Parameterizing the Permanent: Hardness for fixed excluded minors

Radu Curticapean* Mingji Xia†

Abstract

In the 1960s, statistical physicists discovered a fascinating algorithm for counting perfect matchings in *planar* graphs. Valiant later showed that the same problem is #P-hard for *general* graphs. Since then, the algorithm for planar graphs was extended to bounded-genus graphs, to graphs excluding $K_{3,3}$ or K_5 as a minor, and more generally, to any graph class excluding a fixed minor H that can be drawn in the plane with a single crossing. This stirred up hopes that counting perfect matchings might be polynomial-time solvable for graph classes excluding *any* fixed minor H. Alas, in this paper, we show #P-hardness for K_8-minor-free graphs by a simple and self-contained argument.

1 Introduction

A *perfect matching* in a graph G is an edge-subset $M \subseteq E(G)$ such that every vertex of G has exactly one incident edge in M. Counting perfect matchings is a very well-studied problem in counting complexity. It already starred in Valiant's seminal paper [24] that introduced the complexity class #P, where it was shown that counting perfect matchings is #P-complete. The problem has driven progress in approximate counting and underlies the so-called holographic algorithms [25, 6, 4, 5]. It also occurs outside of counting complexity, e.g., in statistical physics, via the partition function of the *dimer model* [22, 16, 17]. In algebraic complexity theory, the matrix *permanent* is an important algebraic analogue of perfect matchings counts [1]. Indeed, evaluating permanents is equivalent to counting perfect matchings in bipartite graphs: Given a bipartite input graph G on $n+n$ vertices with its $n \times n$ bi-adjacency matrix A, the permanent $\mathrm{per}(A)$ counts exactly the perfect matchings in G.

Algorithms for restricted graph classes. A long line of research, dating back to the 1960s, identified structural restrictions on G that facilitate the problem of counting perfect matchings. For the graphs of *regular lattices* [22, 16], and more generally, for *planar graphs G*, it is possible to flip the signs of some entries in the adjacency matrix to obtain a matrix A such that $\sqrt{\det(A)}$ counts the perfect matchings in G [17]. The entries to be flipped are determined by a so-called *Pfaffian orientation* of G, which can be computed in linear time for planar graphs. Overall, a polynomial-time algorithm for counting perfect matchings in planar graphs follows, the so-called *FKT method*.

Little [18] and Vazirani [27] later generalized the FKT method from planar graphs to the more general class of graphs excluding $K_{3,3}$ as a minor. Such graphs can be obtained inductively by "gluing together" planar graphs and K_5. Little showed that $K_{3,3}$-free graphs still admit a Pfaffian orientation by combining Pfaffian orientations of the individual parts, and Vazirani later obtained a polynomial-time and poly-logarithmic space algorithm for finding such an orientation.

Still working with Pfaffian orientations, it was shown by Gallucio and Loebl [14] and Tesler [23] that perfect matchings can be counted in time $4^g n^{O(1)}$ for graphs G that are embedded on a surface of genus g. In other words, the problem is fixed-parameter tractable in the parameter g. These algorithms use Pfaffian orientations to express the number of perfect matchings in G as a linear combination of 4^g determinants. (A simplified algorithm by the authors [11] bypasses the explicit use of Pfaffian orientations and instead reduces in a black-box manner to 4^g instances of counting perfect matchings in planar graphs.)

*IT University of Copenhagen and Basic Algorithms Research Copenhagen (BARC), Copenhagen, Denmark. BARC is supported by Villum Foundation Grant No. 16582.

†State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China. Supported by NSFC 61932002.

The mold of Pfaffian orientations was broken by Straub, Thierauf and Wagner [21], and Curticapean [7], who independently designed polynomial-time algorithms for counting perfect matchings in graphs excluding a K_5-minor. As such graphs do not necessarily admit Pfaffian orientations, a different algorithmic approach was needed: In hindsight, the new algorithms transferred the *protrusion replacement* technique from parameterized complexity [2] into the counting setting.

Moreover, a standard dynamic programming approach shows that counting perfect matchings is fixed-parameter tractable in graphs of bounded tree-width t. This can be improved to $2^t n^{O(1)}$ time [26], where the base 2 is optimal under the strong exponential-time hypothesis [10].

Towards excluding general fixed minors. Every tractable graph class mentioned above excludes some fixed minor H: Starting from a graph G in the class, it is not possible to obtain that fixed graph H by deleting edges/vertices and contracting edges. For example, planar graphs exclude $K_{3,3}$ and K_5, bounded-genus graphs exclude sufficiently large complete graphs, and bounded-treewidth graphs even exclude large grids. It is therefore natural to ask whether we can count perfect matchings in *any* graph class excluding fixed minors.

On the positive side, Curticapean [7] and Eppstein and Vazirani [13] lifted the algorithms for $K_{3,3}$-minor-free and K_5-minor-free graphs to H-minor-free graphs for any graph H that can be drawn in the plane with a single crossing, which includes $H = K_{3,3}$ and $H = K_5$. (Note that excluding single-crossing minors H yields different graph classes than the class of single-crossing graphs themselves. In fact, counting perfect matchings in graphs with a constant number of crossings can be reduced easily to the FKT method.) These algorithms run in fixed-parameter tractable time $f(k)n^{O(1)}$ for some function f depending only on the size k of an excluded single-crossing minor. Note that the exponent of n does not grow with k.

On the negative side, parameterized complexity rules out such fixed-parameter tractable algorithms for counting perfect matchings in *k-apex graphs* [11]; these are the graphs that are planar up to deleting k vertices. More precisely, it was previously shown by the authors that counting perfect matchings is $\#\mathsf{W}[1]$-hard on k-apex graphs, suggesting that algorithms for this problem require $n^{g(k)}$ time for $g \in \omega(1)$. As k-apex graphs exclude K_{k+5}-minors, algorithms for counting perfect matchings in K_t-free graphs in turn require $n^{g(t)}$ time for $g \in \omega(1)$. This result however still leaves open the possibility that, for any fixed $t \in \mathbb{N}$, some polynomial-time algorithm counts perfect matchings in K_t-free graphs with an exponent depending on t.

