Complexity of 3-D Floorplans by Analysis of Graph Cuboidal Dual Hardness

RENSHEN WANG
University of California, San Diego
EVANGELINE YOUNG
The Chinese University of Hong Kong
and
CHUNG-KUAN CHENG
University of California, San Diego

Interconnect dominated electronic design stimulates a demand for developing circuits on the third dimension, leading to 3-D integration. Recent advances in chip fabrication technology enable 3-D circuit manufacturing. However, there is still a possible barrier of design complexity in exploiting 3-D technologies. This article discusses the impact of migrating from 2-D to 3-D on the difficulty of floorplanning and placement. By looking at a basic formulation of the graph cuboidal dual problem, we show that the 3-D cases and the 3-layer 2.5-D cases are fundamentally more difficult than the 2-D cases in terms of computational complexity. By comparison among these cases, the intrinsic complexity in 3-D floorplan structures is revealed in the hard-to-decide relations between topological connections and geometrical contacts. The results show possible challenges in the future for physical design and CAD of 3-D integrated circuits.

Categories and Subject Descriptors: B.7.2 [**Integrated Circuits**]: Design Aids; J.6.1 [**Computer-Aided Engineering**]: —*Computer-aided design*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: 3-D integration, floorplanning, cuboidal dual, hardness

ACM Reference Format:

Wang, R., Young, E., and Cheng, C.-K. 2010. Complexity of 3-D floorplans by analysis of graph cuboidal dual hardness. ACM Trans. Des. Autom. Electron. Syst. 15, 4, Article 33 (September 2010), 22 pages. DOI = 10.1145/1835420.1835426 http://doi.acm.org/10.1145/1835420.1835426

This article is based on work previously published as "On the complexity of graph cuboidal dual problems for 3-D floorplanning of integrated circuit design," in *Proceedings of the Great Lakes Symposium on Very-Large-Scale Integration (GLSVLSI) 2009*, 257-261. ©2009 ACM.
Authors' addresses: R. Wang and C.-K. Cheng, Department of Computer Science and Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0404; email: rewnang@cs.ucsd.edu; E. Young, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin N.T. Hong Kong.

DOI 10.1145/1835420.1835426 http://doi.acm.org/10.1145/1835420.1835426

1. INTRODUCTION

Interconnects are dominating the performance of VLSI circuits, and consuming an increasingly large portion of power in VLSI systems. Scaling of wires [Ho et al. 2001] implies that global interconnect is becoming the bottleneck of system power and performance. New technologies such as three-dimensional integration are a possible solution to reduce interconnect length and to keep the continuation of Moore's law in the nano era. By adding a dimension in current 2-D VLSI circuits, we can greatly enhance integration density and reduce interconnect wire length, thus resolving the largest bottleneck. Ideally, by using n-layer circuit we reduce the interconnect length by a factor of $\sqrt{n}$. Meanwhile, the added dimension may also bring higher complexity and difficulty in design, CAD tools, and fabrication. To fully exploit the advantages of the third dimension in 3-D ICs, we need to measure and understand the complexity it brings, and estimate the hardness in optimization problems for designing 3-D circuits.

Floorplanning and placement can be a design phase that will become much more difficult when migrating from 2-D to 3-D, as the basic blocks change from rectangles to cuboids, and the design space gains a dimension. Regarding the circuit building blocks, there are two classes of 3-D circuits discussed in the literature. Currently 3-D circuits and system-on-chips [Lim 2005] are usually achieved by die stacking, which is a stack of layers of 2-D circuits with identical thickness. This type of placement is also called 2.5-D placement [Deng and Maly 2003] since it does not really contain full 3-D structures. Previous works like Goplen and Sapatnekar [2007] and Yu et al. [2006] also discuss this 2.5-D type.

Full 3-D floorplan and placement representations have been explored in several works since Yamazaki et al. [2000]. Full 3-D means the circuit blocks are cuboids placed freely in a 3 dimensional 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 field programmable gate arrays (FPGAs) [Wu et al. 2001] where time is regarded as another dimension. It is discovered that most of the floorplan representations effective in 2-D do not have an equally effective extension in 3-D, for example, the extension from sequence-pair (2-D) to sequence-triple (3-D) [Yamazaki et al. 2000]. The intrinsic complexity in 3-D structures is a possible reason for these results. 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 Kozminski and Kinnen [1984] and Kant and He [1997], except that our problem is in 3-D and allows empty space in solutions, while in Kozminski and Kinnen [1984] a solution must be a full rectangular dissection. Although the size of each cuboid is not limited, which means the set of cuboids is not naturally a placement of circuit modules, the solution still has practical meanings in the initial

Table I. k-Dimensional Cuboidal Dual Problems

Floorplan Dimensions	Number of Layers	Hardness of Problem
2-D	1	P
2.5-D	2	*open*
2.5-D	≥ 3	NP-complete
Full 3-D	*undefined*	NP-complete

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 heat density, we make $(v_i, v_j) \notin E$. More importantly, this problem differentiates the 2-D and 3-D structures as in following paragraphs.

A 2-D rectangular dual can be decided by a set of conditions in Kozminski and Kinnen [1984] or Kant and He [1997] and can be efficiently generated in linear time by algorithms in Bhasker and Sahni [1986] or Kant and He [1997]. For cuboidal duals, while the 2-D case can be solved with a similar approach, we find the 3-D cases are fundamentally harder in terms of computational complexity. Just as the 2-colorability problem is easy but 3-colorability is NP-hard, one extra color or dimension brings a fundamentally higher level of complexity. In fact we prove the 3-D cuboidal dual problem is NP-complete just by reducing 3-colorability to it. For the 2.5-D case of the cuboidal dual, although the problem looks very similar to the 2-D case, we find it also to be NP-complete when the number of layers reaches 3. The difficulties of all the cases are listed in Table I.

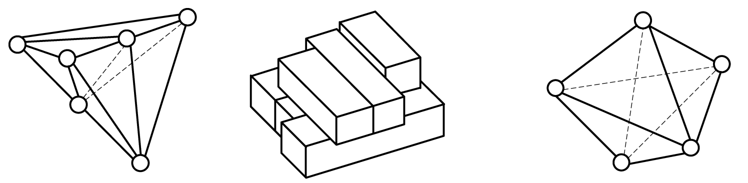
These results imply that the complexity of circuit floorplanning and placement design can be greatly increased when we extend the circuit to the third dimension, even if we limit the extension to just a few layers. To achieve practical 3-D ICs, we are facing a serious challenge of design complexity, thermal behavior, and manufacturing issues.

The rest of this article discusses the details of all the indicated results and the corresponding proofs. Section 2 introduces the basic problem formulation and the difficulty of the upper-bound. Section 3 proves the general 3-D cuboidal dual problem is hard. Section 4 shows that although finding a 2-D cuboidal dual is easy, finding a 2.5-D cuboidal dual with 3 layers is hard. Finally Section 5 gives comparisons and conclusions on these cases.

2. PRELIMINARIES AND FORMULATIONS

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 cuboids. A 2-D case can be regarded as a 3-D case with all the cuboids placed on the floor.

An adjacency graph can be constructed from a floorplan by assigning a vertex to each cuboid and adding on 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 Kozminski and Kinnen [1984] there is a set of necessary and sufficient conditions for a graph to be an



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

Fig. 1. Graph-floorplan relations.

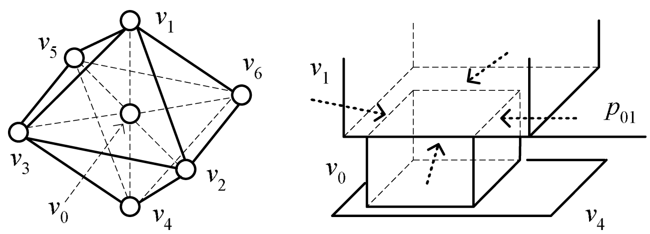


Fig. 2. 7-vertex gadget and part of its cuboidal dual.

