Dynamic Complexity of Directed Reachability and Other Problems

Samir Datta[1], William Hesse[2], and Raghav Kulkarni[3]

[1] Chennai Mathematical Institute, India
sdatta@cmi.ac.in
[2] University of Massachusetts & Google Inc
whessedk@gmail.com
[3] Center for Quantum Technologies, Singapore
kulraghav@gmail.com

Abstract. We report progress on dynamic complexity of well-known graph problems such as reachability and matching. In this model, edges are dynamically added and deleted and we measure the complexity of each update/query. We propose polynomial-size data structures for such updates for several related problems. The updates are in very low level complexity classes such as quantifier-free first order formulas, $\mathsf{AC}^0[2]$, TC^0. In particular, we show the following problems are in the indicated classes:
(a) maximum matching in non-uniform DynTC^0;
(b) digraph reachability in non-uniform $\mathsf{DynAC}^0[2]$;
(c) embedded planar digraph reachability in DynFO(= uniform DynAC^0).

Notably, the part (c) of our results yields the first non-trivial class of graphs where reachability can be maintained by first-order updates; it is a long-standing open question whether the same holds for general graphs. For (a) we show that the technique in [7] can in fact be generalized using [9] and [8] to maintain the determinant of a matrix in DynTC^0. For (b) we extend this technique with the help of two more ingredients namely isolation [1,13] and truncated approximation using rational polynomials. In fact, our proof yields $\mathsf{DynAC}^0[p]$ bound for any prime $p > 1$. For (c) we exploit the duality between cuts and cycles in planar graphs to maintain the number of crossings between carefully chosen primal and dual paths, using several new structural lemmas.

1 Introduction

Conventional complexity assumes that the entire input is available initially and it does not change with time. An alternative to this static view is the dynamic model, where the input evolves with time, such as where edges are added or deleted to a graph. We can measure the efficiency of an algorithm in such a model in terms of the (static) complexity of its updates. The particular complexity classes we focus upon in this paper are some of the simplest ones possible: classes of Boolean functions computable by bounded depth circuits. Yet polynomial-size

J. Esparza et al. (Eds.): ICALP 2014, Part I, LNCS 8572, pp. 356–367, 2014.

data structures with updates in these classes are powerful enough to solve fairly complex problems, as we see in this work. We start with a brief description of this model.

1.1 The Model of Dynamic Complexity

In the *dynamic* (graph) model we start with an empty graph on a fixed-size set of vertices. The graph evolves by the addition/deletion of a single edge in every time step and an algorithm maintains the graph and polynomially many bits of auxiliary data, which allow a property such as reachablity to be queried efficiently. The dynamic complexity of the algorithm is the static complexity of each update step. If a polynomial-size data structure can be updated in a static class $\mathcal{C}$, the dynamic problem is said to belong to $\mathsf{Dyn}\mathcal{C}$. In this paper, $\mathcal{C}$ is often a complexity class defined in terms of bounded depth circuits[1] such as AC^0, $\mathsf{AC}^0[2]$, TC^0, where $\mathsf{AC}^0$is the class of polynomial size constant depth circuits with AND and OR gates of unbounded fan-in; $\mathsf{AC}^0[2]$circuits may additionally have PARITY (sum modulo 2) gates; TC^0circuits may additionally have MAJORITY gates. We encourage the reader to refer to any textbook (e.g. Vollmer [17]) for precise definitions of the standard circuit complexity classes. The archetypal example of a dynamic problem is maintaining reachability ("is there a directed path from s to t"), in a digraph. This problem is known to be maintainable in the class DynTC^0- a class where edge insertions and deletions can be maintained using TC^0circuits.

