Two NP-hard Interchangeable Terminal Problems*

Sartaj Sahni and San-Yuan Wu

University of Minnesota

ABSTRACT

Two subproblems that arise when routing channels with interchangeable terminals are shown to be NP-hard. These problems are:

P1: Is there a net to terminal assignment that results in an acyclic vertical constraint graph?

P2: For instances with acyclic vertical constraint graphs, obtain net to terminal assignments for which the length of the longest path in the vertical constraint graph is minimum.

Key words and phrases

Interchangeable terminals, NP-hard, PLA-routing, Complexity.

*This research was supported, in part, by the National Science Foundation under grant MCS-83-05567

1 Introduction

The problem of routing channels with interchangeable terminals arises, for example, when the cells on either side of the channel are programmable logic cells (e.g. ROMs and PLAs). An instance of this problem is shown in Figure 1(a). This instance consists of eight cells $A - H$. Cells $A - D$ are on one side (top/upper) of the channel while cells $E - H$ are on the other (bottom/lower). On the top side, terminals $1 - 4$ are in cell A. While terminals $12 - 15$ on the bottom side of the channel are in cell H. The nets to be assigned to the terminals of a cell are given in braces. Thus, the nets for cell C's five terminals are NETS(C)={1,1,3,4,5}. Since the terminals in a cell are interchangeable, all net to terminal assignments are permissible. Figures 1(b) and 1(c) show two possible assignments of nets to terminals.

With any assignment of nets to terminals, we may associate a vertical constraint graph (VCG). This graph has one vertex for each net. In addition, the directed edge $<i,j>$ is an edge of the VCG iff there is a k such that net i is assigned to terminal k on the top side of the channel and net j is assigned to terminal k on the bottom side of channel. The VCGs for the assignments of Figures 1(b) and (c) are shown in Figures 1(d) and (e), respectively.

Many channel routing algorithms require the VCG to be acyclic. Futher, the height of the VCG (i.e., the length of the longest path in the VCG) is a lower bound on the number of horizontal tracks needed to route the channel when two routing layers (one for horizontal and the other for vertical routes) are available. Keeping these factors in mind, Kobayashi and Drozd [KOBA85] have proposed a three step method to assign nets to terminals. The three steps are:

1. Permute the terminals in each cell so as to maximize the number of aligned terminal pairs.
2. Exchange terminals that are in nonaligned pairs so as to remove cyclic constraints in the resulting vertical constraint graph.
3. Exchange terminals so as to reduce the height of the vertical constraint graph.

Lin and Sahni [LIN86] have developed a linear time algorithm for step 1. In this paper, we show that the remaining two steps are NP-hard. We show this by reducing the known NP-complete problem monotone 3SAT to each of these. **Monotone 3SAT [GARE79]**

INPUT:Set V of variables, collection C of clauses over V such that each clause $c \varepsilon C$ has $|c| = 3$ and contains either only negated variables or only un-negated variables.

QUESTION:Is there a satisfying truth assignment for C ?

2 Basic constructs

In this section we develop some interchangeable terminal instances that will be used in our proofs. We show how the possible terminal assignments for these instances may be interpreted as truth assignments for logical variables.

2.1 Truth Assignment Box

A truth assignment box for a variable x consists of two cells A and B on the upper side of the channel and one, C, on the lower side (see Figure 2(a)). Cells A and B each have one terminal while cell C has two. NETS(A)= { x }, NETS(B)= { $\overline{x}$ }, and NETS(C)= { T, F }. There are exactly two ways in which the nets can be assigned to the terminals of A, B, and C. These are shown in Figures 2(b) and (c). We shall interpret the assignment of Figure 2(b) as setting variable x to true and $\overline{x}$ to false. This assignment is called a *true assignment*. The assignment of Figure 2(c), called a *false assignment*, corresponds to setting x to false and $\overline{x}$ to true. The VCG corresponding to the assignment of Figure 2(b) is shown in Figure 2(d) and that corresponding to the assignment of Figure 2(c) is shown in Figure 2(e). A truth assignment box for variable x, TAB(x), will be drawn schematically as in Figure 2(f). We will refer to the four terminals as TAB(x).x, TAB(x).$\overline{x}$, TAB(x).T, and TAB(x).F .

