58th Annual IEEE Symposium on Foundations of Computer Science

The Matching Problem in General Graphs is in Quasi-NC

Ola Svensson
EPFL
Lausanne, Switzerland
Email: ola.svensson@epfl.ch

Jakub Tarnawski
EPFL
Lausanne, Switzerland
Email: jakub.tarnawski@epfl.ch

***Abstract*—We show that the perfect matching problem in general graphs is in quasi-NC. That is, we give a deterministic parallel algorithm which runs in $O(\log^3 n)$ time on $n^{O(\log^2 n)}$ processors. The result is obtained by a derandomization of the Isolation Lemma for perfect matchings, which was introduced in the classic paper by Mulmuley, Vazirani and Vazirani [1987] to obtain a Randomized NC algorithm.**

Our proof extends the framework of Fenner, Gurjar and Thierauf [2016], who proved the analogous result in the special case of bipartite graphs. Compared to that setting, several new ingredients are needed due to the significantly more complex structure of perfect matchings in general graphs. In particular, our proof heavily relies on the laminar structure of the faces of the perfect matching polytope.

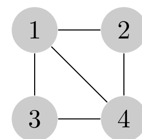
This is an extended abstract. The full version of the paper, which includes all proofs, may be found at https://arxiv.org/abs/1704.01929.

I. Introduction

The perfect matching problem is a fundamental question in graph theory. Work on this problem has contributed to the development of many core concepts of modern computer science, including linear-algebraic, probabilistic and parallel algorithms. Edmonds [1] was the first to give a polynomial-time algorithm for it. However, half a century later, we still do not have full understanding of the deterministic parallel complexity of the perfect matching problem. In this paper we make progress in this direction.

We consider a problem to be efficiently solvable in parallel if it has an algorithm which uses polylogarithmic time and polynomially many processors. More formally, a problem is in the class NC if it has uniform circuits of polynomial size and polylogarithmic depth. The class RNC is obtained if we also allow randomization.

We study the decision version of the problem: given an undirected simple graph, determine whether it has a perfect matching – and the search version: find and return a perfect matching if one exists. The decision version was first shown to be in RNC by Lovász [2]. The search version has proved to be more difficult and it was found to be in RNC several years later by Karp, Upfal and Wigderson [3] and Mulmuley, Vazirani and Vazirani [4]. All these algorithms are randomized, and it remains a major open problem to determine whether randomness is required, i.e., whether either version is in NC.



$$T(G) = \begin{pmatrix} 0 & X_{12} & X_{13} & X_{14} \\ -X_{12} & 0 & 0 & X_{24} \\ -X_{13} & 0 & 0 & X_{34} \\ -X_{14} & -X_{24} & -X_{34} & 0 \end{pmatrix}$$

Figure 1: Example of a Tutte matrix of an undirected graph.

A successful approach to the perfect matching problem has been the linear-algebraic one. It involves the Tutte matrix associated with a graph $G = (V, E)$, which is a $|V| \times |V|$ matrix defined as follows (see Figure 1 for an example):

$$T(G)_{u,v} = \begin{cases} X_{(u,v)} & \text{if } (u,v) \in E \text{ and } u < v, \\ -X_{(v,u)} & \text{if } (u,v) \in E \text{ and } u > v, \\ 0 & \text{if } (u,v) \notin E, \end{cases}$$

where $X_{(u,v)}$ for $(u,v) \in E$ are variables. Tutte's theorem [5] says that $\det T(G) \neq 0$ if and only if G has a perfect matching. This is great news for parallelization, as computing determinants is in NC [6], [7]. However, the matrix is defined over a ring of indeterminates, so randomness is normally used in order to test if the determinant is nonzero. One approach is to replace each indeterminate by a random value from a large field. This leads, among others, to the fastest known (single-processor) running times for dense graphs [8], [9].

A second approach, adopted by Mulmuley, Vazirani and Vazirani [4] for the search version, is to replace the indeterminates by randomly chosen powers of two. Namely, for each edge (u, v), a random weight $w(u, v) \in \{1, 2, ..., 2|E|\}$ is selected, and we substitute $X_{(u,v)} := 2^{w(u,v)}$. Now, let us make the crucial assumption that one perfect matching M is **isolated**, in the sense that it is the *unique minimum-weight perfect matching* (minimizing $w(M)$). Then $\det T(G)$ remains nonzero after the substitution: one can show that M contributes a term $\pm 2^{2w(M)}$ to $\det T(G)$, whereas all other terms

DOI 10.1109/FOCS.2017.70

are multiples of $2^{2w(M)+1}$ and thus they cannot cancel $2^{2w(M)}$ out. The determinant can still be computed in NC as all entries $2^{w(u,v)}$ of the matrix are of polynomial bit-length, and so we have a parallel algorithm for the decision version. An algorithm for the search version also follows: for every edge in parallel, test whether removing it causes this least-significant digit $2^{2w(M)}$ in the determinant to disappear; output those edges for which it does.

The fundamental claim in [4] is that assigning random weights to edges does indeed isolate one matching with high probability. This is known as the Isolation Lemma and turns out to be true in the much more general setting of arbitrary set families:

Lemma I.1 (Isolation Lemma). *Let $\mathcal{M} \subseteq 2^E$ be any nonempty family of subsets of a universe $E = \{1, 2, ..., |E|\}$. Suppose we define a weight function $w : E \to \{1, 2, ..., 2|E|\}$ by selecting each $w(e)$ for $e \in E$ independently and uniformly at random. Then with probability at least $1/2$, there is a* unique *set $M \in \mathcal{M}$ which minimizes the weight $w(M) = \sum_{e\in M} w(e)$.*

We call such a weight function w *isolating*. We take $\mathcal{M}$ in Lemma I.1 to be the set of all perfect matchings.

Since Lemma I.1 is the only randomized ingredient of the RNC algorithm, a natural approach to showing that the perfect matching problem is in NC is the derandomization of the Isolation Lemma. That is, we would like a set of polynomially many weight functions (with polynomially bounded values) which would be guaranteed to contain an isolating one. To get an NC algorithm, we should be able to generate this set efficiently in parallel; then we can try all weight functions simultaneously.

However, derandomizing the Isolation Lemma turns out to be a challenging open question. It has been done for certain classes of graphs: strongly chordal [10], planar bipartite [11], [12], or graphs with a small number of perfect matchings [13], [14]. More generally, there has been much interest in obtaining NC algorithms for the perfect matching problem on restricted graph classes (not necessarily using the Isolation Lemma), e.g.: regular bipartite [15], P_4-tidy [16], dense [17], convex bipartite [18], claw-free [19], incomparability graphs [20]. The general set-family setting of the Isolation Lemma is also related to circuit lower bounds and polynomial identity testing [21].

Recently, in a major development, Fenner, Gurjar and Thierauf [22] have almost derandomized the Isolation Lemma for bipartite graphs. Namely, they define a family of weight functions which can be computed obliviously (only using the number of vertices n) and prove that for any bipartite graph, one of these functions is isolating. Because their family has quasi-polynomial size and the weights are quasi-polynomially large, this has placed the perfect bipartite matching problem in the class $\mathsf{quasi\text{-}NC}$.

Nevertheless, the general-graph setting of the derandomization question (either using the Isolation Lemma or not) remained open. Even in the planar case, with NC algorithms for bipartite planar and small-genus graphs having been known for a long time [23], [24], we knew no $\mathsf{quasi\text{-}NC}$ algorithm for non-bipartite graphs. In general, the best known upper bound on the size of uniform circuits with polylogarithmic depth was exponential.

We are able to nearly bridge this gap in understanding. The main result of our paper is the following:

Theorem I.2. *For any number n, we can in $\mathsf{quasi\text{-}NC}$ construct $n^{O(\log^2 n)}$ weight functions on $\{1, 2, ..., \binom{n}{2}\}$ with weights bounded by $n^{O(\log^2 n)}$ such that for any graph on n vertices, one of these weight functions isolates a perfect matching (if one exists).*

The results of [4] and Theorem I.2 together imply that the perfect matching problem (both the decision and the search variant) in general graphs is in $\mathsf{quasi\text{-}NC}$. See Section II-A for more details on this. We remark that the implied algorithm is very simple. The complexity lies in the analysis, i.e., proving that one of the weight functions is isolating (see Theorem IV.11).

