Monotonic Escape Routing with Matched Ordering

Tan Yan

1 PCB routing

A modern PCB usually host a couple of chip packages whose footprints on board look like pin grids (see Figure 1). Such pins on the board are expected to be connected by non-crossing wires. Not all connections can be routed on one layer, so we may need multiple layers to accommodate all the wire connections. However, on each layer, the routing of the wires must be planar.

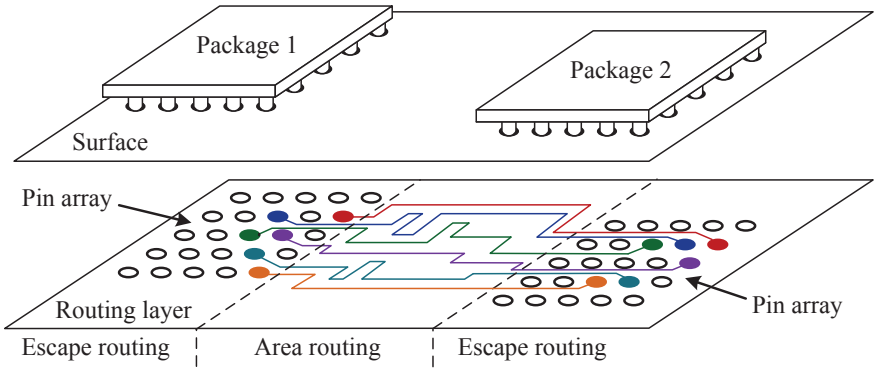

Figure 1. An illustration of PCB routing.

We usually perform the routing in two stages:

1. Escape routing: route from pins inside the grids to the boundary of the grids.
2. Area routing: route between the grids.

The two stages have different tasks. The main task of escape routing is to obtain a planar topology such that the escaped wires have matching ordering along the boundaries of the two pin grid. (See Figure reffig:pcb, the ordering of the wires along both boundaries are the same so that we have a planar topology for area routing. Escape routing must also satisfy certain geometric constraints which can be abstracted as capacity constraints. I will talk about this later. Area routing needs to follow the planar topology (thus the ordering in the middle) produced by escape routing. In some high performance designs, lengths of the wires have very strict min-max bounds. So we sometimes need to bend the wires to meet the min length bounds during area routing. The focus of this research is on escape routing.

2 Escape Routing Problem

We can formulate the escape routing problem as follows: Given two regular pin grids of size $r_l \times c_l$ and $r_r \times c_r$ (denoting the number of rows and columns of the pin grid on the left and on the right), with n pins in each pin grid marked as $\{1, \ldots, n\}$, find planar paths from each marked pin to the boundaries of the two grids such that the ordering of the paths match at the two boundaries (see Figure 2 for an example). When we say the ordering of

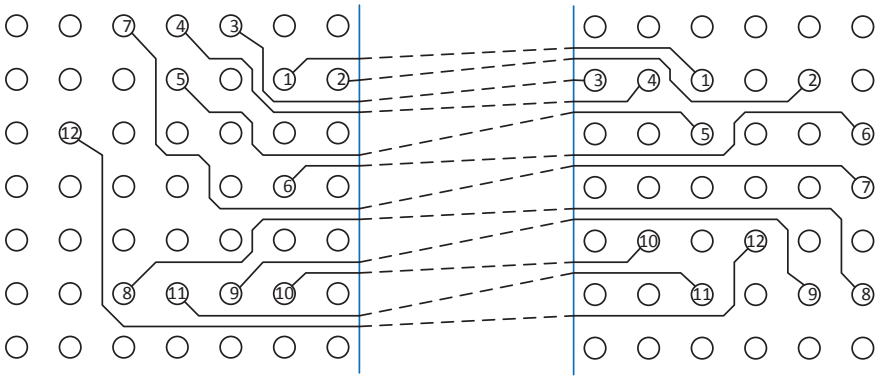

Figure 2. An example of escape routing.

the paths, we mean we mark a path with the same number as its source pin. Then we compare the marked numbers of the paths along the boundary from top to bottom and obtain a permutation of $\{1, \ldots, n\}$. If the permutations of both sides match, then we know that a planar topology is obtained for later area routing. Matched ordering is not the only requirement. We also have capacity constraint which limits the number of paths between adjacent pins. The *wiring capacity* is the maximum number of paths that is allowed between two vertically or horizontally adjacent pins (See Figure 3. In practice, we can assume this capacity is a small constant, usually 2.

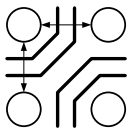

Figure 3. An example with wiring capacity 2.

This escape routing problem is very similar to the general routing problem on regular grid. (The only differences are general routing problem on a 2-D grid has wiring capacity only one and instead of two ordering matching, they have all the pins inside one pin grid.) Therefore, we believe this problem is NP-hard. (In my field, people are becoming lazy on showing something is NP-hard and then provide efficient heuristics. They usually show that the problem is closed related to some NP-hard problem and then just believe the problem is also NP-hard.) Therefore, without adding certain constraint on the problem, we do not think it is solvable.

2.1 Monotonic Routing

A very practical routing constraint is *monotonic* constraint. In our problem, a path is called monotonic if the intersection of the path and a vertical line is either nothing or a single point or a continuous line segment. See Figure 4 for an example. The routing in the left is montonic because any vertical line can intersect it at a single point or a line segment. The right side shows two examples that do not qualify. We can find vertical lines that intersect each of the path at more than one point. In fact, the sample routing in Figure 2 is monotonic.

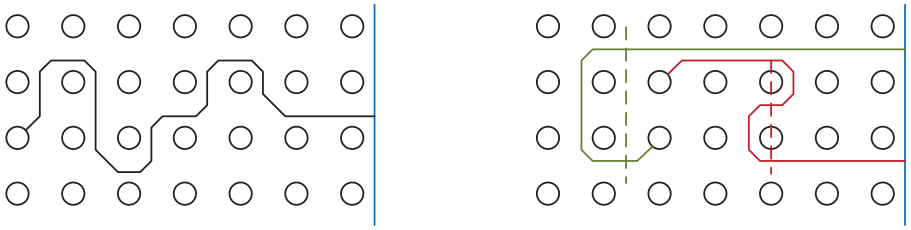

Figure 4. Monotonic routing (left) vs. non-monotonic routing (right).

Now with this monotonic routing constraint, we wonder if there is a polynomial-time algorithm to solve the escape routing problem. Idealy we want the algorithm to be polynomial in n and not depend on the grid size because we may encounter sparse cases. However, polynomial dependency on the size of the grid is also tolerable.
One comment is that if we know the ordering of the paths along the boundary, then a greedy algorithm will produce a monotonic routing on both sides. However, we cannot enumerate all possible ordering because there are $O(n!)$ of them.