A Novel Wire-Density-Driven Full-Chip Routing System for CMP Variation Control

Huang-Yu Chen, *Student Member, IEEE*, Szu-Jui Chou, Sheng-Lung Wang, and Yao-Wen Chang, *Member, IEEE*

Abstract—As nanometer technology advances, the post chemical–mechanical polishing (CMP) topography variation control becomes crucial for manufacturing closure. To improve the CMP quality, dummy-feature filling is typically performed by foundries after the routing stage. However, filling dummy features may greatly degrade the interconnect performance and significantly increase the input data in the following time-consuming reticle enhancement techniques. It is, thus, desirable to consider wire-density uniformity during routing to minimize the side effects from aggressive post-layout dummy filling. In this paper, we present a new full-chip grid-based routing system considering wire density for reticle planarization enhancement. To fully consider a wire distribution, the router applies a novel two-pass top-down planarity-driven routing framework, which employs new density critical area analysis based on Voronoi diagrams and incorporates an intermediate stage of a density-driven layer/track assignment based on incremental Delaunay triangulation. Experimental results show that our methods can achieve a more balanced wire distribution than state-of-the-art works.

Index Terms—Design for manufacturability, layout, physical design, routing.

I. INTRODUCTION

AS IC process geometry shrinks to 65 nm and below, one important yield loss of interconnects comes from the *chemical–mechanical polishing* (CMP), which consists of *electroplating* followed by CMP [29].

As the denotation, CMP contains chemical and mechanical parts. A typical schematic of the CMP process is illustrated in Fig. 1, where abrasive *slurry*, which chemically dissolves the wafer layer, is distributed over the surface of the polishing pad, and a dynamic polishing head mechanically presses the
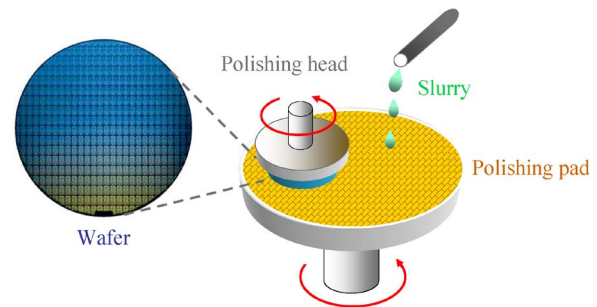

Fig. 1. Schematic of the CMP polisher.

polishing pad and the wafer. After CMP, the surface of the wafer would become smooth and flat.

Because of the difference in hardness between copper and dielectric materials, the CMP planarizing process might generate topography irregularities. A non-uniform feature density distribution on each layer causes CMP to over polish or under polish, generating metal dishing and dielectric erosion [31]. These thickness variations have to be controlled carefully since the variation in one interconnect level is progressively transferred to subsequent levels during manufacturing, and, finally, the compounding variation can be significant on an upper level, which is often called the multilayer accumulative effect [32].

Two key problems arise from the post-CMP thickness variation.

1) The layout surface fluctuates inside or outside the *depth of focus* of the photolithography system, such that the exposed patterns do not appear acceptably sharp, and open/short defects may even occur.
2) These irregular variations greatly change the electrical characteristics of interconnects, particularly for resistance and capacitance, degrading the accuracy of timing analysis and worsening the electromigration.

These issues become critical in the nanometer designs and have recently attracted much attention [23]. As a result, to improve chip thickness uniformity, the Taiwan Semiconductor Manufacturing Company (TSMC) recommends performing virtual CMP analysis to identify the metal and dielectric thickness variation hot spot before chip fabrication for 65-nm manufacturing processes (see [33]).

To improve the CMP quality, modern foundries often impose recommended layout density rules and fill *dummy features* into layouts to restrict the variations on each layer. For example, on each metal layer, local pattern density within every predefined window must be within a specified range, for example, between 10% and 40%. These density bounds can also help in minimizing the multilayer accumulative effect, as reported in [26].

H.-Y. Chen is with the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: yellowfish@eda.ee.ntu.edu.tw).

S.-J. Chou and S.-L. Wang are with the Synopsys Taiwan Ltd., Taipei 110, Taiwan (e-mail: shirleyc@synopsys.com; slwang@synopsys.com).

Y.-W. Chang is with the Graduate Institute of Electronics Engineering and the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: ywchang@cc.ee.ntu.edu.tw).

Dummy features may either be connected to power/ground (P/G; *tied fills*) or left floating (*floating fills*) [27]. The tied fill has predictable but higher capacitance, whereas the floating fill has lower but unpredictable one due to the floating nature. Traditionally, electrical impacts of dummy fills can be negligible, and dummy features are inserted during the post routing stage. Filling algorithms have been proposed to satisfy density bounds and reduce the density variation [22], [34]. However, as reported in [35], these filled dummy features may incur troublesome problems at 65 nm and successive technology nodes. The tied fill may induce crosstalk for its high coupling capacitance to nearby interconnects and would place a heavy burden for the P/G network. On the other hand, the capacitance of floating fills is usually uncertain, and, thus, its capacitance extraction requires additional efforts to solve the same matrix system of nonfloating capacitance with more unknowns and some transformation [36].

Moreover, dummy fills also sheerly increase the data volume of mask, lengthening the time of mask-making processes such as mask synthesis, writing, and inspection verification. In particular, these filled features would significantly increase the input data in the following time-consuming reticle enhancement techniques, such as optical proximity correction and phase shift mask. Therefore, much research focuses on impact-limited dummy-feature-filling algorithms [8], [25].

In the nanometer technology, routing has become a decisive factor for determining chip manufacturability since it presides over most of the layout geometry in the back-end design process. To tackle these manufacturing challenges, routing techniques must handle the increasing complexity. The routing approaches applying the bottom-up *coarsening* and top-down *uncoarsening* techniques have demonstrated the superior capability of handling large-scale routing problems, such as the Λ-shaped multilevel [4], [5], [18], the V-shaped multilevel [6], and the two-pass bottom-up [7] routing frameworks.

Recently, routing considering a wire distribution has attracted much attention in the literature. The earlier studies for CMP processes have indicated that the post-CMP topography is highly correlated to the layout pattern density because, during the polishing step, interlevel dielectric removal rates are varied with the pattern density [32]. Furthermore, the layout pattern (consisting of wires and dummy features) density can be systematically determined by the wire-density distribution, as reported in [13]. Specifically, as shown in Fig. 2, for a given global cell v_i with a metal density m_i, the Cu thickness of v_i, i.e., t_i, can be expressed as follows [13]:

$$t_i = \alpha \left(1 - \frac{m_i^2}{\beta} \right) \quad (1)$$

which shows that the post-CMP copper thickness is determined by the wire density.

Therefore, managing wire density at the routing stage has great potential for alleviating the aggressive dummy-feature-filling-induced problems.

Li *et al.* [28] presented the first routing system in the literature addressing the CMP-induced variation. By setting the desired density in the cost function of global routing, the routing results have a more balanced interconnect distribution.
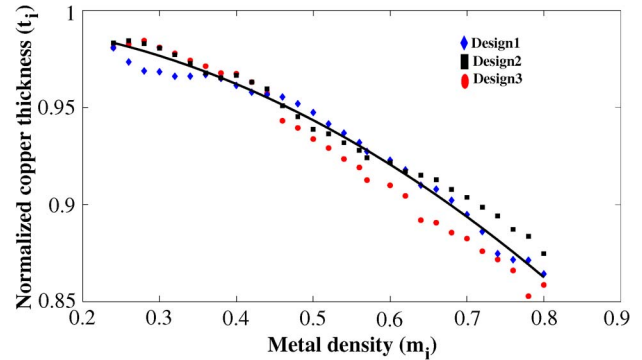

Fig. 2.   Normalized Cu thickness by metal density [13].

Cho *et al.* [13] proposed a pioneering work to consider CMP variation during *global* routing. They empirically developed a predictive CMP density model and showed that the number of inserted dummy features can be predicted by the wire density. Therefore, they proposed a minimum-pin density global routing algorithm to reduce the maximum wire density in each global tile. However, both approaches only consider the wire density *inside* a routing tile. Since the topographic variation is a long-range effect, focusing the density value inside each routing tile may incur larger intertile density difference and result in more irregular post-CMP thickness [see Fig. 3(a)]. Therefore, optimizing wire-density uniformity inside a routing tile is obviously not a right metric and a common pitfall for CMP control. For better CMP control, it is more desirable to minimize the global variation of the wire density, i.e., the density gradient, which is a major objective for dummy fills [10]. As shown by the example in Fig. 3, if the density lower and upper bounds are 20% and 80%, respectively, then the three adjacent routing tiles in Fig. 3(b) all satisfy these rules. However, Fig. 3(c) is a better choice for CMP control because it has the minimum wire-density gradient.

In this paper, we present a new full-chip grid-based routing system, named the two-pass top-down grid-based router (TTR), considering wire-distribution uniformity for density variation minimization. To fully consider the wire distribution, the router is based on a novel two-pass top-down planarization-driven routing framework, which consists of four major stages (see Fig. 4 for an illustration):

1) prerouting: identify the potential density hot spots based on the pin distribution and the wire connection to guide the following global routing;
2) global routing: apply prerouting-guided planarization-aware global pattern routing for local nets and iteratively refine the solution;
3) layer/track assignment: perform density-driven layer/track assignment for long segments panel by panel;
4) detailed routing: use segment-to-segment detailed maze routing to route short segments and reroute failed nets level by level.

Different from the aforementioned works, the TTR has the following distinguished features.

• A new routing framework of performing density prediction in the *prerouting* stage, followed by planarization-aware *global* routing at the first uncoarsening stage, an
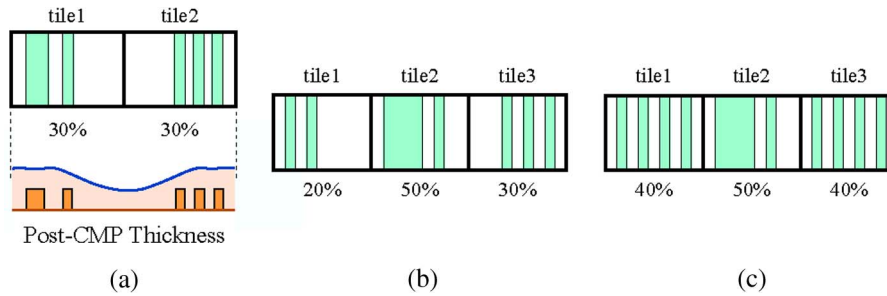
Fig. 3. Density variation among neighboring subregions impacts topography. (a) Different wire distribution in a subregion exists even under the same density. Large density variation among neighboring subregions leads to post-CMP thickness irregularities. (b) Three adjacent routing tiles satisfy density rules but result in an unbalanced wire distribution. (c) Better result for minimizing the density gradient among tiles.

Prerouting Stage
Identify the potential density hot spots based on the pin distribution and wire connection to guide the following global routing.

First Pass Stage
Apply prerouting-guided planarization-aware global pattern routing for local nets and iteratively refine the solution.

Intermediate Stage
Perform density-driven layer/track assignment for long segments panel by panel.

Second Pass Stage
Use segment-to-segment detailed maze routing to route short segments and reroute failed nets level by level.
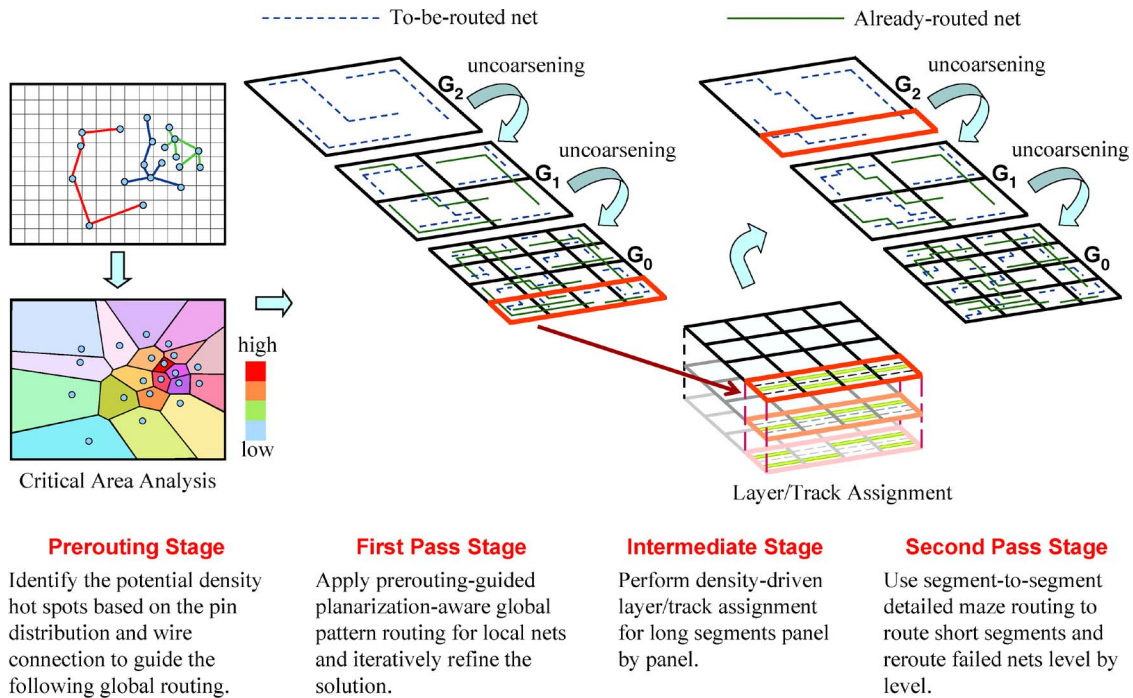
Fig. 4. New two-pass top-down planarity-driven routing framework.

intermediate stage of density-driven *layer/track assignment*, and then *detailed* routing at the second uncoarsening stage.

- An efficient density critical area analysis (CAA) algorithm based on *Voronoi diagrams* is performed *offline* in the prerouting stage, which considers *both* topological information of pins and wire connection to complement the density analysis. As shown in Section IV, the Voronoi-diagram-based CAA algorithm leads to a 3%–5% faster overall routing process due to easier density control for later detailed routing. Furthermore, it can substantially improve the resulting wire-density uniformity.
- A planarization-aware global router is employed to consider the density lower and upper bounds while minimizing the density *gradient* among global tiles.
- A layer assigner for panel-density minimization and a density-driven track assignment (DTA) algorithm based on the incremental *Delaunay triangulation* (DT) are performed before detailed routing to preserve more flexibility for wire-density arrangement.

Compared with the density-driven routing system [28], experimental results show that the TTR can achieve 43% reduction on the maximum number of nets crossing in tiles and obtain at least 35% smaller standard deviations of the wire distribution.

The rest of this paper is organized as follows. Section II describes the routing model and the routing framework. Section III presents our density-driven routing algorithms. Experimental results are reported in Section IV, and conclusions are given in Section V.

II. ROUTING MODEL

We first explain the routing model. As illustrated in Fig. 4, G_k corresponds to the routing graph of level k. Each level contains a number of *global cells* (GCs), and the GCs belonging to different levels have different sizes. We denote GC_k as the GC of level k.

The first top-down routing pass is for global routing, which starts uncoarsening from the coarsest level to the finest level (level 0). At each level k, our global router finds routing paths
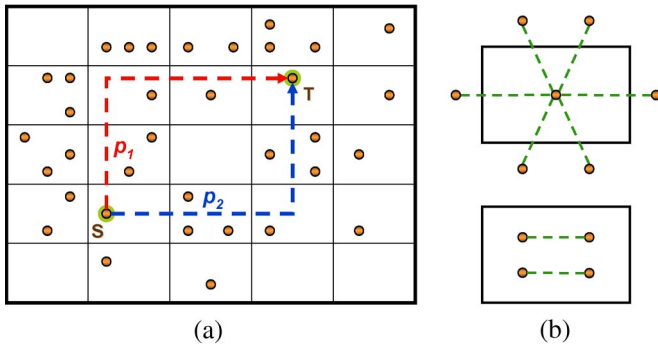
Fig. 5. Limitations of minimum-pin density routing [13]. (a) Path n_1 passes fewer pins but tends to exacerbate the overdense areas in its adjacent regions, whereas path n_2 passes more pins but leads to a better balanced wire density. (b) Pin count cannot reflect the wire density in the global tile well.

for the *local nets* (those nets that entirely sit inside GC_k but not inside GC_{k-1}). After all the global routings of level k are performed, we divide one GC_k into four smaller GC_{k-1} and, at the same time, perform resource estimation for use at level $k - 1$. Uncoarsening continues until the size of GC_k at a level is below a threshold.

The second top-down routing pass is for detailed routing. As the first pass, it processes uncoarsening from the coarsest level to the finest level. At each level, a detailed router is performed, and rip-up/reroute procedures are applied for failed nets. The process continues until we reach level 0 when the final routing solution is obtained.

III. DENSITY-DRIVEN ROUTING

To deal with wire-density optimization, we develop a two-pass top-down full-chip grid-based routing system, named the TTR (see Fig. 4). The rationale for top-down routing lies in the fact that it tends to route longer nets first level by level, which directly contributes to better wire planning since longer nets have greater impacts on planarization than shorter ones. We detail the three distinguished stages of the TTR in the following.

A. Density Critical Area Analysis (CAA)

To guide the following routing for making better decisions, the TTR features a density CAA in the prerouting stage that identifies the potential overdense hot spots. Recently, Cho *et al.* [13] have performed *minimum-pin density* routing to prevent global-routing paths from crossing through overdense areas. The reason is that a path with higher pin density tends to pass through more wire dense areas since the existence of a pin means that, eventually, there is at least one wire connecting to other pins. This approach can help reduce the wire density in each global tile. However, there is some limitation. As the global routing instance shown in Fig. 5(a), although routing path n_1 passes fewer pins, it may exacerbate the overdense areas in its adjacent regions. In contrast, routing path n_2 contains more pins but results in a better balanced wire distribution. Moreover, the pin density is not directly proportional to the wire density. As shown in Fig. 5(b), the small pin count in the global tile may still contribute to the large wire density.

Therefore, it is necessary to consider *both* the topological information and the wire connections of each pin to complement the density analysis. To remedy the deficiency, we develop a new enhanced analysis model based on *Voronoi diagrams*. The Voronoi diagram of a point set P partitions the plane into regions, called *Voronoi cells*, each of which is associated with a point of P. If a point in the plane is closer to the point $p_t \in P$ than to any other point of P, then this point will be in the interior of the Voronoi cell that is associated with p_t. The boundary segments of a Voronoi cell are called the *Voronoi edges*. A Voronoi diagram can efficiently compute the physical proximity and has been well studied in computational geometry [19]. Papadopoulou and Lee [30] used Voronoi diagrams of rectilinear polygons to compute the critical areas for short defects in a circuit layout.

The motivation for the Voronoi diagram approach lies in the following observation.

1) Observation 1: Given the Voronoi diagram of points, the standard deviation for the size of Voronoi cells strongly depends on the distribution of these points.

As illustrated in Fig. 6(a), the Voronoi cells for points with a non-uniform distribution have large variation in sizes; in contrast, as shown in Fig. 6(b), for points with a uniform distribution, the sizes of Voronoi cells are almost the same.

Another observation can quantify the proximity relation to indicate whether a point lies in the dense area.

2) Observation 2: For a point, the number of adjacent Voronoi cells that entirely sit within a specified distance from this point reflects the dense quantity of the region where this point lies.

As shown in Fig. 7(a), the point in the dense area has more Voronoi cells around it within a given circle with its center at this point.

Base on these observations, we specify a range r and associate each pin p with a density cost d_p, which is defined as

$$d_p = \alpha \nu_p + (1 - \alpha) \omega_p \qquad (2)$$

where ν_p is the number of Voronoi cells around p (excluding the Voronoi cell that is associated with p itself) that entirely sit inside the circle with a center at p and radius r, ω_p is the number of wires connecting to p, and α, $0 \leq \alpha \leq 1$, is a user-defined parameter. For the example shown in Fig. 7(b), there are three Voronoi cells around p that entirely sit inside the circle, and four wires are connected to p. Therefore, ν_p and ω_p are equal to 3 and 4, respectively.

In the current implementation, we set radius r as the average distance among the pins of adjacent Voronoi cells. This way, the expected value for ν_p would be zero if p lies in a uniformly distributed region; otherwise, ν_p would increase as a penalty to reflect the density hot spot where p lies. Additionally, since two-pin nets practically dominate the netlist in most designs, the expected value of ω_p would be equal to 1. Therefore, the ranges of ν_p and ω_p in (2) are similar and can be reasonably combined together through the α parameter.

