Manhattan and Rectilinear Wiring*

Raghunath Raghavan, James Cohoon, and Sartaj Sahni
University of Minnesota
Minneapolis, MN 55455

Abstract

The problem of wiring pairs of points in Manhattan and rectilinear fashion is considered. We develop an $O(n^2)$ algorithm to determine whether or not a set of n point pairs can be wired in Manhattan fashion on a single layer. When this is possible, our algorithm generates the layout. We show that determining the maximum number of point pairs that can be wired in Manhattan fashion is NP-hard. The problem of determining the minimum number of layers needed to wire a set of n point pairs is also NP-hard. For the rectilinear wiring problem, we show that determining whether or not a set of n point pairs is wireable on a given grid is NP-complete.

Keywords and phrases

Manhattan and rectilinear wiring; design automation; complexity; NP-hard; NP-complete.

1. THE MANHATTAN AND RECTILINEAR WIRING CONSTRAINTS

Wire routing (equivalently, *wiring*, *wire layout*, etc.) is the problem of defining precisely the conductor paths in a given *wiring medium* for a set of *wires*. A wire is a specification of a pair of points to be connected.*

In the most general wire routing case, the conductor paths can be curves. In fact, it is not unusual to find curved conductor paths in manual layouts. However, in automated systems, conductor paths are usually constrained to contain horizontal and vertical segments only. The resulting loss in generality is offset by the following advantages:

(1) The conductor paths can now be represented far more easily. General curves, on the other hand, are not amenable to compact representation on a computer.

(2) The length calculations are considerably simplified.

(3) Fabrication equipment such as plotters are usually capable of horizontal and vertical movements only. Curves and diagonal lines are simulated by a series of small, alternating horizontal and vertical movements, and take a long time to draw.

Wiring where each path is made up of horizontal and vertical segments only is called *rectilinear wiring*. It is often thought of as wiring on a grid. All the wire end points are grid points; each conductor path is constrained to lie along grid lines only; and conductor paths are not allowed to cross. The separation between grid lines reflects the minimum conductor separation necessary to avoid inductive cross-talk. Thus, the concept of wiring on a grid neatly captures a number of practical constraints inherent in wire routing on a single layer. Figure 1.1 shows an example of wiring on a grid.

Any point on a grid may be specified by its x- and y-coordinates. Assuming that consecutive grid lines are one unit apart, the coordinates of grid points are always natural numbers. Let S = $\{(u_i,v_i,x_i,y_i)|1\leq i\leq n\}$ be a set of n wires to be connected. Wire i is specified by requiring that the point (u_i, v_i) is to be connected to the point (x_i, y_i), $1 \leq i \leq n$. S is called a *wire set*. For the example of Figure 1.1, S = {(2,3,6,3),(3,2,5,5),(2,4,3,4),(3,4,1,5)}.

With the constraint that conductor paths are not allowed to cross, it should be clear that not

---

* This research was supported in part by the National Science Foundation under grant MCS 80-005856.

*With this terminology, the context of usage ensures that, in our discussion, there is no confusion between the problem specification (i.e., the wire) and the problem solution (i.e., the wiring or the wire layout).

Another point is that in general, a *signal net* (a set of points to be interconnected) may be specified, rather than a wire. However, in fast circuit technologies (e.g., ECL), transmission line considerations impose rigid restrictions on the topology of the interconnection tree. Specifically, Steiner trees may not be permitted. In such cases, the nets are effectively decomposed into wires (before wire routing even begins) in ways that guarantee that all restrictions will be satisfied. We shall assume such an environment.

Figure 1.1 Wiring on a grid

all wire sets S can be wired in a rectilinear fashion on a single layer. Hence, we need to consider multilayered wiring media. A multilayered wiring medium has k layers on which wires can be run. Typically, the points to be connected are the terminals of modules located on or in the wiring surface. It is assumed that these wire end points are available on all the layers. Interlayer connections are accomplished by means of holes called *vias*. There may be rules that restrict just where vias may be located. Each layer on which wiring is permitted is treated as a grid, and the same restrictions apply as before.

Usually, the cost of a realization goes up sharply with the number of layers used for the wiring. So it is generally desirable to minimize the number of layers used. Also, engineering considerations might limit the number of layers available to do the wiring. For example, in printed circuit boards with more than 10 or 12 layers, vias tend to be unreliable. As another example, integrated circuits can often have just 2 or 3 layers of metallization.

There are a number of applications where it is necessary to limit the number of bends on each conductor path. For example, when fabricating microwave or millimetric wave integrated circuits, the conductors, called *microstrip lines*, function as transmission lines. Bends on a microstrip line result in reflections, and this reduces the transmission efficiency of the line. Thus, it is desirable to limit the number of bends on each microstrip line.

So, a wiring problem worth considering is that of wiring S on a grid subject to the added constraint that no wire may bend more than k times. This problem has been studied, for example,

by [POME65]. In the wiring of Figure 1.1, the wire joining (2,3) and (6,3) has two bends. When the wiring is restricted such that no wire has more than one bend, it is called a *Manhattan wiring*[*].

Two points (u,v) and (x,y) can be wired with no bends if and only if either u = x or v = y. If u ≠ x and v ≠ y, then there are exactly two ways to wire this pair using only one bend. These two ways are shown in Figure 1.2, and are respectively called the *upper* and *lower* wirings.

Figure 1.2 Possible Manhattan wirings

In Section 2, we show that it is possible to efficiently determine whether S is Manhattan wireable on one layer. Our algorithm has time complexity $O(n^2)$, where n = |S|. For wire sets S that are Manhattan wireable on one layer, our algorithm also obtains the layout in the stated time bound.

