# Improved Bounds for Bipartite Matching on Surfaces

**Samir Datta[1], Arjun Gopalan[2], Raghav Kulkarni[3], and Raghunath Tewari[4]**

1 **Chennai Mathematical Institute, India**
`sdatta@cmi.ac.in`
2 **BITS Pilani, India**
`arjun91@gmail.com`
3 **LIAFA Paris 7 and LRI* Paris 11, France**
`kulraghav@gmail.com`
4 **Indian Institute of Technology – Kanpur, India**
`rtewari@cse.iitk.ac.in`

**Abstract**

We exhibit the following new upper bounds on the space complexity and the parallel complexity of the Bipartite Perfect Matching (BPM) problem for graphs of small genus:

(1) BPM in planar graphs is in UL (improves upon the SPL bound from Datta et. al. [7]);
(2) BPM in constant genus graphs is in NL (orthogonal to the SPL bound from Datta et. al. [8]);
(3) BPM in poly-logarithmic genus graphs is in NC; (extends the NC bound for $O(\log n)$ genus graphs from Mahajan and Varadarajan [22], and Kulkarni et. al. [19].

For Part (1) we combine the flow technique of Miller and Naor [23] with the double counting technique of Reinhardt and Allender [27]. For Part (2) and (3) we extend [23] to higher genus surfaces in the spirit of Chambers, Erickson and Nayyeri [4].

**1998 ACM Subject Classification** F.1.3 Complexity Measures and Classes

**Keywords and phrases** Perfect Matching, Graphs on Surfaces, Space Complexity, NC, UL

**Digital Object Identifier** 10.4230/LIPIcs.STACS.2012.254

## 1 Introduction

### 1.1 Matching Problems in Graphs

A *matching* $M$ in a graph $G$ is a set of vertex disjoint edges. The end-points of the edges in $M$ are said to be *matched*. A *perfect matching* in a graph $G$ is a matching $M$ such that every vertex of $G$ is matched. See [21] for an excellent introduction to matching and related problems.

Historically, matching problems have played a central role in Algorithms and Complexity Theory. Edmond's *blossom* algorithm [10] for Max-Matching was one of the first examples of a non-trivial polynomial time algorithm. It had a considerable share in initiating the study of *efficient computation*, including the class P itself; Valiant's #P-hardness [30] for counting perfect matchings in bipartite graphs provided surprising insights into the counting complexity classes. The study of whether Perfect-Matching is parallelizable

* Supported by the French ANR Defis program under contract ANR-08-EMER-012 (QRAC project)


29th Symposium on Theoretical Aspects of Computer Science (STACS'12).
Editors: Christoph Dürr, Thomas Wilke; pp. 254–265
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

has yielded powerful tools, such as the *isolating lemma* [25], that have found numerous other applications.

The rich combinatorial structure of matching problems combined with their potential to serve as central problems in the field invites their study from several perspectives. The focus of this paper is on the space and parallel complexity of matching problems. The best known upper bound for Perfect-Matching (and other matching problems mentioned above) is *non-uniform* SPL [2] whereas the best hardness known is NL-hardness [5]. Unless otherwise specified, all circuit classes from now on are uniform (say L-uniform).

## 1.2 Matching Problems in Planar Graphs

A well known example where planarity is a boon is that of counting perfect matchings. The problem in planar graphs is in P [17] as opposed to being #P-hard in general graphs [30]. Counting perfect matchings in planar graphs can in fact be done in NC [31]; thus Perfect-Matching (Decision) in planar graphs is in NC. "Is the construction version of Perfect-Matching in planar graphs in NC?" remains an outstanding open question, whereas the bipartite planar case is known to be in NC [23], [22], [19], [7].

The space complexity of matching problems in planar graphs was first studied by Datta, Kulkarni, and Roy [7] where it was shown that Min-Wt-PM in bipartite planar graphs is in SPL. Kulkarni [18] shows that Min-Wt-PM in planar graphs (not necessarily bipartite) is NL-hard. The only known hardness for Perfect-Matching in planar graphs is L-hardness (cf. [6]). For bipartite planar graphs, nothing better than L-hardness is known.

Given a directed graph $G$ and two vertices $s$ and $t$ in $G$, let Dir-Reach denote the problem of deciding if there exists a path from $s$ to $t$ in $G$. Dir-Reach is NL-complete. It turns out that Dir-Reach in planar graphs reduces (in log-space) to Perfect-Matching in bipartite planar graphs [6]; the former was proved to be in $\mathsf{UL} \cap \mathsf{coUL}$ by Bourke, Tewari, and Vinodchandran [3]. In this paper we show that Perfect-Matching in bipartite planar graphs is in UL, leaving the coUL bound as an intriguing open question.

## 1.3 Matching Problems in small genus graphs

Counting perfect matchings in graphs embedded on $O(\log n)$ genus surfaces is in NC (see Galluccio and Loebl [12]). Combining this with a rounding procedure from Goldberg, Plotkin, Shmoys, and Tardos [13], the authors of [22] and [19] obtain an NC algorithm for the decision and construction versions of BPM in $O(\log n)$ genus graphs. In [8], the result of [7] was extended to bipartite graphs of bounded genus and a tighter bound of $\mathsf{SPL} \subseteq \mathsf{NC}$ was obtained. We are able to improve these results using a technique of Miller and Naor [23] and its extension to higher genus graphs by Chambers, Erickson and Nayyeri [4].

