

黄水清20182681人工智能

写出 BP

使用曼哈顿距离写出 open 表和 close 表

模糊关系的

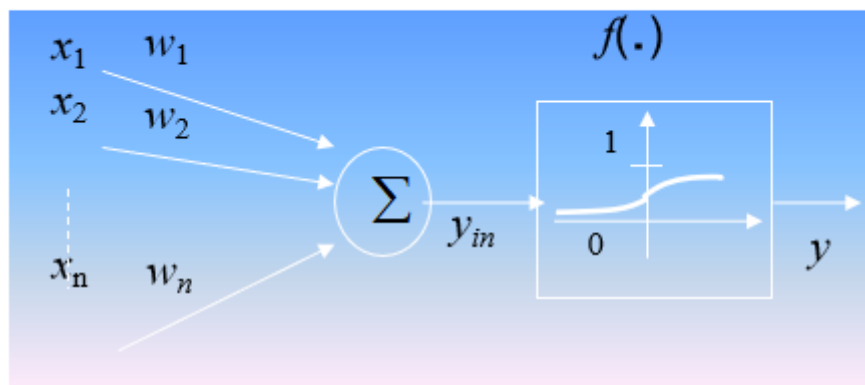
矩阵相乘：对应相乘再相加

关系合成：对应取小再取大

带书，5 道简答题，50 分，在书上还有 5 道题是平时例题，50 分

1.BP 神经网络权值训练，权值修改，可以带计算器

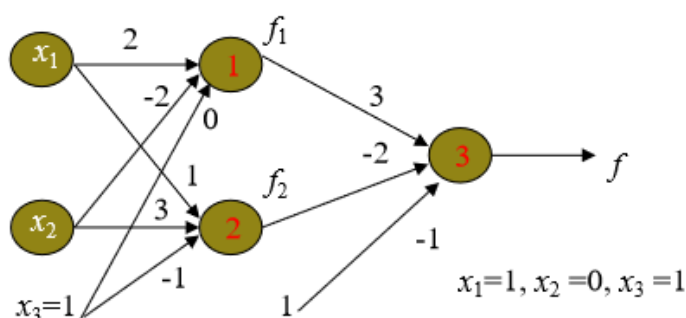
认识BP网络



$$y_{in} = \sum_{i=1}^n w_i x_i = W X^T$$
$$y = f(y_{in}) = \frac{1}{1 + e^{-y_{in}}}$$

(信息) 正向传播过程

➤ 输入数据从输入层传播到输出层的过程为正向传播过程



$$f1=s(1\times2+0\times(-2)+1\times0)=s(2)=1/(1+e^{-2})=0.881$$

$$f2=s(1\times1+0\times3+1\times(-1))=s(0)=1/(1+e^0)=0.5$$

$$f=s(0.881\times3+0.5\times(-2)+1\times(-1))=s(0.643)=0.655$$

求导的链式法则

计算梯度

$$\frac{\partial E_d}{\partial w_{ij}} = \frac{\partial E_d}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{ij}}$$

其中 $net_j = \sum_i w_{ij}x_{ij}$, 为 j 单元的输入

$$\frac{\partial net_j}{\partial w_{ij}} = \frac{\partial \sum_i w_{ij}x_{ij}}{\partial w_{ij}} = x_{ij}$$

$$\frac{\partial E_d}{\partial w_{ij}} = \frac{\partial E_d}{\partial net_j} \cdot x_{ij} = \frac{\partial E_d}{\partial o_j} \cdot \frac{\partial o_j}{\partial net_j} \cdot x_{ij}$$

其中 $o_j = f(net_j)$

$$\text{因} \quad \frac{\partial o_j}{\partial net_j} = \frac{\partial f(net_j)}{\partial net_j} = \frac{\partial}{\partial net_j} \left(\frac{1}{1+e^{-net_j}} \right) = o_j(1-o_j)$$

$$\text{故} \quad \frac{\partial E_d}{\partial w_{ij}} = \frac{\partial E_d}{\partial o_j} \cdot o_j(1-o_j) \cdot x_{ij}$$

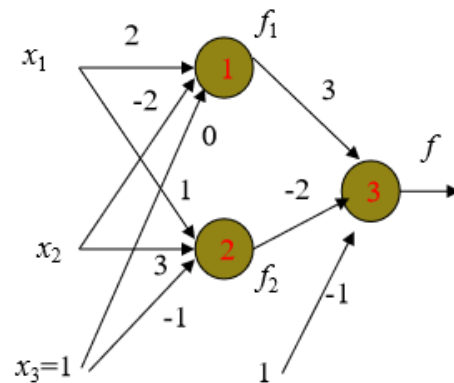
$$\frac{\partial o_j}{\partial net_j} = \frac{\partial f(net_j)}{\partial net_j} = \frac{\partial}{\partial net_j} \left(\frac{1}{1+e^{-net_j}} \right) = o_j(1-o_j)$$

$$\begin{aligned} & \frac{\partial}{\partial net_j} \left(\frac{1}{1+e^{-net_j}} \right) \\ &= \frac{0-(-1)*e^{-net_j}}{(1+e^{-net_j})^2} = \frac{e^{-net_j}}{(1+e^{-net_j})^2} \\ &= \frac{(1+e^{-net_j})-1}{(1+e^{-net_j})^2} = \frac{1}{(1+e^{-net_j})} - \frac{1}{(1+e^{-net_j})^2} \\ &= \frac{1}{1+e^{-net_j}} * \left(1 - \frac{1}{1+e^{-net_j}} \right) \\ &= o_j(1-o_j) \end{aligned}$$

• 奇偶校验网络（异或）

- 用下图BP网络实现奇偶校验(当前参数为随机初始化的结果)，该网络除阈值输入外有两个输入，若输入数据中有奇数个1，网络输出0，否则输出1，训练数据见表

x_1	x_2	x_3	t
1	0	1	0
0	0	1	1
0	1	1	0
1	1	1	1



BP网络应用举例

➤ 第一次训练

✓ 训练数据

$x_1=1, x_2=0, x_3=1, t=0, \eta=1$

$f_1 = s(1 \times 2 + 0 \times (-2) + 1 \times 0) = s(2) = 1 / (1 + e^{-2}) = 0.881$

