HAL
archives-ouvertes.fr

Circuits in graphs through a prescribed set of ordered vertices

David Coudert, Frédéric Giroire, Ignasi Sau

To cite this version:

David Coudert, Frédéric Giroire, Ignasi Sau. Circuits in graphs through a prescribed set of ordered vertices. Journal of Interconnection Networks, World Scientific Publishing, 2011, 11 (3-4), pp.121-141. 10.1142/S0219265910002763 . inria-00585561

HAL Id: inria-00585561

https://hal.inria.fr/inria-00585561

Submitted on 13 Apr 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Circuits in graphs through a prescribed set of ordered vertices*

David Coudert, Frédéric Giroire
MASCOTTE, INRIA, I3S, CNRS, Univ. Nice Sophia Antipolis, France
{firstname.lastname}@sophia.inria.fr

Ignasi Sau
AlGCo team, CNRS, LIRMM, Montpellier, France
Ignasi.Sau@lirmm.fr

Abstract

A *circuit* in a simple undirected graph $G = (V, E)$ is a sequence of vertices $\{v_1, v_2, \ldots, v_{k+1}\}$ such that $v_1 = v_{k+1}$ and $\{v_i, v_{i+1}\} \in E$ for $i = 1, \ldots, k$. A circuit C is said to be *edge-simple* if no edge of G is used twice in C. In this article we study the following problem: which is the largest integer k such that, given any subset of k ordered vertices of a graph G, there exists an edge-simple circuit visiting the k vertices in the prescribed order? We first study the case when G has maximum degree at most 3, establishing the value of k for several subcases, such as when G is planar or 3-vertex-connected. Our main result is that $k = 10$ in infinite square grids. To prove this, we introduce a methodology based on the notion of core graph, in order to reduce the number of possible vertex configurations, and then we test each one of the resulting configurations with an Integer Linear Program (ILP) solver.

Keywords: Simple circuit, prescribed order, square grid, planar graph, connectivity.

*A preliminary version containing some of the results of this work appeared in "D. Coudert, F. Giroire, and I. Sau. *Edge-Simple Circuits Through 10 Ordered Vertices in Square Grids.* Proceedings of the 20th International Workshop on Combinatorial Algorithms (IWOCA), volume 5874 of LNCS, pages 134-145, Opava, Czech Republic, June 2009". This research has been done in the context of ANR AGAPE and DIMAGREEN.

1 Introduction

A *circuit* in a simple undirected graph $G = (V, E)$ is a sequence of vertices $\{v_1, v_2, \ldots, v_{k+1}\}$ such that $v_1 = v_{k+1}$ and $\{v_i, v_{i+i}\} \in E$ for $i = 1, \ldots, k$. A circuit C is said to be *edge-simple* if no edge of G is used twice in C. The existence of a circuit through a prescribed set of vertices or edges has been an important graph-theoretical question for many years [2–4,6–8,10–12,15,16,19]. Typically, high connectivity is a powerful sufficient condition for the existence of such circuits. For instance, it is well known that in a k-vertex-connected graph any subset of k vertices [6] or any subset of $k-1$ independent edges [12] is included in a cycle. A circuit C is a *cycle* if no vertex of G is used twice in C, except for $v_1 = v_{k+1}$.

However, knowing specific properties of the graph often permits to prove much stronger results. In this article we focus, on the one hand, on the existence of edge-simple circuits through specified vertices in the infinite square grid (or equivalently, a large enough toroidal grid), which is a widely studied 4-vertex-connected graph. On the other hand, we also study (planar and general) graphs of maximum degree at most 3. In addition, we do not only require the circuit to visit a subset of vertices, but also to visit them in a *prescribed order*. More precisely, we are interested in the following type of circuits:

Definition 1.1 (good circuit)**.** *Given a set S of k distinct ordered vertices $\{v_1, v_2, \ldots, v_k\}$ in a graph G, an edge-simple circuit C in G is S-*good *if the vertices $v_1, v_2, \ldots, v_k$ appear in C in this order (modulo cyclic permutations).*

If the ordered subset of vertices S is clear from the context, we may refer to an S-good circuit simply as *good*. It is easy to check that, for instance, such a good circuit in a square grid always exists for any ordered subset of at most 4 vertices, and one can also verify (still within a reasonable amount of time) that the same holds for any subset of 5 vertices. On the other hand, it seems intuitive to suspect that this property will not be true for an arbitrary large subset of ordered vertices of a square grid. Therefore, the following natural question arises:

Problem (Longest Good Circuit (LGC for short))**.** *Given a graph G, which is the largest integer k such that for any subset S of k distinct ordered vertices in G, there exists an S-good circuit?*

Our results and organization of the paper. We first provide some background and motivations for our problem in Section 2. In Section 3 we use the notion of edge expansion to provide an upper bound for LGC in general graphs. In Section 4 we investigate the question in graphs of maximum degree at most 3 and graphs with vertices of small degree. In particular, we prove in Subsection 4.1 that $\mathrm{LGC}(G) = 3$ for G being an infinite hexagonal grid, and in Subsection 4.2 that for any 3-vertex-connected planar cubic graph G different from K_4, $\mathrm{LGC}(G) = 3$. This bound on the connectivity is tight, namely we show that there exist 2-vertex-connected planar cubic graphs G for which $\mathrm{LGC}(G) = 2$. In Section 5 we focus on infinite square grids. Namely, we show in Subsection 5.1 that checking the feasibility of a configuration in the grid is equivalent to checking its feasibility in an auxiliary graph, called *internal graph*. In Subsection 5.2 we introduce the notion of *core graphs* in order to reduce the number of internal graphs to be tested. In Subsection 5.3 we give a counterexample establishing the upper bound of 10 in square grids. In Subsection 5.4 we match this upper bound with the help of an Integer Linear Program (ILP) solver to exhaustively test all the orderings for a small list of possible configurations that we obtained after applying the reductions of Subsections 5.1 and 5.2. Finally, Section 6 concludes the article and presents some lines for further research.

Notation. We use standard graph terminology (c.f. for instance [5].) All the graphs considered in this article are undirected. Given a graph G we denote by $V(G)$ and $E(G)$ the sets of

vertices and edges of G, respectively. If H is a subgraph of G, we denote it by $H \subseteq G$. Given a graph G and $S \subseteq V(G)$, we denote by $G[S]$ the subgraph of G containing all the edges with both endpoints in S. Given a vertex $v \in V(G)$, we define $N_G[v]$ to be the set of vertices of $V(G)$ at distance at most 1 from v, and $N_G(v) = N_G[v] \setminus \{v\}$. The *degree* of a vertex $v \in V(G)$ is defined as $\deg_G(v) = |N_G(v)|$. A graph is *cubic* if all its vertices have degree 3. The *maximum degree* of G is defined as $\Delta(G) = \max_{v \in V(G)} \deg_G(v)$. Given an edge $e = \{u, v\}$, we define the operation of *subdividing* e as deleting the edge $e = \{u, v\}$, adding a new vertex w, and making it adjacent to both u and v. Conversely, given a vertex w of degree 2 belonging to edges $\{u, w\}$ and $\{w, v\}$, we define the operation of *dissolving* vertex w as replacing edges $\{u, w\}$ and $\{w, v\}$ with a new edge $\{u, v\}$.

2 Background and Motivation

Connectivity is one of the cornerstone concepts of graph theory. Maybe the most archetypal results are Menger's classical theorems [5], which say that a graph is k-vertex-connected (resp. k-edge-connected) if and only if it contains k vertex-disjoint (resp. edge-disjoint) paths between any two vertices. There is a huge literature concerning extremal problems of cycles in k-connected graphs. See for instance the following theorems that we recall here for the sake of completeness.

