

Colouring Periodic Graphs and Tilings of the Plane

Jessica Paquette, under supervision of Gary MacGillivray

MATH 498, April 2016

Abstract

Doubly periodic (DP) graphs are infinite graphs which exhibit translational symmetries in two dimensions. We discuss k -colourings of DP graphs, and generalize DP graphs to d dimensions. First, we show that every bipartite DP graph permits periodic 2-colourings. Second, we show that the d -fold periodic graphs can be coloured in polynomial time. Finally, we show that k -colouring DP graphs is undecidable by reduction from the domino problem. As a corollary, we show that DP graphs which only admit aperiodic colourings exist.

1 Introduction

A *periodic graph* is a countably infinite graph that exhibits translational symmetries in some number of dimensions. Doubly periodic (DP) graphs, for example, exhibit symmetries in 2D space. Intuitively speaking, DP graphs can be seen as copies of a finite graph, each copy occupying a cell in 2D space. We discuss the k -colourings of periodic graphs.

Decision problems surrounding the colourings of DP graphs are computationally easy in some cases, while undecidable in others. Deciding 2-colouring for DP graphs is polynomial-time [1]. In contrast, extending a 3-colouring of a finite portion of a DP graph to a 3-colouring of the entire graph is undecidable [2]. Colourings of DP graphs can exhibit periodicity or aperiodicity. Thus, another question of interest is when periodic (or aperiodic) colourings exist.

Section 3 proves that every bipartite DP graph admits periodic colourings. In Section 4, we show that 2-colouring d -fold periodic graphs is decidable in polynomial time. Finally, Section 5 proves that k -colouring DP graphs is undecidable by reduction from the domino problem. As a corollary of this, we show that there are DP graphs which only admit aperiodic colourings.

2 Preliminaries

2.1 DP Graphs

Doubly periodic graphs, or *DP graphs* can be constructed by placing countably infinite copies of a finite cell graph G on an infinite two dimensional grid. Each cell can be uniquely determined by a pair of integers i, j . We refer to the specific cell as $G_{i,j}$. Two cells $G_{i,j}$ and $G_{i,k}$ are said to be in the same *row*. We define *column* similarly. Vertices in different cells can be adjacent, but only when their containing cells are adjacent. Two cells $G_{p,q}$ and $G_{r,s}$ are *adjacent* exactly when $p = r \pm 1$ or $q = s \pm 1$. In the former case, we say that $G_{p,q}$ and $G_{r,s}$ are *horizontally adjacent*. In the latter, we call the two cells *vertically adjacent*. If, say $p = r + 1$, then we can say that $G_{p,q}$ is to the *right* of $G_{r,s}$, or we call it $G_{r,s}$'s *right neighbour*. The definitions of left, bottom, and top neighbours are similar; if $q = s + 1$, then $G_{p,q}$ is *above* $G_{r,s}$, and so on.

The vertices of G are labelled v^0, v^1, \dots, v^n . An individual vertex's superscript is known as the *type* of that vertex. We write the vertices of $G_{i,j}$ as $v_{i,j}^0, v_{i,j}^1, \dots, v_{i,j}^n$. This allows us to distinguish between vertices in two different cells.

If two adjacent cells share an edge, say, $G_{i,j}$ and $G_{i+1,j}$, then a "copy" of this edge must appear for every cell and its rightmost neighbour. More specifically, let Γ be a DP graph. If $v_{i,j}^a v_{i+1,j}^b \in E(\Gamma)$, then for all cells $G_{p,q}, G_{p+1,q}$, $v_{p,q}^a v_{p+1,q}^b \in E(\Gamma)$. This implies that any cell of Γ can be "shifted" onto another cell, and the resultant graph will be isomorphic to Γ . It is in this sense that DP graphs exhibit translational symmetry.

The maximum number of cells in the vertical direction intersected by a component of C of Γ is called the *height* of C . We denote the height of C as $H(C)$. The width of C is defined similarly with respect to cells in the horizontal direction. For some vertex v , the specific row of Γ that v lies in is denoted $x(v)$. The specific column is $y(v)$.

2.2 n -fold Periodic Graphs

A d -fold periodic graph generalizes DP graphs to d dimensions, $d > 0$. The cells of a d -fold periodic graph are indexed by d -dimensional vectors. To refer to the cell at index $\vec{x} = [x_1, x_2, \dots, x_d]$, we write $G_{\vec{x}}$, or G_{x_1, x_2, \dots, x_d} . Like DP graphs, if two adjacent cells share an edge, all adjacent cells share an edge. For d -fold periodic graphs, we say that two cells $G_{\vec{a}}, G_{\vec{b}}$ are adjacent if $\vec{b}_i = \vec{a}_i \pm 1$ for some $1 \leq i \leq d$. We call i the *direction* between the two cells. The number of cells intersected in direction i by some component C of Γ is called the *height in direction i* of C . This is denoted as $H_i(C)$. The "column" that some vertex lies in in the w direction is written $w(v)$ for $1 \leq w \leq d$.

2.3 Colourings of Periodic Graphs

Let Z be a colouring of a DP graph Γ . We call Z *periodic* if there is a translation (a, b) which, when applied to any cell of Γ , preserves Z . If Z is periodic and has period (a, b) , then translating Γ over by a cells horizontally and b cells vertically results in a graph isomorphic to Γ which is coloured with Z . It is intuitive to think of such translations as "shifting" Γ by a certain number of cells. Thus, such translations are also referred to as shifts.

Colourings need not be periodic; such translations do not necessarily exist. In this case, we call Z an *aperiodic* colouring. This definition carries over to d -fold periodic graphs, with translations being written (x_1, x_2, \dots, x_d) .

