

人工智能基础第一次课程实验

一、实验背景

你是某座城市的一个物流公司的技术人员，公司计划引进无人机进行配送快递的任务。无人机配送系统需要从仓库（起点S）出发，依次将包裹送到所有指定的配送点（B），并最终返回控制中心（终点T）。假设无人机的运动近似为二维平面，这个城市有一些高层建筑需要避开，我们把它当作障碍物。现在需要你设计无人机的运行路线，完成指定的配送任务并且返回控制中心。

二、问题描述

我们将配送地图简化为 $M \times N$ 的网格地图，你可以向上、下、左、右四个方向其中之一移动，并且只能移动到相邻的可通行区域。请注意，地图（城市）外没有信号，无人机无法飞行。在此过程中，你的无人机电量每运行一步会消耗一个电量，**假设总电量足够**。

地图上的网格含有五种类型：

- S：无人机飞行的起点（仓库）
- T：无人机到达的终点（控制中心）
- B：城市中的配送点
- #：高层障碍区域（不可通行）
- .

如：

	S		.		B		#	
	#		B		.		B	
	.		#		T		.	

你能否顺利完成全部配送并且回到控制中心？如果能，最小的耗电量是多少？

在本次实验中，你的**目标**是：

- 实现迭代加深的深度优先搜索算法，找到路径，使无人机能配送所有的快递并且返回控制中心；
- 实现A*算法，找到最优路径，使无人机能在损失电量最少的情况下配送所有的快递并且返回控制中心；
- 设置启发式函数为0，此时A*退化为一致代价搜索算法，比较并分析A*方法带来的优化效果。
- 比较迭代加深的深度优先搜索算法和A*算法生成的路径，分析最优性。

输入：

./Lab1/input/ 共有10个测试文件（除去input_0.txt），形式如下：

S.B#
#B.B
.#T.

这是一个3x4的网格地图，共有3个配送点。

输出：

结果输出到 `./Lab1/output/` 下，文件命名为 `output_(x).txt`，其中 `x` 表示对应第 `x` 个输入样例。如对于 `input_0.txt`，使用 A* 算法输出文件名为 `output_0_Astar.txt`，输出形式如下：

```
// 一行表示全部配送完并返回控制中心的总耗电量（取正值）
// 一行表示动作的字符串，一共有四种动作，U表示向上，L表示向左，R表示向右，D表示向下
8
RDRUDRLD
```

三、实验要求

1. 编程语言限制为 C/C++，写清核心代码注释。代码主体框架已在 `src/IDS.cpp` 和 `src/Astar.cpp` 中给出，说明：
 - 补全完成迭代加深的深度优先搜索部分、启发式函数定义部分、A* 搜索部分等标注有 TODO 标记代码。
 - 给出的代码任意处均可修改，同学们不受框架约束自行编程，只需保证最后程序产生正确的输出文件即可。
2. 报告中需要描述：
 - 迭代加深的深度优先搜索算法的主要思路。
 - 描述你的 A* 算法的启发式函数，证明它是 admissible 的，并论证其是否满足 consistent 性质。
 - A* 算法的主要思路；
 - 与一致代价搜索进行比较，并分析使用 A* 方法带来的优化效果。
 - 将迭代加深的深度优先搜索算法和 A* 算法生成的路径进行比较，分析最优性。
3. 严禁抄袭以及使用 AI 工具直接生成代码。允许使用 AI 工具辅助检查。
4. 本次实验 IDS 算法满分 40 分，A* 算法满分 60 分，附加实验满分 20 分，上限为 100 分。

四、实验提交

1. 提交方式：bb 系统中提交；
2. 截止日期：4 月 30 日晚 23:59；
3. 提交的目录树结构如下所示：

```
PB22000001_张三_lab1/
    |---src
    |   |---(your code)
    |---input
    |   |---input_0.txt
    |   |---...
    |---output
    |   |---output_0_Astar.txt
    |   |---...
```

请将文件压缩成 .zip 格式进行提交；

4. 请务必按时提交实验。

附加实验——蒙特卡洛树搜索(MCTS)

