THE COMPLEXITY OF WIRE-ROUTING AND FINDING MINIMUM AREA LAYOUTS FOR ARBITRARY VLSI CIRCUITS

M. R. Kramer and J. van Leeuwen

ABSTRACT

Using Thompson's model for VLSI, we prove the following problems to be NP-complete: (a) Given a (connected) graph G and an integer A, can G be embedded in a rectangle of area $\leq A$; and (b) given N pairs of points on a rectangular grid, can wires be routed to connect paired points such that the wires run along grid lines only and the wires do not overlap or cross. The latter problem remains NP-complete if wires are allowed to cross. The results show that the general layout and routing problems in VLSI design are NP-complete, even in the absence of further optimality constraints.

Keywords and Phrases VLSI, chip, circuit, graph, layout, wire, area, routing, NP-completeness.

Advances in Computing Research, vol. 2 pages 129–146.

ISBN: 0-89232-461-9

1. INTRODUCTION

In VLSI theory, circuits are evaluated in terms of their computing time and period and the area required for an embedding on a chip. For our purposes a circuit will simply be a (finite) graph, with nodes corresponding to gates and edges corresponding to wires that connect gates. (We will continue to refer to edges as wires.) Following Thompson [5], we define a chip to be a simple and connected domain of some shape on the two-dimensional grid. (We only consider iso-oriented, rectangular chips, but sometimes weaker assumptions are made.) Each cell of the grid may contain a node or one or two wire segments. Only one wire may cross a given cell boundary and (hence) two wires can at best cross in a single cell. Also, the number of wires incident to a single node is necessarily bounded by four. It will be clear what is meant by an embedding of a graph on a chip, or "in a rectangle."

From a theoretical point of view, problems of (approximate) optimality can be studied adequately within the framework of Thompson's model. (It should be noted that the model assumes a unit size for every node and a unit "width" for every wire, regardless of its length.) Leiserson [4] provides a good survey of some of the results and techniques for obtaining "area-efficient" layouts.

The problem of finding a suitable VLSI design is normally split into two tasks: (1) placement and (2) routing. This division leads to the following questions: First, given an arbitrary graph, can it be embedded in a given amount of area? Second, if one supposes that components have been placed, how hard is it to compute a routing for the necessary connecting wires over the chip? We shall prove both problems to be NP-complete (Garey and Johnson [2]), thereby indicating that fast, exact algorithms are unlikely to exist for either of them and intuitively justifying the use of efficient heuristics. We note that normally a placement of the components is not fixed unless a routing is known to exist and an effort has been made to minimize the total length of the wires used and/or the total area occupied by the design. We shall prove that even the question whether a routing exists at all is NP-complete. Throughout this article we shall assume familiarity with the theory of NP-completeness (see, for example, Garey and Johnson [2]).

2. NP-COMPLETENESS OF THE (CONNECTED) GRAPH EMBEDDING PROBLEM: PRELIMINARY RESULTS

We shall consider the following problems and shall prove each one to be NP-complete in the course of this section and the next:

A: *Given a graph G and a rectangle R, can G be embedded in R?*

B: *Given a graph G and an integer A, can G be embedded in a rectangle of area $\leq A$?*

C: *Given a connected graph G and an integer A, can G be embedded in a rectangle of area $\leq A$?*

We shall first show that Problems A and B are essentially (polynomially) equivalent.

Problem B clearly reduces to $\sqrt{A}$ instances of Problem A, but, because area is given in binary, this is not a polynomial reduction. For practical purposes, though, it is, because of the following result (from which it follows that A can be assumed to be polynomially bounded in the size of the graph).

THEOREM 1. [6] *Every graph of n nodes and degree ≤ 4 can be laid out in $O(n^2)$ area.*

For our main construction we need a very special kind of graph called a "frame graph" (see Figure 1).

Definition. For integers α,γ and b,l with $\alpha > b$ and $\gamma > l$, the frame graph $F(\alpha, \gamma, b, l)$ is defined to be the α-by-γ grid, with a b-by-l "window" cut out in the middle.

$F(\alpha, \gamma, b, l)$ has $\alpha\gamma - bl$ nodes, and its "natural" embedding (see Figure 1) occupies an α-by-γ rectangle.

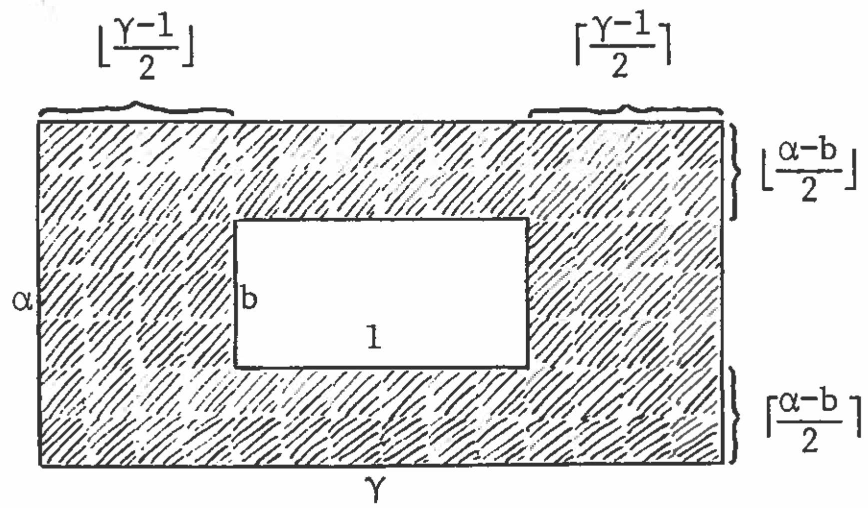

Figure 1. A frame graph. The shaded part is grid-connected.

THEOREM 2. *Let $\alpha \geq b + 36bl$ and $\gamma \geq l + 36bl$. Then the only embedding of $F(\alpha, \gamma, b, l)$ possible in area $\alpha\gamma$ is the natural embedding, in a α-by-γ rectangle.*

PROOF. By just counting how many cells are required for nodes alone, it follows that an embedding in a rectangle of area $\alpha\gamma$ can "waste" no more than bl cells on open space and wiring. Consider an embedding of $F(\alpha, \gamma, b, l)$ in an α_1-by-γ_1 rectangle, with $\alpha_1\gamma_1 = \alpha\gamma$. For the argument below we use the following terminology. Nodes in the left α-by-$\lfloor(\gamma - 1)/2\rfloor$ part of $F(\alpha, \gamma, b, l)$ will be called "red," those in the upper $\lfloor(\alpha - b)/2\rfloor$-by-$\gamma$ part "green." (Never mind that in this way some nodes are both red and green.) We shall show that the red and green nodes, and likewise the remainder of $F(\alpha, \gamma, b, l)$, must occur in natural position, or else more than bl cells would be needed for additional wiring. We consider essentially two different cases.

