Exploring A* Search For Single and Multi Layer Routing

Mohamed Shawky
Computer Engineering
Cairo University
Email: mohamed.sabae99@eng-st.cu.edu.eg

Mahmoud Adas
Computer Engineering
Cairo University
Email: mahmoud.ibrahim97@eng-st.cu.edu.eg

Remonda Talaat
Computer Engineering
Cairo University
Email: Remonda.Bastawres99@eng-st.cu.edu.eg

Evram Youssef
Computer Engineering
Cairo University
Email: evram.narouz00@eng-st.cu.edu.eg

***Abstract*—**
***Index Terms*—**

A. Motivation

Metal routing is a critical step in systems integration process, where each source is connected to its fan-outs using non-crossing metal. Routing in a modern chip, with millions or even billions of transistors, can be very complicated and cumbersome, so the manufacture process has moved to automatic routing.

Automatic routing is a very vast field, with several techniques being developed through time. Automatic routing involves lots of problems like finding shortest non-blocked path and VIAs in multi-layer routing.

Automatic routing techniques are always a trade-off between optimality and speed. Some techniques target execution time by reaching a sub-optimal solution. Others target optimal solution with very high execution time.

In this work, we seek a good balance between performance and optimality of solution by introducing the usage of A* search for automatic routing with a modified cost function.

B. Formulation

Routing is formulated as a grid search problem 1, where an occupancy grid map *(OGM)* is provided with some source and destination cells to be joined. The *OGM* cells can have the value of 1 for occupied cell or 0 for empty cell. The *OGM* can be of multiple levels, in case of multi-layer routing, where the cell can have a third value to indicate that the cell can be used as a VIA. Only 4-neighbor cells are considered, so the path can move up, down, left or right. The path can also move from one level to another through VIA cells.

Given this problem formulation, several grid search and shortest path algorithms can be used to find the optimal path between each source and its given destinations. These algorithms can vary based on optimality and performance.

The two main metrics considered in this work are path length and execution time. Total length of metal, covering the grid, is used as a measure of solution optimality. It's defined as the number of grid cells, on which the metal is placed for routing.

Execution time is used as a measure of algorithm performance. It's simply the total elapsed time by the algorithm to find all required paths.

C. Baseline

To measure the validity and performance of our proposed method, three main baselines are used. These baselines are well-established algorithms that are currently used in industry.

1) Maze Routing (Lee's Algorithm): The first baseline algorithm, to be considered, is *maze routing*. Maze routing is one of the most well-established and widely-used algorithms in metal routing. It consists of a wide range of algorithms and techniques. *Lee's algorithm* is the basic maze routing solution and it's considered as our first baseline for comparison. This algorithm is basically a grid search algorithm for routing.

2) Mikami-Tabuchi's Algorithm: The second considered baseline algorithm is *Mikami-Tabuchi's* algorithm. Unlike *Lee's algorithm*, this algorithm is a line search technique that adopts line-search operations. It can find a path between source and destination cells, but it doesn't guarantee the shortest possible path.

3) Steiner Tree Algorithm: Steiner tree spans though the given subset of vertices in a given graph. Steiner tree is suitable for any situation, where the task is minimize cost of connection among some important locations, like VLSI Design, Computer Networks, etc. So, Steiner tree is considered as our final baseline algorithm.

D. Modified A Search*

The main contribution of this work is the usage of *A* search* as a new routing technique. The main idea of *A* search*, as shown in fig.2, is the usage of some heuristic function to assign a cost for each node. To define a path from a source node to a destination node, the nodes that minimizes the estimated path cost are chosen at each step. The main objective is to minimize the following function:

$$f(n) = g(n) + h(n) \tag{1}$$

Where *f(n)* is the total cost of the path from a specific node, *g(n)* is the cost of the edge between current node and next chosen node and *h(n)* is the estimated cost of the next chosen node based on the used heuristic function.