In fact, such algorithms seemed within close reach: Robertson and Seymour's graph structure theorem [19] shows that graphs G excluding a fixed H-minor can be obtained as clique-sums of graphs that are near-embeddable on surfaces of fixed genus. Very roughly speaking, this means that G is glued together from certain pieces, similar to graphs excluding $K_{3,3}$ or K_5, but with some "upgrades". These upgrades involve raising the *genus* in the decomposition pieces from 0 to some constant depending on H, adding a constant number of *apex* vertices (vertices that can connect arbitrarily to the remainder of the piece), and adding a constant number of *vortices* (graphs of bounded path-width that are aligned with the boundary of a face).

Algorithms for counting perfect matchings support most of these upgrades: The problem is fixed-parameter tractable in bounded-genus graphs, while apex vertices can be handled by brute-force in $n^{O(1)}$ time, and the general gluing operation can be handled similarly as in the simpler case of excluded single-crossing minors. It only remains to handle vortices. However, even a minimal example of vortices was unresolved: We say that a *ring blowup* is a graph obtained from a drawing of a planar graph by cloning each vertex on the outer face into two copies, as shown in Figure 1. In the terminology of the graph structure theorem, ring blowups are certain planar graphs with a single vortex. Progress towards polynomial-time algorithms for counting perfect matchings in H-minor-free graphs was halted because such algorithms were not even known for ring blowups.

Our results. We show that counting perfect matchings remains $\#\mathsf{P}$-hard in ring blowups. By known general results in graph minor theory [15], this implies the existence of a graph H such that counting perfect matchings is $\#\mathsf{P}$-hard in H-minor-free graphs. Additionally, we show that ring blowups exclude K_8-minors. (Note that they *can* contain K_7-minors, as Figure 5 shows.) We then obtain our main theorem:

THEOREM 1.1. *Counting perfect matchings is $\#\mathsf{P}$-hard in graphs excluding K_8-minors.*

A weaker version of this theorem was previously announced in a survey on parameterized counting [9]. To prove Theorem 1.1, we reduce from the $\#\mathsf{P}$-hard problem of counting perfect matchings in a graph G. Our reduction hinges upon a particular *sign-crossing gadget* (see Figure 2) that can be used to remove crossings at

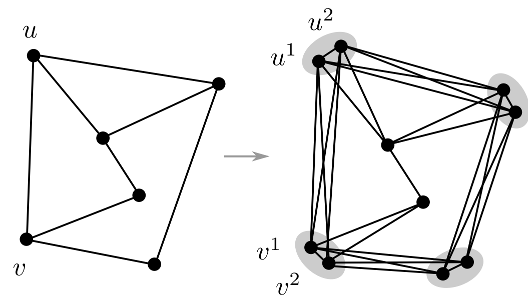


Figure 1: A ring blowup.

the cost of disrupting the perfect matching count: After inserting a sign-crossing gadget between crossing edges $e, f \in E(G)$, perfect matchings that contain *both* e and f are counted with a factor -1, and only perfect matchings containing *at most one* of e or f are counted properly. Such sign-crossing gadgets were previously used in the theory of matchgates [3] and in the hardness proof for counting perfect matchings in k-apex graphs [11]. In our proof, sign-crossings are used to transform G into a ring blowup while preserving the perfect matching count. By a surprisingly simple construction, we can ensure that sign-crossings come in equivalent pairs, so that any -1 factors introduced by sign-crossings cancel via $(-1)^2 = 1$.

2 Preliminaries

To give a self-contained proof of Theorem 1.1, we first state some preliminaries from counting complexity and graph minor theory. Graphs G will be undirected and may be edge-weighted; we implicitly consider $w : E(G) \to \mathbb{Q}$ to be the weight function. Given a vertex $v \in V(G)$, we write $I(v)$ for the set of edges incident with v. Furthermore, given a set $S \subseteq V(G)$, we write $G[S]$ for the subgraph of G induced by S.

2.1 Counting and gadgets. We define counting problems as functions $\#\mathrm{A} : \{0,1\}^* \to \mathbb{Q}$, where inputs (graphs, formulas, numbers) are implicitly encoded as bitstrings. For example, #SAT asks to count the satisfying assignments to Boolean formulas. Likewise, when given as input a graph with edge-weights from a constant-sized[1] set $W \subseteq \mathbb{Q}$, the problem #PerfMatch asks to determine the quantity

$$\#\mathrm{PerfMatch}(G) = \sum_{\substack{M \subseteq E(G) \text{ is a} \\ \text{perfect matching}}} \prod_{e \in M} w(e). \tag{2.1}$$

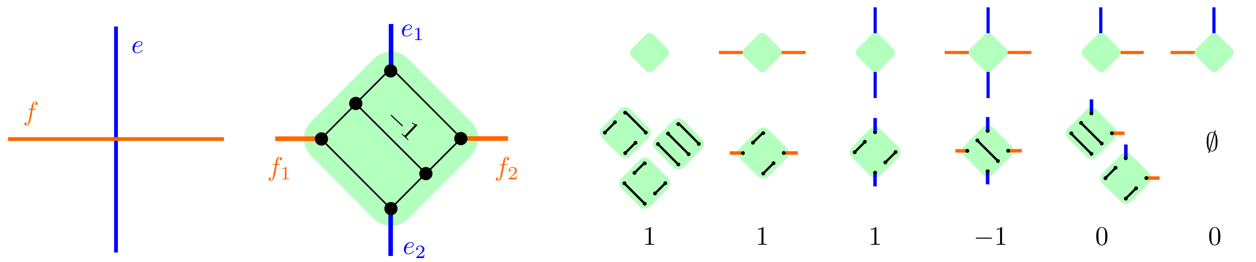
We say that #A admits a polynomial-time Turing reduction to #B if #A can be solved in polynomial time with an oracle for #B, and we say that #B is #P-hard if #SAT admits such a reduction to #B. Our hardness proofs are based on the following theorem:

THEOREM 2.1. ([24, 12]) *Counting perfect matchings is #*P*-hard, even for unweighted* 3*-regular graphs.*

Our reductions rely on certain gadgets, so-called *matchgates*. Most importantly, given a drawing of a not necessarily planar graph G with crossing edges $e, f \in E(G)$, we can replace the crossing by the planar *sign-crossing gadget* shown in Figure 2a. The resulting graph essentially counts perfect matchings in G, but with a significant twist: Any perfect matching $M \subseteq E(G)$ containing both e and f is weighted by an additional factor of -1. In other words, every perfect matching M is weighted by

$$\chi_{e,f}(M) := \begin{cases} -1 & \{e, f\} \subseteq M, \\ 1 & \text{otherwise.} \end{cases} \tag{2.2}$$

[1] Assuming constant size ensures that $\#\mathrm{PerfMatch}(G)$ can be represented with polynomially many bits.



(a) The sign-crossing gadget is inserted into a crossing between edges e, f and drawn into a small disk to avoid introducing further crossings. We call e_1, e_2, f_1, f_2 the *external edges* of the gadget.

(b) The top row shows, up to symmetry in the last two cases, the possible configurations with $0, 2$ or 4 external edges. The middle row shows the possible extensions by matchings within the gadget. The bottom row lists the total weighted sums of these extensions.

Figure 2: The planar sign-crossing gadget introduced by Cai and Gorenstein [3].

Note that a "perfectly planarizing" crossing gadget that does not introduce negative signs would render the #P-hard problem of counting perfect matchings polynomial-time solvable by reduction to the FKT method. It can even be shown unconditionally that no such gadget exists [3].

The claimed functionality of the sign-crossing gadget follows from standard techniques in the area of so-called Holant problems, see [25, 3, 8]. In the following, we give a self-contained proof.

Lemma 2.1. *Let G be a weighted graph that is drawn in the plane with crossing edges $e, f \in E(G)$ of weight 1, and let G' be obtained by inserting a sign-crossing gadget as in Figure 2a. Then, with $\chi_{e,f}$ as in (2.2), we have*

$$\#\mathrm{PerfMatch}(G') = \sum_{\substack{M \subseteq E(G) \text{ is a} \\ \text{perfect matching}}} \chi_{e,f}(M) \prod_{e \in M} w(e).$$

Proof. Let $X = \{e_1, e_2, f_1, f_2\}$ be the external edges of the sign-crossing gadget S inserted at e, f, as shown in Figure 2a. Let $G'' = G' - V(S)$ be the graph obtained from G' by removing all vertices of S. For a subset $T \subseteq X$ of external edges, write $V(T)$ for the endpoints of edges in T. Any perfect matching $M \subseteq E(G')$ can be obtained by (i) choosing a subset $T \subseteq X$ to include into M and then (ii) extending the chosen subset T by *independently* choosing a perfect matching in $S - V(T)$ and one in $G'' - V(T)$. It follows that

$$\#\mathrm{PerfMatch}(G') = \sum_{T \subseteq X} \underbrace{\#\mathrm{PerfMatch}(S - V(T))}_{=: s(T)} \cdot \#\mathrm{PerfMatch}(G'' - V(T)). \tag{2.3}$$

Now observe that any set $T \subseteq X$ with $s(T) \neq 0$ must have even cardinality, as $S - V(T)$ would otherwise have an odd number of vertices, and hence, no perfect matchings. The values of $s(T)$ for sets $T \subseteq X$ of even cardinality are calculated in Figure 2b: Each column in the figure depicts such a set T in the top row, with the perfect matchings of $S - V(T)$ listed below it. The value $s(T) = \#\mathrm{PerfMatch}(S - V(T))$ is then obtained in the bottom row as the weighted count of the listed perfect matchings.

We see that any $T \subseteq X$ with $s(T) \neq 0$ is *consistent* in that it includes none/both of $\{e_1, e_2\}$ and none/both of $\{f_1, f_2\}$. Given such a consistent set T, define $\tilde{T} \subseteq \{e, f\}$ by forgetting subscripts, i.e., include e into $\tilde{T}$ iff $\{e_1, e_2\} \subseteq T$, likewise for f. The term in (2.3) corresponding to T counts precisely those perfect matchings M in G with $M \cap \{e, f\} = \tilde{T}$, except that $s(T)$ introduces a factor of -1 if $\tilde{T} = \{e, f\}$. This proves the lemma. ∎

If several crossings are replaced by sign-crossing gadgets, Lemma 2.1 can be applied inductively; each gadget introduces a sign factor. A particularly interesting situation occurs when edges are drawn as curves rather than straight lines, as two edges e and f may then cross more than once. This will prove very useful in the next section.

Corollary 1. *Let G be an unweighted graph that is drawn in the plane. Choose $t \in \mathbb{N}$ crossings and write $e_i, f_i \in E(G)$ for the edges involved in the i-th crossing. Let G' be obtained by inserting a sign-crossing gadget at* *each of the t chosen crossings. Then we have*

$$\#\mathrm{PerfMatch}(G') = \sum_{\substack{M \subseteq E(G) \text{ is a} \\ \text{perfect matching}}} \prod_{i=1}^{t} \chi_{e_i,f_i}(M).$$

2.2 Graph minor theory. A graph H is a *minor* of G, written $H \preceq G$, if H can be obtained from G by repeated edge deletions and contractions and vertex deletions. This is equivalent to the existence of a minor model of H in G.

DEFINITION 2.1. *A* minor model *of H in G is a collection of pairwise disjoint branch sets $S_v \subseteq V(G)$ for $v \in V(H)$ such that (i) each set S_v for $v \in V(H)$ induces a connected subgraph of G, and (ii) for every edge $uv \in E(H)$, some edge of G runs between S_u and S_v.*

The *Hadwiger number* $\eta(G)$ of a graph G is the maximum $k \in \mathbb{N}$ with $K_k \preceq G$. The colored sets in Figure 5 show minor models of K_7 in ring blowups, proving that the Hadwiger number of such graphs can be at least 7.

