IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

NPTEL

GRID ROUTING

PROF. INDRANIL SENGUPTA
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Introduction

- In the VLSI design cycle, *routing* follows *cell placement*.
- Once routing is completed, precise paths are defined on the layout surface, on which conductors carrying electrical signals are run.
- Routing takes up almost 30% of the design time, and a large percentage of layout area.
  - One main objective is to minimize the area required for routing.

Types of Routing?

- Given a set of blocks placed on a layout surface and defined pin locations:
  - Given a set of obstacles and a set of pins to connect, determine a solution to interconnect the pins on a single layer (**GRID ROUTING**).
  - Determine the approximate regions through which each interconnection net should pass (**GLOBAL ROUTING**).
  - For each routing region, complete the interconnection by assigning horizontal and vertical metal line segments on the layout surface (**DETAILED ROUTING**).

The General Routing Problem

- Given:
  - A set of blocks with pins on the boundaries.
  - A set of signal nets.
  - Locations of the blocks on the layout surface.
- Objective:
  - Find suitable paths on the available layout space, on which wires are run to connect the desired set of pins.
  - Minimize some given objective function, subject to given constraints.

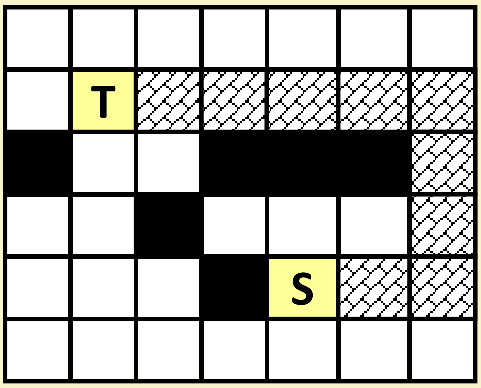
- Types of constraints:
  - Minimum width of routing wires.
  - Minimum separation between adjacent wires.
  - Number of routing layers available.
  - Timing constraints.

GRID ROUTING

Basic Concept

- The layout surface is assumed to be made up of a rectangular array of grid cells.
- Some of the grid cells act as obstacles.
  – Blocks that are placed on the surface.
  – Some nets that are already laid out.
- Objective is to find out a single-layer path (sequence of grid cells) for connecting two points belonging to the same net.

- Two broad classes of grid routing algorithms:
  1. Maze routing algorithms.
  2. Line search algorithms.



IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL

Grid Routing Algorithms

1. Maze running algorithm
   - Lee's algorithm
   - Hadlock's algorithm
2. Line search algorithm
   - Mikami-Tabuchi's algorithm
   - Hightower's algorithm
3. Steiner tree algorithm

Maze Running Algorithms

- The entire routing surface is represented by a 2-D array of grid cells.
  - All pins, wires and edges of bounding boxes that enclose the blocks are aligned with respect to the grid lines.
  - The segments on which wires run are also aligned.
  - The size of grid cells is appropriately defined.
    - Wires belonging to different nets can be routed through adjacent cells without violating the width and spacing rules.
- Maze routers connect a single pair of points at a time.
  - By finding a sequence of adjacent cells from one point to the other.

Lee's Algorithm

- The most common maze routing algorithm.
- Characteristics:
  - If a path exists between a pair of points S and T, it is definitely found.
  - It always finds the shortest path.
  - Uses breadth-first search.
- Time and space complexities are $O(N^2)$ for a grid of dimension $N \times N$.

Phase 1 of Lee's Algorithm

- Wave propagation phase
  - Iterative process.
  - During step i, non-blocking grid cells at Manhattan distance of i from grid cell S are all labeled with i.
  - Labeling continues until the target grid cell T is marked in step L.
    - L is the length of the shortest path.
  - The process fails if:
    - T is not reached and no new grid cells can be labeled during step i.
    - T is not reached and i equals M, some upper bound on the path length.

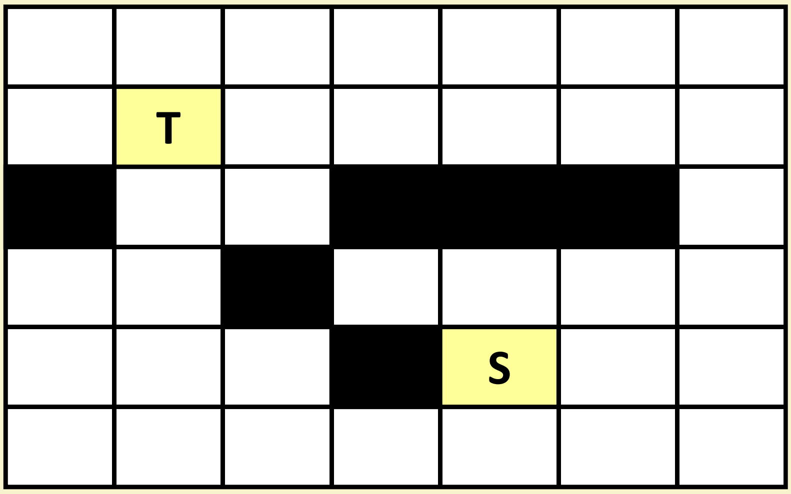
Phase 2 of Lee's Algorithm

- Retrace phase
  - Systematically backtrack from the target cell T back towards the source cell S.
  - If T was reached during step i, then at least one grid cell adjacent to it will be labeled i-1, and so on.
  - By tracing the numbered cells in descending order, we can reach S following the shortest path.
    - There is a choice of cells that can be made in general.
    - In practice, the rule of thumb is not to change the direction of retrace unless one has to do so.
    - Minimizes number of bends.

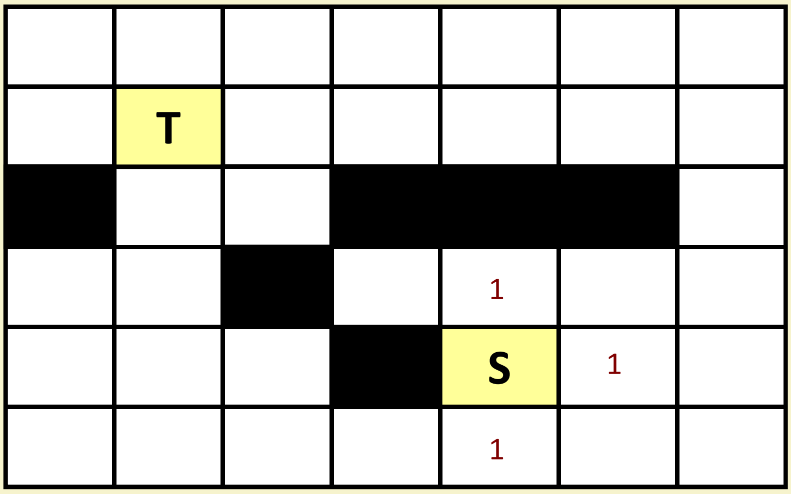
Phase 3 of Lee's Algorithm

- Label clearance
  - All labeled cells except those corresponding to the path just found are cleared.
  - Cells along the path are marked as obstacles.
  - Search complexity is as involved as the wave propagation step itself.