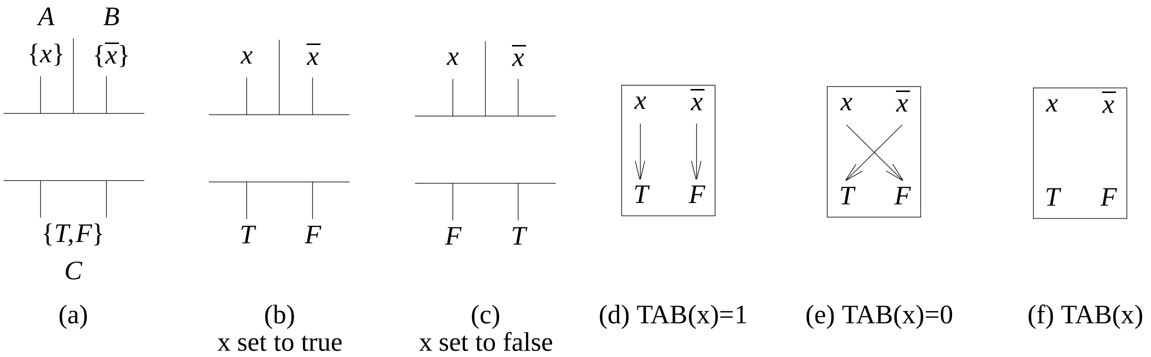


Figure 2: Truth assignment box

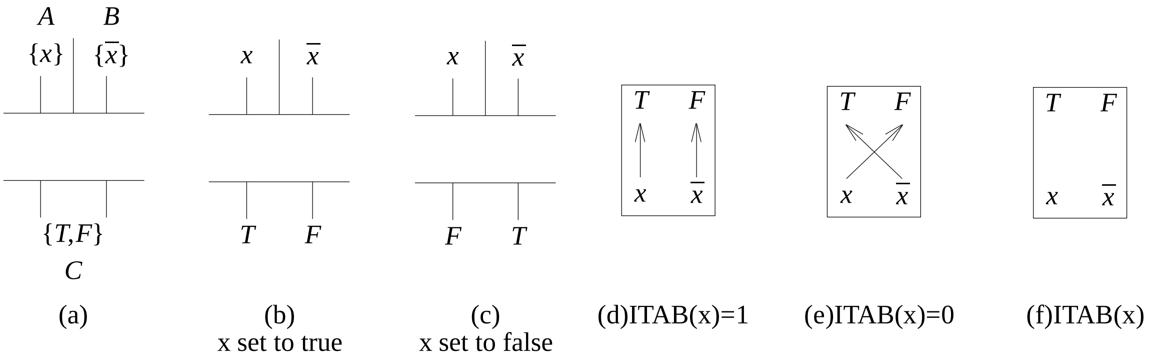
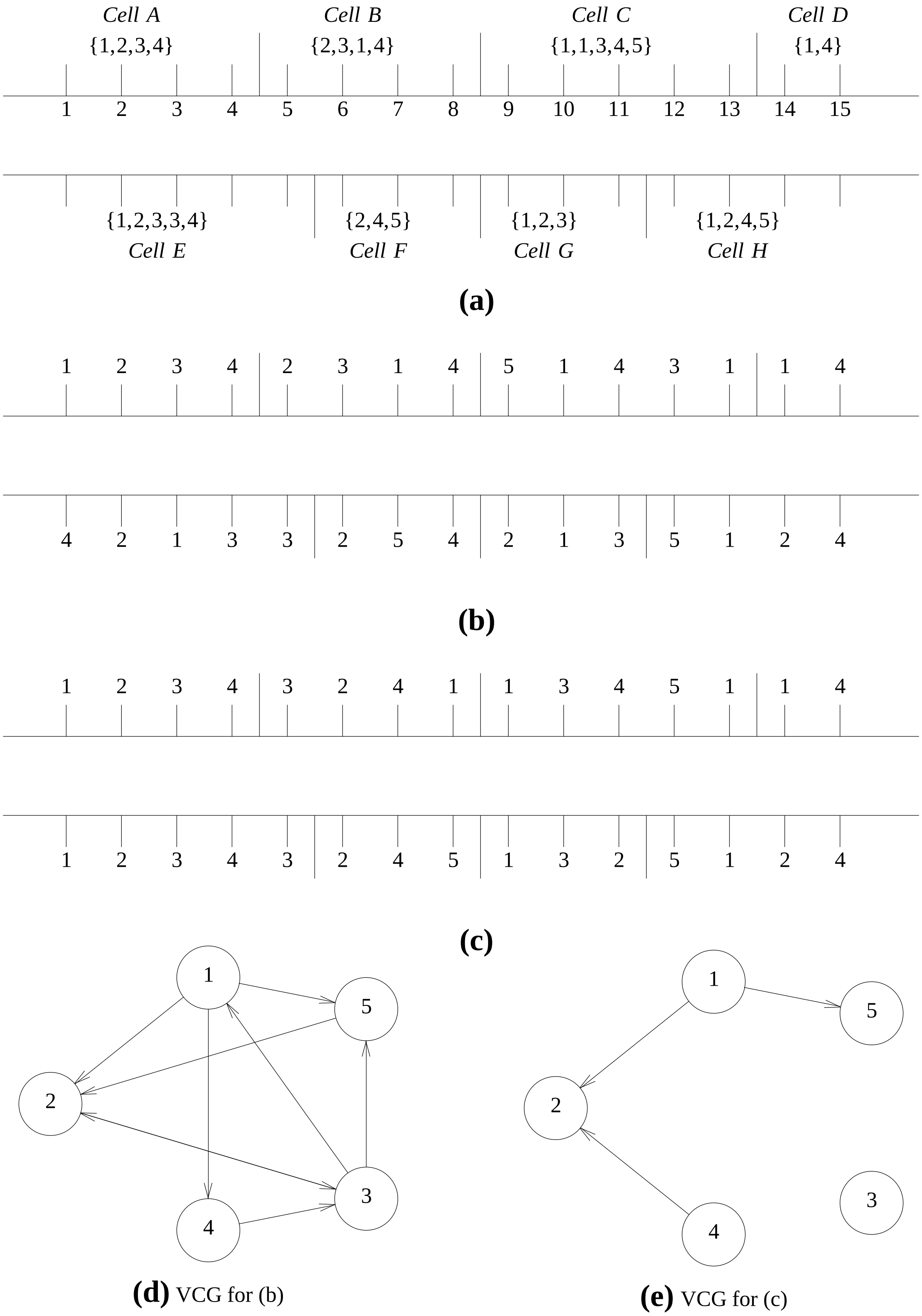


Figure 3:Inverted truth assignment box

2.2 Inverted Truth Assignment Box

An inverted truth assignment box, ITAB, is identical to a TAB except that the schematic is drawn upside down (see Figure 3). The upside down drawing of the schematic results in cleaner figures.



(a)

(b)

(c)

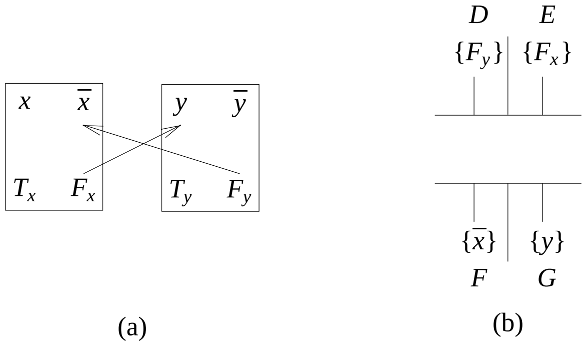
(d) VCG for (b)

(e) VCG for (c)

Figure 1: An instance of the interchangeable terminal routing problem.

