# A Multilevel Analytical Placement for 3D ICs

Jason Cong[1,2] and Guojie Luo[1]

[1]Computer Science Department
University of California, Los Angeles

[2]California NanoSystems Institute

Los Angeles, CA 90095, USA
Tel : 1 (310) 206-2775
Fax : 1 (310) 825-2273
e-mail : {cong, gluo}@cs.ucla.edu

**Abstract - In this paper we propose a multilevel non-linear programming based 3D placement approach that minimizes a weighted sum of total wirelength and TS via number subject to area density constraints. This approach relaxes the discrete layer assignments so that they are continuous in the z-direction and the problem can be solved by an analytical global placer. A key idea is to do the overlap removal and device layer assignment simultaneously by adding a density penalty function for both area & TS via density constraints. Experimental results show that this analytical placer in a multilevel framework is effective to achieve trade-offs between wirelength and TS via number. Compared to the recently published transformation-based 3D placement method [1], we are able to achieve on average 12% shorter wirelength and 29% fewer TS via compared to their cases with best wirelength; we are also able to achieve on average 20% shorter wirelength and 50% fewer TS via number compared to their cases with best TS via numbers.**

## I. Introduction

Three-dimensional (3D) IC technologies can offer the potential to significantly reduce interconnect delays and improve system performance. Furthermore, the shortened wirelength, especially that of the clock net, also lessens the power consumption of the circuit. 3D IC technologies also provide a flexible way to carry out the heterogeneous system-on-chip (SoC) design by integrating disparate technologies, such as memory and logic circuits, radio frequency (RF) and mixed signal components, optoelectronic devices, etc., onto different layers of a 3D IC.

Device layers in a 3D IC are connected using through-silicon vias (TS via). However, TS vias are usually etched or drilled through device layers by special techniques and are costly to fabricate. A large number of the TS vias will increase the area overhead and the cost of the final chip. Also, under the current technologies, TS via pitches are very large compared to the sizes of regular metal wires, usually around 5-10μm. In 3D IC structures, TS vias are usually placed at the whitespace between the macro blocks or cells, so the number of TS vias will not only affect the routing resource but also affect the overall chip or package areas. Therefore, the number of TS vias in the circuit is constrained and needs to be controlled during physical design.

In recent years, 3D IC physical design attracts more and more attention. Along with the technology updates, there are several published works targeted on the 3D placement problem. A thermal-driven force-directed 3D placement method [2] was proposed, where the temperature profile is interpreted as thermal forces to guide the cell placement. In their work, a 3D force-directed placement engine is used to place each cell in a true 3D space, with z position being a real number. Rounding is needed for layer assignment, which may introduce rounding errors. A folding/stacking based 3D placement [1] is proposed to reuse the 2D placement results and perform device layer assignment and other optimizations. A partition-based approach [3] was also applied to the 3D context, where the temperature and TS via counts and thermal effect are modeled in the min-cut objective together with the total wirelength. Although it is a convenient way to consider these effects with this partition-based method, recent comparative analysis suggests that partition-based methods are not as competitive as analytical methods for modern 2D placement problems [4]. A quadratic programming approach [5] for 3D placement was also proposed, which finds an overlap-free placement by modeling the cell distribution with a discrete cosine transformation based cost function. It shares the same problems as [2], which only distribute the cells in a cuboid space but not into layers.

All these techniques try to explore the trade-off among wirelength, TS via number and temperature. But the quality of the solution may be sub-optimal in cases. The goal of our work is to first develop a high-quality solver for the simple 3D placement problem with objective of wirelength and TS via number, so that it can be used as a basic engine to consider other constraints and/objectives in 3D placement.

In particular, we develop a 3D placement approach using a nonlinear optimization method to handle the 3D global placement. The main idea is to do the overlap removal and device layer assignment simultaneously by adding a density penalty function for both area & TS via density constraints. The minimization of this density penalty function have the tendency to achieve overlap-free condition in *(x,y)*-direction and also legal device layer assignment in *z*-direction. The TS via number is also considered by adding TS via number penalty function to the objective. The contributions of our work are listed as following:

- A novel density penalty function is proposed to do the cell distribution in a 3D placement region. The minimization of this function would result in a close-to-legal 3D global placement. A formal proof is also given. Our method relieves the rounding problem of previous analytical 3D placement methods, and thus reduces the effort for the detailed placement phase.
- We observe that multilevel scheme is effective to control

TS via number, which provides extra TS via number reduction to the weighting factor.

- Experiments are performed to compare with [1] on the trade-off curves of wirelength and TS via number. The experimental results show that our method outperforms theirs by achieving on average 12% shorter wirelength and 29% fewer TS vias compared to their cases with best wirelength; we are also able to achieve on average 20% shorter wirelength and 50% fewer TS via number compared to their cases with best TS via numbers.

