A Complexity Trichotomy for Approximately Counting List H-Colourings*†‡§

Andreas Galanis[1], Leslie Ann Goldberg[2], and Mark Jerrum[3]

1 University of Oxford, Oxford, UK
2 University of Oxford, Oxford, UK
3 Queen Mary University of London, London, UK

Abstract

We examine the computational complexity of approximately counting the list H-colourings of a graph. We discover a natural graph-theoretic trichotomy based on the structure of the graph H. If H is an irreflexive bipartite graph or a reflexive complete graph then counting list H-colourings is trivially in polynomial time. Otherwise, if H is an irreflexive bipartite permutation graph or a reflexive proper interval graph then approximately counting list H-colourings is equivalent to #BIS, the problem of approximately counting independent sets in a bipartite graph. This is a well-studied problem which is believed to be of intermediate complexity – it is believed that it does not have an FPRAS, but that it is not as difficult as approximating the most difficult counting problems in #P. For every other graph H, approximately counting list H-colourings is complete for #P with respect to approximation-preserving reductions (so there is no FPRAS unless NP = RP). Two pleasing features of the trichotomy are (i) it has a natural formulation in terms of hereditary graph classes, and (ii) the proof is largely self-contained and does not require any universal algebra (unlike similar dichotomies in the weighted case). We are able to extend the hardness results to the bounded-degree setting, showing that all hardness results apply to input graphs with maximum degree at most 6.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics

Keywords and phrases approximate counting, graph homomorphisms, list colourings

Digital Object Identifier 10.4230/LIPIcs.ICALP.2016.46

1 Overview

We study the complexity of approximately counting the list H-colourings of a graph. List H-colourings generalise H-colourings in the same way that list colourings generalise proper vertex colourings. Fix an undirected graph H, which may have loops but not parallel edges. Given a graph G, an *H-colouring of G* is a homomorphism from G to H – that is, a mapping $\sigma : V(G) \to V(H)$ such that, for all $u, v \in V(G)$, $\{u, v\} \in E(G)$ implies $\{\sigma(u), \sigma(v)\} \in E(H)$. If we identify the vertex set $V(H)$ with a set $Q = \{1, 2, \ldots, q\}$ of "colours", then we can

* A full version of the paper containing detailed proofs is available at `http://arxiv.org/abs/1602.03985`. The numbering here matches the full version.

† The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) ERC grant agreement no. 334828. The paper reflects only the authors' views and not the views of the ERC or the European Commission. The European Union is not liable for any use that may be made of the information contained therein.

‡ This work was partially supported by the EPSRC grant EP/N004221/1.

§ This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing.

43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016).
Editors: Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi;
Article No. 46; pp. 46:1–46:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

think of the mapping σ as specifying a colouring of the vertices G, and we can interpret the graph H as specifying the allowed colour adjacencies: adjacent vertices in G can be assigned colours i and j, if and only if vertices i and j are adjacent in H.

Now consider the graph G together with a collection of sets $\mathbf{S} = \{S_v \subseteq Q : v \in V(G)\}$ specifying allowed colours at each of the vertices. A *list H-colouring of* $(G, \mathbf{S})$ is an H-colouring σ of G satisfying $\sigma(v) \in S_v$, for all $v \in V$. In the literature, the set S_v is referred to as the "list" of allowed colours at vertex v, but there is no implied ordering on the elements of S_v – S_v is just a set of allowed colours.

Suppose that H is a *reflexive* graph (i.e., a graph in which each vertex has a loop). Feder and Hell [4] considered the complexity of determining whether a list H-colouring exists, given an input $(G, \mathbf{S})$. They showed that the problem is in FP if H is an interval graph, and that it is NP-complete, otherwise. Feder, Hell and Huang [5] studied the same problem in the case where H is *irreflexive* (i.e., H has no loops). They showed that the problem is in FP if H is a circular arc graph of clique covering number two (which is the same as being the complement of an interval bigraph [12]), and that it is NP-hard, otherwise. Finally, Feder, Hell and Huang [6] generalised this result to obtain a dichotomy for all H. They introduced a new class of graphs, called bi-arc graphs, and showed that the problem is in FP if H is a bi-arc graph, and NP-complete, otherwise.

We are concerned with the computational complexity of counting list H-colourings. Specifically we are interested in how the complexity of the following computational problem depends on H.

Name $\#\text{List-}H\text{-Col}$.
Instance A graph G and a collection of colour sets $\mathbf{S} = \{S_v \subseteq Q : v \in V(G)\}$, where $Q = V(H)$.
Output The number of list H-colourings of $(G, \mathbf{S})$.

Note that it is of no importance whether we allow or disallow loops in G – a loop at vertex $v \in V(G)$ can be encoded within the set S_v – so we adopt the convention that G is loop-free. As in the case of the decision problem, H is a parameter of the problem – it does not form part of the problem instance. Sometimes we obtain sharper results by introducing an additional parameter Δ, which is an upper bound on the degrees of the vertices of G. Thus $\#\text{List-}H\text{-Col}(\Delta)$ is the special case of $\#\text{List-}H\text{-Col}$ in which the graph G has degree at most Δ. Although $\#\text{List-}H\text{-Col}$ and $\#\text{List-}H\text{-Col}(\Delta)$ are the main objects of study in this paper, we occasionally need to discuss the more basic versions of these problems without lists.

