Balanced Connected Partitioning of Unweighted Grid Graphs

Cedric Berenger
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France

Peter Niebert
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France

Kevin Perrot
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France

Abstract

We consider a partitioning problem for grid graphs with special constraints: a (square) grid graph as well as a number of colors is given, a solution is a coloring approximatively assigning the same number of vertices to each color and such that the induced subgraph for each color is connected. In a "rooted" variant, a vertex to be included in the coloring for each color is specified as well. This problem has a concrete motivation in multimedia streaming applications.

We show that the general problem is NP-complete. On the other hand, we define a reasonable easy subclass of grid graphs for which solutions always exist and can be computed by a greedy algorithm.

2012 ACM Subject Classification Mathematics of computing → Paths and connectivity problems, Hardware → Partitioning and floorplanning

Keywords and phrases grid graphs, connected partitioning, NP-completeness, graph algorithm

Digital Object Identifier 10.4230/LIPIcs.MFCS.2018.39

Acknowledgements We would like to thank Jérémy Chalopin and Victor Chepoi for their comments on earlier versions of this work.

1 Introduction

We study a particular partitioning problem of (square) grid graphs. Consider a finite grid (a rectangle) and a subset of squares present (shown in black on the second leftmost image of Figure 1). The present squares are vertices of a graph where the edges are implicit by the neighboring relation (leftmost image).

The problem we study is to color/partition such a graph with a given number of colors so that the induced subgraph for each color is connected and that the partition is balanced, i.e. the number of vertices for each color is (almost) the same. In the example of Figure 1, we choose to color with three colors. The third image shows a coloring satisfying both constraints. In the "rooted variant", we additionally specify for each color a root node that has to take that color (fourth image). For this choice, the third image is not a solution, but the fifth image is. Of course, the fifth image is also a solution of the unrooted problem.

The practical motivation for this particular problem concerns (broadcast) streaming in physical networks in multimedia applications (where each square is a screen tile and there may be holes and irregular borders), but also routing problems in NoC [10] (network on a chip) with horizontal and vertical communication links and dead vertices (due to manufacturing

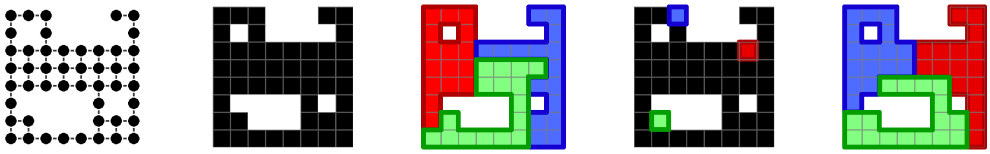

Figure 1 Grid graph and balanced connected coloring.

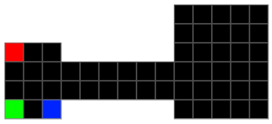

Figure 2 A rooted problem with 3 colors stuck in a corridor of width 2.

issues). For these systems, the limited bandwidth of links can be compensated by partitioning and injecting the data into the network from several "roots". Each root is thus responsible for a subset of vertices and the broadcast messages from distinct roots do not cross.

For the general case, we show that finding a balanced connected partition is NP-complete for both the rooted and the unrooted case (Theorems 1, 2). We achieve the NP-hardness result by a reduction from the Hamiltonian circuit problem on hexagonal graphs [7].

Intuitively, the connectivity of each colored zone is subject to conflicts when there are fewer connections than colors that need to pass. In our reduction, we exploit this by binary conflicts on connections that can only be used by a single color. Beyond binary conflicts, taking the point of view of our grid as a maze, several parallel and adjacent lines of vertices form a corridor of a limited capacity for colors to pass. For instance, in the example of Figure 2, the three roots are connected by a corridor of width 2, hence only two colors can connect to the big zone and obviously, no balanced connected partition exists. In order to define "easy" subclasses of graphs called *q-square connected graphs*, we formalize the idea of a maze with all corridors, whether straight or "diagonal" (zigzaging), large enough to let all colors pass. The condition of q-square connectedness can be verified on bounded windows of the graph and thus decided in linear time. It guarantees that a solution exists. Indeed, we give a greedy polynomial time algorithm for the unrooted case that always finds a solution on such topologies (Theorem 6). We also claim that this simple greedy algorithm can be extended to the rooted case and improved to compute solutions in linear time (for any fixed q).

The definition, intuitively similar but not equivalent to the classical definition of k-connectedness of graphs, is not trivial and is the result of fine-tuning the conditions in order to obtain a correctness proof. However, it is intuitive and may have an interest in itself. We have implemented the algorithm for simulation where it proves to be well behaved (it produces partitions that are "compact" and not unnecessarily intertwined).

Related work. To the best of our knowledge, the hardness of the precise problem studied here, as well as the easy subclass and the greedy algorithm, are unknown.

The much studied Balanced Connected q-Partition Problem (BCP$_q$), related to our work, is the problem of partitioning a weighted graph $G = (V, E)$ into q connected subgraphs of similar size/weight. In this work, we study this problem for unweighted grid graphs.

The unweighted BCP$_q$ is shown NP-Hard in [4] for bipartite graphs for $q \geq 2$, but the result does not cover grid graphs. It has been proven in [1] that the BCP$_q$ is NP-Hard in $G_{m\times n}$ grid graphs for any $n >= 3$, but this result requires weights on the vertices. That work also gives approximation algorithms considering the relative error under some hypothesis. In

[3] it is shown that finding a solution with an absolute error less than $|V|^{1-\epsilon}$ is NP-Hard even for the unweighted variant of the BCP$_q$, but on general graphs only. Our NP-hardness proof requires novel ideas and a different construction from previous works.

In [8, 9], polynomial algorithms for BCP$_q$ are given for unweighted q-connected graphs. However grid graphs are 2-connected only. There exist also polynomial approximation results under certain restrictions. In [3] a 4/3-approximation algorithm is shown to exist for BCP$_2$ without other assumptions. More recently, [11] shows a polynomial time 7/6 approximation algorithm for BCP$_2$ in grid graphs. In [2], it is shown that there is no approximation algorithm with ratio smaller than 6/5 for arbitrary graphs. The same article also shows that BCP$_q$ is strongly NP-Hard even on q-connected graphs with weights, as well as an inapproximability result for (BCP$_2$).

With motivations from VLSI design, a lot of work focuses on minimizing the number of transversal edges between parts, not necessarily keeping them connected, e.g. [5], [6].

Outline. The remainder of this article is structured as follows: Section 2 formalizes the model and the problem statement. In Section 3, we recall the Hamiltonian circuit problem for hexagonal graphs and develop the reduction from this problem to show NP-hardness for the general case. Essentially the same reduction can be used for the unrooted and for the rooted case. NP-completeness follows trivially as it is straightforward to verify a solution by a linear algorithm. In Section 4, we define the subclasses of q-square connected graphs by two alternative definitions (one intuitive, the other for algorithmic purposes) and show their equivalence. Then we define a rule based greedy algorithm for the unrooted case for coloring a graph one color after the other. We prove termination and correctness of the algorithm. Then we discuss how this algorithm can be extended to deal with the rooted case. In Section 5, we conclude and give an outlook on future work.

2 Model

We now formalize the notions of square grid graph and the partitioning problem. We simplify the presentation by assuming all square grid graphs embedded in the plane of pairs of integers.

For any finite set of pairs $V \subset \mathbb{Z}^2$ and the induced set of neighboring edges $E = \{((x_1, y_1), (x_2, y_2)) \in V \times V \mid |x_2 - x_1| + |y_2 - y_1| = 1\}$, we call $G_V = (V, E)$ a *square grid graph*. By $|G_V| := |V|$ we denote the *size* of a square grid graph

For simplicity, by grid graphs we will mean square grid graphs in the following (as opposed to hexagonal grid graphs, for instance).