1.2 Historical Background

Dynamic complexity was introduced by Immerman-Patnaik [14] who defined the class DynFO(where the static update class is Dlogtime-uniform AC^0) and proved that problems such as undirected connectivity and minimum spanning tree are in DynFO. DAG reachability was known to be in this class even before it was formally defined [4]. Hesse proved that directed reachability is in the slightly larger (uniform) class DynTC^0. Recently it has been shown that distance computation in undirected graphs can be maintained in DynFO[5] (see also [10] where 3-connected planar isomorphism is also shown to be maintainable in DynFO^+, a closely related class). The big open question in the area is whether it is possible to maintain directed reachability in DynFO. For surveys on dynamic complexity please see [6,16].

1.3 Our Results and Techniques

We summarize our results below:

Theorem 1. *The determinant of a matrix can be maintained in* DynTC^0.

[1] We will have occasion to refer to both the non-uniform and (dlogtime-)uniform versions of these circuit classes and we adopt the convention that, whenever unspecified, we mean the uniform version.

Applying the reduction from rank to determinant by Mulmuley [12] and further applying the Isolation Lemma [13,1], we obtain:

Corollary 1. *(a) The rank of a matrix can be maintained in* DynTC^0*.*
(b) Maximum Matching in a graph is in non-uniform DynTC^0*.*

The complete proof will be presented in the full version of the paper. Next we show the following on reachability.

Theorem 2. *Reachability in directed graphs is in non-uniform* $\mathsf{DynAC}^0[2]$*.*

A generalization to the weighted case with possibly negative weights yields:

Corollary 2. *(a) Maintaining walks of weight at most W is in* $\mathsf{DynAC}^0[2]$*; (b) Shortest Path in a graph without negative cycle or detecting if the negative cycle exists can be maintained in* $\mathsf{DynAC}^0[2]$*; (c) Perfect Matching in grid graphs is in uniform* $\mathsf{DynAC}^0[2]$*.*

Again, we postpone the proof to the full version of the paper.

Theorem 3. *Reachability in directed embedded planar graphs is in* DynFO*.*

The main building block for Theorem 1 and Theorem 2 is a technique introduced by Hesse [7] for maintaining the reachability in directed graphs. This technique relies on a result of Hesse, Allender, and Barrington [8] showing that multiplying univariate polynomials is in TC^0. We show that this reahcability technique can in fact be generalized using the result of Mahajan and Vinay [9] on clow-sequences to maintain the determinant of a matrix in DynTC^0. This suffices for Theorem 1. For Theorem 2 we extend this technique further with the help of two more ingredients namely the Isolation Lemma (more specifically its usage in Allender, Reinhardt and Zhou [1]) and a simple but subtle use of rational polynomials in truncated form to achieve $\mathsf{DynAC}^0[2]$bound. In fact, our proof yields $\mathsf{DynAC}^0[p]$bound for the same for any prime $p > 1$, putting it in the intersection of these $\mathsf{AC}^0[p]$classes. We note that no total function outside $\mathsf{AC}^0$is known to lie in the intersection of these classes. Thus our result gets tantalizingly close to the non-uniform AC^0bound. The uniform AC^0bound would resolve the long-standing conjecture that directed reachability is in DynFO.

For Theorem 3 we exploit the duality between cuts and cycles in planar graphs to maintain information about all oriented cycles in the dual graph, giving cut sets in the original graph. There are interesting technical complexities to overcome, requiring some extra bookkeeping in our data structures. There seem to be no fundamental problems with extending the algorithm from embedded planar graphs to planar graphs. This generates hope for placing general reachability in DynFO.

1.4 Organization

The organization of this paper is as follows: Section 2 contains the preliminaries. Section 3 contains the proof of Theorem 1. Section 4 contains the proof of Theorem 2. Section 5 contains the proof of Theorem 3.

2 Preliminaries

2.1 Non-uniform Version of Isolation

Mulmuley, Vazirani, and Vazirani [13] introduced this simple but powerful lemma:

Lemma 1 (Isolation Lemma). *Given a non-empty $\mathcal{F} \subseteq 2^{[m]}$, if one assigns for each $i \in [m]$, $w_i \in [2m]$ uniformly at random then with probability at least half, the minimum weight subset of $\mathcal{F}$ is unique; where the weight of a subset S is $\sum_{i \in S} w_i$.*

A surprising part of the above lemma is that no structure is assumed on $\mathcal{F}$. Allender, Reinhardt, and Zhou observe that the above lemma can be derandomized with the help of non-uniformity.

