Unified Redistribution Layer Routing for 2.5D IC Packages

Chun-Han Chiang[1], Fu-Yu Chuang[2], and Yao-Wen Chang[1,2]
[1]Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan
[2]Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan
{chchiang, fychuang}@eda.ee.ntu.edu.tw; ywchang@ntu.edu.tw

***Abstract*—A 2.5-dimensional integrated circuit, which introduces an interposer as an interface between chips and a package, is one of the most popular integration technologies. Multiple chips can be mounted on an interposer, and inter-chip nets are routed on redistribution layers (RDLs). In traditional designs, the wire widths and spacings are uniform (i.e., grid-based). To improve circuit performance in modern designs, however, variable widths and spacings are also often adopted (i.e., gridless designs). In this paper, we propose the first unified routing framework that can handle both grid-based and gridless routing on RDLs based on the modulus-based matrix splitting iteration method (MMSIM) and bipartite matching. The MMSIM-based method assigns each wire a rough position while considering multiple design rules, and bipartite matching is applied to further refine those positions. We also prove the optimality of our RDL routing framework for grid-based designs and validate it empirically. Experimental results show that our framework can solve all the gridless and grid-based designs provided by industry effectively and efficiently. In particular, our framework is general and readily extends to other routing (and some quadratic optimization) problems.**

I. Introduction

Recently, 2.5-dimensional integrated circuits (2.5D ICs, also known as *interposer-based 3D ICs*) have become one of the most popular packaging technologies which support heterogeneous integration and enhance system performance while reducing power consumption and manufacturing complexity [22]. Figure 1(a) shows the side view of a 2.5D IC package structure. The main feature of a 2.5D IC package is the interposer which is introduced as an interface between chips and a package. With multiple chips mounted on the interposer, inter-chip nets can be routed on *redistribution layers (RDLs)* of the interposer by using the same processes as the silicon chips. The integrated chips may come from different vendors, which introduce predefined connections, i.e., *pre-assignment nets*. Moreover, different from stacked 3D IC packages [15, 17] which employ *through-silicon vias (TSVs)* to communicate between different layers and the substrate, a 2.5D IC package contains TSVs only in their interposers. Due to the lower fabrication cost and design complexity, many vendors, including ASE, eSilicon, and GlobalFoundries, have adopted 2.5D IC packages as their next-generation solutions for various applications [2, 8, 12].

In order to improve circuit performance, some optimization techniques, including *wire sizing* and *wire spacing*, are proposed to meet the timing and/or power constraints. However, routing complexity increases dramatically due to the variable widths and spacings.

Furthermore, in a typical 2.5D IC package, connecting multiple bump pairs on different chips often involves long and parallel inter-chip nets, as shown in Figure 2(a). The parallel inter-chip nets may induce severe coupling effects between signals, which degrades signal integrity and circuit performance. Several methods have been developed to mitigate the coupling effects, including shielding, spacing, and other decoupling structures [13]. In this paper, we first focus on the popular industrial practice, where signals are shielded by power/ground nets [2, 8], and then extend our work to more general structures.

This work was partially supported by AnaGlobe, IBM, MediaTek, Synopsys Inc, TSMC, MOST of Taiwan under Grant MOST 105-2221-E-002-190-MY3, MOST 106-2221-E-002-203-MY3, MOST 107-2221-E-002-161-MY3, MOST 108-2911-I-002-544, and MOST 108-2221-E-002-097-MY3

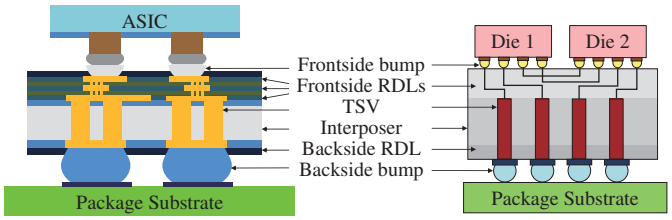

Figure 1: (a) Our problem considers a real 2.5D IC package design [2, 8, 12], which integrates multiple high bandwidth memories (HBMs) with an application-specific integrated circuit (ASIC). The problem is to connect pre-assignment nets among dies through frontside RDLs. (b) A general 2.5D IC package.

To better handle the routing problem on RDLs with various constraints, traditional combinatorial routing algorithms might not be sufficient for solving the problem effectively.

A. Previous Works

Figure 1(b) shows a general 2.5D IC package, which contains an interposer with inter-chip nets routed on RDLs. Traditional RDL routing algorithms can be divided into two categories: (1) mathematical programming and (2) graph-based methods.

For mathematical programming, Fang *et al.* [9] proposed an integer linear programming (ILP) based formulation which completes global routing with several reduction techniques to prune redundant solutions. Though this formulation can often obtain the optimal wirelength for global routing, the high time complexity of an ILP makes it prohibitive for solving large-scale designs.

For graph-based methods, the works [10, 11, 18] adopted network-flow-based methods to deal with RDL routing. Liu *et al.* [18] exploited the geometrical properties of Voronoi diagrams to model global-routing channels and applied a network-flow-based algorithm to solve the routing problem. Based on network-flow-based formulations, flip-chip routing was handled by Fang *et al.* in [10] and also package-board co-design by Fang *et al.* in [11]. These methods can solve the global-routing problems in reasonable runtime. However, they cannot directly apply to designs with variable wire widths and spacings.

To handle designs with non-uniform wire widths and spacings, gridless routing is needed due to its higher flexibility. In the work [7], a V-shaped multilevel framework was proposed for full-chip gridless routing. This method first partitions a layout into an array of rectangular subregions, and then routes wires of different widths sequentially on those subregions. Without considering routing resources and net ordering, however, it may cause a detour when the routing resource has been occupied by other routed wires, as shown in Figure 2(b).