For those wire sets that cannot be Manhattan wired on one layer, the following questions are of interest:

(1) What is the minimum number of layers needed to Manhattan wire S?

This is also called the layering problem. Minimizing the number of layers needed reduces the cost of the realization, as discussed earlier.

(2) What is the maximal subset of S that can be Manhattan wired on one layer?

---

*The term "Manhattan wiring" does not appear to have acquired a single, universally accepted meaning. Sometimes, it is used to denote the same thing as what we term "rectilinear wiring". At other times, it is used to describe multilayer wiring where all the horizontal segments are confined to one set of layers, and all the vertical segments to the remaining layers. While recognizing that there may be other interpretations of the term "Manhattan wiring", we shall use it to denote wiring where the layout of each wire is restricted to contain no more than one bend.

It is reasonable to expect that confining the layout of each wire to any single layer and maximizing the number of wires laid out on any given layer will help reduce the number of feedthroughs required to other layers. Since feedthroughs reduce the reliability of the system, their use is to be avoided wherever possible. Hence, the relevance of this question.

In Section 3, we show that determining the answer to question (2) is NP-hard. In Section 4, we show that determining whether S is Manhattan wireable on two layers is NP-complete. This implies that determining the answer to question (1) is NP-hard. Note that the result obtained in Section 3 rules out the most obvious greedy approach to the layering problem (i.e., put as many wires as possible on the first layer; put as many of the remaining wires as possible on the second layer; etc.).

Then, in Section 5, we tackle the problem of determining the minimum number of layers needed to wire a given wire set S in rectilinear fashion. We show that this problem is NP-hard. In fact, we have established a stronger result, namely that determining whether S can be wired in rectilinear fashion on one layer constitutes an NP-complete problem.

2. THE FEASIBILITY OF MANHATTAN WIRING ON ONE LAYER

In this section, we outline how to determine whether a given wire set S can be Manhattan wired feasibly (i.e., without intersections) on one layer. In general, there are two ways to lay out a wire in Manhattan fashion, as shown in Figure 2.1. However, the presence of other wires may preclude one or even both these layouts (see Figure 2.2).

Figure 2.1 Manhattan layouts

Figure 2.2

Let T S be the set of wires for which there is only one feasible layout. The wires in T are *forced*, i.e., their layouts are the same in all feasible layouts for S. Figure 2.3 shows how forced wires can arise.

Now, let (a,b,c,d) and (e,f,g,h) be two wires in S. Any one of the following relations may

Figure 2.3 Forced wires

hold between these two wires:

(1) The two wires cannot be laid out feasibly on the same layer. See Figure 2.4. Clearly, the wiring is infeasible.

Figure 2.4 Infeasibility

(2) Either layout for each wire does not directly influence the layout for the other. See Figure 2.5. We denote this as (a,b,c,d)**I**(e,f,g,h) (i.e., they are pairwise *i*ndependent).

(3) The upper layout for (a,b,c,d) forces the (upper) layout for (e,f,g,h); however, the lower layout for (a,b,c,d) does not influence the layout for (e,f,g,h) in any way. This situation is depicted in Figure 2.6. We denote this situation as (a,b,c,d)**U**(e,f,g,h).

(4) There is the analogous situaion where only the lower layout for (a,b,c,d) forces the (lower)

Figure 2.5 The relation I

Figure 2.6 The relation U

layout for (e,f,g,h). This situation is denoted as (a,b,c,d)**L**(e,f,g,h).

(5) Finally, there is the situation where the upper (lower) layout for (a,b,c,d) forces the lower (upper) layout for (e,f,g,h). This situation is depicted in Figure 2.7. In this case, we say that the two wires are pairwise dependent, and write it as (a,b,c,d)**D**(e,f,g,h).

We note that the binary relations **D** and **I** are symmetric, while *L* and *U* are not. Also,

(a,b,c,d)**L**(e,f,g,h) (e,f,g,h)**U**(a,b,c,d)

(Some of these conflict types have been considered earlier, but in a totally different context [HIGH73]. In the latter work, the various conflict types have been used to decide the order in

Figure 2.7 The relation D

which connections should be attempted during one-at-a-time wire routing.. There are no restrictions on the wire routing, such as requiring at most one bend per wire layout. Thus, there is no connection at all between our work and that in [HIGH73].)

Having encapsulated all the possible wire pair interactions in these four binary relations **L**, **U**, **D**, and **I**, we now proceed to the wiring algorithm. In the *first phase* of the algorithm, we consider all n(n-1)/2 (where n = |S|) pairs of wires in S. For each pair (A,B), we determine the appropriate interaction relation (i.e., A**D**B, A**I**B, A**U**B, A**L**B, A forced, B forced, infeasible) as described earlier. If it is determined that the wires A and B cannot be wired feasibly , the algorithm terminates. Also, during this phase, certain wires are identified as forced.

In the second phase, this constraint information can be encoded into a Boolean expression, which has the following properties:

(i) it will be in normal form (CNF).

(ii) it will contain at most 2 literals per clause.

(iii) it will be satisfiable if and only if the corresponding wire set can be laid out (without intersections) on one layer in Manhattan fashion.

The encoding is done as follows. One Boolean variable is associated with each wire. The truth value assigned to it corresponds to the layout of the wire, say true for the upper layout and false for the lower layout. Let x_A be the Boolean variable corresponding to wire A, for each wire A in the wire set. Each interaction relation can be turned into a Boolean formula as follows:

A**D**B becomes $(x_A :> \neg x_B) \Lambda (x_B :> \neg x_A)$.

A**I**B is not encoded.

A**U**B becomes simply (x_A :> x_B).

