“八皇后问题”思路初探

□新疆　卢建斌

国际象棋中的皇后威力很大，它可以沿棋盘上任意一条直线或斜线走动，素有“八面威风”之称。显然，双方的皇后是不能在同一行或同一列或同一条斜线上对峙的。那么，在一张空白的国际象棋棋盘上最多可以放置几个皇后并且不让它们互相攻击呢？这个问题是伟大数学家高斯在十九世纪中期提出来的，并作了部分解答，他认为可能有 76 种不同的放法。这就是有名的“八皇后问题”。

现在我们已经知道八皇后问题总共有 92 种不同的解答。如果你亲自动手在棋盘上试着找出一种放法，就一定会发现开始时几个皇后很容易放置，越到后来就越困难。而试图用手工的方法找出全部 92 种解答几乎是不可能的！由于我们的记忆有限，很可能在某个位置放过子，后来证明不行取消了，但是以后又重新放上去试探，这样就会不断地走弯路，浪费大量的时间和精力。因此，必须找到一个简易有效、有条不紊的法则才行。有了明确的法则，就可以借助于计算机来帮我们寻找答案了！

为方便起见，让我们先从四皇后谈起。若要在如图 1 所示的 4×4 棋盘中放置四个互不攻击的皇后，显然在每一行中只能放置一个而且必须放置一个皇后，问题是每行中的皇后应放在第几列？

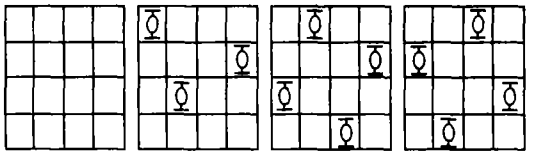

图 1　图 2　图 3　图 4

我们可以从上到下逐行地考虑：如果第一行的皇后放置在第一列，第二行的皇后就不能放置在第一、二列，只能放置在第三列或第四列。如果第二行的皇后放置在第三列，那第三行的皇后就没有安身之地了。那么暂且把第二行的皇后放在第四列，看看情况如何。这时，第三行的皇后可以放在第二列（见图 2）。然而，第四行的皇后却无家可归。要放置好第四个皇后，必须对前几个皇后的位置予以修正。

前面已经说过，第三行以及第二行的所有位置均已考虑过了，只得考虑让第一行的皇后挪动一下。如果把第一行的皇后放在第二列，第二行的皇后放在第四列上，第三行的皇后放在第一列上，第四行的皇后放在第三列上，正好四个皇后互不攻击（图 3）。如果继续这样逐行逐列地考虑下去，还可以找到四皇后问题的另一个解答（图 4）。

对于八皇后问题，也可如此解决：从上到下每行放一个皇后；在每一行中均从左至右逐列试放。能放则放（继续考虑下一行），不能放则换（列数加 1），换不成则退（退回前一行，让前一皇后所在列数加 1 后继续考虑），这就是探索八皇后问题之解答的途径。

为了记住在探索过程中每行皇后放置的位置，需要用到栈这种数据结构。我们可以用数组 T 来作栈，在 T(P) 中存放第 P($1 \leq P \leq 8$) 行皇后所在的列数。例如把第一行的皇后放在第三列，则可令 T(1) = 3。

由于我们在每行中只放一个皇后，故而皇后之间保证不会左右相互攻击了；那么如何检查它们在上下或斜线方向是否会相互攻击呢？

我们用一个数组 C 来记录每一列上是否放了皇后，不管在哪一行上的第 i($1 \leq i \leq 8$) 列上放了一个皇后，就令 C(i) = 1；若把放在第 i 列上的皇后取走放到别的位置上去了，就令 C(i) = 0。这样，我们只要查看 C(i) 是否为 0，就可以知道第 i 列上是否可以放置皇后了！

在 8×8 的国际象棋棋盘上，有 15 条主对角线（如图 5）和 15 条副对角线（如图 6）。每放一个皇后，它所

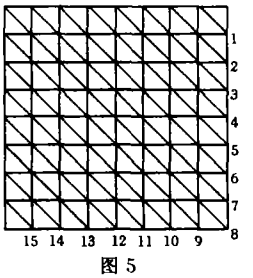

图 5

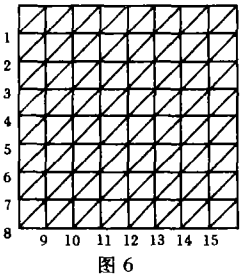

图 6

在的两条对角线上就不能再放其它皇后了。我们可以如前所述，再引进两个数组 S、M（每个数组有 15 个分量），分别记录相应的主对角线和副对角线上是否已经放置了皇后。若在第二行第三列放了一个皇后，则应令 S(7) = 1，M(4) = 1，表示第 7 条主对角线、第 4 条副对角线上已有皇后，以后不能再放置了。而主对角线数 7 可由下式计算得到：

