On the Exact Complexity of Hamiltonian Cycle and q-Colouring in Disk Graphs

Sándor Kisfaludi-Bak[1(✉)] and Tom C. van der Zanden[2]

[1] Department of Mathematics and Computer Science, TU Eindhoven, Eindhoven, The Netherlands
s.kisfaludi.bak@tue.nl

[2] Department of Computer Science, Utrecht University, Utrecht, The Netherlands
T.C.vanderZanden@uu.nl

Abstract. We study the exact complexity of the Hamiltonian Cycle and the q-Colouring problem in disk graphs. We show that the Hamiltonian Cycle problem can be solved in $2^{O(\sqrt{n})}$ on n-vertex disk graphs where the ratio of the largest and smallest disk radius is $O(1)$. We also show that this is optimal: assuming the Exponential Time Hypothesis, there is no $2^{o(\sqrt{n})}$-time algorithm for Hamiltonian Cycle, even on unit disk graphs. We give analogous results for graph colouring: under the Exponential Time Hypothesis, for any fixed q, q-Colouring does not admit a $2^{o(\sqrt{n})}$-time algorithm, even when restricted to unit disk graphs, and it is solvable in $2^{O(\sqrt{n})}$-time on disk graphs.

1 Introduction

Exact algorithms for NP-hard problems have received considerable attention in recent years. The goal of research in this area is to develop 'moderately exponential' algorithms and to prove matching lower bounds under complexity-theoretic assumptions. Most work in this direction concerns fundamental graph problems.

The *square-root phenomenon* is a well-documented occurrence among algorithms on planar graphs [13]. The term illustrates that many problems that have $2^{O(n)}$ algorithms on general graphs can be solved in $2^{O(\sqrt{n})}$ in planar graphs. Moreover, matching lower bounds can be found based on the Exponential Time Hypothesis, i.e., for most of these problems, there are no algorithms with running time $2^{o(n)}$ resp. $2^{o(\sqrt{n})}$, unless the Exponential Time Hypothesis fails.

An important question about the square-root phenomenon is whether we can generalize the results on planar graphs to larger graph classes. One possible direction is to extend to *disk graphs*: the vertices are disks in $\mathbb{R}^2$, and two vertices are adjacent if their disks intersect. Note that disk graphs where the interiors of the disks are disjoint are exactly the planar graphs [12]. *Unit disk graphs* are disk graphs where all radii are one; *bounded-ratio disk graphs* are disk graphs where the ratio of the largest and smallest radius is bounded by some constant.

The research of the first author was supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 024.002.003.

D. Fotakis et al. (Eds.): CIAC 2017, LNCS 10236, pp. 369–380, 2017.
DOI: 10.1007/978-3-319-57586-5_31

In this paper, we demonstrate the square-root phenomenon for the Hamiltonian Cycle problem in bounded-ratio disk graphs that are given by their geometric representations. Note that in planar graphs, the problem has a $2^{O(\sqrt{n})}$-time algorithm [13], and a matching $2^{\Omega(\sqrt{n})}$ lower bound conditional on the Exponential Time Hypothesis. The main obstacles for Hamiltonian Cycle in bounded-ratio disk graphs are the following.

- On the algorithmic side, the $2^{O(\sqrt{n})}$ running time often follows from the fact that planar graphs have treewidth $O(\sqrt{n})$ (see e.g. [4,6,15]). In our setting, bounded-ratio disk graphs are dense and may have unbounded treewidth.
- The lower bounds are based on reductions that planarize a graph by replacing each crossing of edges with a crossover gadget. Since there may be quadratically many crossings in a general graph, these reductions blow up an n-vertex graph to an n^2-vertex one, which results in the $2^{\Omega(\sqrt{n})}$ lower bound. In our setting, the NP-hardness of Hamiltonian Cycle was previously only known through its NP-hardness on grid graphs [8]. However, this reduction has a cubic blowup, giving only a $2^{\Omega(\sqrt[3]{n})}$ lower bound and – to our knowledge – it is an open problem whether this lower bound can be improved to match the best known ($2^{O(\sqrt{n})}$) algorithm [6].

The cubic time blowup of the reduction in [8] showing the hardness of Hamiltonian Cycle in grid graphs follows from two factors: the need to deal with crossings (introducing one factor n) and the need to replace long edges with some suitable 'path structure' (introducing another factor n). Compared to grid graphs, creating a reduction for disk graphs we have one major advantage: even though disk graphs have a structure somewhat similar to planar graphs, they can be (locally) non-planar and the Hamiltonian cycle in a solution can cross itself. Even so, our reduction still uses crossover gadgets and has to replace edges with path structures.

