Supplemental File

Searching k-Optimal Goals for an Orienteering Problem on a Specialized Graph with Budget Constraints

A novel non-randomized anytime orienteering algorithm for finding k-optimal goals

THE UNIVERSITY OF MELBOURNE

Abhinav Sharma

Advait Deshpande

Yanming Wang

Xinyi Xu

Prashan Madumal

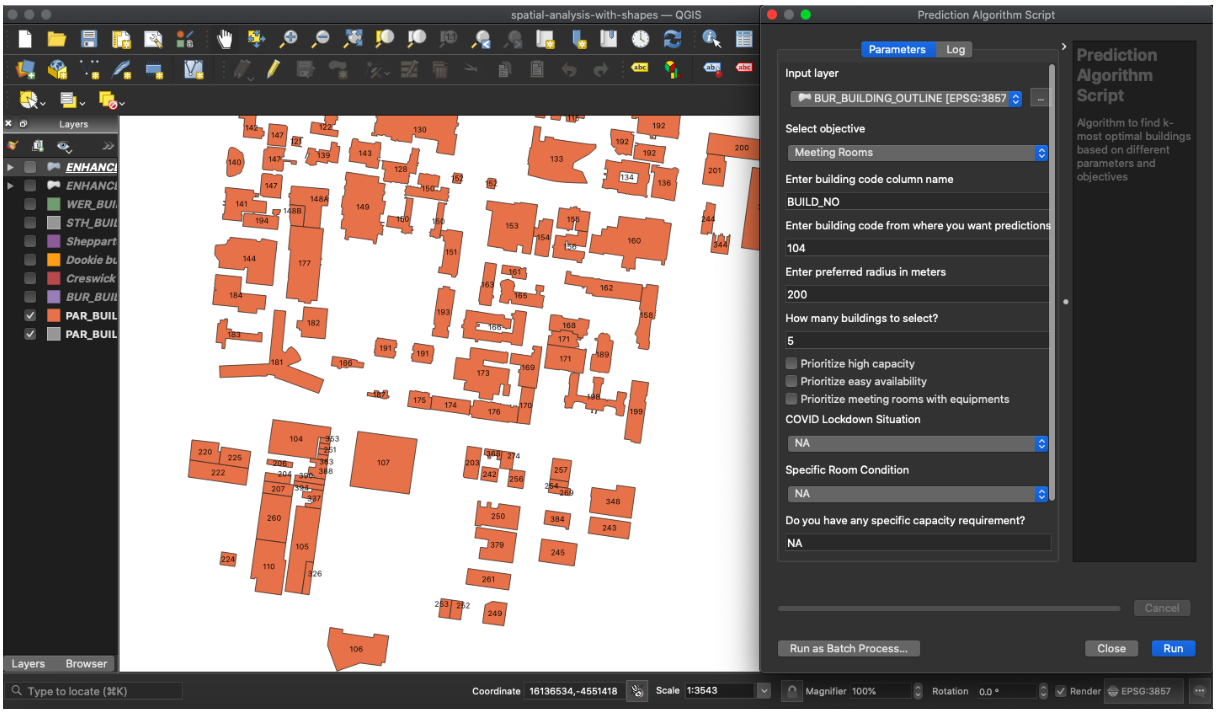
Anbin Hou

Contents

1 QGIS Platform

QGIS is a user friendly Open Source Geographic Information System (GIS) licensed under the GNU General Public License[1]. This platform is widely used to perform spatial data analysis. We have implemented our proposed algorithm as described in the paper using QGIS processing framework[2]. This framework provides a geoprocessing environment that can be used to call native and third-party algorithms from QGIS, making your spatial analysis tasks more productive and easy to accomplish.