The notion of a minimum period is not well-defined, as we are concerned with more than one dimension. Consider a DP graph and a periodic colouring Z of that graph, and the minimum Z -fixing translation of the form $(p, 0)$. Since p defines the least number of cells we need to shift by horizontally to preserve Z , we call it the *horizontal period* of Z . Similarly, if we consider the minimum Z -fixing translation of the form $(0, q)$, then q is the *vertical period* of Z . We can also define periodicity in terms of more than one dimension. For example, we can talk about minimum (p, q) shifts. The minimum such shift defines the number of cells that the graph must be shifted by diagonally in order to preserve Z . Since (p, q) cannot be reduced to a single number in a meaningful way like for horizontal or vertical shifts, we refer to the translation itself as the minimum diagonal period of Z .

3 Periodic 2-Colourings of Bipartite DP Graphs

The following theorem states that only bipartite DP graphs containing infinite components can admit periodic colourings [3].

Theorem 3.1. [3] *A bipartite DP graph Γ has aperiodic 2-colourings if and only if it has finite components.*

This section shows that every bipartite DP graph Γ has at least one periodic 2-colouring. As a corollary of this result, we produce a bound on the lengths of the various periods of Γ 's 2-colourings. We use Theorem 3.1 to restrict our attention to bipartite DP graphs which contain finite components.

3.1 Periodic Colourings

This section proves the following theorem.

Theorem 3.2. *Every bipartite DP graph admits periodic 2-colourings.*

The proof is in two parts. First, we prove that DP graphs which only contain finite components always admit periodic colourings. Second, we show that DP graphs which contain both finite and infinite components always admit periodic colourings.

Lemma 3.3. *DP graphs which only contain finite components always admit periodic colourings*

Proof. Let Γ be a DP graph whose components are all finite. Since Γ 's components are all finite, the type of each vertex in each component is unique. So then for each pair of vertices u, v of the same type, u and v are disconnected. Since u and v are disconnected, each pair of vertices of the same type can be assigned the same colouring. Then every pair of adjacent cells can be coloured in the same way. From this, we know that, a shift of $(1, 0)$, $(0, 1)$ or $(1, 1)$ all preserve the colouring of Γ . Therefore, Γ admits periodic colourings. ■

There thus exists a colouring Z for which has period 1 horizontally and vertically, and $(1, 1)$ diagonally. Notice that we could assign colours to the components of Γ arbitrarily. To see this, let u and v be vertices of the same type. Since u and v are disconnected, the assignment of one colour to u does not impact the colour of v . This intuition leads to the following lemma.

Lemma 3.4. *DP graphs which contain only finite components can be coloured with any horizontal, vertical, or diagonal period*

Proof. As previously shown, it is possible to assign every cell of Γ the same colouring. Let c be the number of components of G . There are then 2^c possible colourings of G . Let two of these colourings be c_1 and c_2 . Suppose that we wish to colour Γ with horizontal period p . Let S be a sequence consisting of 0s and 1s. Write 0^r to denote a string of r 0s, and 1^s to denote a string of s 1s. Then the sequence S generated by concatenating infinitely many copies of 10^{p-1} is periodic, and has period p . Denote the k -th entry of S as $S[k]$. Now consider each row of Γ , and each cell $G_{i,j}$ in that row. If $S[i] = 0$, then assign $G_{i,j}$ colour c_1 . If, on the other hand, $S[i] = 1$, colour $G_{i,j}$ with c_2 . Then the rows of Γ are coloured in a way that is isomorphic to the sequence S , which has period P . Therefore, the colouring of Γ has horizontal period p . A similar construction allows us to colour Γ with arbitrary vertical and diagonal periods. Therefore, Γ can be coloured with any arbitrary period. ■

As a result of this lemma, the finite components of Γ do not impact the overall periodicity of the colourings of Γ . Therefore, we can ignore the finite components of Γ entirely. This allows us to easily prove Theorem 3.2.

Proof of Theorem 3.2. Let Γ be a bipartite DP graph. Remove all finite components from Γ and let the resultant graph be Γ_1 . Let the finite components of Γ be Γ_2 . Since Γ_1 only contains infinite components, it has at least one periodic colouring Z_1 , with, say, horizontal period p . By the previous lemma, Γ_2 can be 2-coloured with arbitrary period, so it has a colouring Z_2 with horizontal period p . Therefore, $\Gamma_1 \cup \Gamma_2$ has a colouring with period p . A similar argument holds for vertical and diagonal periods. Therefore, Γ admits periodic colourings. ■

3.2 Bounding the Period

We have established that all bipartite DP graphs permit periodic colourings. Now we can proceed to prove an upper bound on the periods of those colourings. We first prove that the statement holds for all DP graphs which contain one kind of infinite component. Following this, we generalize the argument to DP graphs with multiple, non-isomorphic infinite components. By Lemma 3.4 the finite components of a DP graph can be coloured with arbitrary period. Thus, we ignore them and focus on the infinite components.

Let c refer to the number of components of G .

Lemma 3.5. *Every connected bipartite DP graph Γ admits periodic colourings with horizontal and vertical period at most $1 + 2^c$, and a diagonal period of at most $(1 + 2^c, 1 + 2^c)$*

Proof. Let Γ be a bipartite DP graph, and let Z be any 2-colouring of Γ . If G has c components, then there are 2^c unique 2-colourings of G . By the pigeonhole principle, for any $i, j \in \mathbb{Z}$, two of the cells $G_{i,j}, G_{i+1,j}, G_{i+2,j}, \dots, G_{i+1+2^c,j}$ are coloured the same. Therefore, there is a Z -fixing translation of at most $1 + 2^c$ along x . A similar argument shows that there is a Z -fixing translation of at most $1 + 2^c$ along y .

