International Journal of Computational Geometry & Applications

TERRAIN DECOMPOSITION AND LAYERED MANUFACTURING*

SÁNDOR P. FEKETE†
Abteilung für Mathematische Optimierung
TU Braunschweig, D–38106 Braunschweig, GERMANY

JOSEPH S. B. MITCHELL‡
Department of Applied Mathematics and Statistics
University of Stony Brook, Stony Brook, NY 11794–3600, USA

Received (01/29/99)
Revised (08/21/01)
Communicated by Godfried Toussaint

ABSTRACT

We consider a problem that arises in generating three-dimensional models by methods of *layered manufacturing*: How does one decompose a given model P into a small number of sub-models each of which is a *terrain polyhedron*? Terrain polyhedra have a base facet such that, for each point of the polyhedron, the line segment joining the point to its orthogonal projection on the base facet lies within the polyhedron. Terrain polyhedra are exactly the class of polyhedral models for which it is possible to construct the model using layered manufacturing (with layers parallel to the base facet), without the need for constructing "supports" (which must later be removed). In order to maximize the integrity of a prototype, one wants to minimize the number of individual sub-models that are manufactured and then glued together.

We show that it is NP-hard to decide if a three-dimensional model P of genus 0 can be decomposed into k terrain polyhedra. We also prove a two-dimensional version of this theorem, for the case in which P is a polygonal region with holes. Both results still hold if we are restricted to isothetic objects and/or axis-parallel layering directions.

1. Introduction

Advanced manufacturing has been identified as one of the critical technologies for economic competitiveness. The goal is to make manufacturing systems more flexible and more automated, from the design stage, to the prototyping, process planning, and quality control stages. Flexibility and automation require sophisticated software tools and systems, often utilizing computationally intensive algorithms. Geometry

*This work was initiated during discussions at the 1993 Barbados Workshop on Geometric and Computational Aspects of Injection Molding, organized by Godfried Toussaint, McGill University.

†`sandor.fekete@tu-bs.de`

‡`jsbm@ams.sunysb.edu`; Partially supported by grants from Boeing Computer Services, Bridgeport Machines, HRL Laboratories, Northrop Grumman, Sun Microsystems, and the National Science Foundation (CCR-9504192, CCR-9732220).

arises in many, if not most, of the computational problems of manufacturing, due to the inherently geometrical nature of form design, of manufacturing processes, and of the physical products themselves.

In the manufacturing process planning problem, the goal is to solve an "inverse geometry" problem: Given a geometric model of a desired part, a specification of the geometry of the input raw material (e.g., a block of metal), and a machine with certain constrained actions for removing or adding material or changing its shape, determine a good method of manufacturing the part. Examples include molding, casting, milling, bending, cutting, spray deposition, etching, etc. For each type of process, the goal is to study the algorithmic difficulty of deciding whether or not a given shape can be manufactured using that process, and, if so, how to go about the process in the most efficient manner. Ultimately, we want CAD systems to be able to verify on-line that a part can be manufactured with a given technique, and to propose the best way to do it.

A motivating application for our work are methods of "layered" rapid prototyping. One example is that of stereolithography, in which a part can be fabricated, layer by layer, by selectively solidifying portions of the surface of a liquid in a vat, as a horizontal platform holding the part is lowered. In this process, problems may arise by "overhanging" material, i.e., if the rest of the surface is not a terrain with respect to a suitable basis, and material on a new layer is not supported by material on the previous level. Given a geometric description of a part, Asberg et al.[7] have given a simple linear-time algorithm to determine if the part can be manufactured, in a somewhat simplified model of the stereolithography process. A problem suggested by their work is: How does one handle parts that *cannot* be manufactured in a single execution of the process? One approach is to use "supports", i.e., add portions of extra material below overhanging material; after manufacturing the whole piece, these supports are removed in an additional production step. Minimization of these supports can be studied for the total volume of support material (corresponding to minimizing the total time for processing liquid plastic material) or the total contact area between supports and manufactured object (corresponding to the time spent on post-processing). In this paper we study another natural approach: decompose a part into a (hopefully) small number of connected pieces, each of which *can* be manufactured, and then "glue" these pieces together. The goal is to minimize the number of such pieces. The only work along these lines is the recent paper of Ilinkin et al.,[19] in which the authors examine decomposition into *two* such pieces by a single plane, for a given normal direction direction d, such that support volume or contact area are minimized.

In this paper, we prove that the problem of minimizing the number of manufacturable pieces that do not require any supports at all is hard: It is NP-complete to determine if a model can be decomposed into k connected pieces, each of which is a "terrain polyhedron", and thus manufacturable using layered prototyping. Specifically, we prove hardness for two-dimensional polygonal domains (with holes) and for three-dimensional polyhedral domains, even of genus 0. These results are in contrast to the work by Liu and Ntafos,[22] who show that partitioning a simple polygon into the minimum number of uniformly monotonic pieces can be done in time $O(n^3)$.

Related Work

There has been a variety of recent work in the area of algorithmic study of manufacturing processes; see the excellent surveys by Bose and Toussaint[10] and by Janardan and Woo.[20] In particular, there has been algorithmic study of the problems of injection molding,[11,28] NC-machining of pockets[17,6] gravity casting,[3,9] and stereolithography.[7]

Rosenbloom and Rappaport[28] give an $O(n)$-time algorithm to decide if a simple polygon having n vertices can have its boundary partitioned into two monotone chains (which is necessary and sufficient for castability in two dimensions). In three dimensions, Bose et al.[9] consider the sand casting model, in which a cast must be cut in two by a plane so that each piece can be moved away from the part without collisions; they give an algorithm requiring $O(n^{3/2+\epsilon})$ time and space or $O(n^2 \log n)$ time and $O(n)$ space to determine if a given polyhedral model is castable, and, if so, to compute a plane that cuts the cast. The assumption that the cast must be cut in two by a plane is removed in the work of Ahn et al.,[3] who consider the problem of determining if there is a direction d, and a way to cut the cast in two along some closed curve, so that the two pieces of the cast can be removed without collisions in directions $\pm d$.

The first algorithmic analysis of the stereolithography process was done by Asberg et al.,[7] who give $O(n)$-time algorithms to determine if a polyhedral object having n vertices can be manufactured using stereolithography, without the need for external supports, but possibly allowing a slight overhang during the process. The use of external support structures during fabrication has been considered by Allen and Dutta.[4] In an attempt to reduce the need for such support structures, the same authors (Ref. [5]) have suggested a method of selectively thickening surface walls, to reduce the use of support structures. Minimizing the total volume of supports or minimizing contact area was studied by Agarwal and Desikan,[2] Mahji et al.,[23] and Schwerdt et al.[29] Ilinkin et al.[19] have studied the problem of optimally decomposing the original piece into two pieces, such that support volume or contact area are minimized.

For general background on polygon decomposition problems, see the book by O'Rourke[25]; in particular, he describes a proof that it is NP-hard to decompose a polygon with holes into a minimum number of star-shaped pieces. See also the more recent survey by Keil.[21]

A somewhat different kind of decomposition problem was studied by Pach and Tardos,[27] who show that a family of pairwise disjoint convex planar sets does not necessarily allow separating a constant fraction from the others by using a series of straight-line cuts.

