OTPlace-Vias: A Novel Optimal Transport Based Method for High Density Vias Placement in 3D Circuits

Lin Chen[a], Qi Xu[b], Hu Ding[a] *
[a]School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China
[b]School of Microelectronics, University of Science and Technology of China, Hefie 230026, China
ischenlin@mail.ustc.edu.cn,{xuqi,huding}@ustc.edu.cn

ABSTRACT

Three-dimensional integrated circuit (3D IC) is an important manufacturing technology. In particular, the Monolithic 3D (M3D) technology stands out as a cutting-edge approach that provides higher integration density. However, M3D also introduces several challenges in terms of high density and computational complexity. In this paper, we propose a new approach for solving the inter-tier vias placement problem through optimal transport, which can be efficiently implemented in parallel with GPUs and consequently achieves significant speedup. Moreover, comparing with previous methods, our approach can also facilitate the processing of high integration density circuits to be more effective.

1 INTRODUCTION

In recent years, the 3D-IC technology has attracted a great amount of attention for its advantages in chip design. For example, comparing with the conventional 2D-ICs, it can yield higher integration density, smaller wirelengths, and enhanced power efficiency [6]. To realize 3D-IC design, a crucial factor that needs to be taken into account is the signal transmission between different layers. The transmission can be conducted through inter-tier vias such as *Through Silicon Vias (TSVs)* [10] and *Monolithic Inter-tier Vias (MIVs)* [4]. TSVs are relatively larger vertical vias (approximate 5um in diameter [8]) that penetrate the silicon substrate to interlink with multiple dies. Recently the Monolithic 3D (M3D) technology has emerged as a cutting-edge approach that yields higher integration density compared to TSV-based 3D ICs [6]; M3D utilizes diminutive MIVs with diameter smaller than 100nm, which are much smaller than TSVs as shown in Fig.1. M3D can also deliver one node PPC (Performace, Power, Cost) benefit, wherease TSV-based 3D can only achieve a half node PPC advantage [6]. Besides, M3D also introduces several other important benefits, such as smaller 3D contact pitch and higher integration density, which are quite helpful to develop compact 3D integration solutions [4].

Several elegant methods have been proposed for the vias placement problem and most of them focus on TSV assignment. Chen et al. [1] designed a min-cost max-flow model to solve the TSV assignment problem on a single device layer, and then proposed an integral min-cost multicommodity (IMCMC) flow model for multiple device layers; due to the NP hardness of IMCMC [12], they also provided a heuristic algorithm to handle it. Liu et al. [5] proved that the general 3D-IC TSV assignment problem, when dealing with more than two dies, is NP-complete, and then they proposed an efficient heuristic algorithm based on the techniques like shortest path and integer linear programming to achieve a good practical performance. To reduce the computational complexity, Hao and Yoshimura [3] proposed a multilevel algorithm that simultaneously reduces the wirelength and runtime.

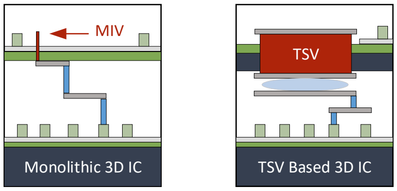

Figure 1: MIV (less than 100nm) vs TSV (about 5um) in 3D integration [8].

Despite of those progresses on TSV assignment, the current research on MIVs is still quite limited to the best of our knowledge. In particular, it is urgent to address the substantial computational challenges arising from high-density MIVs. If we directly apply the aforementioned TSV assignment algorithms, they may suffer several significant issues. Different from TSV-based 3D ICs, the M3D technology features a notably higher integration density and consequently entails the placement for a considerably large number of MIVs. For example, it was shown that some circuits may need more than 160K MIVs [4]. Therefore, the large computational complexity has become a major bottleneck for MIV placement. In our experiments (Section 4) we show that these algorithms are seriously slowed down when dealing with the substantial volume of inter-tier vias in 3D-IC design.

In this paper, we aim to tackle the high-density challenge and design a new inter-tier vias placement approach that can be implemented fast in practice. Our key idea is based on the theory of *Optimal Transport (OT)*, which is also known as the Monge-Kantorovich problem [9]. OT is a fundamental mathematical topic that has many important applications in the real world, and more details are shown in Section 2.2. Our contributions can be summarized as follows:

(1) Stemming from some geometric insights on the inter-tier vias placement problem, we propose a novel OT model for solving both two-layer and multi-layer layouts. In particular,

*Corresponding Author. This work was supported in part by the Fundamental Research Funds for the Central Universities through Grant WK2150110025.

DAC '24, June 23–27, 2024, San Francisco, CA, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0601-1/24/06…$15.00
https://doi.org/10.1145/3649329.3656249

together with a cute greedy rounding idea, we can apply the Sinkhorn algorithm [2] to efficiently solve our OT model through GPU acceleration with preserving high-quality solution, which is a major advantage over most existing vias placement algorithms.