## 1.4 Our Results

▶ **Theorem 1.1.** Perfect-Matching *in bipartite planar graphs is in* UL.

The result holds for both decision and construction versions of the problem. We build on two key algorithms: (1) Miller and Naor's algorithm [23] for perfect matching in bipartite planar graphs; (2) Reinhardt and Allender's [27] UL algorithm for shortest path in *min-unique* graphs: graphs with polynomially bounded edge-weights and having at most one minimum weight path between any pair of vertices. Miller and Naor reduce the Perfect-Matching (Decision) in planar graphs to the following problem in directed planar graphs: Neg-Cycle

(Decision) problem - given a directed graph with polynomially bounded edge-weights, decide whether or not the graph contains a negative weight cycle. The simple observation that this reduction works in log-space combined with the crucial observation that Neg-Cycle problem is in NL yields the somewhat surprising NL bound for perfect matching in bipartite planar graphs. While this upper bound matches the lower bound of NL for matching in bipartite graphs, we are able to improve it to UL by making efficient use of planarity. This brings it tantalizingly close to the best known upper bound of UL ∩ coUL for planar reachability. For the proof of the UL bound in part (a), we first provide a technical extension of (2) when the graph contains negative weight edges but no negative weight cycles. A simple but subtle combination of (1) and (2) then yields the desired result. As opposed to [19] and [7], our space bounded algorithms do not require determinant computation as a subroutine, instead we make use of a variant of planar reachability. However, for the weighted case we do not know how to improve upon the SPL bound in [7]. We also do not know how to improve on $\mathsf{L}^{\mathrm{C_=L}}$ bound for maximum matching due to Hoang [15].

▶ **Theorem 1.2.** Perfect-Matching *in bipartite graphs of constant genus is in* NL.

▶ **Theorem 1.3.** Perfect-Matching *in bipartite graphs of* $(\log n)^{O(1)}$*genus is in* NC.

Again the results hold for both decision and construction versions but we require that a *cellular* embedding of the graph be given as part of the input. We adapt the approach of Chambers et. al. [4] in the context of the flow instance corresponding to the perfect matching problem. Chambers, Erickson and Nayyeri [4] extend the techniques of Miller and Naor to reduce the search space of a max-$s,t$-flow on a surface. In particular, for genus $g$ surface they can formulate the flow problem as a linear program in only $O(g)$ variables. We show that a flow instance corresponding to perfect matching in a bipartite graph embedded on a surface is a yes instance exactly when we can send flows along the $2g$ *basis cycles* such that the residual graph has no negative cycle. Moreover, if we start from a Perfect-Matching instance then the flows must be integral and polynomially bounded. Thus exhaustive search in the $2g$-dimensional space yields an NL algorithm for the existence of a Perfect-Matching when $g$ is a constant. We believe that this NL bound can be improved to UL.

For poly-logarithmic genus, the ellipsoid method yields an NC bound. Our observation is that the separation oracle, the problem of determining if a weighted directed graph contains a negative cycle, can be implemented in NC. For the construction version, we use the rounding procedure of Goldberg, Plotkin, Shmoys and Tardos [13] to obtain an integral solution in NC from the fractional solution coming from the ellipsoid algorithm.

We also consider Even-Path problem: deciding whether or not there is a directed simple path of even length between two specified vertices. Even-Path is NP-complete [20] but restricted to planar graphs it is in P [26]. For directed acyclic graphs (DAGs), the problem is NL-complete. The Even-Path problem can be viewed as a relaxation of the Red-Blue-Path problem - given a directed graph with edges colored Red or Blue, decide whether or not there is a (simple) path between two specified vertices such that consecutive edges in the path are of different colors. The Red-Blue-Path problem is known to be NL-complete for planar DAGs [18]. This provides context for the following theorem.

▶ **Theorem 1.4.** Even-Path *in planar DAG is in* UL.

The hope is that the proof of this theorem contains the seeds for a proof showing that Red-Blue-Path for planar DAGs is in UL which would imply that NL collapses to UL. It is worth noting that our proof of the UL bound for Even-Path in planar DAGs combines two different *deterministic isolation* techniques ([3], [15]).

### 1.5 Organization

Section 2 contains preliminaries. Section 3 contains the UL bound for bipartite planar graphs. Section 4 contains the results for higher genus graphs. Section 5 contains UL bound for Even-Path problem in planar DAG. Section 6 contains some open ends.

## 2 Preliminaries

### 2.1 Matching Problems

We consider the following computational problems related to matching:

- Perfect-Matching (Decision) : decide if $G$ contains a perfect matching.
- Perfect-Matching (Construction) : construct a perfect matching in $G$ (if exists).
- Min-Wt-PM (Decision) : given $G$ together with edge-weights $w : E(G) \to \mathbb{Z}$ such that $|w(e)| \leq n^{O(1)}$, and an integer $k$ - decide if $G$ contains a perfect matching of weight at most $k$.
- Max-Matching (Decision) : given $G$ and an integer $k$, decide if $G$ has a matching of cardinality at least $k$.
- UPM (Decision) : decide if $G$ has a unique perfect matching

### 2.2 Space Complexity Classes

See the monograph by Vollmer [32] for definitions of standard circuit complexity classes. It is known that UL $\subseteq$ NL $\subseteq$ NC $\subseteq$ P and UL $\subseteq$ SPL. It is also known that SPL $\subseteq \oplus$L $\subseteq$ NC. NL and SPL as well as NL and $\oplus$L are not known to be comparable.