Name $\#H\text{-Col}$.
Instance A graph G.
Output The number of H-colourings of G.

Once again, $\#H\text{-Col}(\Delta)$ is the special case of $\#H\text{-Col}$ in which the degree of G is at most Δ. To illustrate the definitions, let K_2' be the first graph illustrated in Figure 1, consisting of two connected vertices with a loop on vertex 2. $\#K_2'\text{-Col}$ is the problem of counting independent sets in a graph since the vertices mapped to colour 1 by any homomorphism form an independent set. Let K_3 be the complete irreflexive graph on three vertices. Then $\#K_3\text{-Col}$ is the problem of counting the proper 3-colourings of a graph.

The computational complexity of computing exact solutions to $\#H\text{-Col}$ and $\#H\text{-Col}(\Delta)$ was determined by Dyer and Greenhill [3]. Dyer and Greenhill showed that $\#H\text{-Col}$ is in FP if H is a complete reflexive graph or a complete bipartite irreflexive graph, and $\#H\text{-Col}$

is #P-complete otherwise. Their dichotomy also extends to the bounded-degree setting. In particular, they showed that if H is not a complete reflexive graph or a complete bipartite irreflexive graph then there is an integer Δ_H such that, for all $\Delta \geq \Delta_H$, $\#H$-Col(Δ) is #P-complete.

Since the polynomial-time cases in Dyer and Greenhill's dichotomy clearly remain solvable in polynomial-time in the presence of lists, their dichotomy for $\#H$-Col carries over to #List-H-Col without change. In other words, there is no difference between the complexity of $\#H$-Col and #List-H-Col as far as exact computation is concerned. However, this situation changes if we consider approximate counting, and this is the phenomenon that we explore in this paper.

With a view to reaching the statement of the main results as quickly as possible, we defer precise definitions of the relevant concepts to Section 2, and provide only indications here. From graph theory we import a couple of well studied hereditary graph classes, namely bipartite permutation graphs and proper interval graphs. These classes each have several equivalent characterisations, and we give two of these, namely, excluded subgraph and matrix characterisations, in Section 2. It is sometimes useful to restrict the definition of proper interval graphs to simple graphs. However, in this paper, as in [4], we consider reflexive proper interval graphs.

From complexity theory we need the definitions of a Fully Polynomial Randomised Approximation Scheme (FPRAS), of approximation-preserving (AP-) reducibility, and of the counting problems #SAT and #BIS. An FPRAS is a randomised algorithm that produces approximate solutions within specified relative error with high probability in polynomial time. An AP-reduction from problem Π to problem Π' is a randomised Turing reduction that yields close approximations to Π when provided with close approximations to Π'. It meshes with the definition of an FPRAS in the sense that the existence of an FPRAS for Π' implies the existence of an FPRAS for Π. The problem of counting satisfying assignments of a Boolean formula is denoted by #SAT. Every counting problem in #P is AP-reducible to #SAT, so #SAT is said to be complete for #P with respect to AP-reductions. It is known that there is no FPRAS for #SAT unless RP = NP. The problem of counting independent sets in a bipartite graph is denoted by #BIS. The problem #BIS appears to be of intermediate complexity: there is no known FPRAS for #BIS (and it is generally believed that none exists) but there is no known AP-reduction from #SAT to #BIS. Indeed, #BIS is complete with respect to AP-reductions for a complexity class $\#\mathrm{RH}\Pi_1$ which is discussed further in the full version.

We say that a problem Π is #SAT-hard if there is an AP-reduction from #SAT to Π, that it is #SAT-easy if there is an AP-reduction from Π to #SAT, and that it is #SAT-equivalent if both are true. Note that all of these labels are about the difficulty of *approximately* solving Π, not about the difficulty of exactly solving it. Similarly, Π is said to be #BIS-hard if there is an AP-reduction from #BIS to Π, #BIS-easy if there is an AP-reduction from Π to #BIS, and #BIS-equivalent if there are both.

Our main result is a trichotomy for the complexity of approximating #List-H-Col.

▶ **Theorem 1.** *Suppose that H is a connected undirected graph (possibly with loops).*

(i) *If H is an irreflexive complete bipartite graph or a reflexive complete graph then* #List-H-Col *is in* FP.

(ii) *Otherwise, if H is an irreflexive bipartite permutation graph or a reflexive proper interval graph then* #List-H-Col *is* #BIS-*equivalent.*

(iii) *Otherwise,* #List-H-Col *is* #SAT-*equivalent.*

▶ Remarks. **1.** The assumption that H is connected is made without loss of generality,

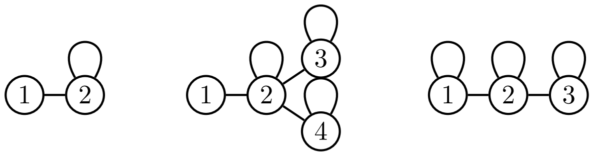

Figure 1 K_2', 2-wrench and P_3^*.

since the complexity of #LIST-H-COL is determined by the maximum complexity of #LIST-H'-COL over all connected components H' of H.

