03_looking_at_results

October 9, 2023

[16]:
```
from algbench import Benchmark, read_as_pandas, describe
import matplotlib.pyplot as plt
import seaborn as sns
import random as rd
```

[17]:
```
benchmark = Benchmark("results")
```

```
---------------------------------------------------------------------------
LookupError                               Traceback (most recent call last)
Cell In[17], line 1
----> 1 benchmark = Benchmark("results")

File ~/Documents/Arbeit/covering_points_by_lines/geometric-covering/venv/lib/
 ↪python3.11/site-packages/algbench/benchmark.py:79, in Benchmark.__init__(self␣
 ↪path, save_output, hide_output, save_output_with_time)
     43 def __init__(
     44     self,
     45     path: str,
   (…)
     48     save_output_with_time: bool = True,
     49 ) -> None:
     50     """
     51     Just specify the path of where to put the
     52     database and everything else happens magically.
   (…)
     77         set to false.
     78     """
---> 79     self._db = BenchmarkDb(path)
     80     self._save_output = save_output
     81     self._hide_output = hide_output

File ~/Documents/Arbeit/covering_points_by_lines/geometric-covering/venv/lib/
 ↪python3.11/site-packages/algbench/benchmark_db.py:16, in BenchmarkDb.
 ↪__init__(self, path)
     14 self.path = path
     15 self._create_or_check_info_file()
---> 16 self._arg_fingerprints =␣
 ↪NfsJsonSet(os.path.join(path, "arg_fingerprints"))
```

```
     17 self._data = NfsJsonList(os.path.join(path, "results"))
     18 self._env_data = NfsJsonDict(os.path.join(path, "env_info"))

File ~/Documents/Arbeit/covering_points_by_lines/geometric-covering/venv/lib/
 ↪python3.11/site-packages/algbench/db/nfs_json_set.py:11, in NfsJsonSet.
 ↪__init__(self, path)
      9 self._db = NfsJsonList(path)
     10 self._values: typing.Set = set()
---> 11 self.load()

File ~/Documents/Arbeit/covering_points_by_lines/geometric-covering/venv/lib/
 ↪python3.11/site-packages/algbench/db/nfs_json_set.py:14, in NfsJsonSet.
 ↪load(self)
     13 def load(self):
---> 14     for row in self._db.load():
     15         self._values.update(row)

File ~/Documents/Arbeit/covering_points_by_lines/geometric-covering/venv/lib/
 ↪python3.11/site-packages/algbench/db/nfs_json_list.py:154, in NfsJsonList.
 ↪load(self)
    153 def load(self) -> typing.List:
--> 154     return list(self)

File ~/Documents/Arbeit/covering_points_by_lines/geometric-covering/venv/lib/
 ↪python3.11/site-packages/algbench/db/nfs_json_list.py:146, in NfsJsonList.
 ↪__iter__(self)
    145 def __iter__(self):
--> 146     for entry in self.iter_compressed():
    147         yield entry
    148     for entry in self.iter_uncompressed():

File ~/Documents/Arbeit/covering_points_by_lines/geometric-covering/venv/lib/
 ↪python3.11/site-packages/algbench/db/nfs_json_list.py:112, in NfsJsonList.
 ↪iter_compressed(self)
    110 compr_path = os.path.join(self.path, "_compressed.zip")
    111 if os.path.exists(compr_path):
--> 112     with ZipFile(compr_path, "r") as z:
    113         for filename in z.filelist:
    114             yield from self._iter_compressed_file(z, filename,␣
 ↪compr_path)

File /opt/homebrew/Cellar/python@3.11/3.11.5/Frameworks/Python.framework/
 ↪Versions/3.11/lib/python3.11/zipfile.py:1302, in ZipFile.__init__(self, file,
 ↪mode, compression, allowZip64, compresslevel, strict_timestamps,␣
 ↪metadata_encoding)
   1300 try:
   1301     if mode == 'r':
-> 1302         self._RealGetContents()
   1303     elif mode in ('w', 'x'):
```

```
   1304         # set the modified flag so central directory gets written
   1305         # even if no files are added to the archive
   1306         self._didModify = True

File /opt/homebrew/Cellar/python@3.11/3.11.5/Frameworks/Python.framework/
 ↪Versions/3.11/lib/python3.11/zipfile.py:1409, in ZipFile._RealGetContents(sel
   1406     filename = filename.decode('utf-8')
   1407 else:
   1408     # Historical ZIP filename encoding
-> 1409     filename = filename.decode(self.metadata_encoding or 'cp437')
   1410 # Create ZipInfo instance to store file information
   1411 x = ZipInfo(filename)

LookupError: unknown encoding: cp437
```

[ ]:
```
for run in benchmark:
    fig, ax = plt.subplots()
    xes, yes = zip(*run["result"]["points"])

    covering_lines = run["result"]["solution"]
    print(run["parameters"])

    for line in covering_lines:
        if len(line) == 1:
            p = run["result"]["points"][line[0]]
            if rd.random() < 0.5:
                second_point = (p[0]+1, p[1])
            else:
                second_point = (p[0], p[1]+1)
            ax.axline(p, second_point, color="red", linestyle="--")
        else:
            ax.axline(run["result"]["points"][line[0]],␣
 ↪run["result"]["points"][line[1]], color="red", linestyle="--")

    ax.scatter(xes, yes, zorder=2, color="black", s=5)
    ax.set_aspect('equal')
    plt.show()
    break
```

[ ]: