Track Routing and Optimization for Yield

Minsik Cho, Hua Xiang, Ruchir Puri, *Fellow, IEEE*, and David Z. Pan, *Senior Member, IEEE*

***Abstract*—In this paper, we propose track routing and optimization for yield (TROY), the first track router for the optimization of yield loss due to random defects. As the probability of failure (POF), which is an integral of the critical area and the defect size distribution, strongly depends on wire ordering, sizing, and spacing, track routing can play a key role in effective wire planning for yield optimization. However, a straightforward formulation of yield-driven track routing can be shown to be integer nonlinear programming, which is a nondeterministic polynomial-time complete problem. TROY overcomes the computational complexity by combining two effective techniques, i.e., the minimum Hamiltonian path (MHP) from graph theory and the second-order cone programming (SOCP) from mathematical optimization. First, TROY performs wire ordering to minimize the critical area for short defects by finding an MHP. Then, TROY carries out optimal wire sizing/spacing through SOCP optimization based on the given wire order. Since the SOCP can be optimally solved in near linear time, TROY efficiently achieves globally optimal wire sizing/spacing for the minimal POF.**

***Index Terms*—Minimum Hamiltonian path (MHP), physical design, random defects, second-order cone programming (SOCP), track routing, yield.**

I. Introduction

SMALLER feature size makes nanometer very large scale integration (VLSI) designs more vulnerable to ever-growing yield loss due to random and systematic causes [1]. Whereas it is believed that the yield loss due to systematic sources is greater than that due to random defects during the technology and process ramp-up stage, the systematic yield loss can be largely eliminated when the process becomes mature and tuned, and systematic variations are extracted/compensated [2]. On the other hand, the random defects that are inherent due to manufacturing facility limitations will still exist for a mature fabrication process [1]. Thus, its relative importance will, indeed, be much bigger for a mature process with systematic variations designed in. Among random defects, the density of back-end-of-line (BEOL) defects (i.e., interconnect defects) is increasing compared to that of front-end-of-line defects (i.e., device defects) [3]. Since the random BEOL defects mainly occur either between physically adjacent interconnects (short defects) or on the interconnect itself (open defects), routing and interconnect optimization should be the suitable place for random-defect-related yield optimization [1], [4], [5].

In general, routing consists of two steps—global routing and detailed routing. Global routing plans an approximate path for each net, whereas detailed routing finalizes the exact design-rule check (DRC)-compatible pin-to-pin connections. Track routing, as an intermediate step between global and detailed routing, can expedite detailed routing by embedding major trunks from each net within a panel (a row/column of global routing cells) in a DRC-friendly manner [6].

Such track routing is an appealing stage to optimize the critical area for yield enhancement, as decent flexibility in routing optimization exists with wire adjacency information [1], [7], [8], which global routing lacks, for an accurate critical area estimation. Meanwhile, detailed routing does not have sufficient flexibility to make radical routing changes for yield enhancement. Therefore, wire ordering to minimize the overlapped wirelength between adjacent wires as well as wire sizing/spacing can be effectively performed in track routing to make the design more robust to random defects.

Due to the criticality of yield in the semiconductor industry, there has been considerable effort to enhance yield by reducing the critical area in routing or postlayout optimization. Wire ordering [9], [10] and spacing [4], [11], [12] to reduce the density of short defects are explored. A redundant link [3] to improve immunity to open defects is studied. Wire spreading in the post routing optimization for yield is given in [13] and [14]. However, there are a few drawbacks in these prior works.

1) One single defect size is considered rather than a defect size distribution [9], [10].
2) The tradeoff between open and short defects due to the limited chip area is ignored [3], [9], [10], [12], [14].
3) Localized/greedy optimization is performed, which may increase the overall critical area [3], [4], [11]–[13].
4) Wire adjacency information is not available for an accurate critical area estimation [15], [16].

Indeed, it is required to find the best tradeoff between open and short defects within a fixed routing area under a given defect size distribution through *wire planning* (wire ordering, sizing, and spacing) in global scope. Accordingly, track routing is the right stage for such optimization.

So far, most recent track routing algorithms have focused on crosstalk/timing optimization [6], [8], [17], [18]; however, none of them have discussed yield optimization. At first glance, crosstalk and random-defect yield optimizations share some common traits as wire spacing helps both. However, the roles that wire ordering, wire sizing, and wire spacing play on yield and crosstalk optimizations are very different. For example, yield optimization has to consider the defect size distribution and all adjacent wires, but crosstalk optimization

Manuscript received June 28, 2007; revised September 25, 2007. This work was supported in part by Semiconductor Research Corporation, IBM Corporation Faculty Award, Fujitsu, and equipment donations from Intel. This paper was recommended by Associate Editor J. Hu.

M. Cho and D. Z. Pan are with the Department of Electrical and Computer Engineering, The University of Texas, Austin, TX 78712 USA (e-mail: thyeros@cerc.utexas.edu; dpan@ece.utexas.edu).

H. Xiang and R. Puri are with the IBM Corporation T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: huaxiang@us.ibm.com; ruchir@us.ibm.com).

Digital Object Identifier 10.1109/TCAD.2008.917589

only needs to consider those adjacent wires with overlapping timing windows; another example is power and ground wires provide shielding against crosstalk, but there is no such shielding counterpart for yield. Moreover, the algorithms in [6], [8], [17], and [18] can neither perform wire planning in a global manner nor consider the tradeoff between random defects.

In this paper, we propose track routing and optimization for yield (TROY), the first track router with yield optimization based on wire planning (wire ordering, sizing, and spacing). TROY first orders wires to minimize the overlapped wirelength between adjacent wires based on the *preference-aware minimum Hamiltonian path* (pMHP) and then performs globally optimal wire sizing and spacing for the ordered wires with efficient *second-order cone programming* (SOCP). As a result, globally optimal wire sizing/spacing as well as the minimal overlapped wirelength decreases the critical area, making a design that is more robust to random defects.

The major contributions of this paper include the following.

1) We propose TROY, a track router with yield optimization. To our best knowledge, this is the first work that yield is optimized during track routing.
2) We propose a simple model of probability of failure (POF) due to random defects. This simple yet effective model enables highly efficient and scalable SOCP.
3) We show that wire ordering within a panel (the first step of wire planning in TROY) can be efficiently solved by pMHP formulation. TROY considers the interaction between adjacent panels to overcome any disadvantage from an isolated panel-by-panel approach.
4) We show that wire sizing and spacing for an entire layer (the second step of TROY) can be formulated as the SOCP, which can be solved optimally and as efficiently as linear programming.

The rest of this paper is organized as follows. Section II presents the preliminaries. General formulation of yield-driven track routing is shown in Section III. Section IV proposes TROY as an efficient algorithm. Experimental results are discussed in Section V followed by the conclusion in Section VI.

II. Preliminaries

A. Track Routing

Track routing is an intermediate step between global routing and detailed routing to reduce routing complexity [6]. Fig. 1(a) illustrates a global routing result where an approximate routing path for a net is determined by a global router. In track routing, each routing is performed with the wires inside each panel, which is a row/column of global routing cells, as shown in Fig. 1(a). The purpose of track routing is to decide how global wires should be embedded inside the panels. An exact vertical/horizontal location should be given to a horizontal/vertical wire in each layer without violating minimum wire sizing/spacing rules. Since the location of each wire is computed during track routing, adjacency information on each wire becomes available during optimization.

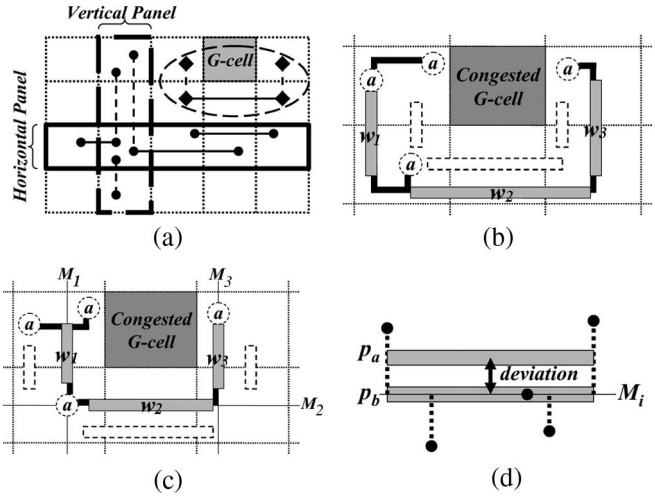

Fig. 1. Example of track routing is shown to illustrate the concept and its impact on design goals. For instance, track routing can result in a different wirelength when trunk Steiner tree is applied to the estimated expected detailed wirelength. (a) Track routing embeds global routes in panel by panel for each layer. (b) This track routing solution incurs a possibly longer wirelength than (c). (c) This track routing solution incurs a possibly shorter wirelength than (b). (d) Optimal trunk Steiner tree can be built by finding a median of all the pins of a net.

