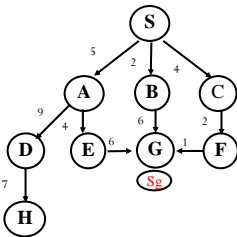


人工智能概论

慕彩红
人工智能学院
mucaihongxd@foxmail.com

第二章 状态空间表示
与问题求解

例



3

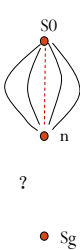
第二章 状态空间表示
与问题求解

- 2.1 问题求解与状态空间表示法
- 2.2 图搜索策略
- 2.3 盲目式搜索
- 2.4 启发式搜索

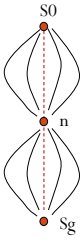
启发式搜索

- 为什么需要启发式搜索?
- ◇盲目搜索的不足
- 效率低, 耗费过多的计算空间与时间
- 可能带来组合爆炸
- 分析前面介绍的宽度优先、深度优先搜索, 或等代价搜索算法, 其主要的差别是OPEN表中待扩展节点的顺序问题。人们就试图找到一种方法用于排列待扩展节点的顺序, 即选择最有希望的节点加以扩展, 那么, 搜索效率将会大为提高。
- ◇什么可以做为启发信息?
- 一般需要某些有关具体问题的领域的特性信息, 把此种信息叫做启发信息。

盲目式搜索



启发式搜索



启发式信息

- 启发性信息是指那种与具体问题求解过程有关的，并可指导搜索过程朝着**最有希望方向前进**的控制信息。
- 启发式就是要猜测：
 - 从节点n开始，找到最优解的可能性有多大？
 - 从起始节点开始，经过节点n，到达目标节点的最佳路径的费用是多少？

启发式搜索

- 启发式搜索是利用与问题有关的启发性信息，并以这些启发性信息指导的搜索的问题求解过程。
 - 需定义一个评价函数，对当前的搜索状态进行评估，找出一个最有希望的节点来扩展。
 - 重排OPEN表,选择**最有希望的节点**加以扩展
- 种类:A算法、A*算法等

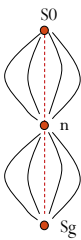
A算法

➢ 估价函数 $f(n)$ ：估算节点“希望”程度的量度

$$f(n) = g(n) + h(n)$$

从起始状态
到当前状态n
的代价

从当前状态到目
标状态的估计代
价 (启发函数)



- A算法
- 局部择优 (尼尔逊, 1964)
 - 有序搜索 (拉斐尔, 1967)

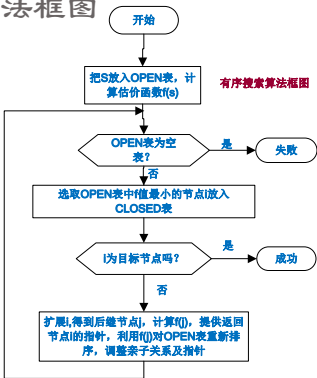
局部择优搜索(瞎子爬山法)

- 搜索过程如下：
 - (1) 把初始节点 S_0 放入OPEN表，计算 $f(S_0)$ ；
 - (2) 如果OPEN表为空，则问题无解，退出；
 - (3) 把OPEN表的第一个节点(记为节点n)取出，放入CLOSED表；
 - (4) 考查节点n是否为目标节点，若是，则求得了问题的解，退出；
 - (5) 若节点n不可扩展，则转第(2)步；
 - (6) 扩展节点n，用估价函数 $f(x)$ 计算每个子节点的估价值，并按估价值从小到大的顺序依次放入OPEN表的首部，并为每一个子节点都配置指向父节点的指针，转第(2)步。

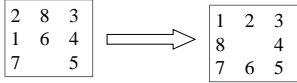
有序搜索

- 概念
- ◇ **有序搜索** (ordered search)，也叫全局择优搜索或最好优先搜索 (best-first search)。
- ◇ 选择OPEN表上**具有最小f值**的节点作为下一个要扩展的节点。