$f_2 = s(1 \times 1 + 0 \times 3 + 1 \times (-1)) = s(0) = 1 / (1 + e^0) = 0.5$

$f = s(0.881 \times 3 + 0.5 \times (-2) + 1 \times (-1)) = s(0.643) = 0.655$

$\delta = 0.655 \times (1 - 0.655) \times (0 - 0.655) = -0.148$

$\delta_1 = 0.881 \times (1 - 0.881) \times (-0.148 \times 3) = -0.047$

$\delta_2 = 0.5 \times (1 - 0.5) \times (-0.148 \times -2) = 0.074$

$w_{13} = 3 + 0.881 \times (-0.148) = 2.870$

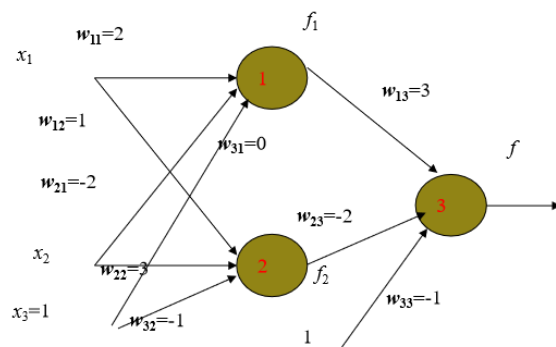
$w_{23} = -2 + 0.5 \times (-0.148) = -2.074$

$w_{33} = -1 + 1 \times (-0.148) = -1.148$

$$\delta_j = (t_j - o_j) o_j (1 - o_j)$$

$$\delta_j = o_j (1 - o_j) \sum_k \delta_k w_{jk}$$

$$\Delta w_{ij} = \eta \delta_j x_{ij}$$



BP神经网络优化

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x_{ij} + \alpha \Delta w_{ij}(t)$$

动量系数 α

增加动量项的目的是为了避免网络训练陷于较浅的局部极小点。

在没有附加动量的作用下，网络可能陷入浅的局部极小值

利用附加动量的作用，则有可能滑过这些局部极小值

理论上其值大小应与权值修正量的大小有关，但实际应用中一般取常量。通常在0-1之间，而且一般比学习率要大。

2.可信度计算

5. 可信度方法 C-F (Certain Factor) Method

肖特里菲(E.Shortliffe) 和 B.Buchanan 在 MYCIN 系统研制过程中产生了可信度方法, 在确定性理论基础上结合概率论等理论提出的一种不精确推理模型。

根据经验对一个事物或现象为真 (相信)的程度称为可信度。每条规则和每个证据都具有一个可信度。

推理规则的一般形式:

If E then H (CF(H, E))

推理规则的一般形式:

If E then H (CF(H,E))

其中 CF(H, E) 是该规则的可信度, 称为可信度因子或规则强度。

CF(H, E) > 0 表示该证据增加了结论为真的程度, 且 CF(H, E) 的值越大则结论 H 越真; 若 CF(H, E) = 1, 则表示该证据使结论为真。

CF(H, E) < 0 表示该证据增加了结论为假的程度, 且 CF(H, E) 的值越小则结论 H 越假; 若 CF(H, E) = - 1, 则表示该证据使结论为假。

CF(H, E) = 0 表示证据 E 和结论 H 没有关系。

$$-1 \leq CF(H, E) \leq 1$$

CF(H, E)的特殊值:

当 CF(H, E) = 1 时, 有 $P(H|E)=1$, 前提 E 真, 结论 H 必真

当 CF(H, E) = -1 时, 有 $P(H|E)=0$ 前提 E 真, 结论 H 必假

当 CF(H, E) = 0 时, 有 $P(H|E)=P(H)$, 前提 E 真假与结论 H 无关, 即 H 与 E 独立

(1) 前提证据事实总 CF 值计算

$$CF(E_1 \wedge E_2 \wedge \dots \wedge E_n) = \min\{CF(E_1), CF(E_2), CF(E_n)\}$$

$$CF(E_1 \vee E_2 \vee \dots \vee E_n) = \max\{CF(E_1), CF(E_2), \dots, CF(E_n)\}$$

其中 E_1, E_2, \dots, E_n 是与规则前提各条件匹配的事实。

(2) 推理结论 CF 值计算

$$CF(H) = CF(H, E) \cdot \max\{0, CF(E)\}$$

其中 E 是与规则前提对应的各事实, CF(H, E) 是规则中结论的可信度, 即规则强度。

可信度方法---CF 模型的计算

(3) 重复结论的 CF 值计算

若同一结论 H 分别被不同的多条规则推出, 但可信度不同, 则可用合成算法求出综合可信度。

多条知识的综合可通过两两的合成实现。

当两条规则推出同一结论 H 时, CF(H) 的计算:

$$CF(H)_1 + CF(H)_2 - CF(H)_1 \cdot CF(H)_2, \text{ 当 } CF(H)_1 \geq 0, \text{ 且 } CF(H)_2 \geq 0$$

$$CF(H) = CF(H)_1 + CF(H)_2 + CF(H)_1 \cdot CF(H)_2, \text{ 当 } CF(H)_1 < 0, \text{ 且 } CF(H)_2 < 0$$

$$CF(H)_1 + CF(H)_2, \text{ 否则}$$

可信度方法举例—例 1

已知 R: IF (E1 or E2) and E3 THEN H (0.8),

CF(E1) = 0.4、CF(E2) = 0.6、CF(E3) = 0.7, 求 CF (H) 。

解:

