

多智能体系统编程作业一

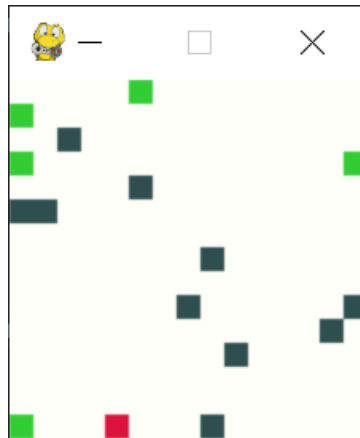
201300005 俞睿 人工智能学院

1 作业概述

本次作业中，我们复现了Commitment and Effective of Situated Agents这篇论文的实验结果。在这篇论文里，作者在Tileworld的环境里探索了世界变化率(rate of world change)、智能体规划时间(agent planning time)、勇敢程度(Degree of commitment)、反应策略(Reaction strategy)对智能体效果的影响。

在论文的复现中，有如下几个特点：

- 实现了论文中的所有设置，补充了部分细节，复现结果与论文较为一致。
- 代码实现上有参数可定制、可复用性好、可扩展性好、可读性强等优点，使得项目在开发、调试和应用过程中都能够更加方便、高效和灵活地满足不同需求。
- 基于PyGame实现了过程的可视化，可以直观地观察智能体与环境交互的过程。（如下图，灰色是障碍物，绿色是洞口，红色是智能体）



2 安装运行

在项目文件夹下执行：

```
pip install -r requirements.txt
python main.py # 运行一局游戏
python test.py # 复现论文实验结果
python display.py # 可视化
```

3 代码实现

3.1 项目架构

```
.
├─ dijkstra.py # 使用Dijkstra算法寻找最小路径
├─ display.py # 可视化智能体与环境的交互
├─ fig        # 复现的结果图片
├─ main.py    # 主函数，运行一局游戏
├─ requirements.txt
├─ res        # 运行结果
├─ test.py    # 复现论文实验结果的实验
└─ tileworld.py # Agent及Tileworld环境的实现
```

3.2 参数设置

- `grid_size`: Tileworld地图的大小, 默认为 15×15
- `n_obstacles`: 障碍物的数量, 默认为10
- `gest_range`: 洞孕育时间的范围, 默认为 $[60, 240]$
- `lt_range`: 洞的寿命, 默认为 $[240, 960]$
- `score_range`: 洞分数的范围, 默认为 $[1, 5]$
- `max_time`: 一局中最长的运行时间, 默认为1000

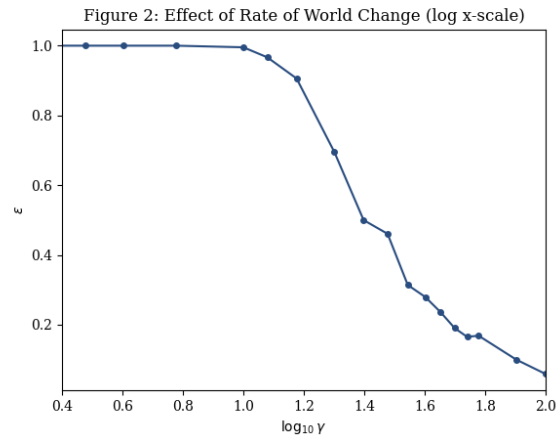
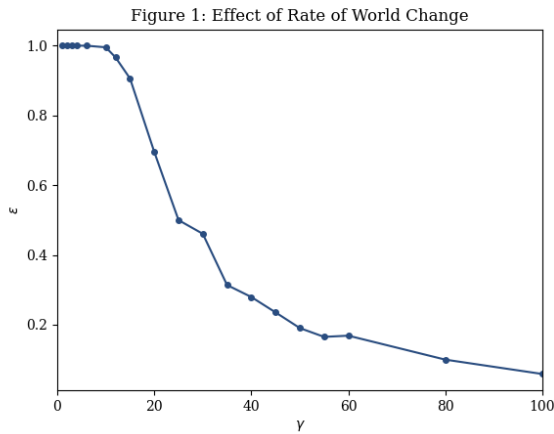
4 复现结果

