BOUNDED TILING, an alternative to SATISFIABILITY ?

Martin W.P. Savelsbergh

Department of Operations Research and System Theory,
Centre for Mathematics and Computer Science, Amsterdam

Peter van Emde Boas

IIW/FVI, University of Amsterdam

1. Introduction

Computational complexity theory has established itself as one of the central subjects of computer science. It has connections with other areas like combinatorial optimization, mathematical logic, algorithm design, and practical topics like cryptology. Among the technical concepts in this field the notion of NP-completeness and the technique of polynomial-time reducibility have become standard tools in different areas of applied mathematics. These notions enable us to show in a convincing technically well defined way that some problems should be considered to be intractable. This fact can then be used as a justification for the use of exponential time enumerative methods and approximative heuristic rules for solving these problems in practice.

The foundations for this part of complexity theory were laid in two classical papers by Cook [2] and Karp [5]. Cook [2] introduced the concept of NP-completeness and proved that the problem SATISFIABILITY has this property of being the most difficult problem in the class NP. He also defined an appropriate reducibility concept which was later used in a modified way by Karp [5] in order to establish NP-completeness results for some twenty well known notorious problems from combinatorial optimization. Many more problems were to follow.

In their 1979 textbook on the subject Garey & Johnson [3] cover a major part of the developments since 1971. Johnson's quarterly column in the Journal of Algorithms, published since December 1981, presents some highlights of the more recent developments. At the same time the subject has become standard material in contemporary textbooks on computation theory and other subjects.

With one notable exception Cook's SATISFIABILITY problem has served as the "master problem" in the theory as appearing in the literature. The exception is found in the recent textbook by Lewis & Papadimitriou [7] where the BOUNDED TILING problem as proposed by Lewis [8] is used as the main entrance into the theory. We should mention at this place also that Levin [6], who discovered the notion of NP-completeness independently, described six different master problems, one of which is a variant of BOUNDED TILING - his paper, however, is practically unknown in the West.

In this paper we investigate whether BOUNDED TILING indeed presents a viable alternative for the foundation of the NP-completeness theory. This investigation was inspired by the experiences of the second author, when he was preparing an introductory course on complexity theory for non-mathematicians interested in cryptography [11]. At that occasion he felt responsible to show his audience the true state of the world and our ignorance concerning complexity, by giving a self contained sketch of the entire theory, culminating in the NP-completeness proof of KNAPSACK. It turned out that BOUNDED TILING enables one to present such a proof without ever talking about SATISFIABILITY.

We are not interested in this paper in proving new NP-completeness results. We rather investigate how the proofs of the standard results will look if the entire theory is based upon BOUNDED TILING rather than SATISFIABILITY. Section 2 therefore contains a complete proof of the analogue of Cook's theorem in which we establish the NP-completeness of BOUNDED TILING. In Section 3 we give reductions for the six basic problems from Garey & Johnson, Chapter 3 [3]. Section 4 contains some discussion of what we have achieved.

2. NP-completeness of BOUNDED TILING

By a "master reduction" we mean a proof establishing NP-completeness of some combinatorial problem by a direct encoding of polynomially bounded nondeterministic computations as instances of this problem. Our master reduction to BOUNDED TILING (which problem will be defined formally in the sequel of this section) uses Turing's model of computation, as was the case with Cook's original reduction to SATISFIABILITY.

For the sake of completeness we present a short sketch of the precise model of computation used. Such a sketch is needed anyhow if this proof is to be used for explaining complexity theory to people outside mathematics and computer science as is our purpose.

In the *Turing machine model* in its simplest form, one considers a machine which operates under control of a finite program on a two way infinite tape. On this tape symbols are written from a finite alphabet Σ; the tape consists of tape squares which are linearly ordered like the integers. During each stage of the computation the machine has visited only a finite number of tape cells, and initially only a finite number of tape cells carry information; those cells which are not yet visited and were not written on in the initial state carry a special symbol, called *blanc,* which represents the absence of information. (Alternatively the tape is always finite, but extended by blanc cells whenever needed.)