Lemma 2 (Non-uniform Version of Isolation Lemma). *There exist m^2 weight assignments $W^{(i)} = (w_1^{(i)}, \ldots, w_m^{(i)})$ (for $i \in [m^2]$) such that: for any non-empty $\mathcal{F} \subseteq 2^{[m]}$ there exists an $i \in [m^2]$ such that the minimum weight subset of $\mathcal{F}$ with respect to $W^{(i)}$ is unique.*

2.2 Clow-sequences and Determinant

Mahajan-Vinay [9] introduced the concept of a *clow-sequence* for an adjacency matrix which extends the notion of a cycle-cover.

Definition 1. *A* clow *is a closed walk starting and ending at the same vertex. The smallest vertex in a clow is called in its* head. *A* clow sequence*, $\mathcal{W}$, is a sequence of clows with some constraints: (a) the heads of the clows in the sequence are in strictly increasing order, and, (b) the number of edges (with multiplicity) in a clow-sequence is exactly n, the size of the matrix. Define $sgn(\mathcal{W})$ to be $(-1)^{n \,+\, \#clows \text{ in } \mathcal{W}}$ and weight $w(\mathcal{W})$ to be the product of its edge weights.*

Armed with the above definition we approach the main theorem of [9]:

Theorem 4 (Mahajan-Vinay[9]).

$$det(M) = \sum_{\mathcal{W}:\ clow\ sequence\ in\ M} sgn(\mathcal{W}) w(\mathcal{W})$$

This nice result is based on the observation that all clow-sequences which are not cycle-covers cancel out in pairs in the signed sum above, leaving behind the signed sum of cycle covers which is the determinant.

2.3 Polynomial Arithmetic and Hesse's Approach

Hesse [7] showed that:

Theorem 5 (Hesse[7]). *Reachability in directed graphs can be maintained in* DynTC^0.

The main idea is to maintain the counts of all s, t-walks parameterized on their lengths for each pair of vertices s, t. These are maintained as integer polynomials and [7] showed that updating them only requires addition and multiplication of polynomials, and finding powers of polynomials. For this the following theorem from Hesse, Allender and Barrington [8] is crucial:

Theorem 6 (Hesse-Allender-Barrington[8]). *Iterated product of integer polynomials is in uniform* TC^0.

We also note that the above theorem easily extends to polynomials in two (more generally any constant number of) variables. Arithmetic on polynomials over the finite field $\mathbb{F}_p$ can be done with even simpler circuits using modulo p gates:

Fact 1 (Folklore). *Iterated polynomial addition and multiplication of two polynomials over* $\mathbb{F}_p$ *is in* $\mathsf{AC}^0[p]$ *for prime* $p > 1$.

2.4 Planarity, Uniform Isolation, Flows and Cycles

The non-uniform edge weights of Lemma 2 can be replaced by a uniformly computed set of weights on certain families of planar graphs. In particular Bourke, Tiwari, and Vinodchandran show that for subgraphs of a planar grid, one can find weights so that minimum weight paths and perfect matchings are unique. This can be extended to perfect matchings in bipartite grid graphs [3].

Lemma 3 (Deterministic Isolation in Grid Graphs, [2,3]). *One can assign* $O(\log n)$*-bit long weights in uniform* AC^0 *to the edges of* $n \times n$ *grid such that in any subgraph of the grid (a) the minimum weight* s*-*t *path is unique; (b) minimum weight perfect matching is unique.*

We use this fact to obtain uniform circuits for dynamic updates in grid graphs.

Further we use planarity in the context of flows. Miller and Naor [11] relate the existence of flow in a planar graph with multiple sources and multiple sinks to the non-existence of negative weight cycle in the dual graph with appropriate edge weights. If the dual graph does not have a negative cycle then the distance between two nodes of the dual graph is well defined. Using this distance computation as a subroutine, Miller and Naor give an algorithm to compute the flow in the primal graph. As a special case of the flow problem they solve the perfect matching question in bipartite planar graphs. In this paper we use their reduction in the dynamic setting.

