中山大学学报（自然科学版）
第30卷 第3期 ACTA SCIENTIARUM NATURALIUM Vol.30 №.3
1991年 UNIVERSITATIS SUNYATSENI 1991

接近最优的九宫排定算法

肖金声 辛小霞
（计算中心）

摘 要

本文致力于减少九宫排定算法的环形移动，从而使平均移动次数降到不足35次．算法用Turbo Pascal验证无误．

关键词 环状初态，移动距离，偶排列，环形移动

1 引 言

对于图1所示的九宫方格，设想有8块小方格大小的小板，上面分别刻有1，2，…，8这8个数字．把这8块小板任意放入九宫方格，空一格，形成一个初态，如图2．这时，与空格相邻的那些小块中可以有一块移到空格位置而留下新的空格．这样便可以不断地进行移动。

九宫排定问题是，对于任给的初态，通过移动其中的数字板，达到图3所示的目标态，例如，图2的初态经3，2，8，7，6，5，4，3，2等9次移动而达目标态．

为了指称小方格，分别给以标记A，B，…，I．对任一I位置不空的初态，都可以经一次或两次移动，使I位置为空，成为环状初态．由于移动是可逆的，我们只须对环状初态讨论可解性就够了．

从位置A开始，按顺时针方向把环状初态的8个数字依次排列起来，称为该初态的原排列，记作S_1，S_2，S_3，S_4，S_5，S_6，S_7，S_8．对其中任一数字S_i($1<i\leqslant 8$)，令t_i是其左方比它大的数字的个数，称为S_i的逆序数．这时，和数

$$T=\sum_{1}^{8} t_i$$

称为原排列的逆序数．我们称T为偶数的原排列为偶排列，否则为奇排列．

A	B	C
H	I	D
G	F	E

图1 九宫
Fig.1 Nine lattices

1		4
2	3	5
8	7	6

图2 一种初态
Fig.2 A initial state

1	2	3
8		4
7	6	5

图3 目标态
Fig.3 The goal state

本文1989年12月15日收到

文[1]证明了下列两个可解性定理。

定理1　一切偶排列可以移成目标态，一切奇排列不能移成目标态。

定理2　对于9!种初态，恰有一半能移成目标态。

为了描述算法，本文沿用文[1]的办法，把九宫方格换成有9个顶点的图，以小方格为顶点，相邻的方格之间以弧相连，如图4。这时，A，C，E，G是二度顶点，B，D，F，H是三度顶点，I是四度顶点。

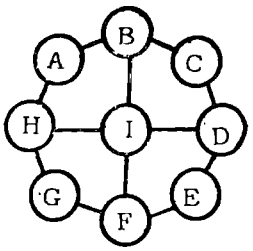

图4　用图表示九宫

Fig.4　Nine lattices in diagram

i对j的移动距离d是指外环上按逆时针方向，i到j之间所夹的顶点数。如图4中E对B的移动距离是2。

2　移动算法

算法的基本思想与文[2]和[3]相同：先把2移成1的后继；当1，…，i－1(i＝2，…，7)已顺时针相邻，便设法把i移成i－1的后继，并保持1，…，i的顺时针相邻。然后，8自然就在7的后继位置上。最后，若1不在A位置，便进行相应的环形移动。本算法的不同之处在于：

(1)随时考虑采用有利于使1靠近A位置的移动方式。即使这样做当时会增加移动次数，但可从避免环形移动而得到补偿。

(2)在移动i的时候，若有可能，也兼顾i＋1，甚至i＋2。

整体算法如下：

```
begin
  读入初态
  if 初态不合法 then goto 99;
  将初态环状化;
  if 不是偶排列 then goto 99;
  for j:=1 to 6 do
  begin
    求j+1到j的移动距离d;
    case d of
        1: move1;
        2: move2;
        3: move3;
        4: move4;
        5: move5;
        6: move6;
    end
  end;
  if 1不在A位置 then 做相应环形移动
99;
end
```