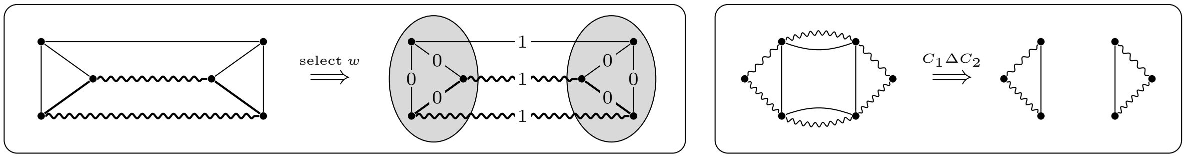
In what follows, we first give an overview of the framework in [22] for bipartite graphs. We then explain how we extend the framework to general graphs. Due to the more complex structure of perfect matchings in general graphs, we need several new ideas. In particular, we exploit structural properties of the perfect matching polytope.

A. Isolation in bipartite graphs

In this section we shortly discuss the elegant framework introduced by Fenner, Gurjar and Thierauf [22], which we extend to obtain our result.

If a weight function w is *not* isolating, then there exist two minimum-weight perfect matchings, and their symmetric difference consists of alternating cycles. In each such cycle, the total weight of edges from the first matching must be equal to the total weight of edges from the second matching (as otherwise we could obtain another matching of lower weight). The difference between these two total weights is called the *circulation* of the cycle. By the above, if all cycles have nonzero circulation, then w is isolating. It is known how to obtain weight functions which satisfy a polynomial number of such non-equalities (see Lemma III.4). However, a graph may have an exponential number of cycles.

A key idea of [22] is to build the weight function in $\log n$ rounds. In the first round, we find a weight function with the property that each cycle *of length 4* has nonzero circulation. This is possible since there are at most n^4



Main difficulty Minor difficulty

Figure 2: An illustration of the difficulties of derandomizing the Isolation Lemma for general graphs as compared to bipartite graphs.
On the left: in trying to remove the bold cycle, we select a weight function w such that the circulation of the cycle is $1 - 0 + 1 - 0 \neq 0$. By minimizing over w we obtain a new, smaller subface – the convex hull of perfect matchings of weight 1 – but every edge of the cycle is still present in one of these matchings. The cycle has only been eliminated in the following sense: it can no longer be obtained in the symmetric difference of two matchings in the new face (since none of them select both swirly edges). The vertex sets drawn in gray represent the new tight odd-set constraints that describe the new face (indeed: for a matching to have weight 1, it must take only one edge from the boundary of a gray set). We will say that the cycle does not *respect* the gray vertex sets (see Section III).
On the right: two even cycles whose symmetric difference contains no even cycle.

such cycles. We apply this function and from now on consider only those edges which belong to a minimum-weight perfect matching. Crucially, it turns out that in the subgraph obtained this way, all cycles of length 4 have disappeared – this follows from the simple structure of the bipartite perfect matching polytope (a face is simply the bipartite matching polytope of a subgraph) and fails to hold for general graphs. In the second round, we start from this subgraph and apply another weight function which ensures that all even cycles of length up to 8 have nonzero circulation (one proves that there are again $\leq n^4$ many since the graph contains no 4-cycles). Again, these cycles disappear from the next subgraph, and so on. After $\log n$ rounds, the current subgraph has no cycles, i.e., it is a perfect matching. The final weight function is obtained by combining the $\log n$ polynomial-sized weight functions. To get a parallel algorithm, we need to simultaneously try each such possible combination, of which there are quasi-polynomially many.

This result has later been generalized by Gurjar and Thierauf [25] to the linear matroid intersection problem – a natural extension of bipartite matching. From the work of Narayanan, Saran and Vazirani [26], who gave an RNC algorithm for that problem (also based on computing a determinant), it again follows that derandomizing the Isolation Lemma implies a quasi-NC algorithm.

B. Challenges of non-bipartite graphs

We find it useful to look at the method explained in the previous section from a polyhedral perspective (also used by [25]). We begin from the set of all perfect matchings, of which we take the convex hull: the perfect matching polytope. After applying the first weight function, we want to consider only those perfect matchings which minimize the weight; this is exactly the definition of a face of the polytope. In the bipartite case, any face was characterized by just taking a subset of edges (i.e., making certain constraints $x_e \geq 0$ tight), so we could simply think about recursing on a smaller subgraph. This was used to show that any cycle whose circulation has been made nonzero will not retain all of its edges in the next subgraph. The progress we made in the bipartite case could be measured by the girth (the minimum length of a cycle) of the current subgraph, which doubled as we moved from face to subface. Unfortunately, in the non-bipartite case, the description of the perfect matching polytope is more involved (see Section II-B). Namely, moving to a new subface may also cause new tight *odd-set constraints* to appear. These, also referred to as odd cut constraints, require that, for an odd set $S \subseteq V$ of vertices, exactly one edge of a matching should cross the cut defined by S. This complicates our task, as depicted in the left part of Figure 2 (the same example was given by [22] to demonstrate the difficulty of the general-graph case). Now a face is described by not only a subset of edges, but also a family of tight odd-set constraints. Thus we can no longer guarantee that any cycle whose circulation has been made nonzero will disappear from the support of the new face, i.e., the set of edges that appear in at least one perfect matching in this face. Our idea of what it means to remove a cycle thus needs to be refined (see Section III), as well as the measure of progress we use to prove that a single matching is isolated after $\log n$ rounds (see Section IV). We need several new ideas, which we outline in Section I-C.

Another difficulty, of a more technical nature, concerns the counting argument used to prove that a graph with no cycles of length at most λ contains only polynomially many cycles of length at most 2λ. In the bipartite case, the symmetric difference of two even cycles contains a simple cycle, which is also even. In addition, one can show that if the two cycles share many vertices, then the symmetric difference must contain one such even cycle that is short (of length at most λ) and thus should not exist. This enables a simple checkpointing argument to bound the number of cycles of length at most 2λ, assuming that no cycle of length at most λ exists. Now, in the general case we are still only interested in removing *even* cycles, but the symmetric difference of two even cycles may not contain an even simple cycle (see the right part of Figure 2). This forces us to remove not only even simple cycles, but all even walks, which may contain repeated edges (we call these *alternating circuits* – see Definition III.1), and to rework the counting scheme, obtaining a bound of n^{17} rather than n^4. Moreover, instead of simple graphs, we work on node-weighted multigraphs, which arise by contracting certain tight odd-sets.

C. Our approach

This section is a high-level, idealized explanation of how to deal with the main difficulty (see the left part of Figure 2); we ignore the more technical one in this description.

Removing cycles which do not cross a tight odd-set: As discussed in Section I-B, when moving from face to subface we cannot guarantee that, for each even cycle whose circulation we make nonzero, one of its edges will be absent from the support of the new face. However, this will at least be true for cycles that do not cross any odd-set tight for the new face. This is because if there are no tight odd-set constraints, then our faces behave as in the bipartite case. So, intuitively, if we only consider those cycles which do not cross any tight set, then we can remove them using the same arguments as in that case. This implies, by the same argument as in Section I-A, that if we apply $\log n$ weight functions in succession, then the resulting face will not contain in its support any even cycle that crosses no tight odd-set. This is less than we need, but a good first step. If, at this point, there were no tight sets, then we would be done, as we would have removed all cycles. However, in general there will still be cycles crossing tight sets, which make our task more difficult.

