Electronic Colloquium on Computational Complexity, Report No. 25 (2020)

Efficient Isolation of Perfect Matching in $O(\log n)$ Genus Bipartite Graphs

Chetan Gupta
Indian Institute of Technology, Kanpur, India
gchetan@cse.iitk.ac.in

Vimal Raj Sharma
Indian Institute of Technology, Kanpur, India
vimalraj@cse.iitk.ac.in

Raghunath Tewari
Indian Institute of Technology, Kanpur, India
rtewari@cse.iitk.ac.in

Abstract

We show that given an embedding of an $O(\log n)$ genus bipartite graph, one can construct an edge weight function in logarithmic space, with respect to which the minimum weight perfect matching in the graph is unique, if one exists.

As a consequence, we obtain that deciding whether the graph has a perfect matching or not is in SPL. In 1999, Reinhardt, Allender and Zhou proved that if one can construct a polynomially bounded weight function for a graph in logspace such that it isolates a minimum weight perfect matching in the graph, then the perfect matching problem can be solved in SPL. In this paper, we give a deterministic logspace construction of such a weight function.

2012 ACM Subject Classification F.1.3 Complexity Measures and Classes

Keywords and phrases logspace computation, high genus, matching isolation

Digital Object Identifier 10.4230/LIPIcs.STACS.2019.32

1 Introduction

Given a graph $G(V, E)$, a *perfect matching* is defined as a set of disjoint edges which covers all the vertices in the graph. The perfect matching problem asks whether a graph has a perfect matching or not. The first polynomial time sequential algorithm to solve this problem was given by Edmonds [6]. Since then, there has been a lot of effort to solve this problem efficiently in a parallel computation model. NC is a class of problem which can be solved efficiently in parallel computation model. Lovász gave the first randomized NC algorithm to solve the perfect matching problem [13]. However, the question whether the problem can be solved in NC or not is still open.

Mulmuley et al. made significant progress in answering this question and gave the famous *isolating lemma* [14].

▶ **Lemma 1.** *(Isolating Lemma [14]) For a set $S = \{x_1, x_2, \ldots x_n\}$, let F be family of subsets of S. If the elements in the set S are assigned integer weights chosen uniformly and independently from the set $\{1, 2, \ldots 2n\}$ then with probability greater than half there is a unique minimum weight set in F.*

Mulmuley et al. used this lemma to get a randomized NC algorithm for finding a perfect matching in graphs. They also showed that if one can construct an isolating weight function in NC (derandomizing the isolating lemma), then a perfect matching can be found in NC. Allender et al. further improved this result and proved that if one can construct an isolating weight function in logspace then the problem can be solved in SPL, which is a subset of NC^2

36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019).
Editors: Rolf Niedermeier and Christophe Paul; Article No. 32; pp. 32:1–32:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

ISSN 1433-8092

[2]. In a recent result, a quasi-polynomial ($O(\log^2 n)$-bit) size isolating weight function was constructed for bipartite graphs which implies that the perfect matching problem can be solved in quasi-NC [7]. This result was subsequently extended to general graphs as well [17]. However, constructing polynomially bounded isolating weight function for general graphs has been elusive so far. Constructing isolating weight function also has ramification in the *directed graph reachability* problem. A logspace construction of a polynomially bounded path isolating weight function will imply that reachability problem in directed graphs can be solved in UL, which will solve the NL vs. UL question, which has been open for a very long time[16]. Also, a logspace construction of a polynomially bounded perfect matching isolating weight function even for bipartite graphs will prove that NL $\subseteq$ SPL [4].

Although constructing polynomially bounded isolating weight function seems to be hard for general graphs, but such weight functions have been constructed for various subclasses of graphs such as planar graphs [18], bounded genus graphs [5], $K_{3,3}$ and K_5-free graphs [3], graph with small number of matchings [9, 1] and graph with small number of nice cycles [11]. The weight function constructed in [5] is a $O(g \cdot \log n)$-bit weight function for g-genus graphs. Thus their result does not yield a polynomial size weight function for the graphs of genus more than constant. The question whether one can construct a polynomially bounded isolating weight function efficiently for graphs of genus beyond constant or not has been open since then. In this work, we settle this question by constructing a $O(g + \log n)$ bit isolating weight function for g-genus graphs. Thus our result gives a polynomial size isolating weight function for $O(\log n)$ genus bipartite graphs.

For a class of bipartite graphs, one way to obtain an isolating weight function is, to construct a *skew-symmetric* weight function for the same class of directed graphs such that every cycle in the graph gets a nonzero weight. This is the common technique in most of the above mentioned results. Having a skew-symmetric weight function such that it gives nonzero weights to every cycle in the graph, is sufficient for both path and matching isolation but is not necessary. Also, a weight function which isolates a path in the graph may not isolate a matching and vice-versa. That is why the weight functions constructed in [12], [19] and [10] are path isolating but do not isolate perfect matching. In this result, we construct a weight function which isolates a perfect matching in g-genus graphs even though it does not give nonzero weight to every cycle in the graphs.

1.1 Our Result

In this paper, we extend the above line of work and prove the following theorem.

▶ **Theorem 2.** *Given an undirected $O(\log n)$ genus bipartite graph along with its polygonal schema, the problem of deciding whether the graph has a perfect matching or not is in* SPL.

