Electronic Colloquium on Computational Complexity, Report No. 24 (2009)

On the Power of Isolation in Planar Structures

Raghav Kulkarni *

February 26, 2009

Abstract

The purpose of this paper is to study the deterministic *isolation* for certain structures in directed and undirected planar graphs. The motivation behind this work is a recent development on this topic. For example, [BTV 07] isolate a directed path in planar graphs and [DKR 08] isolate a perfect matching in bipartite planar graphs. One natural question raised by their work is: "How far is the reach of the deterministic isolation in planar structures ?"

Our first observation is that the restriction of planarity is in fact, fairly general, in the sense that efficiently isolating certain planar structures would significantly bring down the complexities of some fundamental problems in general graphs. For example, we show that efficiently isolating a cycle cover in directed planar graphs would imply Bipartite-Matching $\in$ *NC*, efficiently isolating a minimum weight perfect matching in undirected planar graphs would imply that non-deterministic log-space computations can be made unambiguous, i.e., $NL = UL$ and efficiently isolating a *Red-Blue path* in directed planar graphs would imply $NP \subseteq \oplus P$.

Further, we show that such efficient isolations are indeed possible in *bipartite* planar graphs, thus leaving non-bipartiteness as the only bottleneck to break. A deceptively simple combinatorial puzzle comes out of our investigations where a positive solution to the puzzle would have strong consequences like $NL = UL$. Our main tools are some new simple bijections, which might be of an independent interest combinatorially.

Keywords: Perfect Matching, Planar Graphs, Deterministic Isolation, *NC, NL.*

1 Introduction

Consider a universe $[m] = \{1, 2, \ldots, m\}$. Let $\mathbf{F}$ be a non-empty family (collection) of subsets of $[m]$. Given a weight w_i for each element $i \in [m]$, define for each $S \subseteq [m]$, $w(S) = \sum_{i \in S} w_i$.

Lemma: 1 ([MVV87] The Isolation Lemma)
If one assigns weights to each element of $[m]$ uniformly and independently at random from 1 to $2m$, then with high ($> \frac{1}{2}$) probability, the minimum weight subset of $\mathbf{F}$ will be unique.

The Isolation Lemma, though initially introduced in the context of *Matching* [MVV87], has found other important applications. For instance, it was used to reduce the *Satisfiability Problem* to *Unique SAT* [MVV87] (to obtain an alternate proof for the result of [VV 86]), to reduce the several variable case to the single variable case in *Polynomial Identity Testing*, to prove $NL/poly \subseteq \oplus L/poly$ and $SAC^1 \subseteq \oplus SAC^1$ [GW 96], to prove that Matching $\in$ *Nonuniform-SPL* and $NL \subseteq UL/poly$ [ARZ 99]. Overall, *isolation* [1] turns out to be very powerful tool in complexity theory.

*Department of Computer Science, University of Chicago, USA. raghav@cs.uchicago.edu

[1] the process of efficiently assigning small weights to the elements of a universe so that the minimum weight subset of our interest becomes unique

ISSN 1433-8092

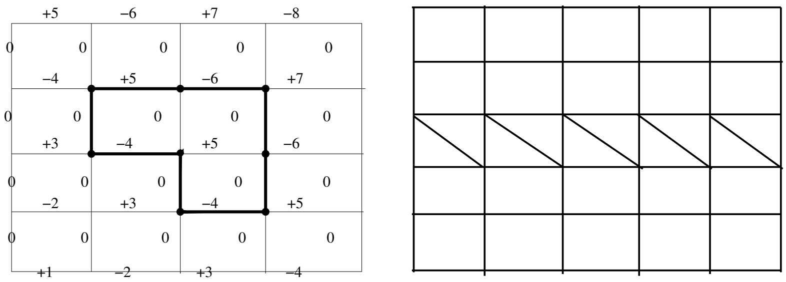

Figure 1: (a) Grid: Log-space computable small size weights exist which give non-vanishing circulations [DKR 08] (b) Near-Grid: Does there exist an efficiently computable small size weighting which gives non-vanishing circulation for every even cycle in Near-Grid?

A typical application of the Isolation Lemma works in two phases. The first phase is *isolation* of a desired object and the second phase is extracting the isolated object. The second phase typically uses some form of counting. In this paper we will focus on the first phase. The crucial components of the isolation phase are the use of small size ($O(\log m)$ bit) weights and the small complexity of assigning the weights. Such an *efficient* deterministic isolation is not possible in the most general setting [Agrawal 07]. However, in most of the important applications of the Isolation Lemma, the universe is highly structured. It is natural to ask whether the isolation becomes simpler if we assume further structure on the family of subsets and on the universe itself. For instance, [ARZ 99] prove that efficient deterministic isolation is indeed possible for *Matching* and NL collapses to UL under the assumption that certain *secure pseudo-random generators* exist.

1.1 Our Focus

We focus on directed and undirected planar graphs and study whether the isolation for certain structures becomes simpler under the restriction of planarity. The motivation comes from recent positive results on this topic. [BTV 07] isolate a directed path in *grid graphs*. Subsequently, building upon [ADR 05] and [ABCDR 06], they show that Directed-Planar-Reachability $\in UL$ (Unambiguous-Log-space) as opposed to the reachability in arbitray directed graphs being *NL-complete*. Further, [DKR 08] isolate a perfect matching in bipartite planar graphs, proving that Bipartite-Planar-Perfect-Matching $\in SPL \subseteq \oplus L$.

We ask the following questions: "What kind of structures in planar graphs admit an efficient deterministic isolation?" and "Does the isolation in planar structures give any insight about the isolation in general graphs?" We provide an evidence that planar isolation is indeed a powerful tool. Sufficiently strong (and plausible) isolation in certain planar structures would imply strong results such as Bipartite-Matching $\in NC$, $NL = UL$ and $NP \subseteq \oplus P$. While we are unable to prove such strong isolations for arbitrary planar graphs, we can prove them for *bipartite* planar graphs. Thus, removing the bipartiteness restriction would be a possible way to attack these problems.

Below we illustrate, with a concrete example, the context in which our results are interesting and the flavor of their implications.

Definition: 1 (Circulation of a Cycle) *Given an even length cycle $C = (e_1, e_2, \ldots, e_{2k})$ in a weighted undirected graph, $circ(C) := |\sum_{i=1}^{2k} (-1)^i w_{e_i}|$.*

Lemma: 2 ([DKR 08]) *Given a bipartite planar graph, one can assign in Log-space, $O(\log n)$ bit weights to its edges so that the* circulation *of any cycle is non-zero.*

[DKR 08] use this lemma to show Bipartite-Planar-Perfect-Matching $\in SPL \subseteq \oplus L$. Generalizing their lemma to non-bipartite graphs would solve positively a longstanding open question: "Is non-bipartite Planar-Perfect-Matching-Search in NC?" Moreover, [DKR 08] also show that it would suffice to find an assignment that yields non-vanishing circulations in *near-grid-graphs* (grid graphs with at most one row allowed to contain diagonal edges, see Figure 1).

Open Question: 1 ([DKR 08]) *Does there exist an efficiently computable weighting of* Near-Grid (Figure 1) *which assigns $O(\log n)$ bit weights to its edges in such a way that for every even length cycle C we get $circ(C) \neq 0$?*

In this paper, we show that a positive answer to the above question would also prove $NL = UL$.

1.2 Our Main Results

Theorem I.
Part (a)
In planar graphs, *efficiently* [2] *isolating*
(1) a Directed Cycle Cover *would imply* Bipartite-Matching $\in NC$
(2) a Minimum Weight Perfect Matching *would imply $NL = UL$*
(3) a Directed Red-Blue Path *would imply $NP \subseteq \oplus P$.*
Part (b)
Such efficient isolations exist for bipartite *planar graphs.*

Theorem II.
The following problems in planar graphs *are NL-hard and their counting versions are #L-hard.*
(1) Shortest-Augmenting-Path
(2) Min-Weight-Perfect-Matching (*even with* 0-1 *weights*)
(3) Exact-Matching.

To put these hardness results in perspective, note that Perfect-Matching-Decision in planar graphs is known to be in NC and Min-Weight-Perfect-Matching in Bipartite Planar graphs is in fact in SPL [DKR 08]. This means that a similar *NL-hardness* for Min-Weight-Perfect-Matching in Bipartite Planar graphs would imply $NL \subseteq SPL \subseteq \oplus L$. Also, our *NL-hardness* for non-bipartite weighted perfect matching might give some insight explaining the lack of progress on obtaining an NC algorithm for non-bipartite perfect matching search even when the bipartite case has been known to be parallelizable for long time. Our main technique for proving a hardness result is to replace a crossing by a suitable planarizing gadget after preprocessing the graph to avoid undesirable cases.

[BTV 07] prove a crucial lemma for grid graphs which says that the edges of a grid graph can be assigned small weights such that sum of the weights of the edges along any directed cycle is nonzero. This lemma turns out to be the key for deterministic isolation. We rephrase that lemma for arbitrary planar graphs as Lemma 3 (See Section 4) and provide an alternate proof for it. Grid

[2] Note that the notion of the *efficiency* depends on the context. For example, if one wants to prove Bipartite-Matching $\in NC$ using an efficient isolation, then the isolation should work in NC, on the other hand to prove $NL = UL$, the isolation should work in UL.

graphs are bipartite and at first look it is not obvious how to accommodate non-bipartite planar graphs in the picture. The interesting part of our alternate proof is its resemblance to the method of finding a *Pffafian Orientation* [Kasteleyn 67] of a planar graph. We leave an open possibility of generalizing our method for any *Pffafian* graph, for instance $K_{3,3}$-free graphs. Finally, we apply Lemma 3 to get efficient isolations for a variety of different structures in planar graphs.

Organization: Section 3 presents four simple bijections that play crucial role in our results. The proofs of Theorem I and II are distributed over Section 4. Section 5 contains the various limited cases where we can achieve efficient deterministic isolation.

2 Preliminaries

See [Vollmer 99] for definitions of standard complexity classes.

Definition: 2 (Some Complexity Classes) *(SPL:) The complexity class SPL is the class of problems which are Log-space reducible to the problem of deciding whether the determinant of a matrix is* 0 *or not under the promise that the determinant is either* 0 *or* 1. *($SPL \subseteq \oplus L$.)*
(UL:) The class UL consists of the problems which are solvable by an NL-machine which has at most one accepting path. ($UL \subseteq NL$, $UL \subseteq SPL$.)
(LogFewNL:) The class LogFewNL consists of the problems that are solvable by an NL-machine which has at most polynomially many accepting paths. ($LogFewNL \subseteq NL$, $LogFewNL \subseteq SPL$.)

Definition: 3 (Red-Blue Graph) *A Red-Blue graph is simply a graph (directed or undirected) in which each edge is colored either Red or Blue. A Red-Blue alternating path in Red-Blue graph is a simple path in which two consecutive edges are of two different colors. A Red-Blue cycle cover in a Red-Blue graph is similarly a cycle cover in which two edge sharing an end point are of two different color.*

3 Four Simple Bijections

3.1 Directed Cycle Covers: General to Planar

Bijection: 1
Part (a) [DKLM 07] *Given a directed graph $\overrightarrow{G}$ one can compute, in Log-space, a directed planar graph* $\mathsf{Planar}(\overrightarrow{G})$ *such that there is a bijection between the cycle covers of $\overrightarrow{G}$ and the cycle covers of* $\mathsf{Planar}(\overrightarrow{G})$.
Part (b) *Moreover, the bijection can be made weight preserving and has the following skew symmetric pullback: If one can isolate a cycle cover in planar graphs using a skew symmetric weighting with each weight at most K bits long, then a cycle cover in arbitrary graphs can be isolated using $O(K)$ bits long weights.*

Proof of Part (a): First consider any straight line drawing of edges of $\overrightarrow{G}$ on plane in which no three straight lines intersect at the same point. Replace each crossing by the gadget in figure 2(a). It is easy to check that the cycle covers remain in one to one correspondence under this transformation. (Details of how to make sure by a Log-space procedure that no three lines intersect at the same point, can be found in [DKLM 07].)
Proof of Part (b): Given a directed graph, replace each crossing with the gadget in figure 2(a).

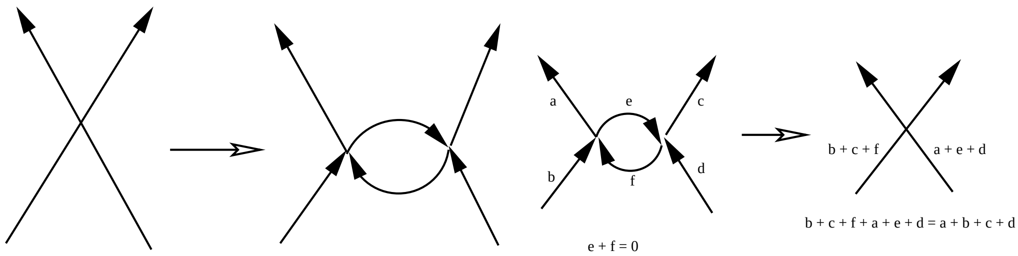

Figure 2: (a) Planarity Transformation Preserving Cycle Covers (b) Skew Symmetric Pullback

Assign skew symmetric weights to the planar graph formed so that a cycle cover in it is isolated. Now, because of the skew symmetric weights, it is easy to see that using the gadget in the figure 2(b), one can actually pull back the weights from planar graph to the original graph so that now minimum weight cycle cover is unique. $\square$

3.2 Construction of Layered DAG with Weights

Bijection: 2
Part (a) *(Shortest s-t Paths: Negative to Positive Weights) If we are given a directed graph* $\overrightarrow{G}$ *with* $O(\log n)$ *bit weights (positive, zero or negative) assigned to its edges and a pair of fixed nodes* s *and* t *in* $\overrightarrow{G}$*, we can construct, in Log-space, another layered directed acyclic graph* $\overrightarrow{G'}$ *with* $O(\log n)$ *bit weights on its edges and a pair of fixed nodes* s' *and* t' *in* $\overrightarrow{G'}$ *such that*
(1) *all weights are non-negative,*
(2) *the minimum weight shortest (least number of edges) s-t paths in* $\overrightarrow{G}$ *are in one to one correspondence with the minimum weight* s'*-*t' *paths in* $\overrightarrow{G'}$*.*
Part (b) *(Min-Weight-Longest s-t Paths to Min-Weight s-t paths in Layered DAG) The above bijection holds for Min-Weight-Longest s-t paths in DAG as well through the same construction of the layered graph with slightly different weights.*

