

Fig. 1. Top view of wire-bond package substrate design.

sion for the connection between them can be made to improve routability and alleviate via use. Therefore, we fine-tune the tree topology and try to route them in the same layer again under maintenance of the tree property if the original connection is unsuccessfully routed. Experimental results based on 6 industrial designs demonstrate that our framework can complete the connection of multi-pin nets under better routability.

The rest of this paper is organized as follows. Section II formulates the FBGA package routing problem into a minimum-cost multiple commodity flow problem. Section III details our routing framework. Section IV demonstrates the experimental results. The conclusion is presented in section V.

II. PRELIMINARIES

In this section, we first review the MCMCF model used in the FBGA package routing problem. It is related to our proposed framework. We then present the problem formulation of multi-pin net routing in the FBGA package.

A. Review of MCMCF for FBGA Substrate Routing

One methodology to complete FBGA package routing is to model it as a minimum-cost multi-commodity flow problem. The work [9] constructs a square lattice grid graph with wire pitch and splits the multi-pin nets into different two-pin nets with the bipartite matching method. For example, a multi-pin net is divided into 2 different nets if it has 3 bonding fingers and 2 bump balls. In order to make the wire bonder work correctly, a keep-out zone is built by the area of the finger and bump ball, respectively. Then, the constraints are formulated to obtain the feasible solution by ILP solver [10]. The following describes how to convert the routing problem to a multi-commodity flow model in the work [9].

First of all, based on the concept of MCMCF, the objective function is to minimize the total wire length used by each route. There are 3 situations of flow capacity at each vertex in a square lattice grid graph. If the vertex is a bonding finger, it is defined as the source. The fan-out flow minus fan-in flow must equal 1. Figure 2(a) draws that 8 green arrows represent all possible fan-out flows. The fan-out flow must be 1, and the fan-in flow must be 0 under the connectivity constraint. Second, if the vertex is a bump ball, it is defined as a target, and the

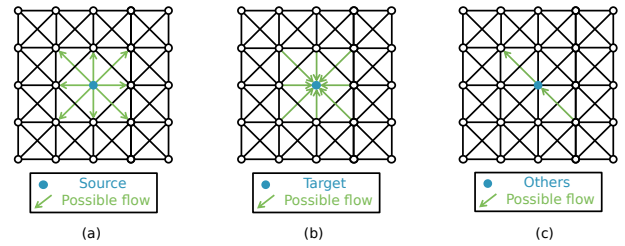


Fig. 2. Connectivity constraints [9].

fan-out flow minus fan-in flow must equal -1 . It is depicted in Figure 2(b), 8 green arrows represent all possible fan-in flows. The fan-out flow must be 0, and the fan-in flow must be 1. An instance of the last case is shown in Figure 2(c). All 8 flows can be used (including fan-in and fan-out flow), while the use of fan-in flow must equal to the fan-out flow. That is, if the use of fan-in flow is 1, then the use of fan-out flow must equal 1. For other vertices, both flows equal 0 for flow conservation. In addition, a short circuit constraint was formulated to prevent two different nets from crossing each other. For via use, constraints are also set up to avoid multiple different routes from passing through the same via and causing design rule violation. Although the work [9] can effectively obtain the routing solution of two-pin nets routing, it mainly focuses on two-pin nets routing. It is unable to take multi-pin nets into account, which implies that the work [9] does not make proper consideration for the part of multi-pin nets. These multi-pin nets are regarded as different and independent two-pin nets, which leads to overuse of routing resource and make worse routability. Therefore, we release the ILP constraints applied to the keep-out zone based on the router [9] and take strategies to complete routing for multi-pin nets.

B. Problem Formulation

Given the substrate designs and corresponding design rules, fingers, bump balls, and netlist. Our objective is to maximize the completion rate of multi-pin nets. Given N nets, including n two-pin nets and m multi-pin nets. P_{m_i} denotes the number of pins in multi-pin net m_i . We build the routing tree to route for each m_i . It contains $P_{m_i} - 1$ edges as routes. We aim to maximize the completion rate of successful routes to $P_{m_i} - 1$ for each multi-pin net m_i .

III. PROPOSED ROUTING ALGORITHM

In this section, we first describe our routing framework and then elaborate on the strategies used in each stage.

A. Algorithm Overview

Figure 3 illustrates our proposed framework. It mainly consists of 3 stages, the pin grouping, the minimum spanning tree topology generation, and the group topology connection. Given the design rules, netlist, fingers, and bump balls, we first group the pins of each multi-pin net. The neighboring bonding fingers or bump balls are grouped together to minimize the

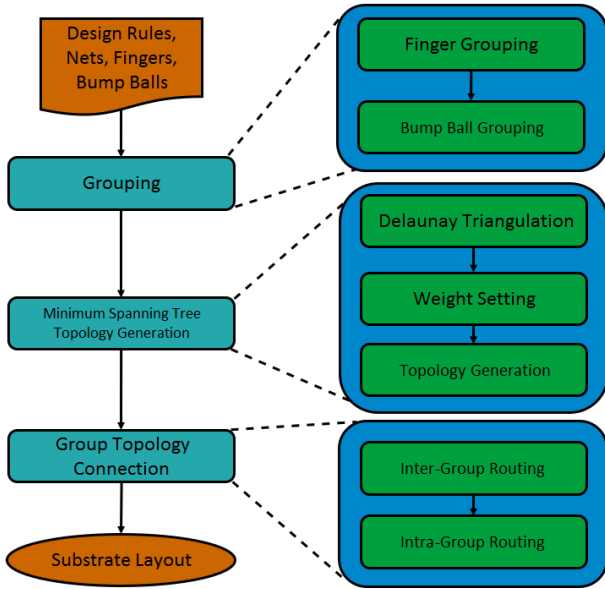


Fig. 3. Overview of the proposed framework.

required routing resource. Therefore, we identify the pins adjacent to each other with pin grouping. In the second stage, the minimum spanning tree is generated as our routing tree based on the Delaunay triangulation result to determine inter-group and intra-group. In the last stage, different strategies are used to complete inter-group and intra-group routing, respectively.

B. Pin Grouping

Since a multi-pin net requires more routing resource than a two-pin net, multi-pin nets have a greater impact on the final routing result. Typically, neighboring pins with shorter pin distances require less routing resource. Therefore, for each multi-pin net, we group neighboring pins of a multi-pin net to alleviate the impact on routing resource. In finger grouping, we first define a constant value α and then calculate the distance between bonding fingers. If the distance between them is less than or equal to α , it is regarded as the same group. As shown in Figure 4, the fingers of net i are marked in yellow and green for net i . The others marked in gray color represent other nets. In this example, nets i and j form 2 groups. For the bump ball case, we define that the distance between the bump balls is less than or equal to $\sqrt{2}$ ball pitch, and they belong to the same net as the same group. As shown in Figure 5, the bump balls of net i are marked in yellow, and the others are marked in gray. In this example, the bump balls labeled yellow are regarded as the same group. Pin grouping results can help us discern intra-group information during the minimum spanning tree topology generation stage in our framework.

C. Minimum Spanning Tree Topology Generation

In this stage, Delaunay triangulation (DT) is performed for each multi-pin net, and the edge weights are set for generating the minimum spanning tree (MST). In work [9], it maps fingers to corresponding bump balls based on a minimum-cost

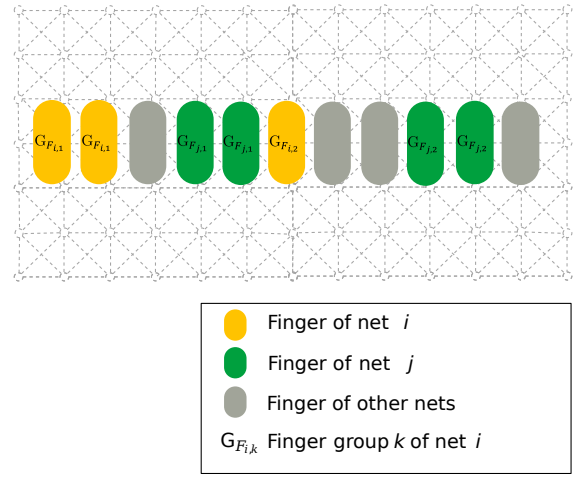


Fig. 4. Finger grouping.