After all density costs of pins have been computed, we transform these costs into the cost of global tiles. For each global tile t, we set its predicted density cost $\widetilde{d_t} = \max\{d_p \mid p$ is inside $t\}$ in the prerouting stage. Then, the TTR feeds the
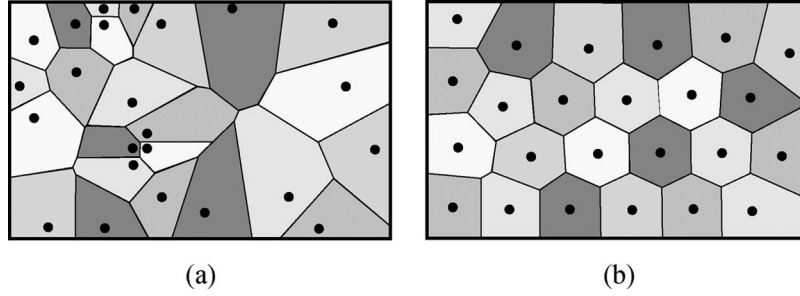
Fig. 6. Voronoi diagram for points with (a) non-uniform distribution and (b) uniform distribution.
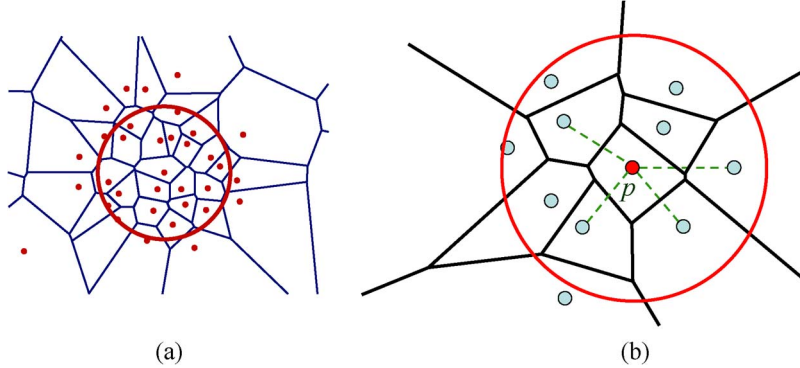

Fig. 7. Voronoi-diagram-based pin density analysis. (a) Proximity relation induced by the Voronoi diagram reflects the dense quantity well. (b) Density cost is measured by the topological proximity and the number of wire connections.

pre-estimated density information to the following routing stages. The density CAA can be performed efficiently. We have the following theorem.

Theorem 1: The Voronoi-diagram-based density CAA runs in $O(|P| \lg |P|)$ time, where $|P|$ is the number of pins.

Proof: The construction for a Voronoi diagram of $|P|$ pins takes $O(|P| \lg |P|)$ time. A Voronoi diagram contains $O(|P|)$ Voronoi edges; therefore, the computation for the average distance among the pins of adjacent Voronoi cells needs $O(|P|)$ time. Moreover, the transformation from the pin costs into the costs of global tiles takes $O(|P|)$ time. Therefore, the overall time complexity is $O(|P| \lg |P|)$. ∎

Note that the Voronoi-diagram-based CAA algorithm is performed only once, and its running time overhead is very small (about 3% of the total running time in our experiment). Furthermore, it even leads to a 3%–5% faster overall routing process due to easier density control for later detailed routing, and it can substantially improve the resulting wire-density uniformity.

B. Planarization-Aware Global Routing

The global routing plans tile-to-tile routing paths for all nets and thereby is an important step to decide the wire distribution and maintain a uniform metal density across the chip. As mentioned in Section I, both previous works [13], [28] consider only the wire density *inside* each global tile, which might incur a larger intertile density gradient and, thus, more irregular post-CMP thickness. As a result, for better CMP control, a global router has to consider the density variation (gradient) among global tiles in addition to the wire density inside each tile.

In our TTR, the global routing performed in the first top-down uncoarsening pass is based on pattern routing [24]. Pattern routing uses an L-shaped (one-bend) or Z-shaped (two-bend) route to make the connection, which gives the shortest path length between two points while reducing the routing bends. Therefore, the obtained routing path is the shortest, and we, thus, can focus on the objectives that we are most concerned.

We define planarization-aware cost Φ_t for each global tile t as follows:

$$\Phi_t = \widetilde{d_t} + \begin{cases} \kappa_p, & \text{if } d_t \geq B_u \\ \beta(2^{d_t} - 1) \\ \quad + (1-\beta)(d_t - \overline{d_t})^2, & \text{if } B_l \leq d_t < B_u \\ \kappa_n, & \text{if } d_t < B_l \end{cases} \quad (3)$$

where d_t is the wire density of t, $\widetilde{d_t}$ is the predicted hot-spot cost calculated in the prerouting stage, $\overline{d_t}$ is the average wire density of tiles adjacent to t (including t), B_l and B_u are the density lower and upper bounds specified in foundry density rules, respectively, and β, where $0 \leq \beta \leq 1$, is a user-defined parameter [note that both the values of $2^{d_t} - 1$ and $(d_t - \overline{d_t})^2$ are between 0 and 1]. κ_p and κ_n are constants, where κ_p is a positive penalty that hinders the overdenseness in the global tile, and κ_n is a negative reward that encourages paths to go through sparse tiles. (Note that, unlike maze routing, the global pattern routing allows such negative rewards.) The second equation simultaneously considers the local density and minimizes the density difference among adjacent regions.

For a more balanced wire distribution, cost function Φ_p of global routing path g_p is defined as follows:

$$\Phi_p = \text{avg}\{\Phi_t \mid \text{tile } t \text{ is on path } g_p\} \quad (4)$$
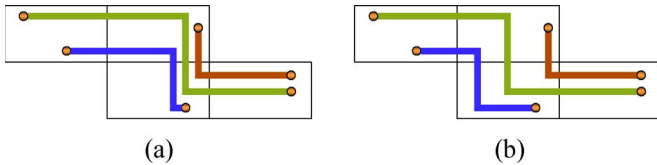
Fig. 8. Traditional routing may result in unbalanced wires. (a) Detailed routing following global routing paths may lead to an unbalanced wire distribution in tiles. (b) Detailed routing guided by wire planning can obtain better results.

in which the average manner can represent the consciousness of an even wire distribution.

Note that, as shown in Fig. 4, the TTR iteratively refines the solution during the first-pass stage. Specifically, a rip-up and reroute procedure applying (3) is performed at the end of each level to progressively fine-tune the global routing solution at that level, which not only helps reduce the net-ordering problem of sequential routing but also facilitates the transitions between levels.

C. Density-Driven Layer/Track Assignment

A recent routing system considering the balanced wire distribution often performs density-driven global routing that is directly followed by detailed maze routing [28]. However, even if the detailed router finds routing paths inside the global tiles that are determined by the global router, an uneven wire distribution may still occur in each global tile. For example, Fig. 8(a) shows a detailed routing result where an unbalanced wire distribution occurs in the tiles. In contrast, if wire planning is performed first after global routing, the detailed router could make wiser decisions and obtain better solutions, as illustrated in Fig. 8(b). Therefore, it is desirable to perform wire planning before detailed routing.

Recently, Cong *et al.* [16] have proposed the first wire-planning scheme between global and detailed routers to reduce congestion. Batterywala *et al.* [2] also suggested adding a track assignment stage between global and detailed routing to improve the routing quality. Ho *et al.* [20] developed a layer/track assignment heuristic in the intermediate stage for crosstalk optimization. Later, Ho *et al.* [21] further extended their track assigner for the wirelength reduction in X-architecture routing. Cho *et al.* [15] also performed wire planning in track assignment using second-order cone programming for yield (critical area) optimization. However, the wire density is not addressed in these works.

1) Density-Driven Layer Assignment (DLA): In this paper, we propose a new layer/track assignment algorithm for wire-density optimization. To our best knowledge, this is the first work of wire planning that addresses the wire-density optimization in the literature.

We handle long horizontal (vertical) segments that span more than one complete global tile in a row (column) in the middle layer/track assignment stage and delegate short segments to the detailed router. The full row (or column) of a global tile array is called a *row (column) panel*. We will refer to a row panel as a panel throughout this paper for brevity, unless specified otherwise.

In a panel, the *local density* of a column is defined as the total number of segments and obstacles at that column, and the *panel density* is the maximum local density among all columns in all layers. For example, Fig. 9(a) gives a row panel with 11 columns, i.e., c_1–c_{11}. There are six segments s_1–s_6 in the panel and two obstacles o_1 and o_2 in layers, and its panel density is equal to 4. We intend to evenly arrange these segments to two horizontal layers (for example, layers 1 and 3) while minimizing the panel density at each layer. The density-driven layer assignment (DLA) problem is defined as follows.

- *The DLA problem.* Given a set L of layers, a set S of disjoint segments in a panel, and a set O of fixed obstacles in layers, assign each segment of S to a layer such that, for each layer, the maximum local density is minimized, and the panel density is minimized.

Note that, for practical concern, in addition to the objectives of the DLA, a good/practical layer assigner shall also assign layers with more segments of the same nets closer to each other to minimize the stacked-via usage (i.e., minimize the layer transition of signal wires to reduce the routing resource and congestion).

To solve the DLA problem, we adopt a two-phase technique of 1) *density-driven layer partitioning* followed by 2) *stacked-via aware layer assignment*. The layer-partitioning step partitions the segments and the obstacles in each panel into $|L|$ layer groups such that the main objective of the DLA is achieved, whereas the stacked-via-aware-layer-assignment step decides the actual layer for each layer group in panels to make layers with more segments of the same nets closer to each other to minimize the stacked-via usage.

In the first layer-partitioning phase, we build the horizontal constraint graph $\text{HCG}(V, E)$ for S and O in the panel. Each vertex $v \in V$ corresponds to a segment or an obstacle, and two vertices v_i and v_j are connected by an edge $e \in E$ if their spans overlap. The cost of edge $e(v_i, v_j)$ is defined as the maximal local density among the overlapping columns between v_i and v_j. With this weighting policy, if two vertices are connected by an edge with a high cost, they should be separated into different layers. Fig. 9(b) shows the HCG of the panel in Fig. 9(a). Here, obstacle o_2 and segment s_3 overlap in columns c_3 and c_4, and the maximal local density of c_3 and c_4 is 3. Therefore, the cost of edge (o_2, s_3) is equal to 3.