Proof: Consider the layered graph $\mathsf{Layer}(\overrightarrow{G})$ associated with $\overrightarrow{G}$. Each layer is a copy of vertices of $\overrightarrow{G}$. The edges go only from one layer to the next. The copy of vertex i in one layer has a directed edge to the copy of vertex j in the next layer iff $\overrightarrow{G}$ has a directed edge from i to j. The number of layers is precisely one plus the number of vertices in $\overrightarrow{G}$, i.e., $n+1$. Additionally, the copy of vertex i in one layer has a directed edge to the copy of vertex i in the next layer. It is clear that there is a directed path from s to t in $\overrightarrow{G}$ iff there is a directed path from s_0 (the copy of s in the very first layer) to t_n (the copy of t in the very last layer) in $\mathsf{Layer}(\overrightarrow{G})$. The advantage of layered graph is that all the paths from s_0 to t_n have the same length. Now, for an edge from i to j in $\overrightarrow{G}$, give the weight of all edges from the copy of vertex i in one layer to the copy of vertex j in the next layer, the same as the weight of the weight of the edge from i to j in $\overrightarrow{G}$. Also, give all the edges going from the copy of vertex i in layer k to the copy of vertex i in the layer $k+1$ a weight of $-(k+W)$ where W is sufficiently large, say n^4 times the sum of absolute values of the weights of the edges of $\overrightarrow{G}$. Thus, a minimum weight s_0 to t_n path has to use maximum possible such edges of weight $-(k+W)$. Thus it corresponds to the shortest s-t path in $\overrightarrow{G}$. Moreover, because of the additional factor of $-k$, the minimum weight path has to start as early as possible and end as early

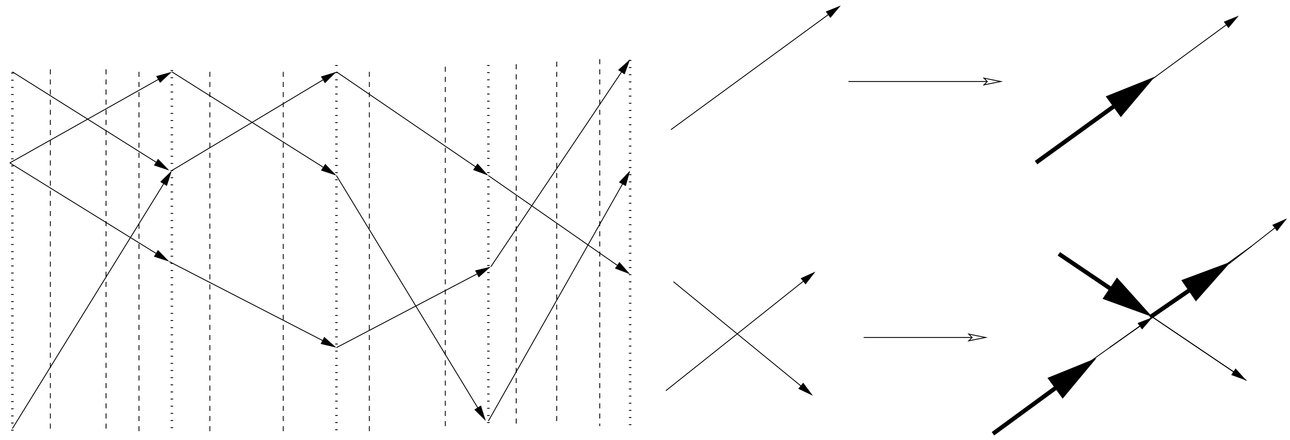

Figure 3: (a) Refining the Layers (b) Red-Blue Gadget.

as possible. Thus, if the shortest length is say l then the minimum weight path from s_0 to t_n now goes from s_0 to t_l and then it just goes from t in one copy to t in the next copy till it reaches t_n. It is easy to see that minimum weight paths from s_0 to t_n are in one to one correspondence with the minimum weight shortest s-t paths in original graph $\overrightarrow{G}$. Now, to make all weights positive, just add a sufficiently large weight , say W' equals to twice that of W, to each of the edges. Since all s_0 to t_n paths have the same length, the weight of every path changes by the same amount and thus the minimum weight path remains the minimum weight path and now we have all the weights non-negative. One can even make them positive by choosing W' large enough. It is easy to see that this bijection is weight preserving, i.e., the ordering of the paths from smallest weight to the largest weight is preserved.

Note that by choosing the weights of the edges going from vertex i in one copy to the vertex i in the next copy to be high positive value, one can get a similar bijection from minimum weight longest paths in a DAG to min weight s-t paths in a layered DAG. $\square$

3.3 Paths in Layered DAG to Red-Blue Paths in Layered Planar DAG

Bijection: 3 *Given a layered DAG, $\overrightarrow{G}$, one can construct in Log-space, a planar DAG, say $\overrightarrow{G'}$, with each edge colored Red or Blue, such that the s-t paths in $\overrightarrow{G}$ are in one to one correspondence with s'-t' Red-Blue alternating paths in $\overrightarrow{G'}$. Moreover, the bijection is weight preserving.*

Proof: Firstly, consider a straight line layout of $\overrightarrow{G}$ in which each layer is put on a vertical line and no three straight lines intersect at the same point. [DKLM 07] describes a Log-space procedure for drawing the layers in such a fashion. Now, at every crossing, imagine a virtual node. Consider the projections of the actual and virtual nodes onto the X-axis. Suppose $x_1, x_2, \ldots, x_k$ are distinct X-co-ordinates in the increasing order. Imagine a virtual vertical line at every $\frac{1}{2}(x_i + x_{i+1})$ and wherever the virtual vertical line intersects the original straight line corresponding to an edge in $\overrightarrow{G}$, split the edge by putting an additional node at the intersection. Paths are preserved by splitting. Moreover, this procedure makes sure that an edge in the new graph can be part of at most one crossing. This will allow us to replace each crossing by a gadget without having to interfere with other crossings. Now, replace each crossing in this new refined graph by a gadget in Figure 3. It

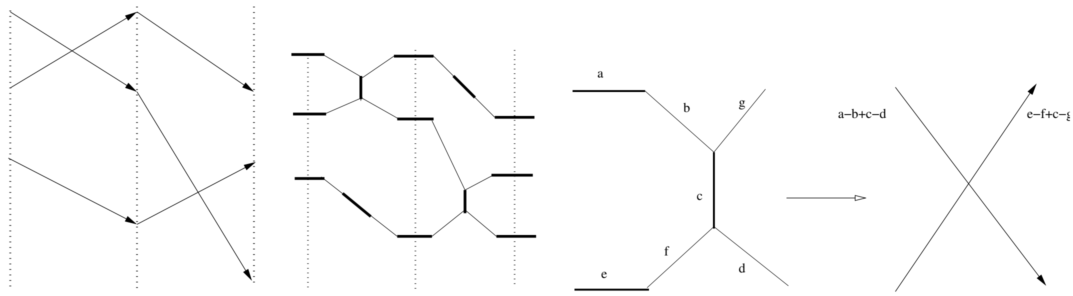

Figure 4: Reducing Layered DAG Reachability to Shortest-Augmenting-Path in Planar Graphs

is easy to check that s-t paths in $\overrightarrow{G}$ remain in bijection with $s' - t'$ Red-Blue alternating paths in the final graph $\overrightarrow{G'}$. Also the procedure described above works in Log-space. $\square$

