Cellular automata for the self-stabilisation of colourings and tilings

Nazim Fatès, Irène Marcovici, Siamak Taati

Cellular automata for the self-stabilisation of colourings and tilings

Nazim Fatès[*], Irène Marcovici [†]and Siamak Taati[‡]

November 26, 2019

Abstract

We examine the problem of self-stabilisation, as introduced by Dijkstra in the 1970's, in the context of cellular automata stabilising on k-colourings, that is, on infinite grids which are coloured with k distinct colours in such a way that adjacent cells have different colours. Suppose that for whatever reason (e.g., noise, previous usage, tampering by an adversary), the colours of a finite number of cells in a valid k-colouring are modified, thus introducing errors. Is it possible to reset the system into a valid k-colouring with only the help of a local rule? In other words, is there a cellular automaton which, starting from any finite perturbation of a valid k-colouring, would always reach a valid k-colouring in finitely many steps? We discuss the different cases depending on the number of colours, and propose some deterministic and probabilistic rules which solve the problem for $k \neq 3$. We also explain why the case $k = 3$ is more delicate. Finally, we propose some insights on the more general setting of this problem, passing from k-colourings to other tilings (subshifts of finite type).

Edited in : *International Conference on Reachability Problems (RP 2019)*, pp. 121-136, Springer. Pleaser refer to the editor's site for the final version :
`https://doi.org/10.1007/978-3-030-30806-3_10`

[*] Inria Nancy, Université de Lorraine, CNRS, LORIA, F-54000, France, `nazim.fates@loria.fr`

[†] IECL, Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

[‡] Bernoulli Institute, University of Groningen, Groningen, The Netherlands – The work of ST was partially supported by NWO grant 612.001.409 of Tobias Müller.

1 Introduction

Self-stabilisation is a property omnipresent in biological systems. Indeed, living cells always need to correct their defects in order to keep their behaviour as stable as possible (see e.g. Ref. [2]). The study of self-stabilisation in computational systems was proposed by Dijkstra [4]. The objective is to incorporate self-stabilisation in discrete parallel models of computation.

In the present article, we explore the phenomenon of self-stabilisation in the context of two-dimensional cellular automata which operate on k-colourings. To illustrate the problem, imagine that an artist has a plan to create a two-dimensional tiling with the constraint that two adjacent tiles necessarily bear different colours. When this tiling is realised, the artist realises that a) some mistakes have occurred during the tiling process and b) the original tiling plan has been lost. The question is to know whether it is possible to correct the tiling to respect the constraints of non-adjacency of colours only by following local rules. In other words, we can reformulate the question as a reachability problem: given a set of admissible states of the system, under which conditions is this set always reachable from the set of its finite perturbations?

The problem of designing self-correcting or self-stabilising cellular automata has been explored since the 1970's. Two main models of errors have been considered: a) the errors can happen at each time step and are thus concurrent with the correction process [7, 8, 11, 12] or, b) the errors are present at the beginning and are then corrected [6]. Pippenger has studied this latter question, for the binary case, where the configurations to correct are those which only contain a unique colour [10]. He has shown that the problems can have positive or negative answers depending on the specification of the problem such as dimension, symmetry constraints, etc.

We re-examine this problem in the setting of k-colourings. The cases $k = 2$ and $k \geq 5$ are the simplest and the case $k = 4$ can be dealt with rather easily. However, the case $k = 3$ is much more delicate. We will also explore the question of symmetries of the rules we use. The case of k-colourings should be considered as a first step towards a wider view of self-stabilisation in cellular automata. We indicate some directions on how to consider more general tiling constraints.

2 Setting of the problem

Let Σ be a finite set that represents the different colours of the tiling. Given two configurations $x, y \in \Sigma^{\mathbb{Z}^2}$, we write $\Delta(x, y) \triangleq \{i \in \mathbb{Z}^2 : x_i \neq y_i\}$ for the set of sites at which x and y disagree.

A *finite perturbation* of a configuration $x \in \Sigma^{\mathbb{Z}^2}$ is a configuration $y \in \Sigma^{\mathbb{Z}^2}$ such that $\Delta(x, y)$ is finite. Given a set $\Lambda \subseteq \Sigma^{\mathbb{Z}^2}$, representing the set of *valid* configurations, we denote by $\tilde{\Lambda}$ the set of finite perturbations of the elements of Λ, that is:

$$\tilde{\Lambda} \triangleq \{y \in \Sigma^{\mathbb{Z}^2} : \exists x \in \Lambda, \Delta(x, y) \text{ is finite}\}.$$

Our goal is to find a parallel procedure acting in a local way that would, from any element of $\tilde{\Lambda}$, reach and stabilise on an element of Λ in a finite number of steps. The locality of the rule is expressed by the definition of a *neighbourhood*, that is, an ordered list $\mathcal{N} = (n_1, \dots, n_k)$ of k elements from $\mathbb{Z}^2$, for some $k \in \mathbb{N}$. We use the model of cellular automata to take into account the distributed aspect of the process: each cell $c \in \mathbb{Z}^2$ is updated according to a *local rule* f that depends only on the states of the cells $c + n_1, \dots, c + n_k$.

Formally, a two-dimensional *cellular automaton (CA)* with neighbourhood $\mathcal{N}$ is a mapping $F : \Sigma^{\mathbb{Z}^2} \to \Sigma^{\mathbb{Z}^2}$ for which there exists a function $f : \Sigma^k \to \Sigma$ satisfying:

$$\forall c \in \mathbb{Z}^2,\ F(x)_c = f(x_{c+n_1}, \dots, x_{c+n_k}).$$

Now that we have set all the elements, we can define the notion of *self-stabilisation*. We say that a cellular automaton $F : \Sigma^{\mathbb{Z}^2} \to \Sigma^{\mathbb{Z}^2}$ is *self-stabilising* on $\Lambda \subseteq \Sigma^{\mathbb{Z}^2}$ if it satisfies the following conditions:

(i) The configurations of Λ are fixed points of F: $\forall x \in \Lambda,\ F(x) = x$.