Given a g-genus bipartite graph G we construct $O(g + \log n)$-bit weight functions $w_1, w_2, \dots w_k$, where $k = O(n^c + 2^g)$, such that there exists a unique minimum weight perfect matching in the G with respect to some w_i, if G has a perfect matching. To achieve this, we first construct a directed graph $\vec{G}$ which is same as G, but its edges are assigned direction as follows. Let L and R be the two sets of the bipartition of G. We assign a direction to all the edges in $\vec{G}$ from L to R. Then we divide the perfect matchings of $\vec{G}$ into different classes according to their *signatures*. Matchings in one class are said to be topologically equivalent to each other in a sense. For a g-genus graph, there are 2^{2g} many classes. We construct our isolating weight function in two steps. In the first step, we construct a weight function which is a linear combination of the weight function constructed in [18] and another weight function defined later in this paper and show that there is at most one minimum weight perfect matching in each class with respect to this weight function. In the second step, we use the hashing scheme of Fredman, Komlós and Szemerédi [8] to get k many weight functions $w_1, w_2, \ldots, w_k$ such that for some $i \leq k$, w_i isolates a minimum weight perfect matching in $\vec{G}$. A matching in $\vec{G}$ corresponds to a unique matching G and vice-versa. Therefore we get a unique minimum weight perfect matching in G with respect to w_i.

For $g = O(\log n)$ we get $k = O(n^{c'})$, for some constant $c' > 0$. That means we get polynomially many weight functions such that there is at most one minimum weight perfect matching in the graph with respect to at least one of the weight function. Then we apply the result of [2] to get an SPL algorithm for perfect matching problem in $O(\log n)$ genus bipartite graphs.

Comparison with the path isolating weight function for $O(\log n)$ genus graphs [10]: Note that the weight function constructed in [10] is also a linear combination of two weight functions, one of which gives nonzero weights to all surface separating cycles in the graph. Therefore, when we divide the paths between a pair of vertices into classes and take any two minimum weight non-intersecting paths with respect to this weight function from the same class, we know that the cycle formed by reversing one of the paths is surface separating. Since every surface separating cycle has nonzero weight, and the weight function is skew-symmetric, this implies that these paths can not be of equal weights. Which means there is at most one minimum weight path in each class with respect to that weight function. Similarly, we handle the case when the paths are intersecting. However, that same weight function does not work here in matching isolation. Here also we first divide the matchings into classes according to their signatures. Now if we consider two minimum weight perfect matchings within a class, all the cycles formed by taking their disjoint union can be surface non-separating. Since the weight of a surface non-separating cycle can be zero with respect to that weight function, this does not give any contradiction to the fact that there can be two minimum weight perfect matchings within a class. In this paper, we overcome this hurdle by constructing a new weight function which isolates a matching within a class. Then we isolate a matching across the classes by the technique mentioned above.

1.2 Organization of the Paper

Rest of the paper is organized as follows. In Section 2, we define the necessary notations and a suitable representation of high genus graphs which we use in this paper. In Section 3, we define the first part of our weight function, which is a linear combination of two weight functions defined in that section. In Section 4, we prove that the number of minimum weight perfect matchings with respect to this weight function is very small. Then we use the hashing scheme of [8] to obtain our final weight function, which isolates a minimum weight perfect matching in the graph.

2 Preliminaries and Notations

A g-genus surface is a sphere with g-many handles on it. A g-genus graph is a graph which can be embedded on a g-genus surface without intersecting its edges. A g-genus surface can be represented by a polygon called *polygonal schema*(see Figure 1). The polygonal schema of a g-genus surface has $4g$-sides $T_1, T_2, T'_1, T'_2, \ldots, T'_{2g-1}, T'_{2g}$ identified in pairs. The sides T_i and T'_i form a pair together. An embedding of a graph G on a g-genus surface can be represented by an embedding of G inside this polygon. In such an embedding an edge $\{u, v\}$ of a graph G is said to cross a side S of the polygonal schema, if u or v is incident on the side

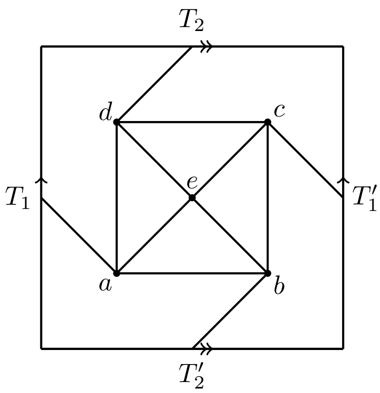

Figure 1 Polygonal schema of K_5, embedded on a surface of genus 1. Edges $\{a, c\}$ and $\{b, d\}$ are crossing the sides T_1 and T_2 respectively. Vertices a and c are said to be incident on the sides T_1 and T_1' respectively.

S (for example in Figure 1, the edge $\{a, c\}$ is crossing the sides T_1 and T_1'). We assume that we are given the combinatorial embedding of the graph G inside this polygon together with the ordered set of edges crossing each side of the polygon. We also assume that no vertex of G lies on the sides of the polygonal schema. Such an embedding is called the *polygonal schema of the graph G.*

In the polygonal schema of a graph G, the edges which do not cross any side of the polygonal schema, we call them *planar edges*. Note that in the polygonal schema of a graph G, the subgraph induced by the planar edges of G, is a planar graph and we call this subgraph G_{planar}.

A piecewise straight-line embedding of a planar graph is an embedding where all the vertices of the graph have integral coordinates and the edges are piecewise straight line segment connecting their two end points. Given a combinatorial embedding of a planar graph, a piecewise straight-line embedding of it can be constructed in logspace [18]. Thus given a polygonal schema of a g-genus graph G, a piecewise straight-line embedding of G_{planar} can be constructed in logspace. We will need such an embedding to construct our desirable weight function.