QGIS processing framework provides the ability to create custom processing logic using python scripts[3]. Using this framework, we were able to design the UI interface for our prediction algorithm implicitly without creating any UI controls and functionalities. The complete QGIS interface with our prediction algorithm processing script is shown in the Figure 1.


spatial-analysis-with-shapes — QGIS
Prediction Algorithm Script
Parameters
Log
Prediction Algorithm Script
Algorithm to find k-most optimal buildings based on different parameters and objectives
Cancel
Run as Batch Process...
Close
Run
Layers
Browser
Type to locate (⌘K)

Input layer: BUR_BUILDING_OUTLINE [EPSG:3857

Select objective: Meeting Rooms

Enter building code column name: BUILD_NO

Enter building code from where you want predictions: 104

Enter preferred radius in meters: 200

How many buildings to select?: 5

- [ ] Prioritize high capacity
- [ ] Prioritize easy availability
- [ ] Prioritize meeting rooms with equipments

COVID Lockdown Situation: NA

Specific Room Condition: NA

Do you have any specific capacity requirement?: NA

Coordinate 16136534,-4551418 Scale 1:3543 Magnifier 100% Rotation 0.0 ° Render EPSG:3857

Figure 1: QGIS interface with map outline and prediction algorithm processing script

2 Problem Objective

In this section, we will briefly describe the objective of our problem. We were provided with the spatial data of the University of Melbourne's Parkville campus as shown in the Figure 2. The provided data contained building names, codes and their corresponding shapes. Our objective was to use supply and demand data of meeting rooms or toilet facilities to predict the most optimal nearest building from a particular building.

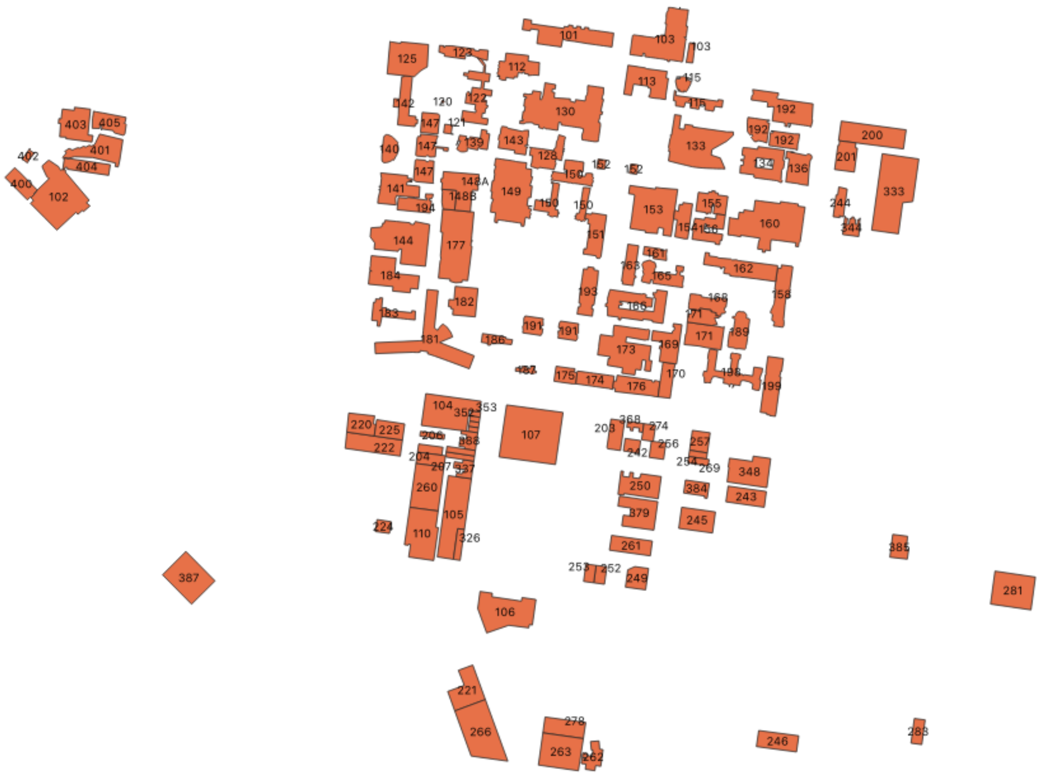
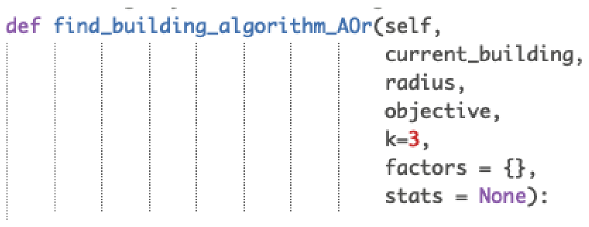


Figure 2: The University of Melbourne's Parkville Campus

For example, our objective was to predict which is the most optimal nearest building from building 104 within 200 metres radius which has the highest probability of providing a meeting room easily.