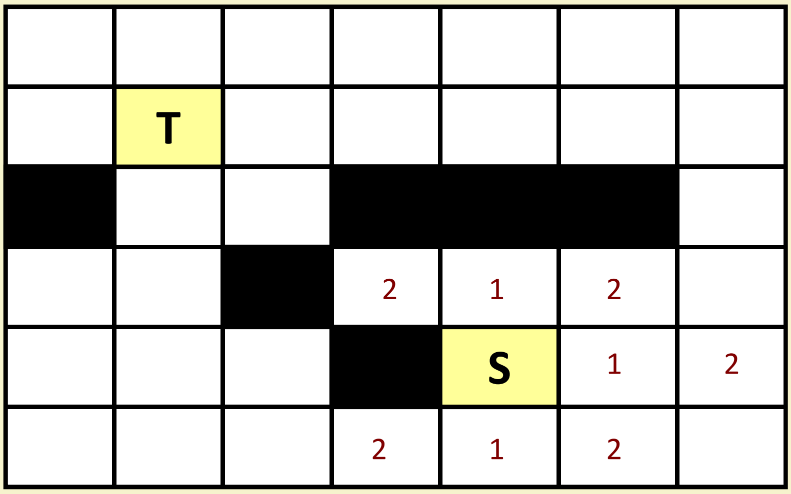
Initial routing problem



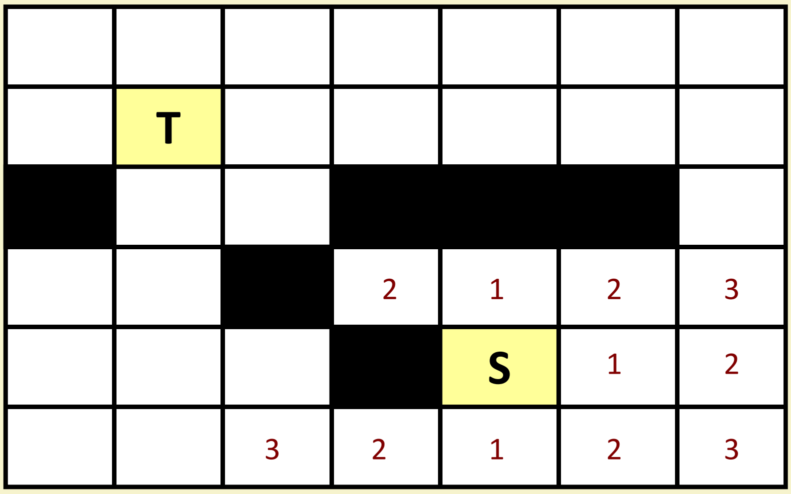
Phase 1 (i = 1)



Phase 1
(i = 2)



Phase 1
(i = 3)



Phase 1
(i = 4)

	T					
						4
			2	1	2	3
		4		S	1	2
	4	3	2	1	2	3

Phase 1
(i = 5)

	T					5
						4
			2	1	2	3
	5	4		S	1	2
5	4	3	2	1	2	3

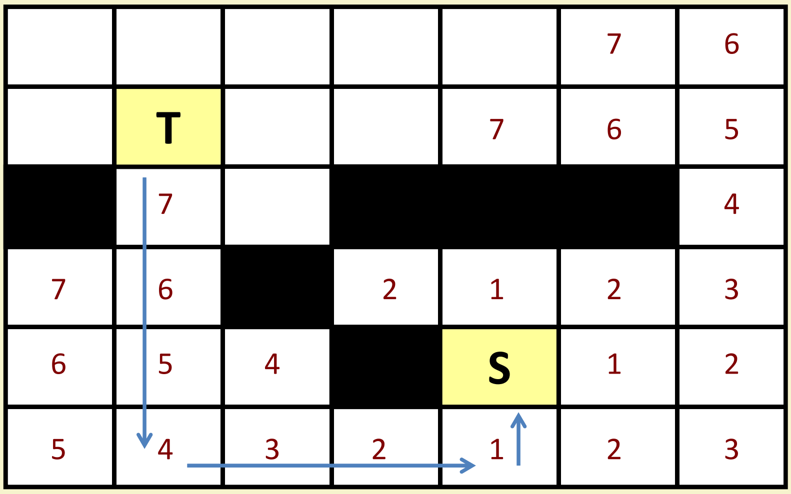
Phase 1
(i = 6)

						6
	T				6	5
						4
	6		2	1	2	3
6	5	4		S	1	2
5	4	3	2	1	2	3

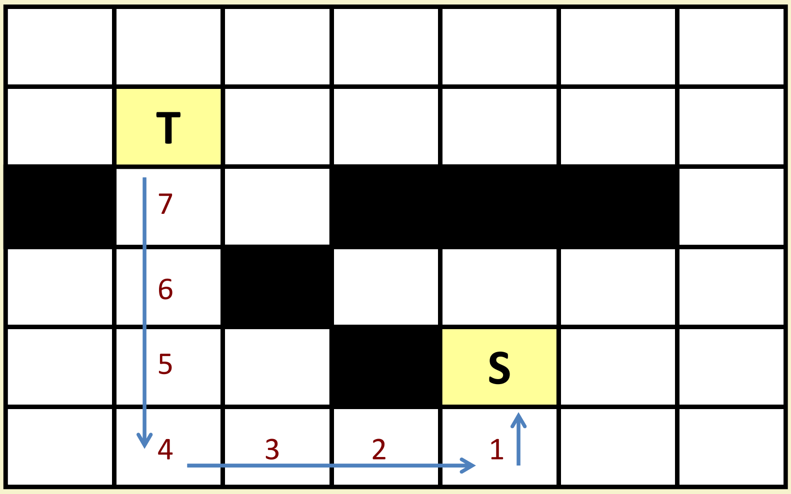
Phase 1
(i = 7)

					7	6
	T			7	6	5
	7					4
7	6		2	1	2	3
6	5	4		S	1	2
5	4	3	2	1	2	3

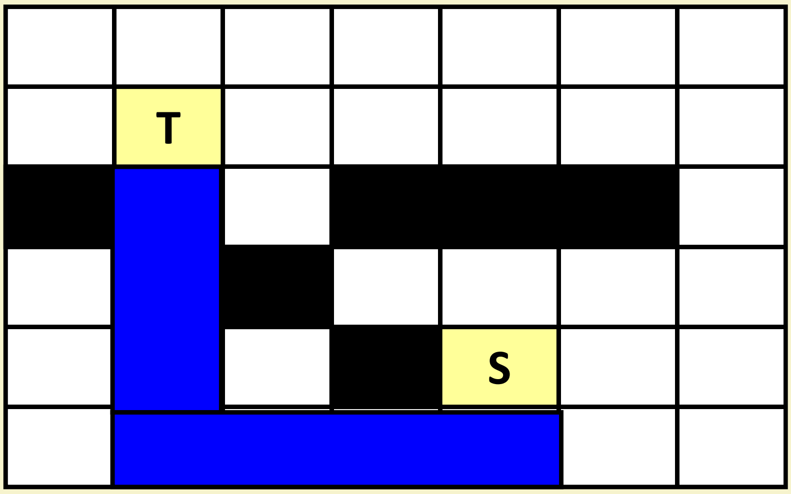
Phase 2 (RETRACE)

