Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Information Processing Letters 99 (2006) 171–174
Information Processing Letters
www.elsevier.com/locate/ipl

TETRAVEX is NP-complete

Yasuhiko Takenaga [a], Toby Walsh [b,*]

[a] *Department of Computer Science, The University of Electro-Communications, Tokyo, Japan*
[b] *National ICT Australia and University of NSW, Sydney, Australia*

Received 24 October 2005; received in revised form 10 April 2006; accepted 17 April 2006
Available online 9 June 2006
Communicated by K. Iwama

Keywords: Computational complexity; NP-completeness; TETRAVEX

TETRAVEX is a widely played one person computer game in which you are given n^2 unit tiles, each edge of which is labelled with a number. The objective is to place each tile within a n by n square such that all neighbouring edges are labelled with an identical number. Unfortunately, playing TETRAVEX is computationally hard. More precisely, we prove that deciding if there is a tiling of the TETRAVEX board given n^2 unit tiles is NP-complete. Deciding where to place the tiles is therefore NP-hard. This may help to explain why TETRAVEX is a good puzzle. This result compliments a number of similar results for one person games involving tiling. For example, NP-completeness results have been show for: the offline version of Tetris [1], KPlumber (which involves rotating tiles containing drawings of pipes to make a connected network) [2], and shortest sliding puzzle problems [3]. It raises a number of open questions. For example, is the infinite version Turing-complete? How do we generate TETRAVEX problems which are truly puzzling as random NP-complete problems are often surprising easy to solve? Can we observe phase transition behaviour? What about the complexity of the problem when it is guaranteed to have an unique solution? How do we generate puzzles with unique solutions?

Theorem 1. TETRAVEX *is* NP-*complete.*

Proof. Clearly it is in NP. Given a solution, we can check it in polynomial time. To show completeness, we use a reduction from 1in3-SAT on purely positive clauses. We will map a problem in n variables and m clauses onto a rectangular TETRAVEX problem of size $O(n)$ by $O(m)$. We can always convert a rectangular TETRAVEX problem into an essentially equivalent but larger square problem by adding suitable tiles.

There are five types of component used in the construction: a horizontal assignment component along the top edge, vertical clause components, vertical and horizontal wiring tiles and junction components to connect vertical to horizontal wires. We assume the variables are labelled from 1 to n. The ith part of the assignment component consists of four tiles that are in one of two configurations:

* Corresponding author.
E-mail addresses: takenaga@cs.uec.ac.jp (Y. Takenaga), tw@cse.unsw.edu.au (T. Walsh).

doi:10.1016/j.ipl.2006.04.010

$$i = \mathit{true}: \quad \boxed{\begin{array}{ccc} & \mathit{top} & \\ i-1 & & i \\ & 0 & \end{array}} \quad \boxed{\begin{array}{ccc} & \mathit{top} & \\ i & & i \\ & i & \end{array}} \quad \boxed{\begin{array}{ccc} & \mathit{top} & \\ i & & i \\ & -i & \end{array}} \quad \boxed{\begin{array}{ccc} & \mathit{top} & \\ i & & i+1 \\ & 0 & \end{array}}$$

$$i = \mathit{false}: \quad \boxed{\begin{array}{ccc} & \mathit{top} & \\ i-1 & & i \\ & 0 & \end{array}} \quad \boxed{\begin{array}{ccc} & \mathit{top} & \\ i & & i \\ & -i & \end{array}} \quad \boxed{\begin{array}{ccc} & \mathit{top} & \\ i & & i \\ & i & \end{array}} \quad \boxed{\begin{array}{ccc} & \mathit{top} & \\ i & & i+1 \\ & 0 & \end{array}}$$

The value *top* is used to ensure that these tiles can only be placed along the top of puzzle. No tile in the puzzle has *top* at its bottom label. Actually, this is not essential and we can label the top of these tiles 0. However, it makes the proof easier if we force the assignment component to be on the top edge of the puzzle. The $i - 1$ value in the leftmost tile, and the $i + 1$ value in the rightmost tile are used to ensure that the assignment components are laid out in order from left to right along the top row of the puzzle. The value 0 is used for internal tiles which are not components. We also start and end the top row with the tiles:

$$\boxed{\begin{array}{ccc} & \mathit{top} & \\ \mathit{left} & & 0 \\ & 0 & \end{array}} \quad \ldots \quad \boxed{\begin{array}{ccc} & \mathit{top} & \\ n+1 & & \mathit{right} \\ & 0 & \end{array}}$$

The value *left* and *right* are used to ensure that a tile appears on the left or right edge of the puzzle. No tile has *left* as its right label. Similarly, no tile has *right* as its left label. This is not essential and we could label them zero, but it again makes the proof easier.

