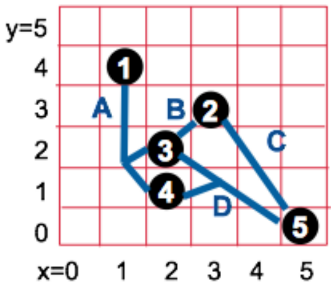
VE527: Assignment #2

Name: Chang Meng
Student ID:118370910019

October 22, 2018

1 (12%) Quadratic Wirelength Calculation

Consider the following simple placement of 5 gates in a small 6×6 grid.



Each gate is drawn as a circle with number (15). Assumer the gate is located at the center of the grid cell, and its (X, Y) coordinates are taken from the column (X) and row (Y) coordinates in the figure. There are 4 nets, labeled A, B, C and D, connected as follows:

- Net A: gates 1, 3, 4
- Net B: gates 2, 3
- Net C: gates 2, 5
- Net D: gates 3, 4, 5

For this placement, what is its total quadratic Wirelength?

Answer:

For Net A, it has 3 points, so the weight = 1/2, and its quadratic wirelength is:

$$\frac{1}{2}[(1-2)^2+(4-2)^2]+\frac{1}{2}[(1-2)^2+(4-1)^2]+\frac{1}{2}[(2-2)^2+(2-1)^2]=8$$

For Net B, it has 2 points, so the weight = 1, and its quadratic wirelength is:

$$[(3-2)^2+(3-2)^2]=2$$

For Net C, it has 2 points, so the weight = 1, and its quadratic wirelength is:

$$[(3-5)^2 + (3-0)^2] = 13$$

For Net D, it has 3 points, so the weight = 2, and its quadratic wirelength is:

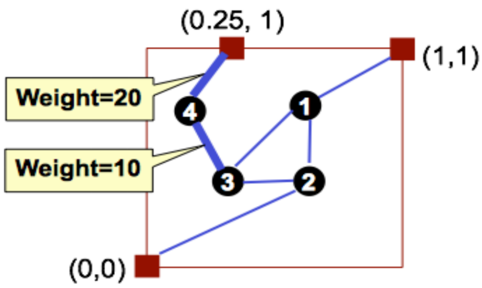
$$\frac{1}{2}[(2-2)^2 + (2-1)^2] + \frac{1}{2}[(2-5)^2 + (2-0)^2] + \frac{1}{2}[(2-5)^2 + (1-0)^2] = 12$$

The total quadratic wirelength is

$$8 + 2 + 13 + 12 = 35$$

2 (24%) Quadratic Placement

Consider the following simple netlist of 4 gates and 3 pads, to be placed in a square that extends from X = 0 to 1, and Y = 0 to 1. Assume that these are the final 2-point nets to be used in a quadratic placement. All nets have weight 1.0, except the two nets from gates 3 to 4 and the net from gate 4 to the pad at (0.25, 1). Their weights are 10 and 20, respectively. To perform a quadratic placement, we solve two systems of linear equations $AX = b_X$ and $AY = b_Y$.



(a) (8%) Build the matrix A using the recipe from the lecture.

(b) (8%) Build the vectors b_X and b_Y using the recipe from the lecture.

(c) (4%) Solve for the (X, Y) coordinates for gates 1, 2, 3, and 4. You can solve this any way you like (i.e., you can use your favorite solver).

(d) (4%) After the initial placement, now suppose we want to do recursive partitioning and we divide the chip in half **vertically**. We want to next formulate a new, smaller QP problem in the region left to the cutline. Which gates should be assigned to this left region? Which gates/pads should be propagated to the vertical cutline as pseudo-pads?

Answer:

(a) The connectivity matrix is:

$$C = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 10 \\ 0 & 0 & 10 & 0 \end{bmatrix}$$

Then we can get:

$$A = \begin{bmatrix} 3 & -1 & -1 & 0 \\ -1 & 3 & -1 & 0 \\ -1 & -1 & 12 & -10 \\ 0 & 0 & -10 & 30 \end{bmatrix}$$

(b) Gate 1 connects to pad (1, 1) with weight 1,

$$b_X[1] = 1 \times 1 = 1, b_Y[1] = 1 \times 1 = 1$$

Gate 2 connects to pad (0, 0) with weight 1,

$$b_X[2] = 1 \times 0 = 0, b_Y[2] = 1 \times 0 = 0$$

Gate 4 connects to pad (0.25, 1) with weight 20,

$$b_X[4] = 20 \times 0.25 = 5, b_Y[4] = 20 \times 1 = 20$$

So, we can get:

$$b_X = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 5 \end{bmatrix}, b_Y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 20 \end{bmatrix}$$

