On the Complexity of Graph Cuboidal Dual Problems for 3-D Floorplanning of Integrated Circuit Design

Renshen Wang
University of California, San Diego
La Jolla, CA 92093-0114
rewang@cs.ucsd.edu

Chung-Kuan Cheng
University of California, San Diego
La Jolla, CA 92093-0114
ckcheng@ucsd.edu

ABSTRACT

This paper discusses the impact of migrating from 2-D to 3-D on floorplanning and placement. By looking at a basic formulation of graph cuboidal dual problem, we show that the 3-D case and the 3-layer 2.5-D case are fundamentally more difficult than the 2-D case in terms of computational complexity. By comparison among these cases, the intrinsic complexity in 3-D floorplan structures is revealed in the hard-deciding relations between topological connections and geometrical contacts. The results show future challenges for physical design and CAD of 3-D integrated circuits.

Categories and Subject Descriptors

J.6 [**Computer Applications**]: Computer-aided design

General terms – Algorithms, theory

Keywords – 3-D integrated circuits, cuboidal dual, computational complexity

1. INTRODUCTION

New technologies such as three-dimensional integration are becoming a new force keeping Moore's law still holding in the nano era. By adding a dimension in current 2-D VLSI circuits, we can greatly enhance integration density and reduce interconnection wire length, which helps to improve system performance and lower power consumption. Meanwhile, the extra dimension also brings higher complexity in design, CAD tools and fabrications. To fully exploit the advantages of the third dimension in 3-D integrated circuits, we first need to measure and understand the complexity it brings, and face the challenge of handling this complexity.

Placement of circuit blocks is an important step of design, which has a large complexity increment migrating from 2-D to 3-D. Current developing 3-D circuits and system-on-chips [8] are usually achieved by die stacking, which is a stack of 2-D circuit layers with same thickness. This type of placement is also called 2.5-D placement [3], [4] since it does not contain full 3-D structures.

Full 3-D floorplan and placement representations are explored in several works since [10]. Full 3-D means the circuit blocks are cuboids placed in a space with no distinguishable "layers". Though we have as yet no 3-D cell library to support this class of 3-D IC design, there are full 3-D applications in reconfigurable FPGAs [9] where time is regarded as another dimension. It is found that most of the floorplan representations effective in 2-D do not have a equally effective extension in 3-D, such as Sequence-pair (2-D) to Sequence-triple (3-D) in [10]. Since a representation is virtually a data structure from which a floorplan can be recovered, we try to explore this complexity through a general type of data structure, graph.

In this work, we discuss the complexity of the two classes of 3-D floorplans through a *"cuboidal dual"* problem in a most basic formulation: Given a graph $G = (V, E)$, can we find a set of cuboids as V with contact relations as E?

The problem is similar to the "rectangular dual" problem in [6], except that the solution in [6] must be a rectangular dissection without empty space. Optimization on this problem can be applied on the initial floorplanning stage of physical design. For example, if a pair of circuit blocks b_i, b_j are heavily connected, we let $(v_i, v_j) \in E$ to make them closer; or if the two blocks both have high power density, we make $(v_i, v_j) \notin E$ for better heat dissipation.

A 2-D rectangular dual can be decided by a set of conditions in [6] or [5], and can be efficiently generated in linear time. For cuboidal duals, while the 2-D case can be solved with a similar approach, we find the 3-D cases are fundamentally more difficult in terms of computational complexity. Like the 2-colorability problem is easy but 3-colorability is NP-hard, one extra color or dimension brings a higher level of complexity. In fact we prove the 3-D cuboidal dual problem is NP-complete by reducing 3-colorability to it. For the 2.5-D cuboidal dual, we find it NP-complete when the number of layers reaches 3. The results imply that the complexity of IC physical design can be greatly increased when we extend the circuit on the third dimension, even for just a few layers of 2.5-D circuits.

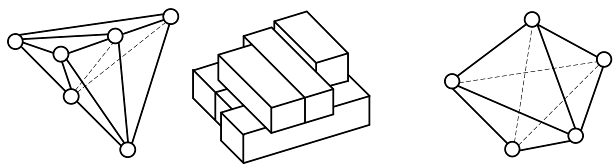
The rest of this paper is organized as follows. Section 2 introduces the basic problem formulation. Section 3 proves the general 3-D cuboidal dual problem is hard and section 4 shows 2.5-D cuboidal dual with 3 layers is hard. Finally section 5 makes comparisons and conclusions on these results.

GLSVLSI'09, May 10–12, 2009, Boston, Massachusetts, USA.

2. PRELIMINARIES

Traditional 2-D floorplanning is to place a set of rectangles in a designated area to meet certain requirements. The basic constraint is that no common area can be shared by two or more rectangles. For 3-D, the problem becomes placing a set of cuboids in a space without common space shared by multiple cuboids. A 2-D case can be regarded as a 3-D case with each cuboid placed on the floor.

An adjacency graph can be constructed from a floorplan by assigning a vertex to each cuboid and add edge (v_i, v_j) when the two corresponding cuboids are contacting on surfaces. While this construction is easy, the reverse construction from graph to floorplan is not trivial. In [6] there is a set of sufficient and necessary conditions for a graph to be an adjacency graph of a rectangular dissection. The dissection is called a *rectangular dual* of the graph. For 3-D, we define a problem based on graph *cuboidal dual*s.



(a) A graph and its cuboidal dual (b) K_5 has no cuboidal dual

Figure 1: Graph-floorplan relations

A general 3-D cuboidal dual of an n-vertex graph $G = (V, E)$ is defined as a set of cuboids, each cuboid C_i corresponds to a vertex $v_i \in V$. No two cuboids share a common part of space. C_i and C_j are adjacent (contacting on surfaces by a non-zero area) if and only if $(v_i, v_j) \in E$. Figure 1 shows a 6-vertex graph and one of its cuboidal duals, and a 5-vertex complete graph has no cuboidal duals.

A 2.5-D cuboidal dual is defined as a 3-D cuboidal dual with an additional constraint that every cuboid has height interval $[l-1, l]$, where l is the layer indicating integer.

A 2-D cuboidal dual is defined as a 2.5-D cuboidal dual with one layer, i.e. every cuboid is placed in height interval [0,1]. It is different from a rectangular dual [6] in that the set of cuboids can be a *subset* of a space dissection.

Our basic problem is to find a cuboidal dual of a given graph G. For any of the 2-D, 2.5-D or 3-D case, the problem is trivially in NP, because it is easy to verify if a given set of cuboids is a solution, i.e. to check whether for each pair of i, j, $(v_i, v_j) \in E \Leftrightarrow C_i$ and C_j are contacting on surfaces.

3. 3-D CUBOIDAL DUAL OF GENERAL GRAPHS

To decide whether a graph has a 3-D cuboidal dual is NP-hard. We prove this by reducing the well known NP-complete problem, 3-colorability, to 3-D cuboidal dual. We construct G from a 3-colorability instance $G_{3C} = (W, E')$.

First we introduce a gadget of 7 vertices for each vertex in W, shown in figure 2. The 7 vertices together with the edges form an octahedron composed of 8 tetrahedrons. There is a cuboidal dual of this graph, and the contact surfaces between different pairs of cuboids are not independent.

LEMMA 1. *In the cuboidal dual of the 7-vertex gadget, the cuboids of two opposite vertices on the octahedron (e.g. v_1, v_4) are on opposite sides of the central cuboid (of v_0).*

PROOF. (Brief) Since v_0 and v_1 are adjacent, the cuboids are contacting on a common plane denoted as p_{01}, and their overlapping surface on p_{01} is a rectangle R (figure 2). Since the four surrounding vertices in a loop are all adjacent to v_0

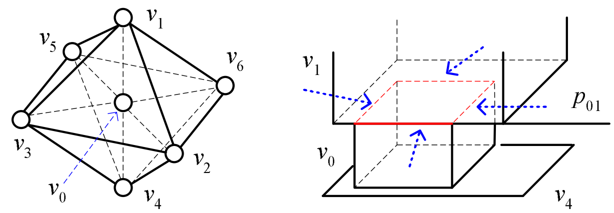


Figure 2: 7-vertex gadget and its cuboidal dual

and v_1, the 4 cuboids must be contacting the outline of R and therefore be on the 4 sides of R. And since v_4 is also adjacent to the 4 surrounding vertices, the projection of v_4's cuboid on p_{01} must be covering R. Therefore v_1 and v_4 have cuboids on opposite sides of v_0. □

So for a 7-vertex gadget N, the contacting directions of $v_1 \to v_0$ and $v_0 \to v_4$ are same, denoted as $d_{1,4}(N)$. In the same way, the other two pairs of vertices (v_2, v_5) and (v_3, v_6) are on opposite sides of v_0. Also from figure 2, cuboids of $v_1, v_2, \cdots, v_6$ cover all the 6 surfaces of cuboid v_0.

Regarding the coordinate axis $d_{1,4}(N)$ is parallel to, it has three possible directions: x, y and z. These directions can be used as the 3 possible colors in the 3-colorability problem, where a gadget N is colored as $d_{1,4}(N)$. For edge $(w, w') \in E'$ in G_{3C}, the two vertices cannot share the same color. This constraint can be realized as a biclique between v_1 and v_4 of two gadgets N and N'. As figure 3 shows, on the axis parallel to $d_{1,4}(N)$, v_1 occupies interval $[a_1, b_1]$ and v_4 occupies interval $[a_4, b_4]$. If there is a biclique between $\{v_1, v_4\}$ and $\{v'_1, v'_4\}$, then both v'_1 and v'_4 must cover interval $[b_4, a_1]$ on the axis, so $d_{1,4}(N')$ cannot be parallel to $d_{1,4}(N)$.

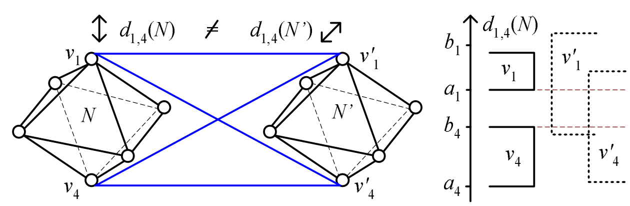


Figure 3: Enforcing 2 gadgets in different directions

To complete the reduction from 3-colorability we need to construct G_{3C} based on the gadget nodes. We add 6 more vertices to the 7-vertex gadget to further restrict the contacting directions among the cuboids of $v_1, \cdots, v_6$.

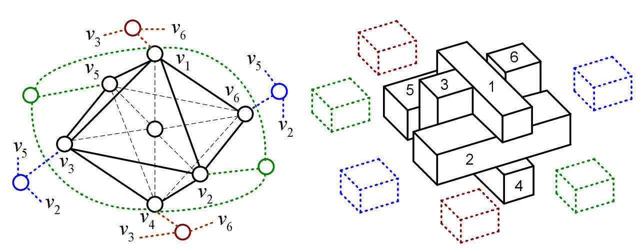


Figure 4: 13-vertex gadget and its cuboidal dual

LEMMA 2. *(figure 4) Adding 3 pairs of vertices to the 7-vertex gadget, pair 1 connected to $\{v_1, v_2, v_4\}$ and $\{v_1, v_5, v_4\}$, pair 2 connected to $\{v_2, v_3, v_5\}$ and $\{v_2, v_6, v_5\}$, pair 3 connected to $\{v_3, v_1, v_6\}$ and $\{v_3, v_4, v_6\}$, then cuboid v_1 and v_4 have same width as cuboid v_0 (along $d_{3,6}$),* *cuboid v_2 and v_5 have same height as cuboid v_0 (along $d_{1,4}$), cuboid v_3 and v_6 have same length as cuboid v_0 (along $d_{2,5}$).*

In the 13-vertex gadget here, the original 7-vertex gadget have a definite shape, so we can easily align multiple gadgets into the same direction with some additional vertices in G.

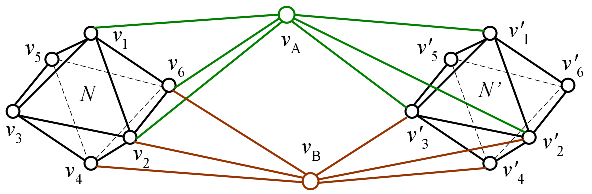


Figure 5: Two 13-vertex gadgets with $d_{1,4}(N)$ and $d_{1,4}(N')$ aligned to the same direction (2-alignment)

As in figure 5 (using the simplified octahedron to represent a 13-vertex gadget), we add two vertices v_A and v_B. Consider their connections with gadget N' on right. Since v_A is simultaneously contacting v_1', v_2' and v_3', by Lemma 2 and figure 4, it must be on the corner formed by the 3 cuboids, and cuboid v_A is therefore above v_2'. Similarly, v_B is contacting v_4', v_2' and v_3', so it must be on the corner and cuboid v_B is below v_2'. As a result, the direction $v_A \to v_B$ is same as $d_{1,4}(N')$.