3 Python Algorithm Implementation

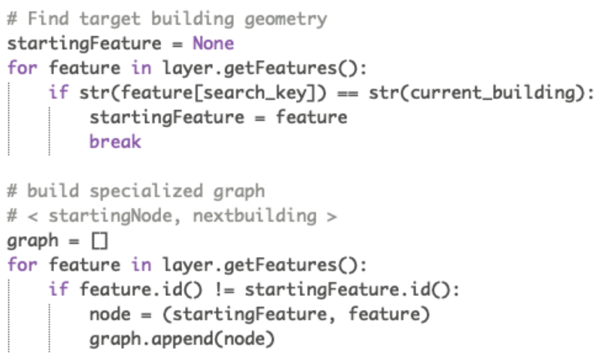
In order to solve the previous stated objective, we implemented our algorithm as described in the paper using QGIS processing framework by creating custom python scripts. Our function has the following signature as shown in Figure 3.



```python
def find_building_algorithm_AOr(self,
                                current_building,
                                radius,
                                objective,
                                k=3,
                                factors = {},
                                stats = None):
```

Figure 3: Algorithm python function

We first created a specialized graph from the provided current building as per the process described in the paper. This is shown in Figure 4.



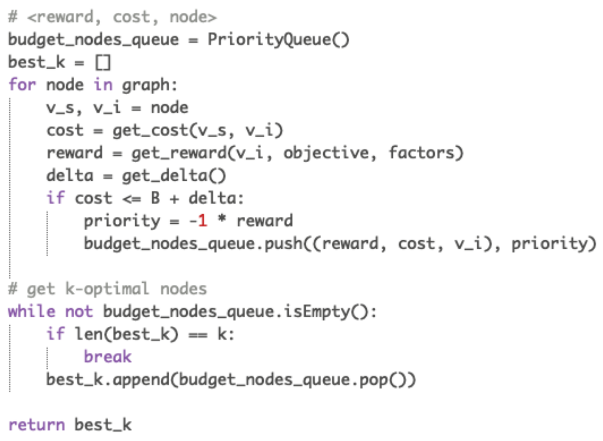
```
# Find target building geometry
startingFeature = None
for feature in layer.getFeatures():
    if str(feature[search_key]) == str(current_building):
        startingFeature = feature
        break

# build specialized graph
# < startingNode, nextbuilding >
graph = []
for feature in layer.getFeatures():
    if feature.id() != startingFeature.id():
        node = (startingFeature, feature)
        graph.append(node)
```

Figure 4: Generating specialized graph

We then executed critical steps of our algorithm by using our customized reward(`get_reward`) and cost function(`get_cost`) to get k-optimal goal nodes as shown in Figure 5. This implementation is solving below constraint satisfaction problem as described in the paper.

$$\arg\max_{r \subseteq V} I(r) \textit{ subject to } C(r) \leq B + \delta \leq L \tag{1}$$

where $\delta \in \mathbb{R}_0^+ \cup \{\infty\}$.



```
# <reward, cost, node>
budget_nodes_queue = PriorityQueue()
best_k = []
for node in graph:
    v_s, v_i = node
    cost = get_cost(v_s, v_i)
    reward = get_reward(v_i, objective, factors)
    delta = get_delta()
    if cost <= B + delta:
        priority = -1 * reward
        budget_nodes_queue.push((reward, cost, v_i), priority)

# get k-optimal nodes
while not budget_nodes_queue.isEmpty():
    if len(best_k) == k:
        break
    best_k.append(budget_nodes_queue.pop())

return best_k
```

Figure 5: Solving budget constraint problem on a specialized graph

4 The Reward Function

In this section, we will explain the critical component of our algorithm which handles how to generate rewards for buildings based on different factors. This function generates rewards based on the logic described below.