(2) To evaluate the performance of our proposed method for addressing the high-density challenges, we also conduct a set of experiments on benchmark datasets. The experimental results suggest that our method can achieve significant improvement of running time upon previous approaches. For example, in a circuit with 50k nets, our OT-based method can compute the solution in 36s, while a classical algorithm takes 6.8h, achieving a remarkable acceleration ratio of 670 times.

2 PRELIMINARIES

We introduce the problem formulation of inter-tier vias placement in Section 2.1, and then introduce optimal transport in Section 2.2.

2.1 Problem Formulation

In Fig.2, we illustrate an example for inter-tier vias placement. The locations of the blocks and pins have already been determined, and the goal is to locate the signal inter-tier vias in the reserved whitespace so that the nets spanning multiple layers can be connected. It is believed that the interconnecting of pins takes a significant part of the timing and power in current design [4]. Thus, to improve the effectiveness of a circuit, the most frequently considered optimization goal in inter-tier vias placement is to minimize the total wirelength [3, 5]. Formally, we are given the following setting:

- A 3D IC with C layers, and the layers are numbered from the bottom to the top as 0, 1, ..., $C-1$. Suppose there is a net spanning the layers from $c^{\perp}$ to $c^{\top}$ ($0 \leq c^{\perp} < c^{\top} \leq C$), then $c^{\top} - c^{\perp}$ inter-tier vias should be placed on the layers $c^{\perp}+1$, $c^{\perp}+2$, ..., $c^{\top}$, so that the net is connected.
- A set of blocks $\texttt{Blo} = \{b_1, b_2, \cdots, b_k\}$. Each block occupies an area on the 3D layout, and inter-tier vias cannot be located in the areas occupied by the blocks.
- A netlist $\texttt{NetL} = \{nt_1, nt_2, \cdots, nt_n\}$, and each net is a collection of pins, where each pin corresponds to a coordinate on the 3D layout. For each nt_i, it corresponds to two numbers, $c_i^{\perp} \geq 0$ and $c_i^{\top} \leq C-1$, respectively representing the lowest and highest layers spanned by the net.

Usually there is an evenly partitioned grid in each layer (see Fig.2 (left) for an example). The blue areas are taken by blocks and we can only consider the whitespace for vias placement. Suppose the grid structure size is $P \times Q$, $P, Q \in \mathbb{Z}^{+}$; for each layer c and $1 \leq j \leq P \times Q$, we use g_j^c to denote the j-th grid cell, and use $\texttt{ws}(g_j^c)$ to denote the area of the whitespace of g_j^c. Let τ be the area size taken by an inter-tier via, then the capacity of g_j^c is defined to be $\texttt{cap}(g_j^c) = \lfloor \texttt{ws}(g_j^c)/\tau \rfloor$, which is equal to the maximum number of vias that can be accommodated in g_j^c.

The goal of inter-tier vias placement is to determine n sets of grid cells $\texttt{Gri}_1, \cdots, \texttt{Gri}_n$, where $\texttt{Gri}_i = \{v_i^c \mid c_i^{\perp} < c \leq c_i^{\top}\}$ for $1 \leq i \leq n$ and each v_i^c denotes the location of the via located on layer c for nt_i, so as to connect those nets spanning multiple layers and minimize the total wirelength of them. In other word, our task is exactly equivalent to finding an appropriate function δ that maps each couple (i, c) to an index $j \in [1, P \times Q]$, i.e., $v_i^c = g_{\delta(i,c)}^c$. Moreover, for $c_i^{\perp} \leq c \leq c_i^{\top}$, we use nt_i^c to denote the subnet of nt_i on layer c when the inter-tier vias have been placed. In particular, except for the top and bottom layers, any middle layer c can be influenced by two vias (one is from the layer c and the other is from the layer $c+1$). See the middle layer of Fig.2 (right) as an example. Formally, our objective is to minimize the following cost function:

$$\text{Cost}(\texttt{Gri}_1, \cdots, \texttt{Gri}_n) \quad = \quad \sum_{i=1}^{n} \sum_{c=c_i^{\perp}}^{c_i^{\top}} \text{HPWL}(nt_i^c, V_i^c) \qquad (1)$$

where $\text{HPWL}(nt_i^c, V_i^c)$ denotes the commonly used half-perimeter wirelength (HPWL) of the 2D bounding box of nt_i^c with the set of placed vias V_i^c, and $V_i^c = \{v_i^{c+1}, v_i^c\}$ if $c_i^{\perp} < c < c_i^{\top}$, $V_i^c = \{v_i^{c+1}\}$ if $c = c_i^{\perp}$, and $V_i^c = \{v_i^c\}$ if $c = c_i^{\top}$. To see the influence of inter-tier vias to wirelength, We consider a toy example in Fig.2 (right): a net has 6 pins: $\{p_0, p_1\}$ on layer 0, $\{p_2, p_3\}$ on layer 1, and $\{p_4, p_5\}$ on layer 2. We have two vias placement solutions (Via_1, Via_2) and (Via_1', Via_2') for connecting these layers; the resulting bounding boxes are labeled in red and blue respectively. Obviously, we can see that the red one has smaller wirelength than the blue one, indicating that (Via_1, Via_2) is the better choice.