设 E = (E1 or E2) and E3

$$CF(E1 \text{ or } E2) = \max\{CF(E1), CF(E2)\} = 0.6$$

$$CF(E) = \min\{CF(E1 \text{ or } E2), CF(E3)\} = 0.6$$

$$CF(H) = CF(E) \cdot CF(H, E) = 0.6 \times 0.8 = 0.48$$

可信度方法举例—例 2

R1:IF E1 THEN H $CF(H, E1) = (0.8)$

R2:IF E2 THEN H $CF(H, E2) = (0.7)$

$CF(E1) = 0.4$ 、 $CF(E2) = 0.6$ ，求 $CF(H)$ 。

解：

$CF_1(H) = CF(E1) \cdot CF(H, E1) = 0.4 \times 0.8 = 0.32$

$CF_2(H) = CF(E2) \cdot CF(H, E2) = 0.6 \times 0.7 = 0.42$

$CF(H) = 0.32 + 0.42 - 0.32 \times 0.42 = 0.6$

基于论据的方法：

在知识库中，如果发现某个知识错了，那么就得撤消此错误知识，同时还得撤消依赖于此知识成立的知识。而要做到这一点，就必须研究知识间的依赖关系。

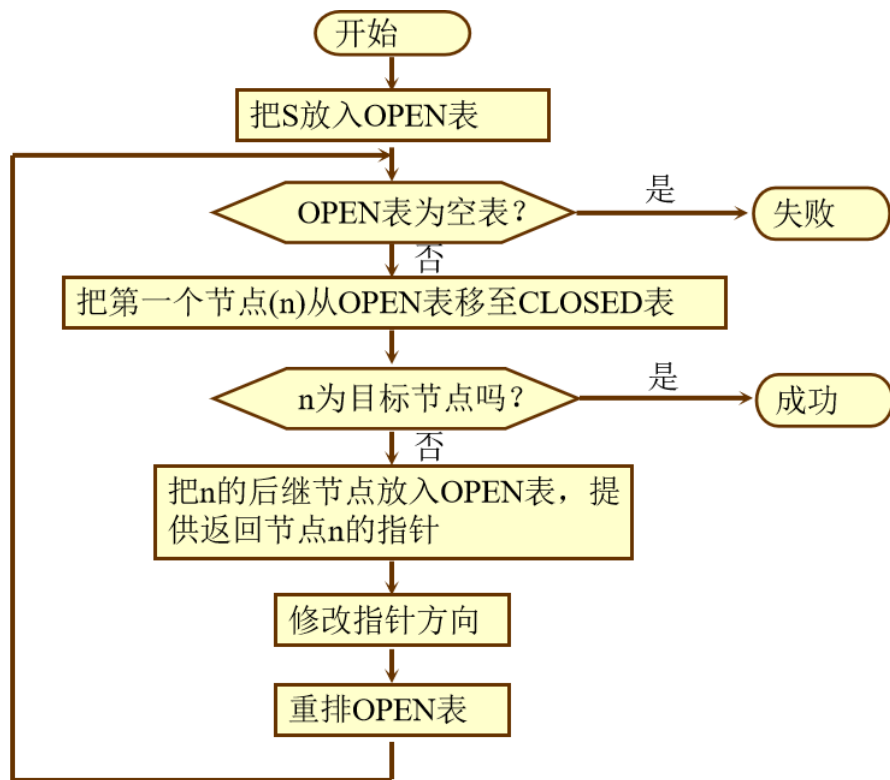
真值维护系统（TMS）完成类似的任务。Doyle 实现了一个基于论据的真值维护系统（JTMS）。在 JTMS 中，知识用节点表示，节点包含节点编号、节点内容（该节点对应的命题）以及该节点的论据集。在 JTMS 中，记录所有可信的节点。节点之间的依赖关系可以用与或树技术来解决。Kleer 克服了 JTMS 的一些不足，实现了一个基于假设的真值维护系统（ATMS）。给定规则，ATMS 可以根据该规则更新命题间的依赖关系。

3.图搜索策略求八数码问题， $f(n) = d(n) + g(n)$ ，写出 open 表和 close 表

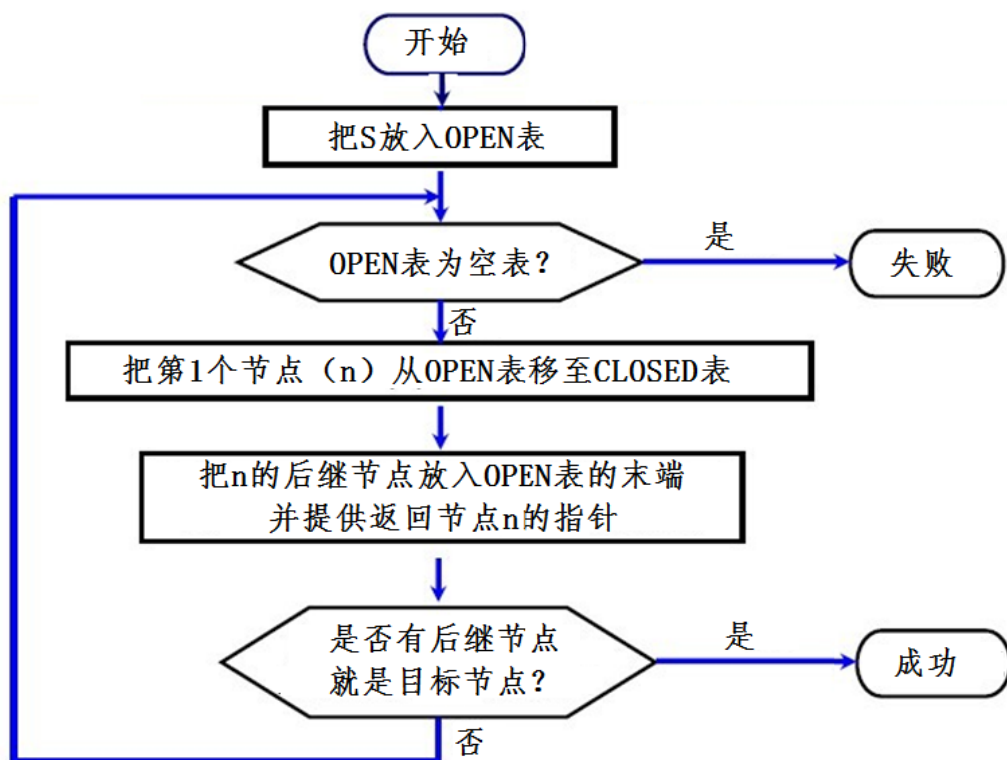
图的搜索过程

- 1 必须记住下一步还可以走哪些点；
- 2 OPEN表(记录还没有扩展的点)；
- 3 必须记住哪些点走过了；
- 4 CLOSED表(记录已经扩展的点)；
- 5 必须记住从目标返回的路径；
- 6 每个表示状态的节点结构中必须有指向父节点的指针；

图的一般搜索策略



•13



宽度优先搜索流程图

•16

● Example

八数码难题 (8-puzzle problem)

2	8	3
1		4
7	6	5

(start state)



1	2	3
8		4
7	6	5

(goal state)

规定：将牌移入空格的顺序为：从空格左边开始
顺时针旋转。不许斜向移动，也不返回先辈节点。
从下图可见，要扩展26个节点，共生成46个节点
之后才求得解（目标节点）。