2. Part (ii) of Theorem 2 can be strengthened. For the graphs H covered by this part of the theorem, #LIST-H-COL is actually complete for the complexity class $\#\mathrm{RH}\Pi_1$. See the full version for a definition of $\#\mathrm{RH}\Pi_1$ and a proof of membership of #LIST-H-COL in $\#\mathrm{RH}\Pi_1$.

Theorem 1 also extends to the bounded-degree case.

▶ **Theorem 2.** *Suppose that H is a connected undirected graph (possibly with loops).*

(i) *If H is an irreflexive complete bipartite graph or a reflexive complete graph then, for all Δ, #LIST-H-COL(Δ) is in* FP.

(ii) *Otherwise, if H is an irreflexive bipartite permutation graph or a reflexive proper interval graph then, for all $\Delta \geq 6$, #LIST-H-COL(Δ) is #BIS-equivalent.*

(iii) *Otherwise, for all $\Delta \geq 6$, #LIST-H-COL(Δ) is #SAT-equivalent. Further, if H is reflexive or irreflexive, #LIST-H-COL(Δ) is #SAT-equivalent for $\Delta \geq 3$.*

▶ Remarks.

1. The condition $\Delta \geq 6$ is necessary for any hardness result that holds for all graphs H. In particular, there is a graph H that is not an irreflexive complete bipartite graph or a reflexive complete graph but for which #LIST-H-COL(Δ) has an FPTAS. An example is the graph $H = K_2'$ for which Weitz's self-avoiding walk algorithm [22] gives an FPTAS for #LIST-H-COL(Δ) for $\Delta \leq 5$.
2. In general, the lowest value of the degree bound Δ such that #LIST-H-COL(Δ) is computationally hard depends on the particular graph H.

Proof of Theorems 1 and 2. Part (i) is trivial. Part (ii) follows from Lemmas 13 and 14. Part (iii) follows from Lemmas 7, 9, and 11. ◀

The most obvious issue raised by our theorems is the computational complexity of approximately counting H-colourings (in the absence of lists). This question was extensively studied by Kelk [14] and others, and appears much harder to resolve, even when there are no degree bounds. It is known [7] that #H-COL is #BIS-hard for every connected undirected graph H that is neither an irreflexive bipartite permutation graph nor a reflexive proper interval graph. It is not known for which connected H the problem is #BIS-easy and for which it is #SAT-equivalent, and whether one or the other always holds. In fact, there are specific graphs H, two of them with as few as four vertices, for which the complexity of #H-COL is unresolved. It is far from clear that a trichotomy should be expected, and in fact there may exist an infinite sequences (H_t) of graphs for which #H_t-COL is reducible to #H_{t+1}-COL but not vice versa. Some partial results and speculations can be found in [14].

As we noted, #H-COL and #LIST-H-COL have the same complexity as regards exact computation. However, for approximate computation they are different, assuming (as is widely believed) that there is no AP-reduction from #SAT to #BIS. An example is provided by the 2-wrench (see Figure 1). It is known [2, Theorem 21] that #2-wrench-Col is #BIS-equivalent, but we know from Theorem 1 that the list version #List-2-wrench-Col is #SAT-equivalent since the 2-wrench is neither irreflexive nor reflexive. One way to see that #List-2-wrench-Col is #SAT-equivalent is to note that the 2-wrench contains K_2' as an induced subgraph, and that this induced subgraph can be "extracted" using the list constraints $S_v = \{1, 2\}$, for all $v \in V(G)$. But #List-K_2'-Col is already known to be #SAT-equivalent [2, Theorem 1]. Indeed, systematic techniques for extracting hard induced subgraphs form the main theme of the paper. It is for this reason that the theory of hereditary graph classes comes into play, just as in [6].

Another recent research direction, at least in the unbounded-degree case, is towards weighted versions of list colouring. Here, the graph H is augmented by edge-weights, specifying for each pair of colours i, j, the cost of assigning i and j to adjacent vertices in G. The computational complexity of obtaining approximate solutions was studied by Chen, Dyer, Goldberg, Jerrum, Lu, McQuillan and Richerby [1], and by Goldberg and Jerrum [11]. There is a trichotomy for the case in which the input has no degree bound, but this is obtained in a context where *individual* spins at vertices are weighted and not just the interactions between *pairs* of adjacent spins. In this paper we have restricted the class of problems under consideration to ones having 0,1-weights on interactions, but at the same time we have restricted the problem instances to ones having 0,1-weights on individual spins. So we have a different tradeoff and the results from the references that we have just discussed do not carry across, even in the unbounded-degree setting. Indeed, towards the end of the paper, in Section 5, we give an example to show that Theorem 1 is not simply the restriction of earlier results to 0,1-interactions (not merely because the proofs differ, but, in a stronger sense, because the results themselves are different).

