Local Edge Colouring of Yao-like Subgraphs of Unit Disk Graphs

J. Czyzowicz[‡‡][**] S. Dobrev[†][**] E. Kranakis[‡][**][††] J. Opatrny[§][**] J. Urrutia[‖]

January 25, 2007

Abstract

The focus of the present paper is on providing a local deterministic algorithm for colouring the edges of *Yao-like* subgraphs of Unit Disc Graphs. These are geometric graphs such that for some positive integers l, k the following property holds at each node v: if we partition the unit circle centered at v into $2k$ equally sized wedges then each wedge can contain at most l points different from v. We assume that the nodes are location aware, i.e. they know their Cartesian coordinates in the plane. The algorithm presented is local in the sense that each node can receive information emanating only from nodes which are at most a constant (depending on k and l, but not on the size of the graph) number of hops away from it, and hence the algorithm terminates in a constant number of steps. The number of colours used is $2kl + 1$ and this is optimal for local algorithms (since the maximal degree is $2kl$ and a colouring with $2kl$ colours can only be constructed by a global algorithm), thus showing that in this class of graphs the price for locality is only one additional colour.

Key Words and Phrases: Edge Colouring, Geometric Graphs, Unit Disk Graphs, Local Algorithm, Location Awareness, Wedge, Wireless Network.

[‡‡] Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada

[†] School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada, on leave from Slovak Academy of Sciences, Bratislava, Slovakia

[‡] School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada K1S 5B6.

[§] Department of Computer Science, Concordia University, 1455 de Maisonneuve Blvd West, Montréal, Qubec, Canada, H3G 1M8.

[‖] Instituto de Matemáticas, Universidad Nacional Autónoma de México, Área de la investigación cientifica, Circuito Exterior, Ciudad Universitaria, Coyoacán 04510, México, D.F. México.

[**] Research supported in part by NSERC (Natural Science and Engineering Research Council of Canada).

[††] Research supported in part by MITACS (Mathematics of Information Technology and Complex Systems).

1 Introduction

The problem of graph edge coloring consists of associating colors to the edges of the graph, so that no two adjacent edges are of the same color. The minimal such number of colors required is called *edge chromatic number* of the graph. The well-known theorem of Vizing [1964] implies that the edge chromatic number of a simple graph is either Δ or $\Delta + 1$, where Δ is the maximum vertex degree of the graph. For arbitrary graphs the problem was shown to be NP-complete by Holyer [1981].

The interest in the wireless networks research community on the edge coloring problem (see Ramanathan [1999]) comes from its applications, e.g. in packet scheduling (Kumar et al. [2004]), channel assignment (Kodialam and Nandagopal [2005]) or link scheduling (Gandham et al. [2005]). Many wireless and ad hoc networks are modeled by *Unit Disk Graphs* (UDGs), where nodes are embedded in the plane and two nodes are adjacent if and only if their Euclidean distance is at most one. In this model it is assumed that wireless nodes possess the same transmission range and two nodes can communicate when they are in each other's transmission range.

To reduce network complexity and enable important network functions, researchers often construct network *spanners*, whereby certain edges of the network are being omitted from consideration while at the same retaining network connectivity. Although many different types of spanners have been considered in the literature, good spanners must possess some useful properties like small node degree, low edge set complexity and stretch factor. Another important property of spanners for wireless and ad hoc networks is that they may enable *local computations*, i.e. distributed algorithms that never need to transmit messages more than a constant number of hops from source to destination. Introduced in the seminal work of Linial [1992] the concept of locality has been explored in different contexts (cf. Peleg [2000]). It follows that in this model, a *local algorithm* is completed in a constant number of steps and thus its complexity is independent of the size of the network. In dynamically changing, in mobile wireless as well as in ad hoc networks locality has proven to be particularly important, because eventual changes are *localized* and need only be performed within the areas affected, without disturbing the entire system.

The most popular spanners of UDGs, which may be computed locally include relative neighborhood graphs, Gabriel graphs and Yao graphs. However among them only Yao graphs achieve *bounded stretch factor* (or dilation), i.e. the distance of two nodes in the spanner is at most a constant factor larger than their distance in the original graph. In this paper we propose local algorithms for edge coloring of (l, k)-edge/wedge graphs. These graphs, besides being generalizations of Yao graphs, include other classes of subgraphs of UDGs, which appear in the literature, like local approximations of minimum spanning trees, Delauney triangulations or half-space proximals (most typically with $l = 1$ and $k = 3$). Notwithstanding proposed heuristics and algorithms for coloring UDGs (e.g. Marathe et al. [1995], Matsui [1998], Barrett et al. [2006]), none of these approaches address edge coloring under the locality conditions defined above. We assume a geometric network with *location aware* nodes, i.e. nodes that know their Cartesian coordinates in the plane.

1.1 Preliminaries and results of the paper

Consider a class of graphs for which the following property holds in each node u. For some integer parameters k and l, if we divide all possible edge directions into $2k$ equally sized wedges (or cones) of width π/k then at most l edges incident to v lie in any fixed wedge. Figure 1 depicts such a partition into eight wedges ($k = 4$) with at most three points per wedge ($l = 3$). We will call such graphs (l, k)-*edge/wedge graphs*. This gives rise to a natural partition of the edges: we partition the edges into k classes $\mathcal{C}_i$ for $i = 0, 1, \dots, k-1$ (edges of two opposite wedges–called double-wedge–are put into the same class). Notice that we want an edge to belong to the same class when looking from both its endpoints, which also explains why we do not consider the case of odd number of wedges. More formally, for $i \leq k-1$, $\mathcal{C}_i$ is the set of all directed edges (u, v) such that the edge's slope lies either within the wedge $\langle i\pi/k, (i+1)\pi/k)$ and the edge's length $|u, v| < 1$, or within the wedge $\langle \pi + i\pi/k, \pi + (i+1)\pi/k)$ and $|u, v| \leq 1$; we call the former *upstream edges* and the latter *downstream edges*. A path p is said to belong to the class $\mathcal{C}_i$, and abusing notation we abbreviate this with $p \in \mathcal{C}_i$, if each of the edges of p is from the class $\mathcal{C}_i$.

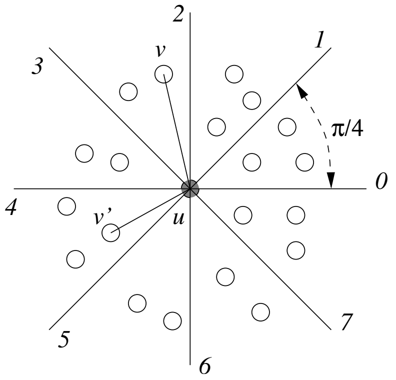

Figure 1: 8 wedges of width $\pi/4$ around the vertex u. Further, we assume that the horizontal slope is not internal to any wedge. Depicted are the eight wedges numbered $0, 1, \dots, 7$, an upstream edge (u, v) and a downstream edge (u, v').