Decomposition into two subinstances: To deal with the tight odd-sets, we will make use of two crucial properties. The first property is easy to see: once we fix the single edge e in the matching which crosses a tight set S, the instance breaks up into two *independent* subinstances. That is, every perfect matching which contains e is the union of: the edge e, a perfect matching on the vertex set S (ignoring the S-endpoint of e), and a perfect matching on the vertex set $V \setminus S$ (ignoring the other endpoint of e).

This will allow us to employ a divide-and-conquer strategy: to isolate a matching in the entire graph, we will take care of both subinstances and of the cut separating them. We formulate the task of dealing with such a subinstance (a subgraph induced on an odd-cardinality vertex set) as follows: we want that, once the (only) edge of a matching which lies on the boundary of the tight odd-set is fixed, the entire matching inside the set is uniquely determined. We will then call this set *contractible* (see Definition IV.1). This can be seen as a generalization of our isolation objective to subgraphs with an odd number of vertices. If we can get that for the tight set and for its complement, then each edge from the cut separating them induces a unique perfect matching in the graph. Therefore there are at most n^2 perfect matchings left in the current face. Now, in order to isolate the entire graph, we only need a weight function w which assigns different weights to all these matchings. This demand can be written as a system of n^4 linear non-equalities on w, and we can generate a weight function w satisfying all of them (see Lemma III.4).

While it is not clear how to continue this scheme beyond the first level or why we could hope to have a low depth of recursion, we will soon explain how we utilize this basic strategy.

Laminarity: The second crucial property that we utilize is that the family of odd-set constraints tight for a face exhibits good structural properties. Namely, it is known that a *laminar* family of odd sets is enough to describe any face (see Section II-B). Recall that a family of sets is laminar if any two sets in the family are either disjoint or one is a subset of the other (see Figure 4 for an example). This enables a scheme where we use this family to make progress in a bottom-up fashion. This is still challenging as the family does not stay fixed as we move from face to face. The good news is that it can only increase: whenever a new tight odd-set constraint appears which is not spanned by the previous ones, we can always add an odd-set to our laminar family.

Chain case: To get started, let us first discuss the special case where the laminar family of tight constraints is a chain, i.e., an increasing sequence of odd-sets $S_1 \subsetneq S_2 \subsetneq ... \subsetneq S_\ell$. For this introduction, assume $\ell = 8$ as depicted in Figure 3. Denote by $U_1, ..., U_8$ the *layers* of this chain, i.e., $U_1 = S_1$ and $U_p = S_p \setminus S_{p-1}$ for $p = 2, 3, ..., 8$. Suppose this chain describes the face that was obtained by applying the $\log n$ weight functions as above that remove all even cycles that do not cross a tight set. Then there is no cycle that lies inside a single layer U_p.

Notice that every layer U_p is of even size and it touches

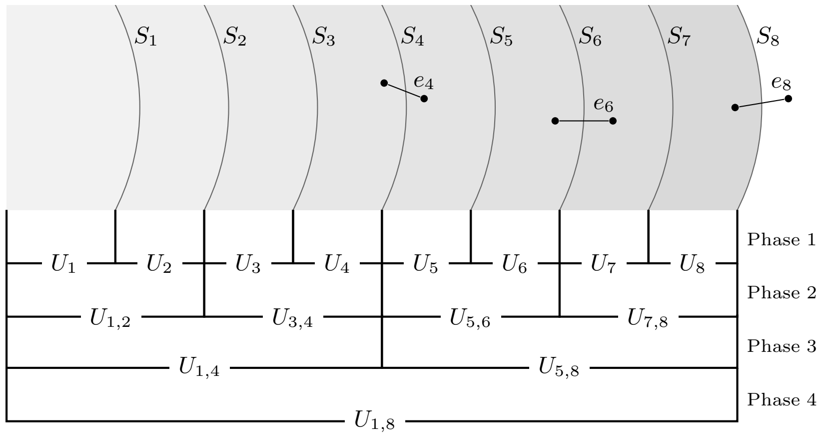


Figure 3: Example of a chain consisting of 8 tight sets, and our divide-and-conquer argument.

two boundaries of tight odd-sets: S_{p-1} and S_p (that is, $\delta(U_p) \subseteq \delta(S_{p-1}) \cup \delta(S_p)$). Any perfect matching in the current face will have one edge from $\delta(S_{p-1})$ and one edge from $\delta(S_p)$ (possibly the same edge), therefore U_p will have two (or zero) boundary edges in the matching. An exception is U_1, which is odd, only touches S_1 and will have one boundary edge in the matching. This motivates us to generalize our isolation objective to layers as follows: we say that a layer U_p is *contractible* if choosing an edge from $\delta(S_{p-1})$ and an edge from $\delta(S_p)$ induces a unique matching inside U_p. This definition naturally extends to layers of the form $S_r \setminus S_{p-1} = U_p \cup U_{p+1} \cup ... \cup U_r$, which we will denote by $U_{p,r}$.

Recall that we have ensured that there is no cycle that lies inside a single layer $U_p = U_{p,p}$. It follows that these layers are contractible. This is because two different matchings (but with the same boundary edges) in the current face would induce an alternating cycle in their symmetric difference.

Let us say that this was the first phase of our approach (see Figure 3). In the second phase, we want to ensure contractibility for double layers: $U_{1,2}$, $U_{3,4}$, $U_{5,6}$ and $U_{7,8}$. In general, we double our progress in each phase: in the third one we deal with the quadruple layers $U_{1,4}$ and $U_{5,8}$, and in the fourth phase we deal with the octuple layer $U_{1,8}$.

Let us now describe a single phase. Take e.g. the layer $U_{5,8}$ and two boundary edges $e_4 \in \delta(S_4)$ and $e_8 \in \delta(S_8)$ (see Figure 3); we want to have only a unique matching in $U_{5,8}$ including these edges. Now we realize our divide-and-conquer approach. Note that the layers $U_{5,6}$ and $U_{7,8}$ have already been dealt with (made contractible) in the previous phase. Therefore, for each choice of boundary edge $e_6 \in \delta(S_6)$ for the matching, there is a unique matching inside both of these layers. Just like previously, this implies that there are only n^2 matchings using e_4 and e_8 in the layer $U_{5,8}$, and we can select a weight function that isolates one of them.We actually select only one function per phase, which works simultaneously for all layers $U_{p,r}$ in this phase (here: $U_{1,4}$ and $U_{5,8}$) and all pairs of boundary edges e_{p-1} and e_r.

By generalizing this strategy (from $\ell = 8$ to arbitrary ℓ) in the natural way, we can deal with any chain in $\log \ell \leq \log n$ phases, even if it consists of $\Omega(n)$ tight sets.

General case: Of course, there is no reason to expect that the laminar family of tight cuts we obtain after applying the initial $\log n$ weight functions will be a chain. It also does not seem easy to directly generalize our inductive scheme from a chain to an arbitrary family. Therefore we put forth a different progress measure, which allows us to make headway even in the absence of such a favorable odd-set structure.

Since a laminar family can be represented as a tree, we might think about a bottom-up strategy based on it; however, we cannot deal with its nodes level-by-level, since it may have height $\Omega(n)$ and we can only afford $\mathrm{poly}(\log n)$ many phases. Instead, we will first deal with all tight odd-sets of size up to 4, then up to 8, then up to 16 and so on, by making them contractible. At the same time, we also remove all even cycles of length up to 4, then up to 8 and so on. These two components of our progress measure, which we call λ-goodness, are mutually beneficial, as we will see below.