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 general 3-D cuboidal dual of an n-vertex graph $G = (V, E)$ is defined as a set of axis-parallel cuboids, where each cuboid, C_i, corresponds to a vertex, $v_i \in V$. No two cuboids share a common space. C_i and C_j are adjacent (contacting on a surface 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 clique, which has no cuboidal duals.

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

A 2-D cuboidal dual is defined as a 2.5-D cuboidal dual with only one layer, that is, every cuboid is placed in height interval [0,1]. It is different from a rectangular dual [Kozminski and Kinnen 1984] in that the set of cuboids can be a subset of a space dissection, so empty space is allowed.

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 cases, the problem is trivially in NP, because it takes polynomial time to verify whether a given set of cuboids is a solution. For example, we check each pair (i, j) to determine whether $(v_i, v_j) \in E \Leftrightarrow C_i$ and C_j are contacting on surfaces, which can be done in $O(n^2)$ time.

3. 3-D CUBOIDAL DUAL OF GENERAL GRAPHS

To decide whether a graph has 3-D cuboidal dual is NP-hard. We prove this by reducing the 3-colorability problem to a 3-D cuboidal dual. It is well known that 3-colorability is NP-complete [Karp 1972]. We construct G from a 3-colorability instance, $G_{3C} = (W, E')$. In the first step, we introduce a gadget of 7 vertices for each vertex in W, as shown in Figure 2.

The 7 vertices together with the edges form an octahedron composed of 8 small tetrahedrons. There is a cuboidal dual of this graph, and the contact surfaces between different pairs of cuboids are not independent of each other.

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 (the cuboid of v_0).*

PROOF. Since cuboid v_0 and cuboid v_1 are adjacent, they are contacting on a common plane (we denote this plane as p_{01}). Their contacting area on p_{01} must be a rectangle, denoted as R, as pointed to by the arrows in Figure 2. Since rectangle R is the only common part of the two cuboids, by the convexity of cuboids, another cuboid (such as v_2) contacting both v_0 and v_1 must be on the boundary of R. Cuboids v_2, v_3, v_5, v_6 are four such cuboids in the gadget, with a closed loop $v_2 - v_3 - v_5 - v_6 - v_2$ of edges. The only way to place v_2, v_3, v_5, v_6 is surrounding the boundary of R, because R's inside area is occupied by v_0 and v_1.

Now we project all seven cuboids onto p_{01}. We have v_2, v_3, v_5, v_6 surrounding rectangle R. Since v_4 is adjacent to these four vertices, the projection of cuboid v_4 on p_{01} must overlap with the area surrounding R—the projection of v_4 covers rectangle R. So on cuboid v_0's six surfaces, cuboid v_4:

—cannot be on the same surface as v_1 because v_4's projection on p_{01} overlaps with R (which is part of v_1's projection);

—cannot be on the four adjacent surfaces of v_1's side because v_4's projection on p_{01} overlaps with R (which is part of v_0's projection).

In conclusion, cuboid v_4 can only be on the surface of v_1's opposite side. □

Lemma 1 implies that the contacting directions of $v_1 \rightarrow v_0$ and $v_0 \rightarrow v_4$ are same; we denote this direction as $d_{1,4}$.

Definition 1. For a 7-vertex gadget N, $d_{1,4}(N)$ denotes the axis (among x, y, z) whose orthogonal plane has overlapping projections from cuboids v_1 and v_4.

In the same manner as Lemma 1, the other two pairs of vertices (v_2, v_5) and (v_3, v_6) also have cuboids on opposite sides of v_0. Another conclusion from Figure 2 is that the cuboids of $v_1, v_2, \cdots, v_6$ cover all 6 surfaces of cuboid v_0; proved as follows.

LEMMA 2. *In the cuboidal dual of the 7-vertex gadget, the six cuboids of $v_1, \cdots, v_6$ each entirely covers one surface of cuboid v_0.*

PROOF. Cuboid v_2 is contacting cuboid v_4, which by Lemma 1 is on the opposite side of cuboid v_1's surface. By the convexity of a cuboid, v_2 cannot be on the same surface as v_1. The same holds for v_3, v_5, and v_6. Thus, cuboid v_1 does not share v_0's surface with any other cuboid. By the symmetry of the gadget, the six cuboids do not share any of v_0's six surfaces, so each covers one. And since each cuboid is contacting four neighbors on adjacent surfaces, it must reach the four edges of its side of v_0; therefore it entirely covers the surface. □

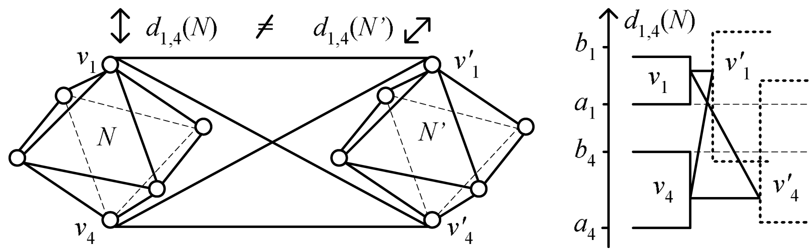


Fig. 3. Enforcing 2 gadgets in different directions.

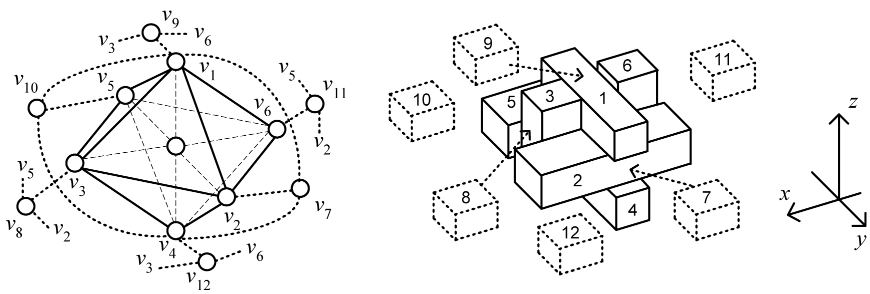


Fig. 4. 13-vertex gadget and its cuboidal dual.

Regarding the coordinate axis that $d_{1,4}(N)$ 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, representing a vertex w, in $G_{3C} = (W, E')$, is colored as $d_{1,4}(N)$. The constraint of 3-colorability is that for edge $(w, w') \in E'$ in G_{3C}, the two vertices cannot share the same color. This constraint can be implemented as a bi-clique connection between v_1 and v_4, of two gadgets N and N'. As Figure 3 shows, we look at an axis parallel to direction $d_{1,4}(N)$, where v_1 occupies interval $[a_1, b_1]$ and v_4 occupies interval $[a_4, b_4]$. If there is a bi-clique connection between $\{v_1, v_4\}$ and $\{v'_1, v'_4\}$, then both v'_1 and v'_4 must cover the interval between v_1 and v_4, which is $[b_4, a_1]$ on the axis, because the cuboids of v'_1 and v'_4 are contacting v_1 and v_4. So the two cuboids of v'_1 and v'_4 have an overlapping interval on this axis, which means $d_{1,4}(N')$ must be in a different direction, and thus $d_{1,4}(N') \neq d_{1,4}(N)$.

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

LEMMA 3. *(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 cuboids v_1 *and* v_4 *have the same width as cuboid* v_0 *(along* $d_{3,6}$*),*
cuboids v_2 *and* v_5 *have the same height as cuboid* v_0 *(along* $d_{1,4}$*),*
cuboids v_3 *and* v_6 *have the same length as cuboid* v_0 *(along* $d_{2,5}$*).*

PROOF. As Figure 4 shows, we already know from Lemma 1 and Lemma 2 that the six cuboids of $v_1, \cdots, v_6$ are each placed with pairs (v_1, v_4), (v_2, v_5), (v_3, v_6) on opposite sides of cuboid v_0, and the six surfaces of cuboid v_0 are covered by $v_1, \cdots, v_6$. In this coordinate system, $d_{1,4}$ of the gadget is parallel to axis z.