If we want to predict the optimal building for booking a meeting room, rewards are calculated based on different factors as shown below.

- **No Factors**: Reward $= \frac{\text{supply of meeting rooms}}{\text{demand of meeting rooms}}$
- **Book a meeting room with required capacity** C:

Reward $= \frac{\text{supply of meeting rooms}}{\text{demand of meeting rooms}}$ subject to $supply >= C$

- **Book a meeting room in COVID lockdown situation**:
  - **Strict - Demand(0%):** Reward = supply of meeting rooms
  - **Medium - Demand(25%):** Reward $= \frac{\text{supply of meeting rooms}}{\text{demand of meeting rooms} * 0.25}$
  - **Low - Demand(50%):** Reward $= \frac{\text{supply of meeting rooms}}{\text{demand of meeting rooms} * 0.50}$
- **Book a meeting room with high capacity**:
  Reward $= \frac{\text{supply of meeting rooms}}{\text{demand of meeting rooms}} * \frac{\text{average room size}}{\text{total average room size}}$
- **Book a meeting room with easy availability**:
  Reward $= \frac{\text{supply of meeting rooms}}{\text{demand of meeting rooms}} * (1 - \frac{\text{total meetings held}}{\text{overall meetings}})$
- **Book a meeting room with equipment**:
  Reward $= \frac{\text{supply of meeting rooms}}{\text{demand of meeting rooms}} * \frac{\text{equipment count}}{\text{total equipment count}}$
- **Book a meeting room with different room conditions**:
  - **Excellent:** Reward $= \frac{\text{supply of meeting rooms}}{\text{demand of meeting rooms}} * \frac{\text{excellent rooms}}{\text{overall excellent rooms}}$
  - **Very Good:** Reward $= \frac{\text{supply of meeting rooms}}{\text{demand of meeting rooms}} * \frac{\text{very good rooms}}{\text{overall very good rooms}}$
  - **Good:** Reward $= \frac{\text{supply of meeting rooms}}{\text{demand of meeting rooms}} * \frac{\text{good rooms}}{\text{overall good rooms}}$

If we want to predict the optimal building for using a toilet facility, rewards are calculated based on different factors as shown below.