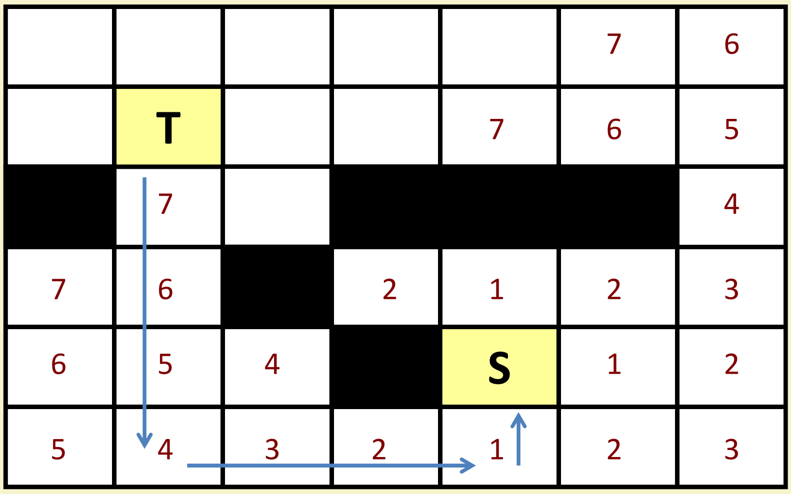


Phase 3 (CLEAR)



Phase 3 (MARK)





- Memory Requirement
  - Each cell needs to store a number between 1 and L, where L is some bound on the maximum path length.
    - For M x N grid, L can be at most M+N-1.
  - One bit combination to denote empty cell.
  - One bit combination to denote obstacles.

$\lceil \log_2(L+2) \rceil$ bits per cell

- Examples:
  1. 2000 x 2000 grid
     - B = $\log_2 4001$ = 12
     - Memory required = 2000 x 2000 x 12 bits = 6 Mbytes
  2. 3000 x 3000 grid
     - B = $\log_2 6001$ = 13
     - Memory required = 3000 x 3000 x 13 bits = 14.6 Mbytes
  3. 4000 x 4000 grid
     - B = $\log_2 8001$ = 13
     - Memory required = 4000 x 4000 x 13 bits = 26 Mbytes

- <u>Improvements</u>:
  - Instead of using the sequence 1,2,3,4,5,..... for numbering the cells, the sequence 1,2,3,1,2,3,... is used.
    - For a cell, labels of predecessors and successors are different. So tracing back is easy.

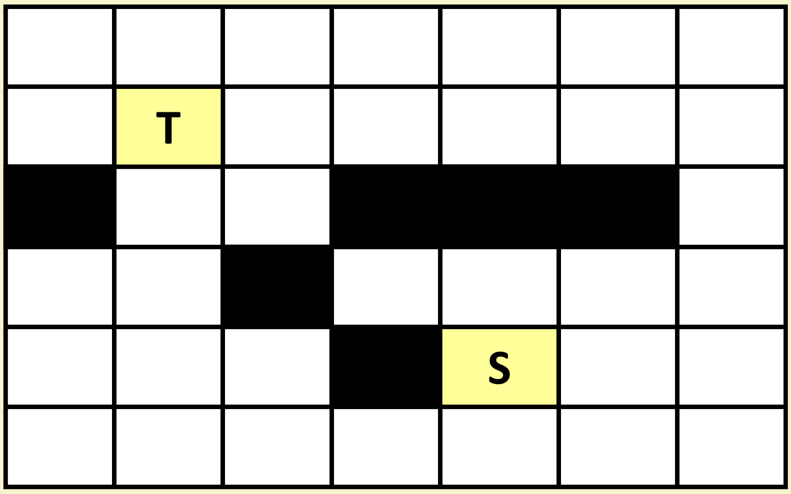
      $\lceil \log_2(3+2) \rceil$ = 3 bits per cell.

      1.5 Mbytes for 2000 x 2000 grid
  - Use the sequence 0,0,1,1,0,0,1,1,.....
    - Predecessors and successors are again different.

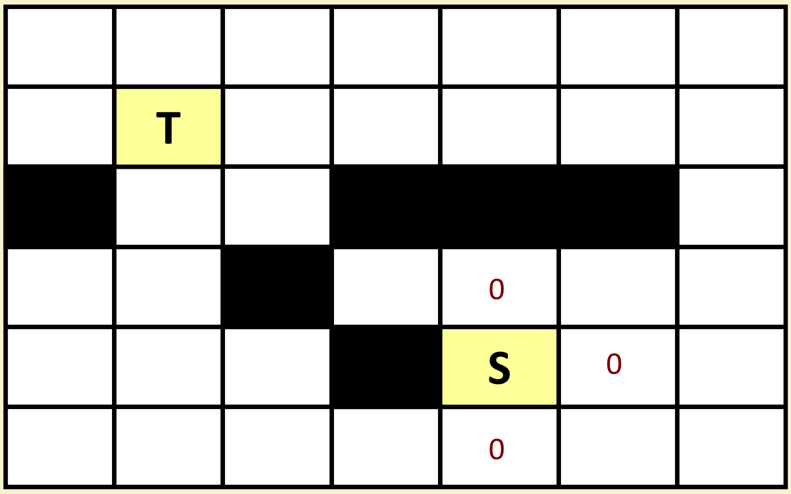
      $\lceil \log_2(2+2) \rceil$ = 2 bits per cell.