The remainder of this paper is organized as follows. Section II formulates the 3D placement problem and describes the placement flow. Section III introduces our 3D analytical placement engine and gives formal proofs for the equivalence between the minimizer of our density penalty function and a legal solution. Section IV describes the multilevel framework with the analytical placement engine. Section V presents the experimental results to demonstrate the effect of our method on the trade-offs between wirelength and TS via number. Finally, section VI concludes the paper and discusses about the future work.

## II. Problem Formulation and 3D Placement Flow

### A. Problem Formulation

Given a circuit represented as a hypergraph $H=(V,E)$, the placement region $R$ (scaled to $[0,1]\times[0,1]$), and the number device layers $K$, the task of 3D placement problem is to assign every cell $v_i \in V$ a triple $(x_i, y_i, z_i)$, which indicates that this cell is placed on the device layer $z_i \in \{1,2,\ldots,K\}$ with its center at $(x_i, y_i) \in R$. The objective is to minimize the weighted sum of total wirelength and TS via number, under non-overlap constraints.

$$\begin{aligned} &\text{minimize} && \sum_{e\in E}\left(l(e)+\alpha\cdot v(e)\right) \\ &\text{subject to} && \left(\text{non-overlap constraints}\right) \end{aligned} \tag{1}$$

We use the traditional half-perimeter model for wirelength calculation, where the wirelength $l(e)$ of a net $e$ is calculated as follows:

$$l(e)=\max_{v_i,v_j\in e}\left|x_i-x_j\right|+\max_{v_i,v_j\in e}\left|y_i-y_j\right| \tag{2}$$

Because the routing information is unknown during the placement process, TS via number is also estimated through a similar model. The TS via number of a net is calculated as the height of the bounding cube of the cells belong to that net. The TS via number $v(e)$ of a net e is calculated as follows:

$$v(e)=\max_{v_i,v_j\in e}\left|z_i-z_j\right| \tag{3}$$

The weighting factor $\alpha$ is used to achieve the trade-offs between the wirelength $l(e)$ and the TS via number $v(e)$.

In modern analytical placers (e.g. those described in [6]), the non-overlap constraints are usually transformed to density constraints during global placement, and are legalized during detailed placement. The formal descriptions will be given in Section III.D.

### B. 3D Placement Flow

The overall placement flow is shown in Fig. 1. The global placement starts from scratch, or takes in the given initial placement. The global placement incorporates the analytical placement engine (Section III) into the multilevel framework that is used in [7]. The global placement is then processed layer-by-layer with the 2D detailed placer [8] to obtain the final placement.

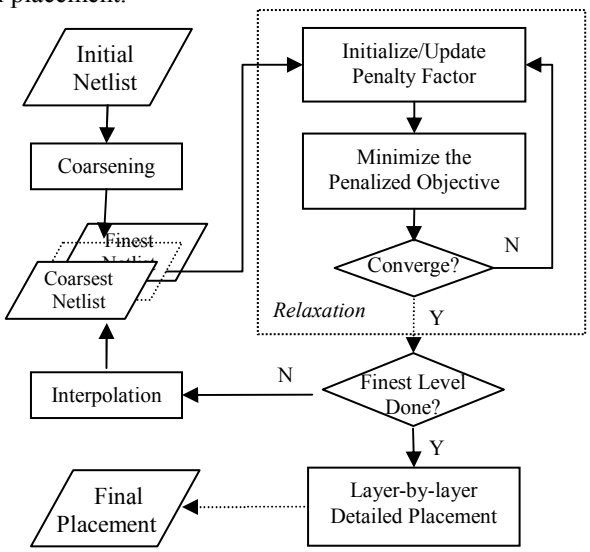


**Fig. 1** – Our 3D Placement Flow

## III. Analytical Placement Engine

The analytical placement engine solves problem (1) by transforming the non-overlap constraints to density penalties.

$$\begin{aligned} &\text{minimize} && \sum_{e\in E}\left(l(e)+\alpha\cdot v(e)\right) \\ &\text{subject to} && Penalty(\vec{x},\vec{y},\vec{z})=0 \end{aligned} \tag{4}$$

The wirelength $l(e)$ (Section III.B), the TS via number $v(e)$ (Section III.C), and the density penalty function $Penalty(\vec{x},\vec{y},\vec{z})$ (Section III.D) will be described in the following subsections in detail.

In order to solve this constrained problem, penalty methods [9] are usually applied:

$$OBJ(\vec{x},\vec{y},\vec{z})=\sum_{e\in E}\left(l(e)+\alpha\cdot v(e)\right)+\mu\cdot Penalty(\vec{x},\vec{y},\vec{z}) \tag{5}$$

This penalized objective function is minimized at each iteration, with a gradually increasing penalty factor $\mu$ to reduce the density violations. It can be shown that the minimizer of equation (5) is equivalent to problem (4) when $\mu\to\infty$ if the penalty function is non-negative.

