Space-Efficient Approximation Scheme for Maximum Matching in Sparse Graphs*

Samir Datta[1], Raghav Kulkarni[2], and Anish Mukherjee[3]

1 **Chennai Mathematical Institute, India**
`sdatta@cmi.ac.in`
2 **Chennai Mathematical Institute, India**
`kulraghav@gmail.com`
3 **Chennai Mathematical Institute, India**
`anish@cmi.ac.in`

Abstract

We present a Logspace Approximation Scheme (LSAS), i.e. an approximation algorithm for maximum matching in planar graphs (not necessarily bipartite) that achieves an approximation ratio arbitrarily close to one, using only logarithmic space. This deviates from the well known Baker's approach for approximation in planar graphs by avoiding the use of distance computation - which is not known to be in Logspace. Our algorithm actually works for any "recursively sparse" graph class which contains a linear size matching and also for certain other classes like bounded genus graphs.

The scheme is based on an LSAS in bounded degree graphs which are not known to be amenable to Baker's method. We solve the bounded degree case by parallel augmentation of short augmenting paths. Finding a large number of such disjoint paths can, in turn, be reduced to finding a large independent set in a bounded degree graph. The bounded degree assumption allows us to obtain a Logspace algorithm.

1 Introduction

Historically, matching problems have played a central role in Algorithms and Complexity Theory. Edmond's blossom algorithm [14] for maximum matching was one of the first examples of a non-trivial polynomial time algorithm. It had a considerable share in initiating the study of efficient computation, including the class P itself; Valiant's #P-hardness [32] for counting perfect matchings in bipartite graphs provided surprising insights into the counting complexity classes. The rich combinatorial structure of matching problems combined with their potential to serve as central problems in the field invites their study from several perspectives.

The study of whether matching is parallelizable has yielded powerful tools, such as the isolating lemma [27], that have found numerous other applications. The RNC bound remains the best known parallel complexity for maximum matching till date. The best known upper bound for Perfect-Matching is non-uniform SPL[1] whereas the best hardness known is NL-hardness [8].

Matching in Planar Graphs

A well known example where planarity is a boon is that of counting perfect matchings. The problem in planar graphs is in P [21] and can in fact be done in NC[33]; thus Perfect-Matching (Decision) in planar graphs is in NC. "Is the construction version of Perfect-Matching in planar graphs in NC ?" remains an outstanding open question, whereas the bipartite planar case is known to be in NC [26, 25, 23, 11].

* The third author was partially supported by a TCS PhD fellowship. The first and the third authors were partially funded by a grant from Infosys foundation.

41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016).
Editors: Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier; Article No. 28; pp. 28:1–28:12
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The space complexity of matching problems in planar graphs was first studied by Datta, Kulkarni, and Roy [11] where it is shown that minimum weight Perfect-Matching (Min-Wt-PM) in bipartite planar graphs is in SPL. Computing a maximum matching for bipartite planar graphs is shown to be in NC by Hoang [16]. Kulkarni [22] shows that Min-Wt-PM in planar graphs (not necessarily bipartite) is NL-hard. The only known hardness for Perfect-Matching in planar graphs is L-hardness (cf. [10]).

1.1 Motivation

Time efficient approximation algorithms are well studied and have a lot of applications. Space is arguably the second most important resource other than time. Although there is an abundance of work on time efficient approximation, work on space efficient approximation seems limited. To the best of our knowledge even some basic problems such as maximum matching have not been considered. Notice that for (the construction version of) this well studied problem we know of no better complexity bound than $\mathsf{P} \cap \mathsf{RNC}$ [14, 27, 20] even in the planar case. In particular we do not know if it is in SC or NC.

Bounded space approximation algorithms in the presence of non-determinism can be obtained by using Baker's approach [4] for some problems in certain sparse graphs, the most prominent being planar graphs. Dispensing with non-determinism in algorithms even for reachability (not to say matching) leads to either a quasipolynomial blow-up in the time requirement via Savitch's theorem [30] or a large space footprint ($O(\sqrt{n})$) if we want to simultaneously keep the algorithms in polynomial time (see e.g. [19, 3] for reachability in planar graphs). For general graphs the tradeoff at the low space side is even worse, with $O(\frac{n}{2^{\sqrt{\log n}}})$ space and polynomial time [5].

In the context of simultaneous polylogarithmic space and polynomial time (i.e. the class SC), Logspace is the gold-standard and therefore a Logspace Approximation Scheme is the desired result we are able to achieve for planar graphs. An LSAS for bounded degree graphs and a plethora of related graph classes is a serendipitous side effect.