下面逐一讨论几个主要移动过程。

2.1 环状化过程 RING

本来把非环初态移成环状只需常规的一或二次移动。为了减少总的移动次数，本算法增加了一些特殊处理：当Ⅰ位置为1或2时，及时排好1，2的相邻位置，或为此作好准备；当1在E位置时，将1适当调离E位置。

```
procedure ring;
begin
  if Ⅰ位置非空 then
    begin
      if Ⅰ位置为1 then 把1移到2之前;
      if Ⅰ位置为2 then 把2移到1之后;
      if 1在E位置
      then 把1适当调离E位置
      else 作常规移动
    end
end
```

2.2 移走Ⅰ位置元素的过程 FROM_K8

此算法在文[3]中已叙述过，此处从略。

2.3 d=1时调用的过程 MOVE1

情形1，i在二度顶点，如图5。这时有多种移动方式：

（1）若$S_1=j+2$，且1在B，C，D三位置之一，便按S_1，i，S_2，S_3，S_4，S_5，S_6，j，i，S_1反时针转动。共移动10次。

（2）若$S_1=j+2$，但1不在B，C，D三位置，则按S_1，j，S_6，S_5，S_4，S_3，S_2，S_1顺时针移动，共移动8次。

此外，在作上述两种移动之前，若S_3，S_4或S_6为$j+3$，则先将其移到S_2的位置。

（3）若$S_1\neq j+2$，且1在F，G，H三位置之一，则视j之大小从S_1开始分别取前面3点、5点或7点作顺时针移动。例如，当j=1，2时，按S_1，j，S_6，S_1移动。

（4）若$S_1\neq j+2$，但1不在F，G，H三位置，则按S_1，i，S_2，S_1移动。在此之前，若S_3，S_4或S_6为$j+2$，则先将其移到S_2的位置。

情形2 i在三度顶点，如图6。这时有两种移动方式：

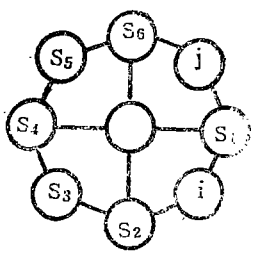

图5 d=1，i在二度顶点的情况

Fig.5 The case in d=1 and i is in a node with two degrees

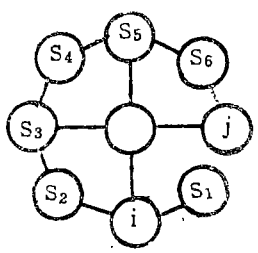

图6 d=1，i在三度顶点的情况

Fig.6 The case in d=1 and i is in a node with three degrees

（1）若1在F，G，H三位置之一，按i，S_1，j，S_6，S_5，S_4，S_3，S_2，S_1，i 顺时针转动，共移动10次。

（2）若1不在F，G，H三位置，则按i，S_2，S_3，S_4，S_5，S_6，j，i逆时针转动，共移动8次。

除开对j+2的考虑不论，MOVE1的平均移动次数为：

$$\frac{5}{8}\times\frac{4+8}{2}+\frac{3}{8}\times\frac{6+10}{2}=6.75\text{次}。$$

2.4　d=2时调用的过程MOVE 2

情形1　i在二度顶点，如图7。这时有多种移动方式：

（1）若1在B，C，D三位置之一，按S_3，i，S_2，S_3，i，S_4，S_5，S_6，j，i移动。共移动10次。

（2）若1不在B，C，D三位置，则视j之大小作不同的移动。j=1时，按j，S_1，S_2，j移动；j=2，3时，按S_2，S_1，j，S_6，S_5，S_2，S_1，i，S_3，S_1移动；j=4，5时，按S_2，S_1，j，S_6，S_5，S_4，S_3，S_2，S_1，i，S_2，S_1移动。分别移动4，10，12次。

此外，当S_1或S_2为j+2时，有特殊的移动方式，不赘述。

情形2　在i三度顶点，如图8。按i，S_2，S_1，i移动，共移动4次。

除开S_1或S_2是j+2的情况不论，求得MOVE2的平均移动次数是

$$\frac{1}{2}\times 4+\frac{1}{2}\left(4\times\frac{1}{5}+10\times\frac{2}{5}+12\times\frac{2}{5}\right)=6.8\text{次}。$$