978-1-7281-4123-7/20/$31.00

In this paper, we propose a unified RDL routing framework based on *the modulus-based matrix splitting iteration method (MMSIM)* [3] and *bipartite matching*. With the unified framework, we can handle both grid-based and gridless designs while achieving high solution quality in wirelength, as shown in Figure 2(c). In particular, different from other gridless routing algorithms that treat grid-based routing as a special case, which may be an overkill, our unified framework exploits the property of the MMSIM and can solve both designs efficiently and elegantly.

B. Our Contributions

We summarize our main contributions as follows:

- We propose a general 2.5D IC package routing framework which can be applied to both gridless and grid-based routing problems. To the best of our knowledge, this is the first package router that solves both the problems in a unified framework which does not rely on the complex data structure of gridless routing to handle simpler grid-based problems.
- This paper is the first work in the literature to formulate a routing problem as a *linear complementarity problem (LCP)* and use the MMSIM to solve this converted problem for RDL routing. By converting a routing problem on RDLs to an LCP, our algorithm can route inter-chip nets with variable widths and spacings while minimizing wirelength.
- We present a bipartite matching method, which honors the rough positions computed by the MMSIM, to assign wires to predefined positions. We theoretically prove the optimality of this method for grid-based designs and empirically validate it by comparing our results with the bipartite matching method presented in [14] which obtains the optimal wirelength for their designs.
- Experimental results show that our algorithm is effective and efficient. Compared with the previous method based on grid-based benchmarks, our routing framework achieves a 68X speedup while maintaining 100% routability and the optimal wirelength. For gridless benchmarks, our framework achieves 100% routability with a 274X speedup, while the previous method cannot.
- Our framework is general and readily extends to other routing problems.

The remainder of this paper is organized as follows. Section II formulates the addressed problem and reviews the LCP and the MMSIM on which our framework is based. Section III details our proposed algorithm. Section IV shows the experimental results. Finally, the conclusion is given in Section V.

II. Preliminaries

In this section, we first formulate a new RDL routing problem, detail the corresponding design rules, and review the LCP and the MMSIM.

A. Problem Formulation

The 2.5D IC RDL routing problem addressed in this paper is different from those in previous works. We consider a new structure and additional issues. Here, we first give some terminologies and notations used throughout this paper:

- A *pre-assignment net* is a net with predefined bump assignments.
- A *track* is a horizontal line which can accommodate potential connections on an RDL. If a wire/connection is assigned to a track, this wire will be routed on the line segment of the track between two chips. For a grid-based design, a track is a predefined position on a horizontal layer. For a gridless design, a track is constructed according to the design rules and routing configuration, described in Section III-B.
- $T = \{t_1, t_2, \ldots, t_{|T|}\}$ is the set of constructed tracks.
- $C = \{c_1, c_2, \ldots, c_{|C|}\}$ is the set of chips mounted on an interposer.
- $N = \{n_1, n_2, \ldots, n_m\}$ is the set of pre-assignment nets for all frontside bump pairs, where m is the number of pre-assignment nets and n_i is the i-th connection between two chips. For easier presentation, we shall first focus on 2-pin nets, and then extend our proposed algorithm to general structures in Section III-D.
- $W^n = \{w_1^n, w_2^n, \ldots, w_m^n\}$ is the set of wire widths, where w_i^n is the given wire width of n_i. For grid-based designs, the wire widths are uniform.
- $W^g = \{w_1^g, w_2^g, \ldots, w_m^g\}$ is the set of shielding widths. For a pre-assignment net n_i, a power/ground net of width w_i^g is required to shield an upper or lower horizontal layer.
- l_g is the basic shielding width, where $w_i^g = w_i^n + l_g$.
- l_s is the minimum spacing between a wire and a power/ground net on the same layer. l_s is also the minimum spacing between wires in adjacent horizontal layers.

In this work, we first present our algorithm by following the industrial structure [2, 8] for easier presentation, and then extend our algorithm to more general structures in Section III-D. Figure 3(a) shows the structure of the considered RDL routing plane. Different from previous works, the structure contains two consecutive horizontal layers and one vertical preferred layer. The structure allows wires to be routed on either predefined positions with uniform wire widths, i.e., grid-based routing, or any real-valued positions with variable wire widths and

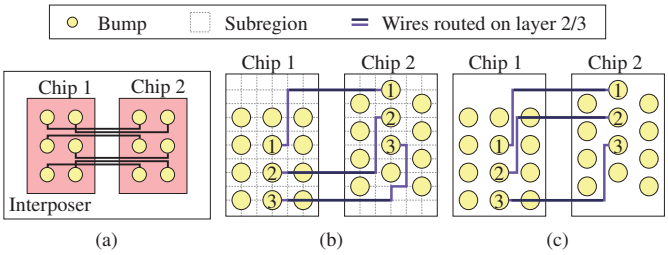

Figure 2: (a) Long and parallel inter-chip nets may cause severe coupling effects in a 2.5D IC package. (b) A result generated from the previous work [7] with reasonable extensions. The detour is caused by the two factors: (1) the layout is partitioned into rectangular subregions without considering the routing resources, and (2) the routing is performed sequentially so that the net ordering problem arises. (c) A better routing result with fewer detours and shorter wirelength from our routing framework.

