第5卷第3期
1994年3月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 5, No. 3
Mar. 1994

关于迷宫排序问题的研究*

吴向军 陈国良

（中山大学计算机系，广州 510275） （中国科学技术大学计算机科学系，合肥 230026）

摘要 本文首先把迷宫排序问题推广为 m×n 迷宫(m>1,n>1)的排序问题，证明了 m×n 迷宫的任一初始状态能经过有限步移动转变成目标状态的充要条件，然后给出了一个 m×n 迷宫排序的算法，该算法的时间复杂度是 O(mn(m+n))，空间复杂度是 O(mn). 最后还指出了它的时间复杂度的一个下界. 这样，关于迷宫排序问题就基本上得到了圆满地解决.

关 键 词 m×n 迷宫(m>1,n>1)，迷宫状态序列，数字 i 的逆序数 l(i)，迷宫状态的逆序数 r(m,n)，迷宫状态的特征值 f(m,n).

1 预备知识

迷宫排序问题的存在已有一百多年了，该问题的原始描述是：在一个 4×4 的棋盘中，将数字 1,2,…,15 以任意的顺序放入该棋盘的方格中，其中空出一格(如图 1(a)所示). 排序的问题是对任意一给定的迷宫状态，希望经过有限次移动，把它转变成目标状态(如图 1(b)所示). 移动的规则是：每次只能在空格周围的四个数字中任选一个移入空格中，当空格位于角或边时，只有二种或三种的移动可能.

10	1	11	14
2	3		5
8	9	4	15
12	13	6	7

(a)初始状态

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

(b)目标状态

图1 4×4迷宫的排序问题

1	1	2	…	n−1	n
2	n+1	n+2	…	2n−1	2n
…	…	…	…	…	…
m	k+1	k+2	…	k+n−1	
	1	2	…	n−1	n

(其中： k=(m−1)*n)

图2 m×n迷宫排序问题的目标状态

对于迷宫排序问题，在许多人工智能和算法设计的书刊中都有不同程度的研究，在这些研究中，有采用面向目标推理的方法，也有用启发式函数来确定下一步移动的方法. 这些方法不但要受许多因素(比如，启发式函数构造的好坏以及迷宫的初始状态)影响，而且还不可避免地需要回溯. 对于 3×3 迷宫和 4×4 迷宫的排序问题，这些方法在最坏情况下所需要

* 本文1991年2月12日收到，1991年11月21日定稿

作者吴向军，29岁，讲师，主要研究领域为算法设计与分析，人工智能. 陈国良，56岁，教授，主要研究领域为非数值计算的并行算法，算法理论，神经网络，智能机体系结构.

本文通讯联系人：吴向军，广州 510275，中山大学计算机系

的搜索空间分别为 9!和 16!. 由于这个搜索空间非常庞大,所以,至今为止,关于迷宫排序问题一般只讨论 3×3 迷宫或 4×4 迷宫的排序情况.

本文在研究迷宫排序问题的过程中,把该问题推广为 m×n 迷宫(m>1,n>1)的排序问题. 它的目标状态如图 2 所示.

对于 m×n 迷宫的任何一初始状态,我们把它按行排成一个序列(定义空格的数字为 m * n),该序列为:

$$\underbrace{S1\ S2\ \cdots\ Sn}_{n}\quad \underbrace{Sn+1\ \cdots\ S2n}_{n}\quad \cdots\cdots\quad \underbrace{S(m-1)n+1\ \cdots\ Smn}_{n}\quad \cdots\cdots \tag{1}$$

我们称序列(1)为迷宫状态序列.

下面定义几个与迷宫状态特征有关的基本概念.

Si 的逆序数 l(Si)是在序列(1)中,数字 Si 后面比 Si 小的数字个数.

迷宫状态的逆序数 r(m,n)是所有数字的逆序数之和. 即:$r(m,n)=\sum_{i=1}^{m*n} l(Si)$.

迷宫状态的特征值 f(m,n)是 r(m,n)与空格坐标(x,y)之和. 即:$f(m,n)=r(m,n)+x+y$.

对于图 1(a),它的迷宫状态序列如下:

10　1　11　14　2　3　16(blank)　5　8　9　4　15　12　13　6　7

这时有:$l(1)=l(2)=l(3)=l(4)=l(6)=l(7)=0, l(5)=1, l(8)=l(9)=3, l(10)=l(16)=9, l(11)=8, l(12)=l(13)=2, l(14)=10, l(15)=4$.