Theorem 2.1 (Dirac [6])**.** *In a k-vertex-connected graph any subset of k vertices is included in a cycle.*

Theorem 2.2 (Häggkvist and Thomassen [12])**.** *In a k-vertex-connected graph any subset of $k-1$ independent edges is included in a cycle.*

There are a number of works giving necessary or sufficient conditions for the existence of a cycle through a specified set of vertices in a general graph [4, 7, 15, 16].

Some stronger results have been given for specific classes of graphs, like 3-connected cubic graphs [7,8]. For this class of graphs it is known that there exists a cycle through any 9 vertices, and that there exists a cycle which passes through any 10 given vertices if and only if the graph is not contractible to the Petersen graph, in such a way that each of the 10 vertices maps to a distinct vertex of the Petersen graph [8]. If, in addition, the 3-connected cubic graph is planar, then there exists a cycle through any 23 vertices [2]. Another example can be found in [10], where the authors provide necessary and sufficient conditions for a given graph embedded on the torus to contain edge-disjoint cycles satisfying prescribed topological properties.

However, all these results do not take into account the *order* in which the cycle visits the prescribed set of vertices. This is a natural constraint, since for example in telecommunication networks it may be important to connect a subset of vertices in such a way that each vertex numbered i has capability to communicate only with the two vertices numbered $i-1$ and $i+1$ (modulo the cardinality of the subset of vertices). This could be the case, for instance, of the classical *token ring* networks defined by the standard *IEEE* 802.5. That is, there exists a whole class of problems to consider when introducing the constraint on the order of the vertices to be visited. In this article we study one of these problems in infinite square grids (or large enough toroidal grids) as well as in graphs maximum degree at most 3 and in graphs with vertices of small degree.

Let us now make a connection to the disjoint paths problem. Observe that, in a general (di)graph, the problem of deciding whether there exist edge-disjoint paths between given pairs of vertices is NP-complete [14] (even if the graph is a square grid [17]). When the number of pairs of vertices is bounded by a constant, the disjoint paths problem is polynomial-time

solvable in undirected graphs [20], NP-complete in directed graphs [18] (even with only two pairs of vertices [9]), and polynomial-time solvable in symmetric directed graphs [13]. In fact, the existence of a good circuit C for a set of k ordered vertices $\{v_1, v_2, \ldots, v_k\}$ corresponds to the existence of k edge-disjoint paths in the graph G between the pairs of vertices (or *commodities*) $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{k-1}, v_k\}, \{v_k, v_1\}$. Thus, for a particular placement of $v_1, v_2, \ldots, v_k$ in G, the existence of a good circuit can be checked by solving the associated integer multicommodity flow problem.

When designing a telecommunication network, the fault tolerance is a crucial issue. Observe that the simplest network which is able to support any single link failure is an edge-simple circuit, and that is one of the main reasons why the study of such circuits is important. The study of the square grid is also natural, due among other reasons to its extensive use in parallel computing. In this context, it is interesting to know which is the largest integer k for which there always exists a circuit visiting any ordered subset of at most k vertices. Observe also that without taking into account the ordering, there exists a cycle (and thus, a circuit) visiting any subset of vertices of the square grid, since a toroidal grid is a Hamiltonian graph.

It is worth mentioning that the square grid is in some sense the common *skeleton* of planar graphs. Indeed, it is well-known that every planar graph of branchwidth at least ℓ contains an $(\lfloor \ell/4 \rfloor \times \lfloor \ell/4 \rfloor)$-grid as a minor [21]. Therefore, a square grid is *inside* every planar graph, and any edge-simple circuit in a minor of a graph can be easily transformed to an edge-simple circuit in the graph itself.

3 General Upper Bound

In this section we use the notion of edge expansion to provide an upper bound for LGC in general graphs, and then we apply this technique to the toroidal grid.

Definition 3.1 (edge expansion)**.** *The* edge expansion *of a graph $G = (V, E)$ is defined as*

$$h(G) = \min \left\{ \frac{|\partial(S)|}{|S|} : \ S \subseteq V, \ 1 \leq |S| \leq \frac{|V|}{2} \right\} ,$$

where $\partial(S)$ stands for the set of edges with exactly one endpoint in S.

Proposition 3.1. *Given a graph $G = (V, E)$ with $h(G) < 2$, it holds*

$$\mathrm{LGC}(G) < 2 \cdot \min \left\{ |S| : \ S \subseteq V, \ \frac{|\partial(S)|}{2} < |S| \leq \frac{|V|}{2} \right\} .$$

Proof. Let S be a smallest subset of vertices of G such that $|\partial(S)|/2 < |S| \leq |V|/2$, and let S' be a subset of vertices of $V - S$ such that $|S| = |S'|$. We assign integers $2i + 1$, $0 \leq i < |S|$, to vertices of S and integers $2i$, $1 \leq i \leq |S|$, to vertices of S'. Now, regardless of the specific assignment of integers to vertices of S and S', we need $2|S|$ edges outgoing from S ($|S|$ edges for the paths from $2i + 1$ to $2i + 2$, $0 \leq i < |S|$, and $|S|$ edges for the paths from $2i$ to $2i + 1$, $1 \leq i \leq |S|$). But since $h(G) < 2$, we have strictly less that $2|S|$ edges outgoing from S. Therefore, we have $\mathrm{LGC}(G) < 2 \cdot \min \{|S| : \ S \subseteq V, \ |\partial(S)|/2 < |S| \leq |V|/2\}$. □

As an example of how to use this technique, one can check that the edge expansion of the toroidal grid T_n of size $n \times n$, with $n \geq 6$, is strictly smaller than 2. Indeed, let $X_{p,q}$ be a subgrid of size $p \times q$ of T_n with $pq \leq \frac{n^2}{2}$. $X_{p,q}$ has pq vertices and $|\partial(X_{p,q})| = 2(p + q)$. So we have $h(T_n) \leq \min \left\{ \frac{2(p+q)}{pq}, \ 1 \leq pq \leq \frac{n^2}{2} \right\}$. When $n = 2p$ and $p = q$, we have $\frac{2(p+q)}{pq} = \frac{4}{p} = \frac{8}{n}$. Consequently, when $n \geq 6$, we have $h(T_n) \leq \frac{8}{6} < 2$. Now, remark that a set of 6 vertices of T_n

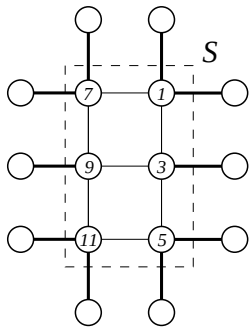


Figure 1: Example showing that $\mathrm{LGC}(G) < 12$ for G being an infinite square grid.

such as the set S illustrated in Figure 1 satisfies $|\partial(S)| = 10$ (that is, there are 10 edges going from S to $V \setminus S$), so by Proposition 3.1 $\mathrm{LGC}(T_n) < 12$. Summarizing,

Corollary 3.1. $\mathrm{LGC}(G) < 12$ *for G being an infinite square grid or a big enough toroidal grid.*

We shall see in Section 5 that the bound given in Corollary 3.1 can be improved using more involved arguments.

4 Hexagonal Grids, Cubic Graphs, and Graphs with Vertices of Small Degree

In this section we study some simple cases before studying in detail square grids in Section 5. Namely, we focus on hexagonal grids on Subsection 4.1, on cubic graphs in Subsection 4.2 and, more generally, on graphs with vertices of small degree in Subsection 4.3.