The same conclusion can be found on gadget N, i.e. the direction $v_A \to v_B$ is same as $d_{1,4}(N)$. Therefore with two additional vertices we make $d_{1,4}(N) = d_{1,4}(N')$.

Besides the alignment of $d_{1,4}$, we also need to align two gadgets so that the directions $d_{1,4}$, $d_{2,5}$ and $d_{3,6}$ of these two gadgets are all in parallel.

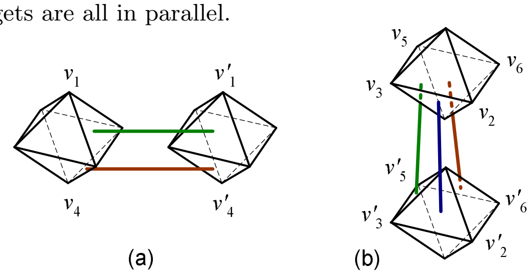


Figure 6: 2-alignment and 3(complete)-alignment

Figure 6(a) is the simplified notation of the alignment illustrated in figure 5, where only the directions of $d_{1,4}(N)$ and $d_{1,4}(N')$ are parallelized. We call this a 2-alignment. In figure 6(b) there are 3 additional vertices (called 3-alignment), the result is $d_{2,5}(N) = d_{2,5}(N')$ and $d_{3,6}(N) = d_{3,6}(N')$, which also implies $d_{1,4}(N) = d_{1,4}(N')$. So in a 3-alignment, the two gadgets are completely aligned in every direction.

Also notice that the direction from one gadget to the other in a 2-alignment is along $d_{2,5}(N)$ or $d_{3,6}(N)$, while this direction in a 3-alignment must be along $d_{1,4}(N)$. These two cases enables the alignment of a pair of 13-vertex gadgets N and N' along any of the 3 axes, with $d_{1,4}(N) = d_{1,4}(N')$. Based on these two alignments we can construct connections between the 13-vertex gadgets as edges in $G_{3C} = (W, E')$ for the reduction from the 3-colorability problem.

THEOREM 1. *3-colorability reduces to 3-D cuboidal dual.*

PROOF. Given a graph from 3-colorability $G_{3C} = (W, E')$ with n vertices denoted as $w_1, w_2, \cdots, w_n$. For each vertex w_i, construct n 13-vertex gadget nodes in G, denoted as $s_{i,1}, \cdots, s_{i,n}$, sequentially connected by 2-alignments. Then for each gadget node $s_{i,j}$, construct 4 auxiliary gadgets as follows: $s_{i,j}$ 2-aligns with $t_{1,i,j}$, $t_{1,i,j}$ 3-aligns with $t_{2,i,j}$, $t_{2,i,j}$ 2-aligns with $t_{3,i,j}$, and finally $t_{3,i,j}$ 2-aligns with $u_{i,j}$.

For each edge $(w_i, w_j) \in E'$, we pick gadget nodes $u_{i,j}$ and $u_{j,i}$, connect $\{v_1(u_{i,j}), v_4(u_{i,j})\}$ with $\{v_1(u_{j,i}), v_4(u_{j,i})\}$ so that the 2 sets of vertices form a biclique. In this way graph G has a cuboidal dual *if and only if* G_{3C} is 3-colorable.

If G_{3C} is not 3-colorable, then no matter how we place the gadgets, there is at least one pair of vertices w_i and w_j such that $(w_i, w_j) \in E'$, and $d_{1,4}(u_{i,j}) = d_{1,4}(u_{j,i})$. So $v_1(u_{i,j}) \to v_4(u_{i,j})$ and $v_1(u_{j,i}) \to v_4(u_{j,i})$ are on the same direction, and by figure 3 it is impossible to form a biclique between the 2 sets.