      1.0 Mbyte for 2000 x 2000 grid

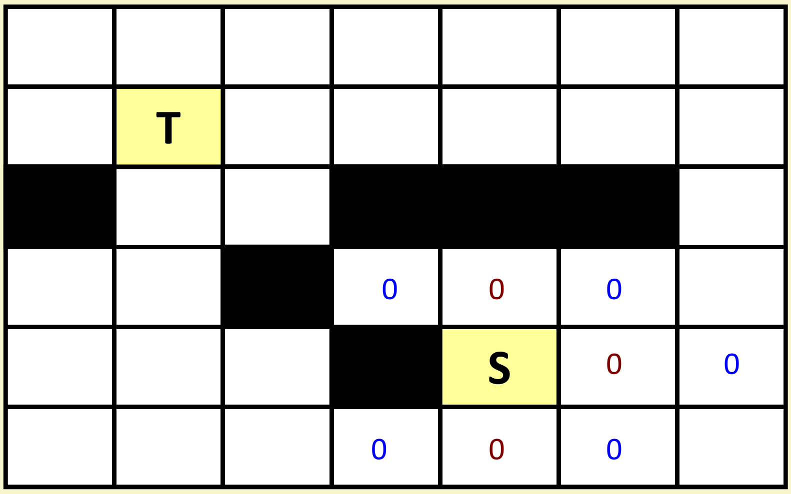
Initial routing problem



Label
0



Label
00



Label
001

	T					
			0	0	0	1
				S	0	0
		1	0	0	0	1

Label
0011

	T					
						1
			0	0	0	1
		1		S	0	0
	1	1	0	0	0	1

Label
0011001

					1	0
	T			1	0	0
	1					1
1	0		0	0	0	1
0	0	1		S	0	0
0	1	1	0	0	0	1

Retrace
0011001

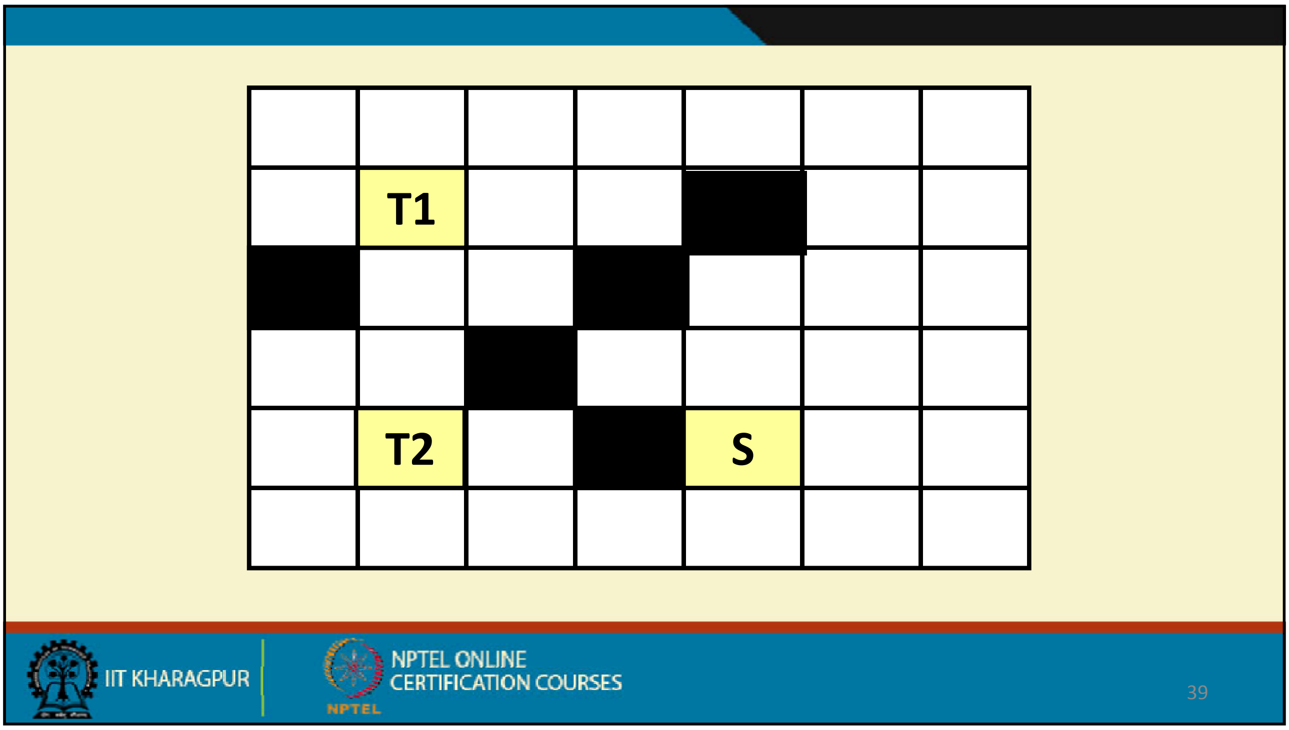
					1	0
	T			1	0	0
	1					1
1	0		0	0	0	1
0	0	1		S	0	0
0	1	1	0	0	0	1