Many subgraphs of UDGs that appear in the literature are in fact (l, k)-edge/wedge graphs, for some l, k. These include Yao graphs as defined in Yao [1982], for k chosen arbitrarily as desired. Similarly, Local Delaunay in Li et al. [2003], Local Minimum Spanning Tree (LMST) Spanners in Chavez et al. [2006b], Wang and Li [2003], and Half-space proximal in Chavez et al. [2006a]) with typical values $l = 1$ and $k = 3$. Interestingly, for many subgraphs of unit disc graphs either there is also a lower bound on the sharpest angle between incident edges (e.g., in LMST as given in Chavez et al. [2006b], Wang and Li [2003] and Half-space proximal Chavez et al. [2006a]) or by construction there is an explicit limit on the number of edges per wedge (e.g. Yao graphs in Yao [1982] and their variants Chavez et al. [2006b]). This means that many spanners of geometric graphs are, in fact, (l, k)-edge/wedge graphs. Finally, some variants of Yao graphs in Yao [1982] are (l, k)-edge/wedge graphs for $l > 1$ (e.g. take in each wedge both the shortest and the longest edge).

1.2 Results and structure of the paper

The focus of the present paper is on providing deterministic, local algorithms for edge-colouring (l, k)-edge/wedge subgraphs of unit disk graphs. The wireless networks considered consist of nodes that have the same *circular* transmission range of size 1. We assume location awareness, i.e., each node of the graph knows its geometric position in the plane. The main result of the paper is a new local, deterministic algorithm for edge-colouring (l, k)-edge/wedge subgraphs of UDGs, for all integers l, k. An important parameter in our algorithms is the *horizon distance* d: a given node u never needs to be aware of the location of nodes beyond its horizon, as measured by the euclidean distance from u. More specifically, in Section 2 we give the basic $2k+1$ edge colouring algorithm for (l, k)-edge/wedge subgraphs of UDGs using a local horizon distance $7.81 \cdot lk$. Section 3 contains two lower bounds. First, we show that no colouring can be locally constructed in UDGs if there is no geometric information available to the nodes, and second give a lower bound of $d = 2kl$ for the horizon distance used in our algorithm. In Section 4 we give improved constructions for the practically important cases of $k = 3$ and $k = 4$ and in the Appendix we provide several refinements and improvements on the basic construction, as well as an asymptotic analysis for large k.

2 Basic Construction

In this section we provide the basic construction in detail. To clarify the ideas we first consider the case of at most one point per wedge and later show how to generalize it to multiple points per wedge.

2.1 Virtual cutting lines and grids

Consider first the case $l = 1$ whereby for each node there is at most one point per wedge. The edges of class $\mathcal{C}_i$ form a set of vertex-disjoint paths. If we colour each path of class $\mathcal{C}_i$ by alternating the colours $2i$ and $2i + 1$, we get a globally consistent edge colouring of the whole graph. However, such an algorithm is inherently global, in fact, even a single path cannot be 2-coloured using a local algorithm since the edges at the endpoints would have to decide on their colour locally, in which case an adversary could adjust the number of edges in the middle segment (which are beyond the horizon of the endpoints) in order to force the use of the third colour.

Therefore the first idea in our algorithm is to use the geometric information to design a tiling of the plane containing the network. This way we can cut the potentially long paths of edges of the same class into path segments of bounded size, each one within a single tile, locally colour the segment edges and use a third colour (per class) to resolve conflicts on the segment boundaries. Let us look more closely at one of the classes $\mathcal{C}_i$, for $i = 0, 1, \ldots, k - 1$. The cutting of paths of class $\mathcal{C}_i$ can be achieved by selecting a set of virtual cutting lines (abbreviated VCL) which are normal to the axis of the double-wedge corresponding to $\mathcal{C}_i$ and spaced at a distance more than 2. We ensure that each edge $e \in \mathcal{C}_i$ intersecting such cutting line will take the third colour, say $2k + i$. The placement of VCLs inside each tile will be identical with respect to the tile boundaries. Hence, the nodes at the endpoints of edges which are cut by the VCLs will use the geometric information to identify the edges which will receive the additional third colour. The fact that the cutting lines are normal to the wedge as well as sufficiently separated ensures that there are no two adjacent edges coloured by the same third colour (if two edges share a node located exactly on a cutting line, only one of them gets assigned the third colour). In this approach the VCLs of different classes do not interfere and can be considered separately, thus resulting in a colouring using overall $3k$ colours.

The main technical contribution of this paper comes from the realization that the VCLs can be designed in such a way that a single conflict-resolution colour can be shared by all classes without interference. The crucial concept is that of *d-cutting the class $\mathcal{C}_i$*, which refers to a set S of line segments that intersects every path $p \in \mathcal{C}_i$ with Euclidean diameter at least d. More formally we define

Definition 1 (d-cutting the class $\mathcal{C}_i$) *We say that a set S of line segments in a plane is d-*cutting *the class $\mathcal{C}_i$ if, for each path $p \in \mathcal{C}_i$ it is true that*

- *if the Euclidean diameter of p is at least d then p intersects S, and*
- *two consecutive edges of p may intersect S only if their shared endpoint is in S*

Note that S is an infinite set; typically it is a pattern of line segments that is repeated infinitely. Also observe that we can deal with the case $l > 1$ by locally ordering (arbitrarily) the edges of class $\mathcal{C}_i$ incident to a vertex v. In this case the edges of the same class and the same rank within this class form a set of vertex disjoint paths. This means that we need as many as lk VCLs (l for each class) which we denote by VCL_i^j for $i = 0, 1, \ldots, k-1$ and $j = 0, 1, \ldots, l-1$, with each VCL_i^j d-cutting the class $\mathcal{C}_i$.

The following definition captures the area reachable by edges of $\mathcal{C}_i$ which also intersect VCL_i^j. Namely we have

Definition 2 (Reachability by edges of $\mathcal{C}_i$) *Let us denote by MS_i^j the Minkowski sum of VCL_i^j and $\mathcal{C}_i$, that is $\mathsf{MS}_i^j = \{v | u \in \mathsf{VCL}_i^j \text{ and } (u, v) \in \mathcal{C}_i\}$*

As we require that the VCLs do not interfere with each other, a *Virtual Cutting Grid* is a union of VCLs for each class and each rank such that their Minkowski sums are pairwise disjoint:

Definition 3 (d-Virtual Cutting Grid (VCG)) *S is a d-Virtual Cutting Grid (abbreviated VCG) for (l, k)-edge/wedge graphs if*

- $S = \bigcup_{i=0}^{k-1} \bigcup_{j=0}^{l-1} \mathsf{VCL}_i^j$,
- *each VCL_i^j is d-cutting $\mathcal{C}_i$, and*
- $\mathsf{MS}_i^j \cap \mathsf{MS}_{i'}^{j'} = \emptyset$, *for all i, j, i', j' such that $i \neq i'$ or $j \neq j'$.*

Algorithm 1 Local Edge Colouring Algorithm

```
Input: $k$ and an implicitly known $d - \mathsf{VCG}$
    // The algorithm at vertex $u$: Independently process each downstream edge $(u, v)$.
1: Determine your class $\mathcal{C}_i$; locally sort all edges of class $\mathcal{C}_i$ incident to $u$ in clockwise
       direction and determine your rank $r$ (from 0 to $l-1$) – this determines $\mathsf{VCL}_i^r$ relevant
       to you.
       // Learn your path upstream up to Euclidean distance $d$.
2: Broadcast your location to all reachable points downstream along the path of class $\mathcal{C}_i$
       and rank $r$; forward downstream any message received from upstream if its origin is
       closer then $d$.
3: if intersecting $\mathsf{VCL}_i^r$ in a node different then $v$ then
4:     select colour $2kl$
5: else if $e$ is the first edge of the path, or immediately follows (in its path) an edge
       coloured $2kl$ then
6:     select colour $2(il + r)$
7: else
       // This is local computation using the information obtained in line 2.
8:     select colour $2(il + r)$ or $2(il + r) + 1$, depending on the parity of the path from an
       edge coloured by the previous rule, ensuring the colours alternate.
9: end if
```

2.2 Local edge coloring algorithm

Before proceeding with the construction of the VCG, let us formally present the colouring Algorithm 1.