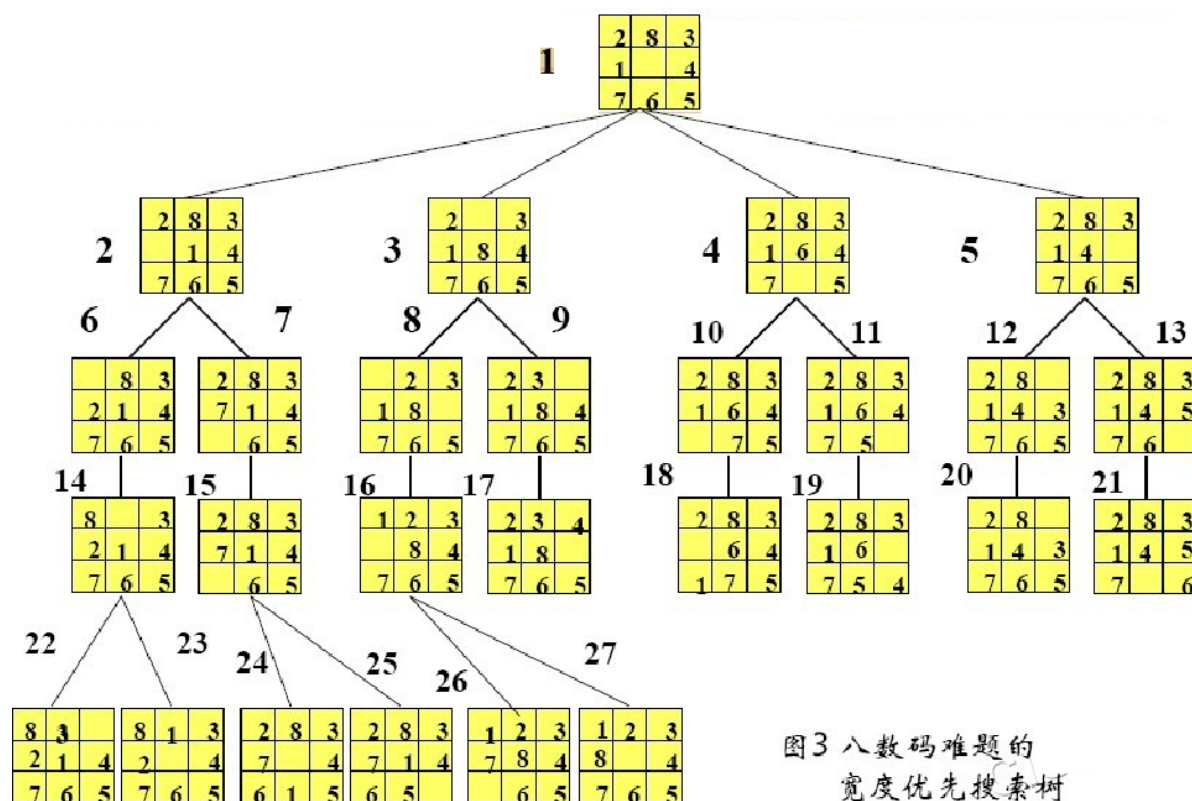


图3 八数码难题的
宽度优先搜索树

启发式搜索

启发式搜索 (Heuristic Search Search)

- 特点：重排OPEN表，选择最有希望的节点加以扩展
- 种类：有序搜索、A*算法等

3.3.1 启发式搜索策略和估价函数

❖ 盲目搜索可能带来组合爆炸

❖ 启发式信息 (Heuristic information)