1.2 Previous Work

The problem of approximating maximum matching has been considered both in time and parallel complexity model. [13] gives a linear-time approximation scheme for maximum matching which has the best known time complexity. An NC approximation scheme for maximum matching is given in [18].

Two papers [31, 36] have strived to rephrase Logspace approximation algorithms in the general approximation framework. Their well directed efforts need to be augmented with more concrete problems.

In this direction [9] studied planar MaxCut and related problems in the context of approximation but had to be satisfied with a $\mathsf{UL} \cap \mathsf{co\text{-}UL}$ approximation scheme which closely follows Baker's approach and is unsatisfactory since it uses non-determinism.

The folklore randomized algorithm for a 1/2-approximation to MaxCut and which can be derandomized in L, with the help of pair-wise independence, is another example in the same spirit.

1.3 Our Results

In this work we first show that there is a Logspace Approximation Scheme for maximum matching in bounded degree graphs.

▶ **Theorem 1.** *Let G be a graph with degrees bounded by a constant d then for any fixed $\epsilon > 0$, we can find a $(1-\epsilon)$ factor approximation to the maximum matching in Logspace.*

The main fact we use here is that any bounded degree graphs (assuming it's connected) always contains a linear size matching. Many planar graph classes are known to have the property of containing a large matching. Such classes include 3−connected planar graphs [7]. In fact our algorithm works for any *recursively sparse* graph containing a large matching.

Next we show that we can actually give Logspace Approximation Scheme for maximum matching in any planar graph by reducing it to the bounded degree graphs by suitable modifications.

▶ **Theorem 2.** *Let G be a planar graph then for any fixed $\epsilon > 0$, we can find a $(1 - \epsilon)$ factor approximation to the maximum matching in Logspace.*

This result extends to many other graph classes, namely for classes which are "biparted" i.e. sparse graphs with bipartite graphs in the class being even *significantly sparser* such as: in bounded genus graphs, k-Apex graphs, (g, k)-Apex graphs, 1-planar graphs, k-page graphs.

Notice that while some of our ideas are similar to the classical sequential algorithm of Hopcroft and Karp [17] for maximum matching in bipartite graphs, we consider graphs which are not necessarily bipartite. Our algorithm trades off Logspace and non-bipartiteness for approximation and sparsity.

1.4 Our Techniques

The primary algorithmic tool is augmentation along short augmenting paths. We prove that in a bounded degree graph, if there are many unmatched but matchable vertices remaining there exist precisely linearly many short augmenting paths. We need to pick a large subset of independent ones from these.

This prompts us to find a large independent set in a bounded degree graph *that works in Logspace*. Notice that the simple greedy strategy that removes a least degree vertex and its neighbourhood will find a linear sized independent set but the algorithm is not implementable in Logspace.

The above method needs the graph to be bounded degree. To convert a planar graph to a bounded degree graph we simply delete high degree vertices and show that this does not affect the size of the matching considerably since the number of high degree vertices is small though possibly still linear in the graph size. This will work if we are sure that the size of the maximum matching is at least linear.

Next we work to whittle the graph down to one containing a linear sized matching without reducing the matching size. We show that removing some small number of vertices ensures this. The proof of this part is based on a lengthy case analysis.

1.5 Organization

After some preliminaries in Section 2, we describe in Section 3 the approximation algorithm for bounded degree graphs where they contain a large (linear in the number of vertices) matching. In Section 4 we then show that our algorithm can be extended for planar graphs also. We conclude in Section 5 with some open ends.

2 Preliminaries

A graph $G = (V, E)$ consists of a finite set of vertices $V(G) = V$ and edges $E(G) = E \subseteq V \times V$.

The class L is the class of languages accepted by deterministic logspace Turing machines. We know that undirected graph connectivity is in L[29]. For the definition of other complexity classes we refer the reader to any standard text book, for example [34, 2]. The concept of Logspace transducer is implicit in Definition 4.16 of [2] and is made explicit in Exercise 4.8 from the same text.

A matching in G is a set $M \subseteq E$, such that no two edges in M have a vertex in common. A matching M is called *perfect* if M covers all vertices of G, M of maximum size is called *maximum* matching. Vertices not incident to an M edge are *free*. An alternating path is one whose edges alternate between M and $E \setminus M$. An alternating path P is augmenting if P begins and ends at free vertices, that is, $M \oplus P = (M \setminus P) \cup (P \setminus M)$ is a matching with cardinality $|M \oplus P| = |M| + 1$. For a complete treatment on matching see [24].