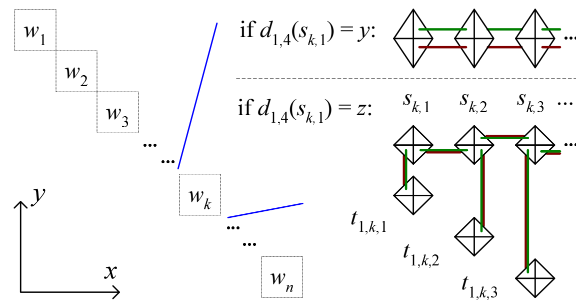


Figure 7: Construction of cuboidal dual from G when G_{3C} is 3-colorable

If G_{3C} is 3-colorable, we can construct a cuboidal dual according to figure 7. Vertices $w_1, \cdots, w_n$ are placed on the xy-plane and figure 7 is a top view. Each vertex w_k has a color of $\{x, y, z\}$, which decides the direction of gadget nodes $d_{1,4}(s_{i,j})$. Every edge in E' is assigned a unique height so the connecting cuboids do not interfere.

(i) If $d_{1,4}(s_{i,1})$ is parallel to z, the auxiliary gadgets $\{t_{1,i,j}\}$ can be placed along a 45° line, and by 3-alignments each $t_{2,i,j}$ is leveraged to the height of edge (w_i, w_j).

(ii) Otherwise $d_{1,4}(s_{i,1})$ is parallel to x or y, then each $t_{1,i,j}$ is leveraged to the height of edge (w_i, w_j), and by 3-alignments the gadgets of $\{t_{2,i,j}\}$ are by top view placed along a 45° line.

In conclusion, by the layout of figure 7, auxiliary gadgets $\{t_{2,i,j}\}$ can be placed along a 45° line by top view. For any i, j such that $(w_i, w_j) \in E'$ and $d_{1,4}(s_{i,1}) \neq d_{1,4}(s_{j,1})$, we can always construct $t_{2,i,j} \to t_{3,i,j} \to u_{i,j}$ along x, $t_{2,j,i} \to t_{3,j,i} \to u_{j,i}$ along y, or vice versa. So $u_{i,j}$ and $u_{j,i}$ can meet at the intersecting point and form the biclique of $\{v_1(u_{i,j}), v_4(u_{i,j})\}$ and $\{v_1(u_{j,i}), v_4(u_{j,i})\}$. Therefore the cuboidal dual of G is successfully constructed. □

COROLLARY 1. *Finding a graph's 3-D cuboidal dual is NP-complete.*

4. LAYERED 3-D (2.5-D) CUBOIDAL DUAL OF LAYERED GRAPHS

In the last section we show that general 3-D cuboidal dual is hard. Now we look at the 2.5-D version of the problem which looks less difficult.

4.1 2-D Cuboidal Dual of Planar Graphs

The 2-D "rectangular dual" problem is first studied in [6] and [1]. By using a 4-completion graph, a simple rule to decide if a graph G has a rectangular dual is Theorem 1 of [6]: *A plane graph G with all interior faces triangular has a rectangular dual if and only if there exists a 4-completion of G.* On our definition of cuboidal duals, the deciding rule can be more general and simplified, without using 4-completions.

THEOREM 2. *A graph G has a 2-D cuboidal dual if and only if G can be drawn as a plane graph with no 3-vertex cycle containing interior vertex (vertices).*

This can be proved by converting the given graph to a 4-completion, which is guaranteed to have a rectangular dual by [6], and the cuboidal dual can then be easily obtained. The flow is shown in figure 8. Construction algorithms in linear time of rectangular duals are introduced in [1] and [5].