Consider the cuboid of v_7 contacting three cuboids of v_1, v_2, and v_4. It must contact cuboid v_2 on the $+y$ side, because:

—cuboid v_2 and v_7 are contacting both v_1 and v_4, so on their projections to axis z (or $d_{1,4}$), v_7 overlaps with v_2, which means v_7 cannot be at $+z$ or $-z$ side of v_2;

—if cuboid v_7 is contacting cuboid v_2 on the $-y$ side, without loosing generality we assume it is on the $+x$ side of cuboid v_3, v_0, and v_6. To make cuboid v_7 contact cuboid v_1 and v_4, we need $x_{1,+} \geq x_{3,+}$ and $x_{4,+} \geq x_{3,+}$ (denoting $x_{k,+}$ as the x coordinate of cuboid k's $+x$ surface). Cuboid v_8 is contacting cuboid v_2, v_3, and v_5, we have $x_{5,+} \geq x_{3,+}$. So in the projection on the xy plane, rectangle v_3 is completely contained in rectangle v_1 and rectangle v_4. Thus, cuboid v_3 has only the $+x$ surface not covered. But it is impossible to make cuboid v_9 on v_3's $+x$ surface while also contacting cuboid v_1 and v_6, which leads to a contradiction;

—if cuboid v_7 is contacting cuboid v_2 on the $+x$ direction (the case for the $-x$ direction is symmetric), then cuboid v_2 cannot protrude among cuboids v_1 and v_4 on the $+x$ direction, that is we have $x_{2,+} \leq x_{1,+}$ and $x_{2,+} \leq x_{4,+}$. So when cuboid v_8 is contacting cuboids v_2, v_3, and v_5, it cannot be contacting the $+z$ or $-z$ surfaces of v_3, so it must be contacting v_3's $+x$ surface, which leads to $x_{2,+} \geq x_{8,-} = x_{3,+}$, and therefore we have $x_{1,+} \geq x_{3,+}$ and $x_{4,+} \geq x_{3,+}$. Hence, rectangle v_3 is also completely contained in rectangle v_1 and rectangle v_4 in the projection on the xy plane, which leads to the same contradiction.

By symmetry, cuboid v_9 must be on the $-y$ side of cuboid v_5, cuboid v_8 must be on the $+x$ side of cuboid v_3, cuboid v_{11} must be on the $-x$ side of cuboid v_6, cuboid v_9 must be on the $+z$ side of cuboid v_1, and cuboid v_{12} must be on the $-z$ side of cuboid v_4. Thus, the contacting surfaces among the cuboids of $v_1, \cdots, v_6$ must be in exactly the same topology as in Figure 4. □

By constructing this 13-vertex gadget, the original 7-vertex gadget has a definite shape, and we can easily align multiple gadgets in the same direction with some additional vertices in G as follows.

In Figure 5 we use a simplified octahedron to represent each 13-vertex gadget, and add two vertices, v_A and v_B. Consider their connections with gadget N' on the right. Since v_A is simultaneously contacting v'_1, v'_2, and v'_3, by Lemma 3 and Figure 4, cuboid v_A must be on the corner formed by the 3 cuboids, and is therefore contacting v'_2 from above. Similarly, v_B is contacting v'_4, v'_2, and v'_3, so cuboid v_B must be on the corner and contacting v'_2 from below. As a result, the direction $v_A \rightarrow v_B$ is the same as $d_{1,4}(N')$.

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

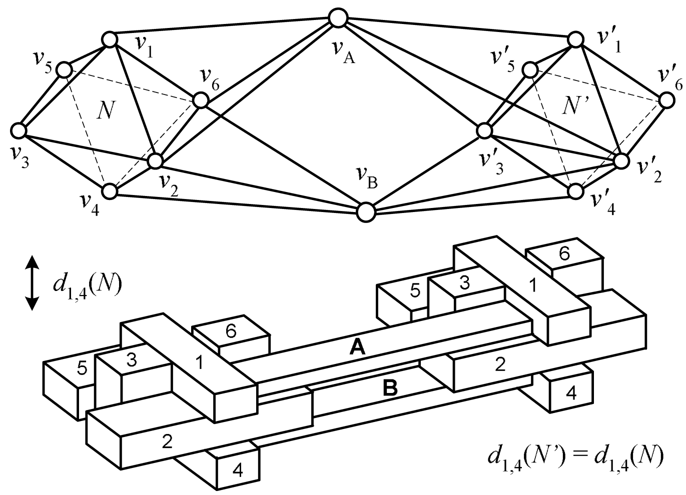


Fig. 5. Two 13-vertex gadgets, N and N', with $d_{1,4}(N)$ and $d_{1,4}(N')$ aligned to the same direction (2-alignment).

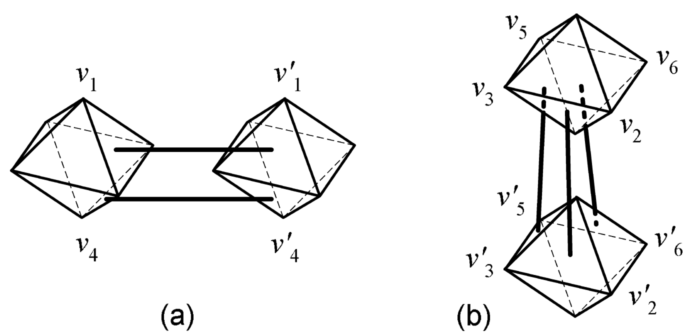


Fig. 6. 2-alignment and 3(complete)-alignment.

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

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 aligned in every direction.

Also notice that the direction of displacement from one gadget to the other in a 2-alignment is along $d_{2,5}(N)$ or $d_{3,6}(N)$, while the displacement in a 3-alignment must be along $d_{1,4}(N)$. These two cases of alignment enable the alignment of a pair of 13-vertex gadgets, N and N', along any of the $\{x, y, z\}$ axes, and always guarantee $d_{1,4}(N) = d_{1,4}(N')$. Based on these cases, we can construct large sets of 13-vertex gadgets, all aligned in the same direction, each set acting as a single vertex in $G_{3C} = (W, E')$, which helps to complete the reduction from 3-colorability.

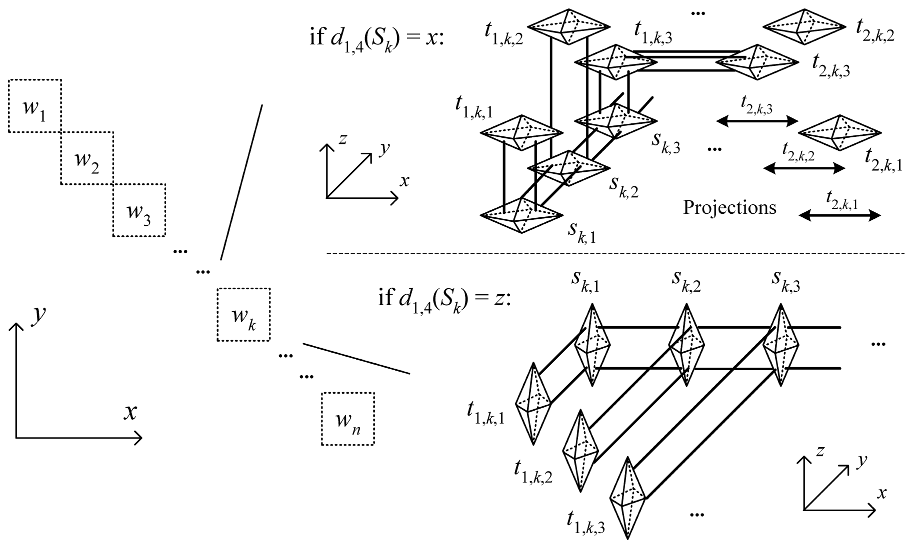


Fig. 7. Construction of cuboidal dual from G when G_{3C} is 3-colorable.

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

PROOF. We use the 13-vertex gadget as a vertex cluster node in our construction of G. Given a graph of 3-colorability $G_{3C} = (W, E')$ with n vertices denoted as $w_1, w_2, \cdots, w_n$, for each vertex w_i, we 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 and v_4 of $u_{i,j}$ with v_1 and v_4 of $u_{j,i}$, so that the two gadgets $u_{i,j}$ and $u_{j,i}$ are enforced in different directions (as in Figure 3). In this way, graph G has a cuboidal dual *if and only if* G_{3C} is 3-colorable.