We can add the two translations together to produce a diagonal Z -fixing translation of at most $2(1 + 2^c)$ cells. To see this, first suppose to the contrary that no such translation exists. This cannot be the case, as shifting right $(1 + 2^c)$ cells always preserves Z . Similarly, shifting up $(1 + 2^c)$ cells always preserves Z . Performing both of these actions produces a diagonal shift in Z . Therefore, there must be a diagonal Z -fixing translation of $2(1 + 2^c)$ cells. Now suppose that this translation is not maximum. If this was the case, then shifting up $(1 + 2^c)$ cells and right $2(1 + 2^c)$ cells would not preserve Z , a contradiction. Therefore, a diagonal Z -fixing translation of $2(1 + 2^c)$ cells is the maximum possible diagonal Z -fixing translation. ■

Next, we consider the case where Γ contains more than one infinite component. Let $\text{hort}(H)$ be the minimum horizontal period of a valid 2-colouring of some bipartite graph H . Define $\text{vert}(H)$ and $\text{diag}(H)$ similarly.

Theorem 3.6. *Let Γ be a bipartite DP graph with k infinite components C_1, C_2, \dots, C_k . Then for any periodic 2-colouring Z of Γ all three of the following hold*

- *The horizontal period of Z is at most $\text{lcm}(\text{hort}(C_1), \text{hort}(C_2), \dots, \text{hort}(C_k))$*
- *The vertical period of Z is at most $\text{lcm}(\text{vert}(C_1), \text{vert}(C_2), \dots, \text{vert}(C_k))$*
- *The diagonal period of Z is at most $\text{lcm}(\text{diag}(C_1), \text{diag}(C_2), \dots, \text{hort}(C_k))$*

Proof. By the previous lemma, each of C_1, C_2, \dots, C_k permit periodic colourings. Suppose that G has c components, and c' of them are used in each cell to form C_i . The subgraph of Γ induced by the c components of Γ is also a DP graph, and so we can apply the previous lemma to it. Therefore, the

horizontal period of C_i is at most $1 + 2^{c'}$. We can apply this argument to each of the components. The colouring of C_i repeats every $1 + 2^{c'}$ horizontal cells. The colourings of the other components repeat similarly. Therefore, after a shift of at most $\text{lcm}(\text{hort}(C_1), \text{hort}(C_2), \dots, \text{hort}(C_k))$ horizontal cells, the colourings of each component will match up. A similar argument holds for vertical and diagonal shifts. Thus, we have that the horizontal period of the colouring is at most $\text{lcm}(\text{hort}(C_1), \text{hort}(C_2), \dots, \text{hort}(C_k))$, the vertical period is at most $\text{lcm}(\text{vert}(C_1), \text{vert}(C_2), \dots, \text{vert}(C_k))$, and the diagonal period is at most $\text{lcm}(\text{diag}(C_1), \text{diag}(C_2), \dots, \text{hort}(C_k))$. ■

4 2-Colouring d -fold Periodic Graphs

This section proves that it is possible to decide 2-colourability for d -fold periodic graphs in time polynomial in n . To illustrate the method of proof, we show the result for 3-fold periodic graphs. From there, we prove the result for d -fold periodic graphs. We build on the following result presented in [1].

Theorem 4.1. [1] *If a DP graph Γ contains an odd cycle, then the subgraph of Γ induced by the cells $G_{i,j}$, $0 \leq i < 16n^2$, $0 \leq j < 4n$ contains an odd cycle.*

This implies that one can check whether or not a DP graph is bipartite by constructing a $16n^2 \times 4n$ cell, and searching for an odd cycle within that cell. Therefore, 2-colourability for a DP graph is decidable in polynomial-time with respect to the number of vertices in the cell graph.

Before proceeding, we will state three lemmas which were proved in [1]. These three lemmas provide the basis for the proof of the desired theorem. We re-state them in terms of d -fold periodic graphs. This is permissible because each lemmas rely only on the length of paths between vertices of certain types. The lengths of the required paths do not change by increasing the dimensionality of Γ .

Lemma 4.2. [1] *If there is a path from a vertex u to a vertex v in Γ , then there is a path of length less than n to a vertex with the same type as v .*

This lemma implies that there is always a short path between two vertices.

Lemma 4.3. [1] *Let Γ be a d -fold periodic graph, with cell graph $G_{\vec{x}}$, and let C be some connected component of Γ . If $H_i(C)$ for some direction $1 \leq i \leq d$ is at least n , then C contains vertices u and v and a path $P = (u = v_0, v_1, \dots, v_r = v)$ such that*

1. $r \leq n$,
2. u and v are the same type,
3. $x_i(u) \neq x_i(v)$

This allows us to bound the distance between vertices of the same type in a predictable way. Given some direction x , if $H_x(C) \geq n$ for some component C of Γ , then there is some type of vertex t where if any vertex u has type t , there is a path from u to another vertex v with type t , and $|x(u) - x(v)| = c$ for some constant $0 < c \leq n$.

Lemma 4.4. [1] *If C is an odd cycle in a d -periodic graph Γ , then Γ contains an odd cycle C' with $H_i(C') < 4n$ for any direction i .*

Thus, we can bound the height of a cycle C in a chosen direction to $4n$.

4.1 3-fold Periodic Graphs

To illustrate the method of proof for d dimensions, we will first prove a similar result for 3 dimensions. The idea behind the proof is to merge together cells so that they must contain the cycle, and continue the process in each dimension. At the end, we will have constructed a DP graph for which any cell can be directly queried for an odd cycle. The cells of the constructed DP graph correspond to a set of known cells in the original graph. By constructing the auxiliary graph, we know exactly where to look in the original graph.

For convenience, let the directions of the 3-fold periodic graph be denoted x , y , and z .