The *finite control* of the machine consists of a *program* containing *quintuples* of the form (q,s,q',s',M), where q and q' are elements from a finite set K whose elements are called *states*, s and s' are symbols in Σ and M is a "move" which equals either R,0 or L representing "move right", "don't move" and "move left" respectively. The intended meaning of the above quintuples reads: "if in state q you are reading symbol s then go to state q', overwriting s by s' and move at most one square along the tape as indicated by M". Next another *instruction* (quintuple) can be executed.

If for every pair (q,s) there exists at most a single instruction in the program starting with this pair the machine is called **deterministic;** otherwise it is called **nondeterministic.**

Initially, the machine is started in a selected *initial state* q_0 looking at the left most *tapecell* of a consecutive block of (presumably nonblanc) cells, representing the *input.* Subsequently instructions present in the program are executed as long as there are instructions present for the current combination of state and symbol read; if no such instruction can be found in the program the machine *halts*, and the computation is completed.

Instantaneous descriptions (ID's) are denoted by a string of tape symbols containing a single occurrence of a state q_i to the left of the symbol currently being scanned by the reading head In our reduction it is useful to consider the pair consisting of the state and the scanned symbol to be a single symbol. Blanco symbols outside the scanned fragment of the tape are omitted.

A Turing machine accepts some input string w whenever there exists a sequence of ID's $T_0,T_1,...,T_t$ such that $T_0=q_0w$, T_t contains the accepting final state q_f and T_{i+1} is obtained from T_i by executing an instruction in the program of the machine. The sequence of ID's represents the *computation* of the machine on input w. For deterministic machines the computation is determined completely by the input. If the machine is nondeterministic there may exist more than one ID T_{i+1} which can follow T_i; this leads to a tree of computations.

For the sequel of this paper it is profitable to restrict the collection of valid Turing machines by enforcing further conditions on the accepting computations. For example we require that a machine, before accepting, erases all information on the tape (printing blanco symbols), returns to its original position on the tape (which position is specially marked in order to retrieve it) and only then halts and accepts. This restriction leads to a machine that has a unique accepting configuration. A rather more technical condition is the requirement that there exists no state such that the machine can move both right and left while going into this state: i.e., no pair of instructions (-,-,q_i,-,R) and (-,-,q_i,-,L) occurs in the program. None of these restrictions will reduce the computation power or speed compared with the Turing machines as introduced above.

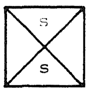
In the tiling problem one is asked whether it is possible to cover a region in the square grid in the Cartesian plane (i.e., a subset of Z×Z), which region may be bounded (e.g., an n×n square or an m×n rectangle) or unbounded (the whole plane, a half plane or a quadrant), using "tiles" from a given finite set, such that specific boundary and adjacency conditions are satisfied. This formulation allows for a great amount of generality. In the sepcific case which we shall consider here, the tiles are represented by unit squares with coloured edges, with colours drawn from a fixed finite alphabet; rotations or reflections of tiles are not allowed. The adjacency conditions stipulate that the tiles placed at adjacent squares (horizontally or vertically) have the same colours on their common edge. The boundary conditions either stipulate that the tiles placed along the boundary of the region to be covered have colours matching with a given colouring along this boundary, or that on specific places specific tiles have to be placed.

It is usual to represent a tile by extending the colour on the edge to the interior of the square, after having divided the square into four parts along the two diagonals. In the sequel we will describe tiles by drawing such figures, and abstain from presenting a formal definition of the tiling problems considered.

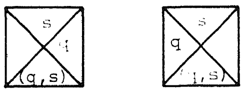
Let there be given a Turing machine with state space K and tape alphabet Σ. We describe a collection of tiles which has the property that a horizontal strip in the plane covered by these tiles may encode an ID of this machine, and such that every extension of this covering to the strip below enforces on this adjacent strip the encoding of an ID resulting from the first one by executing some instruction of the Turing machine program. This set of tiles can also be found in [10].