The "if and only if" holds because of the gadget sets in G constructed from the vertices of G_{3C}. Assume G_{3C} has n vertices. On each vertex, w_i, of G_{3C}, we put $5n$ gadgets $s_{i,1} \sim s_{i,n}$, $t_{1,i,1} \sim t_{1,i,n}$, $t_{2,i,1} \sim t_{2,i,n}$, $t_{3,i,1} \sim t_{3,i,n}$, and $u_{i,1} \sim u_{i,n}$. The $d_{1,4}$ directions of all these gadgets are the same, because they are connected by either 2-alignments or 3-alignments. Denote these $5n$ gadgets as set S_i, and the direction $d_{1,4}(S_i)$ has 3 choices x, y, or z, just like the color of vertex w_i in G_{3C} has 3 choices. And by enforcing $u_{i,j}$ and $u_{j,i}$ in different directions for edge $(w_i, w_j) \in E'$, we also enforce sets S_i and S_j in different directions.

From left to right, if graph G has a cuboidal dual, we color each vertex w_i by the $d_{1,4}$ direction of gadgets in S_i. By the alignments and enforcements among gadgets in G, w_i and w_j have different colors whenever there is an edge between w_i and w_j. Therefore, we have a valid 3-coloring on G_{3C}.

On the other hand, if G_{3C} is 3-colorable, then we can construct a cuboidal dual according to Figure 7. The $s_{i,j}$ gadgets of G_{3C}'s vertices $w_1, \cdots, w_n$, are placed on the xy-plane and with a top view of Figure 7. Each G_{3C}'s vertex w_k

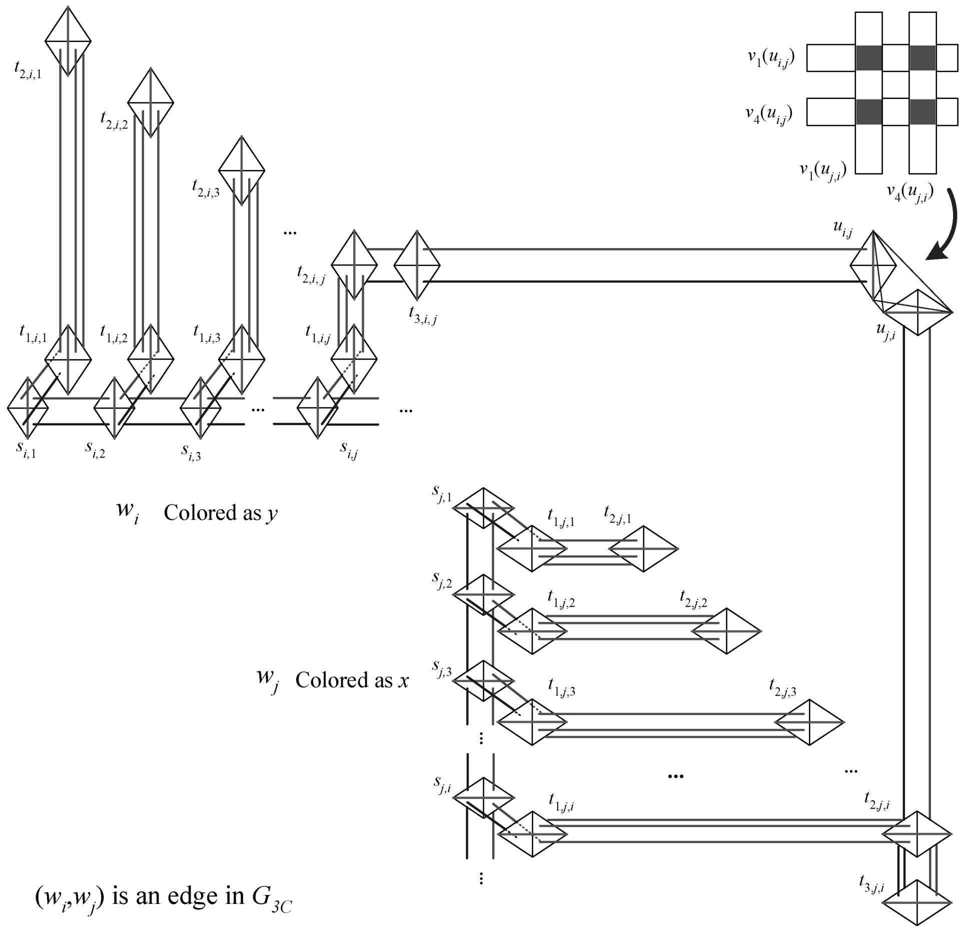


Fig. 8. Connection in the cuboidal dual for an edge in G_{3C}.

has a square area, and is assigned with a direction in $\{x, y, z\}$ according to the coloring of G_{3C}, which decides the direction of the gadget set $d_{1,4}(S_k)$. Every edge (w_i, w_j) in E' has a connection through gadgets as constructed in G. We assign a unique height $(H_{i,j})$ to every edge, so that out of the square areas of w_i and w_j, at height $H_{i,j}$ there are only connection cuboids between w_i and w_j. Thus, we guarantee different edges can have series of gadget connections constructed without conflict. Details of the gadgets' placement are revealed in Figures 7 and 8, explained as follows.

(1) 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 $H_{i,j}$;

(2) 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 $H_{i,j}$, and by 3-alignments the gadgets of $\{t_{2,i,j}\}$ are by top view placed along a 45° line.

Therefore, by the layout of Figure 7, auxiliary gadgets $\{t_{2,i,j}\}$ can always be placed along a 45° line by top view. Then for any (i, j) such that there is an edge

$(w_i, w_j) \in E'$ with $d_{1,4}(s_{i,1}) \neq d_{1,4}(s_{j,1})$, we can always construct $t_{2,i,j} \rightarrow t_{3,i,j} \rightarrow u_{i,j}$ along x, $t_{2,j,i} \rightarrow t_{3,j,i} \rightarrow u_{j,i}$ along y, or vice versa. These gadgets are on the height of $H_{i,j}$, which prevents conflicts with other edges. So as shown in Figure 8, $u_{i,j}$ and $u_{j,i}$ can finally make a connection at the intersecting point and form the biclique of $\{v_1(u_{i,j}), v_4(u_{i,j})\}$ and $\{v_1(u_{j,i}), v_4(u_{j,i})\}$. In this way we construct the edge connections, and the cuboidal dual construction of G is complete.

The reduction is thus proved. This is a polynomial reduction, because for each vertex in G_{3C}, we construct $5n$ gadgets in G, therefore graph G's size is on $O(n^2)$ scale of G_{3C}'s size. □

COROLLARY 1. *The problem of 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 showed that general 3-D cuboidal dual is hard. Now we look at the 2.5-D version of the problem, which looks less complex. We start with the basic single-layer cuboidal dual of a planar graph G.

4.1 2-D Cuboidal Dual of Planar Graphs

The 2-D rectangular dual problem was first studied in Kozminski and Kinnen [1984] and Bhasker and Sahni [1986]. By using a 4-completion graph, a simple rule to decide if a graph G has a rectangular dual is Theorem 1 of Kozminski and Kinnen [1984].

A planar graph G drawn on a plane with all triangular interior faces has a rectangular dual if and only if there exists a 4-completion of G. A *4-completion graph is a graph H drawn on a plane such that*:

(a) the exterior face of H has 4 vertices;

(b) each interior face of H has 3 vertices;

(c) each cycle in H that is not a face has length ≥ 4

A 4-completion of G is a 4-completion graph H that G can be obtained by deleting the 4 exterior vertices of H.

On our definition of cuboidal duals, which allows empty space between cuboids, the deciding rule becomes more general and simplified.

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

PROOF. Given a planar graph G drawn without non-empty 3-vertex cycles, we add 4 exterior vertices v_n, v_s, v_w, v_e, and then add edges so that the resulting graph H is a 4-completion graph.