Given the polygonal schema of a g-genus graph G, we define the *signature* of an edge e in G, denoted as $\text{sign}(e)$, as a $2g$-bit binary string $b_1 b_2 \ldots b_{2g}$, such that $b_i = 1$ if e crosses T_i, otherwise 0. Similarly, for any set of edges say $E = \{e_1, e_2, \ldots, e_k\}$, we define the signature of E as, $\text{sign}(E) = \text{sign}(e_1) \oplus \text{sign}(e_2) \oplus \ldots \oplus \text{sign}(e_k)$, where $\oplus$ represents the bitwise-XOR operator. Note that the i-th bit in the signature of a set E represents the parity of the number of edges from that set, crossing the side T_i, i.e. if the number of edges in the set E, crossing the side T_i are even then i-th bit in the $\text{sign}(E)$ will be 0; otherwise it will be 1. Without loss of generality assume that each edge crosses at most one side of the polygonal schema. If it crosses more than one side of the polygonal schema, we break it into multiple edges by inserting dummy vertices. To preserve matching, we always break an edge into an odd number of edges. Every term defined till now remains the same in case of directed graphs as well.

Since in this paper we work with both directed and undirected graphs, it is essential that we make a demarcation in the notation used for directed and undirected graphs. For a directed edge $\vec{e} = (u, v)$, the edge $e = \{u, v\}$ represents the underlying undirected edge and the edge $\vec{e}^r$ represents the directed edge (v,u) that is the edge $\vec{e}$ with its direction reversed. Similarly, for any set of directed edges $\vec{E}$, set E represents the set of underlying undirected edges of $\vec{E}$ and set $\vec{E}^r$ represents the set where each edge $\vec{e} \in \vec{E}$ is replaced with the edge $\vec{e}^r$.

In a directed graph $\vec{G}$, we call a set of edges $\vec{C}$, a *directed cycle* if (i) edges of C (underlying undirected edges of $\vec{C}$) form a simple cycle and, (ii) for every two adjacent edges of $\vec{C}$, tail of one edge is followed by the head of another edge. When we call $\vec{C}$ just a *cycle* then (ii) may not hold. Similarly we can define *directed path* and *path* in $\vec{G}$.

$(0)^k$ represents the string $\overbrace{00\ldots0}^{k\text{-times}}$, where k is an integer. For an integer $l > 0$, $[l]$ denotes the set $\{1, 2, \ldots, l\}$.

3 Isolating Weight function

As discussed in the introduction, our main goal here is to construct a weight function for graphs efficiently. Let us first define the weight function formally. A weight function for a graph (directed or undirected) $G(V,E)$ is a map $w : E \to Z$ which assigns an integer weight to every edge in the graph. For any set of edges E' in the graph, the weight of the set E' is defined as $w(E') = \sum_{e \in E'} w(e)$. A weight function w for a graph G is called *min-isolating* if there exists at most one minimum weight perfect matching in G with respect to the weight function w.

In case of directed graphs, a weight function w is called *skew-symmetric* if $w(\vec{e}) = -w(\vec{e}^r)$, for all $\vec{e} \in \vec{E}$.

For a g-genus graph $\vec{G}$, we define a weight function w_{comb} which is a linear combination of the following two weight functions.

- The first weight function we define is the same as the one defined in [18] for directed planar graphs. We call it w_{pl}. As we mentioned in Section 2, we can construct a piecewise straight-line embedding of $\vec{G}_{\text{planar}}$ in logspace. For an edge $\vec{e} = (u,v)$, let (x_u, y_u) and (x_v, y_v) be the coordinates of the vertices u and v respectively, in the piecewise straight-line embedding of $\vec{G}_{\text{planar}}$.

$$w_{\text{pl}}(\vec{e}) = \begin{cases} (y_v - y_u)(x_u + x_v), & \text{if } \vec{e} \text{ is a planar edge,} \\ 0, & \text{otherwise.} \end{cases}$$

  We state the following theorem regarding the weight function w_{pl}, which gives us a characterization of the weight of a directed cycle in a directed planar graph, established in [18].

  ▶ **Theorem 3.** *[18] Given a piecewise straight-line embedding of a planar graph $\vec{G}$, there exists a logspace computable weight function w_{pl} such that for any directed cycle $\vec{C}$ in $\vec{G}$, we have $w_{pl}(\vec{C}) = 2 \cdot \text{Area}(\vec{C})$ if $\vec{C}$ is a counter-clockwise cycle and $w_{pl}(\vec{C}) = -(2 \cdot \text{Area}(\vec{C}))$ if $\vec{C}$ is a clockwise cycle, where $\text{Area}(\vec{C})$ is the area of the region enclosed by $\vec{C}$.*

- We define another weight function w_{side} as follows. Let $\sigma = (\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_k)$ be the ordered set of edges crossing the sides of the polygonal schema T_1 to T_{2g}, ordered in a clockwise manner starting from the tail of T_1.

$$w_{\text{side}}(\vec{f}_i) = \begin{cases} i, & \text{if tail}(\vec{f}_i) \text{ is incident on some side } T_j \text{ for } j \in [2g], \\ -i, & \text{if head}(\vec{f}_i) \text{ is incident on some side } T_j \text{ for } j \in [2g]. \end{cases}$$

For all other edges $\vec{e}$, $w_{\text{side}}(\vec{e}) = 0$.

Our weight function w_{side} is somewhat similar to the weight function defined in Theorem 8 of [5]. However, the main difference is that they define $2g$ many weight functions (one for each pair of side of the polygonal schema) similar to w_{side} and their final weight function is a linear combination of those $2g$ weight functions, making it an $O(g \cdot \log n)$-bit size weight function for g-genus graphs. Whereas in this paper w_{side} is a single $O(\log n)$-bit weight function for a g-genus graph.

Since each of these two weight functions are polynomially bounded and are computable in logspace, the overall computation remains in logspace as well. We combine these two weight functions into a single weight function and call it w_{comb}, defined as follow:

$$w_{\text{comb}} = w_{\text{pl}} \cdot n^{10} + w_{\text{side}}. \tag{1}$$

Since for any two subsets of edges $\vec{E_1'}$ and $\vec{E_2'}$ of the graph, both weight functions w_{pl} and w_{side} are bounded by n^{10}, hence $w_{\text{comb}}(\vec{E_1'}) = w_{\text{comb}}(\vec{E_2'})$ if and only if $w_{\text{pl}}(\vec{E_1'}) = w_{\text{pl}}(\vec{E_2'})$ and $w_{\text{side}}(\vec{E_1'}) = w_{\text{side}}(\vec{E_2'})$.