4.1 Hexagonal grids

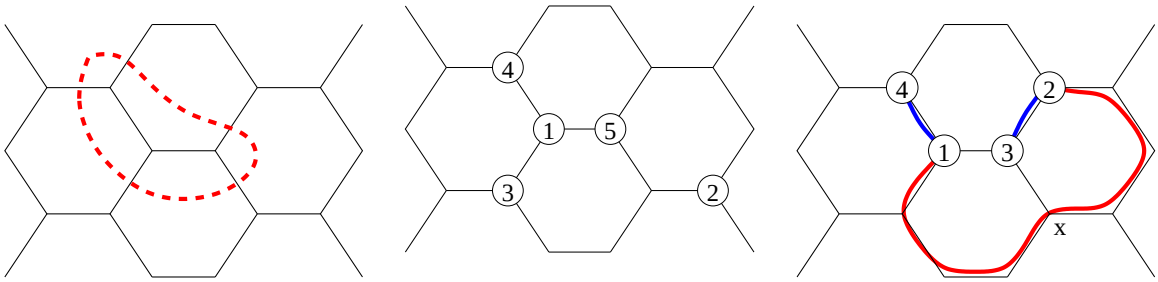
For the sake of presentation, we gradually improve in what follows the bound for LGC in hexagonal grids. First, note that from Proposition 3.1 we have that $\mathrm{LGC}(G) < 6$ for G being an infinite hexagonal grid, as $h(G) < 2$ and the smallest subset of vertices S such that $\frac{|\partial(S)|}{|S|} < 2$ has size 3, as shown in Figure 2(a). Nevertheless, we can obtain a better bound in the hexagonal grid without using Proposition 3.1.

Lemma 4.1. $\mathrm{LGC}(G) < 5$ *for G being an infinite hexagonal grid.*

Proof. Consider the example of Figure 2(b). In order to find a good circuit, we need in particular to find a path from vertex 1 to vertex 2. Suppose without loss of generality that this path contains edge $\{1, 4\}$. Thus, it also contains a second edge adjacent to vertex 4. Since vertex 4 has degree 3, it can no longer be in a path from vertex 3 to vertex 5 without reusing an edge of the path from vertex 1 to vertex 2. □

We can improve the result of Lemma 4.1 by using a more refined argument.

Lemma 4.2. $\mathrm{LGC}(G) < 4$ *for G being an infinite hexagonal grid.*



(a) Example for $k = 6$. (b) Example for $k = 5$. (c) Example for $k = 4$.

Figure 2: Examples for $k = 4, 5, 6$ in the infinite hexagonal grid.

Proof. Consider the example with 4 vertices of Figure 2(c). We distinguish two cases. First, if the path from vertex 1 to vertex 2 contains vertex 3 (resp. 4), then it is no longer possible to find edge-disjoints paths from vertex 2 to vertex 3 and from vertex 3 to vertex 4 (resp. between vertices 1 and 4, and vertices 3 and 4).

Suppose now, as for the example of Figure 2(c), that the path from vertex 1 to vertex 2 does not contain vertices 3 and 4. Since vertices along that path have degree 3, they cannot belong to a path between vertex 3 and vertex 2 or 4. Furthermore, we can assume that the path between vertices 2 and 3 uses only edge $\{2, 3\}$, since otherwise it would use edge $\{1, 3\}$ and it would not be possible to find a disjoint path between vertices 3 and 4. Thus, the path between vertices 3 and 4 has to use edge $\{1, 3\}$ plus another edge incident to vertex 1, and so it is no longer possible to find a path between vertices 1 and 4.

We conclude that there exists no good circuit for the placement of vertices of Figure 2(c). □

From Lemma 4.2 and Dirac's Theorem [6], we conclude that:

Proposition 4.1. $\mathrm{LGC}(G) = 3$ *for G being an infinite hexagonal grid.*

4.2 Cubic graphs

More generally, we obtain that:

Lemma 4.3. *For any cubic graph G,* $\mathrm{LGC}(G) < 5$.

Proof. Let G be a cubic graph. If $|V(G)| \leq 4$, then $\mathrm{LGC}(G) < 5$ by definition. If $|V(G)| \geq 6$ (recall that cubic graphs have an even number of vertices), we choose a vertex of G and we call it 1. Then, we call its three neighbors 3, 4, and 5 respectively. We now choose another vertex in $V(G) \setminus \{1, 3, 4, 5\}$ and we call it 2. Using similar arguments to those in the proof of Lemma 4.1, we obtain the result. □

Lemma 4.4. *For any planar cubic graph G different from K_4,* $\mathrm{LGC}(G) < 4$.

Proof. Note first that the only cubic planar graph without a face containing more than 3 vertices (including the external one) is K_4. Let now G be a cubic planar graph different from K_4, let $u_1, u_2, \ldots, u_l$, $l \geq 4$, be the vertices belonging to a face of G containing at least 4 vertices, and let us place number 4 on u_a, 1 on u_b, 3 on u_c, and 2 on u_d with $1 \leq a < b < c < d \leq l$. Using a proof similar to the one of Lemma 4.2, we obtain that G contains no edge-simple circuit visiting 1, 2, 3, and 4 in this order. □

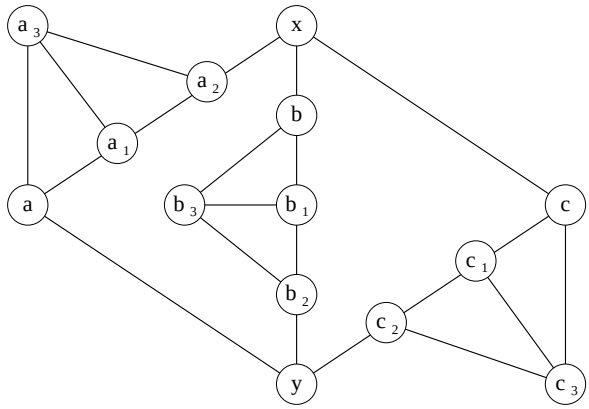


Figure 3: Example showing that it is not true that $\mathrm{LGC}(G) = 3$ for any 2-vertex-connected planar cubic graph G.

Now recall that by Dirac's Theorem [6], for $k \leq 3$, any k-vertex-connected graph G satisfies $\mathrm{LGC}(G) \geq k$. Combining this observation with Lemma 4.4 we obtain that

Corollary 4.1. *For any 3-vertex-connected planar cubic graph G different from K_4,* $\mathrm{LGC}(G) = 3$.

The bound on the connectivity in Corollary 4.1 is tight. Indeed, consider the 2-vertex-connected planar cubic graph G depicted in Figure 3. One can check that there is no edge-simple circuit visiting the vertices a, b, c, in this order. Therefore, for this graph G, $\mathrm{LGC}(G) = 2$.

To conclude this section, we note that it is easy to check that $\mathrm{LGC}(G) = 4$ for G being K_4 and $K_{3,3}$. On the other hand, it is clear that $\mathrm{LGC}(G) = 1$ for any 1-edge-connected graph G or any cubic 1-vertex-connected graph G. An interesting question is to completely characterize cubic graphs G for which $\mathrm{LGC}(G) = 3$.

4.3 Graphs with vertices of small degree

The above results are in fact based on the existence of configurations linking vertices of small degree in the graphs under study. We present here a collection of easy observations for graphs with vertices of small degree.