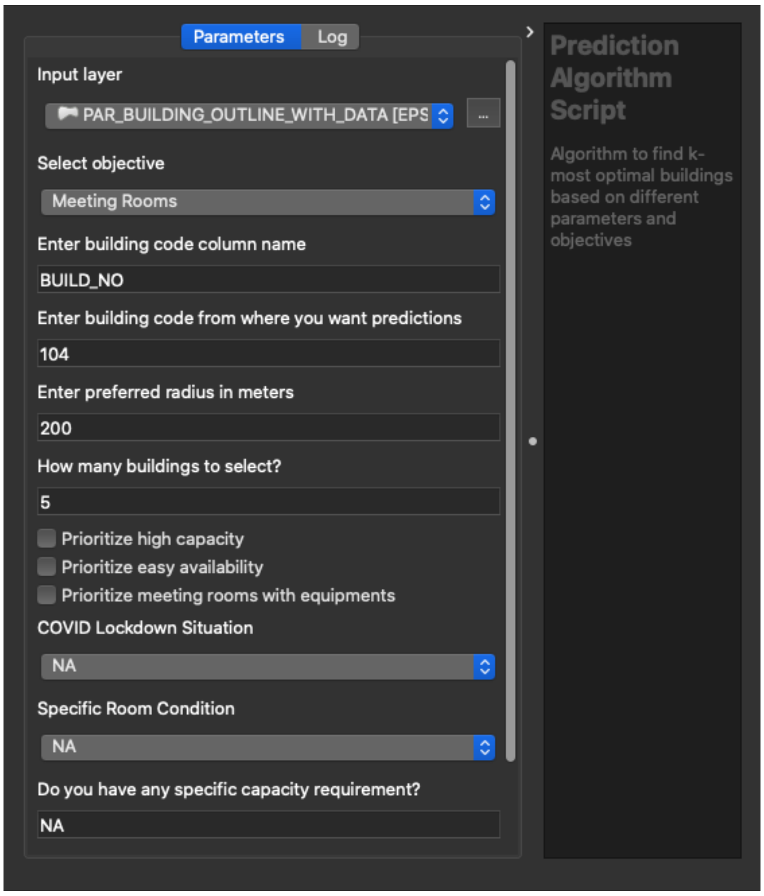
- **No Factors**: Reward $= \frac{\text{supply of toilet facilities}}{\text{demand of toilet facilities}}$
- **Search a toilet facility with required capacity** C:
  Reward $= \frac{\text{supply of toilet facilities}}{\text{demand of toilet facilities}}$ subject to $supply >= C$
- **Search a toilet facility in COVID lockdown situation**:
  - **Strict - Demand(0%):** Reward = supply of toilet facilities
  - **Medium - Demand(25%):** Reward $= \frac{\text{supply of toilet facilities}}{\text{demand of toilet facilities} * 0.25}$
  - **Low - Demand(50%):** Reward $= \frac{\text{supply of toilet facilities}}{\text{demand of toilet facilities} * 0.50}$
- **Search a toilet facility with high capacity**:
  Reward $= \frac{\text{supply of toilet facilities}}{\text{demand of toilet facilities}} * \frac{\text{average room size}}{\text{total average room size}}$
- **Search a toilet facility with different room conditions**:
  - **Excellent:** Reward $= \frac{\text{supply of toilet facilities}}{\text{demand of toilet facilities}} * \frac{\text{excellent rooms}}{\text{overall excellent rooms}}$
  - **Very Good:** Reward $= \frac{\text{supply of toilet facilities}}{\text{demand of toilet facilities}} * \frac{\text{very good rooms}}{\text{overall very good rooms}}$
  - **Good:** Reward $= \frac{\text{supply of toilet facilities}}{\text{demand of toilet facilities}} * \frac{\text{good rooms}}{\text{overall good rooms}}$

5 Prediction Algorithm Results

In this section, we will show different results that were collected after running our implemented algorithm python script using QGIS processing framework. After execution of the script, we selected k-optimal buildings in the provided QGIS layer space. We have only shown results of the algorithm when there are no factors provided and when there is a COVID-19 lockdown situation to give a perspective about the working of the algorithm. We can easily execute this algorithm for different situations and factors as described in Section 4.

5.1 No Factors

We first executed our algorithm with no factors to find 5 most optimal buildings from building 104 within 200 metres radius with δ of around 100 metres for booking a meeting room. This process is shown in the Figure 6.



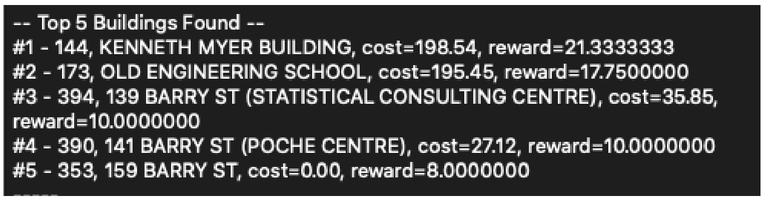
Input layer
PAR_BUILDING_OUTLINE_WITH_DATA [EPS
Select objective
Meeting Rooms
Enter building code column name
BUILD_NO
Enter building code from where you want predictions
104
Enter preferred radius in meters
200
How many buildings to select?
5
☐ Prioritize high capacity
☐ Prioritize easy availability
☐ Prioritize meeting rooms with equipments
COVID Lockdown Situation
NA
Specific Room Condition
NA
Do you have any specific capacity requirement?
NA

Prediction Algorithm Script

Algorithm to find k-most optimal buildings based on different parameters and objectives