A key technique of our reduction is that in replacing long edges with some other structure, we need to ensure that all the vertices of this structure can be visited even if the edge is not used in the Hamiltonian cycle. This can be achieved with a $2 \times n$ grid (a *snake*), which can either be traversed in a zigzag manner (corresponding to using the edge in the cycle) or traversed going back and forth (corresponding to not using the edge). Our snakes are almost identical to the ones proposed by Itai et al. [8]. Unfortunately, it does not appear to be possible to create a crossover gadget for two snakes. To overcome this, we modify the reduction to ensure that some edges will certainly be included in the solution (which we can thus replace with a simple path rather than a snake) and build our reductions such that we only have crossings between simple paths and between simple paths and snakes (for which we *can* build crossover gadgets).

To complement our results for Hamiltonian Cycle, we also show that the same upper and lower bounds hold for q-colouring on disk graphs in the case where q is a constant. The algorithm follows from the observation that q-colorable graphs do not have large cliques, and a separator theorem due to Miller et al. [14]; the lower bound uses an adaptation of a reduction due to Gräf et al. [7].

Some proofs are omitted from this extended abstract.

2 Algorithm for Hamiltonian Cycle in Bounded-Ratio Disk Graphs

In this section, we show that the Hamiltonian Cycle problem can be solved in $2^{O(\sqrt{n})}$ time on bounded-ratio disk graphs. Our algorithm uses techniques due to Ito and Kadoshita [9], who show that Hamiltonian Cycle can be solved in $2^{O(\alpha)}n^{O(1)}$ time on unit disk graphs, where α is the area of a bounding square of the set of disks.

Theorem 1. *There exists a $2^{O(\sqrt{n})}$-time algorithm for Hamiltonian Cycle on bounded-ratio disk graphs (where the graphs are given by their geometric representation).*

Lemma 1. *Given a disk graph of ratio $\beta = O(1)$ with its representation, there are values $\gamma = \gamma(\beta)$ and $\Delta = \Delta(\beta)$ such that if we tessellate the plane using squares of diameter γ, the vertices in each tile induce a clique and the vertices in any given square have neighbours in at most Δ distinct other squares.*

Proof. Ito and Kadoshita [9] prove this lemma for unit disk graphs, where $\gamma = 1$ and $\Delta = 18$. The proof generalizes to bounded-ratio disk graphs. □

Given a bounded-ratio disk graph G, the lemma gives a *clique partition* $Q_1, \ldots, Q_r$ of G, that is, a partition of the vertices of G into cliques, such that the vertices of each clique have neighbours in at most $\Delta = O(1)$ other cliques.

Given a graph G and sets $A, B \subseteq V(G)$, we let $E(A, B)$ denote the set of edges between a vertex in A and a vertex in B. Using the notion of *canonical Hamiltonian cycle* (which we do not need to consider), Ito and Kadoshita [9] then prove the following lemma:

Lemma 2 (Ito and Kadoshita [9]). *Let G have clique partition $Q_1, \ldots, Q_r$ defined by a tessellation as in Lemma 1. Then for each $i \neq j$, we can remove all but $O(\Delta^2)$ edges of $E(Q_i, Q_j)$ to obtain G', such that G' has a Hamiltonian cycle if and only if G has a Hamiltonian cycle.*

If G' is connected, removing the vertices from each clique of the clique partition in G' that do not have an edge to a vertex of some other clique in the partition preserves the Hamiltonicity of G'. We thus obtain the reduced graph G'', which contains at most $O(\Delta^3)$ vertices per tile.

Lemma 3. *The reduced graph has treewidth $O(\sqrt{n})$. A tree decomposition of treewidth $O(\sqrt{n})$ can be found in polynomial time.*

Proof. Alber and Fiala [1] show that unit disk graphs have balanced separators where the disks of the vertices in the separator cover an area of at most $O(\sqrt{n})$. This also holds for bounded-ratio disk graphs, as we can consider the supergraph obtained by making the radius of each disk equal to the largest radius. Since in the reduced graph, each tile contains at most a constant number of points, this gives a balanced separator of size $O(\sqrt{n})$ (in terms of vertices). These separators in turn imply that the reduced graph has treewidth $O(\sqrt{n})$. Using these separators we can also build a tree decomposition of width $O(\sqrt{n})$ in polynomial time [16]. Note that the hidden constant depends on β. □