Starting from the exterior 4-vertex cycle of v_n, v_s, v_w, v_e, we first make the graph 2-connected. For any vertex or 2-connected component v inside a cycle C_1 with at least 4 vertices, (Figure 9(a)):

(1) If v is 1-connected to C_1 by vertex v_1, we find another vertex v_2 on C_1 with no edge to v_1. Such a v_2 exists, otherwise every vertex is connected to v_1, then either C_1 is triangular, or C_1 itself is contained in a 3-vertex cycle, and both

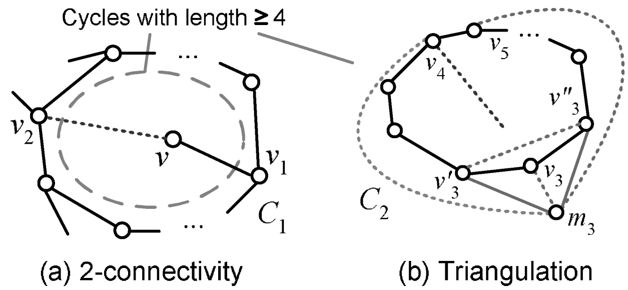


Fig. 9. Triangulation without producing non-empty 3-vertex cycles.

situations contradict our given condition. Add an edge (v, v_2), which makes v 2-connected to C_1. C_1 is then decomposed into 2 smaller cycles, both with length ≥ 4 since there is no edge (v_1, v_2).

(2) If v is not connected to C_1, pick any vertex v_1 on C_1 and add edge (v, v_1). v and C_1 then become 1-connected, and 2-connectivity can be achieved by case (1).

After the graph is 2-connected, every face is a cycle with no repeated vertices and no interior vertices. We then triangulate every face with more than 3 vertices, so that every face except the outer boundary is a triangle.

As shown in Figure 9(b), for a face cycle with length ≥ 4, denote it as C_2 and pick any vertex v_3 on C_2. The two neighbors of v_3 on C_2 are denoted as v_3' and v_3''.

(1) If there is vertex v_4 on C_2 not connected with v_3, and there is no vertex out of C_2 connected with both v_3 and v_4, then we add edge (v_3, v_4), which does not induce non-empty 3-vertex cycles.

(2) Otherwise, add edge (v_3', v_3''). In this case every vertex on C_2 except $\{v_3, v_3', v_3''\}$ has a common neighbor with v_3. If v_3' and v_3'' have a common neighbor m_3, then all the common neighbors of v_3 and other vertices must also be m_3, and we have a 3-vertex cycle $m_3 \to v_4 \to v_5 \to m_3$ containing entire C_2, which contradicts the condition. So v_3' and v_3'' have no common neighbor out of C_2 and adding edge (v_3', v_3'') does not induce non-empty 3-vertex cycles.

Repeating this triangulation until the graph is completely triangulated, we get a 4-completion graph H. By Theorem 1 of Kozminski and Kinnen [1984] there is a rectangular dual of H. To obtain the 2-D cuboidal dual of the original graph G, we remove the 4 rectangles on the outer boundary that correspond to the 4 added vertices. For each edge (v_i, v_j) in H that is not in G, we add a gap between the contacting rectangles so that they are no longer contacting, and the final set of rectangles is the cuboidal dual of G.

The operation of adding a gap can be done individually for each (v_i, v_j) in H but not in G. There are two cases, as shown in Figure 10.

(1) If v_i's contacting edge is contained in v_j's contacting edge, then we only move v_i's edge backward by ϵ. ϵ takes a value less than the minimum distance between any pair of existing points in the rectangle set, so that the only change in the contacting topology is between v_i and v_j.

(2) If neither contacting edge is contained in the other, we denote their common edge as l with end points AB (Figure 10). For each edge on the infinite

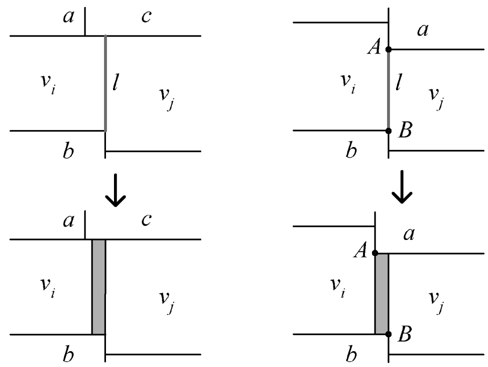


Fig. 10. Addins gas between rectangles for edges not in G.

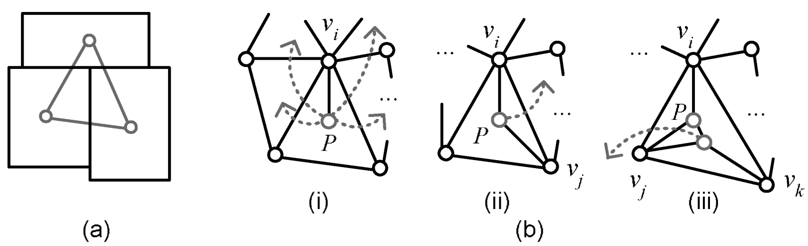


Fig. 11. Empty 3-vertex cycle and operations to resolve non-empty 3-vertex cycle.

line of l, if it is above A or it is v_i's right edge, move it leftward by ϵ; if it is below B or it is v_j's left edge, no operation. Again ϵ is less than the minimum existing distance between points. The operation may involve rectangles other than v_i and v_j, but still the only change in topology is between v_i and v_j.

Finally, if graph G unavoidably contains a non-empty 3-vertex cycle, there is no 2-D cuboidal dual. Because as in Figure 11(a), the 3 contact surfaces of the 3 cuboids on the cycle form a T shape on the plane without internal space, so there is no room for the components inside the 3-vertex cycle. □

When the given graph G is not drawn on a plane, the planar graph testing and embedding algorithm in Chiba et al. [1985] can be used to test planarity and embed G into the plane in linear time. If G is not planar, clearly there is no 2-D cuboidal dual. Otherwise G is planar, and we draw a plane and resolve (if possible) all the non-empty 3-vertex cycles one by one in G with following operations.

Assuming G is connected (otherwise we run the algorithm on each connected component), for each non-empty 3-vertex cycle C with an interior component P, as illustrated in Figure 11(b):

(1) if P is connected to one of the vertices on the cycle (denoted v_i), and there is another face on v_i with with cycle length ≥ 4, move P into that face;

(2) if P is connected to two of the vertices on the cycle (denoted v_i, v_j), and there is another face F' containing both v_i and v_j with cycle length ≥ 4, there are two cases:

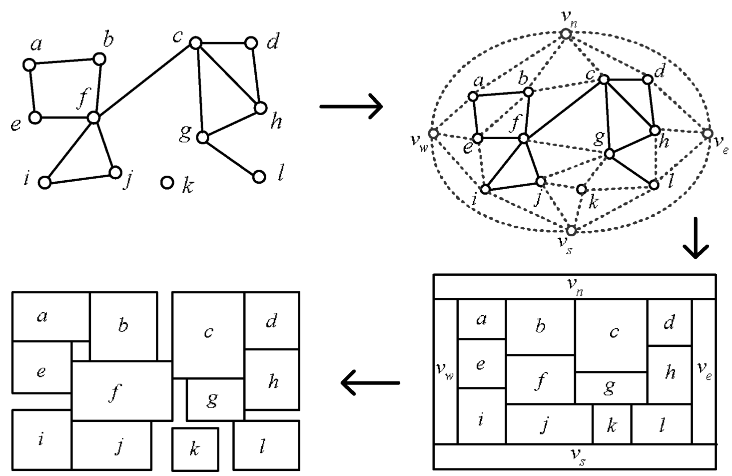


Fig. 12. From a graph to its 2-D cuboidal dual.

—if v_i's neighbor in P and v_j's neighbor in P are not the same vertex, move P into face F';

—if v_i's neighbor in P and v_j's neighbor in P are the same vertex, only when face F'' on the other side of edge (v_i, v_j) has cycle length ≥ 4, move P into face F'';

(3) if P is connected to all the three vertices on the cycle (denoted v_i, v_j and v_k), and there is another face F' with cycle length ≥ 4 and containing both edges of (v_i, v_j) and (v_j, v_k), also v_i's neighbor in P and v_k's neighbor in P are not the same vertex, then we can move P into face F'.