For the set of colours we take $K \cup \Sigma \cup K \times \Sigma \cup \{w\}$, all summands in this union assumed to be disjoint. The colours in $K \times \Sigma$ represent the tape symbols scanned by the reading head in the indicated state; colours in $K(\Sigma)$ are used to transmit states (symbols) along vertical edges (horizontal strips). The colour w represents the "white" colour, indicating the absence of a colour (in our drawings we omit the symbol w in the corresponding places).

For every tape symbol s there is a tile which represents "preservation" of this tape symbol from one ID to the next one if not disturbed by the reading head:

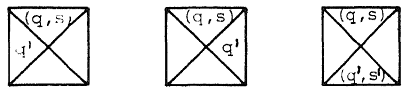


It is also possible that this symbol is scanned by the reading head in the next instruction; the following tiles indicate the accepting of the reading head arriving from an adjacent square:



From the above pair only one tile is present depending on the direction in which the machine moves while going to state q; otherwise two of those tiles might be placed adjacent to each other, creating in this way a pair of "phantom heads" which might destroy the contents of the tape (locally), and next absorb each other again. For this reason we introduced the technical restriction on Turing machines that in no state the machine can move in two directions.

For the symbol scanned by the head an instruction is executed. This is enforced by introducing the following tiles, which correspond to the instructions (q,s,q',s',L), (q,s,q',s',R) and (q,s,q',s',0) respectively:



From the above description it should be evident that the above set of tiles has the required property that a tiling enforces a simulation of a Turing machine computation from one horizontal strip to the next one. The remaining problem is to enforce the establishment of a correct initial ID encoding on some horizontal strip. Here the technique used depends on the precise variant of the tiling problem considered. In this paper we only consider the following variant of the tiling problem:

BOUNDED TILING (called SQUARE TILING in [3]):
INSTANCE: A finite set of tiles; an N×N square with a colouring on the border.
QUESTION: Does there exist a tiling of the entire square, extending the colouring along the border ?

The following fundamental theorem provides a new opening to the theory of NP-completeness.

FUNDAMENTAL THEOREM: BOUNDED TILING is NP-complete.

Proof: Membership of NP is evident. Let A be an arbitrary set in NP and let U be some Turing machine nondeterministically accepting A in time $k.n^k$. Consider the set of tiles which encodes the computations of U as described above. Let x be an arbitrary input string. We transform this string x to the following instance of BOUNDED TILING: let $N = k.|x|^k$; consider the N×N square with the following colouring on its border: on top the initial configuration q_0x is encoded, extended with N-|x| blanco symbols; the left and right border are all white, whereas on the bottom the unique accepting configuration is required (we assume the machine U to be restricted such that it never moves left of the original input, and has a unique accepting configuration). Now a tiling of this square corresponds with an accepting computation, hence x belongs to A if and only if the instance of BOUNDED TILING as described is solvable. It is left to the reader to convince himself that the reduction is P-time computable.

Note that the set of tiles depends on the Turing machine only, whereas the input only determines the boundary conditions to be enforced.

We should emphasize at this point that BOUNDED TILING is just a version of a tiling problem which happens to be complete at the NP-level. By introducing alternative ways of describing the border conditions one easily obtains problems which are complete for other classes of problems. For example a version where the border conditions only describe the top and the bottom row of a rectangle to be tiled, is complete for PSPACE; describing the size of the square to be tiled in binary rather than unary notation makes the problem complete for NEXPTIME. These two observations are again due to Lewis [8]. The origin-constrained unbounded tiling problem, where one wants to tile an entire plane with a prescribed tile at the origin is easily shown to be undecidable (Σ_1^0*-complete*). Harel [4] introduced a recurring variant of the tiling problem where one asks for a tiling of the plane such that a given tile occurs infinitely often, and showed that this variant was Π_1^1*-complete*.

