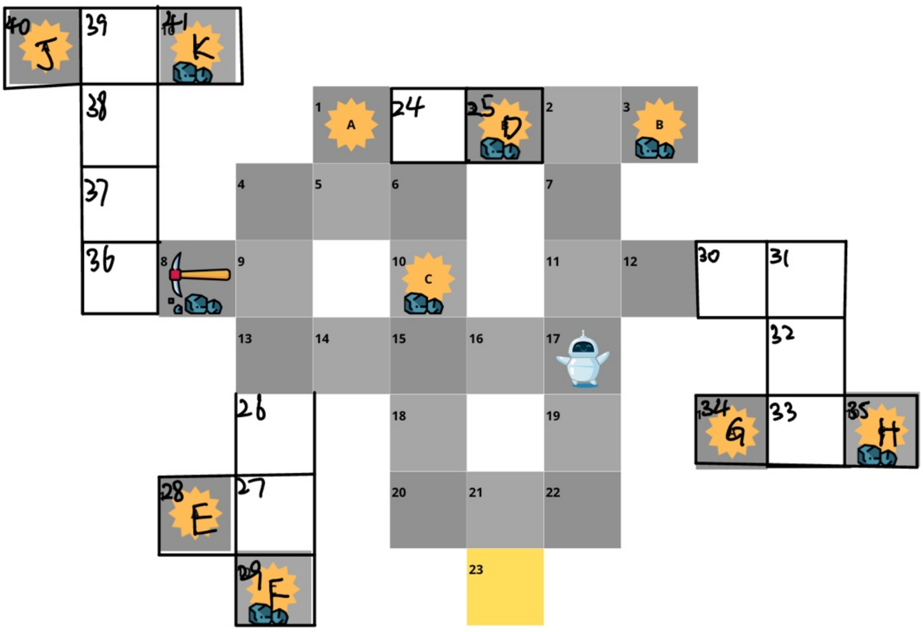
Informatics 2D Coursework 2 Report

1. (5 marks) Task 2.1 Design

In my design, I am trying to make more ore goals so that the mine bot will have more choice to face to make the problem harder. Let the ore be more separated so that the mine bot will have more possible path. Set two ore closer with one blocked and another one unblocked to make mine bot have more time to decide mine which one first and how to find hammer.

Based on the original graph, I add 7 more ore and block 4 of them to make the problem become harder.

Following graph is the hard problem.



2. (10 marks) Task 2.2 Evaluation

My trial

Wg : Wh	Number of actions	Running times
0 : 1	339	0.01
1 : 2	none	none
1 : 3	none	none
1 : 4	none	none

1 : 5	*285*	*12.85*
1 : 6	*285*	*0.2*
1 : 7	*287*	*0.09*
1 : 8	*289*	*0.07*
1 : 9	*289*	*0.07*
1 : 10	*289*	*0.02*
1 : 11	*295*	*0.02*

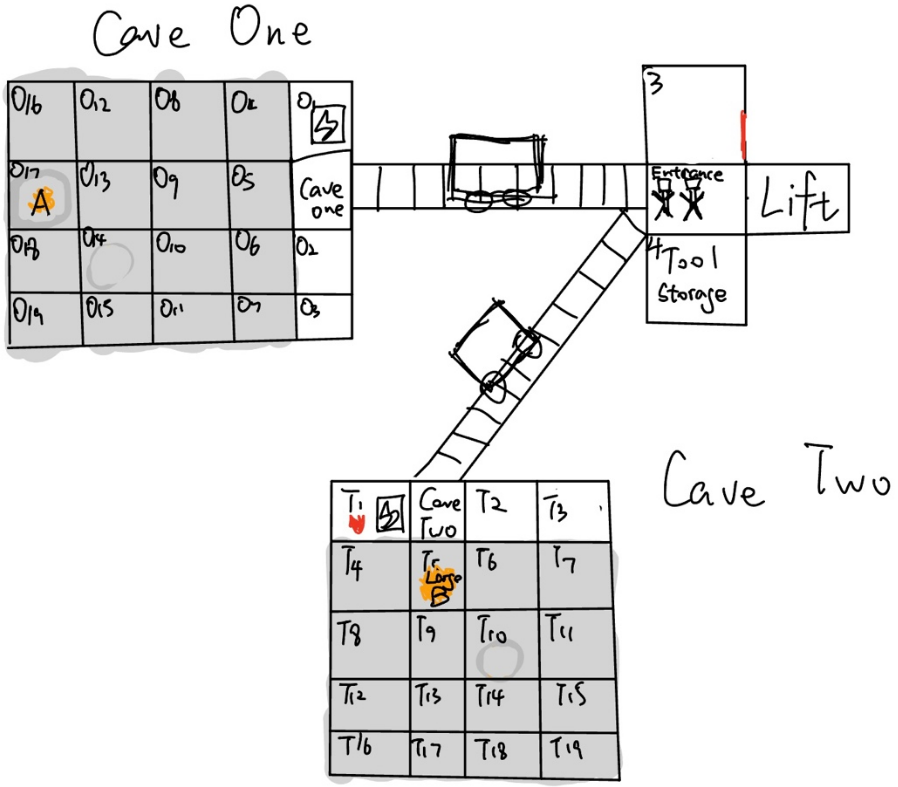
The basic running time for 1 : 5 is about 12 seconds.