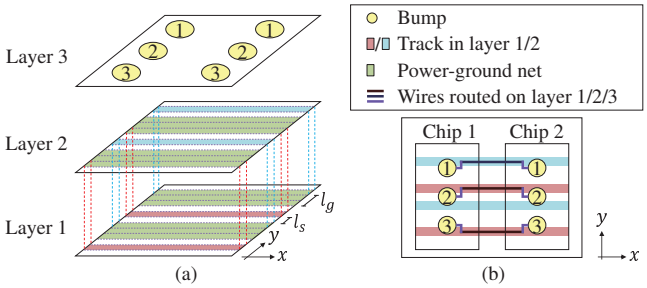

Figure 3: (a) Structure of the considered RDL routing plane. The bumps are distributed on layer 3, vertical wires and partial horizontal wire are routed on layer 3, and horizontal wires on tracks on layers 1 and 2, where power/ground nets are interleaved with the wires. (b) The top view of a routing result corresponding to (a).

spacings, i.e., gridless routing. Our objective is to develop a unified framework to handle both types of designs. Figure 3(b) shows the top view of a routing result satisfying all the design rules corresponding to Figure 3(a). Here, we formally define the problem as follows:

- **The Unified Pre-Assignment Multi-layer Multi-Chip RDL Routing Problem:** Given an RDL layout, a netlist of pre-assignment nets N, and design rules, connect all pre-assignment nets $n_i \in N$ such that there is no net crossing and the total wirelength is minimized.

B. Design Rules

A structure designed for mitigating coupling effects is adopted in this work. This considered RDL structure is different from those of previous works. Several specific design rules are required to reduce the design complexity and/or fabrication difficulties. We detail the major design rules of our problem as follows:

- To reduce coupling effects, wires routed on horizontal layers are interleaved with power/ground nets. A power/ground net of width w_i^g is required to shield a pre-assignment net on an upper or lower horizontal layer [2, 8].
- To achieve higher density while maintaining shielding, the y distance between two horizontal tracks on adjacent layers should be larger than the minimum spacing l_s.
- The distance between any two nets of the same layer should be larger than the minimum spacing l_s.
- Wires should be routed in the interposer region.

C. Review of LCP and MMSIM

Our proposed algorithm is based partly on solving the LCP by the MMSIM. Therefore, we shall give a brief review of the LCP and the MMSIM.

Given a real square matrix M and a real vector q, the LCP finds a real vector s which satisfies the following constraints:

$$Ms + q \geq 0, \quad s \geq 0, \text{ and } s^T(Ms + q) = 0. \tag{1}$$

The MMSIM is considered the most effective and efficient method for solving LCP [3]. The MMSIM for a given LCP(q,M) is defined as follows. Let $M = L - U$ be a splitting of the matrix M. Given an initial real vector $r^{(0)}$, we compute $r^{(k+1)}$ for $k = 0, 1, 2, \ldots$ by solving the following linear system until the iteration sequence $\{s^{(k)}\}_{k=0}^{+\infty}$ is convergent:

$$(L + \Omega)r^{(k+1)} = Ur^{(k)} + (\Omega - M)|r^{(k)}| - \delta q, \tag{2}$$

and set

$$s^{(k+1)} = \frac{1}{\delta}(|r^{(k+1)}| + r^{(k+1)}). \tag{3}$$

Here, Ω is a positive diagonal matrix, and δ is a positive constant. The optimal solution of LCP(q,M) can be derived with the convergence of the MMSIM, whose convergence conditions are proved in [3]; for example, matrices M and L must be *positive definite*.

III. Our Proposed Algorithm

In this section, we first present our routing algorithm for a popular industrial 2.5D IC package structure. We provide an overview of our algorithm, and then detail the methods used in each stage. Finally, we extend our algorithm to more general structures (e.g., with multi-pin nets, general shielding structures, etc.).

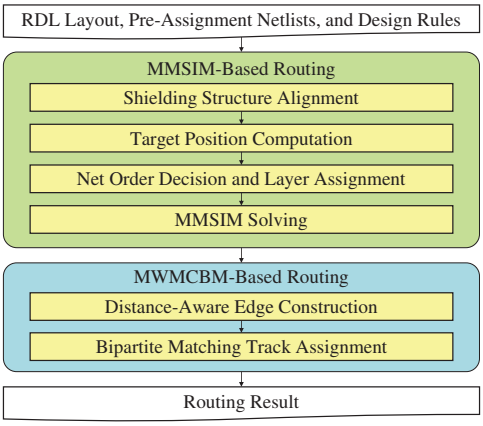

Figure 4: Flow of our proposed routing algorithm.

A. Algorithm Overview

Figure 4 summarizes our algorithm, which consists of two major stages: (1) MMSIM-based routing, and (2) minimum weighted maximum cardinality bipartite matching (MWMCBM) based routing.

First, we relax the routing problem as an LCP and use the MMSIM to solve this converted RDL routing problem. In particular, this MMSIM guarantees to converge to find an optimal solution. After MMSIM-based routing, we reformulate the problem as an MWMCBM one and perform MWMCBM-based routing to refine the solution. With the guidance of the routing result in the LCP/MMSIM stage, MWMCBM-based routing can solve the subproblems efficiently by reducing the number of edges in the constructed graph, while preserving an optimal solution. We theoretically prove that an optimal solution of the subproblem is globally optimal for a grid-based design.

The synergy between MMSIM- and MWMCBM-based routing provides an effective and efficient algorithm for both grid-based and gridless designs, without overdoing the problem. Specifically, the nice properties of the MMSIM mold the input design type for the MMSIM- and MWMCBM-based routing to achieve a better final solution.

B. MMSIM-Based Routing

The unified RDL routing problem is first relaxed as a constrained quadratic programming one. Then, we convert the relaxed problem into an LCP and solve it by the MMSIM to obtain a desired solution (e.g., an optimal solution for a grid-based design).