For further information on the role of computational geometry in manufacturing, we can point to a variety of other resources, including a workshop report by C. Yap,[33] the task force report of Chazelle et al.,[12] a special issue of *Algorithmica*[24] edited by the second author, and a special issue of the journal *Computer-Aided Design* edited by T. Woo and R. Janardan.[31]

A "feature-based" approach to manufacturability analysis is given by Gupta and Nau.[16] For more information on manufacturing technology issues in computer science, we refer the reader to Ref. [1,30,32].

Preliminaries

A *simple polygon*, P, having n vertices, is a closed, simply-connected region whose boundary is a union of n (straight) line segments ("edges"), whose endpoints are the vertices of P. A *polygonal domain*, P, having n vertices and h holes, is a closed, multiply-connected region whose boundary is a union of n line segments, forming $h+1$ closed (polygonal) cycles. (A simple polygon is a polygonal domain with $h = 0$.) A polygonal domain P is said to be *rectilinear* (or *isothetic*) if every edge of P is parallel to the x- or y-axis. A simple polygon P is said to be a *terrain polygon* if it has an edge e, called a *base*, such that every point of P can be connected to e by a line segment within P that is orthogonal to e.

A *polyhedral domain* is a connected subset, P, of $\Re^3$ whose boundary consists of a union of a finite number of triangles. A *simple polyhedron* P is a polyhedral domain whose boundary is a connected set homeomorphic to a sphere (it has "genus" 0 – there are no "holes" or "handles"). We say that P is *isothetic* if every triangle on its boundary is orthogonal to one of the three coordinate axes. A *terrain polyhedron* P is a simple polyhedron that has a facet, f, called a *base*, such that the surface of P consists of f and a terrain in one direction d orthogonal to f; in other words, every point of P can be connected to f by a line segment within P that is orthogonal to f. It is straightforward to see that this is equivalent to P being manufacturable using layered manufacturing (with layers parallel to the base) in a single piece, without supports.

Objectives

Throughout this paper, we will consider partitioning a given (two- or three-dimensional) polyhedral domain P into k connected pieces $P_1, \ldots, P_k$ that can be produced by layered manufacturing without using any supports. We call this *decomposing* P into terrain polyhedra, and the question whether, for input model P and integer k, a decomposition of P into k terrain polyhedra exists is called the TERRAIN DECOMPOSITION PROBLEM (TDP). In the context of our following discussion, we admit *any* partition; in practical manufacturing, most interesting decompositions will arise from a sequence of planar cuts through P. The interested reader will easily verify that restricting decompositions in such a way does not change our results; moreover, our results remain valid even if we restrict the normal directions of terrain polyhedra to be axis-parallel.

In the following Section 2, we consider the two-dimensional case of the TDP on polygons. Clearly, the interest in this case lies more in mathematics than in practical manufacturing; however, practitioners should take note that our main result for the "real" three-dimensional case (described in Section 3) is built upon understanding the two-dimensional situation.

2. Decomposing Polygonal Domains

In this section, we prove our main result in two dimensions:

Theorem 1 *The TDP of deciding if a polygonal domain P can be decomposed into k terrain polygons is NP-hard, even if the domain is rectilinear.*

Proof. We give a reduction from the problem PLANAR 3SAT, which has been shown to be NP-complete by Lichtenstein.[15] For another application of PLANAR 3SAT in showing NP-hardness of a geometric problem, see Fowler et al.[14] Also, see O'Rourke and Supowit[25,26] for a proof that it is NP-hard to decide whether a given polygon can be decomposed into a small number of *star-shaped* pieces. (That proof is quite instructive in that, similar to our own proof, it also uses "variable loops" and "dots" in order to establish correctness of the given construction. However, the main difficulty in our construction lies in coming up with clause gadgets that work for the TDP, while still preserving the main properties of variable gadgets.)

A 3SAT instance I with n variables and m clauses is considered to be *planar* if and only if the following bipartite graph G_I is a planar graph: The nodes of G_I are partitioned into a set corresponding to variables and a set corresponding to clauses; the node for variable x_i is adjacent to the node for clause c_j if and only if variable x_i appears in clause c_j. Refer to Figure 1.

Now the transformation of G_I into a polygon is done in a four-step process:

1. G_I is drawn in the plane in a suitable manner.
2. The set of edges incident on a variable vertex (the "star") is turned into a closed rectilinear *loop*, i.e., a simple closed rectilinear path.
3. Each loop gets turned into a polygonal piece in the shape of a narrow rectilinear annulus, i.e., a polygonal domain with exactly one hole.
4. A polygonal piece is added for each clause, such that all annuli are linked up into one connected domain. This is done such that decomposition into a small number of pieces and satisfying the instance I correspond appropriately.

For another application of a similar multiple-step construction in a hardness proof, see the paper by Baur and Fekete,[8] where this technique is used in the context of geometric packing and dispersion.

For an overview of the construction, refer to Figure 2 for a polygon resulting from the 3SAT instance shown in Figure 1.

Without loss of generality, we may assume that G_I is connected. We assume that G_I has been embedded in a rectilinear manner, where clause vertices (all with degree three) and variable vertices (possibly with degree higher than four) are represented by squares of suitable size; edges are represented by rectilinear paths with a limited number of bends. A class of algorithms providing such a layout is the *Kandinsky* approach by Fößmeier and Kaufmann,[13] which uses network flow techniques to

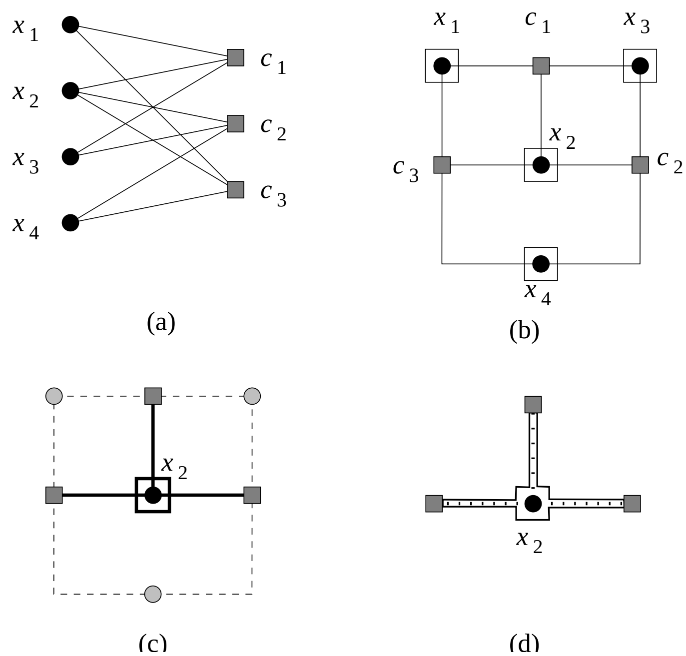

Figure 1: (a) The bipartite graph G_I corresponding to the 3SAT instance $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee x_4) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_4})$. (b) A planar rectilinear embedding of G_I. (c) The "star" of edges adjacent to a variable node, x_2. (d) A "loop" constructed from the star of edges.

