Yield and Timing Constrained Spare TSV Assignment for Three-Dimensional Integrated Circuits

Yu-Guang Chen[1], Kuan-Yu Lai[1], Ming-Chao Lee[2], Yiyu Shi[3], Wing-Kai Hon[1], and Shih-Chieh Chang[1]

[1]Department of Computer Science
National Tsing Hua University
HsinChu, Taiwan 30013

[2] Design Sign-Off Division
Design Technology
MediaTek Inc.
HsinChu, Taiwan 30078

[3]Department of Electrical and
Computer Engineering
Missouri University of Science and Technology
Rolla, Mo 65409

andyron75@yahoo.com.tw, yshi@mst.edu, scchang@cs.nthu.edu.tw

Abstract—**Through Silicon Via (TSV) is a critical enabling technique in three-dimensional integrated circuits (3D ICs). However, it may suffer from many reliability issues. Various fault-tolerance mechanisms have been proposed in literature to improve yield, at the cost of significant area overhead. In this paper, we focus on the structure that uses one spare TSV for a group of original TSVs, and study the optimal assignment of spare TSVs under yield and timing constraints to minimize the total area overhead. We show that such problem can be modeled through constrained graph decomposition. An efficient heuristic is further developed to address this problem. Experimental results show that under the same yield and timing constraints, our heuristic can reduce the area overhead induced by the fault-tolerance mechanisms by up to 38%, compared with a seemingly more intuitive nearest-neighbor based heuristic.**

Keywords- 3D IC; Reliability; Fault-Tolerance; TSV

I. Introduction

While three-dimensional integrated circuits (3D ICs) can provide smaller footprint, higher speed, lower power consumption [1], their reliability has become one of the primary concerns. The Through Silicon Via (TSV), which provides vertical connection in 3D ICs, may suffer from various reliability issues such as undercut, misalignment or random open defects [15][16]. Considering the large number of TSVs in a chip, these issues in turn translate into low chip yields: For example, [15][17] reported a 70% chip yield for a chip with 10,000 TSVs and only 20% yield for 55,000 TSVs in IMEC process technology, which would be a primary concern in 3D ICs design.

To enhance reliability, many fault-tolerance mechanisms have been proposed in literature such as [9][14][15][16][18][19]. Most previous works focus on innovative structures of spare TSVs and the corresponding control circuits, which will be reconfigured to reroute the signal in the presence of TSV failure. In this paper, we focus on a classical fault-tolerance structure, where a number of original TSVs and one spare TSV are grouped together to provide redundancy. Such a structure can be implemented with very simple control logic. Figure 1 shows one such structure with four original TSVs and one spare TSV. Seven multiplexers serve as the control circuit. Figure 1(a) shows the original configuration where all TSVs are good and each signal passes through their corresponding TSV. If TSV T_1 is faulty, the new configuration is shown in Figure 1(b).

There are two types of design constraints when building such a fault-tolerant structure. First, the chip yield imposes a limit on the maximum number of original TSVs *N* that can be in a group. It also sets a lower bound for the distance between the TSVs in the same group due to the clustering effect of TSV faults [7][19]. Second, neighboring TSVs in a group should not be topologically far away to avoid timing violation when the signal is rerouted. These distance constraints prevent us from randomly grouping TSVs to form a TSV fault-tolerance structure.

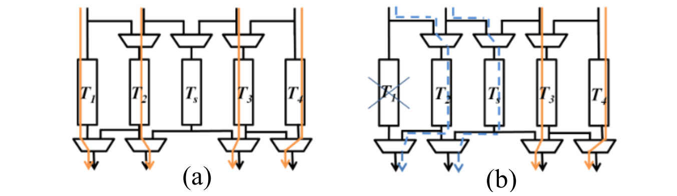


Figure 1. A fault-tolerance structure and the signal routing when (a) all TSVs are good and (b) T_1 fails.

Fault-tolerance structures incur significant area overhead, which is used by spare TSVs, control logics, and rerouting wires [14][18][19]. To reduce cost, recently a few efficient spare TSV allocation algorithms were proposed [18][19]. With detailed example given in Section II, we argue that it is the flexibility to place spare TSVs that provides maximum room to minimize design overhead.