A plane graph is a planar graph that is given together with a concrete planar embedding. We define a particular graph class from plane graphs by "blowing up" their outer faces, see Figure 1.

DEFINITION 2.2. *Given a plane graph $\hat{Q}$ with outer face O, the* blowup *of $\hat{Q}$ is the graph obtained by successively replacing each vertex $v \in O$ by two clones v^1, v^2 having the same neighborhood as v, and then adding the edge v^1v^2. We call v^1 and v^2* blowup vertices*. A* ring blowup *is any subgraph of the blowup Q of a plane graph $\hat{Q}$. We also call $\hat{Q}$ a* reduct *of Q.*

The notion of clique-sums will feature prominently in Section 4.

DEFINITION 2.3. *Let G and G' be two graphs with not necessarily disjoint vertex sets, and assume that $S = V(G) \cap V(G')$ is a clique in G and G'. A* clique-sum *$G \oplus_S G'$ is any graph that can be obtained from the union $G \cup G'$ by deleting some edges with both endpoints in S.*

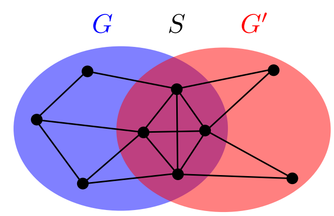


Figure 3: A clique-sum.

A standard separator argument bounds the Hadwiger number of clique-sums by those of their constituents, see [15, Lemma 2.1] for a proof.

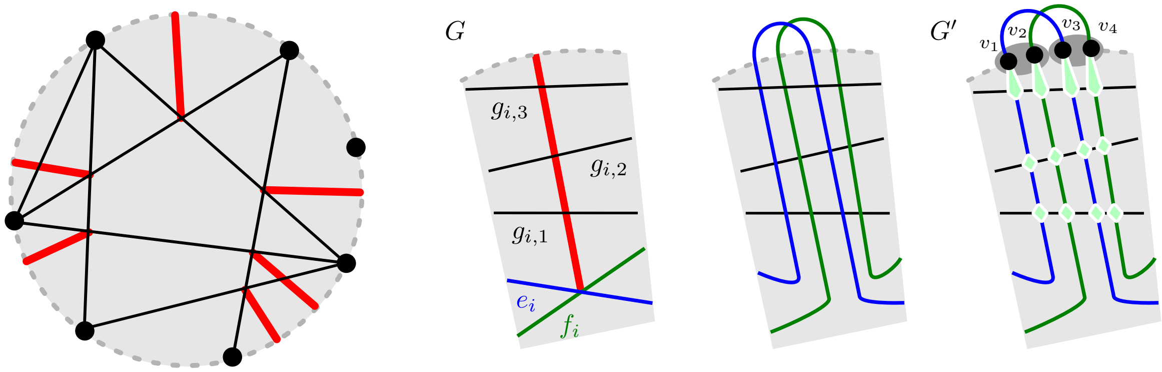
FACT 2.1. *For any graphs G, G' and any clique-sum $G \oplus_S G'$ for $S = V(G) \cap V(G')$, we have $\eta(G \oplus_S G') \le \max\{\eta(G), \eta(G')\}$.*

For a proof sketch, note that no minor model of K_t can simultaneously place some branch sets entirely within $V(G) \setminus S$ and others entirely within $V(G') \setminus S$. Hence all branch sets intersect $V(G)$ or all intersect $V(G')$. In the first case, we can delete all vertices from $V(G') \setminus S$ without losing edges between branch sets, since S is a clique. The second case is symmetric.

3 Reducing to ring blowups

In this section, we show how to transform any unweighted graph G into a ring blowup while preserving the value of #PerfMatch. The main idea, spelled out in Lemma 3.1 and illustrated in Figure 4, is to arrange the vertices of G on a circle and then bend the edges of G to push crossings across the perimeter of the circle, where they are handled via blowups. As an edge e is bent, it will introduce new crossings with other edges g, but our construction ensures that any such edge g is crossed *exactly twice*. When we then introduce sign-crossing gadgets at these crossings, any -1 factors from gadgets are guaranteed to come in pairs, so the overall product of these factors is 1. Hence, going from G to G', the value of #PerfMatch is preserved via Corollary 1. In Lemma 3.2, we then use a standard reduction in counting complexity to remove the -1 weights introduced into G' by sign-crossings.

LEMMA 3.1. *Let G be an unweighted graph with n vertices and m edges. In polynomial time, we can construct a ring blowup G' on $O(n + m^3)$ vertices and edge-weights ± 1 such that $\#\mathrm{PerfMatch}(G) = \#\mathrm{PerfMatch}(G')$ holds and all edges incident with blowup vertices of G' have weight 1.*



(a) Initial drawing of G with circle C (dashed) that induces an inner part (shaded) and outer part. The segments ℓ_i extending from crossings to C are drawn as red lines.

(b) Left: The i-th crossing, involving edges e_i, f_i, and the edges $g_{i,j}$ crossed by segment ℓ_i. Middle: The edges e_i and f_i are bent towards the outer part along ℓ_i. Each edge $g_{i,j}$ now crosses each of e_i and f_i twice. Right: Inserting sign-crossing gadgets and dragging the top vertices $v_1, \ldots, v_4$ of outermost crossing gadgets onto C. The dark ellipses show that G' is a ring blowup.

Figure 4: How Lemma 3.1 moves crossings to the outer part of G to obtain a ring blowup G'.

Proof. As shown in Figure 4a, we first place $V(G)$ on a circle C in the plane and draw the edges of G as straight lines inside of C. The placement is chosen such that no three edges intersect in a common point and such that every ray from the center to the perimeter of C contains at most one point that is a crossing or vertex of G. Both conditions can be ensured by perturbing an arbitrary placement of $V(G)$ on C.

The circle C divides the plane into two regions; we call the induced subgraphs of G contained in these regions (both including C) the *outer* and *inner* part of G. Initially, the outer part contains no edges.