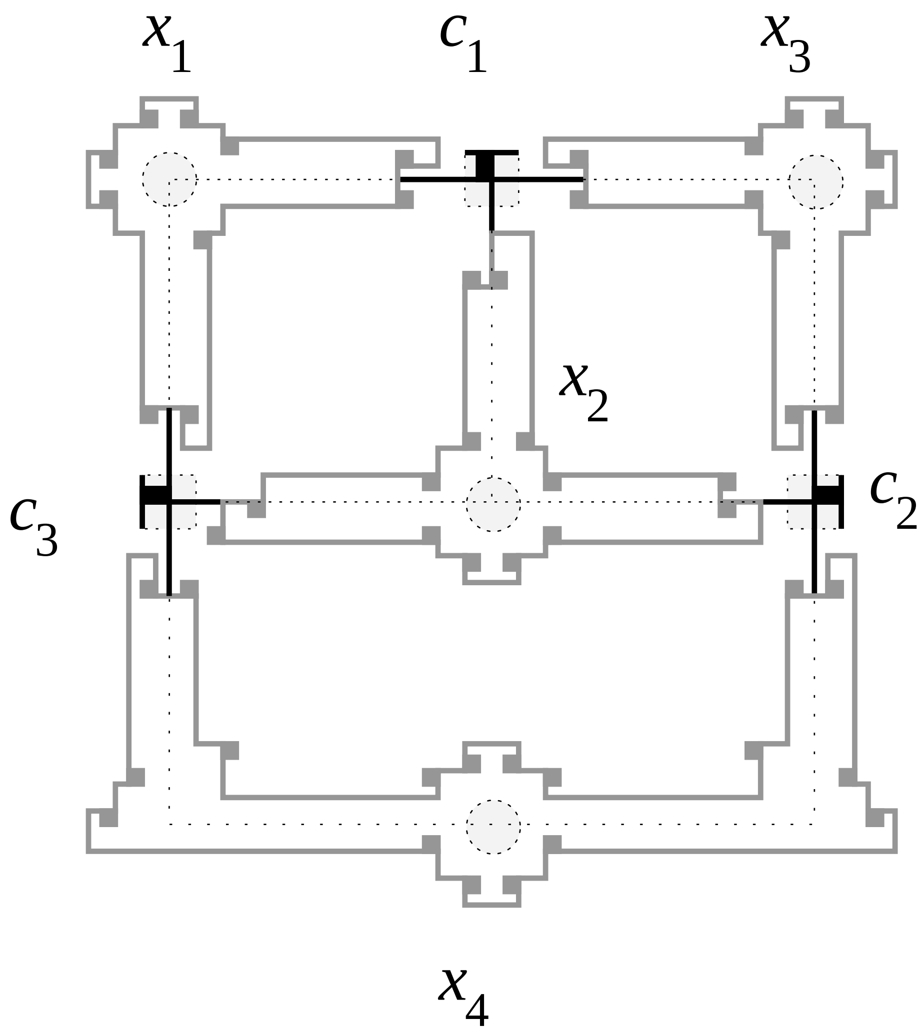

Figure 2: A polygonal domain P representing the 3SAT instance from Figure 1. P consists of four narrow "variable" annuli (the areas shaded in dark gray) and three clause shapes (the areas shaded in black). The underlying rectilinear drawing of the graph G_I from Figure 1(b) is shown dotted and in light gray.

achieve a low number of bends. In a planar embedding of G_I, the set of edges adjacent to the square for one variable node forms a star, as shown in Figure 1(c); thus a variable can be represented by a *loop*, which is obtained by traversing a slightly offset version of the boundary of the star, as shown in Figure 1(d). Given the nature of our embeddings, a loop can be assumed to be axis-parallel and to have a short description—i.e., the number of bends in the path and their coordinates will be integers of size bounded polynomially in the length of the 3SAT instance.

In the following, we describe the third and fourth steps: representing a loop by a polygonal region that can be partitioned into few terrains only in two basically different ways. These two partitions correspond to the possible truth settings of the corresponding variable. Furthermore, we show how the clauses can be represented by polygonal pieces that connect the loops for the involved variables, such that the additional piece for a clause can be decomposed efficiently if and only if the clause is satisfied by at least one of its literals. More precisely, the region for any clause can be decomposed into one additional terrain polygon and pieces that can be added to terrains from the three loops, if and only if the decomposition of variable annuli corresponds to a satisfying truth assignment. Otherwise, two additional pieces will be necessary.

Representing Variables

Refer to Figure 3. Consider a loop L_i representing a variable x_i, consisting of n_i line segments e_j. A set of consecutive segments may either have alternating left- and right-hand turns, resulting in a "staircase" portion of L_i, or have two consecutive turns that are both right- or left-hand, in which case we have the situation of a "U-turn" formed by a triple of consecutive segments. A segment is called *ordinary* if it is not part of a triple forming a U-turn, and a *U-segment* otherwise. For reasons that will become clear in the proof of Lemma 2, we assume that U-turns are well separated, i.e., no U-segment is part of more than one triple forming a U-turn, and there are at least two ordinary segments between different U-segment triples. (If necessary, this can be achieved by adding a small number of ordinary segments between U-turns; for an example, see Figure 2 near the shaded circles.)

Now this loop L_i is turned into a rectilinear annulus, R_i, by extending ("fattening") each horizontal/vertical edge e_j of the loop by a small amount $\varepsilon = 1/n$ in both the horizontal and the vertical direction, resulting in a rectangular region r_j of width ε. (Note that regions r_j and r_{j+1} arising from adjacent edges e_j and e_{j+1} will overlap.) In Figure 3(a') we show the result of fattening a portion of a loop for a "staircase" portion of R_i. For U-turns in the loop, in addition to the fattening, we add small rectilinear extensions called "tabs", as shown in Figure 3(b'). (For easier notation, we still refer to a region extended in this manner by r_j, even though it is not rectangular.) Note that further changes will be made later to accommodate for the clauses; see below.

Overall, we get a resulting rectilinear annulus polygon R_i, consisting of n_i (partially overlapping) pieces r_j, each corresponding to a line segment e_j of L_i.

For easier classification of decompositions, and for use in the correctness proof of