It seems that all the completeness results on tiling problems in the literature are completely elementary and easy to explain except for the difficult case of the undecidability of the unconstrained, unbounded tiling problem which was shown by Berger [1] (Robinson later simplified the proof [9]). Proposals have been put forward for introducing alternating variants as well (which then would turn out to be complete at alternating levels in the polynomial time hierarchy [10]), but these modifications seem to be in contradiction with the simplicity of the basic problem as we use it.

3. Six Basic NP-complete Problems

We will establish the six basic NP-completeness results (Chapter 3 [3]) according to the scheme given in Figure 1. We shall use some intermediate problems which are also among the class of well known NP-complete problems. The proofs presented will be incomplete because we will leave it to the reader to verify both the membership of the class NP for the problems considered and the polynomial boundedness of the reductions. We will only include the proofs of the non-standard reductions. For the others the references given should suffice.

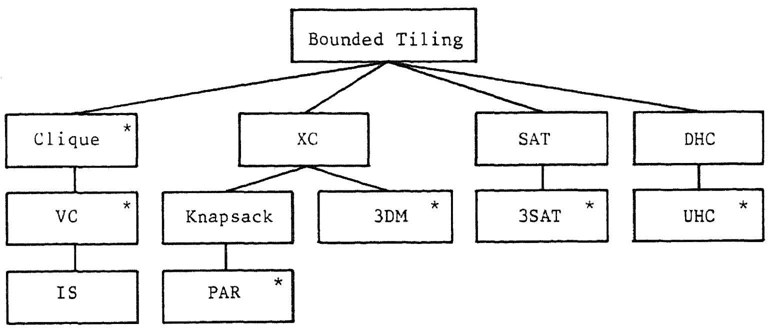


Figure 1: Scheme of reductions (the six basic problems introduced by Garey and Johnson are marked with an asterisk).

3.1. Clique, Vertex Cover and Independent Set

CLIQUE:	Given a graph G=(V,E) and a positive integer J≤|V|, does G contain a clique of size J or more, that is, a subset V'⊆V such that |V|≥J and every two vertices in V' are joined by an edge in E?
VERTEX COVER:	Given a graph G=(V,E) and a positive integer K≤|V|, is there a vertex cover of size K or less for G, that is, a subset V'⊆V such that |V'|≤K and, for each edge (u,v)∈E, at least one of u and v belongs to V'?
INDEPENDENT SET:	Given a graph G=(V,E) and a positive integer K≤|V|, is there an independent set of size K or more for G, that is, a subset V'⊆V, such that |V'|≥K and every two vertices in V' are non-adjacent?

Theorem: BOUNDED TILING <= CLIQUE.

Proof: The proof presented is analogous to the standard reduction of SATSFIABILITY to CLIQUE. We introduce one vertex representing the border (border vertex) and N^2 independent sets of |T| vertices, one for each square of the plane. Each vertex from an independent set represents the possibility of locating one of the |T| tiles on the square which goes with the independent set.

Vertices from an independent set will be connected by an edge to vertices of the independent sets which go with the squares that lie directly left, right, above and beneath if and only if the colours on the border of the tiles match. Furthermore they will be connected to all the vertices of other independent sets. Analogous the border vertex is connected by an edge with the vertices of the independent sets which go with a square directly next to the border if and only if the colours match and with all vertices of other independent sets.

A clique of $1+N^2$ vertices will consist of the border vertex plus N^2 vertices from N^2 (different) independent sets. But that is only possible if there exists a tiling of the plane.

Theorem: CLIQUE <= VERTEX COVER.
Proof: [Karp, 5].

Theorem: VERTEX COVER <= INDEPENDENT SET.
Proof: [Karp, 5].

3.2. Exact Cover, Knapsack, Partition and 3-Dimensional Matching

EXACT COVER: Given a set U and a collection Ξ consisting of subsets F⊆U, does there exist a subcollection Ξ'⊆Ξ such that its members form a partition of U?