Let $s \in O(m^2)$ be the number of crossings in the drawing and let $P_1, \ldots, P_s \in \mathbb{R}^2$ be their locations. For each $i \in [s]$, shoot a ray from the center of C to P_i and write ℓ_i for the segment of this ray from P_i to the perimeter of C. The segments are drawn as red lines in Figure 4a. Note that distinct rays are disjoint and contain no vertices of G. For $i \in [s]$ in sequence, write $e_i, f_i \in E(G)$ for the edges involved in crossing P_i, write m_i for the number of edges crossed by segment ℓ_i and enumerate the crossed edges as $g_{i,1}, \ldots, g_{i,m_i} \in E(G)$. We bend e_i and f_i in a sufficiently narrow neighborhood of ℓ_i to cross C, as shown in the middle part of Figure 4b. For any $j \in [m_i]$, this process adds two crossings between e_i and $g_{i,j}$ as well as two crossings between f_i and $g_{i,j}$.

After all original crossings $P_1, \ldots, P_s$ are processed in this way, we insert a sign-crossing gadget at each crossing in the inner part of G, as shown in the right part of Figure 4b. Note that no sign-crossing gadgets are inserted at crossings in the outer part. To simplify the subsequent argument, we drag some vertices of the sign-crossing gadgets onto C, as shown in Figure 4b. Overall, we obtain a new graph G' with edge-weights 1 and -1, and Corollary 1 shows that

$$\#\mathrm{PerfMatch}(G') = \sum_{\substack{M \subseteq E(G) \text{ is a} \\ \text{perfect matching}}} \prod_{i=1}^{s} \prod_{j=1}^{m_i} \underbrace{\chi^2_{e_i, g_{i,j}}(M)}_{=1} \cdot \underbrace{\chi^2_{f_i, g_{i,j}}(M)}_{=1} = \#\mathrm{PerfMatch}(G). \tag{3.4}$$

Taking inventory, while going from G to G', we replaced all crossings from the inner part with planar gadgets and added no new crossings to the inner part, as different segments ℓ_i, ℓ_j are disjoint. Via sign-crossings, we added $O(\sum_i m_i) = O(sm) = O(m^3)$ vertices into the inner part, where we recall that m_i is the number of crossings between segment ℓ_i and the edges in G. Each crossing is contained in the outer part and involves edges $v_1 v_3$ and $v_2 v_4$ for some consecutive block of vertices $v_1, \ldots, v_4$ on the circle C. The vertex blocks of different crossings are disjoint, so we can define a reduct of G' by identifying $v_1 = v_2$ and $v_3 = v_4$ in each block, as shown in Figure 4b. Then also no edges of weight -1 are incident with blowup vertices. This shows that G' is a ring blowup satisfying the specifications of the lemma. ∎

To conclude this section, we remark that negative edge-weights can be removed from the graphs constructed before while staying in the graph class of ring blowups.

Lemma 3.2. *The problem* #PerfMatch *restricted to graphs that are ring blowups with edge-weights* ± 1 *such that edges of weight* -1 *are not incident with blowup vertices (that is, restricted to the graphs constructed in Lemma 3.1) admits a polynomial-time Turing reduction to* #PerfMatch *in unweighted ring blowups.*

A standard proof of this lemma, see e.g. [8], replaces occurrences of the weight -1 with an indeterminate x; this turns the number of perfect matchings in an n-vertex graph into a polynomial $p \in \mathbb{Z}[x]$ of degree at most $d = n/2$. This polynomial can be evaluated at non-negative integer inputs $0, \ldots, d$ via planar gadgets, and the value $p(-1)$ can then be recovered from $p(0), \ldots, p(d)$ via polynomial interpolation. As Lemma 3.1 guarantees that edges of weight -1 are not incident with blowup vertices, the planar gadgets introduced in Lemma 3.2 can be contained within the inner part of the resulting graphs.

Combining Theorem 2.1 (the #P-hardness of counting perfect matchings) with Lemmas 3.1 and 3.2, we immediately obtain:

Theorem 3.1. *The problem* #PerfMatch *is* #P*-hard in unweighted ring blowups.*

Our proof even shows that #PerfMatch is #P-hard in graphs G that are obtained from plane graphs by adding edges v_1v_3 and v_2v_4 between any disjoint blocks of four consecutive vertices $v_1, \ldots, v_4$ on the outer face. However, in the arguments in Section 4, general ring blowups arise naturally.

4 Bounding the Hadwiger number

In this section, we bound the Hadwiger number of ring blowups by 7. We remark that Seese and Wessel [20] gave an upper bound of 7 on the Hadwiger number of a graph class subsuming the graphs constructed in the previous section. For completeness, we give a self-contained proof for ring blowups.

First, we show in Lemma 4.1 that it suffices to consider *simple ring blowups*, that is, blowups of plane graphs for which all but ≤ 3 vertices are contained on the outer face. In Lemma 4.2, we iteratively remove certain *complications* from simple ring blowups. When this process terminates, we obtain graphs that can be handled by trivial arguments in Lemma 4.3.

Definition 4.1. *A* simple ring *is a plane graph* Q *with outer face* O *such that* $Q - O$ *is a complete graph with* ≤ 3 *vertices. We call* $W = V(Q) \setminus O$ *the* inner face *of* Q.[2] *A* simple ring blowup *is any subgraph of the blowup of a simple ring. (Equivalently, a graph is a simple ring blowup if it has a simple ring as reduct.)*

In the following lemma, we adapt an argument by Joret and Wood [15, Lemma 3.4] to render ring blowups simple without decreasing their Hadwiger number. See Figure 5 for an illustration of the process; the last drawing shows an example of a simple ring blowup with a single vertex on the inner face. The lemma repeatedly uses the fact that, given a minor model of a graph H in another graph G, contracting an edge contained a branch set canonically induces a minor model of H in the resulting graph.