A**L**B becomes ($\neg x_A$:> $\neg x_B$).

For each forced wire A, if the upper layout of A is forced, this fact is encoded as x_A; if the lower layout of A is forced, this is encoded as $\neg x_A$.

The rationale for this encoding scheme is quite easy to see. When all the implications in the encodings are substituted out, the result is a CNF encoding of each interaction relation, with at most two literals per clause and at most two clauses per relation.

The entire set of n(n-1)/2 interaction relations is encoded by *and*ing all the individual encodings. This gives a CNF Boolean expression containing at most O(n^2) clauses, with at most 2 literals per clause. Since the 2SAT problem can be solved in linear time (see [EVEN76]), the existence of a satisfying truth assignment for the encoding can be determined in O(n^2) time.

If a satisfying truth assignment can be found, it is a trivial task to construct the equivalent wire layout for the wire set.

3. MAXIMIZING WIRE LAYOUT UNDER THE MANHATTAN CONSTRAINT

Let S be a wire set. In this section, we consider the problem of determining a maximum cardinality subset T of S (T

S) such that all the wires in T can be laid out in Manhattan fashion on one layer. We call this problem MAXWIRE.

While we are able to determine if there exists a Manhattan layout for S in polynomial time, it appears that a similarly efficient algorithm for MAXWIRE does not exist. In fact, the MAXWIRE problem is NP-hard and the corresponding decision problem is NP-complete. Since MAXWIRE is NP-hard, the existence of a polynomial time algorithm for this problem would imply the existence of polynomial time algorithms for several other problems (e.g. traveling salesperson, knapsack, graph coloring, etc.). None of these other problems have known polynomial time algorithms. Hence, it is unlikely that MAXWIRE can be solved in polynomial time.

We shall show that MAXWIRE is NP-hard by showing that the known NP-complete problem Maximum 2-Satisfiability (abbreviated MAX2SAT) α MAXWIRE.

An arbitrary instance I of the problem MAX2SAT is defined as follows.

Input: a set U of n Boolean variables, a collection C of m clauses over U, and a positive integer K $\leq$ m.

Output: "yes" if and only if there is a truth assignment for U that simulatneously satisfies at least K of the clauses in C; "no" otherwise.

In order to prove that MAX2SAT α MAXWIRE, we need several wire configurations. These are the clause adjunct, the vertical assembly, the clause assembly, the transmitter box, the junction pair, the transmitter assembly, and the variable assembly. The proof that MAX2SAT α MAXWIRE will essentially show how to start from an arbitrary instance I of MAX2SAT and construct, in polynomial time, an instance I' of MAXWIRE such that K or more clauses of I can be satisfied if and only if K' or more wires of I' can be laid out (in Manhattan fashion) on one layer. Hence, our description of the wire configurations named above will often make reference to the number of variables, n, and the number of clauses, m, in I.

Before describing the seven configurations cited above, we introduce three conventions that will greatly simplify the discussion. Let (a,b,c,d) represent a wire. If (a,b,c,d) has two feasible layouts then it will be drawn as in Figures 3.1(a) and 3.1(b). The circles in these figures represent the points (a,b) and (c,d). If we wish to give preference to one of these two layouts, then blocking wires may be introduced as in Figure 3.2(a). Let the number of blocking wires be w_b. This number will be determined later. The configuration of Figure 3.2(a) will simply be drawn as in Figure 3.2(b).

The third convention concerns the drawing of a nested set of wires as shown in Figure

Figure 3.1 Wire drawings

Figure 3.2 A blocked configuration

3.3(a). There are k nested wires in the configuration of this figure (k is called the *weight* of the configuration). Figure 3.3(a) will be drawn as in Figure 3.3(b).

A *clause adjunct* is just w_c wires nested together as in Figure 3.3(a). A *vertical assembly* is an alternating sequence of α- and β-wires, with an end wire at the bottom (Figure 3.4(a)). Figure 3.4(b) shows a convenient way to draw a vertical assembly. If we are dealing with an n variable instance of MAX2SAT, then we shall use a vertical assembly with 2n α- and 2n β-wires

A *clause asssembly* consists of a clause adjunct, two vertical assemblies and two bonus wires (Figure 3.4(c)). E_1 and E_2 denote the end wires of the vertical assemblies. Each of the two bonus wires crosses an α-wire. The exact location (i.e. which of the 2n α-wires of a vertical assembly) of the bonus wire will be specified later. The total number of wires in a clause

Figure 3.3 Weighted configuration

assemby is readily seen to be w_c + 8n + 4.

Later, we shall see that a wire from the transmitter assembly will cross each bonus wire. Let the wire that crosses B_1 be Y_1 and the one that crosses B_2 be Y_2. So, Y_1 (Y_2) can be laid out if and only if B_1 (B_2) is not laid out. With this in mind, we proceed to obtain more properties of a clause assembly.

Lemma 3.1: Of the w_c + 8n + 4 (w_c >> n) wires in a clause assembly exactly w_c + 4n + 2 can be laid if and only if at most one of Y_1 and Y_2 is laid out. If both Y_1 and Y_2 are laid out, then at most w_c+ 4n + 1 wires of the clause assembly can be laid out.

Proof: It is clear that an α-wire and a β-wire that are adjacent cannot both be laid out, as they intersect. Hence, from an examination of Figure 3.4(c), we see that at most w_c + 4n + 2 wires can be laid out. If Y_1 is not laid out, then this bound is achieved by laying out the w_c wires of the clause adjunct using the upper layout for each of these wires. This permits us to lay out E_2 and the 2n α-wires of the right vertical assembly. In the left vertical assembly the bonus wire B_1 and the 2n β-wires can be laid out. The case when Y_2 is not laid out is symmetric to this case.