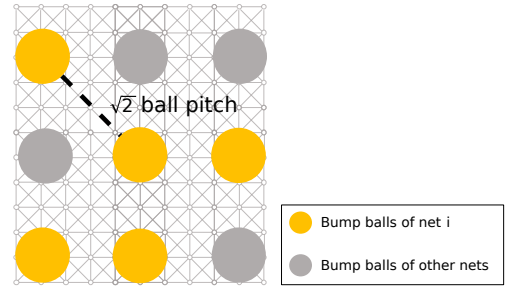


Fig. 5. Bump ball grouping.

bipartite matching algorithm for a multi-pin net. It didn't consider the connections between bump balls and between bonding fingers, which leads to the overuse of routing resource. Therefore, we categorize the edge connections into 2 types, the inter-group connections and the intra-group connections, and construct a routing tree topology for bump balls and for bonding fingers in our proposed framework.

First, we create a graph for each multi-pin net and treat these pins as vertices and perform Delaunay triangulation as shown in Figure 6(a). We then set the edge weight of each pair of adjacent pins. There are 2 types of edges, intra-group and inter-group edges. If the edge connection in the triangulation result belongs to the same group (i.e., intra-group edge), the weight is set to 0; otherwise, the inter-group edges can be subdivided into 3 cases. The first case is an intra-group connection between a bonding finger and a bump ball. We calculate the Euclidean distance between them as weight on edge connection. The second case is an intra-group connection between bump balls. Similarly, we set the Euclidean distance between them as weight. And the third case is an intra-group connection between bonding fingers. For this case, we define a cost function to calculate the weight. Since the spacing between neighboring bonding fingers is typically smaller than the spacing between neighboring bump balls, we consider not only the Euclidean distance between them but also the involved connection between them. The cost function is given in equation 1. $D_{i,j}$

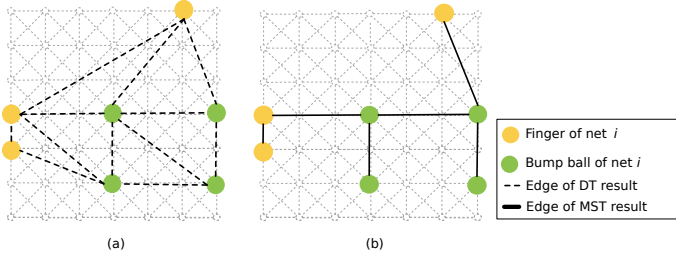


Fig. 6. Triangulation and the corresponding minimum spanning tree.

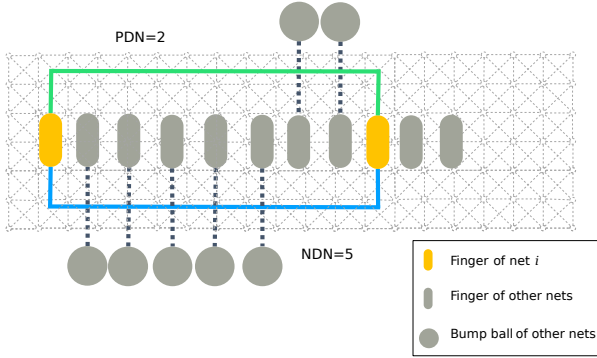


Fig. 7. Example for weight between fingers.

is the weight set on the edge connection between finger i and finger j . $dist_{i,j}$ is the Euclidean distance between finger i and finger j . P is a user-defined large constant value for a penalty. $MC_{i,j}$ denotes the involved connection between finger i and finger j . The calculation for $MC_{i,j}$ is defined as equation 2. PDN denotes the number of involved positive directions, and NDN denotes the number of involved negative directions. An example for calculating $MC_{i,j}$ is shown in Figure 7. The fingers marked in yellow represent the fingers i and j in net i and gray for other nets. We calculate the number of involved directions between the fingers of net i . In this case, PDN is 2, and NDN is 5 due to the number of involved positive directions being 2, and the number of involved negative directions is 5. Therefore, $MC_{i,j}$ is equal to $\min(2,5) = 2$. Finally, the value $D_{i,j}$ is equal to $dist_{i,j} + 2 \times P$.

After edge weight setting, the Kruskal algorithm is performed to obtain a minimum spanning tree as a routing tree to determine the routes to connect these pins of the multi-pin net, which is shown in Figure 6(b).

$$D_{i,j} = dist_{i,j} + P \times MC_{i,j} \quad (1)$$

$$MC_{i,j} = \min(PDN, NDN) \quad (2)$$

D. Group Topology Connection

We obtain intra-group and inter-group based on minimum spanning tree topology, and different strategies are used for different connection types. The following describes the strategies used for inter-group and intra-group, respectively.