2.5 Maintaining the Determinant

In this paper, the "determinant-maintenance-problem" corresponds to queries for the value of the determinant of a dynamic $n \times n$ matrix. Any one entry of the matrix can be changed from zero to a poly(n)-bit integer, or vice-versa, in one update.

3 Proof of Theorem 1: Determinant in DynTC^0

We combine ideas from the dynamic reachability algorithm of Hesse [7], parallel computation of the determinant (Mahajan-Vinay [9]) and the TC^0 algorithm for computing the iterated product of polynomials by Hesse-Allender-Barrington [8], to yield the claimed result.

Reinterpreting the Mahajan-Vinay [9] result in terms of generating functions we observe that we just need to compute a particular coefficient of the product of a bunch of "signed" and weighted Hesse polynomials to compute the determinant. Use of [8] enables us to perform this computation in TC^0 every time a value of the determinant is queried for.

Let $g_u(x)$ be the generating function of the clows with u as a head. The polynomial g_u is similar to the the Hesse polynnomial counting the number of u, u (closed) walks. To make this correspondence precise we notice that the basic difference between the two polynomials is that unlike in an arbitrary closed walk:

1. the vertex u occurs exactly once in a clow, and,
2. no vertex smaller than u is contained in the clow with head u.

Let G_u be the subgraph of G from which all vertices smaller than u, including u iself, have been deleted. Consider the sum of the Hesse polynomials $h_{G_u}(v, w)$ for each pair of vertices v, w such that $(w, u), (u, v)$ are edges in the graph (along with a 1 for the 0-length walk). This is because along with the path w, u, v a v, w walk forms a u, u closed walk satisfying the two properties above and vice-versa.

In other words we have shown that:

$$g_u(x) = \sum_{w,v:(w,u),(u,v)\in E(G)} x^2 h_{G_u}(v, w)$$

Proposition 1. *The generating function for clow sequences is:*

$$\prod_{u\in V} (g_u(x) - 1)$$

The -1 term arises because of the sign $(-1)^{n+k} = (-1)^{n-k}$ of a clow sequence.

Notice that maintaining G_u is easy - we just update an edge in G_u iff both its endpoints are larger than u. Now we can maintain the $h(x)$ for the G_u s, which follow the following update rule (on update of edge (i, j)):

$$h_{s,t}(x) := h_{s,t}(x) + b \cdot x \cdot h_{s,i}(x) \sum_{k=0}^{\infty} (b \cdot x \cdot h_{j,i}(x))^k h_{j,t}(x), \quad (1)$$

where b is 1 for addition update and -1 for deletion update. Moreover it suffices to maintain the summation up to only first polynomially many terms. Further, Hesse-Allender-Barrington [8] show in Corollary 6.5 that iterated sum and multiplication of polynomials is in $\mathsf{DLogtime}$-uniform TC^0. Thus we do not explicitly need to maintain the product of the terms but only compute the product when the determinant is queried. This completes the proof of the 0-1 case. In case of arbitrary weights the same proof goes through with a modified update rule replacing b by $b \cdot w(i, j)$. This completes the proof of Theorem 1. □

4 Proof of Theorem 2: Reachability in $\mathsf{DynAC}^0[2]$

The basic idea is to maintain the number of s, t walks modulo 2 (as usual, parameterised on the length) instead of the exact number. Thus if the graph can be made min-unique by appropriate weighting with polynomially bounded weights i.e. if there is a unique minimum weight path between every pair of connected vertices, then it suffices to check the number of paths modulo 2 for a given pair s, t for every weight from up to n because we are guaranteed that the coefficient of the min weight path will be 1. But Reinhardt-Allender [15] provides such a weighting scheme although a non-uniform one.