An *independent set* is a set of vertices in a graph, no two of which are adjacent. A *maximum* independent set is an independent set of largest possible size in a given graph. A (vertex) colouring of a graph is an assignment of labels (called "colours") to the vertices of a graph such that no two adjacent vertices share the same color.

An *induced* subgraph of a graph is another graph, formed from a subset of the vertices of the graph and all of the edges connecting pairs of vertices in that subset. An *induced path* is a path that is an induced subgraph. A graph is called *recursively sparse* if every subgraph of it is a sparse graph.

▶ **Definition 3** (Approximation Ratio)**.** We call an algorithm A a β-approximation algorithm if, on every instance I, the algorithm outputs a set I_A such that $1/\beta \cdot I_A \leq I_{Opt} \leq \beta \cdot I_A$ where I_{Opt} is the optimal result on the instance I. The β is called the approximation ratio (or approximation factor) of the algorithm.

▶ **Definition 4** (Approximation Scheme)**.** Let X be a minimization (respectively, maximization) problem.

- An approximation scheme is a family of $(1 + \epsilon)$-approximation algorithms A_ϵ (respectively, $(1 - \epsilon)$-approximation algorithms A_ϵ) for problem X for any $0 < \epsilon < 1$.
- A Logspace approximation scheme (LSAS) for problem X is an approximation scheme which runs in Logspace.

For a more general treatment of LSAS, consult [36, 31].

A planar graph is a graph that can be embedded in the plane, i.e., the edges can be drawn on the plane in such a way that no edges cross each other (i.e.the edges intersect only at their endpoints). A graph G is said to have *genus g* if G has a minimal embedding (an embedding where every face of G is homeomorphic to a disc) on a genus g surface. Euler's formula for a genus g graph states that $\chi(g) = |V| - |E| + |F|$ where $\chi(g) = 2 - 2g$ and $|F|$ is the number of faces of G. For planar graphs, this implies $|E| \leq 3n - 6$ and so the *average degree* of a planar graph is at most 6. See standard texts on Graph theory (e.g. [12, 35]) for further information. Consult [28] for definitions and properties of various other sparse graph classes.

3 Approximating maximum matching in bounded degree graphs

In this section we show that given any bounded degree graph, we can give a Logspace approximation scheme for the maximum matching.

Our strategy is to design a Logspace transducer that takes in a bounded degree graph and a matching therein as input and while the matching has size significantly smaller than the size of the maximum matching finds a number of disjoint augmenting paths that can then be augmented in parallel in Logspace. The output of the transducer is thus a somewhat larger matching - in fact a matching which is larger than the previous matching by a constant fraction of the maximum matching. We are of course assuming that we are not already very close to the optimal matching. Since we can compose constantly many Logspace transducers to yield another Logspace transducer we are done.

All the augmenting paths we deal with are short i.e. of length at most $2k + 1$ for some constant k. This is because such paths can be found in Logspace by say exhaustively listing all $(2k + 1)$-tuples of vertices and checking if they form valid augmenting paths.

These short augmenting paths are at most linearly many in n, at most $n(2k+1)^2 d^{2k+1}$ to be more precise where d is an upper bound on the maximum degree. Now suppose that the current matching cardinality differs significantly from the maximum matching size $|M_{opt}|$ (by a factor $\Omega(1/k)$ of the maximum matching) then we show that there are at least $\Omega(|M_{opt}|/k)$ many augmenting paths of length $2k+1$ (which happen to be disjoint - though this fact is not used subsequently).

Having demonstrated that there exist lots of paths, we have to find a large fraction in Logspace, which are mutually disjoint. If we form an intersection graph of these short augmenting paths by making two paths adjacent if they have a vertex in common, then we are looking for a large independence set in this intersection graph. We would be done if we can colour the paths with $O(1)$ colours (so that no two intersecting paths get the same colour) because then the largest colour class serves as the desired constant fraction independence set. Since the original graph is bounded degree so is the intersection graph - so it is, at least existentially, $O(1)$-colourable. We in fact show how to constant colour this graph in Logspace.

3.1 Lower bounding the number of short paths

Let $G = (V, E)$ (where $n = |V|$) be the given bounded degree graph with an upper bound of d on the degrees. Let M_{opt} be an optimal maximum matching contained in G. Let M be any other matching which is not necessarily maximum. We assume that the gap $|M_{opt}| - |M|$ is sufficiently large so that lot of augmenting paths exist since the number of unmatched but matchable vertices is large. Yet conceivably very few or none of these paths may be short. Because we can only hope to explore augmenting paths of a constant length in L such a possibility would be very injurious to the approach. Fortunately, we can show that as long as we are not very close to the maximum matching there are *many* short augmenting paths that survive. The following lemma is an adaptation of Corollary 2 of [17] tailored for augmenting paths of constant length where the number of such paths is also important to us.