KNAPSACK: Given a finite set A and size s(a)∈Z^+ for each a∈A and a number K∈Z^+ does there exist a subset A'⊆A such that

$$\sum_{a\in A} s(a) = K\ ?$$

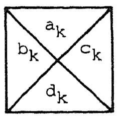
PARTITION: Given a finite set A and a size s(a)∈Z^+ for each a∈A, does there exist a subset A'⊆A such that

$$\sum_{a\in A'} s(a) = \sum_{a\in A-A'} s(a)\ ?$$

3-DIMENSIONAL MATCHING: Given a set M⊆W×X×Y, where W,X and Y are disjoint sets having the same number q of elements, does M contain a matching, that is, a subset M'⊆M such that |M'|=q and no two elements of M' agree in any coordinate?

Theorem: BOUNDED TILING <= EXACT COVER.

Proof: Let HC (VC) denote the collection of horizontal (vertical) colours. All edges in the bounded region along which colours should match are located on N+1 vertical lines of length N and N+1 horizontal lines of length N. The set of these N(N + 1) horizontal (vertical) lines is called HE (VE). For a colour c∈HC let $X_c = \{c\}$ whereas $\bar{X}_c = \text{HC-}\{c\}$. The set Y_c and $\bar{Y}_c$ for the colours c∈VC are defined analogously. The square located at the point with coordinates (i,j) has two horizontal edges $h_{i,j}$ and $h_{i+1,j}$ and two vertical edges $v_{i,j}$ and $v_{i,j+1}$ The collection of tiles which can be used consists of K tiles $T_1,...,T_K$. The colours on the tile T_k are a_k, b_k, c_k and d_k as indicated below.



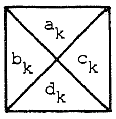
With the above notation we describe the instance of exact cover to which the given instance of BOUNDED TILING will be transformed. The set U is obtained by locating on all edges the colours which may occur along this edge

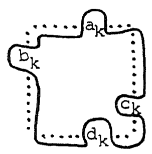
$$U = VE \times VC \cup HE \times HC.$$

The collection Ξ will consist of a family $F_{i,j,k}$ which expresses the collection of coloured points in U which is covered by locating the tile T_k at square (i,j) (0⩽i,j⩽N-1, 1⩽k⩽|T|), and a set F_0 which enforces the given colouring along the border. First we define the set $F_{i,j,k}$:

$$F_{i,j,k} := \{h_{i,j}\} \times X_{a_k} \cup \{h_{i+1,j}\} \times \bar{X}_{d_k} \cup \{v_{i,j}\} \times Y_{b_k} \cup \{v_{i,j+1}\} \times \bar{Y}_{c_k}.$$

We picture a geometrical interpretation below:





With this explanation it should be evident now how the set F_0 should be composed, depending on the colouring of the border. Let the colour of the kth edge-segment of the top of the region be A_k, similarly B_k, C_k and D_k are defined for left, right and bottom border.

$$F_0 = \bigcup_j \left[\{h_{0,j}\} \times \bar{X}_{A_j} \cup \{h_{N,j}\} \times X_{D_j}\right] \cup \bigcup_i \left[\{v_{0,i}\} \times \bar{Y}_{B_i} \cup \{v_{N,i}\} \times Y_{C_i}\right]$$

It should be evident now that the presented proof is correct.

Theorem: EXACT COVER <= KNAPSACK.
Proof: [Karp, 5].

Theorem: KNAPSACK <= PARTITION.
Proof: [Karp, 5].

Theorem: EXACT COVER <= 3-DIMENSIONAL MATCHING.
proof: [Karp, 5]. Note that the reduction of 3SAT to 3DM given by Garey and Johnson [3.1.2 in 3] uses the same components as the reduction given by Karp. We prefer Karp's use of this technique over the more complicated application by Garey and Johnson.

3.3. Satisfiability and 3-Satisfiability

SATIFIABILITY:	Given a collection $C = \{c_1,...,c_m\}$ of clauses on a finite set U of variables, is there a truth assignment for U that satisfies all the clauses in C?

