

Individual Project 1

Vishal Batvia

CMPT 827 Intelligent Systems

Author Note

I am submitting the Project 1 day late because of bad health issues

Individual Project 1

1.1. Searching in Space-Time Domain

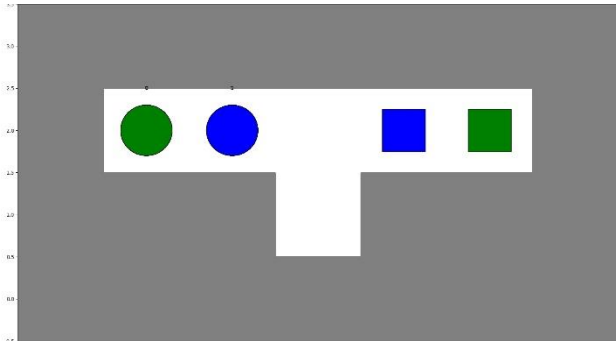


Figure 2 stage 1

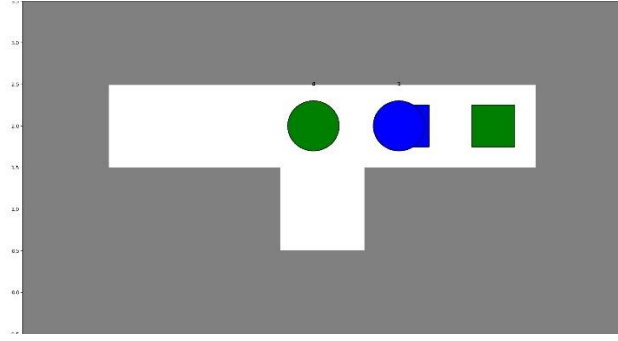


Figure 1 Stage 2

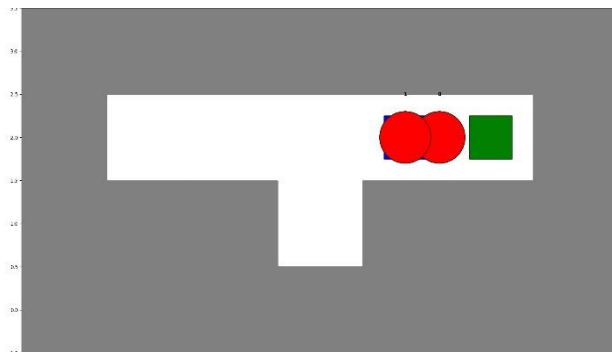


Figure 3 Stage 3

1.2. Vertex Constraint

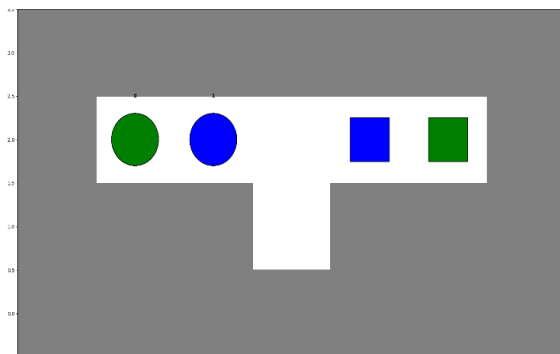


Figure 4 stage 1 Start

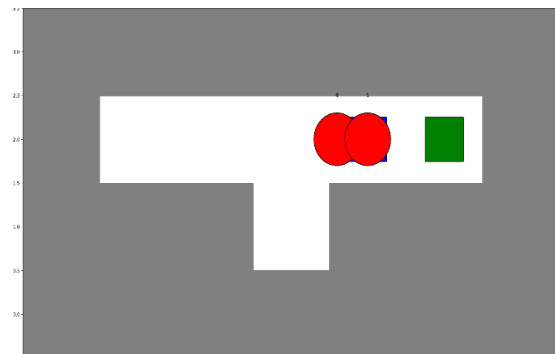
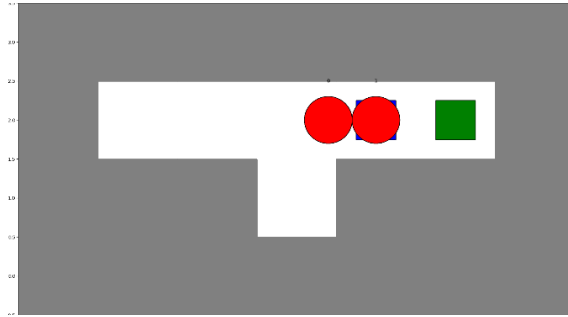


Figure 5 Stage 2 Collision



*Figure 6 Stage 3 Vertex condition at timestep 4
So wait for one timestep before going to the goal*