用来加速搜索过程的有关问题领域的特征信息。

❖ 算法

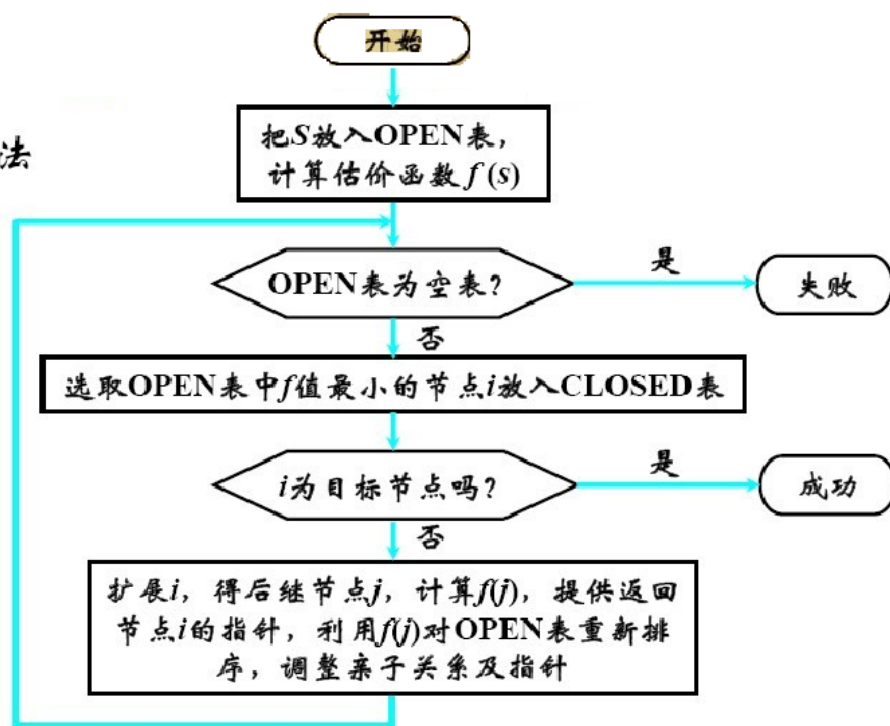


图5 有序搜索算法框图

我们采用简单的估价函数

$$f(n)=d(n)+W(n)$$

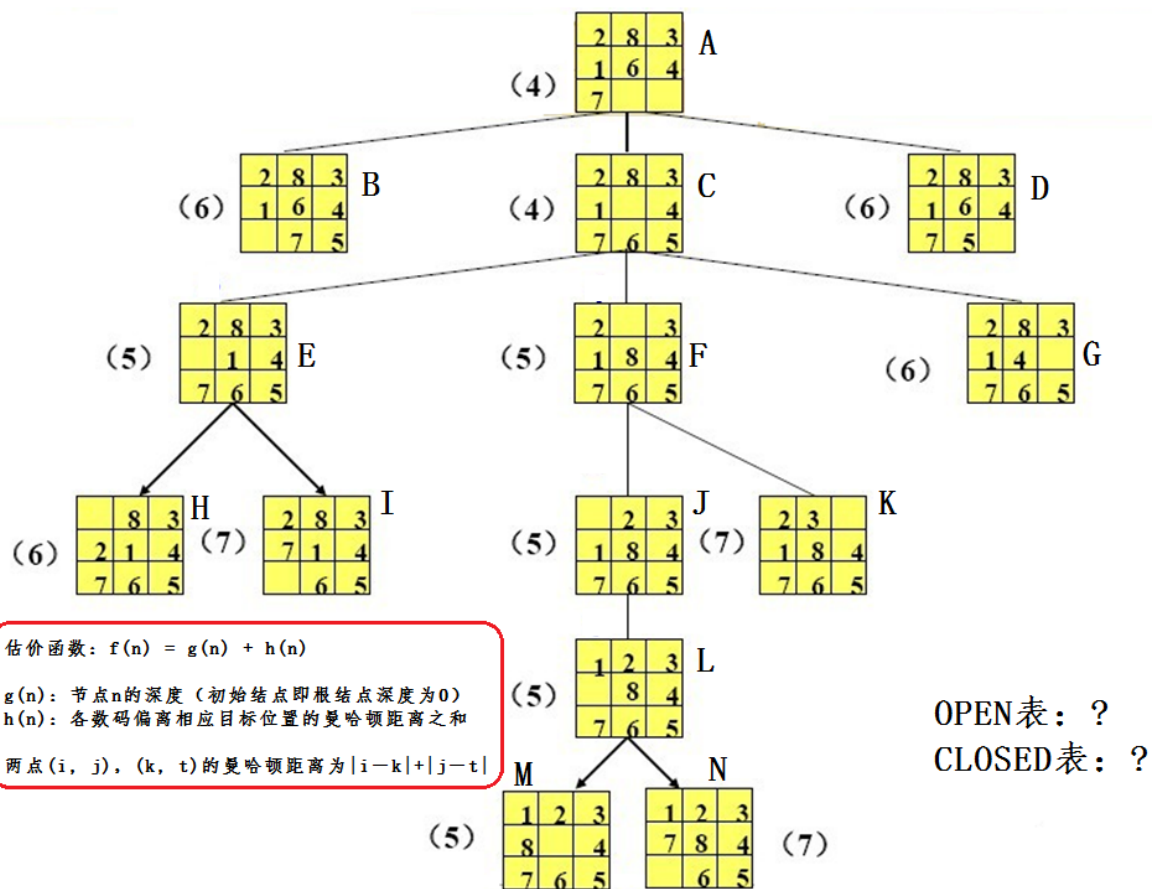
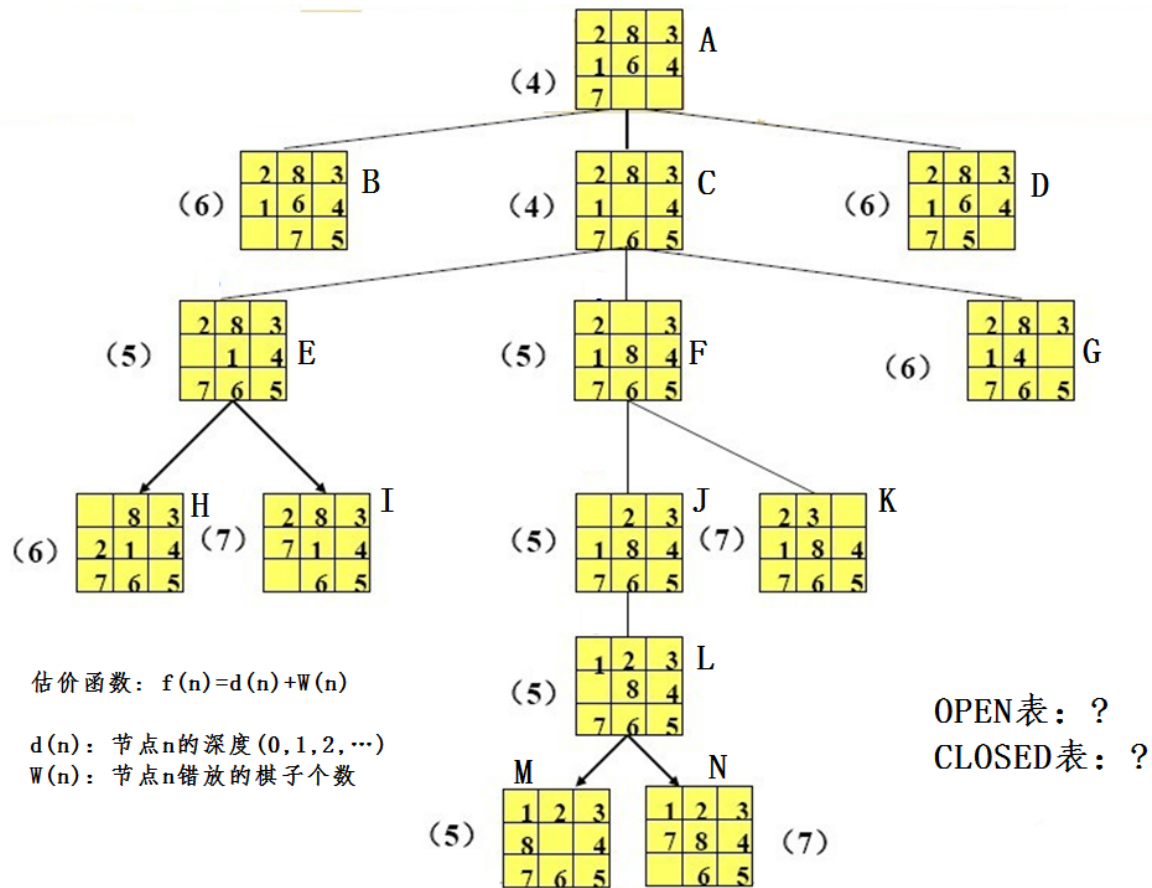
其中：

$d(n)$ 是搜索树中节点 n 的深度(0,1,2,...);

$W(n)$ 用来计算对应于节点 n 的数据库中错放的棋子个数。因此，起始节点棋局

2	8	3
1	6	4
7		5

的 f 值等于 $0+4=4$ 。



A*算法

A*算法 (Algorithm A*)