### 2.3 Flow Terminology

Here we rephrase the terminology used in [23]. An undirected *edge* is a two element unordered set $\{u, v\}$ such that $u, v \in V$. An undirected graph $G = (V, E)$ consists of a set $V$ of *vertices* and a set $E$ of *undirected edges*. An *arc* is an ordered tuple $(u, v) \in V \times V$. A directed graph $\overrightarrow{G} = (V, \overrightarrow{E})$ consists of a set $V$ of *vertices* and a set $\overrightarrow{E} \subseteq V \times V$ of arcs. Given an undirected graph $G = (U, V)$, its directed version is a directed graph $\overleftrightarrow{G} = (V, \overleftrightarrow{E})$ where $\overleftrightarrow{E} := \{(u, v) \mid \{u, v\} \in E\}$.

A *capacity-demand graph* is a triple $(G, c, d)$ where $G = (V, E)$; every arc $(u, v) \in \overleftrightarrow{E}$ is assigned a real value $c(u, v)$ called the *capacity* of the arc and every vertex $v \in V$ is assigned a real value $d(v)$ called the *demand* at the vertex. A *pseudo-flow* in a capacity-demand graph $(G, c, d)$ is a function $f : \overleftrightarrow{E} \to \mathbb{R}$ such that: (i) for every arc $(u, v) \in \overleftrightarrow{E}$, we have: (skew-symmetry) $f(u, v) = -f(v, u)$, and (ii) for every vertex $v \in V$, we have: (demands met) $\sum_{w \in V:(v,w) \in \overleftrightarrow{E}} f(v, w) = d(v)$. A *flow* in a capacity-demand graph $(G, c, d)$ is a function $f : \overleftrightarrow{E} \to \mathbb{R}$ such that: (a) f is a pseudo-flow in $(G, c, d)$; (b) for every $(u, v) \in \overleftrightarrow{E}$, we have: (capacity constraints satisfied) $f(u, v) \leq c(u, v)$. A *zero-demand graph* $(G, c)$ is a capacity-demand graph in which the demand at every vertex is zero.

For a description of other graph theoretic terminology (such as walk, dual graph etc.), we refer the reader to Diestel's excellent text [9].

### 2.4 Main Lemmas from Miller and Naor [23]

▶ **Definition 2.1** (Directed Dual). Let $G$ be a planar graph. Fix an embedding of $G$ in the plane. Let $G^*$ denote the dual of $G$ with respect to the fixed embedding. The directed dual of $G$ is the directed version of $G^*$ denoted by $\overleftrightarrow{(G^*)}$. The arcs of $\overleftrightarrow{G}$ and that of $\overleftrightarrow{(G^*)}$ are in one to one correspondence. If $e = \{u, v\}$ is an edge in $G$ with a directed version $(u, v)$ and $e^* = \{x^*, y^*\}$ is the corresponding dual edge in $G^*$ then in $\overleftrightarrow{(G^*)}$ the directed edge that corresponds to $(u, v)$ is directed $(x^*, y^*)$ where $x^*$ is the face that lies to the *left* of the directed edge $(u, v)$.

▶ Proposition 2.2 (folklore, see for instance [23]). Let $(G, c)$ be a zero-demand graph. Let $f$ be a flow in $(G, c)$. If $C^* = (e_1^*, \ldots, e_k^*)$ is a directed cycle in $\overleftrightarrow{(G^*)}$, then

$$\sum_{e \,:\, e^* \in C^*} f(e) = 0.$$

▶ **Lemma 2.3** (Miller, Naor [23]). *Let $(G, c)$ be a zero-demand planar graph, then: there exists a flow in $(G, c)$ $\iff$ $\overleftrightarrow{(G^*)}$ has no negative weight cycle with respect to weights $c$.*

## 3 Bipartite Planar Matching: The UL Bounds

Suppose we have a directed graph $G$ with polynomially bounded weights on its edges. The weights could be positive or negative. Let $s$ be a fixed vertex in $G$. Let $d(u, v)$ denote the length of the minimum length path from $u$ to $v$. whenever defined. Notice that these definitions are conditional on the non-existence of negative cycles and we show how to deal with these cases below.

$$V_k := \{v \mid d(s, v) \leq k\}.$$

Let $\mathrm{dist}_k^w(u, v)$ denote the weight of the minimum weight walk (with respect to weights $w$) of length at most $k$ from $u$ to $v$. Note that $\mathrm{dist}_k^w(u, v)$ could be negative. We define,

$$\Sigma_k^w := \sum_{v \in V_k} \mathrm{dist}_k^w(s, v).$$

We use an extension of [27] to compute $\mathrm{dist}_k^w(s, v)$ and $V_k$ in UL. We pause to note that the technique of [27], called double counting in [27], is a generalization of the inductive counting technique used by Immerman [16] and Szelepcsényi [28] to show that NL = coNL. We combine this UL algorithm with Miller and Naor's algorithm via Weighting Scheme A in Section 5 to obtain the UL bound for perfect matching in bipartite planar graphs.