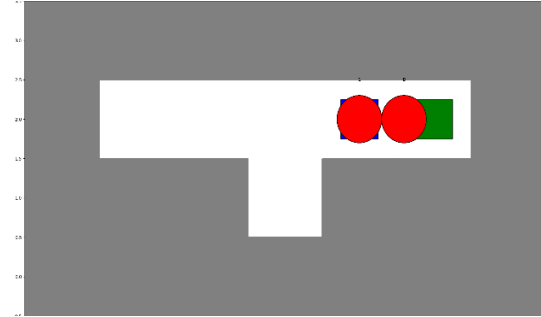


Figure 7 Stage 4 Go to goal after the wait

1.3 Edge Constraints

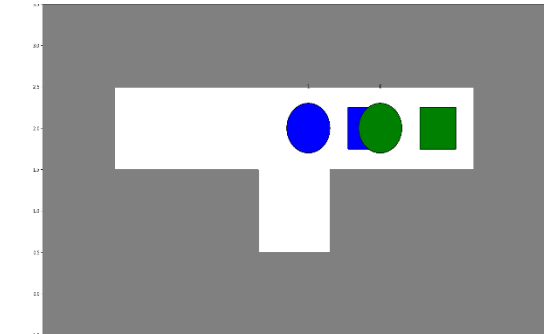
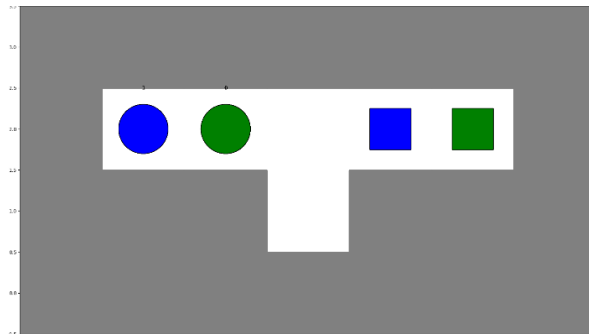
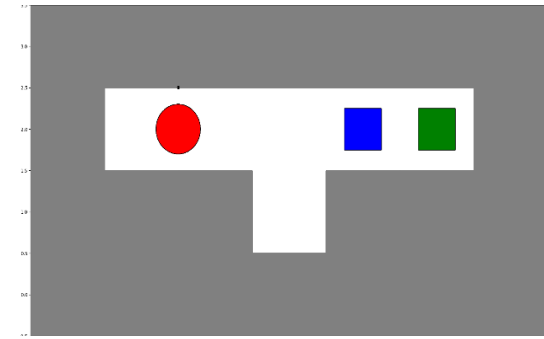
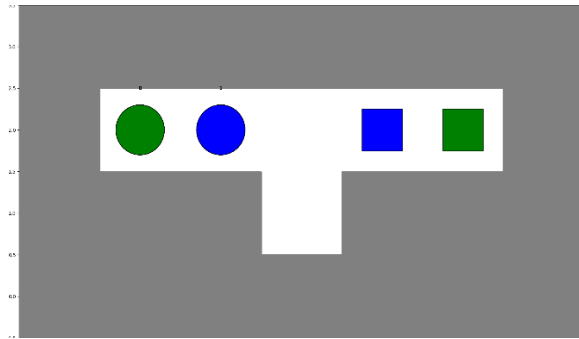


Image 1 is initial state.

Image 2 is agent one cannot go from (1,2) to (1,3) at timestep 1.

Image 3 is time step 2.

Image 4 is going to the goal location.

1.4. Goal Constraint

```
# Constraint for Task 1.2
# constraints = [{'agent': 0, 'loc': [(1,5)], 'timestep': 4}]

# Constraint for Task 1.3
# constraints = [{'agent': 1, 'loc': [(1,2),(1,3)], 'timestep': 1}]

# Constraints for Task 1.4
# constraints = [{'agent': 0, 'loc': [(1,5)], 'timestep': 10},
#               {'agent': 0, 'loc': [(1,3),(1,4)], 'timestep': 5},
#               {'agent': 0, 'loc': [(1,3),(1,4)], 'timestep': 7},
#               {'agent': 0, 'loc': [(1,2),(1,3)], 'timestep': 2}]

# Constraints for Task 1.5
# constraints = [{'agent': 1, 'loc': [(1,4)], 'timestep': 2},
#               {'agent': 1, 'loc': [(1,3),(1,2)], 'timestep': 2}]
```

As shown above, It was made sure that the agents do not reach the goal locations before the timestep ten and then there was another constraint, which makes sure that the agent at time step 10 cannot be at its goal location.

1.5. No collision Constraint

As shown in the figure in section 1.4, it was made sure that the agent one cannot be at the goal location (1,4) at timestep two and also it cannot move from (1,3) to (1,2) at timestep 2.

