

Edge-Simple Circuits Through 10 Ordered Vertices in Square Grids

David Coudert, Frédéric Giroire, Ignasi Sau

▶ **To cite this version:**

David Coudert, Frédéric Giroire, Ignasi Sau. Edge-Simple Circuits Through 10 Ordered Vertices in Square Grids. International Workshop on Combinatorial Algorithms – IWOCA, Jun 2009, Hradec nad Moravicì, Czech Republic. inria-00429146

HAL Id: inria-00429146

https://inria.hal.science/inria-00429146

Submitted on 31 Oct 2009

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.

Edge-Simple Circuits Through 10 Ordered Vertices in Square Grids*

David Coudert[1], Frédéric Giroire[1], Ignasi Sau[1,2]

1. Mascotte, INRIA, I3S, CNRS, University of Nice Sophia, France.
2. Graph Theory and Combinatorics group - Applied Mathematics IV Department of UPC - Barcelona, Spain.

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 $\{v_i, v_{i+i}\} \in E$ for $i = 1, \ldots, k$. A circuit C is said to be *edge-simple* if no edge of G is used twice in C. In this article we study the following problem: which is the largest integer k such that, given any subset of k ordered vertices of an infinite square grid, there exists an edge-simple circuit visiting the k vertices in the prescribed order? We prove that $k = 10$. To this end, we first provide a counterexample implying that $k < 11$. To show that $k \geq 10$, we introduce a methodology, based on the notion of core graph, to reduce drastically the number of possible vertex configurations, and then we test each one of the resulting configurations with an ILP solver.

Keywords: square grid, edge-simple circuit, prescribed vertices, ILP solver.

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. An edge-simple circuit is also called *closed trail* in the literature. The existence of a circuit through a prescribed set of vertices or edges has been an important graph-theoretical question for many years [Dir60, Lov76, HT82, Kaw04, BL81, BH91, EHL84, EGL91, KL82, ABHM99, GHHT04, FS92]. 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 nodes [Dir60] or any subset of $k - 1$ independent edges [HT82] 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 existence of edge-simple circuits through specified vertices in the infinite square grid (or equivalently, a large enough torus), which is a widely studied 4-connected graph. In addition, we do not require the circuit only to visit a subset of vertices, but also to visit them in a prescribed order. It is clear that such a circuit in the square grid always exists for any ordered subset of 4 vertices. After thinking for a few minutes it is also easy to convince oneself that the same holds for 5 vertices. On the other hand, it seems intuitive to

*This work has been partially supported by European project IST FET AEOLUS, PACA region of France, Ministerio de Ciencia e Innovación, European Regional Development Fund under project MTM2008-06620-C03-01/MTM, and Catalan Research Council under project 2005SGR00256.

suspect that this property will not be true for an arbitrary large subset of ordered vertices of the square grid. Therefore, the following question arises: which is the largest integer k such that, given *any* subset of k ordered vertices of an infinite square grid, there exists an edge-simple circuit visiting the k vertices in the prescribed order? Here, we prove that $k = 10$.

To obtain this result, one has a priori to test the existence of an edge-simple circuit visiting k vertices in the prescribed order on the grid, for all possible placements and orderings of the k vertices. Since the number of possible placements and orderings is prohibitively large, we introduce a methodology, based on the notion of *core graph*, to reduce the number of configurations to be tested. We first provide some background and motivations for the problem in Section 2. We then show in Section 3 that checking the feasibility of a configuration on the grid is equivalent to checking its feasibility on an auxiliary graph, called *internal graph*. Then, in Section 4 we introduce the notion of *core graphs* to reduce drastically the number of internal graphs to be tested. In Section 5 we give a counterexample establishing the upper bound $k \leq 10$. In Section 6 we match this upper bound with an ILP solver to exhaustively test all the orderings for a small list of possible configurations that we obtained after applying the reductions of Sections 3 and 4. Finally, Section 7 concludes the article.

2 Background and Motivation

Connectivity is one of the cornerstone concepts of graph theory. Maybe the most archetypal results are Menger's classical theorems [Die05], 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. For instance, it is well known that in a k-vertex-connected graph any subset of k nodes [Dir60] or any subset of $k-1$ independent edges [HT82] 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 [Kaw04, BL81, EGL91, KL82].

Some stronger results have been given for specific classes of graphs, like 3-connected cubic graphs [EHL84, EGL91]. 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 [EHL84] in such a way that each of the 10 vertices maps to a distinct vertex of the Petersen graph. If, in addition, the 3-connected cubic graph is planar, then there exists a cycle through any 23 vertices [ABHM99]. Another example can be found in [FS92], 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.