Fact 4.1. *If a graph G contains*

(i) one vertex of degree 1, then $\mathrm{LGC}(G) = 1$*;*

(ii) two adjacent vertices of degree 2, then $\mathrm{LGC}(G) < 4$*;*

(iii) one vertex of degree 2 adjacent to one vertex of degree 3, then $\mathrm{LGC}(G) < 4$*;*

(iv) one vertex of degree 3 adjacent to three vertices of degree 2, then $\mathrm{LGC}(G) < 3$*. In particular,* $\mathrm{LGC}(K_{2,3}) = 2$*. Also* $\mathrm{LGC}(G) = 2$ *if G is also 2-vertex-connected;*

(v) one vertex of degree 3 adjacent to two vertices of degree 3, then $\mathrm{LGC}(G) < 5$*.*

Proof.

(i) An edge-simple circuit cannot contain a vertex of degree 1.

(ii) If $|V(G)| \leq 3$, it is true by definition.

Let us now suppose that $|V(G)| \geq 4$. We give to the two vertices of degree 2 the numbers 2 and 4. One of the paths from 2 to 1 or from 2 to 3 has to go through 4. Vertex 4 has then no more available edges for the paths from 4 to 1 and from 4 to 3.

(iii) Same proof than (ii).

(iv) Let v be the vertex of degree 3, and u_1,u_2,u_3 be the vertices of degree 2. Now give numbers 1,2,3 to u_1,u_2,u_3, respectively. Since u_1 has degree 2, one of the paths between u_1 and u_2 or u_3 has to go through v, say the path $\{u_1, v, u_2\}$. Furthermore, since u_3 has also degree 2, one of the paths between u_3 and u_1 or u_2 has to go through v, but v has only degree 3.

When G is 2-vertex-connected, we use Dirac's Theorem [6] to conclude that $\mathrm{LGC}(G) = 2$.

(v) As such graph has at least 4 vertices, we can consider an example similar to the one used in the proof of Lemma 4.1, assigning numbers $1, 3, 4, 5$ to u, v_1, v_2, v_3, respectively (see Figure 2(b)).

□

5 Infinite Square Grids

In this section, we investigate the problem in infinite square grids (or large enough toroidal grids). Namely, we show in Subsection 5.1 that checking the feasibility of a configuration in the grid is equivalent to checking its feasibility in an auxiliary graph, which we call *internal graph*. In Subsection 5.2 we introduce the notion of *core graphs* in order to reduce the number of internal graphs to be tested. In Subsection 5.3 we give a counterexample establishing the upper bound of 10 for LGC in square grids. Finally, in Subsection 5.4 we match this upper bound with the help of an ILP solver to exhaustively test all the orderings for a small list of possible configurations that we obtained after applying the reductions of Subsections 5.1 and 5.2.

5.1 Preliminaries

We now introduce some additional definitions to be used throughout this section.

Definition 5.1 (configuration, feasible configuration)**.** *A* configuration *X is a subset of vertices of the infinite square grid. A configuration X is* feasible *if, for any permutation σ of the vertices of X, there exists a good circuit in the infinite square grid joining the vertices of X following the ordering given by σ.*

Definition 5.2 (internal graph, internal and external degree)**.** *Given a subset $X = \{u_1, \ldots, u_n\}$ of vertices in the square grid, the* internal graph *$G = (V, E)$ of X is the graph with $V = \{v_1, \ldots, v_n\}$, and for $u_i, u_j \in X$, $\{v_i, v_j\} \in E$ if and only if u_i and u_j are on the same row (or column) and there is no other $z \in X$ between u_i and u_j on that row (or column).*

Given $u_i \in X$, the internal degree $\deg_{\text{in}}(u_i)$ *of u_i is the degree of u_i in the internal graph G of X, i.e., $\deg_G(u_i)$. Similarly, the* external degree *of $u_i \in X$ is $\deg_{\text{out}}(u_i) = 4 - \deg_{\text{in}}(u_i)$. A vertex $u_i \in X$ is* isolated *if $\deg_{\text{in}}(u_i) = 0$.*

For example, in Figure 4, a configuration X in the square grid (defined by the full dots) and its corresponding internal graph G are depicted. The vertex labeled u satisfies $\deg_{\text{in}}(u) = 3$ and $\deg_{\text{out}}(u) = 1$.

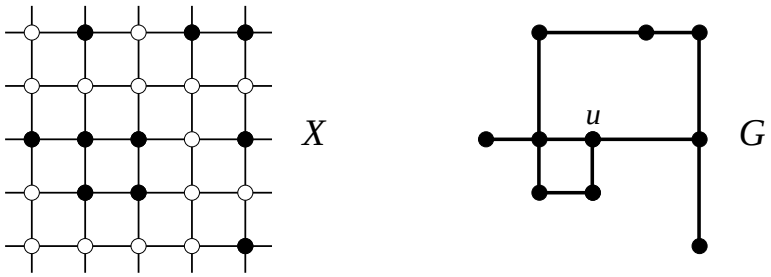


Figure 4: A configuration X (defined by the full dots) and its corresponding internal graph G.

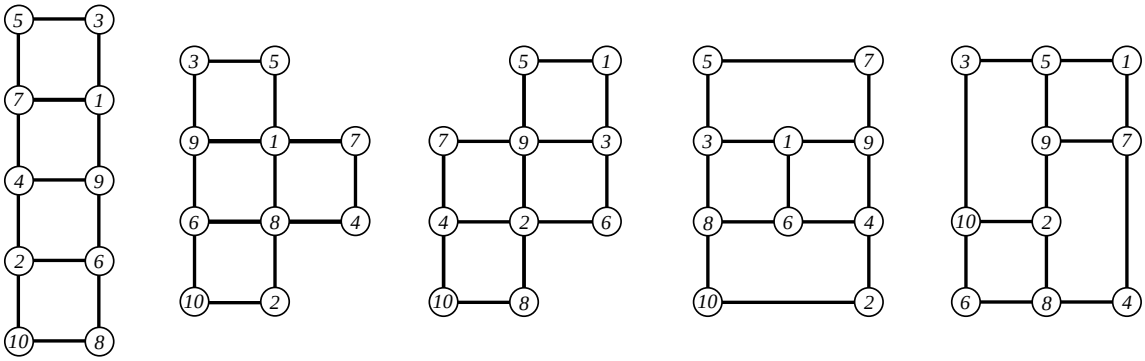


Figure 5: Some feasible internal graphs on 10 vertices.

Since we deal with an infinite square grid, any two vertices of an internal graph G with external degree at least one can be connected with a path in the grid without using any edge of G. This is because a vertex that has external degree at least one has no neighbor in the internal graph along an infinite semirow or semicolumn of the grid. This fact can be modeled in the following way: given an internal graph G, we construct a (multi)graph $\hat{G}$ from G by adding a new vertex ∞ and, for each vertex $u \in V(G)$, $\deg_{\text{out}}(u)$ copies of the edge $\{u, \infty\}$.

Definition 5.3 (feasible internal graph)**.** *An internal graph G is* feasible *if, for all the permutations σ of the vertices of G, there exists an edge-simple circuit in $\hat{G}$ joining the vertices of G following the ordering given by σ.*

The following lemma follows easily from the above definitions.

Lemma 5.1. *A configuration X is* feasible *if its internal graph G is feasible.*

Observe that the fact that G is feasible is a sufficient (but *not* necessary) condition for X to be feasible. Intuitively, the internal graph captures the most difficult case among all the configurations having the same internal graph.

Before getting into technical details, and in order to get familiar with the problem, the curious reader may verify that all the internal graphs on 10 vertices depicted in Figure 5 are feasible. We shall see in Subsection 5.4 that this fact is not a coincidence, as we will prove that any configuration on at most 10 vertices is feasible.

5.2 Reducing the problem

We now prove several technical lemmas to be used in the sequel of this section. The objective is to reduce the number of configurations to be tested.