8+行数-列数=8+2-3=7

副对角线数4可由下式得到：

行数+列数-1=2+3-1=4

因此，一般地说，在第P行、第T(P)列放置皇后以后，应该令：

C(T(P))=1

S(8+P-T(P))=1

M(P+T(P)-1)=1

现在，我们可以为八皇后问题编写程序了！

程序1：

```
0 CLS : 'To input the following Table Characters, use Alt with their
1 'ASCII code; To obtain ASCII code, use ASC.EXE provided by
UCDOS.
2 PRINT "                                                    "
3 FOR I = 1 TO 7
4 PRINT 9 - I; "                                             "
5 PRINT "                                                    "
6 NEXT I
7 PRINT " 1                                                  "
8 PRINT "                                                    "
9 PRINT "     a   b   c   d   e   f   g   h": PRINT
10 DIM T(8), C(8), S(15), M(15): P = 1: T(P) = 1
20 IF C(T(P)) = 1 OR S(8 + P - T(P)) = 1 OR M(P +
T(P) - 1) = 1 THEN 60
30 C(T(P)) = 1: S(8 + P - T(P)) = 1
35 M(P + T(P) - 1) = 1
40 IF P = 8 THEN 80
50 P = P + 1: T(P) = 1: GOTO 20
60 IF T(P) < 8 THEN T(P) = T(P) + 1: GOTO 20
70 P = P - 1: C(T(P)) = 0: S(8 + P - T(P)) = 0
75 M(P + T(P) - 1) = 0: GOTO 60
80 FOR P = 1 TO 8: LOCATE 2 * P, 4 * T(P) + 2
85 PRINT CHR$(232): NEXT P
90 LOCATE 20, 1: PRINT "The solution is :";
100 FOR I = 1 TO 8: FOR J = 1 TO 8
110 IF T(J) = I THEN PRINT " "; CHR$ (96 + I); CHR
$(57 - J);
120 NEXT J: NEXT I
130 END
```

程序1完全按上述思路编写，用传统的带行号格式写成，比较容易读懂，但它只能求出一种解答，要继续求出其它91种解答，必须借助于递归算法。程序2在程序1的基础上稍加改进，采用了递归算法和结构化程序设计方法，能够求出所有的92种解答。相信读者在深入理解程序1的基础上，能够比较顺利地读懂程序2。

程序2：

```
DECLARE SUB Generate (N)
DECLARE SUB Display ()
CLS
DIM T(8), C(8), S(15), M(15)
N = 1
SUM = 0
FOR I = 1 TO 8
 C(I) = 1
NEXT I
FOR I = 1 TO 15
 S(I) = 1
 M(I) = 1
NEXT I
CALL Generate(N)
LOCATE 22, 1
PRINT SPACE$(14); "Total :"; SUM; SPACE$(16)
END

SUB Display
 SHARED T(), SUM
 CLS
 PRINT "                                                    "
 FOR I = 1 TO 7
 PRINT 9 - I; "                                             "
 PRINT "                                                    "
 NEXT I
 PRINT " 1                                                  "
 PRINT "                                                    "
 PRINT "     a   b   c   d   e   f   g   h"
 PRINT
 SUM = SUM + 1
 PRINT "Solution"; SUM; ":";
 FOR I = 1 TO 8: FOR J = 1 TO 8
 IF T(J) = I THEN PRINT " "; CHR$(I + 96); CHR$(57 - J);
 NEXT J: NEXT I
 FOR I = 1 TO 8
  LOCATE 2 * I, 4 * T(I) + 2
  PRINT CHR$(232);
 NEXT I
 LOCATE 22, 1
 PRINT "Press any key to continue,  <Esc>  to EXIT."
 A$ = INPUT$(1): IF A$ = CHR$(27) THEN END
END SUB

SUB Generate (N)
 SHARED T(), C(), S(), M()
 FOR I = 1 TO 8
  IF (C(I) AND S(8 +N -I)AND M(N + I - 1)) THEN
   T(N) = I
   C(I) = 0
   S(8 + N - I) = 0
   M(N + I - 1) = 0
   IF (N <8) THEN CALL Generate (N+1) ELSE CALL Display
   C(I) = 1
   S(8 + N - I) = 1
   M(N + I - 1) = 1
  END IF
 NEXT I
END SUB
```

注：所附程序均在Turbo BASIC、Quick BASIC和MS DOS提供的QBASIC环境下运行通过。程序运行后，屏幕上将以模拟棋盘的形式显示出各种解答方案下皇后所在的位置及其记录方案（程序1只能显示出第一种方案）。