Case 1. $\alpha_1 \leq \alpha$.

Partition the rectangle in 3-by-3 boxes (we ignore the rounding effect at the border). A box is called full when it has red nodes in all its nine cells. (The nodes in a full box necessarily are in natural position, by just observing the degree of the nodes.) Note that there must be at least $1/9 \cdot \alpha \cdot \lfloor(\gamma - 1)/2\rfloor \geq 2\alpha bl$ boxes that contain at least one red node, hence at least $2\alpha bl/\alpha_1 \geq 2bl$ strips of dimension α_1-by-3 that contain such boxes in the partitioned rectangle. Now suppose that every one of these $2bl$ strips contains (1) a nonfull box with a red node, or (2) only full boxes of red nodes but two of them separated by a "blank" box or some other sort of waste on extra wiring. Then $\geq 2bl$ cells would be wasted on necessary excess wiring, thereby contradicting that at most bl cells were available for this purpose. We conclude that there must be an α_1-by-3 strip that contains only full boxes of red nodes, with no waste due to wiring the red nodes to other (red) nodes. The full boxes must be adjacent and are (necessarily) a natural strip from the grid. There are two possibilities: (1) it is a complete "vertical" strip from the red part of the graph (and $\alpha_1 = \alpha$) or (2) it is a complete "horizontal" strip from the red part. In either case the strip would be at least $18bl$ red nodes long. Consider the immediately adjacent single column or row of (red) nodes connected to the strip. If it is not connected to the strip in natural order but (say) one node is further out, then a packing consideration shows that none of the nodes from the column or row can be directly adjacent to the strip either (the problem being the routing of the wires to and within the added column or row). But this gives an excess in cells for wiring $\geq 18bl$, which is more than is permitted. When this argument is continued, it follows that every part of $F(\alpha, \gamma, b, l)$ must be in its natural place and order.

Case 2. $\alpha_1 > \alpha$

This case is the same as requiring that $\gamma_1 < \gamma$. It can be handled exactly like Case 1, considering the green nodes instead of the red ones to find a solid strip and force the natural embedding of the graph.

The rounding effect in the argument is easily absorbed in the slack of the bounds on α and γ, assuming that b and l are not too small. □

PROPOSITION 1. *Problems A and B are "polynomial time Turing equivalent," i.e., equivalent in the sense of Cook.*

PROOF. Consider an instance of Problem A, requesting the embedding of an n-node graph G in some b-by-l rectangle (b and l given in binary). By Theorem 1 we may assume that b and l are at most $O(n^2)$. Now let $\alpha = b + 36bl$ and $\gamma = l + 36bl$ and consider the instance of problem B requesting the embedding of $G \cup F(\alpha, \gamma, b, l)$ in area $\alpha\gamma$. Note that the instance of Problem B has size bounded by $O(n^8)$, which is polynomial in the size of Problem A. By Theorem 2 the only embedding of $F(\alpha, \gamma, b, l)$ in area $\alpha\gamma$ is the natural embedding, and thus the instance of Problem B is solvable if and only if G fits in the middle b-by-l window.

Next consider an instance of Problem B. By Theorem 1 we may assume that the area of the embedding is bounded by $O(n^2)$. Thus, Problem B can be solved by solving $O(n)$ instances of Problem A. □

(From the results below it follows that Problems A and B are, in fact, polynomially equivalent in the sense of Karp. See Garey and Johnson [2], Section 5.2, for a discussion of these concepts.)

In the remainder of this section we shall show that Problem A (hence, Problem B) is NP-complete. We will make use of the following problem, which is known to be NP-complete in the strong sense (i.e., "unary NP-complete," see Garey and Johnson [2] Section 4.2.2):

> *3-PARTITION. Given positive integers m,B and a set of 3m integers* $a_1, \ldots, a_{3m}$ such that $\frac{1}{4}B < a_i < \frac{1}{2}B$ and $\sum_i a_i = mB$, *does there exist a partition of this set into m disjoint subsets such that each subset sums to B?*

(Note that by the size constraint on the a_i each subset of a "3-partition" is forced to contain exactly 3 elements.)

THEOREM 3. *Problem A is NP-complete (in the strong sense).*

PROOF. Problem A obviously is in NP. We shall design a pseudo-polynomial transformation from 3-PARTITION to Problem A. Let an instance of 3-PARTITION be given. For each integer a_i, design a component $C(a_i)$ that is a 4-by-$5ma_i$ grid, as shown in Figure 2A. (It should be obvious that the natural embedding of a component is the only one that is minimum, i.e., that uses no extra cells for wiring.) Let G be the collection of components $C(a_i)$, $1 \leq i \leq 3m$. Consider the instance of Problem A that requests the embedding of G in the rectangle R of size $4m$-by-$5mB$, as shown in Figure 2B. If the instance of 3-PARTITION has a solution, then so does the instance of Problem A. The (3) components corresponding to a subset that sums to B can be packed into a 4-by-$5mB$ strip, and m such strips just fit in R (horizontally). However, the converse is true also. For suppose we have a solution to the instance of Problem A. G has as many nodes as there are cells in R, and hence every component is forced to be embedded in minimum area. This is the "brick" form shown in Figure 2A. Because $5ma_i > 4\,m$, no brick can be placed vertically. It follows that R decomposes into m horizontal strips of width 4 that are completely packed. The strips translate back into subsets that are a solution to the instance of 3-PARTITION.

Observe that the reduction can be computed in time polynomial in m and B. The remaining conditions for pseudopolynomial reductions (Garey and Johnson [2], p. 101) are easily verified. By Garey and Johnson [2], Lemma 1, Problem A is NP-complete (in the strong sense). □

THEOREM 4. *Problem B is NP-complete (in the strong sense).*

PROOF. Problem B again, obviously, is in NP. We shall design a pseudopolynomial reduction from 3-PARTITION that is very similar to that in the proof of Theorem 3. Let an instance of 3-PARTITION be given. For each a_i, we design a component $C(a_i)$ as before. Let G be the graph consisting of all $C(a_i)$ $(1 \leq i \leq 3m)$ together with one $F(4m + 4, 5mB + 4, 4m, 5mB)$, and let $A = (4m + 4)(5mB + 4)$. See Figure 3 for an illustration of the last component of G. Consider now the instance of Problem B that requests an embedding of G in area "A."