Two things are appealing about our theorems. First, unlike the weighted classification theorems [1], here the truth is pleasingly simple. The trichotomies for #List-H-Col and #List-H-Col(Δ) have a simple, natural formulation in terms of hereditary graph classes. Second, the proofs of the theorems are largely self-contained. The proofs do not rely on earlier works such as [1], which require multimorphisms and other deep results from universal algebra. The proof of Theorem 1 is self-contained apart from some very elementary and well-known starting points, which are collected together in Lemma 6. The proof of Theorem 2 is similarly self-contained, though it additionally relies on recent results [18, 8] about approximating the partition function of the anti-ferromagnetic Ising model on bounded degree graphs (these are also contained in Lemma 6).

2 Complexity- and graph-theoretic preliminaries

As the complexity of computing exact solutions of #List-H-Col is well understood, we focus on the complexity of computing approximations. The framework for this has already been explained in many papers, so we provide an informal description only here and direct the reader to Dyer, Goldberg, Greenhill and Jerrum [2] for precise definitions.

The standard notion of efficient approximation algorithm is that of a *Fully Polynomial Randomised Approximation Scheme* (or FPRAS). This is a randomised algorithm that is required to produce a solution within relative error specified by a tolerance $\varepsilon > 0$, in time polynomial in the instance size and ε^{-1}. Evidence for the non-existence of an FPRAS for a problem Π can be obtained through *Approximation-Preserving* (or AP-) *reductions*. These are randomised polynomial-time Turing reductions that preserve (closely enough) the error tolerance. The set of problems that have an FPRAS is closed under AP-reducibility.

Every problem in #P is AP-reducible to #SAT, so #SAT is complete for #P with respect to AP-reductions. The same is true of the counting version of any NP-complete decision problem. It is known that these problems do not have an FPRAS unless RP = NP. On the other hand, using the bisection technique of Valiant and Vazirani [20, Corollary 3.6], we know that #SAT can be approximated (in the FPRAS sense) by a polynomial-time probabilistic Turing machine equipped with an oracle for the decision problem SAT.

In the statement and proofs of our theorems we refer to two hereditary graph classes. A class of undirected graphs is said to be *hereditary* if it is closed under taking induced subgraphs. The classes of bipartite permutation graphs and proper interval graphs have been widely studied and many equivalent characterisations of them are known. We are concerned with the excluded subgraph and matrix characterisations.

A graph is a *bipartite permutation graph* if and only if it contains none of the following as an induced subgraph: X_3, X_2, T_2 or a cycle C_ℓ of length ℓ not equal to four. (Refer to Figure 2 for specifications of X_3, X_2 and T_2.) This characterisation was noted by Köhler [15], who observed that it follows from an excluded subgraph characterisation of Gallai [9, 10]. The argument is given by Hell and Huang [12], in the proof of their Theorem 3.4, in particular parts (iv) and (vi).

A graph is a *proper interval graph* if and only if it contains none of the following as an induced subgraph: the claw, the net, S_3 or a cycle C_ℓ of length ℓ at least four. (Refer to Figure 3 for specifications of the claw, the net and S_3.) This characterisation is due to Wegner [21] and Roberts [17], and is stated is by Jackowski [13] as his Theorem 1.4, specifically the equivalence of (i) and (iii). In this context, note that a *chordal* graph is one that contains no induced cycles of length other than three.

The two graph classes also have matrix characterisations. Say that a 0,1-matrix $A = (A_{i,j} : 1 \le i \le n, 1 \le j \le m)$ has *staircase form* if the 1s in each row are contiguous and the following condition is satisfied: letting $\alpha_i = \min\{j : A_{i,j} = 1\}$ and $\beta_i = \max\{j : A_{i,j} = 1\}$, we require that the sequences (α_i) and (β_i) are non-decreasing. It is automatic that the columns share the contiguity and monotonicity properties, so the property of having staircase form is in fact invariant under matrix transposition.

A graph is a bipartite permutation graph if the rows and columns of its biadjacency matrix can be (independently) permuted so that the resulting biadjacency matrix has staircase form. This characterisation is presented by Spinrad, Brandstädt and Stewart [19], specifically the equivalence of (i) and (ii) in their Theorem 1.

A graph is a proper interval graph if the rows and columns of its adjacency matrix can be (simultaneously) permuted so that the resulting adjacency matrix has staircase form. This fact comes directly from the characterisation of proper interval graphs that gives the class its name, namely, that they are graphs which have an interval intersection model in which no interval is a proper subset of another. The ordering of intervals by left endpoint (which is the same as the ordering by right endpoint) gives the required permutation of rows and columns.

As we mentioned in Section 1, an appealing feature of our theorems is that our proofs are largely self-contained. The only pre-requisites for the proof are complexity results classifying some very well-known approximation problems. These are collected in Lemma 6, which is proved in the full version. For this, we will use the graph K_2' defined in Section 1 (see Figure 1), the path P_4 of length three (with four vertices) and the problem #1P1NSAT of counting the satisfying assignments of a CNF formula in which each clause has at most one negated literal and at most one unnegated literal. We will also use the following definition.