3.4 Paths in Layered DAG to Min-Weight-PM in Planar Graph

Bijection: 4 *Given a layered DAG $\overrightarrow{G}$, one can construct in Log-space, an undirected planar graph G' with polynomially bounded weights on its edges, and a number w, such that*
(a) *$\overrightarrow{G}$ has a s-t path iff the minimum weight perfect matching in G' has weight w.*
(b) *s-t paths in $\overrightarrow{G}$ are in bijection with the perfect matchings of weight w in G'.*

Proof: First, apply the procedure in the previous lemma to refine the layering of $\overrightarrow{G}$ so that each edge belongs to at most one crossing. We assume that $\overrightarrow{G}$ has this property. Now, replace each crossing as shown in the Figure 4. First make two copies of every vertical layer. Put them next to each other and draw an undirected edge connecting the two copies of the the same node. Label these edges as *matched*. Also the middle edge in the gadget is considered *matched* as shown in the Figure 4. Now, add a new node s' to the graph and draw an undirected edge from s' to the copy of s in the very first layer. Similarly, add a new node t' and draw an undirected edge from the copy of t in the very last layer to t'. Also, replace every directed edge that was not part of any crossing by a path of length three and label the middle edge in this path as *matched*. It is easy to see that, if $\overrightarrow{G}$ has k layers, then s-t paths in $\overrightarrow{G}$ are in bijection with alternating paths of length $4k+2$ in G'. In fact, any augmenting path in G' has to use at least $4k+2$ edges, as every path has to cross each layer and a path that goes backwards will end up using more edges. Now, by giving all *matched* edges weight 0 and all unmatched edges weight 1, we get a bijection from s-t paths in $\overrightarrow{G}$ to perfect matchings of weight $2k+2$ in G'. $\square$

4 Implications of the Four Bijections

4.1 Alternate Proofs for Some Known Results

Lemma: 3 (Key to the Deterministic Planar Isolation) *Given an undirected planar graph G, consider an associated directed graph $\overrightarrow{G}$ by replacing each undirected edge (i,j) by two directed edges, one edge directed from i to j and the other from j to i. One can assign, in Log-space, small size ($O(\log n)$ bit) weights to these directed edges such that*
(a) *the weights are skew symmetric, i.e., $w(i,j) = -w(j,i)$.*

(b) *Let $\overrightarrow{C}$ be a simple directed cycle in $\overrightarrow{G}$ and let $\overrightarrow{e}$ be a directed edge of $\overrightarrow{C}$. Then,* $(\forall \overrightarrow{C})(\sum_{\overrightarrow{e} \in \overrightarrow{C}} w_{\overrightarrow{e}} \neq 0)$.

Proof: Without loss of generality, assume that G is connected.

1. Find a spanning tree T of G.
2. For every undirected edge $(i, j) \in T$, set $w(i, j) = 0$ as well as $w(j, i) = 0$.
3. Find the spanning tree T^* in the dual graph G^* consisting precisely edges e^* such that $e \notin T$.
   - Make an Euler traversal along T^*.
   - Every time, while going from $u^* \in T^*$ to $v^* \in T^*$ via an edge e^* where $e = (i, j)$, reset $w(i, j)$ so that the anticlockwise traversal along the face corresponding to u^* sums up exactly to 1.
   - Reset $w(j, i) = -w(i, j)$. Because of the skew symmetry, the clockwise sum will be -1.

Now, for any cycle, the sum of the weights in anticlockwise traversal decomposes into the anticlockwise sum for the faces in the interior of it and hence counts precisely the number of faces in the interior of the cycle. This has to be non-zero for any simple cycle. It is easy to check that the weights remain polynomially bounded and the algorithm works in Log-space. □

We note that the Lemma 2 can be used to give an alternate proof for Lemma 3.

Lemma: 4 *In any class of graphs closed under subdivision of edges, Lemma 2 implies Lemma 3.*

Proof: Given an undirected planar graph, replace every undirected edge (u, v) by a path $u - w - v$ of length two. Now, the graph is bipartite. Use the Lemma 2 to assign weights. Say weight of (u, w) is a and weight of (w, v) is b. Now the directed edge (u, v) will get weight $a - b$ whereas the directed edge (v, u) will get weight $b - a$. The circulation being nonzero will translate to the cycle sum in the new directed graph being nonzero. □

It is not clear though, whether Lemma 3 implies the Lemma 2 in general graphs.

Implication: 1 ([BTV 07]) Directed-Planar-Reachability *is in UL.*

Proof: Assign the weights as described in Lemma 3. Use Bijection 2 to make weights positive and then use [ARZ 99] to extract the isolated path. □

Implication: 2 ([LMN 08]) Longest-Path *in planar DAG is in UL.* [3]

Proof: As observed in [LMN 08], Lemma 3 will also isolate longest path in planar DAG. The Bijection 2 can be used and then the minimum weight path in new graph (which corresponds to the longest path in the original planar DAG) can be extracted using [ARZ 99]. □

[3] This result was proved recently by [LMN 08]. Our proof is arguably simpler but conceptually not much different from that of [LMN 08]. Already [LMN 08] give two different proofs of the same result, out of which one proof has an interesting application of inductive counting which might turn out to be useful in other contexts such as while dealing with extracting other variations of directed paths after isolation.

4.2 Planar Isolation: Powerful but Hard

The next implication follows because of Bijection 1 and the standard bijection from the perfect matchings in bipartite graphs to the cycle covers in an associated directed graph.

Implication: 3 *If a cycle cover in a directed planar graph can be found in NC then bipartite perfect matching is in NC. Moreover, deciding whether a directed planar graph has a cycle cover in NC would imply decision version of bipartite perfect matching is in NC. If an NC-computable $O(\log n)$ bit weighting isolates a cycle cover in planar graphs then* Bipartite-Matching *is in NC.*

Implication: 4 *If a UL-computable $O(\log n)$ bit skew symmetric weighting isolates a cycle cover in planar graphs, then (a) Bipartite Matching is in SPL, and (b) NL = UL.*

Proof: The implication of Bipartite-Matching being in *SPL* will follow from the Bijection 1 because once we transform the graph into planar graph and isolate a cycle cover in it, extracting the isolated cycle cover is in *SPL* with weights being $O(\log n)$ bit. Note that the skew symmetry of the weights is not required for this part. To see the second implication, consider a layered directed acyclic graph and two nodes s and t in it. Reachability in such graphs is complete for NL. Add a self-loop at every node and make the weight of the self-loop to be 0 and also add a directed edge from t to s of weight 0. Other edges have weight 1. Now, replace each crossing by the gadget in Figure 2(a). Now, construct a $O(\log n)$ bit skew symmetric weighting for the planar graph formed so that the minimum weight cycle cover in unique. Using the Bijection 1, these weights can be pulled back so that the minimum weight cycle cover in the original graph is unique. Now the minimum weight cycle cover exactly corresponds to minimum weight s-t path with the weight of the path being the weight of the cycle cover as skew symmetry will force self loops to have weight 0. Thus, minimum weight s-t path in the original graph is unique. Use Bijection 2 and [ARZ 99] to compute such a unique min weight path in *UL*. Note that without the skew symmetry of the weights we can still get the implication that $NL \subseteq SPL$ because extracting out the isolated cycle cover in planar graph is in *SPL*. □

Definition: 4 *Given a weighted undirected graph, suppose one wants to perturb the weights of its edges so that the minimum weight perfect matching becomes unique. One can first multiply the weight of every edge by a large polynomial factor and add the perturbation weight to the edge which guarantees the minimum weight perfect matching becomes unique. If perturbation is done using K bit weights, then we call such perturbation as the isolation using K bit long* perturbation weights.