Theorem 4.5. *If a 3-fold periodic graph Γ contains an odd cycle, then the subgraph of Γ induced by the cells $G_{i,j,k}$, $0 \leq i < 256n^4$, $0 \leq j < 16n^2$, $0 \leq k < 4n$ contains an odd cycle.*

Proof. Let Γ be a 3-fold periodic graph. We will construct a new 3-fold periodic graph Γ' whose cells consist of $4n$ cells of Γ in the z direction merged together. This implies that the cycle will be contained in the z direction by each cell of Γ' by Lemma 4.4. Let $G_{i,j,k} = \bigcup_{r=4nz}^{4n(z+1)-1} G_{i,j,r}$. If u, v are in adjacent cells, then $uv \in E(\Gamma')$ if and only if

- $uv \in E(\Gamma)$ and
- $z(u) \neq 4nr$ and $z(v) \neq 4nr$ and
- $z(u) \neq 4nr - 1$ and $z(v) \neq 4nr - 1$

Intuitively, we have broken Γ up into chunks that are $4n$ cells deep, with no edges between z -adjacent cells. The discarded edges are guaranteed to never be part of the cycle by Lemma 4.4. Observe that $n' = |V(G')| = 4n^2$.

Next, we apply the same process to Γ' . The resultant graph Γ'' has cells that are $4n' = 16n^2$ tall, with no edges between y -adjacent cells. Then by Lemma 4.4, if an odd cycle exists in Γ , there must also be an odd cycle with height at most $16n^2$. Observe that $n'' = 4n'n' = 64n^4$.

By Lemma 4.4, if Γ'' contains an odd cycle, it also contains an odd cycle fewer than $4n'' = 256n^4$ cells wide. Since there are no edges between y -adjacent

or z -adjacent cells, the cycle must be entirely contained within a single cell of Γ'' . Then the cycle must be contained within the cells $G''_{i,0,0}$, $0 \leq i < 256n^4$. Therefore, the cycle must also be contained within the cells $G_{i,j,k}$, $0 \leq i < 256n^4$, $0 \leq j < 16n^2$, $0 \leq k < 4n$. ■

4.2 d -fold Periodic Graphs

We now generalize the result to d -dimensions. We do this by repeating outlined process above for $i = 1, 2, \dots, d$.

Theorem 4.6. *If a d -fold periodic graph Γ contains an odd cycle, then the subgraph of Γ induced by the cells G_{x_1, x_2, \dots, x_d} , $0 \leq x_1 < 4n, 0 \leq x_2 < 16n^2, \dots, 0 \leq x_d < 2^{2^d} n^{2^d}$ contains an odd cycle.*

Proof. The previous process can be generalized to d dimensions. We state it as follows

- Let $n_0 = n$, and $h_0 = 4n_0$.
- For $i = 1, 2, \dots, d$ construct Γ^i by letting $G_{x_1, x_2, \dots, x_d}^i = \bigcup_{r=4n_{i-1}x_i}^{4n_{i-1}(x_i+1)-1} G_{x_1, x_2, \dots, x_i=r, \dots, x_d}$.
- For each of pair of vertices u, v in adjacent cells, add uv to $E(\Gamma^i)$ if and only if $uv \in E(\Gamma)$, $H_i(u) \neq 4nr$ and $H_i(v) \neq 4nr - 1$ (and vice versa)
- Let $n_i = 4n_{i-1}^2$ and $h_i = 4n_i$.
- If Γ contains an odd cycle, then it must be contained within the cells $G_{\vec{x}}$, $0 \leq x_0 < h_0, 0 \leq x_1 < h_1, \dots, 0 \leq x_n < h_n$.

This process holds by repeated application of Lemma 4.4.

Now observe what happens as n_i and h_i grow.

i	n_i	h_i
0	n	$4n$
1	$4n^2$	$16n^2$
2	$64n^4$	$256n^4$
3	$16384n^8$	$65536n^8$

We hypothesize that at $i = k$ $n_k = 2^{2^k-2} n^{2^k}$ and $h_k = 2^{2^k} n^{2^k}$. We will show that the statement holds for n_{k+1} , and then for h_{k+1} .

By definition of n_i , we know that if $n_k = 2^{2^k-2} n^{2^k}$, then,

$$\begin{aligned}
n_{k+1} &= 4n_k^2 \\
&= 4(2^{2^k-2} n^{2^k})^2 \\
&= 4(2^{2^{k+1}-4} n^{2^{k+1}}) \\
&= 2^{2^{k+1}-2} n^{2^{k+1}}
\end{aligned}$$

Which is exactly as expected, and so $n_i = 2^{2^i-2}n^{2^i}$. Now, we can show that $h_{k+1} = 2^{2^{k+1}}n^{2^{k+1}}$.

$$\begin{aligned} h_{k+1} &= 4n_{k+1} \\ h_{k+1} &= 4(2^{2^{k+1}-2}n^{2^{k+1}}) \\ h_{k+1} &= 2^{2^{k+1}}n^{2^{k+1}} \end{aligned}$$

Which is exactly what we hypothesized. So then $h_i = 2^{2^i}n^{2^i}$. Therefore, if Γ contains an odd cycle, it must also contain an odd cycle in the cells $G_{x_1, x_2, \dots, x_d}, 0 \leq x_1 < 4n, 0 \leq x_2 < 16n^2, \dots, 0 \leq x_d < 2^{2^d}n^{2^d}$. ■

From this, we can restrict the space we search in Γ for an odd cycle to a block of $4n \times 16n^2 \times 256n^4 \times \dots \times 2^{2^d}n^{2^d}$ cells. Note that this bound is still polynomial in n , as the dimension is fixed. However, the overall growth of the cells at every step is double exponential in terms of the dimension of Γ .

5 k -Colouring DP Graphs

The k -colourability of a DP graph was conjectured to be related to tiling the plane in [1]. We prove that this is true. Tiling the plane with a set of Wang dominoes is reducible to 3-colouring for DP graphs.

A *Wang domino* or *Wang tile* is an edge-coloured rectangular tile which cannot be rotated or flipped. Two Wang dominoes can be placed beside each other if their touching sides have the same colour. In general, we can say that for each side x of a Wang domino w_1 and each side y of a Wang domino w_2

$$xRy \text{ iff } x = y$$

This is called the *domino relation*. The question of whether or not a set of Wang dominoes tiles the plane is known to be undecidable [4]. Thus, k -colouring for $k \geq 3$ is undecidable for DP graphs.