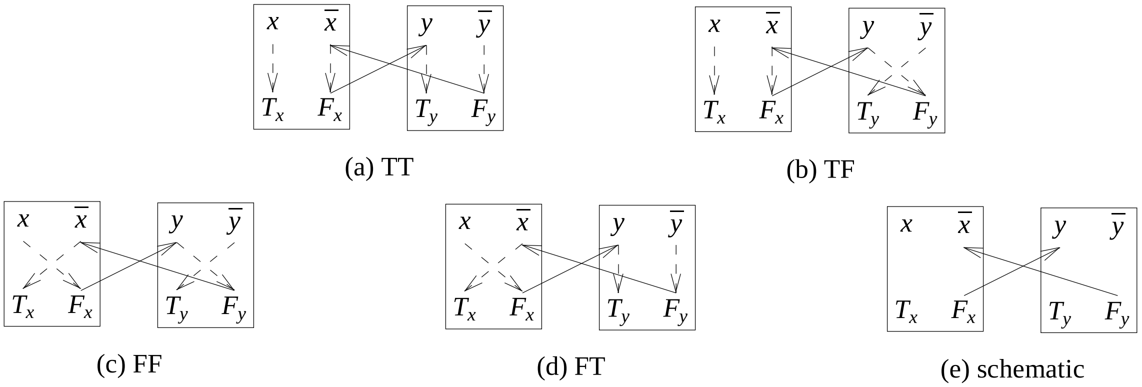
2.3 TAB Connection Box

A TAB connection box is used to connect the VCGs of two TABs. Consider the two TABs of Figure 4(a). The connection box for these consists of four cells *D*, *E*, *F*, and *G* as shown in Figure 4(b). There are four possible VCGs for the interchangeable terminal instance that consists of the two TABs and the connection box. These correspond to the four terminal assignments TT (*x* and *y* are true), TF, FF, and FT and are shown in Figure 5. Figure 5(e) shows the schematic for two TABs and a connection box. >From Figure 5, we observe that the TF terminal assignment results in a VCG with a cycle while the VCG for the remaining three terminal assignments are acyclic.



(a) (b)

Figure 4: TAB Connection Box



(a) TT

(b) TF

(c) FF

(d) FT

(e) schematic

Figure 5

2.4 TAB Chain

A TAB chain consists of k, $k \geq 1$, TABs, TAB(x_1), ..., TAB(x_k) with a connection box for each pair of TABs, (TAB(x_i), TAB(x_{x+1})), $1 \leq i < k$. The schematic for this is shown in Figure 6. The following property is an immediate consequence of our discussion of the preceding section.

Property 1: The VCG for a TAB chain is acyclic iff one of the following is true:

a) The terminal assignment for all TABs is true.

b) The terminal assignment for all TABs is false.

c) There exists a j, $1 \le j < k$, such that the terminal assignment for TAB(x_i), $1 \le i \le j$ is false and that for the remaning TABs is true.

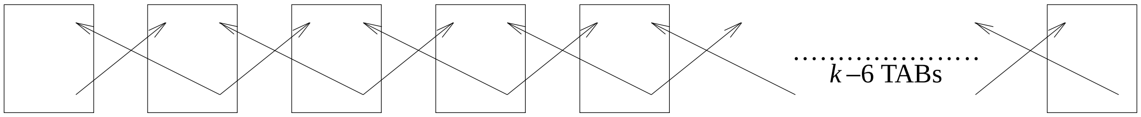


Figure 6: TAB Chain

2.5 ITAB connection box

The connection box for two inverted truth assignment boxes ITAB(x) and ITAB(y) is similar to that for two TABs and is shown in Figure 7. The schematic for the two ITABs and their connection box is shown in Figure 8(e). Figures 8(a)-(d) show the VCGs for the four possible terminal assignments. The VCG for the TF assignment contains a cycle. The remaning three VCGs are acyclic.

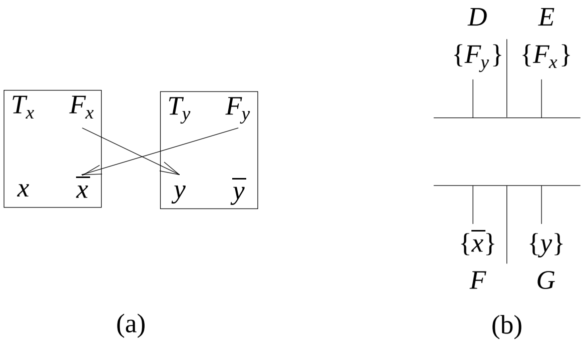


Figure 7: ITAB Connection Box

2.6 ITAB Chain

This is just k ITABs and $k-1$ ITAB connection boxes. The schematic is shown in Figure 9. The following property is readily verified.

Property 2: The VCG for an ITAB chain is acyclic iff one of the following is true:

a) All ITABs are assigned true.

b) All ITABs are assigned false.

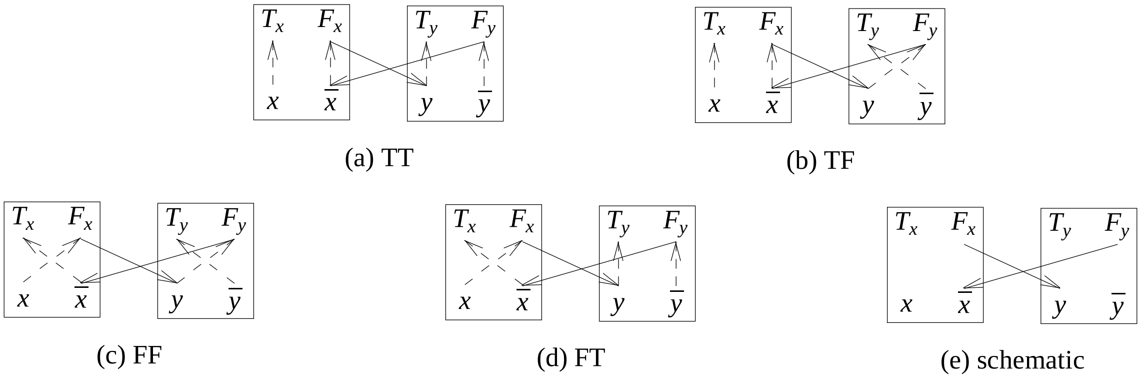


Figure 8

c) There exists a j, $1 \le j < k$, such that the terminal assignment for ITAB(x_i), $1 \le i \le j$, is false and that for the remaining ITABs is true.

