FAST MAZE ROUTER

J. Soukup
Bell-Northern Research
Ottawa, Ontario

Introduction

This paper describes a method of generating a path between two points of a cellular array, with some cells being specified as obstacles. The method combines two techniques: A line search is first directed toward the target. This is a fast and extremely effective method on boards with plenty of space. When the line search hits an obstacle, an expansion technique typical for the Lee-type algorithms is then used to "bubble" around the obstacle. The line search can then continue toward the target again.

The method combines the advantage of both algorithms: It is extremely fast (10-50 times faster than the Lee algorithm on typical two-layer boards), and it guarantees a connection if there is one. It has two minor disadvantages. The routes are suboptimal when the area becomes congested, and both the starting cell and the target cell have to be specified.

The algorithm can be easily coded; our current program for two-layer boards has only 240 lines of Fortran code. However, because of the repetitive nature of the algorithm, a large amount of computer time, and, hence, money can be saved by careful coding.

The Method

For the sake of the simplicity, the algorithm is described for an orthogonal one-layer array. For each cell, two values are maintained:

S signal definition and a trace back pointer, 3 bits required.
C reach flag, 2 bits required.

Two working arrays store coordinates of the cells on the old and new wave front:

RO is the old reach set (2 coordinates for each entry);
RN is the new reach set (2 coordinates for each entry).

The following codes are used for variables S and C:

S = 0 other layer (not used here)

S = 1 left
S = 2 right
S = 3 down
S = 4 up
S = 5 starting point (or subnet)
S = 6 target point (or subnet)
S = 7 other signals or obstructions

C = 0 not reached yet

C = 1 reached through Lee expansion only
C = 2 reached through line search

Before the algorithm starts, C is set to 0 for all cells.

Algorithm

(1) Enter the starting cell as the first entry into stack RN.
Set C = 2 for the starting point.

(2) Move RN into RO; empty RN; reverse order of RO.

(3) One by one, proceeding from the end of the array, take the entries of RO.

For each entry, check values of S and C for its neighbours:

- if C = 2 or S = 7, then do nothing.
- if S = 6, then go to (8).
- if $C \leq 1$ and the neighbour is in the direction toward the target, then go to (5).
- if C = 0, then do:
  - add the neighbour to the end of RN
  - set C = 1 for the neighbour
  - if $S \leq 4$, then set S to the traceback code.

(4) If RN is empty, then EXIT - there is no connection, otherwise go to (2).

(5) Move RN into RO after the last unused entry.

(6) Add the neighbour to the end of RO.
Set C = 2 for the neighbour.
If $S \leq 4$, then set S to the traceback code.

(7) Find the next neighbour continuing in the same direction. Check its values S and C:

- if C = 2 or S = 7, then go to (3).
- if S = 6, then go to (8).
- if not closer to the target, then go to (3).

Go to (6).

(8) A connection has been found, traceback to the starting point, using S.

Comments

Though one starting point and one target point always have to be given, the algorithm can be applied to a general case of connecting a large subnet to another large subnet. The subnets are to be marked by S = 5 or 6 respectively and the algorithm starts with entering whole starting subnet into stack RN.

Sample Problem

Fig. 1 and 2 show a test problem of connecting a pair of points around a simple combination of obstacles. Fig. 1 corresponds to the fast algorithm described above; the router reached the cells represented by a circle through the Lee expansion only. The cells marked by a diamond were on a direct route toward the target.

Fig. 2 corresponds to a pure Lee-expansion. The ratio between the number of cells processed by the router explains why the new router is so fast. It also can be seen that similar route patterns can be expected from both routers.

Acknowledgements

The author is deeply indebted to B.R. Smith and to J. Zelikovitz for their support, for numerous discussions on the routing strategy and for testing the router.