For a subset of vertices $V' \subset V$ of a (grid) graph $G_V = (V, E)$, the graph (V', E') with $E' = E \cap (V' \times V')$ is the induced subgraph of G_V restricted to vertex set V'. Obviously, for a grid graph $G_V = (V, E)$ and a subset of vertices V', the induced subgraph (V', E') equals the grid graph $G_{V'}$.

A graph (V, E) is *connected* if for each pair $u, v \in V$ there exists a sequence $u = v_0, v_1, \ldots, v_l = v$ for some $l \geq 0$ and $(v_i, v_{i+1}) \in E$ for all $0 \leq i < l$.

We will represent grid graphs and induced subgraphs by colored 2D matrices, see Figure 1. An obvious data structure to represent grid graphs is a list of vertices. Alternatively, the *containing rectangle* of a grid graph G_V is defined as $R(V) = \{(x, y) \mid \exists (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) \in V : x_1 - 1 \leq x \leq x_2 + 1 \text{ and } y_3 - 1 \leq y \leq y_4 + 1\}$, a full matrix containing V and an external border without vertices. The containing rectangle allows for matrix representations of grid graphs d.

A *connected part* $G_{V'}$ of a grid graph G_V is an induced subgraph of G_V that is connected. A *q-partition* $\mathcal{P}$ of a grid graph G_V is a set of q parts G_{V_i} with $1 \leq i \leq q$ such that : $V = \bigcup_{G_{V'} \in \mathcal{P}} V'$ and for any $G_{V_i}, G_{V_j} \in \mathcal{P}$ we have that $i \neq j$ implies $V_i \cap V_j = \emptyset$. We call a partition *connected* if all parts are connected.

For a subset of vertices $R = \{r_1, \ldots, r_q\} \subseteq V$ of a grid graph V_G called *roots*, an *R-rooted* partition of G_V is a partition $\{G_{V_i} \mid r_i \in V_i\}$.

A partition $\mathcal{P}$ of a grid graph G is *perfectly balanced* if and only if for all $G_{V_i} \in \mathcal{P}, |V_i| = \frac{|V|}{|\mathcal{P}|}$. Additionnaly, considering a partition $\mathcal{P}$ of a grid graph G, we say a part $G_{V_i} \in \mathcal{P}$ is *perfectly saturated* if it fits the requirement for P to be perfectly balanced, *i.e.* $|V_i| = \frac{|V|}{|\mathcal{P}|}$. If $|V_i|$ is below/above the *saturation threshold* $\frac{|V|}{|\mathcal{P}|}$, we respectively say that the part is *under/over-saturated.*

Perfect balance may often be an undesirably restrictive condition when it comes to balancing load in networks. In order to relax it, we may specify an error tolerance ratio $0 \leq r \leq 1$ with perfect balance corresponding to $r = 0$. For $0 \leq r \leq 1$ we say that a q-partition $\mathcal{P}$ of a grid graph G_V is *r-balanced* if and only if for each part $G_{V_i} \in \mathcal{P}$ we have $\frac{|G_V|}{q} \cdot (1-r) \leq |G_{V_i}| \leq \frac{|G_V|}{q} \cdot (1+r)$. The same way, we say that a part G_{V_i} of a q-partition $\mathcal{P}$ of a grid graph G_V is *r-saturated* if and only if $\frac{|G_V|}{q} \cdot (1-r) \leq |G_{V_i}| \leq \frac{|G_V|}{q} \cdot (1+r)$. If $|V_i|$ sits outside of the *saturation interval* $\left[\frac{|G_V|}{q} \cdot (1-r), \frac{|G_V|}{q} \cdot (1+r)\right]$, we respectively say that the part is *under/over-r-saturated.* When the ratio r is clear from the context, we will simply say that an area is balanced instead of r-balanced, and that a part is (under/over)saturated instead of (under/over)r-saturated.

The *r-balanced connected partitioning problem* (rBCP) is the problem of finding, for any grid graph G_V and number q, an r-balanced connected partition with q parts.

The *rooted r-balanced connected partitioning problem* (RrBCP) is the problem of finding, for any grid graph G_V and set of roots $R \subseteq V$, an R-rooted r-balanced connected partition.

3 NP-Hardness

In this section we present a polynomial time reduction from the Hamiltonian cycle problem (HCP) on hexagonal grid graphs to the rooted r-balanced connected partitioning problem (rRBCP) on square grid graphs, for $r \leq \frac{1}{7}$. We then argue that the reduction also applies to r-balanced connected partitioning problem (rBCP).

For any finite set $V \subset \mathbb{Z}^2$ of vertices, we construct the *hexagonal grid graph* $H_V = (V, E)$ with the induced set of edges $E = \{((x_1, y_1), (x_2, y_2)) \in V \times V \mid (|x_2 - x_1| = 1 \text{ and } |y_2 - y_1| = 0) \text{ or } (x_2 - x_1 = 0 \text{ and } y_2 - y_1 = 1 \text{ and } x_1 + y_1 \mod 2 \equiv 0)\}$. By $|H_V| := |V|$ we denote the *size* of the hexagonal grid graph. An example is shown on Figure 5 (left).

The *Hamiltonian cycle problem* (HCP) is the following decision problem: given a graph G, does there exist a cycle containing exactly once each vertex of G? This problem has been proven to be NP-complete when restricted to hexagonal grid graphs [7].

Given a hexagonal grid graph for the HCP, we will construct a square grid graph and set of roots for rRBCP. The section is split in four parts. In the first part we give intuitions on some simple rRBCP instances. Then we describe the reduction and concepts at stake. Finally, we demonstrate the correctness of the reduction. The last part explains how to apply the same idea for a reduction to rBCP.

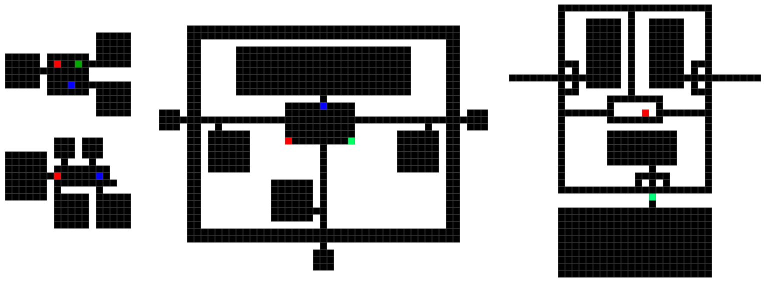

Figure 3 Intuition on the construction: tanks and roots/colors interaction.

3.1 Intuitions on the construction

Let us introduce step by step some straightforward "tricks" used in the reduction.

Consider a large connected region of a square grid graph, uncolored and accessible only via a single vertex or a single path, then this region will own a single color. This is exemplified on Figure 3 (left top): any partitioning must assign the same color to each of the three large outer regions (possibly three different colors, but a single color for each). We call such a region a "tank". In contrast, lines of vertices of width 1 are called "*paths*".

The basis of the construction will be to connect tanks via paths, and choose sizes of tanks such that the total number of vertices used in paths is negligible compared to a single tank.

If the tolerance ratio and the size of tanks are tightly chosen, then coloring a given tank may result in saturating the part corresponding to that color, i.e. a color tied to a tank may not color other tanks without oversaturating. For example, on Figure 3 (left bottom), the red root must color the large tank on the left in any partitioning, nearly saturating that color. Hence, red cannot own any of the two bottom tanks without oversaturating.