▶ **Definition 4.** Let $0 < \lambda < 1$ be a rational number and let Δ be a positive integer. Define

Name $\textsc{AntiFerroIsing}_\lambda(\Delta)$.

Instance *A graph G of maximum degree at most Δ.*

Output *The partition function of the antiferromagnetic Ising model with parameter λ evaluated on instance G, i.e., $Z_\lambda(G) = \sum_{\sigma:V\to\{\pm1\}} \prod_{\{u,v\}\in E(G)} \lambda^{\delta(\sigma(u),\sigma(v))}$, where $\delta(i,j)$ is 1 if $i = j$ and 0 otherwise.*

▶ **Lemma 6.** *The following problems are* #SAT*-equivalent: $\#K_2'$-$\textsc{Col}(\Delta)$ for any $\Delta \geq 6$, and $\textsc{AntiFerroIsing}_\lambda(\Delta)$ for any $\Delta \geq 3$ and $0 < \lambda < (\Delta-2)/\Delta$. The following problems are* #BIS*-equivalent: $\#P_4$-$\textsc{Col}(\Delta)$ for $\Delta \geq 6$ and* #1P1NSat*.*

3 #SAT-equivalence

The aim of this section is to establish the #SAT-equivalence parts of Theorems 1 and 2.

▶ **Lemma 7.** *Suppose that H is a connected undirected graph. If H is neither reflexive nor irreflexive then, for all $\Delta \geq 6$, $\#\textsc{List}$-H-$\textsc{Col}(\Delta)$ is* #SAT*-equivalent. Hence, $\#\textsc{List}$-H-$\textsc{Col}$ is* #SAT*-equivalent.*

Proof. Let $\Delta \geq 6$. Since H is connected, it must contain K_2' as an induced subgraph. So $\#K_2'$-$\textsc{Col}(\Delta)$ is AP-reducible to $\#\textsc{List}$-H-$\textsc{Col}(\Delta)$. By Lemma 6, $\#K_2'$-$\textsc{Col}(\Delta)$ is #SAT-equivalent. ◀

The gadgets that we use in our reductions in the upcoming lemmas are of a particularly simple kind, namely paths.[1] Let the vertex set of the L-vertex path be $\{1, 2, \ldots, L\}$, where the vertices are numbered according to their position on the path. The end vertices 1 and L are *terminals*, which make connections with the rest of the construction. For each vertex $1 \leq k \leq L$ there is a set of allowed colours S_k. We can describe a gadget by specifying L and specifying the sets $(S_1, S_2, \ldots, S_L)$. In our application, each set S_i has cardinality 2, and $S_1 = S_L$.

Fix a connected graph H (note that H may have loops). Our strategy for proving that $\#\textsc{List}$-H-$\textsc{Col}(\Delta)$ is #SAT-equivalent is to find a gadget $(\{i_1, j_1\}, \{i_2, j_2\}, \ldots, \{i_L, j_L\})$ such that

(i) the sequence $(i_1, \ldots, i_L)$ is a path in H, and likewise $(j_1, \ldots, j_L)$;
(ii) it is never the case that both $\{i_k, j_{k+1}\} \in E(H)$ and $\{j_k, i_{k+1}\} \in E(H)$; and
(iii) $i_1 = j_L$ and $j_1 = i_L$.

If we achieve these conditions then, as we shall see, the colours at the terminals will be negatively correlated, and from there we will be able to encode instances of $\textsc{AntiFerroIsing}_\lambda(\Delta)$ for some integer $\Delta \geq 3$ and $\lambda \in (0, \frac{\Delta-2}{\Delta})$, and this is #SAT-equivalent (Lemma 6). Note that although the ordering of elements within the sets S_i is irrelevant to the workings of the gadget, we write the pairs in a specific order to bring out the path structure that we have just described.

Fix H and let $A = A_H$ be the adjacency matrix of H. Denote by $A_{(i,j),(i',j')}$ the 2×2 submatrix of A indexed by rows i and j and columns i' and j'. We regard the indices in the notation $A_{(i,j),(i',j')}$ as ordered; thus the first row of this 2×2 matrix comes from row i of A and the second from row j.

[1] We were also able to make use of path gadgets in [11], though, as noted (see Section 1) the results unfortunately do not carry over to our setting. Here the use of structural graph theory makes the discovery of such gadgets pleasingly straightforward.

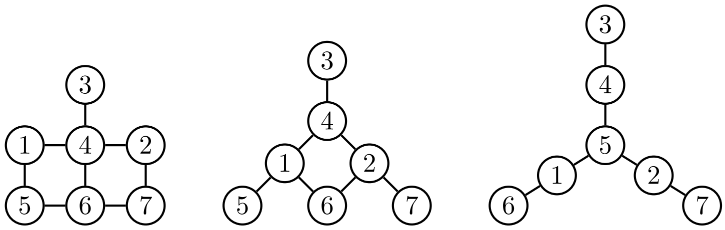

Figure 2 X_3, X_2 and T_2.