Implication: 5 (Hardness of Generalizing the Weighting Scheme of [DKR 08]) *Let C be a complexity class such that $UL \subseteq C$. If an $O(\log n)$ bit perturbation weighting that isolates a minimum weight perfect matching in non-bipartite planar graphs, can be computed in C then $NL \subseteq C$.*

Proof: By Bijection 4, such an isolation is sufficient to isolate a s-t path in layered directed acyclic graph since weights can be pulled back as shown in Figure 4. Now, using Bijection 2 one can make those weights positive and then extract the isolated path in *UL* using [ARZ 99]. □

Theorem: 1 *The following problems are NL-hard. (Counting versions are #L-hard.)*
(1) Shortest-Augmenting-Path *in Planar Graphs.*

(2) Min-Weight-Perfect-Matching *in Planar Graphs (even with* 0-1 *weights on the edges).*
(3) Exact-Matching [4] *in Planar Graphs: Given a positive integer* p *and a planar graph with each edge colored either red or blue, decide whether the graph has a perfect matching containing exactly* p *red edges.*

Proof: (1) follows from the proof of Bijection 4. The Bijection 4 shows that s-t path problem in layered directed acyclic graphs reduces to that of finding a minimum weight perfect matching in a planar graph. Note that the weights in Bijection 4 are either 0 or 1. To see (2) and (3), note that if we colour all matching edges as red and all non-matching edges as blue, then any perfect matching must have at least $2k+2$ blue edges and there exists a perfect matching with exactly $2k+2$ blue edges if and only there is a directed s-t path in the original graph.

□

4.3 Isolating a Cycle Cover in Directed Bipartite Planar Graphs

Lemma: 5 *Any two cycle covers in a directed planar graph* G *form a "direction alternating closed walk of even length", i.e., a closed walk in which reversing the directions of alternating edges gives a directed closed walk.*

Proof: Cycle cover in a directed graph G are in bijection with perfect matchings in a related graph G' which is constructed as follows. Take two disjoint copies of V, say V_1 and V_2, now a directed edge from i to j becomes an undirected edge from i in V_1 to j in V_2. Thus, two cycle covers in G will correspond to two perfect matchings in G'. Two perfect matchings in G' are going to form an alternating cycle, which in turn will correspond to the direction alternating closed walk in G. Note that edges of the direction alternating closed walk alternate between two cycle covers, moreover they alternate the direction of traversal too. Also, note that the vertices might repeat, edges do not. Each vertex can occur at most twice. □

Lemma: 6 *Suppose we have one red cycle cover and one blue cycle cover and a direction alternating closed walk formed by them. Then, from a red cycle cover, if you switch the red edges to blue edges only along the direction alternating closed walk, it is going to give you another valid cycle cover.*

Proof: Again, since cycle covers in G correspond to perfect matchings in a related graph G', the switching along a direction alternating closed walk will correspond to alternating along an alternating cycle in G'. Hence will result in a valid perfect matching in G' which corresponds to a valid cycle cover in G. □

Lemma: 7 *Applying the weighting in the Lemma 3 for a directed bipartite planar graph implies that the minimum weight cycle cover in the directed bipartite planar graph is unique.*

Proof: Suppose there were two minimum weight cycle covers. Say one is red and the other is blue. They are going to form a direction alternating red-blue closed walk. Since the graph is bipartite, we can extract a simple direction alternating cycle from the closed walk, i.e., the vertices in such

[4] This problem was first posed by Papadimitriou and Yannakakis [PY 82]. In general graphs the complexity of this problem is yet unresolved and mysteriously admits a *Randomized-NC* algorithm but not yet known to be in P, neither known nor believed to be *NP-complete.* The *RNC* algorithm is a consequence of the Isolation Lemma.

a cycle can not repeat. This is because consider the first time a vertex is repeated, which gives a simple directed cycle in the graph. Now this directed cycle has to be odd cycle. Otherwise this simple cycle itself will form a simple direction alternating cycle and we can work with this simple cycle. Since a bipartite graph does not have any odd cycles, we get a direction alternating simple cycle. Now, let $c(r)$ be the sum of the weights of the red directed edges in the direction alternating cycle and $c(b)$ be the sum of the weights of the blue directed edges in the direction alternating cycle. Suppose red directed edges are going anticlockwise and blue directed edges are going clockwise along the cycle. Then, $c(r) - c(b)$ is precisely the sum of the weights of the edges in anticlockwise traversal of the underlying undirected cycle, which by the Lemma 3 is guaranteed to be non-zero. Hence, contribution of red edges is not equal to the contribution of blue edges in the direction alternating cycle. Hence, we can switch to get a smaller weight cycle cover which is a contradiction. A minimum weight cycle cover in a planar graph can be isolated using small ($\log n$ bit) weights. $\square$

Theorem: 2 *Finding a cycle cover in directed bipartite planar graphs is in SPL.*

Proof: The above sequence of lemmas will prove the isolation. To extract the isolated cycle cover, one can use the determinant of the adjacency matrix and use an argument similar to that used for extracting the isolated perfect matching [ARZ 99]. $\square$

Implication: 6 (Hardness of Generalizing the Weighting Scheme of [BTV 07]) *If the weighting in Lemma 3 is obtainable in a complexity class C containing SPL, for arbitrary bipartite graphs (even for 3-dimensional grid graphs) then it would imply* Bipartite-Matching $\in$ *C.*

Proof: Note that the planarity is not required for arguing that the direction alternating cycle is a simple cycle. Given an undirected bipartite graph, consider it as both way directed graph. The directed version has a cycle cover if and only if the undirected version has a perfect matching. Now, use the weighting from the assumption. $\square$

4.4 Isolating a Red-Blue Path in Directed Bipartite Planar Graphs

Red-Blue-Path in directed graphs is known to be *NP-complete* (even when restricted to graphs of in-degree and out-degree at most 2) [Vornberger 81]. It is easy to see that Red-Blue-Path in DAG is NL-hard.

Lemma: 8 *(a)* Red-Blue-Path *in directed planar graphs is NP-hard.*
(b) Red-Blue-Path *in planar DAG is NL-hard.*

Proof: The gadget in Figure 3 will transfer the hardness from general case to planar case. $\square$

Implication: 7 *(a) If a $\oplus P$-computable weighting assigns $O(\log n)$ bit weights to the edges of a Red-Blue directed planar graph such that minimum weight Red-Blue path is unique, then $NP \subseteq \oplus P$.*
(b) If a UL-computable weighting assigns $O(\log n)$ bit weights to the edges of a Red-Blue planar DAG such that minimum weight Red-Blue path is unique then $NL = UL$.