(ii) The configurations of $\tilde{\Lambda}$ evolve to Λ in finitely many steps: $\forall y \in \tilde{\Lambda},\ \exists t \in \mathbb{N},\ F^t(y) \in \Lambda$.

We will in particular focus on the case where the set Λ is the set of colourings of $\mathbb{Z}^2$ with k distinct colours. Let $k \geq 2$ be an integer, and let the set $\Sigma = \{0, \dots, k-1\}$ represent the set of possible colours of the cells. We define the set of *k-colourings* of $\mathbb{Z}^2$ by:

$$\Lambda_k \triangleq \{x \in \Sigma^{\mathbb{Z}^2} : c, c' \in \mathbb{Z}^2,\ ||c - c'||_1 = 1 \implies x_c \neq x_{c'}\}.$$

Our aim is to examine if there exist simple self-stabilising rules, depending on the value of k. We also have a look at other families of *subshifts of finite type*, that is, sets Λ defined by local constraints. More specifically, a nonempty set $\Lambda \subseteq \Sigma^{\mathbb{Z}^2}$ is a *subshift of finite type (SFT)* if there exists a finite set $B \subseteq \mathbb{Z}^2$ and a function $u : \Sigma^B \to \{0, 1\}$ such that:

$$\Lambda = \{x \in S^{\mathbb{Z}^d} : \forall c \in \mathbb{Z}^d, u((x_{c'})_{c' \in c+B}) = 1\}.$$

In the definition above, the function u describes the set of allowed (image 1) and forbidden (image 0) patterns of base B.

We will focus on cases where the SFT can be defined in terms of horizontal and vertical constraints. We will call the elements of such an SFT a *proximity tiling*. Formally, let us denote by (e_1, e_2) the standard basis of $\mathbb{Z}^2$. A nonempty set $\Lambda \subseteq \Sigma^{\mathbb{Z}^2}$ is a *proximity tiling space* if there exist functions $v_1, v_2 : \Sigma^2 \to \{0, 1\}$ such that:

$$\Lambda = \{x \in \Sigma^{\mathbb{Z}^2} : \forall c \in \mathbb{Z}^2, v_1(x_c, x_{c+e_1}) = v_2(x_c, x_{c+e_2}) = 1\}.$$

Note that the notion of proximity tilings we have introduced here is reminiscent of tilings by Wang tiles, but this formalism is more adapted to our context. For example, the set of k-colourings is simply the proximity tiling space defined by the function $v = v_1 = v_2$ where $v(a, b) = 1$ if $a \neq b$, and 0 if $a = b$.

In designing self-stabilising rules and proving their correctness, we will often examine the set of cells where the constraints are not respected. We thus introduce different notions of *error*. For a configuration $x \in \Sigma^{\mathbb{Z}^2}$, a cell $c \in \mathbb{Z}^2$ is said to have an e_i*-error* (with respect to v_i) if $v_i(x_c, x_{c+e_i}) = 0$. It has a $(-e_i)$*-error* if $v_i(x_{c-e_i}, x_c) = 0$. We will also use the terminology *E-error, W-error, N-error, S-error* instead of respectively e_1-error, $(-e_1)$-error, e_2-error, and $(-e_2)$-error. The set of cells having an error is defined by:

$$\mathcal{E}(x) \triangleq \{c \in \mathbb{Z}^2 : \exists e \in \{\pm e_1, \pm e_2\},\ c \text{ has an } e\text{-error}\}.$$

A cell $c \in \mathbb{Z}^2$ is said to be *error-free* if it does not belong to $\mathcal{E}(x)$, meaning that it obeys the local constraints in the four directions. Note that in somes cases, even if $\mathcal{E}(x)$ contains only very few cells, it is necessary to modify a much larger set of cells in order to reach a valid configuration (see Prop. 6.1).

We will also consider self-stabilising probabilistic CA. For *probabilistic CA*, the outcome of the local rule is a probability distribution on Σ, and the cells of the lattice are updated simultaneously and independently at each time step, according to the distributions prescribed by the local rule. The local rule in this case is given by a function $\varphi : \Sigma^k \to \mathcal{P}(\Sigma)$, where $\mathcal{P}(\Sigma)$ denotes the set of probability distributions on Σ. The probabilistic CA Φ defined by φ maps a configuration x to a probability measure μ, where for each finite set $C \subseteq \mathbb{Z}^2$, we have

$$\mu\big(\{y : \forall c \in C, y_c = v_c\}\big) = \prod_{c \in C} \varphi(x_{c+n_1}, x_{c+n_2}, \ldots, x_{c+n_k})(\{v_c\}).$$

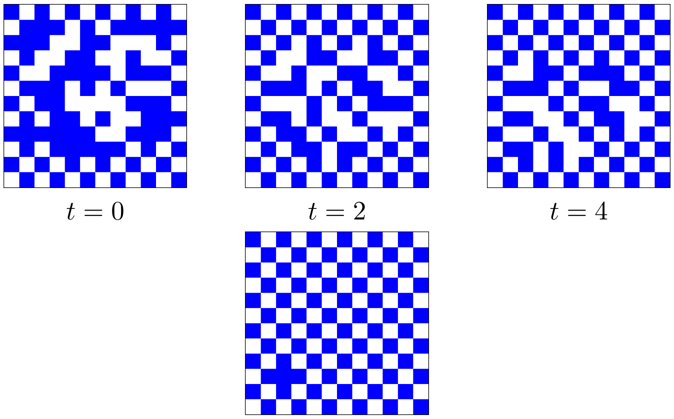
The trajectory of a probabilistic CA Φ with initial configuration x is thus a Markov process $X^0, X^1, \ldots$ with $X^0 = x$ such that, for every $t > 0$, conditioned on the value of the configurations $X^0, X^1, \ldots, X^{t-1}$, the configuration X^t is distributed according to the measure $\Phi(X^{t-1})$. We say that a probabilistic CA Φ is *self-stabilising* on Λ if:

(i) The configurations of Λ are left unchanged by Φ: $\forall x \in \Lambda, \Phi(x) = \delta_x$.