Note that in the perfect matching problem, we are given an undirected graph and asked to find if the graph has a perfect matching or not. However, we have defined the weight function w_{comb} for directed graphs. In order to give weights to an undirected bipartite graph G, we first obtain a directed graph $\vec{G}$ and construct a weight function for $\vec{G}$. Then we use that weight function to build a weight function for G.

Let G be an undirected bipartite graph and (L, R) be its bipartition. We construct a directed graph $\vec{G}$ as follow. For an edge $\{u, v\}$ in G such that $u \in L$ and $v \in R$, we replace it with a directed edge (u, v) in $\vec{G}$. We use Reingold's algorithm [15] to find out whether a vertex belongs to L or R. Let w be a weight function for $\vec{G}$. We define corresponding weight function w^{und} for G as follow. For an edge $\{u, v\} \in G$ such that $u \in L$ and $v \in R$,

$$w^{\text{und}}\big(\{u, v\}\big) = w(u, v), \text{ where } (u, v) \in \vec{G} \tag{2}$$

Note that if $\vec{M}$ is a matching of weight t in $\vec{G}$ then M will be a matching of weight t in G. Thus, if w is a min-isolating weight function for $\vec{G}$ then w^{und} will be min-isolating for G.

In the next section, we will construct a min-isolating weight function for directed g-genus bipartite graphs. Then ultimately we will use that weight function to obtain a min-isolating weight function for undirected g-genus bipartite graphs.

4 Isolating a Minimum Weight Perfect Matching

Let $\vec{G}$ be a g-genus bipartite graph and (L, R) be its bipartition. Let us assume that all the edges in $\vec{G}$ have direction from L to R. We will prove that there are at most 2^{2g} minimum weight perfect matchings in $\vec{G}$ with respect to the weight function w_{comb}, if $\vec{G}$ has a perfect matching.

Let $\vec{M}$ be a perfect matching in $\vec{G}$. As we defined in Section 2, signature of $\vec{M}$ is,

$$\text{sign}(\vec{M}) = \text{sign}(\vec{e_1}) \oplus \text{sign}(\vec{e_2}) \oplus \ldots \oplus \text{sign}(\vec{e_j}), \text{ where } \vec{e_i} \in \vec{M} \text{ for } i \in [j].$$

Note that for a g-genus graph each matching has a $2g$-bit signature. Thus there are 2^{2g} many possible signatures. For each $0 \le i \le 2^{2g} - 1$, let $\text{bin}(i)$ represent the $2g$-bit binary number(with possible leading 0's) equivalent to an integer i. We define a class A_i of perfect matchings in $\vec{G}$ with respect to the signature $\text{bin}(i)$ for all $0 \le i \le 2^{2g} - 1$, as

$$A_i = \{\vec{M} \mid \vec{M} \text{ is a perfect matching in } \vec{G} \text{ and } \text{sign}(\vec{M}) = \text{bin}(i)\}$$

We will prove that there exists at most one minimum weight perfect matching in each class with respect to the weight function w_{comb}.

▶ **Lemma 4.** *For a g-genus bipartite graph $\vec{G}$, there exists at most one minimum weight perfect matching in the class A_i for all $i \in [2^{2g}]$, with respect to the weight function w_{comb}.*

For two matchings $\vec{M}_1$ and $\vec{M}_2$ in $\vec{G}$, we define $\vec{E}_{\vec{M}_1 \Delta \vec{M}_2} = (\vec{M}_1 \cup \vec{M}_2) \setminus (\vec{M}_1 \cap \vec{M}_2)$. Let us first prove the following lemma about the characterization of the edges in the set $\vec{E}_{\vec{M}_1 \Delta \vec{M}_2}$, when $\vec{M}_1$ and $\vec{M}_2$ are two perfect matchings from the same class.

▶ **Lemma 5.** *If $\vec{M}_1$ and $\vec{M}_2$ are the two perfect matchings in the class A_i then* $\text{sign}(\vec{E}_{\vec{M}_1 \Delta \vec{M}_2}) = (0)^{2g}$ *that is, the edges in the set $\vec{E}_{\vec{M}_1 \Delta \vec{M}_2}$ collectively cross each side of the polygonal schema an even number of times.*

Proof. Since $\vec{M}_1$ and $\vec{M}_2$ are the matchings from the same class, we have

$$\begin{aligned}
\text{sign}(\vec{M}_1) &= \text{sign}(\vec{M}_2) \\
\text{sign}(\vec{M}_1) &\oplus \text{sign}(\vec{M}_2) = (0)^{2g} \\
\Big(\text{sign}(\vec{M}_1 \cap \vec{M}_2) \oplus \text{sign}(\vec{E}_{\vec{M}_1 \Delta \vec{M}_2} \setminus \vec{M}_2)\Big) &\oplus \Big(\text{sign}(\vec{M}_1 \cap \vec{M}_2) \oplus \text{sign}(\vec{E}_{\vec{M}_1 \Delta \vec{M}_2} \setminus \vec{M}_1)\Big) = (0)^{2g} \\
\big(\text{sign}(\vec{E}_{\vec{M}_1 \Delta \vec{M}_2} \setminus \vec{M}_2)\big) &\oplus \big(\text{sign}(\vec{E}_{\vec{M}_1 \Delta \vec{M}_2} \setminus \vec{M}_1)\big) = (0)^{2g} \\
\text{sign}(\vec{E}_{\vec{M}_1 \Delta \vec{M}_2}) &= (0)^{2g}.
\end{aligned}$$

◀

We will now show that there is at most one minimum weight perfect matching in each class. Assume that $\vec{M}_1$ and $\vec{M}_2$ are the two minimum weight perfect matchings in the class A_i with respect to the weight function w_{comb}. We know that the edges in the set $\vec{E}_{\vec{M}_1 \Delta \vec{M}_2}$ form vertex disjoint cycles. Let $\vec{C}_1, \vec{C}_2, \ldots, \vec{C}_k$ be those cycles. Notice that all the edges in the cycle $\vec{C}_i$ are directed from L to R therefore $\vec{C}_i$ is not a directed cycle, for all i. Also, note that each $\vec{C}_i$ consists of even number of edges and contain alternating edges from $\vec{M}_1$ and $\vec{M}_2$. Hence we can claim the following.

