

Purdue University

**Purdue e-Pubs**

---

Department of Computer Science Technical  
Reports

Department of Computer Science

---

1981

## **Optimal Tile Salvage**

Francine Berman

Frank Thomsons Leighton

Lawrence Snyder

**Report Number:**

81-396

---

Berman, Francine; Leighton, Frank Thomsons; and Snyder, Lawrence, "Optimal Tile Salvage" (1981).  
*Department of Computer Science Technical Reports*. Paper 322.  
<https://docs.lib.purdue.edu/cstech/322>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.  
Please contact [epubs@purdue.edu](mailto:epubs@purdue.edu) for additional information.

## Optimal Tile Salvage

*Francine Berman\**

Purdue University

*Frank Thomson Leighton*

Massachusetts Institute of Technology

*Lawrence Snyder*

Purdue University

### ABSTRACT

A map is a tiling of a finite region of the plane with unit squares such that some tiles have been removed. The optimal  $x \times y$ -Tile Salvage Problem is: Given a map, find the maximum number of non-overlapping  $x \times y$  tiled rectangles. A polynomial time algorithm is given for the  $1 \times 2$  case, i.e., adjacent pairs. It is shown that the problem is NP-complete for the  $2 \times 2$  case. A polynomial time algorithm is presented for finding  $2 \times 2$  groups that is no worse than one half optimal. The problem is motivated by a technique for increasing the size of very large scale integrated (VLSI) circuit chips.

---

The research described herein is part of the Blue CHIP Project. Funding is provided in part by the Office of Naval Research under Contract No. N00014-80-K-0816 and Contract No. N00014-81-K-0360, Special Research Opportunities Program, Task SRO-100.

\*Additional support from National Science Foundation Grant MCS-80-05387.

## Optimal Tile Salvage

*Francine Berman*

Purdue University

*Frank Thomson Leighton*

Massachusetts Institute of Technology

*Lawrence Snyder*

Purdue University

### Introduction

In very large scale integrated (VLSI) circuit technology, there is a continuing effort to place more transistors on a chip. Although this is often accomplished by making the devices denser with more sophisticated processing technology, the necessary equipment is complex and expensive, and the procedures are very exacting. Another approach is to make the chips larger. But for a fixed fabrication process this causes a reduction in yield, i.e., a reduction in the average number of error-free chips per process run [1, 2]. An alternative method of increasing the number of transistors per chip is to use configurability [3, 4].

Recall that chips are regions of a larger unit called a "wafer." Ordinarily after fabrication a wafer is tested and faulty chips are marked; then the wafer is diced using scribing corridors provided for the purpose, the faulty chips are discarded, and the functional chips are packaged. To use configurability, this process is changed somewhat. Special circuitry is added in the scribing corridors that enables adjacent chips to be connected. Then, after the wafer is tested, adjacent functional chips are connected into a group and the entire group is used as a single enlarged chip.

For example, a large circuit can be divided into two parts,  $A$  and  $B$ , and the wafer can be laid out in a checkerboard pattern of alternating  $A$  and  $B$  die types. After testing, adjacent functional  $A$ ,  $B$  pairs can be located. Each pair can be diced as a unit and configured to yield a double chip that implements the large circuit.

Notice that in the checkerboard pattern, there are four adjacent  $B$ 's with which each  $A$  could be paired. Choosing a particular  $B$  could influence which  $B$ 's are available to pair with other  $A$ 's. The problem is to find the *maximum* number of adjacent  $A$ ,  $B$  pairs given a map of the error-free dice on the wafer. The purpose of this paper is to show that this problem can be solved in polynomial time, that important generalizations of this problem are NP-complete, and that there is a linear-time algorithm for approximating the NP-complete problem.

### **Preliminaries**

A map is an  $n^{\frac{1}{2}} \times n^{\frac{1}{2}}$  region of the plane tiled with unit squares some of which have been removed. The tiles represent functional chips and the interstices represent faulty chips (see Figure 1). The optimal  $x, y$  Tile Salvage Problem is to find the maximum number of non-overlapping  $x \times y$  tiled rectangles. The orientation of the rectangles does not matter. For example, the selection of  $A, B$  pairs mentioned above is an instance of the  $1 \times 2$  Tile Salvage Problem. The intended semantics, that each group contain an  $A$  and a  $B$ , is accomplished by patterning the wafer in an  $A, B$  checkerboard.

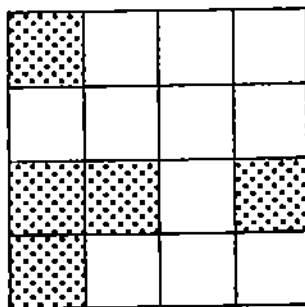


Figure 1. A map;  $n = 16$ .

Related problems have been studied. In the context of the Tile Salvage Problem, Fowler, Paterson and Tanimoto [7] show that the  $3 \times 3$  case is NP-complete. (They left open the question for smaller rectangles.) They also found closely related covering problems NP-complete as has Masek [8]. Partitioning problems have also been considered and these have shown a remarkable variety [9-11]. See also Johnson [12].

#### The $1 \times 2$ Case

The selection of the maximum number of adjacent pairs of tiles, i.e., the  $1 \times 2$  Tile Salvage Problem, can be accomplished in polynomial time. To prove this, label the tiles in a checkerboard pattern of alternate  $A$ 's and  $B$ 's. By interpreting the problem as an instance of the "marriage" problem (2-dimensional matching), we can obtain a polynomial solution [5]. In particular, let the  $A$  tiles in the map correspond to "girls" and the  $B$  tiles correspond to "boys" in the 2-dimensional matching problem. The maximum set of girl-boy pairs gives the maximum set of mutually non-intersecting  $1 \times 2$  rectangles. See Figure 2.