We have to be a bit careful in maintaining the Hesse polynomials $f_{s,t}$ modulo 2. In Hesse's paper he reduces the problem of maintaining the polynomials to iterated integer product and integer division which are in TC^0 by [8]. However, this reduction does not work for small rings like $\mathbb{Z}_2$.

We will of course maintain $h_{s,t}(x)$ mod 2 therefore in (1) the update will be identical for addition and deletion (since $b = 1 \equiv -1$ mod 2). We will also need to deal with $g_{i,j}(x) = \sum_{k=0}^{\infty} (h_{j,i}(x)x)^k$, where and henceforth in the section, arithmetic is modulo 2 unless explicitly mentioned otherwise. We cannot compute this on the fly because we no longer have the power of TC^0at our disposal. We will side-step the issue of computing g versus maintaining it by just considering an implicit representation of it as described below.

Observe that $g_{i,j}(x) = (1 - xh_{j,i}(x))^{-1}$ by just summing up the geometric series representing it. Thus $g_{i,j}(x) \in \mathbb{Z}_2(x)$ We will maintain each $h_{s,t}(x)$ as a rational polynomial i.e. the ratio of two polynomials rather than expand it as an infinite series/truncation of infinite series. If we do it naïvely the degrees of the numerator and denominator will start to grow as more and more updates occur. A simple observation takes care of this:

Observation 7. *Let $\alpha(x) = \frac{\beta(x)}{\gamma(x)} \in \mathbb{Z}_2(x)$ be a rational function such that the constant term $\gamma(0)$ of $\gamma(x)$ is 1. Further, let $\tilde{\beta}(x), \tilde{\gamma}(x)$ be the truncations of $\beta(x), \gamma(x)$ after the degree d terms. Let $\tilde{\alpha}(x) = \frac{\tilde{\beta}(x)}{\tilde{\gamma}(x)}$. Then $\tilde{\alpha}(x)$ is well defined and in the power series expansion of $\tilde{\alpha}(x)$ all the terms of degree at most d have the same coefficients as the corresponding terms in the expansion of $\alpha(x)$.*

Thus we maintain polynomials $\tilde{\beta}_{s,t}(x), \tilde{\gamma}_{s,t}(x)$ with the intention that $\tilde{f}_{s,t}(x)$ will be the ratio of these two polynomials. Thus the update rule Equation (1) is converted to the following equations:

$$\beta'_{s,t}(x) = \tilde{\beta}_{s,t}(x)\left(\tilde{\gamma}_{j,i}(x) - x\tilde{\beta}_{j,i}(x)\right)\tilde{\gamma}_{s,i}(x)\tilde{\gamma}_{j,t}(x) + \tilde{\gamma}_{s,t}(x)\tilde{\beta}_{s,i}(x)x\tilde{\gamma}_{j,i}(x)\tilde{\beta}_{j,t}(x) \quad (2)$$

$$\gamma'_{s,t}(x) = \tilde{\gamma}_{s,t}(x)\tilde{\gamma}_{s,i}(x)\left(\tilde{\gamma}_{j,i}(x) - x\tilde{\beta}_{j,i}(x)\right)\tilde{\gamma}_{j,t}(x) \quad (3)$$

We can do this in $\mathsf{AC}^0[2]$ because of Fact 1.

Next we truncate the two polynomials above at degree d (where d is the sum of all weights in the graph $= O(n^4)$), to obtain the new values $\tilde{\beta'}_{s,t}(x), \tilde{\gamma'}_{s,t}(x)$. This is also doable in $\mathsf{AC}^0[2]$.