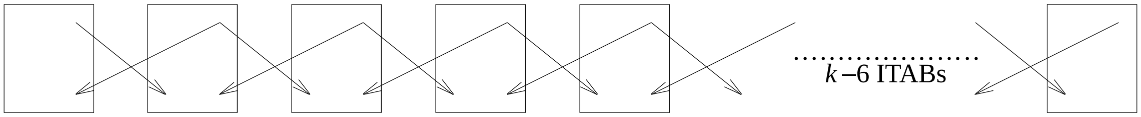


Figure 9: ITAB Chain

2.7 TAB to ITAB Isolation Connection

An isolation connection is used to connect together a truth assignment box TAB(x) and an inverted truth assignment box ITAB(y). The connection is established using six cells as shown in Figure 10(a). As can be seen from Figures 10(b)-(e), if the TAB and ITAB have different truth assignments, then the VCG has a cycle. Figure 11(a) shows the schematic for a TAB chain and an ITAB chain. The rightmost TAB is connected to the leftmost ITAB using an isolation connection.

Property 3:

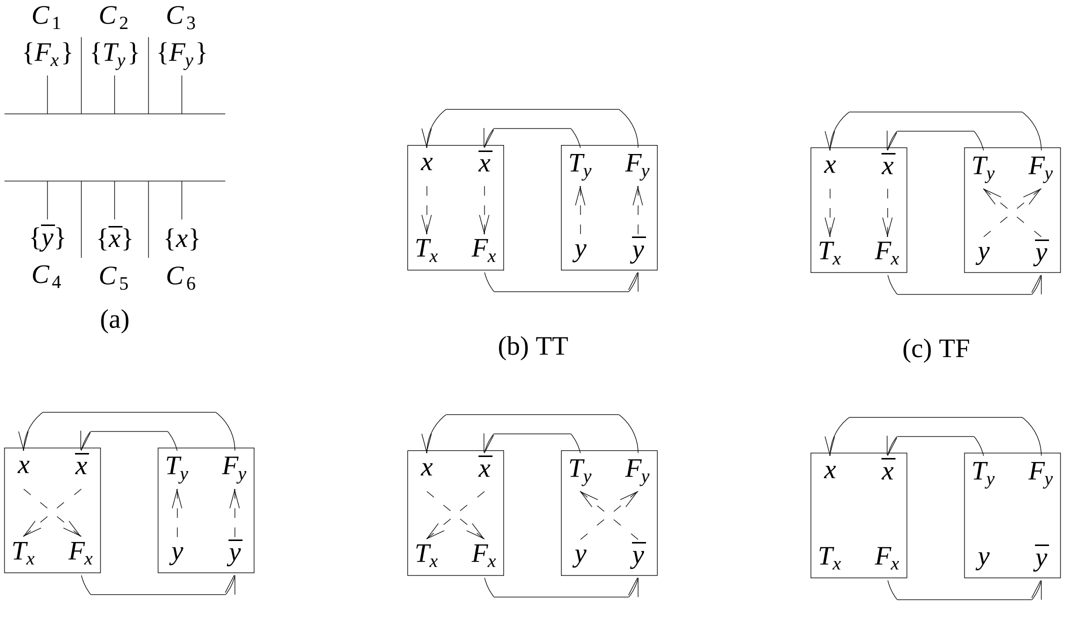
Assume that a TAB and ITAB chain are connected using an isolation connection as in Figure 11(a). If the VCG for the resulting chain has no cycles, then the following are true:

(i) The VCG has no path that begins in a TAB other than the rightmost one and ends in an ITAB other than the leftmost ITAB.

(ii) The VCG has no path that begins in an ITAB other than the leftmost one and ends in a TAB other than the rightmost one.

In other words, an isolation connection isolates the TABs from the ITABs and vice versa.

Proof: >From Figures 10(b)-(e), it follows that if the VCG of Figure 11(a) has no cycle, then the rightmost TAB and leftmost ITAB have the same truth assignment. Hence there are only two possibilities for the VCG for the rightmost TAB, the leftmost ITAB, and the isolation connection. Thus, this part of the VCG has one of the forms given in Figures 11(b) and (c). As can be seen, there can be no path that violates (*i*) or (*ii*). □



(a)

(b) TT

(c) TF

(d) FT

(e) FF

(f) schematic

FIGURE 10: Isolation connection

2.8 ITAB to TAB Isolation Connection

This is shown in Figure 12. It has the same properties as a TAB to ITAB isolaion connection.

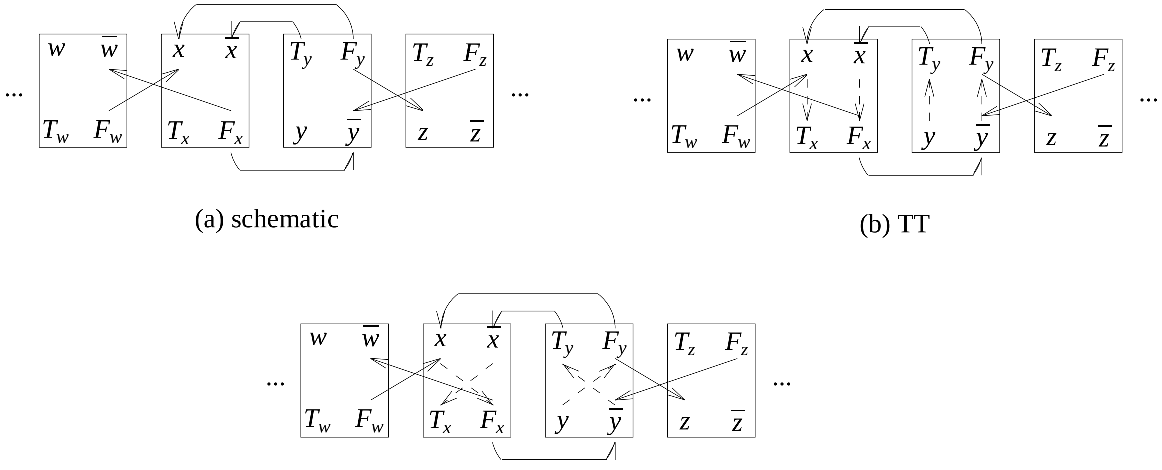
3 Acyclic VCG (Problem P1) is NP-hard

We show this by reducing monotone 3SAT to P1. Specifically, we show how to construct an instance $P(S)$ of P1 for any given instance S of monotone 3SAT. $P(S)$ has the following properties:

(*i*) $P(S)$ may be constructed from S in polynomial time.