Proof. Once a Red-Blue path is isolated using weights from $-W$ to $+W$, one can extract it out in $\oplus P$ by finding the parity of the number of Red-Blue paths of weight at most i, from $i = -W, -W+1, -W+2, \ldots$ and stopping when the first time the parity is odd. $\square$

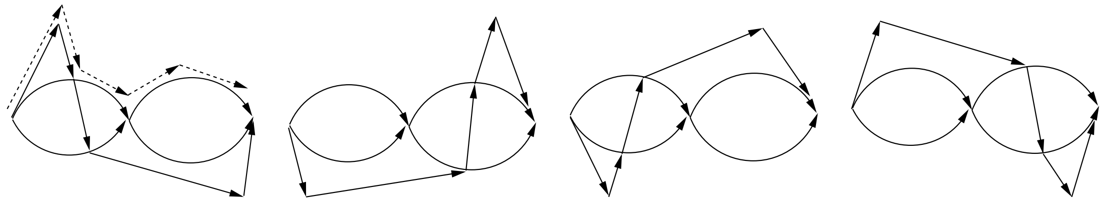

Figure 5: (a) Lighter Even Path (b) Symmetric Case, (c) and (d) are Antisymmetric Cases.

Lemma: 9 *A directed Red-Blue alternating path in bipartite planar graphs can be isolated in UL using $O(\log n)$ bit weights.*

Proof: Use the weighting in Lemma 3 to perturb the weights. If there were two shortest Red-Blue paths, they would form a closed region. Reversing one path along the closed region will give a directed cycle which must have non-zero total sum. Thus, switching along one of the paths, which gives a valid Red-Blue path in case of bipartite graphs, leads to a contradiction. □

Theorem: 3 Red-Blue-Path *in directed bipartite planar graphs is in UL.*

Proof: Once a Red-Blue path is isolated, extracting it out can be done in UL. Look at the auxiliary graph in which a Red edge followed by a Blue edge becomes a new auxiliary edge. Now, carrying the weights to the auxiliary graph, a directed path in auxiliary graph is isolated and thus can be extracted out in UL. Directed path in auxiliary graph will correspond to the Red-Blue path in the original bipartite planar graph. □

4.5 Isolating a Pair of Even Length Paths in Planar DAG

Lemma: 10 *Given a planar DAG, one can assign in Log-space, $O(\log n)$ bit weights to the edges so that there are at most two minimum weight even length s-t paths.*

Proof: The weight of every edge will consists of two co-ordinates. The first coordinate gets the weight as in Lemma 3 whereas the second co-ordinate gets the weight negative of the first co-ordinate. This is double perturbation, firstly the weight of every edge is 1 and we are scaling and perturbing it according to Lemma 3. After this perturbation is performed we will scale and perturb once again but this time using the negative of the perturbation used for the edge first time. Now, firstly observe that any two shortest even length paths can intersect at at most one point. This is because if they intersect at k points then they will form $k+1$ regions and if there are more than two regions then by switching along one or two regions one can get a smaller weight even length path. Moreover, no three paths can intersect at the same point because the portion of the path from the beginning to the first intersection point of these paths will have the same parity for two out of three paths and hence the switching argument will work for these two paths. Now, to get a contradiction, consider three minimum weight even length paths. There are four cases to consider depending on whether the third path intersects the region formed by first two paths from above or from below and in the first region or in the second region. As, shown in the Figure 5 two cases in figure (a) and (b) are symmetric. To prove (a), note that we can form a lighter even length path with respect to the first coordinate. This can be seen by case analysis on the parity of length of each segment shown in Figure 5. Similarly two more cases can be handled, one antisymmetric

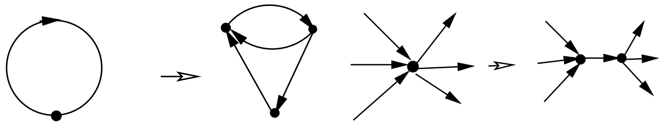

Figure 6: a) Replacing a self-loop b) Making either in-degree or out-degree 1.

to (a), i.e., after negating the weights, becomes same as (a) and other antisymmetric to (b). The paths will differ in the second co-ordinate in antisymmetric cases. □

Theorem: 4 *Finding an even length path in a planar DAG is in LogFewNL.*

Proof: Use the weighting as described in the previous lemma. Note that since we are perturbing the weights by small amount compared to original weights, the new weights are still positive. Form an auxiliary graph by following two edges at time. Now, the number of minimum weight s-t paths in the auxiliary graph is at most 2. Use inductive counting to extract out a pair of even length paths analogous to [ARZ 99]. □

5 Limited Cases where the Isolation is Possible

Finally, by applying Lemma 3, we note that several fundamental structures in planar graphs mentioned below admit an efficient isolation.
The following structures are already known to admit an efficient isolation. (1) s-t paths in directed planar graphs ([BTV 07]) (2) longest path in planar DAG ([LMN 08]) (3) perfect matchings in bipartite planar graphs ([DKR 08]) (4) spanning arborescence in directed planar graphs ([DKR 08]) We augment this list in the following theorem.
Theorem III.
The following structures admit an isolation using a Logspace computable weighting function that assigns $O(\log n)$ bit long weights to the edges. (1) k-factor in bipartite planar graphs (2) cycle cover in directed bipartite planar graphs (3) cycle cover in directed bimodal planar graphs (4) red-blue alternating path in directed bipartite planar graphs (5) red-blue alternating cycle cover in undirected bipartite planar graphs (6) a pair of even length s-t paths in planar DAG (7) a polynomial number s-t paths of length 0 mod k for a constant k, in planar DAG

5.1 Isolating a Cycle Cover in Bimodal Planar Graphs

Lemma: 11 *A simple direction alternating cycle can be found inside a direction alternating closed walk in bimodal planar graphs.*

Proof: Firstly, any bimodal planar graph can be transformed into another bimodal planar graph such that either the in-degree or the out-degree of any vertex is at most one. First replace the self-loops by the gadget in Figure 6 and then use the transformation in Figure 6. Now we have a simplified bimodal planar graph in which the cycle covers are in bijection with those in the original bimodal graph. Again consider the first time a vertex is repeated. This has to form an odd cycle. The vertex which is repeated will have one incoming red edge and one outgoing red edge. Now the closed walk must enter and leave a vertex the same number of times. For this vertex, the red cycle cover has one incoming and one outgoing edge and similar must be true for the blue cycle cover. The edges of red and blue cycle cover are distinct because the red outgoing edge can not be also blue outgoing edge because the rest of the cycle is alternating. Thus, for bimodal graphs the direction alternating cycle is simple. Now, the proof is the same as that of bipartite case. □

Theorem: 5 *A cycle cover in bimodal planar graph can be found in SPL.*

Proof: Since the direction alternating cycle is simple, the weighting in Lemma 2 will isolate a cycle cover and then it can be extracted out in *SPL*. □

5.2 Isolating a 2-factor in Bipartite Planar Graphs

Minimum weight 2-factors in a planar graph are in bijection with minimum weight perfect matchings in an associated planar graph and the bijection is weight preserving. So isolating a minimum weight perfect matching would imply an isolation for minimum weight 2-factor. On the other hand, minimum weight perfect matchings in a planar graph are in bijection with minimum weight perfect matchings in a 3-regular planar graph [BGKMM 86] and thus with minimum weight 2-factors of the 3-regular planar graph by negating the weights. □