有序搜索算法框图



一个算法的例子



定义评价函数：

$f(n) = g(n) + h(n)$

$g(n)$ 为从初始节点到当前节点的路径长度（深度）

$h(n)$ 为当前节点“不在位”的将牌数

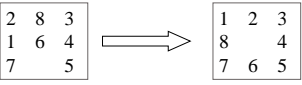
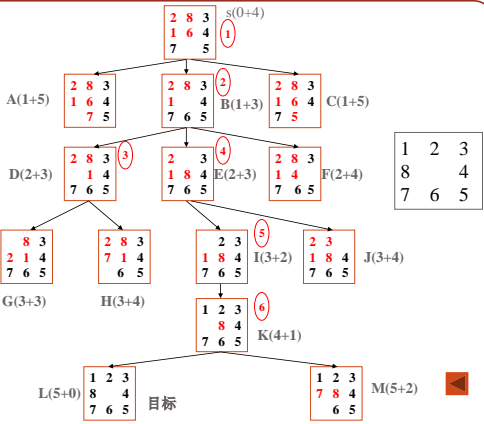
h计算举例



$g(n)=0$

$h(n)=4$

$f(n)=4+0=4$



	宽度搜索	深度搜索（界为5）	有序搜索
扩展节点	26	18	6
生成节点	46	34	13

A*算法

- 1968年，彼得·哈特（PETER E. HART）对A算法进行了很小的修改，并证明了当估价函数满足一定的限制条件时，算法一定可以找到最优解。
- 隶属于A算法，其特点在于对估价函数的定义上

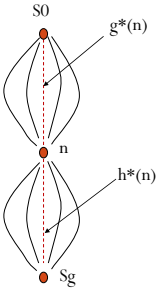
➤ A*算法的限制条件

$f(n) = g(n) + h(n)$

要满足一定条件

A*算法

- 把OPEN表中的节点按估价函数的值从小到大进行排序
- $f(n) = g(n) + h(n)$
- $g^*(n)$ 是从初始节点 S_0 到任意节点 n 的一条最佳路径的代价
- $h^*(n)$ 是从节点 n 到目标节点的一条最佳路径的代价
- $g(n)$ 是对 $g^*(n)$ 的估计， $g(n) \geq g^*(n)$
- $h(n)$ 是 $h^*(n)$ 的下界，即对所有 n 的， $h(n) \leq h^*(n)$



A*算法

定义1

- 在图搜索过程中，如果重排OPEN表是依据 $f(n)=g(n)+h(n)$ 进行的，则称该过程为A*算法；

定义2

- 在A*算法中，如果对所有的n存在 $h(n) \leq h^*(n)$ ，则称 $h(n)$ 为 $h^*(n)$ 的下界，它表示某种偏于保守的估计；

定义3

采用 $h^*(n)$ 的下界 $h(n)$ 为启发函数的A*算法，称为A*算法。

A*算法过程

- 把S放入OPEN表，记 $f=h$ ，令CLOSED为空表。
- 重复下列过程，直至找到目标节点止。若OPEN表为空，则宣告失败。
- 选取OPEN表中未设置过的具有最小值的节点为最佳节点BESTNODE，并把它放入CLOSED表。
- 若BESTNODE为一目标节点，则成功求得一解。
- 若BESTNODE不是目标节点，则扩展之，产生后继节点SUCCSSOR。
- 对每个SUCCSSOR进行下列过程：
 - 建立从SUCCSSOR返回BESTNODE的指针。
 - 计算 $g(SUC)=g(BES)+k(BES,SUC)$ 。
 - 如果SUCCSSOR \in OPEN，则称此节点为OLD，并把它添至BESTNODE的后继节点表中。
 - 比较新旧路径代价。如果 $g(SUC) < g(OLD)$ ，则重新确定OLD的父节点为BESTNODE，记下较小代价 $g(OLD)$ ，并修正 $f(OLD)$ 值。
 - 若至OLD节点的代价较低或一样，则停止扩展节点。
 - 若SUCCSSOR不在OPEN表中，则看其是否在CLOSED表中。
 - 若SUCCSSOR在CLOSE表中，则转向c。
 - 若SUCCSSOR既不在OPEN表中，又不在CLOSE表中，则把它放入OPEN表中，并添入BESTNODE后商表，然后转向(7)。
- 计算 f 值。
- GO LOOP