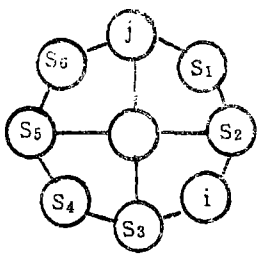

图7　d=2，i在二度顶点的情况

Fig.7　The case in d=2 and i is in a node with two degrees

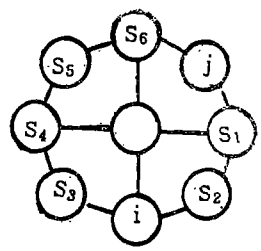

图8　d=2，i在三度顶点的情况

Fig. 8　The case in d=2 and i is in a node with three degrees

2.5　d=3时调用的过程MOVE3

情形1　i在二度顶点，如图9。这时，有两种移动方式：

（1）一般按S_3，i，S_4，S_3，i，S_2，S_1，i移动，共移动8次。

（2）若j=4，则按S_3，i，S_4，S_3，i，S_2，S_1，i，S_3，S_4，S_2，S_3移动，共移动12次。

此外，若S_2或S_3是j+2，则先做S_1，S_2，S_3，S_1或S_3，S_2，S_1，S_3移动，再做类似的8或12次移动。

情形2　i在三度顶点，如图10。这时有多种移动方式：

（1）若j=1，$S_1 \neq 3$，且1在D或F位置，按j，S_6，S_5，S_4，i，j移动。共移动6次。

（2）否则，一般按i，S_4，S_5，S_6，j，i移动。共移动6次。

（3）但若1在G或H位置之一，且j<4，则按S_2，S_1，j，S_6，S_5，S_2，i，S_3，S_1，i移动，共移动10次。

此外，若$S_5=j+2$或$S_1=j+2$，$S_2=j+3$时，有特殊的移动次序，从略。

除开对j+2的考虑，求得MOVE3的平均移动次数为：

$$\frac{1}{2}\left(\frac{1}{4}\times 12+\frac{3}{4}\times 8\right)+\frac{1}{2}\left(\frac{6}{8}+\frac{7}{8}\left(\frac{1}{4}\times 10+\frac{3}{4}\times 6\right)\right)\approx 8\text{次}.$$

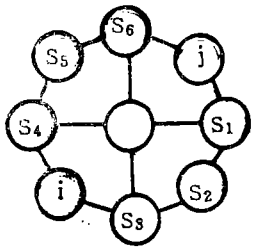

图9 d=3，i为二度的情况
Fig. 9 The case in d=3 and i is in a node with two degrees

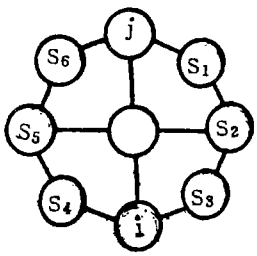

图10 d=3，i为三度顶点的情况
Fig.10 The case in d=3 and i is in a node with three degrees

2.6 d=4时调用的过程MOVE4

情形1 i在二度顶点，如图11。这时，视j的大小和1所处的位置有多种移动方式：

（1）若j=1，且处在D或F位置，按j，S_1，S_2，S_3，S_4，j移动。共移动6次。

（2）若j=1，但不在D，F位置，或者$j=2$，就按S_5，i，S_4，S_5，i，S_6，j，i移动，共移动8次。

（3）若j=3，且1在F，G，H三位置之一，按S_2，S_1，j，S_6，S_5，i，S_4，S_2，i，S_4，S_2，S_3，S_1，i移动，共移动14次。

（4）若j=3，但1不在F，G，H三位置，则按S_4，i，S_5，S_6，j，S_4，i，S_3，S_2，S_1，S_4，i移动，共移动12次。

情形2 i在三度顶点，如图12。这时，一般按i，S_4，S_3，S_2，S_1，i移动。共移动6次。但若S_2或S_3为j+2，则先将其移到S_1的位置，再作类似的6次移动，从略。

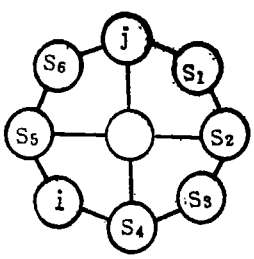

图11 d=4，i在二度的情况
Fig.11 The case in d=4 and i is in a node with two degrees

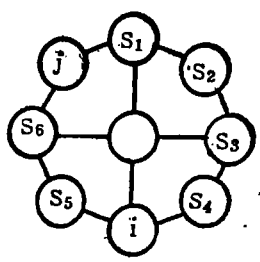

图12 d=4，i在三度的情况
Fig.12 The case in d=4 and i is in a node with three degrees

除开对j+2的考虑，求得MOVE4的平均移动次数为：

$$\frac{1}{2}\times 6+\frac{1}{2}\left(\frac{1}{3}\times\frac{6+8}{2}+\frac{8}{3}+\frac{1}{3}\left(\frac{14+12}{2}\right)\right)=7.66\text{次}.$$

