八数码实验分析

1. **宽度优先搜索算法**

搜索过程：

(1)初始化open表与closed表，把初始节点(序号1)放入open表中。

(2)若open=()，则算法失败终止;否则，转步骤(3)。

(3)把open表的第1个节点i移出，放入closed表的表尾。

(4)若节点i的状态是目标状态，否则转步骤(5)。

(5)若节点i 不可扩展(即在算符集中找不到可用算符)， 则转步骤(2);否则，转步骤(6)。

(6)逐一用可用算符扩展节点，生成i的所有子节点。若 子节点j的状态既不在open表中，也不在closed表中，则节点j是一个新节点。将所有的新子节点逐一放 入open表的表尾，并记载子节点各个域的值，转步骤 (2);否则，不把新生成的节点j放入open表中，放弃 节点j，转步骤(2)。

实验结果：

第1个状态是:

2 8 3

1 6 4

7 0 5

第2个状态是:

2 8 3

1 6 4

7 5 0

第3个状态是:

2 8 3

1 0 4

7 6 5

第4个状态是:

2 8 3

1 6 4

0 7 5

第5个状态是:

2 8 3

1 6 0

7 5 4

第6个状态是:

2 8 3

1 4 0

7 6 5

第7个状态是:

2 0 3

1 8 4

7 6 5

第8个状态是:

2 8 3

0 1 4

7 6 5

第9个状态是:

2 8 3

0 6 4

1 7 5

第10个状态是:

2 8 0

1 6 3

7 5 4

第11个状态是:

2 8 3

1 0 6

7 5 4

第12个状态是:

2 8 3

1 4 5

7 6 0

第13个状态是:

2 8 0

1 4 3

7 6 5

第14个状态是:

2 3 0

1 8 4

7 6 5

第15个状态是:

0 2 3

1 8 4

7 6 5

第16个状态是:

2 8 3

7 1 4

0 6 5

第17个状态是:

0 8 3

2 1 4

7 6 5

第18个状态是:

2 8 3

6 0 4

1 7 5

第19个状态是:

0 8 3

2 6 4

1 7 5

第20个状态是:

2 0 8

1 6 3

7 5 4

第21个状态是:

2 8 3

1 5 6

7 0 4

第22个状态是:

2 0 3

1 8 6

7 5 4

第23个状态是:

2 8 3

0 1 6

7 5 4

第24个状态是:

2 8 3

1 4 5

7 0 6

第25个状态是:

2 0 8

1 4 3

7 6 5

第26个状态是:

2 3 4

1 8 0

7 6 5

第27个状态是:

1 2 3

0 8 4

7 6 5

第28个状态是:

2 8 3

7 1 4

6 0 5

第29个状态是:

8 0 3

2 1 4

7 6 5

第30个状态是:

2 8 3

6 7 4

1 0 5

第31个状态是:

2 8 3

6 4 0

1 7 5

第32个状态是:

2 0 3

6 8 4

1 7 5

第33个状态是:

8 0 3

2 6 4

1 7 5

第34个状态是:

2 6 8

1 0 3

7 5 4

第35个状态是:

0 2 8

1 6 3

7 5 4

第36个状态是:

2 8 3

1 5 6

7 4 0

第37个状态是:

2 8 3

1 5 6

0 7 4

第38个状态是:

2 3 0

1 8 6

7 5 4

第39个状态是:

0 2 3

1 8 6

7 5 4

第40个状态是:

2 8 3

7 1 6

0 5 4

第41个状态是:

0 8 3

2 1 6

7 5 4

第42个状态是:

2 8 3

1 0 5

7 4 6

第43个状态是:

2 8 3

1 4 5

0 7 6

第44个状态是:

2 4 8

1 0 3

7 6 5

第45个状态是:

0 2 8

1 4 3

7 6 5

第46个状态是:

2 3 4

1 8 5

7 6 0

第47个状态是:

2 3 4

1 0 8

7 6 5

第48个状态是:

1 2 3

7 8 4

0 6 5

第49个状态是:

1 2 3

8 0 4

7 6 5

宽度优先算法优点：

问题有解便可找到解，且相对深度优先搜索在本题情况下会更有效。

宽度优先算法缺点：

相比启发式搜索效率低。

1. **深度优先搜索算法**

搜索过程：

(1)初始化open表与closed表，把初始节点放入open表中。

(2)若open=()，则算法失败终止;否则，转步骤(3)。

(3)把open表的第1个节点i移出，放入closed表的表尾。

(4)若节点i的状态是目标状态，则算法成功终止;否 则，转步骤(5)。

(5)若节点i不可扩展(即在算符集中找不列可用算符)， 则转步骤(2);否则，转步骤(6)。

(6)逐一用可用算符扩展节点i，生成i的所有子节点。 若子节点j的状态既不在open表中，又不在closed表中，则节点j是一个允许的新节点。将 所有的新子节点按节点序号从小到大依序放入open表 的表首，并记载子节点各域的值。否则，不把新节点j 放入open表中(放弃节点j)，转步骤(2)。

实验结果：（由于未设置深度，所以实验结果较大，附在文件夹内深度优先结果.txt中，此处展示部分结果）

第49121个状态是:

1 2 3

8 4 5

7 0 6

第49122个状态是:

1 2 3

8 0 5

7 4 6

第49123个状态是:

1 2 3

8 5 0

7 4 6

第49124个状态是:

1 2 3

8 5 6

7 4 0

第49125个状态是:

1 2 3

8 5 6

7 0 4

第49126个状态是:

1 2 3

8 0 6

7 5 4

第49127个状态是:

1 2 3

8 6 0

7 5 4

第49128个状态是:

1 2 3

8 6 4

7 5 0

第49129个状态是:

1 2 3

8 6 4

7 0 5

第49130个状态是:

1 2 3

8 0 4

7 6 5

深度优先算法优点：

存在解时一定可以找到解

缺点：

当深度较深时，浪费时间空间，效率低

1. **分支有界搜索算法**

实验过程：

与深度优先算法一致，但设置了最大深度（99）

实验结果：

深度过浅无法得到结果

在深度较大（50000）时，得到结果在Bound.txt中

分支有界搜索算法优点：

节省时间空间，提高效率。

缺点：

深度选择不当很可能找不到解。

1. **爬山法**

搜索过程：

（1）. 生成第一个可能的解。若是目标，则停止;否则转下一步。

（2）. 从当前可能的解出发，生成新的可能解集。

1 ）用测试函数测试新的可能解集中的元素，若是解，则停止;否则转2）。

2） 若不是解，则将它与至今已测试过的“解”比较。若它 最接近解，则保留作为最佳元素;若它不最接近解，则 舍弃。

（3）. 以当前最佳元素为起点，转（2）

实验结果：

路径寻找错误

爬山法优点：

提高搜索效率

缺点：

遇到波谷导致无法找到正确解。

1. **A\*算法**

搜索过程：

（1）把初始节点S0放入Open表中，f(S0)=g(S0)+h(S0)；

（2）如果Open表为空，则问题无解，失败退出；

（3）把Open表的第一个节点取出放入Closed表，并记该节点为n；

（4）考察节点n是否为目标节点。若是，则找到了问题的解，成功退出；

（5）若节点n不可扩展，则转到第(2)步；

（6）扩展节点n，生成子节点ni(i=1,2,……)，计算每一个子节点的估价值f(ni) (i=1,2,……)，并为每一个子节点设置指向父节点的指针，然后将这些子节点放入Open表中；

（7）根据各节点的估价函数值，对Open表中的全部节点按从小到大的顺序重新进行排序；

（8）转第(2)步。

实验结果：

第1个状态是:

2 8 3

1 6 4

7 0 5

其f值为:5

其g值为:0

其h值为:5

第2个状态是:

2 8 3

1 0 4

7 6 5

其f值为:4

其g值为:1

其h值为:3

第3个状态是:

2 0 3

1 8 4

7 6 5

其f值为:6

其g值为:2

其h值为:4

第4个状态是:

2 8 3

0 1 4

7 6 5

其f值为:6

其g值为:2

其h值为:4

第5个状态是:

0 2 3

1 8 4

7 6 5

其f值为:6

其g值为:3

其h值为:3

第6个状态是:

1 2 3

0 8 4

7 6 5

其f值为:6

其g值为:4

其h值为:2

第7个状态是:

1 2 3

8 0 4

7 6 5

其f值为:5

其g值为:5

其h值为:0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

路径长度为:5

第0层状态为:

2 8 3

1 6 4

7 0 5

第1层状态为:

2 8 3

1 0 4

7 6 5

第2层状态为:

2 0 3

1 8 4

7 6 5

第3层状态为:

0 2 3

1 8 4

7 6 5

第4层状态为:

1 2 3

0 8 4

7 6 5

第5层状态为:

1 2 3

8 0 4

7 6 5

A\*算法优点：

省略大量无效过程，提高了效率。

1. **启发式算法（A）**

整体与Astar算法类似，但在h(n)估价函数上略有不同。在本实验中没有明显区别。

实验结果：

第1个状态是:

2 8 3

1 6 4

7 0 5

其f值为:72

其g值为:0

其h值为:72

第2个状态是:

2 8 3

1 6 4

7 5 0

其f值为:73

其g值为:1

其h值为:72

第3个状态是:

2 8 3

1 0 4

7 6 5

其f值为:73

其g值为:1

其h值为:72

第4个状态是:

2 8 3

1 6 4

0 7 5

其f值为:73

其g值为:1

其h值为:72

第5个状态是:

2 8 3

1 6 0

7 5 4

其f值为:74

其g值为:2

其h值为:72

第6个状态是:

2 8 3

1 4 0

7 6 5

其f值为:74

其g值为:2

其h值为:72

第7个状态是:

2 0 3

1 8 4

7 6 5

其f值为:74

其g值为:2

其h值为:72

第8个状态是:

2 8 3

0 1 4

7 6 5

其f值为:74

其g值为:2

其h值为:72

第9个状态是:

2 8 3

0 6 4

1 7 5

其f值为:74

其g值为:2

其h值为:72

第10个状态是:

2 8 0

1 6 3

7 5 4

其f值为:75

其g值为:3

其h值为:72

第11个状态是:

2 8 3

1 0 6

7 5 4

其f值为:75

其g值为:3

其h值为:72

第12个状态是:

2 8 3

1 4 5

7 6 0

其f值为:75

其g值为:3

其h值为:72

第13个状态是:

2 8 0

1 4 3

7 6 5

其f值为:75

其g值为:3

其h值为:72

第14个状态是:

2 3 0

1 8 4

7 6 5

其f值为:75

其g值为:3

其h值为:72

第15个状态是:

0 2 3

1 8 4

7 6 5

其f值为:75

其g值为:3

其h值为:72

第16个状态是:

2 8 3

7 1 4

0 6 5

其f值为:75

其g值为:3

其h值为:72

第17个状态是:

0 8 3

2 1 4

7 6 5

其f值为:75

其g值为:3

其h值为:72

第18个状态是:

2 8 3

6 0 4

1 7 5

其f值为:75

其g值为:3

其h值为:72

第19个状态是:

0 8 3

2 6 4

1 7 5

其f值为:75

其g值为:3

其h值为:72

第20个状态是:

2 0 8

1 6 3

7 5 4

其f值为:76

其g值为:4

其h值为:72

第21个状态是:

2 8 3

1 5 6

7 0 4

其f值为:76

其g值为:4

其h值为:72

第22个状态是:

2 0 3

1 8 6

7 5 4

其f值为:76

其g值为:4

其h值为:72

第23个状态是:

2 8 3

0 1 6

7 5 4

其f值为:76

其g值为:4

其h值为:72

第24个状态是:

2 8 3

1 4 5

7 0 6

其f值为:76

其g值为:4

其h值为:72

第25个状态是:

2 0 8

1 4 3

7 6 5

其f值为:76

其g值为:4

其h值为:72

第26个状态是:

2 3 4

1 8 0

7 6 5

其f值为:76

其g值为:4

其h值为:72

第27个状态是:

1 2 3

0 8 4

7 6 5

其f值为:76

其g值为:4

其h值为:72

第28个状态是:

2 8 3

7 1 4

6 0 5

其f值为:76

其g值为:4

其h值为:72

第29个状态是:

8 0 3

2 1 4

7 6 5

其f值为:76

其g值为:4

其h值为:72

第30个状态是:

2 8 3

6 7 4

1 0 5

其f值为:76

其g值为:4

其h值为:72

第31个状态是:

2 8 3

6 4 0

1 7 5

其f值为:76

其g值为:4

其h值为:72

第32个状态是:

2 0 3

6 8 4

1 7 5

其f值为:76

其g值为:4

其h值为:72

第33个状态是:

8 0 3

2 6 4

1 7 5

其f值为:76

其g值为:4

其h值为:72

第34个状态是:

2 6 8

1 0 3

7 5 4

其f值为:77

其g值为:5

其h值为:72

第35个状态是:

0 2 8

1 6 3

7 5 4

其f值为:77

其g值为:5

其h值为:72

第36个状态是:

2 8 3

1 5 6

7 4 0

其f值为:77

其g值为:5

其h值为:72

第37个状态是:

2 8 3

1 5 6

0 7 4

其f值为:77

其g值为:5

其h值为:72

第38个状态是:

2 3 0

1 8 6

7 5 4

其f值为:77

其g值为:5

其h值为:72

第39个状态是:

0 2 3

1 8 6

7 5 4

其f值为:77

其g值为:5

其h值为:72

第40个状态是:

2 8 3

7 1 6

0 5 4

其f值为:77

其g值为:5

其h值为:72

第41个状态是:

0 8 3

2 1 6

7 5 4

其f值为:77

其g值为:5

其h值为:72

第42个状态是:

2 8 3

1 0 5

7 4 6

其f值为:77

其g值为:5

其h值为:72

第43个状态是:

2 8 3

1 4 5

0 7 6

其f值为:77

其g值为:5

其h值为:72

第44个状态是:

2 4 8

1 0 3

7 6 5

其f值为:77

其g值为:5

其h值为:72

第45个状态是:

0 2 8

1 4 3

7 6 5

其f值为:77

其g值为:5

其h值为:72

第46个状态是:

2 3 4

1 8 5

7 6 0

其f值为:77

其g值为:5

其h值为:72

第47个状态是:

2 3 4

1 0 8

7 6 5

其f值为:77

其g值为:5

其h值为:72

第48个状态是:

1 2 3

7 8 4

0 6 5

其f值为:77

其g值为:5

其h值为:72

第49个状态是:

1 2 3

8 0 4

7 6 5

其f值为:77

其g值为:5

其h值为:72

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

路径长度为:5

第0层状态为:

2 8 3

1 6 4

7 0 5

第1层状态为:

2 8 3

1 0 4

7 6 5

第2层状态为:

2 0 3

1 8 4

7 6 5

第3层状态为:

0 2 3

1 8 4

7 6 5

第4层状态为:

1 2 3

0 8 4

7 6 5

第5层状态为:

1 2 3

8 0 4

7 6 5

A算法优点：

提高了效率

缺点：

对无效过程的省略没有Astar算法有效。

1. **总结**

**总体来说，启发式算法会比盲目搜索效率更高，但是需避免无法搜索到正确解的情况。**