Space Complexity of Perfect Matching in Bounded Genus Bipartite Graphs

Samir Datta[a], Raghav Kulkarni[b], Raghunath Tewari[c,1,*], N. V. Vinodchandran[d,1]

[a] *Chennai Mathematical Institute, Chennai, India.*
[b] *University of Chicago, Chicago, USA.*
[c] *Indian Institute of Technology Kharagpur, Kharagpur, India.*
[d] *University of Nebraska – Lincoln, Lincoln, USA*

Abstract

We investigate the space complexity of certain perfect matching problems over bipartite graphs embedded on surfaces of constant genus (orientable or non-orientable). We show that the problems of deciding whether such graphs have (1) a perfect matching or not and (2) a unique perfect matching or not, are in the log-space complexity class *Stoic Probabilistic Log-space* (SPL). Since SPL is contained in the log-space counting classes $\oplus$L (in fact in Mod_kL for all $k \geq 2$), $\mathrm{C}_=$L, and PL, our upper bound places the above-mentioned matching problems in these counting classes as well. We also show that the search version, computing a perfect matching, for this class of graphs can be performed by a log-space transducer with an SPL oracle. Our results extend the same upper bounds for these problems over bipartite planar graphs known earlier.

As our main technical result, we design a log-space computable and polynomially bounded weight function which isolates a minimum weight perfect matching in bipartite graphs embedded on surfaces of constant genus. We use results from algebraic topology for proving the correctness of the weight function.

*Corresponding author

Email addresses: sdatta@cmi.ac.in (Samir Datta), raghav@cs.uchicago.edu (Raghav Kulkarni), raghunath@cse.iitkgp.ernet.in (Raghunath Tewari), vinod@cse.unl.edu (N. V. Vinodchandran)

[1]Research supported in part by NSF grants CCF-0830730 and CCF-0916525.

Keywords: space complexity, perfect matching, topological graph theory

1. Introduction

The *perfect matching* problem and its variations are one of the most well-studied problems in theoretical computer science. Research in understanding the inherent complexity of computational problems related to matching has lead to important results and techniques in complexity theory and elsewhere in theoretical computer science. However, even after decades of research, the exact complexity of many problems related to matching is not yet completely understood.

We investigate the *space complexity* of certain well studied perfect matching problems over bipartite graphs. We prove new uniform space complexity upper bounds on these problems for *graphs embedded on surfaces of constant genus*. We prove our upper bounds by solving the technical problem of 'deterministically isolating' a perfect matching for this class of graphs.

Distinguishing a single solution out of a set of solutions is a basic algorithmic problem with many applications. The *Isolation Lemma* due to Mulmuley, Vazirani, and Vazirani provides a general randomized solution to this problem. Let $\mathcal{F}$ be a non-empty set system on $U = \{1, \ldots, n\}$. The Isolation Lemma says, for a random weight function on U (bounded by $n^{O(1)}$), with high probability there is a *unique* set in $\mathcal{F}$ of minimum weight [1]. This lemma was originally used to give an elegant RNC algorithm for constructing a maximum matching (by isolating a minimum weight perfect matching) in general graphs. Since its discovery, the Isolation Lemma has found many applications, mostly in discovering new randomized or non-uniform upper bounds, via isolating minimum weight solutions [1, 2, 3, 4]. Clearly, derandomizing the Isolation Lemma in sufficient generality will improve these upper bounds to their deterministic counterparts and hence will be a major result. Unfortunately, recently it is shown that such a derandomization will imply certain circuit lower bounds and hence is a difficult task [5].

Can we bypass the Isolation Lemma altogether and deterministically isolate minimum weight solutions in specific situations? Recent results illustrate that one may be able to use the structure of specific computational problems under consideration to achieve non-trivial deterministic isolation. In [6], the authors used the structure of directed paths in planar graphs to prescribe a simple weight function that is computable deterministically in logarithmic

space with respect to which the minimum weight directed path between any two vertices is unique. In [7], the authors isolated a perfect matching in planar bipartite graphs. In [8], the authors give a generalized weight function that isolates directed paths in planar graphs and perfect matchings in undirected bipartite planar graphs by bypassing the use of grid graphs altogether. In this paper we extend the deterministic isolation technique of [7, 8] to isolate a minimum weight perfect matching in bipartite graphs embedded on constant genus surfaces. This is more interesting in light of the fact that for constant genus graphs even the existence of a polynomially bounded weight function, that isolates a minimum weight perfect matching, was not known earlier. As a future direction it would be interesting to consider the general bipartite graph $K_{n,n}$, and prove the existence of a polynomially bounded weight function that isolates a minimum weight perfect matching in this case.

Our Contribution

Let G be a bipartite graph and let $\overrightarrow{G}$ be the directed graph obtained by replacing every undirected edge $\{u, v\}$ of G with the directed edges (u, v) and (v, u). The main technical contribution of the present paper can then be stated (semi-formally) as follows.

- **Main Technical Result.** Given an embedding of a undirected bipartite constant genus graph G, there is a log-space matching preserving reduction f, and a log-space computable, polynomially bounded, skew-symmetric weight function w for the class of directed graphs, so that the weight of any simple cycle in $\overrightarrow{f(G)}$ with respect to w is non-zero.

  We use this result to establish (using known techniques) the following new upper bounds. Refer to the next section for definitions.

- **New Upper Bounds.** For bipartite graphs, combinatorially embedded on surfaces of constant genus, we show that (a) checking if the graph has a perfect matching is in SPL, (b) checking if the graph has a unique perfect matching is in SPL, and (c) constructing a perfect matching (if one exists) is in $\mathsf{FL}^{\mathsf{SPL}}$.

SPL is a log-space complexity class that was first studied by Allender, Reinhardt, and Zhou [4]. This is the class of problems reducible to the determinant with the promise that the determinant is either 0 or 1. In [4], the authors show, using a non-uniform version of Isolation Lemma, that perfect

matching problem for general graphs is in a 'non-uniform' version of SPL. In [7], using the above-mentioned deterministic isolation, the authors show that for planar bipartite graphs, DECISION-BPM is in fact in SPL (uniformly). Recently, Hoang showed that for graphs with polynomially many matchings, perfect matchings and many related matching problems are in SPL [9]. SPL is contained in log-space counting classes such as $\mathrm{Mod}_k\mathrm{L}$ for all $k \geq 2$ (in particular in $\oplus\mathsf{L}$), PL, and $\mathrm{C}_=\mathrm{L}$, which are in turn contained in NC^2. PL and $\mathrm{C}_=\mathrm{L}$ contain the class non-deterministic log-space or NL. But no relation is known between the classes $\mathrm{Mod}_k\mathrm{L}$ (for all $k \geq 2$), PL, and $\mathrm{C}_=\mathrm{L}$. Thus the upper bound of SPL that we prove implies that the problems DECISION-BPM and UNIQUE-BPM for the class of graphs we study are in these log-space counting classes as well.

The techniques that we use in this paper can also be used to isolate directed paths in graphs on constant genus surfaces. This shows that the reachability problem for this class of graphs can be decided in the unambiguous class UL (a subclass of NL), extending the results of [6]. But this upper bound is already known since recently Kynčl and Vyskočil show that reachability for bounded genus graphs log-space reduces to reachability in planar graphs [10].

Matching problems over graphs of low genus have been of interest to researchers, mainly from a parallel complexity viewpoint. For the class of bipartite planar graphs, it was shown that finding a perfect matching can be done in NC [11]. In [12], the authors present an NC^2 algorithm for computing a perfect matching for bipartite graphs on surfaces of $O(\log n)$ genus (readers can also find an account of known parallel complexity upper bounds for matching problems over various classes of graphs in their paper). However, the space complexity of matching problems for graphs of low genus has not been investigated before. The present paper takes a step in this direction.

Proof Outline

We assume that the graph G is presented as a combinatorial embedding on a surface (orientable or non-orientable) of genus g, where g is a constant. This is a standard assumption when dealing with graphs on surfaces, since it is NP-complete to check whether a graph has genus $\leq g$ [13]. We first give a sequence of two reductions to get, from G, a graph G' with an embedding on a genus g 'polygonal schema in normal form'. These two reductions work for both orientable and non-orientable cases. At this point we take care of the non-orientable case by reducing it to the orientable case. These reductions

are matching preserving, bipartiteness preserving and computable in log-space. Finally, for $\overrightarrow{G'}$ (the directed version of G'), we prescribe a set of $2g'+1$ weight functions (where $g' = O(g)$), $\mathcal{W} = \{w_i\}_{1 \leq i \leq 2g'+1}$, that are polynomially bounded and computable in log-space, so that for any cycle C in $\overrightarrow{G'}$, there is a weight function $w_i \in \mathcal{W}$ with respect to which the weight of C is non-zero. Since g' is constant, we can take a linear combination of the elements in $\mathcal{W}$, for example $\sum_{w_i \in \mathcal{W}} w_i \times (n^c)^i$ (where n is the number of vertices) for some fixed constant c (say $c = 4$), to get a single weight function (which again is polynomially bounded and log-space computable) with respect which the weight of any cycle is non-zero.