Some instances can "trap" parts/colors inside a restricted area. On Figure 3 (middle), blue is tied to a very large tank and must color it nearly saturating that color (for some suitable ratio r). But blue cannot reach the outer ring, because otherwise it would have to cover at least one of the three remaining *inner tanks* and oversaturate.

A color can prevent another color from creating a fork. In Figure 3 at right, the upper rectangle of paths has three access points (left, right, bottom). Green has to color the lower tank, and we suppose that adding any of the three inner tanks would lead to oversaturation. On the other hand, red requires the three inner tanks to saturate. Suppose that green was connected to the two exit points left and right by paths. Whichever the choice of paths taken, red could no longer reach all three inner tanks. In other words, this construction forces a decision for green between the left and the right exit point. This "gadget" will be used as a *macrocell* representing a vertex of the hexagonal grid, and the green color can only connect two of the three exits of any macrocell in order for the partition to be balanced, forcing green on a Hamiltonian path.

3.2 Reduction construction up to calculations

For a given hexagonal grid graph (instance of HCP), we build a square grid graph and set of roots (instance of rRBCP). This instance will admit a solution if and only if the HCP instance admits a solution: the square grid graph mimics the structure of the hexagonal grid, and the part of a specific color will be forced to correspond to a Hamiltonian cycle of the hexagonal grid graph (if one exists, otherwise no r-balanced partioning exists). Our construction will exclusively be composed of tanks and paths.

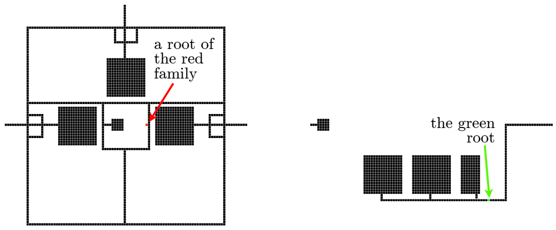

Figure 4 Macrocells with their root vertices for red (left) and green (right) parts.

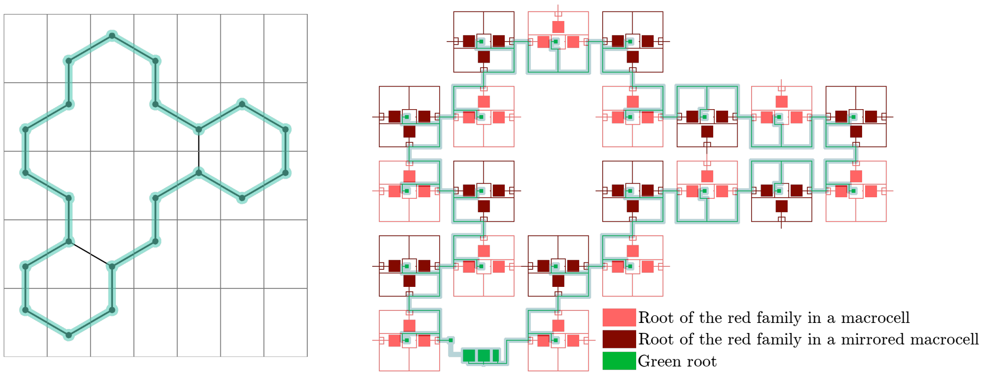

Figure 5 An hexagonal grid graph (left), and the corresponding grid graph construction (right).

Consider a macrocell as Figure 4 (left), which recalls the construction in Figure 3 (right) with an additional tank in the center. As the sizes of the tanks will depend on the hexagonal grid size and on the tolerance ratio r, it is crucial that we can scale up macrocells. Let $T \times T$ be the size of a containing rectangle of such macrocells (such that the three entry paths lie on the center of three sides), let t_i, t_c be the respective sizes of the three inner tanks and of the center tank, and let p be the total number of path vertices (not tank) in one macrocell. It is straightforward to scale up a macrocell so that the number of paths vertices becomes negligible: p grows linearly with T, whereas t_i, t_c grow quadratically.

Let us distinguish a vertex of degree two of the hexagonal grid graph, which we call *primary root* (there always exists one). By using copies of this macrocell or its mirror for each vertex distinct from the primary root, we can mimic the layout of the hexagonal grid graph, as shown on Figure 5 (right). Let m be the number of vertices in the hexagonal grid graph, and let us call *red family* the set of m colors used for roots in these macrocells.

For the primary root vertex (of degree two in the hexagonal grid graph), we replace it with a special macrocell containing (see Figure 4 (right)): on one entry path a tank of size $t_g \approx 3 \cdot \frac{5}{6} t_i$ called *green tank* with a *green* root tied to it, and on the other entry path a tank of size $t_c \approx \frac{t_i}{6m}$. The main idea here is that the green root misses approximately $3 \cdot \frac{1}{6} t_i$ vertices to be balanced (compared to the family of red colors each taking three tanks of size t_i), and will have to take the *central tanks* of size t_c in all macrocells (m times t_c) in order to enter the saturation interval and be r-balanced, hence creating a Hamiltonian path.

A complete construction is shown on Figure 5 (right).

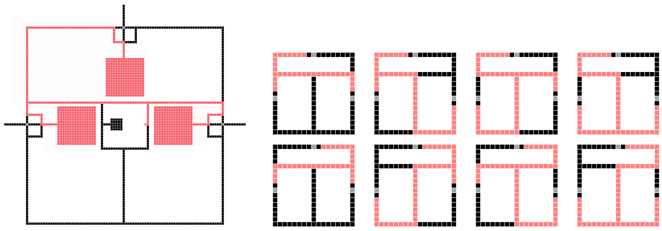

Figure 6 If another color goes across a macrocell, then the red color enforces it to take exactly two out of the three entry paths. In any case red takes the three inner tanks. Detailed schematic of the first possibility on the left, simplified schematics on the right for the eight possibilities.

3.3 Reduction correctness and calculations

Let us recall that $r < \frac{1}{7}$. We now argue about the correctness of a series of facts, leading to the correctness of the reduction.

Figure 6 shows a full case disjunction regarding the fact that macrocells owned by colors in the red family prevent green from creating forks (corresponding to degree three vertices).

▶ **Fact 1.** *If green goes across a macrocell, then it must take exactly two entry paths.*

In other words, green part is constrained by the adversary family of red colors to follow a Hamiltonian cycle.

Now, let us see how to calculate the sizes of tanks and macrocells such that the number of path vertices is negligible compared to the size of any single tank, and that green must take exactly all central tanks of size t_c in each macrocell (and therefore runs across each one).

We introduce variables to specify the construction precisely:

- m: total number of vertices in the hexagonal grid graph,
- n: total number of vertices in the square grid graph,
- k: scaling factor of macrocells,
- t_i: size of each of the three inner tank in each macrocell,
- t_c: size of the central tank in each macrocell (plus one in the primary root macrocell),
- t_g: size of the tank tied to the green root in the primary root macrocell,
- p: number of path vertices (not tank) in one macrocell,
- $s = \frac{n}{m}$: perfect saturation threshold.

By definition, we have the saturation interval ratio $I_s = [s(1-r), s(1+r)]$. We have two kinds of roots (the red family and the green), and four kinds of vertices (path and tanks t_i, t_c, t_g). This gives three groups of inequations to consider. Recall that $t_g \approx 3 \cdot \frac{5}{6} t_i$ and $t_c \approx \frac{t_i}{6m}$.