例题

- 对于如图所示的八数码问题，给出满足A*算法的启发函数，并给出相应的搜索图。

$S_0 =$

2	8	3
1	6	4
7		5

$S_g =$

1	2	3
8		4
7	6	5

21

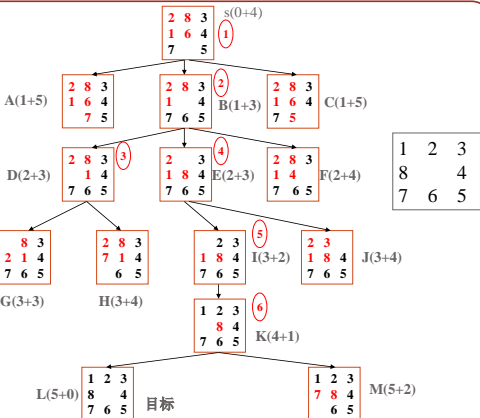
$S_0 =$

2	8	3
1	6	4
7		5

$S_g =$

1	2	3
8		4
7	6	5

- 方案一：
- 启发函数的选取如下： $g(n)$ 表示节点n在搜索树中的深度， $h(n)=\omega(n)$ 表示节点n中“不在位”的将牌数。
- 在上面确定 $h(n)$ 时，尽管并不知道 $h^*(n)$ 具体为多少，但当采用单位代价时，通过对“不在位”的将牌数的估计，可以得出至少需要移动 $h(n)$ 步才能够到达目标，显然 $h(n) \leq h^*(n)$ 。因此它满足A*算法的要求。



S

2	8	3
1	6	4
7		5

S

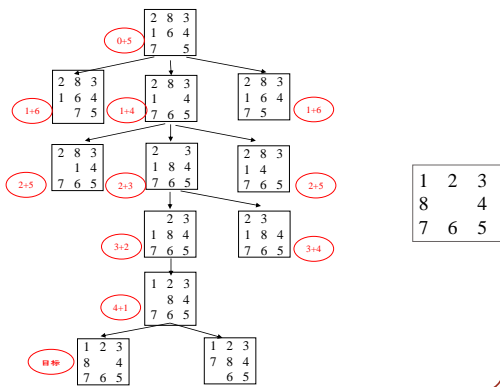
g

1	2	3
8		4
7	6	5

g

- 方案二：
 - 定义启发函数 $h(n)=p(n)$ 为节点n的每一数码与其目标位置之间的距离总和。
- 1 2 3
- 2 8 3
- 8 1 6 4 4
- 7 5 5
- 7 6
- 不在位将牌数
- 将牌 1 1
- 将牌 2 1
- 将牌 6 1
- 将牌 8 2
- $\omega(S)=4$
- $p(S)=5$
- 显然有 $\omega(n) \leq p(n) \leq h^*(n)$ ，相应的搜索过程也是A*算法。

A*算法



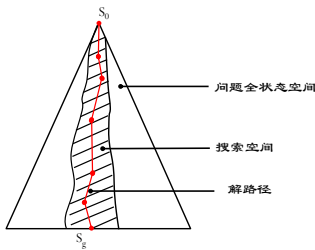
下面给出本题的启发函数的比较结果。

启发函数	$h(n)=0$	$h(n)=\omega(n)$	$h(n)=p(n)$
扩展节点	26	6	5
生成节点	46	13	11

$\omega(n) \leq p(n) \leq h^*(n)$
 $p(n)$ 比 $\omega(n)$ 有更强的启发性信息，由 $h(n)=p(n)$ 构造的启发式搜索树，比 $h(n)=\omega(n)$ 构造的启发式搜索树节点数要少。

➤ 搜索方法好的标准，一般认为有3个：

- (1) 问题有解能否找到
- (2) 搜索空间小
- (3) 解最佳



启发式搜索小结

- 启发式策略就是利用与问题有关的启发信息进行搜索的策略。
- 利用估价函数来估计待搜索节点的希望程度，并以此排序。估价函数一般由两部分组成： $f(n)=g(n)+h(n)$
- $g(n)$ 是从初始节点到节点n已付出的实际代价， $h(n)$ 是从节点n到目的节点的最佳路径的估计代价。 $h(n)$ 体现了搜索的启发信息。
- 在 $f(n)$ 中， $g(n)$ 的比重越大，越倾向于宽度优先搜索，而 $h(n)$ 的比重越大，表示启发性越强。