Depending on the decision of the track router, multiple design goals can be impacted, such as wirelength, crosstalk, and timing. For an example of the wirelength, two possible track routes for a net [$w_1 - w_2 - w_3$, which is shown inside the dashed circle in Fig. 1(a)] are illustrated in Fig. 1(b) and (c), respectively, where four pins are marked with a in the dashed circle. The ideal detailed routes to the pins are also drawn in solid lines, and wires from other nets are in dashed boxes. It is clear that the route in Fig. 1(c) has shorter wirelength, which can be translated into lower congestion and a smaller critical area than the one in Fig. 1(b). To achieve this, wires w_1, w_2, and w_3 need to be aligned with M_1, M_2, and M_3, which are the medians of point a in each panel. Each median (M_1, M_2, and M_3) is the optimal position of each wire in terms of wirelength if the trunk Steiner tree [19] is assumed. In general, it is not always feasible to embed all the wires in their median positions due to either the limited routing area or other design objectives.

B. Notations

Table I shows a list of notations in this paper. All constants are in uppercase, whereas all variables are in lowercase. Fig. 2 shows an example of track routing where six wires from W_1 to W_6 are assumed to be already routed (thus, p_1 to p_6 are known) within a panel P_i, which is bounded by T_i and B_i. Some examples of n_i, s_{ij}, L_{ij}, and l_{ij} are shown as well. Please note that although W_4 is between W_3 and W_5, $l_{35} = 2$ because W_3 and W_5 are adjacent and overlapped immediately before and after W_4. M_i is the median of x/y positions of all the pins in the panel where W_i exists. If $p_i \neq M_i$, we can use the deviation $|p_i - M_i|$ as a metric for a possible wirelength increase because the shortest trunk Steiner tree can be built with the median of pins [19]. Regarding the example in Fig. 1(d), if $p_i = p_b$, then the deviation is zero; however, if $p_i = p_a$, then the deviation is the distance between p_a and M_i, which is shown as

TABLE I
NOTATIONS IN THIS PAPER

W_i	wire i
M_i	preferred position of W_i for minimal wirelength
L_i	wirelength of W_i
L_{ij}	overlapped wirelength between W_i and W_j
p_i	x/y position of the center of W_i
n_i	a set of wires adjacent to W_i
l_{ij}	adjacent and overlapped wirelength between W_i and W_j
w_i	wire width of W_i
s_{ij}	spacing between W_i and W_j, $\lvert p_i - p_j \rvert - \frac{w_i + w_j}{2}$
P_i	the i-th panel
T_i	the top position of P_i
B_i	the bottom position of P_i
W_{min}	the minimum wire width of a layer
W_{max}	the maximum wire width of a layer
S_{min}	the minimum wire spacing of a layer

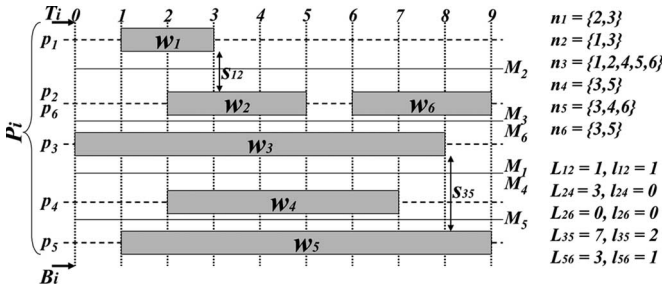

Fig. 2. Example of track routing is shown to explain the notations.

the double-headed arrow. Thus, p_i should be as close as possible to M_i for a shorter wirelength and less random defects.

C. Critical Area and the POF

The critical area for a defect is equal to the area where the center of the defect must fall to cause a circuit failure for a given defect size distribution. The POF based on the critical area analysis with the defect size distribution is a widely used metric for yield prediction and optimization [1], [7]. The defect size distribution $F(x)$ is widely modeled as follows [7], [20]:

$$F(x) = kx^{-r} \qquad \text{for } x_{\min} \le x < \infty \tag{1}$$

where x is the defect size, $x_{\min}$ is the minimum resolvable lithographic feature size, k is a coefficient to ensure $\int_{x_{\min}}^{\infty} F(x)dx = 1$, and, typically, $r \approx 3$ [21]. When the end effect is ignored [15], the critical area $A_i^o(x)$ for open defects on a wire W_i and the critical area $A_{ij}^s(x)$ for short defects between two parallel wires W_i and W_j can be approximated as follows [7], [20], [22]:

$$A_i^o(x) = \begin{cases} 0, & 0 \le x < w_i \\ L_i(x - w_i), & w_i \le x < 2w_i + S_{\min} \\ L_i(w_i + S_{\min}), & 2w_i + S_{\min} \le x < \infty \end{cases} \tag{2}$$

$$A_{ij}^s(x) = \begin{cases} 0, & 0 \le x < s_{ij} \\ l_{ij}(x - s_{ij}), & s_{ij} \le x < 2s_{ij} + W_{\min} \\ l_{ij}(s_{ij} + W_{\min}), & 2s_{ij} + W_{\min} \le x < \infty \end{cases} \tag{3}$$

where L_i, w_i, l_{ij}, and s_{ij} are as in Table I. Since the critical area cannot keep increasing, $A_i^o(x)$ and $A_{ij}^s(x)$ saturate at a defect size of $2w_i + S_{\min}$ and $2s_{ij} + W_{\min}$, respectively [20]. The POF due to open defects on W_i (POF_i^o) and due to short defects between W_i and W_j (POF_{ij}^s) on a given layer can be obtained as follows [7], [20]:

$$\text{POF}_i^o = \int_{x_{\min}}^{\infty} F(x)\frac{A_i^o(x)}{A_{\text{chip}}}dx = \frac{kL_i}{2A_{\text{chip}}}\left(\frac{w_i + S_{\min}}{2w_i^2 + S_{\min}w_i}\right) \tag{4}$$

$$\text{POF}_{ij}^s = \int_{x_{\min}}^{\infty} F(x)\frac{A_{ij}^s(x)}{A_{\text{chip}}}dx = \frac{kl_{ij}}{2A_{\text{chip}}}\left(\frac{s_{ij} + W_{\min}}{2s_{ij}^2 + W_{\min}s_{ij}}\right) \tag{5}$$

where A_{chip} is the total chip area. As POF_i^o and POF_{ij}^s indicate the chance of having a random defect, yield can be improved by minimizing POF_i^o and POF_{ij}^s together. However, minimizing POF_i^o and minimizing POF_{ij}^s are two conflicting objectives due to a fixed routing area, as larger w_i to decrease POF_i^o leads to smaller s_{ij}, which adversely increases POF_{ij}^s. As a result, it is crucial to explore the tradeoff between POF_i^o and POF_{ij}^s (thus, open and short defects) to minimize yield loss due to random defects.

D. SOCP

The SOCP can be mathematically described as a convex optimization problem, where a linear objective is optimized over the intersection of an affine linear space with the Cartesian product of second-order cones [23]–[28]. A second-order cone $\mathcal{C}$ can be classified into three types for a given $\mathbf{x} = [x_1, x_2, x_3, \ldots, x_n] \in \mathbb{R}^n$. The first is when $\mathbf{x} \in \mathbb{R}_+^n$, which degenerates the SOCP to linear programming (LP; a special case of the SOCP). The second is quadratic cone $\mathcal{C}^q$, which can be defined as follows:

$$\mathcal{C}^q = \left\{\mathbf{x} = [x_1, \tilde{\mathbf{x}}^T]^T : x_1 \ge \|\tilde{\mathbf{x}}\|\right\} \tag{6}$$

where $\|\cdot\|$ denotes the Euclidean norm. The last is *rotated quadratic cone* $\mathcal{C}^r$, which can be defined as follows:

$$\mathcal{C}^r = \left\{\mathbf{x} = [x_1, x_2, \tilde{\mathbf{x}}^T]^T : 2x_1x_2 \ge \|\tilde{\mathbf{x}}\|^2, x_1 \ge 0, x_2 \ge 0\right\} \tag{7}$$

where $\|\cdot\|$ also denotes the Euclidean norm. Then, for given $\mathbf{A}_i \in \mathbb{R}^{m \times n_i}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c}_i \in \mathbb{R}^{n_i}$, $\mathbf{x}_i \in \mathbb{R}^{n_i}$, $\mathbf{c} = (\mathbf{c}_1^T, \ldots, \mathbf{c}_r^T)^T$, and $\mathbf{x} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_r^T)^T$, the standard primal SOCP problem can be written as

$$\begin{aligned} \textbf{min} \quad & \mathbf{c}^T\mathbf{x} \\ \textbf{s.t.} \quad & \sum_{i=1}^{r} \mathbf{A}_i\mathbf{x}_i = \mathbf{b} \\ & \mathbf{x}_i \in \mathcal{C}, \quad i = 1, \ldots, r \end{aligned} \tag{8}$$

and the corresponding dual problem is defined by

$$\begin{aligned} \textbf{max} \quad & \mathbf{b}^T\mathbf{y} \\ \textbf{s.t.} \quad & \mathbf{A}_i^T\mathbf{y} + \mathbf{z}_i = \mathbf{c}_i, \quad i = 1, \ldots, r \\ & \mathbf{z}_i \in \mathcal{C}, \quad i = 1, \ldots, r \end{aligned} \tag{9}$$

where $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{z}_i \in \mathbb{R}^{n_i}$, and $\mathbf{z} = (\mathbf{z}_1^T, \ldots, \mathbf{z}_r^T)^T$.