Path vertices. The *unitary macrocell* shown in Figure 4 (left) permits tanks of maximum size c_t with a minimum of c_p path vertices. By multiplying the dimensions of the unitary macrocell by an integer factor k (except for path wideness), we get a *k-scaled macrocell* permitting tanks of maximum size $k^2 \times c_t$ with less than $k \times c_p$ path vertices (the needed path length to link tanks together grows linearly with k whereas maximum possible tank size grows quadraticaly) :

$$\exists k \in \mathbb{N} : \quad p \geq k \times c_p \quad \text{and} \quad t_i, t_c \leq k^2 \times c_t \quad \text{with } c_p = 639 \text{ and } c_t = 256 \tag{1}$$

We want paths to be negligible and dominated by tanks for balance, meaning that modifying the number of path vertices without modifying the number of tanks in a part has no impact on its saturation. The total number of path vertices is mp.

Red family and inner tanks. We want that each root of the red family reaches the saturation interval with exactly the three inner tanks of its macrocell, *i.e.*

$$2t_i + mp < s(1-r) < 3t_i \quad \text{and} \quad 3t_i + mp < s(1+r) < 4t_i. \tag{2}$$

Red family and green tank. In any partitioning, roots of the red family cannot take the green tank.

Green and all tanks. Green always owns the green tank, and we want the green root to reach the saturation interval with exactly the green tank plus all m central tanks, *i.e.*

$$t_g + (m-1)t_c + mp < s(1-r) \leq t_g + mt_c \text{ and } t_g + mt_c + mp < s(1+r) < t_g + t_i. \tag{3}$$

Red family and central tanks. As we constrained green to take all central tanks, Reds can't take any central tank.

▶ **Fact 2.** *For any rational ratio $r = \frac{a}{b} < \frac{1}{7}$ and any $m > 3 \in \mathbb{N}$, if*

$$t_g = \frac{5}{6}s \qquad t_c = \frac{s(\frac{1}{6} - \frac{a}{b})}{m} \qquad t_i = ms - (t_g + m(t_c + p)) = \frac{(-1+12m^2+12m(-1+\frac{a}{b})+6\frac{a}{b})s}{36(m-1)m}$$
$$p = \frac{t_c}{2m} = \frac{s(\frac{1}{6} - \frac{a}{b})}{2m^2} \qquad s = 36 \cdot 71 \cdot 10^6 \cdot b \cdot (m-1) \cdot m^3 \qquad k = \frac{p}{c_p} = s\frac{\frac{1}{6} - \frac{a}{b}}{2 \cdot 639 \cdot m^2}$$

then in any r-balanced connected partitioning green must take exactly all m central tanks (one in each macrocell plus one in the primary root macrocell).

Proof. The statement satisfies Relations (1-3), implying the result (see Appendix A.1). ◀

From Fact 2 in any r-balanced partitioning the green root must go across every macrocell, and from Fact 1 it can do so only via a connected "macropath" of degree two among all m macrocells, which corresponds exactly to a Hamiltonian cycle in the HCP instance. Conversely, a Hamiltonian cycle gives a solution for the green root to do so. In both cases each root of the red family will own the three inner tanks of its macrocell.

This reduction can straightforwardly be done in polynomial time (choose a vertex of degree 2, compute the equations given in Fact 2, and map each vertex of the hexagonal grid graph to its corresponding macrocell) which gives the result.

▶ **Theorem 1.** *rRBCP is NP-hard for any rational ratio $r < \frac{1}{7}$.*

3.4 r-balanced connected partitioning problem (rBCP)

The previously described construction also applies to rBCP (without fixed roots). Indeed, it is enough to notice that when roots are removed, then there are no other solutions to the rBCP instance than those of the rRBCP instance. This is rather straightforward.

The part owning the green tank (previously green, let us now call it *yellow*) must behave the same as green and take all central tanks (it verifies Fact 2). Then, given that yellow must not own any inner tank (Relation (3), these inner tanks must be owned by other parts (previously the red family, let us now call them the *orange family*). It is now enough to notice that each orange part must own three inner tanks belonging to a single macrocell: as green runs across each macrocell to take all central tanks, there is only one remaining entry path in each macrocell, therefore the inner tanks of a macrocell cannot be shared between two parts of the orange family without letting at least one of them under-saturated (Relations (2)). As a consequence Fact 1 also holds and the result follows.

▶ **Theorem 2.** *rBCP is NP-hard for any rational ratio $r < \frac{1}{7}$.*

4 Partitionning q-square Connected Grids

Intuitively, the difficulty of the balanced connected partitioning problem is a consequence of conflicts: if large areas are acessible from a very limited number of paths, you will have to carefully choose which part will cover which area to avoid a part blocking a crucial access.

One might consider as solution to remove these conflicts by assuring that there is always a distinct path for any part to access any area, but the usual graph theoretic notion of k-connectivity is not well suited for square grids where the degree of vertices is limited to 4 in general and which always include vertices with degree less than 2 (e.g. in the corner). Moreover, k-connectivity is a global property of a graph and difficult to verify locally. Hence the need for a new definition.

In the following, we define for each q a subclass of grid graphs called *q-square connected*, which is easy to verify and for which connected balanced q-partitions always exist and which can be efficiently computed. The idea is to make corridors (Figure 2) sufficiently large for all colors to pass. Then a greedy algorithm can expand a connected part (from a root) until it is saturated (reaches $\frac{|G_V|}{q}$ vertices), in such a way that the remaining uncolored graph remains $(q-1)$-square connected. However, while "width" seems obvious for straight "corridors", it is less evident for angles, branching, etc.

4.1 Two definitions of q-square connected graphs

We represent a "square" of a given side length by its lower left corner: for $(x, y) \in \mathbb{Z}^2$, $q > 0$, let $sq(x, y, q) = \{(x + i, y + j) \mid 0 \leq i, j < q\}$ denote *the q-square at (x, y)*. A grid graph G_V is said to *contain* $sq(x, y, q)$ iff $sq(x, y, q) \subseteq V$. A grid graph G_V is *covered by q-squares* if for each $(x, y) \in V$ there exist x', y' such that $sq(x', y', q)$ is contained in G_V and $(x, y) \in sq(x', y', q)$.

Two q-squares $sq(x, y, q), sq(x', y', q)$ are *adjacent* iff $|x - x'| + |y - y'| = 1$. For a set $V \subseteq \mathbb{Z}^2$, let $Sq(V, q) = (V', E')$ such that V' the set of q-squares in V, E' the set of adjacent pairs of q-squares in V, denote the *induced graph of q-squares*. The *distance of two q-squares $sq(x, y, q), sq(x', y', q)$ in $Sq(V, q)$* is denoted by $dist((x, y),(x', y'), V, q)$.

A grid graph G_V covered by q-squares is *q-connected* if for $sq(x, y, q), sq(x', y', q)$ in $Sq(V, q)$ with $\max\{|x - x'|, |y - y'|\} \leq q$ and $\min\{|x - x'|, |y - y'|\} < q$ we have $dist((x, y),(x', y'), V, q) = dist((x, y),(x', y'), \mathbb{Z}^2, q) = |x - x'| + |y - y'|$.

The condition for q-connectedness considers square that are overlapping (their intersection is non-empty) or touching (at least one vertex in one square is adjacent to a vertex in the other square). For such squares, we require the existence of a path of squares that somehow zigzags from one square to the other without changing the direction on either axis.

▶ **Lemma 3.** *Let a connected grid graph G_V be covered by q-squares and q-connected with $q > 1$. Then G_V is also $(q - 1)$-connected.*

The proof is given in the appendix. Now we consider an alternative definition that is equivalent, less intuitive, but better suited for the description of the algorithm.

A grid graph G_V is *q-wide* iff for all $v = (x, y) \in V$ there exist x', y' such that $0 \leq x-x' < q$, $0 \leq y - y' < q$ and for all $0 \leq i < q$ we have $(x' + i, y),(x, y' + i) \in V$, i.e. every vertex is part of a horizontal and a vertical segment of q vertices.