Repeating these three operations, if all the non-empty 3-vertex cycles are resolved, a 2-D cuboidal dual exists. Otherwise, either in case (1), (2) or (3), there is either no adjacent face with cycle length ≥ 4 to place P, or $v_i \to P \to v_{j/k}$ form another 3-vertex cycle, there is no solution to draw G without a non-empty 3-vertex cycle, so G has no 2-D cuboidal dual.

Multiple algorithms for constructing a rectangular dual from G are introduced in Kozminski and Kinnen [1984], Bhasker and Sahni [1986], and Kant and He [1997], among which Bhasker and Sahni [1986] and Kant and He [1997] provide the constructions in linear time. Based on these algorithms, plus the transformation from G to a 4-completion graph, which can also be done in linear time, a 2-D cuboidal dual can be constructed from a planar graph G in linear time. Figure 12 shows an example of the construction flow.

COROLLARY 2. *The problem of finding a graph's 2-D cuboidal dual is in P.*

4.2 2.5-D Cuboidal Dual of Layered Graphs

With the 2-D cuboid dual problem (single layer case of 2.5-D) solved, we see how the difficulty of the problem differs from 2-D to 3-D, and the fundamental complexity with 3-D structures. To further explore the whole set of problems, we look at something between these two versions—a multi-layer 2-D version.

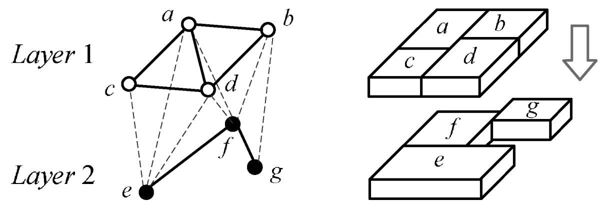


Fig. 13. A 2-layer graph and 2.5-D cuboidal dual.

Multi-layer floorplans are usually called 2.5-D because of their characteristics between 2-D and 3-D. For this purpose, we first define a layered graph.

Definition 2. In a *k-layer graph* $G = (V, E, L : V \mapsto \{1, \cdots, k\})$, each vertex is assigned with a layer between 1 and k, and each edge is either in a layer or between two consecutive layers: $(v_i, v_j) \in E \Rightarrow |L(v_i) - L(v_j)| \leq 1$.

In this problem, we are given a layered graph $G = (V, E, L : V \rightarrow \{1, \cdots, k\})$. The 2.5-D cuboidal dual of G is a special 3-D cuboidal dual in which the cuboid of v_i has fixed z interval of $[L(v_i) - 1, L(v_i)]$. Figure 13 shows an example of a 2-layer graph with its 2.5-D cuboidal dual.

The added constraints on cuboids and contacts lower the freedom of contacting directions. For edge (v_i, v_j), if v_i and v_j are on the same layer, their contacting direction has 2 choices, otherwise $|L(v_i) - L(v_j)| = 1$ and the contact surface must be parallel to the layer dividing planes. However, although the gadgets used in Section 3 have no 3-dimensional freedom, we have other gadgets making the problem harder than the single layer case.

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 Lichtenstein [1984]. 3-SAT is a basic NP-complete problem introduced in Cook [1971]. A Planar 3-SAT instance has the same set of variables $U = \{u_1, \ldots, u_n\}$ and set of clauses $C = \{c_1, \ldots, c_m\}$ as 3-SAT. But in Planar 3-SAT, if we regard each variable or clause as a vertex and add edge (u_i, c_j) if clause c_j contains variable u_i, the resulting graph G_{p3SAT} is a planar graph.

Before we reduce Planar 3-SAT to 2.5-D cuboidal dual, we need a planar graph's rectilinear path embedding on a plane, which can be converted from its Fáry embedding (straight line embedding). For convenience, we use the straight line embedding on the $(n-2) \times (n-2)$ integer grid, which is from Schnyder [1990].

LEMMA 4. *Given a planar graph G_p with n vertices, we can place the set of vertices on a $(n-2) \times (n-2)$ integer grid, such that if each vertex v_i has a non-zero diameter, then each edge (v_i, v_j) has a rectilinear path from v_i to v_j with no more than $2n - 4$ corners, and without intersections between paths.*

PROOF. By Schnyder [1990], there is a straight line embedding of G_p on the $(n-2) \times (n-2)$ integer grid, so that all the vertices are on the grid and edges only intersect at end points.

Starting with this embedding, for each edge that is neither horizontal nor vertical, we take its slope-intercept form $y = mx + b, x \in [x_l, x_r]$. By using

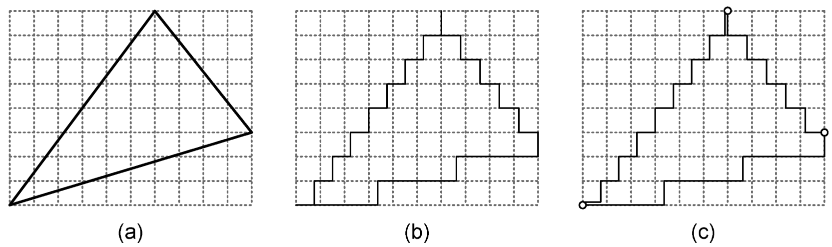


Fig. 14. Converting a straight line embedding into a rectilinear path embedding.

the floor function ($\lfloor x \rfloor$ is the largest integer not greater than x), we turn this line into $y = \lfloor mx + b \rfloor$ which is a series of horizontal segments. Connecting these segments in order by vertical segments, the straight line is turned into a rectilinear path. Figure 14(a→b) is an example of this step.

The rectilinear paths have the following three properties.

(1) Each path has no more than $2n - 4$ corners. Because the vertices are on an $(n-2) \times (n-2)$ integer grid, so any path from (x_1, y_1) to (x_2, y_2) goes through $|y_1 - y_2| \leq n-2$ vertical units. Each vertical unit segment has at most 2 corners, therefore the total number of corners does not exceed $2n - 4$.

(2) There is no crossing for any pair of paths (α, β) with $x_1, x_2 \in [x_\alpha^-, x_\alpha^+] \cap [x_\beta^-, x_\beta^+]$, $y_\alpha(x_1) < y_\beta(x_1)$, and $y_\alpha(x_2) > y_\beta(x_2)$. Because the floor function is monotonic, $\lfloor a \rfloor < \lfloor b \rfloor$ necessarily leads to $a < b$, so a crossing between two paths leads to a crossing between the original straight lines, which is contradictive to the given planar graph embedding.

(3) There are no identical paths, because each path has a unique pair of end points.

With edges converted into rectilinear paths, we then perform the following two operations.

(1) If a path intersects a vertex (because of y coordinates lowered by the floor function), then the original line must be above the vertex, so we move the horizontal intersecting segment upwards $(+y)$ by ϵ. ϵ takes a small value (e.g. $1/n^2$ grid unit).

(2) If two or more paths share segment(s), then we find the diverting point (it exists because there are no identical paths), and look at the shared segment on this point. If the segment is horizontal, then move the segments upward $(+y)$ by ϵ on the set of paths that either turn upward or don't turn downward; if the segment is vertical, then move the segments leftward $(-x)$ by ϵ on the set of paths that either turn leftward or don't turn rightward. Figure 14(b→c) is an example of this operation. Note if there are two diverting points, the operations based on them are consistent because there are no crossing paths.

Repeat these two operations, and the intersections can be all resolved. Because we only need a limited number of spacings to avoid superposition of paths, and since the operations always move the segments towards $-x$ or $+y$, there is no dead loop in the repetition of operations.