In this paper, we formulate the interesting problem of optimal spare TSV assignment for irregular TSV placement, considering both yield and timing constraints, such that the total area overhead is minimized. We show that the problem can be cast into a constrained graph decomposition problem. We then devise an efficient heuristic to address this NP-hard problem. Experimental results show that under the same yield and timing constraint, our heuristic can reduce the area overhead induced by the fault-tolerance mechanisms by up to 38% compared with a seemingly more intuitive nearest-neighbor based heuristic.

The remainder of the paper is organized as follows. Section II describes background and motivation of our work. Section III discusses the assumptions and formulates the problem. Section IV proposes an efficient heuristic to address it. Experimental results are presented in Section V and concluding remarks are given in Section VI.

II. Background and Motivation

It is clear that employing spare TSVs in 3D IC designs can significantly increase the yield, yet extra area overhead may become an inevasible cost. Appropriately assigning spare TSVs can dramatically reduce the number of spare TSVs needed (and accordingly the extra area overhead).

Therefore, Ye *et al.* [18] proposes an ILP based algorithm to group pre-placed original TSVs and spare TSVs to minimize

This work is supported in part by National Science Council of Taiwan under Grant NSC 100-2221-Z-007-126-MY3 and NSC 101-2221-Z-007-119-MY3.
978-3-9815370-2-4/DATE14/©2014 EDAA

total wirelength overhead during re-routing. However, the work assumes that original TSVs and spare TSVs are already placed, which lacks flexibility. We argue that the design overhead can be maximally reduced by optimally allocating the spare TSVs at appropriate locations to form redundancy.

We use Figure 2 as an example to demonstrate why for cases with irregularly placed TSVs, the appropriate assignment of spare TSVs can significantly reduce the number of fault-tolerant structures and therefore, the overall design cost. Figure 2 (a) shows a tier in 3D IC with nine original TSVs. A hollow circle stands for an original TSV and a solid circle for a spare TSV. For yield constraint, we assume that each fault-tolerance structure can have no more than three original TSVs. For timing constraint, a maximum distance is set between two neighboring TSVs. Figure 2 (b) and Figure 2 (c) show two different grouping results for inserting spare TSVs. Obviously, a better solution is shown in Figure 2 (c): Only three spare TSVs are needed in Figure 2 (c) instead of five used in Figure 2(b), a 40% reduction.

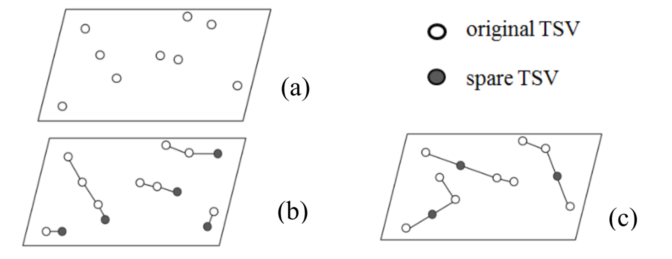


Figure 2. An example for spare TSVs assignment for (a) nine original TSVs, and the fault-tolerant structures formed with (b) five spare TSVs and (c) three spare TSVs.

III. Problem Formulation

Before we present our problem formulation, it is important to clarify the following five assumptions we use.

1) We assume that the original TSVs are placed without consideration of fault-tolerance since the stringent yield and timing constraints may result in significant distortion to the routing of signal nets.
2) Following the assumptions in [5][8][10][12][18], in this work we assume that spare TSVs can be placed irregularly. This will allow more flexibility and show better wirelength [8][10].
3) The spare TSVs placed following our formulation may overlap with logic cells, IP blocks or macro cells. This can be addressed by performing an extra step of local legalization, similar to the method used in [3].
4) If any fault tolerance structure formed by our algorithm cannot be implemented due to blockage, we will remove it and use double TSV technique [14] for each TSV in that structure. This step may increase the number of spare TSVs needed, however, it can guarantee the routability.
5) We assume that in each fault-tolerant structure, only one spare TSV is inserted. This significantly simplifies the related control logic.

As such, our problem can be stated as follows: **Given (i) a chip with all original TSVs are already placed, (ii) the target yield of a chip, (iii) the timing slacks of the signals that go through TSVs, determine an optimal spare TSV assignment such that total area overhead induced by the fault-tolerance mechanisms is minimized.**

Literature has pointed out that the size of TSVs is typically much larger than that of standard cells (about 5-10x in 45nm technology) [11]. As a result, we simply reduce the objective from minimizing the total area overhead to minimizing the number of spare TSVs. Experimental results show that the latter objective can indeed significantly reduce the total area overhead significantly.