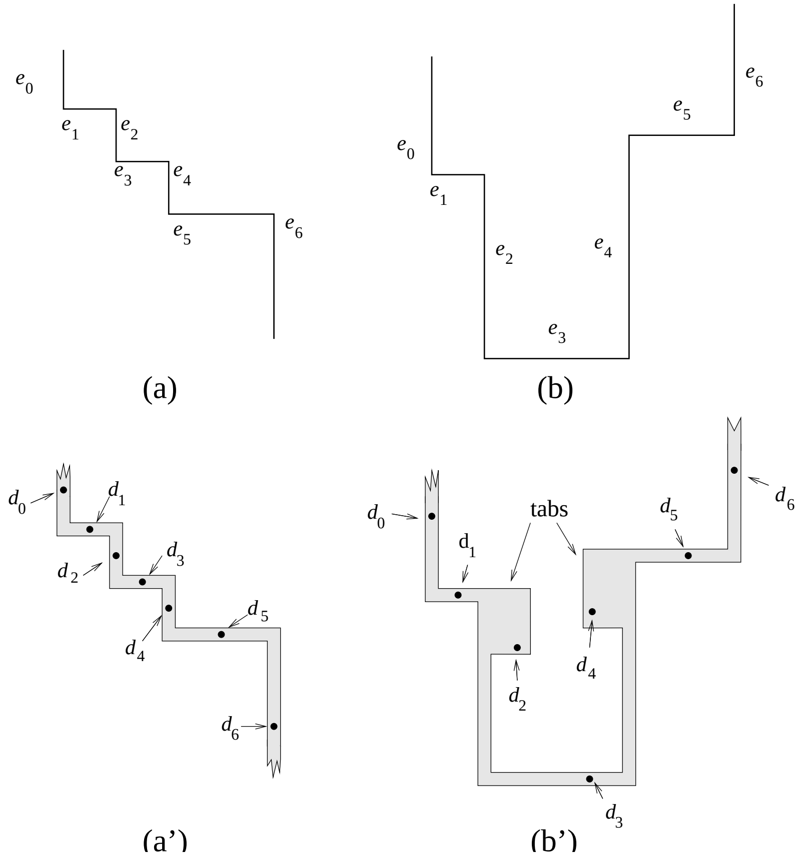

Figure 3: Pieces to build the annulus for representing a variable: (a) A "staircase" sequence of edges, e_0, e_1, ..., e_6. (a') A representing set of rectangles $r_0, r_1, \dots, r_6$, and their dots. (b) A "U-turn" sequence of edges, e_2, e_3, e_4, and their neighbors. (b') A representing set of regions; also shown are the modifications ("tabs") added to rectangles r_2 and r_4, and the position of all dots.

our reduction, we select a set d_1, ..., d_{2n_i-1}, of n_i special points (*"dots"*), one near the middle of each r_j, as shown in Figure 3. If an edge of L_i gets represented by a region with a tab, we make sure the dot gets chosen from that tab. Furthermore, dots from different r_j's cannot see each other.

In the following, we consider what subsets of dots can be covered by the same terrain polygons. We say that a terrain polygon P_ℓ *matches* the dots d_j and $d_{j'}$, if it contains both of them.

How the Variable Gadgets Work

The intuitive idea is that the annulus R_i representing L_i can only be decomposed into $n_i/2$ terrain polygons by using one of two possible matchings of adjacent dot pairs, so that all of the corresponding pairs of polygonal pieces r_j form terrain polygons; each of these two choices corresponds to fixing variable x_i to one of two possible truth values.

At various places, we will make use of the following straightforward observation. Here, the *link distance* of two points is the minimum number of edges in a path that is necessary to connect them. We say that a line segment e *U-links* two points q_1 and q_2, if e sees both of them and the line through e does not separate q_1 and q_2 (i.e., if q_1, e, and q_2 form a U-turn).

Lemma 1 *Let two points q_1 and q_2 be in the same terrain polygon P_ℓ. Then the link distance between them is at most three. If the link distance is three, then there must be a line segment in P_ℓ that U-links them.*

Now we state the main claim concerning variable gadgets more formally:

Lemma 2 *Consider an annulus R_i as described above that represents a variable. Then R_i can be decomposed into $n_i/2$ terrain polygons, if and only if there is a decomposition in which each terrain polygon contains two adjacent dots d_j and d_{j+1}.*

Proof. It is easy to check that the "if" part holds: By construction, for any terrain polygon that matches d_j and d_{j+1}, there is a terrain polygon that contains all of r_j and r_{j+1}.

To see the converse, assume that there is a terrain polygon P_ℓ that contains at least three dots. We will argue that no such polygon can exist.

First, it follows from Lemma 1 that no two dots from pieces r_j and $r_{j+\ell}$ with $3 \leq \ell \leq 2n_i - 3$ can be contained in the same terrain polygon, since the link distance between them is greater than three.

Moreover, two dots d_j and d_{j+2} are at link distance at least three; they can only be contained in the same terrain polygon P_ℓ if there is a line segment that U-links them. This can only be the case if e_j, e_{j+1}, e_{j+2} form a U-turn; this is the situation illustrated by r_2, r_3, r_4 in Figure 3(b). The purpose of the tabs is to guarantee that in this case, the claim still holds: By construction, d_j and d_{j+2} have link distance greater than three.

Therefore, any terrain polygon P_ℓ contains at most two dots and can contain two dots only if they are from overlapping r_j's, as claimed. □

This implies that we can decompose R_i into $\frac{n_i}{2}$ terrains in two combinatorially different ways:

(1) In one decomposition (the "odd" one), dots d_0 and d_1 are matched, d_2 and d_3 are matched, etc.

(2) In the other decomposition (the "even" one), dots d_{2n_i-1} and d_0 are matched, d_1 and d_2 are matched, etc.

One of these two decompositions (say, the odd one) will correspond to setting the variable x_i "false", the other to setting x_i "true".

In the following, we will refer to the pieces in one of these two decompositions as *L-shapes*, since they are basically formed by two orthogonal rectangles. (This does not rule out the various different possibilities at the ends of these L-shapes, and that there will be some tabs; however, this does not affect the way in which the terrain polygons cover the dots, which turns out to be all that matters in our reduction.)

Representing Clauses

Next we describe how to connect the annulus polygons for representing variables by additional polygons representing clauses. Since we have assumed that the graph G_I of the 3SAT instance is connected, this will result in one connected polygonal domain. For the resulting overall picture, refer to Figure 2, in which variable annuli are shown in dark gray shading, while the clause gadgets are shown in solid black.

Consider a clause c with the three variables x_{i_1}, x_{i_2}, and x_{i_3}. By construction, all three annuli R_{i_1}, R_{i_2}, R_{i_3} have U-turns close to the corresponding clause vertex. After representing the variable annuli, we are in a situation as shown in Figure 4(a). As shown in Figure 4(b), we make some small modifications to the annuli. Figure 4(c) illustrates the difference between un-negated and negated variables, by showing the construction with two of the literals having switched logical parity from Figure 4(b).) In the following, we give a detailed description.

In the graph G_I, any vertex v representing a clause has degree three. In a rectilinear drawing of G_I, this implies that there must be two adjacent edges that reach v from opposite directions; we call these the "across" edges, and the loops representing them we call the "across" loops. (These are loops R_{i_1} and R_{i_3} in the figure.) The remaining loop (R_{i_2} in the figure) lies "in between" the others. Any U-turn close to a clause vertex has two corners; depending on which side the corners lie when viewed from inside of the simple closed region surrounded by the variable loop looking towards the clause, we speak of a "left" and a "right" side of a U-turn.

For all three U-turns, we change one of its sides: If a variable occurs in a clause as an un-negated variable, we change the left side; if it occurs as a negated variable, we change the right side. In all cases, one part of the change is "flipping" the respective corner in the loop so that we get two extra turns in the loop. This can be thought of as placing an L-shaped "dent" in the loop, adding two extra regions; for simplicity, we still refer to the set of these regions as r_j, with consecutive indices indicating adjacency along the loop (and, therefore, partially overlapping regions). One of the two new regions will be referred to as the "parallel" region, as its longer side lies parallel to the direction towards the clause; the other lies orthogonal to this direction, so it is called the "orthogonal" region. The three corners replacing the one changed corner of a U-turn are the "orthogonal corner" (incident only to the orthogonal one of the two new regions), the "new corner" (incident to both two new regions), and the "parallel" corner (incident only to the parallel of the two new regions). In addition to introducing the two new regions, the tab on the changed side of the U-turn gets moved to the new corner and placed towards the unchanged side of the U-turn. (This assures that the properties of Lemma 2 still hold at the modified U-turn.) Also, we add an extra tab at each orthogonal corner, placed