### A. Relaxation of Discrete Variables

As mentioned in Section II.A, the placement variables are represented by triples $(x_i, y_i, z_i)$, where $z_i$ is a discrete variable in $\{1,2,...,K\}$. The range of $z_i$ is relaxed from the set $\{1,2,\ldots,K\}$ to a continuous interval $[1,K]$. After relaxation, a nonlinear analytical solver can be used in our placement engine. The relaxed solution is mapped back to the discrete values before the detailed placement phase.

### B. Log-sum-exp Wirelength

The half-perimeter wirelength $l(e)$ defined in equation (2) is replaced by a differentiable approximation with log-sum-exp function [10]:

$$l(e) \approx \eta(\log \sum_{v_i \in e} \exp(x_i/\eta) + \log \sum_{v_i \in e} \exp(-x_i/\eta) + \log \sum_{v_i \in e} \exp(y_i/\eta) + \log \sum_{v_i \in e} \exp(-y_i/\eta)) \quad (6)$$

For numerical stability, the placement region $R$ is scaled into $[0,1]\times[0,1]$, thus variables of $(x_i, y_i)$ are in the range between 0 and 1, and the parameter $\eta$ is set to 0.01 in implementation as [11].

### C. TS Via Number

The TS via number $v(e)$ estimation defined in equation (3) is also replaced by the log-sum-exp approximation:

$$v(e) \approx \eta(\log \sum_{v_i \in e} \exp(z_i/\eta) + \log \sum_{v_i \in e} \exp(-z_i/\eta)) \quad (7)$$

### D. Density Penalty Function

The density penalty function is for overlap removal in both the *(x,y)*-direction and the *z*-direction. The minimization of the density penalty function should lead to a non-overlap placement in theory.

Assume that every cell $v_i$ has a legal device layer assignment (i.e. $z_i \in \{1,2,\ldots,K\}$), then we can define $K$ density functions for these $K$ device layers. Intuitively, the density function $D_k(u,v)$ indicates the number of cells which cover the point $(u,v)$ on the *k*-th device layer. It is defined as:

$$D_k(u,v) = \sum_{i:z_i=k} d_i(u,v) \quad (8)$$

which is the sum of the density contribution $d_i(u,v)$ of cell $v_i$ assigned to this device layer at point $(u,v)$. The density contribution $d_i(u,v)$ is one inside the area occupied by $v_i$, and is zero outside this area. An example is given in Fig. 2 showing the density function with two overlapping cells.

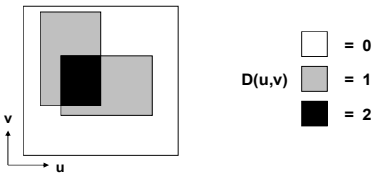


**Fig. 2** – an Example of the Density Function

During global placement, it is possible that cell $v_i$ stays between two device layers, so that the variable $z_i \in [1,K]$ is not aligned to any of two device layers. We borrow the idea from the bell-shaped function [12] to define the density function for this case:

$$D_k(u,v) = \sum_i \eta(k,z_i) d_i(u,v), \text{ for } 1 \le k \le K \quad (9)$$

where

$$\eta(k,z) = \begin{cases} 1-2(z-k)^2 & |z-k| \le 1/2 \\ 2(|z-k|-1)^2 & 1/2 < |z-k| \le 1 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

We call (10) the *bell-shaped density projection function*, which extends the density function (8) from integral layer assignments to the definition (9) for relaxed layer assignments. It is obvious that (9) is consistent with (8) when the layer assignments $\{z_i\}$ are integers.

An example of how this extension works for a 4-layer 3D placement is given in Fig. 3. The x-axis is the relaxed layer assignment in z-direction, while the y-axis indicates the amount of area to be projected in the actual device layers. The four curves colored in red, green, blue and purple represents device layer 1, 2, 3 and 4 respectively. As in this example, a cell is temporarily placed at $z = 2.316$ (yellow triangle) between layer 2 and layer 3. The bell-shaped density projection functions project 80% of its area to layer 2 (green) and 20% of its area to layer 3 (blue). In this way, we establish a mapping from a relaxed 3D placement to the area distributions in discrete layers.