Theorem 2 (Bodlaender et al. [2], Cygan et al. [5]). *Given a graph with a tree decomposition of width w, there exists an algorithm solving Hamiltonian Cycle in $2^{O(w)}n^{O(1)}$ time.*

Applying this algorithm to the reduced graph of Lemma 3 finishes the proof of Theorem 1. □

The techniques described in this section can also be used to solve Longest Path and Exact Path (which respectively are the problems of finding simple path of maximum length and finding a path between two specified vertices (u, v) of given length k):

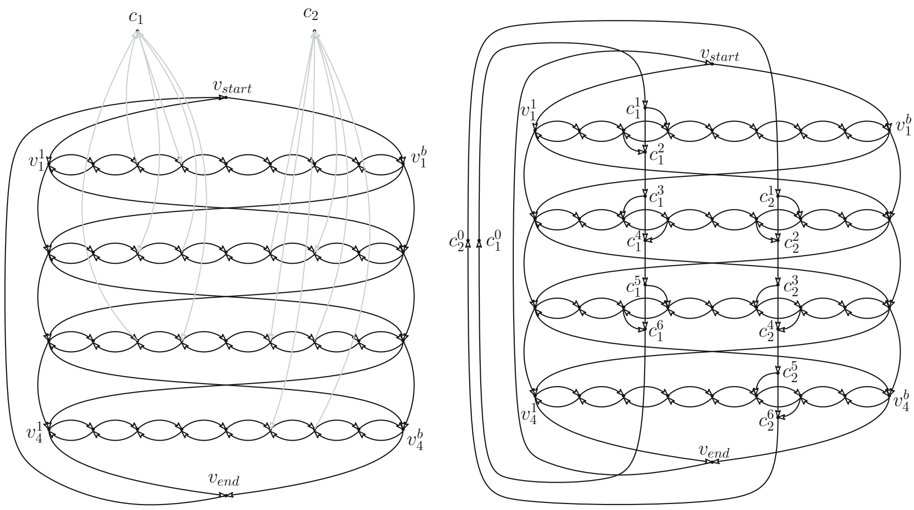
Theorem 3. *There exists a $2^{O(\sqrt{n})}$-time algorithm for Longest Path and Exact Path on bounded-ratio disk graphs.*

Proof. Lemma 2 also holds for Longest Path and Exact Path. However, the subsequent step of removing vertices from the cliques no longer works: the information of how many vertices can be visited within each clique is essential. Instead, from each clique of the partition, we remove every vertex that does not have an edge to a vertex in some other clique of the partition. The removed vertices are then replaced by a path of the same number of vertices, and every vertex of the path is made adjacent to every (remaining) vertex of the clique. This preserves the longest (or exact) path, while only increasing the treewidth by a constant (compared to the reduced graph from Lemma 3). Bodlaender et al. [2] and Cygan et al. [5] also give algorithms for Longest Path and Exact Path parameterized by treewidth, similar to those of Theorem 2. Thus we obtain $2^{O(\sqrt{n})}$-time algorithms for Longest Path and Exact Path, by modifying the graph such that it has treewidth $O(\sqrt{n})$ and then applying one of these algorithms. □

3 Lower Bound for Hamiltonian Cycle in Unit Disk Graphs

In this section, we give a tight lower bound for the running time of a Hamiltonian cycle algorithm in UDGs, assuming the Exponential Time Hypothesis. We use a reduction from 3-SAT.

We begin with a well-known reduction from 3-SAT to directed Hamiltonian cycle [11], and modify it significantly. We introduce the construction briefly; see Fig. 1 for an example of the construction with the formula $(x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$. Let n be the number of variables, and m be the number of clauses. For each variable x_i we introduce the vertices $v_i^1, \ldots v_i^b$, where $b = 3m + 3$. On these vertices we add a double chain (a directed path through the vertices and back); this double chain is the *row* of this variable. The Hamiltonian cycle can traverse this row left to right or right to left, which will indicate the truth setting of this variable. We add edges from the beginning and end of a variable's row to the beginning and end of the next variable's row, and we add a starting and ending point v_{start} and v_{end}, the arc (v_{end}, v_{start}), and arcs $(v_{start}, v_1^1), (v_{start}, v_1^b)$, $(v_n^1, v_{end}), (v_n^b, v_{end})$. In order to check the clauses, we add a vertex c_j for each clause $j = 1, \ldots, m$. We connect the vertices v_i^{3j} and v_i^{3j+1} to clause j if a literal of x_i is present in the j-th clause. By orienting this arc pair correctly (depending on the sign of the literal), we make it possible for the Hamiltonian cycle to make a detour to c_j while traversing the variable's row in the direction (left or right) corresponding to the sign of the literal. For more details about this construction we refer the reader to the write-up in [11].