The remainder of the proof is very similar as for Theorem 3. Note that

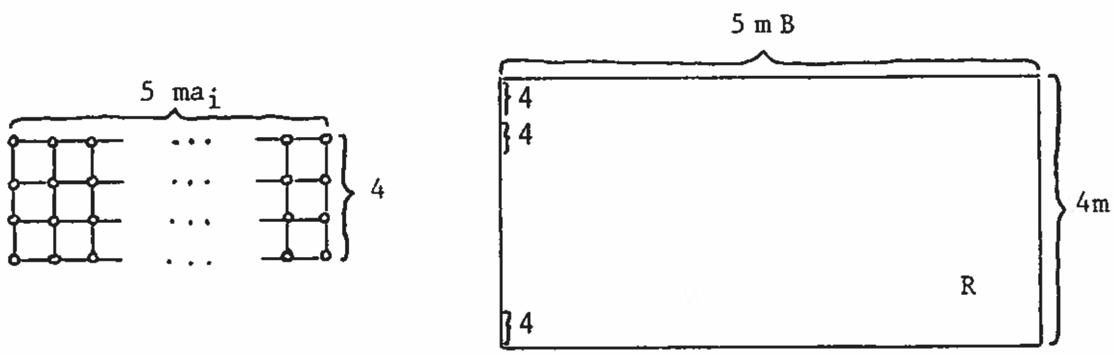

Figure 2. (A) The "brick" form. (B) Packing bricks in R.

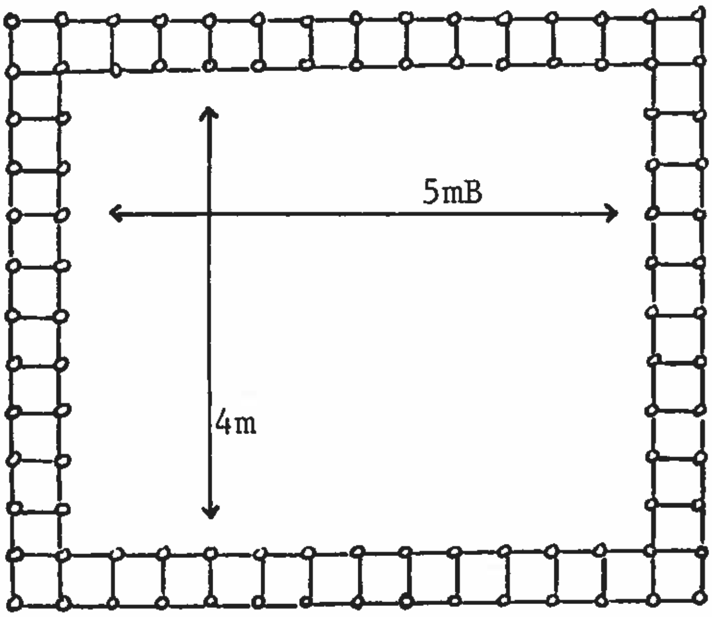

Figure 3. Frame graph belonging to G.

G has as many nodes as there are cells in area A, thus forcing each component to be minimally embedded. But this forces the frame graph into its natural form and the remaining components of G to fill the interior part in completely the same way as before. □

3. NP-COMPLETENESS OF THE (CONNECTED) GRAPH EMBEDDING PROBLEM: GENERAL CASE

The proofs of Theorems 3 and 4 basically use the same reduction from 3-PARTITION and heavily depend on being able to construct a graph G with many components ("a very disconnected graph"). In this section we shall modify the proof of Theorem 4 so as to obtain a connected graph. For an introduction we shall first discuss a way to get a "smaller" proof of Theorem 4. Recall that the proof of Theorem 4 led to a graph G that necessarily had an embedding that divided naturally into m disjoint strips. Thus, we could have used a "frame" with m windows (of width 4, and separated by chains of width 2) and still have gotten the same effect and a valid reduction. The next step is to place the windows horizontally rather than vertically in sequence and to shrink them in size by cutting the factor m in width. Thus, we use for $C(a_i)$ the 4-by-$5a_i$ "brick" from Figure 4A and as "frame" the F(8, $5mB + 2m + 2$, 4, $5mB + 2m - 2$) with m interior, separated windows as in Figure 4B. It is not hard to see that the new construction still yields a valid, pseudopolynomial reduction from 3-PARTITION to Problem B. Observe that A went down from about $20m^2B$ to about $40mB$ here.

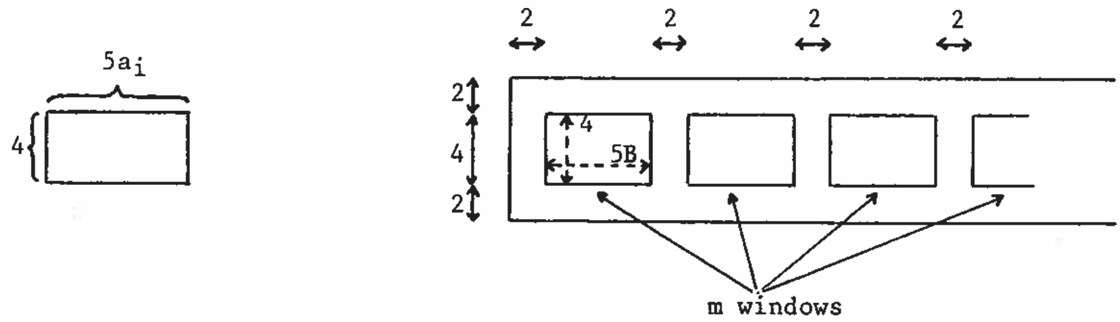

Figure 4. (A) The "brick" form. (B) Frame with m "windows."

To get a connected graph from the frame in Figure 4B and the $3m$ separate components, we proceed as follows. First, we "stretch" the frame by $3m$, by inserting nodes in the supporting columns. (This gives the windows height $3m + 4$, but their width remains $5B$.) Next we span exactly $3m$ lifelines (edges) through the middle of the frame all the way across from one end to the other, and finally we pin each $C(a_i)$ to its own lifeline. In this way each component is connected to the frame, but their position is not more restricted than before ("components can be freely shifted along their lifeline"). The supporting frame will be made bigger, so as to force a unique embedding.