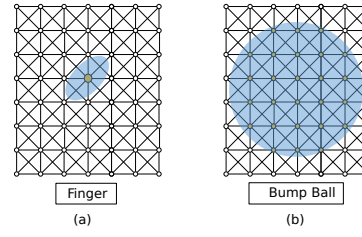


Fig. 8. Deregulation on pin area.

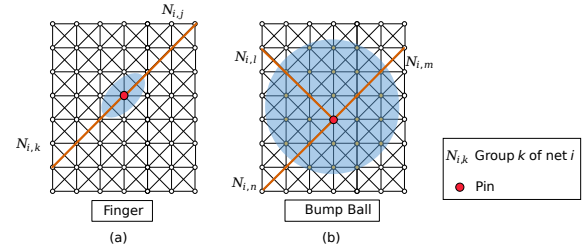


Fig. 9. Deregulation example.

D.1. Inter-Group Routing

There are 3 connection types in inter-group, including finger to bump ball (FB), bump ball to bump ball (BB) and finger to finger (FF). The pins distance of the first two connection types is usually longer than that of intra-group connection. It implies that they require more routing resource to complete wire routing. Consider that the more routing resources are used by one net, the greater impact on resource which can be used by the other nets. Therefore, in order to take a global picture of these longer connections, we use the ILP router [9] to obtain a global solution. However, they regard these multi-pin nets as different nets. It can not make the pins of a multi-pin net connect to each other. As a result, we allow them to share the routing resource under the pin area. That is, we relax the wire constraint effect in the coverage area of the finger and bump ball as shown in Figure 8 (a) and (b), respectively. The covered area is marked in blue. The vertices marked in yellow represent the deregulated vertices, and the vertices marked in red represent the pins. The wire constraint is deregulated in the coverage. It can ensure not only the satisfaction of wire constraints but also the connection for the multi-pin net. An example is depicted in Figure 9. Figure 9 (a) shows the deregulation for finger, given 2 group $N_{i,j}$ and $N_{i,k}$ in multi-pin net i , both groups can connect to each other on pin location marked in red. Similarly, the coverage of the bump ball is drawn in Figure 9(b). Given 3 group $N_{i,l}$, $N_{i,n}$ and $N_{i,m}$ in multi-pin net i , they can share routing resource under the coverage and connect to each other on pin location.

Although inter-group connection usually requires more routing resources as aforementioned, however, the area occupied by the finger on the package is much less than the bump ball, and the distance between fingers is relatively short. It implies that the routing resource required for the connection of inter-

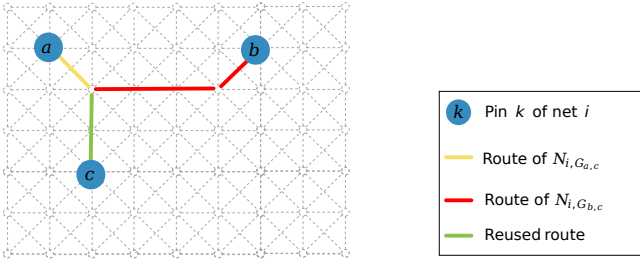


Fig. 10. A example for reusable routing.

group finger to finger is less than inter-group finger to bump ball connections and inter-group bump ball to bump ball connections. Therefore, local routing can be performed to complete the connection of inter-group FF. We use a path-reused routing strategy to accomplish the task. It can make good use of routing resource, reduce wire length and let the pins of a multi-pin net connect to each other. We define a constant parameter α less than 1 and set the weight of the routing resources used by the same multi-pin net to α . Then, we perform the Dijkstra algorithm to obtain the solution. During the procedure, it is allowed to reuse the paths for these routes of the same net. In this way, we can make these pins connect to each other and make them share routing resource to reduce the wire length. An instance is shown in Figure 10. Given 2 groups $N_{i,G_{a,c}}$ and $N_{i,G_{b,c}}$ of multi-pin net i and the routing sequence is route $N_{i,G_{a,c}}$ and route $N_{i,G_{b,c}}$. After completing the routing for $N_{i,G_{a,c}}$, we tend to reuse the routing resource used by $N_{i,G_{a,c}}$ to complete the routing for route $N_{i,G_{b,c}}$. As a result, we can reduce the wire length used by net i and make the connection for pin a , b , and c .

