrp.py](#). This file contains the most stripped down version of steps, which are required to essentially solve a vehicle routing problem using a genetic algorithm. The vehicle routing problem which is considered in this file is one of the easiest imaginable variants, which simply considers some spatially distributed task locations and a number of available vehicles. There are no additional constraints and every other aspect such as task duration and vehicle speed are entirely constant and homogeneous. This file will not touch on any of the details, but rather provide an overview of steps required to operate the library.
- [hsvrsp.py](#). This script contains a more realistic VRP variant abbreviated “HSVRSP”. This variant included additional constraints such as heterogeneity, cooperation and precedence, which make it much harder to solve. This example still follows the same overall pattern as the previous one, but goes into a little bit more details for each step. For example it introduces the possibility of loading problem instances from JSON files, modifying the solution representations objective function, changing the selection operator and the specifics of solution visualization.
- [custom\\_genotype.py](#). This script basically uses the same example VRP problem as the first example did. Although this file introduces how to write a custom genotype representation. This custom genotype can be created by subclassing the abstract base class *AbstractGenotype*. It is important for the design of this custom class that it can properly interface with the Phenotype and Problem classes which will be used in combination with it, but other than that the newly created Genotype class can simply be plugged into the genetic solution representation and all other parts will work the same way they did before.



## CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

### 4.1 Types of Contributions

#### 4.1.1 Report Bugs

Report bugs at <https://github.com/the16thpythonist/hetrob/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

#### 4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

#### 4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

#### 4.1.4 Write Documentation

hetrob could always use more documentation, whether as part of the official hetrob docs, in docstrings, or even on the web in blog posts, articles, and such.

### 4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/the16thpythonist/hetrob/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 4.2 Get Started!

Ready to contribute? Here's how to set up *hetrob* for local development.

1. Fork the *hetrob* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/hetrob.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv hetrob
$ cd hetrob/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 hetrob tests
$ python setup.py test or pytest
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

## 4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.5, 3.6, 3.7 and 3.8, and for PyPy. Check [https://travis-ci.com/the16thpythonist/hetrob/pull\\_requests](https://travis-ci.com/the16thpythonist/hetrob/pull_requests) and make sure that the tests pass for all supported Python versions.

## 4.4 Tips

To run a subset of tests:

```
$ pytest tests.test_hetrob
```

## 4.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.



CREDITS

## 5.1 Development Lead

- Jonas Teufel <[jonseb1998@gmail.com](mailto:jonseb1998@gmail.com)>

## 5.2 Contributors

None yet. Why not be the first?





## HISTORY

### 6.1 0.1.0 (2020-06-30)

- First release on PyPI.

### 6.2 0.2.0 (2020-10-02)

- Added all the stuff from the implementation phase of the bachelor thesis
- **Documentation**
  - Added very basic information about the optimization of a base VRP with a simple GA
  - Added a chapter for advanced topics
  - Added a chapter for a command line interface

### 6.3 0.3.0 (2020-11-23)

- Changed the documentation theme to “read the docs”
- Added the “getting started page” to the documentation
- Added the example “basic\_vrp.py”
- Added the example “hsvrsp.py”

### 6.4 0.4.0 (2020-11-24)

- Added the example “custom\_genotype.py”



## INDICES AND TABLES

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