The reduction is in two parts. In the first step we show that it is possible to construct a DP graph Γ whose colourings all preserve the domino relation. Each tile has four edge colours, or *names*. We encode each name as a binary number using a sequence of vertices. Edges are places between the cells of Γ so that the names that "touch" between cells have the same colouring.

In the second step, we show that it is possible to encode valid tiles using G . We construct G so that it can only be coloured when its four names correspond to a domino in our desired set. To achieve this, we model a Boolean circuit in G which evaluates to true when the four names match those of some domino. Ultimately, the valid 3-colourings of G model valid dominoes and the valid 3-colourings of Γ , if they exist, model tilings of the plane.

The rest of this section is spent proving the following theorem.

Theorem 5.1. *It is undecidable whether or not a DP graph Γ is k -colourable for $k \geq 3$.*

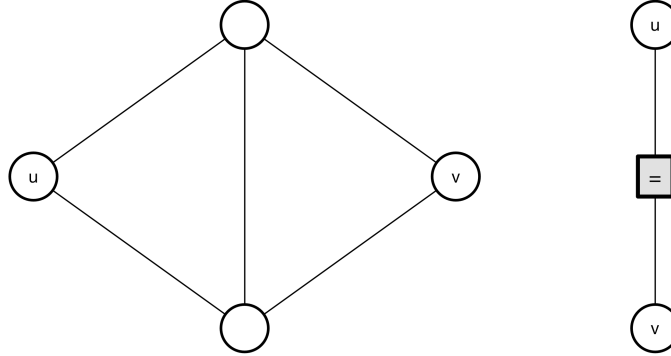
5.1 Constructing Γ : Senders and Evaluators

In order to construct Boolean circuits, and facilitate relationships between cells, we introduce the notion of a *sender* and *evaluator*. Such graphs are shown to exist in [2] and [5].

5.1.1 Senders

A finite graph H is called a (u, v) -*sender* if any valid 3-colouring of H forces u and v to assume the same colour. [2] This allows us to send a "signal" between two vertices u , and v . As a consequence, placing senders within G allows us to enforce that if something is true in one cell, it must be true in a neighbouring cell. The following figure represents a sender.

Figure 1: A (u, v) -sender and the equivalent shorthand for a (u, v) -sender.



This can be easily verified. The two unlabelled vertices cannot be coloured the same as u or v . Thus, the only choices for u and v are the same colour.

5.1.2 Evaluators

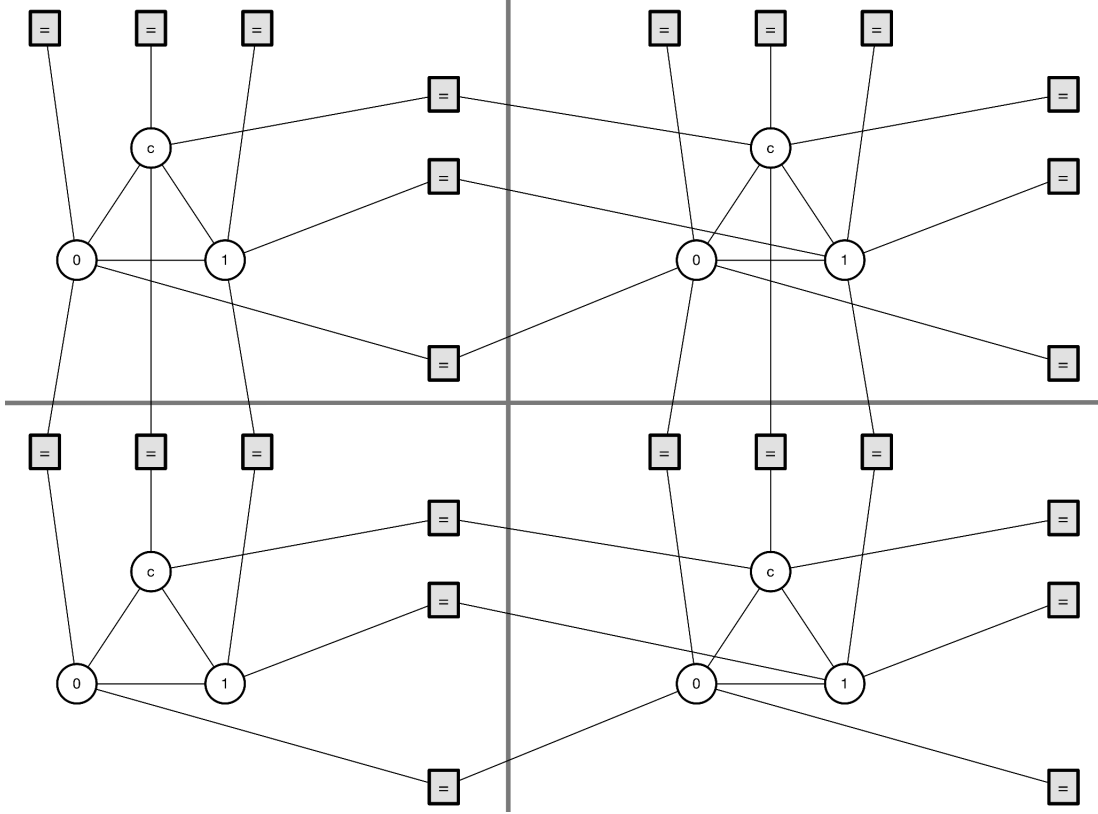
Let ϕ be an r -ary Boolean function. A ϕ -*evaluator* is a finite graph whose vertex set consists of r *input vertices* a_0, a_1, \dots, a_r , and an *output vertex* b . When $a_0 \dots a_r$ are coloured 0 and 1, the colouring of b is forced to assume the result of $\phi(a_0, \dots, a_r)$. Such graphs are shown to exist by Stefan Burr in [5].