To explicitly consider the chip yield and timing constraints, however, we need a formal method to represent them.

A. Chip Yield Constraint

Assume the failure rate for a single TSV is *F,* and to simplify the discussion we assume each TSV fails independently[1]. Consider the fault-tolerant structure with *N* original TSVs and one spare TSV. The structure will be good if all *N*+1 TSVs are good, or if only one of them fails. The yield of the structure can be calculated as

$$\begin{aligned} Y_{structure} &= (1-F)^{N+1} + (N+1)\times(1-F)^{N}\times F \\ &= (1-F)^{N}\times(1+NF). \end{aligned} \quad (1)$$

It is clear that a chip fails if any of the fault-tolerant structures fails. Therefore, the yield of the entire chip, Y_{chip}, can be calculated as

$$Y_{chip} = (Y_{structure})^{\#Structure}, \quad (2)$$

where *#Structure* denotes the number of fault-tolerance structures in a chip.

Given the yield constraint of a chip, $Y_{chip\_constraint}$, we can use (1) and (2) to estimate *N*, the number of original TSVs in a fault-tolerance structure as

$$[(1-F)^{N}\times(1+NF)]^{\frac{M}{N}} \geq Y_{chip\_constraint}, \quad (3)$$

where *M* denotes total number of original TSVs in the chip.

Due to traditional semiconductor manufacturing, TSV faults may have clustering effect rather than random distribution [7][19]. Since we focus on the fault-tolerance structure with only one spare TSV, it is important to avoid grouping very close TSVs in the same structure [7][18]. Therefore, we impose a minimum distance constraint, D_{min}, to any two TSVs that can be in the same fault-tolerance structure, as suggested by [6][18].

B. Timing Constraint

As for the timing constraint, given the slack of the signal passing through a TSV, the maximum rerouting distance of the TSV *i*, $D_{max,i}$, is defined as the maximum distance between this TSV and its neighbor (the TSV used to re-route its signal when it fails). The maximum rerouting distance can be easily calculated, either through simple RC delay model, or through detailed SPICE simulation. The delay induced by control logics such as MUX is also included when calculating maximum rerouting distance. Note that such maximum rerouting distance can be different for different TSVs.

One potential issue here is the asymmetry of the maximum

[1] This may not be the case as TSV faults tend to be clustered in some cases. With proper model, the yield can still be calculated but in a much complicated manner. We omit the details here in the interest of space.

distance: For example, T_1 is within the maximum distance of T_2, but not vice versa. Accordingly, T_1 can be used to reroute the signal for T_2, but T_2 cannot be used for T_1. This significantly increases the problem difficulty. In this paper, as an initial exploration, we will only put two TSVs as neighbors when they are within the maximum distance of each other, and leave the asymmetric distance constraints as our future work.

Based on the above discussion, to form a TSV fault-tolerant structure while considering both yield and timing constraints, the maximum number of original TSVs in the group and distance constraints, $D_{max,i}$ and D_{min}, between neighboring TSVs must be held. As such, our problem can be stated as follows:

Formulation 1 (Yield and Timing Constrained Spare TSV Assignment)*:* Given (i) the locations of all original TSVs, (x_i, y_i) on a tier from placement result, (ii) the maximum number of original TSVs in a group, N, (iii) the routing distance constraint, $D_{max,i}$, for each TSV i, and (iv) the minimum distance constraint, D_{min}, between neighboring TSVs, determine an optimal spare TSV assignment such that the total number of spare TSVs m is minimized.

IV. ALGORITHM

In this section, we first model our problem in Formulation 1 to the constrained decomposition problem, and then propose an efficient polynomial-complexity heuristic to tackle the problem.

A. Problem Modeling

It is interesting that Formulation 1 has a close relationship with graph theory. To see this, we first define an assignment graph as follows.

Definition 1: An assignment graph is an undirected graph constructed as follows.

1) Each vertex, s_i, represents an original TSV T_i.
2) There are two types of edges in the graph: *Primary edges* and *second edges* defined as follows.
3) There is a ***primary edge*** between vertices s_i and s_j if their distance is greater than D_{min} and within the maximum rerouting distance of each other, i.e., $D_{min} \leq d_{ij} \leq \min(D_{max,i}, D_{max,j})$, otherwise
4) There is a ***secondary edge*** between vertices s_i and s_j if they are within the sum of maximum rerouting distance of each other, i.e., $d_{ij} \leq D_{max,i} + D_{max,j}$.