Lemma 5.2. *Any internal graph in which all vertices have external degree at least* 2 *is feasible.*

Proof. Let G be an internal graph in which all vertices have external degree at least 2, and assume that the vertices are ordered $v_1, v_2, \ldots, v_k$ by the permutation σ. Then the circuit defined by vertices $\{v_1, \infty, v_2, \infty, v_3, \ldots, v_{k-1}, \infty, v_k, \infty, v_1\}$ is a solution in G. □

Lemma 5.3. *If an internal graph G is feasible then any internal graph H that is a subgraph of G is feasible.*

Proof. Let G be a feasible internal graph, and let H be a subgraph of G. That is, there exists an injective mapping φ from $V(H)$ to $V(G)$ such that if $\{u, v\} \in E(H)$, then $\{\varphi(u), \varphi(v)\} \in E(G)$. Assume first that $|V(H)| = |V(G)|$ (note that in that case, φ is a bijection), and let $v_1, \ldots, v_k$ be an ordering of the vertices of H. Consider a solution C in G for the ordering $\varphi(v_1), \ldots, \varphi(v_k)$ of the vertices of G. A solution in H is obtained from C by replacing each edge $\{x, y\} \in E(C)$ in the following way: if $\{\varphi^{-1}(x), \varphi^{-1}(y)\} \in E(H)$, we replace $\{x, y\}$ with $\{\varphi^{-1}(x), \varphi^{-1}(y)\}$; otherwise we replace it with the two edges $\{\varphi^{-1}(x), \infty\}, \{\infty, \varphi^{-1}(v)\}$. It is easy to check that the obtained circuit is a solution in H. Finally, if $|V(H)| = k < n = |V(G)|$, given an ordering $v_1, \ldots, v_k$ of $V(H)$, consider a solution C in G for any ordering of $V(G)$ that coincides with $\varphi(v_1), \ldots, \varphi(v_k)$ when restricted to the image of φ. Then the above replacement transforms C into a solution in H. □

Two internal graphs G_1 and G_2 are *equivalent* if G_2 is feasible if and only if G_1 is.

Lemma 5.4. *Any two isomorphic internal graphs G_1 and G_2 are equivalent.*

Proof. Let φ be an isomorphism mapping the vertices of G_1 to the vertices of G_2, and suppose without loss of generality that G_1 is feasible. We shall find a solution in G_2 for an arbitrary ordering $v_1, \ldots, v_k$ of its vertices. Let C be a solution in G_1 for the ordering $\varphi^{-1}(v_1), \ldots, \varphi^{-1}(v_k)$, which exists since G_1 is feasible. Let $P_{i,j}$ be the subpath of C going from $\varphi^{-1}(v_i)$ to $\varphi^{-1}(v_j)$. Since each path P_1 in G_1 corresponds bijectively to a path P_2 in G_2, we can denote without ambiguity the path P_2 by $\varphi(P_1)$. Then the circuit defined by the union of the paths $\varphi(P_{1,2}), \varphi(P_{2,3}), \ldots, \varphi(P_{k-1,k}), \varphi(P_{k,1})$ is a solution in G_2. □

Lemma 5.5. *If an internal graph G is feasible, then any internal graph G' that can be obtained from G via the following transformation T_1 is also feasible:*

(1) Choose from G an isolated vertex u and an edge $\{x, y\}$.

(2) Remove u, add a new vertex v, and replace the edge $\{x, y\}$ with the edges $\{x, v\}, \{v, y\}$.

Proof. Given an ordering σ of the vertices of G', let s (resp. t) be the vertex that precedes (resp. follows) v in σ. Consider a solution C in $\hat{G}$ for the ordering obtained from σ by replacing v with u, and let us obtain from C a solution in $\hat{G}'$. If the edge $\{x, y\}$ is used in C, replace it with the path $\{x, v, y\}$. Since $d_G(u) = 0$, the path from s to u in $\hat{G}$ is of the form $\{s, P, \infty, u\}$, and the path from u to t is of the form $\{u, \infty, Q, t\}$. Since $d_{G'}(v) = 2$, replacing the path $\{s, P, \infty, u\}$ (resp. $\{u, \infty, Q, t\}$) with $\{s, P, \infty, v\}$ (resp. $\{v, \infty, Q, t\}$) yields a solution in $\hat{G}'$. □

Lemma 5.6. *If an internal graph G is feasible, then any internal graph G' that can be obtained from G via the following transformation T_2 is also feasible:*

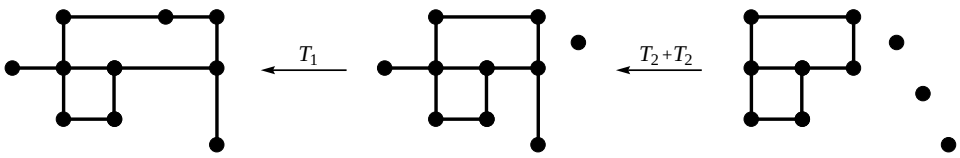


Figure 6: We can restrict ourselves to core graphs. An arrow from a graph G to a graph H means that if G is feasible, so is H (due to either transformation T_1 or transformation T_2).

(1) Choose from G two vertices u and w, such that u is isolated and $deg_{in}(w) \leq 3$.

(2) Remove u, and add a new vertex v and the edge $\{w, v\}$.

Proof. Given an ordering σ of the vertices of G', let s (resp. t) be the vertex that precedes (resp. follows) v in σ. Consider a solution C in $\hat{G}$ for the ordering obtained from σ by replacing v with u, and let us obtain from C a solution in $\hat{G}'$. If any of the edges $\{w, \infty\}$ is used in C, replace one of them with the edges $\{w, v\}, \{v, \infty\}$. Since $d_G(u) = 0$, the path from s to u in $\hat{G}$ is of the form $\{s, P, \infty, u\}$, and the path from u to t is of the form $\{u, \infty, Q, t\}$. Since $d_{G'}(v) = 1$, replacing the path $\{s, P, \infty, u\}$ (resp. $\{u, \infty, Q, t\}$) with $\{s, P, \infty, v\}$ (resp. $\{v, \infty, Q, t\}$) yields a solution in $\hat{G}'$. □

Combining inductively Lemmas 5.5 and 5.6, we deduce that if G' is an internal graph obtained from a feasible graph G with a sequence of the transformations T_1 and T_2, then G' is also feasible. In practice, this means that in any internal graph we can take the vertices of internal degree 2 that lie in the *middle* of a path and the vertices with internal degree 1, and put them as isolated vertices. If the resulting graph is a feasible internal graph, then by Lemmas 5.5 and 5.6, so is the original one. In other words, we can restrict ourselves to internal graphs G whose connected components (except isolated vertices) have at least two vertices in each row and each column.

Definition 5.4 (core graph, ℓ-core graph)**.** *An internal graph is a* core graph *if all its non-edgeless connected components have at least two vertices in each row and each column. A core graph G on k vertices is an ℓ-*core graph *if G has $k - \ell$ isolated vertices.*

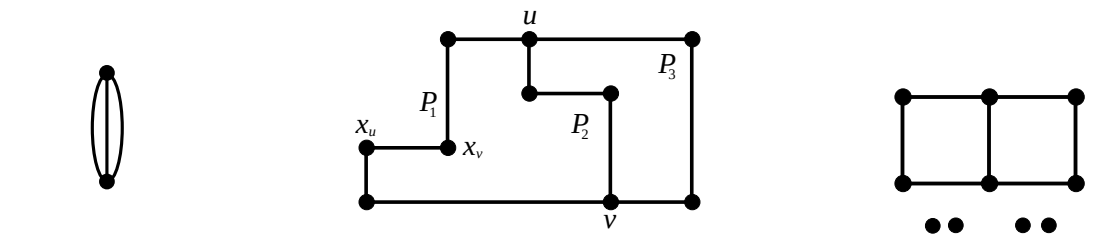
Lemmas 5.5 and 5.6 imply that we can restrict ourselves to core graphs. For instance, consider the example of Figure 6. The leftmost internal graph (which is the same example of Figure 4) can be obtained by a sequence of the transformations T_1 and T_2. Thus, to prove that the three internal graphs of Figure 6 are feasible it is enough to prove it for the rightmost graph, which is a 7-core graph.