Figure 6: Prediction algorithm QGIS processing script

After execution, we collected following results as shown in Figure 7 and 8.



```
-- Top 5 Buildings Found --
#1 - 144, KENNETH MYER BUILDING, cost=198.54, reward=21.3333333
#2 - 173, OLD ENGINEERING SCHOOL, cost=195.45, reward=17.7500000
#3 - 394, 139 BARRY ST (STATISTICAL CONSULTING CENTRE), cost=35.85,
reward=10.0000000
#4 - 390, 141 BARRY ST (POCHE CENTRE), cost=27.12, reward=10.0000000
#5 - 353, 159 BARRY ST, cost=0.00, reward=8.0000000
```

Figure 7: Prediction algorithm results

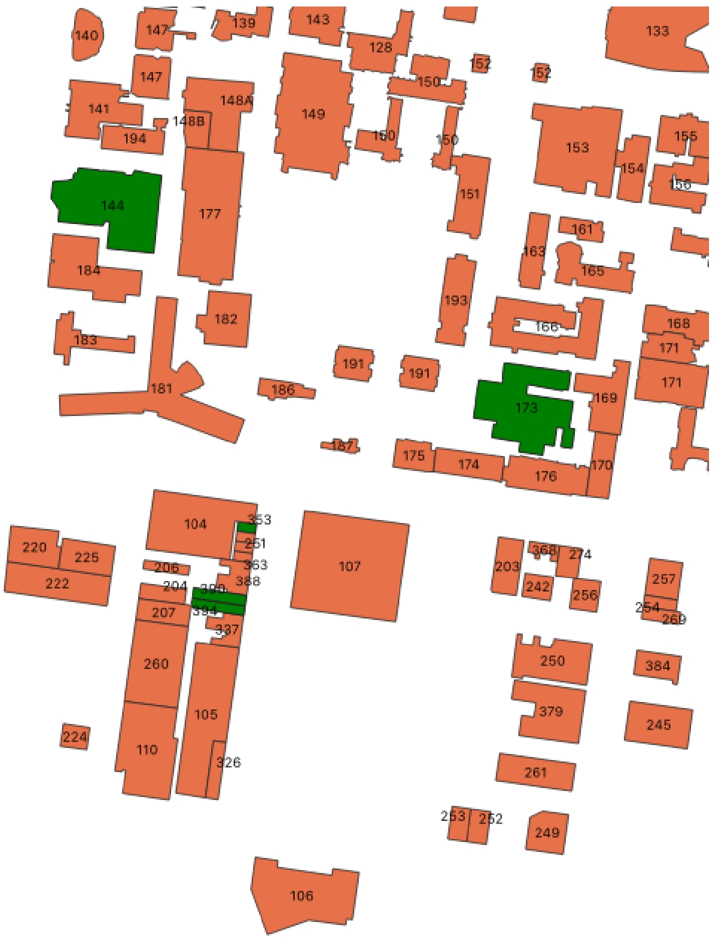
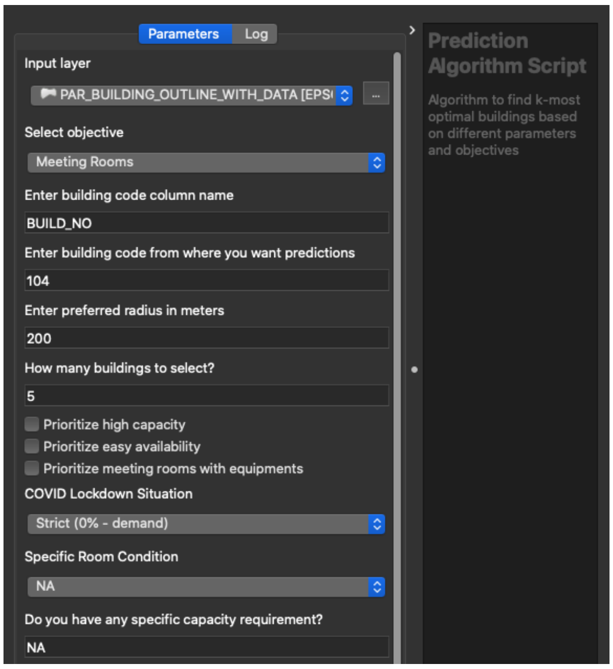


Figure 8: 5 most optimal buildings from building 104 within 300 metres radius

5.2 COVID-19 Lockdown

In this section, we will show the results of our algorithm in the COVID-19 lockdown situation. We will again use the same scenario of finding 5 most optimal buildings from building 104 within 200 metres radius with δ of around 100 metres for booking a meeting room.