The intuition behind these weight functions is as follows (for some of the definitions, refer to later sections). The set $\mathcal{W}$ is a disjoint union $\mathcal{W}_1 \cup \mathcal{W}_2 \cup \{w\}$ of the sets of weight functions $\mathcal{W}_1$, $\mathcal{W}_2$, and $\{w\}$. Consider a graph G embedded on a fundamental polygon with $2g$ sides. There are two types cycles in G: *surface separating* and *surface non-separating*. A basic theorem from algebraic topology implies that a surface non-separating cycle will intersect at least one of the sides of the polygon an odd number of times. This leads to $2g$ weight functions in $\mathcal{W}_1$ to take care of all the surface non-separating cycles. There are two types of surface separating cycles: (a) ones which completely lie inside the polygon and (b) the ones which cross some boundary. Cycles of type (a) behave exactly like cycles in the plane so the weight function w designed for planar graphs works (from [7, 8]). For dealing with cycles of type (b), we first prove that if such a cycle intersects a boundary, it should alternate between 'coming in' and 'going out'. This leads to $2g$ weight functions in $\mathcal{W}_2$ which handle all type (b) cycles.

Figure 1 gives a pictorial view of the components involved in the proof of our main technical result.

Organization of the Paper

The rest of the paper is organized as follows. In Section 2 we give the necessary definitions and state results from earlier work, that we use in this paper. In Section 3 we give log-space reductions that takes a combinatorial embedding of a graph G (orientable or non-orientable) embedded on a surface of genus g, and outputs an orientable graph H embedded on the polygonal schema of a surface of genus g' (where $g' = O(g)$), such that G has a perfect matching if and only if H has a perfect matching and given a perfect matching in H we can construct a perfect matching in G in log-space. In Section 4 we state and prove the complexity upper bounds that we discussed earlier.

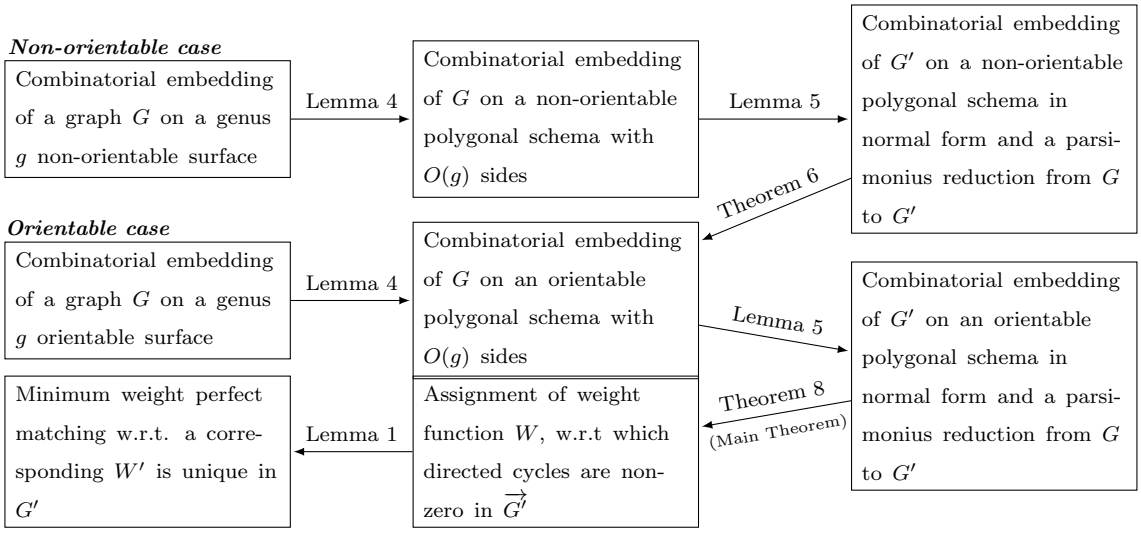


Figure 1: Outline of the steps. Note that all reductions are log-space computable and preserves matchings and bipartiteness.

2. Preliminaries

For a positive integer n, $[n]$ is defined to be the set $\{1, 2, \ldots, n\}$.

2.1. Topological graph theory

We introduce the necessary terminology from algebraic topology. For a more comprehensive understanding of this topic, refer to any standard algebraic topology book such as [14].

A *2-manifold* is a topological space such that every point has an open neighborhood homeomorphic to $\mathbb{R}^2$ and two distinct points have disjoint neighborhoods. A 2-manifold is often called a *surface*. The *genus* of a surface Γ is the maximum number g, such that there exist g pairwise disjoint cycles $C_1, C_2, \ldots, C_g$ on Γ, whose removal still leaves $\Gamma \setminus (C_1 \cup C_2 \cup \ldots \cup C_g)$ connected. A surface is called *orientable* if it has two distinct sides, else it is called *non-orientable*. A cycle C in Γ is said to be *non-separating* if there exists a path between any two points in $\Gamma \setminus C$, else it is called *separating*.

A *polygonal schema* of a surface Γ is a polygon with $2g'$ directed sides, such that the sides of the polygon are partitioned into g' classes, each class containing exactly two sides and glueing the two sides of each equivalence class gives the surface Γ (upto homeomorphism). A side in the ith equivalence

class is labeled σ_i or $\bar{\sigma_i}$ depending on whether it is directed clockwise or anti-clockwise respectively. The *partner* of a side σ is the other side in its equivalence class. By an abuse of notation, we shall sometimes refer to the symbol of a side's partner, as the partner of the symbol. Frequently we will denote a polygonal schema as a linear ordering of its sides moving in a clockwise direction, denoted by X. For a polygonal schema X, we shall refer to any polygonal schema which is a cyclic permutation, or a reversal of the symbols, or a complementation (σ mapped to $\bar{\sigma}$ and vice versa) of the symbols, as being the same as X. A polygonal schema is called orientable (resp. non-orientable) if the corresponding surface is orientable (resp. non-orientable). Note that g' need not be equal to the genus of Γ.

Definition 1. An orientable polygonal schema is said to be in *normal form* if it is in one of the following forms:

$$\sigma_1\tau_1\bar{\sigma_1}\bar{\tau_1}\sigma_2\tau_2\bar{\sigma_2}\bar{\tau_2}\ldots\sigma_m\tau_m\bar{\sigma_m}\bar{\tau_m} \tag{2.1}$$

$$\sigma\bar{\sigma} \tag{2.2}$$

A non-orientable polygonal schema is said to be in normal form if it is of one of the following forms:

$$\sigma\sigma X \tag{2.3}$$

$$\sigma\tau\bar{\sigma}\tau X \tag{2.4}$$

where, X is a string representing an orientable schema in normal form (i.e. like Form 2.1 or 2.2 above) or possibly an empty string.

Table 1 lists some common surfaces and their polygonal schema in normal forms.

Name	Orientability	Genus	Polygonal schema in normal form
Sphere	Orientable	0	$\sigma\bar{\sigma}$
Torus	Orientable	1	$\sigma\tau\bar{\sigma}\bar{\tau}$
Projective Plane	Non-orientable	1	$\sigma\sigma$
Klein bottle	Non-orientable	2	$\sigma\tau\bar{\sigma}\tau$

Table 1: Common surfaces and their properties

We denote the polygonal schema in the normal form of a surface Γ as $\Lambda(\Gamma)$. We will refer to two orientable symbols σ, τ which form the following contiguous substring: $\sigma\tau\bar{\sigma}\bar{\tau}$ as being clustered together while a non-orientable symbol σ which occurs like $\sigma\sigma$ as a contiguous subtring is said to form a *pair*. Thus, in the first and third normal forms above all symbols are clustered. The *connected sum* of two surfaces Γ_1 and Γ_2 is defined to be the surface whose polygonal schema is the concatenation of $\Lambda(\Gamma_1)$ and $\Lambda(\Gamma_2)$.

The first normal form represents a connected sum of tori and the third of a projective plane and tori. In the fourth normal form all but one of the orientable symbols are clustered while the only non-orientable symbol is sort of clustered with the other orientable symbol. This form represents a connected sum of a Klein bottle and tori. The second normal form represents a sphere.

We next introduce the concept of $\mathbb{Z}_2$-homology. Given a 2-manifold Γ, a *1-cycle* is a collection of closed curves in Γ. Given two sets A and B, the *symmetric difference* of A and B, denoted as $A\Delta B$ is the set of elements that belong to A or B but now both. The set of 1-cycles forms an Abelian group, denoted as $\mathcal{C}_1(\Gamma)$, under the symmetric difference operation, Δ. Therefore the identity element of the group is the empty collection and the inverse of an element is itself. Two 1-cycles C_1, C_2 are said to be *homologically equivalent* if $C_1\Delta C_2$ forms the boundary of some region in Γ. Observe that this is an equivalence relation. Then the *first homology group* of Γ, $H_1(\Gamma)$, is the set of equivalence classes of 1-cycles. In other words, if $\mathcal{B}_1(\Gamma)$ is defined to be the subset of $\mathcal{C}_1(\Gamma)$ that are homologically equivalent to the empty set, then $H_1(\Gamma) = \mathcal{C}_1(\Gamma)/\mathcal{B}_1(\Gamma)$.[2] If Γ is a genus g surface then $H_1(\Gamma)$ is generated by a system of $2g$ 1-cycles $\mathcal{L} = \{L_1, L_2, \ldots, L_{2g}\}$, and every L_i $(1 \leq i \leq 2g)$ contains a closed simple curve Q_i, such that $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_{2g}\}$ generates $H_1(\Gamma)$ as well. Moreover the elements of $\mathcal{Q}$ have only one point in common, and whose complement is homeomorphic to a topological disk. Such a disk is also referred to as the *fundamental polygon* of Γ.