(a) Original Construction (b) Modified Construction

Fig. 1. (a) The construction for $(x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$. (b) Modified construction with a directed cycle for each clause.

Our Construction. We replace the clause vertices by a different gadget: for each clause c_i, we introduce a directed cycle containing seven vertices, $c_i^0, \ldots, c_i^6$ (see Fig. 1b). If the first literal is the j-th variable, then we add the arcs $v_j^{3i} c_i^2$ and $c_i^1 v_j^{3i+1}$; if the literal is negated, we add $v_j^{3i+1} c_i^2$ and $c_i^1 v_j^{3i}$. Similarly, we add entry and exit arcs for the second and third literals at c_i^3, c_i^4 and c_i^5, c_i^6. This modified graph has a directed Hamiltonian cycle if and only if the original formula is satisfiable.

Next, we reduce to undirected Hamiltonian cycle. (From this point onward, we use the abbreviation HC for Hamiltonian cycle.) To do this, we start by replacing each vertex u of the construction with three vertices on a path: u^-, u^0 and u^+. An arc previously going from u to w is represented in the new graph by the edge $u^+ w^-$. This reduction from directed to undirected HC is already present in Karp's famous 1972 paper [10]. It is again routine to prove that the graph resulting from this construction has a HC if and only if the original formula is satisfiable. We denote the undirected graph that was obtained in this way by G.

We consider a specific drawing of G depicted on Fig. 2 of the resulting graph; we plan to emulate its properties in a unit disk graph. Intuitively, we would like to replace the edges of this graph by paths — this can be done by using unit disks that induce a path, making sure that the new graph has a HC if and only if the old graph has a HC. We call the set of disks used to represent an edge in such a way a *thread*. The difficulty stems from edges that are not used in the HC: substituting such edges with threads is not allowed. Essentially, we can only use threads if it is guaranteed by the construction that every HC has to pass through. If this cannot be guaranteed, we use *snakes*, which are constructions that allow the HC to either 'use' the edge uv, or to make a detour from one of the endpoints into the gadget, visiting every vertex inside.

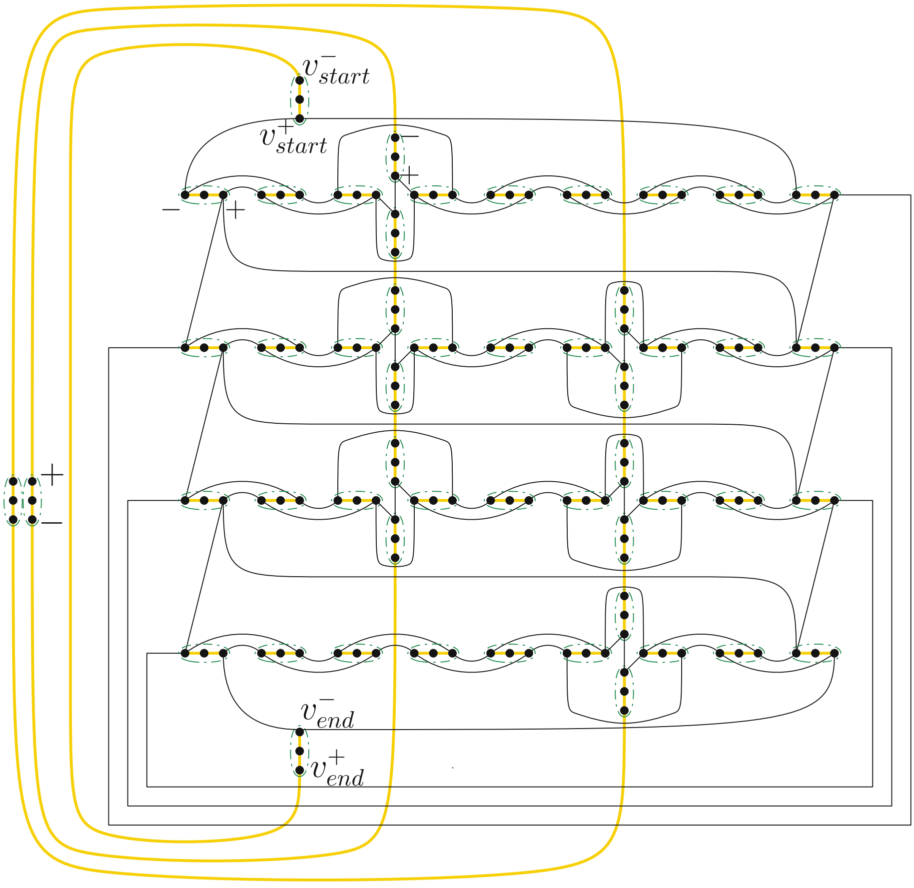


Fig. 2. A drawing of the undirected version.