Lemma: 12 *Finding a minimum weight 2-factor in a planar graph is NL-hard, even with* 0-1 *weights. Counting minimum weight 2-factors is GapL-complete.*

Proof: It follows from the bijection between minimum weight perfect matchings and minimum weight 2-factors in a planar graph. □

Theorem: 6 *A 2-factor in a bipartite planar graph can be found in SPL.*

Proof: Any two 2-factors will form a simple alternating cycle. Now, using the weighting of [DKR 08] one can isolate a 2-factor and counting in previous lemma helps in extracting. □

5.3 Isolating a Red-Blue Alternating Cycle Cover in Undirected Bipartite Planar graphs

Lemma: 13 *A red-blue alternating cycle cover in undirected bipartite planar graphs can be isolated in Logspace, using* $(\log n)$ *bit weights.*

Proof: Use the weighting of [DKR 08]. Now, consider two subgraphs, one consisting only of the red edges and the other consisting only of the blue edges. The original graph has a red-blue alternating cycle cover if and only if both these subgraphs have a perfect matching. Because of the weighting of [DKR 08], a perfect matching in each of these subgraphs is isolated and thus the minimum weight red-blue alternating cycle exactly corresponds to union of the minimum weight perfect matchings from each subgraph. □

Theorem: 7 *Finding a red-blue cycle cover in undirected bipartite planar graphs is in SPL.*

Proof: As described in the previous lemma, we just need to find a perfet matching in sugraphs of each colour. □

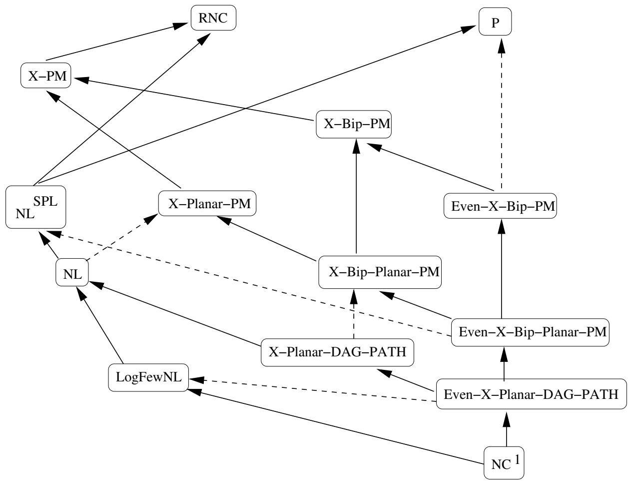

Figure 7: Complexities of Relaxations of Exact-Matching (X-PM) and Exact-Path (X-PATH), the dotted arrows indicate the new results proved in this paper.

5.4 Isolating Polynomial Number of Paths of Length 0 mod k in Planar DAG, for a constant k

Lemma: 14 *Given a planar DAG and a constant k, one can assign in Logspace, $O(\log n)$ bit weights ot its edges so that the number of minimum weight s-t paths of lenght $0 mod k$ is bounded by a polynomial in n.*

Proof: The proof is similar to the 0 mod 2 case. Note first that any two minimum weight paths will form at most k regions. Also, at any point, there can be at most k minimum weight paths intersecting. Thus, since there are only n points, the number of minimum weight paths is bounded by n^k which is a polynomial for a constant k. □

Theorem: 8 *Finding a path of lenght 0 mod k in a planar DAG, for a constant k, is in LogFewNL.*

Proof: First isolate a path of length 0 *mod* k and then form an auxiliary graph by following k edges at a time. There will be polynomially many min weight s-t paths left in the auxiliary graph. Use inductive counting to extract out one such path as in [ARZ 99]. □

5.5 Relaxation of Exact-Matching and Exact-Path

Definition: 5 (Exact-Matching) *Given an undirected graph with each edge labeled either Red or Blue, and an integer k,* Exact-Matching *problem asks to decide whether the graph has a perfect matching containing exactly k Red edges and find one such matching if exists.*

Definition: 6 (Exact-Path) *Given a directed graph with each edge labelled either Red or Blue, an integer k and two specified nodes s and t, decide whether the graph has a simple s-t path that contains exactly k Red edges.*

The Exact-Matching problem is known to be in Randomized NC [MVV87] but not yet known to be in polynomial time. For planar graphs, the decision version of Exact-Matching is in NC and the construction version is known to be in polynomial time [Vazirani 88]. Exact-Path problem, on the other hand, is NP-complete, even when restricted to planar graphs because of the reduction from Hamilon path problem [GJT 68]. We will consider modular relaxations of these problems. Finding whether there is a simple even length path between two specified vertices in a directed graph is known to be NP-complete [LP 83] whereas in directed planar graphs the same problem is solvable in polynomial time [Nedev 99].

Theorem: 9 *Given a bipartite graph with each edge colored either Red or Blue, deciding whether the graph has a perfect matching containing even number of Red edges is in P. Constructing such a perfect matching if exists is also in P.*

Proof: First find any perfect matching in the bipartite graph. Now, construct an auxiliary directed graph with respect to this matching by drawing a directed edge for every path of length two consisting of a matching edge followed by a non-matching edge. If the matching edge as well as the non-matching edge are of the same colour, then give this directed edge a weight 0, otherwise, give it a weight 1. If the original matching constructed itself contains even number of red edges then we are done. Otherwise, there exists such a matching if and only if the auxiliary graph has a path of odd weight. Whether the auxiliary graph has a path of odd weight can be found in polynomial time (in fact in NL). It is easy to see that the whole procedure works in complexity class NL^{Bip-PM}. Note that finding a minimum weight such matching is also in polynomial time when the weights are polynomially bounded. □

Theorem: 10 *Given a bipartite planar graph with each edge coloured either Red or Blue, deciding whether the graph has a perfect matching containing even number of Red edges is in NL^{SPL}. Constructing such a perfect matching if exists is also in NL^{SPL}.*

Proof: Note that in the proof of the previous lemma, the complexity of finding a perfect matching in bipartite planar graphs is SPL. The rest of the proof is identical to the proof of the previous lemma. □

Lemma: 15 *Finding whether a planar DAG contains an s-t path of length exactly k, reduces via a Logspace many one reduction, to the* Exact-Matching *problem in bipartite planar graphs.*

Proof: First add a cycle at every node if necessary to make the indegree plus outdegree at every node at most 3. Thus the graph becomes bimodal. Now use the split graph and colour all the edges corresponding to the original directed edges as Red and the edges corresponding to the edges in the newly added cycles to be Blue. Now it is easy to see that the original graph has a path of length exactly k if and only if the new graph has a perfect matching containing exactly $2k$ Red edges. □

Lemma: 16 *Given a planar DAG with each edge coloured either Red or Blue, deciding whether there is an s-t path that contains exactly k Red edges, i.e,* Exact-Path *problem in planar DAG, reduces in Logspace, to* Exact-Path *in bipartite planar graphs.*