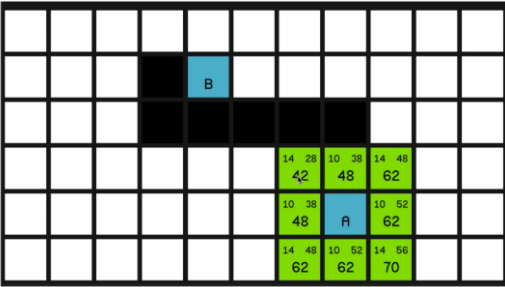
启发式搜索小结

- $f(n)=g(n)+h(n)$
- 在 $f(n)$ 中， $g(n)$ 的比重越大，越倾向于宽度优先搜索，而 $h(n)$ 的比重越大，表示启发性越强。
- $g(n)$ 的作用一般是不可忽略的，保持 $g(n)$ 项就保持了搜索的宽度优先成分，这有利于搜索的完备性，但会影响搜索的效率。

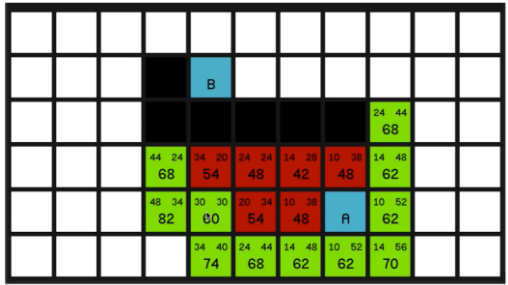
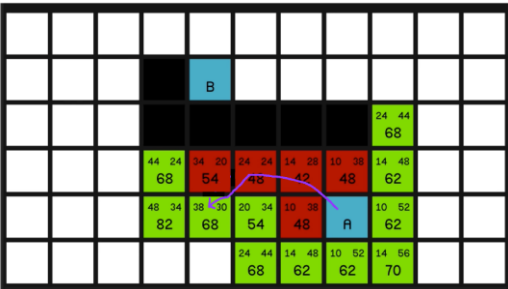
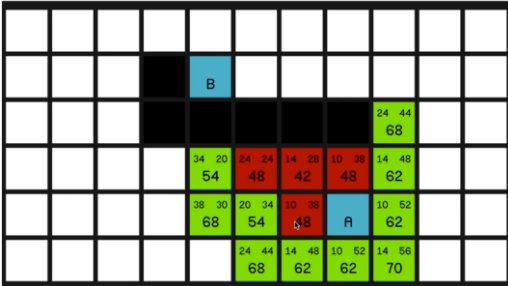
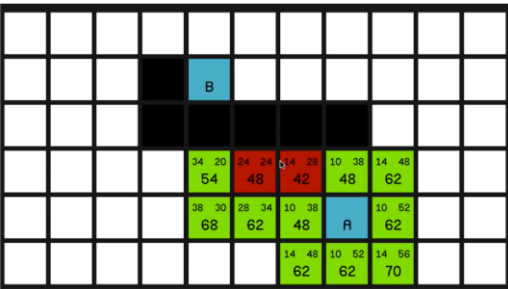
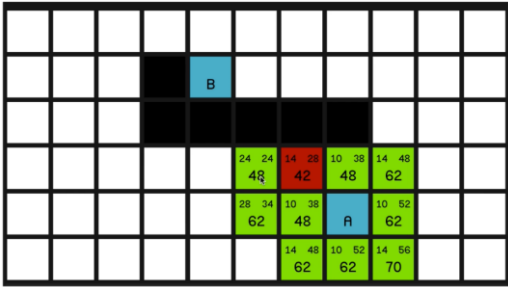
启发式搜索小结