Given a gadget, i.e., sequence $(\{i_1, j_1\}, \{i_2, j_2\}, \ldots, \{i_L, j_L\})$, consider the product of 2×2 submatrices of A:

$$D' = A_{(i_1,j_1),(i_2,j_2)} A_{(i_2,j_2),(i_3,j_3)} \cdots A_{(i_{L-1},j_{L-1}),(i_L,j_L)}. \tag{1}$$

If conditions (i)–(iii) for gadget construction are satisfied then each of the 2×2 matrices in the product has 1s on the diagonal; also, all of them have at least one off-diagonal entry that is 0. Thus, each matrix has determinant 1, from which it follows that $\det D' = 1$.

Now consider the matrix D that is obtained by swapping the two columns of D'. This swap rectifies the "twist" that occurs in the passage from (i_1, j_1) to $(i_L, j_L) = (j_1, i_1)$, but it also flips the sign of the determinant, leaving $\det D = -1$. Let $r = i_1 = j_L$ and $s = j_1 = i_L$. The matrix D can be interpreted as giving the number of list H-colourings of the gadget when the k'th vertex of the gadget (for $k \in \{1, \ldots, L\}$ is assigned the list $\{i_k, j_k\}$, so the terminals are restricted to colours in $\{r, s\}$. Thus the entry in the first row and column of D is the number of colourings with both terminals receiving colour r, the entry in the first row and second column is the number of colourings with terminal 1 receiving colour r and terminal L receiving colour s, the entry in the second row and first column is the number of colourings with terminal 1 receiving colour s and terminal L receiving colour r and finally the entry in the second row and second column is the number of colourings with both terminals receiving colour s. We call $D = D(\Gamma)$ the *interaction matrix* associated with the gadget Γ. Since $\det D < 0$ the gadget provides a negative correlation between the colours at the terminals, which, as we will see, will allow a reduction from $\textsc{AntiFerroIsing}_\lambda(\Delta)$.

In the full version, Lemma 8 first applies the technique to get the #SAT-equivalences in the unbounded-degree case. For these arguments, we intentionally keep the construction of the gadgets as simple as possible. While this would also lead to a value of Δ such that $\#\textsc{List}\text{-}H\text{-}\textsc{Col}(\Delta)$ is #SAT-equivalent, the value of Δ would be much larger than 6. Nevertheless, we are able to refine the constructions to obtain the bounded-degree results of Theorem 2. Lemma 9 here combines Lemmas 8 and 9 of the full version and illustrates the key ideas.

▶ **Lemma 9.** *Suppose that H is a connected undirected graph. If H is irreflexive but it is not a bipartite permutation graph, then for all $\Delta \geq 3$, $\#\textsc{List}\text{-}H\text{-}\textsc{Col}(\Delta)$ is #SAT-equivalent (so $\#\textsc{List}\text{-}H\text{-}\textsc{Col}$ is also #SAT-equivalent).*

Proof (One Case). Graphs that are not bipartite permutation graphs contain one of the following as an induced subgraph: X_3, X_2, T_2, or a cycle of length other than 4. (Refer to Figure 2.) Here we present the case X_3. The remaining cases are similar and can be found in the full version.

We first show that the unbounded problem $\#\textsc{List}\text{-}X_3\text{-}\textsc{Col}$ is #SAT-equivalent. The gadget in this case is $\Gamma = \bigl(\{1, 2\}, \{4, 7\}, \{3, 6\}, \{4, 5\}, \{2, 1\}\bigr)$. The conditions (i)–(iii) for gadget construction are easy to check. Explicit calculation using (1) yields

$$D' = A_{(1,2),(4,7)}A_{(4,7),(3,6)}A_{(3,6),(4,5)}A_{(4,5),(2,1)} = \left(\begin{smallmatrix}1&0\\1&1\end{smallmatrix}\right)\left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right)\left(\begin{smallmatrix}1&0\\1&1\end{smallmatrix}\right)\left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right) = \left(\begin{smallmatrix}2&3\\3&5\end{smallmatrix}\right).$$

Swapping the columns of D' yields the interaction matrix $D = \left(\begin{smallmatrix}3&2\\5&3\end{smallmatrix}\right)$. As we explained earlier, $\det D = -1$. Obtaining a matrix D with negative determinant is moving in the right direction, but in order to encode antiferromagnetic Ising we ideally want the matrix $D = (D_{i,j})$ to also satisfy $D_{1,1} = D_{2,2}$ and $D_{1,2} = D_{2,1}$. Observe that the graph X_3 has an automorphism of order two, $\pi = (1,2)(5,7)$, that transposes vertices 1 and 2, which are the terminals of the gadget Γ. Consider the gadget obtained from Γ by letting π act on the colour sets, namely

$$\begin{aligned}\Gamma^\pi &= \big(\{\pi(1),\pi(2)\},\{\pi(4),\pi(7)\},\{\pi(3),\pi(6)\},\{\pi(4),\pi(5)\},\{\pi(2),\pi(1)\}\big)\\ &= \big(\{2,1\},\{4,5\},\{3,6\},\{4,7\},\{1,2\}\big),\end{aligned}$$