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 [Kar75] (even if the graph is a square grid [Kv82]). When the number of pairs of vertices is bounded by a constant, the disjoint paths problem is polynomial in undirected graphs [RS95], NP-complete in directed graphs [LR80] (even with only two pairs of vertices [FHW80]), and polynomial in symmetric directed graphs [JP09].

However, all these results do not take into account the *order* in which the cycle visits the prescribed set of nodes. This is a natural constraint, since for example in telecommunication networks it may be important to connect a subset of nodes in such a way that each node numbered i has capability to communicate only with the two nodes numbered $i-1$ and $i+1$ (modulo the cardinality of the subset of nodes). 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 the constraint on the order is introduced. In this article we

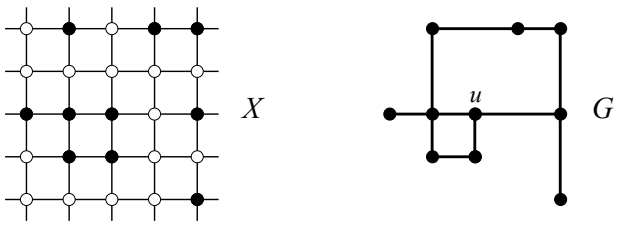


Figure 1: A configuration X (defined by the full dots) and its corresponding internal graph G.

study one of these problems in square grids.

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 nodes. 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 the square 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 [RST94]. Therefore, a square grid is *inside* every planar graph, and any edge-disjoint circuit in a minor of a graph can be easily transformed to an edge-disjoint circuit in the graph itself.

3 Preliminaries

In this section we introduce some definitions to be used throughout. We use standard graph terminology (see, for instance, [Die05]).

Definition 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 an edge-simple circuit in the infinite square grid joining the vertices of X following the ordering given by σ.*

Definition 2 (Internal graph, internal and external degree) *Given a subset $X = \{u_1, \ldots, u_n\}$ of nodes 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 $u \in X$, the internal degree $d_{in}(u)$ *of u is the degree of u in the internal graph G of X, i.e., $d_G(u)$. Similarly, the* external degree *of $u \in X$ is $d_{out}(u) = 4 - d_{in}(u)$. A vertex $u \in X$ is* isolated *if $d_{in}(u) = 0$.*

For example, in Fig. 1, 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 $d_{\text{in}}(u) = 3$ and $d_{\text{out}}(u) = 1$.

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

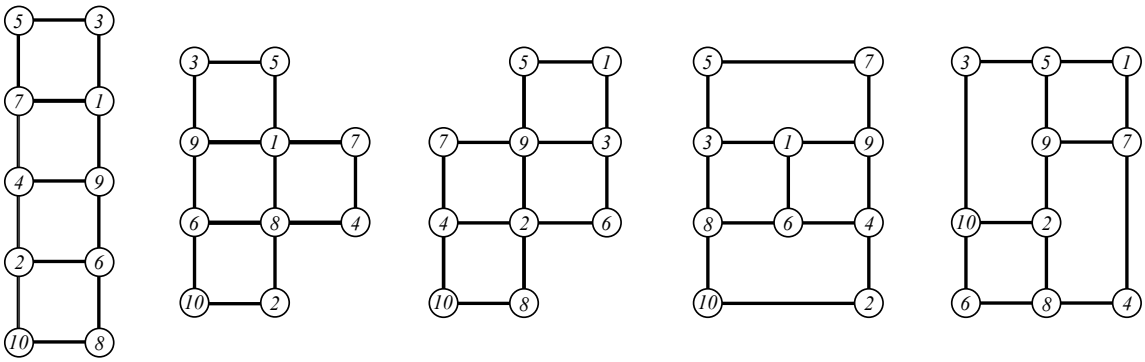


Figure 2: Some feasible internal graphs on 10 vertices.

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 $u \in V(G)$, $d_{\text{out}}(u)$ copies of the edge $\{u, \infty\}$.

Definition 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 3.1 *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 results, and in order to get familiar with the problem, the curious reader may verify that the internal graphs on 10 vertices depicted in Fig. 2 (together with a numbering of their vertices) are feasible. We shall see in Section 6 that this fact is not a coincidence, since any configuration on 10 vertices is feasible.