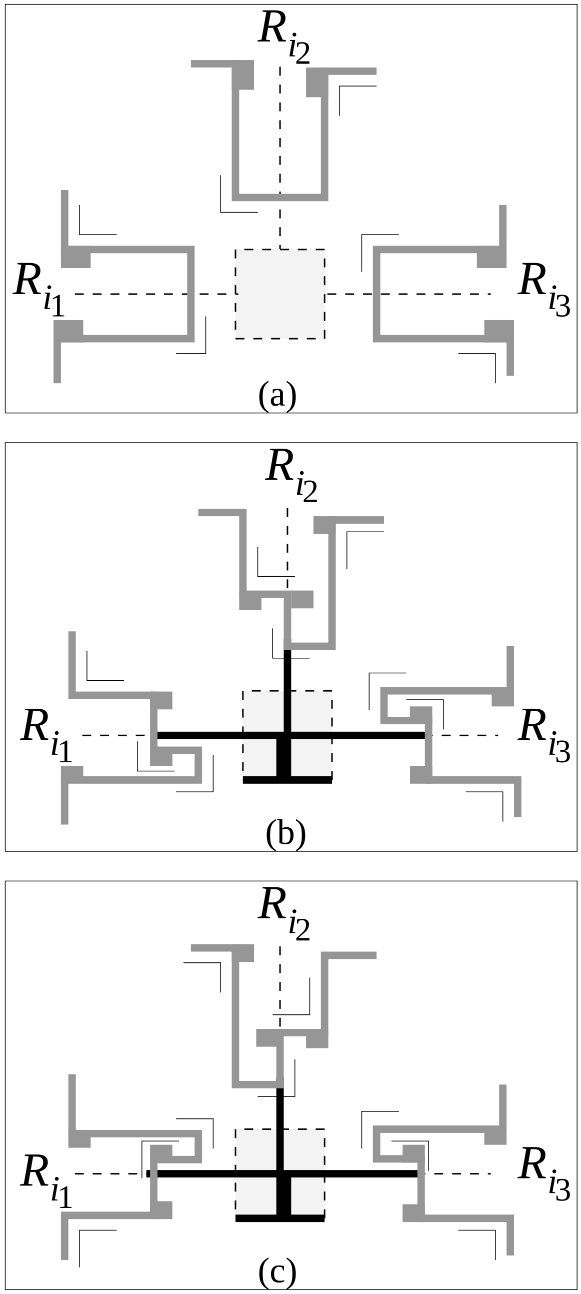

Figure 4: (a) Three different annuli, for variables x_{i_1}, x_{i_2}, x_{i_3}, with U-turns close to the vertex for clause $(x_{i_1} \vee \overline{x_{i_2}} \vee x_{i_3})$; the indicated L-shapes correspond to the truth assignments of x_{i_1}, x_{i_2}, or x_{i_3} that would satisfy c. (b) Local modifications to annuli R_{i_1}, R_{i_2}, R_{i_3} near their U-turns. (c) The same operations for clause $(\overline{x_{i_1}} \vee x_{i_2} \vee x_{i_3})$.

towards the clause vertex, as shown in Figure 4.

Finally, we add a simple rectilinear (16-sided) "clause region" that connects all three annuli, such that three of the sides of the clause region lie along the boundaries of the variable loops. (See Figure 4.) For the across loops, the connection is to the orthogonal region, while the connection for the "in between" loop is just lined up with the parallel region.

Performing these modifications for all clauses results in a polygonal region P that has a description polynomial in the size of the 3SAT instance I. In particular, the number n_i of pieces r_j of annulus R_i after this last series of modifications is polynomially bounded in the number n of variables and the number m of clauses.

How the Clauses Work

Now we claim:

Lemma 3 *There is a decomposition of P into at most $m+\sum_{i=1}^{n} n_i/2$ terrain polygons, if and only if the* PLANAR 3SAT *instance I has a satisfying truth assignment.*

Proof. The "if" part of the claim is relatively straightforward: Each annulus R_i gets decomposed into $n_i/2$ L-shapes, as suggested by Lemma 2; for each annulus, the decomposition that is chosen corresponds to the truth setting of the variable. As shown in Figure 5(a)-(c), any satisfied clause allows an L-shape of annulus R_i of the satisfying variable x_i to be enlarged by a piece of the clause region, such that the remaining part of the clause region is one terrain polygon. This implies the stated total number of terrain polygons.

It remains to establish the "only if" part of the claim. To simplify the argument and for better reference, we choose a set of two additional "clause" dots, $d_c^{(1)}$ and $d_c^{(2)}$, for each clause region, as shown in Figure 6; also, after the modification illustrated in Figure 4, we adjust the choice of "variable" dots for each annulus to preserve the property described in Lemma 2. We will show that there must be a *c-satisfying L-shape*, i.e., a terrain polygon containing two variable dots that corresponds to a truth assignment of variable x_i satisfying clause c.

We will establish this by proving the following sequence of elementary claims about decompositions of P with at most $m+\sum_{i=1}^{n} n_i/2$ terrain polygons:

(1) No terrain polygon can contain dots from two different clause regions.

(2) No terrain polygon can contain more than one clause dot.

(3) No terrain polygon can contain variable dots from two different annuli.

(4) No terrain polygon can contain more than two variable dots.

(5) No terrain polygon containing a dot $d_c^{(2)}$ can contain any other dot.

(6) Any terrain polygon contains either two variable dots, or none at all.

(7) A terrain polygon P_ℓ can contain a clause dot $d_c^{(1)}$ only if it contains a c-satisfying L-shape.

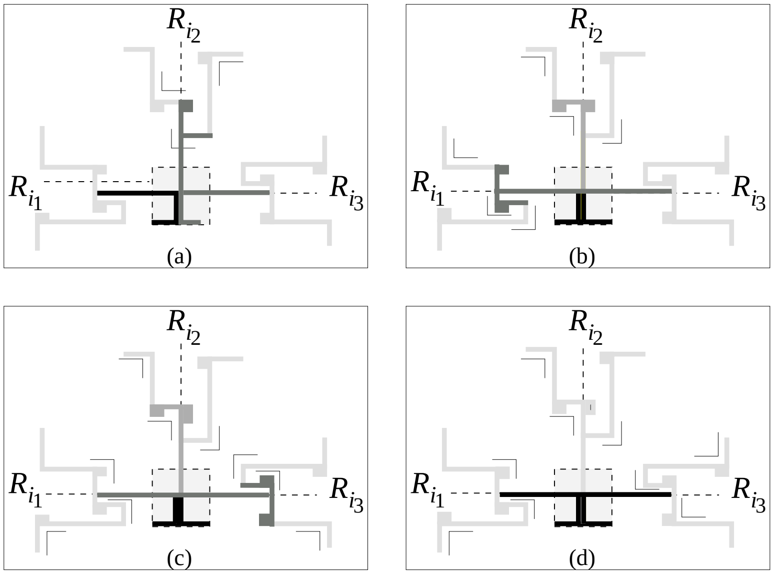

Figure 5: Decomposing a clause region with one extra terrain polygon: (a) Variable x_{i_2} satisfies clause c. (b) Variable x_{i_2} does not satisfy clause c, but variable x_{i_1} does. (c) Neither variable x_{i_2} nor x_{i_1} satisfy clause c, but variable x_{i_3} does. (d) No variable satisfies clause c.