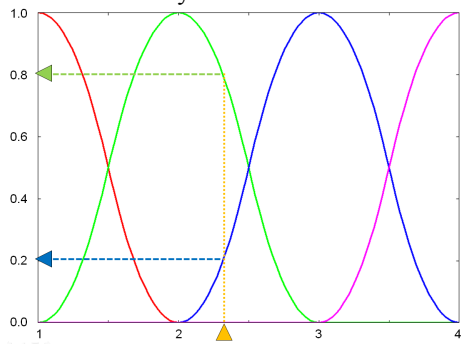


**Fig. 3** – an Example of the Bell-shaped Density Projections

Inspired by the quadratic penalty terms in 2D placement methods [11-13], we define this *density penalty function* to measure the amount of overlaps:

$$P(\vec{x},\vec{y},\vec{z}) = \sum_{k=1}^{K} \int_0^1 \int_0^1 (D_k(u,v)-1)^2 \, dudv \quad (11)$$

***Lemma 1.*** Assume the total area of cells equals the placement area (i.e. $\sum_i area(v_i) = K$, no empty space), every legal placement $(\vec{x}^*,\vec{y}^*,\vec{z}^*)$, which satisfies $D_k(u,v) = 1$ for every $k$ and $(u,v)$ without any non-integer $z_i^*$, is a minimizer of $P(\vec{x},\vec{y},\vec{z})$.

The proof of Lemma 1 is trivial and thus is omitted. Therefore, minimizing $P(\vec{x},\vec{y},\vec{z})$ provides a necessary condition for a legal placement. However, there exist minimizers that cannot form a legal placement. An example is shown in Fig. 4, where placement (b) also minimizes the density penalty function but it is not legal.

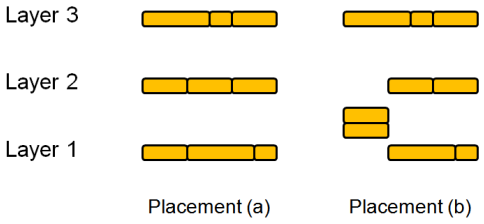


**Fig. 4** – Two Placements with the Same Density Penalties

To avoid reaching such minimizers, we introduce the interlayer density function:

$$E_k(u,v) = \sum_i \eta(k+0.5,z_i) d_i(u,v), \text{ for } 1 \le k \le K-1 \quad (12)$$

and also the interlayer density penalty function:

$$Q(\vec{x},\vec{y},\vec{z}) = \sum_{k=1}^{K-1}\int_0^1\int_0^1 \left(E_k(u,v)-1\right)^2 dudv \qquad (13)$$

Similar to the density penalty function $P(\vec{x},\vec{y},\vec{z})$, the following *Lemma 2* is also true.

***Lemma 2.*** Assume the total area of cells equals the placement area, every legal placement is a minimizer of $Q(\vec{x},\vec{y},\vec{z})$.

Combining the density penalty functions $P(\vec{x},\vec{y},\vec{z})$ and $Q(\vec{x},\vec{y},\vec{z})$, we define the following density penalty function:

$$Penalty(\vec{x},\vec{y},\vec{z}) = P(\vec{x},\vec{y},\vec{z}) + Q(\vec{x},\vec{y},\vec{z}) \qquad (14)$$

***Theorem 1.*** Assume the total area of cells equals the placement area, every legal placement $(\vec{x}^*,\vec{y}^*,\vec{z}^*)$ is a minimizer of $Penalty(\vec{x},\vec{y},\vec{z})$, and vice versa.

***Proof.*** It is obvious that every legal placement is a minimizer of $Penalty(\vec{x},\vec{y},\vec{z})$ by combining *Lemma 1* and *Lemma 2*. We shall prove that every minimizer $(\vec{x}^*,\vec{y}^*,\vec{z}^*)$ of $Penalty(\vec{x},\vec{y},\vec{z})$ is a legal placement. From the proof of *Lemma 1* and *Lemma 2*, we know the minimum value of $Penalty(\vec{x},\vec{y},\vec{z})$ is achieved if and only if $D_k(u,v)=1$ and $E_k(u,v)=1$ for every $k$ and $(u,v)$. First, if all the components of $\vec{z}^*$ are integers, it is easy to see the placement is legal, because all the cells are assigned to a certain device layer, and for any point $(u,v)$ on any device layer $k$ there is only one cell covering this point (no overlaps).

Next, we show that there does not exist a $z_i^*$ with a non-integer value (proof by contradiction). If a cell $v_i$ has a non-integer $z_i^*$, we know that there are $K$ cells covering $(x_i^*,y_i^*)$ because $\sum_{k=1}^{K} D_k(x_i^*,y_i^*) = K$. According to the pigeonhole principal, among these $K$ cells there are at least two cells $v_{i1},v_{i2}$ with the $z$-direction distance $\left|z_{i1}^*-z_{i2}^*\right|<1$, since all the variables $\{z_i^*\}$ are in the range of $[1,K]$. Without loss of generality we may assume $z_{i1}^* \le z_{i2}^*$, therefore there exists an integer $k\in\{1,2,\ldots K\}$ such that either $z_{i1}^*\in(k,k+0.5]$ and $z_{i2}^*\in(k,k+1.5)$, or $z_{i1}^*\in(k-0.5,k]$ and $z_{i2}^*\in(k-0.5,k+1)$. It is easy to verify that in the former case $\left|z_{i1}^*-(k+0.5)\right|+\left|z_{i2}^*-(k+0.5)\right|<1$ and $E_k(x_i^*,y_i^*)\ge \eta(k+0.5,z_{i1}^*)+\eta(k+0.5,z_{i2}^*)>1$; in the later case $\left|z_{i1}^*-k\right|+\left|z_{i2}^*-k\right|<1$ and $D_k(x_i^*,y_i^*)\ge\eta(k,z_{i1}^*)+\eta(k,z_{i2}^*)>1$. Both cases lead to either $E_k(x_i^*,y_i^*)>1$ or $D_k(x_i^*,y_i^*)>1$, which conflict with the assumption that $(\vec{x}^*,\vec{y}^*,\vec{z}^*)$ is a minimizer of $Penalty(\vec{x},\vec{y},\vec{z})$.