The following lemma asserts the locality of the algorithm by specifying the length of the required horizon at each node.

Lemma 1 *Each edge decides on its colour using only information from vertices which are at most* $2d\cos(\pi/k)$ *hops away.*

Proof. According to line 2 of the algorithm, no information is forwarded further than Euclidean distance d from its origin. This may take $2d - 1$ hops if forwarded in a straight line (alternating hops of length 1 and ϵ, where ϵ depends on k and can be chosen arbitrarily small). However the edges of class $\mathcal{C}_i$ can zig-zag only within a wedge of angular width π/k, thus possibly increasing the required distance by a factor of at most $\cos(\pi/k)$. This completes the proof of the lemma. □

Assuming that each edge is "waken up" by any activity within its 1-neighbourhood, Lemma 1 also implies that each edge decides on its colour after at most $4d\cos(\pi/k)$ steps since it may need to wake up the upstream path. The correctness of the algorithm follows by construction and from the definition of the VCG.

Theorem 1 *Algorithm 1 computes a* $2kl + 1$ *edge colouring of a* (l, k)*-edge/wedge subgraph of a UDG.*

Proof. We first prove that each edge can compute its colour. Let us call *start edges* the edges that have determined their colour in lines 4 or 6 of Algorithm 1. From the fact that

VCL_i^j is a d-cutting class $\mathcal{C}_i$ it follows that each edge is at distance d from a start edge and will therefore compute its colour in line 8 (if it did not do so in lines 4 or 6).

The definition of VCG (second property of each VCL_i^j and the fact that all MS_i^j s are pairwise disjoint) implies that no two edges of colour $2kl$ are incident to each other. From the construction (line 8) and the structure of (l,k)-edge/wedge graphs it follows that no two edges of the same colour are incident. □

2.3 Construction of Virtual Cutting Grids

It remains to be shown that given k and l, there indeed exists a $d-\mathsf{VCG}$ for (l,k)-edge/wedge graphs with d depending only on k and l. Without loss of generality, let us rotate all classes $\pi/4$ clockwise. This allows us to partition the classes into three categories as follows:

- *mostly horizontal:* $\mathcal{C}_i \bmod \pi \subseteq (-\pi/4, \pi/4\rangle$,
- *mostly vertical:* $\mathcal{C}_i \bmod \pi \subseteq (pi/4, 3\pi/4\rangle$, or
- *straddling the diagonal:* $\mathcal{C}_i \bmod \pi \subseteq (\pi/6, \pi/3)$

The basic idea is to use vertical VCLs to cut the *mostly horizontal* classes and horizontal VCLs to cut the *mostly vertical lines*. However, as VCLs must be separated, there must be gaps in the vertical cutting lines to allow placement of the horizontal cutting lines, and vice versa. The gaps are offset by an angle of $\pi/4$ from each other, but because of their width this still allows paths to leak from one gap to another. The square areas between the Minkowski sums of the mesh lines are used for placement of additional cutting lines to stop these leaks. The resulting VCG for $k = 2$ and $l = 1$ is depicted in Figure 2.

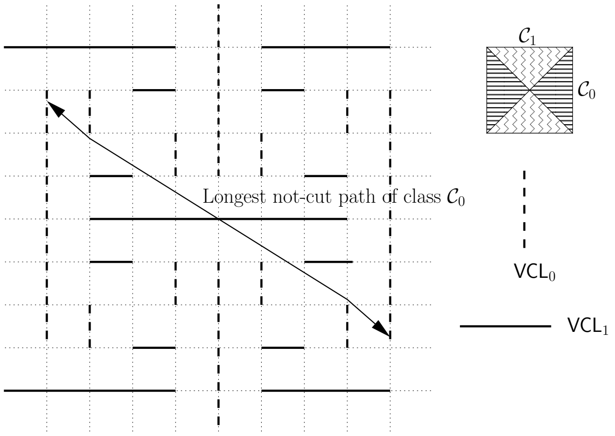

Figure 2: The design of VCG for $k = 2$, showing also the diameter of the longest not-cut path of class $\mathcal{C}_0$. The side of a basic square is 2.

Each leak-stopping square (we call them *leak-stoppers*) contains two vertical and two horizontal line segments, each of them blocking half of the gap width. The leak-stoppers are of two types (called *black* and *white*), placed in chessboard manner, with black leak-stoppers being mirror image of white ones.

This design can be straightforwardly generalized to $l > 1$ by enlarging the basic mesh by a factor of l and having VCL_i^j shifted $2j$ horizontally (for $i = 0$) or vertically (for $i = 1$). This idea is illustrated in Figure 3, refer also to the first line of Table 1).

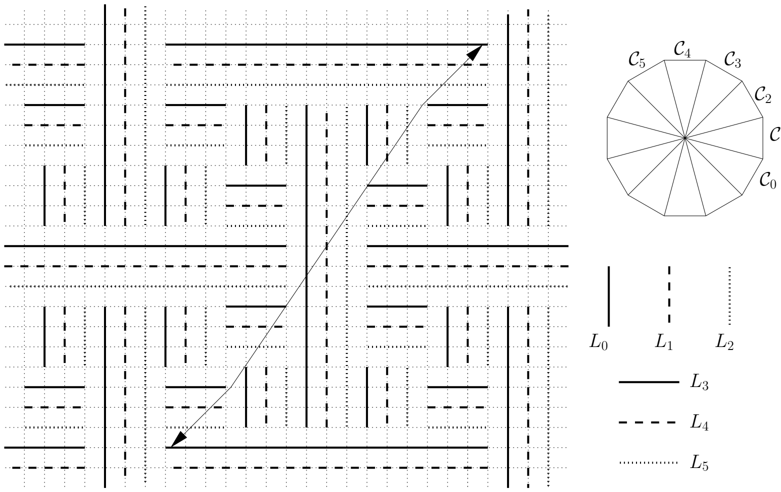

Figure 3: The design of VCG for $lk = 6$, showing also the longest non-cut path. The side of a basic square is 2.