(*ii*) $P(S)$ has a terminal assignment with an acyclic VCG iff S is satisfiable.

Properties (*i*) and (*ii*) imply that P1 is NP-hard. Actually, P1 is readily seen to be solvable

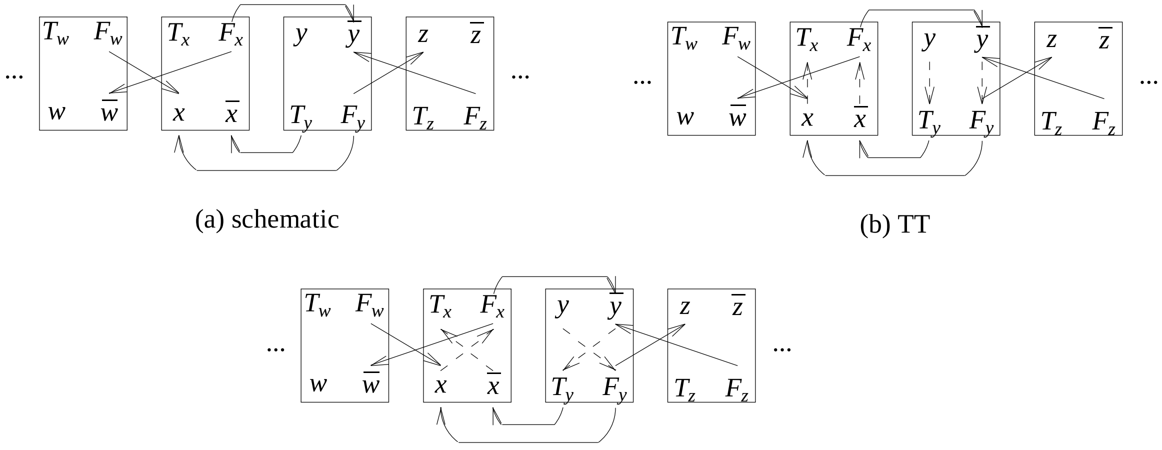


(a) schematic

(b) TT

(c) FF

FIGURE 11: Isolation from TAB chain to ITAB chain



(a) schematic

(b) TT

(c) FF

FIGURE 12: Isolation from ITAB chain to TAB chain

in nondeterministic polynomial time. Hence this observation together with our NP-hard proof imply that P1 is NP-complete.

3.1 Construction of $P(S)$

Let S be any instance of MONOTONE 3SAT. Let m and k, respectively, be the number of variables and clauses in S. The clauses in S may be partitioned into two sets U and N. U contains all clauses that contain only unnegated variables and N contains all clauses that contain only negated variables. Let u and n, respectively, be the number of clauses in U and N. Clearly, $u+n = k$. As an example, consider the case: $S = (A+B+C) \cdot (A+C+D) \cdot (\overline{B}+\overline{C}+\overline{D})$. For this, $U = \{ (A+B+C), (A+C+D) \}$, $N = \{(\overline{B}+\overline{C}+\overline{D}) \}$, $u = 2$, and $n = 1$.

For each of the m variables in S, construct a row of TABs and ITABs. Each such row consists of k (note k is the number of clauses in S) TAB and ITAB chains connected by TAB to ITAB or ITAB to TAB isolation connections as appropriate. Specifically, each row is a TAB chain with two TABs followed by an ITAB chain with 3 ITABs; followed by a TAB chain with 3 TABs; followed by an ITAB chain with 3 ITABs; etc.

For our example S, $m = 4$ and $k = 3$. So we will have 4 rows of TABs and ITABs with each row consisting of a TAB chain with 2 TABs; followed by an ITAB chain with 3 ITABs; followed by a TAB chain with 2 TABs. This is shown in Figure 13. Note that TAB to ITAB isolation connections are used to connect the first TAB chain of each row to the ITAB chain in that row and that ITAB to TAB isolation connections are used to connect the ITAB chain in each row to the second TAB chain in that row.

For each i, $1 \le i \le k$, the truth boxes in column $3i-2$ of the above construction will be used to represent a distinct clause of S. The first u of these columns (i.e., columns $1, 4, 7, ..., 3u-2$) will represent the u unnegated clauses of S while the last n of these columns (i.e., columns $3u+1, 3u+4, ..., 3k-2$) will represent the n negated clauses.

Let $U_1, U_2, ..., U_u$ be the u unnegated clauses. Let $U_i = (x_{i_1}+x_{i_2}+x_{i_3})$. Without loss of generality, we may assume that the TAB/ITAB row for variable x_{i_1} is above that for x_{i_2}, which in turn is above that for x_{i_3}. Column $3i-2$ of the truth box construction of Figure 13 is used to represent U_i. If i is odd, this column contains TABs. Otherwise, it contains ITABs. We introduce the three edges $< TAB(x_{i_1}).F\ , TAB(x_{i_2}).x_{i_2} >$, $< TAB(x_{i_2}).F\ , TAB(x_{i_3}).x_{i_3} >$, $< TAB(x_{i_3}).F\ , TAB(x_{i_1}).x_{i_1} >$ in case this is a column of TABs and the three edges $< ITAB(x_{i_1}).F\ , ITAB(x_{i_3}).x_{i_3} >$, $< ITAB(x_{i_3}).F\ , ITAB(x_{i_2}).x_{i_2} >$, $< ITAB(x_{i_2}).F\ , ITAB(x_{i_1}).x_{i_1} >$ in case this is a column of ITABs.