This "signal" ($i, -i$ which is interpreted as i is *true*, or $-i, i$ which is interpreted as i is *false*) is then transmitted to the vertical clause components via "wires". There are vertical wires, horizontal wires and junctions. A vertical wire is of the form:

$$\boxed{\begin{array}{ccc} & i & \\ X & & X \\ & i & \end{array}} \quad \text{and} \quad \boxed{\begin{array}{ccc} & -i & \\ X & & X \\ & -i & \end{array}}$$

Where X is either 0 (if the wire is passing a blank part of the puzzle), or the value of a horizontal wire being crossed. It is important to note that this value for X is not equal to i. When we cross a wire carrying the signal from the ith variable, we use a junction component. These vertical wiring tiles can appear in either order depending on the polarity of the signal being transmitted. A horizontal wire is of the form:

$$\boxed{\begin{array}{ccc} & X & \\ i & & i \\ & X & \end{array}} \quad \text{and} \quad \boxed{\begin{array}{ccc} & X & \\ -i & & -i \\ & X & \end{array}}$$

Where X is again either 0 (if the wire is passing a blank part of the puzzle), or the value of a vertical wire being crossed. We again note that this value for X is not equal to i. These horizontal wiring tiles can appear in either order depending on the polarity of the signal being transmitted.

A junction connects a vertical pair of wires with a horizontal pair of wires. Each junction is labelled with an unique number j where $j > n + m$. The junction consists of one of two possible arrangements of four tiles:

$$\begin{array}{cc} \boxed{\begin{array}{ccc} & i & \\ i & & -i \\ & -j & \end{array}} & \boxed{\begin{array}{ccc} & -i & \\ -i & & i \\ & j & \end{array}} \\ \boxed{\begin{array}{ccc} & -j & \\ -i & & i \\ & i & \end{array}} & \boxed{\begin{array}{ccc} & j & \\ i & & -i \\ & -i & \end{array}} \end{array} \quad \text{or} \quad \begin{array}{cc} \boxed{\begin{array}{ccc} & -i & \\ -i & & i \\ & j & \end{array}} & \boxed{\begin{array}{ccc} & i & \\ i & & -i \\ & -j & \end{array}} \\ \boxed{\begin{array}{ccc} & j & \\ i & & -i \\ & -i & \end{array}} & \boxed{\begin{array}{ccc} & -j & \\ -i & & i \\ & i & \end{array}} \end{array}$$

The junction turns the vertical signal i, $-i$ into the horizontal signal $\genfrac{}{}{0pt}{}{i}{-i}$, or the vertical signal $-i$, i into the horizontal signal $\genfrac{}{}{0pt}{}{-i}{i}$.

Whilst it is possible to stack two wiring tiles horizontally and place them between the left and right half of the junction component, we cannot then line up with the wiring components coming down from the assignment component. Similarly, it is impossible to put two wiring tiles between the top and bottom half of the junction component. Therefore this junction component must occur in this two by two form.

Finally, there is the clause component. For each clause, we have 12 vertically arranged tiles. Suppose the pth clause is $i \vee j \vee k$. We label this with an unique number, $c = n + p$. The clause component consists of a top tile, a buffer tile, then two tiles connected to the wires bringing in the i, $-i$ or $-i$, i signal, another buffer tile, two tiles connected to the wires bringing in the j, $-j$ or $-j$, j signal, another buffer tile, two tiles connected to the wires bringing in the k, $-k$ or $-k$, k signal, one more buffer tile, and finally the bottom tile. There are thus four buffer tiles in total. Two of them are labelled:

$$\boxed{\begin{array}{ccc} & -c & \\ \mathit{left} & & 0 \\ & -c & \end{array}}$$

The *left* label ensures that this tile is placed along the left edge of the puzzle. Such tiles will be adjacent to any pair of wires bringing in a *false* signal. The other two buffer tiles, which are immediately above and below the pair of wires bringing in the *true* signal, are:

$$\boxed{\begin{array}{ccc} & -c & \\ \mathit{left} & & 0 \\ & c & \end{array}} \quad \text{and} \quad \boxed{\begin{array}{ccc} & c & \\ \mathit{left} & & 0 \\ & -c & \end{array}}$$

The top and bottom tiles of the clause component are:

$$\boxed{\begin{array}{ccc} & 0 & \\ \mathit{left} & & 0 \\ & -c & \end{array}} \quad \text{and} \quad \boxed{\begin{array}{ccc} & -c & \\ \mathit{left} & & 0 \\ & 0 & \end{array}}$$

Finally, the two tiles connecting the wires bringing in the i signal are:

$$\boxed{\begin{array}{ccc} & c & \\ \mathit{left} & & i \\ & -c & \end{array}} \quad \text{and} \quad \boxed{\begin{array}{ccc} & -c & \\ \mathit{left} & & -i \\ & c & \end{array}}$$

A similar pair of tiles connect the wires bringing in the j, $-j$ or $-j$, j signal, and the k, $-k$ or $-k$, k signal.