A *combinatorial embedding* of a graph is a map which given a vertex, gives a cyclic ordering of the edges around the vertex. An undirected graph G is said to be embedded on a surface Γ if it can be drawn on Γ so that no two edges cross. We assume that the graph is given together with a combinatorial

[2]For examples of the homology group of various surfaces please refer to any standard text on algebraic topology such as [14].

embedding on a surface of constant genus. Refer to the book by Mohar and Thomassen [15] for details. The genus of a graph G is the minimum number g such that G has an embedding on a surface of genus g (an embedding where every face of G is homeomorphic to a disc). Such an embedding is also called a *2-cell embedding*. A genus g graph is said to be orientable (non-orientable) if the surface is orientable (non-orientable).

Definition 2. The *polygonal schema of a graph* G is a combinatorial embedding given on the polygonal schema of some surface Γ together with the ordered set of vertices on each side of the polygon. Formally it is a tuple $(\phi, \mathcal{S})$, where ϕ is a cyclic ordering of the edges around a vertex (also known as the *rotation system* of G) and $\mathcal{S} = (S_1, S_2, \ldots, S_{2g})$ is the cyclic ordering of the directed sides of the polygon. Each S_i is an ordered sequence of the vertices (which we shall refer to as *boundary vertices*), from the tail to the head of the side S_i and is paired with some other side, say S_i' in $\mathcal{S}$, such that the jth vertex of S_i (say from the tail of S_i) is the same as the jth vertex of S_i' (from the tail of S_i'). Moreover, no edge in G crosses any of the sides S_i $(1 \leq i \leq 2g)$ and every edge can have at most one of its endpoints as a boundary vertex.

An example of the polygonal schema of the complete graph on 5 vertices (K_5) on the torus is shown in Figure 2. In the following definition we

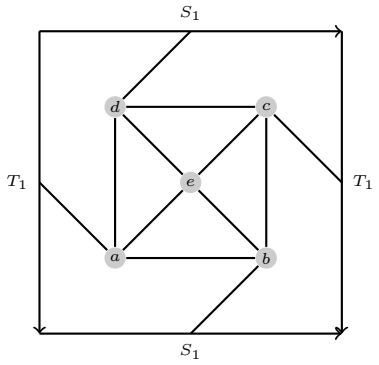


Figure 2: Polygonal schema of K_5 on the fundamental polygon of the torus.

formalize the class of such graphs.

Definition 3. - Let BIPARTITEGENUS$_g$ be the class of genus g bipartite graphs G, given together with a rotation system (or *combinatorial embedding*) of G on a surface of genus g.

- Let $g \geq 0$ be an integer. Then $\text{BDDBIPGENUS}_g = \cup_{0 \leq i \leq g} \text{BIPARTITEGENUS}_i$.

We also define the directed version of the above class of graphs as follows:

Definition 4. - Given G, define $\overrightarrow{G}$ to be the directed graph formed by replacing every edge $\{u, v\}$ in G with the directed edges (u, v) and (v, u). Let $\text{DIRBIPARTITEGENUS}_g$ be the class of directed graphs $\overrightarrow{G}$, such that G is in BIPARTITEGENUS_g.

- Let $g \geq 0$ be an integer. Then $\text{DIRBDDBIPGENUS}_g = \cup_{0 \leq i \leq g} \text{DIRBIPARTITEGENUS}_i$.

2.2. Complexity Theory

For a nondeterministic machine M, let $acc_M(x)$ and $rej_M(x)$ denote the number of accepting computations and the number of rejecting computations respectively on an input x. Denote $gap_M(x) = acc_M(x) - rej_M(x)$.

Definition 5. A language L is in SPL if there exists a log-space bounded nondeterministic machine M so that for all inputs x, $gap_M(x) \in \{0, 1\}$ and $x \in L$ if and only if $gap_M(x) = 1$. $\mathsf{FL}^{\mathsf{SPL}}$ is the class of functions computed by a log-space machine with an SPL oracle. UL is the class of languages L, decided by a nondeterministic log-space machine (say M), such that for every string in L, M has exactly one accepting path and for a string not in L, M has no accepting path.

Alternatively, we can define SPL as the class of problems log-space reducible to the problem of checking whether the determinant of a matrix is 0 or not under the promise that the determinant is either 0 or 1. For definitions of other complexity classes refer to any standard textbooks such as [16, 17]. All reductions discussed in this paper are log-space reductions.

Given an undirected graph $G = (V, E)$, a *matching* M is a subset of E such that no two edges in M have a vertex in common. A *maximum matching* is a matching of maximum cardinality. M is said to be a perfect matching if every vertex is an endpoint of some edge in M.

Definition 6. We define the following computational problems related to matching:

- DECISION-BPM : Given a bipartite graph G, checking if G has a perfect matching.

- SEARCH-BPM: Given a bipartite graph G, constructing a perfect matching, if one exists.

- UNIQUE-BPM: Given a bipartite graph G, checking if G has a unique perfect matching.

UNIQUE-BPM have been studied earlier (see [7]), and we include it here for the sake of completeness.

2.3. Necessary Prior Results

Lemma 1 is an adaptation of Theorem 6 in [8] for the case of constant genus bipartite graphs.

Lemma 1. *Let $g \geq 0$ be some fixed constant and let w be a polynomially bounded, skew-symmetric edge weight function defined for the class of graphs* DIRBDDBIPGENUS$_g$*, such that for any graph $G \in$* DIRBDDBIPGENUS$_g$ *and any cycle $C \in G$, $w(C) \neq 0$. Then in log-space we can construct a polynomially bounded, edge weight function w' for the class of graphs* BDDBIPGENUS$_g$*, such that for any graph $H \in$* BDDBIPGENUS$_g$ *the minimum weight perfect matching in H is unique with respect to w'.*

Proof. Let H be a graph in BDDBIPGENUS$_g$. By definition, the graph $\overrightarrow{H}$ is in DIRBDDBIPGENUS$_g$ and hence there exists a polynomially bounded weight function w with respect to which every cycle in $\overrightarrow{H}$ has non-zero weight. Using Reingold's reacability algorithm [18], construct a bipartition (L, R) of H. Now for an edge $e = \{u, v\}$ in H such that $u \in L$ and $v \in R$, set $w'(e) = w(\overrightarrow{e})$ where $\overrightarrow{e}$ is the directed edge (u, v) in $\overrightarrow{H}$.

Now suppose H has two distinct minimum weight perfect matchings, M_1 and M_2, with respect to w'. Then the symmetric difference of M_1 and M_2 is a collection of disjoint, even length, simple cycles, where the edges of the cycle alternate between the matchings M_1 and M_2. Since M_1 and M_2 are distinct, there is at least one such cycle. Let $C = (v_0, v_1, \ldots v_{2k-1}, v_0)$ be such a cycle. Let $e_i = (v_i, v_{(i+1) \pmod{2k}})$ for $0 \leq i \leq 2k-1$. Without loss of generality assume, $v_0 \in L$ and the edge e_0 is in M_1. Therefore if i is even (resp. odd), then $e_i \in M_1$ (resp. $e_i \in M_2$) and $\overrightarrow{e_i}$ is directed from L to R (resp. from R to L). Thus $w'(e_{2i}) = w(\overrightarrow{e_{2i}})$ and $w'(e_{2i+1}) = -w(\overrightarrow{e_{2i+1}})$ for $0 \leq i \leq k-1$, due to skew symmetry of w.

The weight of the restriction of M_1 to C, $w'(M_1 \cap C) = \sum_{i=1}^{k} w'(e_{2i-1})$. Similarly $w'(M_2 \cap C) = \sum_{i=1}^{k} w'(e_{2i})$. Now,

$$\begin{aligned} w'(M_1 \cap C) - w'(M_2 \cap C) &= \sum_{i=1}^{k} w'(e_{2i-1}) - \sum_{i=1}^{k} w'(e_{2i}) \\ &= \sum_{i=1}^{k} w(\overrightarrow{e_{2i-1}}) + \sum_{i=1}^{k} w(\overrightarrow{e_{2i}}) = \sum_{i=1}^{2k} w(\overrightarrow{e_i}) \\ &\neq 0. \end{aligned}$$

Therefore either $M_1 \cap C$ or $M_2 \cap C$ has higher weight with respect to w'. Without loss of generality assume it is M_2. Thus we get a perfect matching $M' = M_2 \setminus (M_2 \cap C) \cup (M_1 \cap C)$ in H of lesser weight, which is a contradiction. □

Lemma 2 ([4]). *For any weighted graph G assume that the minimum weight perfect matching in G is unique and also for any subset of edges $E' \subseteq E$, the minimum weight perfect matching in $G \setminus E'$ is also unique. Then deciding if G has a perfect matching is in* SPL. *Moreover, computing the perfect matching (in case it exists) is in* $\mathsf{FL}^{\mathsf{SPL}}$.