(ii) For every $y \in \tilde{\Lambda}$, there exists a finite (random) time T such that $X^T \in \Lambda$ almost surely.

3 The case of 2-colourings

In this section, we study the self-stabilisation problem for 2-colourings. We thus set $\Sigma = \{0, 1\}$, and consider the set Λ_2. Note that Λ_2 contains only two elements, corresponding to the two (odd and even) chequerboard configurations.



$t = 0$ $t = 2$ $t = 4$

$t = 10$

Figure 1: Evolution of the cellular automaton for self-correction of the 2-colourings.

3.1 Directional self-stabilisation by a deterministic CA

Let us define a cellular automaton F on $\Sigma^{\mathbb{Z}^2}$ by:

$$\forall c \in \mathbb{Z}^2, \quad F(x)_c = \begin{cases} 1 - x_c & \text{if } x_c = x_{c+e_1} = x_{c+e_2}, \\ x_c & \text{otherwise.} \end{cases}$$

The rule above is similar to the well-known majority rule of Toom used to correct errors that appear on a uniform background [3,12].

Proposition The cellular automaton F defined above is self-stabilising on Λ_2.

Proof. It is clear from the definition that $\forall x \in \Lambda_2,\ F(x) = x$.

For each $n \in \mathbb{N}$, define the triangle $T_n = \{(i,j) \in \mathbb{Z}^2 : i + j \leq n,\ i, j \geq 0\}$ on the grid. Let $x \in \Lambda_2$ (recall that x is thus a chequerboard configuration) and take $y \in \tilde{\Lambda}_2$ such that $\Delta(x, y)$ is finite. By translating x and y if needed, we can assume without loss of generality that the difference set $\Delta(x, y)$ is included in the triangle T_n for some n. It is then easy to verify that $\Delta\big(x, F(y)\big) \subseteq T_{n-1}$. Indeed, for every cell outside T_n, the local rule does not modify the state, whereas for the cells (i, j) which are inside T_n and satisfy $i + j = n$, we have $F(y)_{i,j} = x_{i,j}$. Iterating F we obtain $\Delta\big(x, F^t(y)\big) \subseteq T_{n-t}$ for each $t \geq 0$. That is, as time goes by, the set of disagreements becomes smaller (see Fig. 1). In particular, for $t = n+1$, we get $\Delta\big(x, F^{n+1}(y)\big) \subseteq T_{-1} = \varnothing$, hence $F^{n+1}(y) = x \in \Lambda_2$. This means that the configuration y has been corrected in $n + 1$ steps. □

3.2 Isotropic self-stabilisation by a probabilistic CA

The cellular automaton F above provides a directional solution: the cells need to distinguish the North and East directions. In the context of deterministic CA, Pippenger has shown that requiring all the symmetries for the local rule can lead to negative results [10]. By contrast, we now propose a probabilistic CA that achieves the self-stabilisation with an isotropic rule, that is, a rule which treats the neighbours "equally" and does not distinguish between the four directions of the grid. This shows that the use of randomness can extend the range of possibilities. More precisely, the rule we propose consists in applying a minority function with probability α, and keeping the state unchanged with probability $1-\alpha$. Such rules are called *α-asynchronous* and their study has received a continuous attention in the last years [5] ; this structure is here used to get out of the potential cyclic behaviours that would prevent the system from reaching the desired stable configurations.

Formally, let $\mathcal{N}$ denote the *von Neumann neighbourhood* $\mathcal{N} = (0, e_1, e_2, -e_1, -e_2)$. We define a probabilistic cellular automaton Φ on $\Sigma^{\mathbb{Z}^2}$ by the local rule $\varphi : \Sigma^5 \to \mathcal{P}(\Sigma)$ given by:

$$\varphi(q_0, q_1, \ldots, q_4) = \alpha\delta_{\text{minority}\,(q_0,q_1,\ldots,q_4)} + (1-\alpha)\delta_{q_0},$$

where δ_q is the Dirac measure on q, meaning that $\delta_q(\{q'\}) = 1$ if $q = q'$, and 0 otherwise, and where minority (a, b, c, d, e) equals 1 if $a + b + c + d + e \leq 2$ and 0 otherwise.

Proposition For $\alpha \in (0, 1)$, the probabilistic cellular automaton Φ is self-stabilising on Λ_2.

Proof. Let us take $x \in \Lambda_2$ (recall that x is thus a chequerboard configuration) and $y \in \tilde{\Lambda}_2$ such that $\Delta(x, y)$ is finite. Let $X_0, X_1, \ldots$ be the Markov process described by Φ with initial configuration $X_0 = y$. Let R be a rectangle such that $\Delta(x, y) \subseteq R$. For any $c \notin R$, we have $\varphi(x_c, x_{c+e_1}, x_{c+e_2}, x_{c-e_1}, x_{c-e_2}) = \varphi(y_c, y_{c+e_1}, y_{c+e_2}, y_{c-e_1}, y_{c-e_2}) = \delta_{x_c}$, so that for all $t \geq 0$, $\Delta(X_t, x) \subseteq R$ almost surely. Furthermore, inside R, Φ behaves like an absorbing finite-state Markov chain that eventually reaches the chequerboard configuration $(x_c)_{c \in R}$. Note that from any state, with positive probability the chequerboard configuration can be reached in at most $|R|$ steps. This is because $\alpha < 1$. Otherwise, a monochromatic rectangle could blink between the two states all 0's and all 1's. □

3.3 Extension to finite SFT

The methods presented above for the case $k = 2$ can be readily extended to all the cases where Λ is a subshift of finite type that contains only a finite number of configurations (with an arbitrary set of symbols Σ). Indeed, the configurations of such a finite SFT are necessarily spatially periodic.