A grid graph G_V satisfies the *q-completion property* iff for every $(x, y) \in V$ and $a, b \in \{-1, 1\}$ and $c \in \{0, 1\}$ such that $(x + i \times a, y + j \times b) \in V$ with $i = 1$ and $1 \leq j \leq q$ or $j = 1$ and $1 \leq i \leq q$ as well as $(x + c \times a, y + (1 - c)b) \in V$ also $(x, y + j) \in V$ for all $1 \leq j \leq q$ or $(x + i, y) \in V$ for all $1 \leq i \leq q$.

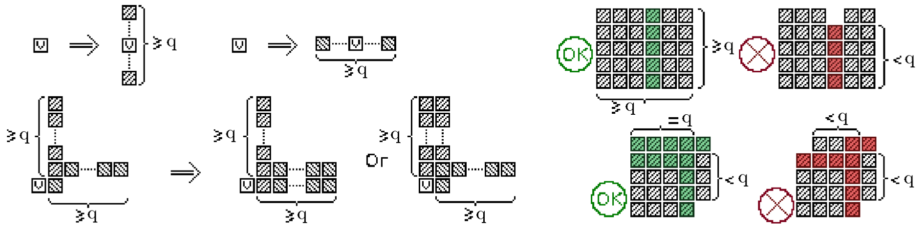

Figure 7 q-width (top) and q-completion (bottom) constraints, and examples.

The q-width and q-completion properties are depicted in Figure 7 at left ($a = b = c = 1$).

▶ **Lemma 4.** *For a connected grid graph G_V, the following two conditions are equivalent:*
1. *G_V is q-square covered and q-connected.*
2. *G_V is q-wide and satisfies the q-completion property.*

Holes in q-square connected grid graphs require a subtle definition of connectedness : two vertices $u, v \notin V$ are *close* iff their distance in $\mathbb{Z}^2$ is at most 2. For instance, two vertices on diagonal positions are close. A subset $H \subseteq \mathbb{Z}^2 \backslash V$ is *close-closed* iff for a vertex $v \in H$ and a vertex $v' \in \mathbb{Z}^2 \backslash V$ which is close to v, also $v' \in H$. The unique maximal close-closed subset $H \subseteq \mathbb{Z}^2 \backslash V$ that is infinite is called *the outside of G_V*, a finite H is called a *hole* of G_V. Note, that holes or the outside are not connected in the neighboring sense as is the grid graph. A grid graph G_V for which there exist no non-empty holes is *solid*.

A *segment* is a straight line of vertices: for $(x, y) \in \mathbb{Z}^2$, $a \in \{0, 1\}$ and $k > 0$ the *segment* $S((x, y), a, k)$ denotes the set of vertices $S((x, y), a, k) = \{(x + ai, y + (1 - a)i) \mid 0 \leq i < k\}$.

4.2 An algorithm for the unrooted problem

We now describe an algorithm for the unrooted balanced connected partitioning problem on q-square connected grid graphs. The algorithm recursively colors a connected zone of $\frac{m}{q}$ vertices leaving $\frac{m(q-1)}{q}$ vertices forming a $(q - 1)$-square connected subgraph. Then we pass to the next color and repeat the process leaving a $(q - 2)$-square uncolored remainder and so forth. This process can intuitively be compared to coloring the edges of a graph with q-wide corridors leaving $(q - 1)$-wide corridors and thus preserving sufficiently large access for $(q - 1)$ colors.

For a given q, the coloring phase of the algorithm preserves a complex invariant, *q-compliance*: a *q-compliant coloring* of G_V is a subset $C \subset V$ such that G_C is connected and $G_{V \backslash C}$ is $(q - 1)$-covered and $(q - 1)$-connected and that there exists a vertex v outside G_V with a neighbor in C.

The algorithm is easier to understand on solid grid graphs, as holes introduce an additional problem. We therefore invite the reader to first try to understand the algorithm in ignoring holes and then read the text again with regard to the way we treat holes: we *cut* them.

A hole H of a grid graph G_V is *cut by coloring C* if there exists a vertex $v \in H$ with a neighbor in C. The vertices of a cut hole are considered outside of G_V with respect to C.

The coloring phase starts by coloring an external corner (a vertex $v \in V$ with two neighbors outside G_V), and then augments q-compliant colorings by three operations:

1. Add a vertex v to C, such that $v \in V \backslash C$ and v has a neighbor in C.
2. Add a segment $S = S((x, y), a, k) \subseteq V \backslash C$ such that $k < q$, (x, y) has a neighbor in C and
   a. either all vertices in $(x + ai, y + (1 - a)i)$ have a neighbor in C or outside V_G and in particular S is terminated by $(x + ak, y + (1 - a)k) \notin V \backslash C$,
   b. or $k = q - 1$, and $(x + ak, y + (1 - a)k) \in H$ for some uncut hole H.

Algorithm 1 Simplified partitioning algorithm.

```
input int q, vertexSet V, such that G_V is q-square connected
int part= |V|/q;
// compute the set of all vertices in uncut holes
vertexSet holeVertices = computeUncutHoles(V)
while(q>0){
   vertexSet C={v} where v is an external corner of V
   while(|C| < part){
      // CLAIM 1, the following extension is always possible :
      C=C∪S for some simple extension S with C∪S q-compliant
      // remove vertices of holes that are cut
      holeVertices = updateUncutHoles(holeVertices,S)
   }
   output part C
   V=V\C , q=q-1
}
```

We call an extension S satisfying one of these three conditions a *simple extension*. Our algoritm below incrementally computes such colorings for determinig a part. An illustrating execution trace is given in the appendix.

▶ **Lemma 5** (Extension lemma). *In Algorithm 1, CLAIM 1 is always true, i.e. for a q-compliant coloring C of a graph G_V such that $|C| < \frac{|V|}{q}$ there exists simple extension S such that $C \cup S$ is q-compliant.*

▶ **Theorem 6.** *For a q-square connected grid graph G_V, Algorithm 1 terminates and computes an (up to $q-2$ vertices per part) balanced connected q-partition in time*[1] *$O(|R||V|q^2)$, where R is the smallest rectangle containing V, or in time $O(|V|^3q^2)$.*

Proof. The Extension lemma (proof: appendix) states that the line of CLAIM 1 is always possible. The invariant ensures that each iteration of the outer loop works on a q-compliant coloring. The termination condition of the inner loop guarantees that after the loop $\frac{|V|}{q} \leq |C| < \frac{m}{q} + q - 1$, the maximal deviation of a part. The algorithm is simplified and can, under bad circumstances, produce a final partition that lacks $(q-1)^2$ vertices, but it is easy to improve this to $q-1$ by balancing the limit between iterations of the outer loop. Moreover, a simple improvement of the algorithm could distributed the last segment attributed to a part in the inner loop between the current part and the next part, limiting the deviation from $\frac{|V|}{q}$ to at most 1 and in fact to 0, perfect balance, in the case of $|V|$ a multiple of q.

For the complexity considerations, we suppose a matrix representation of the input V as a subset of a rectangle R. This rectangle can of course be computed in $O(|R|)$ and $|R| \leq |V|^2$ since V is connected. For practical purposes, the problem can be stated such that $|R|$ itself is a measure of the input size, but for the question of Algorithm 1 being polynomial or not, this detail is of no consequence.

The functions `computeUncutHoles` and `updateUncutHoles` compute reachable sets, e.g. by depth first search, in linear time. Moreover, since `updateUncutHoles` removes vertices from the set of uncut holes and removes each vertex at most once, the total computation time for `updateUncutHoles` is linear in $|R|$.

[1] We neglect non-constant access time to the matrix R, which is of no practical consequence.