图 1(a)的逆序数 $r(4,4)=l(1)+l(2)+\cdots+l(16)=51$.

图 1(a)的特征值 $f(4,4)=r(m,n)+2+3=56$.(空格的为置为(2,3))

2　解存在的充要条件

对于 m×n 迷宫的任一初始状态,它是不一定能经过有限步移动转变成目标状态的. 对于能转变成目标状态的初始状态,我们称它的解存在. 对于 3×3 迷宫排序问题,曹新谱[1]在 1981 年用无向图的方法证明了它的任一初始状态解存在的条件,在该证明过程中,他引入了一些特定的概念,从而使得该证明方法很难推广. 另外,在文献[2]中,虽然有关于 4×4 迷宫排序问题解存在的条件,并说可以利用[1]的结果来证明,但未见其证明过程.

下面对于 m×n 迷宫排序问题,给出它的任一初始状态解存在的充要条件,并给予简单、直观地证明.

引理 1. 在迷宫状态序列中,任意去掉一已在目标位置的数字,该迷宫状态逆序数的奇偶性保持不变.

证明:假设:迷宫状态的逆序数为 r(m,n),该状态的序列为 $S_1, S_2, \cdots, S_{mn}$,去掉其中已在目标位置的数字为 Si,Si 的逆序数 l(Si)为 t,去掉该数字后的逆序数为 $r'(m,n)$.

因为数字 Si 已经在目标位置且迷宫中的数字是唯一的,所以,如果在数字 Si 后面有 t 个数字比它小,那么在 Si 前面就一定有 t 个数字比它大. 这就使得:在迷宫状态序列中去掉数字 Si,将会有 t 个数字的逆序数减少 1. 这样,在迷宫状态序列中去掉数字 Si,迷宫状态的

逆序数将总共减少

$$t+\underbrace{1+1+\cdots+1}_{t}=2*t,\text{即}:r(m,n)=r'(m,n)+2t.$$

故有 $r(m,n)=r'(m,n)\ \mathrm{mod}\ 2.$ 证毕.

引理 2. 对于任意 $m\times n$ 迷宫,空格的任何移动都不改变迷宫状态特征值的奇偶性.

证明:假设:当前迷宫状态的逆序数为 $r(m,n)$,该迷宫状态的特征值为 $f(m,n)$,空格位于(x,y)方格中.空格移动一步后,迷宫状态的逆序数改变为 $r'(m,n)$,迷宫状态的特征值改变为 $f'(m,n)$.

由于空格向上和向下移动,向左和向右移动是对称的,所以下面只考虑空格向左和向上这两种移动情况.

1. 空格向左移动一步

由于空格的数字定为 $m*n$,所以它向左移动一步,它的逆序数将增加 1.其它数字的逆序数则保持不变,这时空格的位置改变为$(x,y-1)$.空格向左移动一步的情况如图 3 所示.

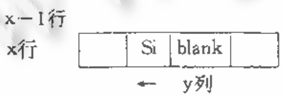


图3 空格向左移动一步示意图

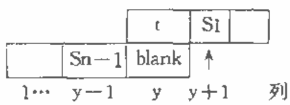


图4 空格向上移动一步示意图

此时有 $r'(m,n)=r(m,n)+1$

$$\begin{aligned}f'(m,n)&=r'(m,n)+x+(y-1)\\&=r(m,n)+1+x+y-1\\&=f(m,n)\end{aligned}$$

故 $f'(m,n)=f(m,n)\ \mathrm{mod}\ 2$

2. 空格向上移动一步

假设:空格上方的数字是 t,在从方格$(x-1,y+1)$到方格$(x,y-1)$之间的 $n-1$ 个数字中,有 $x1$ 个数字比 t 小,有 $x2$ 个数字比 t 大(其中:$x1+x2=n-1$).空格向上移动一步的情况如图 4 所示.

空格向上移动一步,它的逆序数将增加 n,空格的位置改变为$(x-1,y)$.

数字 t 向下移动一步后,它的逆序数将减少 $x1$.另外,由于在方格$(x-1,y+1)$和$(x,y-1)$之间有 $x2$ 个数字比 t 大,所以将有 $x2$ 个数字的逆序数增加 1.这样空格向上移动一步,迷宫状态逆序数的修改量为:$n-x1+x2$.即:$r'(m,n)=r(m,n)+n-x1+x2$.