Lemma 4.1. *For any ring blowup* G*, there is a simple ring blowup* Q *with* $\eta(G) \leq \eta(Q)$*.*

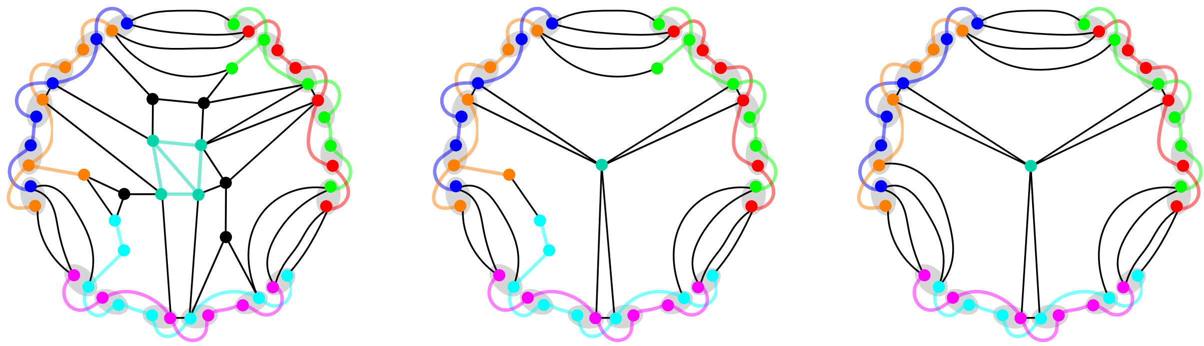
Proof. We abbreviate $t = \eta(G)$. The lemma holds for $t \leq 4$, since there clearly are simple ring blowups containing K_4-minors. We may therefore assume $t \geq 5$ in the following.

Let $\hat{G}$ be a reduct of G, with outer face O, and let G be drawn as a blowup of $\hat{G}$, as shown in Figure 5. Fix a minor model $S_1, \ldots, S_t \subseteq V(G)$ of K_t in G, define $G' := G$ and proceed as follows.

1. Delete from G' all vertices not contained in $S_1 \cup \ldots \cup S_t$. Then the sets $S_1, \ldots, S_t$ still yield a minor model of K_t in the resulting graph (which we still call G') as no edges between branch sets were deleted.

2. Contract every branch set S_i not containing blowup vertices. This yields a minor model of K_t in the resulting graph. The $\ell \leq t$ vertices $W = \{w_1, \ldots, w_\ell\}$ resulting from the contraction induce a planar drawing of K_ℓ, so we have $\ell \leq 4$. We may even assume $\ell = 3$: Otherwise, if $\ell = 4$, then one of the vertices, say w_4, would be enclosed by the cycle on $W \setminus \{w_4\}$ in the drawing of G'. But then w_4 cannot have edges to the $t - \ell$ other branch sets, so the t overall branch sets cannot form a minor model of K_t for $t \geq 5$.

[2] Technically, this need not be a face.



(a) The initial graph G with a minor model of K_7, indicated by vertex colors. For clarity, edges contained in branch sets are also colored.

(b) Steps 1-2 removed vertices not contained in branch sets and contracted the turquoise branch set, which contains no blowup vertices.

(c) Step 3 successively contracted all edges that run between blowup vertices and non-blowup vertices. The result is a simple ring blowup.

Figure 5: How Lemma 4.1 cleans up a ring blowup while maintaining a K_t-minor.

3. For every edge uv fully contained in a branch set, where u is a blowup vertex and v is not, contract uv into u. This still yields a minor model of K_t in the resulting graph G'.

Summing up, we see that G' contains a K_t-minor, so it suffices to prove that G' is a simple ring blowup. Note that the ≤ 3 vertices in W form a clique in G', while all other vertices are blowup vertices. By applying the operations used to transform G into G' on the reduct of G, we obtain a reduct of G' that is a simple ring. ◻

By removing certain "complications", it is possible to simplify simple rings (and their blowups) even further.

Definition 4.2. *Given a simple ring $\hat{Q}$ with outer face O and inner face W, a* complication *in $\hat{Q}$ is*

(a) any edge between vertices $u, v \in O$ that are not consecutive in the cyclic order of O, or

(b) for any vertex $w \in W$, any neighbor $o \in O$ of w after its first two neighbors in the cyclic order of O.

Figure 6 illustrates complications of the different types. We show in the following lemma that it suffices to bound the Hadwiger number of the blowups of triangulated complication-free simple rings. Here, we say that a graph is *triangulated* if every face but its outer face is a triangle.

Lemma 4.2. *If $\eta(Z) \leq 7$ holds for the blowup Z of any triangulated and complication-free simple ring $\hat{Z}$, then $\eta(Q) \leq 7$ holds for any simple ring blowup Q.*

Proof. Let $\hat{Q}$ be a simple ring with complications, outer face O, and inner face W. Let Q be the blowup of $\hat{Q}$. We may assume $\hat{Q}$ to be triangulated; this can be ensured by adding edges, which does not decrease $\eta(Q)$.

The goal is to decompose $\hat{Q}$ along clique-sums into simple rings with strictly less complications; the triangulation will be kept intact along the way. An inductive application of Fact 2.1 then proves the lemma. The following notation will be useful: For vertex sets $S \subseteq V(\hat{Q})$, let $B(S)$ be obtained by replacing each vertex $v \in S \cap O$ with v^1 and v^2.

We start by removing all complications of type (a). To this end, consider such a complication involving $u, v \in O$. Define $S = \{u, v\}$. There exist sets $L, R \subseteq V(\hat{Q})$ such that $\hat{Q}$ is a clique-sum $\hat{Q}[L] \oplus_S \hat{Q}[R]$, as shown in the left part of Figure 6. This in turn means that the blowup Q is a clique-sum $Q[B(L)] \oplus_{B(S)} Q[B(R)]$. Via Fact 2.1, it suffices to show that $Q[B(L)]$ and $Q[B(R)]$ are K_8-minor-free. But $Q[B(L)]$ and $Q[B(R)]$ are the blowups of $\hat{Q}[L]$ and $\hat{Q}[R]$, which are triangulated simple rings with at least one complication of type (a) less. By induction, we may therefore assume in the following that $\hat{Q}$ contains no complications of type (a).