In the end, we give each vertex a diameter to reach all its path segments that are moved upwards or leftwards; the result is a rectilinear path

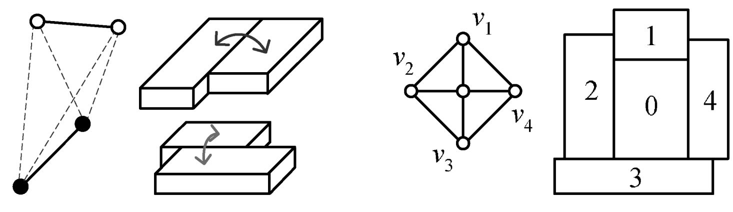


Fig. 15. Basic gadgets in 2.5-D cuboidal dual.

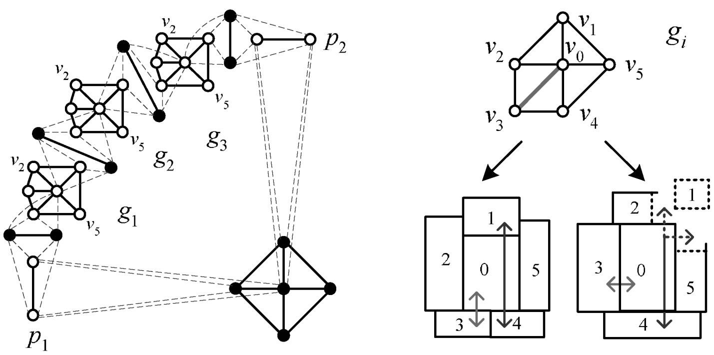


Fig. 16. 2-layer subgraph of a clause gadget.

embedding of G_p, with each path having no more than $2n - 4$ corners and no path intersections. □

Based on Lemma 4, the reduction from Planar-3SAT to 2.5-D cuboidal dual can be constructed. We start with two basic gadgets shown in Figure 15. In figure 15(a), 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 of the same reason as in Figure 3, if the two pairs have the same direction, a complete connection is impossible. In Figure 15(b), the diamond gadget is similar to the 7-vertex gadget in lemma 1 and figure 2, so that the rectangles of v_1 and v_3 must be on opposite sides of the central rectangle.

For a clause c_i in Planar-3SAT, we can construct a 2 layer gadget as in Figure 16(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 1, are enforced to have orthogonal contact surfaces by the diamond gadget on layer 2. Meanwhile the two pairs are also connected through three 6-vertex gadgets, which have a different property from the diamond gadget: the contact directions of $v_0 \rightarrow v_1$, $v_0 \rightarrow v_2$, $v_0 \rightarrow v_3$, and $v_0 \rightarrow v_4$ have more freedom than in the diamond gadget, but are still dependent on each other. For instance, assume the direction of $v_0 \rightarrow v_4$ of such a gadget is determined like in Figure 16(b), there are two cases.

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

(2) 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 either of the two sides of v_0.

LEMMA 5. *The clause gadget in Figure 16(a) has a 2-layer cuboidal dual if and only if at least one 6-vertex gadget has $v_0 \rightarrow v_3$ horizontal.*

PROOF. Starting from the vertical pair p_1, which is connected to the horizontal pair p_2 through three 6-vertex gadgets, 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 enforced by vertex pair p_2.

If all the 6-vertex gadgets here have $v_0 \rightarrow v_3$ vertical: starting at g_1, its $v_0 \rightarrow v_3$ and $v_0 \rightarrow v_4$ are both vertical, so v_3 and v_4 must be at the same side of v_0, which makes g_1 work as a diamond gadget, and $v_0 \rightarrow v_1$ is also vertical. The same situation propagates through g_2 and g_3, and finally vertex pair p_2 is also vertical, which leads to contradiction. Thus, 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, that is, $v_0 \rightarrow v_1$ horizontal. By this propagation, $v_0 \rightarrow v_1$ of g_3 is horizontal and the 2.5-D cuboidal dual of Figure 16(a) can be constructed. □

With Lemma 5, the reduction from Planar 3-SAT becomes straightforward, since in 3-SAT a clause is true if and only if at least one of its literals is true.

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

PROOF. We construct a 3-layer graph $G = (V, E, L : V \mapsto \{1, 2, 3\})$ from $G_{p3SAT} = (U \cup C, E')$. G's overall topology is shown in Figure 17. A pair of vertices on layer 2 defines the vertical direction, orthogonally connected with m pairs of vertices (like Figure 15(a)) on layer 3, which are thus in the horizontal direction. Each vertex pair on layer 3 is orthogonally connected with the lower pair of vertices in a clause gadget c_i (so that p_1 is vertical and p_2 is horizontal).

We create n diamond gadgets on layer 2 corresponding to variable $u_1, \cdots, u_n$. For each u_i, we put m pairs of vertices $p_{i1}, \cdots, p_{im}$ over u_i's diamond gadget on layer 1, which are all enforced in the same direction (as in Figure 18(a)). Then for u_i's each appearance in clause c_j, or an edge in G_{p3SAT} between u_i and c_j, p_{ij} is connected to a path $R_{i,j}$ consisting of a series of layer 2 diamond gadgets interleaved with layer 1 vertex pairs, and the end of $R_{i,j}$ is connected to a 6-vertex gadget g_k in clause gadget c_j. $k \in \{1, 2, 3\}$ is determined by the rectilinear path embedding of G_{p3SAT} by Lemma 4, such that the paths connected to g_1, g_2, g_3 are in clockwise order. For example, in the instance of Figure 19, clause gadget c_1 can have g_1, g_2, g_3 connected to the paths from u_1, u_3, u_2 respectively.

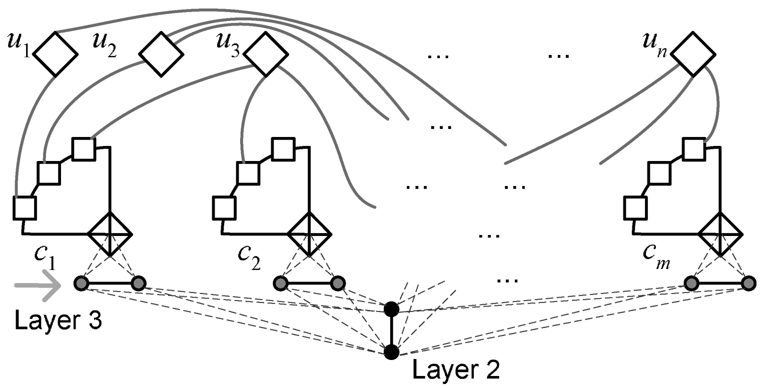


Fig. 17. Construction of a 3-layer graph G from a Planar 3-SAT formula.

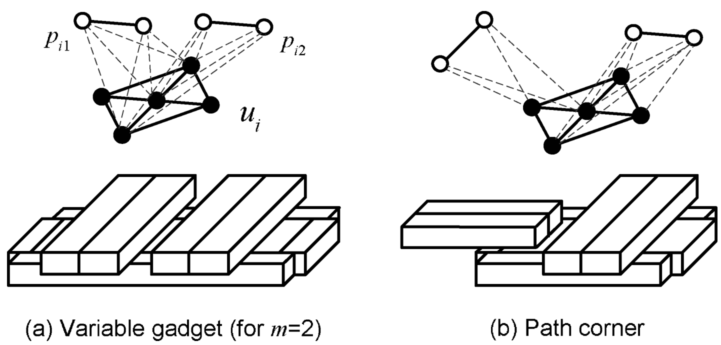


Fig. 18. Variable gadget and path corner gadget.

$R_{i,j}$ starts from vertex pair p_{ij} and ends at vertex pair (v_0, v_3) of one of c_j's 6-vertex gadget (g_k). Along $R_{i,j}$, the number of layer 2 diamond gadgets in $R_{i,j}$ is:

—$2(m+n)$, if u_i appears in c_j in non-negated form (as "u_i");
—$2(m+n)+1$, if u_i appears in c_j in negated form (as "$\overline{u_i}$").

A layer-2 diamond gadget connects two layer-1 vertex pairs as in Figure 18(b). This connection both enables and enforces the two vertex pairs to have orthogonal directions, so that a corner of a rectilinear path can be realized, regardless whether it is a left turn or right turn. Also,

—for u_i, vertex pair (v_0, v_3) in g_k is in the same direction as p_{i1};
—for $\overline{u_i}$, vertex pair (v_0, v_3) in g_k is in the orthogonal direction of p_{i1}.