Agent 1 is at (1,3), and thus the only location it can go is (2,3), which gives a chance for agent 0 to pass agent one before the collision. Therefore there was no collision, and the agents reached the target.

```
File Edit View Search Terminal Help
(base) vishal@vishal-Strix-GL704GM-GL704GM:~/CMPT827/code$ python run_experiments.py --instance instances/exp1.txt --solver Prioritized
***Import an instance***
Start locations
@ @ @ @ @ @ @
@ 0 1 . . . @
@ @ @ . @ @ @
@ @ @ @ @ @ @

Goal locations
@ @ @ @ @ @ @
@ . . . 1 0 @
@ @ @ . @ @ @
@ @ @ @ @ @ @

***Run Prioritized***

Found a solution!

CPU time (s): 0.00
Sum of costs: 8
[[ (1, 1), (1, 2), (1, 3), (1, 4), (1, 5)], [(1, 2), (1, 3), (2, 3), (1, 3), (1, 4)] ]
***Test paths on a simulation***
(base) vishal@vishal-Strix-GL704GM-GL704GM:~/CMPT827/code$
```

2.1. Vertex Constraints

```
constraints.append({'agent' : next_agent, 'loc' : [path[nPath]], 'timestep' : nPath})
```

2.2. Edge Constraints

```
constraints.append({'agent' : next_agent, 'loc' : [path[nPath],path[nPath-1]], 'timestep' : nPath})
```

2.3. Additional Constraints (For goal location)

```
# constraints.append({'agent' : next_agent, 'loc' : [path[len(path) - 1]], 'timestep' : nPath})
```

2.4. Addressing the failures

Mentioned a timestep of 10 as a failure condition, so if the steps are more than ten, then it says no solution. The number changes if the size of the map changes. It was just chosen as a random number, and it works.

2.5. Showing Prioritized is not optimal.

- It returns no solution for exp2_3.txt given that ordering but it answers if the ordering is reversed
- For the below map, even if the order is changed, it gives no solution even though one order solution exists. The plan is present in custom maps.

```

4 7
@ @ @ @ @ @ @
@ @ . . . @
@ @ @ . @ @ @
@ @ @ @ @ @ @
2
1 2 1 4
2 3 2 3

```

- For test_4, it provides a solution, but it contains a collision, so it is not able to give a collision-free answer.

3.1. Detecting collisions

```

detect_keys([])
Generate node 0
[{'a1': 0, 'a2': 1, 'loc': [(1, 4)], 'timestep': 3}]

```

3.2. Converting collisions to constraints

```
Generate node 0
[{'a1': 0, 'a2': 1, 'loc': [(1, 4)], 'timestep': 3}]
[{'agent': 0, 'loc': [(1, 4)], 'timestep': 3}, {'agent': 1, 'loc': [(1, 4)], 'timestep': 3}]
```

3.3. Implement High-Level Search

```

Activities  Terminal v
vishal@vishal-Strix-GL704GM:~$ python run_experiments.py --instance instances/exp4.txt --solver CBS
***Import an instance***
Start locations
0 0 0 0 0 0 0
0 0 1 . . . 0
0 0 . . . . 0
0 . . . . . 0
0 . . . . . 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0

Goal locations
0 0 0 0 0 0 0
0 . . . . . 0
0 . . . . . 0
0 . . . . . 0
0 . . . . . 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0

***Run CBS***
Generate node 0
[{'a1': 0, 'a2': 1, 'loc': [(2, 4)], 'timestep': 3}]
[{'agent': 0, 'loc': [(2, 4)], 'timestep': 3}, {'agent': 1, 'loc': [(2, 4)], 'timestep': 3}]
Expand node 0
Generate node 1
Generate node 2
Expand node 1
Generate node 3
Generate node 4
Expand node 2
Generate node 5
Generate node 6
Expand node 4
Generate node 7
Generate node 8
Expand node 5
Generate node 9
Generate node 10
Expand node 6
Generate node 11
Generate node 12
Expand node 8
Generate node 13
Generate node 14
Expand node 10
Generate node 15
Generate node 16
Expand node 11
Generate node 17
Generate node 18
Expand node 13
Generate node 19
Generate node 20
Expand node 16
Generate node 21
Generate node 22
Expand node 18
Generate node 23
Generate node 24
Expand node 24
Generate node 25
Generate node 26
Expand node 3
No collision
***Test paths on a simulation***
vishal@vishal-Strix-GL704GM:~$