● 估价函数的定义

对节点 n 定义 $f^*(n)=g^*(n)+h^*(n)$,表示从 S 开始约束通过节点 n 的一条最佳路径的代价。

希望估价函数 f 定义为: $f(n)=g(n)+h(n)$

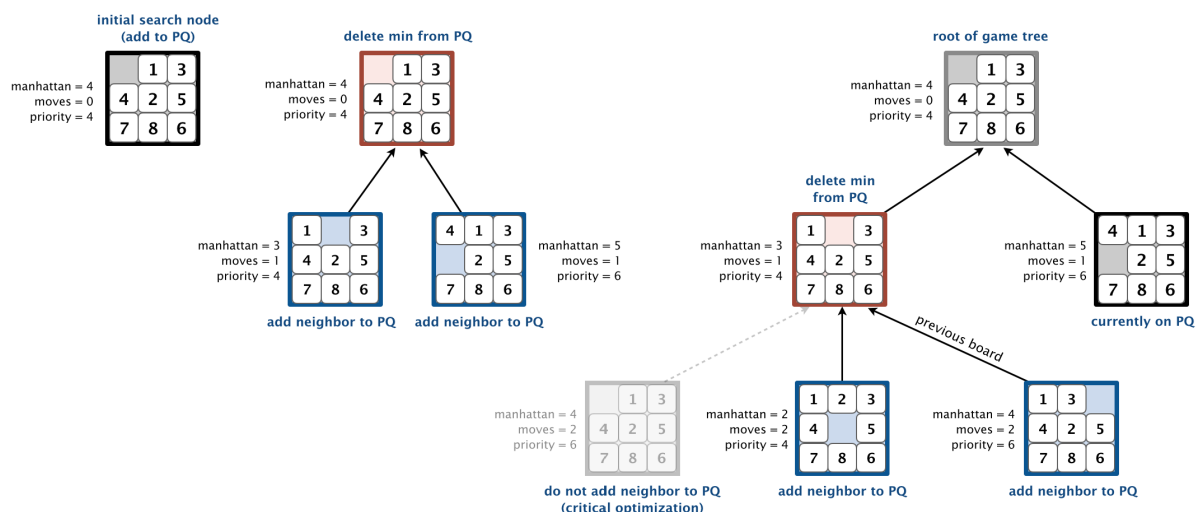
—— g 是 g^* 的估计, h 是 h^* 的估计

● A*算法的定义

定义1 在GRAPHSEARCH过程中, 如果第8步的重排OPEN表是依据 $f(x)=g(x)+h(x)$ 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 x 存在 $h(x) \leq h^*(x)$, 则称 $h(x)$ 为 $h^*(x)$ 的下界, 它表示某种偏于保守的估计。

定义3 采用 $h^*(x)$ 的下界 $h(x)$ 为启发函数的A算法, 称为A*算法。当 $h=0$ 时, A*算法就变为有序搜索算法。



练习

• 请编程完成上述奇偶校验网络的训练

• 提示

• 网络的输入层、隐层、输出层数量见前面的图

• 将前述公式转换为矩阵表示形式, 定义两个权矩阵分别为 3×2 的矩阵和 3×1 的矩阵, 其初始值分别为

$$w^1 = \begin{bmatrix} 2 & 1 \\ -2 & 3 \\ 0 & -1 \end{bmatrix} \quad w^2 = \begin{bmatrix} 3 \\ -2 \\ -1 \end{bmatrix}$$

$$\text{则 } [f_1 \ f_2] = \text{Sigmoid}([x_1 \ x_2 \ x_3]w^1)$$

$$f = \text{Sigmoid}([f_1 \ f_2 \ 1]w^2)$$

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x_{ij}$$

学习率 η

大的学习率可能使网络权值每一次的修正量过大，甚至会导致权值在修正过程中超出某个误差的极小值呈不规则跳跃而不收敛；但过小的学习率导致学习时间过长，不过能保证收敛于某个极小值。所以，一般倾向选取较小的学习率以保证学习过程的收敛性（稳定性），通常在0.01-0.8之间。

BP神经网络优化

4.你跟爷爷奶奶像不像=>模糊关系的合成:对应取小再取大，类似矩阵相"乘"法:对应相乘再相加

and 取小, or 取大



用模糊矩阵R来表示为

$$S = \begin{bmatrix} 0.5 & 0.7 \\ 0.1 & 0 \end{bmatrix}$$



那么家中孙子、孙女与祖父、祖母的相似程度如何？

$$\begin{aligned} R \circ S &= \begin{bmatrix} 0.2 & 0.8 \\ 0.6 & 0.1 \end{bmatrix} \circ \begin{bmatrix} 0.5 & 0.7 \\ 0.1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} (0.2 \wedge 0.5) \vee (0.8 \wedge 0.1) & (0.2 \wedge 0.7) \vee (0.8 \wedge 0) \\ (0.6 \wedge 0.5) \vee (0.1 \wedge 0.1) & (0.6 \wedge 0.7) \vee (0.1 \wedge 0) \end{bmatrix} \\ &= \begin{bmatrix} 0.2 & 0.2 \\ 0.5 & 0.6 \end{bmatrix} \end{aligned}$$



P	祖父	祖母
孙子	0.2	0.2
孙女	0.5	0.6

5.传教士过河的状态序列

1.问题重述

在河的左岸有N个传教士、N个野人和一条船，传教士们想用这条船把所有人都运过河去，但有以下条件限制：

- (1) 修道士和野人都会划船，但船每次最多只能运K个人；
- (2) 在任何岸边野人数目都不能超过修道士，否则修道士会被野人吃掉。

假定野人会服从任何一种过河安排，请规划出一个确保修道士安全过河的计划。

2.问题分析

2.1约束条件:

- ① $M \geq C$ 任何时刻两岸、船上都必须满足传教士人数不少于野人数 ($M=0$ 时除外, 既没有传教士)
- ② $M+C \leq K$ 船上人数限制在 K 以内

2.2求解:

传教士与野人全部安全渡到对岸的解决方案

3.求解过程

3.1变量假设:

设 $N=3$, $K=2$ (三个 M 和三个 C , 每次渡河二人以下)

L: 左岸, R: 右岸,

B: 是否有船 (0: 无船, 1: 有船)

3.2状态表示

定义: 用三元组 (ML, CL, BL) 表示左岸状态, 其中:

$0 \leq ML, CL \leq 3$, $BL \in \{0,1\}$

如: $(0, 3, 1)$ 表示左岸有三个野人, 船在左岸。

从 $(3, 3, 1)$ 到 $(0, 0, 0)$ 的状态转换

状态空间: 32 种状态, 其中:

12种不合理状态: 如 $(1, 0, 1)$ 说明右岸有2个 M , 3个 C ;