▶ **Lemma 5.** *If $|M| < (1-\frac{3}{k})|M_{opt}|$ for some positive integer k then there are at least $3|M_{opt}|/2k$ augmenting paths consisting of at most $2k+1$ edges.*

Proof. The maximum number of vertices that can be matched in any matching is precisely, $2|M_{opt}|$. The symmetric difference $M \oplus M_{opt}$ consists of $|M_{opt}| - |M|$ augmenting paths and a number of alternating cycles, which are all mutually disjoint. Suppose the length of the i-th augmenting path is ℓ_i. Then $\sum_{i=1}^{|M_{opt}|-|M|}(\ell_i - 1) \leq 2|M_{opt}|$. This is because an augmenting path of length ℓ contains $\ell - 1$ matched vertices which are distinct across other paths. Thus, $(|M_{opt}| - |M|)\ell_{avg} \leq 3|M_{opt}| - |M| \leq 3|M_{opt}|$ where ℓ_{avg} is the average path length. Thus, $\ell_{avg} \leq k$.

Since at least half fraction of the paths have length at most double the average, we get that at least $3|M_{opt}|/2k$ paths have length at most $2k$. ◀

3.2 Approximating Maximum Independent Set

As graph G still contains a large set of (linearly many) disjoint augmenting paths, we find a constant factor approximation to the maximum independent set in intersection graph of bounded length augmenting paths of G.

Let H be the intersection graph of augmenting paths of length at most $2k+1$ in G.

▶ **Lemma 6.** *A β-factor approximation to the maximum independent set in the graph H can be computed in L where $\beta = 2^{-(2k+1)^2 d^{2k+1}}$*

Proof. The graph H has maximum degree upper bounded by $D = (2k+1)^2 d^{2k+1}$ since there are at most $d^i d^{2k+1-i} = d^{2k+1}$ paths in which a fixed vertex appears as the i-th vertex[1]. Since d, k are constants D is also a constant. If we can colour the intersection graph by at most $f(D)$ colours then we would be done because the largest colour class will be a constant (say β) factor approximation to the maximum independent set. Now we give a simple procedure to do this.

For a graph with maximum degree bounded by D, we can find at most D disjoint forests that partition the edge set. This can be done by running Reingold's algorithm for undirected connectivity [29] at most D times on the graph. Now we colour each forest with 2 colours and it gives D bit colours to every node (1 bit for every colouring). This yields an $f(D) = 2^D$ colouring of the graph because two vertices that are adjacent must belong to at least one common forest. ◀

▶ **Theorem 7.** *Let G be a graph with degrees bounded by a constant d then for any fixed $\epsilon > 0$, we can find a $(1-\epsilon)$ factor approximation to the maximum matching in Logspace.*

Proof. Fix integer $k = \lceil \frac{3}{\epsilon} \rceil$. If the current matching is of size at most $(1 - 3/k)$ fraction of the maximum matching there are a lot (at least $|M_{opt}|/2k$ from Lemma 5) of augmenting paths of length $2k+1$ remaining. Thus the number of vertices in H is at least linear in $|M_{opt}|$.

By Lemma 6 we can find an independent set of size at least $\beta|V(H)| = \beta|M_{opt}|/2k$. This yields a linear number in the size of the maximum matching, of short (length $\leq 2k+1$) augmenting paths which are vertex disjoint and thus are augmentable in parallel. In fact a L-transducer can do the augmentation and output the new matching (it just has to interchange the matched and the unmatched edges in every picked augmenting path).

At every step we increase the matching size by an additive term of $|M_{opt}|/(2k/\beta)$ (unless we get closer than a factor of $(1-3/k)$ to the maximum matching). We chain $(1-3/k)2k/\beta$ such transducers. Note that since we start with an empty matching, after K rounds the approximation ratio would be at least $(1-3/k)$. Thus we get an approximation ratio of at least $(1-3/k) \leq 1-\epsilon$. ◀

4 Approximating Planar Maximum Matching

In this section we show that we can give Logspace Approximation Scheme for finding maximum matching in planar graphs using the LSAS for bounded degree graphs. We first show that a *tame* graph and so a minimum degree 3 planar graph contains a linear size matching in Subsection 4.1. In Subsection 4.2 we describe the Logspace Approximation Scheme.