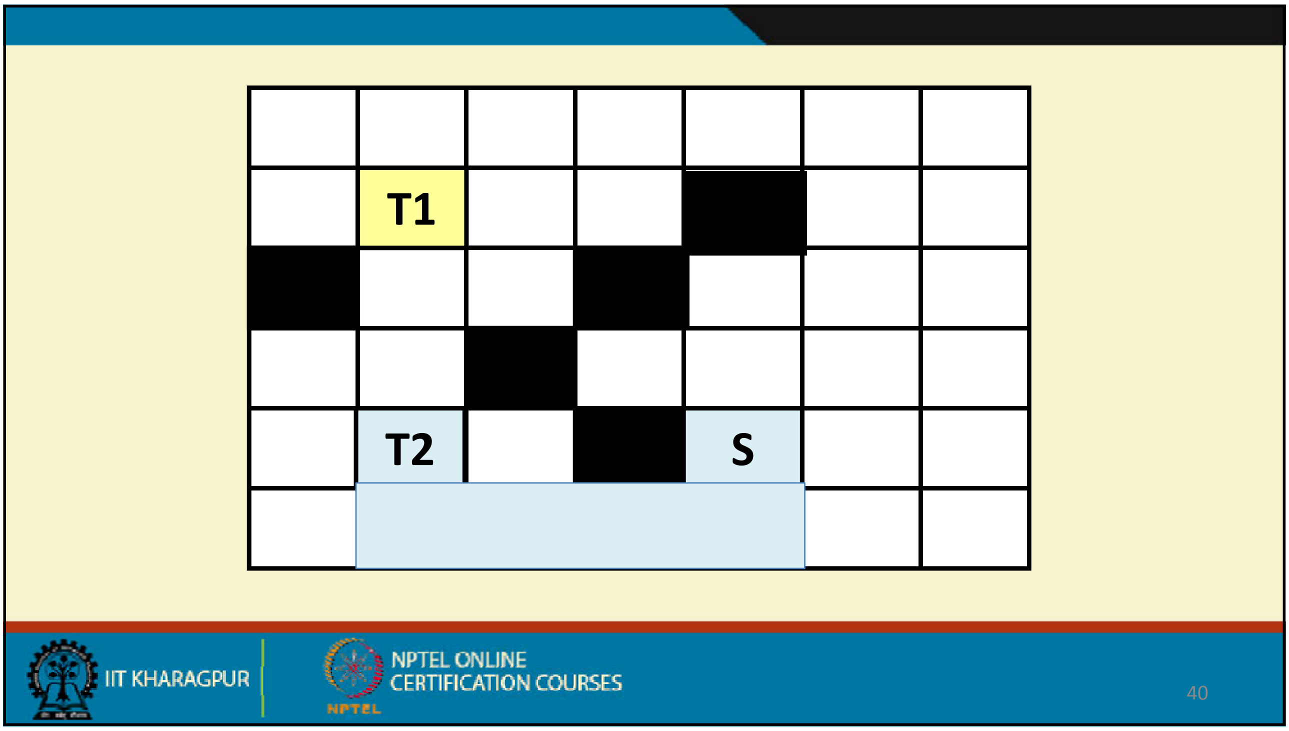
Reducing Running Time

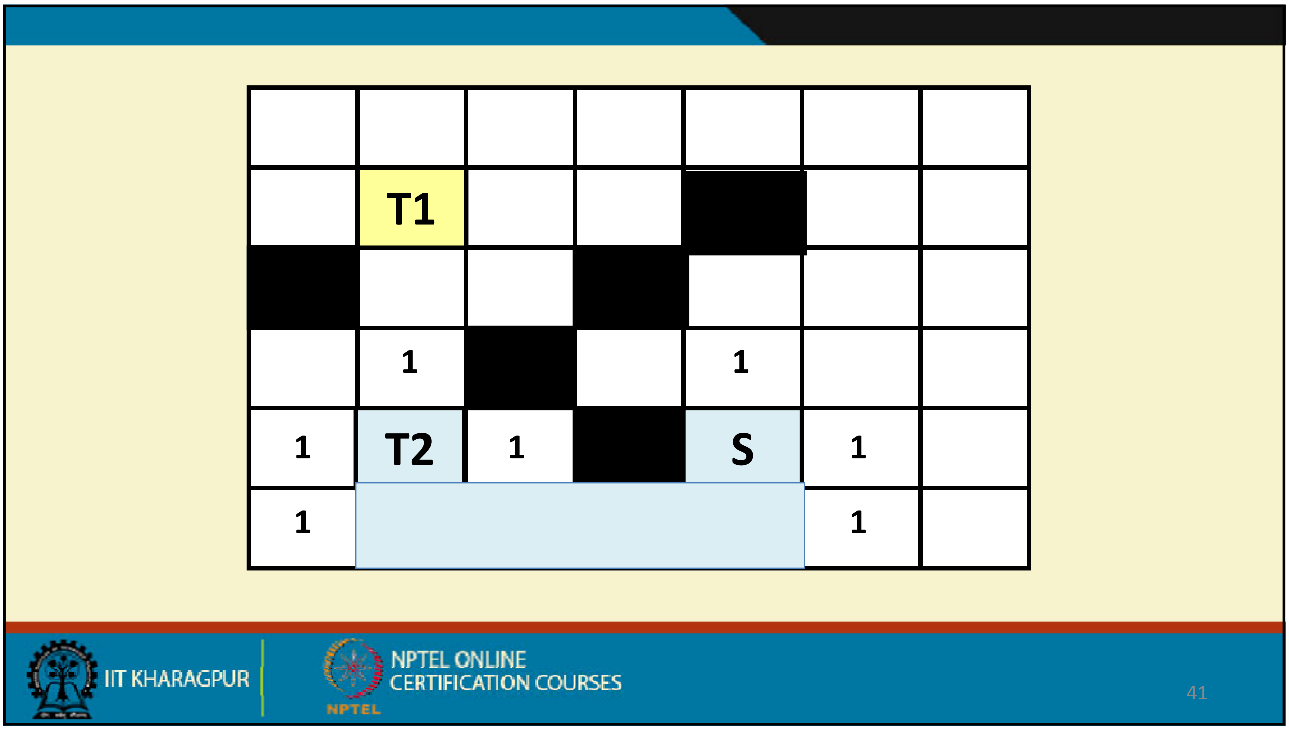
- Starting point selection
  - Choose the starting point as the one that is farthest from the center of the grid.
- Double fan-out
  - Propagate waves from both the source and the target cells.
  - Labeling continues until the wavefronts touch.
- Framing
  - An artificial boundary is considered outside the terminal pairs to be connected.
  - 10-20% larger than the smallest bounding box.