THEOREM 5. *Problem C is NP-complete (in the strong sense).*

PROOF. Problem C obviously is in NP. We shall design a pseudo-polynomial reduction from 3-PARTITION to Problem C very much as in the previous theorems. Let an instance of 3-PARTITION be given. We shall construct a connected graph C as outlined above. To this end, we start with a frame $F(\alpha, \gamma, 3m + 4, 5mB + 2m - 2)$ and divide its interior into m compartments of width $5B$ as shown in Figure 5A. (α and γ will be chosen later.) Span the $3m$ lifelines across. For each integer a_i ($1 \leq i \leq 3m$), design $C(a_i)$ as shown in Figure 5B, which is essentially the 4-by-$5a_i$ brick but with an extra row in the middle with one node that connects $C(a_i)$ to the ith lifeline. (Note that besides the end points the lifelines all carry precisely one additional node this way, which enables us to position the bricks freely in any one of the compartments.) Consider the instance of Problem C that requests the embedding of G in $A = \alpha\gamma$ area, with $\alpha = (3m + 4) + 36(3m + 4)(5mB + 2m - 2)$ and the width of the frame chosen as $\gamma = (5mB + 2m - 2) + 36(3m + 4)(5mB + 2m - 2)$. If the instance of 3-PARTITION has a solution, then it follows by design that G can be laid out in the natural way with the components corresponding to every subset neatly arranged in the m windows. Conversely, suppose we have a layout of G in $\alpha\gamma$ area. By the choice of α and γ, it

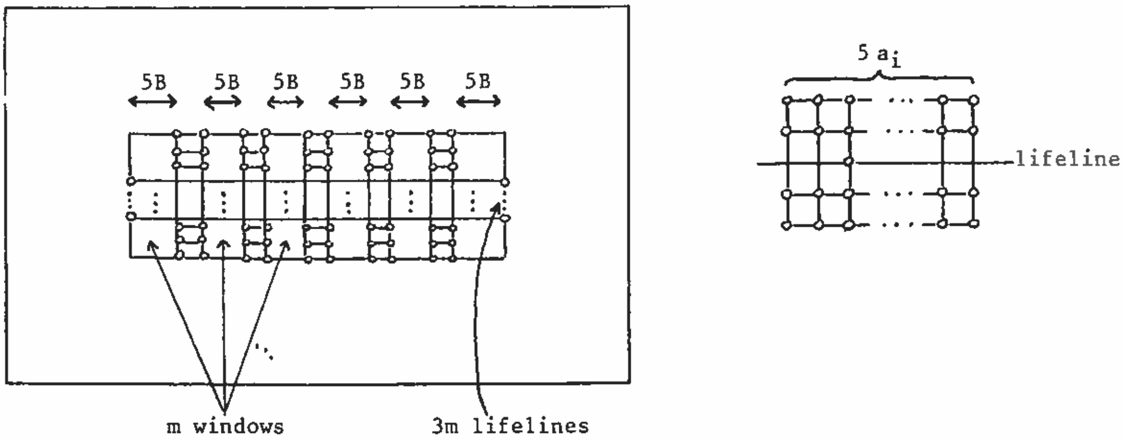

Figure 5. (A) Frame with m windows and $3m$ lifelines. (B) Modified brick with extra row in the middle.

follows from Theorem 2 that the surrounding frame can be laid out in only one way, namely, the natural embedding. The remaining part of G must be laid out in the $(3m + 4)$-by-$(5mB + 2m - 2)$ interior window. Note that all cells beside those occupied by the lifelines in Figure 5A are needed just to accommodate the nodes from all $C(a_i)$ and the separating columns. This forces the lifelines to occupy no more cells than they do in the natural embedding, or otherwise no embedding could exist for the remaining part of G. It is easily seen that this forces the lifelines to be embedded exactly as in the natural embedding (it is the only possible minimum embedding within the frame) and all other parts to be minimally embedded as well. This puts the $C(a_i)$s neatly in groups of three in the windows $5B$ wide, and the embedding translates directly into a solution of the instance of 3-PARTITION. The reduction is easily seen to be pseudopolynomial, and as before we can conclude that Problem C is NP-complete. □

Several further observations can be made. Theorem 2 remains valid even if we measure "area" just by the number of occupied cells and not by the size of the (smallest) enclosing rectangle. This means that Theorem 5 (the embedding problem for connected graphs) remains valid for this, least restrictive, notion of area as well. (This is also true for Theorem 4 by a direct argument.) By adding more lifelines and frame connections to the graph constructed in the proof of Theorem 5, one can easily show that the embedding problem remains NP-complete for connected graphs of any (fixed) higher degree of connectivity.

Although the proof of Theorem 5 came "close," it stopped short of proving the NP-completeness of embedding a planar connected graph in minimum area. Also, the NP-completeness question seems open for (connected) graphs of degree ≤ 3.

4. NP-COMPLETENESS OF WIRE-ROUTING: PRELIMINARY RESULTS

In this section and the next we shall consider the following problem and shall prove it to be NP-complete.

D: *Given N pairs of points on a grid, is there a routing of the N wires connecting the pairs of points (intersections of wires allowed)?*

Problem D (ROUTING) is proved NP-complete by means of a polynomial transformation from 3-SAT (cf. [2]). The problem remains NP-complete if wires are not allowed to cross (CROSS FREE ROUTING). Both problems remain NP-complete if the pairs of points are required to be fully disjoint.

The NP-completeness proof is facilitated by considering a useful intermediate problem. Let an "obstacle" be any (connected) rectangular domain of cells.

E: *Given N pairs of points and M (rectangular) obstacles on a grid, is there a routing of N wires connecting the pairs of points such that no wire intersects an obstacle (i.e., no wire is routed through an obstructed cell)?*

For technical reasons we must assume that obstacles are given by an explicit listing of the cells they cover. (We shall get rid of this assumption later when we construct instances of Problem E in which the obstacles have a size that is polynomially bounded in N.) We shall prove that both Problem E and its cross-free version, CROSS FREE OBSTACLE ROUTING, are NP-complete.