4.1 Existence of a linear matching subgraph

We say that a maximal induced path is k-isolated if its length is $k > 1$ edges and each of its $(k-1)$-internal vertices have degree precisely two in G. A k-isolated path is long if $k > 2$. An endpoint of an isolated path is called a branch vertex if its degree is G is at least 3 and a pendant vertex if its degree is 1.

Consider the set P of all isolated paths in a graph G. Let P_0 represent the paths in P which contain an even number of edges. Let B_0 represent the set of pairs of endpoints of all the paths in P_0 which support at least two paths from P_0. For each pair in B_0 pick exactly two paths from P_0 supported by vertices of B_0 to yield set P_0'. Let E_0 be the set of extreme[2] edges of all paths in $P_0 \setminus P_0'$.

▶ **Definition 8.** A graph is *tame* if all pairs in B_0 support exactly two paths from P_0.

---

[1] This is a very crude upper bound which does not take into account that the $2k+1$ length length path is augmenting so the bound of $k^2 d^k$ is closer to truth. Our bound however suffices for the purpose at hand

[2] i.e. the first and the last

We can use the following Lemma to compress the graph preserving maximum matching size:

▶ **Lemma 9.** *The size of the maximum matching in $G - E_0$ is the same as in G.*

Proof. Every matching in $G - E_0$ is a matching in G. Thus we just need to prove that for any maximum matching in G there is a matching in in $G - E_0$ of the same cardinality. To see this notice that for any pair $\{u, v\} \in B_0$ in any matching M it is the case that $0, 1$ or 2 edges from E_0 are used. If 1 or 2 edges of E_0 are used in the matching then 1 or 2 paths, respectively, in P'_0 which are incident on u, v have at least one unmatched vertex (because they contain odd number of vertices apart from their externally matched endpoints). Switching the $1, 2$ matched edges incident on u, v to these $1, 2$ paths in P'_0 so that the unmatched vertices on these paths are matched we reach a matching with the same cardinality as M. ◀

Notice that for a tame graph there may be zero, one or two isolated even length paths between any pair of vertices. Removing the edges in E_0 ensures that we are left with a tame graph. The following is the property of tame graphs that we plan to exploit:

▶ **Lemma 10.** *A tame planar graph has a linear sized maximum matching.*

Proof. Let N_0 be a yet to be fixed threshold[3]. We use a case analysis:

1. The total length of long isolated paths $N \geq N_0$. We have a matching of size at least $N_0/4$ in this case by Lemma 11.
2. The total length of long isolated paths $N < N_0$: In this case for every pair of endpoints of long paths.
   a. We replace each such long path by a path of length 2 or 3 depending on whether the path was even and odd. This reduces the max matching size by at most $N/2$ without increasing the number of vertices.
   b. If there are more than 2 paths of length 3 between u, v then delete all but 2. This further reduces the max matching size by at most 2ν without increasing the number of vertices. Here ν is the number of odd paths in the initial graph. Thus the loss in matching in this step is at most $2N/3$.
   c. Attach the Lollipop graph (i.e. a K_4 with a pendant edge attached to one of the vertices) to each of the 2 internal vertices of the 3-isolated paths. This does not decrease the matching size. The number of vertices goes up by at most $4N$. In the resulting graph only 2-isolated paths have degree 2 vertices.
      i. If there are at least $N' \geq N'_0$ isolated 2-paths in the graph.
         A. Consider the subgraph of this graph where all edges not incident on vertices of degree 2 have been deleted and all isolated vertices formed as a result have been deleted. The resulting subgraph has at least $2N'$ edges and N' (degree 2) vertices.
         B. Find a spanning forest of this graph and root every tree in the forest at a vertex which wasn't a degree 2 vertex in G. It is easy to see that all the vertices of degree 2 in G are matchable in the forest - just match them to their unique child in the rooted forest. Thus a matching size of $N' \geq N'_0$ is guaranteed in G.
      ii. If there are at most $N' < N'_0$ isolated 2-paths in the graph.
         A. Attach the Lollipop graph to each degree 2 vertex of the graph. This does not decrease the matching size and increases the number of vertices by at most $4N'$. We obtain a min degree 3 graph.