Making odd-sets contractible enables us not only to achieve progress, but also to simplify our setting. A contractible tight set can be, for our purposes, thought of as a single vertex – much like a blossom in Edmonds' algorithm. This is because such a set has exactly one boundary edge in a perfect matching (as does a vertex), and choosing that edge determines the matching in the

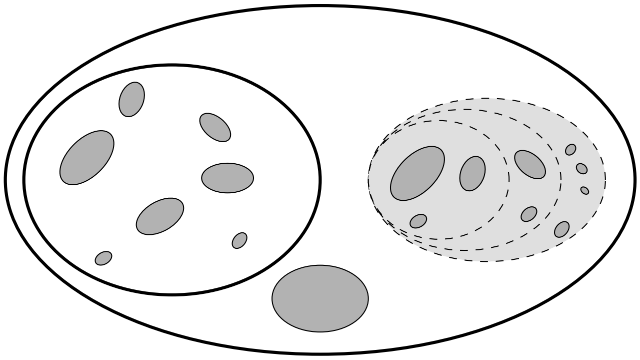


Figure 4: Example of a general laminar family.
Dark-gray sets are of size at most λ and thus contractible.
Dashed sets are of size more than λ but at most 2λ; they must form chains (due to the cardinality constraints). We make them contractible in the first step. Then we contract them (so now all light-gray and dark-gray sets are contracted).
Thick sets are of size more than 2λ. For the second step, we erase the edges on their boundaries. Then we remove cycles of length up to 2λ from the resulting instance (the *contraction*), which has no tight odd-sets (and no cycles of length up to λ).

interior. As the name suggests, we will contract such sets.

Suppose that our current face is already λ-good. Roughly, this means that we have made odd-sets of size up to λ (which we will call small) contractible and removed cycles of length up to λ. Now we want to obtain a face which is 2λ-good.

The first step is to make odd-sets of size up to 2λ contractible. Let us zoom in on one such odd-set – a maximal set of size at most 2λ (see the largest dashed set in Figure 4). Once we have contracted all the small sets into single vertices, all interesting sets are now of size more than λ but at most 2λ, and any laminar family consisting of such sets must be a chain, since a set of such size cannot have two disjoint subsets of such size (see Figure 4). But this is the chain case that we have already solved!

Having made odd-sets of size up to 2λ contractible, we can contract them. The second step is now to remove cycles of length up to 2λ. However, here we do not need to care about those cycles which cross an odd-set S of size larger than 2λ – the reason being, intuitively, that in our technical arguments we define the length of a cycle based on the sizes of sets that it crosses, and thus such a cycle actually becomes longer than 2λ. In other words, we can think about removing cycles of length up to 2λ from a version of the input graph where all small odd-sets have been contracted and all larger ones have had their boundaries erased (see Figure 4). We call this version the *contraction* (see Definition IV.5). Our λ-goodness progress measure (see Definition IV.7) is actually defined in terms of cycles in the contraction.

Now the second step is easy: we just need to remove all cycles of length up to 2λ from the contraction, which has no tight odd-sets and no cycles of length up to λ – a simple scenario, already known from the bipartite case. Applying one weight function is enough to do this.

Finally, what does it mean for us to remove a cycle? When we make a cycle's circulation nonzero, it is then eliminated from the new face in the following sense: either one of its edges disappears from the support of the face (recall that this is what always happened in the bipartite case), or a new tight odd-set appears, with the following property: the cycle crosses the set with fewer (or more) even-indexed edges than odd-indexed edges (see the example in Figure 2). In short, we say that the cycle does not *respect* the new face (see Section III). This notion of removal makes sense when viewed in tandem with the contraction, because once a cycle crosses a set in the laminar family, there are two possibilities in each phase: either this set is large – then its boundary is not present in the contraction, which cancels the cycle, or it is small – then it is contracted and the cycle also disappears (for somewhat more technical reasons).

To reiterate, our strategy is to simultaneously remove cycles up to a given length and make odd-sets up to a given size contractible. We can do this in $\log n$ phases. In each such phase we need to apply a sequence of $\log n$ weight functions in order to deal with a chain of tight odd-sets (as outlined above). In all, we are able to isolate a perfect matching in the entire graph using a sequence $O(\log^2 n)$ weight functions with polynomially bounded weights.

D. Future work

The most immediate open problem left by our work is to get down from quasi-NC to NC for the perfect matching problem. Even for the bipartite case, this will require new insights or methods, as it is not clear how we could e.g. reduce the number of weight functions from $\log n$ to only a constant.

Proving that the search version of the perfect matching problem in planar graphs is in NC is also open. While the quasi-NC result of [22] gives rise to a new NC algorithm for bipartite planar graphs, which proceeds by verifying at each step whether the chosen weight function has removed the wanted cycles (it computes the girth of the support of the current face in NC), our λ-goodness progress measure seems to be difficult to verify in NC.

A related problem which has resisted derandomization so far is exact matching [27]. Here we are given a graph, whose some edges are colored red, and an integer k; the question is to find a perfect matching containing exactly k red edges. The problem is in RNC [4], but not known to even be in P.

Finally, our polyhedral approach motivates the question of what other zero-one polytopes admit such a derandomization of the Isolation Lemma. One class that comes to mind are totally unimodular polyhedra.

E. Outline

The rest of the paper is organized as follows. In Section II we introduce notation and define basic notions related to the perfect matching polytope and to the weight functions that we use. In Section III we define alternating circuits (our generalization of alternating cycles), discuss what it means for such a circuit to respect a face, and develop our tools for circuit removal. In Section IV we introduce our measure of progress (λ-goodness), contractible sets and the contraction multigraph. We also state Theorem IV.11, which implies our main result. We defer to the full version of the paper the proof of our key technical theorem: that applying $\log_2 n + 1$ weight functions allows us to make progress from λ-good to 2λ-good.

II. Preliminaries

Throughout the paper we consider a fixed graph $G = (V, E)$ with n vertices. We remark that the isolating weight functions whose existence we prove can be generated without knowledge of the graph. For notational convenience, we assume that $\log_2 n$ evaluates to an integer; otherwise simply replace $\log_2 n$ by $\lceil \log_2 n \rceil$. We also assume that n is sufficiently large.

We use the following notation. For a subset $S \subseteq V$ of the vertices, let $\delta(S) = \{e \in E : |e \cap S| = 1\}$ denote the edges crossing the cut $(S, V \setminus S)$ and $E(S) = \{e \in E : |e \cap S| = 2\}$ denote the edges inside S. We shorten $\delta(\{v\})$ to $\delta(v)$ for $v \in V$. For a vector $(x_e)_{e\in E} \in \mathbb{R}^{|E|}$, we define $x(\delta(S)) = \sum_{e\in\delta(S)} x_e$, as well as $\mathrm{supp}(x) = \{e \in E : x_e > 0\}$. For a subset $X \subseteq E$ we define $\mathbb{1}_X$ to be the vector with 1 on coordinates in X and 0 elsewhere. We again shorten $\mathbb{1}_{\{e\}}$ to $\mathbb{1}_e$ for $e \in E$. Sometimes we identify matchings M with their indicator vectors $\mathbb{1}_M$.

A matching is a set of edges $M \subseteq E$ such that no two edges in M share an endpoint. A matching M is perfect if $|M| = \frac{n}{2}$.

A. Parallel complexity

The complexity class $\mathsf{quasi\text{-}NC}$ is defined as $\mathsf{quasi\text{-}NC} = \bigcup_{k\geq 0} \mathsf{quasi\text{-}NC}^k$, where $\mathsf{quasi\text{-}NC}^k$ is the class of problems having uniform circuits of quasi-polynomial size $2^{\log^{O(1)} n}$ and polylogarithmic depth $O(\log^k n)$ [28]. Here by "uniform" we mean that the circuit can be generated in polylogarithmic space.

By the results of [4], Theorem I.2 implies that the perfect matching problem (both the decision and the search variant) in general graphs is in $\mathsf{quasi\text{-}NC}$. The same can be said about maximum cardinality matching, as well as minimum-cost perfect matching for small costs (given in unary); see Section 5 of [4].