Let us consider an arbitrary finite SFT Λ on $\Sigma = \{0, \dots, k-1\}$, for some $k \geq 1$. Then, for each configuration $x \in \Lambda$, there exist integers $m, n \geq 1$ such that ${\sigma_{\mathrm{h}}}^m(x) = {\sigma_{\mathrm{v}}}^n(x) = x$, where σ_{h} and σ_{v} denote the horizontal and vertical shift maps. Taking the least common multiple of the collection of integers obtained for the different configurations $x \in \Lambda$ (all these integers are bounded by the cardinality of Λ), we can find horizontal and vertical periods $p_{\mathrm{h}}, p_{\mathrm{v}} \geq 1$ such that $\forall x \in \Lambda,\ {\sigma_{\mathrm{v}}}^{p_{\mathrm{v}}}(x) = {\sigma_{\mathrm{h}}}^{p_{\mathrm{h}}}(x) = x$. This means that the elements of Λ are constant on every sublattice $\mathbb{L}_{a,b} = \{(a+p_{\mathrm{h}}i, b+p_{\mathrm{v}}j) : i, j \in \mathbb{Z}^2\}$. Therefore, we can simply use Toom's majority rule on each sublattice, that is, we define a cellular automaton F on $\Sigma^{\mathbb{Z}^2}$ by:

$$F(x)_{a,b} = \mathrm{majority}(x_{a,b}, x_{a+p_{\mathrm{h}},b}, x_{a,b+p_{\mathrm{v}}}),$$

where the majority function associates to three symbols the symbol which is most present in this three symbols, with the convention that when the three symbols are distinct, one can choose arbitrarily the value of the function. Note however that even if Λ is a proximity tiling, the neighbourhood of the cellular automaton F depends on the periods p_{h} and p_{v} and can be much larger than 1.

We can also design an isotropic probabilistic rule that corrects finite SFT. If a state appears strictly more than twice among $x_{a+p_{\mathrm{h}},b}, x_{a,b+p_{\mathrm{v}}}, x_{a-p_{\mathrm{h}},b}, x_{a,b-p_{\mathrm{v}}}$, this state becomes the new value of $x_{a,b}$. Otherwise, we randomly choose a new state in the alphabet Σ. Again, all the errors stay within some enveloping rectangle and are eventually corrected. We can replace both p_{h} and p_{v} by $LCM(p_{\mathrm{h}}, p_{\mathrm{v}})$ in order to have an isotropic rule.

4 The case of k-colourings, for $k \geq 5$

We now consider the case $k \geq 5$. Recall that we have $\Sigma = \{0, 1, \dots, k-1\}$.

4.1 Directional self-stabilisation by a deterministic CA

Let us introduce the following terminology. We say that a cell (i,j) has a *NE-error* if it has either an N-error or an E-error. For $x \in \Sigma^{\mathbb{Z}^2}$, we denote by $\mathcal{E}_{NE}(x)$ the set of cells having a NE-error, that is:

$$\mathcal{E}_{NE}(x) \triangleq \{(i,j) \in \mathbb{Z}^2 : x_c = x_{c+e_1} \text{ or } x_c = x_{c+e_2}\}.$$

Let $\psi : \Sigma^4 \to \Sigma$ be a function which assigns to each quadruplet of colours (a,b,c,d) a colour which is not in the set $\{a,b,c,d\}$, for example $\psi(a,b,c,d) = \min \Sigma \setminus \{a,b,c,d\}$.

We define a cellular automaton F on $\Sigma^{\mathbb{Z}^2}$ by:

$$\forall c \in \mathbb{Z}^2, \quad F(x)_c = \begin{cases} \psi(x_{c-e_1}, x_{c-e_2}, x_{c+e_1}, x_{c+e_2}) \text{ if } c \in \mathcal{E}_{NE}(x) \\ x_c \text{ otherwise.} \end{cases}$$

Proposition Let $k \geq 5$, the cellular automaton F defined above is self-stabilising on Λ_k.

Proof. It is clear from the definition that $\forall x \in \Lambda_k,\ F(x) = x$. Let us now take $x \in \tilde{\Lambda}_k$. Without loss of generality, we can assume that there exists an integer $n \geq 0$ such that $\mathcal{E}_{NE}(x) \subseteq T_n$. (Recall that $T_n = \{(i,j) \in \mathbb{Z}^2 : i + j \leq n,\ i,j \geq 0\}$). One can also check that after t steps, we have $\mathcal{E}_{NE}(F^t(x)) \subseteq T_{n-t}$. Indeed, the set of NE-errors can only decrease under the action of F: if $c \notin \mathcal{E}_{NE}(x)$, then $c \notin \mathcal{E}_{NE}(F(x))$, since by definition of F, if $c + e_1$ or $c + e_2$ takes a new colour in $F(x)$, that new colour is different from x_c. Furthermore, if $c \in \mathcal{E}_{NE}(x)$ is such that $c + e_1, c + e_2 \notin \mathcal{E}_{NE}(x)$, then $c \notin \mathcal{E}_{NE}(F(x))$, so that the set of NE-errors is progressively eroded, from the NE to the SW. After $n + 1$ steps, we have: $\mathcal{E}_{NE}(F^{n+1}(x)) = \varnothing$, meaning that the configuration is thus fully corrected: $F^{n+1}(x) \in \Lambda_k$. □

4.2 Isotropic self-stabilisation by a probabilistic CA

Let $\psi : \Sigma^4 \to \Sigma$ be a function as above. For $x \in \Sigma^{\mathbb{Z}^2}$, let us recall that we denote by $\mathcal{E}(x)$ the set of cells having an error, that is:

$$\mathcal{E}(x) \triangleq \{c \in \mathbb{Z}^2 : x_c \in \{x_{c \pm e_1}, x_{c \pm e_2}\}\}$$

We define a probabilistic cellular automaton Φ on $\Sigma^{\mathbb{Z}^2}$ which leaves the state of cell c unchanged if $c \notin \mathcal{E}(x)$ and updates it to a random value with distribution $\alpha\delta_{\psi(x_{c-e_1}, x_{c-e_2}, x_{c+e_1}, x_{c+e_2})} + (1-\alpha)\delta_{x_c}$ if $c \in \mathcal{E}(x)$. Once again, the use of an α-asynchronous rule is destined to break the potential cycles that could be created by the situations where the value of the update function is *not* deterministic (in the case where only one colour is missing in the neighbourhood).