The objective of our routing problem is to route nets such that the total routed wirelength is minimized while the design rules are satisfied. We apply a combination of L- and Z-shaped routes to make a connection in the horizontal range of each bump pair for higher flexibility to a dense-bump design. For a pre-assignment net n_i, we define the coordinates of two involving bumps as (x_i^α, y_i^α) and (x_i^β, y_i^β), where a combined pattern route gives the shortest horizontal path length $|x_i^\alpha - x_i^\beta|$ for connecting the bump pair. Therefore, the problem can be converted as assigning pre-assignment nets to tracks such that the total vertical distance from each bump pair to the assigned

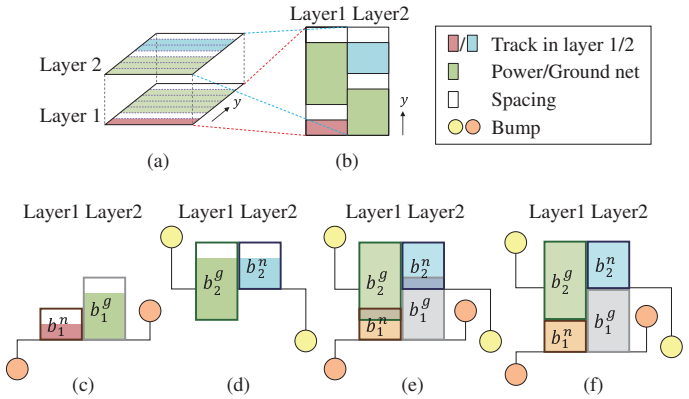

Figure 5: (a) A desired routing structure. (b) The projection to the column representation of (a). (c)(d) For the pre-assignment nets assigned on layers 1 and 2, the shielding blocks are aligned with the bottom and the top of the track blocks, respectively. (e) An initial result by combining (c) and (d). (f) A feasible solution after reserving sufficient spacing of (e).

track is minimized, which can be formulated as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} |y_i - y_i^{\alpha}| + |y_i - y_i^{\beta}| \\ \text{s.t.} \quad & \text{a) } n_i \text{ is shielded by a power/ground net of width } w_i^g, \\ & \text{b) net spacing on the same layer} \geq l_s, \\ & \text{c) track spacing on different layers} \geq l_s, \\ & \text{d) wires are routed in the interposer region,} \end{aligned} \tag{4}$$

where y_i is the y coordinate of the track to which the pre-assignment net n_i is assigned. Note that doglegs in the considered structure incur bends and lower the routability, so we apply combined patterns without doglegs.

Since the objective in Equation (4) is neither smooth nor differentiable, we relax the objective function and the top boundary condition to a constrained quadratic programming problem as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} (y_i - y_i^{\alpha})^2 + (y_i - y_i^{\beta})^2 \\ \text{s.t.} \quad & \text{a) } n_i \text{ is shielded by a power/ground net of width } w_i^g, \\ & \text{b) net spacing on the same layer} \geq l_s, \\ & \text{c) } |y_i - y_j| \geq l_s, \ \forall i \neq j \in \{1, 2, \ldots, m\}, \\ & \text{d) } y \geq 0. \end{aligned} \tag{5}$$

A desired routing structure is shown in Figure 5(a). We project the horizontal layers into a column representation shown in Figure 5(b), where tracks, power/ground nets, and spacings are considered as rectangles in columns, and the width of each instance is transformed into the height of the rectangle in a column.

As the minimum spacing l_s is required between every pair of nets, we cluster a net with a spacing as a *block*. For each pre-assignment net n_i, two blocks are created, a *track block* b_i^n and a *shielding block* b_i^g. The track block b_i^n is a block clustered by a spacing and the track for the pre-assignment net n_i, and the shielding block b_i^g is a cluster of a spacing and the power/ground net that shields n_i on the upper or lower horizontal layer.

We introduce *target positions* p_i^n and p_i^g for b_i^n and b_i^g, respectively. A target position is a desired location of a block such that the objective in Equation (5) is minimized, and the desired structure is formed. A desired structure is shown in Figure 5(b), where the wires are interleaved and aligned with the power/ground nets on corresponding layers. The target position p_i^n for a track block b_i^n is computed by minimizing the objective function in Equation (5), while p_i^g for a shielding block b_i^g by aligning it with the bottom and the top of b_i^n for the pre-assignment net n_i assigned on layers 1 and 2, respectively, as shown in Figure 5(c) and 5(d).

We summarize them by the following equations:

$$\begin{aligned} p_i^n &= \frac{1}{2}(y_i^{\alpha} + y_i^{\beta}), \\ p_i^g &= \begin{cases} p_i^n, & \text{if } n_i \in \text{layer 1}, \\ p_i^n - l_g, & \text{if } n_i \in \text{layer 2}, \end{cases} \end{aligned} \tag{6}$$

where $n_i \in$ layer j represents that the pre-assignment net n_i is assigned to a track on layer j.

We define the *sequence order* as the order of pre-assignment nets in a nondecreasing y coordinate order of the assigned tracks. In order to meet the convergence requirements of the MMSIM, the sequence order of pre-assignment nets has to be fixed. Since the target position p_i^n is representative for the bump pair locations y_i^{α} and y_i^{β}, we set the sequence of pre-assignment nets by the order of target positions. For those nets with same target positions, their sequence order can be decided by nondecreasing vertical distances to the target position. With the defined sequence order, by assigning track blocks to horizontal layers alternately and shielding blocks to the corresponding shielding layer, a desired structure can be obtained, where power/ground nets are interleaved with the tracks.

Figure 5(e) shows the result after placing blocks to their target positions and corresponding layers. However, the result may contain overlaps between blocks. By reserving sufficient spacing for adjacent blocks in the sequence order, we can obtain a feasible solution, as shown in Figure 5(f). Therefore, Equation (5) can be reformulated as follows:

$$\min \quad \sum_{i=1}^{m} (y_i - p_i^n)^2 + \sum_{i=1}^{m} (z_i - p_i^g)^2 \tag{7a}$$

$$\text{s.t.} \quad z_{i+1} - y_i \geq w_i^n + l_s, \qquad \forall i \leq m-1, \tag{7b}$$

$$y_{i+1} - z_i \geq w_i^g + l_s, \qquad \forall i \leq m-1, \tag{7c}$$

$$z_i = y_i, \qquad \forall n_i \in \text{ layer 1}, \tag{7d}$$

$$z_i = y_i - l_g, \qquad \forall n_i \in \text{ layer 2}, \tag{7e}$$

$$y \geq 0, \ z \geq 0, \tag{7f}$$

where y_i and z_i are the y coordinates of b_i^n and b_i^g, respectively.