▷ Claim 6. Let $\vec{E}_{1i}$ and $\vec{E}_{2i}$ be the set of edges of $\vec{M}_1$ and $\vec{M}_2$ respectively in $\vec{C}_i$ then $w_{\text{comb}}(\vec{E}_{1i}) = w_{\text{comb}}(\vec{E}_{2i})$, for all $i \in [k]$.

Proof. Let us assume that there exists some $j \in [k]$ such that $w_{\text{comb}}(\vec{E}_{1j}) \neq w_{\text{comb}}(\vec{E}_{2j})$. Without loss of generality assume that $w_{\text{comb}}(\vec{E}_{1j}) > w_{\text{comb}}(\vec{E}_{2j})$. Now consider a new perfect matching $\big((\vec{M}_1 \setminus \vec{E}_{1j}) \cup \vec{E}_{2j}\big)$. This matching has strictly lesser weight than $\vec{M}_1$, which is a contradiction because we have assumed that $\vec{M}_1$ is a minimum weight perfect matching. ◀

Now consider another graph $\vec{G'}$ which is same as $\vec{G}$ but direction of the edges belonging to $\vec{M}_2$ is reversed in $\vec{G'}$. Let $\vec{M}'_1$ and $\vec{M}'_2$ be the matchings in $\vec{G'}$ corresponding to the matchings $\vec{M}_1$ and $\vec{M}_2$ in $\vec{G}$, where $\vec{M}'_1$ is same as $\vec{M}_1$, but $\vec{M}'_2 = \vec{M}_2^r$. We know that the edges in the set $\vec{E}_{\vec{M}'_1 \Delta \vec{M}'_2}$ will form vertex disjoint cycles. Let $\vec{C}'_1, \vec{C}'_2, \ldots, \vec{C}'_k$ be those cycles and $\vec{E}'_{1i}$ and $\vec{E}'_{2i}$ be the edges of matching $\vec{M}'_1$ and $\vec{M}'_2$ respectively, in the cycle $\vec{C}'_i$. By claim 6 we know that

$$w_{\text{comb}}(\vec{E}_{1i}) = w_{\text{comb}}(\vec{E}_{2i}), \text{ for all } i \in [k].$$

Also $\vec{E}_{1i} = \vec{E}'_{1i}$ and $\vec{E}_{2i} = \vec{E}'^{\,r}_{2i}$, therefore

$$w_{\text{comb}}(\vec{E}'_{1i}) = w_{\text{comb}}(\vec{E}'^{\,r}_{2i}), \text{ for all } i \in [k].$$

Since w_{comb} is skew-symmetric, we have

$$\begin{aligned} w_{\text{comb}}(\vec{E}'_{1i}) &= -w_{\text{comb}}(\vec{E}'_{2i}), \\ w_{\text{comb}}(\vec{E}'_{1i}) &+ w_{\text{comb}}(\vec{E}'_{2i}) = 0, \\ w_{\text{comb}}(\vec{C}'_i) &= 0, \text{ for all } i \in [k]. \end{aligned} \tag{3}$$

Note that the edges in the set $\vec{E}'_{1i}$ have direction from L to R and the edges in set $\vec{E}'_{2i}$ have direction from R to L therefore the cycles $\vec{C}'_1, \vec{C}'_2, \ldots, \vec{C}'_k$ are the directed cycles in $\vec{G}'$. We will now prove that $w_{\text{comb}}(\vec{C}'_i) \neq 0$ for some $i \in [k]$, which will be a contradiction with Equation 3.

Since changing the direction of an edge does not change its signature, by Lemma 5 we know that $\text{sign}(\vec{C}'_1) \oplus \text{sign}(\vec{C}'_2) \oplus \ldots \oplus \text{sign}(\vec{C}'_k) = (0)^{2g}$.

▶ **Lemma 7.** *Let $\vec{G}'$ be a g-genus graph which contains directed cycles $\{\vec{C}'_1, \vec{C}'_2, \ldots, \vec{C}'_k\}$ such that* $\text{sign}\,(\vec{C}'_1) \oplus \text{sign}\,(\vec{C}'_2) \oplus \ldots \oplus \text{sign}\,(\vec{C}'_k) = (0)^{2g}$*. Then there exists $i \in [k]$, such that $w_{comb}(\vec{C}'_i) \neq 0$.*

Proof. First consider the case, when no edge of the cycles $\{\vec{C}'_1, \vec{C}'_2, \ldots, \vec{C}'_k\}$ crosses any side of the polygonal schema. In that case each cycle $\vec{C}'_i$ is a planar cycle i.e. consists of only planar edges. By Theorem 3 we know that $w_{\text{pl}}(\vec{C}_i) \neq 0$, which implies that $w_{\text{comb}}(\vec{C}_i) \neq 0$ for all $i \in [k]$. Hence the lemma holds in this case.

We will now prove the lemma for the case when some edges of the cycles $\{\vec{C}'_1, \vec{C}'_2, \ldots, \vec{C}'_k\}$ cross some sides of the polygonal schema.