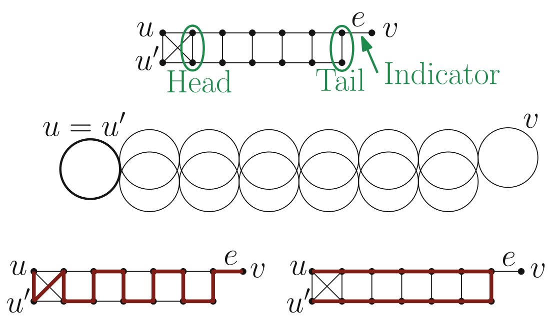


Fig. 3. A snake and a corresponding unit disk realization. Below: simulating a HC that passes or avoids edge uv.

Lemma 4. *The maximum vertex degree in G is four, and vertices of degree four induce a subgraph in which the maximum degree is two.*

Proof. The upper bound on the vertex degree follows from the fact that the original directed graph has maximum indegree and maximum outdegree three.

For the second statement, notice that vertices of degree four are either in- or out-vertices inside the row of a variable x_i, i.e., they are of the form $(v_i^j)^+$ or $(v_i^j)^-$. Moreover, notice that every vertex of degree four has a neighbour of degree two – the middle vertex $(v_i^j)^0$. Thus it is sufficient to show that for any degree four vertex v there is an additional neighbour of degree at most three.

The proof is for the case $(v_i^j)^+$. If $j = 1$ or $j = b = 3m + 3$, then $(v_i^j)^+$ has $(v_i^2)^-$ or $(v_i^{b-1})^-$ as a neighbour, and these are vertices of degree three: in the directed construction, the corresponding vertices had in- and outdegree two, because they are vertices of the form v_i^j, $j \equiv 2 \pmod 3$. If $1 < j < n$, then the vertex has a neighbour in the clause loop, where the maximum degree is three. □

Representing Edges with Snakes. The snake is simply a $2 \times k$ grid graph for some $k \in \mathbb{N}$, with an extra disk at the *head* of the snake. In Fig. 3 we illustrate how a snake replacing an edge uv works. We need to add a disk u' which has the same neighbourhood as u - (this can be done by taking an identical or slightly perturbed copy of the disk of u). At the other end of the snake (at the *tail*) no such operation on v is required. Through the snake a HC can simulate passing the edge uv and it can also make a detour from u that covers all inner vertices if uv is not in the original HC. We define the indicator edge of the snake to be the edge connecting the snake to v, indicated by e in Fig. 3. It is easy to verify that if e is not used then we must detour (corresponding to avoiding uv in the HC), otherwise we must zigzag (corresponding to using uv).

Crossing Gadgets. Notice that in all the crossings in Fig. 2, exactly one of the crossing edges is a thick golden edge. These edges share the property that at least one of their endpoints has degree two, thus any Hamiltonian cycle of the

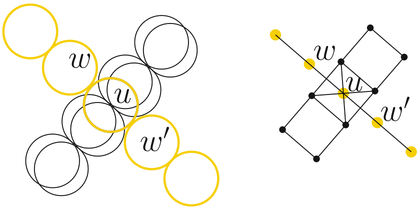


Fig. 4. Crossing a snake and a thread.

construction must pass through the golden edges. Therefore, we can replace the golden edges by threads; all other edges can be replaced by snakes. We have a crossing gadget for thread–snake crossings that we describe below.

The crossing gadget is depicted in Fig. 4. A Hamiltonian cycle passing through the snake in any way cannot enter the edges spanned by the thread: it can only enter at vertex u, and continue on one of the outgoing thread edges; that would render one of the points w and w' unreachable to the HC.