From the table above, it is easy to see that lower ratio of wg to wh will have a lower running time and more number of actions. Higher ratio of wg to wh will face the situation that ff cannot run out a solution or have a huge amount of time to do this.

After trying different weights of cost so far to reach and estimated cost to get from s to the goal state, we can find that larger weight on estimated cost to get from s to the goal state will result in a faster running time, since this algorithm will be an approximate greedy search and only care about the goal. Thus, they will only care about the goal instead of the cost from the current state to goal state, which less likely result in an optimal solution. If we increase the cost, then the search algorithm will more likely be a breadth first search. Thus, it is huge amount of searching branch for more goals, since the growth is exponential.

3. (25 marks) Task 3.4 Your Extension

Based on the real world, we know that there will be more than one mine caves and we need to take tramcar to get to different mine caves from the platform where we store tools and get from the Lift. After arriving underground through the Lift, we can go to entrance of the two train stations to get on tramcar here. We also know that all the ore should be blocked by the rock when we first enter the cave. If we break the rock, then rock will become into small pieces. We need to pick them from cave and then transport them outside the cave. It is also important that charging station will cause fire in the cave, so we must put out fire first in order *to minimize losses and keep mine bot safe.*



Above graph is the layout of problem 5. In domain 5, we add two train objects to treat them as tramcar in the cave. In predicates, we add TrainOne and TrainTwo to distinguish the station of two trains from other cells. RockGet predicate is used to show whether we put blocked rock into train that the train will automatically bring these pieces of broken stones out of the cave.

HardRock shows the cell is filled with rock and you cannot go forward. RockOnBot can help detect whether mine bot is carry rock on.

Here, two actions TurnOnTrainOne and TurnOnTrainTwo are actions that mine bot can turn on the train and the train will move from one station to another. There are two actions called SitOnTrainOne and SitOnTrainTwo showing that mine bot can take on the train and the train will take it to another station. Similarly, SitOnTrainOneTogether and SitOnTrainTwoTogether will take both of mine bots from one station to another one. Here, Break action will be a little bit different with former one, since it needs to break hard rock in the cell connected to hard rock. All move actions add one restriction that we cannot move on to the cell blocked by hard rock. PickupPieces action shows that mine bot can pick up broken stones to put it on the train so that there is no broken stone in the cave stuck mine bot's way. Two actions PutRockOnTrainOne and PutRockOnTrainTwo is that mine bot will put the broken stone into train and the train will take all broken stone away. Finally, before getting the ore, we need to put out fire first to make the cave safe, so we add NoFire predicate to make put out fire become the top priority. If we put out the fire, NoFire will be true, and all actions can be taken on. All tools will be stored in tool storage first.

In the example problem, we simplified the task so that ff can get an action. In this problem, our goal is getting Ore A and Ore B(large ore), putting Rock in O17, O13, O9, O5 and T5 on the *train and putting out fire in T1 first.*