#### The $2 \times 2$ Case

The complexity of the problem changes dramatically if we consider circuits which can be divided into four mutually adjacent tiles. Consider

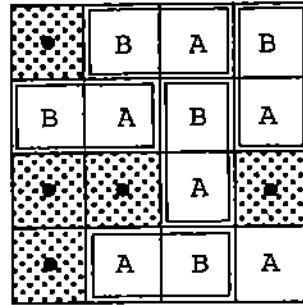


Figure 2. An optimal selection for the  $1 \times 2$  Tile Salvage Problem. maps with alternating rows of two types: one type with alternating  $A$  and  $B$  dice and one type with alternating  $C$  and  $D$  dice. See Figure 3.

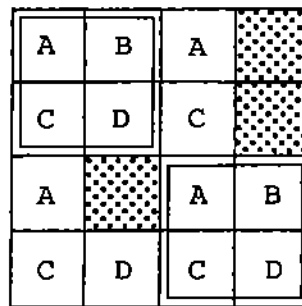


Figure 3. Checkerboard Pattern with  $A, B, C, D$  Die Types.

Each die can participate as a component in four different  $2 \times 2$  tiles. The problem, as before, is to find the maximum number of non-overlapping  $2 \times 2$  tiled rectangles given a map. In this section we show that the  $2 \times 2$  Tile Salvage Problem is NP-complete. It is interesting to notice that in both the  $1 \times 2$  case and the  $2 \times 2$  case the number of rectangles a single die can participate in is the same, although the complexity of the two problems is quite different.

Let the Tile Salvage Predicate be defined as the statement

*"Given a map and an integer  $z$ , are there at least  $z$  non-overlapping  $2 \times 2$  tiled rectangles in the map?"*

Clearly the problem is in NP. To complete the argument, our strategy

will be to show that the NP-complete problem Planar 3-SAT [6] can be reduced to the Tile Salvage Predicate in polynomial time.

Planar 3-SAT can be defined as follows: Let  $B$  be a formula with clauses  $c_1, \dots, c_m$  and variables  $v_1, \dots, v_n$ . Assume that  $B$  is in conjunctive normal form with at most 3 literals per clause. Define  $G(B)$  to be the undirected graph with vertices  $V$  and edges  $E$  where

$$V = \{v_1, \dots, v_n\} \cup \{c_1, \dots, c_m\}$$

$$E = \{\{v_i, v_{i+1}\} \mid 1 \leq i < n\} \cup \{\{v_n, v_1\}\} \cup$$

$$\{\{c_i, v_j\} \mid \text{if } v_j \text{ or } -v_j \text{ occurs in } c_i\}.$$

Lichtenstein defines Planar 3-SAT to be the statement

*"Given a formula  $B$  in conjunctive normal form with at most 3 literals per clause and for which  $G(B)$  is planar, is there an assignment of values to variables for which  $B$  has the value true?"*

Lichtenstein has shown that Planar 3-SAT is NP-Complete [6].

It will be convenient to adopt a new representation for maps and new vocabulary. For the remainder of the paper, a map will be a graph formed by placing a vertex at the center of each  $(1 \times 1)$  tile and connecting it to any vertices that are a unit distance away (see Figure 4). It is easy to see that the  $2 \times 2$  Tile Salvage Problem on the graph form of the map is simply that of selecting the maximum number of non-intersecting unit square regions. For brevity, we will hereafter call these unit squares "tiles," and understand that they actually represent  $2 \times 2$  rectangles.

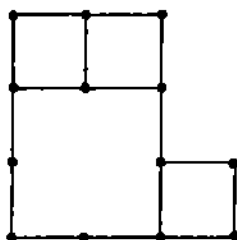


Figure 4. Alternate map representation based on Figure 3.

**Theorem:** *The  $2 \times 2$  Tile Salvage Problem is NP-complete.*

**Proof:** It is easy to show that the  $2 \times 2$  Tile Salvage Problem is in NP. For the reduction, we will take an arbitrary instance  $B$  of Planar 3-SAT and show that there is a satisfying assignment of truth values to  $B$  if and only if there are at least  $z$  non-intersecting  $2 \times 2$  rectangles on a map  $C_B$ . The map  $C_B$  and parameter  $z$  will depend upon the structure of  $B$ .

By definition,  $G(B)$  is planar and of a special form: There are nodes representing both variables and clauses, edges linking variables, and edges linking variables and clauses. We will represent the variables in  $G(B)$  by constructs called *generators*, the clauses in  $G(B)$  by constructs called *receptors*, and edges in  $G(B)$  by constructs called *transmission lines* in the map  $C_B$ .

To construct the generator for a node  $v_i$ , consider the graph  $G(B)$ .  $v_i$  has edges to  $v_{i-1}$ ,  $v_{i+1}$  and to some set of clauses  $c_{i_1}, \dots, c_{i_k}$ . Construct the following generator to represent  $v_i$  in  $C_B$ : The ring of the generator is a rectangular structure without corners of  $2g_i$  tiles. Adjacent transmission lines should be separated by at least two tiles. The transmission lines corresponding to consecutive positive or consecutive negative occurrences of  $v_i$  should be separated by an odd number of tiles. The transmission lines corresponding to consecutive complementary occurrences (a positive occurrence and a negative occurrence) should be



separated by an even number of tiles. The transmission lines linking the generator for  $v_i$  with the generators for  $v_{i-1}$  and  $v_{i+1}$  should each have an odd number of tiles. See Figure 5.

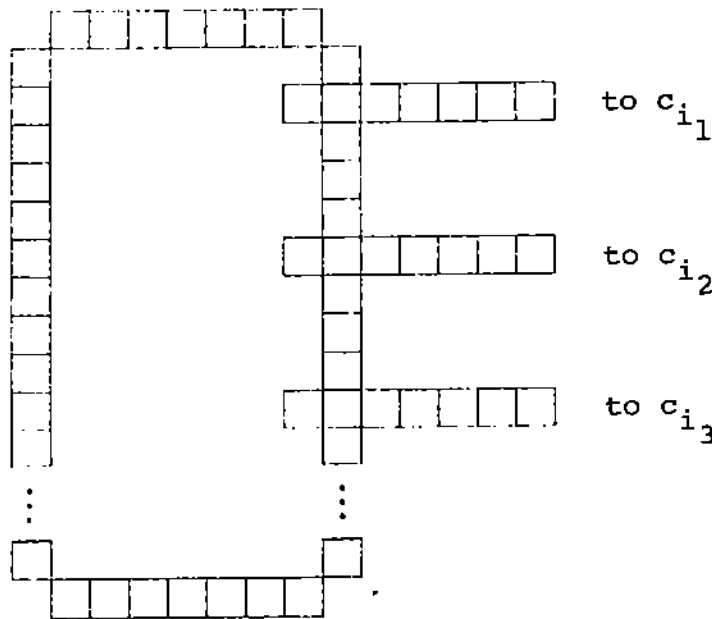


Figure 5. Generator for  $v_i$  Transmission Lines.