When both Y_1 and Y_2 are laid out, neither B_1 nor B_2 can be laid out. An optimal layout for the clause assembly is obtained by choosing either the upper or lower layout for all wires in the clause adjunct. If the upper layout is used, then E_2 and the 2n α-wires of the right vertical assembly may be laid out. In addition, we can lay out either the 2n α- or β-wires of the left vertical assembly. This gives us a total of w_c + 4n + 1 wires laid out. Similarly, if the lower layout for the clause adjunct is chosen, w_c + 4n + 1 wires can be laid out. □

A *transmitter box* consists of four sets of nested wires (Figure3.5(a)). Each of these sets consists of w_t wires. The regions A, B, C, and D will be used as end points for wires that will be specified later. There are only two ways to lay out all 4 w_t wires in a transmitter box. These are

Figure 3.4 Vertical assembly and clause assembly

shown in Figures 3.5(b) and 3.5(c). Figure 3.5(d) shows the symbolic drawing for a transmitter box.

A *junction pair* consists of 8 w_j wires as shown in Figure 3.6 (each solid line represents w_j wires).

A *transmitter assembly* consists of two transmitter boxes connected together by junction pairs. This conncection is obtained by simply placing the points A, B, C, and D of the junction pair in the appropriate labeled regions in the two transmitter boxes (Figure 3.7(a)). In addition, a vertical assembly (from a clause assembly) crosses the wires of the junction pair. The appropriate vertical assemblies are shown as broken lines in Figure 3.7(a). Figure 3.7(b) shows the symbolic drawing for a transmitter assembly.

Lemma 3.2: If the 8 w_t wires in the two transmitter boxes are laid out, then at most 3 w_j of the 8 w_j junction pairs (i.e. 3 of the 8 weighted wires) in a transmitter assembly can be laid out. Furthermore, if 3 of these weighted wires are laid out, then either both T and T' or both F and F' are laid

Figure 3.5

Figure 3.6 Junction pairs

out.

Proof: From Figures 3.5(b) and 3.5(c), we see that only one of the weighted wires T and F and one of the weighted wires T' and F' may be laid out. Suppose that the left transmitter box is wired such that T can be laid out (the case when F can be laid out is symmetric). If T is not laid

Figure 3.7

out then we can lay out only one of p, q, and T' and one of r, s, and F'. So, we may assume that T is in fact laid out. Hence, p and q cannot be laid out; but T' can. If we do not lay out T' then only one additional weighted wire (i.e. one of r, s, and F') can be laid out. So, we may assume that T' is also laid out. Consequently, F' cannot be laid out. However, regardless of how the vertical assembly is laid out, exactly one of r and s may be laid out (recall that adjacent α- and β-wires cannot both be laid out). Hence, exactly 3 of the weighted junction pair wires can be laid out. □

A *variable assembly* is a sequence of transmitter assemblies connected as in Figure 3.8. The variable assemblies corresponding to MAX2SAT instances with m clauses will consist of 2m + 1 transmitter boxes and 2m junction pairs.

Figure 3.8 Variable assembly

Lemma 3.3: If all the (8m + 4) w_t wires in the transmitter boxes of a variable assembly are laid out, then at most 6m w_j junction pair wires can be laid out. In addition, these 6m weighted wires must include one of the weighted wire sets T and F where:

$$T = \{T_i \mid 1 \le i \le 2m\} \cup \{T_i' \mid 1 \le i \le 2m\}$$
$$F = \{F_i \mid 1 \le i \le 2m\} \cup \{F_i' \mid 1 \le i \le 2m\}$$

Proof: Immediate consequence of Lemma 3.2. □

We are now ready to show that MAXWIRE is NP-hard.

Theorem 3.1: MAXWIRE is NP-hard.

Proof: We shall show that MAX2SAT α MAXWIRE. Let (C = $\{C_1, c_2, .. , C_m\}$, K) be an arbitrary instance of MAX2SAT. Let $x_1, x_2, .. , x_n$ be the n variables in this instance. An equivalent MAXWIRE instance is constructed by setting up n variable assemblies (Figure 3.9). Variable assembly i represents variable x_i, $1 \le i \le n$, and clause assembly i represents clause C_i, $1 \le i \le m$.

Only the location of the bonus wires B_1 and B_2 for each clause assembly needs to be specified. Suppose that $C_1 = (\neg x_i \vee x_j)$. The bonus B_1 crosses the T wire (this corresponds to Y_1) emanating from the first transmitter box in the variable assembly i while the bonus B_2 crosses the

Figure 3.9

F wire (this corresponds to Y_2) emanating from the second transmitter box in the the variable assembly j. In general, if $C_a = (l_i \vee l_j)$ then the B_1 wire for clause a crosses the T (F) wire emanating from the (2a-1)th transmitter box of variable assembly i if $l_i = \neg x_i$ (if $l_i = x_i$). The B_2 wire crosses the T (F) wire from the (2a)th transmitter box of variable assembly j if $l_j = \neg x_j$ (if $l_j = x_j$).

Let Q be a maximal subset of the wires in Figure 3.9 that can be laid out. By choosing w_c, w_b, w_j, and w_t suitably large, we can ensure that all blocking wires, all wires in clause adjuncts, all wires in transmitter boxes, and 3 w_j wires per junction pair are in Q. Let R be the set of remaining wires in Q. Suppose that at least K of the clauses in C are satisfiable. Let $z_1, z_2, .., z_n$ be a truth assignment that satisfies at least K clauses of C. If the 3 w_j wires per junction pair are

chosen such that all the T wires (see Lemma 3.3) of variable assembly i are in Q if and only if z_i = true (hence all the F wires are in Q if and only if z_i = false), then we may choose the following additional wires for layout:

(1) If C_a is satisfied by the assignment $z_1, z_2, .. z_n$, then 4n + 2 wires from the clause assembly a (exluding the wire from the clause adjunct) may be laid out, $1 \leq a \leq m$.