Note that if k is even, the classes $\mathcal{C}_0, \mathcal{C}_1, \ldots, C_{k/2-1}$ are *mostly horizontal* and the remaining $k/2$ classes are *mostly vertical*. This means that we can reuse solution for $k' = 2$, $l' = lk/2$, i.e. the pattern from Figure 3 works for $k = 6, l = 1$ as well (refer to the second line of Table 1).

If k is odd, there will be exactly one class $\mathcal{C}_{\lfloor k/2 \rfloor}$ straddling the diagonal, with $\lfloor k/2 \rfloor$ classes being *mostly horizontal* and *mostly vertical*, respectively. For such class, neither vertical, nor horizontal VCL works. Consequently, in this case, we use the pattern for $k+1$ classes and let $\mathsf{VCL}^j_{\lfloor k/2 \rfloor}$ be a union of a horizontal and a vertical VCL– this also facilitates uniform construction with the same number of horizontal and vertical VCLs. For example, for $k = 5$ and $l = 1$, this result in the construction illustrated in the last row of Table 1. Summing up, this means that we can design a VCG for every l and k.

Theorem 2 *Every (l, k)-edge/wedge graph, for $k \geq 2$ and $l \geq 1$, has a d-VCG, for $d \geq \left(2\sqrt{61(l\lceil k/2 \rceil)^2 + 34(l\lceil k/2 \rceil) + 5}\right)$.*

Table 1: Possible assignments of VCLs for $lk = 6$ or $k = 5$, $l = 1$.

k	l	$L0$	$L1$	$L2$	$L3$	$L4$	$L5$
2	3	VCL_0^0	VCL_0^1	VCL_0^2	VCL_1^0	VCL_1^1	VCL_1^2
6	1	VCL_0^0	VCL_1^0	VCL_2^0	VCL_3^0	VCL_4^0	VCL_5^0
5	1	VCL_0^0	VCL_1^0	VCL_2^0	VCL_3^0	VCL_4^0	VCL_2^0

Proof. Firstly, we need to prove that the constructed set is indeed a VCG for l and k. For this purpose we have to show that (1) Minkowski sums are disjoint and that the VCLs satisfy the requirements of Definition 1, in particular that (2) no two consecutive edges are incident to a VCL unless they share an endpoint and (3) there is a finite (depending only on l and k) diameter d such each VCL_i^j d-cuts all paths of class $\mathcal{C}_i$ and rank j.

Proving (1): The MS_i^j's are pairwise disjoint because the closest distance between two points of different VCL_i^j and $\mathsf{VCL}_{i'}^{j'}$ is 2, i.e. two edges of length at most 1 connecting them must lie on the same line – and therefore must lie in the same class. Hence, one is an upstream edge and the other one is a downstream edge. However, by definition $\mathcal{C}_i$ does not contain upstream edges of length 1, i.e. the Minkowski sums do not overlap even when the VCLs are at distance exactly 2.

Proving (2): Note that for each i, no line segment of VCL_i has a slope inside $\mathcal{C}_i$. This is also true for the classes with bounding slopes being multiples of $\pi/4$ (the classes contain only the smaller of their boundary angles, by definition the *mostly horizontal/vertical* classes contain only the larger one). Together with (1) this means that the second property of Definition 1 is satisfied as well.

Proving (3): It is not difficult to observe that the diameter points shown in Figures 2 and 3 are indeed the furthest possible points of a path of class $\mathcal{C}_0$ which is not cut by its VCL. By symmetry, their distance is the diameter distance for other classes as well (the diameter of the *straddling the diagonal* class, if any, is even smaller). As the VCL line separation is 2, the diameter points are offset by $2(5l\lceil k/2\rceil + 1)$ horizontally and $2(6l\lceil k/2\rceil + 2)$ vertically. The statement of the theorem is obtained using the Pythagorean theorem. □

3 Lower Bounds

In this section we look at lower bounds. First, at the impossibility of local construction of edge colourings if the nodes are not location aware and second at lower bounds for the horizon distance d.

3.1 Impossibility of edge coloring when no geometric information is available

In this subsection we show that having the geometric information is indeed crucial in achieving a local algorithm. If the nodes only have distinct IDs without additional information, no non-trivial edge colouring can be locally constructed. Erdös [1962] addressed this question by using the probabilistic method to show that for all k and $\epsilon > 0$ there exist graphs with sufficiently large number n of nodes that have chromatic number $> k$ but such that every set of vertices of size at most ϵn is 3 colourable.

In the present paper we adapt the proof technique of Naor and Stockmeyer [1995]. The main difference is that since we are dealing with UDGs, our concept of locality is somewhat different. Instead of requiring limited hop-count in the actual graph, we limit the hop-count to the underlying UDG. The principal result of Naor and Stockmeyer [1995] we use is based on Ramsey's theorem and states that we can limit ourselves to algorithms that do not depend on the actual values of the IDs, but only on their relative ordering.

Theorem 3 (Naor and Stockmeyer [1995]) *Fix a class of graphs $\mathcal{G}$ and a* locally checkable labeling problem $\mathcal{L}$. *If there is a local algorithm A with time bound t that solves a problem $\mathcal{L}$ in $\mathcal{G}$, then there is an order-invariant local algorithm A' that solves $\mathcal{L}$ in $\mathcal{G}$ in time t.*

Informally, a problem has a *locally checkable labeling* if it can be locally verified whether a given vertex (or edge) labeling satisfies the requirements. Refer to Naor and Stockmeyer [1995] for the formal statement. For our purposes it is sufficient to state that vertex colouring and edge colouring are both locally checkable labellings.

Theorem 4 *For every $l \geq 1$, $k \geq 2$ there is a (l,k)-edge/wedge subgraph of a UDG that is not locally colourable.*

Proof. Consider a mesh of vertices with the distance w between neighboring mesh vertices equal to $w = \sqrt{2}l\tan(\pi/k)$. Define the graph G in which each vertex v of the mesh is connected to the closest l vertices in each wedge. The choice of w ensures that all these edges are of length at most 1, i.e. G is a subgraph of the UDG of these mesh vertices. Since neighboring vertices of the mesh are neighbors in G, for each edge e of the UDG there exists a path in G of length at most $x = \sqrt{2}/w$ connecting the endpoints of e.

Assume now, by contradiction, there exists a local (in the UDG) algorithm A, examining the neighbourhood up to distance t. Therefore, there must exist a constant x and a local (in G) algorithm A' examining the neighbourhood up to distance xt. By Theorem 3 there is also an order invariant algorithm A''.

Consider now a mesh of size $(2xt + 3) \times (2xt + 3)$, with the vertex at location $[i, j]$ labeled $j(2xt+3)+i$. As A' examines the neighbourhood up to distance at most xt, nodes $v_1 = [xt, xt]$, $v_2 = [xt + 1, xt]$ and $v_3 = [xt + 2, xt]$ all see the same pattern (monotonically increasing from upper left to lower right) and cannot deterministically decide on different colours for edges (v_1, v_2) and (v_2, v_3). This contradicts the fact that A' is a local edge-colouring algorithm. □

3.2 Lower bound on d

Theorem 2 gives a bound for d of about $7.81 \cdot lk + O(1)$. As the construction seems quite ad hoc and not optimized (it gives the same bound for fixed lk, despite the fact that the case $l = 1$ and large k is much more restrictive than $k = 2$ and large l), an obvious question is how for from the optimal it actually is. A somewhat surprising answer is that it is not far at all, namely we have the following result.