4.1 世界变化率(rate of world change)的影响

在论文的设定中, 每个洞的特征是: 孕育时间(Gestation g), 寿命(Life-expectancy l)和分数。在基线情况下, 孕育时间从 $[60, 240]$ 中随机均匀采样, 寿命从 $[240, 960]$ 中随机均匀采样。我们这么定义世界变化率的影响: 当世界变化率为 γ 时, 孕育时间从 $[60/\gamma, 240/\gamma]$ 中随机均匀采样, 寿命从 $[240/\gamma, 960/\gamma]$ 中随机均匀采样。分数 α 从 $[1, 5]$ 随机均匀采样。

Tileworld环境每隔 g 在随机位置生成一个分数为 α 的洞, 这个洞将在 l 时间后堙灭。 ϵ 的计算方法是: 拿智能体吃掉的洞的分数之和除以所有生成的洞的分数之和。

我们可以观察到, 随着世界变化率 γ 的增加, ϵ 递减, 且趋势和原论文一致。



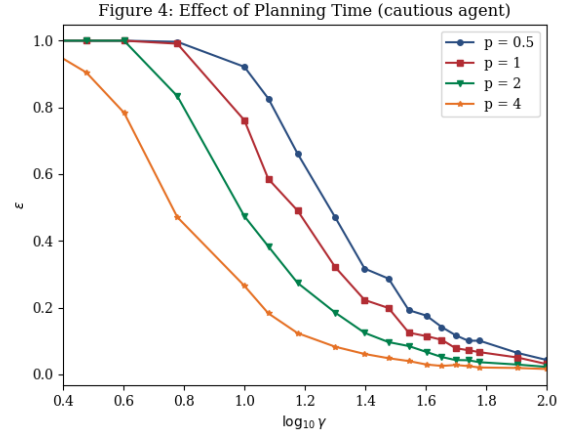
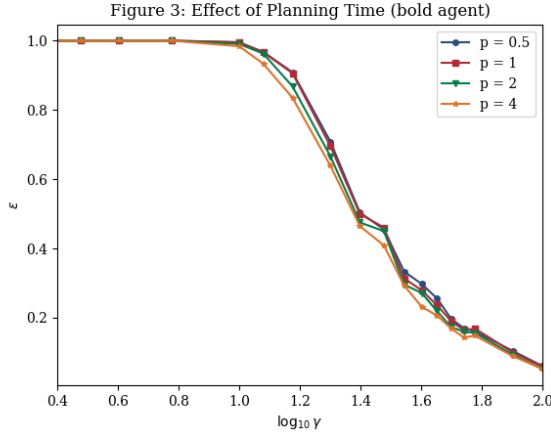
4.2 智能体规划时间(agent planning time)的影响

参考论文, 我们这么定义智能体挪动单步的时间:

$$t = \frac{p}{k} + m$$

这个 p 就是智能体规划时间, k 的设定即智能体每隔多少步重新考虑其选项, 这决定了代理程序的行为, 也就是4.3中的勇敢程度, m 是单步移动的时间, 在本文设定中一直为1。

对于不同的智能体规划时间, 我们发现对于Bold Agent和Cautious Agent来说, 均是规划时间越短效果越好, 且趋势和原论文一致。



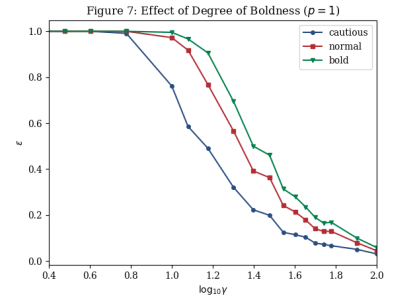
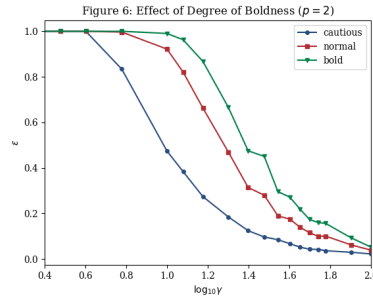
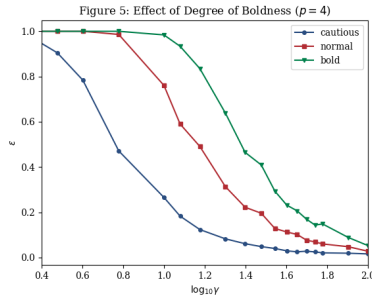
4.3 勇敢程度(Degree of commitment)的影响

我们这样设定勇敢程度：

- $k = 1$ ，即每走一步都要考虑重新考虑选项，为Cautious agent
- $k = 4$ ，即每走4步都要考虑重新考虑选项，为Normal agent
- $k = 30$ ，由于30已经是地图边长的两倍，我们近似地认为这样Agent直到走完整条路径都不会重新考虑选项，为Bold Agent