A primary edge implies the two TSVs can be connected directly as neighboring TSVs, while a secondary edge means that they can only be joined through a spare TSV in the middle. For the primary edge, we have imposed the symmetric constraint as discussed in Section III. As an example, Figure 3 shows the assignment graph constructed from Figure 2(a) where primary edge is represented by solid lines and secondary edge by dashed lines.

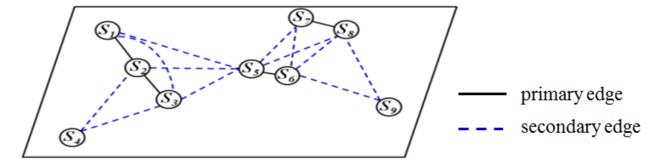


Figure 3. An example of assignment graph constructed from Figure 2 (a).

With the assignment graph, our problem can be casted as follows.

Formulation 2 (Constrained Decomposition of Assignment Graph): Given an assignment graph $G_{assignment}$ and an integer N, decompose the graph into a minimum number of disjointed sub-graphs, such that

1) each sub-graph contains no more than N vertices;
2) all the vertices in each sub-graph can be connected by a spanning tree, and
3) the spanning tree contains no more than one secondary edge.

Each resulting sub-graph will be used to form a fault-tolerant structure by inserting a spare TSV This problem can be proved as an NP-hard problem by reducing to the well-known Exact Cover by 3-Sets (X3C) problem [2]. Due to the space limitation, we omit the detailed description.

B. Our Heuristic

This section describes our proposed heuristic for the constrained decomposition problem as described in Formulation 2. In our heuristic, we decompose the assignment graph by constructing disjointed sub-graphs iteratively, and eventually output all the sub-graphs and the spanning tree in each sub-graph.

Our heuristic is based on two simple observations. First, a vertex with lower degree (fewer edges) has fewer choices to form a sub-graph. This implies that a higher priority should be given to vertices with lower degrees. Second, when forming sub-graphs, we should always try to first include the vertices that are connected to the sub-graph by primary edges, because once a secondary edge is included, we have less flexibility in selecting the remaining vertices.

Based on the two observations, we start with an empty sub-graph, find a vertex with the lowest degree in the graph, and add it to the sub-graph. We then search all the candidate vertices to find one with lowest residual degree[2], and add it to the sub-graph. The way to determine candidate vertices is explained later. This process continues until N vertices have been added, or no more candidate vertices can be found. The vertices in the sub-graph now form a fault-tolerant structure. These vertices, along with all the edges connected to them, are removed from the graph. And the same process repeats to form the next sub-graph.

To keep track of the spanning tree in the sub-graph, when expanding the sub-graph we will only keep the edges that form the spanning tree. The following rules are used to decide which edge to keep.

E1) A secondary edge will always be removed before a primary edge.

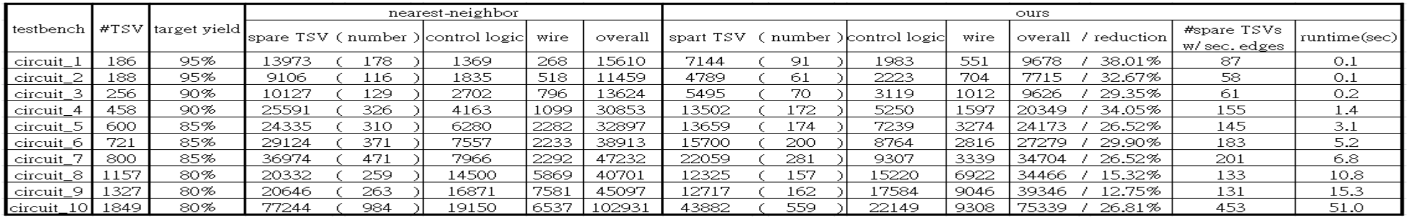
E2) When both edges are of the same type, the longer edge will always be removed before the shorter edge.

E3) Randomly select the edge to be removed if still tied after E1 and E2.

With the above edge deletion technique, a vertex is a candidate during sub-graph expansion if

[2] The residual degree of a vertex is defined as the number of edges between the vertex and all other vertices not in the sub-graph.