To illustrate the clause component, we give the 12 tiles representing the clause $i \vee j \vee k$, with i being the only variable set true. For brevity, we lay the tiles out horizontally in four (connected) rows (see Fig. 1).

With the clause component, we pick out one (and only one) of the three input pairs of wires to be *true*. Since the top-most and bottom-most labels of each clause component is zero, we can lay out the clause components in any order. However, they must be along the left edge of the puzzle as the left side of each tile is labelled *left* and no tile has *left* as its right-hand label.

Finally, we fill in the rest of the puzzle with tiles labelled just zero. The puzzle is $4n+2$ tiles wide. In the assignment component along the top edge, each of the n variables is assigned a value using a block of 4 tiles. We also have the start and end of row tiles. This makes $4n + 2$ in total. The puzzle is $12m + 1$ tiles high. Each of the m clauses using a component with 12 tiles. There is also the top left tile which starts the assignment component. This makes $12m + 1$ in total. Within the puzzle, there are $3m$ junction components. Each clause requires 3 junctions, one for each variable. There are $24nm - 12m$ vertical wiring tiles. Each of the n variables has a vertical wire running the $12m$ tiles from the assignment component at the top of the puzzle to the bottom edge of the puzzle. Each of these wires is two tiles wide. This gives $24nm$ tiles. There are, however, no wiring tiles where we have junction components. There are $3m$

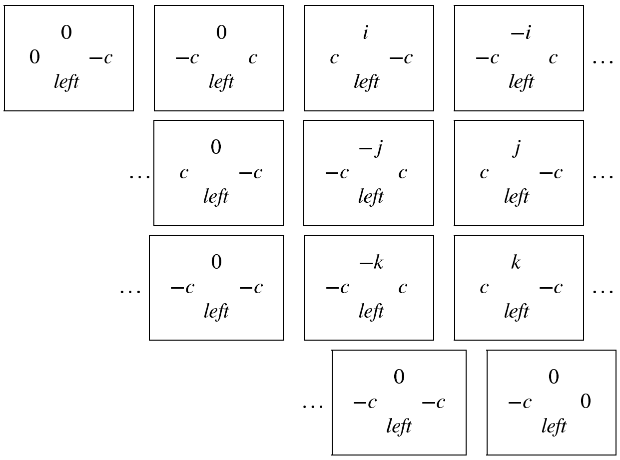

Fig. 1.

junctions in total, each consisting of 4 tiles. Hence, there are $24nm - 12m$ vertical wiring tiles in total. Finally, there are $24nm - 6m$ horizontal wiring tiles. Each of the m clauses has three horizontal wires running the $4n + 1$ tiles from the clause component at the left edge of the puzzle to the right edge of the puzzle. Each of these wires is two tiles wide. This gives $6m(4n + 1)$ tiles. There are, however, no wiring tiles where we have junction components. There are $12m$ such tiles. Hence, there are $24nm - 6m$ horizontal wiring tiles in total.

Suppose there is an assignment which satisfies just one variable in each clause. Then it is easy to see that there is a proper tiling of the puzzle. Suppose, on the other hand, that there is no such assignment. Assume there was a proper tiling of the puzzle. This means that there must be an assignment component along the top edge. Now the vertical wires can only fit in the puzzle if they are connected to this component. Similarly, there must be a clause component along the left edge. The horizontal wires can only fit in the puzzle if they are connected to this component. Finally, the junction components can only fit into the puzzle if they wire up the horizontal and vertical wires correctly. Thus we have a correct wiring of the circuit. However, this is only possible if we can satisfy the 1in3-SAT problem. Hence, the 1in3-SAT problem is satisfiable iff there is a proper tiling of this puzzle. □

Some observations can be made about this result. We can add the circular boundary condition that the top edge of the square matches the bottom, and the left edge matches the right. TETRAVEX with such boundary conditions remains NP-complete (since it is easy to modify the reduction so all edges are labelled zero). We can also generalize TETRAVEX to 3 (or more) dimensions. The problem remains NP-complete as we need only one plane for the reduction and can use (hyper)cubes labelled with zero everywhere else. Finally, our reduction requires $O(n + m)$ integer labels. It is an open question if the problem remains NP-complete when we have just $O(1)$ different labels.

References

[1] E.D. Demaine, S. Hohenberger, D. Liben-Nowell, Tetris is hard, even to approximate, in: Proceedings of the 9th International Computing and Combinatorics Conference (COCOON 2003), 2003, pp. 351–363.

[2] D. Kral, V. Majerech, J. Sgall, T. Tichy, G. Woeginger, It is tough to be a plumber, Theoret. Comput. Sci. 313 (3) (2004) 473–484.

[3] D. Ratner, M. Warmuth, Finding a shortest solution for the $(n \times n)$-extension of the 15-puzzle is intractable, J. Symbolic Comput. 10 (2) (1990) 111–137.