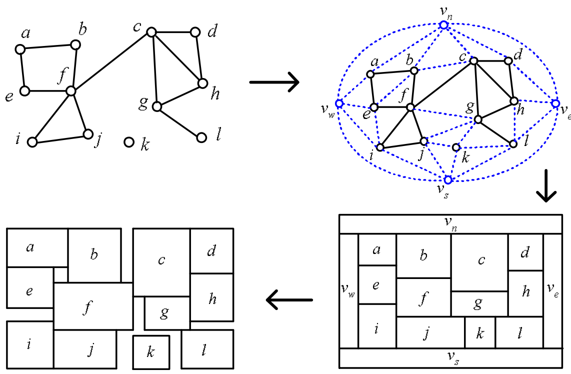


Figure 8: From a graph to its 2-D cuboidal dual

COROLLARY 2. *Finding a graph's 2-D cuboidal dual is in P.*

4.2 2.5-D Cuboidal Dual of Layered Graphs

In the problem here, we are given a layered graph $G = (V, E, n, L : V \rightarrow \{1, \cdots, n\})$, with each vertex assigned a layer and each edge either in a layer or between two consecutive layers, i.e. $(v_i, v_j) \in E \Rightarrow |L(v_i) - L(v_j)| \leq 1$. The 2.5-D cuboidal dual is a 3-D cuboidal dual that each cuboid v_i must be on layer $L(v_i)$. Figure 9 shows an example.

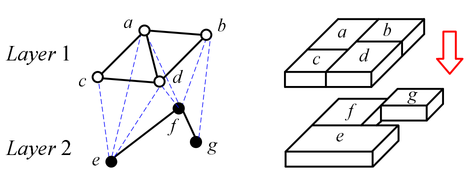
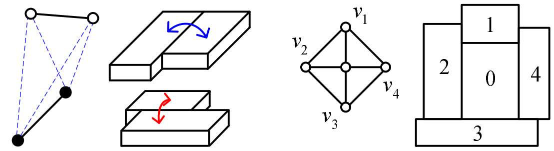


Figure 9: A 2-layer graph and 2.5-D cuboidal dual

The restrictions on cuboids and contacts reduce the freedom of contacting directions. For edge (v_i, v_j), if v_i and v_j are on the same layer, the contacting direction has only 2 choices. Yet we also have some gadgets which introduce complexity.

As figure 10(a) shows, if two vertices on layer i and two vertices on layer $i+1$ are completely connected as a clique K_4, then in the cuboidal dual the contact surface between the two cuboids in layer i must be orthogonal to the one in layer $i+1$. Because as in figure 3, if the two pairs have the same direction, a complete connection is impossible. The diamond gadget in figure 10(b) is similar to the 7-vertex gadget in lemma 1 and figure 2, except it is in 2-D.

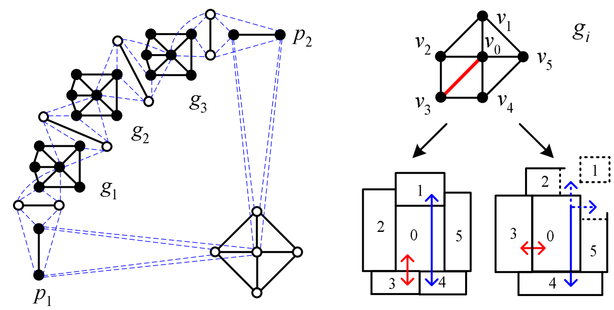


(a) Orthogonal contacts (b) Diamond gadget

Figure 10: Basic gadgets in 2.5-D cuboidal dual

We find that when graph G has 3 layers, deciding its 2.5-D cuboidal dual is no less difficult than Planar 3-SAT, which is proved to be NP-complete in [7].

3-SAT is a basic NP-complete problem introduced in [2]. A Planar 3-SAT instance has the same set of variables $U = \{u_1, \ldots, u_n\}$ and the same set of clauses $C = \{c_1, \ldots, c_m\}$ as 3-SAT. Regarding each variable and clause as a vertex, adding edge (u_i, c_j) if clause c_j contains u_i, the resulting graph G_{p3SAT} is a planar graph.



(a) Clause gadget (b) 6-vertex gadget

Figure 11: 2-layer subgraph of a clause gadget

A gadget for a clause c_i can be constructed as in figure 11(a), where the white vertices are on layer 1 and the black vertices are on layer 2. Two pairs of vertices p_1 and p_2 on layer 2 are enforced to have orthogonal contact surfaces by the diamond gadget on layer 1. Meanwhile the two pairs are also connected through three 6-vertex gadgets which have following properties. Assume the direction of $v_0 \rightarrow v_4$ of such a gadget is determined, (figure 11(b))

(i) if $v_0 \rightarrow v_3$ is on the same direction, i.e. v_3 and v_4 are on the same side of v_0, the 6-vertex gadget acts as a diamond gadget, so v_1 must be on the opposite side of v_0;

(ii) if v_3 is not on the same side of v_4, since this gadget has one more vertex than the diamond gadget, v_1 has the freedom of being on one of the two sides of v_0.