This simplification reduces the number of configurations dramatically. In particular, the above discussion together with Lemma 5.2 proves that all forests are feasible. Therefore, if we want to know if all the configurations on k vertices are feasible, it suffices to test all the core graphs on k vertices; this is the topic of Subsection 5.4 for $k = 10$. Summarizing,

Proposition 5.1. *If all the core graphs on k vertices are feasible, then all the configurations on k vertices are feasible.*

Note that if all the configurations on k vertices are feasible, then clearly so are all the configurations on k' vertices, for every $k' < k$.

We introduce a last criterium to deduce the feasibility of an internal graph on 10 vertices.



(a) A triple edge. (b) A 10-core obtained from subdivision. (c) The only 6-core.

Figure 7: Graphs obtained from a triple edge by subdividing edges.

Lemma 5.7. *All the core graphs on 10 vertices whose non-edgeless connected components can be obtained from a triple edge by subdividing edges are feasible.*

Proof. Let G be a core graph whose non-edgeless connected components can be obtained from a triple edge (see Figure 7(a)). It is easy to check that the smallest core in the square grid that can be obtained in this way has 6 vertices (see Figure 7(c)), so since G has 10 vertices, G has exactly one such non-edgeless component. By construction, all the vertices of G have internal degree 2, except possibly isolated vertices and 2 vertices u and v that have internal degree 3, as it is exemplified in Figure 7(b). Suppose we are given an arbitrary ordering $x_1, \ldots, x_i = u, \ldots, x_j = v, \ldots, x_{10}$ of the vertices of G. By an argument analogous to the one of the proof of Lemma 5.2, to prove the feasibility of G it is sufficient to find two paths using only edges of G from u (resp. v) to x_{i-1} or x_{i+1} (resp. x_{j-1} or x_{j+1}), with $x_0 = x_{10}$ and $x_{11} = x_1$. We distinguish three cases according to the relative position of u and v in the ordering.

1. If u and v are consecutive in the ordering, any path from u to v in G fulfills the conditions.

2. If u and v are at distance 2 in the ordering, let w be the vertex which lies between u and v. Then take a path in G from u to v through w.

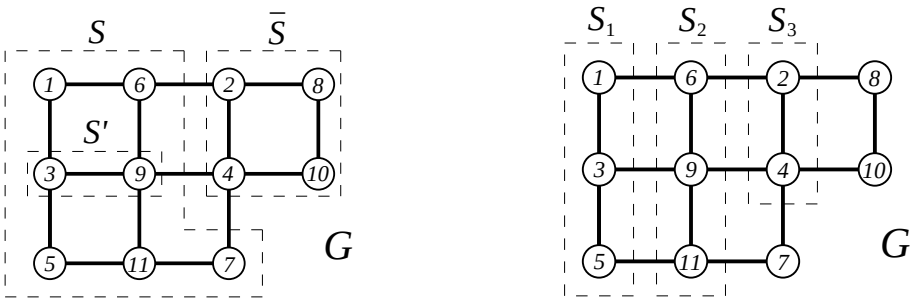
3. Otherwise, if u and v are at distance at least 3, u (resp. v) has at least one vertex at distance 1 in the ordering x_u (resp. x_v) inside G. Observe that there are 3 edge-simple paths P_1, P_2, P_3 between u and v in G. Suppose that $x_u \in P_i$ and $x_v \in P_j$. If $i \neq j$, take a path from u to x_u in P_i and a path from v to x_v in P_j. If $i = j$, assume without loss of generality that $i = j = 1$. If, beginning from u, x_u comes before x_v in P_1, consider the subpaths in P_1 from u to x_u and from v to x_v. Otherwise, if x_v comes before x_u in P_1 (see Figure 7(b)), consider the paths $\{u, P_2, v \xrightarrow{\text{in } P_1} x_u\}$ and $\{v, P_3, u \xrightarrow{\text{in } P_1} x_v\}$.

□

5.3 Upper bound

We have seen in Corollary 3.1 that $\mathrm{LGC}(G) < 12$ for G being an infinite square grid. In this section we present an unfeasible configuration proving that $\mathrm{LGC}(G) < 11$. We start with the following definition.

Definition 5.5 (internal path)**.** *Given an internal graph G, a permutation σ of X, a solution C to the instance (G, σ), and a subset $S \subseteq X$, an* internal path *in S is a subpath P of C linking two consecutive vertices of X according to σ, such that P is a subgraph of $G[S]$.*



(a) (b)

Figure 8: The counterexample (G, σ) used in Proposition 5.2 showing that $k < 11$, together with the vertex sets defined in the proof.

Given a subset of vertices $S \subseteq X$, the *paths originating from* S are paths with at least one endpoint in S.

Proposition 5.2. $\mathrm{LGC}(G) < 11$ *for* G *being an infinite square grid.*

Proof. Let (G, σ) be the internal graph on 11 vertices together with the ordering depicted in Figure 8. Suppose for the sake of contradiction that there exists a solution C to the instance (G, σ). Let $S = \{1, 6, 3, 9, 5, 11, 7\} \subseteq X$, and let $\overline{S} = \{2, 8, 4, 10\}$ (see Figure 8(a)). Note that there are 12 edges outgoing from $G[S]$ to the rest of the grid.

Claim 5.1. *C contains exactly 1 internal path in S.*

Proof. Suppose first that there is no internal path in S. Therefore, each path originating from S uses at least 2 edges outgoing from $G[S]$. Since $|S| = 7$, there must be 14 edges in C outgoing from $G[S]$ to the rest of the grid, but there are only 12. Therefore, C contains at least 1 internal path in S.

Suppose now that C contains at least 2 internal paths in S. Let $S' = \{3, 9\}$ (see Figure 8(a)), and note that there are 6 edges outgoing from $G[S']$. Note also that the only possible internal paths in S are $5 \to 6$ (we adopt this notation here for simplicity), $6 \to 7$, and $11 \to 1$, so any internal path in S must cross S'. Therefore, there can be at most 2 such internal paths, and those 2 paths use 4 edges outgoing from $G[S']$. Thus, only $6 - 4 = 2$ outgoing edges from $G[S']$ are left, which are not enough to route the 4 subpaths in C containing the vertices of S'. Therefore, C contains exactly 1 internal path in S. □

Claim 5.1 implies that all the edges outgoing from $G[S]$ are used by C to route paths originating at S. Let now $S_1 = \{1, 3, 5\}$ and $S_3 = \{2, 4\}$ (see Figure 8(b)).

Claim 5.2. *C contains at least 2 internal paths from S_1 to S_3.*

Proof. Note that subgraph $G[S_3]$ has 6 outgoing edges. Since all the edges outgoing from $G[S]$ are used by C, exactly 3 paths go from S to $\overline{S}$ in C. Clearly, the 4 subpaths in C containing the vertices of S_3 use 4 outgoing edges from $G[S_3]$. Note that all paths from S to S_3 are from S_1.

If there is no path in C from S_1 to S_3, then the 3 paths from S to $\overline{S}$ cross S_3, so no edge outgoing from $G[S_3]$ would be left to route the paths originating from S_3, which is a contradiction.