Now consider a complication of type (b) with $w \in W$ and neighbors $u_1, u_2, u_3 \in O$ of w that are consecutive vertices in the cyclic order of O. Note that we may indeed assume the neighbors to be consecutive, since $\hat{Q}$ is triangulated and all complications of type (a) were processed. Define $S = \{w, u_1, u_3\}$ and $T = S \cup \{u_2\}$. Then

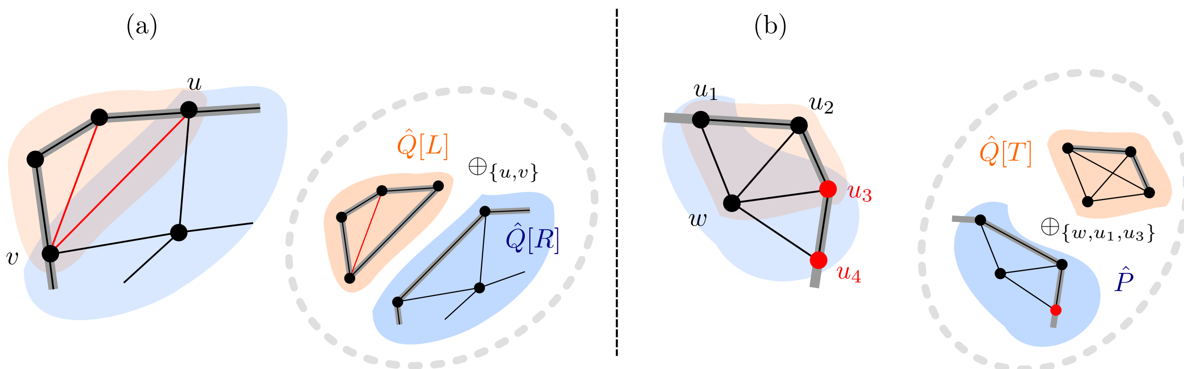


Figure 6: This figure exemplifies complications of type (a) and (b) in a simple ring $\hat{Q}$ and illustrates the main steps in the proof of Lemma 4.2. The outer face of $\hat{Q}$ is drawn as a thick line. Complications are marked red. The shading and encircled drawings indicate the clique-sums arising in the proof of Lemma 4.2.

$\hat{Q}$ is a clique-sum $\hat{P} \oplus_S \hat{Q}[T]$ for the simple ring $\hat{P}$ obtained from $\hat{Q}$ by removing u_2 and adding the edge u_1u_3, as shown in the right part of Figure 6. This in turn implies that Q is a clique-sum $P \oplus_{B(S)} Q[B(T)]$, where P is the blowup of $\hat{P}$. We observe that $Q[B(T)]$ has only 7 vertices, so it suffices to exclude K_8 from P, where $\hat{P}$ is a triangulated simple ring that still has no complications of type (a), and complication of type (b) less. By induction, we may therefore assume that $\hat{Q}$ has no complications at all. This proves the lemma. ∎

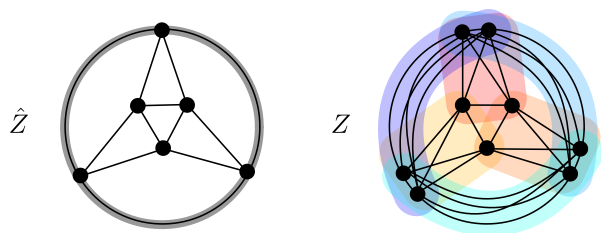
Finally, an elementary case distinction allows us to handle complication-free ring blowups.

Lemma 4.3. *For any triangulated and complication-free simple ring $\hat{Z}$ with blowup Z, we have $\eta(Z) \leq 7$.*

Proof. Let O and W with $|W| \leq 3$ be the outer and inner faces of $\hat{Z}$. Every edge $e \in E(\hat{Z})$ either has both endpoints in W, one endpoint in O and W each, or both endpoints of e lie consecutively on O since Z has no complications of type (a). If $W = \emptyset$, then $|V(\hat{Z})| \leq 3$, since $\hat{Z}$ is triangulated. In this case, we already obtain $\eta(Z) \leq |V(Z)| \leq 6$.

Therefore, assume $W \neq \emptyset$ in the following. Then any vertex $w \in W$ has ≤ 2 neighbors in O, since $\hat{Z}$ has no complications of type (b). Furthermore, since $\hat{Z}$ is triangulated, any pair of vertices in W shares a neighbor in O, because $\hat{Z}$ would otherwise contain a chordless cycle of length 4. This implies the following:

- If $|W| \leq 2$, then $|O| \leq 2$ and thus $\eta(Z) \leq |V(Z)| \leq 6$.
- If $|W| = 3$, then $\hat{Z}$ and its blowup Z are the following graphs, where $|V(Z)| = 9$ and $|E(Z)| = 30$.



To obtain a K_8-minor from the 9-vertex graph Z, one vertex must be deleted or one edge must be contracted. Deleting a vertex removes at least 6 edges. Contracting an edge reduces the number of edges by least 3, since every edge in Z is contained in some K_4-subgraph (shown above as colored blobs) and contracting an edge of K_4 yields a K_3, thus losing 3 edges. It follows that any 8-vertex minor of Z has at most $27 < 28 = |E(K_8)|$ edges and therefore cannot be K_8.

This covers all cases for $|W|$, thus proving the lemma. ∎

The proof of our main theorem is now immediate.