Since each iteration of the inner loop increments C and each iteration of the outer loop reduces V by C, the total number of iterations of the inner loop is limited by $|V|$. In each iteration of the inner loop, in principle, all vertices of $V \backslash C$ have to be examined for a possible simple extension. The examination of whether the extension violates the q-compliance however is local around the extension point and can be done[2] in $O(q^2)$. Note also that there is no need to test econnectedness of the uncolored part after an extension: indeed, each path leading through the extension can be replaced by a path avoiding it: width constraints imply this for extensions of type (1) and (2a), whereas an uncut hole is surrounded by a cyclic path that also allows to circumvent the cutting segment. ◀

5 Perspectives and future work

Algorithm optimization. Above, we stated the complexity of Algorithm 1 as $O(|R||V|q^2)$ or $O(|R|^2q^2)$. The complexity is quadratic in $|R|$ because of blindly sweeping R at each iteration in search of the next extension. Several options for improvement exist, for instance, tracking in parallel C and a doubly linked list of actually possible simple extensions in connection with R, we can always pick the first extension of the list and update the list in each iteration in essentially $O(q^3)$ since the impact to other extensions is local. Cutting holes may temporarily require slower updates, but their global impact remains linear in $|R|$. On the other hand, prioritizing simple extensions may allow to bound the set of extensions at any point of the algorithm execution, potentially bringing the complexity to $O(|R|q^2)$.

Towards an algorithm for the rooted problem. We conjecture that Algorithm 1 can be modified to solve the rooted variant of the BCP as follows: In the rooted variant, we start with a grid graph G an a set of q vertices that are already colored with distinct colors: $\forall 1 < i \leq n, C_i = \{v_i\}$, where the C_i designate parts we are building. Here, we cannot greadily extend colors one after the other and keep the q-connection of the remaining uncolored subgraph $G \backslash \bigcup C_i$. Indeed, while a part C_i is not saturated, this part must keep at least a single access to the uncolored part of the graph, i.e $\exists (u, v), u \in C_i, v \in G \backslash \bigcup C_i$. This greatly restricts the possibilities for extensions if we consider a single color at any step.

Instead of extending a single color until the corresponding part is saturated, we can search possible extensions of any color that keeps unsaturated part connected to the uncolored graph, i.e, we search for a set S of maximum $q-1$ vertices $v_1...v_q$ that we will add to a colored part C_i, such that $G \backslash \cup C_i$ is $q-1$-connected and $\forall 0 < i < q, \exists (u, v), u \in C_i, v \in G \backslash \cup C_i$. When a colored part C_i becomes saturated, we can remove it and recursively consider the smaller problem on the subgraph $G \backslash C_i$, with the remaining $q-1$ unsaturated parts still connected to the uncolored subgraph which is $(q-2)$-connected.

An important difficulty introduced in the rooted problem is the possible emergence of access conflicts that will exclude a completion of the partition. If these conflicts are not avoided, the greedy algorithm can get stuck. As an example of a simple conflict, consider two unsaturated parts C_i and C_j which both access the uncolored part of the graph by same single vertex. Obviously, such a coloring cannot be completed (c.f. Figure 8 at left for green and red). More general cases can involve up to q parts when all the k parts get their unique

[2] Whereas extensions by a single vertex can produce a violation of q-width or the q-completion property, the extensions by a segment S can only produce a violation of the q-width property.

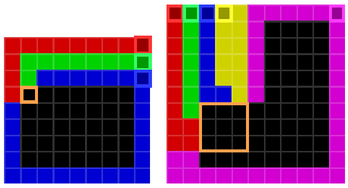

Figure 8 Binary (red and green, at left), and 4-nary (red, green, blue, yellow, at right) conflicts.

access to the uncolored subgraph into the same square of size $(k-1) \times (k-1)$: In Figure 8 (right), extending either red, green, blue, or yellow will recursively create a $(q-1)$-ary conflict between the remaining colors.

We conjecture that the Extension lemma can be modified to take into account such q-ary conflicts and avoid them, at the price of a limited imbalance when approching the saturation of all colors. In the future, we will try to extend Algorithm 1 with this reasoning.

References

1 Ronald Becker, Isabella Lari, Mario Lucertini, and Bruno Simeone. Max-min partitioning of grid graphs into connected components. *Networks: An International Journal*, 32(2):115–125, 1998.

2 Frédéric Chataigner, Liliane RB Salgado, and Yoshiko Wakabayashi. Approximation and inapproximability results on balanced connected partitions of graphs. *Discrete Mathematics and Theoretical Computer Science*, 9(1):177–192, 2007.

3 Janka Chlebíková. Approximating the maximally balanced connected partition problem in graphs. *Information Processing Letters*, 60(5):225–230, 1996.

4 Martin E Dyer and Alan M Frieze. On the complexity of partitioning graphs into connected subgraphs. *Discrete Applied Mathematics*, 10(2):139–153, 1985.

5 Andreas E Feldmann. *Balanced partitioning of grids and related graphs.* ETH Zurich, 2012.

6 Andreas Emil Feldmann. Fast balanced partitioning is hard even on grids and trees. *Theoretical Computer Science*, 485:61–68, 2013.

7 Kamrul Islam, Henk Meijer, Yurai Núñez Rodríguez, David Rappaport, and Henry Xiao. Hamilton circuits in hexagonal grid graphs. In *CCCG*, pages 85–88, 2007.

8 László Lovasz. A homology theory for spanning tress of a graph. *Acta Mathematica Hungarica*, 30(3-4):241–251, 1977.

9 Jun Ma and Shaohan Ma. Ano (k 2 n 2) algorithm to find ak-partition in ak-connected graph. *Journal of Computer Science and Technology*, 9(1):86–91, 1994.

10 Srinivasan Murali and Giovanni De Micheli. Bandwidth-constrained mapping of cores onto noc architectures. In *Proceedings of the conference on Design, automation and test in Europe-Volume 2*, page 20896. IEEE Computer Society, 2004.

11 Bang Ye Wu. A 7/6-approximation algorithm for the max-min connected bipartition problem on grid graphs. In *Computational Geometry, Graphs and Applications*, pages 188–194. Springer, 2011.

A Appendix

A.1 Idea behind the solution given in Fact 2

In Fact 2, we gave a set of dimensions for macrocells to build a square grid instance that satisfies Relations (1) to (3) for given values of tolerance ratio r and number of macrocells m. Let us present the reasoning behind these values.

As the cumulated surface of three inner tanks will be chosen close to the saturation threshold s, we first choose the size of the green tank $t_g = \frac{5}{6}s$, so green will be unable to take inner tanks (Relation (3) right).

Then we choose the size of central tanks so that green exactly enters the saturation interval by covering the green tank and all the central tanks: $t_g + mt_c = s(1-r)$ (Relation (3) left). Now we choose total path length in such a way that it is neglectible in front of tanks: $mp = \frac{t_c}{2}$, so collecting all the path vertices in the topology accounts for less that collecting a single central tank (mp in Relations (2) and (3)). The inner tanks get all the remaining vertices: $mt_i = ms - t_g - m(t_c + p)$ (Relation (2)).

Finally, the last dimension to set is the saturation threshold s. As all previously set dimensions must be integers and are multiples of s, we choose s to be a multiple of all denominators. Finally we also take s sufficiently large so that there exists $k = \frac{p}{c_p}$ such as $p \geq k \times c_p$, *i.e.* there is enough path vertices to wire all the tanks (Relation (1)).