(c) The solution of the equations are:

$$X = \begin{bmatrix} \frac{95}{184} & \frac{49}{184} & \frac{13}{46} & \frac{6}{23} \end{bmatrix}, Y = \begin{bmatrix} \frac{155}{184} & \frac{109}{184} & \frac{43}{46} & \frac{45}{46} \end{bmatrix}$$

So the coordinates are: $(X_1, Y_1) = (0.52, 0.84)$, $(X_2, Y_2) = (0.27, 0.59)$, $(X_3, Y_3) = (0.28, 0.93)$, $(X_4, Y_4) = (0.26, 0.98)$

(d) We sorted the gates by their X coordinates and get a sequence of X_4, X_2, X_3, X_1. So gate 2 and gate 4 will be assigned to the left region. Gate 1 and gate 3 will be propagated to the vertical cutline as pseudo-pads because they are both connected to gate 2 in the left region.

3 (6%) Maze Routing: Basics

Which of the following statements about maze routing are correct?

a) Maze routing can only handle a maximum of two routing layers.

b) Maze routing routes a set of nets simultaneously.

c) To efficiently find a minimum-cost item from the wavefront, we can use a data structure such as a min heap.

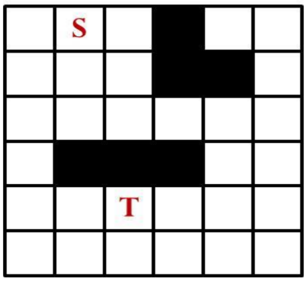
d) If we add to the cost function a predictor that is a lower bound on the actual extra pathcost to the target, then maze routing will still find the minimum cost path. This kind of search is called A* search.

Answer:

c), d) are correct.

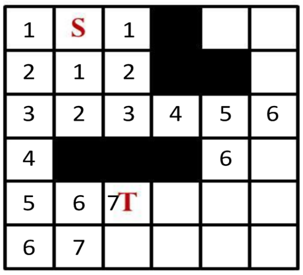
4 (12%) Basic 2-point net routing in 1 layer

Consider the simple 6×6 routing grid shown below. We label a source S cell, a target T cell, and several obstacle cells in black. All white cells have unit cost. Use the simple unit-cost maze routing expansion method, and perform wavefront expansion by hand on the grid, to find a route from S to T. Mark the route you obtain. What is the pathcost of this route?

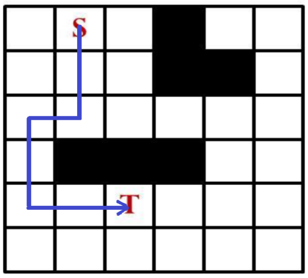


Answer:

The expansion process is like this:



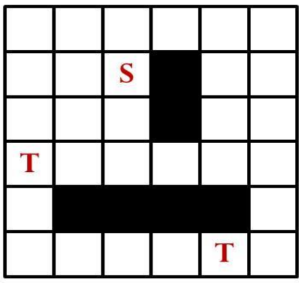
The route is shown in the graph:



The pathcost of this route is 7.

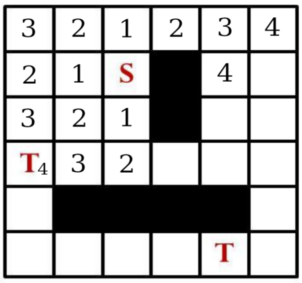
5 (20%) Multi-point net routing in 1 layer

Consider the simple 6×6 grid shown below. We label one source cell S, two target cells T, and several obstacle cells in black. All white cells have unit cost. Use the simple unit-cost maze routing expansion method and the techinique for completing mutil-point nets, and perform wavefront expansion by hand on the grid, to find a route for this 3-point net. Mark the route you obtain. What is the total pathcost of this route?