Now we look at figure 11(a), starting from the vertical pair p_1. The first gadget g_1 has vertical $v_0 \rightarrow v_4$ due to the orthogonality enforcements from p_1. By the same enforcements, $v_0 \rightarrow v_1$ of g_1 is parallel to $v_0 \rightarrow v_4$ of g_2, and $v_0 \rightarrow v_1$ of g_2 is parallel to $v_0 \rightarrow v_4$ of g_3. Finally $v_0 \rightarrow v_1$ of g_3 is horizontal as vertex pair p_2.

With these connections, if all the 6-vertex gadgets here have $v_0 \rightarrow v_3$ vertical, then all the gadgets are like the diamond gadget, resulting in $v_0 \rightarrow v_1$ of g_1, g_2 and g_3 must be all vertical, which leads to contradiction, i.e. the 2.5-D cuboidal dual does not exist. Otherwise if we have at least one 6-vertex gadget with $v_0 \rightarrow v_3$ horizontal, then we can place $v_0 \rightarrow v_1$ horizontal on this gadget, and the following gadget also has horizontal $v_0 \rightarrow v_4$. Regardless of the direction of $v_0 \rightarrow v_3$ on following gadgets, we can always make v_1 on the opposite side of v_4, i.e. $v_0 \rightarrow v_1$ horizontal. By this propagation, $v_0 \rightarrow v_1$ of g_3 is horizontal and the 2.5-D cuboidal dual of figure 11(a) can be constructed.

In summary, the 2-layer subgraph of figure 11(a) has a 2.5-D cuboidal dual if and only if at least one 6-vertex gadget has horizontal $v_0 \rightarrow v_3$. This makes the reduction from Planar 3-SAT straight forward, since in 3-SAT a clause is true if and only if at least one of its variables is true.

THEOREM 3. *Planar 3-SAT reduces to 2.5-D cuboidal dual with 3 layers.*

PROOF. We construct a 3-layer graph $G = (V, E, 3, L : V \rightarrow \{1, 2, 3\})$ from $G_{p3SAT} = (U \cup C, E')$ as shown in figure 12. Only two vertices are on layer 3, which are used to align m pairs of vertices on layer 2, by which all the m clause gadgets are aligned in the same direction. Assume the "vertical" direction here is that of the pair on layer 3, as it is drawn in figure 12. n diamond gadgets are placed on layer 2 for the n variables $u_1, \ldots, u_n$.

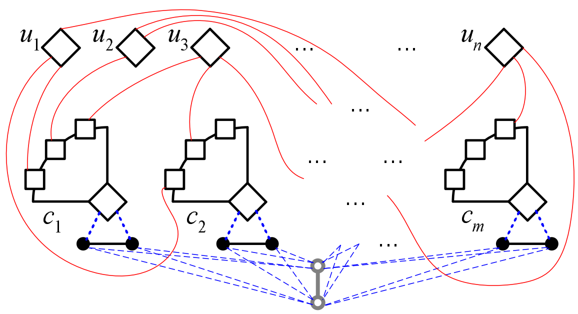


Figure 12: Reduction from Planar 3-SAT to 3-layer 2.5-D cuboidal dual

For each u_i appearing in c_j, we connect the ith diamond gadget $D(u_i)$ to a 6-vertex gadget in the jth clause gadget through $2(m + n)$ diamond gadgets on layer 2, which are connected with vertex pairs on layer 1. The first pair p_1 coming out of $D(u_i)$ is in direction $d_{2,4}(D(u_i))$ if u_i appears as $\overline{u_i}$ in c_j, otherwise in direction $d_{1,3}(D(u_i))$. In this way, vertex pairs of u_i and $\overline{u_i}$ are always orthogonal. The red curves in figure 12 represent the connections of vertex pairs "$p_1 \rightarrow \Diamond \rightarrow p_2 \rightarrow \Diamond \rightarrow \cdots \rightarrow \Diamond \rightarrow p_{2m+2n+1}$". Each diamond gadget ($\Diamond$) both enables and enforces a 90° turn, either left or right, and the final vertex pair $p_{2m+2n+1}$ is orthogonal to $v_0 \rightarrow v_3$ of the 6-vertex gadget.