A.2 Proofs of Lemma 3 and Lemma 4

In order to prove these properties, we first prove auxiliary observations:

▶ **Lemma 7.** *Let a connected grid graph G_V be covered by q-squares and q-connected. Then the induced graph $Sq(V,q)$ is also connected.*

Proof. We show that for any two squares $sq(x_1, y_1, q)$ and $sq(x_2, y_2, q)$ in G_V there exists a path in $Sq(V, q)$. The proof uses induction on the least distance d of pairs of vertices $(x'_1, y'_1) \in sq(x_1, y_1, q)$ and $(x'_2, y'_2) \in sq(x_2, y_2, q)$: if this distance is 0 then the squares are intersecting and the q-connected property implies that there is a path between the two squares. Otherwise, there is a first vertex (x'_3, y'_3) on the shortest path from (x'_1, y'_1) to (x'_2, y'_2) that is not in $sq(x_1, y_1, q)$. Since G_V is q-square covered, there exists $sq(x_3, y_3, q)$ in V_G with $(x'_3, y'_3) \in sq(x_3, y_3, q)$. Moreover, because of q-connectedness, there exists a path from $sq(x_1, y_1, q)$ to (x_3, y_3, q) in $Sq(V, q)$. On the other hand, the distance of (x'_3, y'_3) and (x'_2, y'_2) is less than d, hence by induction, there exists a path from $sq(x_3, y_3, q)$ to $sq(x_2, y_2, q)$ in $Sq(V, q)$ and the two paths joint connect $sq(x_1, y_1, q)$ and $sq(x_2, y_2, q)$ as desired. ◀

▶ **Lemma 8.** *Let a connected grid graph G_V be covered by q-squares and q-connected with $q > 1$. Then each $(q-1)$-square in G_V is contained in a q-square of G_V.*

Proof. Sketch: let $sq(x, y, q-1)$ be a $(q-1)$-square. In order to show the existence of a q-square in G_V containing it, we first show that there exists a q-square containing one or several lines from one side of the $(q-1)$-square and from there by induction we show that there exists a q-square containing all the $(q-1)$ lines from that side. The proof necessarily makes use of the q-connected property. ◀

Proof of Lemma 3. Suppose that two $(q-1)$-squares are touching or intersecting. Any two q-squares containing them also touch or intersect. We consider two q-squares containing them of minimal distance and consider a path between the two. A path consists of steps "up", "left", "right" or "down" and it never contains both "up" and "down" on the one hand or "left" and "right" on the other. A $(q-1)$-square can imitate the same directions and always be included in the q-square of a path, however, it may end up on the "wrong" among the 4 $(q-1)$-squares in the final q-square and require correction. However, it is easy to see that this correction can either be achieved by skipping a step of the q-square path (thus relatively moving in the opposite direction) or by adding a step in the final q-square and we thus end up with a path of the desired distance. ◀

Proof of Lemma 4. (1) implies (2): q-width trivially follows from q-square covering. Suppose a constellation (x, y) and a, b, c concerning the q-completion property. Without loss of generality we suppose $a = b = c = 1$, other cases are very similar. Supposing the property fails to hold for this instance of parameters, then there exist vertices $(x + i, y), (x, y + j) \notin V$ with $1 \leq i, j \leq q$. We suppose i, j minimal, i.e. for all $1 \leq i' < i$ we have $(x + i', y) \in V$ and $(x, y + j') \in V$. We show the existence of q-squares in G_V that touch or overlap, yet violate the q-connected property because of the obstacles represented by $(x + i, y), (x, y + j)$. First assume $j = 1$, then the only q-square containing $(x + 1, y + 1)$ can be $sq(x + 1, y + 1, q)$. This q-square touches the q-square containing $(x + 1, y)$ yet a direct path to the letter q-square starting at $sq(x + 1, y + 1, q)$ would have to start either down or left, but there are no edges in that direction, a contradiction. So $j > 1$. Now consider the q-squares containing vertices $(x + i', y + 1)$ with $0 \leq i' <\leq i$, all of which belong to G_V. Some of them (e.g. $(x, y + 1)$) must be below $y + j' - q + 1$, others above $y + 1$ and hence there must exist an i' such that $(x + i', y + 1)$ is in a q-square below $y + j' - q + 1$ and $(x + i' + 1, y + 1)$ in a q-square above $y + 1$ and these q-squares touch or intersect, yet there is no direct path between them because of the two obstacle vertices. Another contradiction. Hence, we cannot at the same time have both $(x + i, y), (x, y + j) \notin V$, hence the constellation satisfies the q-completion property.

(2) implies (1):

First we show that the grid graph is q-square covered. We do this by a construction of coloring a part of the vertices of the graph incrementally in such a way, that the colored zone is invariantly a union of q-squares. We show that this process can be continued until all vertices are colored. The construction starts by choosing a vertex (x, y) with x minimal. Since the graph is q-wide this gives us q vertices $(x, y_1)...(x, y_q)$ with $y = y_i$, all with x minimal. Again because of q-width, it follows that $sq(x, y_1, q) \subseteq V$ and this is the first square we color. Now suppose that there are uncolored vertices left. Since the graph is connected, at least one such vertex is adjacent to a colored square. Without loss of generality, let us suppose $sq(x, y + 1, q)$ fully colored and $(x + i, y)$ with $0 \leq i < q$ not yet colored. Because of q-width, there exists x' such that $x' + k = x + i$ with $0 \leq k < q$ and for all $0 \leq l < q$ we have $(x' + l, y) \in V$. We suppose furthermore x, y, i, x' chosen with these conditions such that $|x - x'|$ is minimal. Then either $x = x'$, but then $sq(x, y, q) \subseteq$ and we can color $(x, y)...(x + q - 1, y)$ and continue. Or $x! = x'$ and let us suppose that $x' < x$. Then the preconditions for q-completion are satisfied for $(x - 1, y)$ and $a = b = c = 1$ and either completion contradicts the assumption of having chosen x, y, i, x' such that $|x - x'|$ is minimal.

Now we want to show that G_V is q-connected. Supposing it is not, then there are two q-squares $sq(x, y, q), sq(x', y', q)$ such that $\max\{|x-x'|, |y-y'|\} \leq q$ and $\min\{|x-x'|, |y-y'|\} < q$ but their distance in $Sq(V, q)$ is greater than $|x - x'| + |y - y'|$. Assume $|x - x'| + |y - y'|$ minimal for such a pair of q-squares. Obviously, $\min\{|x - x'|, |y - y'|\} > 0$ otherwise a straight path between the two q-squares obviously exists contradicting assumptions. The non-trivially overlapping or touching squares give rise to a constellation of the q-completion property which in turn yields a third square at distance 1 from one of the two squares and 1 unit closer to the other square. Since we assumed $|x - x'| + |y - y'|$ minimal, this results in an intermediate q-square on a path of length $|x - x'| + |y - y'|$ between the original q-squares, contradicting assumptions. ◀

A.3 Proof of the Extension Lemma

The proof of the Extension lemma is based on a (constructive) recursive analysis of a q-compliant coloring: it is based on the identification of potential extension points, and if the latter induces a conflict to q-compliance, there is a way to subdivide the area of extension points until necessarily we reach an area too small to contain a conflict.

In order to formalize the search area, we introduce the notion of *coloring border* as the edge between the outside and the colored part of V_G on the one hand and the uncolored part of V_G on the other hand: Visually, this border is between vertices and we formalize it as couples: let (u, v) be a pair of neighboring vertices such that $v \in V\backslash C$ and $u \in C$ or $u \in H$ for some cut hole of V_G or such that u is outside of V_G *coloring border candidate.* If $u \in C$ we call it a *colored edge*, in the case of $u \in H$ we call it an *uncolored edge.* Coloring border candidates $(u, v), (u', v')$ are *connected* iff u is a neighbor of u' and v is a neighbor of v' or $u = u'$ and $v = (x, y), v' = (x', y')$ satisfy $|x - x'| + |y - y'| = 2$, i.e. they are on diagonal positions, or conversely $v = v'$ and u, u' are on diagonal positions. Graphically, connected coloring border candidates can be drawn without lifting the pen and share a common point. An extension point is a coloring border candidate (u, v) with $u \in C$ and v the first vertex of a simple extension S. A *coloring border* is a connected set of coloring border candidates.