Proposition For $k \geq 5$ and $\alpha \in (0, 1)$, the probabilistic cellular automaton Φ defined above is self-stabilising on Λ_k.

Proof. Let $x \in \tilde{\Lambda}_k$ be an initial configuration. Let $X^0, X^1, \ldots$ denote the Markov process described by Φ starting from $X^0 = x$. For any $c \notin \mathcal{E}(X^t)$, the state of cell c remains unchanged, and the neighbouring cells of c cannot take the state X^t_c, so that for each $t' \geq t$, $\mathcal{E}(X^{t'}) \subseteq \mathcal{E}(X^t)$ almost surely. Furthermore, inside $\mathcal{E}(x)$, Φ behaves like an absorbing finite state Markov chain, that eventually reaches an allowed configuration. Indeed, let us consider the cells of $\mathcal{E}(X^t)$ that have at least two correct neighbouring cells (there necessarily exist such cells, since $\mathcal{E}(x)$ is finite). If the function ψ is applied to such a cell c, and if the values of its neighbours remain the same, then $c \notin \mathcal{E}(X^{t+1})$. This happens with probability at least $\alpha(1-\alpha)^2 > 0$. Consequently, the probability of decreasing the set of errors is strictly larger than this probability at each time step. □

4.3 Extension to single-site fillable proximity tilings

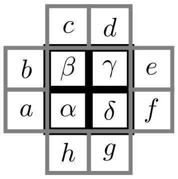
We say that a proximity tiling is *single-site fillable* if there exists a map $\psi : \Sigma^4 \to \Sigma$ such that, for any possible choice $(a, b, c, d) \in \Sigma^4$ of symbols surrounding a cell, assigning the value $\alpha = \psi(a, b, c, d)$ to the central cell ensures that it is error-free [9]. The two constructions above (directional self-stabilisation by a deterministic CA, and isotropic self-stabilisation by a probabilistic cellular automaton) naturally extend to all proximity tiling spaces that are single-site fillable.

5 The case of 4-colourings

5.1 Directional self-stabilisation by a deterministic CA that corrects by blocks

The case of 4-colourings ($\Sigma = \{0, 1, 2, 3\}$) is more delicate. Obviously, it is no longer possible to use a function ψ with the same properties as above. Nevertheless, we propose a solution where we show that the number of errors is decreased by updating 2-squares, that is, 2×2-blocks of cells. We explain the possibility of this update in the next lemma, and the show how to apply this update without generating conflicts.

Lemma For any possible choice $(a, b, c, d, e, f, g, h) \in \Sigma^8$ of symbols surrounding a 2-square (see right), there exist a choice $(\alpha, \beta, \gamma, \delta) \in \Sigma^4$ for the cells of the 2-square such that the four cells of the 2-square are error-free.



Proof. If $\{a, d, e, h\} \subsetneq \Sigma$, then we can choose a colour from $\Sigma \setminus \{a, d, e, h\}$ and assign it to both α and γ. We are then sure that we can find suitable colours for the two remaining cells, since each of these two cells is surrounded by at most three different colours. In the same way, if $\{b, c, f, g\} \subsetneq \Sigma$, we can find a valid pattern.

Let us now assume that $\{a, d, e, h\} = \{b, c, f, g\} = \Sigma$. Without loss of generality, we can assume that $a = 0, h = 1, d = 2, e = 3$. The set of allowed colours for α is then $\{2, 3\}$, and the set of allowed colours for γ is $\{0, 1\}$. If the allowed colours for β and δ are $\{0, 1\}$ and $\{2, 3\}$ respectively, then a valid pattern is given by $(\alpha, \beta, \gamma, \delta) = (2, 0, 1, 3)$. If the allowed colours for β and δ are $\{0, 2\}, \{1, 3\}$ respectively, then a valid pattern is given by $(\alpha, \beta, \gamma, \delta) = (2, 0, 1, 3)$. The other cases are analogous. □

We can now design a CA that corrects finite perturbations of Λ_4. Let $\psi : \Sigma^8 \to \Sigma^4$ be a function that maps some $(a, b \ldots, h) \in \Sigma^8$ to a quadruplet $(\alpha, \beta, \gamma, \delta) \in \Sigma^4$ such that the pattern formed by these values as illustrated above is an error-free pattern.

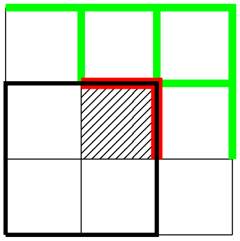


Figure 2: Illustration of the definition of the cellular automaton used to correct 4-colourings. The central cell is a NE-corner if one of the red/dark gray lines (North or East or both) presents a mistake and all the green/light gray lines are free of errors. The 2-square whose NE-corner is the central cell is then corrected by the cellular automaton.

Our aim is to use this function ψ to correct non-overlapping 2-squares, by ensuring that the correcting rule applies without conflicts. In order to do this, we first identify a set of cells that will play the role of the top-right cells of the 2-squares that will be updated.