3-SATISFIABILITY: Given a collection $C=\{c_1,...,c_m\}$ of clauses on a finite set U of variables such that $|c_i|=3$ for $1 \leqslant i \leqslant m$, is there a truth assignment for U that satisfies all the clauses in C?

Theorem: BOUNDED TILING <= SATISFIABILITY.

Proof: The presented proof is similar to the one given by Lewis & Papadimitriou [7]. We include the proof because of its elegance; in particular this proof (contrary to the one given in [7]) produces a formula in disjunctive normal form without intermediate transformations. We introduce the literals x_{ijk}, $1 \leqslant i,j \leqslant N$ and $1 \leqslant k \leqslant |T|$, representing the possibility of locating the tile T_k on the square (i,j). We need four types of clauses:

(1) to guarantee that at least one tile will be placed upon each square of the plane:

$$\bigwedge_{i,j} (\bigvee_{k} x_{ijk});$$

(2) to guarantee that exactly one tile will be placed upon each square of the plane:

$$\bigwedge_{i,j} \bigwedge_{k \neq k'} (\bar{x}_{ijk} \vee \bar{x}_{ijk'});$$

(3) to guarantee that the adjacency conditions between tiles are satisfied:

$$\bigwedge_{i,j} \bigwedge_{k,k'} (\bar{x}_{ijk} \vee \bar{x}_{ij+1k'})$$

for all pairs $(T_k, T_{k'})$ which can not occur as a horizontally adjacent pair.

$$\bigwedge_{i,j} \bigwedge_{k,k'} (\bar{x}_{ijk} \vee \bar{x}_{i+1jk'})$$

for all pairs $(T_k, T_{k'})$ which can not occur as a vertically adjacent pair.

(4) to guarantee that the boundary conditions are satified:
$\bar{x}_{1jk}$ for all tiles T_k which can not be placed upon the square (1,j);
$\bar{x}_{Njk}$ for all tiles T_k which can not be placed upon the square (N,j);
$\bar{x}_{i1k}$ for all tiles T_k which can not be placed upon the square (i,1);
$\bar{x}_{iNk}$ for all tiles T_k which can not be placed upon the square (i,N).

Theorem: SATISFIABILITY <= 3-SATISFIABILITY.
Proof: [Karp, 5].

3.4. Directed Hamiltonian Circuit and Undirected Hamiltonian Circuit

Directed Hamiltonian Circuit (DHC):	Given a directed graph H=(V,A), does H have a directed cycle passing through each vertex exactly once.
Undirected Hamiltonian Circuit (UHC):	Given an undirected graph G=(V',E), does G have a cycle passing through each vertex exactly once.

Theorem: BOUNDED TILING <= DHC

Proof: For the colours on the border we construct the digraph on $2N$ vertices pictured in Figure 2 (the bordergraph) and for each of the N^2 squares of the plane we will construct a "choice-component" combining all possible ways to cover the square by one of the tiles $T_k \in T$. "External" arcs will be added to the graph under construction, connecting the border to components or one component to another if and only if the colours match. These external arcs will always be directed in the way pictured in Figure 3. In combination with the bordergraph these directions force a Hamiltonian cycle to enter and leave a choice-component exactly twice.

To establish the relation between a tiling of the plane and a Hamiltonian cycle in the graph under construction, the choice-component should enforce the entering and leaving to take place in a part of the choice-component that represents one and the same tile (thus choosing a tile). A choice-component that will take care of this consists of |T| subgraphs each representing a tile $T_k \in T$ and is pictured in Figure 4.