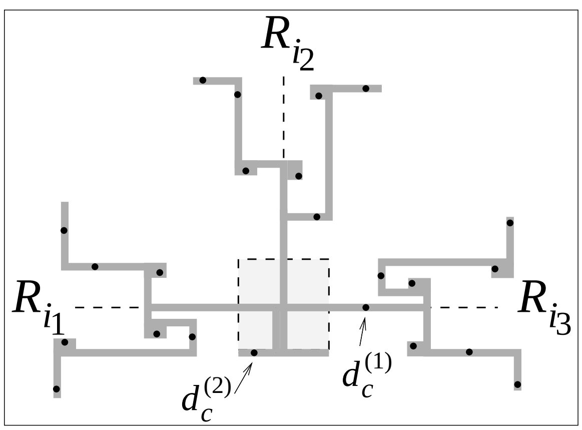

Figure 6: The dots around the clause region from Figure 4(b)

Claim 1 follows immediately from Lemma 1 and the fact that, by construction, different clause regions have link distance greater than three.

Claim 2 follows from Lemma 1: $d_c^{(1)}$ and $d_c^{(2)}$ are at link distance three, and they cannot both be U-linked by any line segment.

Claim 3: The additional tabs assure that the link distance between any two dots from two different annuli is greater than three.

Claim 4 is also implied by Lemma 2.

Claim 5: It is easy to check that there is only one (variable) dot within link distance three of any clause dot $d_c^{(2)}$. However, this dot and $d_c^{(2)}$ cannot be U-linked by any line segment.

Claim 6: By Claim 5, we need m terrain polygons just to cover all dots $d_c^{(2)}$, and none of them can contain a variable dot. This leaves only $\sum_{i=1}^{n} n_i/2$ polygons to cover the $\sum_{i=1}^{n} n_i$ variable dots. By Claim 4, no terrain polygon can contain more than two variable dots, so they must each contain two of them.

Claim 7: For any clause c, the dot $d_c^{(1)}$ cannot be contained in the same terrain polygon as $d_c^{(2)}$. It follows that $d_c^{(1)}$ must be in the same polygon as two variable dots; by Lemma 2, these must be from overlapping regions. Now it is easy to check that this is only possible for a c-satisfying L-shape.

This proves the lemma. □

Therefore, there is a decomposition into

$$m + \sum_{i=1}^{n} \frac{n_i}{2}$$

terrains, if and only if the 3SAT instance is satisfiable, completing the proof of Theorem 1. □

3. Decomposing Polyhedra in 3-Space

In this section, we consider the problem of decomposing a polyhedral object into k terrain polyhedra. In this case, the extra dimension gives us enough flexibility to modify our previous two-dimensional proof, which applied to polygonal domains with holes, to yield a hardness proof based on the construction of an isothetic simple polyhedron of genus 0.

Theorem 2 *It is NP-complete to decide if a polyhedron P can be decomposed into k terrain polyhedra, even if P is isothetic and of genus 0.*

Proof. We use the same basic construction as in the previous section. The idea is to turn the polygon representing a 3SAT instance into a 3-dimensional cylinder. This cylinder is then put onto narrow "rails" that connect it to one base plate, such that we get an isothetic polyhedron of genus 0. In order to rule out other decompositions than those corresponding to the basic L-shapes of the previous section, we also add a sufficiently large number of "ridges" to some vertical faces of the arrangement. Refer to Figure 9 for the idea, demonstrated for a U-turn section of a variable annulus.

Marked Boundary Edges

Having an extra dimension makes it necessary to rule out a larger variety of undesired shapes. For this purpose, a certain subset of the boundary edges of the polygon will be "marked" (shown thicker in Figures 7– 9). The motive behind this is to rule out those edges as base edges of terrains. The base edge of a U-turn is always marked, as shown in Figure 7; from then on, we mark every other edge. In order to get the correct parity for the next bottom edge of a U-turn, we do a parity shift halfway between two U-turns. Figure 8 shows which boundary edges will be marked in the polygonal arrangement representing a clause. Again, see Figure 2 for an overview of the resulting polygon for the instance from Figure 1.

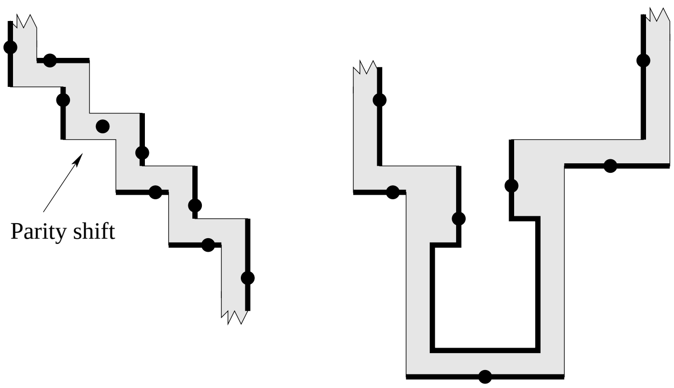

Figure 7: Staircase and U-turn pieces that are used to compose the variable loops for representing variables. Shown are marked boundary edges and the choice of variable dots.

Transforming a Marked Polygon into a Polyhedron

Once we have obtained a polygon with marked edges, we transform it into a polyhedron. See Figure 9 for a close-up at a U-turn section. The polygon forms the z-parallel projection of a cylinder. The cylinder is put onto narrow "rails", which are put onto a large z-orthogonal "base plate", thus creating an isothetic polyhedron of genus 0. Finally, the vertical faces corresponding to marked edges are modified by adding $\Theta(n^3)$ x- or y-parallel "ridges" into them; the z-parallel height of a ridge is chosen to be $\Theta(\varepsilon^3)$, while the y-or x-parallel thickness is $\Theta(\varepsilon^2)$. This transforms the "rectangles" r_j that represented the edges e_j in the loops of the previous section into "walls" w_j. Each wall has two sides; depending on whether the edges were marked or not, we get "smooth" sides (without ridges) and "rough" sides (with ridges). There are walls with a smooth and a rough side, two smooth sides (at parity shifts), and walls with two rough sides (at clauses and U-turns).

A Satisfying Truth Assignment Induces a Small Decomposition

As in the previous section, it is straightforward to see that if there is a satisfying

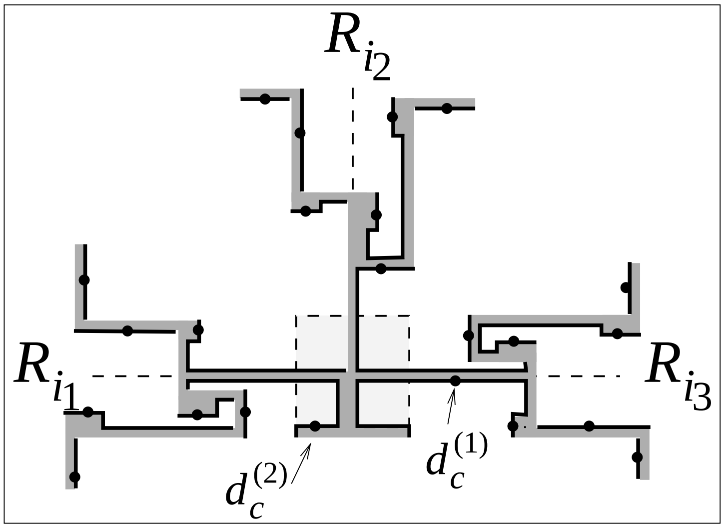

Figure 8: Marked boundary edges and dots around a clause construction, similar to Figure 4(b).

truth assignment of the variables, there is a decomposition into

$$m + 1 + \sum_{i=1}^{n} \frac{n_i}{2}$$

terrain polyhedra: For each polyhedral piece corresponding to an annulus in the two-dimensional proof (i.e., representing a variable loop), choose the decomposition into $n_i/2$ pairs that corresponds to the truth assignment. For each clause region, collect one piece from the c-satisfying L-shape, just as shown in Figure 5; again, the remaining portion of the clause region is covered by an additional piece. It remains to be shown that if there is no truth assignment, then there is no decomposition of this size. The remaining terrain polyhedron consists of the base plates and the rails.