Therefore there does not exist a non-integer $z_i^*$, and every minimizer of $Penalty(\vec{x},\vec{y},\vec{z})$ is a legal placement in the z-dimension. ■

In the analytical placement engine, the densities $D_k(u,v)$ and $E_k(u,v)$ are replaced by smoothed densities $\widehat{D}_k(u,v)$ and $\widehat{E}_k(u,v)$ for differentiability. As in [11], the densities are smoothed by solving Helmholtz equations:

$$\begin{aligned}\widehat{D}_k(u,v) &= -\left(\frac{\partial^2}{\partial u^2}+\frac{\partial^2}{\partial v^2}-\varepsilon\right)^{-1} D_k(u,v)\\ \widehat{E}_k(u,v) &= -\left(\frac{\partial^2}{\partial u^2}+\frac{\partial^2}{\partial v^2}-\varepsilon\right)^{-1} E_k(u,v)\end{aligned} \qquad (15)$$

And the smoothed density penalty function

$$\begin{aligned}Penalty(\vec{x},\vec{y},\vec{z}) = &\sum_{k=1}^{K}\int_0^1\int_0^1\left(\widehat{D}_k(u,v)-1\right)^2 dudv\\ &+\sum_{k=1}^{K-1}\int_0^1\int_0^1\left(\widehat{E}_k(u,v)-1\right)^2 dudv\end{aligned} \qquad (16)$$

is used in our implementation, whose gradient is computed efficiently with the method in [14].

## IV. Multilevel Framework

The optimization problem below summarizes our analytical placement engine:

$$\begin{cases}\text{minimize} & \sum_{e\in E}\left(l(e)+\alpha v(e)\right)\\ & +\mu\left(\sum_{k=1}^{K}\int_0^1\int_0^1\left(\widehat{D}_k(u,v)-1\right)^2 dudv\right.\\ & \left.+\sum_{k=1}^{K}\int_0^1\int_0^1\left(\widehat{D}_k(u,v)-1\right)^2 dudv\right)\\ \text{increase } \mu & \text{until the density penalty is small enough}\end{cases} \qquad (17)$$

This analytical engine is incorporated into the multilevel framework in [7], which consists of coarsening, relaxation, and interpolation.

The purpose of coarsening is to build a hierarchy for the multilevel diagram, where we use the best-choice hypergraph clustering [15].

After the hierarchy is set up, multiple placement problems are solved from the coarsest level to the finest level. In a coarser level, clusters are modeled as cells and the connections between clusters are modeled as nets, so that there is one placement problem for each level. The placement problem at each level is solved (relaxed) by the analytical engine (17).

These placement problems are solved in the order from the coarsest level to the finest level, where the solution at a coarser level is interpolated to obtain an initial solution of the next finer level. The cell with highest degree in a cluster is placed in the center of this cluster (C-points), while the other cells are placed at the weighted average locations of their neighboring C-points, where the weights are proportional to the connectivity to those clusters.

## V. Experimental Results

Our experiment is performed on the IBM-PLACE benchmark [16], which is a standard cell circuit without I/O ports, as was done in [1]. In this experiment, we also assume a 4-layer implementation of 3D IC. The floorplan size is scaled from [16] by dividing the original area by 4, and then enlarging it to obtain 10% white space. Filler cells without connection to any other cells are added to ensure the existence of feasible solutions for the equality constraints in problem (4).