一、MCTS算法介绍

蒙特卡洛树搜索简而言之就是使用蒙特卡洛方法估算每一种走法的胜率，具体来说，是通过不断的模拟每一种走法，直至终局，该走法的模拟总次数 N ，与胜局次数 W ，即可推算出该走法的胜率为 W/N 。

蒙特卡罗树搜索大概可以被分成四步。选择 (Selection)， 拓展 (Expansion)， 模拟 (Simulation)， 反向传播 (Back Propagation)。在开始阶段，搜索树只有一个节点，也就是我们需要决策的局面。搜索树中的每一个节点包含了三个基本信息：代表的局面，被访问的次数，累计评分。

• 选择 (Selection)

在选择阶段，需要从根节点，也就是要做决策的局面 R 出发向下选择一个最急需需要被拓展的节点 N ，局面 R 是每一次迭代中第一个被检查的节点；对于被检查的局面而言，有三种可能：

1. 该节点所有可行动作都已经被拓展过
2. 该节点有可行动作还未被拓展过
3. 这个节点游戏已经结束了(例如已经连成五子的五子棋局面)

对于这三种可能：

1. 如果所有可行动作都已经被拓展过了，那么我们将使用UCT公式计算该节点所有子节点的UCT值，并找到值最大的一个子节点继续检查。反复向下迭代。
2. 如果被检查的局面依然存在没有被拓展的子节点(例如说某节点有20个可行动作，但是在搜索树中才创建了19个子节点)，那么我们认为这个节点就是本次迭代的目标节点 N ，并找出 N 还未被拓展的动作 A ，执行**拓展**。
3. 如果被检查到的节点是一个游戏已经结束的节点。那么从该节点直接执行**反向传播**。

每一个被检查的节点的被访问次数在这个阶段都会自增。在反复的迭代之后，我们将在搜索树的底端找到一个节点，来继续后面的步骤。

• 扩展 (Expansion)

在选择阶段结束时候，我们查找到了一个最迫切被拓展的节点 N ，以及他一个尚未拓展的动作 A 。在搜索树中创建一个新的节点 N_n 作为 N 的一个新子节点。 N_n 的局面就是节点 N 在执行了动作 A 之后的局面。

• 模拟 (Simulation)

为了让 N_n 得到一个初始的评分。我们从 N_n 开始，让游戏随机进行，直到得到一个游戏结局，这个结局将作为 N_n 的初始评分。一般使用胜利/失败来作为评分，只有1或者0。

• 回溯 (Back Propagation)

在 N_n 的模拟结束之后，它的父节点 N 以及从根节点到 N 的路径上的所有节点都会根据本次模拟的结果来添加自己的累计评分。如果在**选择**中直接发现了一个游戏结局的话，根据该结局来更新评分。每一次迭代都会拓展搜索树，随着迭代次数的增加，搜索树的规模也不断增加。当到了一定的迭代次数或者时间之后结束，选择根节点下最好的子节点作为本次决策的结果。

信任度上限树 (Upper Confidence bound applied to Trees (UCT))

上文中 N 表示总模拟次数， W 表示胜局次数。每次都选择胜率最大的节点进行模拟。但是这样会导致新节点无法被探索到。为了平衡探索与利用，UCT被引入： $w_i/n_i + c * \sqrt{\ln N_i / n_i}$ 。其中 w_i 是 i 节点的胜利次数， n_i 是节点的模拟次数， N_i 是所有模拟次数， c 是探索常数，理论值为 $\sqrt{2}$ ，可根据经验调整。公式的后半部分，探索次数越少，值会越大，所以，那些被探索比较少的点，会获得更多的探索机会。

二、问题描述

同上。你的**目标**是：尝试使用MCTS算法找到最优路径，使无人机能在损失电量最少的情况下配送所有的快递并且返回控制中心。

三、实验要求

1. 编程语言限制为 C/C++，写清核心代码注释。代码主体框架已在 src/MCTS.cpp 中给出。同学们可以不受框架约束自行编程，只需保证最后程序产生正确的输出文件即可。
2. 报告中需要描述：
 - 算法的主要思路；
 - 算法效果及调参分析。
3. 本次实验能够**正确运行并合理解释算法的思路及效果**即可。代码正确运行占10分，效果及调参分析占10分。
4. 严禁抄袭以及使用AI工具直接生成代码。允许使用 AI 工具辅助检查。