To construct the receptor for the clause  $c_i$ , we observe that in  $G(B)$ , the node representing  $c_i$  has degree between 1 and 3. Let  $v_{i_1}, v_{i_2}, v_{i_3}$  be the literals in clause  $c_i$ . Construct the following receptor to represent  $c_i$ .

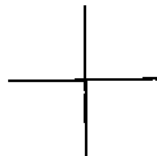


Figure 6. Receptor for  $c_i$  without transmission lines.

The transmission lines from  $v_{i_1}, v_{i_2}, v_{i_3}$  each fit into a corner of the receptor (in no particular order) as follows:

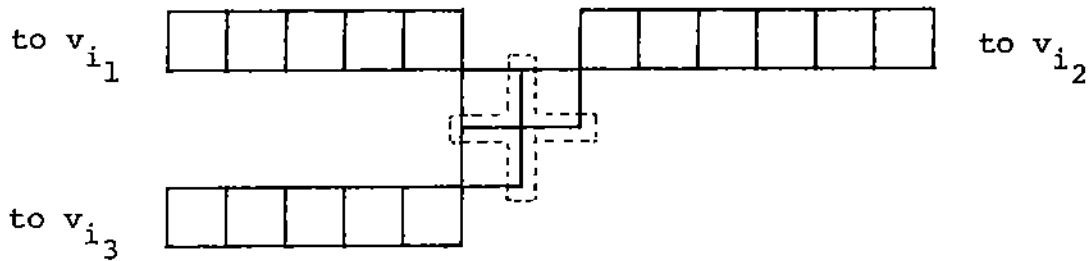
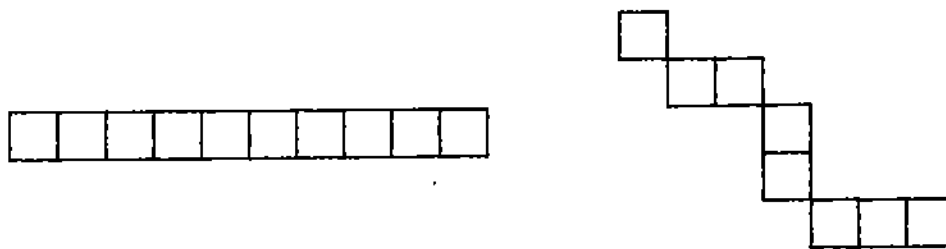


Figure 7. Receptor with Transmission Lines to  $v_{i_1}, v_{i_2}, v_{i_3}$ .

The edges connecting the variables and linking the variables and clauses in  $G(B)$  will be represented by transmission lines in the map. A horizontal transmission line is essentially a row of tiles. These lines may be bent vertically or diagonally on the lattice as shown in Figure 8.



Horizontal Transmission Line    Non-Horizontal Transmission Line

Figure 8.

We define the length of a transmission line to be the number of tiles which compose the transmission line. We will always count the tiles on a transmission line from the *innermost* tile emanating from the generator.

This completes the translation of the components of the graph  $G(B)$  to portions of the map  $C_B$ .  $C_B$  can now be constructed from the above specifications with the following additional requirements:

- 1) Transmission lines between generators should be composed of an odd number of tiles.
- 2) Transmission lines between generators and receptors should be

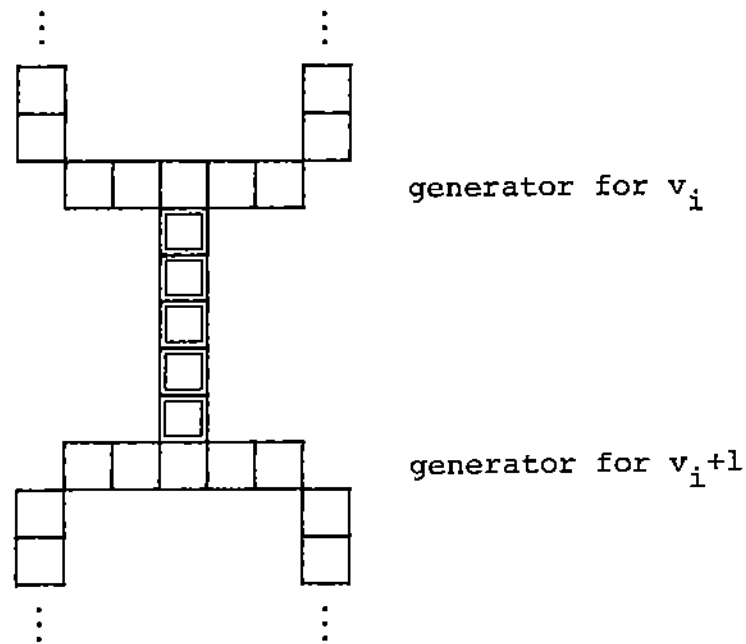


Figure 9. Transmission Line (Highlighted) Between the Generators for  $v_i$  and  $v_{i+1}$ .