5.2.1 COVID-19 Strict Lockdown - Demand (0%)


Parameters
Log

Input layer
PAR_BUILDING_OUTLINE_WITH_DATA [EPS(

Select objective
Meeting Rooms

Enter building code column name
BUILD_NO

Enter building code from where you want predictions
104

Enter preferred radius in meters
200

How many buildings to select?
5

- [ ] Prioritize high capacity
- [ ] Prioritize easy availability
- [ ] Prioritize meeting rooms with equipments

COVID Lockdown Situation
Strict (0% - demand)

Specific Room Condition
NA

Do you have any specific capacity requirement?
NA

Prediction Algorithm Script

Algorithm to find k-most optimal buildings based on different parameters and objectives

Figure 9: Prediction algorithm QGIS processing script for COVID-19 strict lockdown

After execution, we collected the following results as shown in Figure 10 and 11.

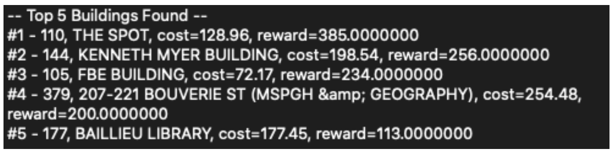

```
-- Top 5 Buildings Found --
#1 - 110, THE SPOT, cost=128.96, reward=385.0000000
#2 - 144, KENNETH MYER BUILDING, cost=198.54, reward=256.0000000
#3 - 105, FBE BUILDING, cost=72.17, reward=234.0000000
#4 - 379, 207-221 BOUVERIE ST (MSPGH &amp; GEOGRAPHY), cost=254.48,
reward=200.0000000
#5 - 177, BAILLIEU LIBRARY, cost=177.45, reward=113.0000000
```

Figure 10: Prediction algorithm results for COVID-19 strict lockdown

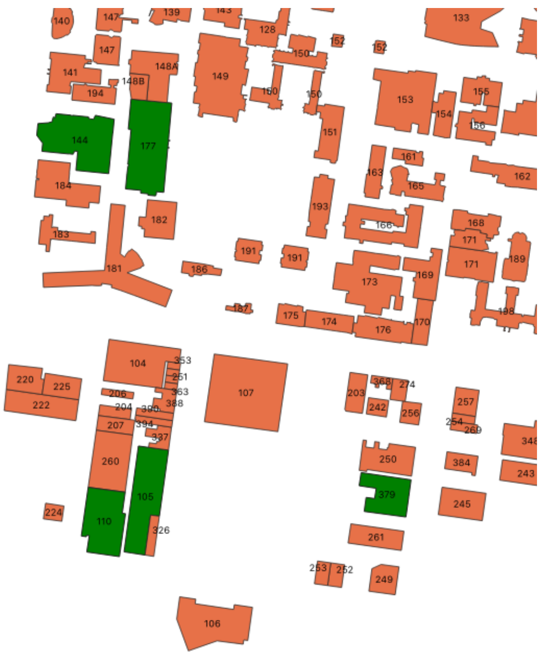
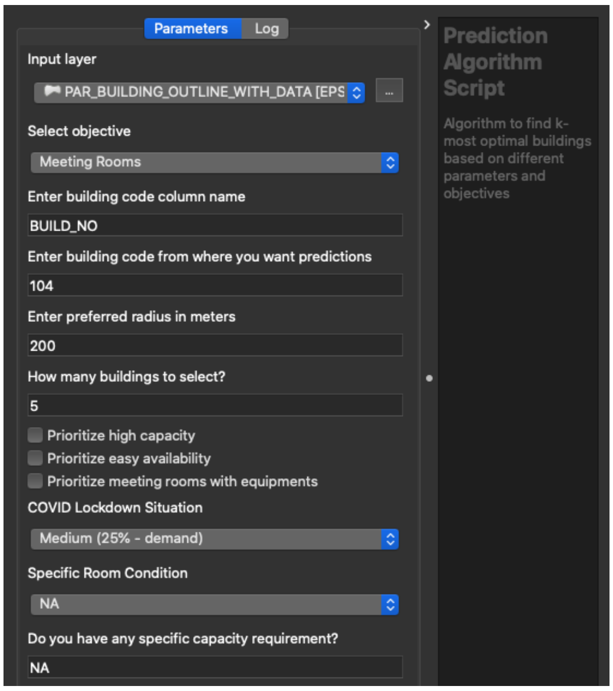