4种不可能状态: 如 $(3, 3, 0)$ 说明所有 M 和 C 都在左岸, 而船在右岸

\therefore 可用的状态共16种, 组成合理的状态空间

(m, c, b)	(m, c, b)
(0 0 1)达不到	(0 0 0)
(0 1 1)	(0 1 0)
(0 2 1)	(0 2 0)
(0 3 1)	(0 3 0)达不到
(1 0 1)不合法	(1 0 0)不合法
(1 1 1)	(1 1 0)
(1 2 1)不合法	(1 2 0)不合法
(1 3 1)不合法	(1 3 0)不合法
(2 0 1)不合法	(2 0 0)不合法
(2 1 1)不合法	(2 1 0)不合法
(2 2 1)	(2 2 0)
(2 3 1)不合法	(2 3 0)不合法
(3 0 1)达不到	(3 0 0)
(3 1 1)	(3 1 0)
(3 2 1)	(3 2 0)
(3 3 1)	(3 3 0)达不到

状态空间具体描述

状 态	m, c, b	状 态	m, c, b	状 态	m, c, b	状 态	m, c, b
S ₀	3,3,1	S ₈	1,3,1	S ₁₆	3,3,0	S ₂₄	1,3,0
S ₁	3,2,1	S ₉	1,2,1	S ₁₇	3,2,0	S ₂₅	1,2,0
S ₂	3,1,1	S ₁₀	1,1,1	S ₁₈	3,1,0	S ₂₆	1,1,0
S ₃	3,0,1	S ₁₁	1,0,1	S ₁₉	3,0,0	S ₂₇	1,0,0
S ₄	2,3,1	S ₁₂	0,3,1	S ₂₀	2,3,0	S ₂₈	0,3,0
S ₅	2,2,1	S ₁₃	0,2,1	S ₂₁	2,2,0	S ₂₉	0,2,0
S ₆	2,1,1	S ₁₄	0,1,1	S ₂₂	2,1,0	S ₃₀	0,1,0
S ₇	2,0,1	S ₁₅	0,0,1	S ₂₃	2,0,0	S ₃₁	0,0,0

3.2操作集

定义：Pmc操作：从左岸划向右岸

Qmc操作：从右岸划向左岸

船上人数组合 (m, c) 共5种 (1, 0) , (1, 1) , (2, 0) , (0, 1) , (0, 2)

∴每一种船上的人数组合同时对应P, Q二种操作

∴系统共有 $5 \times 2 = 10$ 种操作（规则）

如：P10: if (ML, CL, BL=1) then (ML-1, CL, BL-1)

如果船在左岸，那么一个传教士划船到右岸

Q01: if (ML, CL, BL=0) then (ML, CL+1, BL+1)

如果船在右岸，那么一个野人划船回到左岸

总共有10种操作

$F = \{P10, P20, P11, P01, P02, Q10, Q20, Q11, Q01, Q02\}$

P10 if(ML,CL , BL=1) then (ML-1 , CL , BL-1)

P01 if(ML,CL , BL=1) then (ML , CL-1 , BL-1)

P11 if(ML,CL , BL=1) then (ML-1 , CL-1 ,BL -1)

P20 if(ML,CL , BL=1) then (ML-2 , CL , BL-1)

P02 if(ML,CL , BL=1) then (ML , CL-2 , BL-1)

Q10 if(ML,CL , BL=0) then (ML+1 , CL ,BL+1)

Q01 if(ML,CL , BL=0) then (ML , CL+1 , BL+1)

Q11 if(ML,CL , BL=0) then (ML+1 , CL +1,BL +1)

Q20 if(ML,CL , BL=0) then (ML+2 , CL +2,BL +1)

Q02 if(ML,CL , BL=0) then (ML , CL +2, BL+1)

3.3控制策略

最短路径有4条，由11次操作构成。

(P11、Q10、P02、Q01、P20、Q11、P20、Q01、P02、Q01、P02)