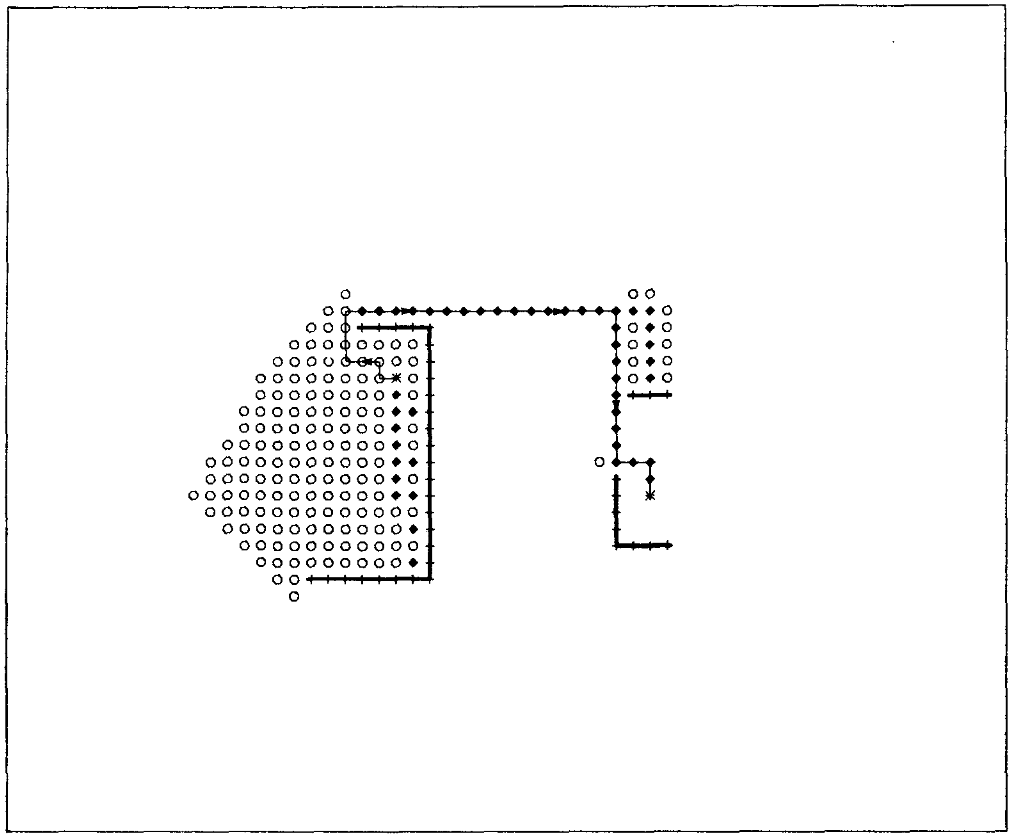

Figure 1 Cells Reached when Using the Fast Router. Diamonds Denote Cells Reached on a Line Toward the Target

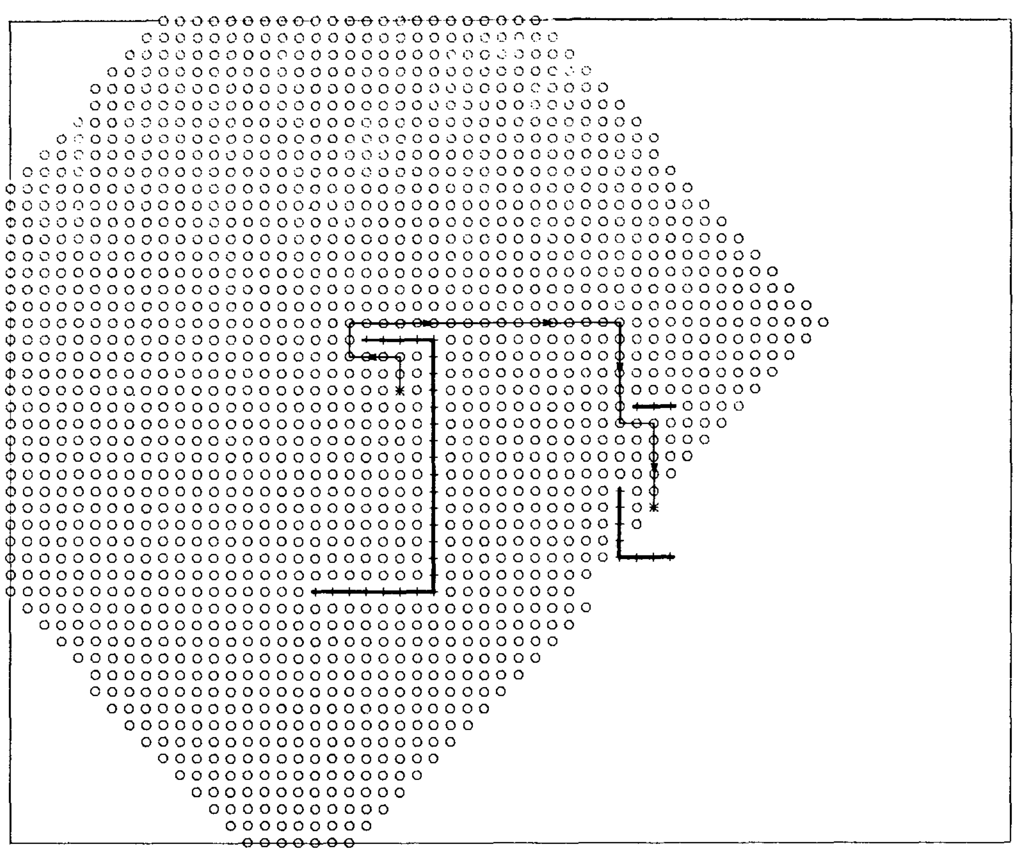

Figure 2 Cells Reached When Using the Pure Lee Expansion