Let us consider a graph G'' such that edges of G'' are the underlying undirected edges of the cycles $(\vec{C}'_1, \vec{C}'_2, \ldots, \vec{C}'_k)$. Let $\mathcal{C} = (C''_1, C''_2, \ldots, C''_k)$ be the cycles in G'' corresponding to cycles $(\vec{C}'_1, \vec{C}'_2, \ldots, \vec{C}'_k)$. We will construct another directed graph $\vec{G}''$ from G''(by assigning direction to the edges of G'') such that either $\vec{C}''_i = \vec{C}'_i$ or $\vec{C}''_i = \vec{C}'^{\,r}_i$, for all $i \in [k]$. Let $E_{\mathcal{C}}$ be the set of edges of the cycles in $\mathcal{C}$. We assign direction to the edges of $E_{\mathcal{C}}$ in two steps. In the first step, we assign direction to only those edges of $E_{\mathcal{C}}$ which are crossing some side of the polygonal schema. In the second step, we assign direction to the planar egdes of $E_{\mathcal{C}}$, based on the direction of the edges which were assigned direction in the first step.

We know that all the cycles in $\mathcal{C}$ collectively cross each side of the polygonal schema an even number of times. Let $E = (e_1, e_2, \ldots e_{2l})$ for some integer $l > 0$, be the edges in the set $E_{\mathcal{C}}$, which cross some of the sides of the polygonal schema, indexed in clockwise order from T_1 to T_{2g}, starting from the tail of T_1. Without loss of generality assume that no two edges in E share a vertex because if they do, we insert dummy vertices in the edge so that our assumption holds. We will need this assumption to simplify our analysis.

- *Step 1:* In this step, we assign direction to the edges in the set E. Let $e_i = \{u, v\}$ be an edge in E such that u and v are incident on sides T_j and T'_j respectively, of the polygonal schema. We assign direction to $e_i \in E$ as follows:
  - Assign direction to e_i from u to v, if i is odd, i.e. assign direction to e_i in such a way that u becomes the tail of $\vec{e_i}$ and v becomes the head of $\vec{e_i}$ in $\vec{G}''$.
  - Similarly, assign direction to e_i from v to u, if i is even.

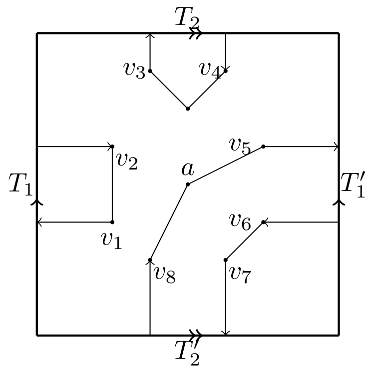

Figure 2 $(v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8)$ are the vertices of the edges which are crossing sides of the polygonal schema. Path v_8av_5 is a planar path.

Before going to *Step* 2, let us make the following observations. Let $\vec{E} = (\vec{e_1}, \vec{e_2} \dots \vec{e_{2l}})$ are the edges in $\vec{G''}$ corresponding to edges in E after *Step 1*. Let $X = \{v_1, v_2, \dots v_{4l}\}$ be the vertices of the edges of $\vec{E}$ ordered in a clockwise manner, according to their incidence on the side of the polygonal schema, starting from the tail of T_1(see Figure 2). Note that,

$$\vec{e_i} = (v_{d_1}, v_{d_2}), \text{ where } d_1 \text{ is odd and } d_2 \text{ is even, for all } i \in [2l]. \tag{4}$$

We define a function $\tau : X \to X$. $\tau(v_i) = v_j$ if there is a simple path P from v_i to v_j which consists of only planar edges of $E_{\mathcal{C}}$, for $i, j \in [4l]$. We call such paths as *planar paths* (see Figure 2). Since vertices in X are the part of simple cycles, the function τ is a bijective function.

▶ **Lemma 8.** *If $\tau(v_i) = v_j$, then $|i - j|$ is odd.*

Proof. Assume that both i and j are odd. Without loss of generality assume that $j > i$. This implies that there are an odd number of vertices in the set X, between v_i and v_j namely, $X' = (v_{i+1}, v_{i+2}, \dots v_{j-1})$. Note that vertices in X' are part of non-intersecting simple cycles therefore they must be connected to each other through a simple planar path. Since τ is a bijective function we know that there is some vertex $v' \in X'$ such that $T(v') = v_t$ where $t \in [4l]$ and, $t > j$ or $t < i$. This is not possible because it will imply that planar paths say from v' to v_t and from v_i to v_j say P_1 and P_2 respectively, must intersect each other. This is a contradiction since P_1 and P_2 are the parts of non-intersecting cycles. ◀

▶ **Lemma 9.** *Let P be a planar path between vertices v_i and $v_j, i, j \in [4l]$. If v_i is the head of some edge then v_j will be the tail of some edge, in $\vec{E}$ and vice versa.*

Proof. Let v_i and v_j both the vertices are the heads of the edges e_{c_1} and e_{c_2}, where $c_1, c_2 \in [k]$. We know that if i is even then j is odd and if i is odd then j is even. Without loss of generality assume that i is even and j is odd. However, from Equation 4 we know that j must be even. Hence we get a contradiction to Lemma 8.
Similarly, we can handle the case when v_i and v_j are the tail of some edges. ◀

- *Step 2:* Now we will assign the direction to the planar edges of $\vec{G''}$. This step is pretty straight forward. Take a planar path P of $\vec{G''}$. Let v' and v'' be its end vertices such that v' is the head of an edge $\vec{e'}$ and v'' is the tail of some edge $\vec{e''}$, where $\vec{e'}, \vec{e''} \in \vec{E}$. Assign direction to all the edges in P in such a way that the path $\vec{P'} = \vec{e'}\vec{P}\vec{e''}$ becomes a directed path in $\vec{G''}$.