A Small Decomposition Induces a Satisfying Truth Assignment

As in the 2-dimensional case, we can select a set of dots in the polyhedron and consider how to cover these dots by terrain polyhedra. For this purpose, choose one variable dot d_j in each of the "mixed" walls w_j (having a smooth and a rough side) as follows: Pick each dot from the center of a ridge, at $\Theta(\varepsilon^3)$ from the boundary, such that no two dots are at the same level with respect to the ridges. (Recall that there are $\Theta(n^3)$ levels of ridges.) These dots will be referred to as *ridge dots.* For any wall w_j with two smooth sides (i.e., the walls at parity shifts), choose a variable dot such that it is not directly above the rail; these dots will be called *parity dots.* Furthermore, select two clause dots, $d_c^{(1)}$ and $d_c^{(2)}$ (corresponding, respectively, to $d_c^{(1)}$ and $d_c^{(2)}$ in the two-dimensional case), from ridges at unused levels within each

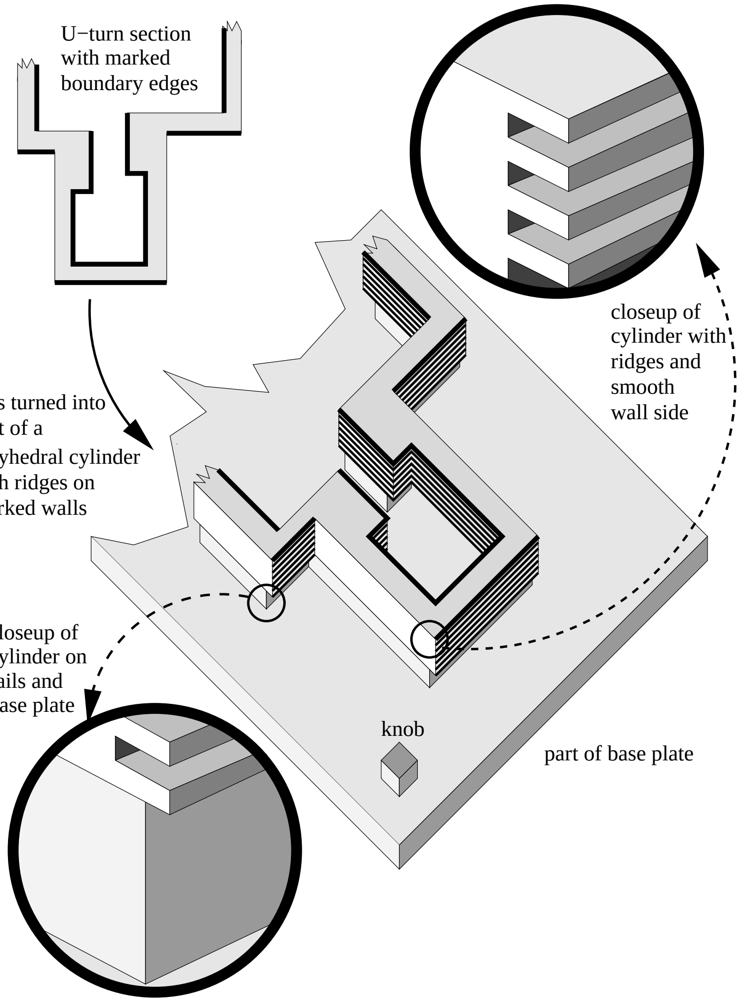

Figure 9: Turning the polygon into a polyhedron: Shown is a polygonal piece for a U-turn with marked boundary edges; the whole polygon consisting of many pieces like this turned into a cylinder with ridges, and mounted onto rails and a base plate. The knob ensures that the base plate and rails require their own terrain polyhedron in a decomposition.

clause region, as shown in Figure 8. One final dot, d_B is chosen from the knob on the base plate. Analogous to the matching of point pairs in the two-dimensional case, we claim that the following must hold for a decomposition into $m+1+\sum_{i=1}^{n}\frac{n_i}{2}$ terrain polyhedra:

(1) No terrain polyhedron can contain more than two ridge dots; moreover, two ridge dots can only be contained in the same terrain polyhedron if they are from overlapping walls.

(2) No terrain polyhedron containing a variable dot $d_c^{(2)}$, or the extra dot d_B, can contain any other dots.

(3) $\sum_{i=1}^{n}\frac{n_i}{2}$ dots are necessary just to cover all ridge dots.

(4) Each P_ℓ contains two variable dots from adjacent dots, or no variable dot.

(5) All variable dots $d_c^{(1)}$ must be contained in the same terrain polyhedron as two variable dots from overlapping walls.

(6) A variable dot $d_c^{(1)}$ can be contained in the same terrain polyhedron as two variable dots, only if these dots belong to a c-satisfying L-shape.

Claim 1 is not hard to check: There is no polygon completely contained in P that sees two nonadjacent ridge dots in the same normal direction, i.e., that U-links them, as in the two-dimensional case. It is the main purpose of the ridges and their dimensions to make sure that this property holds: any polygon fully contained in P that sees two nonadjacent ridge dots along its normal direction must be "almost" axis-parallel, i.e., its normal direction can differ from an axis-parallel unit vector only by a vector of the form $(O(\varepsilon), O(\varepsilon), O(\varepsilon))$. However, the plane defined by such a polygon separates the two dots if they are not from adjacent walls. (Note that Claim 1 is not immediately true for parity dots; in fact, there exists one terrain polygon in P containing all parity dots at once. However, we will see in the following that in decompositions with a minimum number of polyhedra, indeed no two parity dots can be contained in the same terrain polyhedron.)

Claim 2 is also easy to check.

For **Claim 3**, let p_i be the number of parity shifts of variable x_i, let w_j be the wall corresponding to parity shift j and let $s_{i,j}$ be the number of dots between parity shifts j and $j+1$. By construction, there is an odd number of walls between any two successive parity shifts of the same variable; i.e., $s_{i,j}$ is always odd. By Claim 1, this means that we need $\sum_{j=1}^{p_i}\lceil\frac{s_{i,j}}{2}\rceil$ terrain polyhedra to cover all ridge dots for variable x_i, which is equal to $\frac{n_i}{2}$; summing over all variables, the claim follows.

Now **Claim 4** follows by a counting argument: By Claim 2, $m+1$ terrain polyhedra are necessary to cover all dots $d_c^{(2)}$ and d_B, and by Claim 3, the remaining $\sum_{i=1}^{n}\frac{n_i}{2}$ terrain polyhedra are necessary to cover the variable dots. Since no parity dot can be in the same terrain polyhedron as a dot $d_c^{(2)}$ or d_B, they must be covered by the polyhedra containing the ridge dots. For those it is again straightforward to check that no three variable dots can be contained in the same terrain polyhedron, and a ridge and a parity dot can only be contained together when they are from overlapping walls.