(2) If C_a is not satisfied by the assignment $z_1, z_2, .. , z_n$, then 4n + 1 additional wires from clause assembly a can be laid out (Lemma 3.1).

Thus, if K or more clauses of C are satisfiable, $|R| \geq (4n+1)m + K$. Next, suppose that $|R| \geq (4n+1)m + K$. Since Q is maximal, it is required to contain all clause adjuncts, blocking wires, transmitter box wires, and 3 w_j wires per junction pair. From Lemma 3.3, this means that for every variable assembly i, either all the weighted T wires or F' wires are in Q. Set x_i = true if and only if all the T wires are in Q. Since $|R| \geq (4n+1)m + K$, it follows from Lemma 3.1 that there are at least K clause assemblies each contributing 4n + 2 wires to Q. From the proof of Lemma 3.1, it is evident that clause assembly i can contribute 4n + 2 wires (in addition to the w_c wires in the clause adjunct) to Q if and only if at least one of the bonus wires B_1 and B_2 is in Q. From the placement of the bonus wires, it is clear that a bonus wire can be laid out if and only if the literal whose vertical assembly it crosses is true under the truth assignment determined by the junction wires in Q. So, only clause assemblies that represented clauses that are satisfied under the above truth assignment can contribute 4n + 2 additional wires to R (and hence to Q).

So, $|R| \geq (4n+1)m + K$ if and only if at least K clauses of C are satisfiable. Hence, if MAXWIRE could be solved in polynomial time, then MAX2SAT could also be solved in polynomial time (by using the above construction). Therefore, MAXWIRE is NP-hard. □

4. MINIMIZING LAYERS UNDER THE MANHATTAN CONSTRAINT

The problem of determining the minimum number of layers needed to Manhattan wire a wire set S is NP-hard. This result follows from a much stronger result that we obtain in this section; namely, that determining whether S can be Manhattan wired on two layers is NP-complete. Let us define the problem at hand:

2MAN (Feasibility of Manhattan wiring, given two layers)

Input: A wire set S.

Output: 'Yes' if and only if S can be Manhattan wired on two layers; 'No' otherwise.

2MAN is readily seen to be solvable in nondeterministic polynomial time. So, to show 2MAN NP-complete, it is sufficient to show that it is NP-hard. We do this by showing that the known NP-complete problem 3-Satisfiability (abbreviated 3SAT) α 2MAN.

An arbitrary instance of 3SAT may be defined as follows:

Input: a set U of n Boolean variables; a set C of m clauses over U. Each clause is the disjunction of exactly 3 literals.

Output: "yes" if and only if there is a truth assignment to the variables in U that satisfies each one of the clauses in C; "no" otherwise.

To simplify the discussion, we adopt the following conventions:

(1) We use the tuple (A,B) to denote a wire which was previously denoted by the 4-tuple (a,b,c,d); A denotes the point (a,b) and B the point (c,d). As before, we shall refer to the upper and lower layouts of a wire that has two possible Manhattan layouts.

(2) Let (A,B) be a wire with two possible Manhattan layouts. If we are restricted to two layers, then (A,B) can be forced to use one of its two possible layouts by adding a *blocker assembly*, as in Figures 4.1(a) and (b).

When a layout has been blocked by a blocker assembly, only the remaining feasible layout will be shown. Thus, Figures 4.1(a) and (b) will be represented as in Figures 4.2(a) and (b) respectively. Figure 4.2(c) represents a wire that has neither of its two layouts blocked.

As in Section 3, several wire constructs are needed here. Figure 4.3(a) shows a *clause assembly*. The other endpoint of each of the three wires W_A, W_B and W_C will be specified later. The symbolic form for this assembly is shown in Figure 4.3(b).

Lemma 4.1: The wires (a,a'), (b,b') and (c,c') of a clause assembly (Figure 4.3(a)) cannot be laid out on the same layer.

Proof: By inspection. □

Lemma 4.2: Any two of the wires (a,a'), (b,b') and (c,c') can be laid out on the same layer.

Proof: By inspection. □

Figure 4.1 Blocking a layout in two layers

Figure 4.2 Blocked and unblocked wires

Figure 4.3 Clause assembly

Lemma 4.3: The set of wires in clause assembly can be laid out on two layers if and only if at least one of $\{W_A,W_B,W_C\}$ is not constrained to lie on the same layer as {(A,A'),(B,B'),(C,C')}. Note that because of the wire (E,E'), the three wires (A,A'), (B,B') and (C,C') must be on the same layer in every two-layer layout of a clause assembly.

Proof: Without loss of generality, let us assume that (E,E') is assigned to layer 2. This forces (A,A'), (B,B') and (C,C') to layer 1.

(a) Assume that W_A, W_B and W_C are also required to be laid out on layer 1. There is only one feasible way to lay out the the six wires forced onto layer 1. This is shown in Figure 4.4. It is now easy to see that (a,a'), (b,b') and (c,c') are forced onto layer 2. From Lemma 4.1, we know that these three wires cannot be laid out on the same layer. So, no two-layer layout of a clause assembly is possible when W_A, W_B and W_C are required to be on the same layer as (A,A'), (B,B') and (C,C').