Sketch of proof. Let w_{max} and w_{min} be the maximum and minimum possible weights respectively, that an edge in G can get. Note that w_{max} and w_{min} are polynomially bounded in the size of the graph G. Then any perfect matching in G will have a weight from the set $W = \{k : k \in \mathbb{Z}, n \cdot w_{min} \leq k \leq n \cdot w_{max}\}$. Similar to [4], there exists a GapL function f, such that if for some $k \in W$, G has a perfect matching of weight k, then $(f(G,k))^2 = 1$ and otherwise for each k, $f(G,k) = 0$. Note that in [4] the authors actually give a GapL/poly function since the weight function for the graphs (which are unweighted to begin with) are required as an advice in their GapL machine. Here we consider weighted graphs, thus eliminating the need for any advice. Now consider the function

$$g(G) = 1 - \prod_{k} \left(1 - (f(G,k))^2\right).$$

By definition, $g(G) = 1$ if G has a perfect matching, else it is 0.

To compute a perfect matching in G, we will construct a log-space transducer that makes several queries to the function f defined above. For a graph G' having a unique minimum weight perfect matching (say M'), the weight of M' can be computed by iteratively querying the function $f(G', k)$ for values of $k \in W$ in an increasing order, starting from $n \cdot w_{min}$. The value k, for

which the function outputs a non-zero value for the first time, is the weight of M'. We denote this weight by $w_{G'}$. First compute w_G. For an e in G, define the graph $G^{-e} = G \setminus \{e\}$. Now compute $w_{G^{-e}}$ for every edge e in G. Output the edges e for which $w_{G^{-e}} > w_G$. The set of edges output by this procedure comprise a perfect matching (in fact the minimum weight perfect matching) because deleting an edge in this set had increased the weight of the minimum weight perfect matching in the resulting graph. □

3. Embedding on a Polygonal Schema in Normal Form

In this section we give a log-space reduction that takes a graph $G \in \textsc{BddBipGenus}_g$ for some fixed constant g and outputs a graph $H \in \textsc{BddBipGenus}_{g'}$ (where $g' = O(g)$), such that H is given as an embedding on the polygonal schema in normal form of an orientable surface of genus g', and no edge of H has both its endpoints on the boundary of the polygon. Moreover, there is a perfect matching in G if and only if there is a perfect matching in H, and given a perfect matching in H, one can construct a perfect matching in G in log-space.

A crucial point to be noted here is that irrespective of the fact that G is orientable or non-orientable, H is always orientable, since its embedding is given in an orientable surface.

3.1. Combinatorial Embedding to a Polygonal Schema

Here we show that given a graph $G \in \textsc{BddBipGenus}_g$, in log-space we can output the polygonal schema of a graph G', such that there is a perfect matching in G if and only if there is a perfect matching in G' and given a perfect matching in G' one can construct a perfect matching in G in log-space.

To prove this, we recall a lemma from [19].

Lemma 3 ([19])**.** *Let G be a graph embedded on a surface of genus $g > 0$, and let T be a spanning tree of G. Then there is an edge $e \in E(G)$ such that $T \cup \{e\}$ contains a non-separating cycle.*

Note that in [19] the graph was required to be embedded on an orientable surface but the proof did not use this requirement. To give a polygonal schema embedding, we first define a "cut" operation on the graph.

Given a cycle (or path) C in an embedded graph G, define by G✂C the graph constructed by "cutting" the edges incident on the cycle from the right. In other words, the neighbors of $u \in C$ (which are not on the cycle)

can be partitioned into two sets, arbitrarily called left and right. For every neighbor v of u which lies to the right of C, replace the the edge (u, v) with three edges (u, x_{uv}), (y_{uv}, v') and (v', v) where x_{uv} and y_{uv} are copies of the same vertex. Replacing an edge with a path of length 3 preserves perfect matchings in the newly constructed graph. We add *spurious edges* between adjacent x_{uv}'s and adjacent y_{uv}'s along the cut and label all the newly formed spurious edges with the label L_C along the left set and L_C^{-1} along the right set as shown in Figure 3. Note that the spurious edges are only added for ease of understanding while constructing the polygonal schema of the graph. Once the polygonal schema is constructed, we remove all spurious edges.

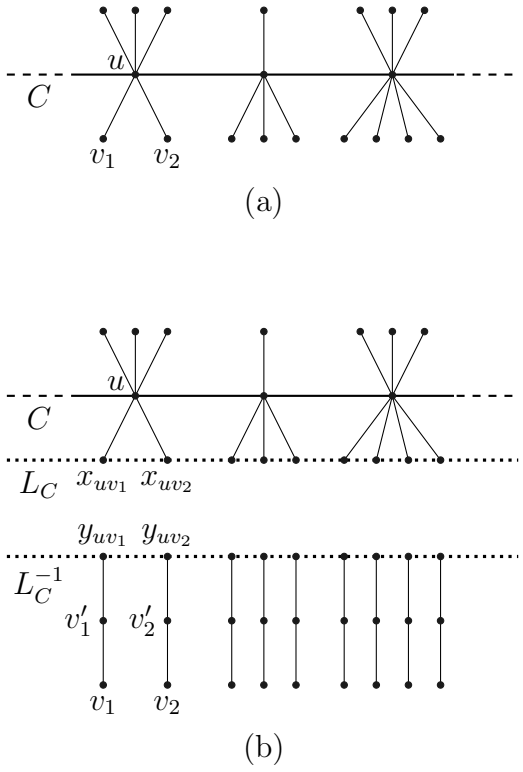


Figure 3: An example of the cut operation $\bowtie$, cutting graph G along cycle (or path) C. (a) Part of graph G and cycle C. (b) Part of the resulting graph $G \bowtie C$, with the dotted lines representing the spurious edges.

Also, if C is a path, its endpoints will lie on two paths. Consider the first path – if the two edges on either side of C on this path have the same label L_1. This can be broken into two cases – firstly, if the left and right side of

this endpoint are the same (in other words, the path is a cycle). In this case, we just keep the same label L_1. When the left and right side of this endpoint are distinct, we will need to split the label into two or three new labels as detailed below and similarly for the other path and common label L_2. We will only describe the case when L_1, L_2 are both defined – the other cases are similar and simpler.

First assume that $L_1 \neq L_2$ and $L_1 \neq L_2^{-1}$. Then we will split remove labels L_1, L_2 and replace them by four new labels say $L'_{1,C}, L''_{1,C}$ and $L'_{2,C}, L''_{2,C}$, respectively for the two sides of the intersection. If, on the other hand, L_1 is the same as L_2 or its inverse – then there are two subcases. Firstly, if the path C is between two copies of the same vertex then we replace L_1 by two new labels $L'_{1,C}, L''_{1,C}$ one for either side of the cut. L_2 being a copy or an inverse copy of L_1 splits automatically. The second case is if C is between two distinct points on two copies or inverse copies. Then we split L_1 into three parts according to the two points. The rotation system is modified appropriately. We illustrate this operation in Figure 4.

Notice that in the process of cutting, for every new label L_C we are adding at most 4 new labels.

Lemma 4. *Let g be a fixed constant. Then given a graph $G \in$ BddBipGenus$_g$ we can find a polygonal schema of a corresponding graph $H \in$ BddBipGenus$_g$ (in fact having the same genus) in log-space such that G has a perfect matching if and only if H has a perfect matching. Moreover, given a perfect matching in H, one can construct a perfect matching in G in log-space.*

Proof. Given a graph G_i embedded on a surface, potentially with spurious edges, we can find C_{i+1}, a non-separating cycle (which does not use a spurious edge) by invoking Lemma 3. Define G_{i+1} to be $G_i \ltimes C_{i+1}$.

Starting with $G_0 = G$ of genus g and repeating the above operation at most g times, we get a planar graph H with at most $2g$ spurious faces (which consist of spurious edges).