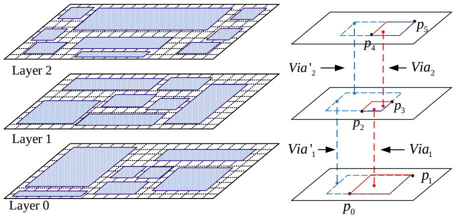

Figure 2: . The left figure displays a grid structure of 3D-IC: the blue areas are occupied by blocks, and we can only place the vias in the whitespace. The right figure illustrates an example to show the impact of vias placement to wirelength.

2.2 Optimal Transport

In this section, we briefly introduce the formulation of OT that is used throughout our paper. Roughly speaking, OT is used to compute the optimal matching between two sets of points, which are usually called "supply" and "demand". A number of different real-world applications in the areas like computer vision, machine learning, and economics, can be formulated as OT problems [9].

Suppose the numbers of supply and demand nodes are m_1 and m_2, respectively. For the supply nodes, we have an m_1-dimensional vector $p = [p_1, p_2, \cdots, p_{m_1}]$ with each entry $p_i \geq 0$ representing the amount of the resource that the corresponding node can supply; similarly, we have an m_2-dimensional vector $q = [q_1, q_2, \cdots, q_{m_2}]$ for the demand nodes, where each entry $q_j \geq 0$ represents the amount of the resource that the corresponding node demands. We require their sums to be equal, i.e., $\sum_{i=1}^{m_1} p_i = \sum_{j=1}^{m_2} q_j$. We also have an $m_1 \times m_2$ cost matrix A, where each entry $A_{i,j}$ indicates the unit transportation cost from the i-th supply node to the j-th demand node. The OT objective is to find a non-negative $m_1 \times m_2$ transportation plan matrix F, where each entry $F_{i,j}$ represents the transportation amount from the i-th supply node to the j-th demand

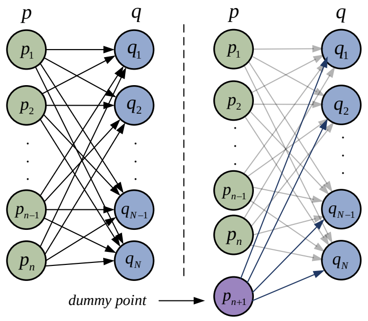

Figure 3: The optimal transport model with p representing supply nodes and q representing demand nodes. The right figure illustrates the use of a "dummy" point to balance the total weights.

node, such that the total cost is minimized:

$$\texttt{argmin}_{F\geq 0} \sum_{i,j} F_{i,j} * A_{i,j} \tag{2}$$

where $\sum_j F_{i,j} = p_i$ for each $1 \leq i \leq m_1$, and $\sum_i F_{i,j} = q_j$ for each $1 \leq j \leq m_2$. The optimal transport problem can be solved by various methods. One common approach is to solve it as a linear programming problem and the representative algorithm is the *network simplex* method [7]. Recently, a more efficient algorithm called "*Sinkhorn's algorithm*" was proposed, which is based on the Sinkhorn's matrix scaling technique [2]. For a more detailed introduction on OT, we refer the reader to the book [9].

3 OUR PROPOSED APPROACH

As the warm-up, we first consider the OT model for the basic two-layer inter-tier vias placement problem in Section 3.1. Subsequently, in Section 3.2 we present our OT-based approach for addressing the general multiple-layers case. In Section 3.3, we introduce the efficient implementation of our method using GPUs.

3.1 Warm-up: Two-Tier Vias Placement

The inter-tier vias placement problem with two layers can be modeled as a min-cost max-flow problem [1, 14], which can be solved via some off-the-shelf min-cost max flow algorithms [7]. However, for high-density circuits with considerable number of vias, solving such a min-cost max flow problem is quite expensive. Fortunately, this min-cost max flow problem can be modeled as an optimal transport problem directly.

For the two-layer case ($c = 0, 1$), each cross-layer net only requires one via, and the vias should only be placed on layer 1. In the OT model built for two layers, the supply nodes can be viewed as the vias set (corresponding to the vector p in Section 2.2); each p_i has the weight 1, indicating that a via can be assigned to only one whitespace cell. The demand nodes are viewed as the grid cells (corresponding to the vector q in Section 2.2); each q_j is corresponding to the grid g^1_j, and it has the weight $\mathsf{cap}(g^1_j)$. The unit transportation cost is consistent with Equation (1), i.e., $A_{i,j} = \text{HPWL}(nt^0_i, \{g^1_j\}) + \text{HPWL}(nt^1_i, \{g^1_j\})$, which denotes the cost by placing the via in the grid cell g^1_j for nt_i.