$$\textbf{min} \quad \alpha \sum_i (POF_i^o + POF_i^{o*}) + (1-\alpha) \sum_{i,j>i} POF_{ij}^s$$

s.t.(a)	$\lvert p_i - M_i \rvert \le d_i$	$\forall i$
(b)	$S_{min} \le s_{ij} \le \lvert p_i - p_j \rvert - \frac{(w_i+w_j)}{2}$	$\forall i, \forall j \in n_i$
(c)	$B_k + \frac{w_i}{2} \le p_i \le T_k - \frac{w_i}{2}$	$\forall i \in P_k$
(d)	$W_{min} \le w_i \le W_{max}$	$\forall i$

Fig. 3. Proposed yield-driven track routing formulation is shown.

The strong duality theorem [27] guarantees that the primal and dual problems in (8) and (9) will have optimal solutions with zero duality gap (e.g., $\mathbf{b}^T\mathbf{y}^* = \mathbf{c}^T\mathbf{x}^*$, where $\mathbf{x}^*$ and $\mathbf{y}^*$ denote the optimal solutions) if strictly feasible solutions exist for both problems (e.g., $\mathbf{x}_i \in \mathcal{C}$ and $\mathbf{z}_i \in \mathcal{C}, \forall i$). For more detailed information on the SOCP, please refer to [24], [26], and [28]–[32].

The SOCP can be efficiently solved by primal–dual interior point solvers in polynomial time, and its solution is globally optimal [23], [24]. Consequently, the SOCP has found a wide variety of applications in engineering, such as filter designs, antenna array designs, robotics, neural networks, and VLSI designs [25], [33]–[35].

III. Yield-Driven Track Routing

In this section, we show yield-driven track routing in a mathematical formulation. To maximize yield, we need to minimize both POF_i^o and POF_{ij}^s in (4) and (5) by tuning the following design variables.

1) L_i: Smaller L_i linearly decreases POF_i^o. However, since L_i is mostly determined by global routing, track routing does not have enough control on this.
2) w_i: Larger w_i exponentially decreases POF_i^o.
3) l_{ij}: Smaller l_{ij} linearly decreases POF_{ij}^s.
4) s_{ij}: Larger s_{ij} exponentially decreases POF_{ij}^s.

Therefore, w_i, l_{ij}, and s_{ij} are the key variables to optimize yield. Meanwhile, we also want to minimize the expected detailed wirelength, which will be added to the current wirelength. This can be achieved by minimizing the deviation of each wire from its preferred location (see Section II-B). To accomplish this objective in the yield-driven track routing framework, we regard the deviation as the expected detailed wirelength, which is also a potential victim of open defects. Hence, we take an additional term into consideration, i.e., POF_i^{o*}, for each wire with a minimum wire width assumed, which is

$$\text{POF}_i^{o*} = \frac{kd_i}{2A_{\text{chip}}} \left(\frac{W_{\min} + S_{\min}}{2W_{\min}^2 + S_{\min}W_{\min}} \right) \tag{10}$$

where d_i is the expected detailed wirelength (or the deviation) of W_i. As POF_i^{o*} is linearly proportional to the expected detailed wirelength, we can still focus on yield maximization, which will automatically reduce the expected detailed wirelength as well.

Based on our observations, yield-driven track routing is proposed as a mathematical formulation in Fig. 3, where the objective is the weighted total POF, and α is a user-defined parameter $(0 \le \alpha \le 1)$ to control the tradeoff between open and short defects. Constraint a is about the deviation of W_i from M_i (the expected detailed wirelength) used in POF_i^{o*}, and constraint b is to guarantee that $s_{ij} \ge S_{\min}$ for any adjacent wires. Constraint c is to keep wires within the corresponding panel (this is the decision made by a global router), and constraint d is to control wire width w_i. The objective in Fig. 3 is nonlinear, and constraint b is concave. In fact, this formulation has high combinatorial complexity, as neither the order of wires is fixed nor p_i is identified. We can easily convert the formulation in Fig. 3 into an integer nonlinear programming (INLP) as in Fig. 4 by reformulating constraint b with a binary integer variable o_{ij}, which is set to 1 if $p_i > p_j$ and 0 otherwise. N is a huge constant. Optimally solving the formulation in Fig. 4 maximizes yield w.r.t. the random defects in track routing. However, this formulation is unacceptably expensive to compute even with a linearized objective function by first-order Taylor approximation (not to mention that this linearization can introduce significant suboptimality). Therefore, as an efficient and effective algorithm to solve this problem, we propose TROY in Section IV.

$$\textbf{min} \quad \alpha \sum_i (POF_i^o + POF_i^{o*}) + (1-\alpha) \sum_{i,j>i} POF_{ij}^s$$

s.t.	$\lvert p_i - M_i \rvert \le d_i$	$\forall i$
	$S_{min} \le s_{ij} \le p_i - p_j - \frac{(w_i+w_j)}{2} + (1-o_{ij})N$	$\forall i, j$
	$S_{min} \le s_{ij} \le p_j - p_i - \frac{(w_i+w_j)}{2} + o_{ij}N$	$\forall i, j$
	$o_{ij} \in \{0,1\}$	$\forall i, j$
	$B_k + \frac{w_i}{2} \le p_i \le T_k - \frac{w_i}{2}$	$\forall i \in P_k$
	$W_{min} \le w_i \le W_{max}$	$\forall i$

Fig. 4. We reformulate the one in Fig. 3 into INLP by introducing a binary variable o_{ij}, which determines the precedence between W_i and W_j in terms of x/y location in the design.

IV. TROY Algorithm

In this section, we present our track routing algorithm for yield optimization, i.e., TROY, to solve the INLP formulation in Fig. 4. TROY can solve it by combining two techniques—the *MHP* and the *SOCP*.

A. Motivation and Strategy

The key observation we make is that the INLP formulation in Fig. 4 is a kind of discrete convex optimization problem that will be degenerated to a convex optimization problem if the value of each binary variable (o_{ij}) is given. As long as it becomes a convex optimization problem, we can find a global optimal solution [23]. However, it is not sufficient to be able to find a global optimal solution itself due to the large scale of modern VLSI designs: it should be efficiently solvable by a powerful optimization technique. After further analyzing the degenerated convex formulation, we discover the following: the degenerated convex optimization problem can be cast into highly efficient SOCP (see Section II-D) if we further simplify

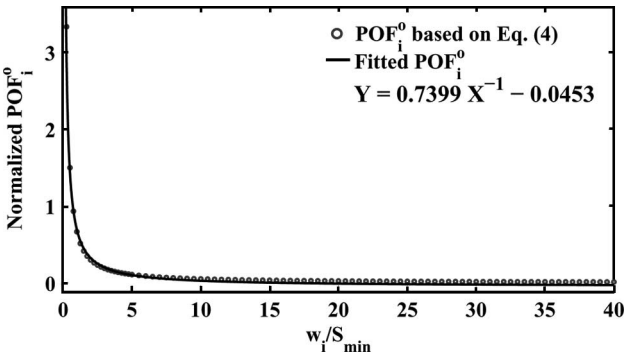

Fig. 5. Our simplified POF_i^o in (11) fits over 99.8% with the data points from (4), enabling a highly efficient SOCP formulation.