这里重新考虑会花费更多的时间，但是可以找到更优的目标，存在一个tradeoff的关系。

我们观察到，总体来说效果Bold agent > Normal agent > Cautious agent，且趋势和原论文一致。因为Cautious agent会花费更多的时间用于规划。我们应该设置更为平衡的 k 来达成最好的效果。



4.4 反应策略(Reaction strategy)的影响

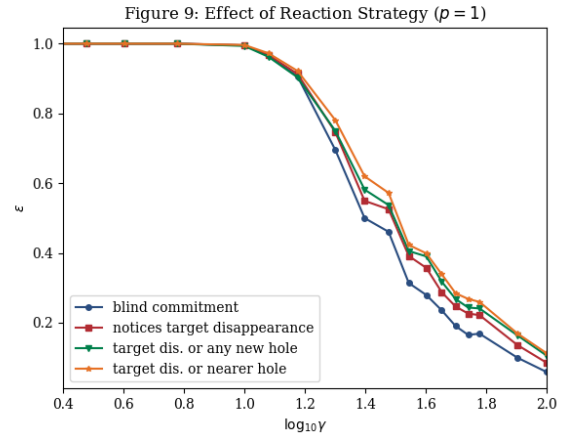
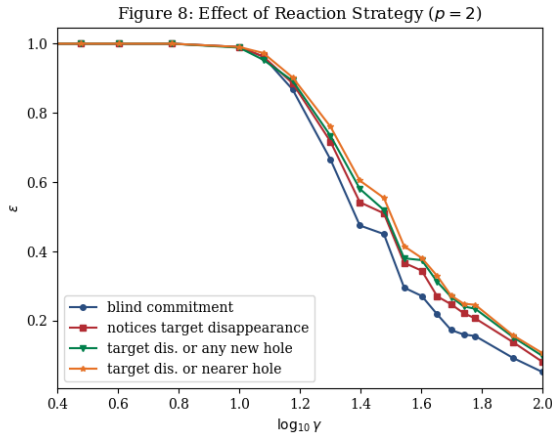
我们参考论文，复现了四种反应策略：

- Blind commitment: 一旦确定目标，直到达到目标的位置都不会重新考虑目标
- Notices target disappearance: 如果目标消失了则重新考虑目标
- Target disappearance or any new hole: 如果目标消失了或有新的目标出现，则重新考虑目标
- Target disappearance or nearer hole: 如果目标消失了或出现了更近的目标（用曼哈顿距离估算目标），则重新考虑目标

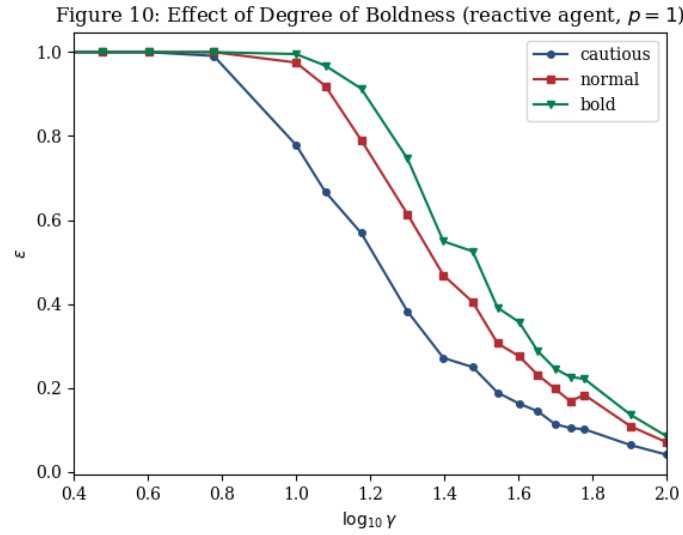
重新考虑时，我们会综合考虑距离洞的位置、洞的年龄和洞的分数。我们自己定义公式为：

$$(dist_score \times dist_weight + age_score \times age_weight) \times hole_score$$

我们观察到，总体来说效果near_hole > new_hole > disappear > blind，且趋势和原论文一致。



我们也讨论了在disappear策略下bold agent的运行效果，可见在更理性的策略类型下，一个更加谨慎的代理程序比一个勇敢的代理程序表现更出色。这点也和论文中一致。



5 Reference

- [1] David N. Kinny. Commitment and Effectiveness of Situated Agents. In IJCAI-91.