A Study of Three Vacuum Cleaner Agent: Simple Deterministic Reflex Agent, Random Reflex Agent, and Deterministic Model-based Reflex Agent

Abstract

Three different vacuum cleaning agents, a simple memory-less deterministic reflex agent, a randomized reflex agent, and a deterministic model-based agent with memory, were designed and implemented for an experiment to compare their performance and learn their characteristic differences. A simple memory-less deterministic reflex agent decides its action only by its percepts from sensors such as a wall sensor and a dirt sensor. However, a randomized reflex agent decides its action just randomly. With internal state, a deterministic model-based reflex agent maintains 3 bits of memory, and the memory helps the agent chooses an action. They were tested in two different environments, and their total action count is 2000, and each agent ran 50 times. As the result, a deterministic model-based reflex agent showed the best performance followed by a simple memory-less deterministic reflex agent. We found that the performance of deterministic agents is better than a randomized reflex agent in a single-agent environment. Based on these findings, we discussed the future design of a deterministic model-based reflex agent for more complex environments.

I. Introduction

As per the description of PA1: Vacuum cleaner agent, we designed and implemented three different vacuum-cleaning agents: A simple memory-less deterministic reflex agent, a randomized reflex agent, and a deterministic model-based agent with memory. The objective of this experiment is to compare the performance of each agent and figuring out the difference between a deterministic and a stochastic agent programs. There are 3 percepts with a wall sensor, a dirt sensor, and a home sensor for each agent. In this experiment, a home sensor was only used when an agent starts because we ignored turn-off action. Thus, there are 4 main actions: go forward, turn right, turn left, and suck up dirt. Also, we designed 2 different environments by dividing the space. Based on it, we specify each agent's design strategy in the second section, then the performance of each agent will be analyzed in the third section. In the last section, we will discuss the agent design strategy for more complex environment and our findings.

II. Agent & Program Design

In order to create the vacuum-cleaner world, we considered three different kinds of intelligent agents. The brief description of each agent and its program algorithm are introduced as follow.

1. A Simple Memory-less Deterministic Reflex Agent

In this agent, its program decides action only by its sensors: a dust sensor and a wall sensor. Below is the its sudo-code with if-then rule.

```
if (dirt sensor detects dirt) {
       clean dirt
}
else {
              if(wall sensor detects wall)
                     turn left or turn right // (50% probability)
              else if(wall sensor does not detect wall)
                     go forward or turn left or turn right // (33.3 % probability)
}
```

2. A Randomized Reflex Agent

We simply designed a randomized reflex agent, literally saying, randomize its action no matter what it perceives. So, each action, go forward, turn right, turn left, and suck, has the same probability, 25%.

```
if (random action == suck up)
              suck up
else if (random action == go forward)
       if (no wall)
                  go forward
else if (random action == turn left)
              turn left
else if (random action == turn right)
              turn right

// the probability of each action is 25%
```

3. A Deterministic Model-Based Agent with Memory

This agent should maintain an internal state related to statements, “how the world evolves” and “what my actions do.” It becomes a model of the world. UPDATE-STATE is the critical part of this agent. Thus, we developed 3 bits memory to record the current state and its previous action as follow:

```
Struct flag{
    int a:1; // dust state on current grid a:0 clean, a:1 dirty
    int b:1;
    int c:1;
    // bc:00 previous action is ‘turn left’
    // bc:01 previous action is ‘turn right’
    // bc:10 previous action is ‘go forward’
}
```

Below is the model-based agent’s sudo-code designed for this report.

```
if (dirt sensor detects dirt) {
       clean dirt
}
else {
       if (previous action was turn left) {
              if (wall sensor detects wall)
                      turn left
              else if (wall sensor does not detect wall)
                     go forward or turn left // (50% probability)
       }
       else if (previous action was turn right) {
              if (wall sensor detects wall)
                     turn right
              else if (wall sensor does not detect wall)
                     go forward or turn right // (50% probability)
       }
       else if (previous action was go forward) {
              if (wall sensor detects wall)
                     turn left or turn right // (50% probability)
              else if (wall sensor does not detect wall)
                     go forward or turn left or turn right // (33.3% probability)
       }
}
// record state: dust sensor and current action in the memory
```

4. Two Different Environments

In order to test the performance of each agent, two environments were created. Basically, the vacuum cleaner environment consists of 10 X 10 grid. The first environment is the empty grid, but the second environment consists of 4 divided rooms with one grid cell doors in between every pair of adjacent rooms. Below are the presentation of two environments:

```
Dirt: 1, Clean: 0
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
```