In each cell we place a *control structure*. The control structure is a triangle whose vertices are assigned the colours 0, 1, and c . The vertex coloured c is called the *control vertex*. The other two vertices are referred to as *false* and *true* respectively. No vertex other than the control vertex is coloured with c . The control vertex ensures that the rest of the cell agrees on the meaning of true and

false. To represent connections via senders to the control structure, we draw a grey rectangular vertex with labelled the desired colour.

The meaning of true and false is maintained among all cells in Γ by connecting $true_{i,j}$ to $true_{i+1,j}$ and $true_{i,j+1}$ for all cells $G_{i,j}$ via senders. We connect $false_{i,j}$ and $control_{i,j}$ to its neighbours similarly. The following figure illustrates the connections between the control structures of four cells.

Figure 2: The connections between the control structures of four cells



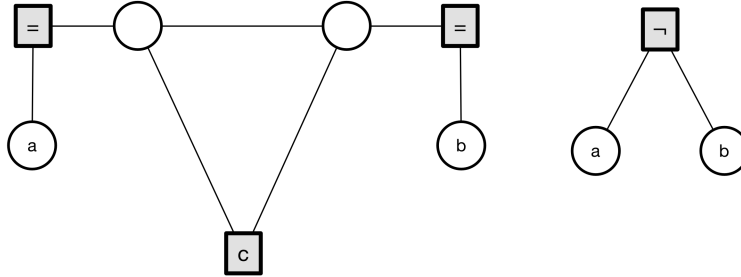
By the construction of a sender, each of the control structures must be coloured the same. Therefore, we can fix a vertex whose colouring always means true, a vertex whose colouring always means false, and a vertex whose colouring is control.

To represent a complete logical system within G , we need only two evaluators: an AND evaluator and a NOT evaluator.

5.1.3 A NOT evaluator

The following graph evaluates the NOT function.

Figure 3: A NOT evaluator which determines $\neg \text{col}(a)$ and outputs in b

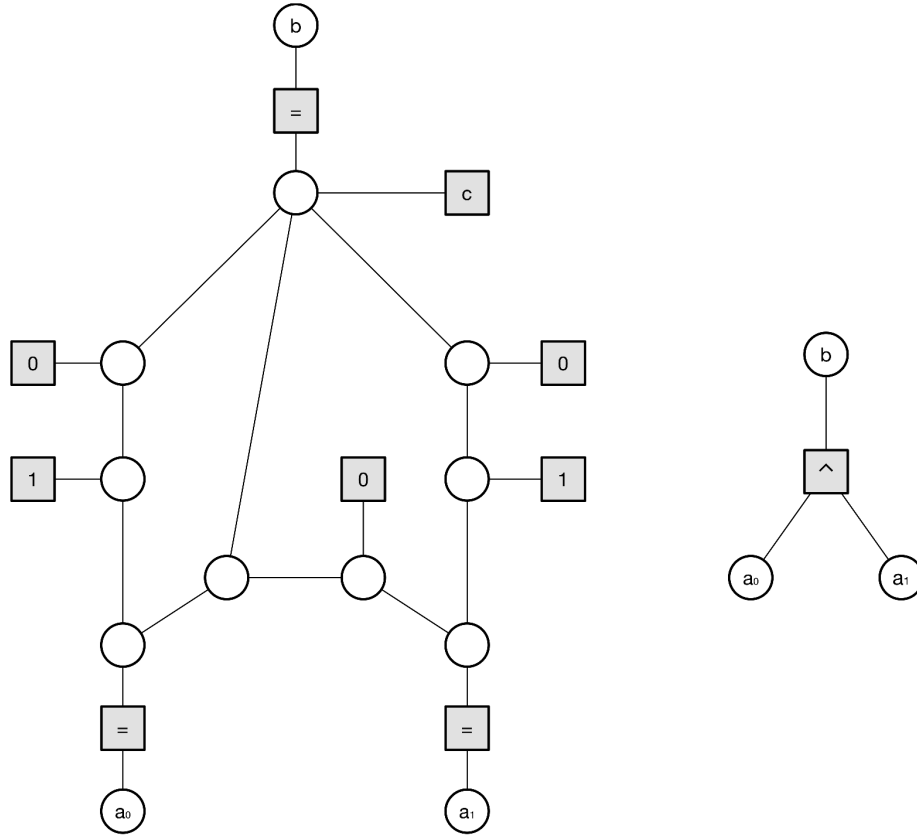


The input vertex a and b can never be coloured the same. Since the bottom vertex is forced to assume the colour c , a and b must assume the opposite truth values.

5.1.4 An AND evaluator

The following graph evaluates the AND function. This graph is presented in [2].

Figure 4: An AND evaluator which determines $\text{col}(a_0) \wedge \text{col}(a_1)$ and outputs the result in the vertex b



5.2 Constructing Γ : Preserving the domino relation

Preserving the domino relation has two requirements. First, we need to encode the names of the dominoes. Second, we need to encode the orientation of the names to disallow rotations and flipping.