Let $\vec{C_1''}, \vec{C_2''}, \ldots, \vec{C_k''}$ be the cycles in $\vec{G''}$ after assigning direction to the underlying undirected cycles $C_1'', C_2'', \ldots, C_k''$. After assigning direction using the above procedure, we can ensure that no two adjacent edges in the cycle $\vec{C_i''}$ for all $i \in [k]$ get opposite direction i.e. if $\vec{e}$ and $\vec{e'}$ are two adjacent edges in the cycle $\vec{C_i''}$ then the tail of e will be followed by the head of $\vec{e'}$ or vice-versa (because of *Step* 2). This implies that $\vec{C_1''}, \vec{C_2''}, \ldots, \vec{C_k''}$ are the directed cycles in $\vec{G''}$. Note that the way we have defined weight function w_{side}, we know that

$$\begin{aligned}
& w_{\text{side}}(\vec{e}_i) < -\big(w_{\text{side}}(\vec{e}_i + 1)\big), \text{ for all } 1 \le i < 2l \\
\implies \quad & w_{\text{side}}(\vec{e}_1) + w_{\text{side}}(\vec{e}_3) + \ldots + w_{\text{side}}(\vec{e}_{2l-1}) < -\big(w_{\text{side}}(\vec{e}_2) + w_{\text{side}}(\vec{e}_4) + \ldots + w_{\text{side}}(\vec{e}_{2l})\big) \\
\implies \quad & w_{\text{side}}(\vec{e}_1) + w_{\text{side}}(\vec{e}_3) + \ldots + w_{\text{side}}(\vec{e}_{2l-1}) + w_{\text{side}}(\vec{e}_2) + w_{\text{side}}(\vec{e}_4) + \ldots + w_{\text{side}}(\vec{e}_{2l}) \neq 0.
\end{aligned}$$

Since for all planar edges $\vec{e}$, $w_{\text{side}}(\vec{e}) = 0$,

$$\sum_{i=1}^{k} w_{\text{side}}(\vec{C_i''}) \neq 0.$$

Thus there exist some $i \in [k]$ such that

$$w_{\text{side}}(\vec{C_i''}) \neq 0 \implies w_{\text{comb}}(\vec{C_i''}) \neq 0, \tag{5}$$

Note that $\vec{C_i'}$ and $\vec{C_i''}$ for all $i \in [k]$, are the directed cycles such that their underlying undirected cycle is same. In a directed cycle there are only two directions possible. Therefore, we can say that

$$\begin{aligned}
& \vec{C_i'} = \vec{C_i''} \text{ or } \vec{C_i''}^{r}, \\
\Rightarrow \quad & w_{\text{comb}}(\vec{C_i'}) = w_{\text{comb}}(\vec{C_i''}) \text{ or } w_{\text{comb}}(\vec{C_i''}^{r}), \\
\Rightarrow \quad & w_{\text{comb}}(\vec{C_i'}) = w_{\text{comb}}(\vec{C_i''}) \text{ or } -w_{\text{comb}}(\vec{C_i''}), \text{ for all } i \in [k], \text{ since } w_{\text{comb}} \text{ is} \\
& \text{skew-symmetric.}
\end{aligned} \tag{6}$$

From Equation 5 and 6 we can conclude that there exists some $i \in [k]$ such that $w_{\text{comb}}(\vec{C_i'}) \neq 0$, which is a contradiction with Equation 3. Thus we can conclude that there can not exist two minimum weight perfect matchings in a class A_i for all $i \in [2^{2g}]$. This finishes the proof of Lemma 4. ◀

Note that we have proved that there is at most one minimum weight perfect matching in each class and there are total 2^{2g} many classes. Therefore, we can say that there are at most 2^{2g} minimum weight matchings in $\vec{G}$ with respect to the weight function w_{comb}. As we mentioned in Section 3 that given a weight function w_{comb} for a directed bipartite graph $\vec{G}$ such that edges of $\vec{G}$ are directed from L to R, we can get a weight function $w^{\text{und}}_{\text{comb}}$ for underlying undirected graph G such that if $\vec{M}$ is a matching of weight t in $\vec{G}$ then M will be a matching of weight t in G.

▶ **Lemma 10.** *Given a g-genus graph G along with its polygonal schema we can construct a weight function w^{und}_{comb} for G in logspace such that there are at most 2^{2g} minimum weight perfect matchings in G with respect to w^{und}_{comb}.*

Now that given an undirected graph G we have obtained at most 2^{2g} many minimum weight perfect matchings in G, we will use the following hashing scheme by Fredman, Komlós and Szemerédi [8] to isolate a minimum weight perfect matching among them. Let us first state their result in a form suitable to our purpose.

▶ **Theorem 11.** *[8] Let $S = \{x_1, x_2, \ldots, x_k\}$ be a set of n-bit integers. Then there exists a $O(\log n + \log k)$-bit prime number p so that for all $x_i \neq x_j \in S$, $x_i \bmod p \neq x_j \bmod p$.*

Let $\mathcal{M}$ be the set of minimum weight perfect matchings in G with respect to $w^{\text{und}}_{\text{comb}}$. Assume edges of the graph G are indexed as $e_1, e_2, \ldots, e_m$. Let w_b be a weight function that assigns weight 2^i to the edge e_i. This is an m-bit weight function, where $m \leq n^2$. All matchings in G get different weight with respect to this weight function therefore, any two matchings $M_1, M_2 \in \mathcal{M}$, $w_b(M_1) \neq w_b(M_2)$. Also, note that $|\mathcal{M}| \leq 2^{2g}$, because each class has at most one minimum weight perfect matching. Thus by Theorem 11 there exists an $O(\log n + g)$-bit prime p such that with respect to weight function $w_{\text{fks}} := w_b \bmod p$, every matching in $\mathcal{M}$ gets a different weight. Hence our final min-isolating weight function for G will be,

$$w_p := w^{\text{und}}_{\text{comb}} \cdot n^{10} + w_{\text{fks}},$$

Note that for every $O(\log n + g)$-bit prime p we get a corresponding weight function w_p and by Theorem 11 we know that there will be at least one $O(\log n + g)$-bit prime p_1 such that w_{p_1} isolates a minimum weight perfect matching in G. Thus we can conclude the following.