To answer a query about whether there exists an s, t-path in the graph, we need to check the polynomials $\tilde{\beta}_{s,t}(x)$ for the various weighting functions promised by [15] and if any of them is non-zero we know that indeed there exists such a path. Conversely, because we have shown that in a weighting under which there exists a unique min-weight path (of some weight, say $w \leq d = O(n^4)$) between s, t, it must be the case that the coefficient of x^w in the power series expansion of $\frac{\tilde{\beta}_{s,t}(x)}{\tilde{\gamma}_{s,t}(x)}$ is non-zero hence $\tilde{\beta}_{s,t}(x)$ must also be non-zero. This completes the proof of Theorem 2. □

5 Proof of Theorem 3: Planar Reachability in DynFO

In this section we show that reachability in an embedded directed planar graph can be maintained in DynFO when adding or removing edges. Maintaining the transitive closure of a directed graph when adding an edge is easy, and we handle the deletion of edges by maintaining a type of dual to the directed graph, so that removing an edge from the original graph corresponds to adding an edge to the dual, and vice versa. To complete the proof, we show that an extended reachability relation on the dual graph is first-order definable in terms of the relation on the original graph, and vice versa.

We present the proof first in the context of a changing directed graph G which is a directed subgraph of a fixed, embedded planar graph H. Maintaining our data structures when planar edges are added to or removed from H presents no serious difficulties, and the details of this are left to the full version of the paper.

Let H be an embedded planar multigraph, and H' be its planar dual, with self-loops allowed in both graphs. Observe that a cycle in H gives a cut set in H': If all edges in H' dual to (crossing) the edges of a cycle in H are removed, then the remaining graph is not connected. This will be the basic idea of our proof, but our data structures become more complex to handle directed edges and to keep track of what is inside of a cycle and what is outside of a cycle.

The directed graph G is a directed subgraph of H, with up to two oppositely directed edges of G corresponding to each edge of H. We will define a directed dual complement graph G^{ddc} that is similarly a directed subgraph of H', and with the property that an edge is in G^{ddc} iff its corresponding dual edge is missing from G, where dual edge is as defined below.

To determine whether directed paths in G and G^{ddc} cross each other, we associate a crossing number with each pair of directed paths, (p, q) where p is a path in G, and q a path in G^{ddc}. This is the net number of times p crosses q, where a crossing is positive if q crosses p in a right to left direction, when oriented in the direction that p is going. If q crosses in the opposite direction, the crossing is a negative crossing. If the pair of paths is a pair of edges (e, e'), the crossing number is 0 if they do not cross, and is 1 or -1 if the two edges cross. The crossing number of paths is just the bilinear extension of this map to sets of edges, denoted $p \otimes q$.

We can now define the dual directed complement of G, G^{ddc}, as the directed subgraph of H' that contains only the edges whose crossing number with every edge of G is 0 or -1. In other words, for each edge e of G, we remove from the graph G^{ddc} the unique edge whose crossing number with e is $+1$. By construction, then, all paths in G have a negative or zero net crossing number with all paths in G^{ddc}.

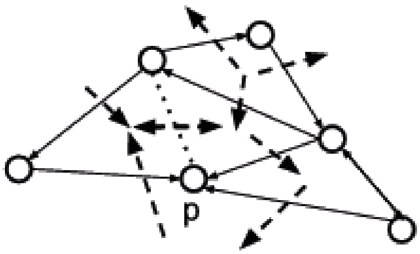

Fig. 1. Directed Dual Complement

In Figure 1, the graph G is shown with circles and solid arrows, and the graph G^{ddc} is shown with dashed arrows for directed edges, and without its nodes drawn. A cycle in G^{ddc} separates two points in G. For example, in Figure X, there is a clockwise cycle of dashed edges in G^{ddc} surrounding point p, and there is therefore no path in G connecting p to a point outside this cycle. If there was a path like that, it would have a crossing number of 1 with the cycle in G^{ddc}, which is impossible, because we excluded all edges with positive crossing numbers from G^{ddc}. It remains, now, to construct a data structure that will tell us about all cycles in G and G^{ddc}, and which points are inside and outside them.

5.1 Canonical Paths and Possible Crossing Numbers

If s and t are both inside or both outside a simple cycle in G^{ddc}, then all paths from s to t will have a net crossing number of 0 with the cycle. If the cycle separates s and t, the net crossing number will be 1 or -1. So we can pick any path from s to t to compute the net crossing number with the cycle. Therefore, we will maintain a spanning tree T of H, and a spanning tree T' of H', and will only compute the net crossing numbers of paths in G^{ddc} with paths in the spanning tree T, and vice versa.