We need an extension of [27], when the graph contains negative weight edges but no negative weight cycles. We call this extension (Algorithm 2) as the *Extended-RA Algorithm* . Following lemmas are simple consequences of the Extended-RA Algorithm and *min-uniqueness* achieved via generalized BTV weights (Weighting Scheme A).

### The weighting scheme A

*Weighting scheme A* is a generalization of the weight function in [3] to planar graphs. In other words, given a directed planar graph $G$, we construct a log-space computable edge weight function with respect to which any simple cycle in $G$ has non-zero weight. Tewari and Vinodchandran [29] give a log-space construction of such a weight function by an application of Green's Theorem. We give an alternate procedure (see Algorithm 1) that achieves the same result.

**Input** : A planar graph $G$
**Output** : An edge weight function $w_A$ such that for any simple cycle $C$ in $G$ $w_A(C) \neq 0$

Compute a spanning tree $T$ in $G$;
For any arc $e \in \overleftrightarrow{T}$, set $w_A(e) = 0$;
Let $R$ denote the spanning tree in $G^*$ consisting of the edges that do not belong to $T$. Fix a root $r$ for $R$ (say the unbounded face) and let $\overrightarrow{R}$ denote the orientation of $R$ where each edge is oriented towards the root;
An arc $e^* = (u, v) \in \overrightarrow{R}$ separates the tree $R$ into two subtrees. Let $\alpha_u$ denote the number of vertices in the subtree containing $u$. Set $w_A(u, v) = \alpha_u$ and $w_A(v, u) = -\alpha_u$;
Set $w_A(e) = w_A(e^*)$ for every $e \in E(G)$ where $e^*$ is the (directed) dual edge of $e$;

**Algorithm 1:** Weighting Scheme A

▶ **Lemma 3.1** (adaptation of [3]). *With respect to the weight function $w_A$ the absolute value of the sum of the weights of the arcs along any simple directed cycle is equal to the number of faces in the interior of the cycle.*

*Proof of Lemma 3.1:* For a simple cycle $C$ of $G$, let us define the weight of $C$, $w(C)$, to be the sum of weights of the edges lying along $C$ in clockwise order. Note that $w_A$ is skew symmetric. Thus clockwise and anti-clockwise weights of the cycle $C$ are the same in absolute value but opposite in sign. We are denoting the clockwise weight of $C$ by $w(C)$.

It suffices to show that for a facial cycle $F$ of $G$, $w(F) = +1$. This is because for a simple cycle $C$:

$$w(C) = \sum_{F \in \text{Interior}(C)} w(F).$$

But $w(F)$ equals the sum of the weights of dual edges (in $G^*$) outgoing from the dual vertex $F^* \in V(G^*)$, so it suffices to show that for every vertex $u \in V(G^*)$:

$$\sum_{v:(u,v) \in E(G^*)} \alpha_v = +1.$$

Observe that the number of nodes in the subtree rooted at $u$ is one more than sum of the number of vertices in the subtrees rooted at $v$ for various $v$, such that $(u, v)$ is a dual edge. This, together with the skew symmetry of the weights $w_A(u, v)$, completes the proof. ☐

**Input** : A directed graph $G$ on $n$ vertices; edge-weights $w : E(G) \to \mathbb{Z}$ such that $|w(e)| \leq n^{O(1)}$; $s, v \in V(G)$; and an integer $t$
**Output** : $\text{dist}_t^w(s, v)$

Initialize $V_0 \leftarrow \{s\}$ and $\Sigma_0^w \leftarrow 0$;
**for** $k = 1$ *to* $t$ **do**
    Compute $(|V_k|, \Sigma_k^w)$ from $(|V_{k-1}|, \Sigma_{k-1}^w)$;
**end**
Compute $\text{dist}_t^w(s, v)$ from $(|V_t|, \Sigma_t^w)$ and output;

**Algorithm 2:** Extended-RA Algorithm (adapted from [27])

▶ **Lemma 3.2.** *Given a directed planar graph with polynomially bounded weights $w$ on its arcs such that there are no negative weight cycles, the shortest distance* $\mathrm{dist}^w(u,v)$ *between any pair of vertices with respect to weights $w$ can be computed in* UL.

▶ **Lemma 3.3.** *Given a directed planar graph with polynomially bounded weights $w$ on its arcs, deciding whether or not the graph contains a negative weight cycle is in* coUL.

**Input** : A bipartite planar graph $G$
**Output** : A perfect matching in $G$ if one exists; else reject

1 Construct a capacity-demand graph $(G, c, d)$ as follows: for each vertex $v \in A$, set $d(v) = 1$ and for each vertex $v \in B$, set $d(v) = -1$. For $u \in A, v \in B$, set $c(u,v) = 1$ and $c(v,u) = 0$;
2 Construct a pseudo-flow $f'$ in $(G, c, d)$ (see [23]);
3 Construct a zero-demand graph $(G, c - f')$;
4 Run Extended-RA Algorithm on $\overleftrightarrow{(G^*)}$ with weights $w = n^4(c - f') + btv$ to compute the shortest distances $\mathrm{dist}_n^w(u,v)$ in $\overleftrightarrow{(G^*)}$, where $btv$ denotes generalized BTV weights (Weighting Scheme A defined in the beginning of this section);
5 Compute $\mathrm{dist}_n^{c-f'}(u,v)$ in $\overleftrightarrow{(G^*)}$ from the above by ignoring the lower order weights from $btv$;
6 Run Miller and Naor's algorithm to compute $f$ (see algorithm in Section 5.1 in [23]);
7 If $f$ is a flow then for $u \in A$ and $v \in B$, output "$u$ is matched to $v$" $\iff f(u,v) = 1$;
8 otherwise reject and output "No perfect matching";

**Algorithm 3:** UL algorithm for Perfect-Matching in bipartite planar graphs

Combining Algorithm 2 with Miller and Naor's algorithm, we obtain the UL algorithm (Algorithm 3) for Perfect-Matching in bipartite planar graphs.