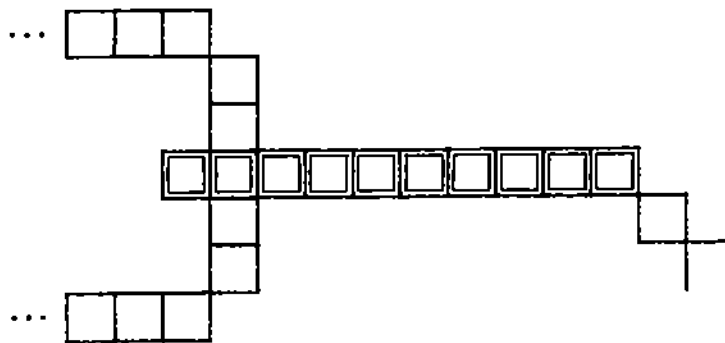


Figure 10. Transmission Line (Highlighted) from a Generator to a Receptor with 10 Tiles.

composed of an even number of tiles. This does **not** include the tile incident to the middle node of the receptor.

- 3) Non-intersecting components of the map should always be separated by at least two tiles.
- 4) The construction can proceed without regard to parity constraints. Once the diagram has been constructed, the parity of

any line segment can be set by first refining the grid, say by a factor of 10. If the line segment has the right parity, we are done. Otherwise, by the dilation, there is a straight line segment of length at least seven tiles that can be replaced by the sequence given in Figure 11. This changes the parity of the number of tiles and because of the dilation, preserves the tile separation constraints.

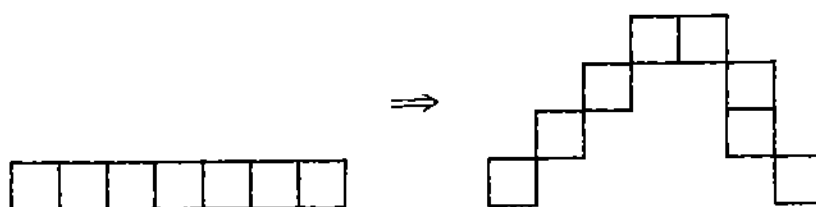


Figure 11. Dilation to Change the Parity of a Line.

As an example of our construction, let  $B$  be the formula  $(v_1 + v_2 + v_3)(-v_1 + v_4 + v_5)$ . One way of constructing  $G(B)$  is shown in Figure 12.

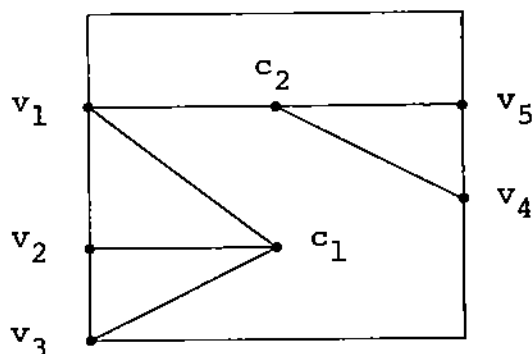


Figure 12.  $G(B)$ .

Clearly  $G(B)$  is planar. Figure 13 shows the map  $C_B$  produced using the preceding construction. Note the parity fix used in the transmission line from the generator for  $v_3$  to the receptor for  $c_1$ .

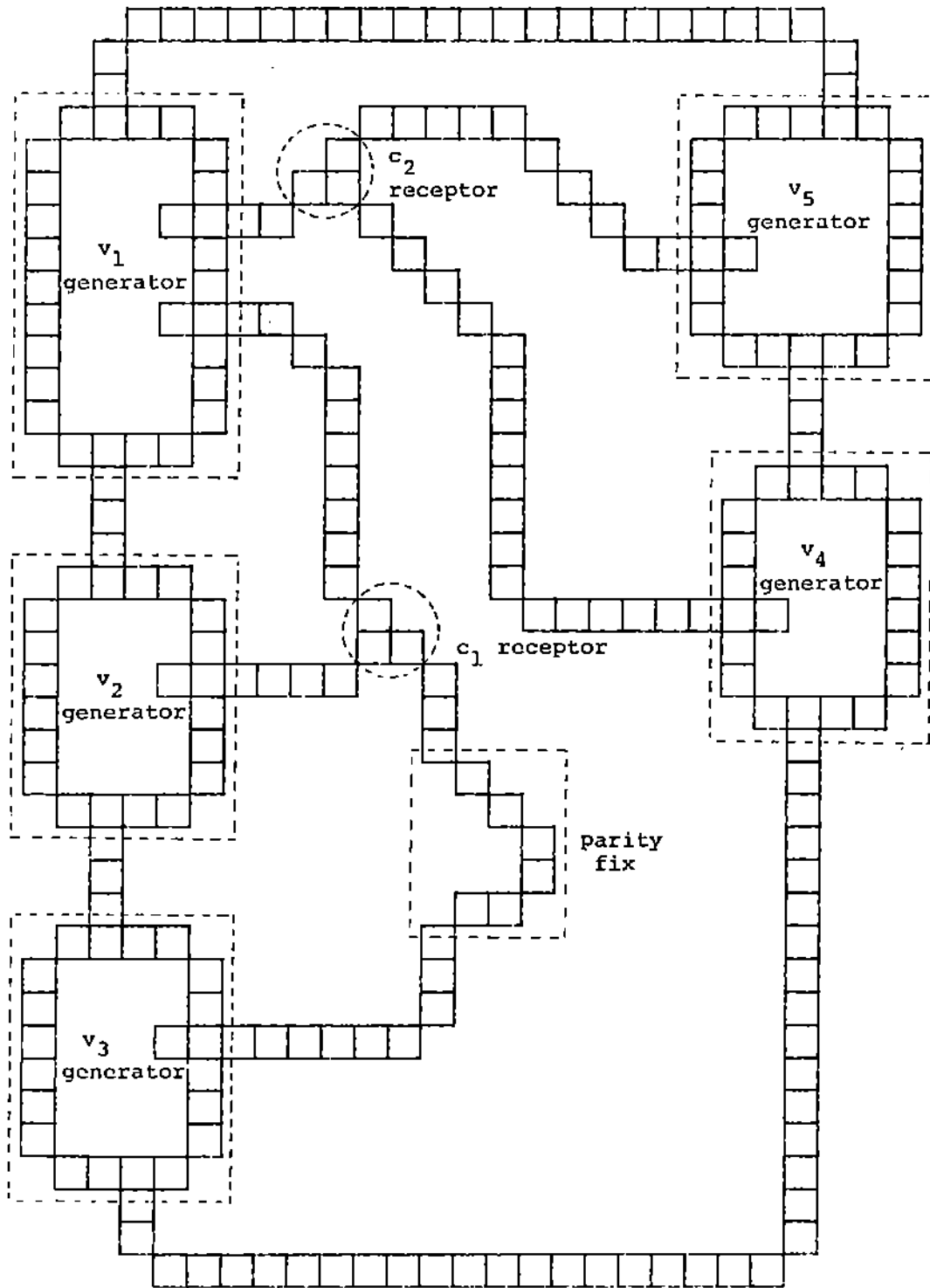


Figure 13. The map  $C_B$  for  $G(B)$ .