The only place that needs paying more attention to is that the total weights of these two sets of nodes may not be equal (for simplicity, we let $N = P \times Q$), i.e., it is possible $\sum_{i=1}^{n} 1 = n < \sum_{j=1}^{N} \mathsf{cap}(g^1_j)$ (note that n cannot be larger than $\sum_{j=1}^{N} \mathsf{cap}(g^1_j)$, otherwise, the problem has no feasible solution). It is worth emphasizing that this requirement of equivalent total weights is particularly important to our following acceleration in Section 3.3, since otherwise the Sinkhorn's algorithm cannot run (the Sinkhorn's matrix scaling does not work for this imbalanced case).

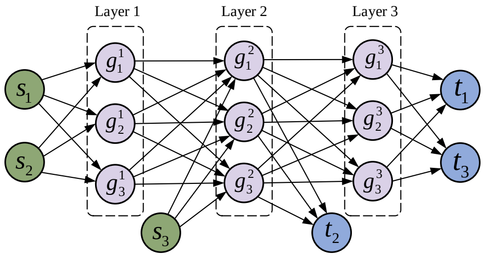

Figure 4: The IMCMC network for a layout with 4 layers and 3 nets: nt_1 spans layers 0 to 3, nt_2 spans layers 0 to 2, and nt_3 spans layers 1 to 3.

We can resolve this issue by adding a "dummy" node p_{n+1} in the left column with the weight $\sum_{j=1}^{N} \mathsf{cap}(g^1_j) - n$ (i.e., the total residual amount of $\sum_{j=1}^{N} \mathsf{cap}(g^1_j) - n$ flow comes from p_{n+1}); the edge cost from p_{n+1} to each demand node q_j is set to be a unified value $\omega > 0$. See Fig. 3 for an illustration. Then the total transportation cost of (2) becomes

$$\begin{aligned} \sum_{i=1}^{n+1}\sum_{j=1}^{N} F_{i,j} * A_{i,j} &= \sum_{i=1}^{n}\sum_{j=1}^{N} F_{i,j} * A_{i,j} + \omega \sum_{j=1}^{N} F_{n+1,j} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{N} F_{i,j} * A_{i,j} + \omega \underbrace{\Big(\sum_{j=1}^{N} \mathsf{cap}(g^1_j) - n\Big)}_{\text{fixed value}}. \end{aligned} \tag{3}$$

Since the second term of (3) is fixed, we can claim that optimizing "$\sum_{i=1}^{n+1}\sum_{j=1}^{N} F_{i,j} * A_{i,j}$" is equivalent to optimizing "$\sum_{i=1}^{n}\sum_{j=1}^{N} F_{i,j} * A_{i,j}$". Therefore we can compute the optimal vias placement solution for two layers through this OT model. However, for the general case with more than two layers, it needs some significant new ideas to adapt it to an OT problem.

Remark 1. *Actually there is a remaining issue for the OT model. Our vias assignment solution should be an integral solution. But the solution obtained from the objective function (2) may not be an integral solution, i.e., the values $F_{i,j}$s are not necessary to be integers, especially when using the entropic smoothed Sinkhorn's algorithm. We just omit this issue in this section (as well as Section 3.2), and leave the detailed discussion to Section 3.3.*

3.2 The General Multi-tier Vias Placement

As mentioned before, the real-world inter-tier vias placement often needs to consider more than two layers [1], which is a much more challenging task than solving the case with only two layers. To address this problem, the integral min-cost multicommodity (IMCMC) flow model has been widely studied [1, 3, 5]. Before introducing our OT model, we revisit the IMCMC flow model first.

The IMCMC flow model. We build a directed graph $G(V, E)$ as illustrated in Fig.4. Suppose there are $C \geq 2$ layers, and each c-th layer contains $N_c \geq 1$ available grid cells for locating the vias.

The vertex set V consists of three parts: $V = S \cup V_g \cup T$, where the source nodes $S = \bigcup_{i=1}^{n} s_i$, the sink nodes $T = \bigcup_{i=1}^{n} t_i$, and the grid cell nodes $V_g = \bigcup_{c=1}^{C-1} \bigcup_{j=1}^{N_c} g_j^c$ with each g_j^c representing the j-th grid cell in the c-th layer. The set V_g is partitioned to $C-1$ columns with each column corresponding to one placement layer. Also, the nodes in adjacency columns are fully connected (i.e., the subgraph of any two adjacency columns is a complete bi-partite graph). For each net nt_i, $1 \le i \le n$, there is a unique source-sink pair (s_i, t_i) corresponding to it; s_i is connected to all the nodes in the $(c_i^{\perp}+1)$-th column, and t_i is connected to all the nodes in the $c_i^{\top}$-th column (recall $c_i^{\perp}$ and $c_i^{\top}$ are the indices for the lowest and highest layers spanned by nt_i, as defined in Section 2).