(4) and (5), which are already convex, by performing curve fitting to the following functions:

$$\text{POF}_i^o(L_i, w_i) \approx \frac{kL_i}{2A_{\text{chip}}}\left(a\frac{S_{\min}}{w_i} - b\right) \tag{11}$$

$$\text{POF}_{ij}^s(l_{ij}, s_{ij}) \approx \frac{kl_{ij}}{2A_{\text{chip}}}\left(a\frac{W_{\min}}{s_{ij}} - b\right). \tag{12}$$

According to our results, $a \approx 0.7399$ and $b \approx 0.0453$ show a regression coefficient of over 99.8% for a wide range of wire sizing and spacing. Fig. 5 shows the accuracy of our simplified POF_i^o. In general, the SOCP is known to have $O(N^{1.3})$ complexity [34], [35], where N is the number of variables and requires at most 30 iterations to solve even large problems [24]. Thus, it should be adequate to handle a VLSI track routing problem.

Finding the optimal order of wires (thus, o_{ij}) for yield can be well approximated by minimizing the total overlapped wirelength ($\sum l_{ij}$), which can be deduced to an MHP problem. In spite of the fact that the MHP is nondeterministic polynomial-time hard, it has been comprehensively studied for several decades; therefore, there exist highly efficient and near-optimal heuristics [36]. As l_{ij} solely affects POF_{ij}^s, it should have a negligible impact on the tradeoff between open and short defects, which will be optimally determined by solving the SOCP. These observations motivate our two-step TROY algorithm as follows.

1) *Wire ordering*. The goal of wire ordering is to compute yield-maximizing o_{ij} (thus, l_{ij}). In TROY, wire ordering is done in each panel such that the total overlapped wirelength between adjacent wires is minimized by finding the MHP to reduce short defects. We further propose a variant of the MHP, i.e., the pMHP, to minimize the expected detailed wirelength together. This is discussed in Section IV-B.
2) *Wire sizing/spacing*. The goal of wire sizing/spacing is to tune the wire width and the spacing such that the maximum immunity to random defects (thus, maximum yield) can be achieved. As wire sizing and spacing are conflicting objectives due to the fixed routing area, the optimal tradeoff is found by the SOCP. This is discussed in Section IV-C.

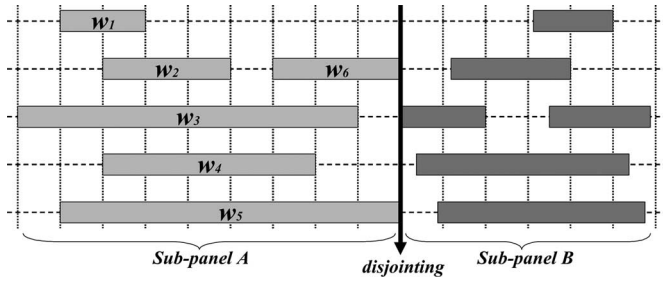

Fig. 6. Example of two disjoint subpanels.

Algorithm 1 Overall flow of TROY
Input Global routing result GR
1: **for** $\forall$ layer $L \in GR$ **do**
2: wire set $C = \emptyset$ *//to store contour wires*
3: **for** $\forall P_i \in L$ in ascending order **do**
4: Wire ordering with $P_i \bigcup C$ *// Solve pMHP*
5: $C =$ Find contour of P_i
6: **end for**
7: Wire sizing and spacing for L *// Solve SOCP*
8: **end for**

B. Wire Ordering Optimization

The goal of wire ordering is to find an order of wires such that the overlapped wirelength l_{ij} between adjacent wires is minimized to effectively reduce POF_{ij}^s. We first identify a set of disjoint subpanels within each panel such that there is no shared wire between any two identified subpanels. Fig. 6 shows an example of two disjoint subpanels, which is similar to the concept of zone in [37]. Then, wire ordering is performed from the lowest panel to the highest panel for each subpanel in each panel.

Wire ordering for each subpanel to minimize the total overlapped wirelength can be achieved by the well-known MHP [8], [10], [36]. Consider the example in Fig. 7 where six wires (W_1–W_6) are to be routed within a subpanel of a panel P_i for maximum yield. Fig. 7(a) illustrates the problem in this example. First, assuming a minimum wire width and spacing, a feasible track routing (not exceeding the number of available tracks) needs to be found through interval packing [38], as shown in Fig. 7(b), which will serve as an initial solution. Other design objectives can be considered while finding the initial solution as long as they do not conflict with the feasibility. Then, a clique as in Fig. 8(a) can be constructed by regarding each row as a vertex, and edge weight E_{ij} between two rows (thus, two vertices) V_i and V_j can be computed as follows:

$$E_{ij} = \sum_{W_i \in V_i, W_j \in V_j} L_{ij}. \tag{13}$$

Since finding an MHP from the clique is well studied, we skip the details; however, the Lin–Kernighan heuristic is shown to be very successful [36]. From the MHP, a routing solution like Fig. 7(c) may be found. However, a naive MHP approach

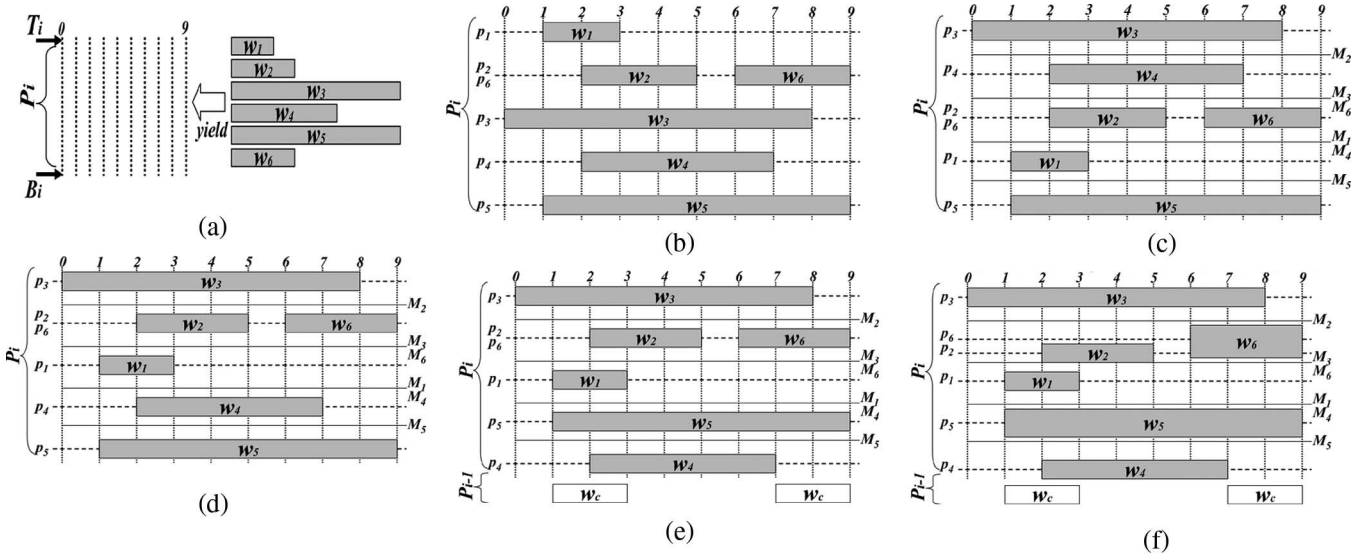

Fig. 7. TROY example. (a) Tracking routing problem. (b) One feasible routing solution can be obtained by interval packing algorithm. (c) One optimal solution from the MHP without taking an expected detailed wirelength into account. (d) Another optimal solution from the MHP with an expected detailed wirelength considered. (e) Another optimal solution from the MHP when the boundary interaction is considered. (f) Final solution after wires are sized and spaced for yield by the SOCP.

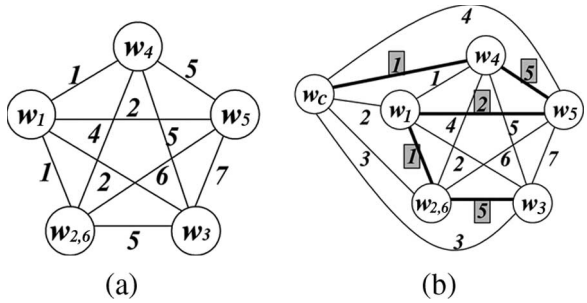

Fig. 8. Clique for wire ordering in track routing. (a) Clique for Fig. 6(b)–(d), where the edge cost is computed by (13). (b) Clique for Fig. 6(e) after adding a new vertex W_c to take the boundary interaction into account.

$$
\begin{aligned}
\textbf{min} \quad & \alpha \sum_i \{\delta_i + (1-\frac{b}{a})d_i\} + (1-\alpha)\sum_{i,j>i}\gamma_{ij} \\
\textbf{s.t.} \quad & |p_i - M_i| \le d_i && \forall i \\
& S_{min} \le s_{ij} = p_i - p_j - \tfrac{w_i+w_j}{2} && \forall o_{ij}=1, \forall j \in n_i \\
& l_{ij} W_{min} \le s_{ij}\gamma_{ij} && \forall i, \forall j \in n_i \\
& L_i S_{min} \le w_i \delta_i && \forall i \\
& B_k + \tfrac{w_i}{2} \le p_i \le T_k - \tfrac{w_i}{2} && \forall i \in P_k \\
& W_{min} \le w_i \le W_{max} && \forall i
\end{aligned}
$$

Fig. 9. After wire ordering is done, the INLP formulation in Fig. 4 can be cast into highly efficient SOCP.

has two drawbacks regarding yield, which we further address in TROY.