Thus, the required information to solve an *A* search* problem is the cost of edge between each two nodes and the heuristic function to get an estimated cost for each node.

The *OGM* representation of the routing problem can be formulated to become an *A* search* problem. The edge cost *g(n)* is always 1, as the path can only move from one cell to an adjacent cell. The heuristic function, proposed in this work, is *Euclidean distance* between next chosen cell and destination cell, as shown in fig.3. So, the cost of taking any of the adjacent cells is *Euclidean distance* + 1, which can be simplified to *Euclidean distance* between an adjacent cell and destination cell.

The proposed algorithm can be summarized as follows:

IV. Experimental Results

V. Conclusion

References

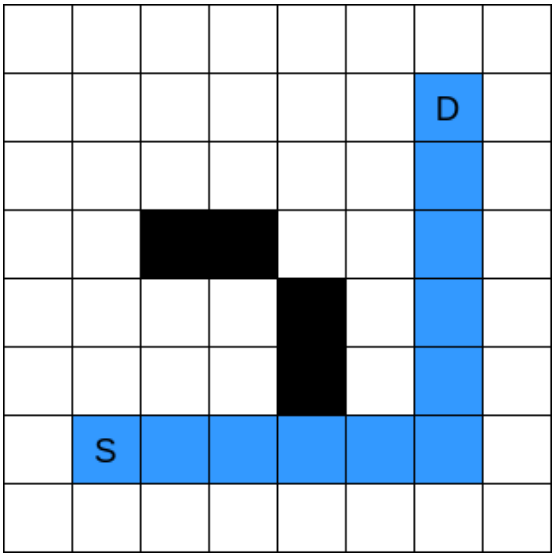

Fig. 1. Example of occupancy grid map (OGM), where S is the source, D is the destination, white cells aren't occupied, black cells are occupied and blue cells represent the path.

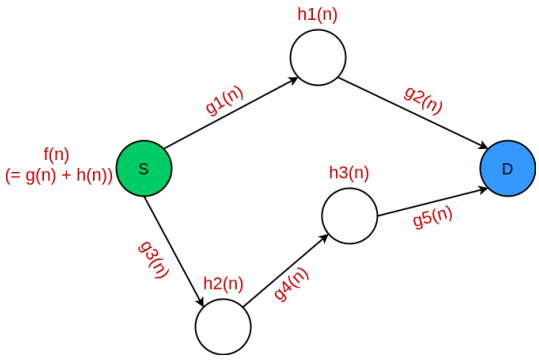

Fig. 2. An illustrative example for A* search process and cost function.

Algorithm 1: Modified A* Search For Automatic Routing

```
Result: Optimal path between source and
        destination cells.
append source cell to the path;
while destination not reached do
    get adjacent cell that minimizes euclidean
     distance;
    append cell to the path;
    if current cell is blocked then
        remove current cell from path;
        backtrack to the previous cell;
        continue;
    end
end
```

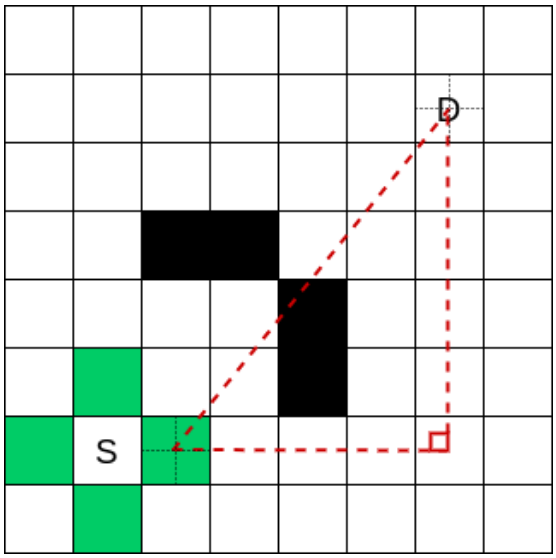

Fig. 3. An illustrative example for the new heuristic function based on Euclidean distance.