Figure 4.4

(b) Suppose that W_X (where $X \in \{A,B,C\}$) is not required to be on layer 1. Now, it is possible to lay out (x,x') (where x=a if X=A, etc.) on layer 1 (Figure 4.5). From Lemma 4.2, we know that the other two wires, i.e., $\{(y,y')|y \in \{a,b,c\}-\{x\}\}$, can be laid out on layer 2. Hence, the clause assembly can be laid out on two layers. □

A *variable assembly* is shown in Figure 4.6. The broken line is not a part of the assembly and may be ignored at present. m is the number of clauses in the 3SAT instance for wich this assembly is to be used. The variable assembly for variable i is symbolically drawn as in Figure 4.7.

Lemma 4.4: A variable assembly can be laid out on two layers. Furthermore, every two-layer layout of a variable assembly has the following characteristics:

(1) (E,E') and the wire set $U = \{(u_i,u_i')|1 \le i \le m\}$ are all forced to be on the same layer.

(2) All wires in the wire set $T = \{(t_i,t_i')|1 \le i \le m\}$ are wired on the same layer, and on a different layer from the wire set $F = \{(f_i,f_i')|1 \le i \le m\}$.

(3) The layouts for the wires in U are either all upper or all lower.

Figure 4.5

Proof: One may readily verify that a variable assembly can be laid out on two layers. As for the three characteristics:

(1) It is easy to see that in every two-layer layout, the wires (l_i,l_i'), $1\le i\le m$, are on the same layer. Hence, all the wires in the wire set U and the wire (E,E') must all be on the other layer.

(2) By inspection.

(3) From (2), T and F are on different layers. Hence, all the wires in U are on the same layer as either the wires in T or those in F. Suppose that the wires in U and T are on the same layer (say, layer 1). The wires in F must be on layer 2. Consequently, the wires in $Y = \{(y_i,y_i')|1\le i\le m\}$ are on layer 1. This in turn implies that all the wires in U use their upper layout. This does not prevent the wires in $X = \{(x_i,x_i')|1\le i\le m\}$ from being laid out, as layer 2 is still available for these wires.

The case when all wires in U and F are on the same layer is similar. This time, all the wires in U must use their lower layouts. □

The wire configuration in Figure 4.8 is a *crossover junction*.

Figure 4.6 Variable assembly

Lemma 4.5: When a crossover junction is laid out on two layers, the wires (P,P') and (R,R') are always on different layers.

Proof: There are only two possible two-layer layouts for a crossover junction. These are shown in Figures 4.9(a) and (b). Solid lines represent layouts on one layer while broken lines represent layouts on the other. Figure 4.9(a) corresponds to the case when both (P,P') and (X,X') are on the same layer, and Figure 4.9(b) to the case when they are on different layers. In both layouts, (P,P') and (R,R') are on different layers. □

Remark: (P,P') and (R,R') can be thought of as segments of a *super wire* W_S as it crosses an ordinary wire (X,X') (hence the arrows in Figure 4.9). Lemma 4.5 states that when a super wire crosses an ordinary wire, it is forced to change layers, regardless of whether or not the super wire and the ordinary wire are initially on the same layer.

Figure 4.7 Symbolic form of variable assembly i

Figure 4.8 Crossover junction

Figure 4.9 Possible layouts for a crossover junction

Lemma 4.6: When a super wire crosses two ordinary wires, it ends up on the same layer that it started out on. This is true regardless of the layers on which these two ordinary wires are laid out. In fact, when any even number of ordinary wires are crossed by a super wire, the end segments of the super wire are on the same layer.

Proof: Follows from Lemma 4.5 and the earlier remark. □

Theorem 4.1: 2MAN is NP-complete.

Proof: As remarked at the start of this section, 2MAN can be solved in nondeterministic polynomial time. So, it is sufficient to show that it is NP-hard. We do this by showing 3SAT α 2MAN.

Let C = $\{C_1, C_2, .. , C_m\}$ be any instance of 3SAT. Let the variables in the clauses be v_i, $1 \le i \le n$. Corresponding to C, we construct a 2MAN instance as follows:

Step1 Construct n variable assemblies, all using the same (E,E') wire (see Figure 4.10).

Step2 Place m clause assemblies over the (E,E') wire by identifying this (E,E') with the (E,E') of Figure 4.3(a).

Figure 4.10

The wires W_{Ai}, W_{Bi}, W_{Ci}, 1≤i≤m, are super wires, and correspond to the three literals l_a, l_b, l_c, in clause C_i, i.e., super wire W_{Xi} corresponds to the literal l_x, ∀ X ∈ {A,B,C}. Let v_x be the variable in literal l_x, i.e., $l_x \in \{v_x, \neg v_x\}$, ∀ x ∈ {a,b,c}. Super wire W_{Xi} originates in the clause assembly for C_i, and terminates in the variable assembly for v_x. If $l_x = v_x$, W_{Xi} terminates as in Figure 4.11(a). Otherwise, it terminates as in Figure 4.11(b).

In between, W_{Xi} crosses the variable assemblies for the variables v_j, 1≤j<x. The broken line in Figure 4.6 shows the path of such a super wire crossing a variable assembly. It uses a cross-over junction as it crosses each ordinary wire. It is seen that super wires always cross an even number of ordinary wires. Figure 4.12 shows the details for the case $C_i = (v_1 \vee \neg v_2 \vee \neg v_3)$.

This completes the construction of the 2MAN instance corresponding to C. It is easily verified that the number of wires in the 2MAN instance constructed is polynomial in n and m.