Consequently, we can formulate the DLA problem as a max-cut k-coloring problem (MCP) [14] on the HCG graph, where k is equal to $|L|$. This way, we can guarantee that the partitioning result can evenly distribute the segments of the maximal local density to different layer groups. However, the MCP is NP-complete [14]. Thus, we resort to a simple, yet efficient, heuristic by constructing a maximum spanning tree on the HCG and applying a k-coloring algorithm on this tree. Note that the k-coloring algorithm on a *tree* can be solved in linear time. Fig. 9(c) shows a layer-partitioning result of Fig. 9(a), where s_1, s_2, s_3, and s_6 are partitioned as one layer group, and o_2, s_4, s_5, and o_1 are partitioned as another one. Note that the objects o_1, s_3, s_5, and s_6 at columns c_9 and c_{10} that induce the maximum local density are separated into two different layer groups.
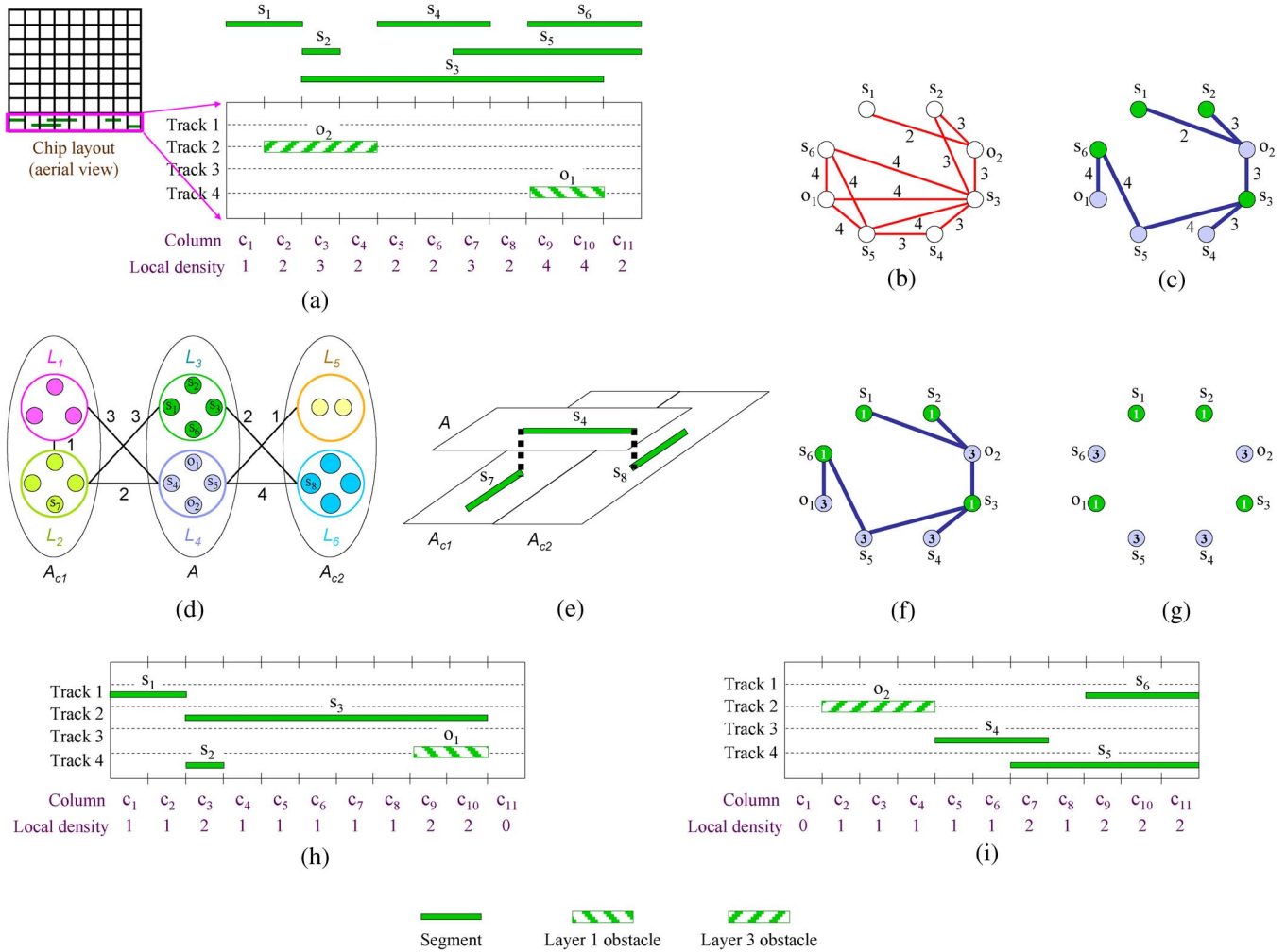
Fig. 9. Density-driven layer assignment example. (a) Row panel A consists of six segments and two obstacles. We intend to evenly assign these segments to two horizontal layers (layers 1 and 3). (b) Horizontal constraint graph. (c) Layer-partitioning result for two layer groups by applying the maximum spanning tree and k-coloring algorithms. (d) Connection graph among A and other panels A_{c1} and A_{c2}. (e) Connection among segments s_7, s_4, and s_8 that increases the connectivity by 1 among layer groups L_2, L_4, and L_6 of (d). (f) Stacked-via-aware-layer-assignment result after computing the maximum-weighted Hamiltonian path of (d). (g) Final layer assignment result by applying a minimum-impact repair procedure to exchange the layers of s_6 and o_1. (h) and (i) Final local densities of layers 1 and 3, respectively. Note that the segments are to be assigned to tracks in the following track assignment step.

In the second stacked-via-aware-layer-assignment phase, we determine the actual layer for each layer group in panels to minimize the layer transitions and, thus, the stacked-via usage. We build the *connection graph* $C(V, E)$ for all layer groups that are obtained in the first step. Each vertex $v \in V$ corresponds to a layer group, and the weight of an edge $e(v_i, v_j) \in E$ is equal to the *connectivity* between v_i and v_j. The connectivity between two layer groups is defined as the number of nets with segments in both groups. For example, if a net n has one segment s_i in layer group L_i and another segment s_j in layer group L_j, the connectivity between L_i and L_j is increased by 1. Fig. 9(d) depicts an example connection graph consisting of six layer groups, i.e., $L_1, L_2, \ldots, L_6$, in the three panels A_{c1}, A, and A_{c2}, where A is a row panel, and A_{c1} and A_{c2} are the column panels. (Note that, to simplify the presentation, the edges with zero weight are not shown in this graph). Row panel A denotes the panel described in Fig. 9(a), where layer group L_3 contains s_1, s_2, s_3, and s_6, and L_4 contains o_1, o_2, s_4, and s_5 after the layer-partitioning step in Fig. 9(c). Fig. 9(e) shows a connection among segments s_7, s_4, and s_8 that increases the

connectivity by 1 between layer groups L_2 and L_4, and L_4 and L_6 of Fig. 9(d).

To minimize the stacked-via usage, we shall make layers with more segments of the same nets closer to each other. In other words, those layer groups with larger connectivity are assigned to closer layers. This problem can be solved by first computing the maximum-weighted Hamiltonian path (MWHP) on $C(V, E)$ and then assigning layer groups with the largest connectivity closer to each other. For the example in Fig. 9(d), the MWHP is $\langle L_1, L_2, L_3, L_6, L_4, L_5 \rangle$. Let $f(\vec{l})$ be the layer assignment for layer groups $\vec{l} = (l_1, l_2, \ldots)$. Suppose that we have four metal layers, and layers 1, 2, 3, and 4 are dedicated for horizontal, vertical, horizontal, and vertical wires, respectively. Both the two layer assignment solutions $f(L_1, L_2, L_3, L_6, L_4, L_5) = (4, 2, 1, 2, 3, 4)$ or $(2, 4, 3, 4, 1, 2)$ can achieve the minimum layer transitions and, thus, the stacked-via usage. (Note that A is a row panel, and A_{c1} and A_{c2} are column panels.)

Since the MWHP problem is NP-hard, we resort to a greedy algorithm that is similar to Kruskal's minimum spanning tree

algorithm to handle the MWHP problem. We first sort edges by their weights and then add edges in a nonincreasing weight order if they form a path. Fig. 9(f) shows the layer group assignment after assigning L_3 to layer 1 and L_4 to layer 3 in Fig. 9(d).

At the last step, since obstacles are already in fixed layers, we applied a minimum-impact repair procedure for obstacles. If an obstacle is not placed in the right layer [e.g., o_1 in Fig. 9(f)], the layer of a vertex v_o of an obstacle is exchanged with that of a vertex v_s of a segment such that the edge cost (v_o, v_s) is the maximum among the edges connected with v_o in the maximum spanning tree. If there does not exist such a vertex v_s, we can just assign v_o to the correct layer since there is no segment there (otherwise, there must be an edge connected with v_o). The final assignment result after the repair procedure for exchanging the layer of vertex o_1 with that of vertex s_6 is shown in Fig. 9(g). As a result, the final assignment has a very balanced density distribution that the average local density of layer 1 is 1.18 and that of layer 3 is 1.27, while the panel densities in both layers are equal to 2. See Fig. 9(h) and (i) for the resulting segment assignments for layers 1 and 3, respectively.

2) Density-Driven Track Assignment (DTA): After the layer assignment, we intend to uniformly spread the segments in each layer of panels and balance the segment distribution among neighboring panels. For convenience, we hereafter refer to a layer of a panel as a *panel* since the layer assignment has already been performed. Let $\mathcal{T}$ be the set of tracks inside a panel. Each track $\tau \in \mathcal{T}$ can be represented by the set of its constituent contiguous intervals. Denoting these intervals by x_i, we have $\tau \equiv \biguplus x_i$. Each x_i is either

1) a blocked interval, where no segment from S can be assigned;
2) an occupied interval, where a segment from S has been assigned;
3) a free interval, where no segment from S has yet been assigned.

A segment $s \in S$ is said to be assignable to $\tau \in \mathcal{T}, \tau \equiv \biguplus x_i$, iff $x_i \cap s \neq \varnothing$, which implies that either x_i is a free interval or is an interval that is occupied by a segment of the same net. Thus, the DTA problem is defined as follows.

- *The DTA problem.* Given a panel A and its two neighboring panels A_u and A_b, a set of tracks $\mathcal{T} \in A$, a set of segments $S \in A$, and a set of fixed obstacles $O \in A$, for a given cost function, $\Psi : S \times \mathcal{T} \to \mathbb{R}$, which represents the density cost of the assignment of S to $\mathcal{T}$ that minimizes Ψ.

To solve this problem, we propose an incremental Delaunay-triangulation-based track assignment (IDTA) algorithm. In Observation 1, we have discovered the relation between the density uniformity and the Voronoi diagram. Instead of using the Voronoi diagram, we can leverage the good properties of its dual graph, called the Delaunay Triangulation (DT), to evaluate the segment distribution. The DT for a point set is triangulation that minimizes the standard deviations of angles among all triangles, and the circumscribed circle of every triangle will not contain any other point in its interior [19].

In hypsography, the DT has been successfully applied for modeling terrain to study the distribution of elevations on
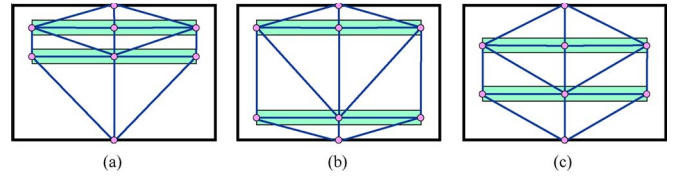

Fig. 10. DT for segments with a different distribution. Each segment is represented by three points, and two center points of the boundary line (upper and lower) are also included. (a, b) Non-uniform segment distribution. (c) Uniform segment distribution.

the surface by given a set of sample points since the DT tends to make equilateral triangles (with smaller circumcircles compared to their areas) and, thus, gives a nice set of triangles to use as polygons in the model [3]. Therefore, the DT is a suitable metric of the track-assignment problem to reflect the segment distribution since, for CMP planarization, the existence of segments in the panel is analogous to elevations on the surface. Similar to the Voronoi diagram, the standard deviation for the size of triangles in the DT can reflect the distribution of these points. Thus, we can represent each segment by three points—two endpoints and one center point—and analyze the corresponding DT of these points. (Note that, if a segment cannot be assigned as a single entity, we can separate it into subnets and then use three points to model each subnet. It is also possible to use more points for each segment, but its time complexity will be higher, which is a tradeoff between accuracy and efficiency.) Therefore, we define Ψ in the DTA problem as the standard deviation of the triangle sizes in the DT resulting from the track assignment.