[3] which will turn out to be $n/35$

Thus we have a matching of size at least min $(N_0/4, N_0', m - (N_0/2 + 2N_0/3))$ and the number of vertices is at most $n + 4N_0 + 4N_0'$ in the last case of the minimum. Since the ratio of matching edges and vertices cannot be better than $1/140$ from Lemma 12, we just need to assume that: $N_0/4 \geq n/140, N_0' \geq n/140, (m - 5N_0/6) \geq (n + 4N_0 + 4N_0')/140$. Taking $N_0 = 4n/140 = n/35$ and $N_0' = n/140$, we get:

$$m - n/42 \geq (n + n/7)/140$$

or

$$m \geq n/42 + n/140 + n/980 > n/140.$$

Thus, overall, $m \geq n/140$. ◀

▶ **Lemma 11.** *A graph in which the total length of long isolated paths is N has a matching of size at least $N/4$.*

Proof. Let the sum of lengths, number of odd, even isolated paths be denoted by respectively N_{odd}, N_{even} and ν_{odd}, ν_{even} An isolated path of odd length N_i has a matching of size at least $(N_i - 1)/2$ (leaving out the two extreme edges). Similarly, an even length isolated path has a matching of size at least $N_i/2 - 1$. Thus the size of a matching from long odd isolated paths is at least $N_{odd}/2 - \nu_{odd}/2$ and from even isolated paths is at least $N_{even}/2 - \nu_{even}$. Now each long even isolated path has length at least 4 so $4\nu_{even} \leq N_{even}$ and each long odd isolated path has length at least 3 so that $3\nu_{odd} \leq N_{odd}$. Thus the total size of matchings is at least

$$\sum_i N_i/2 - \nu_{odd}/2 - \nu_{even} \geq N/2 - N_{odd}/6 - N_{even}/4 \geq N/2 - N_{odd}/4 - N_{even}/4 = N/4$$

◀

▶ **Lemma 12.** *A minimum degree 3 planar graph has a matching of size at least $n/140$.*

Proof. Consider the set S of all vertices of degree at least d in G. Let S_0 be the isolated vertices in $G - S$ i.e. those vertices in $V(G) - S$ all whose neighbours are in S. Consider the bipartite graph G' with bipartitions S_0, S where we connect a vertex $u \in S_0$ to all $v \in S$ such that $(u, v) \in E(G)$. Now the number of edges incident on S_0 is at least $3|S_0|$ (because every edge incident on $u \in S_0$ is still present in G'). On the other hand, the number of edges from average degree is at most $2(|S| + |S_0|)$. Thus $|S_0| \leq 2|S|$. But $|S| \leq 6n/d$. thus together the number of vertices deleted is at most $3|S| = 18n/d$. Hence the number of remaining vertices is at least $(1 - 18/d)n$.

Now suppose the graph has c components. Find a spanning forest of this graph. Vertices in each spanning tree have degree at most $d - 1$. Then,

▶ Claim 1. Any tree on n vertices and maximum degree d supports a matching of size at least $(n - 1)/d$.

Proof. To see this fix a root to the tree and consider a deepest leaf v in the tree. Remove the other endpoint w of the pendant edge (v, w) leads to a tree containing d lesser vertices. Since at the end we might be left with just the root as an isolated vertex, the claimed bound follows. ◀

Thus the tree supports a matching of size at least $(n'' - 1)/(d - 1)$ where n'' is the number of vertices in the component. Therefore the total size of the matching is at least $(n' - c)/(d - 1)$ where $n' \geq (1 - 18/d)n$ is the number of vertices spanned by the forest. Since none of the components is a singleton it must be that $c \leq n'/2$. So the size of the maximum matching is at least $(1 - 18/d)(1/(2d - 2))n$. Putting $d = 36$, we get that the size of the maximum matching is at least $n/140$. ◀

4.2 Finding a large planar matching

▶ **Theorem 13.** *There is a Logspace Approximation Scheme for maximum matching in planar graphs.*

Proof. We first convert the original graph G into a tame graph G' by using Lemma 9. This preserves the maximum matching size. Suppose there are least αn matching edges in G' for some $\alpha < 1/2$. Fix a positive $\epsilon < \alpha$.

We will delete all vertices of degrees greater than d from G' to yield graph G'' which is of degree bounded by d. Since the number of vertices of degree at least d in G' is at most $6n/d$, the number of matching edges removed by deleting the high degree vertices is at most $6n/d$. So we will have a large $= (\alpha - 6/d)n$ sized matching remaining after this if $\alpha - 6/d = \epsilon/2$ i.e. $d = \frac{12}{2\alpha-\epsilon}$. Thus it suffices to find a $(1 - \epsilon/2)$ factor approximation to the maximum matching using Theorem 7 in Logspace. ◀

Notice that, here we had to tame the graph only to ensure the existence of a linear size matching. But given promise that the graph contains a linear size matching, we can get a approximation scheme, for any recursively sparse graph, without taming it.