If there is 1 path in C from S_1 to S_3, then 2 paths from S to $\overline{S}$ cross S_3, so altogether the 3 paths from S to $\overline{S}$ use 5 out of the 6 outgoing edges from $G[S_3]$. However, 3 additional outgoing edges from $G[S_3]$ would be needed to route the 3 remaining paths originating from S_3, which is a contradiction. □

Consider now $S_2 = \{6, 9, 11\}$ (see Figure 8(b)). The subgraph $G[S_2]$ has 8 outgoing edges, and 6 of them are required in C to route the paths originating at S_2, so only 2 edges outgoing from $G[S_2]$ are still available in C. But, by Claim 5.2, C contains at least 2 internal paths from S_1 to S_3 (which cross S_2), hence using 4 outgoing edges from $G[S_2]$. The proposition follows. □

5.4 Lower bound

To show that $\mathrm{LGC}(G) \geq 10$ for G being an infinite square grid, one has a priori to test whether all the configurations with 10 vertices on the grid are feasible or not. But, the number of such configurations is prohibitively big, and testing a single configuration may take a non-negligible time (see discussion below). Hence we introduce a methodology, based on the notion of *core graph* (see Subsection 5.2) in order to reduce the number of configurations to be tested.

A naïve strategy to generate all configurations is to consider all the possibilities of placing 10 points in the square grid. However, we showed in Proposition 5.1 that we only need to consider *core graphs* with 10 vertices, introduced in Definition 5.4. In addition, by Lemma 5.4 these core graphs can be considered modulo isomorphism. It is clear that the smallest integer i such that an i-core on 10 vertices exists is 4, and in that case the non-edgeless connected component of the 4-core is a 4-cycle. Such a core is always feasible due to Lemma 5.2, because all the vertices have external degree at least 2. It is also easy to check that, due to the topology of the grid, a 5-core cannot exist. One can also verify that the only 6-core in which not all vertices have external degree at least 2 is a 2×3-grid (see Figure 7(c)). Therefore, it is enough to test this 6-core plus all the ℓ-cores on 10 vertices, for $\ell = 7, 8, 9, 10$. The procedure to generate the core graphs to be tested is detailed in Algorithm 1. The complete code and some examples can be found at [1].

Proposition 5.3. *The feasibility of any configuration on 10 vertices follows from Algorithm 1.*

Proof. To prove the correctness of Algorithm 1, we analyze each step sequentially.

Step 2: As by definition, core graphs have at least 2 vertices in each row (resp. column) (besides isolated vertices), the number of rows (resp. column) of an ℓ-core is at most $\lfloor \frac{\ell}{2} \rfloor$. Hence, it is enough to consider internal graphs fitting into an $(\lfloor \frac{\ell}{2} \rfloor \times \lfloor \frac{\ell}{2} \rfloor)$-grid, except possibly a set of isolated vertices. Moreover, two internal graphs G_1 and G_2 such that G_2 is obtained from G_1 by a translation or a symmetry are clearly equivalent (Lemma 5.4).

In steps 3-5, the algorithm removes from $\mathcal{T}_\ell$ all the internal graphs with some vertex of internal degree 1. This can be done because, by Lemma 5.6, it is enough to test the internal graphs in which all vertices, except the isolated ones, have internal degree at least 2.

In steps 6-8, the algorithm removes from $\mathcal{T}_\ell$ all the internal graphs such that all vertices have external degree at least 2, which are feasible due to Lemma 5.2.

In steps 9-15, the algorithm identifies the isomorphism classes of core graphs (step 10) and kept at most one representative per class. More precisely, in steps 11-13, the algorithm removes from $\mathcal{T}_\ell$ all the isomorphism classes that have some representative without at least 2 vertices per row and column. This can be done by combining Lemma 5.4 and Proposition 5.1. Then, by Lemma 5.4, all the graphs in an isomorphism class are equivalent, so it is enough to keep one graphs of each class, as it is done in step 15.

The correctness of steps 16-19 (resp. 20-22) follows directly from Lemma 5.7 (resp. Lemma 5.3).

Algorithm 1 Test configurations on 10 vertices

```
for ℓ = 6 to 10 do
    generate a list T_ℓ of all the internal graphs on ℓ vertices in an (⌊ℓ/2⌋ × ⌊ℓ/2⌋)-grid
    (modulo translations, symmetries, and compression of empty rows or columns)
    // INTERNAL DEGREE 1:
    for all G ∈ T_ℓ such that G has some vertex of internal degree 1 do
        remove G from T_ℓ {Lemma 5.6}
    // EXTERNAL DEGREE AT LEAST 2:
    for all G ∈ T_ℓ such that all the vertices of G have external degree at least 2 do
        remove G from T_ℓ {Lemma 5.2}
    // ISOMORPHIC GRAPHS:
    partition T_ℓ into classes G_1, ..., G_n of isomorphic graphs
    for i = 1 to n do
        if there exists G ∈ G_i without at least two vertices per row and column then
            remove from T_ℓ all the graphs in G_i {Lemma 5.4 and Proposition 5.1}
        else
            remove from T_ℓ all the graphs in G_i except one {Lemma 5.4}
    // SUBDIVISION OF TRIPLE EDGE:
    if ℓ = 10 then
        for all G ∈ T_ℓ such that G can be obtained from a triple edge by subdividing edges
        do
            remove G from T_ℓ {Lemma 5.7}
    // SUBGRAPHS:
    for each pair of graphs G, H ∈ T_ℓ such that H is a subgraph of G do
        remove H from T_ℓ {Lemma 5.3}
    b_ℓ ← 1
    for each G ∈ T_ℓ do
        G' ← G + (10 − ℓ) isolated vertices
        for each permutation σ of the vertices of G' do
            test if (G', σ) is feasible using an ILP solver
            if (G', σ) is not feasible then
                b_ℓ ← 0
if (b_6 · b_7 · b_8 · b_9 · b_10) == 1 then
    k = 10
else
    k < 10
```

In steps 23-29, the feasibility of each pair (G', σ) is tested using an ILP solver to find a solution of the associated integer multicommodity flow problem, as explained in Section 2 (more details about the implementation are available at [1]).

Finally, by Proposition 5.1, it is clear that $k = 10$ if and only if all the ℓ-core graphs are feasible, for each $\ell \in \{6, 7, 8, 9, 10\}$, as tested in steps 30-33. □

Remark 5.1. *In step 10 of Algorithm 1, we partition* T_ℓ *into isomorphism classes. This step could take a non-negligible time if we just test if each pair of graphs are isomorphic. To deal with this problem, we first carry out a sieve according to the sorted degree sequence of the vertices and the sorted degree sequences of the neighbours of each vertex. That is, if two graphs do not have the same sequence of degrees and degrees of the neighbours of each vertex, we infer directly that these two graphs are not isomorphic. This sieve reduces the computation time considerably.*

Remark 5.2. *Observe that, due to Lemma 5.5, the internal graphs without at least 2 vertices per row and column could have been already removed from* T_ℓ *after step 2. The reason why we kept those graphs until step 13 is that some graphs that do have at least 2 vertices per row and column are isomorphic to graphs without at least 2 vertices per row and column, so we can also remove them from* T_ℓ*.*

Table 1 summarizes the number of ℓ-cores obtained while running Algorithm 1, for $\ell \in \{6, 7, 8, 9, 10\}$. The numbers given in the first row (initial number of internal graphs) follow from the introduction of internal graphs; without it, we would have a much greater number of configurations to test. Note that the results of Subsection 5.2 imply an overall reduction from 4714 to 52 graphs.