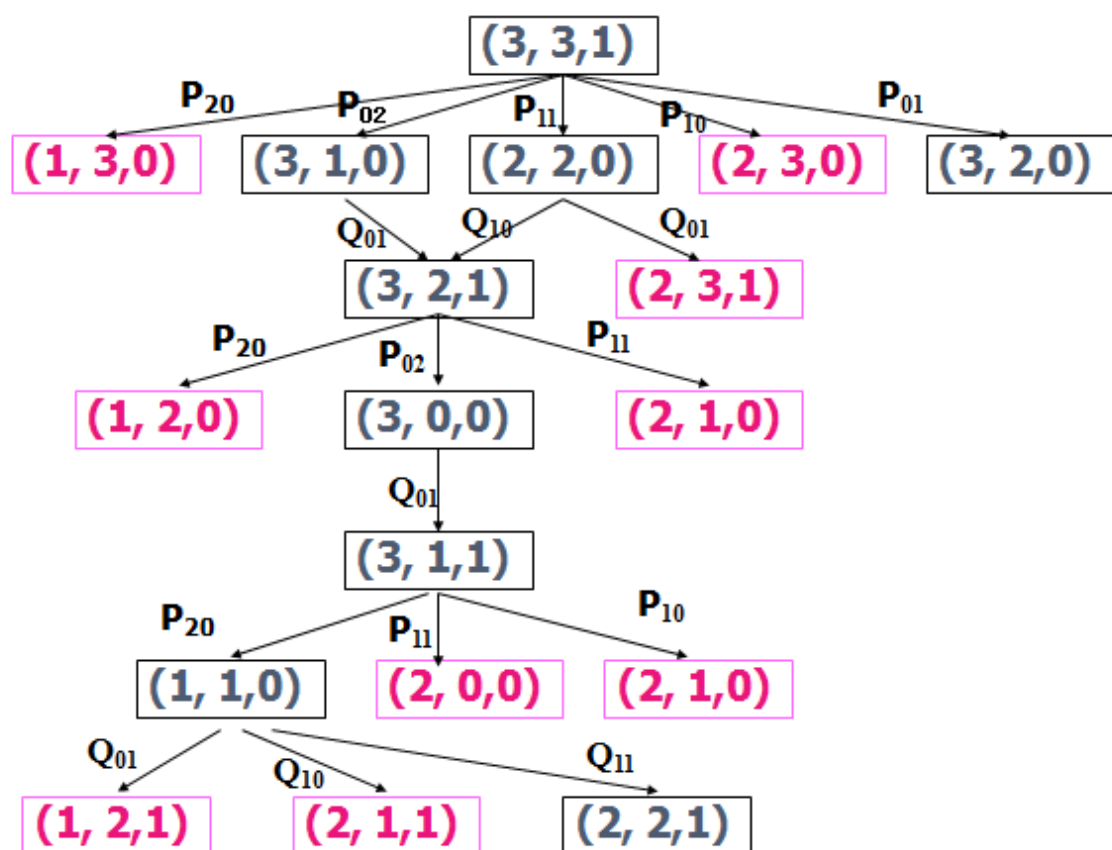
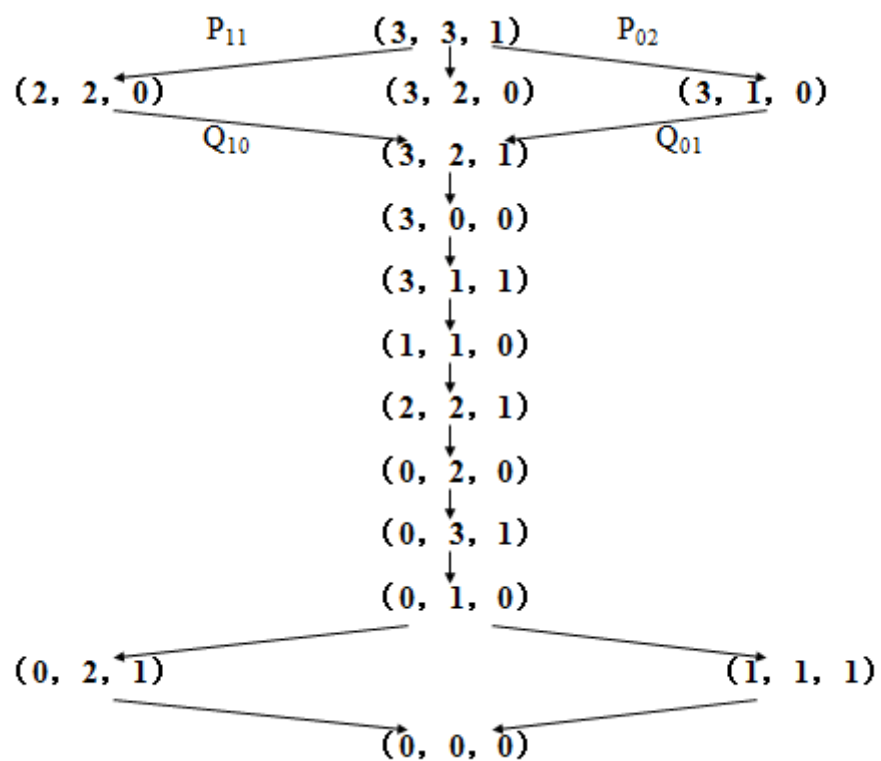
(P11、Q10、P02、Q01、P20、Q11、P20、Q01、P02、Q10、P11)

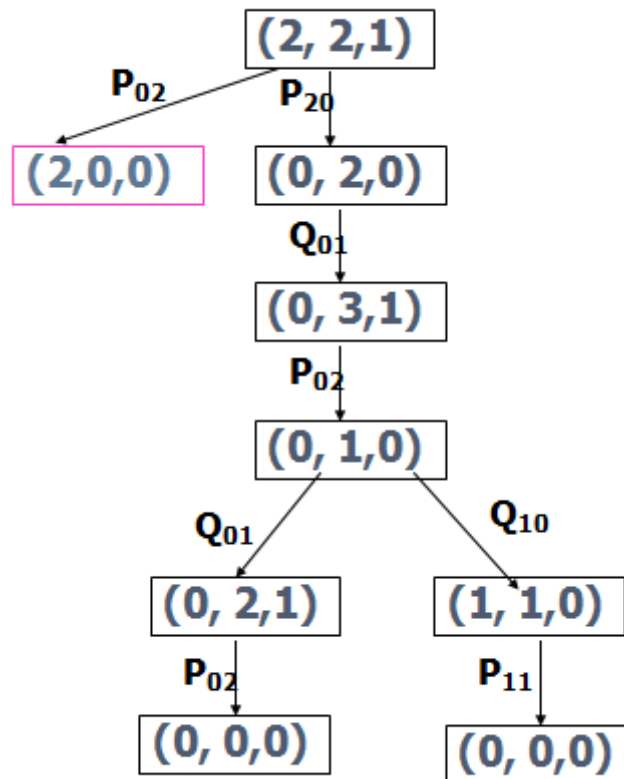
(P02、Q01、P02、Q01、P20、Q11、P20、Q01、P02、Q01、P02)

(P02、Q01、P02、Q01、P20、Q11、P20、Q01、P02、Q10、P11)

3.4状态空间图

状态空间图是一个有向图，图中的节点代表状态，节点之间的连线代表操作，箭头代表状态的转换方向。





3.5状态空间的一般搜索过程

OPEN表：用于存放刚生成的节点

CLOSE表：用于存放将要扩展或已扩展的节点

1) 把初始节点S0放入OPEN表,并建立只含S0的图,记为G

OPEN: =S0,G: =G0 (G0=S0)

2) 检查OPEN表是否为空,若为空则问题无解,退出

LOOP: IF (OPEN) = () THEN EXIT (FAIL)

3) 把OPEN表的第一个节点取出放入CLOSE表,记该节点为节点n

N: =FIRST (OPEN) ,REMOVE (n,OPEN) ,ADD (n,CLOSE)

4) 观察节点n是否为目标节点,若是,则求得问题的解,退出

IF GOAL (n) THEN EXIT (SUCCESS)

5) 扩展节点n,生成一组子节点,把其中不是节点n先辈的那些子节点记作集合M,并把这些节点作为节点n的子节点加入G中.

EXPAND (n) -->M (mi) ,G: =ADD (mi,G)

针对M中子节点的不同情况,分别进行如下处理

对于那些未曾在G中出现过的M成员设置一个指向父节点 (n) 的指针,并把它放入OPEN表

对于那些先前已在G中出现过的M成员,确定是否要修改指向父节点的指针

对于那些先前已在G中出现,并且已经扩展了的M成员,确定是否需要修改其后继节点指向父节点的指针

6) 按某种搜索策略对OPEN表中的节点进行排序

7) 转第2步

针对本题，设置OPEN、CLOSED两个队列分别存放待扩展节点和已扩展节点；

对每个生成的新节点，要检查上述两表中是否已存在，是否是非法节点；

(合法节点满足：

ML=0||ML=3||ML=CL)

每扩展一个节点应记录产生合法后继节点的操作，以便最后给出操作序列；