▶ **Corollary 14.** *There is a Logspace Approximation Scheme for maximum matching in recursively graphs which contains a linear size matching.*

▶ Note 1. We require only the following properties of planar graphs in proving Lemma 12:

- Sparsity: The average degree is upper bounded by 6.
- Bipartite sparsity: The average degree of every bipartite subgraph is even lower i.e. 4.
- Min-degree: The minimum degree is at least 3 i.e. at least half the average degree.

Thus the proof of Lemma 12 goes through for any family of graphs satisfying these properties. Also notice that Lemma 9 works for arbitrary graphs and Lemma 10 works for any family of graphs satisfying the first two properties above. Hence we also get Logspace Approximation Schemes for the following families of graphs [15]:

1. Genus g graphs: graphs that are embeddable on a surface of genus $g = O(1)$.
2. k-Apex graphs: graphs such that deleting k vertices leads to planar graphs.
3. (g, k)-Apex graphs: graphs such that deleting k vertices leads to genus g graphs.
4. 1-planar graphs: graphs that can be drawn with at most one crossing per edge.
5. k-page graphs: graphs such that all edges can be accommodated on a k-page book with vertices on the spine.

4.3 The Algorithm

Here we present the full algorithm for finding the approximate maximum matching. First we present the algorithms for finding the approximate maximum matching in bounded degree graphs in Algorithm 1 and then we present our main algorithm, describing the *taming* procedure and using the previous algorithm as subroutine, in Algorithm 2.

5 Conclusion and Open-Ends

The main open question which remains is to show that whether we can devise an LSAS for maximum matching in *general* graphs or at least in arbitrary sparse graphs. In this work, we have been able to resolve this for bounded degree graphs, planar graphs and some related classes of sparse graphs.

Algorithm 1 (Matching in Bounded Degree Graphs)

Input : (G, ϵ, M) where $G = (V, E)$ is bounded degree graph with $deg(v) \leq d$ for all $v \in V, \epsilon > 0$ and M is a set of matched edges.
Output : A set $M' \subseteq E$ of matched edges.

Fix integer $k = \lceil \frac{3}{\epsilon} \rceil$.
Construct the intersection graph of augmenting paths of length at most $2k + 1$ in G.
Let the graph be H with maximum degree $\leq D = (2k + 1)^2 d^{2k+1}$
Find at most D disjoint forests that partition the edge set.
Colour each forest with 2 colours, giving D bit colours to every node
Augment the vertex disjoint augmenting paths in parallel using L-transducer
Add the new matching to M
return M

Biedl [6] showed that there exists a linear-time (also in Logspace) reduction from maximum matching in arbitrary graphs to maximum matching in 3-regular graphs, though it is not immediate that it is approximation preserving. It will interesting to show such a reduction which is also approximation preserving.

Proving lower bounds for maximum matching in the context of approximation is another important goal. Currently we do not know of any non-trivial hardness results including NC^1-hardness or even TC^0-hardness let alone a L-hardness for approximation to any factor.

Acknowledgements. The first author would like to thank Abhishek Kadian for work on a previous avatar of this paper.

References

1. 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.
2. Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach.* Cambridge University Press, New York, NY, USA, 1st edition, 2009.
3. Tetsuo Asano, David G. Kirkpatrick, Kotaro Nakagawa, and Osamu Watanabe. $\tilde{O}(\sqrt{n})$-space and polynomial-time algorithm for planar directed graph reachability. In *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II*, pages 45–56, 2014. doi:10.1007/978-3-662-44465-8_5.
4. Brenda S. Baker. Approximation algorithms for np-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994. doi:10.1145/174644.174650.
5. Greg Barnes, Jonathan F. Buss, Walter L. Ruzzo, and Baruch Schieber. A sublinear space, polynomial time algorithm for directed *s-t* connectivity. *SIAM J. Comput.*, 27(5):1273–1282, 1998. doi:10.1137/S0097539793283151.
6. Therese C. Biedl. Linear reductions of maximum matching. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA.*, pages 825–826, 2001. URL: http://dl.acm.org/citation.cfm?id=365411.365789.
7. Therese C. Biedl, Erik D. Demaine, Christian A. Duncan, Rudolf Fleischer, and Stephen G. Kobourov. Tight bounds on maximal and maximum matchings. *Discrete Mathematics*, 285(1-3):7–15, 2004. doi:10.1016/j.disc.2004.05.003.
8. Ashok K. Chandra, Larry Stockmeyer, and Uzi Vishkin. Constant depth reducibility. *SIAM Journal on Computing*, 13(2):423–439, 1984. doi:10.1137/0213028.