As shown in the example in Fig. 10(a) and (b), if the segment distribution is non-uniform, it would result in a larger area difference among all triangles (i.e., the larger area gap between the minimum and maximum triangles); on the other hand, if the segment distribution is uniform, then the area difference would be small, as shown in Fig. 10(c). This way, the best assignment configuration in A would be the one with the minimum area difference among all triangles in the DT.

Before performing the IDTA algorithm, we first model the distribution of segments and obstacles in each neighboring panel into an *artificial segment* lying on the boundary of A to consider the boundary interaction [15]. To reflect the distribution of objects in a neighboring panel A_n of A, we set the length of an artificial segment as the average occupied length per track in A_n, and the center of this artificial segment is determined by the center of gravity of all segments and obstacles in A_n. For example, given a panel A [layer 1 after the layer assignment in Fig. 9(a)] and its two neighboring panels A_u and A_b as shown in Fig. 11(a), the length of artificial segment s_u for A_u is equal to $(3 + 6 + 3 + 4)/4 = 4$, and its center lies in $(3 \times 2.5 + 6 \times 5 + 3 \times 7.5 + 4 \times 9)/(3 + 6 + 3 + 4) = 6$. Similarly, the length of artificial segment s_b for A_b is equal to 3, and its center is at 4.5. Fig. 11(b) shows the two artificial segments s_u and s_b on the boundaries of A for modeling A_u and A_b in Fig. 11(a), respectively.

Fig. 12 shows the IDTA algorithm. Without loss of generality, we discuss the track assignment at a row panel, and the case for a column panel is similar. For the track assignment problem,
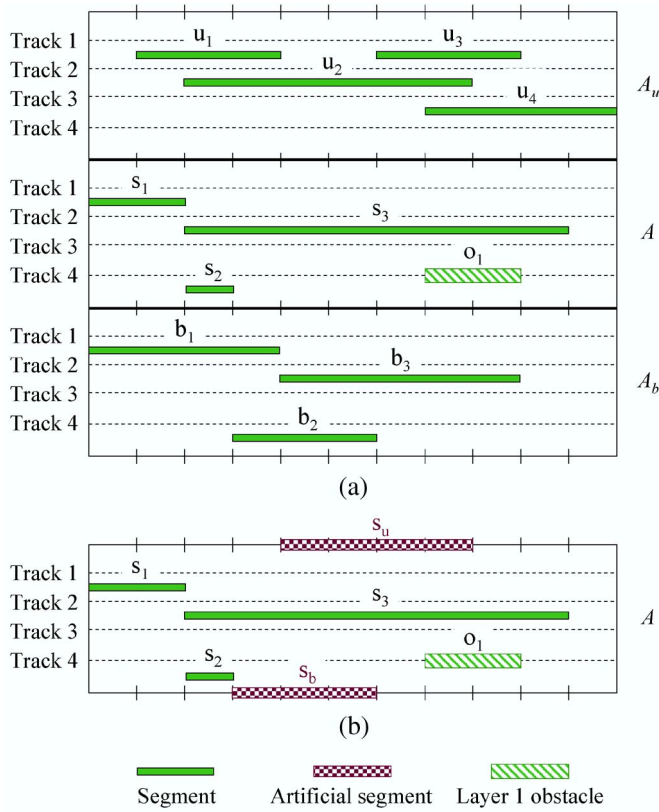
Fig. 11. Artificial segments construction. (a) Panel A and its two neighboring panels A_u and A_b. (b) Two artificial segments s_u and s_b on the boundaries of A for modeling A_u and A_b in (a).
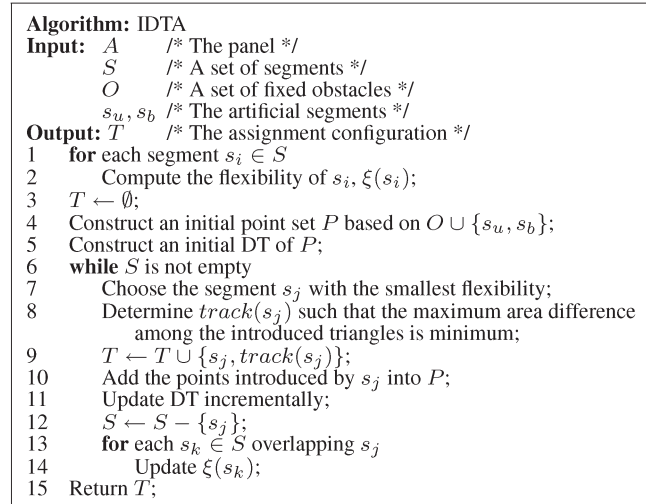

Fig. 12. IDTA algorithm.

the x-coordinates of segments are fixed (i.e., the segments in row panels can only move in the vertical direction); therefore, we can focus on the y-direction. At the beginning, we define the *flexibility* of a segment s_i as

$$\xi(s_i) = t_i + \frac{1}{\ell_i}$$

where t_i is the number of assignable tracks of s_i, and ℓ_i is the length of s_i. Since the x-coordinate of s_i is fixed, t_i can easily
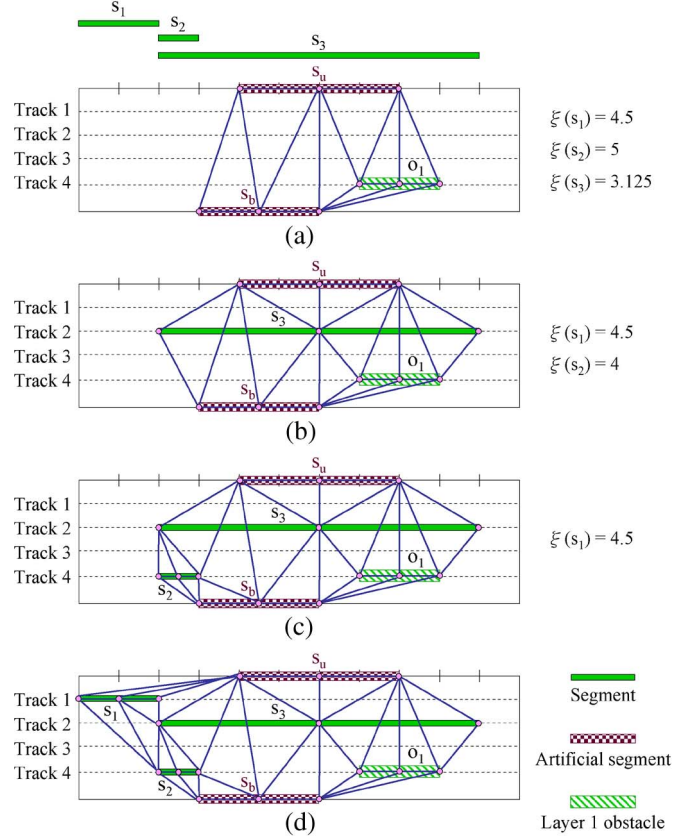
Fig. 13. Track assignment results of Fig. 11(b). (a) Initial DT. (b) Track assignment for segment s_3. (c) Track assignment for segment s_2. (d) Track assignment for segment s_1.

be computed. If the flexibility of s_i is smaller, which means that s_i might have longer length or less space to insert, then s_i should be assigned first.

After the flexibility computation, we construct initial DT that includes only the obstacles and two artificial segments. Each segment or obstacle is represented by three points—its left endpoint, center point, and right endpoint. Fig. 13(a) shows the initial DT. The construction of the DT takes $O(|P| \lg |P|)$ time, where $|P|$ is the number of points. Note that the DT can be updated incrementally; if a new point is added into the existing DT, we only need to update the triangles that are introduced by this new point. Therefore, the process can be performed very efficiently. The update will be frequently used in the following steps.

Lemma 1: Adding a new point into the existing DT of $|P|$ points takes $O(\lg |P|)$ time.

Segments are sequentially assigned in the nondecreasing order of their flexibility. Suppose that segment s_j has the smallest flexibility among all unassigned segments. Then, we assign s_j to a proper track. To minimize the area difference among all triangles, the track that results in the DT with a smaller area difference is preferred. Note that, as mentioned in Section II, since the uncoarsening stage recursively partitions the chip until the size of global cells at a level is below a threshold regardless of the design size, the number of tracks in a panel is a constant. Therefore, it requires only constant time to find the best track.

After assigning s_j to track $track(s_j)$, we need to update the DT and the flexibility of segments. Since we can incrementally update the DT, only the new triangles that are introduced by s_j need to be regenerated. Only the segments that overlap s_j and are originally assignable to $track(s_j)$ need to update their values of flexibility. For those segments, the new flexibility would be the original flexibility minus 1. The number of segments overlapping with s_j is bounded by $\ell_j \times t_j$, which is bounded by the constant size of the panel; here, ℓ_j is a value, and t_j is bounded by the number of tracks in a panel, which is predetermined before the routing and is around 10–20 in our implementation. Therefore, the total time complexity of updating the DT and the flexibility of segments is $O(\lg |S|)$, and we have the following theorem for the overall time complexity of the IDTA algorithm.

Theorem 2: The IDTA algorithm runs in $O(|S| \lg |S|)$ time, where $|S|$ is the number of segments in a panel.

    Proof: For the assignment of a segment s_j of the length ℓ_j in the DT with $|P|$ points, it takes $O(t_j \lg |P|)$ time to choose the best track location (where t_j is the number of assignable tracks of s_j), $O(\lg |P|)$ time to update the DT, and $O(\ell_j t_j)$ time to recompute the flexibility. Since ℓ_j and t_j are constants and $|P|$ is bounded by $3|S|$, the overall time complexity for assigning s_j is $O(\lg |S|)$, and, thus, the overall time complexity for the IDTA in a panel is $O(|S| \lg |S|)$. ∎

Fig. 13 shows the track assignment results coming from Fig. 11(b). Fig. 13(a) shows the initial DT, including only obstacles and artificial segments, and Fig. 13 (b)–(d) shows the assignment results of s_3, s_2, and s_1, respectively. The flexibility of unassigned segments is listed on the right side of the figures. Note that, each time when a segment is assigned, the flexibility of unassigned segments is updated incrementally.

After the track assignment, the actual track position of a segment is known. Thus, we can perform classical segment-to-segment maze routing in the detailed routing stage to connect shorter nets that span at most two routing tiles, and the whole routing process is finished.