所以有 $f'(m,n)=r'(m,n)+(x-1)+y$

$$\begin{aligned}&=r(m,n)+n-x1+x2+x+y+1\\&=r(m,n)+x+y+n-1-x1+x2\qquad(\text{其中 } x1+x2=n-1)\\&=f(m,n)+2*x2\end{aligned}$$

故 $f'(m,n)=f(m,n)\ \mathrm{mod}\ 2$ 证毕.

定理 1. 对于 $m\times n$ 迷宫$(m>1,n>1)$的任何一初始状态,它能经过有限步移动转变成

目标状态的充要条件是：该初始状态的特征值 f(m,n)与 m+n 同奇偶.

证明：1. 必要条件

假设：m×n 迷宫的一个初始状态的特征值为 f(m,n)，它能经过有限步移动，最终到达目标状态. 由迷宫状态特征值的定义可知：迷宫目标状态的特征值 $f_0(m,n)=m+n$，由引理 2 可得：空格移动有限步后，迷宫状态特征值的奇偶性保持不变.

所以有　$f(m,n)=f_0(m,n) \bmod 2$

故　$f(m,n)=(m+n) \bmod 2$

2. 充分条件

假设：m×n 迷宫的一个初始状态的特征值 f(m,n)与 m+n 同奇偶，下面用数学归纳法来证明：该迷宫状态一定能转变成目标状态.

(1)当 m=n=2 时

对于任何符合条件的初始状态，它一定能经过若干步移动达到图 5 所示的目标状态.

1	2
3	

图5 2×2迷宫的目标状态

行				
1	1	2	…	n
2	S1	S2	…	Sn
…	…	…	…	…
m+1	Sk+1	Sk+2	…	Smn
	1	2	…	n 列

图6 (m+1)×n迷宫排序的中间状态

1	1	S1	…	Sn
2	n+1+1	Sn+1	…	S2n
…	…	…	…	…
m	k+1	Sk+1	…	Smn
	1	2	…	n+1 列

图7 m×(n+1)迷宫排序的中间状态

(2)假设：对 m×n 迷宫(m>1,n>1)排序问题，如果它的一个状态的特征值 f(m,n)与 m+n 同奇偶，那么该状态一定能转变成目标状态.

现在来证明对(m+1)×n 迷宫和 m×(n+1)迷宫的排序问题，上述命题同样成立.

(a)(m+1)×n 迷宫的排序情况

假设：(m+1)×n 迷宫的一个初始状态的特征值 f(m+1,n)与 m+1+n 同奇偶.

这时，我们可以经过有限步移动，把原状态转变成第一行与目标状态完全相同的中间状态(数字的移法在后面的算法中有详细叙述，在此从略). 转变后的中间状态如图 6 所示.

不妨再假设：此时的迷宫状态的逆序数为 $r'(m+1,n)$，迷宫状态特征值为 $f'(m+1,n)$，空格的位置为(x,y)，由引理 2 可得：

$$f'(m+1,n)=f(m+1,n) \bmod 2$$

又由前面的假设可知：$f(m+1,n)=(m+1+n) \bmod 2$

所以　$f'(m+1,n)=(m+n+1) \bmod 2$

由于第一行已排好，去掉该行不会改变原问题的性质. 对于剩下的 m×n 迷宫排序问题，令它的逆序数为 $r''(m,n)$，它的特征值为 $f''(m,n)$，空格的位置修改为(x−1,y). 这时，由引理 1 可得　$r''(m,n)=r'(m+1,n) \bmod 2$

$$\begin{aligned}\text{此时有}\quad f''(m,n)&=r''(m,n)+(x-1)+y\\&=r'(m+1,n)+x+y-1 \quad \bmod 2\\&=f'(m+1,n)-1 \quad \bmod 2\end{aligned}$$

$=m+n+1-1 \mod 2$

$=m+n \mod 2$

由归纳假设可知：对于剩下的 $m\times n$ 迷宫状态，它能够经过有限步移动到达 $m\times n$ 迷宫的目标状态. 现在加上已排好的第一行，就得到了 $(m+1)\times n$ 迷宫的目标状态.

故　对 $(m+1)\times n$ 迷宫排序问题，命题成立.

(b) $m\times(n+1)$ 迷宫的排序情况

假设：$m\times(n+1)$ 迷宫的一个初始状态的特征值 $f(m,n+1)$ 与 $m+n+1$ 同奇偶.