TABLE I – Benchmark Characteristics and 3D Placement Results from [1]

| Circuit | #cell | #net | 2D WL (x $10^7$) | LST (r=10%) | | LST (8x8 win) | | Folding-2 | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | WL (x $10^7$) | #TSV (x $10^3$) | WL (x $10^7$) | #TSV (x $10^3$) | WL (x $10^7$) | #TSV (x $10^3$) |
| ibm01 | 12282 | 11507 | 0.52 | 0.25 | 18.52 | 0.35 | 6.69 | 0.46 | 1.67 |
| ibm03 | 22207 | 21621 | 1.37 | 0.66 | 30.43 | 0.84 | 12.32 | 1.14 | 4.13 |
| ibm04 | 26633 | 26163 | 1.67 | 0.85 | 37.41 | 1.10 | 15.32 | 1.55 | 2.94 |
| ibm06 | 32185 | 33354 | 2.20 | 1.10 | 50.14 | 1.44 | 19.32 | 2.02 | 4.12 |
| ibm07 | 45135 | 44394 | 3.73 | 1.83 | 65.09 | 2.37 | 25.02 | 3.18 | 5.93 |
| ibm08 | 50977 | 47944 | 3.94 | 1.98 | 70.32 | 2.56 | 25.21 | 3.48 | 5.80 |
| ibm09 | 51746 | 50393 | 3.46 | 1.72 | 72.79 | 2.34 | 23.84 | 3.19 | 4.54 |
| ibm13 | 81508 | 83806 | 6.58 | 3.24 | 121.14 | 4.50 | 42.57 | 6.03 | 7.70 |
| ibm15 | 158244 | 161196 | 16.50 | 8.26 | 246.51 | 11.40 | 72.96 | 14.50 | 15.13 |
| ibm18 | 210323 | 200565 | 24.30 | 12.60 | 297.77 | 17.40 | 83.38 | 22.40 | 12.08 |
| geo-mean | | | 3.58 | 1.78 | 70.73 | 2.38 | 24.99 | 3.20 | 5.32 |
| normalized | | | 2.25 | 1.12 | 1.29 | 1.50 | 0.45 | 2.01 | 0.10 |

TABLE II – Experimental Results for our Multilevel Analytical Placement Method with α=10

| Circuit | 1-Level Placement (flat) | | | | 2-Level Placement | | | | 3-Level Placement | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GP WL (x $10^7$) | DP WL (x $10^7$) | #TSV (x $10^3$) | Runtime (min) | GP WL (x $10^7$) | DP WL (x $10^7$) | #TSV (x $10^3$) | Runtime (min) | GP WL (x $10^7$) | DP WL (x $10^7$) | #TSV (x $10^3$) | Runtime (min) |
| ibm01 | 0.27 | 0.28 | 8.11 | 5.90 | 0.34 | 0.35 | 1.63 | 6.66 | 0.36 | 0.37 | 0.87 | 7.19 |
| ibm03 | 0.65 | 0.67 | 16.83 | 10.77 | 0.80 | 0.81 | 4.06 | 12.09 | 0.84 | 0.84 | 2.92 | 12.31 |
| ibm04 | 0.86 | 0.85 | 28.10 | 20.61 | 1.01 | 0.99 | 7.68 | 24.21 | 1.13 | 1.11 | 3.36 | 24.21 |
| ibm06 | 1.01 | 1.00 | 38.48 | 22.65 | 1.25 | 1.22 | 10.03 | 26.26 | 1.48 | 1.45 | 3.40 | 27.07 |
| ibm07 | 1.55 | 1.55 | 53.87 | 28.97 | 1.78 | 1.75 | 18.60 | 35.06 | 2.30 | 2.27 | 4.46 | 36.00 |
| ibm08 | 1.68 | 1.68 | 53.86 | 26.60 | 1.99 | 1.93 | 18.48 | 29.80 | 2.39 | 2.36 | 4.43 | 30.81 |
| ibm09 | 1.43 | 1.44 | 61.79 | 26.31 | 1.69 | 1.69 | 18.62 | 29.97 | 2.11 | 2.08 | 3.37 | 30.14 |
| ibm13 | 2.60 | 2.64 | 110.51 | 39.18 | 2.95 | 2.96 | 36.86 | 50.41 | 4.18 | 4.14 | 4.37 | 45.29 |
| ibm15 | 6.83 | 6.72 | 260.19 | 96.33 | 7.50 | 7.28 | 107.48 | 141.69 | 8.99 | 8.74 | 27.53 | 114.04 |
| ibm18 | 9.84 | 9.69 | 333.02 | 128.09 | 10.77 | 10.33 | 140.53 | 165.71 | 13.43 | 12.88 | 38.35 | 156.42 |
| geo-mean | 1.58 | 1.59 | 55.00 | 27.95 | 1.86 | 1.84 | 16.82 | 33.72 | 2.23 | 2.20 | 4.95 | 33.07 |
| normalized | 1.00 | 1.00 | 1.00 | 1.00 | 1.17 | 1.16 | 0.31 | 1.21 | 1.41 | 1.38 | 0.09 | 1.18 |

In TABLE I, we list the statistics of the 10 out of 18 circuits in the benchmark, which are the only circuits with available experimental results in [1]. We also list partial results from the previous transformation-based 3D placement method [1] for later comparison. Here only the results of the three typical transformation schemes are listed, including the local-stacking transformation "LST (r=10%)", window-based transformation "LST (8x8 win)" and a folding transformation "Folding-2". The geometric averages are computed to measure the overall results. These results listed are for trade-offs between wirelength and TS via number without awareness of temperature, thus the following comparisons are valid.