The following is ff running outcome on problem 5.

```
 1: MOVECLEAR MB2 ENTRANCE C4
 2: PICKUPITEM MB2 E C4
 3: MOVEHOLDI MB2 C4 ENTRANCE E
 4: MOVECLEAR MB1 C3 ENTRANCE
 5: MOVECLEAR MB1 ENTRANCE C4
 6: SITONTRAINTWO MB2 ENTRANCE CAVETWO
 7: PUTOUT MB2 CAVETWO T1 E
 8: DROP MB2 E CAVETWO
 9: PICKUPITEM MB1 H1 C4
10: MOVEHOLDI MB1 C4 ENTRANCE H1
11: MOVECLEAR MB2 CAVETWO T1
12: SITONTRAINONE MB1 ENTRANCE CAVEONE
13: BREAK MB1 O5 CAVEONE RO5 H1
14: MOVEHOLDI MB1 CAVEONE O5 H1
15: BREAK MB1 O9 O5 RO9 H1
16: MOVECLEAR MB2 T1 CAVETWO
17: TURNONTRAINTWO MB2 CAVETWO ENTRANCE
18: MOVEHOLDI MB1 O5 O9 H1
19: BREAK MB1 O13 O9 RO13 H1
20: MOVEHOLDI MB1 O9 O13 H1
21: BREAK MB1 O17 O13 RO17 H1
22: MOVEHOLDI MB1 O13 O9 H1
23: MOVEHOLDI MB1 O9 O5 H1
24: DROP MB1 H1 O5
25: PICKUPPIECES MB1 O5 RO5
26: MOVEHOLDI MB1 O5 CAVEONE RO5
27: PUTROCKONTRAINONE MB1 RO5
28: MOVECLEAR MB1 CAVEONE O5
29: MOVECLEAR MB1 O5 O9
30: PICKUPPIECES MB1 O9 RO9
31: MOVEHOLDI MB1 O9 O5 RO9
32: MOVEHOLDI MB1 O5 CAVEONE RO9
33: MOVEHOLDI MB1 CAVEONE O1 RO9
34: RECHARGE MB1 O1 BS1
35: MOVEHOLDI MB1 O1 CAVEONE RO9
36: PUTROCKONTRAINONE MB1 RO9
37: MOVECLEAR MB1 CAVEONE O5
38: MOVECLEAR MB1 O5 O9
39: MOVECLEAR MB1 O9 O13
40: PICKUPPIECES MB1 O13 RO13
41: MOVEHOLDI MB1 O13 O9 RO13
42: MOVEHOLDI MB1 O9 O5 RO13
43: MOVEHOLDI MB1 O5 CAVEONE RO13
44: PUTROCKONTRAINONE MB1 RO13
45: MOVECLEAR MB1 CAVEONE O5
46: MOVECLEAR MB1 O5 O9
47: MOVECLEAR MB1 O9 O13
48: MOVECLEAR MB1 O13 O17
49: PICKUPPIECES MB1 O17 RO17
50: MOVEHOLDI MB1 O17 O13 RO17
51: MOVEHOLDI MB1 O13 O9 RO17
52: MOVEHOLDI MB1 O9 O5 RO17
53: MOVEHOLDI MB1 O5 CAVEONE RO17
54: PUTROCKONTRAINONE MB1 RO17
55: MOVECLEAR MB1 CAVEONE O5
56: PICKUPITEM MB1 H1 O5
57: MOVEHOLDI MB1 O5 CAVEONE H1
58: SITONTRAINONE MB1 CAVEONE ENTRANCE
59: SITONTRAINTWO MB1 ENTRANCE CAVETWO
60: BREAK MB1 T5 CAVETWO RT5 H1
61: MOVECLEAR MB2 CAVETWO T5
62: PICKUPPIECES MB2 T5 RT5
63: DROP MB1 H1 CAVETWO
64: MOVEHOLDI MB2 T5 CAVETWO RT5
65: PUTROCKONTRAINTWO MB2 RT5
66: SITONTRAINTWO MB2 CAVETWO ENTRANCE
67: SITONTRAINONE MB2 ENTRANCE CAVEONE
68: MOVECLEAR MB2 CAVEONE O5
69: MOVECLEAR MB2 O5 O9
70: MOVECLEAR MB2 O9 O13
71: MOVECLEAR MB2 O13 O17
72: MINEALONE MB2 O17 OA
73: MOVEHOLDI MB2 O17 O13 OA
74: MOVEHOLDI MB2 O13 O9 OA
75: MOVEHOLDI MB2 O9 O5 OA
76: MOVEHOLDI MB2 O5 CAVEONE OA
77: SITONTRAINONE MB2 CAVEONE ENTRANCE
78: MOVEHOLDI MB2 ENTRANCE C1 OA
79: LIFTOREALONE MB2 OA C1
80: MOVECLEAR MB1 CAVETWO T1
81: RECHARGE MB1 T1 BS2
82: MOVECLEAR MB1 T1 CAVETWO
83: MOVECLEAR MB2 C1 ENTRANCE
84: SITONTRAINTWO MB2 ENTRANCE CAVETWO
85: MOVECLEAR MB1 CAVETWO T5
86: MOVECLEAR MB2 CAVETWO T5
87: MINETOGETHER MB2 MB1 T5 OB
88: MOVEHOLDORETOGETHER MB2 MB1 T5 CAVETWO
89: SITONTRAINTWOTOGETHER MB2 MB1 CAVETWO ENTRANCE
90: MOVEHOLDORETOGETHER MB2 MB1 ENTRANCE C1
91: LIFTORETOGETHER MB2 MB1 OB C1


time spent:    0.00 seconds instantiating 49924 easy, 0 hard action templates
               0.01 seconds reachability analysis, yielding 580 facts and 11320 actions
               0.00 seconds creating final representation with 482 relevant facts, 2 relevant fluents
               0.02 seconds computing LNF
               0.02 seconds building connectivity graph
              65.47 seconds searching, evaluating 100444 states, to a max depth of 44
              65.52 seconds total time
```

Originally, I want to realize the capacity and lift of rock, but if I add them to the action, it will be huge amount of work for ff. Thus, the problem is simplified with automatic rock collecting train and no capacity of the train.