We also assign capacities to both the nodes and edges. The capacity of g_j^c is $\mathsf{cap}(g_j^c)$ indicating the maximum number of vias that grid g_j^c can accommodate; the capacities of s_i and t_i are assigned to be ∞. The capacities of the edges connecting the source or sink nodes are all set to be 1. Additionally, the capacities of the edges between adjacency layers are all set to be ∞.

Each edge of the graph G may have "multiple" costs. For an edge pointing to g_j^c in the IMCMC model graph G, if nt_i need a via in this layer c, then we assign a cost equal to $\text{HPWL}(nt_i^c, V_i^c)$ (or $\text{HPWL}(nt_i^c, V_i^c)+\text{HPWL}(nt_i^{c-1}, V_i^{c-1})$ if $c = c_i^{\perp}+1$) as defined in the objective function (1).

Given these edge costs, the goal of the IMCMC flow model is to compute an "integral min-cost flow" path from s_i to t_i for each net nt_i under the capacity constraints, so that each passed node in each column (say, the c-th column) represents the locating grid cell for nt_i in the c-th layer. The requirement of "integral" means that the flow can pass only one node in each column (i.e., there is no fractional flow passing multiple nodes in the same column).

From IMCMC to OT. It is well known that computing the optimal solution for IMCMC is NP-hard, where the major obstacle is from the capacity constraints. For example, there may be multiple flows passing through a grid cell g_j^c, but the total number should be no more than the capacity $\mathsf{cap}(g_j^c)$. To design our OT-based approach for solving this IMCMC problem, we begin by a simple heuristic method below.

The inter-tier vias placement problem can be significantly simplified if all the capacity constraints are omitted. Namely, we can compute the n flow paths separately, since there is no any restriction if the paths share the same grid cell. Also, for each pair (s_i, t_i), the corresponding flow can be obtained by computing the shortest path from s_i to t_i based on the edge costs in the graph G. Let the set of traversed grid cells be $\{v'^c_i \mid c_i^{\perp} < c \le c_i^{\top}\}$ for each $1 \le i \le n$.

Since we ignore the capacity constraints, the obtained solution from the above heuristic idea may be illegitimate, i.e., some grid cells may be occupied by too many vias. We call it a "fake" solution. Actually this fake solution yields a lower bound cost for the IMCMC flow model, because the capacity constraints can only induce an increase on the total cost. So it is natural to consider to compute a valid solution as close as possible to this fake solution. Our intuition is as follows. Because the wirelength depends on the geometric locations of the vias, if we can mildly move these vias from the fake solution, the total wirelength should not be affected too much. To optimize the geometric difference between the fake solution and our solution, we use "optimal transport" and propose the optimization problem **OTPlace-Vias**.

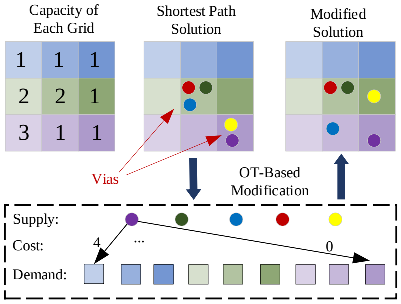

Figure 5: The placement solution obtained by using shortest path may be invalid. The OT-based modification is applied to compute a solution satisfying the capacity constraints.

Following the notations defined in Section 2.1, let $\{v_i^c \mid c_i^{\perp} < c \le c_i^{\top}, 1 \le i \le n\}$ be the solution we attempt to find, where v_i^c is the grid solution for nt_i in c-th layer. Then we propose the following objective:

$$\min \sum_{i=1}^{n} \sum_{c=c_i^{\perp}+1}^{c_i^{\top}} \mathsf{Manh}\left(v_i^c, v'^c_i\right) \tag{4}$$

where $\mathsf{Manh}\left(v_i^c, v'^c_i\right)$ is the Manhattan distance for measuring the distance between v_i^c and v'^c_i. Also, we have the capacity constraints: for any grid cell g_j^c, the number of vias assigned to it should be no larger than $\mathsf{cap}(g_j^c)$, i.e., the size of the set $\{i \mid 1 \le i \le n, v_i^c = g_j^c\}$ is at most $\mathsf{cap}(g_j^c)$.

To see why the objective (4) can be solved by OT, we need to transform it to another form. Note that the number of vias on each layer is always fixed, e.g., if there are three nets spanning the layer, the number of vias should be three. For $1 \le c \le C-1$, denote by T_c the set of nets who require placing via on layer c. Also note that these $|T_c|$ vias are not necessary to be placed to separated grid cells, as long as they do not violate the capacity constraints. So the objective function (4) can be rewritten as:

$$\min \sum_{c=1}^{C-1} \sum_{i \in T_c} \mathsf{Manh}\left(v_i^c, v'^c_i\right). \tag{5}$$

