Name: Aryan Ritwajeet Jha
ID: 011807182

Project 2 Report: Set Covering Problem

Problem Description

Using various heuristic techniques, solve the Set Covering Problem for the two given test cases: *phase1* and *cap360*, each representing a 'graph' of various poles (facilities) and meters (customers). Report the results as the total number of poles used to cover all meters, and the time taken for the simulation.

Methodology and Parameter Values

cleanupRepeats = Number of iterations (pole additions) a cleanup is run on the entire P'. Cleanup is also always done at least once at the end, if all the meters are covered. Value set to 100 if preprocessing is used or 1 otherwise.

Preprocess_steps = Number of time the sequence of Preprocess1, Preprocess2 and Preprocess3 are run per iteration. Set as 1 typically.

Important Note regarding Preprocessing: In all my preprocessing functions including Preprocess1, Preprocess2 and Preprocess3, the decisions are NOT taken in a sequential manner, rather a batch of poles/meters are identified for the cleanup procedure and they're all subsequently cleaned up afterwards. This is because my 'graph' altering functions such as `addPole!`, `discardPole!`, `removePole!`, `ignoreMeter!` have been found to be more time-consuming, and calling them after every single decision was not allowing me to meet time benchmarks.

Preprocess2_steps = Maximum number of poles discarded in one *sweep* of Preprocess2. Set as 100.

Preprocess3_steps = Maximum number of meters ignored in one *sweep* of Preprocess3. Set as 50.

Preprocess2_check_steps = Maximum number of pairs of poles compared as part of one *sweep* of Preprocess2. Set as 60_000 for greedy and 120_000 for score 2.

Preprocess3_check_steps = Maximum number of pairs of meters compared as part of one *sweep* of Preprocess3. Set as 60_000 for greedy and 120_000 for score 2.

Note regarding Modified Greedy: Score 1 results were typically inferior to Score 2 results, so I've only given Score 2 results for that row.

For cap360 test case with preprocessing, I've also attached snips from the terminal after the script ran.

Result 1 (Without Preprocessing)

phase1:

Scoring Function	#Poles Used	Time Taken [s]
Greedy	24	0.001
Modified Greedy (Score 2)	25	0.001

Name: Aryan Ritwajeet Jha
ID: 011807182

cap360:

Scoring Function	#Poles Used	Time Taken [s]
Greedy	617	2
Modified Greedy (Score 2)	618	2

```
Solved in 0.0013756 s
value=24

Solved in 0.0013964 s
value=25

Solved in 0.0014176 s
value=25
```

```
Solved in 1.8892614 s
value=617

Solved in 1.6786586 s
value=632

Solved in 1.586568 s
value=618
```

Result 2 (With Preprocessing)

Note only for *phase1*: For *phase1* test case, at the time of taking the snapshots, preprocess3 was not implemented (but benchmarks were being achieved anyway).

preprocess_steps was set to 100 (i.e. pp1 and pp2 were run 100 times one after the other each iteration) and preprocess2_step was set to 1 (i.e. for each pp1 and pp2 call, exactly one pole was added or discarded).

phase1:

Scoring Function	#Poles Used	Time Taken [s]
Greedy	24	2
Modified Greedy (Score 2)	23	0.11

```
**********
Simulation Time[s]: 1.9548152
value=24
scoring_function=greedy
testCase=phase1
cleanupRepeats=1
**********
preprocess1_steps=18
preprocess2_steps=93
preprocess3_steps=0
```

```
**********
Simulation Time[s]: 5.0413527
value=23
scoring_function=score2
testCase=phase1
cleanupRepeats=1
**********
preprocess1_steps=11
preprocess2_steps=94
preprocess3_steps=0
```

cap360:

Scoring Function	#Poles Used	Time Taken [s]
Greedy	580	100
Modified Greedy (Score 2)	552	211

```
**********
Simulation Time[s]: 100.433225
value=580
scoring_function=greedy
testCase=cap360
cleanupRepeats=100
**********
preprocess1_steps=151
preprocess2_steps=5967
preprocess3_steps=154
```

```
**********
Simulation Time[s]: 211.2306388
value=552
scoring_function=score2
testCase=cap360
cleanupRepeats=100
**********
preprocess1_steps=18
preprocess2_steps=5987
preprocess3_steps=752
```

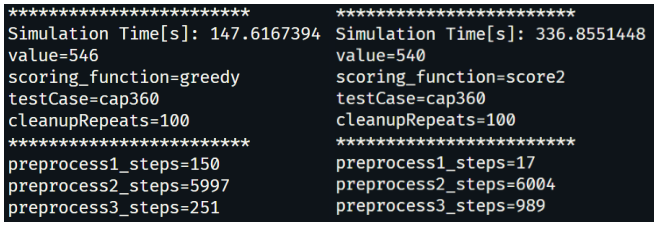
Result 3 (With 'Full' Preprocessing)

Increased Preprocess_check_steps and Preprocess_steps to 120_000, 120 for both Preprocess2 and Preprocess3

cap360:

Scoring Function	#Poles Used	Time Taken [s]
Greedy	546	148
Modified Greedy (Score 2)	540	337

Name: Aryan Ritwajeet Jha
ID: 011807182



Codebase Description

Code can be run via `main.jl` file in the project folder. Functions for Set Covering Heuristics can be found in `./src/setCoveringHeuristics.jl` file.

References

1. Prof. Bala Krishnamoorthy's course notes as part of MATH 567 Integer and Combinatorial Optimization taught at Washington State University, Spring 2025. Problem Statement and Lecture notes retrieved Integer Optimization: Lecture Notes and videos. (2025, April 24). Retrieved from https://bala-krishnamoorthy.github.io/FilesMath567/S25/LecNotes/index.html.