Now find a spanning tree of this graph which does not use a spurious edge – that such a tree exists follows from noticing that the graph without spurious edges is still connected. Find a tree path connecting any two spurious faces. Cut along this path to combine the two spurious faces into one larger spurious face. Repeat the operation till all the spurious faces are merged into one spurious face and re-embed the planar graph so that it forms the external face. Finally, remove all spurious edges to get the desired polygonal schema of the given graph. Note that the final graph has a perfect matching if and only

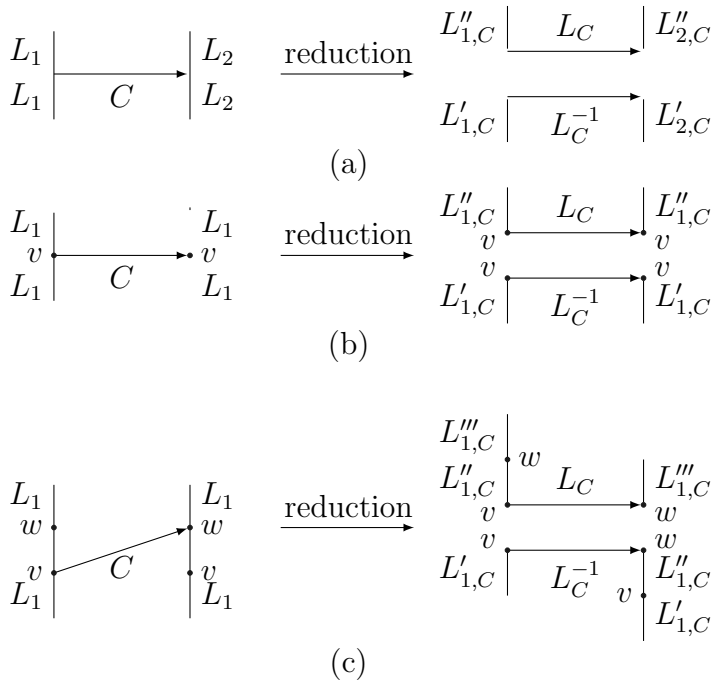


Figure 4: Cutting along a path C when (a) $L_1 \neq L_2$ and $L_1 \neq L_2^{-1}$, (b) $L_1 = L_2$ or $L_1 = L_2^{-1}$ and C is between copies of the same vertex v, and (c) $L_1 = L_2$ or $L_1 = L_2^{-1}$ and C is between distinct vertices v and w.

if the G had a perfect matching, since each cut operation preserves matchings. Also, given a perfect matching in the final graph, we can construct a perfect matching in G in log-space.

It is easy to see that the procedure above can be performed in log-space, provided that g is constant. □

3.2. Normalizing a Polygonal Schema

We next give a log-space algorithm that takes an embedding of a graph on a polygonal schema and outputs an embedding of the graph on a polygonal schema in normal form. We adapt the algorithmic proof of Brahana-Dehn-Heegaard (BDH) [20, 21] classification theorem as described in Vegter-Yap [22] so that it runs in log-space for constant genus graphs. The algorithm starts with a polygonal schema having $2m$ sides and uses certain transformations $O(m)$ times to yield a polygonal schema in normal form.

Lemma 5. *Let g be a fixed constant. Given the polygonal schema of a graph $G \in$ BddBipGenus$_g$, we can find a polygonal schema in normal form of a corresponding graph $H \in$ BddBipGenus$_g$ in log-space, such that G has a perfect matching if and only if H has a perfect matching. Moreover, given a perfect matching in H, one can construct a perfect matching in G in log-space.*

Proof. Consider the following 6 transformations of a polygonal schema:

A. Replace $X\sigma\bar{\sigma}$ by X (Example given in Figure 5).

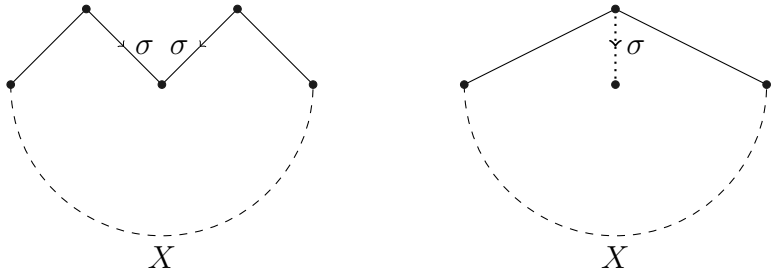


Figure 5: Reduction A (pasting along σ)

B. Replace $\sigma\tau X\bar{\tau}Y$ by $\rho X\bar{\rho}\sigma Y$ (Example given in Figure 6).

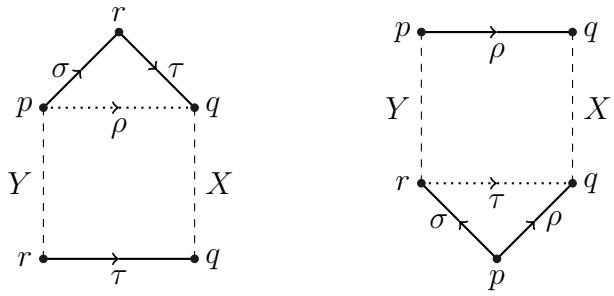


Figure 6: Reduction B (Cutting along ρ followed by pasting along τ). Note that the number of vertices in the equivalence class of r, reduces by 1.

C. Replace $\sigma X\sigma Y$ by $\tau\tau Y^*X$, where Y^* is reverse complement of Y (Example given in Figure 7).
D. Replace $\sigma X\tau Y\bar{\sigma}U\bar{\tau}V$ by $\rho\pi\bar{\rho}\bar{\pi}UYXV$ (Example given in Figure 8).
E. Replace $\sigma_1\sigma_1X\sigma_2\sigma_3\bar{\sigma}_2\bar{\sigma}_3Y$ by $\tau_1\tau_1\tau_2\tau_2\tau_3\tau_3XY$ (Example given in Figure 9).

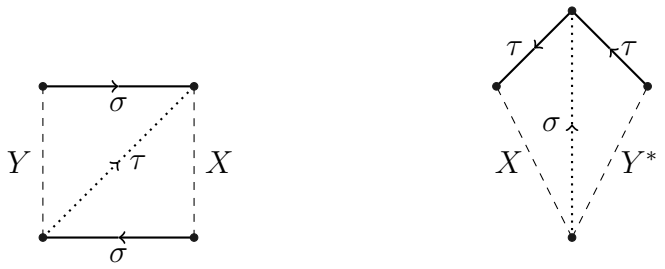


Figure 7: Reduction C (Cutting along τ followed by pasting along ρ)

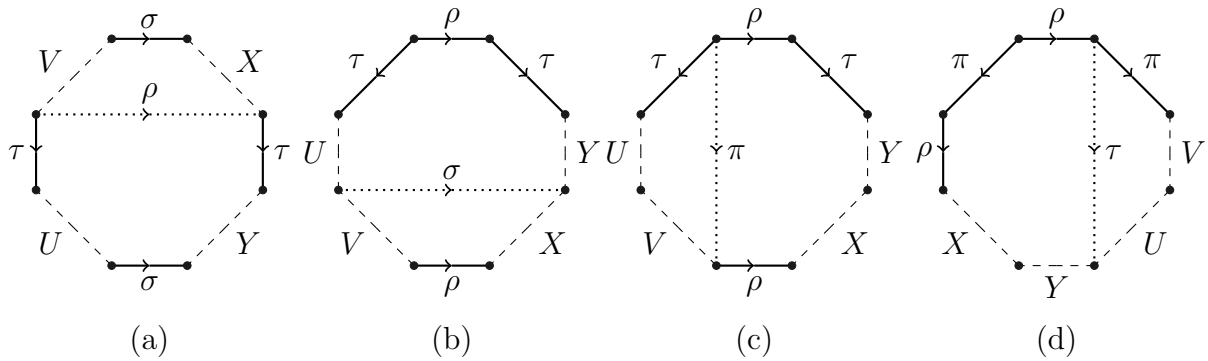


Figure 8: Reduction D (a) Cutting along ρ. (b) Pasting along σ. (c) Cutting along π. (d) Pasting along τ.

F. Replace $\sigma\sigma\tau\tau X$ by $\sigma\rho\bar{\sigma}\rho X$ (Example given in Figure 10).

The procedure to construct the polygonal schema in normal form is as follows:

1. Use reductions A,B,C several times to ensure that all the sides of the polygonal schema have a common endpoint.
2. (a) Orientable case: Use transform D repeatedly to bring the polygon in normal form.
   (b) Non-orientable case:
   - Use reductions C,D to convert the schema into a form where the orientable symbols are clustered and non-orientable symbols are paired.
   - Use reduction E repeatedly (in the forward direction) to eliminate all orientable symbols.

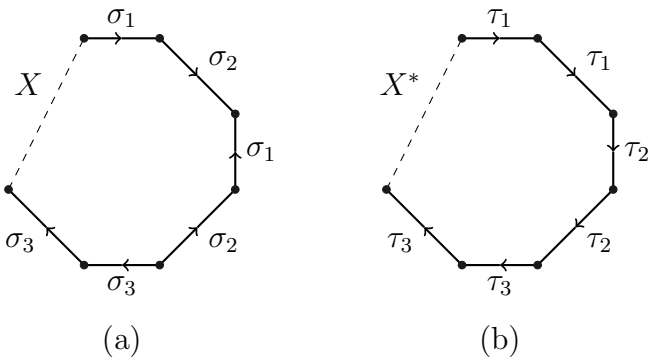


Figure 9: Reduction E (a) Initial polygonal schema (b) Polygonal schema obtained after applying Reduction E (replacing (i) σ_1 with $\bar{\tau_3}\bar{\tau_2}$, (ii) σ_2 with $\tau_1\tau_2$ and (iii) σ_3 with $\tau_1\tau_2\tau_3X^*$).