We note that snakes and threads can be used to represent bending edges, and a bend introduces only constant overhead; furthermore, a vertex can be the starting or ending point of up to five internally disjoint snakes or threads; since the maximum degree of G is four by Lemma 4, this threshold is not reached. We place snakes so that vertices of degree four have at most one connecting snake head – this can be done since the vertices of degree four in G span a collection of vertex disjoint paths and cycles by Lemma 4.

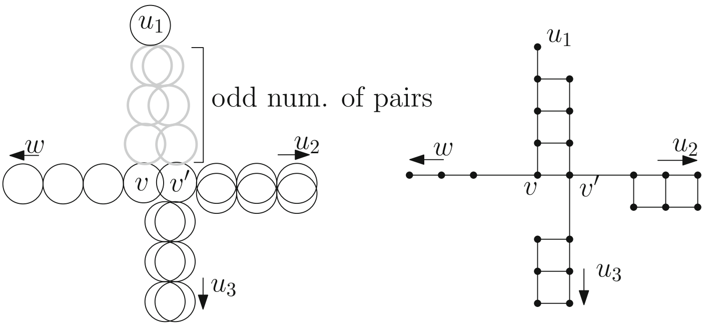


Fig. 5. Construction for degree four vertices.

Modifying the Neighbourhood of Vertices of Degree Four with a Snake Head. All degree four vertices have a neighbour of degree two (the in- and outvertices u^+ and u^- are connected to the degree two vertex u^0), thus the connecting edge is always used by all HCs in G. Let v be a degree four vertex with neighbours w, u_1, u_2 and u_3; let w be the neighbour of degree 2, and let u_1 be the neighbour whose snake head is at v. If the vertex has a connecting snake head, then we modify the neighbourhood of v so that v' is not a duplication of the disk representing v. Connect v to the tail of the w_1v snake, and connect it to the vw_2 snake together with v; the two remaining snake tails are only connected to v' (see Fig. 5 for the case when wv and u_1v are consecutive edges of v in the drawing). We can also ensure that the length of the vw_2 snake is odd, i.e., the snake is a $2 \times 2k+1$ grid for some $k \in \mathbb{N}$, not including v and v'. This is required to make sure that the HC must pass both v and v' when the snake is used.

Finally, if wv and u_1v are non-consecutive edges around v in the drawing of G (Fig. 2), then we can change the drawing by introducing a new crossing between vw and vu_2 to make wv and u_1v consecutive around v. This requires a new snake-thread crossing, for which we can use our crossing gadget.

Fig. 6. Adding an extra thread-snake crossing around degree four vertices might be necessary. Here we added an extra crossing around the brown vertex w^- by changing a thread incident to w^- (in violet). (Color figure online)

The Final Construction. We begin by recreating a drawing of G in the plane with integer coordinates for all vertices, similar to the one seen in Fig. 2. This fits in a rectangle of size $O(n+m) \times O(n+m)$: there are $O(n)$ variable rows of length $O(m)$, and they require $O(1)$ vertical space each; together with the n long edges, we can fit these in $O(n+m)$ horizontal and $O(n)$ vertical space. The loop edges require $O(m)$ more vertical and horizontal space.

We apply a large constant scaling to make enough room for gadgets. Next, we define an orientation on the snakes so that degree four vertices have at most one snake head. Such an orientation exists due to Lemma 4. We also introduce extra crossings around degree four vertices when needed to ensure that the thread edge and the snake head are neighbours (see the change around w^- in Fig. 6). Finally, we exchange the golden edges with threads and the snake edges with snakes (Fig. 7); if our initial constant scaling was large enough, we have enough space to bend threads and snakes, without introducing intersections between independent snakes and threads. For disks representing vertices, for every incoming snake we introduce slightly perturbed disks (according to the original definition of snakes). These extra disks are indicated by a green number in Fig. 7.

Lemma 5. *Given an initial undirected graph G corresponding to a 3-CNF formula of n variables and m clauses, the unit disk graph G' constructed above is computable in time polynomial in $n+m$, has $O((m+n)^2)$ vertices and it is equivalent to G in the sense that G' has a HC if and only if G has a HC.*

Proof. First, we show that if G' has a HC then G has a HC. (The other implication is trivial.) Let H' be a HC in G'. In each thread we designate an arbitrary