Let $N_1, N_2, ..., N_n$ be the n negated clauses. Let $N_i = (\overline{x}_{i_1}+\overline{x}_{i_2}+\overline{x}_{i_3})$. Once again , we assume that the row for x_{i_1} is above that for x_{i_2} which is above that for x_{i_3}. Column $3(u+i)-2$ is used to represent N_i. Into this column, the three edges $< TAB(x_{i_1}).F\ , TAB(x_{i_2}).\overline{x}_{i_2} >$, $< TAB(x_{i_2}).F\ , TAB(x_{i_3}).\overline{x}_{i_3} >$, $< TAB(x_{i_3}).F\ , TAB(x_{i_1}).\overline{x}_{i_1} >$ are introduced if this is a column of TABs. The three edges $< ITAB(x_{i_1}).F\ , ITAB(x_{i_3}).\overline{x}_{i_3} >$, $< ITAB(x_{i_3}).F\ , ITAB(x_{i_2}).\overline{x}_{i_2} >$, $< ITAB(x_{i_2}).F\ , ITAB(x_{i_1}).\overline{x}_{i_1} >$ are introduced if this is a column of ITABs.

The resulting edges for our example are shown in Figure 13. Recall that to introduce an edge $<a, b>$ into the VCG, we need merely add two cells to the terminal assignment instance. Each of these has exactly one terminal; one cell is above the other; the upper cell has net a; the lower cell has net b. Futher, note that while we have used the symbols T and F for the lower nets of all TABs and the upper nets of all ITABs, the intent is that all these nets are different. This

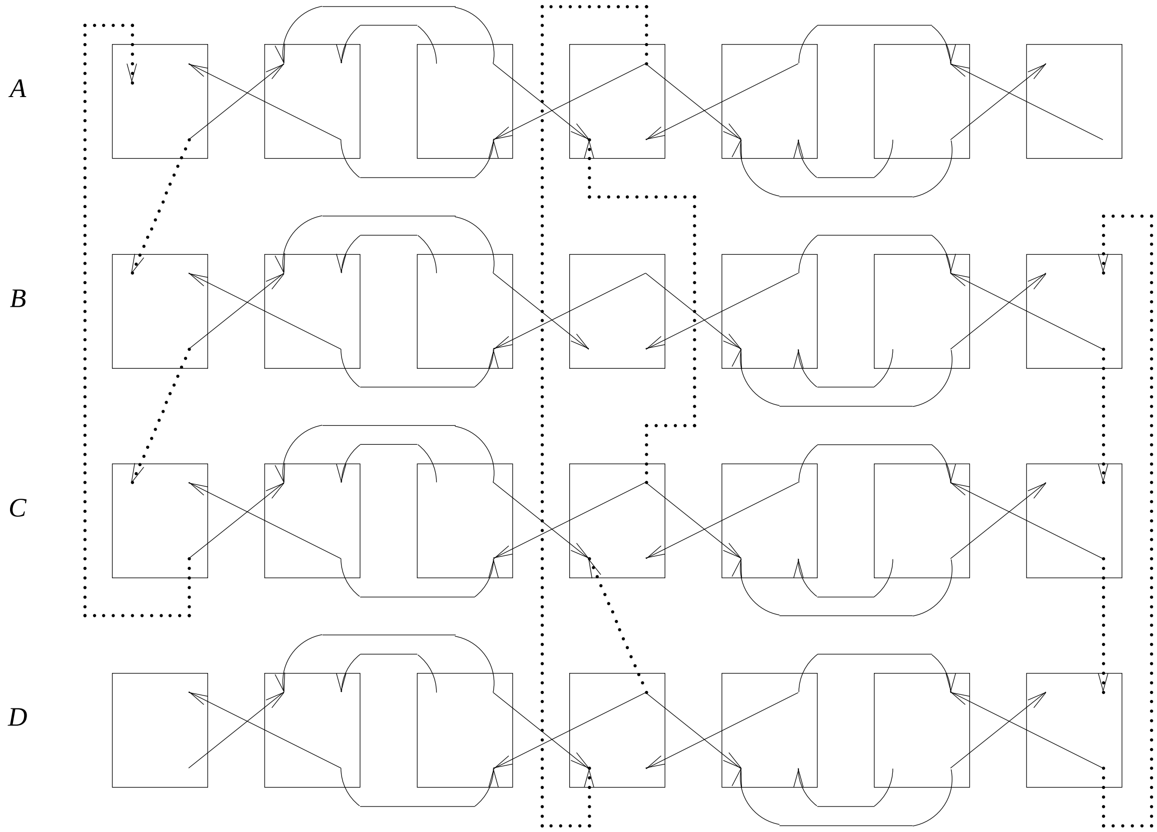


FIGURE 13: $F = (A + B + C) \cdot (D + C + A) \cdot (\overline{B} + \overline{C} + \overline{D})$

completes the construction of $P(S)$.

Property 4:If all the literals in clause i are false, then column $3i-2,\ 1 \le i \le k$ of $P(S)$ contains a cycle.

3.2 Correctness Proof

We need to show that $P(S)$ as constructed in Section 3.1 has the two properties listed in Section 3. It is obvious that property (i) (i.e., $P(S)$ may be constructed in polynomial time) is satisfied. For property (ii), first suppose that S is satisfiable. Consider any truth assignment that satisfies S. Use this truth assignment for all the truth boxes (TABs and ITABs) in the construction. >From properties 1-4, it follows that the resulting VCG contains no cycles. Note that because of the isolation property (property 3) there can be no path that includes boxes from two or more clause columns.

Next, suppose that $P(S)$ has a terminal assignment that contains no cycles. >From Property

4, it follows that this assignment corresponds to a truth assignment in which every clause has at least one literal set to true. However, since the truth assignment along a TAB or ITAB chain may change from false to true without introducing a cycle (Properties 1 and 2), it is possible that the terminal assignment that results in an acyclic VCG results in an inconsistent truth assignment to the variables of S (i.e., a variable is false in some clauses and true in others). If this is the case, the truth assignment may be made consistent in the following way:

> Suppose x_j is false in the first q of the truth boxes in columns $3i-2,\ 1 \le i \le k$ (only those boxes in the row for x_j are considered) and true in the remaining $k-q$ boxes in columns $3i-2,\ 1 \le i \le k$.
>
> (a) If $q \le u$, then set x_j to true and ($\overline{x}_j$ to false) in all the clauses in which x_j or $\overline{x}_j$ appear. This makes the truth assignment for x_j the same in all clauses and does not reduce the number of true literals in any clause.
>
> (b) If $q > u$, then set x_j to false (and $\overline{x}_j$ to true). This does not reduce the number of true literals in any clause and results in a consistent truth assignment for x_j.