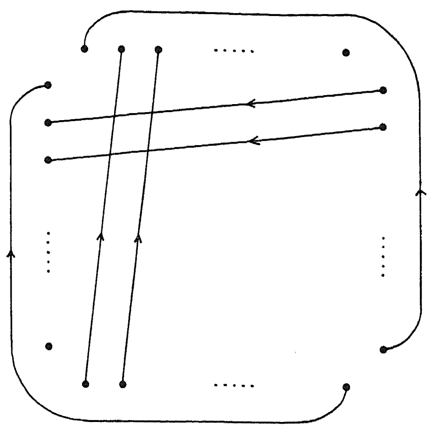


Figure 2: The bordergraph.

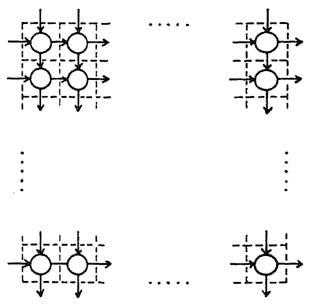


Figure 3: Directions of the "external arcs" (= choice-component).

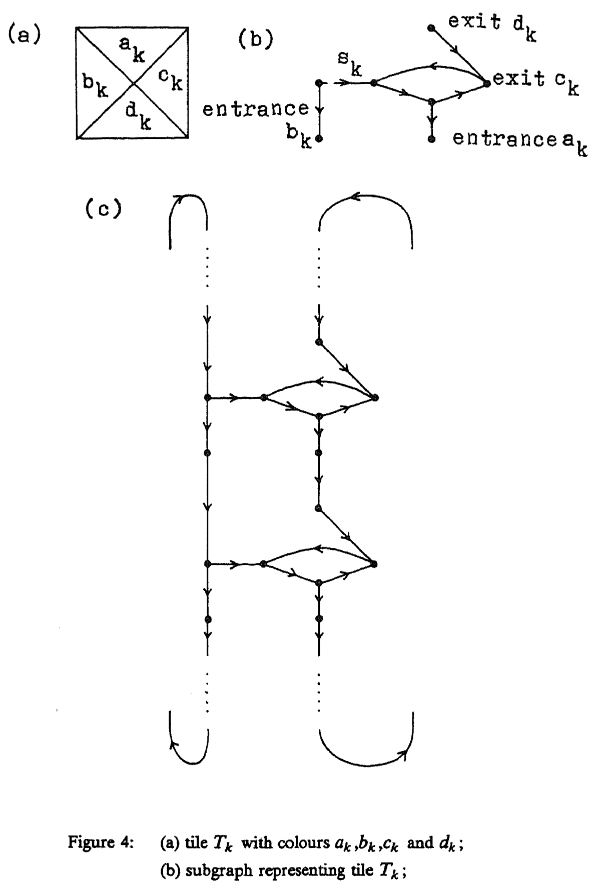
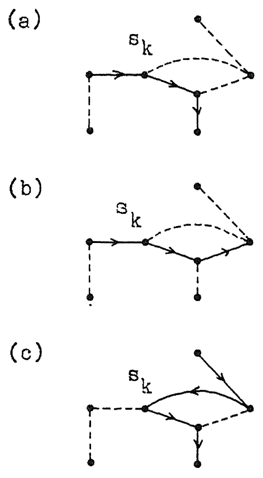


Figure 4: (a) tile T_k with colours a_k, b_k, c_k and d_k;
(b) subgraph representing tile T_k;
(c) choice-component consisting of |T| subgraphs.

We make the following observations:

- The bordergraph and the external arcs force a Hamiltonian cycle to enter every choice-component exactly once in a vertex b_k and once in a vertex $a_{k'}$.
- The only possible way for a Hamiltonian cycle to include all the vertices on the "left" side of the choice-component when entering in vertex b_k is to crossover to the "right" side via s_k.
- There are exactly three possible ways to include a vertex s_k in a Hamiltonian cycle (see Figure 5).

By a simple case analysis we find that choosing option (a) to include s_k will never lead to a Hamiltonian cycle. Therefore traversing a component will only use options (b) and (c). Another close examination now reveals that any Hamiltonian cycle incorporating a choice-component enters and leaves the choice-component in one and the same subgraph representing a tile T_k, thus identifying the tile to be placed on the plane (see Figure 6).