LEMMA 1. *Problem D and problem E are polynomially equivalent in the sense of Karp (with or without wire crossings allowed).*

PROOF. Clearly Problem D is a special instance of Problem E, and we only have to consider the converse reduction. Given an instance I of Problem E, we construct an equivalent instance I′ of Problem D. Take the N pairs of points, but replace each of the M obstacles by an additional set of pairs (called "obstruction pairs") as follows. Put a point in each cell of the obstacle and combine adjacent points into pairs. This can be done in such a manner that all pairs are disjoint (Figure 6). Only when an obstacle consists of an odd number of cells will there be one point that cannot be paired to a buddy and thus must be paired to itself. (We shall see later that this can be avoided in the application of the transformation in Theorem 7.) Clearly I′ can be constructed in polynomial time.

To prove that I and I′ are equivalent, we observe the following. Clearly

Figure 6. Construction of obstruction pairs.

obstruction pairs can be connected by drawing the "straight" wire of unit length through the common boundary of the two cells containing the pair. Thus, any solution to I immediately translates into a solution of I′. (The wires connecting the obstruction pairs do not interfere with any other wires and leave a routing cross free if the given one was.) Conversely, consider any solution to I′. As the two points of any obstruction pair necessarily are in adjacent cells, we may assume that their connecting wire runs directly through the common cell boundary. (If it did not, we could change the wiring so that it does.) It means that in I′ the obstruction pairs together block out certain regions (the original obstacles!) for use by other wires and leave the remaining area completely free and open. It follows that the solution to I′ translates back immediately into a solution of I. (Again, if the solution to I′ is cross free, then so is the solution to I.) □

5. NP-COMPLETENESS OF WIRE-ROUTING: GENERAL CASE

By lemma 1 it suffices to prove the NP-completeness of Problem E (OBSTACLE ROUTING).

THEOREM 6. *Problem E is NP-complete.*

PROOF. It should be clear that both Problem E and its cross-free variant are in NP. Thus, for the NP-completeness proof, it suffices that a known NP-complete problem can be polynomially transformed to them. We will transform from 3-SAT (cf. [2]).

Let an instance of 3-SAT be given. It consists of a collection C of clauses in disjunctive form that must be simultaneously satisfied, with three literals per clause and with variables (and their negations) chosen from x_1 to x_n. To construct an equivalent instance of OBSTACLE ROUTING, we need some intuitive terminology first. An "i-street" ($i \geq 1$) consists of two parallel lanes (rows or columns) of the grid, bordered by blocked cells and separated by i fully blocked lanes (see Figure 7). Only where a side street begins (a junction) or another street intersects it (a crossing) will the regular lane structure be modified somewhat (within the bound-

Figure 7. A 1-street, with points x and y.

aries of the street, though). When a wire must be routed from a point x at one end of the street to a point at the other end (see Figure 1), there essentially are only two possibilities: either through the first lane, or through the second. We will identify these options with "false" and "true" and label one lane with x^0 and the other with x^1 to distinguish them for our purposes. In a number of cases it will be necessary to let lanes switch roles. This is achieved by redirecting the wire to an intermediate point and forcing the continuation from another, such that the role-switching is effectuated. Figure 8 shows the basic inverter that can be inserted in an i-street. Inversion makes it possible to switch to the truth-value assignment for $\bar{x}$ (the negation of x) when necessary. But inverters also allow for the possibility of blocking both lanes of a street for routing alien wires. By placing one inverter (or two, to neutralize the effect on the lane interpretation) between any two consecutive sites where a special construction has taken place, one can assure that wires can only be routed through the streets we want them to use.

Given a collection of clauses C as specified, we construct an equivalent instance of Problem E as follows (see Figure 9). In brief, the instance will consist of n vertical 5-streets representing the n variables and $3 \mid C \mid$ horizontal 2-streets that connect, in couples of 3 corresponding to the literals of a clause, to "plazas" representing the individual clauses. Horizontal streets begin at a junction with the vertical street corresponding to the proper variable x contributed to the clause. If $\bar{x}$ is to be contributed, then an inverter is put in the vertical street just before the junction to get the

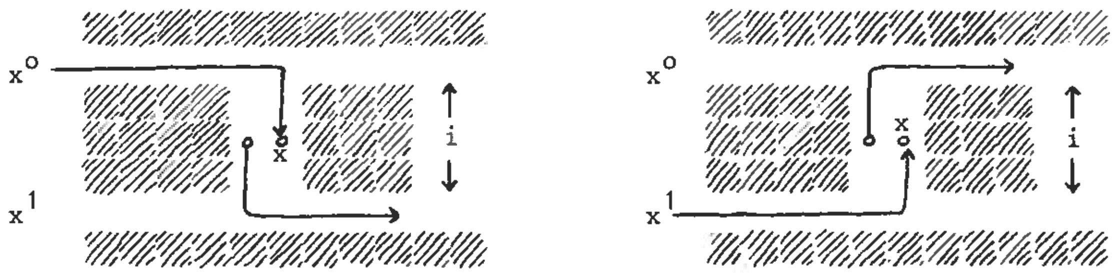

Figure 8. An inverter.