1) The possible detailed wirelength increase due to deviation from the preferred location (see Section II-B) is not considered, which, in turn, increases the density of random defects.
2) The interaction between adjacent panels is ignored. As short defects can occur on the boundary of adjacent panels, it is required to take this into account.

We observe that there can be multiple optimal MHP solutions, as the distribution of edge weights is rather narrow. Thus, we need to find the minimum deviation solution estimated by $\sum_i |p_i - M_i|$ among all the optimal MHP solutions. We call our modified MHP the *preference-aware minimum Hamiltonian path*. For example, although Fig. 7(c) and (d) shows the MHPs of Fig. 8(a) (the same overlapped wirelength), one can recognize that Fig. 7(d) shows less deviation from the preferred positions ($\sum_i |p_i - M_i|$), which can result in a shorter expected detailed wirelength as well as less random defects.

We further improve our wire ordering by considering the contour of the adjacent panel. Consider the example in Fig. 7(e), where W_c are the wires from a panel P_{k-1}, assuming that the wires in P_{k-1} are already ordered. Fig. 7(e) shows a better wire ordering than Fig. 7(d) when the interaction between P_k and P_{k-1} is considered. This can be done with a new clique in Fig. 8(b), where W_c is added and set as a starting vertex, and the bold lines indicate the pMHP. The edge weights between W_c and other vertices can be computed with (13) as well. When all the panels on a layer are finished with wire ordering, the wires on the layer will be sized and spaced as in Section IV-C.

C. Globally Optimal Wire Sizing and Spacing

After wires in every panel on a layer are ordered, the formulation in Fig. 4 can be further deduced to the formulation in Fig. 9 after plugging in (11) and (12), filling all the integer variables (o_{ij}) with the corresponding values, and eliminating constant terms from the objective. Auxiliary variables γ_{ij} and δ_i are introduced to translate the nonlinear objective terms into the rotated conic constraints of (7), which enable the SOCP [23]–[28]. In detail, we first set $o_{ij} = 1$ if $p_i > p_j$, and $o_{ij} = 0$ otherwise, based on the given wire ordering, which will eliminate half of the minimum spacing constraints. Then, we

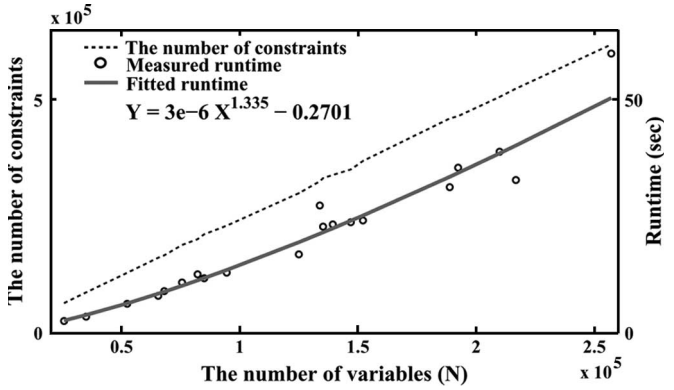

Fig. 10. Empirical runtime complexity of our SOCP is $O(N^{1.335})$, where N is the number of variables. Such near-linear complexity makes TROY to VLSI track routing.

can rewrite the objective function in Fig. 4 as follows by setting $S_{\min} = W_{\min}$ and taking off all the constant terms:

$$\alpha \sum_i \left\{ \frac{L_i S_{\min}}{w_i} + \left(1 - \frac{b}{a}\right) d_i \right\} + (1-\alpha) \sum_{i,j>i} \frac{l_{ij} W_{\min}}{s_{ij}}. \quad (14)$$

With this simplified objective function, we can introduce auxiliary variables γ_{ij} and δ_i to define the upper bounds of two key terms as follows:

$$\frac{l_{ij} W_{\min}}{s_{ij}} \le \gamma_{ij} \quad (15)$$

$$\frac{L_i S_{\min}}{w_i} \le \delta_i. \quad (16)$$

Hence, (14) can be minimized by suppressing the upper bounds [γ_{ij} in (15) and δ_i in (16)], which is essentially the objective function in Fig. 9. Equations (15) and (16) can be further cast into the rotated quadratic cone in (7). For example, since $s_{ij} > 0$, $l_{ij} > 0$, and $W_{\min} > 0$, (15) is equivalent to

$$2 s_{ij} \gamma_{ij} \ge c_{ij}^2, \quad c_{ij} = \sqrt{(2 l_{ij} W_{\min})} \quad (17)$$

where c_{ij} is a known value, as the wire order defines l_{ij}. Since (17) is in the form of $2x_1x_2 \ge \|x_3\|^2$, $x_1 \ge 0$, $x_2 \ge 0$, it describes a rotated quadratic cone in (7). The same transformation can be done for (16). Then, the formulation in Fig. 9 can be solved optimally and efficiently by the primal–dual interior-point method with $O(N^{1.3})$ bound, where N is the number of variables [34], [35]; thus, the solution will provide the optimal wire sizing and spacing for maximum yield.

Fig. 10 shows the empirical runtime complexity of our SOCP formulation in TROY. The number of constraints is linearly proportional to the number of variables. Hence, even with a larger circuit, the number of constraints will not explode, and the problem size will be tractable. When we perform curve fitting to the measured runtime samples, it has $O(N^{1.335})$, where N is the number of variables.

The optimal wire sizing and spacing for an entire layer by the SOCP can find the optimal tradeoff between open and short defects in terms of yield. Thus, TROY is far superior to traditional local or iterative approaches. Fig. 7(f) shows a track routing solution after wire sizing and spacing are done by the SOCP. Intuitively, the longer overlapped wirelength l_{ij} between two adjacent wires needs wider spacing to minimize POF_{ij}^s. Meanwhile, the spacing has to be larger than minimum spacing ($S_{\min}$) at least, and all the wires should be posed within the corresponding panel. If the wire has enough space around it, the wire width will be increased to minimize POF_i^o. Although this example shows a case for one panel, wire sizing and spacing will be performed for all the wires in a layer. In practice, the wire width may be discrete. For this case, we can change the continuous wire width found from the SOCP to the closest discrete wire width that does not violate the minimum wire spacing rule.

For some designs, aggressive wire sizing can cause routing congestion for local wires by leaving insufficient spaces. The seriousness of this issue can be different in different layers, as lower layers tend to be more crowded by local wires. This issue can be overcome in TROY by adjusting the α parameter in the objective function in Fig. 9. By applying a smaller value, TROY will search for the solution with higher weight on open defect optimization, which will increase the spacing between wires for local wires. Therefore, in higher layers, aggressively configured TROY can be applied, whereas a conservative approach can be taken in lower layers.

D. Runtime Complexity Analysis

As TROY consists of two steps, we will analyze the runtime complexity of each step.

1) *Wire ordering*. The main bottleneck in wire ordering is to find an MHP. However, we can regard the time complexity of finding an MHP as constant, as the number of maximum wires in a panel is fixed by a global routing cell size. Hence, the complexity of each MHP instance does not scale according to the design size. Let L and C denote the number of layers and the number of cells (the chip area), respectively. Then, the runtime complexity of wire ordering is $O(LC)$.
2) *Wire sizing/spacing*. The complexity of the SOCP is shown as $O(N^{1.335})$ in Fig. 10. However, since the relationship between the number of variables and the number of cells is not clear, we empirically measure the runtime complexity of the SOCP w.r.t. the number of cells. As shown in Fig. 11, it has $O(C^{1.276})$, where C is the number of cells. Since the SOCP needs to be solved for each layer, the runtime complexity of wire size/spacing is $O(LC^{1.276})$.

Therefore, the overall runtime complexity of TROY can be shown as $O(LC^{1.276})$ based on our analysis, which can be fast enough for a VLSI design.