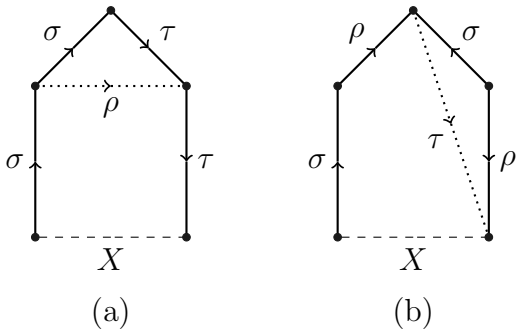


Figure 10: Reduction F (a) Cutting along ρ. (b) Pasting along τ.

- Use Reduction E in the *reverse* direction repeatedly to eliminate all but at most one non-orientable symbol.
- Use Reduction F, if necessary, to ensure that there is at most one non-orientable symbol.

Possibly the only step requiring any explanation is the last one. We apply Reduction E in reverse with X as the empty string to replace three non-orientable symbols by two orientable ones forming a cluster of 4 and a single non-orientable one which forms a pair. The way we apply the reduction, ensures that both the orientable and the non-orientable parts are contiguous.

Finally we will be left with a string in one of the first two normal forms or a string of the form $\sigma\sigma\tau\tau X$ (where X is an orientable schema in normal form) in which case Reduction F is applicable.

To see that the above procedure can be carried out in log-space it suffices to prove that each of the above reductions can be carried out in log-space, the number of reductions is bounded by a constant and we can decide in log-space when to carry out a reduction.

The Vegter-Yap paper does careful book-keeping in order to ensure that the number of operations in Step 1 is linear in the original genus. We can alternatively, follow the brute force approach and keep on applying Reductions A,B,C while the sides of the polygon do not have a common end-point. This will require at most linear number of applications of the first two reductions.

Observe that for the orientable case, each application of reduction D reduces the number of unclustered symbols by two. Thus we are done in $O(m)$ applications of this reduction. Similarly, each application of reduction C reduces the number of unpaired non-orientable symbols by one and as before every application of reduction D reduces the number of unclustered orientable symbols by two. So in $O(m)$ steps all the orientable symbols are clustered and the non-orientable symbols are paired. Now every application of reduction E in the forward direction gets rid of two orientable symbols so in $O(m)$ steps all the orientable symbols are removed. Finally $O(m)$ applications of reduction E in reverse lead to removal of all but one non-orientable symbols.

To see that each of the steps is in log-space observe that each of the steps involves one or more of the following operations:

- find a path through the interior of the polygon between two points on its boundary,
- cut along a path, and
- paste two paired sides of (a cut) polygon together.

We know how to do the second operation in log-space while the third, being the reverse of the second one is even easier, since we just have to identify corresponding boundary vertices and then excise them out of the corresponding edge. The first operation is just an undirected reachability question in the graph (minus its boundary) hence is in log-space by Reingold's Theorem [18].

Finally, a determination of when to apply a particular reduction is easily seen to be in log-space for all but, possibly, reduction D. In this case, for an orientable symbol σ separated from its mate $\bar{\sigma}$ on both sides, sequentially test for each other symbol τ if it lies in one of the two stretches that σ and

its mate divide the schema into, while its mate $\bar{\tau}$ lies in the other. Having found the first such τ suffices to enable a use of the reduction. □

3.3. Reducing the Non-orientable case to the Orientable case

Let $G \in \text{BDDBIPGENUS}_g$ be a non-orientable graph. As a consequence of Lemma 5 we can assume that we are given a combinatorial embedding (say Π) of G on a (non-orientable) polygonal schema, say $\Lambda(\Gamma)$, in the normal form with $2g'$ sides. (Here g' is a function of g.)

Let $Y = (X_1, X_2)$ be the cyclic ordering of the labels of the sides of $\Lambda(\Gamma)$, where X_2 is the 'orientable part' and X_1 is the 'non-orientable part'. More precisely, for the polygonal schema in the normal form, we have: X_1 is either $\sigma\sigma$ (thus corresponds to the projective plane) or it is $\sigma\tau\bar{\sigma}\tau$ (thus corresponds to the Klein bottle). See Figure 11.

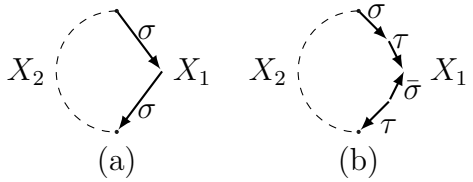
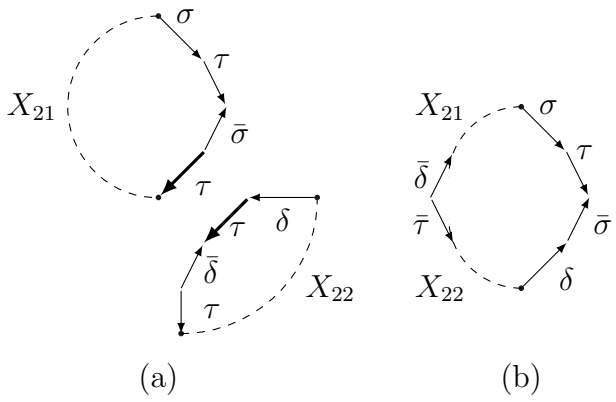


Figure 11: **(a)** $\Lambda(\Gamma)$ when the surface is a sum of an orientable surface and the projective plane. **(b)** $\Lambda(\Gamma)$ when the surface is a sum of an orientable surface and the Klein bottle.

Now let G be a bipartite graph embedded on a non-orientable polygonal schema $\Lambda(\Gamma)$ with $2g'$ sides. We will construct a graph G' embedded on an *orientable* polygonal schema with $4g' - 2$ sides such that G has a perfect matching iff G' has a perfect matching. Moreover, given a perfect matching in G' one can retrieve in log-space a perfect matching in G. This is illustrated in Theorem 6.

Theorem 6. *Let g be a fixed constant. Given a non-orientable graph $G \in \text{BDDBIPGENUS}_g$ as an embedding on a polygonal schema in normal form $\Lambda(\Gamma)$, with $2g'$ sides, where the cyclic ordering of the sides of $\Lambda(\Gamma)$ are $\sigma\sigma X$ or $\sigma\tau\bar{\sigma}\tau X$ and X is the orientable part, in log-space we can construct an orientable graph $G' \in \text{BDDBIPGENUS}_{4g'-2}$ together with its embedding on the polygonal schema of an orientable surface Γ' of genus $4g' - 2$ such that: G has a perfect matching if and only if G' has a perfect matching. Moreover, given a perfect matching in G', we can construct a perfect matching in G in log-space.*

Proof. We first show the case when Γ is the connected sum of an orientable surface and a Klein bottle. Consider the polygonal schema formed by taking two copies of $\Lambda(\Gamma)$ and glueing the side τ of one copy with its partnered side τ of the other copy. We relabel the edge labelled σ in the second copy with some unused symbol δ to avoid confusion. The entire reduction is shown in Figure 12. Let G' be the resulting graph.



(a) (b)

Figure 12: *Klein bottle.* **(a)** The two copies of $\Lambda(\Gamma)$ with the side that is being glued shown in dark. **(b)** Polygonal schema obtained after the glueing operation.

Note that the polygonal schema obtained as a result represents an orientable surface and has $4g' - 2$ sides. Also every vertex and edge in G has exactly two copies in G' and G' is also bipartite. Let M be a matching in G. Let M' be the union of the edges of M from both the copies of G . It is easy to see that M' is a matching in G'. Now consider a matching M' in G'. The *projection* of M' to G gives a subgraph of G where every vertex has degree (counted with multiplicity) exactly two. Since G is bipartite, one can obtain a perfect matching within this subgraph.

Now consider the case when Γ is the connected sum of an orientable surface and a projective plane, that is, following the notation above X_1 corresponds to the labels of a polygonal schema for the projective plane and X_2 corresponds to the labels of a polygonal schema of an orientable surface. Take two copies of $\Lambda(\Gamma)$, and glue σ of one copy with its partner σ in the other copy. We show this operation in Figure 13.

The rest of the proof is similar to the Klein bottle case. □

As a consequence of 6 we see that the non-orientable case can be reduced to the orientable case. The resulting polygonal schema need not be in the

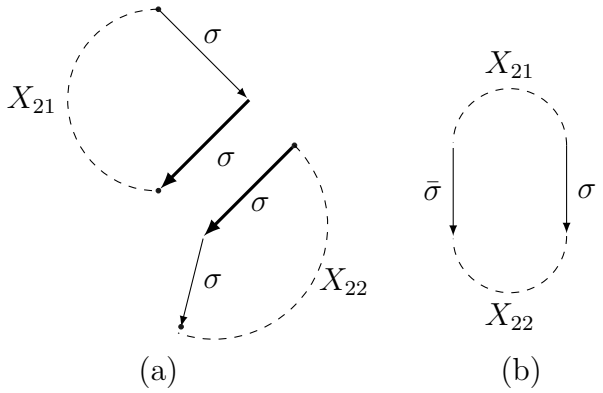


Figure 13: *Projective plane.* **(a)** The two copies of $\Lambda(\Gamma)$ with the two pair of sides that are being glued shown in dark. **(b)** Polygonal schema obtained after the glueing operation.

normal form. To get a combinatorial embedding on a polygonal schema in the normal form we apply Lemma 5 once more.