Proof. [Proof of Theorem 1.1.] By Theorem 3.1, the problem #PerfMatch is #P-hard in unweighted ring blowups G. It remains to show that such graphs G exclude K_8-minors. By Lemma 4.1, we have $\eta(G) \leq \eta(Q)$ for some simple ring blowup Q. By Lemma 4.2, we have $\eta(Q) \leq 7$ if $\eta(Z) \leq 7$ holds for all blowups of triangulated and complication-free ring graphs $\hat{Z}$, which in turn is true by Lemma 4.3. This concludes the proof. ∎

5 Conclusion and outlook

We showed that the FKT method for planar graphs cannot be extended to graphs excluding arbitrary fixed minors. Our work leaves open an exhaustive classification of the minors whose exclusion renders #PerfMatch polynomial-time solvable. This is not an artifact of our analysis: As Figure 5 shows, the graphs constructed by our reduction *can* contain K_7-minors, so our reduction inherently fails to address the open case of K_7-minor-free #PerfMatch. This prompts the obvious question:

QUESTION 1. *What is the complexity of* #PerfMatch *in graphs excluding K_6 or K_7? More generally, given any fixed graph H, what is the complexity of H-minor-free* #PerfMatch*?*

Turning towards a bigger picture, it is also interesting to investigate which other counting problems benefit from excluded minors. This can be studied systematically in the framework of *Holant problems*, of which counting perfect matchings constitutes a representative example.

In a future version of this paper, we rule out $\exp(o(\sqrt{n}))$ time algorithms for #PerfMatch with edge-weights ± 1 under the exponential-time hypothesis. Note that graphs excluding fixed minors have tree-width $O(\sqrt{n})$, and therefore standard algorithms for counting perfect matchings in graphs of bounded tree-width yield matching $\exp(O(\sqrt{n}))$ time upper bounds on H-minor-free graphs.

Acknowledgements

The first author thanks Dániel Marx for pointing out connections between counting problems and graph minors at the Dagstuhl seminar 10481 in 2010. He also thanks Marcin Pilipczuk for discussions on this topic at the Simons Institute for the Theory of Computing in 2015.

References

[1] Manindra Agrawal. Determinant versus permanent. In *Proceedings of the 25th International Congress of Mathematicians, ICM 2006*, volume 3, pages 985–997, 2006.

[2] Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (meta) kernelization. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 629–638. IEEE Computer Society, 2009.

[3] Jin-Yi Cai and Aaron Gorenstein. Matchgates revisited. *Theory of Computing*, 10:167–197, 2014.

[4] Jin-Yi Cai and Pinyan Lu. Holographic algorithms: From art to science. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 401–410, New York, NY, USA, 2007. ACM.

[5] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms by Fibonacci gates and holographic reductions for hardness. In *Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 644–653. IEEE Computer Society, 2008.

[6] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms with matchgates capture precisely tractable planar #CSP. *CoRR*, abs/1008.0683, 2010. Also appeared in FOCS 2010.

[7] Radu Curticapean. Counting perfect matchings in graphs that exclude a single-crossing minor. *CoRR*, abs/1406.4056, 2014.

[8] Radu Curticapean. *The simple, little and slow things count: on parameterized counting complexity.* PhD thesis, Saarland University, 2015.

[9] Radu Curticapean. Counting problems in parameterized complexity. In Christophe Paul and Michal Pilipczuk, editors, *13th International Symposium on Parameterized and Exact Computation, IPEC 2018, August 20-24, 2018, Helsinki, Finland*, volume 115 of *LIPIcs*, pages 1:1–1:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[10] Radu Curticapean and Dániel Marx. Tight conditional lower bounds for counting perfect matchings on graphs of bounded treewidth, cliquewidth, and genus. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1650–1669. SIAM, 2016.

[11] Radu Curticapean and Mingji Xia. Parameterizing the permanent: Genus, apices, minors, evaluation mod 2k. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 994–1009. IEEE Computer Society, 2015.

[12] Paul Dagum and Michael Luby. Approximating the permanent of graphs with large factors. *Theor. Comput. Sci.*, 102(2):283–305, 1992.

[13] David Eppstein and Vijay V. Vazirani. NC algorithms for computing a perfect matching, the number of perfect matchings, and a maximum flow in one-crossing-minor-free graphs. In Christian Scheideler and Petra Berenbrink, editors, *The 31st ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2019, Phoenix, AZ, USA, June 22-24, 2019*, pages 23–30. ACM, 2019.

[14] Anna Galluccio and Martin Loebl. On the theory of Pfaffian orientations. I. Perfect matchings and permanents. *Electronic Journal of Combinatorics*, 6, 1998.

[15] Gwenaël Joret and David R. Wood. Complete graph minors and the graph minor structure theorem. *Journal of Combinatorial Theory, Series B*, 103(1):61–74, 2013.

[16] Pieter W. Kasteleyn. The statistics of dimers on a lattice: I. The number of dimer arrangements on a quadratic lattice. *Physica*, 27(12):1209 – 1225, 1961.

[17] Pieter W. Kasteleyn. Graph Theory and Crystal Physics. In *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, 1967.

[18] Charles Little. An extension of Kasteleyn's method of enumerating the 1-factors of planar graphs. In *Combinatorial Mathematics*, LNCS, pages 63–72. 1974.

[19] Neil Robertson and Paul D. Seymour. Graph minors. XVI. excluding a non-planar graph. *J. Comb. Theory, Ser. B*, 89(1):43–76, 2003.

[20] Detlef Seese and W. Wessel. Grids and their minors. *J. Comb. Theory, Ser. B*, 47(3):349–360, 1989.

[21] Simon Straub, Thomas Thierauf, and Fabian Wagner. Counting the number of perfect matchings in K_5-free graphs. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 66–77, 2014.

[22] H. N. V. Temperley and Michael E. Fisher. Dimer problem in statistical mechanics - an exact result. *Philosophical Magazine*, 6(68):1478–6435, 1961.

[23] Glenn Tesler. Matchings in graphs on non-orientable surfaces. *J. Comb. Theory, Ser. B*, 78(2):198–231, 2000.

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

[25] Leslie G. Valiant. Holographic algorithms. *SIAM J. Comput.*, 37(5):1565–1594, 2008.

[26] Johan M. M. van Rooij, Hans L. Bodlaender, and Peter Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In Amos Fiat and Peter Sanders, editors, *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, volume 5757 of *Lecture Notes in Computer Science*, pages 566–577. Springer, 2009.

[27] Vijay V. Vazirani. NC algorithms for computing the number of perfect matchings in $K_{3,3}$-free graphs and related problems. *Inf. Comput.*, 80(2):152–164, 1989.