D.2. Intra-Group Routing

For intra-group connection, the pins of the intra-group are neighbors. As a result, we can perform a local routing strategy to complete it. There are 2 connection types in the intra-group, including FF and BB. We complete the routing for intra-group FF first. It is trivial to connect them due to the very small distance between them. For the case of bump balls connection, in addition to using the path-reused routing strategy, we use the spanning tree reconstruction strategy to alleviate via usage.

Considering the routing cost for the multi-pin net, we perform the routing for it with the least number of routes. That is, given n pins, we use $n - 1$ routes to complete the routing based on the minimum spanning tree topology. However, routing for multi-pin nets based on the topology cannot be guaranteed to be completed. In addition, it is unwise to use excess vias for one net because it will occupy lots of routing resources and it is harmful to the routing resources which can be used by the other nets. For the above reasons, we use spanning tree reconstruction strategy for the connection of intra-group BB. It is illustrated in Figure 11. We define the bump ball within $\sqrt{2}$ ball pitch from the source bump ball as neighbor. When the original routing fails, we try to make a new connection with neighbors to maintain the spanning tree. The bump balls with index marked in blue belong to multi-pin net i . Given the orig-

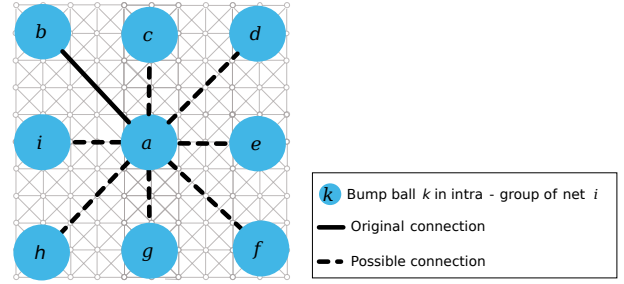


Fig. 11. Definition of minimum spanning tree reconstruction.

TABLE I
DESIGN SPECIFICATIONS

<i>Design</i>	<i>BA</i>	<i>Substrate</i>	<i>VD</i>	<i>BD</i>	<i>WW</i>
<i>Industrial 1</i>	8×8	4,500×4,500	230	350	40
<i>Industrial 2</i>	10×10	4,500×4,500	230	400	50
<i>Industrial 3</i>	16×16	17,000×17,000	250	580	60
<i>Industrial 4</i>	20×20	11,000×11,000	200	355	45
<i>Industrial 5</i>	22×22	15,000×15,000	180	355	35
<i>Industrial 6</i>	23×23	19,000×19,000	230	550	50

inal connection $route(a,b)$ with solid line, a and b are source and target, respectively. The neighbor bump balls of a are b , c , d , e , f , g , h and i . The possible connections are represented by dash lines. When the routing of $route(a,b)$ fails, we select a new neighboring bump ball within $\sqrt{2}$ ball pitch as a new target, including i , c , e , g , b , d , h , and f , to form a new spanning tree while to keep the group of bump balls connected. To maintain the local spanning tree, we try to route with new connection. If all the aforementioned cases failed, the routing for a new target with $\sqrt{2}$ ball pitch is performed again, including b , d , h , f .

IV. EXPERIMENTAL RESULTS

We implement our framework in the C++ programming language based on a Linux-based workstation with a 2.2 GHz Intel Xeon Gold 5120 processor and 128 GB memory. The Gurobi optimizer [10] is used as our ILP solver. Our benchmarks are all real industrial designs. The design specifications are listed in table I. The design specifications are shown in the following table I. Column "*Design*" lists the industrial design index. The size of ball grid array, the size of substrate, the diameter of via, the diameter of bump ball and wire width are listed in Column "*BA*", "*Substrate*", "*VD*", "*BD*" and "*WW*", respectively. The unit of "*#Layers*", "*#TN*" and "*#MN*" are number and the others are micrometer (μm).