3.4. Matching-preserving Reduction to a graph in the required form

As a consequence of Theorem 6, we can assume that the input graph (orientable or non-orientable) is embedded on an orientable polygonal schema in normal form. Now the following theorem completes the claim given at the beginning of this section.

Theorem 7. *Let g be a fixed constant. Then given a 2-cell combinatorial embedding of a graph $G \in$ BddBipGenus$_g$ we can find a polygonal schema in normal form of a corresponding orientable graph $H \in$ BddBipGenus$_{g'}$ (where $g' = O(|g|)$) in log-space, such that, no edge of H has both its end points on the boundary of the polygon. Moreover there is a perfect matching in G if and only if there is a perfect matching in H. Moreover, given a perfect matching M' in H, in log-space we can construct a perfect matching M in G.*

Proof. Given a combinatorial embedding of a graph G we apply Lemma 5 to get a combinatorial embedding of G' on a polygonal schema in normal form. At this point, there are no vertices lying on the boundary of the polygonal schema, only edges crossing it. From G' we construct another graph $G'' \in$ ConstantGenusBipartite as follows: for each such edge $e = (u, v)$, such that u and v lie on different segments of the polygon, we

introduce internal vertices $u' = v', v''$ on the edge e(converting it to a path $u, u' = v', v'', v$ of length 3) so that u', v' lie on the boundary of the polygon on the sides nearer to u, v respectively (its important to note that u' and v' are copies of the same vertex in G'' that lie on the boundary of the polygon of the respective paired sides). Now, due to this construction, the edge e gets replaced by a path of length 3 and thus the number of perfect matchings in G'' is preserved.[3] If G was orientable, we output $H = G''$ and we are done.

If G was non-orientable, we apply Theorem 6 to get a graph G_2 along with its embedding on an orientable polygonal schema (need not be in the normal form), such that there is a one to one correspondence between perfect matchings in G_1 and G_2. Once again we apply Lemma 5 to get a graph say G_2' on a polygonal schema in normal form. Then we repeat the same procedure as above to get the desired orientable graph in ConstantGenusBipartite as earlier. □

4. New Upper Bounds

In this section we establish new upper bounds on the space complexity of certain matching problems on the class of constant genus bipartite graphs, given as a combinatorial embedding.

4.1. Main Technical Theorem

In this section we prove Theorem 8 which would help us in establishing the desired upper bounds.

Theorem 8 (Main Theorem)**.** *Let g be a fixed constant. Then given an orientable graph $G \in$ DirBddBipGenus$_g$ in terms of its polygonal schema, such that no edge of G has both its end points along the boundary of the polygon, there exists a log-space computable, polynomially bounded, skew-symmetric weight function $w : E(G) \to \mathbb{Z}$, such that for any cycle $C \in G$ of length at least 3, $w(C) \neq 0$.*

Proof. Let $(\phi, \mathcal{S})$ be the polygonal schema (having $2g'$ sides) of the given graph G, where $\mathcal{S} = (S_1, S_2, \ldots, S_{2g'})$ and $g' = O(g)$. We shall denote the pair

[3]That is if e was part of a matching in G' then we include the edges (u, u') and (v, v'') in the matching in G'', and if e was not part of a matching in G' then we include the edge (v', v'') in the matching in H, the rest being the same.

of a side S_i by S'_i. Let $\mathcal{T} = \{T_1, T_2, \ldots T_{g'}\}$ be the set of distinct sides of the polygon, that is no two elements in $\mathcal{T}$ are pairs of each other. We define w as a linear combination of the $2g'+1$ weight functions $w_1, w'_1, \ldots, w_{g'}, w'_{g'}, w_{TV}$.[4]

For each edge $e = (u, v)$ in G, define the $2g' + 1$ weight functions as follows:

- For each $i \in [g']$,

$$w_i(e) = \begin{cases} 1 & \text{if } u \text{ lies on the side } T_i \\ -1 & \text{if } v \text{ lies on the side } T_i \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

- For each $i \in [g']$,

$$w'_i(e) = \begin{cases} j & \text{if } u \text{ lies on the side } T_i \text{ at index } j \text{ from the tail of } T_i \\ -j & \text{if } v \text{ lies on the side } T_i \text{ at index } j \text{ from the tail of } T_i \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

- w_{TV} is defined to be the weight function defined by Tewari and Vinodchandran in [8] by assuming G to be a planar graph.

Firstly note that each of the three kinds of weight functions that we defined above are skew-symmetric. Therefore a linear combination of them is also skew-symmetric.

Now think of G as a planar graph by ignoring the identification of vertices along respective sides of the polygon, which we shall call G_{planar}. G_{planar} is bipartite since G is bipartite. Now w_{TV} is the weight function defined in Lemma 3 in [8]. Therefore for every cycle $C \in G$ that does not cross any of the sides S_i (and thus is a valid cycle in G_{planar}), $w_{TV}(C) \neq 0$ [8].

Let C be a simple cycle in G that crosses the boundary of the polygon at some point. That is, there exists a vertex v on side T_i such that the cycle enters v from the side T_i and exits from the vertex corresponding to v on side T'_i. Let E_j^C be the set of edges of C whose one end-point (either head or tail) lies on the side T_j. From the definition of w_j it follows that $w_j(C) = w_j(E_j^C)$ (same thing holds for w'_j as well). Now if there exists a k such that $|E_k^C|$ is

[4]We can construct such a w in log-space since g' is constant.

odd, then $w_k(C) \neq 0$. Otherwise, if for all k, $|E_k^C|$ is even then by Lemma 10 it follows that the edges of E_k^C, alternate between going out and coming into the side T_k if at all it passes through the side T_k. Existence of one such side is guaranteed since C crosses the boundary of the polygon. Without loss of generality we assume that side is T_1. Now by using Lemma 11 we get that $w_1'(E_1^C) \neq 0$ and thus $w_1'(C) \neq 0$. (See below for Lemma 10 and 11) □

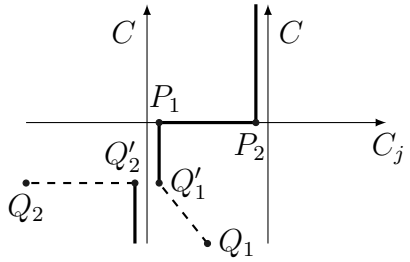


Figure 14: Construction of a path from Q_1 to Q_2 in $\Gamma \setminus C$ (the dotted path is a path between Q_1 and Q_1' (resp. between Q_2 and Q_2').

To establish Lemma 10 we use an argument (Lemma 9) from homology theory. For two cycles (directed or undirected) C_1 and C_2, let $I(C_1, C_2)$ denote the number of times C_1 and C_2 cross each other (that is one of them goes from the left to the right side of the other, or vice versa).

Next we adapt the following Lemma from Cabello and Mohar [23]. Here we assume we are given an orientable surface (Cabello and Mohar give a proof for a graph on a surface).

Lemma 9 ([23])**.** *Given a genus g orientable, surface Γ, let $\mathcal{C} = \{C_i\}_{i\in[2g]}$ be a set of cycles that generate the first homology group $H_1(\Gamma)$. A cycle C in Γ is non-separating if and only if there is some cycle $C_i \in \mathcal{C}$ such that $I(C, C_i) \equiv 1 \pmod 2$.*

Proof. Let $\tilde{C}$ be some cycle in Γ. We can write $\tilde{C} = \sum_{i\in[2g]} t_i C_i$, where $t_i \in \{0, 1\}$, since $\mathcal{C}$ generates $H_1(\Gamma)$. Define $I_{\tilde{C}}(C) = \sum_{i\in[2g]} t_i I(C, C_i) \pmod 2$. One can verify that $I_{\tilde{C}} : \mathcal{C}_1(\Gamma) \to \mathbb{Z}_2$ is a group homomorphism. Now since $\mathcal{B}_1(\Gamma)$ is a normal subgroup of $\mathcal{C}_1(\Gamma)$, $I_{\tilde{C}}$ induces a homomorphism from $H_1(\Gamma)$ to $\mathbb{Z}_2$.

Any cycle is separating if and only if it is homologically equivalent to the empty set. Therefore if C is separating, then $C \in \mathcal{B}_1(\Gamma)$ and thus every

homomorphism from $H_1(\Gamma)$ to $\mathbb{Z}_2$ maps it to 0. Hence for every $i \in [2g]$, $I(C, C_i) \equiv I_{C_i}(C) = 0$.