这时，我们同样可以经过有限步移动，把原状态转变成第一列与目标状态完全相同的中间状态（数字的移法与情况(a)是一致的，在此从略）. 转变后的中间状态如图 7 所示.

再假设：此时迷宫状态的逆序数为 $r'(m,n+1)$，迷宫状态的特征值为 $f'(m+1,n)$，空格的位置为 (x,y). 由引理 2 可得：$f'(m,n+1)=f(m,n+1) \mod 2$

又由前面的假设可知：$f(m,n+1)=(m+n+1) \mod 2$

所以　$f'(m,n+1)=(m+n+1) \mod 2$

由于第一列已排好了，去掉该列不会影响原问题的性质. 对于剩下的 $m\times n$ 迷宫的排序问题，不妨设：迷宫状态的特征值为 $f''(m,n)$.

与情况(a)一样可证得：$f''(m,n)=(m+n) \mod 2$

由归纳假设可知：对于剩下的 $m\times n$ 迷宫状态，它能够经过有限步移动到达 $m\times n$ 迷宫的目标状态. 现在加上已排好的第一列，就得到了 $m\times(n+1)$ 迷宫的目标状态.

故　对于 $m\times(n+1)$ 迷宫排序问题，命题成立.

综合上面两种情况可得：对于 $m\times n$ 迷宫 $(m>1,n>1)$ 的排序问题，如果它的一个状态的特征值 $f(m,n)$ 与 $m+n$ 同奇偶，那么该状态一定能转变成目标状态.　　证毕.

根据上述定理，我们很容易得到下面的推论.

推论. 对于 $n\times n$ 迷宫 $(n>1)$ 的任何一初始状态，它能经过有限步移动转变成目标状态的充要条件是：该初始状态的特征值 $f(n,n)$ 是偶数.

3　m×n 迷宫的排序算法

从上面定理的证明过程中，我们不难看出 $m\times n$ 迷宫排序算法的总体思想和所采用的移动策略.

$m\times n$ 迷宫排序算法的总体思想如图 8 所示.

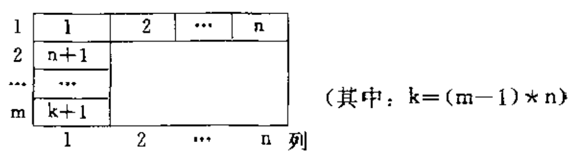


图8　m×n迷宫排序过程示意图

(1) 当 $m\geqslant n$ 时

排好最上面的若干行,然后向下缩小,直至剩下迷宫的行数小于列数.

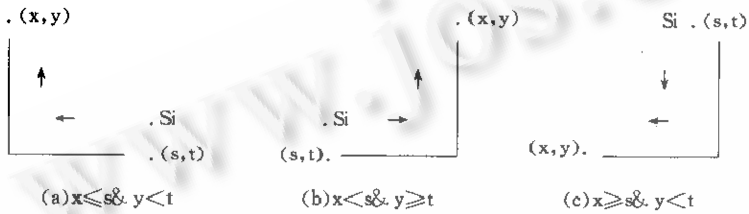
(2)当 $m<n$ 时

排好最左边的若干列,然后向右缩小,直至剩下迷宫的列数等于行数.

迷宫排序过程中所采用的移动策略是:在把某个数字从当前位置移到它的目标位置的过程中,遵守不破坏已排好数字位置的原则.

迷宫排序过程中的最基本问题是如何排好某一指定的数字的问题,排好某一行或某一列数字的问题只不过是连续排好若干个指定的数字而已.所以下面只分析在迷宫排序过程中排好某一指定数字所要移动的情况.

假设:当前所指定的数字是 Si,它的当前位置是(s,t),目标位置是(x,y).在把数字从当前位置移到目标位置的过程中,总共有 3 种直线移动组合.这些移动组合情况如图 9 所示.