Algorithm 2 (Main Algorithm)

Input : A graph $G = (V, E)$ and an $\epsilon > 0$.
Output : A set $M \subseteq E$ of matched edges in G such that $|M| \geq (1-\epsilon)|M_{Opt}|$

Let $M = \emptyset$ and $d = 36$
Let P_0 be the set of all isolated paths containing an even number of edges.
Let B_0 be the set of pairs of endpoints of all the paths in P_0 supporting at least two paths from P_0.
$P'_0 = \emptyset$
for each pair $(a, b) \in B_0$ pick exactly two paths from P_0 supported by a, b **do**
    Add the two paths to P'_0
end for
Let E_0 be the set of extreme edges of all paths in $P_0 \setminus P'_0$.
$G = G \setminus E_0$
Remove vertices of degree at least d
Remove all the isolated vertices
Let the modified graph be G'
for $i = 1$ to $(2k-6)/\beta$ **do**
    Call Algorithm 1 on $G', \epsilon/2$
    Remove the matched edges along with endpoints from G'
end for
return M

9 Samir Datta and Raghav Kulkarni. Space complexity of optimization problems in planar graphs. In *Theory and Applications of Models of Computation - 11th Annual Conference, TAMC 2014, Chennai, India, April 11-13, 2014. Proceedings*, pages 300–311, 2014. `doi:10.1007/978-3-319-06089-7_21`.

10 Samir Datta, Raghav Kulkarni, Nutan Limaye, and Meena Mahajan. Planarity, determinants, permanents, and (unique) matchings. *ACM Trans. Comput. Theory*, 1(3):1–20, 2010. `doi:10.1145/1714450.1714453`.

11 Samir Datta, Raghav Kulkarni, and Sambuddha Roy. Deterministically isolating a perfect matching in bipartite planar graphs. *Theory of Computing Systems*, 47:737–757, 2010. `doi:10.1007/s00224-009-9204-8`.

12 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

13 Ran Duan and Seth Pettie. Linear-time approximation for maximum weight matching. *J. ACM*, 61(1):1:1–1:23, 2014. `doi:10.1145/2529989`.

14 J. Edmonds. Paths, trees and flowers. *Canad. J. Math.*, 17:449–467, 1965.

15 David Eppstein. Sparser bipartite graphs? Theoretical Computer Science Stack Exchange. URL: `http://cstheory.stackexchange.com/q/31567`.

16 Thanh Minh Hoang. On the matching problem for special graph classes. In *IEEE Conference on Computational Complexity*, pages 139–150, 2010. `doi:10.1109/CCC.2010.21`.

17 John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973. `doi:10.1137/0202019`.

18 Stefan Hougardy and Doratha E. Drake Vinkemeier. Approximating weighted matchings in parallel. *Inf. Process. Lett.*, 99(3):119–123, 2006. `doi:10.1016/j.ipl.2006.03.005`.

19 Tatsuya Imai, Kotaro Nakagawa, Aduri Pavan, N. V. Vinodchandran, and Osamu Watanabe. An o(n½+$\sum$)-space and polynomial-time algorithm for directed planar reachability. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 277–286, 2013. doi:10.1109/CCC.2013.35.

20 Richard M. Karp, Eli Upfal, and Avi Wigderson. Constructing a perfect matching is in random NC. *Combinatorica*, 6(1):35–48, 1986. doi:10.1007/BF02579407.

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

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

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

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

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

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

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

28 Jaroslav Nesetril and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4.

29 Omer Reingold. Undirected connectivity in log-space. *J. ACM*, 55(4), 2008. doi:10.1145/1391289.1391291.

30 Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. Syst. Sci.*, 4(2):177–192, 1970. doi:10.1016/S0022-0000(70)80006-X.

31 Till Tantau. Logspace optimization problems and their approximability properties. *Theory Comput. Syst.*, 41(2):327–350, 2007. doi:10.1007/s00224-007-2011-1.

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

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

34 Heribert Vollmer. *Introduction to Circuit Complexity: A Uniform Approach.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999.

35 Douglas B. West. *Introduction to Graph Theory.* Prentice Hall, 2 edition, September 2000.

36 Tomoyuki Yamakami. *Combinatorial Optimization and Applications: 7th International Conference, COCOA 2013, Chengdu, China, December 12-14, 2013, Proceedings*, chapter Uniform-Circuit and Logarithmic-Space Approximations of Refined Combinatorial Optimization Problems, pages 318–329. Springer International Publishing, Cham, 2013. doi:10.1007/978-3-319-03780-6_28.