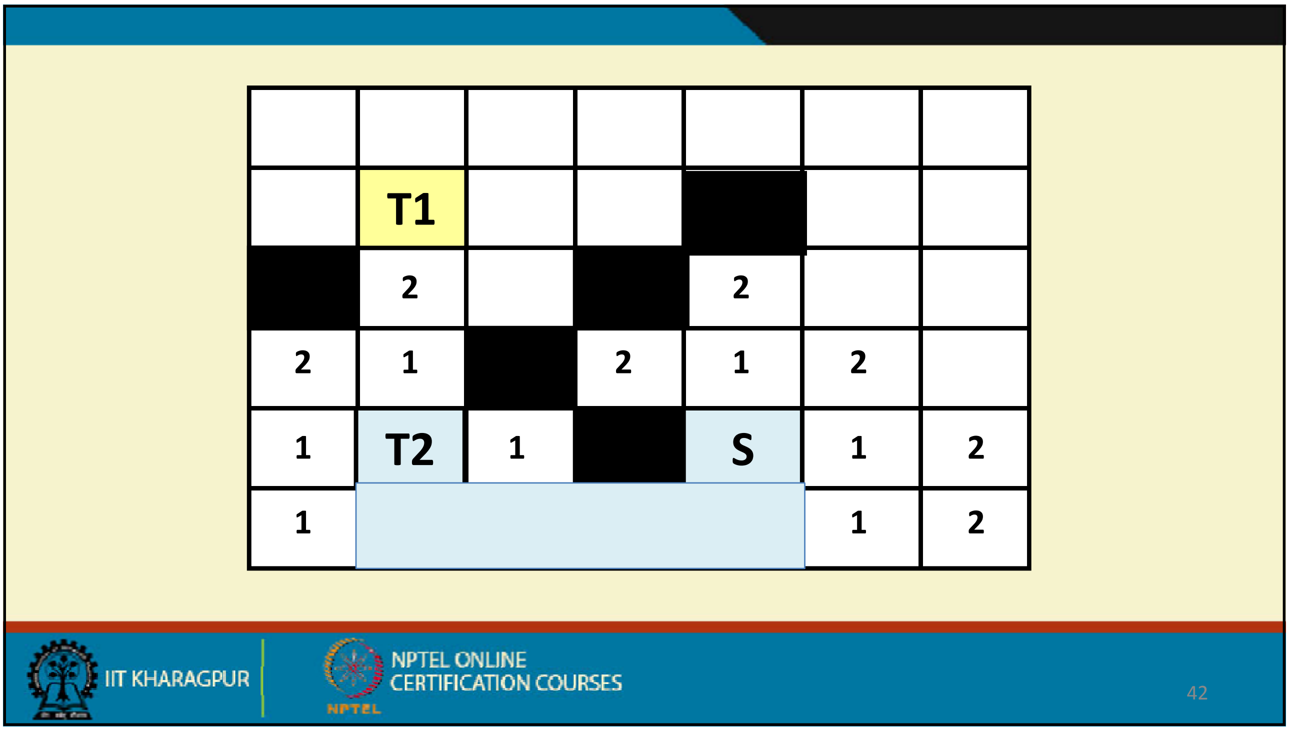
Connecting Multi-point Nets

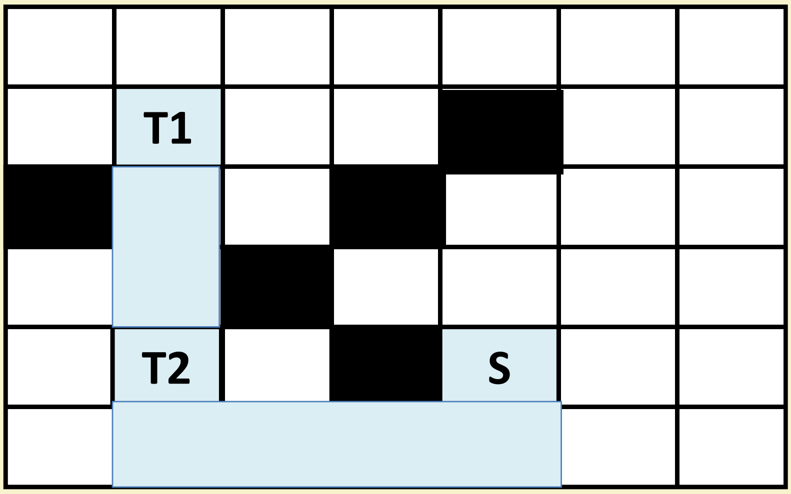
- A multi-pin net consists of three or more terminal points to be connected.
- Extension of Lee's algorithm:
  - One of the terminals of the net is treated as source, and the rest as targets.
  - A wave is propagated from the source until one of the targets is reached.
  - All the cells in the determined path are next labeled as source cells, and the remaining unconnected terminals as targets.
  - Process continues.


T1
T2
S
IIT KHARAGPUR
NPTEL ONLINE
CERTIFICATION COURSES
NPTEL
39


T1
T2
S
IIT KHARAGPUR
NPTEL ONLINE
CERTIFICATION COURSES
NPTEL
40


T1
1
1
1
T2
1
S
1
1
1
IIT KHARAGPUR
NPTEL ONLINE
CERTIFICATION COURSES
NPTEL
41


T1
2
2
2
1
2
1
2
1
T2
1
S
1
2
1
1
2
IIT KHARAGPUR
NPTEL ONLINE
CERTIFICATION COURSES
NPTEL
42



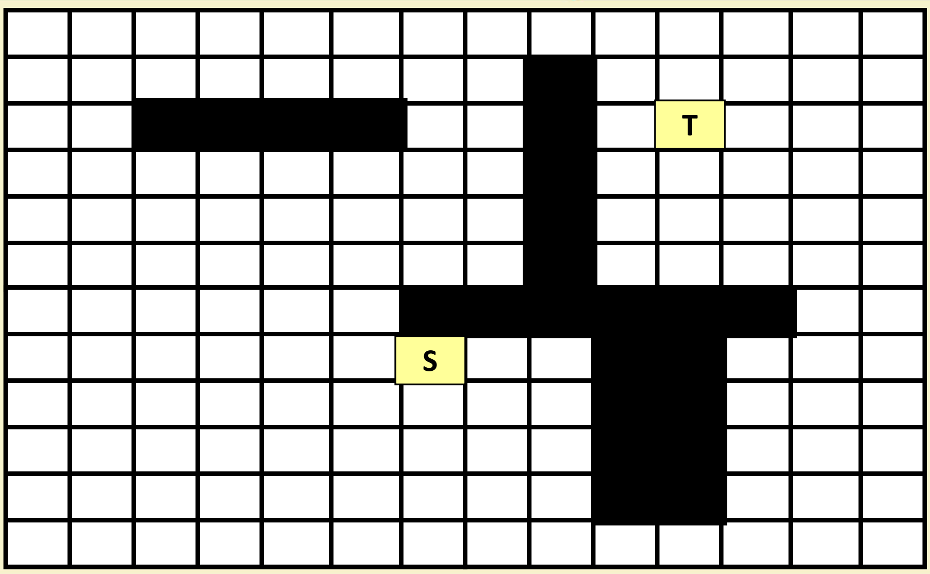
Hadlock's Algorithm

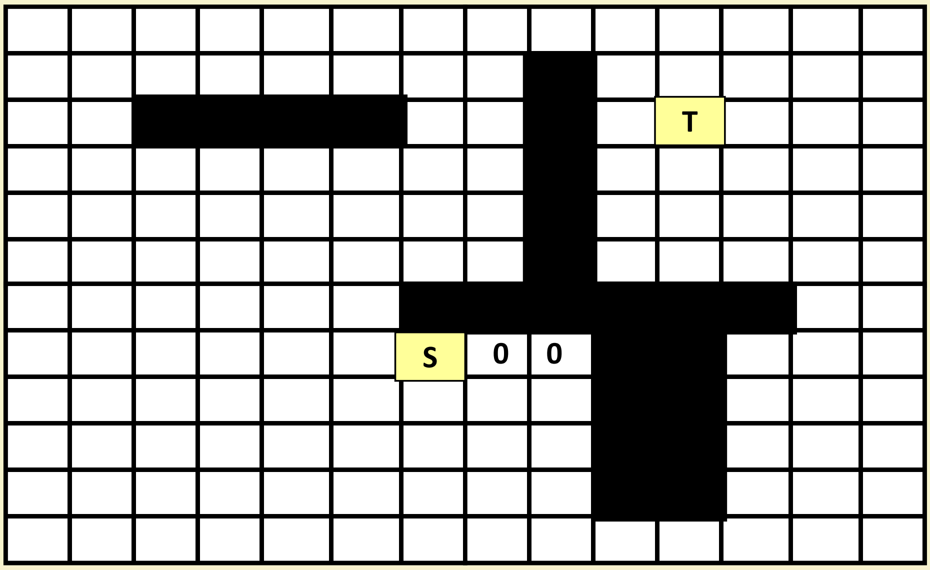
- Uses a new method for cell labeling called detour numbers.
  - A goal directed search method.
  - The detour number d(P) of a path P connecting two cells S and T is defined as the number of grid cells directed away from its target T.
  - The length of the path P is given by

    len(P) = MD (S,T) + 2 d(P)

  where MD (S,T) is the Manhattan distance between S and T.

- The cell filling phase of Lee's algorithm can be modified as follows:
  - Fill a cell with the detour number with respect to a specified target T (not by its distance from source).
  - Cells with smaller detour numbers are expanded with higher priority.
- Path retracing is of course more complex, and requires some degree of searching.