IV. EXPERIMENTAL RESULTS

The TTR routing system was implemented in the C++ programming language on a 1.2-GHz SUN Blade-2000 workstation with 8-GB memory. We used the LEDA packages to compute the Voronoi diagrams and the DT. We conducted the experiments based on the 11 MCNC routing benchmarks (provided by Cong *et al.* [17]) and five real industrial Faraday benchmarks (introduced in [1]). Tables I and II list the two sets of benchmark circuits. Note that, in the tables, "Tile Size" gives the tile size of each circuit that the TTR used, "#Level" lists the number of uncoarsening stages that the TTR used, and r reports the radius of the Voronoi-diagram-based CAA of each circuit, i.e., the average distance among pins of adjacent Voronoi cells.

In our implementation, parameter α in (2) was set to 0.5, and parameters β, κ_p, κ_n, B_l, and B_u in (3) for all benchmarks were given as 0.5, 2, -2, 10%, and 40%, respectively. (Note that, if the wire width is equal to the wire spacing, the maximum wire density for a fully wire-occupied routing tile is equal to 50%; therefore, we set the respective density lower and upper bounds

TABLE I
MCNC BENCHMARK CIRCUITS

Circuit	Size (um^2)	#Layer	#Net	#Pin	#Level	Tile Size (um^2)	r (um)
Mcc1	45000×39000	4	802	3101	6	1407×1219	3325.1
Mcc2	152400×152400	4	7118	25024	7	2382×2382	5994.1
Struct	4903×4904	3	1920	5471	8	78×78	127.9
Primary1	7522×4988	3	904	2941	8	60×40	230.9
Primary2	10438×6488	3	3029	11226	9	83×52	209.9
S5378	435×239	3	1694	4818	6	15×8	10.8
S9234	404×225	3	1486	4260	7	14×8	10.9
S13207	660×365	3	3781	10776	7	22×12	8.0
S15850	705×389	3	4472	12793	7	23×13	7.9
S38417	1144×619	3	11309	32344	7	19×11	7.9
S38584	1295×672	3	14754	42931	8	21×12	8.2

TABLE II
FARADAY BENCHMARK CIRCUITS

Circuit	Size (um^2)	#Layer	#Net	#Pin	#Level	Tile Size (um^2)	r (um)
DMA	408.4×408.4	6	13256	73982	8	6×6	4.8
DSP1	706×706	6	28447	144872	8	10.8×10.8	6.4
DSP2	642.8×642.8	6	28431	144703	8	9.6×9.6	6.1
RISC1	1003.6×1003.6	6	34034	196677	8	15.2×15.2	6.5
RISC2	959.6×959.6	6	34034	196670	8	14.8×14.8	6.7

as 10% and 40%. Also, 0.5 means the same weighting for the two costs.)

We compared the proposed two-pass top-down routing framework of the TTR with the grid-based full-chip multilevel router considering balanced routing density in [28] (named MROR). The MROR program was provided by Li *et al.* [28] and was run on the same machine. For fair comparison, the TTR used the same setting for the size of routing tiles in all benchmarks as the MROR. Note that, as reported in [28], the MROR achieves better solutions than the previous work [4], and, thus, we shall directly compare the TTR with the MROR.

In addition, we also examined the effects of the Voronoi-diagram-based density CAA in the TTR by comparing with the minimum-pin density routing algorithm presented in [13]. Note that Cho *et al.* [13] applied their algorithm in an ILP-based global router called the BoxRouter [12]. Therefore, to focus on the comparison of the two CAA algorithms, we integrated the minimum-pin density routing algorithm into the TTR. In other words, we removed the prerouting of the TTR and replaced the cost function of the global router in (3) by the minimum-pin density routing algorithm. Note that we did not intend to directly compare with that of Cho *et al.* [13] since that work and ours have different objectives—[13] works on global routing alone, whereas ours is for the complete routing solution.

Tables III and IV show the comparison results on the MCNC and Faraday benchmarks, respectively. Note that, since the MROR program can only handle the designs with all pins lying in layer 1 (as in the MCNC benchmarks), we did not conduct the experiments on the Faraday benchmarks (where pins are distributed between layers 1 and 3) for the MROR.

Note that, according to the wire-density predictive CMP model characterized in [13], the post-CMP copper thickness is proportional to the square of the wire density. Therefore, it is desirable to minimize the variation of the wire density for CMP control, which is the main objective of this paper. As a result,

TABLE III
COMPARISON FOR THE WIRE DENSITY CONTROL ON THE MCNC BENCHMARKS

Circuit	MROR [28]						Minimum pin density global routing [13] + TTR's routing framework								TTR (Ours)							
	$\#Net_{max}$	$\#Net_{avg\_v}$	$\#Net_{avg\_h}$	σ_v	σ_h	CPU (sec)	#LG	#Seg	$\#Net_{max}$	$\#Net_{avg\_v}$	$\#Net_{avg\_h}$	σ_v	σ_h	CPU (sec)	#LG	#Seg	$\#Net_{max}$	$\#Net_{avg\_v}$	$\#Net_{avg\_h}$	σ_v	σ_h	CPU (sec)
Mcc1	45	9.9	11.3	7.6	7.3	77.4	124	2600	41	10.3	11.1	5.1	7.6	36.1	124	2639	30	10.3	11.0	5.9	6.4	33.4
Mcc2	96	18.7	20.9	17.3	18.5	2714.9	256	15814	119	20.6	22.2	14.4	19.6	798.0	256	16644	87	20.5	22.2	13.9	16.0	645.0
Struct	7	1.4	1.4	1.1	1.6	61.4	193	2128	5	1.2	0.8	0.9	0.8	66.8	167	2124	6	1.1	0.8	1.1	1.0	58.2
Primary1	15	0.7	0.6	1.2	1.8	69.1	328	2423	12	0.8	0.7	0.9	1.4	27.0	215	2207	6	0.7	0.3	0.9	0.8	24.3
Primary2	25	2.1	1.9	1.6	4.5	322.2	387	8338	22	2.5	1.9	1.3	2.8	144.0	303	7693	8	1.8	0.9	1.3	1.6	131.0
S5378	15	4.4	3.5	3.4	2.1	4.5	87	1091	8	2.5	2.4	1.6	1.5	8.1	91	1193	9	2.5	2.4	1.8	1.5	8.2
S9234	14	4.0	2.6	3.2	1.6	3.2	95	912	7	1.7	1.6	1.4	1.3	5.2	95	1003	9	1.7	1.6	1.6	1.2	5.4
S13207	27	9.3	5.9	5.2	2.8	15.8	97	1727	13	3.4	3.0	2.1	1.8	24.8	97	1821	11	3.3	3.0	2.3	1.7	24.2
S15850	26	10.3	7.4	5.4	2.9	23.8	97	1834	12	4.0	3.8	2.3	1.9	34.2	97	1915	13	3.9	3.8	2.4	1.9	33.5
S38417	23	7.3	4.3	4.4	2.2	54.2	188	5043	10	3.0	2.4	1.8	1.4	62.5	188	5462	11	2.9	2.4	2.0	1.4	62.4
S38584	29	9.1	5.8	5.4	2.9	137.7	189	6004	16	3.3	3.1	2.3	1.6	112.0	189	6328	15	3.3	3.1	2.3	1.6	112.0
Comp.	1.00	1.00	1.00	1.00	1.00	1.00	-	-	0.68	0.72	0.74	0.59	0.65	1.01	-	-	0.57	0.66	0.64	0.64	0.65	0.98