A coloring border B is *promising* if

- it contains colored segments;
- for any simple extension $S(x, y, a, k)$ with (x, y) having a colored neighbor on the border its addition to C either preserves q-compliance or it causes a conflict with an uncut hole or with a colored edge that is either on the coloring border or directly connected to it;
- one of the following cases holds :
  1. It contains uncolored segments $S(x, y, a, q-1)$ such that all vertices are connected to coloring border candidates and such that $(x - a, y - (1 - a)), (x + a(q - 1), y + (1 - a)(q - 1)) \notin V\backslash C$.
  2. It does not satisfy (1), there exists a vertex $v \in V\backslash C$ having one neighbor $v_1 \in C$ and another neighbor $v_2 \in \mathbb{Z}^2\backslash V$ (a corner with one side colored, the other outside V), with (v, v_1) in the coloring border.
  3. It does not satisfy (1) or (2), but it contains edges (v, v') with $v' \in \mathbb{Z}^2\backslash V$.
  4. It does not satisfy (1) or (2) or (3), but it contains two "colored corners", i.e. pairs of edges (v_i, v_i') and (v_i, v_i'')such that $v_i \in V\backslash C$ and $v_i', v_i'' \in C$ (concave corners with color on both sides), and such that no convex corner (with color on both sides) exists between them (such pairs of concave corners naturally contain a segment all along the edges between the two corners).

Proof of Lemma 5 (Extension lemma). By induction, we claim that a promising coloring border contains an extension that is q-compliant. Then, since the complete border of the partially colored graph is trivially promising, the result follows.

Now let B be a promising coloring border. We examine the cases and show that for each case either a q-compliant simple extension can be identified, or that a smaller promising coloring border can be identified. Case (1) allows immediate coloring, as such a segment cannot break $(q-1)$-completion and the only way it could break the $(q-1)$-width would be if $|V\backslash C| = (q-1)^2$, which contradicts the assumptions $|C| < \frac{|V|}{q}$.

Suppose case (2): adding a single vertex in the corner cannot break $(q-1)$-width, because this contradicts that (1) is not satisfied. Without loss of generality, let us assume $v=(x,y)$ such that $(x-1,y)\in C$ and $(x,y-1)\notin V$ (other cases are symmetric). In order to generate a conflict to $(q-1)$-completion, we must have $(x+q-1,y),(x+q-2,y)\ldots(x+q-2,y+q-1)\in V\backslash C$ but that $(x',y')\notin V\backslash C$ for an $(x',y')\in\{(x+q-1,y+1),\ldots,(x+q-1,y+q-1)\}$. Since V_G is q-square connected, it is not difficult to see that $\{(x+q-1,y+1),\ldots,(x+q-1,y+q-1)\}\subseteq V$ hence $(x',y')\in C$. With $(x'-1,y')\in V\backslash C$ it forms a coloring border candidate and since the coloring border is promising, it is connected to the corner. Considering the shortest path between the corner and the conflicting edge, then by removing the edge linking the corner to the colored vertex, we obtain a new coloring border. We claim that it is promising : it contains concave corners because of the orientation of the edges at the two vertices, and moreover, if an extension of any of the types (1)-(4) on this sub-border have conflicts, they must occur within the sub-border.

For the case (3), let us suppose (other cases are symmetric) that $v=(x,y),(x+1,y)\in V\backslash C$ and moreover $(x,y-1)\in C$ and $(x+1,y-1)\notin V$. If adding v produces a conflict to q-compliance, it is either a conflict to $(q-1)$-width or to $(q-1)$-completion.

Let us first consider the case of an $(q-1)$-width conflict. Such a conflict can concern $(x,y+q-2)$, $(x-k,y)$ or $(x+k,y)$ for some $1<k<q-1$. In the first case, since G_V is q-square connected, it is easy to see that $(x,y+q-2)\in V$, and consequently, it is in C. By removing the edges $(x,y),(x,y-1)$ and $((x,y+q-2),(x,y+q-3))$ from the border coloring and by keeping a path between these two edges connecting them and containing colored edges, we obtain a coloring border of reduced size, which we claim to be promising : for example, the two removed edges are parallel and the connecting path necessarily contains concave corners (otherwise, we end up with a contradiction to the assumption that (1) does not hold), and finally, given that the remaining coloring border is “almost closed”, a conflict with respect to some extension point cannot be situated beyond the two removed edges. In the second case, we consider the simple extension $(x-k+1,y)$. It cannot have an $(q-1)$-width conflict (otherwise we would have (1), but we assume that we did not) and because of $(x+1,y-1)\notin V$ we find that a conflicting vertex with respect to $(q-1)$-completion must be $(x-k+q-1,y+q-2)\in C$ we apply a similar reasoning as above in order to identify a sub coloring border. Finally, In the third case (and excluding the second case), the segment $\{(x,y),\ldots,(x+k-1,y)\}$ is a simple extension that cannot introduce a conflict to $(q-1)$-width and as before, if there is a conflict to $(q-1)$-completion, this conflict is between the edge below (x,y) and some colored edge and again, we apply a reduction of the coloring border. In any case, either the extension preserves q-compliance, or the reduced coloring border is promising and contains by induction an extension preserving q-compliance, that extends obviously to the complete bordering color.

For the case (4), we suppose (up to symmetry) corners $(x,y),(x+k,y)$ with $(x-1,y),(x+k+1,y)\in C$ and $(x,y-1),\ldots,(x+k,y-1)\in C$. If adding (x,y) to C causes a conflict, it is either due to a colored vertex and we proceed by induction as above. If, on the other hand, there exists a conflicting vertex $v'\notin V$ then this vertex cannot be on the coloring border (assumption of case (4)), hence it must be part of an uncut hole at position $(x+q-1,y+q-1)$. We claim that in this case, the simple extension $S=\{(x+q-1,y),\ldots,(x+q-1,y+q-2)\}$ preserves q-compliance. First, it cannot introduce a conflict with $(q-1)$-completion, but also, it cannot introduce a conflict with $(q-1)$-width. Also observe that the uncut hole is surrounded by some closed path in $V\backslash C$, cutting it cannot cause $V\backslash(C\cup S)$ to become disconnected. ◀

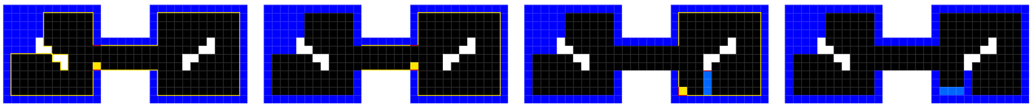

Figure 9 Promising coloring border and extension search.

The recursive reasoning of the proof is illustrated in Figure 9 for the case $q = 4$, from left to right: (a) A partial coloring and a promising coloring border (initially the inner contour) is given and a possible simple extension (Case 3) is shown in yellow, but it induces a conflict concerning $(q-1)$-width, the edges cut are marked red, we keep the path connecting these edges at right. A second attempt of Case 3 is shown in the second picture. In the third picture, there are no more uncolored edges (Case 4) in the coloring border so we try a concave corner, but there is a conflict with an uncut hole, so we cut it instead, which conclude the search for the extension. The last picture shows an application of Case 1.

A.4 Example computation of our coloring algorithm

Below, we show a trace of the partitioning by Algorithm 1 for illustration.