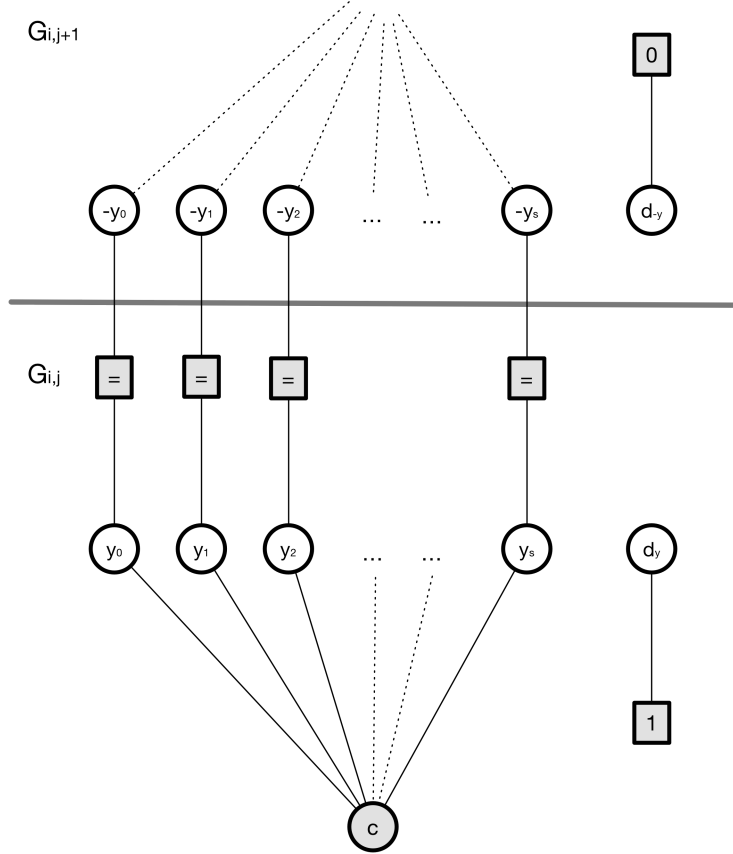
Let W be a set of Wang tiles whose edges can be labelled $1, 2, \dots, s$. We can encode the s names using $\lceil \lg(s) \rceil$ bits. Thus, we add $\lceil \lg(s) \rceil$ vertices for each edge of the tile. Label these vertices x_0, x_1, \dots, x_s for the "right" side, y_0, y_1, \dots, y_s for the "top" side, $-x_0, -x_1, \dots, -x_s$ for the "left" side, and $-y_0, -y_1, \dots, -y_s$ for the "bottom" side. Connect each of these vertices via a sender to the control vertex. Then each of these vertices can be coloured with 0 or 1. At this point, we have encoded each of the s possible names.

Now, we need to satisfy the domino relation between cells. Consider y_0, y_1, \dots, y_s in $G_{i,j}$. We need to ensure that $G_{i,j+1}$ only has valid colourings if its copies of $-y_0, -y_1, \dots, -y_s$ are coloured the same as y_0, y_1, \dots, y_s in $G_{i,j}$. To do this, we connect y_0, y_1, \dots, y_s in $G_{i,j}$ to $-y_0, -y_1, \dots, -y_s$ in $G_{i,j+1}$ via s senders. This enforces that $G_{i,j}$ and $G_{i,j+1}$ only have valid colourings if they agree on their positive and negative y vertices respectively. Therefore, every pair of vertically adjacent cells must agree on its positive and negative y vertices. A similar construction forces horizontally adjacent cells to agree on their positive and negative x vertices.

Finally, we have to enforce that tiles cannot be flipped or rotated. For each side of G , add a *direction vertex*. We label these vertices as d_x, d_{-x}, d_y, d_{-y} depending on their respective sides. The direction vertex of a side is coloured 1 or 0, with "positive" direction vertices being labelled 1, and "negative" direction vertices being labelled 0. This is enforced by connecting the positive direction vertices to *true* via senders, and the negative direction vertices to *false* via senders. By doing this, we force the meaning of "up", "down", "left", and "right" to remain consistent among all cells. This allows us to take the side of a name into account when we evaluate whether or not the colouring of our cell models a valid tile.

The following figure illustrates the connections between two vertically adjacent cells. The connections between two horizontally adjacent cells are defined similarly.

Figure 5: The connections between the positive and negative y vertices of $G_{i,j}$ and $G_{i,j+1}$



5.3 Constructing Γ : Modeling Valid Dominoes

Next, we must ensure that $G_{i,j}$ is only 3-colourable when the colours of its names match up with some valid domino. Using the AND and NOT evaluators, we can construct a Boolean circuit which evaluates to true only when we have a valid tile. We can then force the cell to only have valid colourings when the circuit evaluates to true by connecting its output to $true_{i,j}$ via a sender.

To illustrate the method of constructing the circuit, we'll start with a small example. Suppose that we have a Wang tile whose left side has colour 0, top side has colour 1, right side has colour 2, and bottom side has colour 3. Then we can write the colours as $0 = 00_2$, $1 = 01_2$, $2 = 10_2$, and $3 = 11_2$.

The colours of each side are stored in the vertices $x_0, x_1, -x_0, -x_1, y_0, y_1$, and $-y_0, -y_1$. The physical side of the tile that the name is on is stored in d_{-x}, d_x, d_y and d_{-y} . By construction, d_x and d_y are always coloured with 1, while d_{-x} and d_{-y} are always coloured with 0. There are $\binom{4}{2} = 6$ combinations

of two sides of any tile. So, we can compare the combinations of the names of our colourings with the combinations of the names of some given tile. If they all match up exactly, then we must have the tile. So, a Boolean function which evaluates the tile $(0, 1, 2, 3)$ is

$$((\neg d_{-x} \wedge (\neg -x_0 \wedge \neg -x_1)) \quad \wedge (d_x \wedge (x_0 \wedge \neg x_1))) \quad \wedge \quad (1)$$

$$((\neg d_{-x} \wedge (\neg -x_0 \wedge \neg -x_1)) \quad \wedge (d_y \wedge (\neg y_0 \wedge y_1))) \quad \wedge \quad (2)$$

$$((\neg d_{-x} \wedge (\neg -x_0 \wedge \neg -x_1)) \quad \wedge (\neg d_{-y} \wedge (\neg y_0 \wedge \neg y_1))) \quad \wedge \quad (3)$$

$$((d_y \wedge (\neg y_0 \wedge y_1)) \quad \wedge (\neg d_{-y} \wedge (\neg y_0 \wedge \neg y_1))) \quad \wedge \quad (4)$$

$$((d_y \wedge (\neg y_0 \wedge y_1)) \quad \wedge (d_x \wedge (x_0 \wedge \neg x_1))) \quad \wedge \quad (5)$$

$$((d_x \wedge (x_0 \wedge \neg x_1)) \quad \wedge (\neg d_{-y} \wedge (\neg y_0 \wedge \neg y_1))) \quad (6)$$

Lines (1) through (3) ensure that if the left side of the tile is coloured with 00_2 , then the right side of the tile must be coloured with 10_2 , the top side must be coloured with 01_2 , and the bottom side must be coloured with 11_2 . Lines (4) and (5) ensure that if the top side is coloured with 01_2 , then the bottom side must be coloured with 00_2 , and the right side must be coloured with 10_2 . Finally, line 6 ensures that if the right side is coloured with 10_2 , then the bottom side must be coloured with 00_2 . This exhausts every combination of sides in the tile, and so it is satisfied if and only if the current tile is exactly as described.