Testing the feasibility of core graphs. Recall that for each core graph G on 10 vertices, G is feasible if for any ordering of $V(G)$ there is a good circuit visiting $V(G)$ in the prescribed order. Without loss of generality, we can assign to one of the vertices of G the number 1 of the permutation (modulo cyclic permutations), and then for each core graph one has to test $9! = 362.880$ possibilities.

For each core graph G and permutation σ, the problem we study can be easily formulated as an integer multicommodity flow problem in a graph with unitary capacity on the edges and so as an integer linear program (ILP). Indeed, the existence of a good circuit C_σ in a core graphs G is equivalent to the existence of k edge-disjoint paths in G between the pairs of vertices (or *commodities*) $\{\sigma(1), \sigma(2)\}, \ldots, \{\sigma(k-1), \sigma(k)\}, \{\sigma(k), \sigma(1)\}$. Thus, a feasible solution of the ILP implies the existence of a good circuit, and this feasibility can be quickly checked using an ILP solver (for instance, CPLEX or GLPK).

In average, testing the 9! permutations for each internal graph takes around 40 minutes on a PC with an Intel Core 2 Duo CPU 2.33GHz running Fedora 8 (see [1]), so testing the 4714 internal graphs would take around 4 months and a half. Testing the 52 remaining graphs has taken just 35 hours and 37 minutes [1].

Running the ILP solver on the configurations given by Algorithm 1, we obtained that all ℓ-cores are feasible for each $\ell \in \{6, 7, 8, 9, 10\}$. Therefore, combining Propositions 5.2 and 5.3 yields that

Theorem 5.1. *There exists an edge-simple circuit through any set of 10 ordered vertices of an infinite square grid or a large enough toroidal grid.*

We would like to stress that the unfeasible configuration used in the lower bound of Proposition 5.2 has been found using our computer-based techniques, and then we analyzed it to formally prove that it is indeed unfeasible.

Graphs \ ℓ	6	7	8	9	10	Total
Initial number of internal graphs	1	7	53	485	4166	4714
Number of isomorphisms	0	3	42	453	4051	4581
Number of subgraphs	0	0	5	10	58	73
Number of single graphs	0	2	6	22	74	104
Final number of internal graphs	1	2	4	10	35	52

Table 1: Number of ℓ-core graphs on 10 vertices in Algorithm 1. A *single* graph is a graph with a line or column with only one vertex.

6 Concluding Remarks

In this article we studied the problem that consists in, given a graph G, determining which is the largest integer k such that for any collection of k ordered vertices in G, there exists an edge-simple circuit visiting those vertices in the prescribed order. We denote this integer by $\mathrm{LGC}(G)$. After providing an upper bound for general graphs, we first investigated this question for graphs with maximum degree at most 3, or graphs with vertices of small degree. In particular, we proved that $\mathrm{LGC}(G) = 3$ for G being an infinite hexagonal grid, and that for any 3-vertex-connected planar cubic graph G different from K_4, $\mathrm{LGC}(G) = 3$. This bound on the connectivity is tight, namely we showed that there exist 2-vertex-connected planar cubic graphs G for which $\mathrm{LGC}(G) = 2$.

Our main result is that $\mathrm{LGC}(G) = 10$ for G being an infinite square grid. To prove this, we introduced a methodology in order to reduce the problem to a small number of configurations, which were then exhaustively tested using an ILP solver. The details about the implementation of our algorithm are available at [1]. Finding a purely combinatorial proof of this result remains open.

Another avenue for further research could be to impose a bound on the size of the square (or toroidal) grid, namely to consider LGC on a $(n_1 \times n_2)$-grid. Considering triangular grids may be also interesting, mainly due to its extensive use in telecommunication networks.

A more ambitious problem is, given a positive integer k, characterizing graphs G for which $\mathrm{LGC}(G) = k$. In particular, it would be interesting to characterize all graphs G with maximum degree at most 3 for which $\mathrm{LGC}(G) = 3$.

Finally, adding the constraint of the prescribed order to the classical problems concerning the existence of circuits (see related work in Section 2), creates a whole family of new problems to consider.

Acknowledgement. We would like to thank the anonymous referees for helpful remarks that improved the presentation of the paper.

References

[1] *http://www-sop.inria.fr/members/Frederic.Giroire/circuit.*

[2] R.E.L. Aldred, S. Bau, D.A. Holton, and B. McKay, *Cycles through 23 vertices in 3-connected cubic planar graphs*, Graphs and Combinatorics **15** (1999), 373–376.

[3] S. Bau and D. Holton, *Cycles containing 12 vertices in 3-connected cubic graphs*, J. Graph Theory **15** (1991), no. 4, 421–429.

[4] J.A. Bondy and L. Lovász, *Cycles through specified vertices of a graph*, Combinatorica **1** (1981), 117–140.

[5] R. Diestel, *Graph Theory*, Springer-Verlag, 2005.

[6] G. Dirac, *In abstrakten Graphen vorhandene vollständige 4-Graphen u nd ihre Unterteilungen*, Math. Nachr. **22** (1960), 61–85.

[7] Y. Egawa, R. Glas, and S.C. Locke, *Cycles and paths through specified vertices in k-connected graphs*, J. Comb. Theory Ser. B **52** (1991), 20–29.

[8] M.N. Ellingham, D.A. Holton, and C.H.C. Little, *Cycles through ten vertices in 3-connected cubic graphs*, Combinatorica **4** (1984), 265–273.

[9] S. Fortune, J. Hopcroft, and J. Wyllie, *The directed subgraph homeomorphism problem*, J. of Theor. Comp. Sci. **10** (1980), no. 2, 111–121.

[10] A. Frank and A. Schrijver, *Edge-Disjoint Circuits in Graphs on the Torus*, J. Comb. Theory Ser. B **55** (1992), no. 1, 9–17.

[11] F. Gringa, J. Harant, E. Hexel, and Zs. Tuzac, *On short cycles through prescribed vertices of a graph*, Discr. Math. **286** (2004), no. 1-2, 67–74.

[12] R. Häggkvist and C. Thomassen, *Circuits through specified edges*, Discr. Math. **41** (1982), 29–34.

[13] A. Jarry and S. Pérennes, *Disjoint paths in symmetric digraphs*, Discr. Appl. Math. **157** (2009), no. 1, 90–97.

[14] R. M. Karp, *On the complexity of combinatorial problems*, Networks **5** (1975), 45–68.

[15] K. Kawarabayashi, *Cycles through a prescribed vertex set in n-connected graphs*, J. Comb. Theory Ser. B **90** (2004), no. 2, 315–323.

[16] A.K. Kelmans and M.V. Lomonosov, *When m vertices in a k-connected graph cannot be walked round along a simple cycle*, Discr. Math. **38** (1982), 317–322.

[17] M.R. Kramer and J. van Leeuwen, *Wire-routing is NP-complete*, Tech. Report RUU-CS-82-4, Department of Computer Science, University of Utrecht, 1982.

[18] A.S. LaPaugh and R.L. Rivest, *The subgraph homeomorphism problem*, J. Comput. Syst. Sci. **20** (1980), no. 2, 133–149.

[19] L. Lovász, *On some connectivity properties of Eulerian graphs*, Acta Math. Acad. Sci. Hungar. **28** (1976), 129–138.

[20] N. Robertson and P.D. Seymour, *Graph minors. XIII. The disjoint paths problem*, J. Comb. Theory Ser. B **63** (1995), 65–110.

[21] N. Robertson, P.D. Seymour, and R. Thomas, *Quickly excluding a planar graph*, J. Comb. Theory Ser. B **62** (1994), no. 2, 323–348.