▶ **Theorem 3.4.** *(Theorem 1.1) In bipartite planar graphs, both the decision as well as the construction versions of the* Perfect-Matching *are in* UL.

**Proof.** The correctness of the above algorithm follows from [23]. To see the UL bound, note that the Extended-RA algorithm computes $\mathrm{dist}_n^w$ correctly along a unique path assuming min-uniqueness of the weights. If there are no negative weight cycles then the generalized BTV weights (Section 5) guarantee min-uniqueness.

Thus, if there are no negative weight cycles in $\overleftrightarrow{(G^*)}$ then we obtain a valid flow and a perfect matching along the unique accepting path. Otherwise, we realize that $f$ is not a valid flow and reject. ◀

We also obtain the following corollary on similar lines

▶ **Corollary 3.5.** *Single-source, single-sink maximum flow problem in planar networks with polynomially bounded capacities is in* $\mathsf{L}^{\mathsf{UL}}$.

## 4 Bipartite Perfect Matching in higher genus graphs

We need $G$ to be given together with its *cellular* embedding [4] on a surface of genus $g$. Every graph admits a cellular embedding as the embedding on the minimal genus surface is always cellular [24]. We also need the $2g$ *basis* cycles in $G$ explicitly given to us. The advantage of cellular embedding of $G$ is that every vertex of $G$ corresponds to a face in the dual graph $G^*$ and vice versa. Let $G$ be a graph with a cellular embedding on a surface of genus $g$ and let $C_1, C_2, \ldots, C_{2g}$ be the basis cycles. Using Steps 1, 2, and 3 of Algorithm 3, we first obtain a multiple source multiple sink flow problem and then transform it to a zero demand instance. None of these reductions use planarity. Let $(G, c)$ denote the zero-demand instance associated with $G$ with capacity function $c$. We fix an arbitrary orientation for each $C_i$. For $i = 1, \ldots, 2g$, let $F_i$ denote the flow that is zero everywhere outside $C_i$, i.e., $F_i(e) = 0$ if $e \notin C_i$ and for each $e \in C_i$ the flow value is $f_i$, i.e., $F_i(e) = f_i$ Let $f = (f_1, \ldots, f_{2g})$ and let $c - f$ denote the graph with the weight of edge $e$ defined as $c(e) - \sum_i F_i(e)$.

The following is a generalization of Lemma 2.3 in Section 2 (same as Lemma 4.1 in [23]) for higher genus graphs with cellular embeddings. After we obtained the proof of this lemma, we learned that a similar lemma is already noted by Chambers et al. [4].

▶ **Lemma 4.1.** *The zero demand instance $(G, c)$ admits a valid flow if and only if there exists $f_1, \ldots, f_{2g}$ such that the dual graph $G^*$ with weights $c - f$ has no negative cycles.*

*Moreover: if the capacities $c$ are integral then we can assume $f_i$ to be integral.*

**Proof.** Analogous to the proof of Lemma 3.1 in [4].

If $(G, c)$ admits a valid flow $F$ then we fix $2g$ basis cycles $C_1^*, \ldots, C_{2g}^*$ in the dual with an arbitrarily chosen orientation and we take $f_i = \sum_{e \in C_i^*} F(e)$. We need that $C_i^*$ crosses $C_i$ exactly once, and $C_i^*$ does not cross $C_j$ if $j \neq i$. We *claim* that this choice leaves no negative cycles in the dual with respect to weights $c - f$ (cf. Lemma 3.1 in [4]).

If there exists $f$ such that there are no negative cycles in the dual with respect to weights $c - f$ then using the shortest distance in dual (proof of Lemma 4.1 in [23]) we can get a valid flow in the zero-demand instance. Here we use the fact that the embedding is cellular and hence every vertex corresponds to a cycle in the dual; and the flow obtained by the shortest distance in the dual sums up to zero on every cycle in the dual. ◀

We use the fact that if $(G, c)$ admits a valid flow then there exists $f$ such that the values of $f_i$ are at most $c_{max} \cdot n$, where $c_{max}$ is the maximum absolute value of the capacites to obtain the following.

▶ **Theorem 4.2.** *Given a bipartite graph $G$ together with a cellular embedding on a constant genus surface,* Perfect-Matching *(Decision + Construction) in $G$ is in* NL.

▶ **Theorem 4.3.** *Given a bipartite graph $G$ together with a cellular embedding on a surface of poly-logarithmic genus,* Perfect-Matching *(Decision + Construction) in $G$ is in* NC.