Constraints (7b) and (7c) are for reserving sufficient spacing for adjacent blocks. The former is for each track block b_i^n of width $w_i^n + l_s$, and the latter for each shielding block b_i^g of width $w_i^g + l_s$.

Constraints (7d) and (7e) are for block alignments. The former aligns shielding blocks with the bottom of track blocks, while the latter aligns with the top. According to the target positions from Equation (6), by aligning blocks with Constraints (7d) and (7e), we have that the second term is identical to the first term in Equation (7a). As a result, Equation (7) is equivalent to Equation (5) with the decided sequence order.

In fact, to satisfy the shielding condition, aligning one layer is sufficient. The spacing satisfies the design rules if one layer is aligned, so we can remove Constraint (7e). As a result, we can rewrite Equation (7) as follows:

$$\min \quad \frac{1}{2} x^T I x + p^T x \tag{8a}$$

$$\text{s.t.} \quad Hx \geq h, \tag{8b}$$

$$Dx = 0, \tag{8c}$$

$$x \geq 0, \tag{8d}$$

where $x = \begin{bmatrix} y \\ z \end{bmatrix}$, $p = \begin{bmatrix} -p^n \\ -p^g \end{bmatrix}$. I is an identity matrix, and H, h, and D are defined as follows. H is the constraint matrix for Equations (7b) and (7c), and h is the corresponding vector. Each row in H contains only two nonzero elements 1 and -1 either in b^g_{i+1} and b^n_i or in b^n_{i+1} and b^g_i, which corresponds to the height of b^n_i or b^g_i in h. D is the constraint matrix for Equation (7d). Each row also contains only two nonzero elements 1 and -1 in b^g_i and b^n_i, respectively, for n_i assigned to layer 1. Here gives an example of the matrices for Figure 5(f):

$$H = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 \end{bmatrix}, \; h = \begin{bmatrix} w^n_1 + l_s \\ w^g_1 + l_s \end{bmatrix}, \; D = \begin{bmatrix} 1 & 0 & -1 & 0 \end{bmatrix}$$

To ensure the MMSIM convergence, the constraint matrix is required to be a rectangular matrix of full row rank. Hence, we introduce a penalty factor λ for relaxing Constraint (8c) to the objective function of Equation (8). As long as λ is large enough, the corresponding blocks will be aligned. Then the equation can be reformulated as follows:

$$\begin{array}{ll} \min & \frac{1}{2}x^T I x + p^T x + \lambda x^T D^T D x \\ \text{s.t.} & Hx \geq h, \\ & x \geq 0. \end{array} \tag{9}$$

Proposition 1. *In Equation (9), $I + \lambda D^T D$ is a symmetric positive definite matrix, and H is a rectangular matrix of full row rank.*

Proof. Since I is an identity matrix and $D^T D$ is a positive semi-definite matrix, $I + D^T D$ is apparently symmetric and positive definite. H is a $(2m-2) \times 2m$ rectangular matrix. In matrix H, since there are only two nonzero elements in each row and at most two nonzero elements in each column, we can simply transform H to the row echelon form and derive an identity matrix with $2m-2$ pivots in the columns of b^n_i and b^g_i for $i \leq m-1$. Hence, H in Equation (9) is of full row rank. □

Since $I + D^T D$ is a symmetric positive definite matrix, by the Karush-Kuhn-Tucker (KKT) condition [21], x is the optimal solution of Equation (9) if and only if there exist vectors u, σ, and τ satisfying the following KKT conditions:

$$\begin{cases} \sigma = Ix + p + \lambda D^T D x - H^T u, \\ \tau = Hx - h, \\ \sigma^T x = 0, \\ u^T \tau = 0, \\ x, u, \sigma, \tau \geq 0. \end{cases} \tag{10}$$

We can rewrite Condition (10) as the following LCP(q,M):

$$Ms + q \geq 0, \quad s \geq 0, \text{ and } s^T(Ms + q) = 0, \tag{11}$$

where

$$M = \begin{bmatrix} I + \lambda D^T D & -H^T \\ H & 0 \end{bmatrix}, \; s = \begin{bmatrix} x \\ u \end{bmatrix}, \; q = \begin{bmatrix} p \\ -h \end{bmatrix}.$$

By the work [19], the optimal solution of the constrained quadratic programming problem (9) gives the solution of the LCP (11), and vice versa. Hence, we obtain the solution of Problem (11) by using the MMSIM to solve the LCP of Equation (11). We choose the splitting matrix L and U as the parameterized Uzawa splitting of the saddle point in matrix M [4, 5] as follows:

$$L = \begin{bmatrix} \frac{1}{\gamma}(I + \lambda D^T D) & 0 \\ H & \frac{1}{\theta}A \end{bmatrix}, U = \begin{bmatrix} (\frac{1}{\gamma} - 1)(I + \lambda D^T D) & H^T \\ 0 & \frac{1}{\theta}A \end{bmatrix} \tag{12}$$

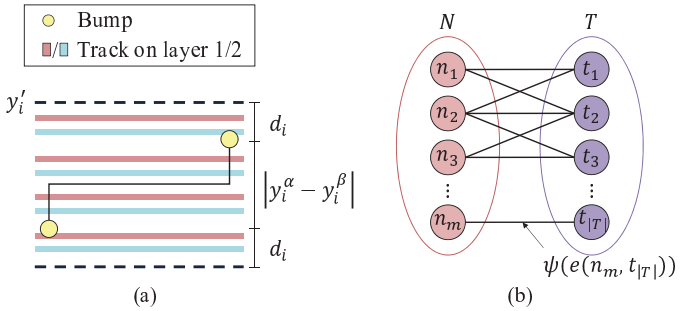

Figure 6: (a) Excessive distance d_i is the distance that y'_i exceeds the y coordinate range of the bump pair. (b) Bipartite graph construction. An edge between a pre-assignment net and a track is introduced if the track is in the range of the maximum excessive distance.