2.7 d=5时调用的过程MOVE5

情形1　i在二度顶点，如图13。移动的办法是先做五点旋转，再按i在三度顶点处理。做五点旋转有两个方向：

(1)若1在F,G，H三位置之一，按S_1，j，S_6，i，S_5,S_1顺时针旋转，移动6次。

(2)若1不在F,G，H三位置，则按S_5，i，S_6，j，S_1，S_5逆时针旋转，移动6次。若S_3或S_5为j+2，则先将其移到S_1的位置，再作逆时针旋转。

情形2　i在三度顶点，如图14。这时有两种移动方式：

(1)若S_6=j+2，或者j=1不在H位置，且$S_1\neq j+2$，按j,S_6，i，j移动。共4次。

(2)否则，按i，S_6，j，i移动，也移动4次。

除开对j+2的考虑，求得MOVE5的平均移动次数为：$\frac{1}{2}(10+4)=7$次。

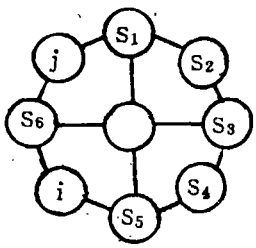

图13　d=5，i在二度顶点的情况

Fig.13　The case in d=5 and i is in a node with two degrees

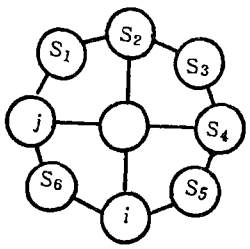

图14　d=5，i在三度顶点的情况

Fig.14　The case in d=5 and i is in a node with three degrees

2.8 d=6时调用的过程MOVE6

情形1　i在二度顶点，如图15。这时一般按j，S_1，S_2，S_3，S_4，S_5，S_6，j移动，共移动8次；若j=1在H位置，则按j，i，S_6，S_5，S_4，S_3，S_2，S_1，i，j移动。共移动10次。

情形2　i在三度顶点，如图16。这时，一般按i，S_6，S_5，S_4，S_3，S_2，S_1，i移

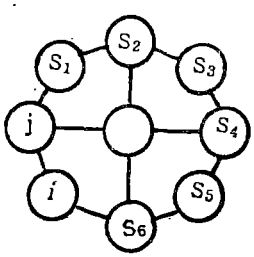

图15　d=6，i在二度顶点的情况

Fig.15　The case in d=6 and i is in a node with two degrees

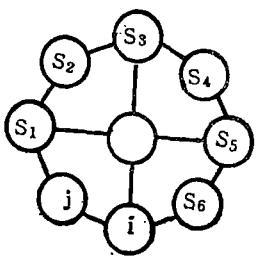

图16　d=6，i在三度顶点的情况

Fig.16　The case in d=6 and i is in a node with three degrees

动，共移动8次。此外，若$S_3=j+2$，可先做S_3，S_2，S_1，S_3，移动；若$S_5=j+2$，可先做S_5，S_4，S_3，S_2，S_1，S_5，移动。然后再进行类似的8次移动。

除开对j+2的考虑，求得MOVE6的平均移动次数是：

$$\frac{8}{2}+\frac{1}{2}\left(\frac{3}{4}\times 8+\frac{1}{4}\times 10\right)=8.25\text{次}.$$

3 算法的平均移动次数

这里，只讨论平均移动次数，而不讨论最坏情况移动次数。因为最坏情况移动次数不可能达到，意义不大。

算法中可能的移动可分为3个部份：环状化，按顺时针排递增顺序，以及最后的环形移动。

把非环状初态排成环状，平均移动1.5次。非环状初态出现的概率为8/9。故环状化的平均移动次数是4/3次。这里不计入对1或2特殊处理的移动次数，因为它不会超过把2移到1后所需平均移动次数。