Some care is required to obtain our postulated running time, i.e., that the perfect matching problem has uniform circuits of size $n^{O(\log^2 n)}$ and depth $O(\log^3 n)$. We could get a $\mathsf{quasi\text{-}NC}^4$ algorithm by applying the results of [29, Section 6.1] to compute the determinant(s). To shave off one $\log n$ factor, we use the following Chinese remaindering method, pointed out to us by Rohit Gurjar (it will also appear in the full version of [22]). We first compute determinants modulo small primes; since the determinant has $2^{O(\log^3 n)}$ bits, we need as many primes (each of $O(\log^3 n)$ bits). For one prime this can be done in NC^2 [7]. Then we reconstruct the true value from the remainders. Doing this for an n-bits result would be in NC^1 [30], and thus for a result with $2^{O(\log^3 n)}$ bits it is in $\mathsf{quasi\text{-}NC}^3$.

B. Perfect matching polytope

Edmonds [31] showed that the following set of equalities and inequalities on the variables $(x_e)_{e\in E}$ determines the perfect matching polytope (i.e., the convex hull of indicator vectors of all perfect matchings):

$$\begin{aligned} x(\delta(v)) &= 1 && \text{for } v \in V, \\ x(\delta(S)) &\geq 1 && \text{for } S \subseteq V \text{ with } |S| \text{ odd}, \\ x_e &\geq 0 && \text{for } e \in E. \end{aligned}$$

Note that the constraints imply that $x_e \leq 1$ for any $e \in E$. We refer to the perfect matching polytope of the graph $G = (V, E)$ by $\mathrm{PM}(V, E)$ or simply by PM. Our approach exploits the special structure of faces of the perfect matching polytope. Recall that a face of a polytope is obtained by setting a subset of the inequalities to equalities. We follow the definition of a face from the book of Schrijver [32] – in particular, every face is nonempty.

Throughout the paper, we will only consider the perfect matching polytope and so the term "face" will always refer to a face of PM. We sometimes abuse notation and say that a perfect matching M is in a face F if its indicator vector is in F. When talking about faces, we also use the following notation:

Definition II.1. *For a face F we define*

$$E(F) = \{e \in E : (\exists x \in F)\ x_e > 0\}$$

and

$$\mathcal{S}(F) = \{S \subseteq V : |S| \text{ odd and } (\forall x \in F)\ x(\delta(S)) = 1\}\,.$$

In other words, $E(F)$ contains the edges that appear in a perfect matching in F and $\mathcal{S}(F)$ contains the tight cut constraints of F.

Notice that if a set is tight for a face, then it is also tight for any of its subfaces.

Standard uncrossing techniques imply that faces can be defined using laminar families of tight constraints. This is proved using Lemma II.2 below, which is also useful in our approach.

Two subsets $S, T \subseteq V$ of vertices are said to be crossing if they intersect and none is contained in the other, i.e., $S \cap T, S \setminus T, T \setminus S \neq \emptyset$. A family $\mathcal{L}$ of subsets of vertices is *laminar* if no two sets $S, T \in \mathcal{L}$ are crossing. Furthermore, we say that $\mathcal{L}$ is a *maximal laminar subset* of a family $\mathcal{S}$ if no set in $\mathcal{S} \setminus \mathcal{L}$ can be added to $\mathcal{L}$ while maintaining laminarity.

Note that any single-vertex set is tight for any face, and therefore a maximal laminar family contains all these sets. The laminar families in our arguments will always contain all singletons.

The following lemma is known.

Lemma II.2. *Consider a face F. For any maximal laminar subset $\mathcal{L}$ of $\mathcal{S}(F)$ we have*

$$\mathrm{span}(\mathcal{L}) = \mathrm{span}(\mathcal{S}(F)),$$

where for a subset $\mathcal{T} \subseteq \mathcal{S}(F)$, $\mathrm{span}(\mathcal{T})$ denotes the linear subspace of $\mathbb{R}^E$ spanned by the boundaries of sets in $\mathcal{T}$, i.e., $\mathrm{span}(\mathcal{T}) = \mathrm{span}\{\mathbb{1}_{\delta(S)} : S \in \mathcal{T}\}$.

Intuitively, Lemma II.2 implies that a maximal laminar family $\mathcal{L}$ of $\mathcal{S}(F)$ is enough to describe a face F (together with the edge set $E(F)$). Furthermore, given a subface $F' \subseteq F$, we can extend $\mathcal{L}$ to a larger laminar family $\mathcal{L}' \supseteq \mathcal{L}$ which describes F'.

As the perfect matching polytope PM is defined as the convex hull of the indicator vectors of all perfect matchings, it is an integral polytope. In particular, it follows that every face of PM is also integral.

C. Weight functions

For our derandomization of the Isolation Lemma we will use families of weight functions which are possible to generate obliviously, i.e., by only using the number of vertices in G. We define them below.

Definition II.3. *Given $t \geq 7$, we define the family of weight functions $\mathcal{W}(t)$ as follows. Number the edge set $E = \{e_1, ..., e_{|E|}\}$ arbitrarily. Let $w_k : E \to \mathbb{Z}$ be given by $w_k(e_j) = (4n^2+1)^j \bmod k$ for $j = 1, ..., |E|$ and $k = 2, ..., t$. We define $\mathcal{W}(t) = \{w_k : k = 2, ..., t\}$.*

For brevity, we write $\mathcal{W} := \mathcal{W}(n^{20})$.

In our argument we will obtain a decreasing sequence of faces. Each face arises from the previous by minimizing over a linear objective (given by a weight function).

Definition II.4. *Let F be a face and w a weight function. The subface of F minimizing w will be called $F[w]$:*

$$F[w] := \mathrm{argmin}\{\langle w, x\rangle : x \in F\}.$$

Instead of minimizing over one weight function and then over another, we can *concatenate* them in such a way that minimizing over the concatenation yields the same subface. In particular, we will argue that one just needs to try all possible concatenations of $O(\log^2 n)$ weight functions from $\mathcal{W}$ in order to find one which isolates a unique perfect matching in G (i.e., it produces a single extreme point as the minimizing subface).

Definition II.5. *For two weight functions w and w', where $w : E \to \mathbb{Z}$ and $w' \in \mathcal{W}$, we define their* concatenation *$w \circ w' := n^{21}w + w'$, i.e.,*

$$(w \circ w')(e) := n^{21} \cdot w(e) + w'(e).$$

We also define $\mathcal{W}^k$ to be the set of all concatenations of k weight functions from $\mathcal{W}$, i.e.,

$$\mathcal{W}^k := \{w_1 \circ w_2 \circ ... \circ w_k : w_1, w_2, ..., w_k \in \mathcal{W}\}.$$

Fact II.6. *We have $F[w][w'] = F[w \circ w']$.*

The proof can be found in the full version of the paper.

III. Alternating circuits and respecting a face

In this section we introduce two notions which are vital for our approach. Before giving the formal definitions, we give an informal motivation.

Our argument is centered around *removing* even cycles. As discussed in Section I-B and Figure 2, the meaning of this term in the non-bipartite case needs to be more subtle than just "removing an edge of the cycle".

In order to deal with a cycle, we find a weight function w which assigns it a nonzero circulation. Formally, given an even cycle C with edges numbered in order, define a vector $(\pm\mathbb{1})_C \in \{-1, 0, 1\}^E$ as having 1 on even-numbered edges of C, -1 on odd-numbered edges of C, and 0 elsewhere. Then, nonzero circulation means that $\langle (\pm\mathbb{1})_C, w\rangle \neq 0$. Now, in the bipartite case, if such a cycle survived in the new face $F[w]$, that is, $C \subseteq E(F[w])$, then the vector $(\pm\mathbb{1})_C$ could be used to obtain a point in the face F with lower w-weight than the points in $F[w]$, a contradiction. This argument is possible because of the simple structure of the bipartite perfect matching polytope.