Theorem 5 *There is no $d-\mathsf{VCG}$ for (l,k)-edge/wedge graphs with $d < 2kl$.*

Proof. The idea is to show that each MS_i^j covers a $1/2d$ fraction of the area. Since all these Minkowski sums must be pairwise disjoint, this would prove the theorem. Consider VCL_i^j. Take a line l whose slope is the axis of the class $\mathcal{C}_i$. Note that l is cut by VCL_i^j into line segments of length at most d. The parts of l at distance less then 1 both downstream and upstream from the cutpoints belong to MS_i^j, i.e. a $2/d$ fraction of l is in MS_i^j. As this is true for every position of l in the plane, (and the behaviour is continuous, as VCLs are sets of line segments) MS_i^j indeed takes $1/2d$ fraction of the area and the statement of the theorem follows. □

4 Refined Constructions

As noted above, the construction used in Theorem 2 is based on the case $k = 2$, with the angular width of the wedges being $\pi/2$. This means that the paths have a lot of 'wiggle room' to avoid the VCL s, forcing a conservative construction with relatively large horizon distance d. The horizon distance can be noticeably reduced by more careful analysis for specific k. In particular, the case $k = 3$ warrants special attention, as this is the most relevant case in practical situations (both LMST and Half-space proximal lead to $(1, 3)$-edge/wedge graphs and $k = 3$ is the lowest k for which the Yao graph is connected). An ad-hoc construction of the VCG using hexagonal symmetry and careful analysis of Minkowski sums yields:

Theorem 6 *There exists* 15.466-*VCG for* $(1, 3)$-*edge/wedge graphs.*

Due to space constraints, the proofs of results stated in this section have been moved to the appendix. The case $k = 4$ is practically important as well, as it corresponds to the smallest k for which the Yao graph is connected and has a convenient east/north/west/south symmetry.

Theorem 7 *There exists* 26.989-*VCG for* $(1, 4)$-*edge/wedge graphs.*

Finally, for the limit case of $k \to \infty$ the wedges are very narrow, which allows the use of several additional techniques to further reduce the horizon distance:

Theorem 8 *There exists* $3.3745lk + o(lk)$-*VCG for* (l, k)-*edge/wedge graphs.*

5 Conclusions

In this paper we identified a new subclass of unit disk graphs, called (l, k)-edge/wedge graphs, that encompass a variety of well-known and commonly used graphs in the wireless networks literature (Yao graphs, Local Delaunay, Local Minimum Spanning Tree spanners, and Half-Space Proximals). We gave local, deterministic algorithms for edge coloring these graphs and provided several ways for improving the algorithm depending on the density of the point set that determines the graph. Several questions concerning the optimality of our algorithm remain open and would be interesting to consider in future investigations.

References

T. C. Barrett, G. Istrate, A. Kumar, M. Marathe, S. Thite, and S. Thulasidasan. Strong edge coloring for channel assignment in wireless radio networks. In *IEEE International Workshop on Foundation and Algorithms for Wireless Networking*. IEEE, 2006.

E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho, H. Tejeda, and J. Urrutia. Half-space proximal: A new local test for extracting a bounded dilation spanner. In *Proceedings of OPODIS, 2005*, Lecture Notes in Computer Science. Springer Verlag, 2006a.

E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho, and J. Urrutia. Local construction of planar spanners in unit disk graphs with irregular transmission ranges. In M. K. j. Correa, A. Hevia, editor, *Latin 06*, volume 3387 of *Lecture Notes in Computer Science*, pages 286–297. Springer Verlag, 2006b.

P. Erdös. On circuits and subgraphs of chromatic graphs. *Mathematika*, 9:170–175, 1962.

S. Gandham, M. Dawande, and R. Prakash. Link scheduling in sensor networks: distributed edge coloring revisited. In *INFOCOM*, pages 2492–2501. IEEE, 2005.

I. Holyer. The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10(4): 718–720, 1981.

M. S. Kodialam and T. Nandagopal. Characterizing achievable rates in multi-hop wireless mesh networks with orthogonal channels. *IEEE/ACM Trans. Netw.*, 13(4):868–880, 2005.

V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. End-to-end packet-scheduling in wireless ad-hoc networks. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, pages 1021–1030. SIAM, 2004.

X.-Y. Li, G. Calinescu, P.-J. Wan, and Y. Wang. Localized delaunay triangulation with application in ad hoc wireless networks. *IEEE Trans. Parallel Distrib. Syst*, 14(10):1035–1047, 2003.

N. Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1): 193–201, Feb. 1992.

M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25(1):59–68, 1995.

T. Matsui. Approximation algorithms for maximum independent set problems and fractional coloring problems on unit disk graphs. In J. Akiyama, M. Kano, and M. Urabe, editors, *JCDCG*, volume 1763 of *Lecture Notes in Computer Science*, pages 194–200. Springer Verlag, 1998.

M. Naor and L. Stockmeyer. What can be computed locally? *SICOMP: SIAM Journal on Computing*, 24, 1995.

D. Peleg. *Distributed Computing: A Locality-Sensitive Approach.* SIAM, 2000.

S. Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 5(2):81–94, 1999.

V. G. Vizing. On the estimate of the chromatic class of a p-graph. *Diskret. Analiz.*, 3:25–30, 1964.

Y. Wang and X.-Y. Li. Localized construction of bounded degree and planar spanner for wireless ad hoc networks. In *DialM: Proceedings of the Discrete Algorithms and Methods for Mobile Computing & Communications*, 2003.

A. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM Journal on Computing*, 11(4):721–736, 1982.

APPENDIX

This appendix contains proofs of the theorems stated in Section 4, as well as related discussion.

Asymptotic analysis for large k

Theorem 8 *There exists* $3.3745lk + o(lk)$*-*VCG *for* (l,k)*-edge/wedge graphs.*

Proof. The structure of the solution from Figure 3 can be interpreted as a main mesh of interrupted *super-lines* (each super-line containing $lk/2$ VCLs, the combined width of their Minkowski sums being lk) and *leak-stoppers* (i.e., square areas containing additional segments of VCLs with the purpose to cut the paths that 'leaked' through the openings in the main mesh). Note that the leak-stoppers are of two types (let us call them *black* and *white*), distributed in chess-board style and being mirror image of each other.

As we are interested in d as a function of lk for large k, we employ some simplifications which do not alter the asymptotic result but allow us to avoid tedious case analysis and notation. In particular, we represent the class $\mathcal{C}_i$ by the corresponding angle $i/k\pi - \pi/4$ and ignore terms that go to 0 as k approaches infinity.