When there is a solution of the planar 3-SAT boolean formula, we place $D(u_i)$ with $d_{1,3}(D(u_i))$ vertical if u_i is true, or $d_{2,4}(D(u_i))$ vertical if u_i is false. Then for every clause gadget at least one 6-vertex gadget can be constructed with horizontal $v_0 \rightarrow v_1$, so the 2.5-D cuboidal dual of the clause is constructed. Each connection can be placed through the area with $2(m + n)$ 90° turns around the $m + n$ gadgets as obstacles. And since the graph G_{p3SAT} is planar, the connections have no intersections. Although by top view, the connections may intersect with the cuboids on layer 3, we can always pick the connection cuboids on layer 1 to cross the intersection, and there is no contact between layer 1 and layer 3. So the 2.5-D cuboidal dual of G is fully constructed.

When there is no solution of the planar 3-SAT formula, no matter how we place $D(u_1), \cdots, D(u_n)$, there is always a clause gadget with all 6-vertex gadgets aligned in the same direction and therefore inconstructible. □

COROLLARY 3. *Finding a layered graph's 2.5-D cuboidal dual is NP-complete if the number of layers ≥ 3.*

5. CONCLUSIONS

We have looked at three cuboidal dual problems of different dimensions, and come to the results of one efficient algorithm and two hardness proofs. Naturally, the difficulty of the problem migrating from 2-D to 3-D is increasing.

Dimensions	Number of layers	Hardness
2-D	1	P
2.5-D	2	*open*
2.5-D	≥ 3	NP-complete
3-D		NP-complete

A surprising finding is that just a few layers of the 2-D cases, which can be decided by a simple rule (Theorem 2, or [6], [5]), being stacked together, can make the problem so much more complex that there is no effective algorithm can decide the solution, unless P=NP. The relation between topological connections and geometrical contacts in 2-D floorplans is not inherited when extended to 3-D structures. This may also explain why 3-D packing instances are more difficult to encode or represent than 2-D instances.

With the much increased complexity in 3-D structures, we expect a big challenge for both designers and CAD tool developers in future 3-D IC design. Human intelligence will play a more important role in the design flow and in devising heuristic algorithms in 3-D floorplanning, placement and routing tools. Further research will be helpful to understand the nature of 3-D physical design problems.

6. REFERENCES

[1] J. Bhasker and S. Sahni. A linear algorithm to find a rectangular dual of a planar triangulated graph. *ACM/IEEE DAC*, pages 108 – 114, 1986.

[2] S. A. Cook. The complexity of theorem-proving procedures. *3rd ACM STOC*, pages 151 – 158, 1971.

[3] Y. Deng and W. Maly. Physical design of the 2.5D stacked system. *IEEE ICCD*, pages 211 – 217, 2003.

[4] B. Goplen and S. Sapatnekar. Placement of 3D ICs with thermal and interlayer via considerations. *ACM/IEEE DAC*, pages 626 – 631, 2007.

[5] G. Kant and X. He. Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems. *Theoretical Computer Science 172*, pages 175–193, 1997.

[6] K. Kozminski and E. Kinnen. An algorithm for finding a rectangular dual of a planar graph for use in area planning for VLSI integrated circuits. *ACM/IEEE DAC*, pages 655–656, 1984.

[7] D. Lichtenstein. Planar formulae and their uses. *SIAM J. Comput. Volume 11, Issue 2*, pages 329–343, 1984.

[8] S. K. Lim. Physical design for 3D system-on-package: Challenges and opportunities. *IEEE DATE*, pages 532–536, 2005.

[9] G.-M. Wu, J.-M. Lin, and Y.-W. Chang. An algorithm for dynamically reconfigurable FPGA placement. *IEEE ICCD*, pages 501 – 504, 2001.

[10] H. Yamazaki, K. Sakanushi, S. Nakatake, and Y. Kajitani. The 3D-packing by meta data structure and packing heuristics. *IEICE Trans. Fundamentals*, pages 639–645, Apr. 2000.