4 Reducing the Problem

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

Lemma 4.1 *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 $\{v_1, \infty, v_2, \infty, v_3, \ldots, v_{k-1}, \infty, v_k, \infty, v_1\}$ is a solution in G. □

Lemma 4.2 *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. Assume first that $|V(H)| = |V(G)|$, and let $v_1, \ldots, v_k$ be an ordering of the vertices of H. Consider a solution C in G for the same ordering $v_1, \ldots, v_k$ of the vertices of G. A solution in H is obtained from

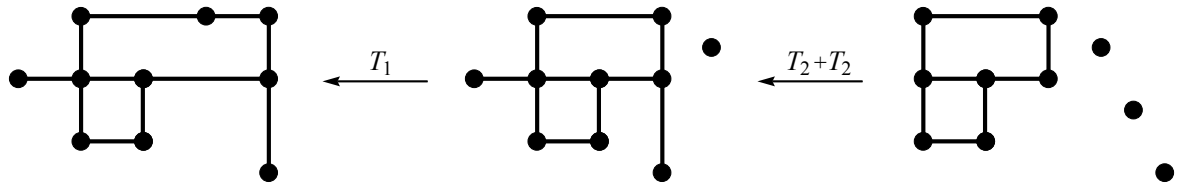


Figure 3: We can restrict ourselves to core graphs. An arrow from a graph G to a graph H means that if G is feasible, so is H (due to either transformation T_1 or transformation T_2).

C by replacing each each $\{u, v\} \in E(G) \setminus E(H)$ with the edges $\{u, \infty\}, \{\infty, v\}$. Otherwise, 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 an ordering of $V(G)$ that coincides with $v_1, \ldots, v_k$ when restricted to $V(H)$. Then the above replacement transforms C into a solution in H. □

The proofs of the three following lemmas can be found in [CGS09]. Two internal graphs G_1 and G_2 are said to be *equivalent* if G_2 is feasible if and only if G_1 is.

Lemma 4.3 *Any two isomorphic internal graphs G_1 and G_2 are equivalent.*

Lemma 4.4 *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 $\{x, v\}, \{v, y\}$.

Lemma 4.5 *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:*

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

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

Combining inductively Lemmas 4.4 and 4.5, 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 that lie in the *middle* of a path and the vertices with internal degree one, and put them as isolated vertices. If the resulting graph is a feasible internal graph, then by Lemmas 4.4 and 4.5, 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 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 4.4 and 4.5 imply that we can restrict ourselves to core graphs. For instance, consider the example of Fig. 3. The leftmost internal graph (which is the same example of Fig. 1) can be obtained by a sequence of the transformations T_1 and T_2. Thus, to prove that the three internal graphs of Fig. 3 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 4.1 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 Section 6 for $k = 10$. Summarizing,

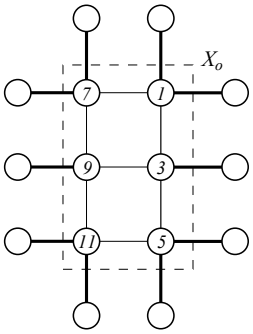


Figure 4: Counterexample showing that $k < 12$.

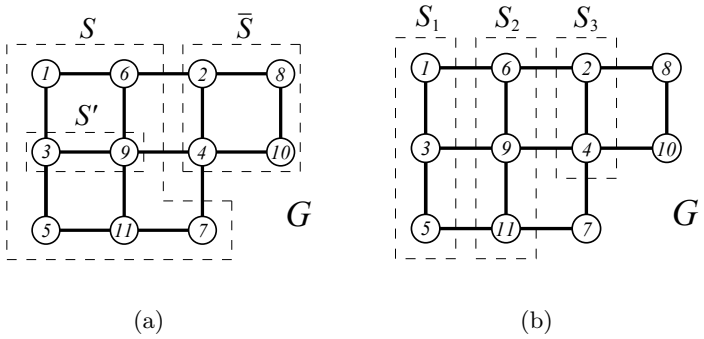


Figure 5: Counterexample (G, σ) of Proposition 5.1 showing that $k < 11$, together with the vertex sets defined in the proof.

Proposition 4.1 *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. The proof can be found in [CGS09].

Lemma 4.6 *All the 10-core graphs on 10 vertices whose non-edgeless connected components can be obtained from a triple edge by subdividing edges are feasible.*

5 Upper Bound: $k < 11$

In this section we show an unfeasible counterexample proving that $k < 11$. For the sake of the presentation, we first describe a simple configuration showing that $k < 12$.

Given a set $X = \{1, 2, 3, \ldots, 12\}$ of ordered nodes in a square grid G, let X_e (resp. X_o) be the subset of nodes of X with an even (resp. odd) number, and note that any path joining two consecutive vertices must go from X_o to X_e, or viceversa. Let X_o be the set displayed in Fig. 4. Then, regardless of the placement of X_e, we need at least 12 edges outgoing from the graph induced by X_o to route the 12 paths, but there are only 10 such edges (the thick edges in Fig. 4). So, this configuration is unfeasible for any placement of X_e.