The first observation is that the separation between the VCLs need not be 2, as the Minkowski sum of VCL_α has horizontal width $2\cos(\alpha)$ for $\alpha \in \langle -\pi/4, \pi/4\rangle$. Summing together for all mostly-vertical classes and approximating the sum with the integral[*] we get that the total width of the horizontal lines in the main mesh can be expressed as

$$\frac{2kl}{\pi}\int_{-\pi/4}^{\pi/4}\cos(x)dx = \frac{2kl}{\pi}2(\sin(\pi/4) - \sin(0)) = 2\sqrt{2}kl/\pi \approx 0.9003 \cdot kl.$$

The same observation applies to the vertical VCLs as well.

Second, note that among the mostly horizontal classes, half are *descending* (i.e. $\mathcal{C}_i \in \langle -\pi/4,\ 0)$ and the other half are *ascending*[†] (and similar classification can be applied to the mostly vertical classes). That means that the number of auxiliary cutting lines used in leak-stoppers to cut the paths that leaked through the mesh openings can be halved. The white leak stoppers have to only deal with *descending* horizontal classes and *ascending* vertical classes, while the black leak-stoppers handle the *ascending* horizontal classes and *descending* vertical classes. We will show how to design the white leak stoppers, since the design for black ones is symmetric.

Actually, the situation is even better. Let the size of leak stoppers be expressed as zkl for some z and let the width of mesh superlines be akl (we already know that $a = \sqrt{2}/\pi$). (Let us also, from this moment on, use normalized distances with respect to kl, e.g., instead of zkl we will use z.) As the gaps in the mesh are offset by $z+a$ horizontally and z vertically, no line of class C_α for $\alpha \leq \arctan(z/(a+z)) = \phi$ can leak from one gap to another. The combined width of Minkowski sums of these classes can then be expressed as (similarly as in the first observation, but now only for the descending classes) $\frac{2}{\pi}(\sin(\pi/4) - \sin(\phi))$.

[*] This is one of the simplifications mentioned above that does not alter the asymptotic result for large k.

[†] This is another simplification which is insignificant for large k. Since there might be two classes straddling the horizontal and vertical axis, we treat them similarly as before by handling them on both sides.

In fact, we can do even more refined analysis and divide descending mostly-horizontal leak-prone classes into three groups: $A = \mathcal{C}_\alpha$ for $\alpha \in (-\pi/4, -\psi_1\rangle$, $B = \mathcal{C}_\beta$ for $\beta \in (-\psi_1, -\psi_2\rangle$ and $C = \mathcal{C}_\gamma$ for $\gamma \in (-\psi_2, \phi\rangle$ and analogously D, E and F for ascending mostly vertical classes, where the boundary angles ψ_1 and ψ_2 are chosen so that the width of the Minkowski sums of these classes are equal: $\sin(\pi/4) - \sin(\psi_1) = \sin(\psi_1) - \sin(\psi_2) = \sin(\psi_2) - \sin(\phi)$. Let us denote this width by $w = \frac{2}{3\pi}(\sin(\pi/4) - \sin(\phi))$. The basic idea is that the classes in B and C (E and F) need progressively less blocking as they are more horizontal (vertical) than classes in A and D.

Let us choose $z \approx 0.5807$, this results in $\phi \approx 0.3737$, $\psi_1 \approx 0.6347$, $\psi_2 \approx 0.4997$ and $w = z/8$. This allows us to conveniently design the leak stopper square as a 8×8 square of basic tiles of size w, with each tile containing a segment stopping the paths of one class (see Figure 5).

Consider the group of classes B (the other groups are analogous). No gap-crossing path π_B of this group can descend below the line of angle $-\psi_1$ passing through the lower lip of the gap, and no such path can come from above the line of the same angle passing through the top end of the gap. These two lines define a swath that has to be blocked by segments of type B in the leak-stopper (see Figure 4 for the swath lines of all groups).

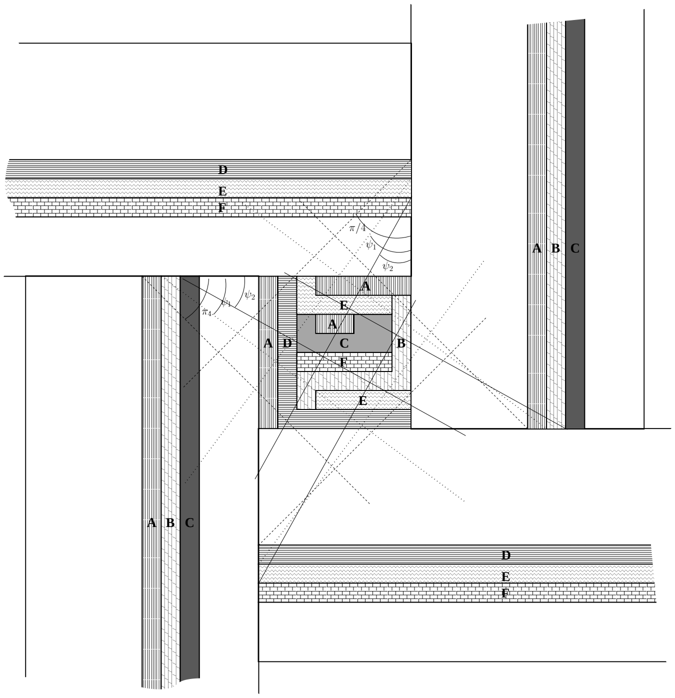

Figure 4: The swath lines all together. The white areas in the mesh superlines correspond to classes that are stopped by black leak-stoppers, and to classes corresponding to angles less then ϕ from vertical/horizontal – the ones that cannot pass from one gap to another.

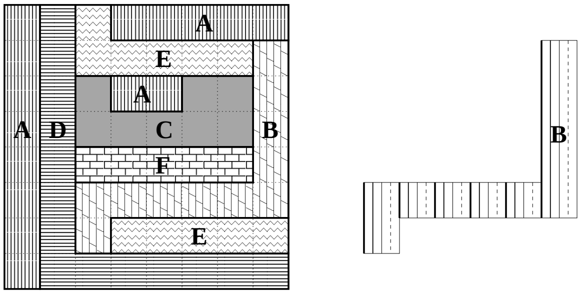

Figure 5: Left: The design of a white leak-stopper. Right: Detailed look at the segment blocking classes of group B.

The ordering of the classes within the mesh superline does matter, as it influences the offsets of the swaths – in the middle of the superline corresponds to a centered swath. As the swaths are widest for the steepest (closest to diagonal) groups, putting those groups to the middle of the mesh superline is the best choice. The left side of a vertical superline starts with $\mathcal{C}_0$ and the class angles increase until $\mathcal{C}_{\pi/4}$ is reached in the middle, then the classes switch to $\mathcal{C}_{-\pi/4}$ and increase to $\mathcal{C}_0$ at the right side of the superline. The ordering is analogous for the horizontal superlines, with the bottom part reserved for the increasing angles and the top part for decreasing angles.

Figure 6 shows the swaths and the resulting designs for all 6 groups of classes in white leak-stoppers.