The interaction matrix $D^\pi = \left(\begin{smallmatrix}3&5\\2&3\end{smallmatrix}\right)$ corresponding to Γ^π is the same as D, except that the rows and columns are swapped. Placing Γ and Γ^π in parallel, identifying the terminals, yields a composite gadget Γ^* whose interaction matrix is $D^* = \left(\begin{smallmatrix}D_{1,1}D_{2,2} & D_{1,2}D_{2,1}\\ D_{2,1}D_{1,2} & D_{2,2}D_{1,1}\end{smallmatrix}\right) = \left(\begin{smallmatrix}9&10\\10&9\end{smallmatrix}\right)$. Note that the gadget Γ^* has maximum degree 2 (this observation will be important for the bounded-degree case). Also, $\det D^* = D_{1,1}^2D_{2,2}^2 - D_{1,2}^2D_{2,1}^2 = (D_{1,1}D_{2,2}+D_{1,2}D_{2,1})\det D < 0$. So we have an AP-reduction from $\textsc{AntiFerroIsing}_\lambda$ with $\lambda = D_{1,1}D_{2,2}/(D_{1,2}D_{2,1})$ to #$\textsc{List}$-H-$\textsc{Col}$: given an instance G of $\textsc{AntiFerroIsing}_\lambda$, simply replace each edge $\{u,v\}$ of G with a copy of the gadget Γ^*, identifying the two terminals of Γ^* with the vertices u and v, respectively. (Since Γ^* is symmetric, it does not matter which is u and which is v.) The problem $\textsc{AntiFerroIsing}_\lambda$ is #SAT-equivalent by Lemma 6. So for this case ($H = X_3$), we have $\lambda = \frac{9}{10}$.

We next show that, for $\Delta \geq 3$, #$\textsc{List}$-X_3-$\textsc{Col}(\Delta)$ is #SAT-equivalent. The smallest Δ such that $\textsc{AntiFerroIsing}_{\frac{9}{10}}(\Delta)$ is #SAT-equivalent is $\Delta = 21$, so the argument above would only give that #$\textsc{List}$-X_3-$\textsc{Col}(\Delta)$ is #SAT-equivalent for $\Delta \geq 21$ (in fact, $\Delta \geq 2\cdot 21 = 42$, since the terminals of Γ^* have degree 2). To improve this, we will implement thickenings of the gadget Γ^* using carefully chosen list colourings to keep the degree of the gadget small. More precisely, for integer $t \geq 0$, we will construct inductively gadgets Γ_t^* such that:

(i) The allowed colours of the terminals of Γ_t^* will be $\{1,2\}$ for odd t and $\{5,7\}$ for even t.

(ii) The two terminals of Γ_t^* will each have degree 1, and all other vertices of Γ_t^* will have degree at most 3.

(iii) The interaction matrix of Γ_t^* will be $D_t^* = \begin{pmatrix} 9^{2^t} & 10^{2^t}\\ 10^{2^t} & 9^{2^t}\end{pmatrix}$.

By taking t sufficiently large, the reduction above, using Γ_t^* as gadget instead of Γ^*, yields that #$\textsc{List}$-X_3-$\textsc{Col}(\Delta)$ is #SAT-equivalent for $\Delta \geq 3$. It remains to build the gadgets Γ_t^*.

Γ_0^* is obtained from Γ^* by connecting each terminal of Γ^* to a new vertex whose allowed set of colours is $\{5,7\}$ (recall that the allowed colours of the terminals of Γ^* are in $\{1,2\}$). The terminals of Γ_0^* are the two new vertices. Γ_0^* clearly satisfies properties (i) and (ii) for $t = 0$. To find the interaction matrix of Γ_0^*, note that colour 1 is adjacent to colour 5 in X_3 but not to colour 7. Similarly, colour 2 is adjacent to colour 7 in X_3 but not to colour 5. Thus, $D(\Gamma_0^*) = \left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)\left(\begin{smallmatrix}9&10\\10&9\end{smallmatrix}\right)\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right) = D_0^*$, proving that Γ_0^* satisfies all properties (i)–(iii), as desired.