Now we prove the planar 3-SAT formula is satisfiable if and only if G has a cuboidal dual. The main part is from left to right: if there is a set of values assigned to $u_1, \cdots, u_n$ to make the formula true, then there is a 3-layer cuboidal dual of G, which can be constructed from the set of u_i values.

The n variable gadgets and m clause gadgets are placed on the grid points of G_{p3SAT}'s rectilinear path embedding in Lemma 4. For each variable u_i, its m

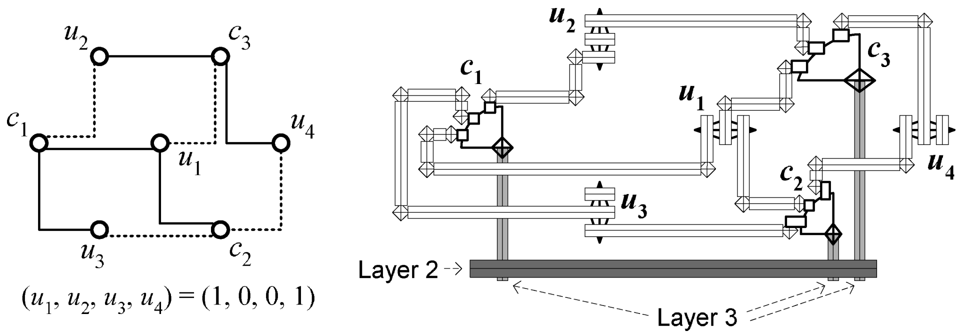


Fig. 19. 3-layer cuboidal dual from 3-SAT formula $(u_1 \vee \overline{u_2} \vee u_3) \wedge (u_1 \vee \overline{u_3} \vee \overline{u_4}) \wedge (\overline{u_1} \vee u_2 \vee u_4)$.

pairs of vertices $p_1, \cdots, p_m$ are placed horizontal if $u_i = 1$, or vertical if $u_i = 0$. So by each path $R_{i,j}$ connected to (v_0, v_3) in g_k of clause gadget c_j, $v_0 \rightarrow v_3$ is horizontal if and only if u_i or its negation evaluated to 1 in clause c_j. The clause gadgets can be constructed by Lemma 5, because assuming the formula is evaluated to 1, each clause has at least one literal evaluated to 1, and the corresponding g_k has horizontal $v_0 \rightarrow v_3$.

To make a complete 3-layer cuboidal dual, we add the layer-3 alignment gadgets and all the $R_{i,j}$ paths. As in the example of Figure 19, the vertical layer-2 cuboid pair can be placed at bottom ($-y$ side), from which m horizontal layer-3 cuboid pairs connect to the m clause gadgets. By the shape of the clause gadget, the layer-3 pairs are placed at the $+x$ side of the grid points; and if multiple clause gadgets exist on the same vertical line, we can make the upper gadgets wider so that the layer-3 pairs have no conflict. The layer-1 cuboid pairs of each path $R_{i,j}$ go along the path in the rectilinear path embedding, which needs no more than $2(m+n) - 4$ corners, by Lemma 4. Consider that g_1, g_2, g_3 are located at the left or upper side of each clause gadget c_j, this path may need 2 more corners to go around c_j. $R_{i,j}$ has at least $2(m+n)$ corner gadgets (Figure 18(b)), which is enough to make the path, and redundant corners can be placed at any corner. (For convenience, in Figure 19 we draw an even number of corners for u_i and odd number of corners for $\overline{u_i}$.) Each corner gadget has a diamond gadget on layer 2, which is not contacting any layer-3 vertex. This is true at the two ends of each path, since the layer-3 vertex pairs are at the $+x$ side of grid points, and the paths are on the grid or at the $-x$ side of grid points (by operation in Lemma 4). And if a path segment in the middle overlaps with a layer-3 vertex pair, we move the path segment towards $-x$ and resolve the overlapping.

By this construction, the 3-layer cuboidal dual is guaranteed to exist if provided an assignment of $u_1, \cdots, u_n$ satisfying the planar 3-SAT Boolean formula. On the other hand, if there is a 3-layer cuboidal dual of graph G, then the 3-SAT formula is satisfiable, because this by Lemma 5 the orientations of the n variable gadgets give a solution set $u_1, \cdots, u_n$, which makes every clause evaluated to 1.

In conclusion, by constructing layered graph G from the 3-SAT instance, we reduce Planar 3-SAT to a 2.5-D cuboidal dual problem with 3 layers. This is a polynomial reduction, because the number of vertices and edges in G is on $O(n^2)$ scale of G_{p3SAT}'s size or the 3-SAT formula's length. □

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

The hardness of finding a multiple-layer cuboidal dual is also fundamentally greater than 2-D (single-layer) cases, except that the exact hardness of the 2-layer case is still unknown. Considering that the gadgets we introduced are all in 2 layers, and the proof of NP-hardness only uses $2m$ vertices in the third layer, the 2-layer case of this problem may also be hard.

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 of migrating from 2-D to 2.5-D, to 3-D is generally increasing.

A surprising finding among these results is that just a few layers of the 2-D cases, which can be decided by a simple rule (Theorem 2, or Kozminski and Kinnen [1984] and Kant and He [1997]), being stacked together, can make the problem so much more complex that there exists no effective algorithm to 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 may 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. More research in this cuboidal dual problem or other problems with graph-geometry formulations could be helpful for us to understand the nature of 3-D physical design problems.

REFERENCES

BHASKER, J. AND SAHNI, S. 1986. A linear algorithm to find a rectangular dual of a planar triangulated graph. *Proceedings of the ACM/IEEE Design Automation Conference*, 108–114.

CHIBA, N., NISHIZEKI, T., ABE, A., AND OZAWA, T. 1985. A linear algorithm for embedding planar graphs using pq-trees. *J. Comput. Syst. Sci. 30*, 54–76.

COOK, S. A. 1971. The complexity of theorem-proving procedures. *Proceedings of the 3rd ACM Symposium on the Theory of Computing*. 151–158.

DENG, Y. AND MALY, W. 2003. Physical design of the 2.5D stacked system. *Proceedings of the IEEE International Conference on Computer Design*. 211–217.

GOPLEN, B. AND SAPATNEKAR, S. 2007. Placement of 3D ICs with thermal and interlayer via considerations. *Proceedings of the ACM/IEEE Design Automation Confrence*. 626–631.

HO, R., MAI, K. W., AND HOROWITZ, M. A. 2001. The future of wires. *Proc. IEEE 89*, 490–504.

KANT, G. AND HE, X. 1997. Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems. *Theoret. Comput. Sci. 172*, 175–193.

KARP, R. M. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations*. New York: Plenum. 85–103.

KOZMINSKI, K. AND KINNEN, E. 1984. An algorithm for finding a rectangular dual of a planar graph for use in area planning for VLSI integrated circuits. *Proceedings of the ACM/IEEE Design Automation Conference*. 655–656.

LICHTENSTEIN, D. 1984. Planar formulae and their uses. *SIAM J. Comput. 11*, 2, 329–343.

LIM, S. K. 2005. Physical design for 3D system-on-package: Challenges and opportunities. In *Proceedings of the Design, Automation and Test in Europe*. 532–536.

SCHNYDER, W. 1990. Embedding planar graphs on the grid. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*. 138–148.

WU, G.-M., LIN, J.-M., AND CHANG, Y.-W. 2001. An algorithm for dynamically reconfigurable FPGA placement. *Proceedings of the IEEE International Conference on Computer Design*. 501–504.

YAMAZAKI, H., SAKANUSHI, K., NAKATAKE, S., AND KAJITANI, Y. 2000. The 3D-packing by meta data structure and packing heuristics. *IEICE Trans. Fundamentals*, 639–645.

YU, H., HO, J., AND HE, L. 2006. Simultaneous power and thermal integrity driven via stapling in 3D ICs. *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. 802–808.

Received July 2009; revised January 2010, April 2010; accepted May 2010