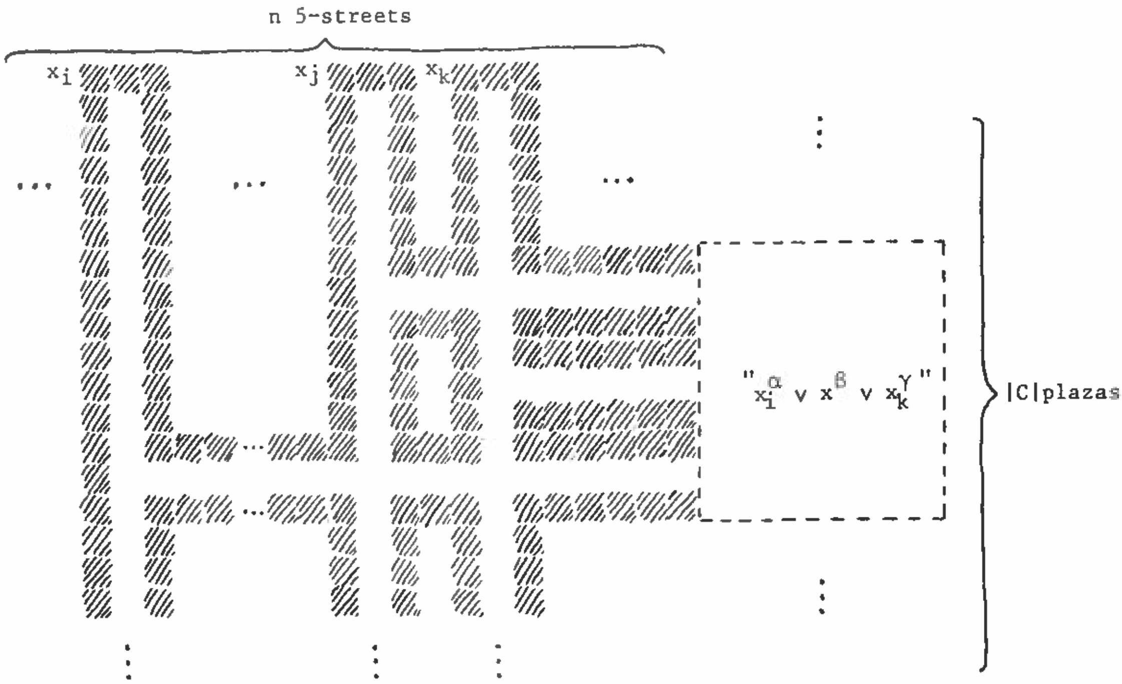

Figure 9. Outline of the routing problem. ($x_i^\alpha \vee x_j^\beta \vee x_k^\gamma$ is the general form of a clause of 3-SAT.)

desired effect and another one immediately after it is used to reenact the original interpretation of the lanes. Note that horizontal streets must cross vertical ones to their right, and we shall need a special interrupt construction to let wires "cross over" while preserving the interpretation of the lanes.

A junction should be constructed such that the truth-value assignment of the corresponding variable x (as reflected by the wiring down the street) is copied into the horizontal street consistently, while a downward routing is reestablished afterward. Figure 10 shows how this can be done using only three extra points. Observe that the routing in the junction is com-

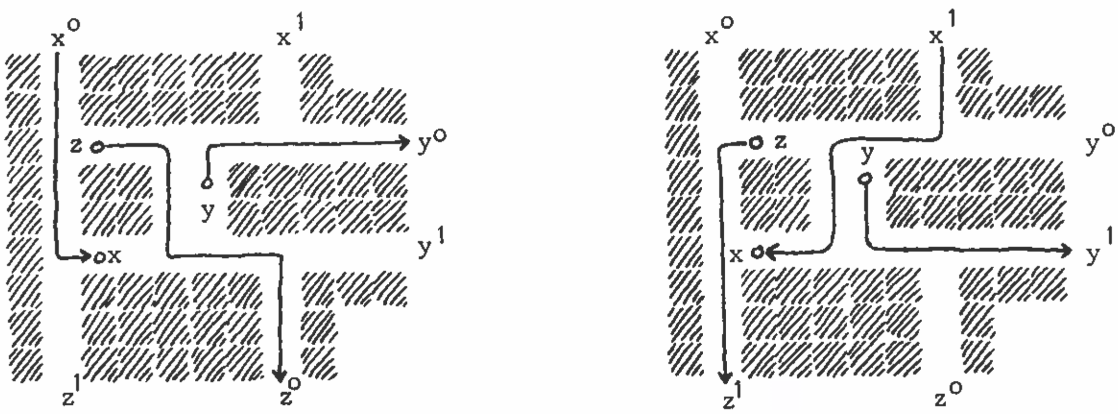

Figure 10. A junction ($x \equiv y \equiv z$).

pletely determined, depending on whether the x-wire comes in through the left (x^0) or through the right (x^1) lane. It correctly splits off the 0–1 interpretation into the horizontal street, but in the process the wire down the vertical street switched lanes. Thus, an inverter must be put in immediately afterward, to bring the wire back into its original lane. Note that wires cannot (and do not) cross in a junction.

To enable a horizontal street and a vertical street to cross while preserving their "value," we need a cross-over construction that interrupts (when necessary) and reestablishes the wire routing in the various intersecting lanes. In fact, we need two separate cross-over constructions: one in case wires are allowed to cross (Figure 11) and one in case they are not (Figure 12). In Figure 11 the wires are, in fact, forced to run straight on, as conveniently located pairs (s,t) and (u,v) would be blocked otherwise. Although it seems that we could have let the wires cross without further steps, these two pairs are necessary to prevent one of the wires from switching lanes. One easily verifies that the routing in a cross-over is uniquely determined by the lanes through which the x-wire and the y-wire enter. Note that the cross-over makes essential use of the fact that wire crossings are permitted. Figure 12 is more complicated and does require that the routing in one direction (down) the vertical street, for