V. Experimental Results

We implement TROY in C++. The initial global routing results are generated from the publicly available BoxRouter binary [39]. All the experiments are performed on a 3.0-GHz Pentium machine with 1-GB RAM. A solver in [36] and [40] is properly modified to find the pMHP for wire ordering in Section IV-B, and MOSEK 4.0 [24] is used to solve the SOCP

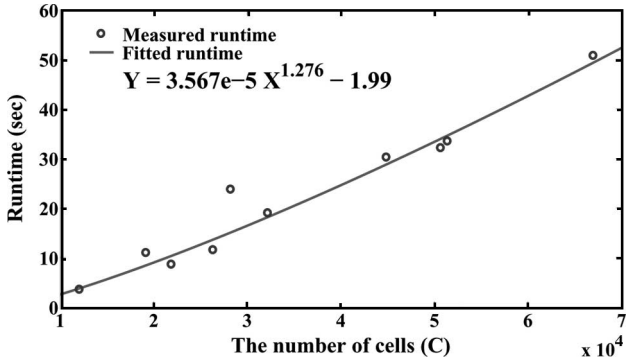

Fig. 11. Average empirical runtime complexity of our SOCP for one layer is $O(C^{1.276})$, where C is the number of cells.

TABLE II
ISPD98 IBM BENCHMARKS

circuit				global routing result		
name	cells	nets	wires	grids	g.s[a](μm^2)	wlen[b](μm)
ibm01	12036	11507	35K	64x64	26.9	135172
ibm02	19062	18429	68K	80x64	119.7	1049388
ibm03	21924	21621	56K	80x64	96.0	899694
ibm04	26346	26163	73K	96x64	73.6	714612
ibm05	28146	27777	110K	128x64	423.4	4917756
ibm06	32185	33354	112K	128x64	105.6	1693308
ibm07	44848	44394	141K	192x64	121.0	2652531
ibm08	50691	47944	160K	192x64	107.5	2456022
ibm09	51461	50393	156K	256x64	62.7	2094085
ibm10	66948	64227	216K	256x64	172.8	4701936

[a] global routing cell size
[b] global wirelength

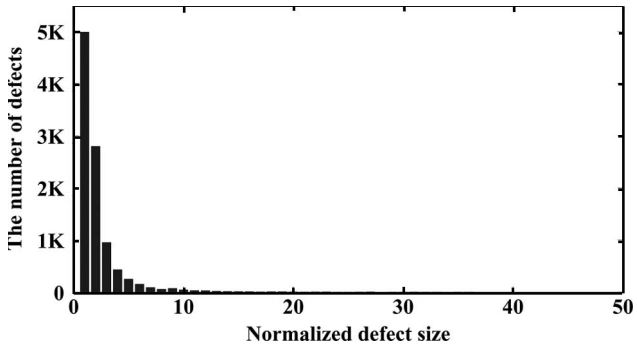

Fig. 12. Distribution of 10 000 defects for Monte Carlo simulation.

for wire sizing and spacing in Section IV-C. Since the ISPD98 IBM Corporation benchmarks lack technology information, we assume the 0.13-μm technology to use the defect size distribution parameter in [7] and set $S_{\min} = W_{\min} = 0.2$ μm. We further assume that $W_{\max} = 0.4$ μm, and 0.2, 0.3, and 0.4 μm are the only allowed wire widths. Table II shows the detail for each benchmark circuit. Since the benchmarks lack detailed pin locations, one to five pins for each global routing cell on each wire are randomly generated to define the preferred position of each wire (M_i). A Monte Carlo simulation [41] with 10 000 random defects based on (1) is performed to estimate yield loss. Also, these random defects are assumed to be uniformly distributed on the chip for fair estimation. Fig. 12 shows our defect distribution.

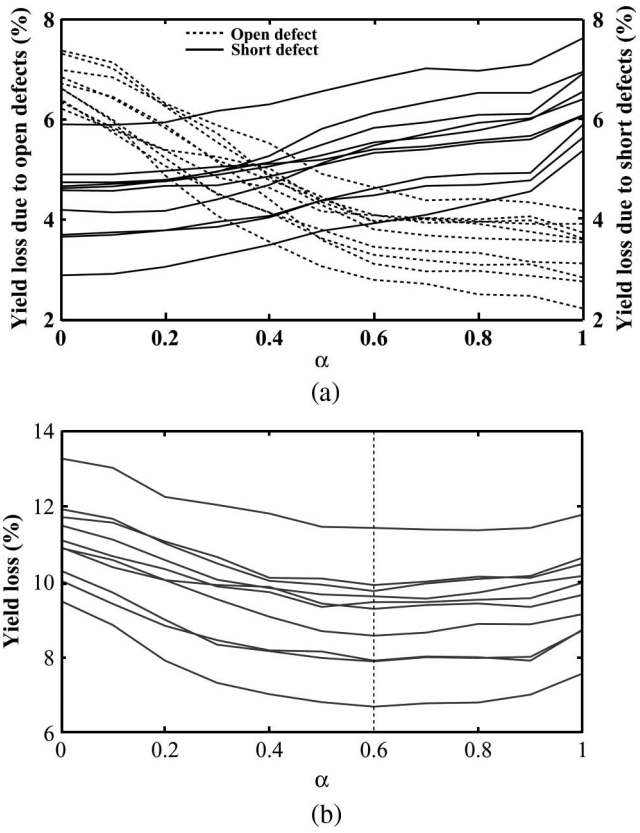

Fig. 13. Tradeoff between open and short defects is shown by α. (a) With larger α, the number of open defects decreases, whereas the number of short defects increases. (b) Minimum yield loss due to random defects (open and short defects) can be achieved around $\alpha = 0.6$.

We explore the tradeoff between random defects controlled by α in Section III, assuming a continuous wire width. In Fig. 13(a), yield loss changes due to open and short defects by different α values are plotted for all the benchmark circuits in Table II. Overall, with larger α, yield loss due to open defects decreases at a cost of more short defects. Fig. 13(b) shows total yield loss (short + open defects) by different α and indicates that the minimum yield loss can be obtained around $\alpha = 0.6$. Whether the open defects are dominant over the short defects or not is still controversial [1], [3], [7], [42], but our result $(\alpha = 0.6)$ shows that both are similarly important. We set $\alpha = 0.6$ for all the experiments in the rest of the section.

For comparison, we implement a greedy algorithm similar to [17], the only track routing algorithm, to our best knowledge, that can handle arbitrary wire spacing. As the original algorithm in [17] optimizes crosstalk and timing without wire sizing, we add a wire sizing feature along with a wire spacing functionality. We also modify the optimization objective such that its wire ordering and wire sizing/spacing greedily seek for the minimization of POF^o and POF^s. Let those greedy wire ordering and wire sizing/spacing be denoted by **g.wo** and **g.wss**, and MHP-based wire ordering and SOCP-based wire sizing/spacing be denoted by **h.wo** and **s.wss**, respectively.

Table III investigates the effect of two main techniques in TROY by pairing each technique with a greedy algorithm ($\mathbf{h.wo} + \mathbf{g.wss}$ and $\mathbf{g.wo} + \mathbf{s.wss}$), and compares TROY

TABLE III
COMPARISON BETWEEN THE GREEDY TRACK ROUTER AND TROY ($\alpha = 0.6$)

algorithm	evaluation		ibm01	ibm02	ibm03	ibm04	ibm05	ibm06	ibm07	ibm08	ibm09	ibm10	sum	ratio
g.wo[a] + g.wss[b] (d.wd[e])	defects[g]	open	527	660	666	692	568	776	587	691	681	599	6447	1.00
		short	521	513	457	527	406	507	453	481	539	456	4860	1.00
		total	1048	1173	1123	1219	974	1283	1040	1172	1220	1055	11307	1.00
	wlen inc[h] (μm)		19606.6	97720.2	69128.0	86668.5	290296.0	139420.2	224507.6	201960.8	168111.3	44649.2	1642068.3	1.00
	cpu (sec)		1	1	1	1	2	2	4	4	5	7	28	1.00
h.wo[c] + g.wss (d.wd)	defects	open	414	482	484	515	391	612	393	518	470	387	4666	0.72
		short	635	609	571	662	538	646	541	634	666	548	6050	1.24
		total	1049	1091	1055	1177	929	1258	934	1152	1136	935	10716	0.95
	wlen inc (μm)		19626.2	95338.9	69544.6	88901.8	287752	140344.5	224130.2	206126.6	171498.7	346590.2	1649853.7	1.00
	cpu (sec)		7	64	30	19	164	38	59	35	29	93	538	19.21
g.wo + s.wss[d] (d.wd)	defects	open	454	584	523	567	388	657	506	541	581	504	5305	0.82
		short	518	490	455	525	341	582	430	505	535	439	4820	0.99
		total	972	1074	978	1092	729	1239	936	1046	1116	943	10125	0.90
	wlen inc (μm)		19052.9	96149.4	68154.6	86960.4	286044.9	138334.9	218944.2	200589.4	165633.8	337722.7	1617587.2	0.99
	cpu (sec)		10	29	23	33	58	55	76	83	86	125	579	20.68
TROY (d.wd)	defects	open	561	511	519	618	389	671	459	562	551	441	5282	0.82
		short	320	445	447	399	293	491	351	425	423	376	3970	0.82
		total	881	956	966	1017	682	1162	810	987	974	817	9252	0.82
	wlen inc (μm)		19119.7	94747.6	68808.3	88547.1	292148.4	139782.8	221714.3	205223.8	169543.3	344612.4	1644247.7	1.00
	cpu (sec)		13	90	50	50	217	87	131	117	110	214	1079	38.54
TROY (c.wd[f])	defects	open	547	539	532	612	391	679	461	582	553	447	5343	0.83
		short	310	407	396	379	278	463	328	393	408	344	3706	0.76
		total	857	946	928	991	669	1142	789	975	961	791	9049	0.80
	wlen inc (μm)		19119.7	94747.6	68808.3	88547.1	292148.4	139782.8	221714.2	205223.8	169543.2	344612.4	1644247.5	1.00
	cpu (sec)		13	90	51	50	218	86	136	116	108	219	1087	38.82