TABLE IV
COMPARISON FOR THE WIRE DENSITY CONTROL ON THE INDUSTRIAL FARADAY BENCHMARKS

| Circuit	Minimum pin density global routing [13] + TTR's routing framework										TTR (Ours)									
	Rout.	#Fail	#LG	#Seg	$\#Net_{max}$	$\#Net_{avg\_v}$	$\#Net_{avg\_h}$	σ_v	σ_h	CPU (sec)	Rout.	#Fail	#LG	#Seg	$\#Net_{max}$	$\#Net_{avg\_v}$	$\#Net_{avg\_h}$	σ_v	σ_h	CPU (sec)
DMA	99.19%	114	272	5168	14	3.14	2.77	1.70	1.77	48.8	99.29%	101	272	5325	10	3.08	2.70	1.75	1.64	47.0
DSP1	99.11%	157	264	4241	11	2.91	2.50	1.95	1.89	124.2	99.18%	145	263	4529	10	2.85	2.44	2.24	1.95	117.3
DSP2	99.10%	158	268	4676	14	2.78	2.78	1.71	1.92	87.2	99.06%	164	268	4892	10	2.72	2.70	1.90	1.91	82.3
RISC1	99.16%	223	265	5864	21	3.63	3.79	2.95	3.78	355.3	99.16%	221	265	6226	17	3.59	3.73	3.08	3.29	333.4
RISC2	99.23%	199	260	6141	21	3.64	3.70	2.55	3.08	297.4	99.19%	209	260	6533	13	3.59	3.62	2.77	2.89	280.0
Comp.	99.16%	-	-	-	1.00	1.00	1.00	1.00	1.00	1.00	99.18%	-	-	-	0.75	0.98	0.98	1.08	0.95	0.95
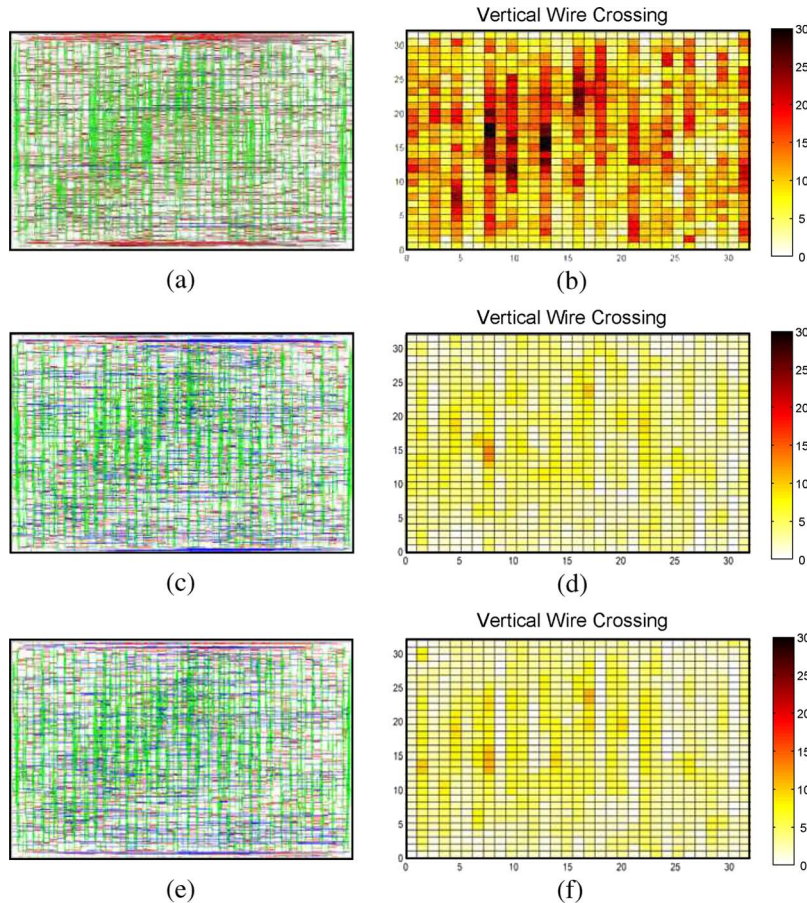
Fig. 14. Routing result and the vertical wire-crossing map in tiles for "S13207." (The red, green, and blue lines represent metals 1, 2, and 3, respectively.) (a) and (b) Routing layout and its vertical wire crossing of the MROR [28]. The maximum vertical wire crossing is 27. (c) and (d) Routing layout and its vertical wire crossing obtained from the minimum-pin density global routing [13] + TTR's routing framework. The maximum vertical wire crossing is 13. (e) and (f) Routing layout and its vertical wire crossing of the TTR (ours). The maximum vertical wire crossing is only 11.
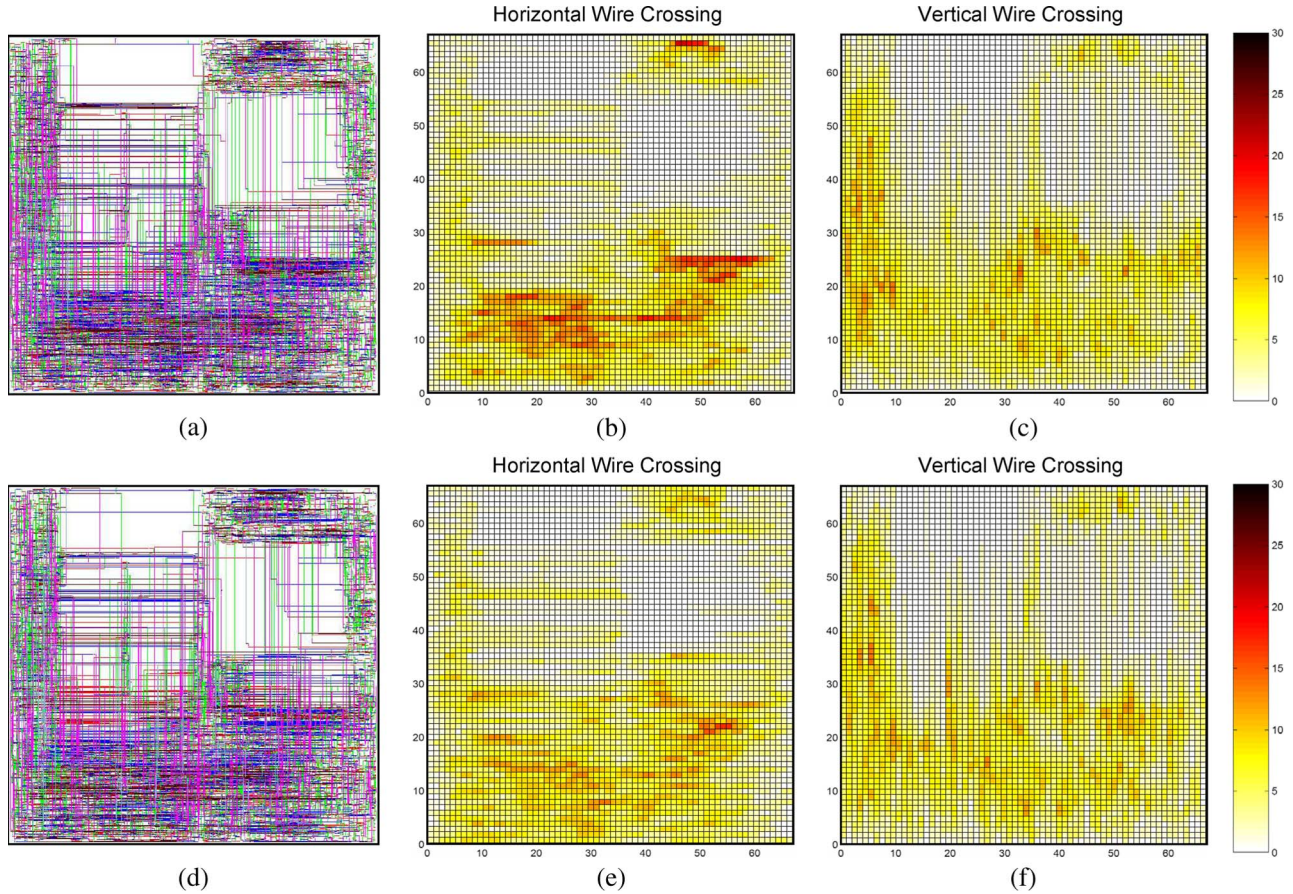
Fig. 15. Routing result and the wire-crossing map in tiles for "RISC1." (The red, green, blue, magenta, coffee, and aqua blue lines represent metals 1, 2, 3, 4, 5, and 6, respectively, and the white space is allocated by seven macros.) (a)–(c) Routing layout and its horizontal and vertical wire crossings obtained from the minimum-pin density global routing [13] + TTR's routing framework. Maximum horizontal and vertical wire crossings are 21 and 14, respectively. (d)–(f) Routing layout and its horizontal and vertical wire crossings of the TTR (ours). The maximum horizontal and vertical wire crossings are 17 and 14, respectively.

we adopted the same metrics as those used in [28], which evaluate the uniformity of the wire distribution in the routing stage.

In the tables, "Rout." stands for routability, "#Fail" gives the number of failed two-pin nets, "#Net$_{max}$" denotes the maximum number of nets crossing a level-0 tile, "#Net$_{avg\_h}$" represents the average number of nets horizontally crossing a tile ("σ_h" gives its standard deviation), and "#Net$_{avg\_v}$" gives the average number of nets vertically crossing a tile ("σ_v" gives its standard deviation). For the TTR routing systems, "#LG" denotes the total number of layer groups for the layer assignment, and "#Seg" shows the total number of segments. As shown in the tables, all routers obtain 100% routing completion on the MCNC benchmarks, and both routers applying the new framework of the TTR outperform the multilevel router MROR in wire uniformity. Compared with the MROR, the TTR incorporated with the minimum-pin density global routing algorithm reduces #Net$_{max}$, #Net$_{avg\_v}$, and #Net$_{avg\_h}$ by 32%, 28%, and 26%, respectively, and the TTR with the Voronoi-diagram-based CAA can achieve 43%, 34%, and 36% reductions on #Net$_{max}$, #Net$_{avg\_v}$, and #Net$_{avg\_h}$, respectively. Moreover, the routers using the TTR framework also result in at least 35% smaller standard deviations of the wire distribution in both directions (which implies better density smoothness) than the MROR. (Note that the measure is on the statistics of wire crossings, not for the CMP thickness directly.) From the results

reported in Tables III and IV, the global routing guided by the Voronoi-diagram-based CAA can achieve better wire uniformity than the minimum-pin density global router; in particular, it can reduce the running time of the overall routing process, although the Voronoi-diagram-based CAA itself has higher time complexity than the minimum-pin density scheme alone. The results show the effectiveness of the Voronoi-diagram-based CAA. Fig. 14 shows the routing layouts of "S13207" and the corresponding wire-crossing maps in the vertical direction for the aforementioned three routers, and Fig. 15 shows the results for the Faraday circuit "RISC1" and the horizontal and vertical wire-crossing maps. The experimental results consistently show the superior effectiveness and efficiency of our routing algorithm and framework in wire-density control.

Tables V and VI show the detailed running times of the TTR routing system on the MCNC and Faraday benchmarks. In the two tables, "PR," "GR," "LA," "TA," and "DR" stand for prerouting, global routing, layer assignment, track assignment, and detailed routing, respectively. As shown in the tables, the Voronoi-diagram-based CAA incurs only about 3% of the total running time. In particular, it even leads to a 5% faster overall routing process than the minimum-pin density global routing algorithm; the reason for the better efficiency is due to easier density control for later detailed routing for the Voronoi-diagram-based CAA.

TABLE V
DETAILED RUNNING TIME OF TTR SYSTEMS ON THE MCNC BENCHMARKS

Circuit	Minimum pin density global routing [13] + TTR's routing framework					TTR (Ours)					
	GR time(s)	LA time(s)	TA time(s)	DR time(s)	Total time(s)	PR time(s)	GR time(s)	LA time(s)	TA time(s)	DR time(s)	Total time(s)
Mcc1	0.09	0.11	6.31	29.10	36.10	0.48	0.18	0.11	6.47	25.60	33.40
Mcc2	1.03	2.20	74.40	717.00	798.00	5.61	2.23	2.38	77.80	552.00	645.00
Struct	0.02	0.03	7.84	57.80	66.80	0.41	0.02	0.03	8.37	48.20	58.20
Primary1	0.06	0.04	6.94	18.60	27.00	0.27	0.10	0.07	6.38	16.00	24.30
Primary2	0.16	0.35	31.80	110.00	144.00	1.38	0.28	0.43	29.80	96.20	131.00
S5378	0.02	0.01	1.32	6.30	8.11	0.40	0.02	0.01	1.44	5.78	8.15
S9234	0.02	0.01	0.97	3.80	5.20	0.38	0.02	0.01	1.08	3.54	5.44
S13207	0.04	0.01	2.88	20.80	24.80	0.87	0.04	0.02	3.00	19.20	24.20
S15850	0.06	0.02	3.07	29.70	34.20	1.05	0.06	0.03	3.26	27.80	33.50
S38417	0.15	0.04	7.79	50.40	62.50	2.83	0.16	0.06	8.33	46.60	62.40
S38584	0.23	0.06	9.97	83.10	112.00	5.65	0.24	0.07	10.50	76.60	112.00
Percent	0.20%	0.14%	14.84%	79.87%	100.00%	3.00%	0.26%	0.17%	15.22%	70.85%	100.00%
Comp.	-	-	-	-	1.00	-	-	-	-	-	0.95

TABLE VI
DETAILED RUNNING TIME OF TTR SYSTEMS ON THE INDUSTRIAL FARADAY BENCHMARKS

Circuit	Minimum pin density global routing [13] + TTR's routing framework					TTR (Ours)					
	GR time(s)	LA time(s)	TA time(s)	DR time(s)	Total time(s)	PR time(s)	GR time(s)	LA time(s)	TA time(s)	DR time(s)	Total time(s)
DMA	0.09	0.14	2.50	39.53	48.80	1.45	0.18	0.17	2.94	34.11	47.00
DSP1	0.09	0.12	3.70	116.99	124.20	1.80	0.14	0.12	4.53	84.64	117.30
DSP2	0.10	0.13	3.68	80.34	87.20	1.70	0.12	0.13	4.53	67.13	82.30
RISC1	0.15	0.24	6.60	338.15	355.30	2.59	0.24	0.30	8.33	261.72	333.40
RISC2	0.13	0.22	7.11	280.24	297.40	2.47	0.19	0.29	8.68	209.12	280.00
Percent	0.09%	0.14%	3.31%	91.35%	100.00%	1.67%	0.16%	0.16%	4.25%	75.90%	100.00%
Comp.	-	-	-	-	1.00	-	-	-	-	-	0.95

V. CONCLUSION

We have presented a new two-pass top-down full-chip grid-based router, named the TTR, considering the wire density for CMP variation control. The TTR features a new Voronoi-diagram-based density critical area analyzer, a planarization-aware global router, a layer assigner for panel-density minimization, and an effective track assigner based on the incremental DT. Experimental results have shown the effectiveness and the efficiency of the proposed methods.