(No Wall Environment)

```
Dirt: 1, Clean: 0, Wall: 2, Door: 3
1 1 1 1 1 2 1 1 1 1 1
1 1 1 1 1 2 1 1 1 1 1
1 1 1 1 1 3 1 1 1 1 1
1 1 1 1 1 2 1 1 1 1 1
1 1 1 1 1 2 1 1 1 1 1
2 2 3 2 2 2 2 2 3 2 2
1 1 1 1 1 2 1 1 1 1 1
1 1 1 1 1 2 1 1 1 1 1
1 1 1 1 1 3 1 1 1 1 1
1 1 1 1 1 2 1 1 1 1 1
1 1 1 1 1 2 1 1 1 1 1
```

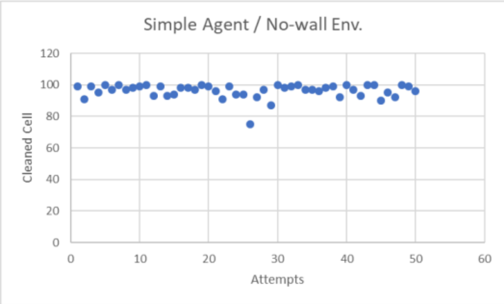
(4 Room Environment)

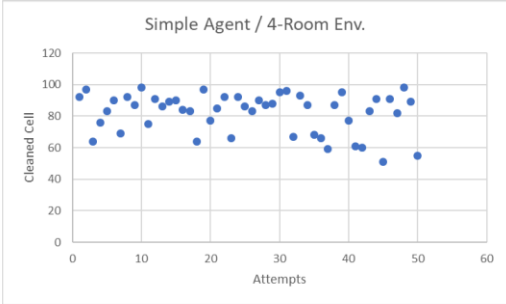
III. Test Result Analysis

For the improved performance of the random agent, we set action_threshold for the maximum action count as 2000, and we applied it to all agents for the fair comparison. In addition, each agent ran 50 times repeatedly.

1. A Simple Memory-less Deterministic Reflex Agent

The best performance of the no-wall environment was 100 % cleanness after 1504 actions. The agent cleaned the 4-room environment 98%. After 50 trials, the no-wall environment was cleaned 100 % 10 times (cleaned cell mean: 96.18, min: 75) , the agent in the 4-room environment cleaned perfectly only once (cleaned cell mean: 82.08 , min: 51).

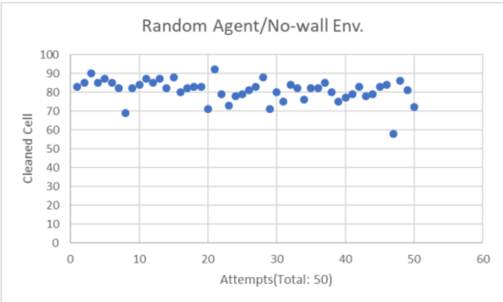


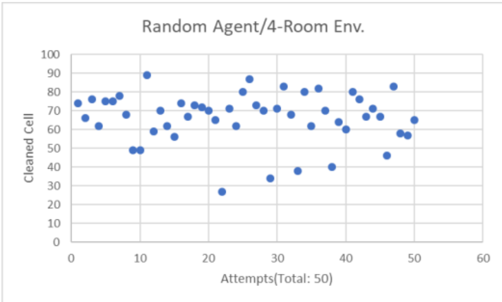


This result shows that the existence of the wall and the limited access to the next room through one grid cell door might have prevented it from achieving the goal of cleaning the room perfectly.

2. A Randomized Reflex Agent

After experimenting 50 times, the best performance in the no-wall environment was 92% cleanness (cleaned cell mean: 80.9, min: 58), and 89% cleanness was the best performance in the 4-room environment (cleaned cell mean: 66.42, min: 27).



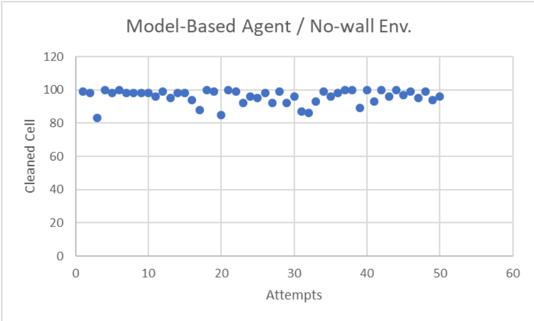


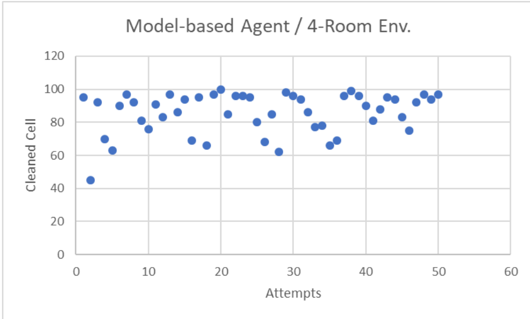
In order to see its performance more specifically, we ran the random agent in both environments for 500 times and provided a table showing the number of actions it to clean 90% of the room for each trial (See **Appendix A**). In the no-wall environment, the agent cleaned 90% of the room 18 times, but the agent never cleaned 90% of the room in the 4-room environment(The best performance of it was 89 cleaned cells). Both of them never cleaned 100% of the room. The average of the number of cleaned cell of the best 45 trials are 89.6 and 83.4 respectively.