Table 1. Comparison between nearest-neighbor based heuristic and our heuristic in terms of area overhead induced by spare TSVs, control logics, rerouting wires, and the overall area overhead (in μm^2)



testbench	#TSV	target yield	nearest-neighbor				ours					
			spare TSV (number)	control logic	wire	overall	spart TSV (number)	control logic	wire	overall / reduction	#spare TSVs w/ sec. edges	runtime(sec)
circuit_1	186	95%	13973 (178)	1369	268	15610	7144 (91)	1983	551	9678 / 38.01%	87	0.1
circuit_2	188	95%	9106 (116)	1835	518	11459	4789 (61)	2223	704	7715 / 32.67%	58	0.1
circuit_3	256	90%	10127 (129)	2702	796	13624	5495 (70)	3119	1012	9626 / 29.35%	61	0.2
circuit_4	458	90%	25591 (326)	4163	1099	30853	13502 (172)	5250	1597	20349 / 34.05%	155	1.4
circuit_5	600	85%	24335 (310)	6280	2282	32897	13659 (174)	7239	3274	24173 / 26.52%	145	3.1
circuit_6	721	85%	29124 (371)	7557	2233	38913	15700 (200)	8764	2816	27279 / 29.90%	183	5.2
circuit_7	800	85%	36974 (471)	7966	2292	47232	22059 (281)	9307	3339	34704 / 26.52%	201	6.8
circuit_8	1157	80%	20332 (259)	14500	5869	40701	12325 (157)	15220	6922	34466 / 15.32%	133	10.8
circuit_9	1327	80%	20646 (263)	16871	7581	45097	12717 (162)	17584	9046	39346 / 12.75%	131	15.3
circuit_10	1849	80%	77244 (984)	19150	6537	102931	43882 (559)	22149	9308	75339 / 26.81%	453	51.0

C1) no secondary edge exists in the sub-graph, and the vertex is connected to at least one of the vertices in it; or

C2) one secondary edge exists in the sub-graph, and the vertex is connected to at least one of the vertices in it with a primary edge.

V. Experimental Results

We implement our algorithm in C, and perform experiments on 10 circuits, which are obtained by applying 3D-Craft [4] to five 2D industrial designs in 45nm technology. Original TSVs and their locations in each case are extracted from layouts as our inputs. Target yields are set depending on the size of each circuit. Maximum distance constraint for each TSV is obtained from timing analysis results, and ranges between 20μm and 50μm. Minimum distance constraint is set to 10μm, as [6][18] suggested. We also consider the Tezzaron design rule [13] for TSV density. After our algorithm completes, we check for routing congestions. If any fault tolerance structure formed by our algorithm cannot be implemented, we remove it and use double TSV technique [14] for each TSV in that structure, as discussed in Section III.

Since no prior art exists, we implement another heuristic that sounds more intuitive and simpler. Instead of using an assignment graph, each time a sub-graph is constructed, we iteratively select the vertex that has the smallest distance to the current sub-graph[3], and add that vertex to it. The process is continued until no more vertices are within the maximum rerouting distance of any vertex in the sub-graph, or the limit of the vertices number N has reached. We call it nearest-neighbor based heuristic.

We compare the area overhead induced by spare TSVs, control logics and rerouting wires as well as the overall area overhead of the two algorithms as shown in Table 1. Apparently, the majority of the overall area overhead is contributed by the spare TSVs. This validates our reduction in the problem formulation from minimizing the overall area overhead to minimizing the number of spare TSVs. From the table we can find the overall area overhead is reduced by up to 38%. We also present the number of spare TSVs with secondary edges needed for each circuit. This actually shows the proposed heuristic gets much better result mainly because it can insert spare TSVs between two TSVs that form a secondary edge. The corresponding runtimes of our heuristic are also reported in Column 12. They are quite short if we take into account the NP-hardness of the problem.

VI. Conclusions

In this paper, we study the optimal spare TSV assignment problem under yield and timing constraints to minimize design cost. We show that such problem can be modeled through constrained graph decomposition. An efficient heuristic is further developed to address the problem. Experimental results show that under the same yield and timing constraint, our heuristic can reduce the overall area overhead by up to 38% compared with the nearest-neighbor based heuristic.

References

[1] J. Burns, L. Mcllrath, C. Keast, C. Lewis, A. Loomis, K. Warner and P. Wyatt, "Three-Dimensional Integrated Circuit for Low Power, High-Bandwidth Systems on a Chip," in *Proc. of IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 268-269, Feb., 2001.