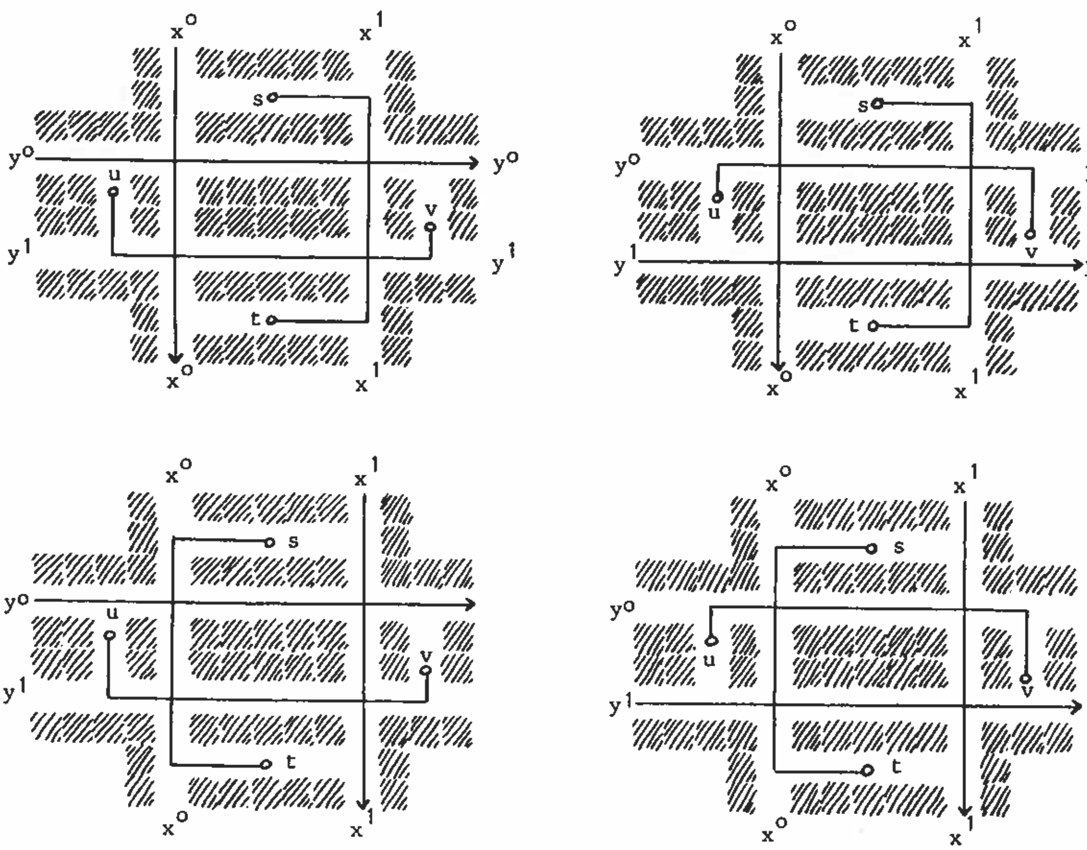

Figure 11. A cross-over.

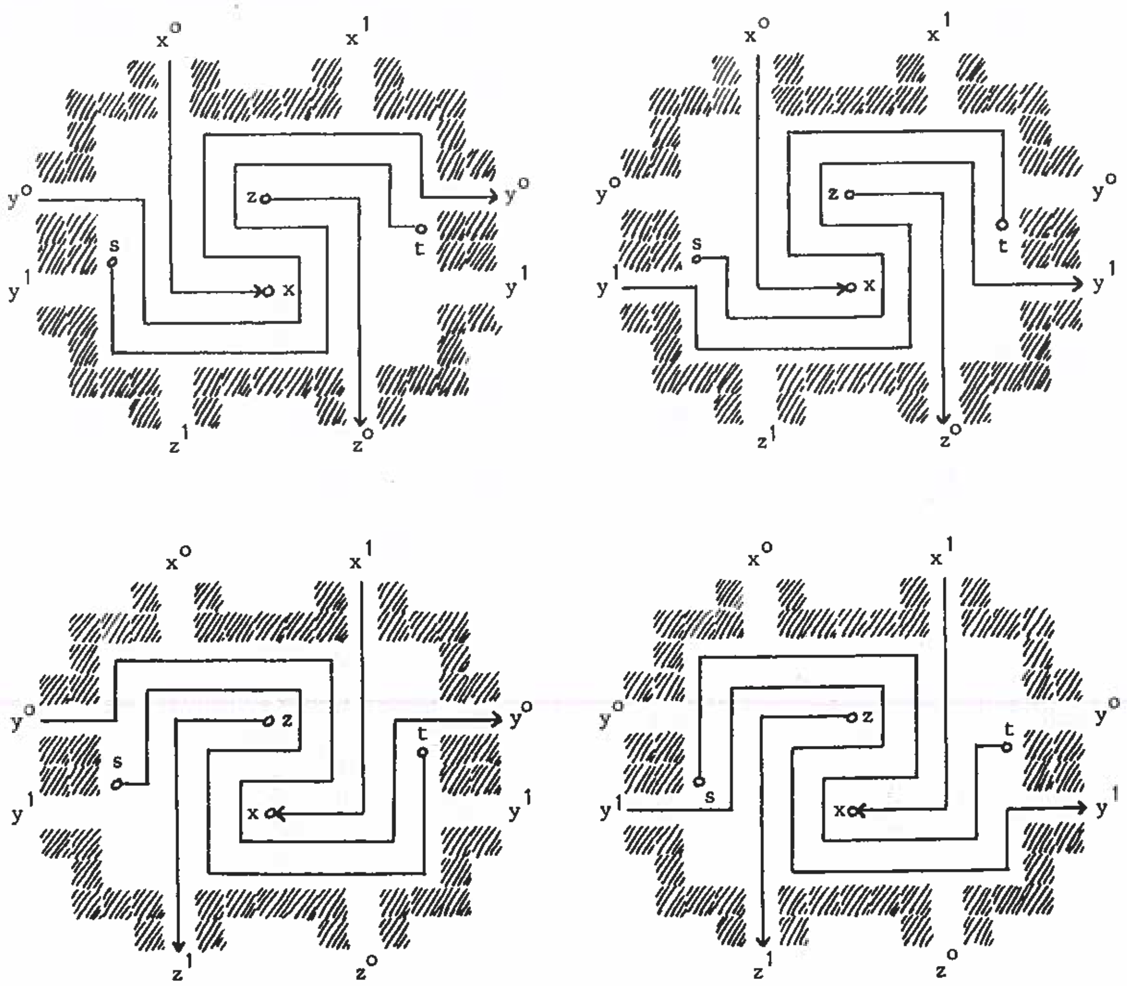

Figure 12. A cross-over for the model in which no wire intersections are allowed ($x \equiv z$).

example, is interrupted. There is no other way if wires are not allowed to cross. The extra points and the modified pairing not only guarantee that a cross-free "passage" can be effectuated but force it to be as shown in the various instances of Figure 6, which differ depending on where the x-wire and y-wire enter. Note that the x-wire (essentially turned into the z-wire) is forced to switch lanes, and thus an inverter must be inserted in the vertical street, just below the cross-over. Note that Figure 12 is correct by virtue of the condition that wires should not cross.

Finally plazas must be designed that properly reflect the evaluation of the clauses. Thus, they should allow for a (internal) routing if and only if at least one of three incoming streets brings in a wire through its "true" lane. Figure 13 shows a suitable plaza for clauses $x \vee y \vee z$. Note that the horizontal street corresponding to y must include an inverter just before entering the plaza and that the horizontal street corresponding to z must be led around to enter the plaza at the required right side. (This