For each of the tiles in our set, we construct Boolean functions like above. In general, let $LNAME$ denote the name on the left of the desired tile, $UNAME$ denote the name on the top, $RNAME$ denote the name on the right, and $BNAME$ denote the name on the bottom. Then the following Boolean function evaluates to true if and only if the sides match those of the tile exactly:

$$((\neg d_{-x} LNAME) \quad \wedge (d_x \wedge RNAME)) \quad \wedge \quad (7)$$

$$((\neg d_{-x} LNAME) \quad \wedge (d_y \wedge UNAME)) \quad \wedge \quad (8)$$

$$((\neg d_{-x} LNAME) \quad \wedge (\neg d_{-y} BNAME)) \quad \wedge \quad (9)$$

$$((d_y \wedge UNAME) \quad \wedge (\neg d_{-y} BNAME)) \quad \wedge \quad (10)$$

$$((d_y \wedge UNAME) \quad \wedge (d_x \wedge RNAME)) \quad \wedge \quad (11)$$

$$((d_x \wedge RNAME) \quad \wedge (\neg d_{-y} \wedge BNAME)) \quad (12)$$

Say we have t tiles. Then we will have t separate functions T_1, T_2, \dots, T_t such that T_i evaluates to true if and only if the current colouring represents T_i . By the previous construction of AND and NOT gates, it is possible to construct these functions. Our final desired function is

$$\phi = T_1 \vee T_2 \vee \dots \vee T_t$$

Which evaluates to true if and only if the colouring represents exactly one tile. Since $\{\wedge, \neg\}$ forms a complete logical system, ϕ is constructible. Each cell contains a copy of ϕ which takes as inputs the names and directions of the cells.

In order to make sure that $G_{i,j}$ is only colourable when ϕ evaluates to true, we connect the output vertex of ϕ via a sender to $true_{i,j}$.

Thus, we have constructed a graph Γ which is only colourable when the colourings of each cell represent a valid tile from a set of Wang tiles. Each of the colourings of the cells obey the domino relation, and the orientation is fixed. Therefore, Γ can only be coloured if the set of Wang tiles it has been constructed for tiles the plane. Deciding whether or not tiling the plane is undecidable, so colouring Γ is undecidable. Therefore, 3-colouring, and thus k -colouring DP graphs is undecidable. ■

There are several well-known sets of Wang tiles which tile the plane only aperiodically [6]. By our reduction, we obtain the following corollary.

Corollary 5.2. *There exist 3-colourable DP graphs which only admit aperiodic colourings.*

Since k -colouring DP graphs for $k \geq 3$ is undecidable, k -colouring d -fold periodic graphs for $k \geq 3$ is also undecidable. From this we can state the following corollary.

Corollary 5.3. *Every tiling problem can be reformulated as a problem of deciding 3-colourability for a periodic graph.*

6 Concluding Remarks

Every bipartite DP graph admits periodic colourings, but this does not hold for all DP graphs in general. It is still possible to colour bipartite d -fold periodic graphs in time polynomial in n , but double-exponential in d . For $k > 2$, k -colouring DP graphs is undecidable. From this, we can infer that k -colouring d -fold periodic graphs for $d > 2$ is also undecidable. As a result of the undecidability proof, every tiling problem using square dominoes can be reformulated as a colouring problem for periodic graphs.

The periods of the colourings of bipartite DP graphs is bounded by the lcm of its components' colourings' periods. However, a lower bound is not currently known. There may be promise in exploring certain subclasses of DP graphs and examining the lower bounds of their colourings' periods. For example, it may be simple to bound the colouring periods of bipartite DP graphs which are constructible by unioning together DP graphs whose cell graphs are connected. If the cell graph is guaranteed to be connected, then each cell only permits two possible colourings. Therefore, for a single component, the horizontal and vertical periods are guaranteed to both be 1.

It may also be of interest to explore whether or not all DP graphs which possess only aperiodic colourings are related to tiles. The graph construction presented for the reduction from tiling is likely not minimal. Thus, it may be interesting to find a lower bound on the size and order of a cell graph which only exhibits aperiodic colourings would be.

References

- [1] Gary MacGillivray Bruce L. Bauslaugh. 2-colouring doubly-periodic graphs. *Bulletin of the ICA*, 45:63–69, 2005.
- [2] Stefan A. Burr. Some undecidable problems involving the edge-coloring and vertex-coloring of graphs. *Discrete Mathematics*, 50:171 – 177, 1984.
- [3] Bethany Joy-Smith. Components and colourings of singly and doubly-periodic graphs. Master’s thesis, University of Victoria, 2010.
- [4] Robert Berger. *The Undecidability of the Domino Problem*. Memoirs ; No 1/66. American Mathematical Society, 1966.
- [5] Stefan A. Burr and Vojtěch Rödl. *Mathematics of Ramsey Theory*, chapter On the Computational Complexity of Ramsey—Type Problems, pages 46–52. Springer Berlin Heidelberg, Berlin, Heidelberg, 1990.
- [6] Karel Culik II. An aperiodic set of 13 wang tiles. *Discrete Mathematics*, 160(1–3):245 – 251, 1996.