Then **Claim 5** follows easily in a similar manner as Claim 4.

Finally, **Claim 6** is straightforward to check, analogous to the two-dimensional case. Since each dot $d_c^{(1)}$ must be contained in some terrain polyhedron, it follows that each clause c must have a c-satisfying L-shape.

This concludes the proof of Theorem 2. □

4. Conclusion

We have provided hardness results for terrain decomposition in two and three dimensions. From our above proofs, the following are easy to verify:

Corollary 1 *It is NP-hard to decide whether an isothetic polygon can be decomposed into k* isothetic *terrain polygons.*

Corollary 2 *It is NP-hard to decide whether an isothetic polyhedron of genus 0 can be decomposed into k* isothetic *terrain polyhedra.*

We conclude with three open questions:

1. Is the terrain decomposition problem NP-hard for *simple* polygons? Our proof of hardness has relied on a polygonal domain with holes.

2. Can one give efficient approximation algorithms for the terrain decomposition problem?

3. Can our results be extended to the problem of computing a decomposition of a *surface* into terrain surfaces? (Such decomposition is often utilized in methods of approximating general polygonal surface models; see Ho, Mitchell, and Silva.[18])

Acknowledgments

We are most grateful to Godfried Toussaint for useful discussions on these problems, help in formulating the problem, and organizing the 1993 Barbados Workshop on Geometric and Computational Aspects of Injection Molding, where this work began. We also thank two anonymous referees for various comments that greatly helped to improve the presentation of this paper.

References

1. *Information Technology for Manufacturing: A Research Agenda.* National Academy Press, 1995.
2. P. Agarwal and P. Desikan. Approximation algorithms for layered manufacturing. *Proc. Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 528–537, 2000.

3. H.-K. Ahn, M. de Berg, P. Bose, S.-W. Cheng, D. Halperin, J. Matousek, and O. Schwarzkopf. Separating an object from its cast. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pp. 221–230, 1997. (To appear, *Comput. Aided Design.*)
4. S. Allen and D. Dutta. Determination and evaluation of support structures in layered manufacturing. *J. of Design and Manufacturing*, 5:153–162, 1995.
5. S. Allen and D. Dutta. Wall thickness control in layered manufacturing. *Comput. Geom. Theory Appl.*, 10:223–238, 1998.
6. E. M. Arkin, S. P. Fekete, J. S. B. Mitchell. Approximation algorithms for lawn mowing and milling. *Computational Geometry*, 17:25–50, 2000.
7. B. Asberg, G. Blanco, P. Bose, J. Garcia-Lopez, M. Overmars, G. Toussaint, G. Wilfong, and B. Zhu. Feasibility of design in stereolithography. *Algorithmica*, 19(1–2):61–83, 1997.
8. C. Baur, S. P. Fekete. Approximation of geometric dispersion problems. *Algorithmica*, 30:451–470, 2001.
9. P. Bose, D. Bremner, and M. van Kreveld. Determining the castability of simple polyhedra. *Algorithmica*, 19(1–2):84–113, 1997.
10. P. Bose and G. Toussaint. Geometric and computational aspects of manufacturing processes. *Comput. & Graphics*, 18:487–497, 1994.
11. P. Bose, M. van Kreveld, and G. Toussaint. Filling polyhedral molds. In *Proc. 3rd Workshop Algorithms Data Struct.*, Springer LNCS vol. 709, pp. 210–221. Springer-Verlag, 1993.
12. B. Chazelle et al. Application challenges to computational geometry: CG impact task force report. Technical Report TR-521-96, Princeton Univ., Apr. 1996. Available at `http://www.cs.princeton.edu/~chazelle/taskforce/CGreport.ps`
13. U. Fößmeier and M. Kaufmann. Drawing high degree graphs with low bend numbers. *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)*, Springer LNCS vol. 1027, pp. 254–266. Springer-Verlag, 1996.
14. R. J. Fowler, M. S. Paterson, and S. L. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Inform. Process. Lett.*, 12(3):133–137, 1981.
15. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, NY, 1979.
16. S. K. Gupta and D. S. Nau. Systematic approach to analyzing the manufacturability of machined parts. *Comput. Aided Design*, 27(5):323–342, 1995.
17. M. Held. *On the Computational Geometry of Pocket Machining*, Springer LNCS vol. 500. Springer-Verlag, 1991.
18. T.-C. Ho, J. S. B. Mitchell, and C. T. Silva. The simplified patch boundary merging algorithm for polyhedral surface simplification. *Technical report*, available at `http://www.ams.sunysb.edu/~jsbm`.
19. I. Ilinkin, R. Janardan, J. Mahji, J. Schwerdt, M. Smid and R. Sriram. A decomposition-based approach to layered manufacturing. *Technical Report CS-TR-00-041*, Dept. of Computer Science & Engineering, U. of Minnesota. Available at `http://www.cs.umn.edu/tech_reports/`.
20. R. Janardan and T. C. Woo. Manufacturing processes. In: *Handbook of Discrete and Computational Geometry*, ch. 46, 851–862. CRC Press LLC, Boca Raton FL, 1997.
21. J. M. Keil. Polygon Decomposition. In: *Handbook of Computational Geometry*, ch. 11, 491–518. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

22. R. Liu and S. Ntafos. On decomposing polygons into uniformly monotone parts. *Inf. Proc. Letters*, 27:85–89, 1988.

23. J. Mahji, R. Janardan, M. Smid, and P. Gupta. On some geometric optimization problems in layered manufacturing. *Comput. Geom. Theory Appl.*, 12:219–239, 1999.

24. J. S. B. Mitchell, editor. Computational Geometry in Manufacturing. Special issue of *Algorithmica*, 19(1–2), 1997.

25. J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford University Press, New York, NY, 1987, pp. 231–239.

26. J. O'Rourke and K. Supowit. *Some NP-hard polygon decomposition problems. IEEE Transactions on Information Theory*, 29:181–190, 1983.

27. J. Pach and G. Tardos. Cutting glass. *Discrete and Computational Geometry*, 24:481–496, 2000.

28. D. Rappaport and A. Rosenbloom. Moldable and castable polygons. *Comput. Geom. Theory Appl.*, 4:219–233, 1994.

29. J. Schwerdt, M. Smid, R. Janardan, E. Johnson, and J. Mahji. Protecting critical facets in layered manufacturing. *Comput. Geom. Theory Appl.*, 16:187–210, 2000.

30. P. Stucki, J. Bresenham, and R. Earnshaw, editors. *Rapid Prototyping Technology.* Special issue of IEEE Computer Graphics and Applications, Volume 15, 1995.

31. T. C. Woo and R. Janardan. Preface: Computational geometry and computer-aided design and manufacturing. *Computer-Aided Design*, 30(4), 1998. Special issue on computational geometry and manufacturing.

32. M. J. Wozny and W. C. Regli, editors. *Computer Science in Manufacturing.* Special issue of Communications of the ACM, Volume 39, 1996.

33. C. K. Yap. Manufacturing and computational geometry workshop report. *Computational Science and Engineering*, pages 82–84, 1995.