Paper Title, 6-page limit*

[*]Sept. 9th 3PM (VAN, CA)/Sept. 10th 6AM (Taipei, TW) Abstract submission

Sept. 16th 3PM (Van, CA)/Sept. 17th 6AM (Taipei, TW) Full manuscript submission

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

***Abstract*—To be written**

***Index Terms*—Back-end design, physical design, placement, routing, routability [1]**

I. Introduction

A. Motivation

With the rapid development of technology nodes, design rules become complicated, and numerous of those have imposed upon layouts to secure viable fabrications. In other words, routability has raised as one of the most important factor to consider during the circuit design workflow. For example, Take the minimum spacing rule for example. Space between two adjacent routes began to depend on the metal width at the 130nm node, and it started to rely on the width of these two adjacent wires at the 65nm node. This minimum spacing rule becomes more complicated at the 32nm node. The spacing depends not only on the factors mentioned above but also on their runlength, which is the total length of two adjacent wires run parallel. This phenomenon indicates that one cannot simply sum the widths of all wires in a certain area and assume that this suffices to check whether the routability constraint is satisfied for that said region.

(And/or other examples that show the unscalable and unviable of conventional approaches for solving routability problems)

B. Previous Works

With the growing design complexity, ”congestion map” is being used as the metric to deal with routability-driven applications, it indicates regions where routing will be difficult to achieve. Attempts to solve this problem are normally taken place in the early physical design stage. Existing approaches can be categorized as follows:

- Static approaches, where congestion maps are fixed for placement, such as using rent’s rule or pin density to determine the routability.
- Probabilistic approaches, where net topologies are probabilistically generated based upon placement.
- Constructive approaches, where a global router is used to generate approximated congestion maps by performing a fast routing.

(Cite other works, briefly talk about the methodologies and limitations of those works in a couple of sentences)

C. Overview of Our Work

(Briefly talk about how our work can solve the scalability issue that conventional methods cannot, and what novelties we are to bring out)

Our key contributions are as follows:

- We present a machine learning model to efficiently generate congestion information.
- Define and extract effective features at post-placement stage
- A probabilistic algorithm is developed and embedded into our new algorithm as a feature for model training
- *(More to be added)*

II. Preliminaries

(Most terminologies will be given here)

A. Routability Problem Formulation

The key objective of global routing is to maximize routability of the design, which is the quality of routing the circuit subject to net and design constraints.

The routing problem can be defined as:

$$maximize \quad f(X) \tag{1}$$

subject to

$$\begin{aligned} d(X) \leq 0 \\ n(X) \leq 0 \end{aligned} \tag{2}$$

Here the constraints are categorized into two classes, network constraints $n(X)$ and design constraints $d(X)$. Network constraints indicate that the topology of the network must satisfy certain requirements, for example, pins of each net has to be all connected without loops formed. Design constraints, on the other hand, state that the implementation of the network topology is legal, such as no overlapping routes, routing direction follows the layer specifications. Our research fits within the routability optimization problem. In particular, we are minimizing the net constraint of congestion.

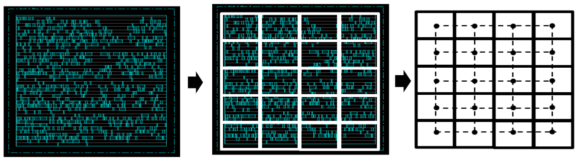

Fig. 1. Design is divided into rectangular tiles (solid lines). The dark circles and dashed lines represent the vertices and the edges of the grid graph, respectively.

B. Grid Model

Global routing is performed on a grid which is the logic graph representation of the physical chip. The grid graph G is tile-based, which means that the entire chip is partitioned into multiple sub-regions, as illustrated in Fig. 1. Each sub-region is called grid cell, or global routing cell. The cells are further gridded with vertical and horizontal gridlines, referred to as routing tracks, along which routes are determined and wires can be created. Each vertex in G represents a corresponding tile, and each edge in G between vertices corresponds to the shared boundary of two tiles.

C. Routing Metrics for Network Constraints

1) Routing Capacity: The graph G used for global routing needs to capture the capacities of the routing regions. The capacity of an edge $e \in G$ between two vetices u and v is defined as the maximum number of available routing tracks between the routing regions of u and v.

High precision capacity metric can be complex and possible in practice, such simple track-based capacity can be extended to consider specific routing elements such as blockages, vias and pins.

2) Congestion: If the total usage of one edge is larger than its capacity, then the tile containing that edge is congested. Detailed routing will not be able to route all nets assigned to congested areas due to lack of routing resources. However, it is in some degree tolerable to the detailed router as it can spread the wires from the congested tiles to adjacent less-congested tiles, if there are any.

3) Wirelength: The minimization of wirelength is another important metric for global routing. Decreased wirelength implies smaller power consumption and delays, which are two key factors for performance-driven optimization. Inherently, congestion minimization conflicts with wirelength minimization, because detours may be introduced to avoid congested regions and that leads to longer wirelength. Therefore, the trade-off between these features has to be carefully tuned based on the requirements of circuit design.

D. Supervised Data Learning technique

Data-learning is a powerful technique which can drive knowledge from big data, predict and generalize unseen data pattern. There are four main categories: Supervised, Unsupervised, Semi-supervised, and reinforcement learning. In the research field of physical design algorithms, supervised learning is widely used.