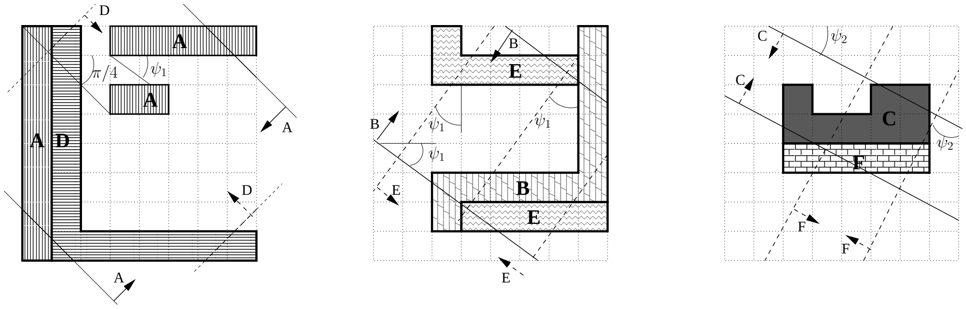

Figure 6: The swaths for different groups of classes, and the corresponding blocking patterns.

The final optimization comes from observing that for large k the path can only go in almost a straight line, i.e. the largest d cannot be in a path that makes 'bends' (as in Figure 3). In order to find the diameter path for a given groups of classes, we examine what is the steepest angle that misses the leak-stopper segments of that type. This can be straightforwardly calculated for all six classes as shown in Table 2:

The steepest paths for groups A, B, D and E are not steep enough to belong to their group (trying to achieve that for B and E lead to some of the peculiarities of the design from Figure 5), meaning that no path from these groups can cross all the way from a mesh superline through a gap to the other mesh superline. The steepest paths for groups C and F are steeper the ψ_2, i.e. the diameter path for these groups is limited by the maximal angle

Table 2: The diameter paths for each group of classes. The second and third column correspond to the locations in the leak-stoppers touched by the extreme line, expressed as multiples of w. The angle is calculated as $\arctan((a+z+r_y w - l_y)/(a+z+r_x w - l_x w))$ for groups A, B and C, for the other groups the x and y axis are swapped.

Group	$[l_x, l_y]$	$[r_x, r_y]$	Angle	Note
A	$[0,8]$	$[7,0]$	0.4250	$< \psi_1$, stops prematurely
B	$[2,7]$	$[7,0]$	0.4854	$< \psi_2$, stops prematurely
C	$[2,4]$	$[6,2]$	0.6461	$> \psi_2$, ψ_2 is steepest
D	$[1,0]$	$[8,7]$	0.4548	$< \psi_1$, stops prematurely
E	$[2,0]$	$[8,6]$	0.4993	$< \psi_2$, stops prematurely
F	$[2,4]$	$[7,4]$	0.6466	$> \psi_2$, ψ_2 is steepest

ψ_2, not by the leak-stoppers. The resulting diameter d is therefore $d = 2(a+z)/\cos(\psi_2) \approx 3.3745$. □

Further improvement is possible by more detailed analysis, i.e. by dividing the classes between ϕ and $\pi/4$ into 4 or more groups, however the possible improvements are only minor: z must be at least $a/2$ (otherwise the whole leak-stopper does not cut the swath for the group containing the diagonal class). A diameter path bypassing both upper left and lower right leak-stopper is therefore of angle $\theta = \arctan(a/(a+2z)) = \arctan(1/2)$ and length $2(a+z)/cos(\theta) \approx 3.01975$, i.e. our result is within 12% of the best possible using the mesh-with-gaps approach.

Improvements for $k = 3$ and $k = 4$

The case $k = 3$ warrants special attention, as this is the most relevant case in practical situations (Both LMST and Half-space proximal lead to $(1,3)$-edge/wedge graphs and $k = 3$ is the lowest k for which the Yao graph is connected). The case $k = 4$ is practically important as it corresponds to the smallest k for which the Yao graph is connected and has a convenient east/north/west/south symmetry. In this part we employ additional techniques (hexagonal symmetry, use of asymmetry) to improve upon Theorem 2.

Theorem 6 *There exists* 15.466-*VCG for* $(1,3)$-*edge/wedge graphs.*

Proof. The detailed design of VCL_i for $i = 0, 1, 2$ is shown in Figure 7, the hexagon is of side(radius) 6. Minkowski sums for parts of VCL_0 and VCL_1 are shown as shaded areas to demonstrate that they are indeed disjoint. Note that no line segment of VCL_i is inside $\mathcal{C}_i$, ensuring (together with disjointness of the Minkowski sums) the second property of VCLs.

Figure 8 shows how are the per-hexagon VCLs combined to form the global VCG. The location is fixed by having one of the hexagons centered at $[0,0]$.

It is straightforward to verify that indeed each VCL_i is d-cutting its $\mathcal{C}_i$, where the longest non-cut path (of length d) is marked on Figure 8 for the class $\mathcal{C}_0$. Using the Pythagorean theorem we derive $d = \sqrt{9^2 + (12 + 1/\sqrt{3})^2} = \sqrt{225 + 8\sqrt{3} + 1/3} \approx 15.46576$. □

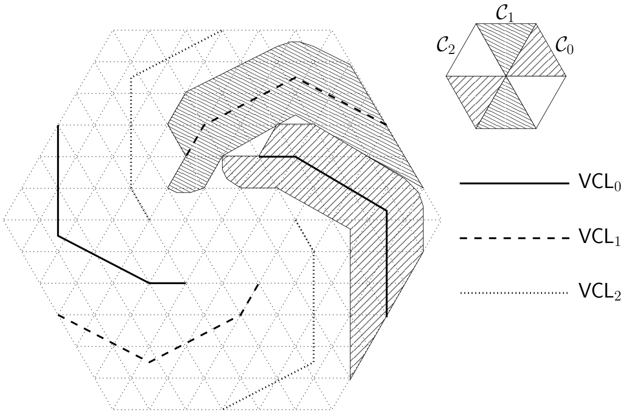

Figure 7: The design of VCLs for $k = 3$

Note that Theorem 2 yields $d \approx 35.609$ for $k = 3$.

The construction can be straightforwardly generalized to larger l and $k' = 3k$: Enlarge the scale by a factor of lk and put lk equally-spaced cutting lines for class $\mathcal{C}_i$ in the space previously occupied by the single cutting line for that class (see Figure 9).

Theorem 9 *There exists* $15.466lk$*-*VCG *for* $(l, 3k)$*-edge/wedge graphs.*

Note that for large k Theorem 8 achieves a better bound.

While only few of the techniques introduced in Section 5 are applicable for the case $k = 4$, a careful ad-hoc construction making use of asymmetry allows us to design an efficient VCG:

Theorem 7 *There exists* 26.989*-*VCG *for* $(1, 4)$*-edge/wedge graphs.*

Proof. The design is based on a careful design of leak-stoppers, shown in Figure 11, allowing reduction of their size from 10×10 to 6×6. Minor benefit is also obtained by having the top of rightmost blocking line for VCL_0 extend to $[5, 1/sqrt1]$ instead of ending at $[5, 1]$ (for other classes the benefit is higher and resulting d smaller).