Let  $C_B$  be the map composed of the generators, receptors and transmission lines which represent the nodes and edges of  $G(B)$ . Clearly the construction of  $C_B$  from  $G(B)$  can be accomplished in polynomial time. To determine the parameter  $z$ , let  $\{v_1, \dots, v_n\}$  be the variables occurring in  $B$  and  $\{c_1, \dots, c_m\}$  be the clauses. By construction, for each  $i$ , the ring of the generator for  $v_i$  is composed of  $2g_i$  tiles. Let  $2t_i + 1$  be the number of tiles in the transmission line linking the generator for  $v_i$  with the generator for  $v_{i+1}$  ( $2t_n + 1$  will be the number of tiles in the line between the  $v_n$  generator and the  $v_1$  generator). Similarly, let  $2t$  represent the *total* number of tiles in all transmission lines linking a generator and a receptor. Say there are  $y$  of these.

Given the parameters described above, we let  $z = \sum_{i=1}^n g_i + \sum_{i=1}^n t_i + t + m - y$ . Recall that if  $2g_i$  gives the number of tiles which are in the generator for  $v_i$  and  $2t$  gives the number of tiles in all transmission lines between generators and receptors, then subtracting  $y$  ensures that we do not count twice the tiles which are both in a transmission line and on the ring of some generator. Recall also that  $m$  is the number of receptors (clauses) each of which will contribute one tile if and only if  $B$  is satisfiable. For example, for the formula  $B$  with map  $C_B$  given in Figure 13,  $z=120$ . An assignment of non-overlapping tiles to  $C_B$  is shown in Figure 14.

To review, given formula  $B$  in 3CNF and planar  $G(B)$ , we have shown how to construct the map  $C_B$  and determine the parameter  $z$  in polynomial time. To complete our reduction, we must show that there are at least  $z$  non-intersecting tiles in  $C_B$  iff there is a satisfying assignment of truth values to  $\{v_1, \dots, v_n\}$  in  $B$ .

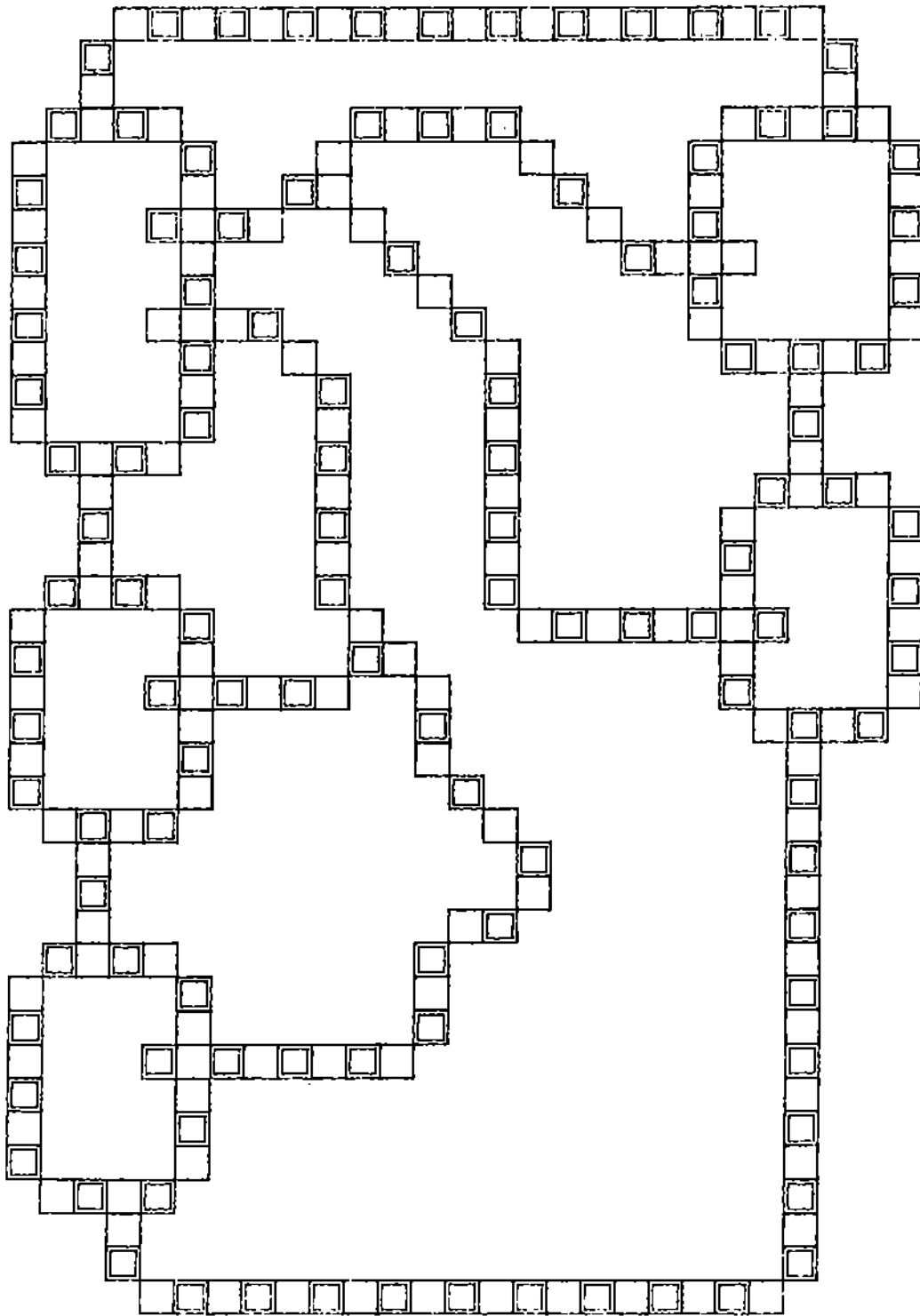


Figure 14. Non-Intersecting Tiles on the Map  $C_D$ .

