

Assignment Report

刘锦坤 2022013352 行健-烽火2班

- [Assignment Report](#)
 - [Single-Agent Search](#)
 - [Finding a Fixed Food Dot](#)
 - [深度优先搜索策略 \(DFS\)](#)
 - [广度优先搜索策略 \(BFS\)](#)
 - [一致代价搜索策略 \(UCS\)](#)
 - [A* with nullHeuristic策略](#)
 - [A* with manhattanHeuristic策略](#)
 - [Finding All the Corners](#)
 - [CornersProblem Search Problem](#)
 - [Corners Problem: Heuristic](#)
 - [Mutli-Agent Search](#)
 - [MinimaxAgent and AlphaBetaAgent](#)

Single-Agent Search

Finding a Fixed Food Dot

这一部分的任务是给定起始位置、地图形状、以及目标食物位置，找寻到从起始位置前往目标食物位置的路径。算法采用图搜索的方式，通过记录已探索节点避免对已探索节点的重复搜索，在搜索策略上分别应用深度优先搜索策略（DFS）、广度优先搜索策略（BFS）、一致代价搜索策略（UCS）、和A*搜索策略进行搜索。在mediumMaze中，各搜索策略的结果如下表所示：

Strategy	DFS	BFS	UCS	A* with nullHeuristic	A* with manhattanHeuristic
Nodes Expanded	144	267	267	267	221
Total Cost	130	68	68	68	68
Score	380	442	442	442	442

各搜索策略在mediumMaze中表现

深度优先搜索策略（DFS）

深度优先搜索策略优先向节点的后继节点搜索，空间复杂度较小，但是不能保证解的最优性，在迷宫寻径问题中，由于可以到达终点的路径较多，所以在本问题中深度优先搜索策略以最小的展开节点数114完成了路径搜索，但是由于深度优先搜索并不保证搜索结果的最优性，所以可以看到深度优先搜索策略所得结果的总花费为130，高于其他搜索策略搜索到的最优总花费68。

广度优先搜索策略 (BFS)

广度优先搜索策略优先搜索深度较小的这些节点，因为在本问题中路径的花费事实上等价于对应节点的深度，所以在本问题中广度优先搜索策略可以保证结果的最优性，所以在最后结果中，广度优先搜索通过展开267个节点搜寻到了总花费为68的最优路径。

一致代价搜索策略 (UCS)

一直代价搜索策略类似于广度优先搜索策略，但是优先展开待搜索节点中目前花费最小的那些节点，这个特性可以保证解的最优性。如之前所说，由于本问题中路径的花费等价于节点的深度，所以可以UCS和BFS在这个问题中是完全等价的，展开267个节点最终找到总花费为68的最优路径。

A* with nullHeuristic策略

不同于之前的搜索策略，A*搜索属于informed search的一种类型，由于额外信息的引入使得搜索更有方向性，在A* with nullHeuristic搜索策略中，由于启发式函数（heuristic）使用的是平庸（trivial）的 $h(n) = 1$ ，根据A*搜索的展开顺序 $f(n) = g(n) + h(n) = g(n) + 1$ 可以看出，采用平凡的启发式函数的A*搜索就等价于一致代价搜索，在本问题中也等价于广度优先搜索策略，均展开267个节点找到总花费为68的最优路径。

A* with manhattanHeuristic策略

这里引入曼哈顿距离作为启发式函数，容易看出，这样的启发式函数有admissible和consistent两个性质，因此在图搜索形式的A*搜索可以保证解的最优性，又由于额外信息的引入有效的减少了展开节点的个数，所以A* with manhattanHeuristic仅仅展开了221个节点（优于其他最优性算法267个节点）找到了总花费为68的最优路径。

Finding All the Corners

在CornersProblem里，地图的四个角都有食物，需要寻找路径吃掉四个角所有的食物。

CornersProblem Search Problem

这一问中要求完成searchAgents.py中的CornersProblem类，我这个类中定义了CornersProblem的state，其state由一个元组组成，元组内有两个元素，一个是当前pacman的位置，另一个是一个由bool型变量组成的列表cornersEaten，分别记录self.corners中各个角的变量是否被吃掉，当列表中全部为true时则为goalstate。getSuccessors方法则以(nextState, action, cost)的元组形式返回其后继动作、状态、花费。

Corners Problem: Heuristic

这一问要求设计一个满足admissible和consistent的启发式函数，基本思路是仍然采用曼哈顿函数，但是计算方式为先计算当前位置到最近食物的曼哈顿距离，然后计算到达最近食物后新的最近食物的曼哈顿距离，重复计算一直到到达最后一个食物，将各个曼哈顿距离相加即为启发式函数的值。经验证

其性质确实为admissible且consistent。最终在mediumMaze中仅展开691个节点即找到了最优路径，优于nullHeuristic的1921个节点。

Mutil-Agent Search

MinimaxAgent and AlphaBetaAgent

这一问中要实现多智能体的对抗搜索，但是略有不同的是，由于问题中ghost的个数有多个，因此在搜索树中有多个min层，因此在搜索minvalue的过程中需要告知当前是第几个min层，直到最后一个ghost选择完后才进入maxvalue函数。

在实际测试中主要有以下发现：

- 采用alphabeta剪枝后，搜索的速度明显高于普通的minimaxsearch，这体现在其在同等下探深度下，agent每走一步的时间变短了。
- 由于下探深度有限（或者少数问题本身无解），pacman在部分情况下仍然会被吃掉。
- 由于评估函数不够合理，在地图较大、附近没有ghost即将吃掉pacman的情况下，pacman会停在原地不动，在附近没有食物时这种情况尤为明显，这是由于评估函数并没有很好的指引pacman兼顾吃东西和逃命，显然默认的评估函数过于重视逃命，pacman成为了胆小的pacman。
- 然后这个算法在一些情况下会主动寻思，这是因为minimax算法默认以最坏的情况考虑问题，但是事实上ghost并不总是向着对于pacman最坏的方向移动，这就使得pacman在本还可挣扎的情况下自杀了。