The resulting diameter path has therefore length

$$\sqrt{20^2 + (16 - \sqrt{2} - 1/\sqrt{2})^2} = \sqrt{660.5 + 48\sqrt{2}} \approx 26.989.$$

One minor problem is that the VCLs as shown in Figure 11 contain segments that might belong to the corresponding class (i.e. VCL_1 contains horizontal segments and VCL_3 contains vertical ones). This can be remedied by slightly tilting these segments – there is space for that as can be seen examining the Minkowski sums. Still, this prevents adaptation of this approach to the case $l > 1$. □

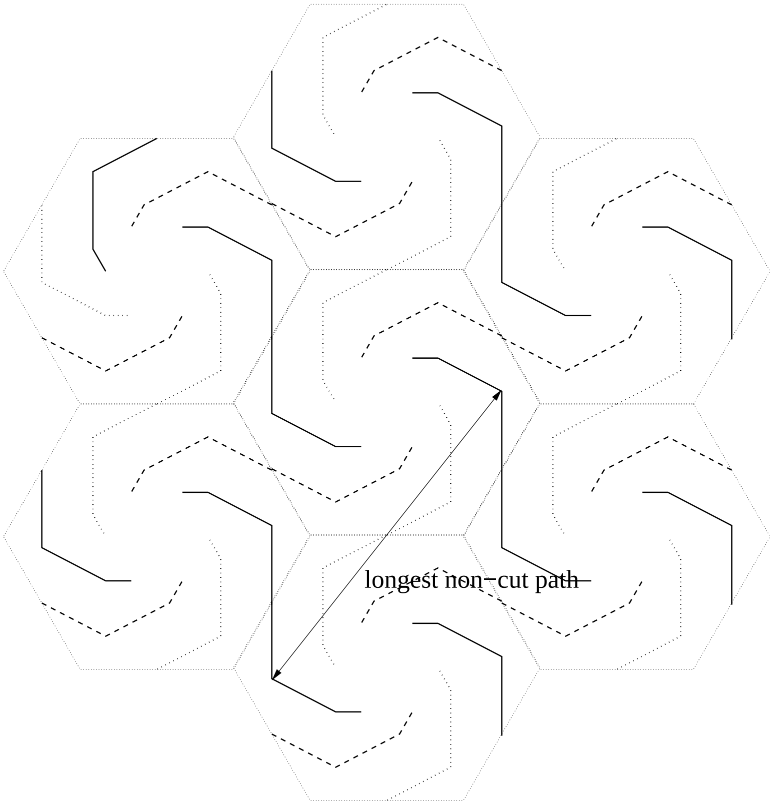

Figure 8: VCG for $k = 3$, also showing the longest non-cut path of class $\mathcal{C}_0$.

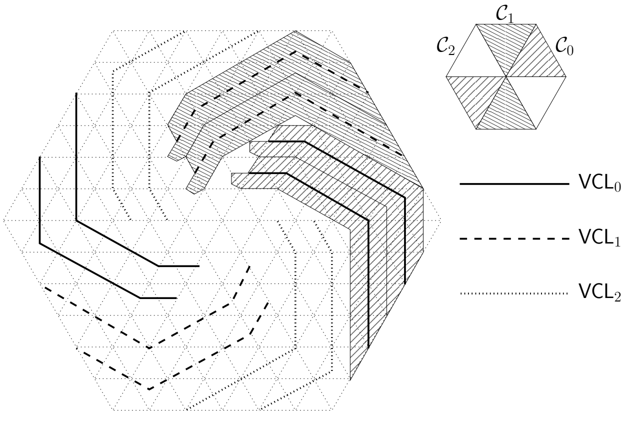

Figure 9: The design of VCLs for $l = 2$, $k' = 3$.

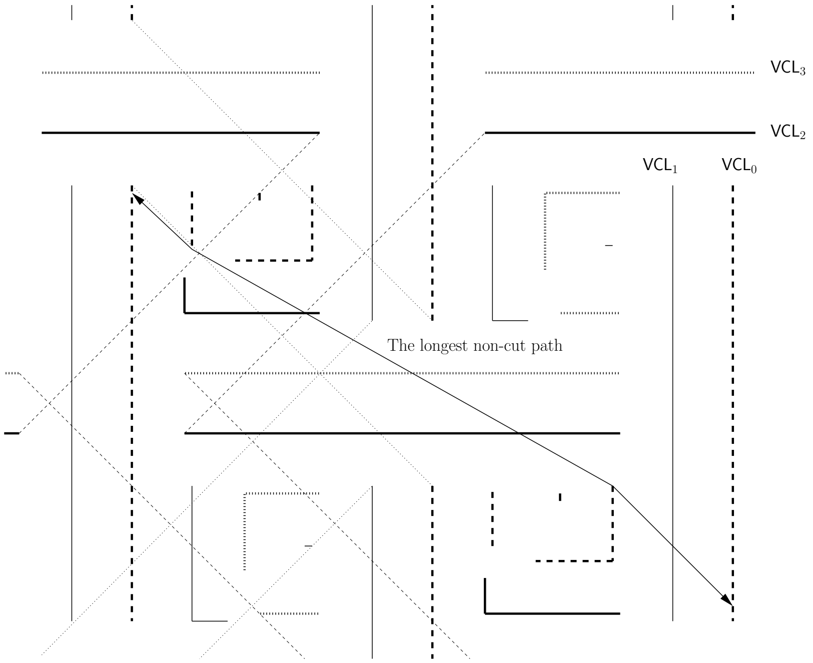

Figure 10: The design of VCLs for $k = 4$

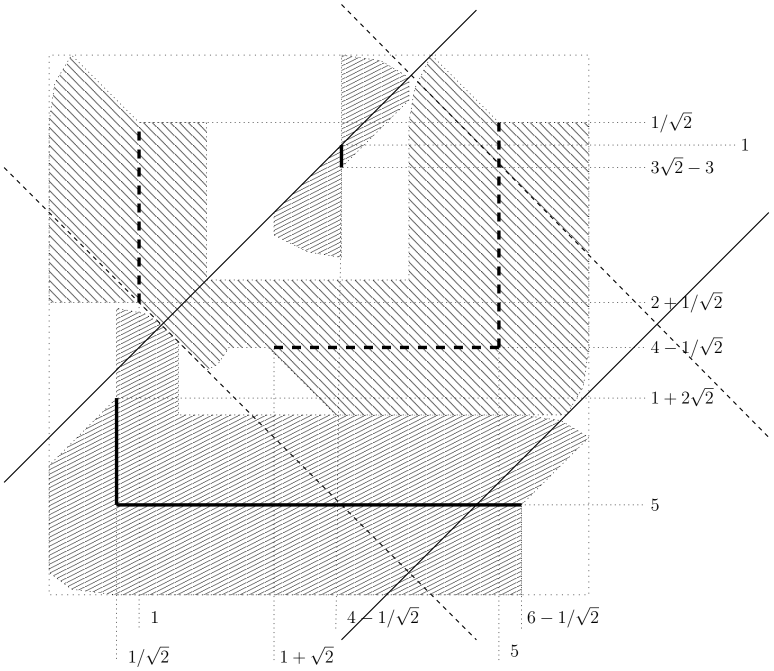

Figure 11: Detailed design of a leak-stopper for $k = 4$, showing also Minkowski sums.