For the first direction, assume that there are  $z$  non-intersecting tiles in the map  $C_B$ . Then by construction, for each  $j$ , the receptor for  $c_j$  contains a transmission line whose assignment of non-intersecting tiles includes a tile incident to the middle node of  $c_j$ .

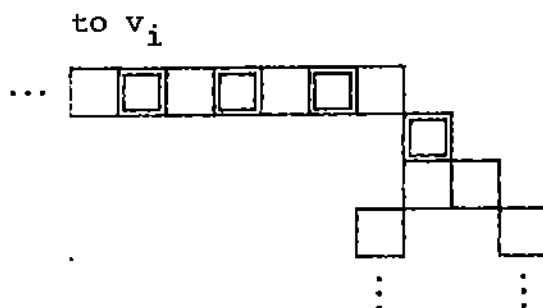


Figure 15. Receptor for  $c_j$  in  $S$ .

Let  $v_i$  be the variable generator from which this transmission line emanates. Then assign the literal corresponding to  $v_i$  in  $c_j$  with the value true.

Do this for every receptor (clause). Note that if there is an assignment of values to  $\{v_1, \dots, v_n\}$  consistent with this assignment then  $B$  will be satisfiable. We will show that we can find at least  $z$  non-intersecting tiles only if it is impossible for literals  $v_i$  and  $\neg v_i$  both to be assigned the value true by the given procedure.

Assume without loss of generality that  $v_i$  appears in  $c_1$  and  $\neg v_i$  appears in  $c_2$ . Then the  $v_i$  generator has the form given in Figure 16

where the ring has  $2g_i$  tiles. Assume towards a contradiction that there are  $z$  non-intersecting tiles in  $C_B$  with the corresponding assignment of true to both  $v_i$  and  $\neg v_i$ . Then the assignment of non-intersecting tiles to both the transmission lines to  $c_1$  and  $c_2$  are incident to the middle nodes of those receptors. Hence without loss of generality, the generator for  $v_i$  can be assigned non-intersecting tiles as in Figure 17.



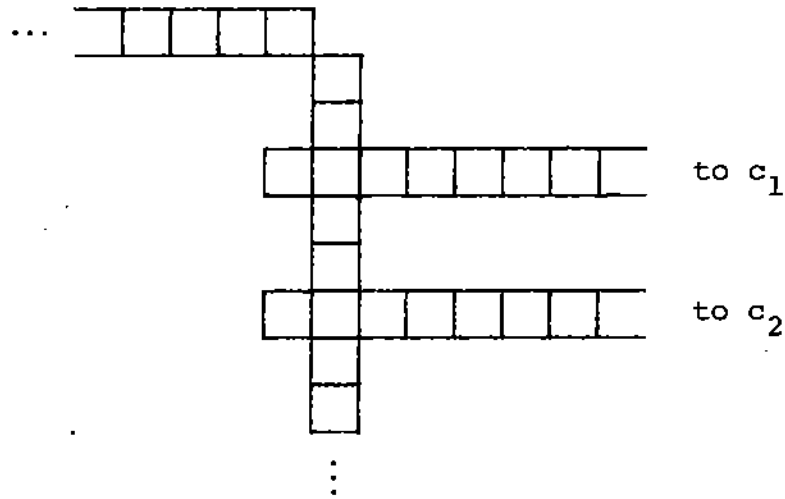


Figure 16. The Generator for  $v_i$ .

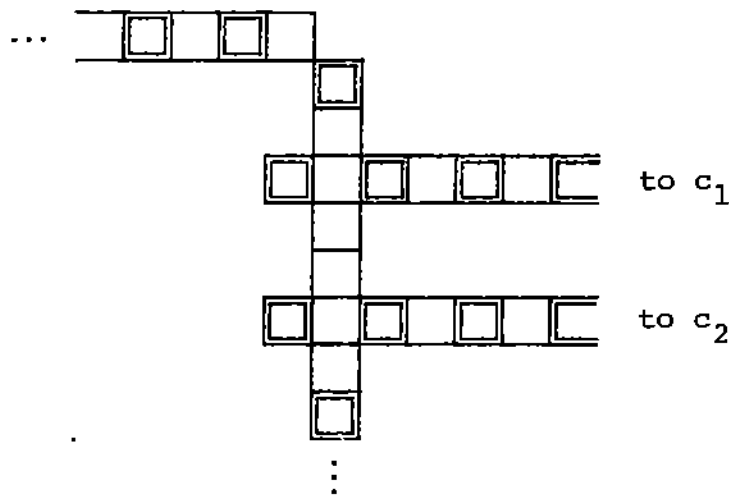
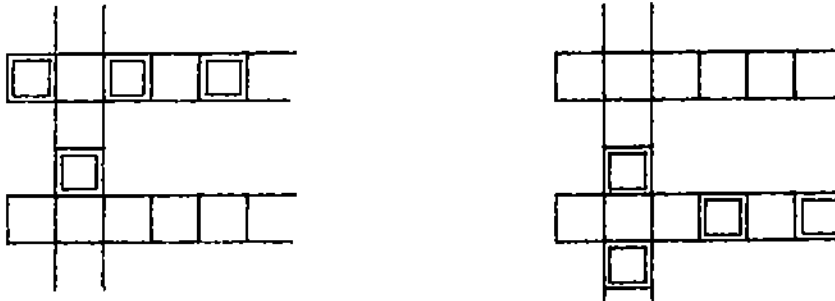


Figure 17. Assignment of Tiles to the  $v_i$  generator.