Given a fixed path from s to t, such as the canonical spanning tree path, then if there are two paths from u to v in G^{ddc} with net crossing numbers i and j with that path from s to t, then for any k between i and j ,there is a path from u to v with crossing number k. The proof is left to the full version of the paper. This fact allows us to store only two numbers in our data structure for every such quadruple (s, t, u, v): the lowest number and highest number in the set of possible crossing numbers, if these bounds exist.

5.2 The Paths in G are First-Order Definable from the Paths in G^{ddc}

It should be clear that the bilinearity of net crossing number makes it easy to compute the new sets of possible crossing numbers when an edge is added to G^{ddc}. Let the set of possible net crossing numbers of a path from u to v in G^{ddc} with the canonical spanning tree path from s to t in H be denoted $Crossings((u, v), (s, t))$. If we denote the dual concept, of possible crossing numbers of paths from s to t in G with canonical paths in H' by $Crossings'$, then we can easily update $Crossings$ when deleting an edge from G, because this is just adding an edge to G^{ddc}. We can also update $Crossings'$ when adding an edge to G. If we can compute $Crossings'$ easily from $Crossings$, then we can maintain both of them when adding or deleting edges.

The first part of the formulas relating the two is relatively easy.

Theorem 8. *If the net crossing number of the canonical paths from s to t and u to v is k_0, then if $i \in Crossings((u, v), (s, t))$, and $j > k_0 - i$, then $j \notin Crossings'((s, t), (u, v))$*

Proof. This follows nicely from the bilinearity of $\otimes$ and the fact that the net crossing number of two cycles is zero (on a genus 0 surface, like the plane or a sphere). Let c be the canonical path from s to t in H and c' be the canonical path from u to v. Then if p is the path with crossing number i from u to v in G^{ddc} and q is any path from s to t in G, then $q - c$ is a cycle and $p - c'$ is a cycle. We will show that the crossing number of q and c' must be less than or equal to $k_0 - i$.

$$\begin{aligned} 0 &= (p - c') \otimes (q - c) \\ &= p \otimes q - c' \otimes q - p \otimes c + c' \otimes c \\ &= p \otimes q - c' \otimes q - i + k_0. \end{aligned}$$

But since p is a path in G and q is a path in G^{ddc}, we know that $p \otimes q \leq 0$. So

$$\begin{aligned} 0 &\leq -c' \otimes q - i + k_0 \\ c' \otimes q &\leq k_0 - i. \end{aligned}$$

The opposite direction, that if $j \notin Crossings'((s, t), (u, v))$, then certain paths must exist, is more complex. The path that must exist is not necessarily a single path from u to v in G^{ddc}, but a path from a cycle in G^{ddc} containing u to a cycle containing v. The configuration of two cycles connected by a path looks like a pair of eyeglasses, so we call this the "Eyeglass Lemma", and prove it in the full version of the paper. The proof constructs these paths from the boundaries between regions reachable from s with crossing number i, and those not reachable from s with that crossing number. Any edge in this boundary must be in G^{ddc} or in the canonical spanning path from s to t, since otherwise the edge crossing the boundary would be in G and the path could be extended by this edge without changing the crossing number. Querying whether such an eyeglass configuration exists is a first-order query, since we can check all possible pairs of endpoints of the path between the two cycles, for the existence of the cycles at those points and the path between them.

The final lemmas, whose proofs are left to the full version of the paper, are that these quantities can be updated when the planar graph H is changed by adding or removing an edge, and that all of these quantities can be updated by first order formulas, putting the data structure in the dynamic complexity class DynFO.