[a] greedy wire ordering
[b] iterative greedy wire sizing/spacing
[c] preference-aware minimum Hamiltonian path based wire ordering in Section IV-B
[d] second order cone programming based wire sizing/spacing in Section IV-C
[e] discrete wire width (W_{min}, $1.5W_{min}$, and $2W_{min}$ are allowed.)
[f] continuous wire width (any wire width between W_{min} and $2W_{min}$ are allowed.)
[g] estimated based on Monte-Carlo simulation [41] with 10K random defects
[h] expected detailed wirelength (the summation of the deviation of each wire from its preferred location)

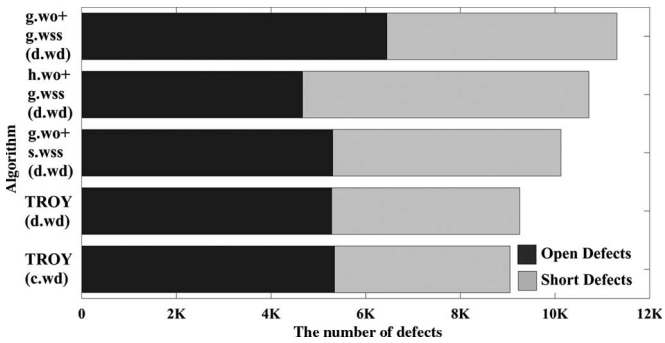

Fig. 14. Total number of defects from all the benchmark circuits is shown by different algorithms. Each step of TROY (MHP-based wire ordering and SOCP-based wire sizing/spacing) is superior to the counterpart in the greedy approach, and TROY can achieve 18% yield loss reduction compared with the greedy yield-driven track router.

($\alpha = 0.6$) with a greedy yield-driven track router ($\mathbf{g.wo} + \mathbf{g.wss}$). Note that there are two experiments on TROY—one in a continuous wire width (**c.w**) and the other one in a discrete wire width (**d.w**). First, we observe that $\mathbf{g.wo} + \mathbf{s.wss}$ reduces yield loss by 10% on average, whereas $\mathbf{h.wo} + \mathbf{g.wss}$ has only 5% improvement, compared with the bottom line ($\mathbf{g.wo} + \mathbf{g.wss}$). This implies that $\mathbf{s.wss}$ is more effective than $\mathbf{h.wo}$ mainly due to two reasons: 1) POF^o and POF^s are highly sensitive to wire sizing/spacing, as shown in (4) and (5); and 2) $\mathbf{g.wss}$ fails to achieve a decent tradeoff between random defects by nature, resulting in a highly biased solution as shown in the $\mathbf{h.wo} + \mathbf{g.wss}$ row of Table III. It also clearly shows that TROY, which is, in fact, $\mathbf{h.wo} + \mathbf{s.wss}$ in a discrete wire width, can significantly reduce yield loss by 18% on average, and it can be even over 30% for ibm05, compared with the greedy approach ($\mathbf{g.wo} + \mathbf{g.wss}$) in a discrete wire width. Also, the total number of open and short defects is consistently reduced. The discrete wire width incurs only 2.2% more yield loss on average when TROY in **d.w** and in **c.w** are compared. Fig. 14 summarizes the key results of Table III.

Although the runtime becomes longer, all test cases can be finished within a few seconds/minutes. More importantly, TROY has near-linear runtime complexity, as discussed in Section IV-D. Therefore, it should be applicable to a VLSI design.

VI. CONCLUSION

In nanometer designs, routing becomes a key optimization phase for yield. To cope with yield loss due to random defects in the advanced technology, we present TROY, an efficient yield-driven track router. With effective wire ordering and wire sizing/spacing optimization based on the MHP and the SOCP, experimental results show that TROY significantly reduces yield loss. As TROY finds globally optimal wire sizing and spacing for a given wire order, it may be easily modified for any wire-sizing- and spacing-related optimization such as crosstalk and timing [43].

REFERENCES

[1] I. Koren, "Should yield be a design objective?" in *Proc. Int. Symp. Quality Electron. Des.*, Mar. 2000, pp. 115–120.

[2] W. Maly, H. Heineken, J. Khare, and P. K. Nag, "Design for manufacturability in submicron domain," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 1996, pp. 690–697.

[3] A. B. Kahng, B. Liu, and I. I. Mandoiu, "Non-tree routing for reliability and yield improvement," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2002, pp. 260–266.

[4] Y. Bourai and C.-J. R. Shi, "Layout compaction for yield optimization via critical area minimization," in *Proc. Des. Autom. Test Eur.*, Mar. 2000, pp. 122–127.

[5] 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.

[6] S. 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.

[7] P. Cristie and J. P. de Gyvez, "Prelayout interconnect yield prediction," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 1, pp. 55–59, Feb. 2003.

[8] D. Wu, J. Hu, M. Zhao, and R. Mahapatra, "Timing driven track routing considering coupling capacitance," in *Proc. Asia South Pac. Des. Autom. Conf.*, Jan. 2005, pp. 1156–1159.

[9] S.-Y. Kuo, "YOR: A yield-optimizing routing algorithm by minimizing critical areas and vias," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 12, no. 9, pp. 1303–1311, Sep. 1993.

[10] A. Pitaksanonku, S. Thanawastien, C. Lursinsap, and J. Gandhi, "DTR: A defect-tolerant routing algorithm," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 1989, pp. 795–798.

[11] V. K. I. Chiluvuri and I. Koren, "Layout-synthesis techniques for yield enhancement," *IEEE Trans. Semicond. Manuf.*, vol. 8, no. 2, pp. 178–187, May 1995.

[12] C. Bamji and E. Malavasi, "Enhanced network flow algorithm for yield optimization," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 1996, pp. 746–751.

[13] J. Z. Su and W. Dai, "Post route optimization for improved yield using a rubber-band wiring model," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 1997, pp. 700–706.

[14] G. A. Allan, "Targeted layout modifications for semiconductor yield/reliability enhancement," *IEEE Trans. Semicond. Manuf.*, vol. 17, no. 4, pp. 573–581, Nov. 2004.

[15] E. P. Huijbregtz, H. Xue, and J. A. Jess, "Routing for reliable manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 8, no. 2, pp. 188–194, May 1995.

[16] D. Muller, "Optimizing yield in global routing," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 2006, pp. 480–486.

[17] H.-P. Tseng, L. Scheffer, and C. Sechen, "Timing- and crosstalk-driven area routing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 4, pp. 528–544, Apr. 2001.

[18] R. Kay and R. A. Rutenbar, "Wire packing: A strong formulation of crosstalk-aware chip-level track/layer assignment with an efficient integer programming solution," in *Proc. Int. Symp. Phys. Des.*, 2000, pp. 61–68.

[19] H. Chen, C. Qiao, F. Zhou, and C.-K. Cheng, "Refined single trunk tree: A rectilinear Steiner tree generator for interconnect prediction," in *Proc. Int. Workshop Syst.-Level Interconnect Prediction*, Apr. 2002, pp. 85–89.

[20] W. Maly, "Modeling of lithography related yield losses for CAD of VLSI circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. CAD-4, no. 3, pp. 166–177, Jul. 1985.

[21] R. Glang, "Defect size distribution in VLSI chips," *IEEE Trans. Semicond. Manuf.*, vol. 4, no. 4, pp. 265–269, Nov. 1991.