For a configuration $x \in \Sigma^{\mathbb{Z}^2}$, let us denote again the set of cells having a NE-error by $\mathcal{E}_{NE}(x) = \{c \in \mathbb{Z}^2 : x_c = x_{c+e_1} \text{ or } x_c = x_{c+e_2}\}$. We say that a cell $c \in \mathbb{Z}^2$ is a *NE-corner* if: $c \in \mathcal{E}_{NE}(x)$ and $c+e_1, c-e_1+e_2, c+e_2, c+e_1+e_2 \notin \mathcal{E}_{NE}(x)$, see Fig. 2 for an illustration of the definition. We denote by $\mathcal{C}_{NE}(x)$ the set of NE-corners in a configuration $x \in \Sigma^{\mathbb{Z}^2}$, that is:

$$\mathcal{C}_{NE}(x) \triangleq \{c \in \mathcal{E}_{NE}(x)\,;\; c+e_1, c-e_1+e_2, c+e_2, c+e_1+e_2 \notin \mathcal{E}_{NE}(x)\}.$$

Note that if $x \in \tilde{\Lambda}_4$, then $\mathcal{E}(x) \neq \varnothing \iff \mathcal{C}_{NE}(x) \neq \varnothing$. Indeed, if $\mathcal{E}(x)$ is a non-empty set, then it contains at least one NE-error. Let us sweep the configuration x by NW-SE diagonals, from the NE to the SW. Since $\mathcal{E}(x)$ is finite, we can consider the first diagonal which contains a NE-error, and on this diagonal, we consider the leftmost NE-error (which is also the uppermost). By definition of a NE-corner, this NE-error is a NE-corner.

We define a CA F by the following rule: if a cell $c = (i,j) \in \mathbb{Z}^2$ is a NE-corner, then apply ψ to the 2-square whose NE-corner is c, that is, we replace the colours of the cells $(i-1,j-1), (i-1,j), (i,j), (i,-j-1)$ by $\psi(a,b,\ldots,h)$, where $a = x_{i-2,j-1}, b = x_{i-2,j}, \ldots, h = x_{i-1,j-2}$ (see above). Let us first observe that the CA F given by this rule is well-defined. Indeed, by definition of a NE-corner, one can check that there are no two consecutive NE-corners, vertically or horizontally, or in diagonal. Consequently, at each step, the 2-squares that are updated do not overlap (note however that they can share some edges, in which case there can be errors at these edges after applying the CA rule).

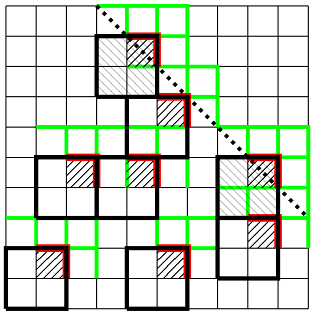
Proposition The cellular automaton F defined above is self-stabilising on Λ_4.

Proof. Since the initial configuration x is assumed to be a finite perturbation of a valid colouring, the number of NE-corners is finite. We prove that on any configuration in $\tilde{\Lambda}_4 \setminus \Lambda_4$, the number of NE-corners is strictly decreasing. Since every configuration in $\tilde{\Lambda}_4 \setminus \Lambda_4$ has at least one NE-corner, this implies that the self-correction succeeds in finite time.

Let us consider the NE-corners of $F^t(x)$. The rule F consists in updating the 2-squares associated to these NE-corners. At the next time step, one can check that all possible new NE-corners belong to these 2-squares that were updated. Indeed, if a cell does not belong to such a 2-square, then it cannot become a NE-corner at the next time step: if a neighbour of this cell were modified, then its new colour respects the colour constraint. Furthermore, there is at most one new NE-corner in each 2-square that is updated, by definition of a NE-corner.

Now, to end the proof, let us show that there exists at least one of these 2-squares that does not contain a NE-corner any more. This will prove that the number of NE-corners is strictly decreasing. Let us sweep the configuration by NW-SE diagonals, from the NE to the SW.

We consider the first diagonal which contains a NE-corner. After applying F, the 2-squares defined by the NE-corners that are on that diagonal do not contain a NE-corner any more. Indeed, our method of sweeping ensures that the two cells to the North and the two cells to the East of this 2-square were not modified; for an illustration, see this figure:



□

5.2 Isotropic self-stabilisation by a probabilistic CA

The problem of finding a rule which is isotropic and self-stabilising for four colours is not straightforward. We now propose a rule which we believe answers the problem, but for which we have no formal proof of success yet. Our idea is to modify the method used for the case $k \geq 5$, and make an exception when there is no colour available to directly correct a cell.

So, we now define ψ as a random function which assigns to each quadruplet of colours (a, b, c, d) a colour uniformly chosen in $\Sigma \setminus \{a, b, c, d\}$ if this set is not empty, and a colour uniformly chosen in Σ otherwise. We then consider the probabilistic cellular automaton that, for any configuration $x \in \Sigma^{\mathbb{Z}^2}$, updates to state $\psi(x_{c-e_1}, x_{c-e_2}, x_{c+e_1}, x_{c+e_2})$ the cell c if it has an error ($c \in \mathcal{E}(x)$), and keeps the value x_c otherwise.

Experimentally, we observe that this rule succeeds in correcting rapidly most of the initial perturbations of valid tilings. However, unlike the case $k \geq 5$, for $k = 4$, we cannot ensure with the PCA above that the errors stay in some bounded area.

We conjecture that from any finite perturbation of a valid tiling, this probabilistic cellular automaton almost surely reaches in finite time a valid 4-colouring.

To support this claim, one can try to find configurations for which this rule may fail in correcting in finite time for some particular configurations.

Consider the following configuration:

1	2	3	0	1	2	3	0	1	2	3	0	1	2
3	0	1	2	3	0	1	2	3	0	1	2	3	0
2	3	0	1	2	3	**0**	**0**	1	2	3	0	1	2
0	1	2	3	0	1	2	3	0	1	2	3	0	1
2	3	0	1	2	3	0	1	2	3	0	1	2	3

It has two cells in error and is such that all cells, even the two that are in error, see the three other colours in their neighbourhood. Consequently, if a cell changes its state alone, it will remain in error. For this specific configuration, some kind of coordination is thus necessary, which cannot here occur by a specific mechanism as for the deterministic case.