6 Open Ends

A promising direction for future work on planar directed reachability is extending our proof from fixed embedded planar graphs to planar graphs whose embedding may change dramatically when adding and removing edges. It appears that a data structure maintaining the decomposition of the planar graph into 3-vertex-connected components (the SPQR-decomposition) would be necessary to keep track of this. Further directions include reducing the complexity of the update computations of this paper's algorithm from AC^0 circuits to quantifier-free first-order formulas, which use function terms but no quantifiers. These correspond to constant-depth circuits with selection gates (multiplexers), but only Boolean gates with constant fan-in. The technique of cut-cycle duality does not seem to extend much beyond the family of planar graphs. However, the techniques imported in this paper such as the Isolation Lemma and its de-randomization, might be useful to further reduce the dynamic complexity of reachability in general directed graphs. As a step towards the long-standing conjecture that reachability in arbitrary directed graphs is in DynFO, one may first want to improve our $\cap_p \mathsf{AC}^0[p]$ bound for updates, to AC^0. We note again that with the current knowledge $\cap_p \mathsf{AC}^0[p]$ may in fact be equal to AC^0.

Acknowledgements. The authors would like to thank Thomas Schwentick for bringing them together for collaboration. The first author would like to acknowledge Jenish C. Mehta's contribution in the initial part of this project without which it would not have been possible. The second author's research was supported by NSF grant CCR-987707.

References

1. Allender, E., Reinhardt, K., Zhou, S.: Isolation, matching, and counting uniform and nonuniform upper bounds. J. Comput. Syst. Sci. 59(2), 164–181 (1999)
2. Bourke, C., Tewari, R., Vinodchandran, N.V.: Directed planar reachability is in unambiguous log-space. TOCT 1(1) (2009)
3. Datta, S., Kulkarni, R., Roy, S.: Deterministically isolating a perfect matching in bipartite planar graphs. Theory Comput. Syst. 47(3), 737–757 (2010)
4. Dong, G., Su, J.: Incremental and decremental evaluation of transitive closure by first-order queries. Information and Computation 120(1), 101–106 (1995)

5. Grädel, E., Siebertz, S.: Dynamic definability. In: ICDT, pp. 236–248 (2012)
6. Hesse, W.: Dynamic computational complexity. Ph.D. thesis, U. Mass. (2003)
7. Hesse, W.: The dynamic complexity of transitive closure is in DynTC^0. Theor. Comput. Sci. 296(3), 473–485 (2003)
8. Hesse, W., Allender, E., Barrington, D.A.M.: Uniform constant-depth threshold circuits for division and iterated multiplication. J. Comput. Syst. Sci. 65(4), 695–716 (2002)
9. Mahajan, M., Vinay, V.: Determinant: Combinatorics, algorithms, and complexity. Chicago J. Theor. Comput. Sci. 1997 (1997)
10. Mehta, J.C.: Dynamic Complexity of Planar 3-connected Graph Isomorphism. In: CSR (accepted, 2014)
11. Miller, G.L., Naor, J.: Flow in planar graphs with multiple sources and sinks. SIAM J. Comput. 24(5), 1002–1017 (1995)
12. Mulmuley, K.: A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. Combinatorica 7(1), 101–104 (1987)
13. Mulmuley, K., Vazirani, U.V., Vazirani, V.V.: Matching is as easy as matrix inversion. Combinatorica 7(1), 105–113 (1987)
14. Patnaik, S., Immerman, N.: Dyn-FO: A parallel, dynamic complexity class. Journal of Computer and System Sciences 55(2), 199–209 (1997)
15. Reinhardt, K., Allender, E.: Making nondeterminism unambiguous. SIAM J. Comput. 29(4), 1118–1131 (2000)
16. Schwentick, T.: Perspectives of Dynamic Complexity. In: Libkin, L., Kohlenbach, U., de Queiroz, R. (eds.) WoLLIC 2013. LNCS, vol. 8071, pp. 33–33. Springer, Heidelberg (2013), `http://dx.doi.org/10.1007/978-3-642-39992-3-6`
17. Vollmer, H.: Introduction to circuit complexity - a uniform approach. Texts in theoretical computer science. Springer (1999)