Before providing the counterexample showing that $k < 11$, we need the following definition.

Definition 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 $G[S]$.*

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

Proposition 5.1 $k < 11$.

Proof: Let (G, σ) be the internal graph on 11 vertices together with the ordering depicted in Fig. 5.

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 Fig. 5(a)). Note that there are 12 edges outgoing from $G[S]$ to the rest of the grid.

Claim 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 Fig. 5(a)), and note that there are 6 edges outgoing from $G[S']$. Note also that the only possible internal paths in S are $5 \to 6$, $6 \to 7$, 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 1 implies all the edges outgoing from $G[S]$ are used by C to route paths originating at S. Let $S_1 = \{1, 3, 5\}$ and $S_3 = \{2, 4\}$ (see Fig. 5(b)).

Claim 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 Fig. 5(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 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. □

6 Lower Bound: $k \geq 10$

To show that $k \geq 10$, one has a priori to test all the configurations with 10 vertices on the grid are feasible. But, the number of such configurations is prohibitively big, as testing a single configuration may take a non-negligible (see discussion below). Hence we introduce a methodology, based on the notion of *core graph* (see the results of Section 4), 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 4.1 that we only need to consider *core graphs* with 10 vertices (Definition 4). In addition, these core graphs can be considered

modulo isomorphism (Lemma 4.3). 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 4.1, 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. 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 as well can be found at [cod].

Algorithm 1 Test configurations on 10 vertices

for $\ell = 6$ to 10 **do**
  generate a list $\mathcal{T}_\ell$ of all the internal graphs on ℓ vertices in an $(\lfloor \frac{\ell}{2} \rfloor \times \lfloor \frac{\ell}{2} \rfloor)$-grid (modulo translations, symmetries, and compression of empty rows or columns)
  // `INTERNAL DEGREE 1:`
  for all $G \in \mathcal{T}_\ell$ such that G has some vertex of internal degree 1 **do**
    remove G from $\mathcal{T}_\ell$ {*Lemma 4.5*}
  // `EXTERNAL DEGREE AT LEAST 2:`
  for all $G \in \mathcal{T}_\ell$ such that all the vertices of G have external degree at least 2 **do**
    remove G from $\mathcal{T}_\ell$ {*Lemma 4.1*}
  // `ISOMORPHIC GRAPHS:`
  partition $\mathcal{T}_\ell$ into classes $\mathcal{G}_1, \ldots, \mathcal{G}_n$ of isomorphic graphs
  for $i = 1$ to n **do**
    if there exists $G \in \mathcal{G}_i$ without at least two vertices per row and column **then**
      remove from $\mathcal{T}_\ell$ all the graphs in $\mathcal{G}_i$ {*Lemma 4.3 and Proposition 4.1*}
    else
      remove from $\mathcal{T}_\ell$ all the graphs in $\mathcal{G}_i$ except one {*Lemma 4.3*}
  // `SUBDIVISION OF TRIPLE EDGE:`
  if $\ell = 10$ **then**
    for all $G \in \mathcal{T}_\ell$ such that G can be obtained from a triple edge by subdividing edges **do**
      remove G from $\mathcal{T}_\ell$ {*Lemma 4.6*}
  // `SUBGRAPHS:`
  for each pair of graphs $G, H \in \mathcal{T}_\ell$ such that H is a subgraph of G **do**
    remove H from $\mathcal{T}_\ell$ {*Lemma 4.2*}
  $b_\ell \leftarrow 1$
  for each $G \in \mathcal{T}_\ell$ **do**
    $G' \leftarrow G +$ $(10 - \ell)$ isolated vertices
    for each permutation σ of the vertices of G' **do**
      test if (G', σ) is feasible using an LP solver
      if (G', σ) is *not* feasible **then**
        $b_\ell \leftarrow 0$
if $(b_6 \cdot b_7 \cdot b_8 \cdot b_9 \cdot b_{10}) == 1$ **then**
  $k = 10$
else
  $k < 10$

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.

Proposition 6.1 *The feasibility of any configuration on 10 vertices follows from Algorithm 1.*

The proof can be found in [CGS09].

Remark 6.1 *In step 12 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 6.2 *Observe that, due to Lemma 4.4, 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 15 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 Section 4 induce 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 an edge-simple circuit visiting $V(G)$ in the prescribed order. W.l.o.g. 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 an edge-simple 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 an edge simple circuit, and this feasibility can be quickly checked using an ILP solver (for instance, CPLEX).

