Lab3 - Routing with solver

Deadline: 6/19/2023 23:59

Problem description

Implement a grid-based 2-pin net router based on the routing part mentioned in the MCell paper. The goal is to complete all routing under the specified routing boundary. You need to output your clauses based on the parts mentioned in the paper to MaxSAT / glsol to complete the routing.

Input/Output Format

Input

max_layer # (the maximum layer you can use, 0-indexed. That is, you can use layer 0 to layer #-1)
boundary # # # # (routing boundary: min x, min y, max x, max y coordinate)
min_pitch_size # (the minimum pitch size for the gridline)
via_cost # (one via equal to how much wirelength. If a via pass 3 layer, it would count twice.)
nets #
(The following # lines are the coordinates of the pins that need to be routed)
<netname source_x source_y source_layer target_x target_y target_layer>

All inputs can be stored as 32-bit integers, and the boundary and pitch size of each layer are the same.

Output

x_coors #
(The x-coordinates of the grid lines you draw, followed by # grids, which will be the actual coordinates of your grid lines.)
....#
y_coors #
(The y-coordinates of the grid lines you draw, followed by # grids, which will be the actual coordinates of your grid lines.)
....#
<netname1>
<x y layer> (the first grid net1 passed, that is your source)
<x y layer> (the second grid net1 passed)
...

<x y layer> (the last grid net1 passed, that is your target)
<netname2>
<x y layer> (the first grid net2 passed)
<x y layer> (the second grid net2 passed)
...

Any nodes that are not positioned on the grid nodes will be regarded as failures. When constructing the mesh structure, the pin coordinates can be used as the reference line.
The provided output file represents the routing result. Points will not be awarded if the net exhibits spill-over, short circuits, cycles, or crossings. It is important to note that a net positioned with its center on the boundary is not classified as a spill-over.

Appendix

The solver that can be used for this assignment is Open-WBO MaxSAT, which can be downloaded from here:
https://github.com/sat-group/open-wbo
Although using a SAT solver is recommended for this assignment, if you want to use ILP to complete this assignment, you can use GLPK (GNU Linear Programming Kit). The installation and usage instructions are as follows:
https://en.wikibooks.org/wiki/GLPK/Using_GLPSOL
After choosing the solver to use, you can set the command you want to execute through the Router::getSysCommand() part of the return.

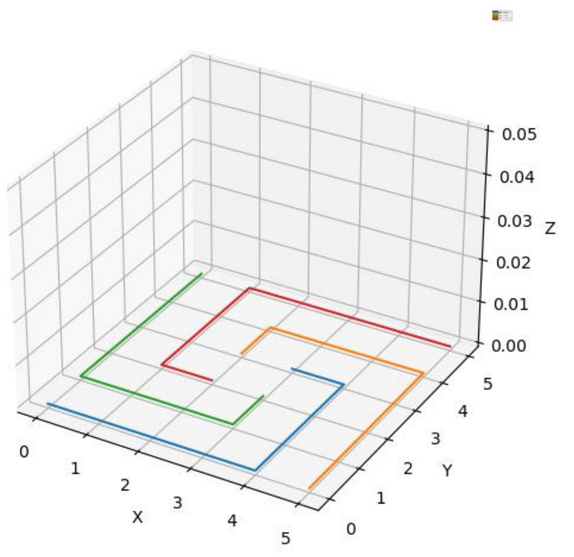
Verfier

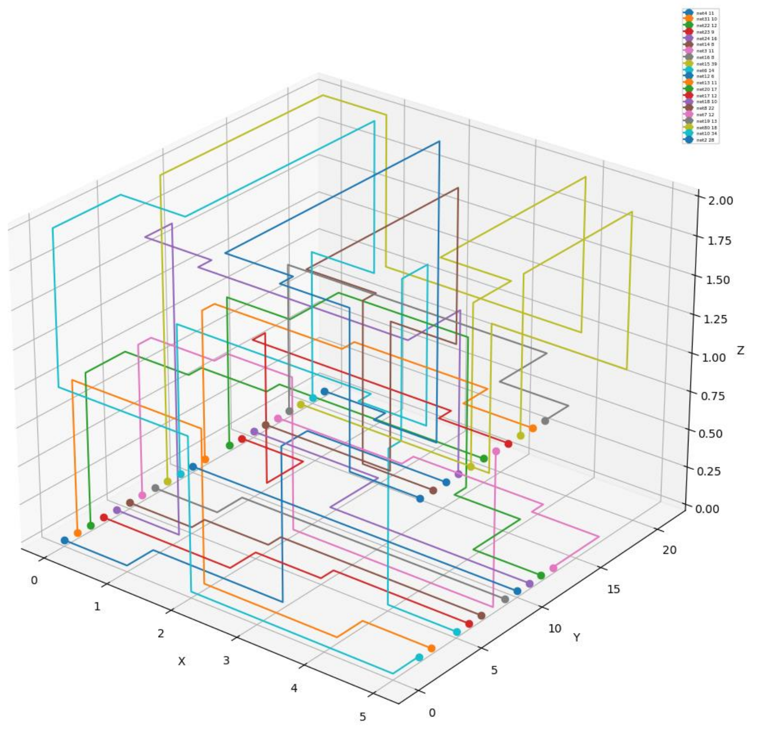
We provide a verifier to check the correctness of your program.
./verifier [output.txt] [input.txt]
If the terminal shows " Your routing result is correct", it means your result is correct. The information also includes your track count and total cost (including via cost), which are the basis of our grading.
If there is any issues, please contact TA.