If the transmission lines are completely assigned with non-intersecting tiles, the generator ring can be assigned with at most  $g_i - 3$  non-intersecting tiles. Note that if all other transmission lines and receptors are completely assigned with non-intersecting tiles, we can still only achieve a count of  $z - 1$  tiles contradicting our hypothesis. In particular, unless we shift the non-intersecting tiles towards their receptor three places on one of the transmission lines *and* their receptor can preserve

its contribution of one tile (incident to the middle node but from another transmission line), we will not be able to achieve an assignment of  $z$  non-intersecting tiles in the map. Hence if there is an assignment of  $z$  non-intersecting tiles in  $C_p$ , we can construct a consistent assignment of values to variables so that  $B$  will be satisfiable.

For the other direction, assume that the formula  $B$  is satisfiable. Then there is an assignment of values to variables so that in each clause, there is at least one literal with the value true. Consider the generator for variable  $v_i$ . Assume without loss of generality that  $v_i$  has been assigned the value true by the satisfying assignment. (The argument is analogous if  $\neg v_i$  is assigned the value true). Mark the innermost tile of each transmission line in the  $v_i$  generator which corresponds to  $v_i$ . For the complementary literal  $\neg v_i$ , mark the first tile incident to the innermost three tiles of each transmission line in the  $v_i$  generator. In addition, mark the tiles in the generator ring adjacent to all transmission lines corresponding to the literal  $\neg v_i$ . See Figure 18.



Assignment for a True Literal

Assignment for a False Literal

Figure 18.

Propagate this assignment of non-intersecting tiles around the generator ring until the ring is filled as completely as possible. Since complementary occurrences of the variable are separated by an even number

of tiles and same-value occurrences of the variable are separated by an odd number of tiles, this assignment will contribute  $g_i - x$  tiles where  $x$  is the number of transmission lines corresponding to a true literal. Next, fill the transmission lines from the generator as completely as possible to the receptor with an assignment of non-intersecting tiles. Leave unmarked all tiles in each receptor incident to the middle node.

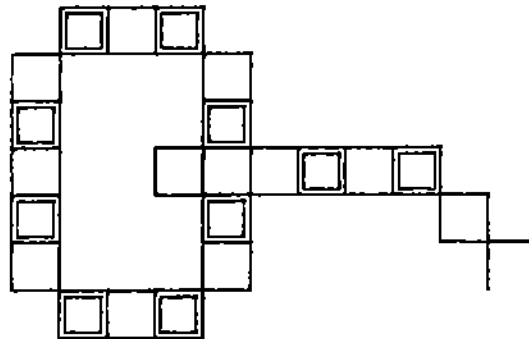


Figure 19. Assignment of Tiles to Generator and Transmission Lines for a false literal.

Assign tiles to the transmission lines linking the generators as completely as possible.

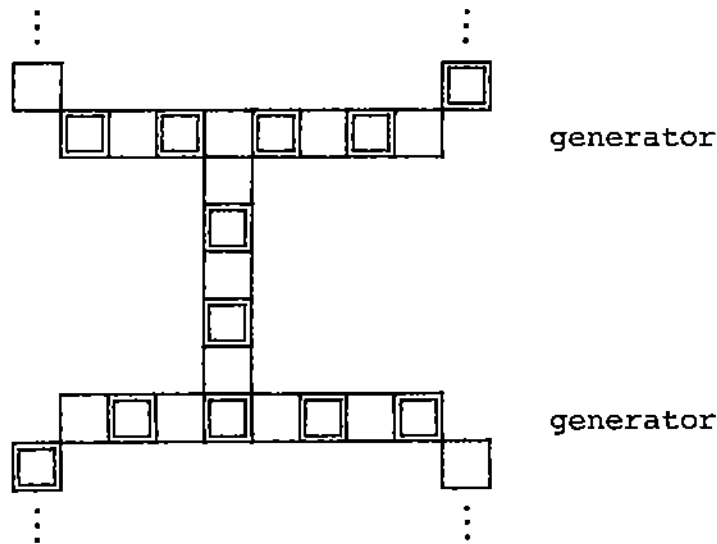


Figure 20. Assignment of Tiles to Lines Between Generators.

The assignment is now complete except for the receptors. Since we were given a satisfying assignment to the formula  $B$ , there is at least one literal  $v^*$  in each clause  $c_i$  which has been assigned the value true. By construction, the transmission line for  $v^*$  has an assignment of non-intersecting tiles which has been propagated from the innermost tile of its generator to the receptor for  $c_i$ . Since the length of  $v^*$ 's transmission line from the innermost tile to the receptor (not including the tile in the receptor incident to the middle node) is even, the last tile in the transmission line is unmarked. Hence we can mark the tile in the receptor incident to both the middle node and the transmission line for  $v^*$  adding one tile from  $c_i$  to the assignment of non-intersecting tiles on the map. Do so for every receptor (clause).

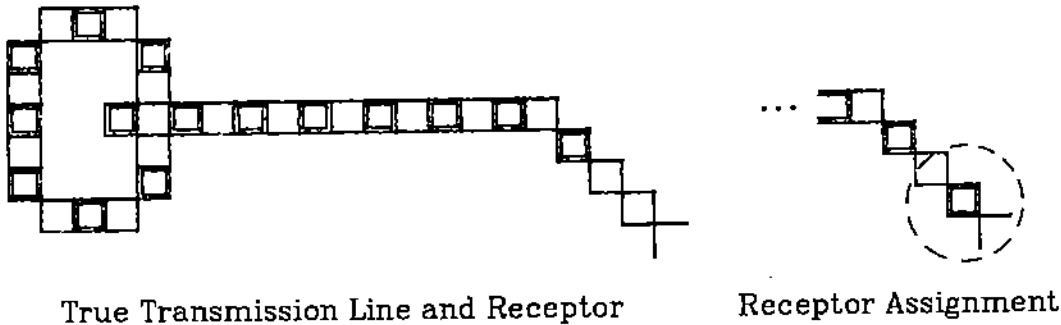


Figure 21.