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 [cod]), 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 [cod].

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 5.1 and 6.1 yields that

Theorem 6.1 *There exists an edge-simple circuit through any set of 10 ordered vertices of an infinite square grid.*

7 Concluding Remarks

In this article we showed that given any subset of 10 ordered vertices of an infinite square grid, there exists an edge-simple circuit visiting the 10 vertices in the prescribed order, and that the number 10 cannot be replaced by 11. To do so, we introduced a methodology 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 [cod]. 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 grid or torus, namely to consider an $(n_1 \times n_2)$-torus and to find the largest integer $k(n_1, n_2)$ such that given any subset of $k(n_1, n_2)$ ordered vertices in an $(n_1 \times n_2)$-torus, there exists an edge-simple circuit visiting the $k(n_1, n_2)$ vertices in the prescribed order.

Another direction is to consider another graphs instead of the square grid, like triangular and hexagonal grids and, more generally, general planar graphs or graphs of bounded treewidth.

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.

References

[ABHM99] 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*, 15:373–376, 1999.

[BH91] Sheng Bau and Derek Holton. Cycles containing 12 vertices in 3-connected cubic graphs. *J. Graph Theory*, 15(4):421–429, 1991.

[BL81] J.A. Bondy and L. Lovsz. Cycles through specified vertices of a graph. *Combinatorica*, 1:117–140, 1981.

[CGS09] D. Coudert, F. Giroire, and I. Sau. Circuit visiting 10 ordered vertices in infinite grids. Technical Report RR-6910, INRIA, 2009.

[cod] `http://www-sop.inria.fr/members/Frederic.Giroire/circuit`.

[Die05] Reinhard Diestel. *Graph Theory*. Springer-Verlag, 2005.

[Dir60] G. Dirac. In abstrakten Graphen vorhandene vollständige 4-Graphen u nd ihre Unterteilungen. *Math. Nachr.*, 22:61–85, 1960.

[EGL91] Y. Egawa, R. Glas, and S.C. Locke. Cycles and paths through specified vertices in k-connected graphs. *J. Comb. Theory Ser. B*, 52:20–29, 1991.

[EHL84] M.N. Ellingham, D.A. Holton, and C.H.C. Little. Cycles through ten vertices in 3-connected cubic graphs. *Combinatorica*, 4:265–273, 1984.

[FHW80] S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Journal of Theoretical Computer Science*, 10(2):111–121, 1980.

[FS92] A. Frank and A. Schrijver. Edge-Disjoint Circuits in Graphs on the Torus. *J. Comb. Theory Ser. B*, 55(1):9–17, 1992.

[GHHT04] F. Gringa, J. Harant, E. Hexel, and Zs. Tuzac. On short cycles through prescribed vertices of a graph. *Discrete Mathematics*, 286(1-2):67–74, 2004.

[HT82] R. Häggkvist and C. Thomassen. Circuits through specified edges. *Discrete Math.*, 41:29–34, 1982.

[JP09] Aubin Jarry and Stéphane Pérennes. Disjoint paths in symmetric digraphs. *Discrete Applied Mathematics*, 157(1):90–97, 2009.

[Kar75] R. M. Karp. On the complexity of combinatorial problems. *Networks*, 5:45–68, 1975.

[Kaw04] Ken Kawarabayashi. Cycles through a prescribed vertex set in n-connected graphs. *J. Comb. Theory Ser. B*, 90(2):315–323, 2004.

[KL82] A.K. Kelmans and M.V. Lomonosov. When m vertices in a k-connected graph cannot be walked round along a simple cycle. *Discrete Math.*, 38:317–322, 1982.

[Kv82] M.R. Kramer and J. van Leeuwen. Wire-routing is NP-complete. Technical Report RUU-CS-82-4, Department of Computer Science, University of Utrecht, 1982.

[Lov76] L. Lovász. On some connectivity properties of Eulerian graphs. *Acta Math. Acad. Sci. Hungar.*, 28:129–138, 1976.

[LR80] Andrea S. LaPaugh and Ronald L. Rivest. The subgraph homeomorphism problem. *J. Comput. Syst. Sci.*, 20(2):133–149, 1980.

[RS95] N. Robertson and P.D. Seymour. Graph minors XIII: the disjoint paths problem. *J. Comb. Theory Ser. B*, 63:65–110, 1995.

[RST94] Neil Robertson, Paul Seymour, and Robin Thomas. Quickly excluding a planar graph. *J. Comb. Theory Ser. B*, 62(2):323–348, 1994.