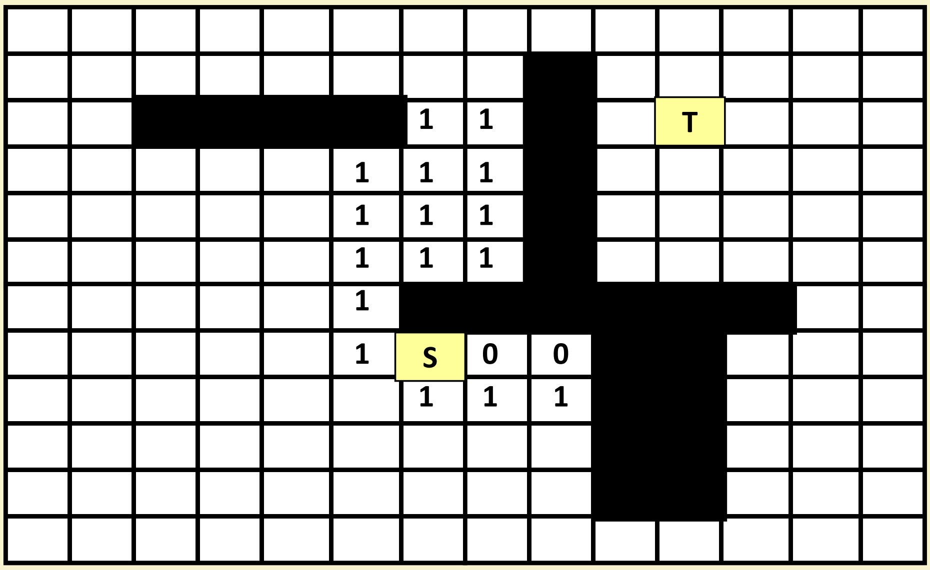


IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

NPTEL

47

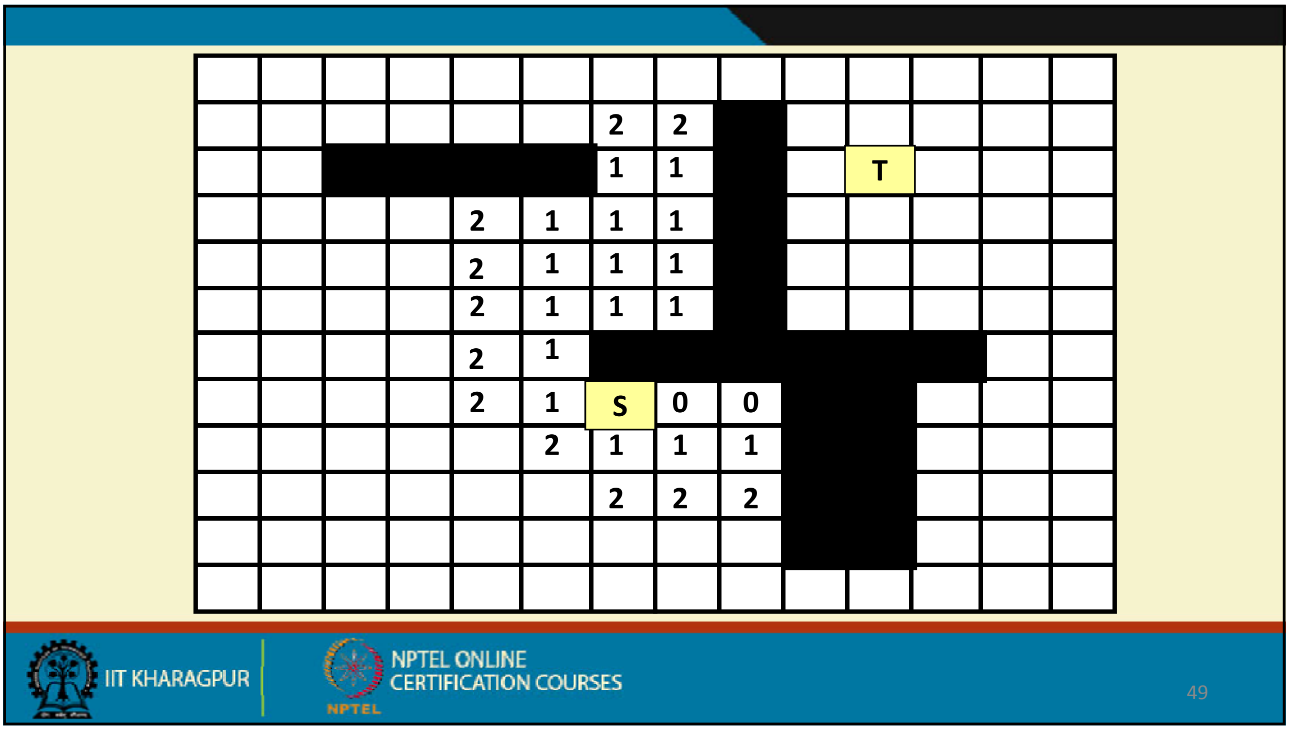


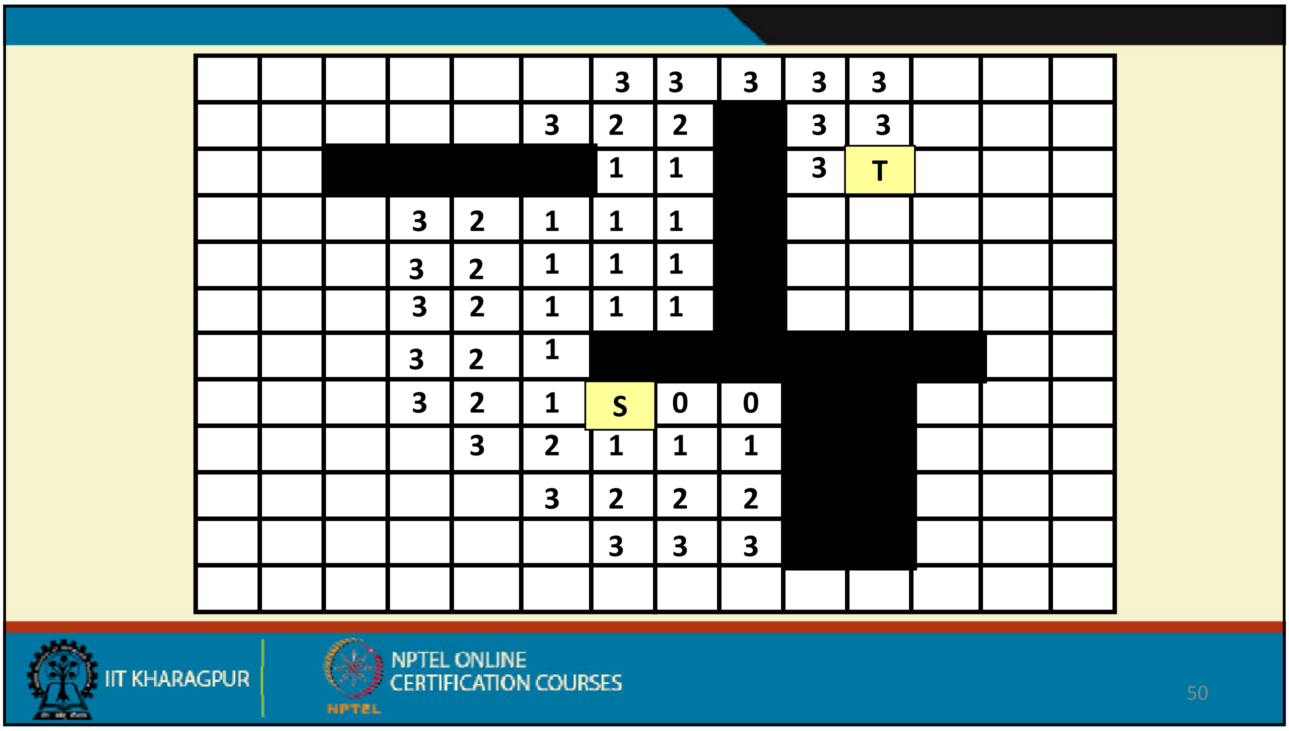
IIT KHARAGPUR

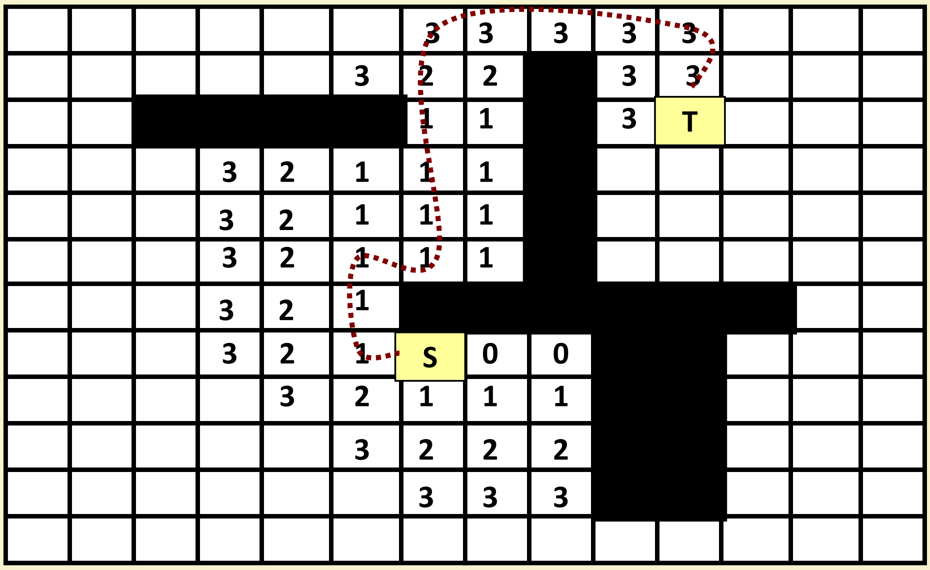
NPTEL ONLINE
CERTIFICATION COURSES

NPTEL

48


IIT KHARAGPUR
NPTEL
NPTEL ONLINE
CERTIFICATION COURSES
49


IIT KHARAGPUR
NPTEL
NPTEL ONLINE
CERTIFICATION COURSES
50



- Advantages:
  - Number of grid cells filled up is considerably less as compared to Lee's algorithm.
  - Running time for an NxN grid ranges from O(N) to $O(N^2)$.
    - Depends on the obstructions.
    - Also locations of S and T.

Line Search Algorithm

- In maze running algorithms, the time and space complexities are too high.
- An alternative approach is called line searching, which overcomes this drawback.
- Basic idea:
  - Assume no obstacles for the time being.
  - A vertical line drawn through S and a horizontal line passing though T will intersect.
    - Manhattan path between S and T.
  - In the presence of obstacles, several such lines need to be drawn.

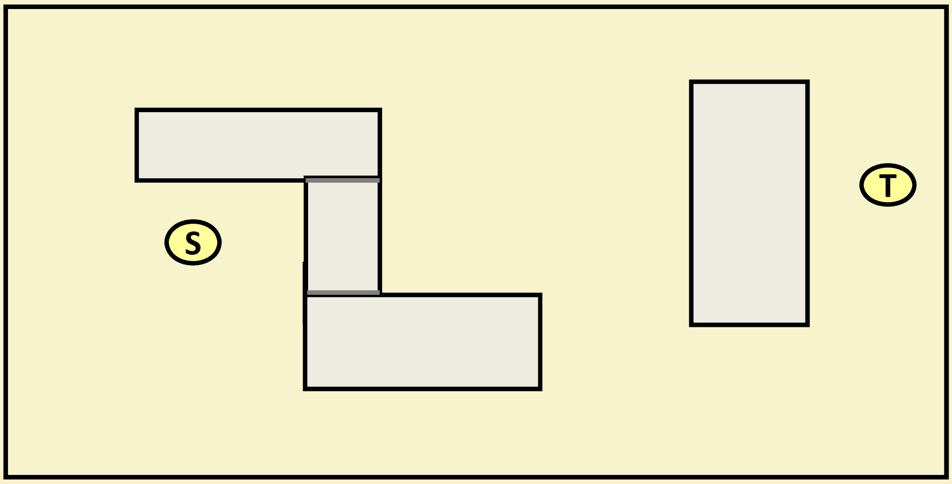
- Line search algorithms do not guarantee finding the optimal path.
  - May need several backtrackings.
  - Running time and memory requirements are significantly less.
  - Routing area and paths are represented by a set of line segments.
    - Not as a matrix as in Lee's or Hadlock's algorithm.

Mikami-Tabuchi's Algorithm

- Let S and T denote a pair of terminals to be connected.
- Step 0:
  - Generate four lines (two horizontal and two vertical) passing through S and T.
  - Extend these lines till they hit obstructions or the boundary of the layout.
  - If a line generated from S intersects a line generated from T, then a connecting path is found.
  - If they do not intersect, they are identified as trial lines of level zero.
    - Stored in temporary storage for further processing.

- Step i of Iteration: (i > 0)