We compared our routing results with (1) the routing for only two-pin nets (denoted by "*OTN*") and (2) the routing for two-pin nets and multi-pin nets, which uses bipartite matching for multi-pin and regard them as different and independent two-pin nets (denoted by "*MBM*") [9]. The comparison results

TABLE II
COMPARISON RESULT WITH ORIGINAL ROUTER [9]

Design	OTN [9]			MBM [9]			Ours		
	TSR	TTR	Routability(%)	TSR	TTR	Routability(%)	TSR	TTR	Routability(%)
Industrial 1	45	57	79.0	44	59	74.6	49	61	80.3
Industrial 2	89	92	96.7	91	97	93.8	96	101	95.1
Industrial 3	240	254	94.5	239	256	93.4	250	257	97.3
Industrial 4	146	175	83.4	236	301	78.4	412	486	84.8
Industrial 5	197	258	76.4	280	394	71.1	468	627	74.6
Industrial 6	223	293	76.1	360	466	77.3	573	682	84.0
Average			84.4			81.4			86.0

are shown in Table II. For each method, the numbers of successful routes and the total routes are listed in Columns "TSR" and "TTR", respectively. The Column "Routability" denotes the ratio of TSR to TTR. For OTN, the TTR of each design is least among the 3 method because it just performs routing for two-pin nets. It implies that the usable routing resources are the largest. In contrast, the routing result demonstrates that our TSR are all larger than OTN. Our average routability is better than OTN. For MBM, the TTR of each design is the second largest because it takes multi-pin nets into account. However, MBM treats each multi-pin net as multiple different two-pin nets. This results in the routing resource over-used issue for all multi-pin nets. Compared to the MBM, our proposed TSR achieves better routability in 6 industrial designs. The average routability is 86%.

V. CONCLUSIONS

As the process advances, it is time-consuming and error-prone for engineers to complete the substrate routing manually. As a result, the demand for automatic routing is critical for packaging substrate. While routing is a complex optimization problem and it is known to be NP-complete. There are many researchers devoted their valuable methodologies to this field. However, most of these studies focus on the part of a two-pin net. Therefore, in this thesis, we have proposed a substrate routing framework for wire-bonding FBGA package design. The routing algorithms for two-pin nets and multi-pin nets are integrated into our router. In addition, we also take the connection between bump balls and between fingers into account. The experimental result indicates that our proposed framework can complete the routing for multi-pin nets with better routability.

REFERENCES

- [1] Santosh Nimbale, IC Packaging, <https://documents.in/reader/full/ic-packaging>.
- [2] Kubo, Yukiko and Takahashi, Atsushi, "A Global Routing Method for 2-Layer Ball Grid Array Packages", *Proceedings of the 2005 International Symposium on Physical Design*, pp.36-43, 2005.
- [3] Kubo, Yukiko and Takahashi, Atsushi, "Global Routing by iterative improvements for two layer ball grid array packages", *IEEE Transactions on computer-aided design of integrated circuits and system*, vol. 25, no. 4, pp. 725-733, April 2006.
- [4] Yoichi Tomioka and Atsushi Takahashi, "Routability driven modification method of monotonic via assignment for 2-layer Ball Grid Array packages", *2008 Asia and South Pacific Design Automation Conference* pp. 238-243, 2008.
- [5] Chen Shuenn-Shi, Chen Jong-Jang, Lee Trong-Yen, T. Chia-Chun, and Chen Sao-Jie, "A New Approach to the Ball Grid Array Package Routing", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* vol. 82, pp.2599-2608, 1999.
- [6] Fengxian Jiao and Sheqin Dong, "Ordered Escape routing for grid pin array based on Min-cost Multi-commodity Flow", *2016 21st Asia and South Pacific Design Automation Conference*, pp.384-389, 2016.
- [7] Han, Kwangsoo and Kahng, Andrew B. and Lee, Hyein, "Evaluation of BEOL design rule impacts using an optimal ILP-based detailed router", *2015 52nd ACM/EDAC/IEEE Design Automation Conference* pp.1-6, 2015.
- [8] Jia-Wei Fang, I-Jye Lin, Ping-Hung Yuh, Yao-Wen Chang and Jyh-Herng Wang, "A routing algorithm for flip-chip design", *ICCAD-2005. IEEE/ACM International Conference on Computer-Aided Design, 2005* pp. 753-758. 2005.
- [9] Jun-Sheng Wu, Chi-An Pan, and Yi-Yu Liu "ILP-Based Substrate Routing with Mismatched Via Dimension Consideration for Wire-bonding FBGA Package Design", in *ACM Transactions on Design Automation of Electronic Systems*, Vol. 28, No. 5, 2023.
- [10] Gurobi Optimizer 8.1, <http://www.gurobi.com/>.