[2] W. L. Chang, M. Guo, "Solving the set cover problem and the problem of exact cover by 3-sets in the Adleman–Lipton model," in *Biosystems, Vol.72(3), pp.263-275*, Dec. 2003.

[3] M. Cho, H. Ren, H. Xiang, R. Puri, "History-based VLSI legalization using network flow", in *Proc. of ACM/IEEE Design Automation Conference (DAC)*, pp. 286-291, June 2010.

[4] J. Cong and G. Luo, "A 3D physical design flow based on Open Access," in *Proc. of International Conference on Communications, Circuits and Systems* (*ISSCAS*), pp.1103-1107, 2009.

[5] J. Cong, G. Luo, Y. Shi, "Thermal-aware cell and through-silicon-via co-placement for 3D ICs", in *Proc. of ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp.670-675, June 2011.

[6] R. Glang, "Deffect size distrubution in VLSI chips", in *Proc. of International Conference on Microelectronic Test Structures (ICMTS)*, pp.57-60, March 1990.

[7] L. Jiang, Q. Xu, and B. Eklow, "On effective TSV repair for 3D-stacked ICs", in *Proc. of Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp.793-798, March 2012

[8] M. Jung, J. Mitra, David Z. Pan, and S. K. Lim, "TSV stress-aware full-chip mechanical reliability analysis and optimization for 3D IC", in *Proc. of Design Automation Conference (DAC)*, pp. 188-193, June 2011.

[9] Uksong Kang and et al., "8Gb 3D DDR3 DRAM using through-silicon-via technology", in *Proc. of IEEE International Solid-State Circuits Conference (ISSCC)*, pp130-131, 131a, Feb. 2009.

[10] 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 Design Automation Conference (DAC)*, pp. 674-680, Nov. 2009.

[11] D. H. Kim, S. Mukhopadhyay, and S. K. Lim, "TSV-aware interconnect length and power prediction for 3D stacked ICs", in *Proc. of International Interconnect Technology Conference (IITC)*, pp.26-28, June 2009.

[12] D. H. Kim, R. O. Topaloglu, and S. K. Lim, "TSV density-driven global placement for 3D stacked ICs", in *Proc. of International SoC Design Conference (ISOCC)*, pp.135-138, Nov. 2011

[13] D. H. Kim, S. K. Lim and et al., "3D-MAPS: 3D Massively parallel processor with stacked memory", in *Proc of IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 188-190, Feb. 2012

[14] M. Laisne, K. Arabi and T. Petrov, "System and methods utilizing redundancy in semiconductor chip interconnects", *United State Patent Application Publication*. US 2010/0060310 A1 (Mar. 11, 2010).

[15] I. Loi, S. Mitra, T. H. Lee, S. Fujita and L. Benini, "A low-overhead fault tolerance scheme for TSV-based 3D network on chip links", in *Proc. of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp.598-602, Nov. 2008.

[16] C.-L Lung, Y.-S. Su, S.-H Huang, Yiyu Shi, and S.-C. Chang, "Fault-Tolerant 3D Clock Network", in *Proc. of ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp.645-651, June 2011.

[17] B. Swinnen, W. Ruythooren, P. D. M. L. Bogaerts, L. Carbonell, K. D. Munck, B. Eyckens, S. Stoukatch, Tezcan, D. Sabuncuoglu, Z. Tokei, J. Vaes, J. V. Aelst, and E. Beyne, "3D integration by cu-cu thermocompression bonding of extremely thinned bulk-si die containing 10 μm pitch through-si vias", in *Proc. of International Electron Devices Meeting (IEDM)*, pp. 1-4, Dec. 2006.

[18] F. Ye and K. Chakrabarty, "TSV open defects in 3D integrated circuits: Characterization, test, and optimal spare allocation", in *Proc. of ACM/IEEE Design Automation Conference (DAC)*, pp. 1024-1030, June 2012.

[19] Y. Zhao, S. Khursheed, and B. M. Al-Hashimi, "Cost-Effective TSV Grouping for Yield Improvement of 3D-ICs", *in Proc. of Asian Test Symposium (ATS)*, pp.201-206, Nov. 2011

[3] The distance between a vertex and a sub-graph is defined as the minimum distance between that vertex and all the vertices in the sub-graph. The distance must satisfy distance constraints (iii) and (iv) in Formulation 1.