```

3.4. Testing on Implementation

The image on the left is from instance folder and on the right is from the test on the code

instances/	41	instances/test_1.txt	42
instances/	19	instances/test_10.txt	19
instances/	35	instances/test_11.txt	35
instances/	36	instances/test_12.txt	36
instances/	36	instances/test_13.txt	37
instances/	24	instances/test_14.txt	24
instances/	50	instances/test_15.txt	51
instances/	51	instances/test_16.txt	51
instances/	39	instances/test_17.txt	39
instances/	32	instances/test_18.txt	32
instances/	47	instances/test_19.txt	47
instances/	18	instances/test_2.txt	18
instances/	28	instances/test_20.txt	28
instances/	46	instances/test_21.txt	46
instances/	51	instances/test_22.txt	51
instances/	32	instances/test_23.txt	32
instances/	47	instances/test_24.txt	47
instances/	40	instances/test_25.txt	40
instances/	42	instances/test_26.txt	42
instances/	40	instances/test_27.txt	41
instances/	41	instances/test_28.txt	41
instances/	48	instances/test_29.txt	48
instances/	28	instances/test_3.txt	28
instances/	43	instances/test_30.txt	44
instances/	39	instances/test_31.txt	39
instances/	30	instances/test_32.txt	30
instances/	28	instances/test_33.txt	29
instances/	33	instances/test_34.txt	33
instances/	30	instances/test_35.txt	30
instances/	23	instances/test_36.txt	23
instances/	38	instances/test_37.txt	39
instances/	28	instances/test_38.txt	28
instances/	35	instances/test_39.txt	35
instances/	32	instances/test_4.txt	32
instances/	24	instances/test_40.txt	24
instances/	45	instances/test_41.txt	46
instances/	57	instances/test_42.txt	57
instances/	43	instances/test_43.txt	43
instances/	33	instances/test_44.txt	34
instances/	24	instances/test_45.txt	24
instances/	57	instances/test_46.txt	58
instances/	65	instances/test_47.txt	69
instances/	36	instances/test_48.txt	36
instances/	42	instances/test_49.txt	44
instances/	26	instances/test_5.txt	27
instances/	48	instances/test_50.txt	49
instances/	24	instances/test_6.txt	24
instances/	34	instances/test_7.txt	34
instances/	38	instances/test_8.txt	39
instances/	24	instances/test_9.txt	24

4.1. Detecting collisions

```
# Task 4.1 add key positive

if not ("positive" in constraint.keys()):
    constraint['positive'] = False
```

4.2. Converting collision to constraints

```
if random.randint(0,1):
    random_agent = 'a1'
else:
    random_agent = 'a2'

if len(loc)>1 and random_agent == 'a1':

    first_constraint = {'agent' : collision['a1'], 'loc' : [loc[1],loc[0]], 'timestep' : timestep, 'positive' : True}
    first_constraint = {'agent' : collision['a1'], 'loc' : [loc[1],loc[0]], 'timestep' : timestep, 'positive' : False}
    return [first_constraint,second_constraint]

if len(loc)>1 and random_agent == 'a2':
    second_constraint = {'agent' : collision['a2'], 'loc' : [loc[0],loc[1]], 'timestep' : timestep, 'positive' : True}
    second_constraint = {'agent' : collision['a2'], 'loc' : [loc[0],loc[1]], 'timestep' : timestep, 'positive' : False}
    return [first_constraint,second_constraint]

if len(loc) == 1:
    first_constraint = {'agent' : collision[random_agent], 'loc' : [loc[0]], 'timestep' : timestep, 'positive' : True}
    second_constraint = {'agent' : collision[random_agent], 'loc' : [loc[0]], 'timestep' : timestep, 'positive' : False}
    return [first_constraint,second_constraint]
```

4.3. Adjusting the High-Level Search

```

File Edit View Search Terminal Help
***Import an instance***
Start locations
@ @ @ @ @ @ @
@ . 1 . . . @
@ 0 . . . . @
@ . . . . . @
@ . . . . . @
@ @ @ @ @ @ @
@ @ @ @ @ @ @

Goal locations
@ @ @ @ @ @ @
@ . . . . . @
@ . . . . . @
@ . . . . 0 @
@ . . . 1 . @
@ @ @ @ @ @ @
@ @ @ @ @ @ @

***Run CBS***
Generate node 0
[{'a1': 0, 'a2': 1, 'loc': [(2, 4)], 'timestep': 3}]
[{'agent': 0, 'loc': [(2, 4)], 'timestep': 3}, {'agent': 1, 'loc': [(2, 4)], 'timestep': 3}]
Expand node 0
Generate node 1
Generate node 2
Expand node 1
Generate node 3
Generate node 4
Expand node 2
Generate node 5
Generate node 6
Expand node 6
Generate node 7
Generate node 8
Expand node 8
Generate node 9
Generate node 10
Expand node 10
Generate node 11
Generate node 12
Expand node 9
No collision
***Test paths on a simulation***
(base) vishal@vishal-Strix-GL704GM-GL704GM:~/CMPT827/code$ █

```

5. Benchmarking on MAPF solvers

I tried to run it on different maps, but it was getting Killed for anything more than a map of 16X16. I checked the locations of the start and goal, which I was mentioning, and it was correct.

A few of them are placed in the custom instances folder