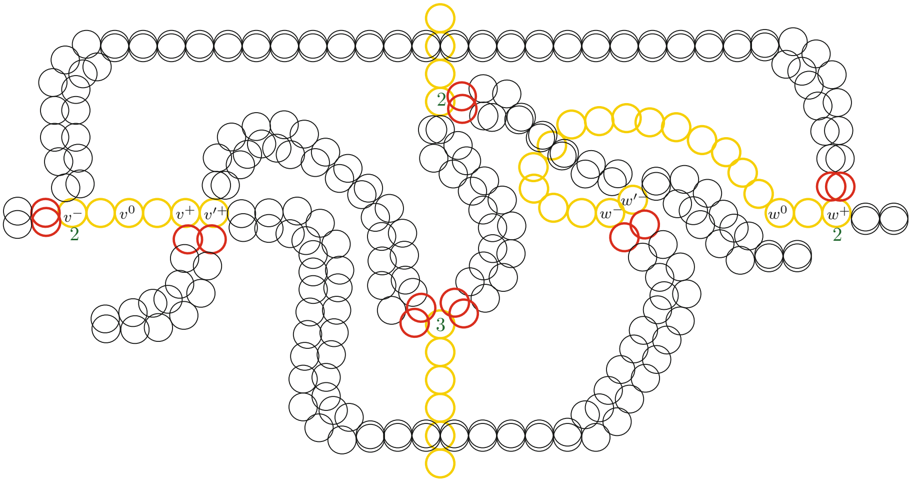


Fig. 7. The part of the final construction corresponding to Fig. 6. Snake heads are represented by two red disks, plus multiplicity of the end vertex when needed. Golden disks correspond to threads. (Color figure online)

inner edge as indicator. Mark an edge in G if the indicator edge of the corresponding thread or snake is contained in H'. We claim that the set of marked edges (denoted by H) is a HC in G. Observe that a cut $C \subseteq E(G)$ corresponds to a cut of the same size in G': the indicator edges corresponding to the threads or snakes of the edges in C define a cut of G'. Consequently, for any cut C the number of H-edges contained in it is an even, positive number, since the corresponding cut C' is crossed by H' an even, positive number of times. It follows that H is a spanning connected Eulerian subgraph.

It remains to show that the maximum degree in H is two; since the maximum degree of G is four, it is sufficient to show that any vertex v of degree four has degree two in H. If v has no snake heads, then this follows from the fact that the disk corresponding to v has four independent neighbours in G'.

Let v be a vertex of degree four with a snake head. We denote by $S(x, y)$ the snake from x to y, with the head at x. We use the notation w, u_1, u_2, u_3 for the neighbours of v, where $\deg(w) = 2$ and $S(v, u_1)$ is the snake whose head is at v (see Fig. 5). Since there is a thread between w and v, the edge (w, v) is marked. If H' uses $S(v, u_1)$, then by the odd length of the snake and our construction, vv' is an edge of H' — this can be verified by stepping back through $S(v, u_1)$ from the indicator edge at u_1. So in this case, H' must detour on both $S(u_2, v')$ and $S(u_3, v')$. Otherwise (if $S(v, u_1)$ is only a detour in H'), then one of the neighbours of v' in H' is inside $S(v, u_1)$, so H' can use only one of $S(u_2, v')$ and $S(u_3, v')$. Thus, the degree of v in H is two in both cases.

The construction can be created in polynomial time from an initial graph G. It is placed in a rectangle of size $O(n + m) \times O(n + m)$; since every point in this rectangle is covered by at most four disks (which can occur if there are two snake heads at a degree three vertex), it follows that the number of disks used is $O((n+m)^2)$. □

Theorem 4. *There is no $2^{o(\sqrt{n})}$ algorithm for Hamiltonian cycle in unit disk graphs, unless the ETH fails.*

Proof. Suppose that the initial formula has $\hat{m}$ clauses and $\hat{n}$ variables. Without loss of generality, suppose that $\hat{m} = \Theta(\hat{n})$ (see the Sparsification Lemma in [3]). The graph G can be created in polynomial time starting from our formula, and by Lemma 5 we can create G' in polynomial time from G. The resulting unit disk graph G' has a HC if and only if the original formula is satisfiable. Since the resulting UDG has $O(\hat{n}^2)$ vertices, a $2^{o(\sqrt{n})} = 2^{o(\sqrt{\hat{n}^2})}$ algorithm would mean that we could decide the satisfiability of the formula in $2^{o(\hat{n})}$ time, which contradicts the Exponential Time Hypothesis. □

4 Colouring Disk Graphs

To complement our results on Hamiltonian Cycle, we show that the square root phenomenon also holds for q-colouring on disk graphs when q is a constant.