已知，各MOVEi的平均移动次数V_i分别为：$V_1=6.75$，$V_2=6.8$。$V_3=8$，$V_4=7.66$，$V_5=7$，$V_6=8.25$。

注意，各MOVEi中为考虑j+2而增加的移动次数没有计算在内，因为其所增不会超过j+2单独移动时的平均移动次数。

排递增顺序由总体算法的FOR语句完成。执行第j次循环时，数字1到j已排好。这时，j+1可能距j为0，1，…，7—j。因此，第j次循环的平均移动次数为

$$L_j=(V_1+V_2+\cdots+V_{7-j})/(8-j).$$

所以，执行该FOR语句的平均移动次数为：

$$(6.75+6.8+8+7.66+7+8.25)/7+(6.75+6.8+8+7.66+7)/6+(6.75+6.8+8+7.66)/5+(6.75+6.8+8)/4+(6.75+6.8)/3+6.75/2=31.52$$

由于算法中处处采用使1向A位置靠近的移动方式，除初态为5 6 7 8 1 2 3 4外，已不会再有EMOVE出现，故可忽略不计；基于同样的理由，需要移动16次的环形移动也大幅度减少。我们保守地估计有7/8不需环形移动，故环形移动的平均移动次数为16/8=2。

于是，整个算法的平均移动次数应为：1.3+31.52+2=34.82

4 实例比较

由表1可以看出，33个实例中，有18个减少了移动次数，其余的也没有增加。需要环形移动的实例由17个降为2个，其中1个是初态如此。总移动次数从1196降为854。平均次数从36.24降为25.88。下降了28.6个百分点。尽管如此，仍无法证明此算法是最优的。

表1　两个算法的结果比较*（对照）

Tab.1　A comparison of results for two algorithms

输入初态	文[3]的移动次数		本文的移动次数		输入初态	文[3]的移动次数		本文的移动次数	
012345678	8	AMOVE	8	AMOVE	357812640	18	GMOVE	14	GMOVE
120345678	6	AMOVE	6	AMOVE	367812450	30	AMOVE	30	AMOVE
561327480	40	EMOVE	30	AMOVE	156327480	20	GMOVE	26	AMOVE
132485670	18	CMOVE	20	AMOVE	761254380	24	EMOVE	24	AMOVE
132867540	24	AMOVE	24	AMOVE	674835210	42	AMOVE	38	AMOVE
712654380	30	AMOVE	30	AMOVE	104567823	9	AMOVE	9	AMOVE
612574380	18	AMOVE	18	AMOVE	650342178	46	EMOVE	40	AMOVE
715438260	32	AMOVE	28	AMOVE	821437605	29	AMOVE	29	AMOVE
712546380	20	AMOVE	20	AMOVE	357406182	44	EMOVE	44	AMOVE
683524170	48	GMOVE	34	AMOVE	618457230	34	GMOVE	44	AMOVE
214387605	35	AMOVE	35	AMOVE	546321807	35	AMOVE	35	AMOVE
645237180	28	EMOVE	36	AMOVE	087563421	34	EMOVE	30	AMOVE
075863421	22	GMOVE	28	AMOVE	125670834	11	GMOVE	17	AMOVE
256710834	25	EMOVE	33	AMOVE	876543210	44	CMOVE	38	AMOVE
234567180	8	GMOVE	10	AMOVE	123456780	0	AMOVE	0	AMOVE
145826730	20	AMOVE	20	AMOVE	345678120	0	GMOVE	0	GMOVE
367581240	24	AMOVE	24	AMOVE					

● 环形移动AMOVE需0次移动；CMOVE和GMOVE需16次；EMOVE需30次

参 考 文 献

1　曹新谱。计算机学报，1983；（4）：252～257

2　黄琳。计算机工程与应用，1985；（9）：30～37

3　肖金声。计算机工程与应用，1986；（8）：58～64

A Scheduling Algorithm in 9-lattices Closing to Optimization

*Xiao Jinsheng** *Xin Xiaoxia*

Abstract

This paper devotes to reducing the number of times of movements for the scheduling algorithm in 9-lattices, as such the average number of times of movements is small then 35. The algorithm was verified in Turbo Pascal.

Keywords ring initial state, even permutation, movement distance, ring movement

● Computer Centre