As mentioned, we increased the total action counts from 1000 to 2000. Before this parameter tuning, overall percentage of room cleanness was between 50% to 60%. Thus, the increment of the total action count has an effect on the performance improvement. We believe that if we simply increase the number of action counts would change the performance of the agent in both environment. However, in the 4-room environment, the wall and doors seems to have significant effects on agent's performance. In our experiment, we divided the grid evenly into 4 rooms. Some result showed that the agent suffered from moving to the next room because of the narrow door size, one cell grid. Therefore, if we divide 4-rooms into different sizes to increase the probability of visiting, it would be helpful to increase the performance. Also, if we change the probability of each action, we could expect the performance improvement.

3. A Deterministic Model-Based Agent with Memory

The best performance of the no-wall environment was 100 % cleanness after 1365 actions. The agent cleaned the 4-room environment 100% after 1499 actions. After 50 trials, the no-wall environment was cleaned 100 % 9 times (cleaned cell mean: 95.96, min: 83) , the agent in the 4-room environment cleaned perfectly only once (cleaned cell mean: 85.74 , min: 45).





We used three bits of memory to update the current state and the previous action. If we can increase the size of memory as much as the agent can store the previous location and the event of suck-up, the performance would be improved because the agent can choose the best action referring to the previous action choice.

IV. Discussion & Conclusion

Often, infinite loop is a problem for a deterministic simple reflex agent like an infinitely moving vacuum cleaner. So, a randomized simple reflex agent might show the better performance than the deterministic one. However, a randomized simple reflex agent is usually not rational in single-agent environment. Our experiments also shows that a simple deterministic flex agent's performance is better than the randomized simple reflex agent since both environments have only one agent. Therefore, even in more complex environments with polygonal obstacles, a deterministic agent should be a better option. Corresponding more complex environments, refining and strengthening the deterministic agent are needed. We believe that a deterministic model-based agent could be the best option among above three agents for that environments.

While experiments, we found interesting results; when an agent does not remember the previous action, if any action is not chosen by random probability, there must be uncleaned cells. Also, when we set rules that if there is no wall in front of an agent, it chooses 'go forward,' and if there is a wall, it chooses 'turn right,' it just cleaned the border of the room. We tried to figure out the exact reason of these results but failed. However, we believe that paradoxically saying, this is the reason why the agent should be designed more sophisticated and intelligently.

Appendix A.

	Random Agent with No-Wall			Random Agent with 4 Rooms		
Top 45	**action count**	**go_forward (could not go)**	**cleaned cell**	**action count**	**go_forward (could not go)**	**cleaned cell**
1	1939	61	94	1868	132	89
2	1934	66	93	1854	146	88
3	1927	73	92	1846	154	88
4	1904	96	92	1869	131	88
5	1934	66	92	1831	169	88
6	1965	35	91	1882	118	87
7	1889	111	91	1848	152	87
8	1880	120	91	1868	132	87
9	1919	81	91	1867	133	87
10	1931	69	91	1854	146	86
11	1920	80	91	1840	160	86
12	1930	70	91	1834	166	85
13	1917	83	90	1872	128	85
14	1909	91	90	1862	138	84
15	1927	73	90	1869	131	84
16	1909	91	90	1865	135	84
17	1920	80	90	1851	149	84
18	1902	98	90	1860	140	83
19	1923	77	89	1878	122	83
20	1924	76	89	1873	127	83
21	1929	71	89	1856	144	83
22	1927	73	89	1895	105	83
23	1887	113	89	1837	163	83
24	1899	101	89	1840	160	83
25	1895	105	89	1827	173	83
26	1921	79	89	1846	154	82
27	1931	69	89	1837	163	82
28	1929	71	89	1855	145	82
29	1934	66	89	1851	149	82
30	1904	96	89	1846	154	82
31	1934	66	89	1873	127	82
32	1921	79	89	1863	137	82
33	1914	86	89	1838	162	82
34	1891	109	89	1820	180	82
35	1879	121	88	1886	114	81
36	1908	92	88	1840	160	81
37	1888	112	88	1870	130	81
38	1887	113	88	1841	159	81
39	1914	86	88	1860	140	81
40	1915	85	88	1831	169	80
41	1916	84	88	1853	147	80
42	1936	64	88	1851	149	80
43	1917	83	88	1860	140	80
44	1910	90	88	1830	170	80
45	1927	73	88	1845	155	80
mean	**1915.911**	**84.09**	**89.6**	**mean**	**145.73**	**83.42**