Suppose that C is satisfiable. Let $v_i = z_i$, 1≤i≤n, be a truth assignment for which all the clauses in C are true. A two-layer layout of the wires in the constructed 2MAN instance is obtained as follows:

(1) (E,E') and (u_{ij},u'_{ij}), 1≤j≤m, 1≤i≤n, are laid out on layer 1. If z_i = true then (u_{ij},u'_{ij}), 1≤i≤m, are laid out using the upper layout. If z_i = false, the lower layout is used. From Lemma 4.4, it

Figure 4.11 Termination of a super wire

follows that the remaining wires in the variable assemblies get forced onto one or the other of the layers.

(2) Since $v_i = z_i$, $1 \leq i \leq n$, satisfies all the clauses, at least one literal in every clause is true under this assignment. Hence, the terminating segment (Figure 4.11(a)) of at least one of the super wires of every clause can be laid out on layer 1. The remaining terminating segments can be laid out on layer 2. From Lemma 4.6 and the fact that every super wire crosses an even number of ordinary wires, it follows that exactly one of the W_A, W_B, W_C wires of each clause assembly will be on layer 1. From Lemma 4.3, it follows that the remaining wires in the clause assemblies can be laid out on layer 2.

Next, suppose that the 2MAN instance can be laid out on two layers. From Lemma 4.4, we know that all the U wires of all n variable assemblies are on one layer (say layer 1). The wire (E,E') is also on layer 1. Also, from Lemma 4.4, all the U wires of a particular variable assembly use either the upper layout or the lower layout. Set v_i to true if and only if the U wires of variable assembly i use the upper layout.

An examination of Figure 4.3 reveals that all the (A,A'), (B,B') and (C,C') wires of the clause assemblies must be on layer 2. From Lemma 4.3, it follows that the starting segment of at least one of the W_A, W_B, W_C super wires of each clause assembly must be laid out on layer 1.

Figure 4.12 $C_i = v_1 \vee \neg v_2 \vee \neg v_3$

From Lemma 4.6 and the fact that every super wire crosses an even number of ordinary wires, it follows that the end segment of at least one of the super wires in every clause assembly is laid out on layer 1. From Figure 4.11, we see that this implies that at least one literal in every clause is

true. Hence, the 3SAT instance is satisfiable.

Thus, the 3SAT instance C has answer ’Yes’ if and only if the constructed 2MAN instance can be laid out on two layers. The construction takes only a polynomial amount of time (as a function of n and m). Therefore, 3SAT α 2MAN. □

5 RECTILINEAR WIRING

The rectilinear wiring problem (RECT) may formally be stated as follows:

RECT

Input: A p by q grid, and a set S of n wires. Each wire is specified by its end points. These end points are themselves grid points.

Output: 'Yes' if and only if there exists a rectilinear layout for the wires in S. Layouts are constrained to lie along grid lines, and may not intersect one another.

Theorem 5.1 shows that RECT is NP-complete. The proof of this is similar to the proof provided by [LYNC75] to show that the disjoint paths problem in a graph is NP-complete. As in the case of Theorem 4.1, the known NP-complete problem 3SAT is used. Before proceeding to Theorem 5.3, we introduce some conventions and wire assemblies.

Let (Y,Z) be a wire. We may force the layout of this wire to follow a particular grid path by introducing several wires that have end points around the desired path. Figure 5.1(a) shows a forced layout path for the wire (Y,Z). The X's denote points a wire cannot cross: these points are end points of other wires. Adjacent X's could, for instance, denote the two end points of a wire. A wire (Y,Z) whose layout is forced will be drawn as in Figure 5.1(b). The solid line denotes a feasible layout for the wire.

Figure 5.1 Forced path

Consider the grid of Figure 5.2(a). A connection is to be made between pairs of points that have the same label. Thus, a total of 8 wires are to be layed out. By examining this figure, one

sees that all eight wires can be layed out only if the (1,1) wire is layed out using grid segments from either only the upper half or only from the lower half of the grid. Furthermore, if (1,1) is layed out using grid segments from only the upper (lower) half of the grid, then (8,8) must also be layed out using only the upper (lower) half of the grid. When both (1,1) and (8,8) are layed out using only grid segments from the upper (lower) half, we shall say that the upper or true (lower or false) layout has been used. Figures 5.2(b) and (c), respectively, show a possible true and false layout. Observe that in both cases it is possible t also layout a wire connecting points C and C'. The grid of Figure 5.2(a) together with the points C and C' defines a *neutral crossover box*. Figure 5.4(a) shows a symbolic representation for a neutral crossover box.

False and *true crossover boxes* are obtained from a neutral crossover box by blocking off the layout of (C,C') that occurs in the "true" and "false" layouts respectively of the neutral crossover box. This is done by placing an X at the point marked T (F) in the case of a true (false) crossover box (see Figure 5.3). Figures 5.4(b) and (c) show the symbolic representation for true and false crossover boxes.

Two crossover boxes can be connected together as follows:

1) Remove the leftmost column of Xs from the right box and the the three rightmost columns of the left box.

2) Butt the two boxes together so that the point labeled 8 in the left box overlaps the leftmost 1 of the right box. Delete the point labeled 8 in the left box (but not the one labeled 1 that overlaps it). Introduce a point labeled 7 just after the first 1 of the right box. Change the labels of the original point of the right box so that they are different from those used in the left box.

Clearly, this scheme can be generalized to cascade, or serially connect together, a number of crossover boxes. All the crossover boxes so connected must be laid out the same, all using either the upper or the lower layout. Figure 5.5 shows the symbolic form for two crossover boxes (one neutral and the other false) that are connected together.

Theorem 5.1: RECT is NP-complete.

Proof: It is easy to see that RECT can be solved in nondeterministic polynomial time. So, we need only to show that 3SAT α RECT.