Hence, if there is a terminal assignment for $P(S)$ that results in an acyclic VCG, S is satisfiable.

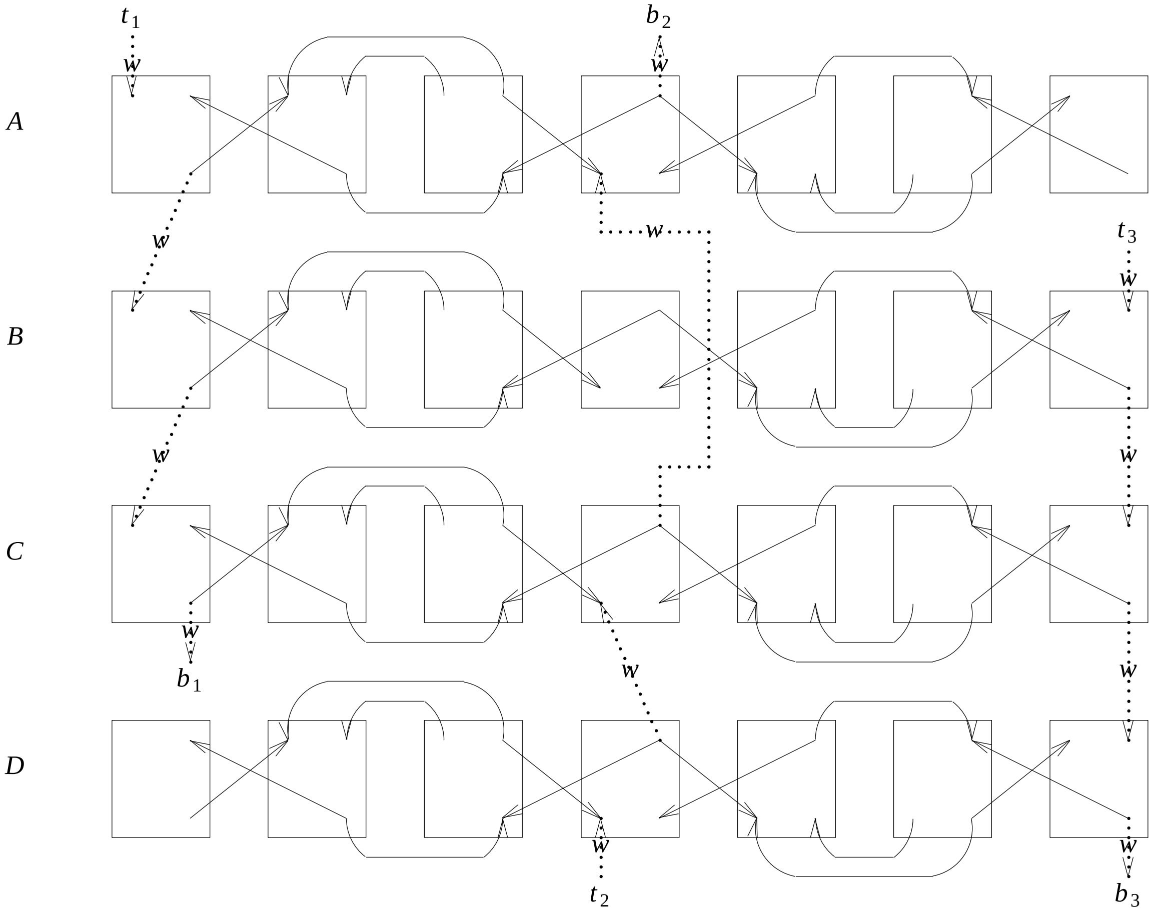


FIGURE 14: $F = (A + B + C) \cdot (D + C + A) \cdot (\overline{B} + \overline{C} + \overline{D})$

4 Minimum Height VCG (Problem P2) Is NP-hard

We actually prove a stronger result than this in this section. We show that the corresponding decision problem (P3) is NP-complete:

P3: Let Q be a terminal assignment instance with an acyclic VCG and r an integer. Does Q have a terminal assignment for which the length of the longest path in the VCG is $< r$?

P3 is easily seen to be solvable in nondeterministic polynomial time. So we need only show that P3 is NP-hard. For this, we show how to construct an instance $Q(S)$ of P3 corresponding to any instance S of monotone 3SAT. This instance $Q(S)$ satisfies properties (i) and (ii) of Section 3. Once we know that the decision problem P3 is NP-complete, we can conclude that the optimization problem P2 is NP-hard.

4.1 Construction of $Q(S)$

For any instance S of monotone 3SAT, we first construct $P(S)$ as in Section 3. Next, we replace the edges introduced for clauses by edges with weight w as shown in Figure 14. The cyclic clause constructs of Figure 13 are replaced by path constructs. Each clause is represented by a (possible) path from a newly introduced top vertex t_i to a newly introduced bottom vertex b_i. Each of the weighted edges of Figure 14 is really a path of length w *and is realized by the construct of Figure 15. Let* $Q(S)$ *be the instance constructed in Figure 14.*