**Proof.** Note that Lemma 4.1 reduces the decision version of the Bipartite Perfect Matching problem to the problem of solving the feasibility of a linear program in variables $f_1, \ldots, f_{2g}$ with the linear constraints that every cycle in the dual is non-negative with respect to weights $c - f$. We use ellipsoid method to solve this problem.

The crucial observation is that the separation oracle for this problem is in NC. The separation oracle in our context is, given a weighted graph, the problem of determining whether or not it contains a negative cycle. This problem is equivalent to checking if all pair shortest paths are well-defined (because otherwise vertices lying on a negative cycle will have negative shortest paths to themselves). Thus a parallelized version of Floyd-Warshall which runs in NC even when the weights are exponential [14] is sufficient for our purpose.

The running time of the algorithm modulo the separation oracle is polynomial in the number of variables and hence in $g^{O(1)}$ time. This yields an NC algorithm for the decision version for poly-log genus graphs, given their embedding in the required form.

The construction version is also in NC,: A solution to the linear program in the $2g$ variables naturally translates to a point inside the Perfect Matching Polytope of $G$ [19]. Pulling back a point from $\mathbb{R}^{2g}$ to $\mathbb{R}^{|E(G)|}$ can be accomplished in L via an argument similar to the proof of Lemma 4.1. An NC procedure to obtain a Perfect Matching, given a point inside the Perfect Matching Polytope is described in [13] (also see Section 3 in [19]). ◀

## 5 Even-Path in planar DAG is in UL

▶ **Definition 5.1** (Red-Blue-Path). Given a directed graph with each edge colored either Red or Blue, a *Red-Blue-Path* from $s$ to $t$ is a (simple) directed path from $s$ to $t$ such that consecutive edges are of different colors. The RED-BLUE-PATH problem is to decide if there is a Red-Blue-Path from $s$ to $t$.

▶ **Definition 5.2** (Even-Path). Given a directed graph and two nodes $s$ and $t$, an *Even-Path* from $s$ to $t$ is a (simple) directed path from $s$ to $t$ containing even number of edges. The EVEN-PATH problem is to decide if there is an Even-Path from $s$ to $t$.

▶ **Theorem 5.3** ([18]). RED-BLUE-PATH *in planar DAGs is* NL*-complete.*

In this section, we prove that the EVEN-PATH problem (which can be viewed as a relaxation of the RED-BLUE-PATH problem as a path starting with say Red edge and ending with say Blue edge is always of even length) in planar DAG is in fact in UL. Our proof involves a combination of two different isolation techniques that are currently available.

▶ **Lemma 5.4.** *Let $G$ be a planar DAG and $u$ and $v$ be any two vertices in $G$. Then with respect to the weight function $w_A$, (a) if $P_1$ and $P_2$ are two minimum weight Even-Paths from $u$ to $v$, then $P_1 \oplus P_2$ (the symmetric difference between the sets of edges of $P_1$ and $P_2$) divides the plane into at most two bounded regions; (b) no three minimum weight Even-Paths from $u$ to $v$ share a common vertex $w$ other than $u$ and $v$, such that the path segments between the vertices $u$ and $w$ and between $w$ and $v$ are not identical. (c) there are at most $2n^4$ minimum weight Even-Paths from $u$ to $v$.*

**Proof.** (a) For the sake of contradiction let $C_1$, $C_2$ and $C_3$ be any three bounded regions of $P_1 \oplus P_2$. Let $P_{ij}$ be the restriction of the $i$-th path to the $j$-th region for $i \in \{1,2\}$ and $j \in \{1,2,3\}$. Observe that $w_A(P_{1j}) \neq w_A(P_{2j})$ since $C_j$ is a simple cycle and by Lemma 3.1 we have that $w_A(C_j) \neq 0$. Now the parity of the lengths of the path segments $P_{1,j}$ and $P_{2,j}$ are different since if they were the same, we could replace the higher weighted segment with the lower weighted one and get an even length path of lesser weight. This implies that $|C_1| + |C_2| + |C_3|$ is odd since each $|C_i|$ is odd. Let $P'_i = \bigcup_j P_{ij}$ for $i \in \{1,2\}$. Therefore either $|P'_1|$ is odd or $|P'_2|$, but not both. Without loss of generality lets assume $|P'_1|$ is odd. For each $j$ pick the path segment between $P_{1j}$ and $P_{2j}$ that has lesser weight to create a set say $P'$. Now $w_A(P')$ is strictly smaller than both $w_A(P'_1)$ and $w_A(P'_2)$. If $|P'|$ is odd then replace $P'_1$ with $P'$ and if $|P'|$ is even then replace $P'_2$ with $P'$ to get a path of smaller weight and same parity. This is a contradiction. Thus $P_1 \oplus P_2$ has at most two bounded regions.