[22] T. Iizuka, M. Ikeda, and K. Asada, "Exact wiring fault minimization via comprehensive layout synthesis for CMOS logic cells," in *Proc. Int. Symp. Quality Electron. Des.*, Mar. 2004, pp. 377–380.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[24] [Online]. Available: http://www.mosek.com

[25] M. S. Lobo, L. Vandenverghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra Appl.*, vol. 248, no. 1, pp. 193–228, Nov. 1998.

[26] E. Andersen, C. Roos, and T. Terlaky, "On implementing a primal-dual interior-point method for conic quadratic optimization," *Math. Program.*, vol. 95, no. 2, pp. 249–277, 2003.

[27] F. Alizadeh and D. Goldfarb, "Second-order cone programming," *Math. Program. Ser. B*, vol. 95, no. 1, pp. 3–51, 2003.

[28] H. D. Mittelmann, "An independent benchmarking of SDP and SOCP solvers," *Math. Program. Ser. B*, vol. 95, no. 2, pp. 407–430, 2003.

[29] S. J. Wright, *Primal–Dual Interior-Point Methods*. Philadelphia, PA: SIAM, 1997.

[30] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA: SIAM, 1994.

[31] F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization," *SIAM J. Optim.*, vol. 5, no. 1, pp. 13–51, 1995.

[32] Y. E. Nesterov and M. J. Todd, "Self-scaled barriers and interior-point methods for convex programming," *Math. Oper. Res.*, vol. 22, no. 1, pp. 1–42, 1997.

[33] I. W.-H. Tsang and J. T.-Y. Kwok, "Efficient hyperkernel learning using second-order cone programming," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 48–58, Jan. 2006.

[34] T.-Y. Ling, I.-J. Lin, and Y.-W. Chang, "Statistical circuit optimization considering device and interconnect process variations," in *Proc. Int. Workshop Syst.-Level Interconnect Prediction*, Mar. 2007, pp. 47–54.

[35] M. Mani, A. Devgan, and M. Orshansk, "An efficient algorithm for statistical minimization of total power under timing yield constraints," in *Proc. ACM/IEEE Des. Autom. Conf.*, Jun. 2005, pp. 309–314.

[36] K. Helsgaun, "An effective implementation of the Lin–Kernighan traveling salesman heuristic," *Eur. J. Oper. Res.*, vol. 126, no. 1, pp. 106–130, Jan. 2000.

[37] T. Yoshimura and E. Kuh, "Efficient algorithms for channel routing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. CAD-1, no. 1, pp. 25–35, Jan. 1982.

[38] U. I. Gupta, D. T. Lee, and J. Y.-T. Leung, "An optimal solution for the channel-assignment problem," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. CAD-28, no. 11, pp. 807–810, Nov. 1979.

[39] [Online]. Available: http://www.cerc.utexas.edu/utda

[40] [Online]. Available: http://www.akira.ruc.dk/~keld/research/LKH

[41] D. Walkter, *Yield Simulation for Integrated Circuits*. Norwell, MA: Kluwer, 1987.

[42] J. P. de Gyvez, "Yield modeling and BEOL fundamentals," in *Proc. Int. Workshop Syst.-Level Interconnect Prediction*, Mar. 2001, pp. 135–163.

[43] H. K.-S. Leung, "Advanced routing in changing technology landscape," in *Proc. Int. Symp. Phys. Des.*, Apr. 2003, pp. 118–121.

Minsik Cho received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 1999, and the M.S. degree in electrical and computer engineering from the University of Wisconsin, Madison, in 2004. He is currently working toward the Ph.D. degree in electrical and computer engineering at The University of Texas, Austin.

He was with Intel during the summer of 2005 and with IBM Corporation T. J. Watson Research, Yorktown Heights, NY, during the summers of 2006 and 2007. His current research interests include the manufacturability-driven routing algorithm and the variation-tolerant VLSI physical design.

Mr. Cho was the recipient of the Korean Information Technology Scholarship in 2002, the IBM Ph.D. Scholarship in 2007, the ISPD routing contest Awards in 2007, the SRC Inventor Recognition Award in 2008, and the Best Paper Award Nominations at the Asian and South Pacific Design Automation Conference and at the Design Automation Conference in 2006.

Hua Xiang received the B.S. and M.S. degrees in computer science from Peking University, Beijing, China, in 1997 and 2000, respectively. She received another M.S. degree from The University of Texas, Austin, in 2002, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, Urbana, in 2004.

She was previously a Senior Member of the Technical Staff with Cadence Design Systems. She is currently a Research Staff Member with the IBM Corporation T. J. Watson Research Center, Yorktown Heights, NY. Her main research interest is computer-aided design for VLSI, primarily in the area of physical design, including routing, buffer planning, floor planning and placement, and DFM.

Ruchir Puri (F'07) received the B.S. degree in electronics and communication engineering from the Regional Engineering College, National Institute of Technology, Kurukshetra, India, in 1988, the M.Tech. degree in electrical engineering from Indian Institute of Technology, Kanpur, India, in 1990, and the Ph.D. degree in electrical and computer engineering from the University of Calgary, Alberta, Canada, in 1994.

He was an Adjunct Professor with the Department of Electrical Engineering, Columbia University, New York, NY, where he taught VLSI design and circuits. He joined the VLSI Design Automation group at the IBM Corporation Thomas J. Watson Research Center, Yorktown Heights, NY, in 1995, where he manages a research group focused on logic and physical synthesis. He has been working on design and automated synthesis solutions for IBM's high-performance and power-efficient microprocessors and ASICs in advanced CMOS technologies. He is the holder of 22 U.S. patents and is the author of over 80 publications on the design and synthesis of low-power and high-performance circuits.

Dr. Puri is currently an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I and has served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II. He was the recipient of the Association for Computing Machinery (ACM)/IEEE Design Automation Scholarship in 1993 and the Alberta Microelectronics research scholarships in 1992 and 1993 for his doctoral research. He was also the recipient of many IBM awards for his work, including two Outstanding Technical Achievement awards and the IBM Execute Now award. He has served on the program committees of most major VLSI Design Automation conferences and National Science Foundation and Semiconductor Research Corporation panels. He has been an invited speaker at numerous conferences such as the International Solid State Circuit Conference, the Design Automation Conference, and the International Conference on CAD. He is currently an ACM Distinguished Speaker and is a member of the ACM Special Interest Group on Design Automation Physical Design Technical Committee as well as of the Logic Synthesis Technical Committee.

David Z. Pan (S'97–M'00–SM'06) received the Ph.D. degree in computer science from the University of California, Los Angeles (UCLA), in 2000.

He was a Research Staff Member with the IBM Corporation T. J. Watson Research Center, Yorktown Heights, NY, from 2000 to 2003. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, The University of Texas, Austin. He is in the Design Technology Working Group of the International Technology Roadmap for Semiconductor. He has published over 65 technical papers and is the holder of five U.S. patents. His research interests include nanometer physical designs, designs for manufacturing, low power, vertical integration design and technology, and CAD for emerging technologies.

Prof. Pan is a member of the Association for Computing Machinery (ACM)/Special Interest Group on Design Automation (SIGDA) Technical Committee on Physical Design. He is an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, IEEE TRANSACTIONS ON VLSI SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, and IEEE CAS Society Newsletter. He has served in the technical program committees of major VLSI/CAD conferences, including the International Conference on Computer-Aided Design, the International Symposium on Physical Design (ISPD), the Design Automation and Test in Europe Conference, the Asia and South Pacific Design Automation Conference (ASPDAC), the International Symposium on Quality Electronic Design (ISQED), the International Symposium on Circuits and Systems (ISCAS), the International Workshop on System-Level Interconnect Prediction, the Great Lake Symposium on VLSI Design, and the International Conference on Integrated Circuit Design & Technology. He is the IEEE COMPUTER-AIDED NETWORK DESIGN Workshop Chair in 2007, the Program Committee Chair of ISPD in 2007, the CAD track Co-Chair of ISCAS in 2006 and 2007, and the Design of Reliable Circuits and Systems Track Chair of ISQED in 2007. He was the recipient of the Best Paper in Session Award from the Semiconductor Research Corporation (SRC) Techcon in 1998, the IBM Research Fellowship in 1999, the Dimitris Chorafas Foundation Award and the SRC Inventor Recognition Award in 2000, the Outstanding Ph.D. Award from UCLA in 2001, the IBM Research Bravo Award in 2003, the IBM Faculty Award from 2004 to 2006, the ACM/SIGDA Outstanding New Faculty Award in 2005, the Best Paper Award Nominations at Design Automation Conference and at ASPDAC in 2006, and the ISPD Routing Contest Awards (second place in 3-D and third place in 2-D) and the National Science Foundation CAREER Award in 2007.