In this paper, we have developed a metric for the DTA problem by taking three sampling points per segment and evaluating the resulting DT. Future work shall include the development of a more sophisticated model/metric for the DTA problem, *e.g.*, the investigation of the tradeoff between the number of sample points and the effectiveness/accuracy of the metric.

In this paper, we have focused on signal nets. Further work shall include the routing for special nets such as power/ground/clock nets. Power/ground/clock nets are typically routed in an earlier stage than the signal nets. Therefore, during the routing of signal nets, we can treat these routed nets as obstacles and consider their metal density by adding this preoccupied density to the planarization-aware cost [d_t in (2)] for further processing. Furthermore, large-scale designs with millions of nets are already in production, and modern router designs often need to consider multiple objectives such as wiring congestion, timing, area, reliability, and manufacturability; these requirements have reshaped the routing problem and impose significant challenges to the development of modern routers.

REFERENCES

[1] S. N. Adya, S. Chaturvedi, J. A. Roy, D. Papa, and I. L. Markov, "Unification of partitioning, placement and floorplanning," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, Nov. 2004, pp. 550–557.

[2] S. H. Batterywala, N. Shenoy, W. Nicholls, and H. Zhou, "Track assignment: A desirable intermediate step between global routing and detailed routing," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, Nov. 2002, pp. 59–66.

[3] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry*, 2nd ed. New York: Springer-Verlag, 2000.

[4] Y.-W. Chang and S.-P. Lin, "MR: A new framework for multilevel full-chip routing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 5, pp. 793–800, May 2004.

[5] T.-C. Chen and Y.-W. Chang, "Multilevel gridless routing considering optical proximity correction," in *Proc. IEEE/ACM Asia South Pacific Des. Autom. Conf.*, Jan. 2005, pp. 1160–1163.

[6] T.-C. Chen, Y.-W. Chang, and S.-C. Lin, "A novel framework for multilevel full-chip gridless routing," in *Proc. IEEE/ACM Asia South Pacific Des. Autom. Conf.*, Jan. 2006, pp. 636–641.

[7] H.-Y. Chen, M.-F. Chiang, Y.-W. Chang, L. Chen, and B. Han, "Novel full-chip gridless routing considering double-via insertion," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jul. 2006, pp. 755–760.

[8] Y. Chen, P. Gupta, and A. B. Kahng, "Performance-impact limited area fill synthesis," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 2003, pp. 22–27.

[9] H.-Y. Chen, S.-J. Chou, S.-L. Wang, and Y.-W. Chang, "Novel wire density driven full-chip routing for CMP variation control," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, Nov. 2007, pp. 831–838.

[10] Y. Chen, A. B. Kahng, G. Robins, and A. Zelikovsky, "Closing the smoothness and uniformity gap in area fill synthesis," in *Proc. ACM/IEEE Int. Symp. Phys. Des.*, Apr. 2002, pp. 137–142.

[11] M. Cho, J. Mitra, and D. Z. Pan, "Manufacturability aware routing," in *The Handbook of Algorithms for VLSI Physical Design Automation*. Boca Raton, FL: CRC Press, 2008.

[12] M. Cho and D. Z. Pan, "A new global router based on box expansion and progressive ILP," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jul. 2006, pp. 373–378.

[13] M. Cho, D. Z. Pan, H. Xiang, and R. Puri, "Wire density driven global routing for CMP variation and timing," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, Nov. 2006, pp. 487–492.

[14] J. D. Cho, S. Raje, and M. Sarrafzadeh, "Approximation for the maximum cut, k-coloring and maximum linear arrangement problems," in *Manuscript*. Evanston, IL: Dept. EECS, Northwestern Univ., 1993.

[15] M. Cho, H. Xiang, R. Puri, and D. Z. Pan, "TROY: Track router with yield-driven wire planning," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 2007, pp. 55–58.

[16] J. Cong, J. Fang, and K. Y. Khoo, "DUNE—A multilayer gridless routing system," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 5, pp. 633–647, May 2001.

[17] J. Cong, J. Fang, M. Xie, and Y. Zhang, "MARS—A multilevel full-chip gridless routing system," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 3, pp. 382–394, Mar. 2005.

[18] J. Cong, M. Xie, and Y. Zhang, "An enhanced multilevel routing system," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, Nov. 2002, pp. 51–58.

[19] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*. New York: Springer-Verlag, 1997.

[20] T.-Y. Ho, Y.-W. Chang, S.-J. Chen, and D.-T. Lee, "Crosstalk- and performance-driven multilevel full-chip routing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 6, pp. 869–878, Jun. 2005.

[21] T.-Y. Ho, C.-F. Chang, Y.-W. Chang, and S.-J. Chen, "Multilevel full-chip routing for the X-based architecture," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 2005, pp. 597–602.

[22] A. B. Kahng, G. Robins, A. Singh, and A. Zelikovsky, "Filling algorithms and analyses for layout density control," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 18, no. 4, pp. 445–462, Apr. 1999.

[23] A. B. Kahng and K. Samadi, "CMP fill synthesis: A survey of recent studies," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 1, pp. 3–19, Jan. 2008.

[24] R. Kastner, E. Bozorgzadeh, and M. Sarrafzadeh, "Pattern routing: Use and theory for increasing predictability and avoiding coupling," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 21, no. 7, pp. 777–790, Jul. 2002.

[25] A. Kurokawa, T. Kanamoto, T. Ibe, A. Kasebe, W. F. Chang, T. Kage, Y. Inoue, and H. Masuda, "Dummy filling methods for reducing interconnect capacitance and number of fills," in *Proc. IEEE Int. Symp. Quality Electron. Des.*, Mar. 2005, pp. 586–591.

[26] S. Lakshminarayanan, P. Wright, and J. Pallinti, "Design rule methodology to improve the manufacturability of the copper CMP process," in *Proc. IEEE Int. Interconnect Technol. Conf.*, Jun. 2002, pp. 99–101.

[27] K.-S. Leung, "SPIDER: Simultaneous post-layout IR-drop and metal density enhancement with redundant fill," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, Nov. 2005, pp. 33–38.

[28] K. S.-M. Li, Y.-W. Chang, C.-L. Lee, C. Su, and J. E. Chen, "Multilevel full-chip routing with testability and yield enhancement," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 9, pp. 1625–1636, Sep. 2007.

[29] J. Luo, Q. Su, C. Chiang, and J. Kawa, "A layout dependent full-chip copper electroplating topography model," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, Nov. 2005, pp. 133–140.

[30] E. Papadopoulou and D. T. Lee, "Critical area computation via Voronoi diagrams," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 18, no. 4, pp. 463–474, Apr. 1999.

[31] T. H. Park, "Characterization and modeling of pattern dependencies in copper interconnects for integrated circuits," Ph.D. dissertation, Dept. EECS, Cambridge, MA, May 2002.

[32] R. Tian, D. F. Wong, and R. Boone, "Model-based dummy feature placement for oxide chemical–mechanical polishing manufacturability," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 2000, pp. 667–670.

[33] Taiwan Semiconductor Manufacturing Company (TSMC), *Reference Flows 7.0*.

[34] X. Wang, C. C. Chiang, J. Kawa, and Q. Su, "A min-variance iterative method for fast smart dummy feature density assignment in chemical–mechanical polishing," in *Proc. IEEE Int. Symp. Quality Electron. Des.*, Mar. 2005, pp. 258–263.

[35] D. White and B. Moore, "An 'intelligent' approach to dummy fill," *EE Times*, Jan. 3, 2005.

[36] W. Yu, M. Zhang, and Z. Wang, "Efficient 3-D extraction of interconnect capacitance considering floating metal fills with boundary element method," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 1, pp. 12–18, Jan. 2006.

Huang-Yu Chen (S'05) received the B.S. degree in electrical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 2004. He is currently working toward the Ph.D. degree in the Graduate Institute of Electronics Engineering, National Taiwan University (NTU), Taipei, Taiwan.

His research interests include the VLSI design automation, the manufacturability-driven large-scale routing, and the design for manufacturability/reliability.

Mr. Chen was a winner of the 2008 Association for Computing Machinery International Symposium on Physical Design Global Routing Contest. He was also a recipient of the 2006 Outstanding Research Award from the Graduate Institute of Electronics Engineering, NTU, a Best Paper Nomination from the 2007 International Conference on Computer-Aided Design, and a Best Paper Award from the 2008 VLSI Design/CAD Symposium, Taiwan.

Szu-Jui Chou received the B.S. degree in electrical engineering and the M.S. degree in electronics engineering from the National Taiwan University, Taipei, Taiwan, in 2005 and 2007, respectively.

She is currently an R&D Engineer with the Implementation Group, Synopsys Taiwan Ltd., Taipei. Her research interests include CMP-aware routing and dummy metal filling.
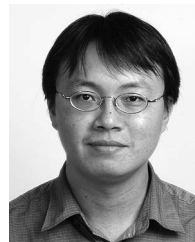
Sheng-Lung Wang received the B.S. degree in electronics engineering from the National Changhua University of Education, Changhua, Taiwan, in 1997 and the M.S. degree from the National Taiwan University, Taipei, Taiwan, in 2003.

He is currently an R&D Engineer with the Synopsys Taiwan Ltd., Taipei. His research interests include computer-aided design and design for manufacturability.

Yao-Wen Chang (S'94–A'96–M'99) received the B.S. degree from the National Taiwan University (NTU), Taipei, Taiwan, in 1988 and the M.S. and Ph.D. degrees from the University of Texas at Austin, Austin, in 1993 and 1996, respectively, all in computer science.

He is currently a Professor with the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, NTU. He is also currently a Visiting Professor with the Waseda University, Kitakyushu, Japan. His current research interests include VLSI physical design and design for manufacturability/reliability.