

jvkersch / pyconcorde

Python wrapper around the Concorde TSP solver

47 commits

2 branches

0 releases

2 contributors

BSD-3-Clause

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

jvkersch

Merge pull request #18 from chq-matteo/patch-1

Latest commit c736389 on 6 Dec 2018

concorde	Merge remote-tracking branch 'origin/master' into maint/rename-pyconc...	11 months ago
examples	Add requirements.txt for examples	4 months ago
tools	MAINT: Remove build shell script.	11 months ago
.gitignore	DEV: Download/build concorde from setup.py.	11 months ago
.travis.yml	BUG: Fixes for Python 2.7	11 months ago
COPYING	MAINT: Rename module	11 months ago
MANIFEST.in	DEV: fixes to build an actual working wheel.	11 months ago
README.md	DOC: Note about other solvers.	11 months ago
requirements.txt	REP: Travis setup	2 years ago
setup.py	DEV: fixes to build an actual working wheel.	11 months ago

README.md

PyConcorde


build

passing

What is it?

PyConcorde is a Python wrapper around the [Concorde TSP solver](#).

PyConcorde allows you to compute solutions to the Traveling Salesman Problem with just a few lines of Python code. It uses the state-of-the-art Concorde solver and provides a convenient Python layer around it.



Note: until commit e065497 (pre version 0.1) PyConcorde was called PyTSP. It was renamed to emphasize the central role of the underlying Concorde solver.

How do I install it?

https://github.com/jvkersch/pyconcorde

1/3


PyConcorde runs under Python 2.7 and 3.5 and up. It needs the [Concorde TSP solver](#) and [QSOpt linear programming library](#). Further instructions on building/downloading those can be found below.

To build PyConcorde, clone the repository:

```
git clone https://github.com/jvkersch/pyconcorde
cd pyconcorde
```

Then run

```
pip install -e .
```

This will download and build Concorde (and its dependency QSOpt) and then build PyConcorde. While this may take a few minutes, downloading ncorde only happens the first time the install script is run (unless you remove the `data` directory).

If you already have Concorde and/or QSOpt installed (or an equivalent linear solver), you can use those instead. Simply set the environment variable `CONCORDE_DIR` (or `QSOPT_DIR`) to point to the folder where you installed Concorde (or QSOpt).

Detailed instructions to build Concorde can be found at [this web page](#).

What can I do with it?

PyConcorde is a very light-weight library. The main entry point is the `TSPSolver` class. Here we use it to read in the Berlin52 dataset, a dataset of 52 locations in Berlin (part of the TSPLib test data).

```
>>> from concorde.tsp import TSPSolver
>>> from concorde.tests.data_utils import get_dataset_path
>>> fname = get_dataset_path("berlin52")
>>> solver = TSPSolver.from_tspfile(fname)
Problem Name: berlin52
Problem Type: TSP
52 locations in Berlin (Groetschel)
Number of Nodes: 52
Rounded Euclidean Norm (CC_EUCLIDEAN)
```

As you can see above, PyConcorde (or rather, Concorde) is somewhat chatty and will print various message to the standard output. Now that we have a solver instance, let's compute a solution. On my machine this is almost instantaneous.

```
>>> solution = solver.solve()
(... output snipped for brevity ...)
```

Again, Concorde will display a variety of messages while it's running. The end result is a `ComputedTour` object called `solution` with information about the tour that we just computed:

```
>>> solution.found_tour
True
>>> solution.optimal_value
7542.0
>>> solution.tour
array([ 0, 48, 31, 44, 18, 40, 7, 8, 9, 42, 32, 50, 10, 51, 13, 12, 46,
       25, 26, 27, 11, 24, 3, 5, 14, 4, 23, 47, 37, 36, 39, 38, 35, 34,
       33, 43, 45, 15, 28, 49, 19, 22, 29, 1, 6, 41, 20, 16, 2, 17, 30,
       21], dtype=int32)
```

Other TSP solvers for Python

If you are looking for a pure Python TSP implementation, check out [tsp-solver](#). It is not competitive with Concorde in terms of speed or memory usage, but has the advantage of being written in pure Python, which makes it more readily inspectable.

If you have another TSP algorithm that can be called from within Python, and you want to have it added here, please open an issue.

Technical Notes

PyConcorde needs Concorde and QSOpt. Downloading and building these packages should happen automatically on Linux/Mac OS, but please file an issue if you experience any trouble during this step.

Note that Windows is currently unsupported. If you get the library to work on Windows, please open a ticket to describe any tweaks (or better yet, a Pull Request).

License

PyConcorde is licensed under the [Modified BSD license](#). Note that Concorde and QSOpt are released under different licenses, and that PyConcorde does not include any code from these packages.