(b) Let $P_1$, $P_2$ and $P_3$ be three minimum weight paths from $u$ to $v$ that share a common vertex (say $w$) such that the segments of each of the three paths between the vertices $u$ and $w$ and between $w$ and $v$ are distinct. In other words, if $P'_i$ and $P''_i$ are the segments of $P_i$ between the vertices $u$ and $w$ and between $w$ and $v$ respectively (for $i \in \{1,2,3\}$), then $\{P'_i\}$ are pairwise non-identical and so are $\{P''_i\}$. There exists at least two path segments between $P'_1$, $P'_2$ and $P'_3$ whose lengths have the same parity. Without loss of generality assume its $P'_1$ and $P'_2$. Now if $w_A(P'_1) \neq w_A(P'_2)$ then since they have the same parity we can pick the lesser weight path between $P_1'$ and $P_2'$ and similarly the lesser weight path between $P_1''$ and $P_2''$ and append them to get an even path of weight less than either that of $P_1$ or $P_2$ from $u$ to $v$. Thus we can assume $w_A(P_1') = w_A(P_2')$. By Lemma 3.1, this implies that $P_1' \oplus P_2'$ as at least two bounded regions. Moreover since $P_1''$ and $P_2''$ are also not identical, therefore $P_1'' \oplus P_2''$ has at least one one bounded region. Thus $P_1 \oplus P_2$ has at least 3 bounded regions, thus contradicting part (a).

(c) Let $a$, $b$, $c$ and $d$ be four vertices in $G$ and let $\mathcal{P}_{a,b,c,d}$ be the set of all minimum weight even length paths from $u$ to $v$ that pass through the vertices $a, b, c$ and $d$ in that order and are vertex disjoint between the vertices $a$ and $b$ and between the vertices $c$ and $d$ respectively. Then by part (b), $\mathcal{P}_{a,b,c,d}$ will have at most 2 paths. Since the total number of such tuples is at most $n^4$, therefore the number of minimum weight, even length $u$-$v$ paths is bounded by $2n^4$. ◀

## Constructing an auxiliary graph

Construct a directed (multi)graph $G'$ from $G$ as follows: the vertex set of $G'$ is the vertex set of $G$. An edge $(v_i, v_j)$ is in $G'$ if and only if there exists a vertex $v_k$ in $G$ and the edges $(v_i, v_k)$ and $(v_k, v_j)$ are in $G$. The weight $w$ of an edges in $G'$ is the sum of the weights of the corresponding two edges in $G$.

Now Lemma 5.5 follows by definition of $G'$ and part (c) of Lemma 5.4.

▶ **Lemma 5.5.** *(a) $G$ has a directed Even-Path from $u$ to $v$ if and only if $G'$ has a directed path from $u$ to $v$; (b) the number of minimum weights paths from $u$ to $v$ in $G'$ with respect to $w_A$ is at most $2n^4$.*

## Weighting scheme $B$

Our weighting scheme $B$ is based on a well known hashing scheme based on primes, due to Fredman, Komlós and Szemerédi [11].

▶ **Lemma 5.6** ([11])**.** *Let $c$ be a constant and $S$ be a set of $n$-bit integers with $|S| \leq n^c$. Then there is a $c'$ and a $c' \log n$-bit prime number $p$ so that for any $x \neq y \in S$ $x \not\equiv y \pmod p$.*

Hoang used this scheme to give better upper bounds for Perfect-Matching in certain classes of graphs [15]. Aduri, Tewari and Vinodchandran showed that reachability in graphs where the number of paths from $s$ to any vertex is bounded by a polynomial is in UL, by applying this hashing scheme. We use Lemma 5.6 here to define a weight function with respect to which $G'$ is min-unique.

Let $p_i$ be the $i^{th}$ prime number. Consider the lexicographical ordering of the edges of $G'$ and denote the $j^{th}$ edge in this ordering by $e_j$. Define the $i^{th}$ weight function (for $1 \leq i \leq q(n)$ and an appropriate polynomial $q(n)$ dictated by Lemma 5.6), $w_{B_i}(e_j) = 2^j (\bmod\ p_i)$.

▶ **Lemma 5.7** (Adapted from [1])**.** *There exists an $i \leq q(n)$ such that the graph $G'$ with respect to the weight function $W_i = w_A \cdot n^{10} + w_{B_i}$ is min-unique.*

**Proof.** Let $\mathcal{P}_v$ be the set of minimum weight paths from $s$ to a vertex $v$ in $G'$, with respect to $w_A$. Then by Lemma 5.5, $|\mathcal{P}_v|$ is bounded by $2n^4$. It follows from Lemma 5.6 that with respect to some $w_{B_i}$, all paths in $\bigcup_v \mathcal{P}_v$ will have distinct weights. Therefore $G'$ is min-unique with respect to $W_i$ for some $i$. ◀

For each $i \in [q(n)]$, check if $G'$ is min-unique with respect to $W_i$ or not. Once we have an appropriate $i$, we can decide reachability in $G'$ in UL [27]. By Lemma 5.5 a path in $G'$ corresponds to an EvenPath in $G$ and thus we have Theorem 5.8.

▶ **Theorem 5.8.** *(Theorem 1.4)* Even-Path *in planar DAGs is in* UL*.*

## 6 Open Ends

Is Neg-Cycle (Decision) in planar graphs in UL? Is Odd-Cycle in planar graphs in ⊕L? Is Perfect-Matching (Decision) in bipartite planar graphs in coUL? Is Min-Wt-PM in bipartite planar graphs in NL? Is Max-Matching in bipartite planar graphs in NL?

## Acknowledgement

We would like to thank Prajakta Nimbhorkar for discussion in the initial stages of the work, in particular for pointing out that the Neg-Cycle problem is in NL. We would like to thank V. Vinochandran for pointing out references [11] and [1] which are crucially used in the proof of Theorem 5.8. The second author has been supported by Microsoft Research India Travel Grant to attend the conference.