▶ **Theorem 12.** *Given a g-genus graph along with its polygonal schema, we can construct weight functions $w_1, w_2, \ldots, w_k$ in $O(\log n + g)$ space such that if graph has a perfect matching then for some $i \in [k]$ and, G has a unique perfect matching M of weight j with respect to weight function w_i, where $j, k = O(n^c + 2^g)$ for some constant $c > 0$.*

For a graph of genus $g = O(\log n)$ we get polynomially many weight function $w_1, w_2, \ldots w_t$ where $t = O(n^c)$ for some constant c, such that each w_i is polynomially bounded and there is a unique minimum weight perfect matching in graph with respect to at least one w_i if G has a perfect matching. Then we apply the algorithm given in [2] to get an SPL algorithm for perfect matching in $O(\log n)$ genus bipartite graphs.

References

1 Manindra Agrawal, Thanh Minh Hoang, and Thomas Thierauf. The polynomially bounded perfect matching problem is in NC ^{2}. In *STACS 2007, 24th Annual Symposium on Theoretical Aspects of Computer Science, Aachen, Germany, February 22-24, 2007, Proceedings*, pages 489–499, 2007. URL: https://doi.org/10.1007/978-3-540-70918-3_42, doi:10.1007/978-3-540-70918-3_42.

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 Rahul Arora, Ashu Gupta, Rohit Gurjar, and Raghunath Tewari. Derandomizing isolation lemma for $k_{3,3}$-free and k_5-free bipartite graphs. In *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, pages 10:1–10:15, 2016.

4 Ashok K. Chandra, Larry J. Stockmeyer, and Uzi Vishkin. Constant depth reducibility. *SIAM J. Comput.*, 13(2):423–439, 1984. URL: https://doi.org/10.1137/0213028, doi:10.1137/0213028.

5 Samir Datta, Raghav Kulkarni, Raghunath Tewari, and N.V. Vinodchandran. Space complexity of perfect matching in bounded genus bipartite graphs. *Journal of Computer and System Sciences*, 78(3):765 – 779, 2012. In Commemoration of Amir Pnueli. URL: http://www.sciencedirect.com/science/article/pii/S002200001100136X, doi:https://doi.org/10.1016/j.jcss.2011.11.002.

6 Jack Edmonds. Paths, trees and flowers. *CANADIAN JOURNAL OF MATHEMATICS*, pages 449–467, 1965.

7 Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-nc. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 754–763. ACM, 2016. URL: https://doi.org/10.1145/2897518.2897564, doi:10.1145/2897518.2897564.

8 Michael L. Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with 0(1) worst case access time. *J. ACM*, 31(3):538–544, June 1984. URL: http://doi.acm.org/10.1145/828.1884, doi:10.1145/828.1884.

9 Dima Grigoriev and Marek Karpinski. The matching problem for bipartite graphs with polynomially bounded permanents is in NC (extended abstract). In *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*, pages 166–172, 1987. URL: https://doi.org/10.1109/SFCS.1987.56, doi:10.1109/SFCS.1987.56.

10 Chetan Gupta, Vimal Raj Sharma, and Raghunath Tewari. Reachability in o(log n) genus graphs is in unambiguous logspace. In *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, pages 34:1–34:13, 2019. URL: https://doi.org/10.4230/LIPIcs.STACS.2019.34, doi:10.4230/LIPIcs.STACS.2019.34.

11 Thanh Minh Hoang. On the matching problem for special graph classes. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, USA, June 9-12, 2010*, pages 139–150, 2010. URL: https://doi.org/10.1109/CCC.2010.21, doi:10.1109/CCC.2010.21.

12 Vivek Anand T. Kallampally and Raghunath Tewari. Trading determinism for time in space bounded computations. In *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, pages 10:1–10:13, 2016. URL: https://doi.org/10.4230/LIPIcs.MFCS.2016.10, doi:10.4230/LIPIcs.MFCS.2016.10.

13 László Lovász. On determinants, matchings, and random algorithms. In *FCT*, pages 565–574, 1979.

14 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987. URL: https://doi.org/10.1007/BF02579206, doi:10.1007/BF02579206.

15 Omer Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4), 2008.

16 Klaus Reinhardt and Eric Allender. Making nondeterminism unambiguous. *SIAM J. Comput.*, 29(4):1118–1131, 2000. URL: http://dx.doi.org/10.1137/S0097539798339041, doi:10.1137/S0097539798339041.

17 Ola Svensson and Jakub Tarnawski. The matching problem in general graphs is in quasi-nc. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 696–707. IEEE Computer Society, 2017. URL: https://doi.org/10.1109/FOCS.2017.70, doi:10.1109/FOCS.2017.70.

18 Raghunath Tewari and N. V. Vinodchandran. Green's theorem and isolation in planar graphs. *Inf. Comput.*, 215:1–7, 2012. URL: https://doi.org/10.1016/j.ic.2012.03.002, doi:10.1016/j.ic.2012.03.002.

19 Dieter van Melkebeek and Gautam Prakriya. Derandomizing isolation in space-bounded settings. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga,*

Latvia, pages 5:1–5:32, 2017. URL: `https://doi.org/10.4230/LIPIcs.CCC.2017.5`, doi: `10.4230/LIPIcs.CCC.2017.5`.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il