It might thus be thought at first that errors may propagate arbitrary far from their origin. However, we experimentally observe that it is not the case: errors have a tendency to stay in the same area, and the correcting process is more rapid than the error-diffusion process. Surprisingly enough, even when the cells are updated successively at random (fully asynchronous case), we also noticed that the rule succeeds in correcting errors. Indeed, when the errors propagate, they modify the configuration in such a way that the property of seeing three different colours in the neighbourhood is lost, which finally enables a correction to take place. By comparison, we believe that for this configuration and this rule, having the possibility to make parallel updates, even if it means using α-asynchronous updates, can only increase the possibilities of correction. It is an open problem to give a formal proof of this self-stabilisation property.

5.3 Extension to ℓ-fillable proximity tilings

We say that a proximity tiling is *strongly ℓ-fillable* if there exists a map $\psi : \Sigma^{4\ell} \to \Sigma^{\ell^2}$ such that, for any possible choice $(a_1, \dots, a_{2\ell}) \in \Sigma^{4\ell}$ of symbols surrounding an ℓ-square, assigning the values $\psi(a_1, \dots, a_{2\ell})$ to the inner cells of the ℓ-square ensures that each cell of the ℓ-square is error-free. (Note that here, we do not assume any further condition on $(a_1, \dots, a_{2\ell}) \in \Sigma^{4\ell}$; we refer again to [1] for a similar but weaker condition of ℓ-fillability). The self-stabilisation by a deterministic CA described above extends to all proximity tilings that are strongly ℓ-fillable. One can indeed extend the notion of NE-corner in that context, see Fig. 3 for an illustration in the case $\ell = 3$. The definition of the CA and the proof that it is self-stabilising can then be easily adapted.

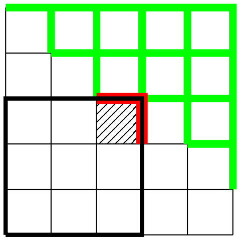


Figure 3: Illustration of the definition of the cellular automaton used to correct a 3-fillable proximity tiling. The central cell is a NE-corner if one of the red lines (North or East or both) presents a mistake and all the green lines are free of errors. The 3-square whose NE-corner is the central cell is then corrected by the cellular automaton.

6 The case of 3-colourings

6.1 Necessity to correct arbitrarily far from the locations of errors

For $k \geq 4$, with the rules defined in the previous sections, one can correct the errors in a local way: if we observe a finite island of errors, then we can always correct the island without modifying the configuration at a distance larger from 1 or 2 from the island. Let us now show that this property no longer holds for $k = 3$. To this end, we will change our representation and associate to each configuration that is a 3-colouring a configuration in the so-called *six-vertex model*.

This model is obtained by associating an arrow to each couple of neighbouring cells (horizontal or vertical), these arrows are represented at the boundary between the two cells according to the following rules. Let q and q' be the colours of the two neighbouring cells. As we have $q' \neq q$, it follows that we either have $q' = q + 1 \mod 3$ or $q' = q - 1 \mod 3$. Depending on this, we draw the arrow in one direction or the other.

- The vertical boundaries which separate q and $q + 1$ (resp. $q - 1$) have an arrow pointing up (resp. down).
- The horizontal boundaries which separate q and $q + 1$ (resp. $q - 1$) have a right (resp. left) arrow.

These conventions are represented on Fig. 4.

One can then check that starting from a 3-colouring, the resulting arrow configuration is such that at each vertex, there are exactly two incoming arrows and two outgoing arrows. Conversely, from a six-vertex configuration, there are three 3-colourings giving that arrow configuration. (Once we choose the colour of one cell, all the other colours can be deduced).

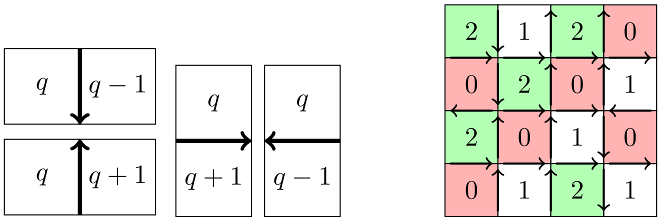


Figure 4: The convention used for encoding 3-colouring configurations in the six-vertex model and an example of a configuration with its associated six-vertex image.

Figure 4 shows an example of such an encoding of a valid colouring. By contrast, Fig. 5 displays a configuration which holds a finite perturbation of a 3-colouring.

Notice that we have drawn in bold the arrows pointing to the South and the ones pointing to the West. The knowledge of the position of these two types of arrows is sufficient to fully describe the configuration; indeed, the other horizontal or vertical arrows have to be East or North arrows, respectively.

In the example given, let us imagine that we have fixed the value of a set of cells that are located at the boundary of a square. We call this set of cells the *boundary square*, and we want to fill the inner part of that boundary square with an admissible configuration. One can verify that the only way to fill this inner part corresponds to a six-vertex configuration that would have a direct South vertical line: indeed, there is only one bold incoming arrow and one bold outgoing arrow in the boundary square, and we have to connect them. So, we can construct finite perturbations of 3-colourings that present only two cells in error (one single interface with same colours), but for which we need to modify a domain of size arbitrary large in order to recover a valid configuration. This is expressed by the following proposition.

Proposition For any $m \in \mathbb{N}$, there exists a configuration $y \in \tilde{\Lambda}_3$ such that $\operatorname{card} \mathcal{E}(y) = 2$, and $\forall x \in \Lambda_3, \operatorname{card} \Delta(x, y) \geq m$.