## References

1 Pavan Aduri, Raghunath Tewari, and N. V. Vinodchandran. On the power of unambiguity in logspace. Technical Report TR10-009, Electronic Colloquium on Computational Complexity, 2010.
2 Eric Allender, Klaus Reinhardt, and Shiyu Zhou. Isolation, matching, and counting: Uniform and nonuniform upper bounds. *Journal of Computer and System Sciences*, 59:164–181, 1999.
3 Chris Bourke, Raghunath Tewari, and N. V. Vinodchandran. Directed planar reachability is in unambiguous log-space. *ACM Trans. Comput. Theory*, 1(1):1–17, 2009.
4 Erin W. Chambers, Jeff Erickson, and Amir Nayyeri. Homology flows, cohomology cuts. In *STOC*, pages 273–282, 2009.
5 Ashok K. Chandra, Larry Stockmeyer, and Uzi Vishkin. Constant depth reducibility. *SIAM Journal on Computing*, 13(2):423–439, 1984.
6 Samir Datta, Raghav Kulkarni, Nutan Limaye, and Meena Mahajan. Planarity, determinants, permanents, and (unique) matchings. *ACM Trans. Comput. Theory*, 1(3):1–20, 2010.
7 Samir Datta, Raghav Kulkarni, and Sambuddha Roy. Deterministically isolating a perfect matching in bipartite planar graphs. *Theory of Computing Systems*, 47:737–757, 2010. 10.1007/s00224-009-9204-8.
8 Samir Datta, Raghav Kulkarni, Raghunath Tewari, and N. V. Vinodchandran. Space complexity of perfect matching in bounded genus bipartite graphs. Technical Report TR10-079, Electronic Colloquium on Computational Complexity, 2010.
9 Reinhard Diestel. *Graph Theory (Graduate Texts in Mathematics)*. Springer, August 2005.
10 J. Edmonds. Paths, trees and flowers. *Canad. J. Math.*, 17:449–467, 1965.
11 Michael L. Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with 0(1) worst case access time. *J. ACM*, 31:538–544, June 1984.
12 Anna Galluccio and Martin Loebl. On the theory of pfaffian orientations. i. perfect matchings and permanents. *Electr. J. Comb.*, 6, 1999.

13 Andrew V. Goldberg, Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Using interior-point methods for fast parallel algorithms for bipartite matching and related problems. *SIAM J. Comput.*, 21(1):140–150, 1992.

14 Yijie Han, Victor Y. Pan, and John H. Reif. Efficient parallel algorithms for computing all pair shortest paths in directed graphs. *Algorithmica*, 17(4):399–415, 1997.

15 Thanh Minh Hoang. On the matching problem for special graph classes. In *IEEE Conference on Computational Complexity*, pages 139–150, 2010.

16 Neil Immerman. Nondeterministic space is closed under complementation. *SIAM J. Comput.*, 17(5):935–938, 1988.

17 P. W. Kasteleyn. Graph theory and crystal physics. *Graph Theory and Theoretical Physics*, 1:43–110, 1967.

18 Raghav Kulkarni. On the power of isolation in planar graphs. Technical Report TR09-024, Electronic Colloquium on Computational Complexity, 2009.

19 Raghav Kulkarni, Meena Mahajan, and Kasturi R. Varadarajan. Some perfect matchings and perfect half-integral matchings in NC. *Chicago Journal of Theoretical Computer Science*, 2008(4), September 2008.

20 Andrea Lapaugh and Christos Papadimitriou. The even-path problem for graphs and digraphs. *Networks Volume 14, Issue 4 , Pages 507 - 513*, 1983.

21 L. Lovász and M.D. Plummer. *Matching Theory*, volume 29. North-Holland Publishing Co, 1986.

22 Meena Mahajan and Kasturi R. Varadarajan. A new NC-algorithm for finding a perfect matching in bipartite planar and small genus graphs (extended abstract). In *STOC*, pages 351–357, 2000.

23 Gary L. Miller and Joseph Naor. Flow in planar graphs with multiple sources and sinks. *SIAM J. Comput.*, 24(5):1002–1017, 1995.

24 Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces.* John Hopkins University Press, 2001.

25 Ketan Mulmuley, Umesh Vazirani, and Vijay Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7:105–113, 1987.

26 Zhivko Prodanov Nedev. Finding an even simple path in a directed planar graph. *SIAM Journal on Computing, Volume 29 , Issue 2, Oct 99, 685-695*, 1999.

27 Klaus Reinhardt and Eric Allender. Making nondeterminism unambiguous. *SIAM Journal of Computing*, 29:1118–1131, 2000. An earlier version appeared in FOCS 1997, pp. 244–253.

28 Róbert Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Inf.*, 26(3):279–284, 1988.

29 Raghunath Tewari and N. V. Vinodchandran. Green's theorem and isolation in planar graphs. Technical Report TR10-151, Electronic Colloquium on Computational Complexity, 2010.

30 Leslie G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.

31 Vijay Vazirani. NC algorithms for computing the number of perfect matchings in $k_{3,3}$–free graphs and related problems. In *Proceedings of SWAT '88*, pages 233–242, 1988.

32 Heribert Vollmer. *Introduction to Circuit Complexity - A Uniform Approach.* Springer-Verlag, 1999.