Plotter

python3 plotter.py [output.txt]
We provide a Python plotter that can help you debug your codes with the visualized routing result. There are two examples in different angles. You can check the detail with plt.show(). The second one has some detours that can be refined. The hidden

case would not be harder than the second one (about 20 net).



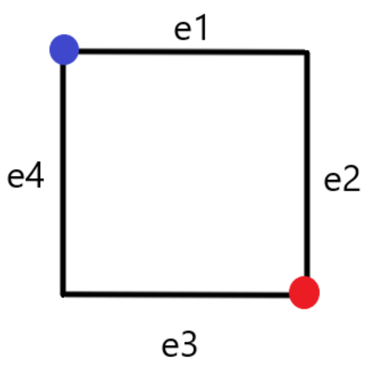


Hint

Generally, SAT solvers require describing input files in CNF (Conjunctive Normal Form), which can be simply regarded as multiple Boolean variables or consisting of clauses with **OR** operators and finally combined with **AND** operators.
For example: (x1 OR NOT x2) AND (x2 OR x3 OR x4)

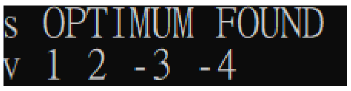
For connectivity between nets, the main consideration is that each pin must be connected by one edge, and each non-pin grid node must use none or two edges.

Between different nets, a grid node cannot be used by two different nets, except for storing the grid node with Boolean values to indicate whether the node is used or not. Alternatively, you can focus on those edges which cannot be used at the same time. Then, apply XOR operation on them.
If the above two conditions are satisfied, the found routing result should be legal.
There is an example for one net on a 2 * 2 mesh structure. And there are two pins:



For blue pin, it should select one edge at least:
(e1 or e4)
And not more than one edge
(-e1 or -e4)
For red pin,
(e2 or e3) or (-e2 or -e3)
You can choose both e1 and e2 at the same time, or choose not to choose either.
(e1 or -e2) and (-e1 or e2)
And,
(e3 or -e4) and (-e3 or e4)
How about 3, 4, 5 or even 6 edge?

Putting them into the solver, we can get the result:



```
s OPTIMUM FOUND
v 1 2 -3 -4
```

That is, it identifies the path which is e1 and e2.

Executing Procedure

1. We'll use "make" command to build your code,
please make sure you're your Makefile can generate an executable named Lab3
2. ./Lab3 [input.txt] [output.txt]

3. Search for [output.txt], found→ break → 0 point

4. ./ verifier [output.txt] [input.txt]

5. If fail → break→ 0 point

Submission

Please submit the following materials in a .zip file to E3 by the deadline, specifying your student ID in the subject field (e.g., StudentID.zip):

1. Source codes (.cpp, .h …)
2. Makefile

Grade

A. The execution time of each case is 60 mins, and exceeding it will be considered a routing failure.

B. No errors such as spill-over, short, cycle, and crossing are allowed for any case. Otherwise, it is also considered a failure.

C. Please use the two solvers (**open-wbo SAT or glsol**) and do not modify the **main.cpp** file. However, if you have any issues or other ideas, please feel free to raise them.

D. There are 4 given case and 2 hidden case. If you can't finish the bonus case, 96 will be your highest score

1. (10 pts + 10 pts) If your routing results for case1 and case2 are correct, you can obtain full grades for each. The two cases are simple, you can draw the result for debugging.

2. (20 pts for each test case (two public + two hidden)) If your routing results for case3, case4, hidden_case1, and hidden_case2 are correct, you can obtain 70% of the score for each case. Additionally, 10% of the score is based on the cost ranking (wirelength + via num * via cost), calculated as follows: (highest - yours) / (highest - lowest). If you can finish the case in 30 minutes, you can obtain 10% of the score. If you can finish the case in 15 minutes, you can obtain an additional 10% of the score.

3. (Bonus 10 pts) The case5 is for the bonus, you must use some acceleration clauses to finish the case. If the result is correct you can obtain 70% of the score. If you can finish the case in 30 minutes, you can obtain the remaining score.

E. Late submission: Score * 0.95 before 6/26, Score * 0.8 6/27 ~ 7/3. After 7/4, submission is not allowed.

F. Wrong submission format: 10 points punishment

G. Any work by fraud will absolutely get 0 point