(a)x≤s& y<t　(b)x<s& y≥t　(c)x≥s& y<t

图9 数字Si移动情况示意图

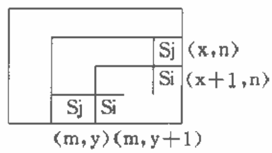


图10 迷宫排序过程中的特殊情况示意图

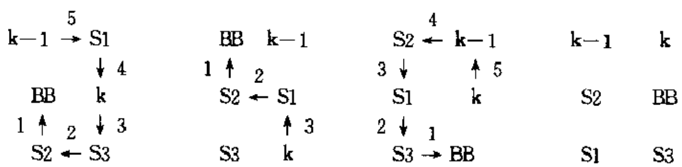
对于迷宫排序过程中的一般移动情况,读者结合图 9 和"不破坏已排好数字位置"的原则是不难想象出来的,空格所移动的步数不会超过 $6(|x-s|+|y-t|)$(详见后面的算法分析).除一般情况外,还要考虑以下二种特殊情况.

1. 数字 Si 的目标位置是(x,n),而 Si 的当前位置(也可能是被移到的位置)是(x+1,n),并且空格不在(x,n)中.如图 10 的右上方所示.

由于 Sj 左边的数字都是已排好的,它不能按一般方法来处理.对于这种情况,又要分迷宫的"宽度"为 2 和大于 2 两种情况(在理论上,"n>2"的情况可以不考虑,这里只是为了减少移动步数),排好它们是可以在固定步内完成的(不考虑空格的初始位置).移动情况如图 11 所示 (BB 为空格).

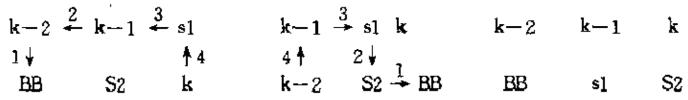
2. 数字 Si 的目标位置是(m,y),而 Si 的当前位置(也可能是被移到的位置)是(m,y+1),并且空格不在(m,y)中.如图 10 的左下方所示.

由于 Sj 上方的数字都是已排好的,它不能按一般方法来处理,排好它也是可以在固定步内完成的(不考虑空格的初始位置).移动情况如图 12 所示.



(a)n=1(13)步

(本文原来的方法需要移动 16 步,审稿者改进为 13 步.)



(b)n>2(8步)

图 11 在 n 列排好 Si 的移动示意图

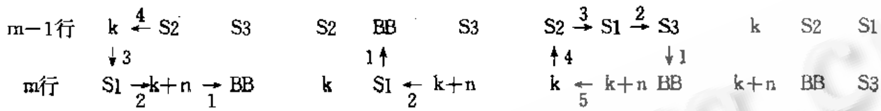


图 12 在 m 行排好 Si 的移动示意图(11 步)

根据数字移动情况的上述分析，把该迷宫排序算法形式化是很容易的，所以本文省去对该算法的形式描述，下面来分析该迷宫排序算法的时间复杂度和空间复杂度.

假设：在把 $m\times n$ 迷宫的任一初始状态转变成目标状态的过程中，空格所需要的移动步数为 $T(m,n)$. 现在来分析把一个数字从当前位置移到目标位置所需要的移动的步数.

不妨再设：当前移动的数字是 Si，它的当前位置是 (s,t)，目标位置是 (x,y)，当前空格位置是 $(b1,b2)$，数字 Si 到达目标位置时空格所需移动的步数为 $B(Si)$.

移动一指定数字有以下三种情况.

(1)(x,y)不在最后一列上并且也不在最后一行上

由图 9 的移动路线可知：数字 Si 从(s,t)移到(x,y)总共需要移动$|x-s|+|y-t|$步. 除了在它的第一次移动时空格需要移动$|s-b1|+|t-b2|\pm 1$ 步外(极少情况，需要移动$|s-b1|+|t-b2|+4$ 步)，其余的每次移动，空格只需要移动 5 步.

所以有
$$\begin{aligned} B(Si) &< |s-b1|+|t-b2|+4+5*(|x-s|+|y-t|-1) \\ &\ll m+n+4+5*(m+n-1) \\ &= O(m+n) \end{aligned}$$

(2)(x,y)在最后一列上

在数字 Si 的前$|x-s|+|y-t|-1$ 步移动过程中，空格所需要的移动步数如情况(1)中的分析. 数字的最后一步移动是属于特殊情况(1)，处理它，空格需要移动 13 步(当 $n=2$时)或 8 步(当 $n>2$ 时).