In the non-bipartite case, it is not enough that $C \subseteq E(F[w])$ in order to obtain such a point (and a contradiction). It is also required that, if the cycle C enters a tight odd-set S on an even-numbered edge, it exits it on an odd-numbered edge (and vice versa). This makes intuitive sense: if C were obtained from the symmetric difference of two perfect matchings which both have exactly one edge crossing S, then C would have this

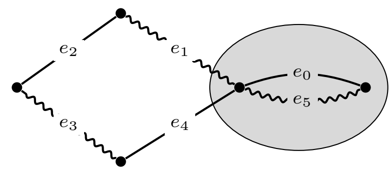


Figure 5: An example of an alternating circuit C of length 6 with indicator vector $(\pm\mathbb{1})_C = \sum_{i=0}^{5}(-1)^i \mathbb{1}_{e_i} = -\mathbb{1}_{e_1} + \mathbb{1}_{e_2} - \mathbb{1}_{e_3} + \mathbb{1}_{e_4}$ (since $\mathbb{1}_{e_0}$ and $\mathbb{1}_{e_5}$ cancel each other). Also note that $\langle (\pm\mathbb{1})_C, \mathbb{1}_{\delta(S)} \rangle = 0$ for the tight set S depicted in gray.

property. Formally, we require that $\langle (\pm\mathbb{1})_C, \mathbb{1}_{\delta(S)} \rangle = 0$ for each $S \in \mathcal{S}(F[w])$. If C satisfies these two conditions, i.e., that $C \subseteq E(F[w])$ and that $\langle (\pm\mathbb{1})_C, \mathbb{1}_{\delta(S)} \rangle = 0$ for every $S \in \mathcal{S}(F[w])$, then we say that C *respects* the face $F[w]$. The notion of respecting a face exactly formalizes what is required to obtain a contradictory point as above (see the proof of Lemma III.3).

In other words, if we assign a nonzero circulation to a cycle, then it will not respect the new face, and this is what is now meant by removing a cycle.

To deal with the second, more technical difficulty discussed in Section I-B, we need to remove not only simple cycles of even length, but also walks with repeated edges. However, we would run into problems if we allowed all such walks (up to a given length). Consider for example a walk C of length 2; such a walk traverses an edge back and forth. It is impossible to assign a nonzero circulation to C, because its vector $(\pm\mathbb{1})_C$ is zero. We overcome this technicality by defining alternating circuits to be those even walks whose vector $(\pm\mathbb{1})_C$ is nonzero (see Figure 5 for an example). For generality, we also formulate the definition of respect in terms of the vector $(\pm\mathbb{1})_C$.

Definition III.1. *Let $C = (e_0, ..., e_{k-1})$ be a nonempty cyclic walk of even length k.*

- *We define the* alternating indicator vector $(\pm\mathbb{1})_C$ of C *to be $(\pm\mathbb{1})_C = \sum_{i=0}^{k-1}(-1)^i \mathbb{1}_{e_i}$, where $\mathbb{1}_e \in \mathbb{R}^E$ is the indicator vector having 1 on position e and 0 elsewhere.*
- *We say that C is an* alternating circuit *if its alternating indicator vector is nonzero. We also refer to C as an* alternating (simple) cycle *if it is an alternating circuit that visits every vertex at most once.*
- *When talking about a graph with node-weights, the node-weight of an alternating circuit is the sum of all node-weights of visited vertices (with multiplicities if visited multiple times).*

Definition III.2. *We say that a vector $y \in \mathbb{Z}^E$* respects *a face F if:*

- *$\mathrm{supp}(y) \subseteq E(F)$, and*
- *for each $S \in \mathcal{S}(F)$ we have $\langle y, \mathbb{1}_{\delta(S)} \rangle = 0$.*

Furthermore, we say that an alternating circuit C respects a face F if its alternating indicator vector $(\pm\mathbb{1})_C$ respects F.

Clearly, if $F' \subseteq F$ are faces and a vector respects F', then it also respects F.

Now we argue that we can remove an alternating circuit by assigning it a nonzero circulation. The proof of this lemma (which generalizes Lemma 3.2 of [22]) motivates Definition III.2.

Lemma III.3. *Let $y \in \mathbb{Z}^E$ be a vector and F a face. If $w : E \to \mathbb{R}$ is such that $\langle y, w \rangle \neq 0$, then y does not respect the face $F' = F[w]$.*

Proof: Suppose towards a contradiction that y respects F'. Assume that $\langle w, y \rangle < 0$ (otherwise use $-y$ in place of y). We pick $x \in F'$ to be the average of all extreme points of F', so that the constraints of PM which are tight for x are exactly those which are tight for F'. Select $\varepsilon > 0$ very small. Then $\langle x + \varepsilon y, w \rangle < \langle x, w \rangle$, which will contradict the definition of $F' = \mathrm{argmin}\{\langle w, x \rangle : x \in F\}$ once we show that $x + \varepsilon y \in F$. We show that $x + \varepsilon y \in F' \subseteq F$ by verifying:

- If $e \in E(F')$ (i.e., e is an edge with $x_e > 0$), then $(x+\varepsilon y)_e = x_e + \varepsilon y_e \geq 0$ if ε is chosen small enough.
- If $e \in E \setminus E(F')$ (i.e., e is an edge with $x_e = 0$), then from y respecting F' we get $e \notin \mathrm{supp}(y)$ and so $(x + \varepsilon y)_e = 0$.
- If $S \notin \mathcal{S}(F')$ is an odd set not tight for F', i.e., $\langle x, \mathbb{1}_{\delta(S)} \rangle > 1$, then $\langle x + \varepsilon y, \mathbb{1}_{\delta(S)} \rangle = \langle x, \mathbb{1}_{\delta(S)} \rangle + \varepsilon \langle y, \mathbb{1}_{\delta(S)} \rangle \geq 1$ if ε is chosen small enough.
- If $S \in \mathcal{S}(F')$ is an odd set tight for F' (this includes all singleton sets), then from y respecting F' we get $\langle y, \mathbb{1}_{\delta(S)} \rangle = 0$ and thus $\langle x + \varepsilon y, \mathbb{1}_{\delta(S)} \rangle = \langle x, \mathbb{1}_{\delta(S)} \rangle = 1$.

∎

The following lemma says that we can assign nonzero circulation to many vectors at once using an oblivious choice of weight function from $\mathcal{W}$. It is a minor generalization of Lemma 2.3 of [22] and the proof remains similar.

Lemma III.4. *For any number s and for any set of s vectors $y_1, ..., y_s \in \mathbb{Z}^E \setminus \{0\}$ with the boundedness property $\|y_i\|_1 \leq 4n^2$, there exists $w \in \mathcal{W}(n^3 s)$ with $\langle y_i, w \rangle \neq 0$ for each $i = 1, ..., s$.*

We usually invoke Lemma III.4 with vectors y_i being the alternating indicator vectors of alternating circuits. Then the quantities $\langle y_i, w \rangle$ are the circulations of these circuits.

Lemmas III.3 and III.4 together imply the following:

Corollary III.5. *Let F be a face. For any finite set of vectors $\mathcal{Y} \subseteq \mathbb{Z}^E \setminus \{0\}$ with the boundedness property $\|y\|_1 \leq 4n^2$ for every $y \in \mathcal{Y}$, there exists $w \in \mathcal{W}(n^3|\mathcal{Y}|)$ such that each $y \in \mathcal{Y}$ does not respect the face $F' = F[w]$.*

IV. Contractible sets and λ-goodness

We will make progress by ensuring that larger and larger parts of the graph are "isolated" in our current face F. By "parts of the graph" we mean sets S which are tight for F. As discussed in Section I-C, for such a set S, the following isolation property is desirable: once the (only) edge of a matching which lies on the boundary of S is fixed, the entire matching inside S is uniquely determined. This motivates the following definition:

Definition IV.1. *Let F be a face and let $S \in \mathcal{S}(F)$ be a tight set for F. We say that S is* F-contractible *if for every $e \in \delta(S)$ there are no two perfect matchings in F which both contain e and are different inside S.*

Note that, in the above definition, there could be no such perfect matching for certain edges $e \in \delta(S)$ (this is the case if and only if $e \notin E(F)$). Intuitively, a contractible set can be thought of as a single vertex with respect to the structure of the current face of the perfect matching polytope. The notion of contractibility enjoys the following two natural monotonicity properties:

Fact IV.2. *Let $F' \subseteq F$ be two faces. If S is F-contractible, then it is also F'-contractible.*

Lemma IV.3. *Let F be a face and $S \subseteq T$ two sets tight for F (i.e., $S, T \in \mathcal{S}(F)$). If T is F-contractible, then so is S.*

The proof can be found in the full version of the paper.