Figure 11: 5 most optimal buildings from building 104 under COVID strict lockdown

5.2.2 COVID-19 Medium Lockdown - Demand (25%)



Parameters Log

Input layer

PAR_BUILDING_OUTLINE_WITH_DATA [EPS

Select objective

Meeting Rooms

Enter building code column name

BUILD_NO

Enter building code from where you want predictions

104

Enter preferred radius in meters

200

How many buildings to select?

5

- [ ] Prioritize high capacity
- [ ] Prioritize easy availability
- [ ] Prioritize meeting rooms with equipments

COVID Lockdown Situation

Medium (25% - demand)

Specific Room Condition

NA

Do you have any specific capacity requirement?

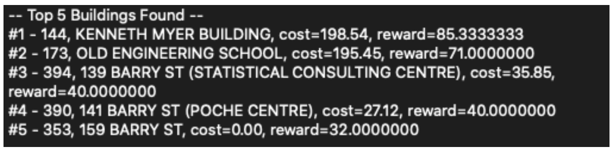
NA

Prediction Algorithm Script

Algorithm to find k-most optimal buildings based on different parameters and objectives

Figure 12: Prediction algorithm QGIS processing script for COVID-19 medium lockdown

After execution, we collected the following results as shown in Figure 13 and 14.



```
-- Top 5 Buildings Found --
#1 - 144, KENNETH MYER BUILDING, cost=198.54, reward=85.3333333
#2 - 173, OLD ENGINEERING SCHOOL, cost=195.45, reward=71.0000000
#3 - 394, 139 BARRY ST (STATISTICAL CONSULTING CENTRE), cost=35.85,
reward=40.0000000
#4 - 390, 141 BARRY ST (POCHE CENTRE), cost=27.12, reward=40.0000000
#5 - 353, 159 BARRY ST, cost=0.00, reward=32.0000000
```

Figure 13: Prediction algorithm results for COVID-19 medium lockdown

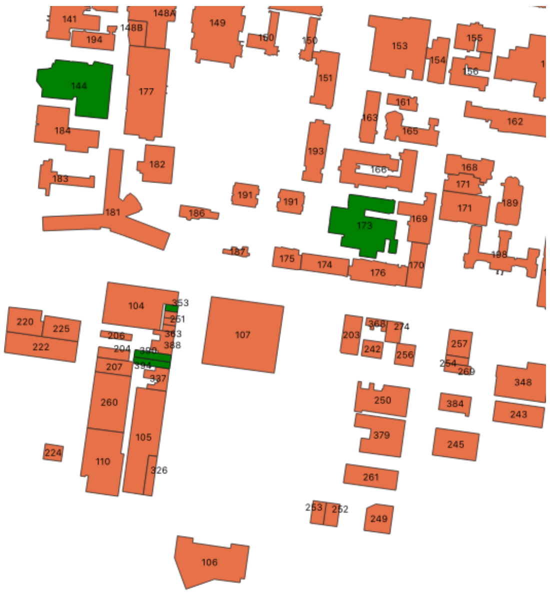


Figure 14: 5 most optimal buildings from building 104 under COVID-19 medium lockdown

References

[1] QGIS Project. https://www.qgis.org/en/site/.

[2] QGIS Processing Framework. https://docs.qgis.org/3.10/en/docs/user_manual/processing/intro.html.

[3] Writing new Processing algorithms as Python scripts. https://docs.qgis.org/3.10/en/docs/user_manual/processing/scripts.html.