By construction, we have created an assignment of  $z$  non-intersecting tiles in  $C_B$  from the satisfying assignment of values to variables in  $B$ . Hence the  $2 \times 2$  Tile Salvage Problem is NP-Complete. ■

### An Approximation Algorithm

In this section we show that a greedy algorithm serves as a linear time approximation algorithm for the  $2 \times 2$  Tile Salvage Problem. (N.B. We continue to use our "map" and "tile" notions.)

Given a map, the algorithm is simply stated: Repeatedly select the leftmost, uppermost tile, and remove it and its incident edges from the map. See Figure 22. The algorithm terminates when the map is empty or has become a collection of disconnected line segments, and obviously requires only a linear amount of time.

In order to assess the algorithm's effectiveness, let  $OPT_M$  be the solution for a given map  $M$  of the  $2 \times 2$  Tile Salvage problem.

**Theorem:** For a given map  $M$ , the algorithm finds a solution that is at least  $\frac{1}{2}OPT_M$ .

**Proof:** Let  $M$  be a map and apply the algorithm. The algorithm induces a partitioning of the tiles of the map as follows: Each subset of tiles contains a selected tile and those tiles having as a side one of the removed incident edges. (See Figure 23.)

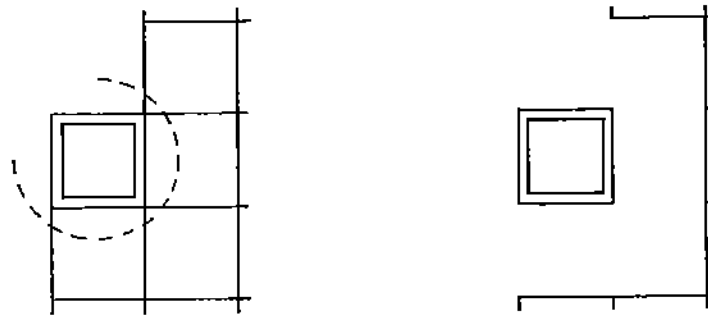


Figure 22. Application of one step of the algorithm.

By the leftmost, uppermost condition, selections will be performed in neighborhoods that are subgraphs of Figure 23(a), although there may also be some extraneous edges. Thus each subset of the partitioning will contain at most five tiles.

Consider an optimal solution for  $M$  of the  $2 \times 2$  Tile Salvage Problem and mark a satisfying set of  $OPT_M$  tiles. Because of the number and configuration of tiles in each subset from the partitioning induced by the

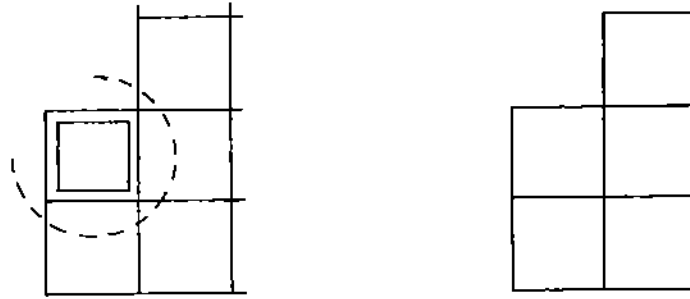
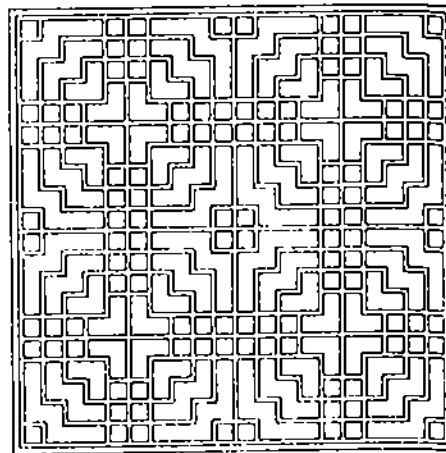


Figure 23. Subset induced by a tile selection.

algorithm, no subset contains more than two marked tiles. Since the greedy algorithm selected one tile each subset, its result can be no worse than  $\frac{1}{2}OPT_M$ . ■



East gate suburb, Chengtu, Szechwan, 1875 A.D.

## REFERENCES

1. Arthur B. Glaser and Gerald E. Subak-Sharpe  
*Integrated Circuit Engineering*  
Addison-Wesley, 1979
2. Carver Mead and Lynn Conway  
*Introduction to VLSI Systems*  
Addison-Wesley, 1980
3. Kye S. Hedlund and Lawrence Snyder  
*Model for Wafer Scale Testing*  
Purdue University *Technical Report TR-389*, 1981

4. Kevin Smith  
Water Prepared to Turn Itself into a Computer  
In *Electronics*, September 22, 1981, pp. 73-74
5. John E. Hopcroft and Richard M. Karp  
An  $n^{\frac{5}{2}}$  Algorithm for Maximum Matching in Bipartite Graphs  
*SIAM Journal of Computing*, 2(4): 225-231, 1973
6. David Lichtenstein  
Planar Formulae and their Uses  
*SIAM Journal of Computing*, 11(2): 329-343, 1982
7. Robert J. Fowler, Michael S. Paterson and Steven L. Tanimoto  
Optimal Packing and Covering the the Plane are NP-Complete  
*Information Processing Letters*, 12(3): 133-137, 1981
8. William J. Masek  
Some NP-Complete Set Covering Problems  
Typescript, Massachusetts Institute of Technology, 1978
9. Andrzej Lingas, Ron Y. Pinter, Ronald L. Rivest and Adi Shamir  
Minimum Edge Length Decomposition of Rectilinear polygons  
Typescript, Massachusetts Institute of Technology, 1981
10. Andrzej Lingas  
The Power of Non-rectilinear Holes  
Typescript, Massachusetts Institute of Technology, 1981
11. Andrzej Lingas  
Heuristics for Minimum Edge Length Rectilinear Partitions  
Typescript, Massachusetts Institute of Technology, 1981
12. David S. Johnson  
The NP-Completeness Column: An Ongoing Guide  
*Journal of Algorithms*, 3(2): 182-195, 1982