  - Pick up trial lines of level i-1, one at a time.
    - Along the trial line, all its grid points are traced.
    - Starting from these grid points, new trial lines (of level i) are generated perpendicular to the trial line of level i-1.
  - If a trial line of level i intersects a trial line (of any level) from the other terminal point, the connecting path can be found.
    - By backtracing from the intersection point to S and T.
    - Otherwise, all trial lines of level i are added to temporary storage, and the procedure repeated.
- The algorithm guarantees to find a path if it exists.

Illustration

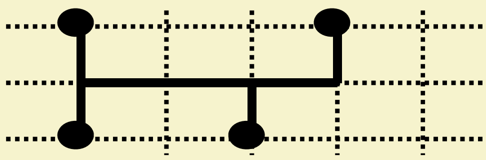


Hightower's Algorithm

- Similar to Mikami-Tabuchi's algorithm.
  - Instead of generating all line segments perpendicular to a trial line, consider only those lines that can be extended beyond the obstacle which blocked the preceding trial line.
- <u>Steps of the algorithm:</u>
  - Pass a horizontal and a vertical line through source and target points (called first-level probes).
  - If the source and the target lines meet, a path is found.
  - Otherwise, pass a perpendicular line to the previous probe whenever it intersects an obstacle.
    - Concept of *escape point* and *escape line*.

Steiner Trees

- A tree interconnecting a set $P=\{P_1,\ldots,P_n\}$ of specified points in the rectilinear plane and some arbitrary points is called a (rectilinear) Steiner tree of P.



- A Steiner tree with minimum total cost is called a Steiner minimal tree (SMT).
  – The general SMT problem is NP-hard.

Steiner Tree Based Algorithms

- Minimum length Steiner trees:
  – Goal is to minimize the sum of the length of the edges of the tree.
  – Both exact and approximate versions exist.
- Weigted Steiner trees:
  – Given a plane partitioned into a collection of weighted regions, an edge with length L in a region with weight W has cost LW.
- Steiner trees with arbitrary orientations:
  – Allows lines in non-rectilinear directions like $+45^o$ and -45^o.