Proof: The proof is similar to the proof of the previous lemma. □

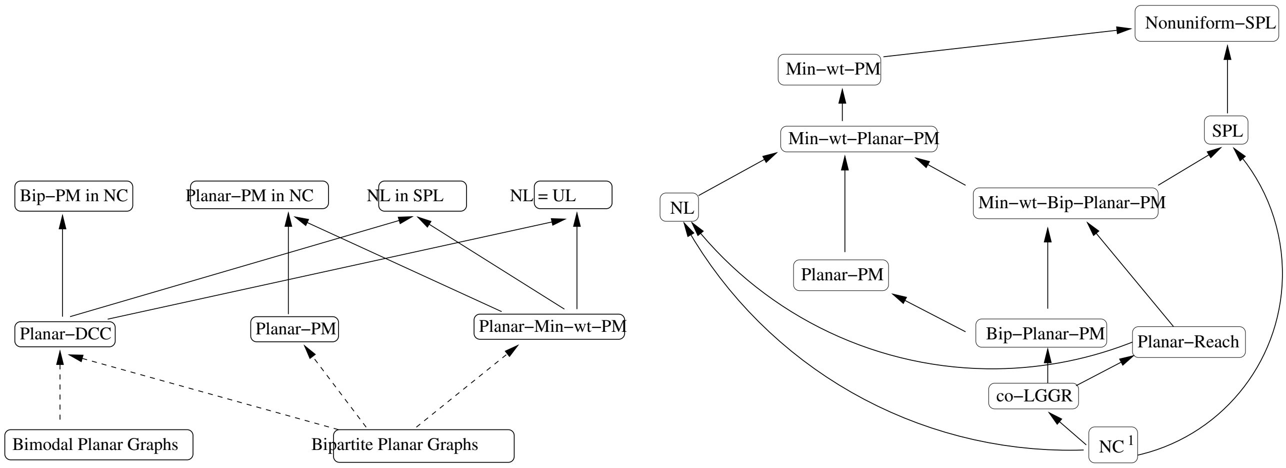

Figure 8: (a) Collapses implied by isolations in planar structures (Dotted arrows indicate the restricted cases where the isolation is possible for that planar structure whereas the normal arrows indicate the results that may be influenced by isolation in that planar structure.) (b) Current Picture about the Complexity of Planar Perfect Matching in Weighted and Unweighted Cases (LGGR stands for Layered Grid Graph Reachability [ABCDR 06])

5.6 Open Questions

Open Question: 2 *Does Lemma 3 generalize for $K_{3,3}$-free graphs ?*

Open Question: 3 *Is unweighted* Planar-Perfect-Matching *NL-hard ?*

Open Question: 4 *Is* Min-wt-Bipartite-Planar-Perfect-Matching *NL-hard ?* (Note that a positive answer to this would imply (from [DKR 08]) that $NL \subseteq \oplus L$.)

Open Question: 5 *We show that isolating a cycle cover in directed bipartite planar graphs is possible. Is the same possible for non-bipartite directed planar graphs ?* (From Bijection 1, such a result would imply that Bipartite-Matching $\in NC$.)

6 Acknowledgement

We thank Samir Datta, Meena Mahajan and Sambuddha Roy for helpful discussions. We thank Laci Babai and Janos Simon for their comments to improve the presentation.

References

[Agrawal 07] M. Agrawal: Rings and Integer Lattices in Computer Science. *Barbados Workshop on Computational Complexity, Lecture no 9, 2007.*

[ABCDR 06] Eric Allender, David A. Mix Barrington, Tanmoy Chakraborty, Samir Datta, Sambuddha Roy: Grid Graph Reachability Problems. IEEE Conference on Computational Complexity 2006: 299-313

[ADR 05] Eric Allender, Samir Datta, Sambuddha Roy: The Directed Planar Reachability Problem. FSTTCS 2005: 238-249

[ARZ 99] Eric Allender, Klaus Reinhardt, Shiyu Zhou: Isolation, Matching, and Counting: Uniform and Nonuniform Upper Bounds. J. Comput. Syst. Sci. 59(2): 164-181 (1999)

[BGKMM 86] E. Bampis, A. Giannakos, A. Karzanov, Y. Manousakis, I. Millis: Perfect matching in general vs. cubic graphs : A note on the planar and bipartite cases. *EDP Sciences, Paris, FRANCE* (1986) (Revue)

[BTV 07] Chris Bourke, Raghunath Tewari, N. V. Vinodchandran: Directed Planar Reachability is in Unambiguous Log-Space. IEEE Conference on Computational Complexity 2007: 217-221

[DKLM 07] Samir Datta, Raghav Kulkarni, Nutan Limaye, Meena Mahajan: Planarity, Determinants, Permanents, and (Unique) Matchings. CSR 2007: 115-126

[DKR 08] Samir Datta, Raghav Kulkarni, Sambuddha Roy: Deterministically Isolating a Perfect Matching in Bipartite Planar Graphs. STACS 2008: 229-240

[GJT 68] Garey M.R., Johnson D.S., Tarjan R.E., The plane Hamiltonian problem is NP-complete, SIAM J. Comput. 5 (1968), 704-714.

[GW 96] Anna Gál, Avi Wigderson: Boolean complexity classes vs. their arithmetic analogs. *Random Struct. Algorithms 9(1-2): 99-111* (1996)

[Kasteleyn 67] P. W. Kasteleyn. Graph theory and crystal physics. In F. Harary, editor, *Graph Theory and Theoretical Physics*, page 43-110, Academic Press, 1967.

[KUW 86] Richard Karp, Eli Upfal, Avi Wigderson. Constructing a perfect matching is in random NC. *Combinatorica, 6:35-48*, 1986.

[LMN 08] Nutan Limaye, Meena Mahajan and Prajakta Nimbhorkar. Longest Path in Planar DAG is in UL. *arXive*

[LP 83] Andrea S. Lapaugh , Christos H. Papadimitriou. The even-path problem for graphs and digraphs. *Networks Volume 14, Issue 4 , Pages 507 - 513* 1983.

[MVV87] Ketan Mulmuley, Umesh Vazirani, Vijay Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1): 105-131, 1987.

[Nedev 99] Zhivko Prodanov Nedev. Finding an Even Simple Path in a Directed Planar Graph. *SIAM Journal on Computing archive Volume 29 , Issue 2 (October 1999) table of contents Pages: 685 - 695 Year of Publication: 1999*

[PY 82] C. H. Papadimitriou and M. Yannakakis. The Complexity of Restricted Spanning Tree Problems. *Journal of ACM, 29 (1982), page 285-309.*

[TO 91] Seinosuke Toda, Mitsunori Ogiwara: Counting Classes Are at Least as Hard as the Polynomial-Time Hierarchy. Structure in Complexity Theory Conference 1991: 2-12

[Vazirani 88] Vijay Vazirani. NC Algorithms for Computing the Number of Perfect Matchings in $K_{3,3}$–free Graphs and Related Problems. SWAT 1988: 233-242

[Vollmer 99] Heribert Vollmer, Introduction to Circuit Complexity - A Uniform Approach; Texts in Theoretical Computer Science. An EATCS Series. Springer Verlag, 1999.

[Vornberger 81] Oliver Vornberger: Alternating Cycle Covers and Paths. *Lecture Notes in Computer Science Publisher: Springer Berlin / Heidelberg Volume 100* 1981

[VV 86] Leslie G. Valiant, Vijay V. Vazirani: NP is as Easy as Detecting Unique Solutions. Theor. Comput. Sci. 47(3): 85-93 (1986)

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/