where $A = tridiag(H(I + \lambda D^T D)^{-1} H^T)$ is a tridiagonal approximation to the Schur complement $H(I + \lambda D^T D)^{-1} H^T$ of the matrix M, and γ and θ are two positive constants determined according to the formulas given in [5]. As the computation of inverse matrices could be time-consuming, we use the Sherman07-Morrison formula [20] to obtain the matrix A. Then A can be computed by $tridiag(HH^T - \frac{\lambda}{2\lambda+1} HD^T(HD^T)^T)$. Since matrix $I + \lambda D^T D$ is a positive definite matrix and H is a rectangular matrix of full row rank, the convergence of MMSIM can be assured [3].

Based on the MMSIM for LCP(q,M) in [3], we can derive the optimal solution of Equation (9) by solving the linear system shown in Equations (2) and (3). We apply the MMSIM solver in [6].

After the solution is derived, we make some adjustment to meet the design rules, e.g. shift tracks within the interposer boundary. For a grid-based design, the positions are mapped to the nearest tracks to obtain a feasible solution. For a gridless design, tracks are constructed to the corresponding positions. Then we derive a solution y' that satisfies all the design rules.

C. MWMCBM-Based Routing

After the MMSIM-based routing is completed, we reformulate the problem as a graph matching subproblem and perform an MWMCBM-based routing to refine the solution. Our MWMCBM-based router supports local refinement and obtains an optimal solution for the subproblem. We show that such an optimal solution of a subproblem is globally optimal for a grid-based design.

Given the set N of pre-assignment nets and the set T of tracks, a bipartite graph $G = (V, E)$ can be constructed, where $V = N \cup T$ and E is the set of edges connecting $n_i \in N$ and $t_j \in T$ for all n_i and t_j with the same width. The goal is to minimize the objective function as follows:

$$\min \sum_{i=1}^{m} |y_i - y^\alpha_i| + |y_i - y^\beta_i|, \tag{13}$$

where y_i is the y coordinate of the track matched to n_i. Hence, we set the cost ψ of the edge $e(n_i, t_j)$ as follows:

$$\psi(e(n_i, t_j)) = \frac{1}{2}(|y_i - y^\alpha_i| + |y_i - y^\beta_i| - |y^\alpha_i - y^\beta_i|), \tag{14}$$

which implies the distance exceeding the y coordinate range between the two involving bumps of n_i. Note that this formulation matches the overall objective function in Equation (4).

Theorem 1. *An optimal solution for Problem (13) is a globally optimal solution for a grid-based design.*

Proof. For a grid-based design, the tracks are constructed on all the predefined positions. Since the wire widths of all pre-assignment nets are uniform, G is constructed as a complete bipartite graph with $|N||T|$ edges. Hence, we can derive the exact minimum cost for Problem (13) by applying minimum weighted maximum cardinality bipartite matching (MWMCBM), which is an optimal solution for a grid-based design. □

Though an optimal solution for Problem (13) can be obtained by applying MWMCBM, the number of edges is $O(|N||T|)$, which grows rapidly when the input size increases. As a result, to speed up the process while maintaining the solution quality, we consider connecting pre-assignment nets only to those tracks with the same width and close enough to the solution obtained in Section III-B.

According to Equation (14), for each pre-assignment net n_i, we introduce the *excessive distance* d_i as the distance that y_i' exceeds the y coordinate range of the bump pair, as shown in Figure 6(a). To minimize wirelength, we connect edges between tracks and pre-assignment nets if the tracks are within the range shown in Figure 6(a). By uniforming d_i, we can guarantee the optimality for the given matching problem (13) if G has a matching of size $|N|$. Hence, we set the uniform excessive distance $d^* = \max_i d_i$. Corresponding to a feasible solution in the previous stage, G has a matching of size $|N|$ by connecting edges within the fixed excessive distance d^*, implying that edges are constructed if the edge costs satisfy the following equations:

$$\begin{aligned} \psi(e(n_i, t_j)) &\leq \max_i d_i = d^*, \\ d_i &= \tfrac{1}{2}(|y_i' - y_i^\alpha| + |y_i' - y_i^\beta| - |y_i^\alpha - y_i^\beta|). \end{aligned} \tag{15}$$

Note that since an approximate solution is computed in Section III-B, the edge number will be significantly reduced. Finally, according to the matching result, we can connect the bumps to the assigned horizontal tracks from the bump distributed layer by L-shaped routes with shifting and bending if necessary.

Here, we prove the optimality for Problem (13) with edge reduction.

Theorem 2. *In Problem (13), if G has a matching of size $|N|$ by connecting edges within a fixed excessive distance d^*, we can guarantee an optimal solution.*

Proof. MWMCBM can be implemented by a minimum cost maximum flow algorithm [1, 16]. By finding the shortest path for $|N|$ iterations, we can derive a solution. For $d^* = 0$, we build an edge for each track between y_i^α and y_i^β, with a zero edge cost. As d^* increases, the new edges all have the cost $d^* + \varepsilon$, where ε is a positive variable. If we can match all $n_i \in N$ within an excessive distance d^*, any of the new constructed edges of the cost $d^* + \varepsilon$ will not be chosen by the shortest path. Hence, we can derive an optimal solution if we can match all $n_i \in N$ within a fixed excessive distance d^*. □

By Theorems 1 and 2, our algorithm can obtain an optimal solution for a grid-based design when no doglegs and detours are required. For designs with limited routing resource, our algorithm can be integrated with some commonly used detailed routing techniques (e.g., A^* search) to handle congested areas. The techniques presented in this paper are sufficient to solve the RDL routing problem on current industrial designs [2, 8].