Basically, we just exchange the order of summations from (4) to (5). Then we can regard each inner summation "$\sum_{i \in T_c} \mathsf{Manh}(v_i^c, v'^c_i)$" of (5) as an independent optimal transport instance. Please see Fig.5 for an illustration. Consider computing the inter-tier vias placement solution on layer c. There are $|T_c|$ supply nodes and N_c demand nodes; we also use the vectors p and q to denote the supplying and demanding resource amounts. Similar with the setting in Section 3.1, each $p_i = 1$ and $q_j = \mathsf{cap}(g_j^c)$. To resolve the weight imbalance issue, we also need to add a dummy point as introduced in Section 3.1. The unit cost $A_{i,j} = \mathsf{Manh}(g_j^c, v'^c_i)$. Then the optimal solution for $\sum_{i \in T_c} \mathsf{Manh}(v_i^c, v'^c_i)$ can be obtained by solving the OT from the constructed p to q. Also, as mentioned in the previous Remark 1, we just omit the discussion on the "integral" issue and address it in the next section.

Remark 2 (the OT models for two-tier vs multi-tier). *The above OT model of each layer for multi-layer setting is similar to that for two-layer setting introduced in Section 3.1, where the major difference lies in the cost setting. In a two-layer setting, the quality of a via position for a net is evident, as it can be directly computed based on the HPWL of the bounding boxes on the upper layer and lower layer. However, in a multi-layer setting, the problem is more complex since the vias for different layers are interdependent. This is also the essential difficulty for multi-tier vias placement problem.*

3.3 Solving OTPlace-Vias with GPU

In this section, we show that our proposed OTPlace-Vias problem can be efficiently solved with GPU. The Sinkhorn algorithm proposed by Cuturi [2] is an elegant iterative optimization method designed to solve the optimal transport problem, where the objective is smoothed by an entropic regularization term. Compared to the original objective (2), the modified objective function is:

$$\texttt{argmin}_{F\geq 0}\sum_{i,j}F_{i,j}*A_{i,j}-\frac{1}{\lambda}h\left(F\right) \qquad (6)$$

where $h\left(F\right)=-\sum_{i,j}F_{i,j}\log F_{i,j}$ represents the entropy of the matrix F, and λ is a positive regularization parameter. Using the entropic regularization term, the computation can be implemented by the Sinkhorn's matrix scaling algorithm [11]. It also has been proved that the time complexity is nearly linear in the input size.

One notable advantage of the Sinkhorn algorithm is its suitability for parallel computation on GPUs. The iterative nature of the algorithm, which involves matrix and vector operations, can be significantly accelerated with the benefit from the parallel processing capability of modern GPUs. Moreover, for the multi-tier vias placement problem, the computations on the shortest paths for different (s_i, t_i)s are independent, and the OT models for different layers are also independent. As a consequence, these computational tasks can also be solved in parallel. This nice property enhances the computational efficiency especially for large-scale instances.

Nevertheless, it is worth noting that the result returned by the Sinkhorn algorithm is an approximation OT solution due to the entropic regularization term. Moreover, the obtained transport plan usually contains decimal numbers, which is prohibitive to our inter-tier vias placement problem (a via can only be inserted into a single grid cell) . To resolve this issue, we propose a greedy modification strategy that is easy to implement in practice.

Greedy rounding for integral solution. We regard the transportation matrix F returned from the Sinkhorn algorithm as a probability matrix, where each entry $F_{i,j}$ denotes the likelihood of the via v_i^c to be assigned to g_j^c. A simple idea is to assign v_i^c to the grid g_j^c with the largest likelihood $F_{i,j}$ in the i-th row $F_{i:}$ of F. However, this strategy may result in the violation on the capacity constraints (similar with the issue caused by the shortest path solution), so we need to develop some sophisticated method to avoid this issue.

We design a priority order to assign these vias, and our intuition is as follows. Before placing the vias on the layer c, each of the subnets $\{nt_i^c \mid i \in T_c\}$ can be enclosed by a bounding box, say, $E(nt_i^c)$. If we insert the via v_i^c to nt_i^c, the bounding box $E(nt_i^c)$ should be enlarged to be $E(nt_i^c \cup \{v_i^c\})$. Obviously, we expect the bounding box to be as small as possible according to our optimization formulation defined in Section 2.1. Moreover, if the initial $E(nt_i^c)$ is small, it is more likely that the size of $E(nt_i^c \cup \{v_i^c\})$ is very sensitive to the location of v_i^c; on the other hand, if $E(nt_i^c)$ is big, the size of $E(nt_i^c \cup \{v_i^c\})$ should have a relatively low sensitivity to the location of v_i^c (e.g., if v_i^c is inserted into $E(nt_i^c)$, the bounding box will not change). Therefore, we order the vias $\{v_i^c \mid i \in T_c\}$ based on the corresponding bounding box sizes; the smaller the size, the higher the order. Then we can place these vias one by one following this order: let v_i^c be the current via that is waiting to assign; if the grid cell corresponding to the maximum probability in $F_{i:}$ is already full, we can move on to the next grid cell with the second highest probability and so on so forth, until reaching an available grid cell.