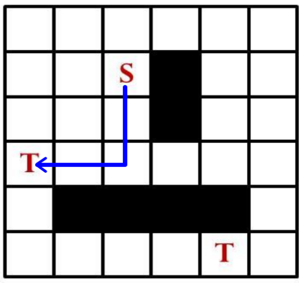


Answer:

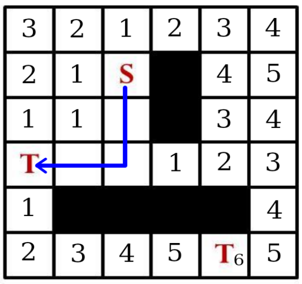
We expand from S and find the T in the left first, as is shown in the following figure:



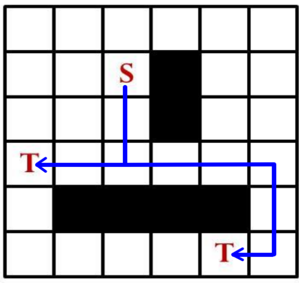
Then we choose one of the shortest paths from S to T in the left like this:



We set the cost on the path above as 0, and expand from the path again:

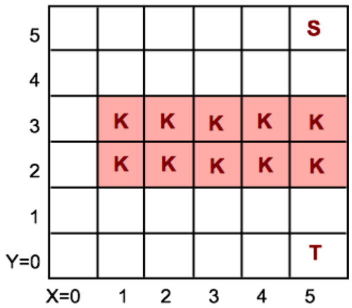


Finally, we obtain a route with total pathcost = 10



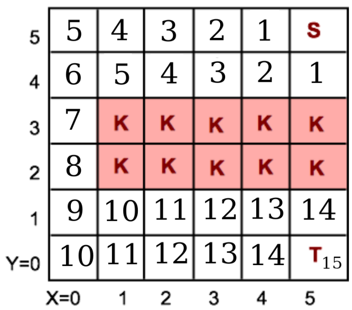
6 (10%) Routing with non-unit costs in 1 layer

Consider the simple 6×6 routing grid shown below. We label one source S, one target T, and several non-unit-cost cells in pink with **integer** cost K. All white cells have unit cost. Assume your router uses the Cheapest-cell-first maze routing expansion algorithm. Suppose that your router finally returns a path that **avoids** all non-unit-cost cells. In this case, K must be large enough so that the path through the unit-cost cells will be cheaper than the path through the unit-cost cells will be cheaper than the path through the non-unit-cost cells. Then, what is minimum value of K?

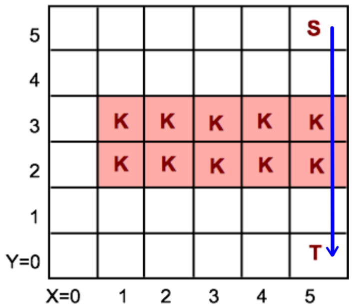


Answer:

If we treat non-unit-cost grids as obstacle, it will expand as is shown below, we can get a path avoiding all non-unit-cost cells with a cost of 15.



The shortest path with non-unit-cost is like the following figure, and its cost is $2K + 3$.



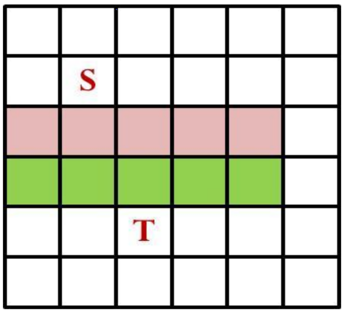
To avoid non-unit-cost cells,

$$2K + 3 > 15, K > 6$$

So the minimum value of K is 7.

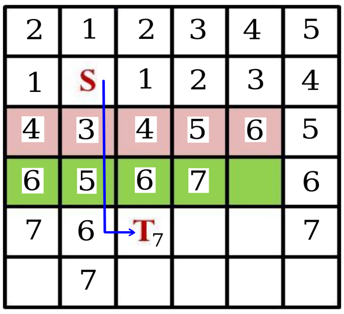
7 (16%) Cheapest-cell-first expansion

Consider the simple 6×6 routing grid shown below. We label one source S and one target T. The white cells have cost 1, the green cells have cost 2, and the pink cells have cost 3. Apply the Cheapest-cell-first maze routing expansion algorithm to find a route from S to T. Mark the route you have found. What is its pathcost?

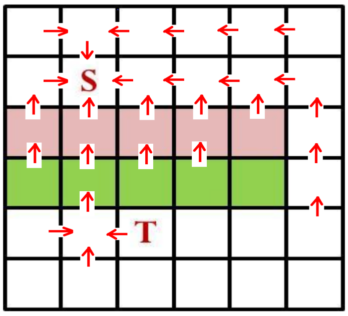


Answer:

The expansion process and the founded route is shown below:



Notably, to reconstruct the final path, I record the parents for each grids like this graph:



The minimum pathcost is 7.