TABLE II presents the results for our multilevel analytical placer with the TS via weight $\alpha = 10$. Due to the comparability and the page limit, we only show our results on those 10 circuits in the benchmark. Three sets of results are collected, for 1-level placement, 2-level placement and 3-level placement, respectively. The 1-level placement is to run the analytical placement engine directly without any clustering, while 2-level or 3-level placements construct a 2-level or 3-level hierarchy by clustering. The wirelength after global placement (GP WL), the wirelength after detailed placement (DP WL), the number of TS vias (#TSV), and the runtime are all collected.

In TABLE II, we see that with the same weight for TS via number, 1-level placement achieve the shortest wirelength, while the 3-level placement achieve the fewest TS via number. We compare our multilevel analytical placement method and the previous transformation-based placement method [1] by comparing our 1-level placement with the "LST (r=10%)" (the best wirelength case), and comparing out 2-level placement with the "LST (8x8 win)", and compare the 3-level placement with the "Folding-2" method (the best TS via case). From the data shown in TABLE I and TABLE II, it is clear that our 1-level placement achieves on average 12% shorter wirelength and 29% fewer TS via than "LST (r=10%)"; our 3-level placement also achieves on average 30% shorter wirelength with slightly fewer TS via number than "Folding-2".

In order to obtain a more complete comparison, different trade-off curves are generated for the circuit *ibm13* in [16]. In the curves "1-Level", "2-Level" and "3-Level", the TS via number control is achieved by increasing the weighting factor $\alpha$, from 10 to 10000. In the meantime, the results of ibm13 with "LST (r=10%)", "LST (r=20%)", "LST (8x8 win)", "Folding-4" and "Folding-2" in [1] are also connected by a curve to visualize the trade-offs. In this example, it is very clear that the analytical placement engine outperforms the transformation-based method both in wirelength and TS via number. Moreover, the multilevel framework enables a sharp reduction of TS via number, without much degradation in the wirelength. Be aware that the results for other circuits have similar behaviors.

We also notice that in the comparison between "Folding-2" and our 3-level placement, the latter does not achieve as few TS via number as "Folding-2" for two large circuits. It is because the analytical engine minimizes the objective $\sum_{e \in E}(l(e)+10v(e))$ and the fixed weighting factor $\alpha = 10$ makes the results scale differently from the "Folding-2"

method. By measuring the results with this objective function, the 3-level placement still outperforms the "Folindg-2" method.

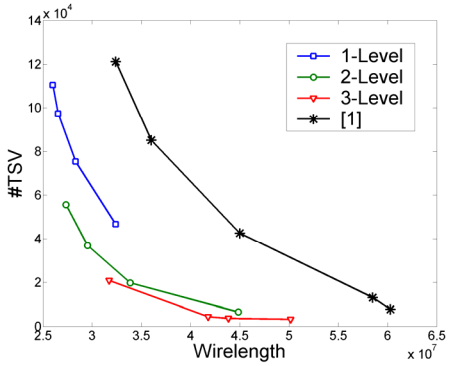


**Fig. 5** – Trade-off Curve for the Circuit ibm13

Moreover, to demonstrate the ability of controlling the TS via number with our multilevel analytical 3D placer, we run the experiments by a 3-level placement with $\alpha = 1000$. The experimental results are shown in TABLE III. Compared to "Folding-2", this set of results achieves on average 20% shorter wirelength and 50% fewer TS via number. In addition, the TS via number obtained by 4-way mincut using hMetis [17] is also listed in this table for reference, which focuses on TS via minimization only by minimizing the total cutsize. Our TS via numbers are very close to the mincut results which can be viewed almost as a lower bound.

TABLE III – Experimental Results for 3-Level Placement with α=1000

| Circuit | 3-Level Placement | | | 4-way Mincut | |
|---|---|---|---|---|---|
| | GP WL (x $10^7$) | DP WL (x $10^7$) | #TSV (x $10^3$) | cutsize (x $10^3$) | #TSV (x $10^3$) |
| ibm01 | 0.39 | 0.39 | 0.92 | 0.35 | 0.42 |
| ibm03 | 0.92 | 0.91 | 2.10 | 1.28 | 2.02 |
| ibm04 | 1.36 | 1.31 | 2.01 | 1.41 | 1.89 |
| ibm06 | 1.67 | 1.62 | 2.60 | 1.63 | 2.63 |
| ibm07 | 2.79 | 2.70 | 2.72 | 2.13 | 2.97 |
| ibm08 | 2.99 | 2.89 | 2.83 | 2.02 | 2.60 |
| ibm09 | 2.36 | 2.29 | 2.47 | 1.35 | 1.90 |
| ibm13 | 5.02 | 4.89 | 3.20 | 1.62 | 2.21 |
| ibm15 | 12.05 | 11.40 | 8.27 | 4.20 | 6.19 |
| ibm18 | 18.36 | 17.37 | 9.82 | 2.95 | 4.68 |
| geo-mean | 2.66 | 2.58 | 2.95 | 1.61 | 2.29 |
| normalized | 1.68 | 1.62 | 0.05 | | 0.04 |