4 EXPERIMENTAL RESULTS

Given the available Python libraries for network flow algorithms and optimal transport, we implement our algorithms in the Python environment. The experiments were performed on a 64-bit Linux machine equipped with 8 NVIDIA RTX 2080Ti GPUs. Our evaluation utilized MCNC[1] and GSRC[2] benchmarks, including two MCNC circuits (ami33 and ami49), and four GSRC circuits (n50, n100, n200, and n300), and the details are shown in Table 1. The multi-layer floorplans are generated by an effective fixed-outline multi-layer floorplanner [13].

Table 1: Information for the datasets. #Nets, #Blocks, and #Pins respectively represent the number of nets, blocks, and pins in the circuit.

	ami33	ami49	n50	n100	n200	n300
#Nets	123	408	485	885	1585	1893
#Blocks	33	49	50	100	200	300
#Pins	522	953	1050	1873	3599	4358

As the warm-up, we conduct a comparative analysis on our proposed OT-based algorithm and the commonly used min-cost max flow method [1, 14] for the two-layer case. Subsequently, we study the performance of our OTPlacer-Vias method in a more general case with four layers. Finally, we demonstrate the computational advantage of our method for dealing with high density layouts with a large number of cross-layer nets ranging from 1K to 50K.

The Case with Two Layer. The running time and wirelength performance of our proposed OT-based method and the min-cost max-flow method are presented in Table 2. We can see that our algorithm consistently achieves significant speedups (often by factors of tens), while incurring only neglectable (less than 1%) increases in wirelength on the testing instances. The results indicate the computational efficiency and promising performance of our proposed OT-based approach.

The General Case. Our OT-based algorithm for multi-layers consists of two stages. First, we compute the shortest paths between all the pairs (s_i, t_i)s in the IMCMC flow graph; these n paths can be computed in parallel since they are independent with each other. Then, in the second stage, we solve our proposed OTPlacer-Vias problem by using the method of Section 3.3. In Figure 6, we present the results of our method and the existing algorithms [1, 3]. The results suggest that our approach achieves lower wirelengths.

[1] http://vlsicad.eecs.umich.edu/BK/MCNCbench/

[2] http://vlsicad.eecs.umich.edu/BK/GSRCbench/

Table 2: "Gird Size" is the size of the grid structure. "Speed" indicates the time acceleration factor, and "Wire↑" represents the increase of the OT-based algorithm compared to the optimal wirelength obtained by the min-cost max-flow method.

	Grid Size	Flow(s)	OT(s)	Speed	Wire↑
ami33	40×40	0.5646	0.0869	4.7X	0.0%
ami49	80×80	2.6561	0.1921	15.9X	0.1%
n50	20×20	1.1834	0.1997	12.7X	0.7%
n100	20×20	2.6620	0.2105	16.6X	0.3%
n200	20×20	12.1482	0.6897	27.7X	0.0%
n300	20×20	15.0489	0.5472	24.3X	0.1%

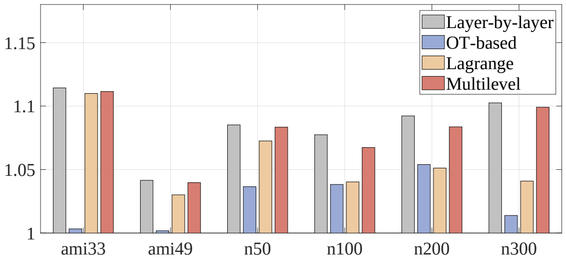

Figure 6: Performance for multi-layer cases. "Layer-by-layer" denotes placing vias layer by layer by applying a series of min-cost max flow models. "Lagrange" means the Lagrangian relaxation based algorithm [1]. "Multilevel" denotes the hierarchical heuristic method [3]. The y-axis indicates the ratio of the obtained wirelength to the lower bound (based on the fake solution by the shortest path method).

Moreover, our approach has a significant advantage in terms of the computational efficiency. For the largest circuit n300, the "Layer-by-layer" method takes 50s, while our OT-based method requires only 1.08s. The "Lagrange" and "Multilevel" methods both consume even more time than the "Layer-by-layer" method. Due to the space limit, we provide the detailed runtimes for each algorithm to our full version. More comparisons on computational efficiency between the "Layer-by-layer" method and our OT-based method can be found in Table 3.

High Density Circuits. We further conduct the experiment for the scenario that the number of nets is large (and so the number of vias is large too). We devised two types of benchmarks, using layouts from ami33 and n300 circuits respectively, since they represent two different types of circuits from MCNC and GSRC. We keep their layouts, and randomly generate a large number of nets as shown in Table 3. Our algorithms can achieve significant speedups, often by factors of tens or even hundreds. For the two-layer case, the min-cost max-flow model can yield the optimal solution, and our proposed method incurs only minimal (less than 0.5%) increases in terms of the wirelength. For the multi-layer case, comparing with the layer-by-layer min-cost max-flow method of [1], our algorithm achieves about 4%-10% reduction in the wirelength and also exhibits substantial speedup. For instance, in the n300 circuit with 50K cross-layer nets of a 4-layer setting, the layer-by-layer algorithm requires 6.8h, whereas our OT-based method produces the results within just 36s, achieving a remarkable acceleration ratio of 670 times.