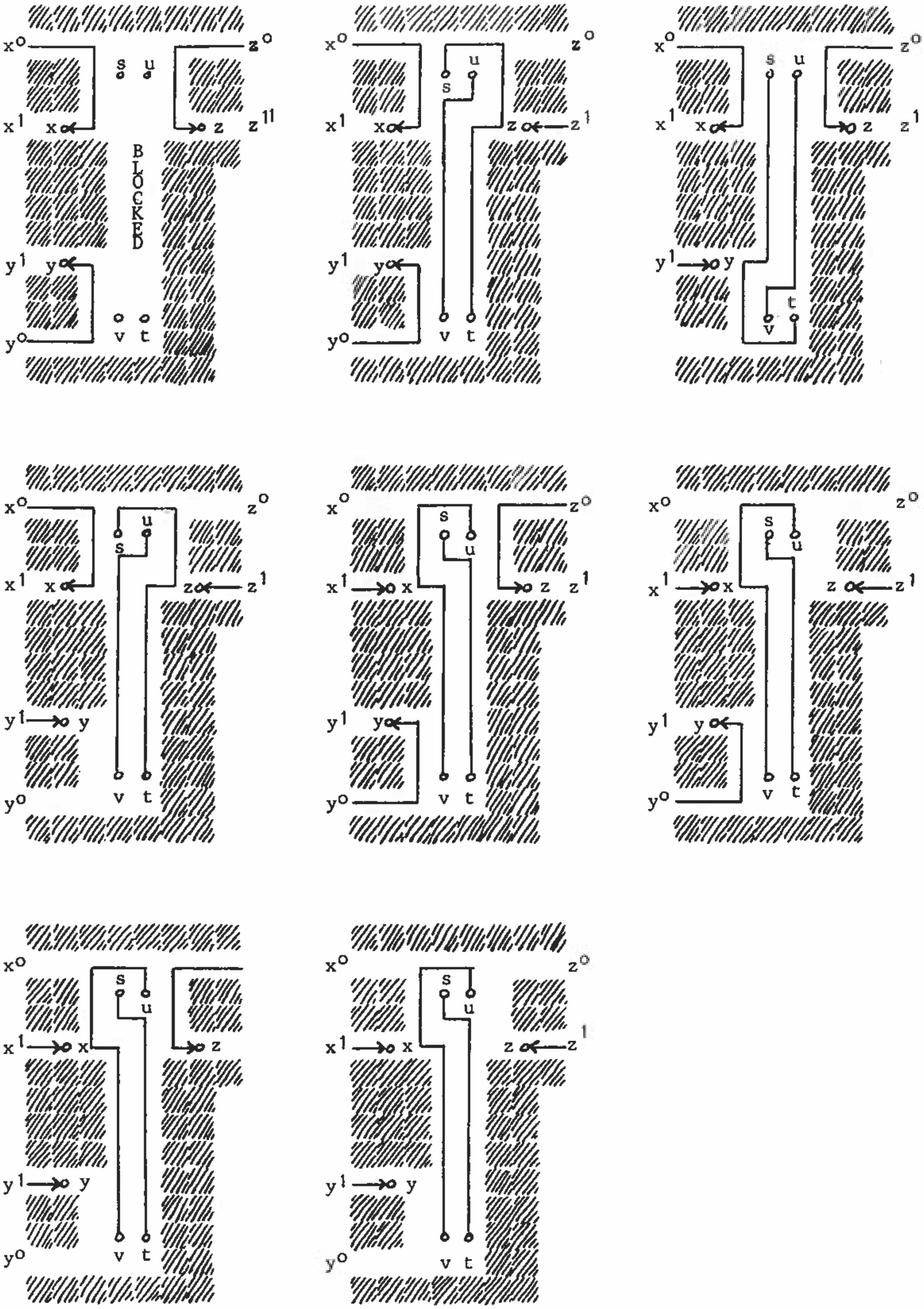

Figure 13. A plaza for the clause $x \vee y \vee z$.

requires an inverter in the z-street as well because, in bending around, the lanes switch positions.) It is also necessary to put an inverter in the x-street, to prevent a wire that should be routed on the plaza from running into an unblocked lane. Although we left open exactly by what distance horizontal streets are to be separated (a distance of 12 blocked lanes would certainly do), Figure 13 assumed rather arbitrarily a distance of 4. This can always be achieved by bending streets closer to one another. The pairs (s,t) and (u,v) are strategically chosen so as to let a routing through the narrow "gorge" of the plaza exist in case there is some room either at the s–u or at the v–t end to lead a wire around. Just in case $x \equiv y \equiv z \equiv$ false(0), all this room is taken by the x-, y-, and z-wires (and necessarily so, for otherwise the chances for a routing on the plaza are nil anyway) and no routing for the pairs exists, unless the overlap constraint of the model is violated.

We conclude that the instance of Problem E is a consistent image of the instance of 3-SAT and that the clauses of C are simultaneously satisfied if and only if a complete (cross-free) routing exists. The transformation requires the construction of $O(n \mid C \mid)$ special elements (streets, junctions, cross-overs, inverters, and plazas) that have size $O(n)$, $O(\mid C \mid)$, or $O(1)$. The entire construction is easily completed in time polynomial in the size of the given instance of 3-SAT. Thus, 3-SAT polynomially transforms to Problem E. □

Observe in the proof of Theorem 6 that the instance of Problem E obtained from the instance of 3-SAT fits in only $O(n \mid C \mid)$ area, which means in particular that the size of the obstacles needed remains bounded by a fixed polynomial in n (as $\mid C \mid \leq 8n^3$) and thus in the number of pairs N actually constructed. The proof of Theorem 6 carries some similarities to a construction in [1].

THEOREM 7. *Problem D is NP-complete.*

PROOF. Clearly Problem D belongs to NP. The result now follows by combining Lemma 1 and Theorem 6. □

Note that Problem E only served as a useful intermediate problem and that the proofs together give an immediate polynomial transformation of 3-SAT to Problem D. By being a bit more careful, one can make sure that all intermediate obstacles have an even number of cells, which means that pairs not only are disjoint but may be assumed to consist of distinct points only.

REFERENCES

1. Fowler RJ, Paterson MS, Tanimoto SL: Optimal packing and covering in the plane are NP-complete. *Inf Proc Lett* 12:133–137, 1981.
2. Garey MR, Johnson DS: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco CA, 1979.
3. Kung HT, Sproull B, Steele G (eds): *VLSI Systems and Computations*. Computer Science Press, Rockville MD, 1982.
4. Leiserson CE: *Area-Efficient VLSI Computation*. MIT Press, Cambridge MA, 1983.
5. Thompson CD: *A Complexity Theory for VLSI*. PhD thesis (Technical Rep. CMU-CS-80-140) Carnegie-Mellon University, Pittsburgh PA, 1980.
6. Valiant LG: Universality considerations in VLSI circuits. *IEEE Trans Comp* C-30: 135–140, 1981.