

3.14

- a.错误。由于 A*启发函数的不同，可能会存在深度优先搜索扩展结点少于 A* 结点。例如一个 n 层二叉树，解位于第 n 层的左边节点。对于深度优先搜索，其只需要探索 $2n-1$ 个结点，A*至少需要探索 $2n-1$ 个节点。于是此时深度优先搜索拓展的结点小于等于 A*探索的结点。
- b.正确。 $h(n)=0$ 满足非负的要求，此时等价于宽度优先搜索，对于八数码问题是可采纳的启发式。
- c.错误。尽管感知器、状态、行动是连续的，但是可以通过一定程度的离散化趋近连续，故 A*算法在机器人学中有用处。
- d.正确。宽度优先搜索在有限深度、有限状态情况下，总能保证寻找到最优路径，是完备的。
- e.错误。由于有车无法跳跃棋子，曼哈顿距离并不能作为可采纳的启发式。

3.16

a.形式化

状态：状态由轨道块的组合方式决定

初始状态：任选一个工件均可作为初始状态

行动：与下一个工件进行拼接(拐角对应，拼合无缝隙)

转移模型：行动会产生期待的后果

目标测试：所有零件拼接成铁路，无重叠的轨道，无松动

路径耗散：无

- b.深度优先搜索。由于问题有解，于是通过深度优先方式就能够得到最优解，只是最糟糕的情况变成了遍历求解，问题求解速度可能比较慢。
- c.要是想组成一张能够运行的铁路路线，那么所有的积木块紧密贴合，即图上的扇入等于扇出。图 3.32 一共有 $12+16+2*2+2$ 个扇出， $12+16+2+2*2$ 个扇入，此时扇入等于扇出，问题有解。如果拿出任何一个开叉块，扇入不等于扇出，问题无解。
- d.考虑最差的情况-遍历，当积木不可翻转时，一共有 $32!$ 种可能。由于翻转的作用，上界应为 $32!*2*2*(2*2)=8*32!$

3.20

a.深度优先搜索-图搜索

b.记有灰尘坐标位置为(1,1)(1,2)(1,3)，Agent 坐标(2,2)。走一步的代价为 1，清扫灰尘代价为 1，扫干净奖励 10。Agent 拥有全局探测灰尘的能力。

最优行动序列为:

(2,2)->(2,3)->(1,3)->(1,2)->(1,1)

(2,2)->(2,1)->(1,1)->(1,2)->(1,3)

走路总代价为 4，清扫代价为 3，奖励 30。性能 23

c.随即生成灰尘坐标位置，仍为 b 中的(1,1)(1,2)(1,3)，agent 坐标(2,2)。走一步的代价为 1。当前 agent 只能搜索以自身为中心 3*3 的灰尘位置。初始时进行搜索，当清扫完上一个时间搜索的地方时，进行下一次搜索。每次搜索代价为 2，走一步代价为 1，清扫灰尘代价为 1，扫干净奖励 10。

最优行动序列为:

(2,2)->(2,3)->(1,3)->(1,2)->(1,1)

(2,2)->(2,1)->(1,1)->(1,2)->(1,3)

走路总代价为 4，清扫代价为 3，搜索代价为 4，奖励 30。性能 19

d.设环境仍是 3*3，最好的搜索 agent 性能为 19。

由于搜索 agent 走了 4 步，假设简单的随机反射型 agent 也走了 4 步，那么性能为 23(性能最好)的概率为 $\frac{2}{4^4}$ ，性能为-4 的概率至少为 $\frac{1}{2}$ ，则性能期望一定小于 19。在这种情况下，最好的搜索 agent 性能高于简单的随机反射型 agent。

e.当为 $n*n$ 时，灰尘的状态空间有 2^{n^2} 。当 $n \rightarrow \infty$ 时，通过搜索 agent 求解复杂度非常高。此时随机反射型 agent 的效率会比较高。

3.25

当 $w=1$ 时，算法是最优的。

$W=0$ 时，算法等价于不计路径耗散的搜索

$W=1$ 时，算法等价于 A*算法

$W=2$ 时，算法等价于贪婪算法