In our proof, we will be working with faces and laminar families which are compatible in the following sense:

Definition IV.4. *Let F be a face and $\mathcal{L}$ a laminar family. If $\mathcal{L} \subseteq \mathcal{S}(F)$, i.e., all sets $S \in \mathcal{L}$ are tight for F, then we say that $(F, \mathcal{L})$ is a* face-laminar pair.

Given a face-laminar pair $(F, \mathcal{L})$, we will often work with a multigraph obtained from G by contracting all small sets, i.e., those with size being at most some parameter λ (which is a measure of our progress). This multigraph will be called the *contraction* (see Figure 6 for an example).

In the contraction, we will also remove all boundaries of larger sets (i.e., those with size larger than λ). This is done to simulate working inside each such large set independently, because the contraction then decomposes into a collection of disconnected components, one per each large set. Because, in the contraction, each set in $\mathcal{L}$ has either been contracted or has had its boundary removed, our task is reduced to dealing with instances having no laminar sets.

Moreover, we only include those edges which are still in the support of the current face F, i.e., the set $E(F)$.

Definition IV.5. *Given a face-laminar pair $(F, \mathcal{L})$ and a parameter λ (with $1 \leq \lambda \leq 2n$), we define the* $(F, \mathcal{L}, \lambda)$-contraction *of G as a node-weighted multigraph as follows:*

- *the node set is the set of maximal sets of size (cardinality) at most λ in $\mathcal{L}$,*
- *each node has a node-weight equal to the size of the corresponding set,*
- *the edge set is obtained from $E(F) \setminus \bigcup_{T \in \mathcal{L}:|T|>\lambda} \delta(T)$ by contracting each of these maximal sets. That is, an edge of G maps to an edge of the contraction if it is in $E(F)$, it is not inside any of the contracted sets and it does not cross any cut defined by a set $T \in \mathcal{L} : |T| > \lambda$.*

In the $(F, \mathcal{L}, \lambda)$-contractions arising in our arguments, we will always only contract sets $S \in \mathcal{L}$ which are F-contractible (i.e., the vertices of a contraction will always correspond to F-contractible sets). Then, a very useful property is that alternating circuits in the contraction can be lifted to alternating circuits in the entire graph G in a canonical way.

Finally, we need the following extension of Definition III.2 for vectors defined on the contraction.

Definition IV.6. *Denote the $(F, \mathcal{L}, \lambda)$-contraction of G as H, and let $z \in \mathbb{Z}^{E(H)}$ be a vector on the edges of H. We say that z respects a subface $F' \subseteq F$ if*[1]

- $\mathrm{supp}(z) \subseteq E(F')$, *and*
- *for each $S \in \mathcal{S}(F')$ which is a union of sets corresponding to vertices in $V(H)$, we have $\langle z, \mathbb{1}_{\delta(S)} \rangle = 0$.*

As before, we say that an alternating circuit C in H respects a subface F' if its alternating indicator vector $(\pm\mathbb{1})_C \in \mathbb{Z}^{E(H)}$ respects F'.

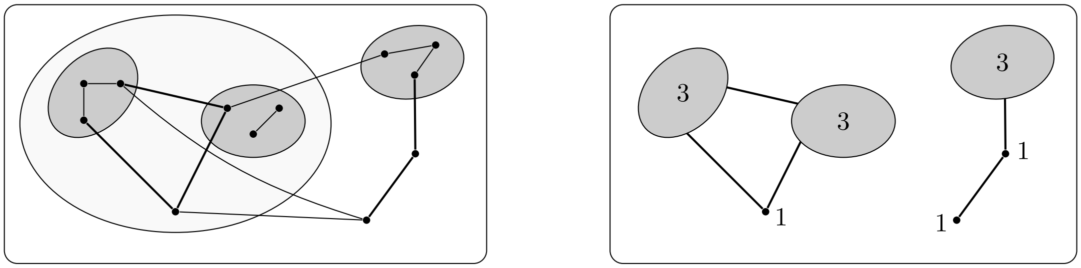
Now we are able to define our measure of progress. On one hand, we want to make larger and larger laminar sets contractible. On the other hand, there could very well be no laminar sets, so we also proceed as in the bipartite case: remove longer and longer alternating circuits.

Definition IV.7. *Let $(F, \mathcal{L})$ be a face-laminar pair and λ a parameter (with $1 \leq \lambda \leq 2n$). We say that $(F, \mathcal{L})$ is* λ-good *if $\mathcal{L}$ is a maximal laminar subset of $\mathcal{S}(F)$ and:*

(i) each $S \in \mathcal{L}$ with $|S| \leq \lambda$ is F-contractible,
(ii) in the $(F, \mathcal{L}, \lambda)$-contraction of G, there is no alternating circuit of node-weight at most λ.

We begin with $\lambda = 1$, which is trivial, and then show that by concatenating enough weight functions we can

[1]In the following conditions we abuse notation and think of z as a vector in $\mathbb{Z}^E$ obtained by identifying each edge of H with its preimage in G and letting those edges of G without a preimage in H have value 0.



(a) A graph G and a laminar family $\mathcal{L}$. We only draw the edges in $E(F)$. We also do not draw ellipses for the singleton sets in $\mathcal{L}$. The dark-gray sets are F-contractible.

(b) The $(F, \mathcal{L}, 4)$-contraction of G. Its vertices are labeled by their node-weights.

Figure 6: An example of the $(F, \mathcal{L}, \lambda)$-contraction of G.

obtain face-laminar families which are 2-good, 4-good, 8-good, and so on. We are done once we have a λ-good family with $\lambda \geq n$. The components of this proof strategy are given in the following three claims. The first step is clear:

Fact IV.8. *Let $\mathcal{L}_0$ be a maximal laminar subset of $\mathcal{S}(PM)$. Then the face-laminar pair* $(\mathrm{PM}, \mathcal{L}_0)$ *is* 1*-good.*

We then proceed iteratively in $\log_2 n$ rounds using the following theorem. Its proof, which constitutes the bulk of our argument, can be found in the full version of the paper.

Theorem IV.9. *Let $(F, \mathcal{L})$ be a λ-good face-laminar pair. Then there exists a weight function $w \in \mathcal{W}^{\log_2 n+1}$ and a laminar family $\mathcal{L}' \supseteq \mathcal{L}$ such that $(F[w], \mathcal{L}')$ is a 2λ-good face-laminar pair.*

We are done once λ exceeds n:

Lemma IV.10. *Suppose $(F, \mathcal{L})$ is λ-good for some $\lambda \geq n$. Then $|F| = 1$.*

The proof can be found in the full version of the paper.

Let us see how Fact IV.8, Theorem IV.9, and Lemma IV.10 together give our desired result:

Theorem IV.11. *There exists an isolating weight function $w \in \mathcal{W}^{(\log_2 n+1)\log_2 n}$, i.e., one with $|\mathrm{PM}[w]| = 1$.*

Proof: We iteratively construct a sequence of face-laminar pairs $(F_i, \mathcal{L}_i)$ for $i = 0, 1, ..., \log_2 n$ such that $(F_i, \mathcal{L}_i)$ is 2^i-good and $F_i = F_{i-1}[w_i]$ for some weight function $w_i \in \mathcal{W}^{\log_2 n+1}$. We begin by setting $F_0 = \mathrm{PM}$ and $\mathcal{L}_0$ to be a maximal laminar subset of $\mathcal{S}(PM)$. By Fact IV.8, $(F_0, \mathcal{L}_0)$ is 1-good. Then for $i = 1, ..., \log_2 n$ we use Theorem IV.9 to obtain the wanted weight function w_i along with a laminar family $\mathcal{L}_i \supseteq \mathcal{L}_{i-1}$. Finally, we have that $(F_{\log_2 n}, \mathcal{L}_{\log_2 n})$ is $2^{\log_2 n}$-good, so that by Lemma IV.10, $|F_{\log_2 n}| = 1$.