D. Extensions to General Structures

For easier presentation, we presented our algorithm based on an industrial structure. Here, we extend our algorithm to more general structures.

Circuits	$\|C\|$	#layers	#pins	$\|N\|$
design1	2	3	173408	1726
design2	2	3	116518	1748
design3	2	3	252514	3304
design4	2	3	573426	6944
design5	3	3	715843	7664
design6	3	3	1074362	12188

Table I: Benchmark Statistics

- For a general 2.5D IC package structure as in [14], inter-chip connections among different chips are connected through I/O buffers. For this structure, we can simply replace the vias in the bump distributed layer with I/O buffers.
- In this work, the coupling effects are mitigated by a shielding structure. However, there are many popular alternatives to deal with this issue such as spacing. A coupling violation occurs when the coupling amount of two wires exceeds a threshold k_{cp}. As a result, we can compute the decoupling spacing from the following equation:

$$k_{cp} = \frac{overlap\ wirelength}{spacing}. \tag{16}$$

  For pre-assignment nets n_i and n_j, the decoupling spacing is computed as l_{ij}. We extend our algorithm to deal with various spacings by modifying the spacing clustered by track blocks. We can remove the shielding blocks and replace l_s with the corresponding l_{ij} in Equation (7).
- For the structure without consecutive horizontal or vertical layers, we can simply remove the constraints for alignment and revise the spacing.
- As mentioned earlier, our presented structure was based mainly on a set of industrial benchmarks provided to us, which contain only 2-pin nets. In a modern design, however, a circuit might contain multi-pin nets. For each multi-pin net, we can compute a desired track position and connect bumps to the assigned track. Unlike a 2-pin connection, a multi-pin net connects more bumps to the assigned track.
- For most industry applications, a 2.5D IC package is composed of a central chip and peripheral ones, and contains only tracks in either the x or y direction. Our proposed algorithm readily extends to this type of 2.5D IC packages. However, there are also other types of multiple chip connections. For a 2.5D IC package with multiple chips, more layers will be introduced for vertical connections. As a result, we can apply our routing framework for the x and y directions, one by one. In this paper, we focus on minimizing the vertical distance from each bump pair to its assigned track for the y direction. By applying our framework to the horizontal direction as well, multiple chips can be connected.

IV. Experimental Results

We implemented our proposed routing algorithm in the C++ programming language and used LEDA as our graph-based algorithm solver. All the experiments were performed on an Intel Xeon 2.93GHz Linux workstation with 48GB memory. We compared our algorithm with the grid-based router presented in [14] and the gridless router presented in [7] based on six industrial benchmarks. Table I lists the statistics of the benchmark circuits. "$|C|$", "#layers", "#pins", and "$|N|$" denote the numbers of chips, layers, pins, and pre-assignment nets, respectively.

Circuits	Routability (%)		Total Wirelength (μm)		Runtime (sec.)	
	Bipartite	Ours	Bipartite	Ours	Bipartite	Ours
design1	**100.0**	**100.0**	**5611732**	**5611732**	5.17	**0.20**
design2	**100.0**	**100.0**	**5656492**	**5656492**	5.88	**0.21**
design3	**100.0**	**100.0**	**10690142**	**10690142**	27.09	**0.57**
design4	**100.0**	**100.0**	**22470260**	**22470260**	120.19	**1.80**
design5	**100.0**	**100.0**	**24795361**	**24795361**	145.13	**2.13**
design6	**100.0**	**100.0**	**39439739**	**39439739**	358.54	**4.80**
Comp.	**1.00**	**1.00**	**1.000**	**1.000**	68.18	**1.00**

Table II: Comparisons of grid-based RDL routing results.

Circuits	Routability (%)		Total Wirelength (μm)		Runtime (sec.)	
	VMGR	Ours	VMGR	Ours	VMGR	Ours
gl_design1	**100.0**	**100.0**	5609250	**5607762**	12.28	**0.27**
gl_design2	**100.0**	**100.0**	5655958	**5649798**	13.16	**0.24**
gl_design3	**100.0**	**100.0**	10704690	**10677131**	56.38	**0.61**
gl_design4	**100.0**	**100.0**	22479864	**22442314**	278.16	**1.90**
gl_design5	94.5	**100.0**	N/A	**24780221**	671.07	**2.24**
gl_design6	89.8	**100.0**	N/A	**39414675**	1782.77	**4.98**
Comp.	0.97	**1.00**	1.002	**1.000**	274.79	**1.00**

Table III: Comparisons of gridless RDL routing results.

A. Grid-Based Designs

For grid-based designs, we compared our algorithm with the 2.5D IC package router presented in [14]. The algorithm proposed in [14] constructs a complete bipartite graph between bumps and I/O buffers in the same chip. We extended the work [14] by modifying I/O buffers to the constructed tracks to handle our designs. It is known that bipartite matching can obtain an optimal solution of the minimum weight. The second stage of our algorithm also performs bipartite matching. With the guidance of an approximate solution obtained in the first stage, however, we can significantly reduce the number of edges while preserving the optimality. The experimental results are shown in Table II, where the routability, the total wirelength, and the runtime are reported. From the results, our algorithm runs 68X faster than the bipartite matching algorithm alone and obtains optimal solutions for grid-based designs.

B. Gridless Designs

For gridless designs, we also performed experiments on the benchmark circuits listed in Table I. We modified the original circuits of uniform wire widths to generate a set of circuits of non-uniform wire widths by using the design rules from industry, where 20% pre-assignment nets were widened to 150%, 200%, or 250% the original widths.