从而可得
$$\begin{aligned} B(Si) &< \begin{cases} |s-b1|+|t-b2|+4+5*(|x-s|+|y-t|-2)+13 & (n=2) \\ |s-b1|+|t-b2|+4+5*(|x-s|+|y-t|-2)+8 & (n>2) \end{cases} \\ &\ll \begin{cases} m+n+4+5*(m+n-2)+13 & (n=2) \\ m+n+4+5*(m+n-2)+8 & (n>2) \end{cases} \\ &= \begin{cases} 6(m+n)+7 & (n=2) \\ 6(m+n)+2 & (n>2) \end{cases} \end{aligned}$$

故 $B(Si)=O(m+n)$

(3)(x,y) 在最后一行上

在数字 Si 的前 $|x-s|+|y-t|-1$ 步移动过程中,空格所需要的移动步数如情况(1)中的分析. 数字的最后一步移动是属于特殊情况(2),处理它,空格需要移动 11 步.

这样就有

$$B(Si)<|s-b1|+|t-b2|+4+5*(|x-s|+|y-t|-2)+11$$
$$\ll m+n+4+5*(m+n-2)+11$$
$$=O(m+n)$$

综合上面三种情况可得:不论移动迷宫中的那个数,把它从当前位置移到目标位置所花的时间不会超过 $O(m+n)$.

所以有　$T(m,n)=\sum_{i=1}^{mn}B(Si)=O(mn(m+n))$

上面对时间复杂度的分析在一些方面是比较粗略的,该迷宫排序算法的实际时间复杂度在多项式的系数上要比上面的结果好,如果要详细分析的话,可以用下面简单的递推式或更详细的递推式来分析.

$$\begin{cases} T(2,2)=C \\ T(m,n)<6(m+n)n+T(m-1,n) & (m\geq n) \\ T(m,n)<6(m+n)m+T(m,n-1) & (m<n) \end{cases}$$

其中 C 是解决 2×2 迷宫排需问题空格所需要移动步数的最大值.

虽然利用递推式分析能降低最高次幂的系数,但是不会降低算法时间复杂度的数量级,因此,不再利用递推式来详细分析该算法的时间复杂度.

在该迷宫排序算法中,除了需要存储迷宫状态所必须要的空间外,不需要大量的额外存储空间,所以该迷宫排序算法的空间复杂度是 $O(mn)$.

4　迷宫排序问题的时间复杂度下界

引理 3. 对于任意一个 $m\times n$ 迷宫,在最坏情况下,它的可移到目标状态的初始状态的逆序数是 $O((mn)^2)$.

证明:下面用构造法来进行证明.

(1)当 m,n 都是偶数时

我们可以构造出这样一个迷宫状态:把空格放入(1,1)方格中,其它的数字以递降的次序按行放入迷宫的方格中,这时所得的迷宫状态序列如下所示.

$$mn(\text{blank}),mn-1,\cdots,3,2,1$$

显然迷宫状态的逆序数 $r(m,n)=mn-1+mn-2+\cdots+2+1+0$

$$=mn(mn-1)/2$$

由 m,n 都是偶数,得到:$r(m,n)$ 也是偶数,即:$r(m,n)=(m+n) \bmod 2$

由迷宫状态特征值 $f(m,n)$ 的定义可知:

$$f(m,n)=r(m,n)+1+1$$
$$=(m+n) \bmod 2$$

故:上面构造出来的迷宫状态是可以移动到目标状态的,它的逆序数为 $O((mn)^2)$.

(2)当 m 是奇数，n 是偶数时

构造这样的迷宫状态：按照目标状态的数字排列排好第一行，对于剩下的 $(m-1)\times n$ 迷宫，此时 $m-1$，n 都是偶数，按第一种情况构造其余的数字排列. 显然此时得到的迷宫状态的逆序数也是偶数，其值为 $(m-1)n[(m-1)n-1]/2$.

由于此时的空格在(2,1)方格中，所以有

$$\begin{aligned} f(m,n) &= r(m,n)+2+1 \\ &= (m+n) \bmod 2 \end{aligned}$$

故：上面构造出来的迷宫状态同样可以移动到目标状态，它的逆序数为 $O((mn)^2)$.

当 m 是偶数，n 是奇数时，先按照目标状态的数字排列排好第一列；当 m，n 都是奇数时，则先按照目标状态的数字排列排好第一行和第一列；而对于剩下的迷宫数字再按第一情况来处理. 同样可以证明构造出来的迷宫状态是可以移动到目标状态的，它们的逆序数也是 $O((mn)^2)$.