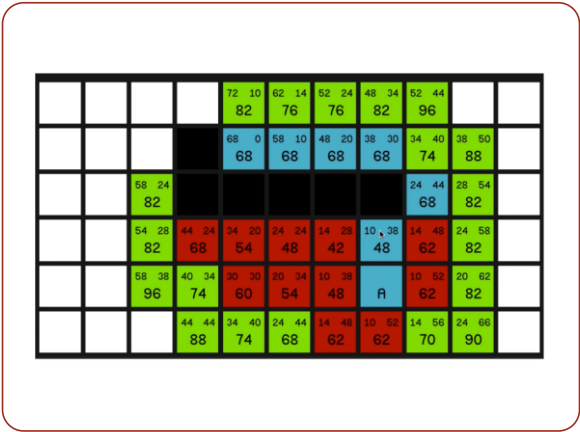
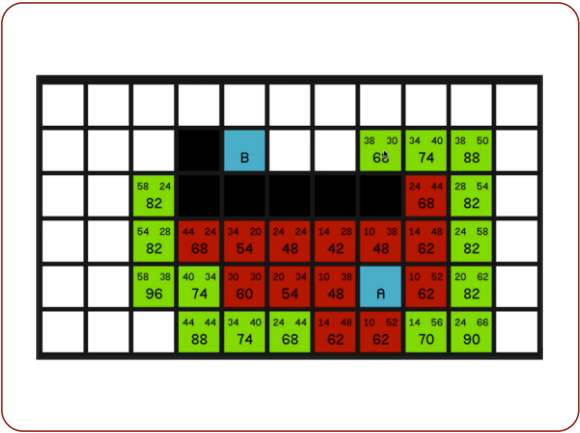
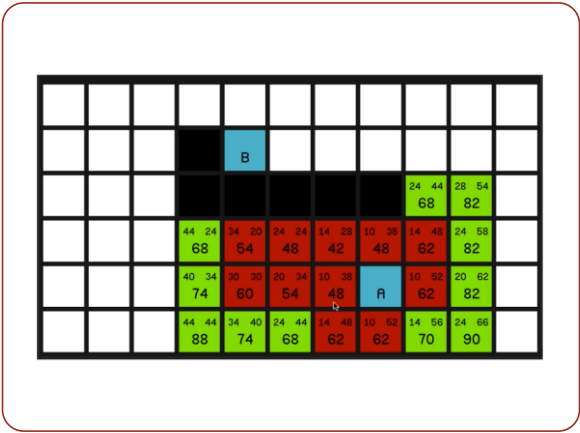
- **A算法**
- 在 GRAPHSEARCH 过程中，如果第8步的重排 OPEN 表是依据 $f(n)=g(n)+h(n)$ 进行的，则称该过程为 **A算法**；
- 显然，A算法对 $h(n)$ 没有明确的限制。
- **A*算法**
- 采用 $h^*(n)$ 的下界 $h(n)$ 为启发函数的 A 算法，称为 **A*算法**。
- **A*算法** 要求 $h(n) \leq h^*(n)$ 。它表示某种偏于保守的估计。
- 如果算法有解，**A*算法** 一定能够找到最优的解答。
- 一般说来，在满足 $h(n) \leq h^*(n)$ 的前提下， $h(n)$ 的比重越大越好， $h(n)$ 的比重越大表示启发性越强。

例



- 每个方块左上角的值记为G，表示该点到A的距离，右上角值为H，H取其到B的距离。F=H+G。





小结

● 状态空间法是一种基于解答空间的问题表示和求解方法，它是以状态和操作符为基础的。在利用状态空间图表示时，我们从某个初始状态开始，每次加一个操作符，递增地建立起操作符的试验序列，直到达到目标状态为止。

40

小结

- 对OPEN表中节点排序方式产生了不同的搜索策略，不同的搜索策略效率不同。
- 这种排序可以是任意的即盲目的(属于盲目搜索)，也可以用各种启发思想或其它准则为依据(属于启发式搜索)。
 - 无信息搜索（盲目式搜索）
 - 宽度优先搜索
 - 深度优先搜索
 - 等代价搜索
 - 有信息搜索（启发式搜索）
 - A算法
 - A*算法

图搜索策略

习题

• 2: A*算法寻路

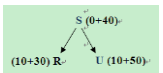
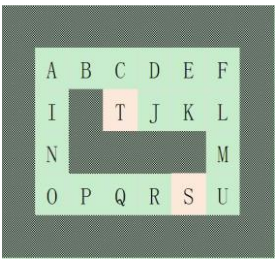


图1 图2

- 2: A*算法寻路
- 机器人从图1中的S点出发，利用A*算法寻找一条到达T点的最短路径，每次只能向与当前位置相邻的上下左右四个邻域块移动（移动一次的代价为10），墙壁（标阴影部分）处不可移动。用G表示S到当前点的路径长度，H表示当前点到T的曼哈顿距离(两点间的水平距离和垂直距离之和)， $F = G + H$ 。如果遇到F值相等，优先考虑H值较小的点。
- （1）用A*算法搜索，在图2中画出搜索树，如图2所示标出当前扩展节点的G+H值。
- （2）用A*算法搜索，在表1中列出搜索中的OPEN、CLOSED表的内容。要求OPEN表中的点从左至右根据F值升序排列，F值相同时，根据H值升序排列，F值和H值都相同时，新生成节点靠左排列。

2: A*算法寻路
表1

OPEN	CLOSED
{S}	{}
{R,U}	{S}

Copyright by Lrc&Mch