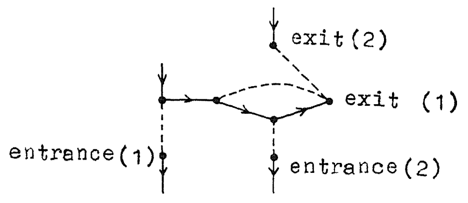


Figure 5: Three ways to include vertex s_k in a Hamiltonian path.

Figure 6: The right way to traverse a component.

From the above presented observations and arguments it should be evident now that a tiling of the plane exists if and only if the constructed graph contains a Hamiltonian cycle.

Theorem: DHC $<=$ UHC
Proof: [Karp, 5].

4. Conclusions

Our purpose in this paper was to experiment with the use of BOUNDED TILING as an alternative "master problem" for the theory of NP-completeness.

In Section 3 we presented a number of reductions, four of which started directly from BOUNDED TILING. The reduction to CLIQUE is essentially the same as the reduction from SATISFIABILITY. For EXACT COVER we see that the reduction has become much easier than the traditional road along 3DM. The reduction to SATISFIABILITY is rather simple, writing down the necessary and sufficient conditions for the existence of a tiling of the plane immediatly gives the desired disjunctive normal form. Finally the reduction to Directed Hamiltonian Cycle. Here we require, as before, an instance of component design. These four reductions all share the fact that it is intuitively clear that the proposed reduction is correct. All that remains is working out the details of the proof, which is not always trivial.

The main advantage of the BOUNDED TILING problem over SATISFIABILITY therefore lies in the problem itself. The conceptual simplicity of the problem, the proof of its NP-completeness and the presented reductions will appeal to a far larger audience. Harel reports on his experience while explaining the Tiling problem to complete novices [4]. Combined with the flexibility as indicated at the end of section 2, which makes the problem useful at various levels of undecidability or intractability, this simplicity makes it a valuable tool for the purpose of introducing the theory of NP-completeness to non-mathematicians.

References

[1] Berger, R.: The undecidability of the domino problem. Mem. Amer. Math. Soc. 66 (1966).

[2] Cook, S.A.: The complexity of theorem-proving procedures. Proc. ACM STOC 3 (1971), 151-158.

[3] Garey, M.R., and D.S. Johnson: Computers and Intractability. W.H. Freeman and Co., San Francisco 1979.

[4] Harel, D.: Recurring dominoes: making the highly undecidable highly understandable, In: Karpinski, M. (ed.): Foundations of Computing Theory. Proc. 1983 FCT Conf. Borgholm, Sweden. Lecture Notes in Comput. Sci., vol. 158. Springer, Berlin, Heidelberg, New York 1983, 177-194.

[5] Karp, R.M.: Reducibility among combinatorial problems. In: Müller, R.E., and J.W. Thatcher (eds.): Complexity of Computer Computations. Plenum Press, New York 1979, 85-103.

[6] Levin, L.A.: Universal'nie Zadachi perbora. Probl. Pered. Inf. IX (1973), 115-116 (Engl. transl.: Universal sequential search problems. Problems of Inform. Transmiss. 9 (1973), 265-266).

[7] Lewis, H.R., and C.H. Papadimitriou: Elements of the Theory of Computation. Prentice Hall, Englewood Cliffs 1981.

[8] Lewis, H.R.: Complexity of solvable cases of the decision problem for predicate calculus. Proc. IEEE FOCS 19 (1978), 35-47.

[9] Robinson, R.M.: Undecidability and non periodicity for tilings of the plane, Invent. Math. 12 (1971), 177-209.

[10] Stockmeyer, L.J.: The polynomial-time hierarchy. Theor. Comput. Sci. 3 (1976), 1-22.

[11] van Emde Boas, P.: Cryptografie en Complexiteit. In: Syllabus Cursus Cryptografie. MC Syllabus 51. Mathematisch Centrum, Amsterdam 1983.