盲目搜索实验报告

武自厚 20336014 保密管理

2022 年 3 月 28 日

本次实验选择 BFS 算法以及 IDS 算法进行分析并实现.

1 原理分析

1.1 搜索

几乎所有的搜索问题都可以形式化并抽象化为以下描述:

根据已知的初始状态、行动、前进成本以及目标状态, 通过各种算法获得从初始状态到某一个满足目标状态的状态序列.

其中, 具体的方式取决于算法采用的策略, 在这里 BFS 算法和 IDS 算法就具有一定区别. 但是都一定具有搜索的**边界**以便于寻找新的更靠近目标的状态, 即边界的扩展.

1.2 宽度优先搜索 (BFS)

每一次即将扩展的状态放在边界的最后, 也就是说使用队列数据结构来维护这个算法. 只有当深度更小的所有状态都扩展后才会扩展一个状态.

1.3 迭代加深搜索 (IDS)

迭代加深搜索的基础是深度优先搜索 (DFS), 即将扩展得到的状态置于边界的最前端, 也就是用栈维护这个策略. 而如果限制扩展的深度, 超过规定深度的状态不予加入边界, 这样的算法称为深度限制搜索 (DLS).

在上述两个算法的基础上, 迭代加深搜索可以描述为: 迭代执行深度为 $1, 2, 3, \ldots$ 的深度限制搜索, 直至查找出结果或者达到设定的最大深度, 返回 *false*.

2 效果分析

2.1 完备性分析

算法执行的结果记录于 `result.txt` 中, 可以发现两个算法给出了同样的结果, 经过验证, 这个结果是正确的. 而深度优先搜索和迭代加深搜索算法都具有完备性 (因为该问题是有限的而且行动代价一致), 所以对于任何有解搜索问题都可以找到答案.

2.2 时间复杂度分析

理论上, 设 b 为分支因子 (一个状态最多能扩展状态的数量), d 为最短解的动作数量. 则深度优先搜索的时间复杂度为 $O(b^d)$, 迭代加深搜索的时间复杂度为 $O(b^d)$.

而代码实际运行状态记录于 `result.txt`, BFS 算法花费约 0.004 秒, IDS 算法花费约 0.1 秒, 速度相差约 25 倍, 但是就问题的规模而言, 可以认为这两个算法处于同一时间复杂度.

2.3 空间复杂度分析

设 b 与 d 的含义与上文相同, 则在理论上, 深度优先搜索的空间复杂度为 $O(b^d)$, 而迭代加深搜索的空间复杂度为 $O(bd)$.

`bfs.txt` 和 `ids.txt` 分别记录了两种算法在扩展时边界的元素个数. 可以看到 BFS 算法在这个问题中最大容量是 9, 而 IDS 算法最大需要的容量是 7, 对于这个规模的问题而言, 可以说 IDS 算法显然优于 BFS 算法.

2.4 最优性分析

理论上, 对于行动成本一致的问题而言 (迷宫问题就具有这个性质),BFS 以及 IDS 算法都具有最优性. 因为在 BFS 算法中, 节点会按照深度升序扩展, 最先找到的路径必然是所有可能路径中深度最小的, 再加上行动成本一致的条件, 所以最先找到的路径即是最优路径. 而对于 IDS 算法, 由于每一次迭代会逐渐加深 DFS 的深度, 最先找到的路径也必然是所有路径中深度最小的, 也就是最优的.

3 思考题: 算法优缺点分析

3.1 宽度优先搜索 (BFS)

BFS 算法的优点在于易于实现, 在行动成本一致的时候具有最优性; 缺点是时间、空间复杂度都比较平庸.

介于它在行动成本一致时具有最优性, 它可以适用于类似游戏中 NPC 自动寻路的 “方格寻路” 问题.

3.2 深度优先搜索 (DFS)

DFS 算法的优点在于空间复杂度相较于其他算法来说非常小; 但缺点是容易 “不撞南墙不回头”, 即不能保证完备性和最优性.

这个算法适用于类似于 “走迷宫” 的 b 较小的问题.

3.3 深度受限搜索 (DLS)

DLS 算法具有 DFS 算法的优点, 且时间复杂度稍有降低; 但缺点依然是不能保证完备性和最优性.

这个算法依然适用于类似 “走迷宫” 的问题, 尤其是知道目标状态行动数量的时候.

3.4 迭代加深搜索 (IDS)

IDS 算法具有 DFS 算法的优点, 且在迭代过程中可以保证完备性, 在行动成本一致的时候具有最优性; 缺点是无法应用于行动成本不一致的问题, 即使采用 “成本边界” 代替, 也会造成极大重复浪费.

这个算法适用的场景类似于 BFS 算法的适用场景, 且行动成本需要一致.

3.5 统一代价搜索 (UCS)

UCS 算法的时间与空间复杂度都较为优秀, 且在行动成本不一致的问题中依然具有完备性和最优性; 缺点是在行动成本不一致的问题中会退化为 BFS 算法, 且由于优先队列弹出的时间复杂度大于队列的弹出, 此时时间复杂度甚至会大于 BFS 算法.

此算法适用于任何行动成本不一致的场景.

3.6 双向搜索

双向搜索算法的时间、空间复杂度都优于其他算法, 且在双向 BFS 中可以满足完备性和最优性; 缺点在于某些问题中可能很难实现向前搜索.

这个算法适用于类似于无向图寻路的便于向前搜索的问题.