6.2 Deterministic self-stabilisation by a CA with additional states

In order to decide if a boundary square is fillable or not, we just need to know if it is possible to associate each incoming arrow with an outgoing arrow. This is easy to do with sequential operations and additional symbols which do not appear in the initial condition. Starting from the NE-corner, let us enumerate the incoming arrows on the North and then West sides, from 1 to n_i, and the outgoing arrows on the East and then South sides, from 1 to n_o. The boundary square is fillable if we can match each incoming arrow number k

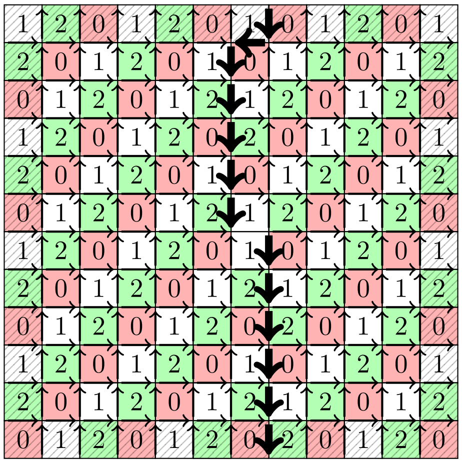
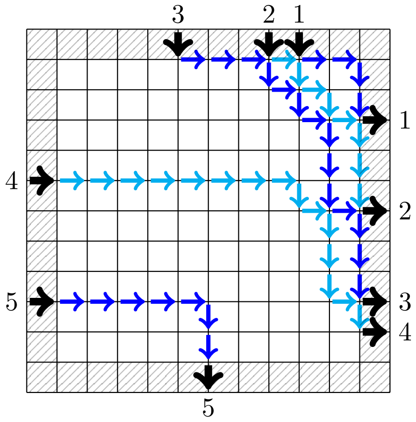


Figure 5: Example of a finite perturbation of a 3-colouring and its associated six-vertex configuration. South and West arrows are shown in bold, dashed cells indicate the boundary square.

with the outgoing arrow number k by a SE-path of arrows (which implies that $n_i = n_o$). In order to know if this can be done, we try to match successively the incoming and outgoing arrows from 1 to $n_i = n_o$ by disjoint paths, by moving E if the edge has not already been selected, and S otherwise. As an additional condition, we need to ensure at each time step that the path does not go beyond the corresponding outgoing arrow or come across another path. This procedure succeeds if and only if there is at least one admissible matching, see the diagram below for an illustration:



Using this method, let us sketch how to design a deterministic CA that corrects finite perturbations of 3-colourings with additional states. First, we mark error cells and create a boundary square around them. We then use a kind of Turing machine that calculates if this boundary square is fillable or not, with the procedure above. If it is fillable, we fill it with the solution associated to the six-vertex configuration given by the procedure. Otherwise, we consider a new boundary square with a box of size increased by one unit. When different boundary squares meet, they merge and restart their process.

It is an open problem to know if a solution without additional symbols exists.

Moreover, in contrast with the previous the sections, here we cannot use the method of taking an available colour or a random colour when no colour is available. We noticed experimentally that the errors diffuse and we could not find any rule that keeps them confined, even in statistical terms.

7 Conclusion

We presented the study of self-stabilisation problems for k-colourings and for some more general tilings spaces. The easiest cases are $k = 2$ and $k \geq 5$. For $k = 4$, deterministic rules still exist but are not as straightforward to design. In the probabilistic setting, we could propose symmetric rules, which experimentally perform well, but for which no formal proofs are available yet. The three-colour case is the most challenging and it is an open problem to know if efficient deterministic solutions do exist.

In this work, we have searched for solutions that operate in a "reasonable" time scale. However, when no such rules are found, it is still possible to use a kind of "brute-force" process where errors are initially at the centre of a self-correcting zone. The cellular automaton should then test sequentially if there are admissible solutions inside this zone. If the answer is positive, then the part is corrected, if not, then the zone is extended by one cell in each direction. When two such zones meet, there should be some procedures to merge the zones and "restart" the process. It is clear that even though each step can be thought of separately in a clear way, putting all the steps together in a cellular automaton that effectively works is a huge task. Moreover, the time needed for such a rule to operate would be more than exponential in the number of errors.

The question might also be raised for the solutions which make use of additional symbols: can one find rules which also resist the introduction of additional symbols in the initial condition? Another important problem that we are currently addressing is to consider the case where the errors are initially randomly distributed on *all the grid*.

References

[1] N. Alon, R. Briceño, N. Chandgotia, A. Magazinov, and Y. Spinka. Mixing properties of colorings of the $\mathbb{Z}^d$ lattice. Preprint arXiv 1903.11685, 2019.

[2] A. Bębenek and I. Ziuzia-Graczyk. Fidelity of DNA replication—a matter of proofreading. *Current Genetics*, (64):985–996, 2018.

[3] A. Bušić, N. Fatès, J. Mairesse, and I. Marcovici. Density classification on infinite lattices and trees. *Electron. J. Probab.*, 18:no. 51, 22, 2013.

[4] E. W. Dijkstra. Self-stabilization in spite of distributed control. In *Selected writings on computing: a personal perspective*, pages 41–46. Springer, 1982.

[5] N. Fatès. Asynchronous cellular automata. In *Encyclopedia of Complexity and Systems Science*, pages 1–21. Springer Berlin Heidelberg, 2018.

[6] P. Gach, G. L. Kurdyumov, and L. A. Levin. One-dimensional uniform arrays that wash out finite islands. *Problems of Information Transmission*, 14(3):223–226, 1978.

[7] P. Gács. Reliable computation with cellular automata. *Journal of Computer and System Sciences*, 32(1):15–78, 1986.

[8] P. Gács and J. Reif. A simple three-dimensional real-time reliable cellular array. *Journal of Computer and System Sciences*, 36(2):125–147, 1988.

[9] B. Marcus and R. Pavlov. An integral representation for topological pressure in terms of conditional probabilities. *Israel Journal of Mathematics*, 207(1):395–433, 2017.

[10] N. Pippenger. Symmetry in self-correcting cellular automata. *Journal of Computer and System Sciences*, 49(1):83–95, 1994.

[11] A. L. Toom. Nonergodic multidimensional system of automata. *Problemy Peredachi Informatsii*, 10(3):70–79, 1974.

[12] A. L. Toom. Stable and attractive trajectories in multicomponent systems. In *Multicomponent Random Systems*, volume 6 of *Adv. Probab. Related Topics*, pages 549–575. Dekker, New York, 1980.