Suppose C is non-separating. One can construct a cycle C' on Γ, that intersects C exactly once. Let $C' = \sum_{i\in[2g]} t'_i C_i$. Now $1 \equiv I_{C'}(C) \equiv \sum_{i\in[2g]} t'_i I(C, C_i) \pmod 2$. This implies that there exists $i \in [2g]$ such that $I(C, C_i) \equiv 1 \pmod 2$. □

Lemma 10. *Let Γ be a genus g orientable surface and let $\mathcal{C} = \{C_i\}_{i\in[2g]}$ be a system of $2g$ directed cycles on Γ, having exactly one point in common and $\Gamma \setminus \mathcal{C}$ is the fundamental polygon of Γ. Let C be a simple directed cycle on Γ. If $I(C, C_i)$ is even for all $i \in [2g]$ then for all $j \in [2g]$, C alternates between going from left to right and from right to left of the cycle C_j in the direction of C_j (if C crosses C_j at all).*

Proof. Suppose there exists a $j \in [2g]$ such that C does not alternate being going from left to right and from right to left with respect to C_j. Thus if we consider the ordered set of points where C intersects C_j, ordered in the direction of C_j, there are two consecutive points (say P_1 and P_2) such that at both these points C crosses C_j in the same direction.

We will show that for any two points Q_1 and Q_2 in $\Gamma \setminus C$, there exists a path in $\Gamma \setminus C$ between Q_1 and Q_2. This implies that C is a non-separating cycle. Hence by Lemma 9, there exists some $i \in [2g]$, such that $I(C, C_i) \equiv 1 \pmod 2$, thus contradicting the hypothesis.

Consider a path from Q_1 to C. Let Q'_1 be the point on this path that is infinitesimally close to C, without intersecting C. Similarly define a point Q'_2 corresponding to Q_2. Note that it is sufficient for us to construct a path between Q'_1 and Q'_2 in $\Gamma \setminus C$. If both Q'_1 and Q'_2 locally lie on the same side of C, then we get a path from Q'_1 to Q'_2 not intersecting C, by traversing along the boundary of C. Now suppose Q'_1 and Q'_2 lie on opposite sides (w.l.o.g. assume that Q'_1 lies on the right side) of C. From Q'_1 start traversing the cycle until you reach cycle C_j (point P_1 in Figure 14). Continue along cycle C_j towards the adjacent intersection point of C and C_j, going as close to C as possible, without intersecting it (point P_2 in Figure 14). Essentially this corresponds to switching from one side of C to the other side without intersecting it. Next traverse along C to reach Q'_2. Thus we have a path from Q'_1 to Q'_2 in $\Gamma \setminus C$ (an example of this traversal in Figure 14).

□

Lemma 11. *Let G be an orientable, bipartite graph, given as an embedding on a polygonal schema of some surface. Let C be a simple cycle in G and E_1^C be the set of edges of C that lie on the side T_1 (a side of the polygon on which the embedding of the graph is given). Assume $|E_1^C|$ is even and the edges in E_1^C alternate between going out and coming into the polygon. Let $i_1 < i_2 < \ldots < i_{2p-1} < i_{2p}$ be the distinct indices on T_1 where C is incident upon. Then, $\left|w_1'(E_1^C)\right| = \left|\sum_{k=1}^{p}(i_{2k} - i_{2k-1})\right|$ and thus non-zero unless E_1^C is empty.*

Proof. Let $e_j = (u_j, v_j)$ for $j \in [2p]$ be the $2p$ edges of G incident on the side T_1. If u_1 lies on T_1 (that is the edge e_1 is directed away from the side T_1), then by Lemma 10, for every i in $[2p]$ such that i is odd, u_i is incident on T_1 (that is the edge e_i is directed away from the side T_1) and for every i such that i is even, v_i is incident on T_1 (that is the edge e_i is directed towards the side T_1). Hence $w_1'(E_1^C) = i_1 - i_2 + i_3 - i_4 \ldots - i_{2p}$. Similarly, if v_1 lies on T_1, then $w_1'(E_1^C) = -(i_1 - i_2 + i_3 - i_4 \ldots - i_{2p})$. Therefore, in either case $w_1'(E_1^C)$ is non-zero. □

4.2. Complexity Upper Bounds

Theorem 12. *Let g be a fixed constant. Then given a graph $G \in$ BddBipGenus$_g$,*

(a) Decision-BPM *is in* SPL,

(b) Search-BPM *is in* $\mathsf{FL}^{\mathsf{SPL}}$ *and*

(c) Unique-BPM *is in* SPL.

Proof. As a result of Theorem 7, we can assume that our input graph G is orientable, has an embedding on a polygonal schema of genus g' where $g' = O(g)$ and no edge of G has both its endpoints on the boundary of the polygon. This implies that the directed graph $\overrightarrow{G}$ is in the class DirBddBipGenus$_g$ with the above properties as well. Using Theorem 8 and Lemma 1 we get a polynomially bounded, log-space computable weight function W, such that the minimum weight perfect matching in G with respect to W is unique. Moreover, for any subset $E' \subseteq E$, Theorem 8 is valid for the subgraph $\overrightarrow{G} \setminus E'$ also, with respect to the same weight function W. Now (a) and (b) follows from Lemma 2. To check uniqueness, compute a perfect matching M in G. Now M is unique if and only if for every $e \in M$, $G \setminus \{e\}$ has no perfect matching. □

Theorem 8 also gives an alternative proof of directed graph reachability for constant genus graphs.

Theorem 13 ([6, 10])**.** *Directed graph reachability for constant genus graphs is in* UL*.*

The proof of Theorem 13 follows from Theorem 8 and [6], since a path-preserving version of Theorem 7 is easily seen to hold.

Acknowledgment

The third author would like to thank Prof. Mark Brittenham from the Mathematics department at the University of Nebraska-Lincoln, for numerous discussions that they had and for providing valuable insight into topics in algebraic topology. We thank the reviewer's for their very useful comments which substantially improved the presentation of this paper.

References

[1] K. Mulmuley, U. Vazirani, V. Vazirani, Matching is as easy as matrix inversion, Combinatorica 7 (1987) 105–113.

[2] K. Reinhardt, E. Allender, Making nondeterminism unambiguous, SIAM Journal of Computing 29 (2000) 1118–1131. An earlier version appeared in FOCS 1997, pp. 244–253.

[3] A. Gal, A. Wigderson, Boolean complexity classes vs. their arithmetic analogs, Random Structures and Algorithms 9 (1996) 1–13.

[4] E. Allender, K. Reinhardt, S. Zhou, Isolation, matching, and counting: Uniform and nonuniform upper bounds, Journal of Computer and System Sciences 59 (1999) 164–181.

[5] V. Arvind, P. Mukhopadhyay, Derandomizing the isolation lemma and lower bounds for circuit size, in: Proceedings of RANDOM '08, pp. 276–289.

[6] C. Bourke, R. Tewari, N. V. Vinodchandran, Directed planar reachability is in unambiguous log-space, ACM Trans. Comput. Theory 1 (2009) 1–17.

[7] S. Datta, R. Kulkarni, S. Roy, Deterministically isolating a perfect matching inbipartite planar graphs, Theory of Computing Systems 47 (2010) 737–757. 10.1007/s00224-009-9204-8.

[8] R. Tewari, N. V. Vinodchandran, Green's Theorem and Isolation in Planar Graphs, Technical Report TR10-151, Electronic Colloquium on Computational Complexity, 2010.

[9] T. M. Hoang, On the matching problem for special graph classes, in: IEEE Conference on Computational Complexity, pp. 139–150.

[10] J. Kynčl, T. Vyskočil, Logspace reduction of directed reachability for bounded genus graphs to the planar case, ACM Trans. Comput. Theory 1 (2010) 1–11.

[11] G. L. Miller, J. Naor, Flow in planar graphs with multiple sources and sinks, SIAM J. Comput. 24 (1995) 1002–1017.

[12] R. Kulkarni, M. Mahajan, K. R. Varadarajan, Some perfect matchings and perfect half-integral matchings in NC, Chicago Journal of Theoretical Computer Science 2008 (2008).

[13] C. Thomassen, The graph genus problem is NP-complete, J. Algorithms 10 (1989) 568–576.

[14] W. S. Massey, A Basic Course in Algebraic Topology, Springer-Verlag, 1991.

[15] B. Mohar, C. Thomassen, Graphs on Surfaces, John Hopkins University Press, 2001.

[16] S. Arora, B. Barak, Computational Complexity: A Modern Approach, Cambridge University Press, 1 edition, 2009.

[17] H. Vollmer, Introduction to Circuit Complexity - A Uniform Approach, Springer-Verlag, 1999.

[18] O. Reingold, Undirected connectivity in log-space, J. ACM 55 (2008) 1–24.

[19] E. Allender, D. A. M. Barrington, T. Chakraborty, S. Datta, S. Roy, Planar and grid graph reachability problems, Theory Comput. Syst. 45 (2009) 675–723.

[20] H. R. Brahana, Systems of circuits on two-dimensional manifolds, The Annals of Mathematics 23 (1921) 144–168.

[21] M. Dehn, P. Heegaard, Analysis situs, Enzyklopädie der mathematischen Wissenschaften mit Einschluß ihrer Anwendungen III.AB (1907) 153–220.

[22] G. Vegter, C.-K. Yap, Computational complexity of combinatorial surfaces, in: Proceedings of the 6th Annual Symposium on Computational Geometry, pp. 102–111.

[23] S. Cabello, B. Mohar, Finding shortest non-separating and non-contractible cycles for topologically embedded graphs, Discrete Comput. Geom. 37 (2007) 213–235.