Table 3: Comparison of the min-cost max-flow model and our OT-based model with different numbers of nets. For the two-layer case, we still use the min-cost max-flow method as the baseline, and the meanings of "Wire↑" and "Speed" are as same as Table2. For the 4-layer case, since there is no optimal solution, we directly take the layer-by-layer method of [1] as the baseline, and our obtained wirelength is lower and so the value "Wire↑" is negative.

	2-Layer				4-Layer			
	ami33		n300		ami33		n300	
#Nets	Wire↑	Speed	Wire↑	Speed	Wire↑	Speed	Wire↑	Speed
1K	0.2%	44X	0.0%	43X	-6.9%	37X	-8.5%	32X
2K	0.2%	58X	0.1%	70X	-7.9%	109X	-7.0%	76X
3K	0.4%	89X	0.2%	84X	-9.3%	118X	-6.8%	104X
4K	0.1%	138X	0.0%	97X	-8.7%	51X	-7.4%	97X
6K	0.2%	138X	0.1%	122X	-8.8%	71X	-8.0%	176X
8K	0.2%	166X	0.1%	146X	-9.2%	151X	-7.7%	322X
10K	0.2%	198X	0.3%	219X	-9.5%	231X	-7.9%	282X
15K	0.3%	339X	0.0%	210X	-10.2%	381X	-8.2%	367X
30K	0.0%	366X	0.0%	227X	-8.2%	449X	-5.8%	388X
50K	0.0%	575X	0.0%	343X	-7.7%	370X	-4.1%	671X

5 CONCLUSION

In this paper, we introduce a novel optimal transport based approach for solving the inter-tier vias placement problem. In particular, our algorithm can be effectively implemented by using GPUs with only neglectable sacrifice on the solution accuracy. With these computational advantages, our method has great potential to improve high integration density 3D-IC designs. In the future, it is also worth considering the applications of OT for other chip design problems.

REFERENCES

[1] Song Chen, Liangwei Ge, Mei-Fang Chiang, and Takeshi Yoshimura. 2009. Lagrangian relaxation based inter-layer signal via assignment for 3-D ICs. *IEICE transactions on fundamentals of electronics, communications and computer sciences* 92, 4 (2009), 1080–1087.

[2] Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* 26 (2013).

[3] Cong Hao and Takeshi Yoshimura. 2017. An efficient multi-level algorithm for 3D-IC TSV assignment. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 100, 3 (2017), 776–784.

[4] Young-Joon Lee and Sung Kyu Lim. 2013. Ultrahigh density logic designs using monolithic 3-D integration. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 32, 12 (2013), 1892–1905.

[5] Xiaodong Liu, Yifan Zhang, Gary Yeap, and Xuan Zeng. 2011. An integrated algorithm for 3D-IC TSV assignment. In *Proceedings of the 48th Design Automation Conference*. 652–657.

[6] Deepak Kumar Nayak, Srinivasa Banna, Sandeep Kumar Samal, and Sung Kyu Lim. 2015. Power, performance, and cost comparisons of monolithic 3D ICs and TSV-based 3D ICs. In *2015 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*. IEEE, 1–2.

[7] James B Orlin. 1997. A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming* 78 (1997), 109–129.

[8] Neela Lohith Penmetsa, Christos Sotiriou, and Sung Kyu Lim. 2015. Low power monolithic 3D IC design of asynchronous AES core. In *2015 21st IEEE International Symposium on Asynchronous Circuits and Systems*. IEEE, 93–99.

[9] Gabriel Peyré, Marco Cuturi, et al. 2017. Computational optimal transport. *Center for Research in Economics and Statistics* 2017-86 (2017).

[10] N Sillon, A Astier, H Boutry, L Di Cioccio, D Henry, and P Leduc. 2008. Enabling technologies for 3D integration: From packaging miniaturization to advanced stacked ICs. In *2008 IEEE International Electron Devices Meeting*. IEEE, 1–4.

[11] Richard Sinkhorn. 1967. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly* 74, 4 (1967), 402–405.

[12] Steven S Skiena. 1998. *The algorithm design manual*. Vol. 2. Springer.

[13] Qi Xu, Song Chen, and Bin Li. 2016. Combining the ant system algorithm and simulated annealing for 3D/2D fixed-outline floorplanning. *Applied Soft Computing* 40 (2016), 150–160.

[14] Qi Xu, Song Chen, Xiaodong Xu, and Bei Yu. 2017. Clustered fault tolerance TSV planning for 3-D integrated circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 36, 8 (2017), 1287–1300.