Let C = $\{C_1, C_2, .., C_m\}$ be an arbitrary 3SAT instance. Let v_i, $1 \le i \le n$, be the variables in C. For each variable v_i, a variable assembly consisting of 3m crossover boxes cascaded together serially is constructed. The exact type of each crossover will be specified later. The relative

placement of these variable assemblies is as shown in Figure 5.6. The (C,C') wires of the 3m crossover boxes of variable assembly i are connected to the 3m (C,C') wires of variable assembly i+1, 1≤i<n (see Figure 5.6). Finally, the (C,C') wires are grouped in threes and their tops and bottoms made common. Each group of three (C,C') wires represents a clause. These groups have the end points (C_i,C_i'), 1≤i≤m, as in Figure 5.6. Thus there are three different ways in which the wires (C_i,C_i') can be laid out. Each of these three paths represents a literal in the clause.

Figure 5.6 The constructed RECT instance

The choice of crossover boxes depends on the 3SAT instance C. Let clause C_i be $l_a \vee l_b \vee l_c$, where $l_x \in \{v_x,\neg v_x\}$, $\forall$ x ∈ {a,b,c}. The first path for (C_i,C_i') represents l_a, the second l_b and the third l_c. The path representing l_a uses neutral crossover boxes to cross over all variable

Figure 5.2 Neutral crossover box

Figure 5.3 True and false crossover boxes.

Figure 5.4 Symbolic drawings for crossover boxes

assemblies other than that for v_a. In crossing the variable assembly for v_a, it uses a true crossover box if $l_a = v_a$ and a false crossover box if $l_a = \neg v_a$. Figure 5.6 shows the case when $C_1 = (v_1 \vee \neg v_2 \vee v_n)$ and $C_2 = (\neg v_1 \vee v_2 \vee v_i)$.

Figure 5.5

Suppose that C is satisfiable. Let $v_i = z_i$, $1 \le i \le n$, be a truth assignment that satisfies all the clauses. A rectilinear layout for the RECT instance created above is obtained as follows:

(1) If z_i = true, the crossover boxes in variable assembly i use the "true" (upper) layout. If z_i = false, they use the "false" (lower) layout.

(2) For each clause, there is at least one literal that is true under the specified truth assignment. Consider clause $C_i = l_a \vee l_b \vee l_c$. Thus, l_x must be true for some $x \in \{a,b,c\}$. The wire (C_i, C_i') can be laid out using the path corresponding to l_x. This path goes through n-1 neutral crossovers and one other crossover. It is always possible to layout the neutral crossovers so that the (C_i, C_i') wire can pass through it (Figure 5.2). If $l_x = v_x$ $(\neg v_x)$, this (C_i, C_i') path goes through a true (false) crossover in variable assembly i. Since z_x is true (false), the layout specified in (1) above would permit this, and (C_i, C_i') can be completely laid out. Thus, if C is satisfiable, all the wires in the constructed RECT instance can be laid out.

Next, suppose that all the wires can be laid out. Consider any such complete layout. From our earlier discussion, it is clear that all the crossover boxes of a variable assembly must be laid out the same, all using either the "true" path or the "false" path. If the wires in the assembly for v_i use the "true" layout, v_i is set to true. Otherwise, it is set to false. We claim that this assignment results in all the clauses evaluating to true. This follows from the discussion in (2) above.

Hence, the RECT instance constructed can be laid out if and only if the given 3SAT instance is satisfiable. Also, the RECT instance contains only a polynomial (in m and n) number of wires, and can be obtained from C in polynomial time. So, 3SAT α RECT. □

6 CONCLUSIONS

Manhattan and rectilinear wiring problems are important problems in the design automation of digital systems. We have shown that there is an efficient algorithm to determine whether a wire set can be laid out in Manhattan fashion on one layer. If the given wire set cannot be laid out on one layer, it is natural to seek the largest subset of the given wire set that can be laid out in Manhattan fashion on one layer. We have shown that this is NP-hard. Also, we have established that determining whether a given wire set can be Manhattan wired on two layers is NP-hard. Finally, we have shown that determining whether a given wire set can be wired in rectlinear fashion on one layer is NP-hard.

It is very unlikely that any of these three NP-hard problems can be solved efficiently. This motivates the search for heuristic and approximation algorithms for these problems.

Acknowledgements

We are grateful to an anonymous referee who pointed out that once the relations **L**, **D**, **I**, and **U** had been formulated, the algorithm for 2 satisfiability could be used to solve the one layer Manhattan wiring problem in $O(n^2)$ time. Our earlier algorithm had complexity $O(n^3)$.

REFERENCES

BREU72 M.A.Breuer (ed.), *Design Automation of Digital Systems*, Prentice Hall, 1972.

EVEN76 S.Even, A.Itai, and A.Shamir, "On the complexity of timetable and multicommodity flow problems," SICOMP, vol 5, no 4, pp 691-703, 1976.

GARE79 M.R.Garey and D.S.Johnson, *Computers and Intractability: a Guide to the Theory of NP-Completeness*, W.H.Freeman and Co., 1979.

HIGH73 D.W.Hightower, "The Interconnection Problem - a Tutorial", Proc. 10th Design Automation Workshop, 1973, pp.

LYNC75 J.F.Lynch, "The Equivalence of Theorem Proving and the Interconnection Problem", SIGDA Newsletter 5:3, 1975.

POME65 T.Pomentale, "An Algorithm for Minimizing Backboard Wiring Functions", CACM 8:11, November 1965.

SAHN80 S.Sahni, A.Bhatt and R.Raghavan, "The Complexity of Design Automation Problems", Dept. of Computer Science, Univ. of Minnesota, Tech. Rept. 80-23, 1980.