We compared our algorithm with a multilevel gridless routing algorithm, namely VMGR proposed by [7]. VMGR first partitions a chip into an array of rectangular subregions, and then routes each net by the evaluated channel density for its uncoarsening and coarsening stages. VMGR performs pattern routing during uncoarsening and maze routing for the failed connections during coarsening. In order to handle our designs, we extended the work [7] by modifying the rectangular subregions to avoid doglegs in the inter-chip area. Also, to enhance the routability of VMGR, we added a combined pattern of an L- and a Z-shaped route to the pattern routing during uncoarsening. The experimental results are shown in Table III, where "N/A" represents incomplete routing results. From the results, our algorithm outperforms VMGR in routability, wirelength, and runtime. The experimental results reveal the effectiveness of our routing framework for both grid-based and gridless designs of 2.5D IC packages.

V. Conclusions

In this paper, we have proposed a unified routing framework that can handle both grid-based and gridless designs, which does not rely on the complex data structure of a gridless router to handle simpler grid-based problems. Also, this paper is the first in the literature to formulate a routing problem as an LCP and solve the LCP with the MMSIM. The optimality of our framework for grid-based routing problems have been theoretically proved and also empirically validated. By integrating the bipartite matching method with the MMSIM, our framework for grid-based designs achieves a 68X speedup while maintaining 100% routability and the optimal wirelength. For gridless benchmarks, our framework achieves 100% routability with a 274X speedup, while the previous method cannot. The results have demonstrated the high quality and efficiency of our framework. Furthermore, our framework is general and readily extends to other routing (and some quadratic optimization) problems.

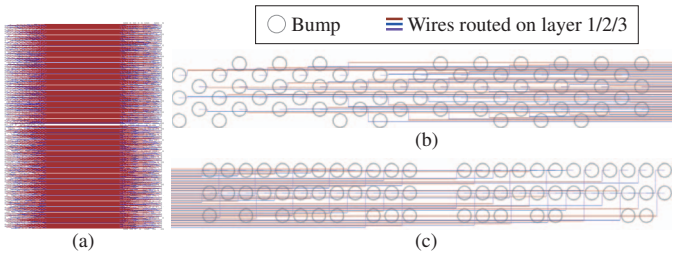

Figure 7: (a) The grid-based RDL routing solution of design2. (b)(c) Partial layouts from left and right side of (a).

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., 1993.

[2] ASE Group, 2.5D IC. http://ase.aseglobal.com/en/technology/advanced_25dic.

[3] Z.-Z. Bai. Modulus-based matrix splitting iteration methods for linear complementarity problems. *Numerical Linear Algebra with Applications*, 17:917–933, 2010.

[4] Z.-Z. Bai, B. N. Parlett, and Z.-Q. Wang. On generalized successive overrelaxation methods for augmented linear systems. *Numerische Mathematik*, 102(1):1–38, November 2005.

[5] Z.-Z. Bai and Z.-Q. Wang. On parameterized inexact Uzawa methods for generalized saddle point problems. *Linear Algebra and its Applications*, 428(11):2900 – 2932, 2008.

[6] J. Chen, Z. Zhu, W. Zhu, and Y.-W. Chang. Toward optimal legalization for mixed-cell-height circuit designs. In *Proc. of ACM/IEEE DAC*, pages 1–6, June 2017.

[7] T.-C. Chen, Y.-W. Chang, and S.-C. Lin. A novel framework for multilevel full-chip gridless routing. In *Proc. of IEEE/ACM ASP-DAC*, pages 636–641, January 2006.

[8] eSilicon, Inc, 2.5D/HBM2 Packaging and Solutions. https://www.esilicon.com/capabilities/custom-2-5d-3d-packaging/.

[9] J.-W. Fang, C.-H. Hsu, and Y.-W. Chang. An integer-linear-programming-based routing algorithm for flip-chip designs. In *IEEE TCAD*, volume 28, pages 98–110, January 2009.

[10] J.-W. Fang, I.-J. Lin, Y.-W. Chang, and J.-H. Wang. A network-flow-based RDL routing algorithm for flip-chip design. In *IEEE TCAD*, volume 26, pages 1417–1429, August 2007.

[11] J.-W. Fang, M. D. F. Wong, and Y.-W. Chang. Flip-chip routing with unified area-i/o pad assignments for package-board co-design. In *Proc. of ACM/IEEE DAC*, pages 336–339, July 2009.

[12] GLOBALFOUNDRIES, Inc. https://www.globalfoundries.com/.

[13] T.-Y. Ho, Y.-W. Chang, S.-J. Chen, and D.-T. Lee. Crosstalk- and performance-driven multilevel full-chip routing. In *IEEE TCAD*, pages 869–878, June 2005.

[14] Y.-K. Ho and Y.-W. Chang. Multiple chip planning for chip-interposer codesign. In *Proc. of ACM/IEEE DAC*, pages 1–6, May 2013.

[15] D. H. Kim, K. Athikulwongse, and S. K. Lim. A study of Through-Silicon-Via impact on the 3D stacked IC layout. In *Proc. of IEEE/ACM ICCAD*, pages 674–680, November 2009.

[16] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97, 1955.

[17] X. Liu, Y. Zhang, G. Yeap, and X. Zeng. An integrated algorithm for 3D-IC TSV assignment. In *Proc. of ACM/IEEE DAC*, pages 652–657, June 2011.

[18] X. Liu, Y. Zhang, G. K. Yeap, C. Chu, J. Sun, and X. Zeng. Global routing and track assignment for flip-chip designs. In *Proc. of ACM/IEEE DAC*, pages 90–93, June 2010.

[19] J. Nocedal and S. Wright. *Numerical Optimization*. New York: Springer, 2006.

[20] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes*. Cambridge University Press, 2007.

[21] B. Stephen and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[22] M. Sunohara, T. Tokunaga, T. Kurihara, and M. Higashi. Silicon interposer with TSVs (Through Silicon Vias) and fine multilayer wiring. In *Proc. of ECTC*, pages 847–852, May 2008.