根据上述证明，我们可以得出：不论 m，n 是奇数还是偶数，在最坏情况下，$m\times n$ 迷宫状态的逆序数都可达到 $O((mn)^2)$.　　证毕.

定理 2. $m\times n$ 迷宫排序问题的时间复杂度下界为 $\Omega(m^2n)$.

证明：在前面引理 2 的证明过程中，我们可以很容易看出：空格向左、向右、向上和向下移动一步，迷宫状态的逆序数的减少量分别为：-1、1、$-(2x+1)$ 和 $(2x+1)$，(其中 $0<x<n-1$). 所以，空格的任意一步移动，迷宫状态的逆序数减少量将小于 2n，从另外一方面来看：迷宫排序的过程就是把迷宫状态逆序数 $r(m,n)$ 减少到 0 的过程. 所以，在最坏情况下，把一迷宫初始状态移动到目标状态，空格至少要移动 $(mn)^2/2n$ 步，即 $O(m^2n)$ 步.

故：$m\times n$ 迷宫排序问题的时间复杂度下界为 $\Omega(m^2n)$.　　证毕.

从表面上来看，迷宫排序问题的时间复杂度下界只与 m 的二次方和 n 的乘积相关，其实，这只是一种假象，因为，在定理 2 的证明过程中，把空格的任意一步移动所引起的迷宫状态逆序数的减少量只简单地放大成与 n 有关的函数，而给出迷宫状态逆序数减少量的更准确的函数是比较复杂的研究工作，这将有待于以后进一步研究.

本文所给出的迷宫排序算法的时间复杂度是 $O(mn(m+n))$，当 $m=n$ 时，它达到了迷宫排序问题的时间复杂度下界，所以它是最佳的.

结束语：本文所描述的迷宫排序算法已在微机上用 C 语言实现了，它可以完成 $m\times n$ 迷宫 $(1<m<21, 1<n<21)$ 的排序问题，并且把每一步的移动在屏幕上直观地表现出来，当程序运行结束时，原迷宫状态就转变成目标状态.

虽然本文所提供的算法不是对任意一特定的迷宫状态的排序都是最佳的(设计这样的最佳算法几乎是不可能的，这是因为它要试探各种可能的移动，这样势必要导致搜索空间高达 $(mn)!$)，但是该算法对所有迷宫状态的排序都是非常令人满意的，而且对有些迷宫状态的排序还是最佳的. 在微机上的运行情况已证实了这一点.

本文所述的迷宫排序算法还存在着可进一步改进的地方，因为所给的算法在数字从当前位置移动到目标位置的过程中所采用的移动控制方式是最简单的直线式移动方式，如果把该过程中的移动控制方式改为动态的台阶式移动方式，那么可能会减少一些空格的移动步数，但是，算法的时间复杂度是不会降低的.

致谢　在本文的完稿过程中，张永民提出了极其宝贵的建议，在此表示衷心地感谢！还要非常感谢郭炜在其它方面所给予的大力帮助！

参考文献

1 曹新谱. 九宫排定问题的解法. 计算机学报，1983；6(4).

2 曹新谱. 算法设计与分析. 长沙：湖南科学技术出版社，1984.

3 卢开澄. 组合数学算法与分析. 北京：清华大学出版社，1983.

4 E. 丽奇等，李卫华等译. 人工智能引能. 广州：广东科技出版社，1986.

5 Horowitz E, Sahni. Foundamentals of computer algorithms. United States of America, 1978.

6 Sara Baase. Computer algorithms: introduction to design and analysis. Addison－Wesley Publishing Company Inc., 1978.

ON SORTING OF M×N MAZE

Wu Xiangjun

(*Department of Computer Science, Zhongshan University, Guangzhou* 510275)

Chen Guoliang

(*Department of Computer Science, University of Science and Technology of China, Hefei* 230026)

Abstract　This paper considers the problem of sorting $m\times n$ maze ($m>1$, $n>1$) and proves an sufficient and necessary condition that anoriginal state can be changed into specified_target state in finite step, for $m\times n$ maze, then presents a sorting algorithm for $m\times n$ maze, its time complexity is $O(mn(m+n))$, its space complexity is $O(mn)$, finaly, this paper gives a lower bound and shows that the algorithm is optimal where $m=n$.

Key words　$m\times n$ maze ($m>1, n>1$), state sequence, reverse order number $l(i)$ of i, reverse order number $r(m,n)$ of state, eigenvalue $f(m,n)$ of state.