Theorem 5. *(a) For any constant q, there is an algorithm running in time $O(2^{O(\sqrt{n})})$ that solves the q-colouring problem on disk graphs. (b) There is no $2^{o(\sqrt{n})}$ algorithm for q-colouring in unit disk graphs for any constant $q \geq 3$, unless the ETH fails.*

5 Conclusions

We have shown that the HC problem and q-colouring both have $2^{O(\sqrt{n})}$ algorithms in bounded-ratio disk graphs, and matching lower bounds $2^{\Omega(\sqrt{n})}$ if ETH holds. We have also seen that in case of the colouring problem, the same result applies in general disk graphs.

Some preliminary work shows that it should be possible to get a $2^{\Omega(\sqrt{n})}$ lower bound for HC in the more restricted case of grid graphs, although the proof and the gadgets used will be more complicated.

A major remaining open problem is to find a $2^{O(\sqrt{n})}$ algorithm for HC in disk graphs. Finally, we remind the reader that reducing the coefficient of $\sqrt{n}$ in the exponents of these running times is also a worthwhile effort; in the case of Hamiltonian Cycle on general graphs, a steady wave of improvements yielded impressive results; can the community achieve something similar for these square-root type algorithms?

Acknowledgements. This work was initiated at the Lorentz Center workshop 'Fixed-Parameter Computational Geometry'. We are grateful to Hans L. Bodlaender and Mark de Berg for discussions and their help with improving this paper.

References

1. Alber, J., Fiala, J.: Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. J. Algorithms **52**(2), 134–151 (2004). http://dx.doi.org/10.1016/j.jalgor.2003.10.001
2. Bodlaender, H.L., Cygan, M., Kratsch, S., Nederlof, J.: Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. Inf. Comput. **243**, 86–111 (2015)
3. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Heidelberg (2015). doi:10.1007/978-3-319-21275-3
4. Cygan, M., Kratsch, S., Nederlof, J.: Fast hamiltonicity checking via bases of perfect matchings. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, STOC 2013, pp. 301–310. ACM, New York (2013). http://doi.acm.org/10.1145/2488608.2488646
5. Cygan, M., Nederlof, J., Pilipczuk, M., Pilipczuk, M., van Rooij, J.M.M., Wojtaszczyk, J.O.: Solving connectivity problems parameterized by treewidth in single exponential time. In: Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Washington, DC, USA, pp. 150–159 (2011). http://dx.doi.org/10.1109/FOCS.2011.23
6. Deineko, V.G., Klinz, B., Woeginger, G.J.: Exact algorithms for the hamiltonian cycle problem in planar graphs. Oper. Res. Lett. **34**(3), 269–274 (2006). http://dx.doi.org/10.1016/j.orl.2005.04.013
7. Gräf, A., Stumpf, M., Weißenfels, G.: On coloring unit disk graphs. Algorithmica **20**(3), 277–293 (1998). http://dx.doi.org/10.1007/PL00009196
8. Itai, A., Papadimitriou, C.H., Szwarcfiter, J.L.: Hamilton paths in grid graphs. SIAM J. Comput. **11**(4), 676–686 (1982). http://dx.doi.org/10.1137/0211056
9. Ito, H., Kadoshita, M.: Tractability and intractability of problems on unit disk graphs parameterized by domain area. In: Proceedings of the 9th International Symposium on Operations Research and Its Applications (ISORA 2010), pp. 120–127. Citeseer (2010)
10. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W., Bohlinger, J.D. (eds.) Proceedings of a Symposium on the Complexity of Computer Computations. The IBM Research Symposia Series, pp. 85–103. Springer, Heidelberg (1972). doi:10.1007/978-1-4684-2001-2_9
11. Kleinberg, J.M., Tardos, É.: Algorithm Design. Addison-Wesley, Boston (2006)
12. Koebe, P.: Kontaktprobleme der konformen Abbildung. Hirzel (1936)
13. Marx, D.: The square root phenomenon in planar graphs. In: Automata, Languages, and Programming - 40th International Colloquium, p. 28 (2013)
14. Miller, G.L., Teng, S., Thurston, W.P., Vavasis, S.A.: Separators for sphere-packings and nearest neighbor graphs. J. ACM **44**(1), 1–29 (1997). http://doi.acm.org/10.1145/256292.256294
15. Pino, W.J.A., Bodlaender, H.L., van Rooij, J.M.M.: Cut and count and representative sets on branch decompositions. In: Proceedings of IPEC (2016, to appear)
16. Reed, B.: Tree width and tangles: a new connectivity measure and some applications. Surv. Comb. **241**, 87–162 (1997)