It remains to argue that $F_{\log_2 n} = \mathrm{PM}[w]$ for some $w \in \mathcal{W}^{(\log_2 n+1)\log_2 n}$. To do this, we proceed as in Section II-C: define the concatenation $w' \bullet w'' := n^{21(\log_2 n+1)} w' + w''$ for two weight functions w' and w'', where $w'' \in \mathcal{W}^{\log_2 n+1}$. By the same reasoning as for Fact II.6 we get that $F_{\log_2 n} = \mathrm{PM}[w_1][w_2]...[w_{\log_2 n}] = \mathrm{PM}[w_1 \bullet w_2 \bullet ... \bullet w_{\log_2 n}]$. We put $w = w_1 \bullet w_2 \bullet ... \bullet w_{\log_2 n}$. ■

Theorem IV.11 implies Theorem I.2 because we have $|\mathcal{W}^{(\log_2 n+1)\log_2 n}| = |\mathcal{W}|^{(\log_2 n+1)\log_2 n} \leq n^{20(\log_2 n+1)\log_2 n}$, the values of any $w \in \mathcal{W}^{(\log_2 n+1)\log_2 n}$ are bounded by $n^{21(\log_2 n+1)\log_2 n}$, and the functions $w \in \mathcal{W}$ can be generated obliviously using only the number of vertices n.

Acknowledgment

We thank the anonymous FOCS 2017 reviewers for their exceptionally detailed and insightful comments.

This work is supported by the ERC Starting Grant 335288-OptApprox.

References

[1] J. Edmonds, “Paths, trees, and flowers,” *Canadian Journal of Mathematics*, vol. 17, pp. 449–467, 1965.

[2] L. Lovász, “On determinants, matchings, and random algorithms,” in *FCT*, 1979, pp. 565–574.

[3] R. M. Karp, E. Upfal, and A. Wigderson, “Constructing a perfect matching is in random NC,” *Combinatorica*, vol. 6, no. 1, pp. 35–48, 1986.

[4] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani, “Matching is as easy as matrix inversion,” *Combinatorica*, vol. 7, no. 1, pp. 105–113, 1987.

[5] W. T. Tutte, “The factorization of linear graphs,” *Journal of the London Mathematical Society*, vol. 22, pp. 107–111, 1947.

[6] L. Csanky, “Fast parallel inversion algorithm,” *SIAM Journal of Computing*, vol. 5, pp. 618–623, 1976.

[7] S. J. Berkowitz, "On computing the determinant in small parallel time using a small number of processors," *Information Processing Letters*, vol. 18, no. 3, pp. 147–150, 1984.

[8] M. Mucha and P. Sankowski, "Maximum matchings via Gaussian elimination," in *45th Symposium on Foundations of Computer Science (FOCS)*, 2004, pp. 248–255.

[9] N. J. A. Harvey, "Algebraic algorithms for matching and matroid problems," *SIAM J. Comput.*, vol. 39, no. 2, pp. 679–702, 2009.

[10] E. Dahlhaus and M. Karpinski, "Matching and multidimensional matching in chordal and strongly chordal graphs," *Discrete Applied Mathematics*, vol. 84, no. 1-3, pp. 79–91, 1998.

[11] S. Datta, R. Kulkarni, and S. Roy, "Deterministically isolating a perfect matching in bipartite planar graphs," *Theory Comput. Syst.*, vol. 47, no. 3, pp. 737–757, 2010.

[12] R. Tewari and N. V. Vinodchandran, "Green's theorem and isolation in planar graphs," *Inf. Comput.*, vol. 215, pp. 1–7, 2012.

[13] D. Grigoriev and M. Karpinski, "The matching problem for bipartite graphs with polynomially bounded permanents is in NC," in *28th Annual Symposium on Foundations of Computer Science (FOCS)*, 1987, pp. 166–172.

[14] M. Agrawal, T. M. Hoang, and T. Thierauf, "The polynomially bounded perfect matching problem is in NC^2," in *STACS 2007, 24th Annual Symposium on Theoretical Aspects of Computer Science*, 2007, pp. 489–499.

[15] G. F. Lev, N. Pippenger, and L. G. Valiant, "A fast parallel algorithm for routing in permutation networks," *IEEE Transactions on Computers*, vol. C-30, no. 2, pp. 93–100, Feb 1981.

[16] I. Parfenoff, "An efficient parallel algorithm for maximum matching for some classes of graphs," *Journal of Parallel and Distributed Computing*, vol. 52, no. 1, pp. 96 – 108, 1998.

[17] E. Dahlhaus, P. Hajnal, and M. Karpinski, "On the parallel complexity of Hamiltonian cycle and matching problem on dense graphs," *Journal of Algorithms*, vol. 15, no. 3, pp. 367 – 384, 1993.

[18] E. Dekel and S. Sahni, "A parallel matching algorithm for convex bipartite graphs and applications to scheduling," *Journal of Parallel and Distributed Computing*, vol. 1, no. 2, pp. 185 – 205, 1984.

[19] M. Chrobak, J. Naor, and M. B. Novick, *Using bounded degree spanning trees in the design of efficient algorithms on claw-free graphs*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989, pp. 147–162.

[20] D. Kozen, U. V. Vazirani, and V. V. Vazirani, "NC algorithms for comparability graphs, interval gaphs, and testing for unique perfect matching," in *Proceedings of the Fifth Conference on Foundations of Software Technology and Theoretical Computer Science*. London, UK, UK: Springer-Verlag, 1985, pp. 496–503.

[21] V. Arvind and P. Mukhopadhyay, "Derandomizing the isolation lemma and lower bounds for circuit size," in *APPROX and RANDOM*, 2008, pp. 276–289.

[22] S. A. Fenner, R. Gurjar, and T. Thierauf, "Bipartite perfect matching is in quasi-NC," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, 2016, pp. 754–763.

[23] G. L. Miller and J. Naor, "Flow in planar graphs with multiple sources and sinks," in *30th Annual Symposium on Foundations of Computer Science*, Oct 1989, pp. 112–117.

[24] M. Mahajan and K. R. Varadarajan, "A new nc-algorithm for finding a perfect matching in bipartite planar and small genus graphs," in *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing, STOC*. New York, NY, USA: ACM, 2000, pp. 351–357.

[25] R. Gurjar and T. Thierauf, "Linear matroid intersection is in quasi-NC," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, 2017, pp. 821–830.

[26] H. Narayanan, H. Saran, and V. V. Vazirani, "Randomized parallel algorithms for matroid union and intersection, with applications to arboresences and edge-disjoint spanning trees," *SIAM J. Comput.*, vol. 23, no. 2, pp. 387–397, 1994.

[27] C. H. Papadimitriou and M. Yannakakis, "The complexity of restricted spanning tree problems," *J. ACM*, vol. 29, no. 2, pp. 285–309, Apr. 1982.

[28] D. A. M. Barrington, "Quasipolynomial size circuit classes," in *Proceedings of the Seventh Annual Structure in Complexity Theory Conference*, Jun 1992, pp. 86–93.

[29] M. Mahajan and V. Vinay, "Determinant: Combinatorics, algorithms, and complexity," Tech. Rep., 1997.

[30] P. W. Beame, S. A. Cook, and H. J. Hoover, "Log depth circuits for division and related problems," *SIAM J. Comput.*, vol. 15, no. 4, pp. 994–1003, Nov. 1986.

[31] J. Edmonds, "Maximum matching and a polyhedron with $0, 1$ vertices," *Journal of Research of the National Bureau of Standards*, vol. 69, pp. 125–130, 1965.

[32] A. Schrijver, *Combinatorial Optimization - Polyhedra and Efficiency*. Springer-Verlag, Berlin, 2003.