## VI. Conclusions and Future Work

In this paper a multilevel analytical 3D placement algorithm is proposed, which minimizes the weighted sum of wirelength and TS via number with careful handling of overlap removal and device layer assignment in an analytical solver. The overlap removal and device layer assignment are handled by a density penalty function, whose minimizer is guaranteed to be a legal placement in the z-dimension. Experimental results demonstrate that our multilevel framework combined with the analytical engine is very effective in controlling the TS via numbers.

We note that the multilevel placement does not obtain wirelength as short as the flat placement. A possible explanation is that clustering limits solution space by requiring all cells in a cluster to be in the same layer. Further study to overcome this limitation is underway.

## Acknowledgements

This research is partially supported by IBM under a DARPA subcontract, and supported by National Science Foundation under grants CCF-0430077 and CCF-0528583.

## References

[1] J. Cong, G. Luo, J. Wei, and Y. Zhang, "Thermal-Aware 3D IC Placement via Transformation," *Proceedings of the 12th Asia and South Pacific Design Automation Conference*, pp. 780-785, 2007.

[2] B. Goplen and S. Sapatnekar, "Efficient Thermal Placement of Standard Cells in 3D ICs using a Force Directed Approach," *Proceedings of the 2003 IEEE/ACM International Conference on Computer-Aided Design*, pp. 86, 2003.

[3] B. Goplen and S. Spatnekar, "Placement of 3D ICs with Thermal and Interlayer Via Considerations," *Proceedings of the 44th Annual Conference on Design Automation*, pp. 626-631, 2007.

[4] G.-J. Nam, "ISPD 2006 Placement Contest: Benchmark Suite and Results," *Proceedings of the 2006 International Symposium on Physical Design*, pp. 167-167, 2006.

[5] H. Yan, Q. Zhou, and X. Hong, "Efficient Thermal Aware Placement Approach Integrated with 3D DCT Placement Algorithm," *Proceedings of the 9th International Symposium on Quality Electronic Design*, pp. 289-292, 2008.

[6] G.-J. Nam and J. Cong, "Modern Circuit Placement : Best Practices and Results," *Springer*, New York, 2007.

[7] J. Cong and J. Shinnerl, "Multilevel Optimization in VLSICAD," *Kluwer Academic Publishers*, Boston, 2003.

[8] J. Cong and M. Xie, "A Robust Mixed-Size Legalization and Detailed Placement Algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 27, no. 8, pp. 1349-1362, August 2008.

[9] J. Nocedal and S.J. Wright, "Numerical Optimization 2nd ed.," *Springer*, 2006.

[10] W.C. Naylor, R. Donelly, and L. Sha, "Non-linear Optimization System and Method for Wire Length and Delay Optimization for an Automatic Electric Circuit Placer," *US Patent 6301693*, October, 2001.

[11] T. Chan, J. Cong, and K. Sze, "Multilevel Generalized Force-directed Method for Circuit Placement," *Proceedings of the 2005 International Symposium on Physical Design*, pp. 185-192, 2005.

[12] A.B. Kahng, S. Reda, and Q. Wang, "Architecture and Details of a High Quality, Large-scale Analytical Placer," *Proceedings of the 2005 IEEE/ACM International Conference on Computer-Aided Design*, pp. 891-898, 2005.

[13] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, and Y.-W. Chang, "A High-quality Mixed-size Analytical Placer Considering Preplaced Blocks and Density Constraints," *Proceedings of the 2006 IEEE/ACM International Conference on Computer-Aided Design*, pp. 187-192, 2006.

[14] J. Cong and G. Luo, "Highly Efficient Gradient Computation for Density-constrained Analytical Placement Methods," *Proceedings of the 2008 International Symposium on Physical Design*, pp. 39-46, 2008.

[15] C. Alpert, A. Kahng, G.-J. Nam, S. Reda, and P. Villarrubia, "A Semi-persistent Clustering Technique for VLSI Circuit Placement," *Proceedings of the 2005 International Symposium on Physical Design*, pp. 200-207, 2005.

[16] http://er.cs.ucla.edu/benchmarks/ibm-place/

[17] G. Karypis and V. Kumar, "Multilevel k-way Hypergraph Partitioning," *Proceedings of the 36th ACM/IEEE Conference on Design Automation*, pp. 343-348, 1999.