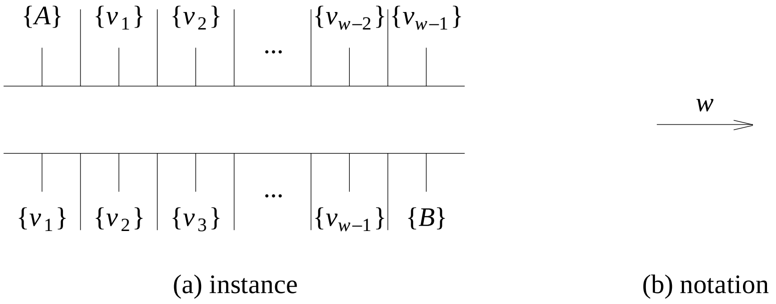
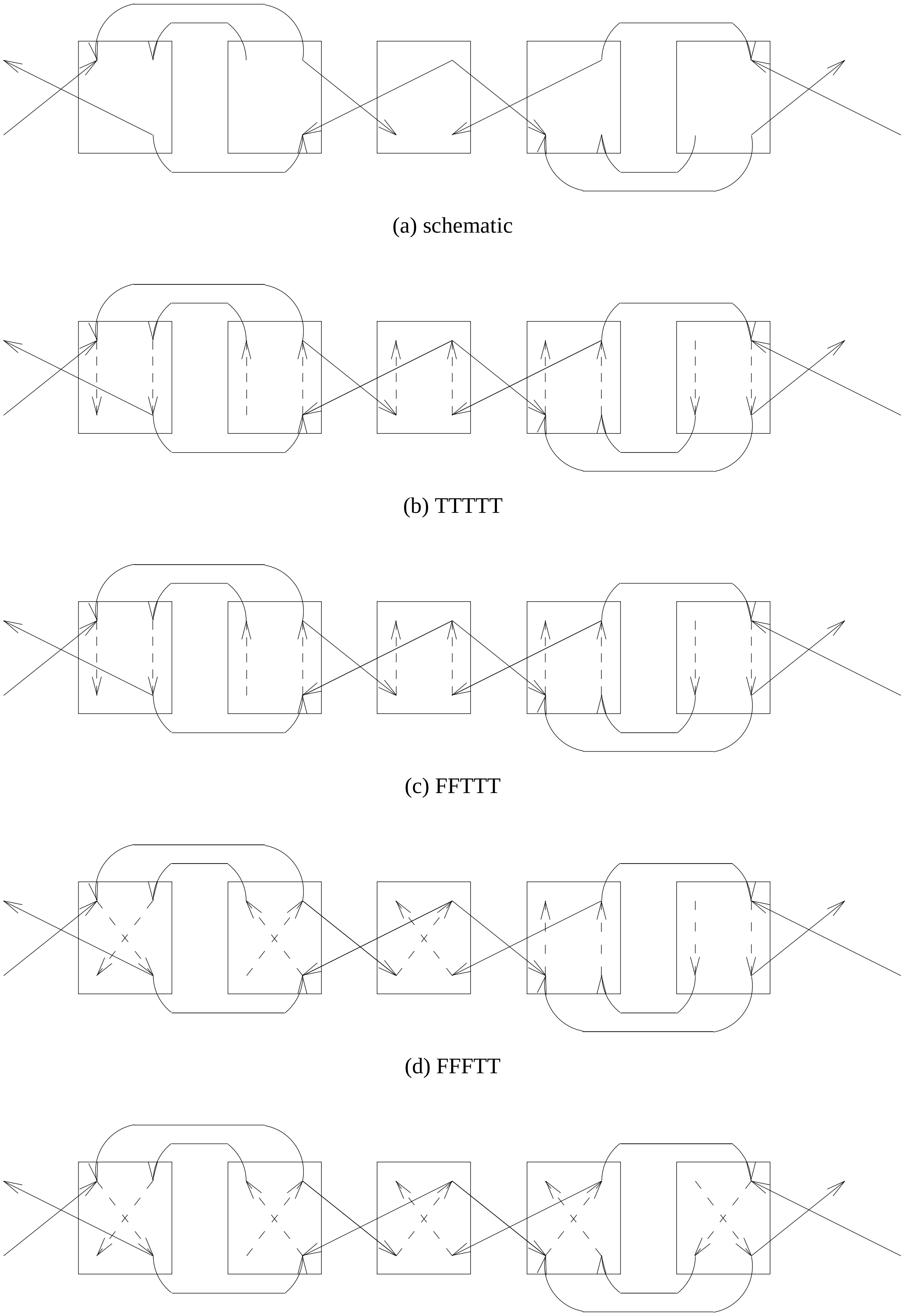


FIGURE 15: Weighted edge with weight = w

4.2 Correctness Proof

Consider a segment of one row of $Q(S)$. This segement consists of five truth boxes as shown in Figure 16 (the case when the middle box is a TAB is similar). As can be seen, no path confined to a row of truth boxes can have length larger than 9 (unless the row contains a cycle). Suppose that $w \geq 9$. If $P(S)$ has no acyclic terminal assignment, then every terminal assignment for $Q(S)$ has at least one path from a t_i to a b_i. This path has length $4w+3$. If $P(S)$ has an acyclic terminal assignment, then the corresponding assignment in $Q(S)$ results in a VCG for which the maximum path length is no more than $3w+2+9 = 3w+11 < 4w+3$.

Hence, $Q(S)$ has a terminal assignment for which the length of the longest path in the VCG is $< 4w + 3$ iff $P(S)$ has a terminal assignment for which the VCG is acyclic. This, in turn, is possible if and only if S is satisfiable.



(a) schematic

(b) TTTTT

(c) FFTTT

(d) FFFTT

(e) FFFFF

FIGURE 16

5 Conclusions

We have shown that both problems P1 and P2 are NP-hard. Consequently, it is very unlikely that either of these is solvable in polynomial deterministic time. Our results motivate the study of fast heuristic methods for these two problems.

6 References

[BURS83] M. Burstein and R. Pelavin, "Hierarchical wire routing", IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, Vol. CAD-2, No. 4, Oct. 1983.

[DEUT76] D. N. Deutsh, "A ’Dogleg’ channel router", Proc. 13th Design automation conference, pp. 425 - 433 (1976).

[GARE79] M. R. Garey and D. S. Johnson, "Computers and Intractability", W. H. Freeman and Company, 1979.

[KOBA85] H. Kobayashi and C. E. Drozd, "Efficient algorithms for routing interchangeable terminals", IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, Vol. CAD-4, No. 3, pp. 204 - 207, (1985).

[LIN86] L. Lin and S. Sahni, "Maximum alignment of interchangeable terminals", University of Minnesota, Technical Report # 86-13, 1986.

[RIVE82] R. L. Rivest and C. M. Fiduccia, "A greedy channel router", Proc. 19th Design automation conference, pp. 418 - 424 (1982).

[YOSH82] T. Yoshimura and E. S. Kuh, "Efficient algorithms for channel routing", IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, Vol. CAD-1, No. 1, pp. 25-35, Jan. 1982.