As is illustrated in Fig. 2, given certain features as the input, the training process builds a set of models, which will then be evaluated. One way of testing the quality of models is by feeding the same input features as were fed during training to see if predicted values are close enough to the original target output. After the evaluation, acceptable models are selected based on different purposes to make predictions and decisions for various applications. There are two types of applications fall under this *ML* category:

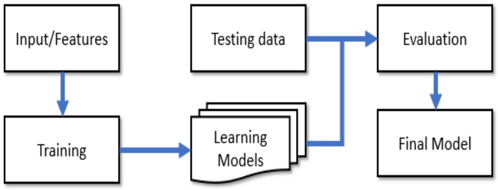

Fig. 2. Supervised learning flow

- Regression. Given input to predict continuous values. For example, guess the height of one individual human based on one's age.
- Classification. The goal is to predict discrete values. Determine if an email is spam or not, a human is female or male.

Examples of supervised learning algorithms include, decision trees, linear regression, support vector machines, neural networks, Multivariate Adaptive regression Splines.

III. Proposed Methodology

(Picture diagram of the algorithm to be added, also talk about why NTHU is selected, because it has open source available, the performance is the best among all other academic routing tools, etc.)

The machine learning algorithm is required to predict the global routing pattern of a design. Thus, it needs to be trained to create an accurately predicted model. For the purpose of training, certain attributes need to be extracted from a design that has already undergone global routing. Along with these attributes, post-routing congestion information for each edge is also extracted. Two matrices, one containing these attributes and another populated with the congestion information act as input and output values required to train the machine learning model. Depending on the size of designs, more than one can be used for training.

Since the congestion in the NTHU router is calculated using an edge-based perspective, all the attributes of the design are observed using a similar approach. The following attributes of the design were observed and extracted based on a theoretical formulation of what would impact congestion values the most for each edge.

A. Attribute Extraction

- The first-degree pins. Those of an edge are the pins that lie within either of the two tiles that edge is a part of. Fig. 3 depicts a visual representation of this attribute.These are one of the closest pins to an edge and are deemed to have a significant impact on the probability of the routing wire to cross-over the chosen edge, thus altering its congestion value. An edge is most likely to be crossed by a wire while routing pins that are closest to it. This attribute provides a metric for proximity or vicinity of pins with respect to an edge. A pin is closest to all the four edges surrounding it than any other edge. To route that pin, at least one of these four edges will be crossed by the wire. Thus, we used this feature to train the machine learning engine.

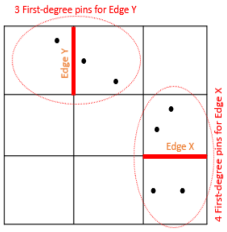
3 First-degree pins for Edge Y
Edge Y
Edge X
4 First-degree pins for Edge X

Fig. 3. First degree pins for two edges

- The first-degree nets. A tile can contain various pins which belong to different nets. Different pins in a tile can be a part of the same net or of different nets. Analogous to first degree pins, this attribute gives the number of nets that are closest to an edge. The first-degree nets of an edge are the nets that cross-over the space enclosed by either of the two tiles that the chosen edge is a part of. Fig.4 gives an example of this attribute.

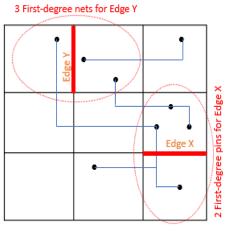
3 First-degree nets for Edge Y
Edge Y
Edge X
2 First-degree pins for Edge X

Fig. 4. First degree nets for two edges

- The pin density. This attribute makes use of First-degree pins to calculate the pin-density. It gives information regarding the number of pins per unit area. Pin density for a chosen edge is given by the number of first-degree pins divided by twice the tile area i.e., area of both tiles the edge is a part of. Pin density helps to give information of how congested the pins are within a tile which directly translates to a higher probability for the tiles corresponding edges to be used for routing. A higher pin density means that the edges of this tile can mostly be used to route pins within that tile. This deems the edges un-routable for wire connecting other tiles and thus a detour must be made.
- RSMT-aware two-pin-pair usage. The NTHU router initially creates multi-pin net structures using the Rectilinear Steiner Minimal Tree (RSMT) algorithm and then fragments each structure into two-pin pairs. To create these fragments, two pins are chosen and then a virtual boundary is laid such that it makes a rectangle using the two pins as its opposite vertices. Then, either side of the rectangular perimeter is selected to create an L-shaped path. In an edge-based routing approach, the two-pin boundary attribute is a value assigned to each edge which is updated when each edge is crossed over perpendicularly by the perimeter of this virtual rectangle created using a two-pin pair. An example is demonstrated in Fig.5.

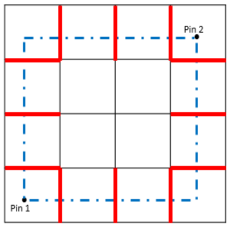

Fig. 5. Edges in red with the usage attribute updated

- *(Blockage-aware detouring usage, to be articulated)*

B. Prediction Model Embedded Routability optimization

To be written

IV. Experimental Result

(explain the experiment conditions, compare results, claim advantages and disadvantages of our work)

(Provide raw data in tables)

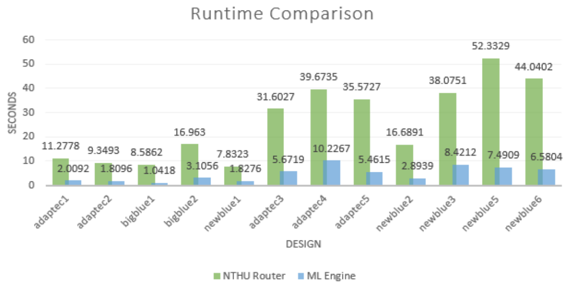

Fig. 6. Time spent on estimating congestion information.

V. Conclusion

To be written, conclude the work, talk about possible future extensions

References

[1] S.Zhou, “void reference to prevent corrupted .bbl file,” in *Proc. ECOC’00*, 1234, p. 000.