To construct Γ_{t+1}^* from Γ_t^*, take two copies of Γ_t^*, place them in parallel, identifying their terminals in the natural way. Now, analogously to the construction of Γ_0^*, connect each (doubled-up) terminal of Γ_t^* to a new vertex whose allowed set of colours is $\{1,2\}$ if t is even and $\{5,7\}$ if t is odd. The final graph is the gadget Γ_{t+1}^* and the new vertices

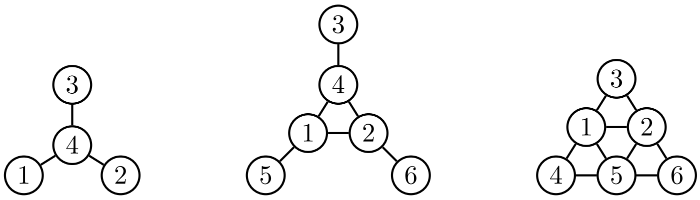

Figure 3 The claw, the net and S_3.

introduced in the second step of the construction are its terminals. It is clear that Γ^*_{t+1} satisfies property (i) and, using the fact that Γ^*_t satisfies property (ii), we have that Γ^*_{t+1} satisfies property (ii) as well. Arguing analogously as for Γ^*_0, the interaction matrix of Γ^*_{t+1} is given by $D(\Gamma^*_{t+1}) = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) \begin{pmatrix} (9^{2^t})^2 & (10^{2^t})^2 \\ (10^{2^t})^2 & (9^{2^t})^2 \end{pmatrix} \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) = D^*_{t+1}$, where the squared entries account for the 2-thickening of the gadget Γ^*_t. This proves that Γ^*_{t+1} has property (iii), concluding the proof for $H = X_3$. The other cases are similar, and are given in the full version (where more of the general principles are explained). ◀

The remaining cases of parts (iii) of Theorems 1 and 2 are covered by Lemmas 10 and 11 of the full version, which we combine here.

▶ **Lemma 11.** *Suppose that H is a connected undirected graph. If H is a reflexive graph that is not a proper interval graph, then, for $\Delta \geq 3$, #*List*-H-*Col*(Δ) is #SAT-equivalent (so #*List*-H-*Col* is #SAT-equivalent).*

Proof Sketch. The line of argument is similar to those used in Lemma 9. Graphs that are not proper interval graphs contain one of the following as an induced subgraph: the claw, the net, S_3, or a cycle of length at least four. (Refer to Figure 3 but note that loops are omitted.) We show that #List-H-Col is #SAT-equivalent when H is any of these (and the bounded-degree analogue). The details can be found in the full version. ◀

4 #BIS-equivalence

We now deal with the #BIS-equivalent cases in Theorems 1 and 2.

▶ **Lemma 13.** *Suppose that H is a connected undirected graph. If H is not a reflexive complete graph or an irreflexive complete bipartite graph then, for all $\Delta \geq 6$, #*List*-H-*Col*(Δ) is #BIS-hard. Hence, #*List*-H-*Col* is #BIS-hard.*

Proof Sketch. In the full version, we show that any graph covered by the lemma contains one of the following as an induced subgraph: K_2', P_3^*, P_4 or an odd cycle. These are (at least) #BIS-hard: by Lemma 6 for K_2' and P_4, by Lemma 12 of the full version for P_3^*, and by Lemma 9 for an odd cycle. ◀

▶ **Lemma 14.** *Suppose that H is a connected undirected graph. If H is an irreflexive bipartite permutation graph or a reflexive proper interval graph, then #*List*-H-*Col* is #BIS-easy.*

Proof Sketch. The reduction is done in a more general weighted setting by Chen, Dyer, Goldberg, Jerrum, Lu, McQuillan and Richerby [1]: see the proofs of Lemmas 45 and 46 of that article. However, in the current context, we can simplify the reduction significantly (eliminating the need for multimorphisms and other concepts from universal algebra), and we can also extract (see the full paper) the slightly stronger statement that #List-H-Col is in #RHΠ$_1$. The target problem for our reduction is #1P1NSAT which is #BIS-equivalent by Lemma 6.

We will treat the case where H is an irreflexive bipartite permutation graph. The other case is similar (as explained in the full version). Without loss of generality, suppose that H is connected and that its biadjacency matrix B has q_1 rows and q_2 columns and is in staircase form. Let A be the adjacency matrix $\left(\begin{smallmatrix} B & 0 \\ 0 & B^T \end{smallmatrix}\right)$, which is formally defined as follows.

$$A_{i,j} = \begin{cases} B_{i,j}, & \text{if } 1 \le i \le q_1,\ 1 \le j \le q_2 \\ B_{j-q_2, i-q_1}, & \text{if } q_1+1 \le i \le q_1+q_2,\ q_2+1 \le j \le q_2+q_1 \\ 0, & \text{otherwise.} \end{cases}$$

Let $q = q_1 + q_2$. For each $i \in \{1, \ldots, q\}$, let $\alpha_i = \min\{j : A_{i,j} = 1\}$ and let $\beta_i = \max\{j : A_{i,j} = 1\}$. Since B is in staircase form, so is A, so the sequences (α_i) and (β_i) are non-decreasing. Let $r_1, \ldots, r_q$ be the colours associated with the rows of A and $c_1, \ldots, c_q$ be the colours associated with the columns of A, in order. Note that $\{r_1, \ldots, r_q\}$ and $\{c_1, \ldots, c_q\}$ are different permutations of the vertices of H,

Suppose that $(G, \mathbf{S})$ is an instance of #LIST-H-COL. Assume without loss of generality that G is bipartite. Otherwise, it has no H-colourings. Let $V_1(G) \cup V_2(G)$ be the bipartition of $V(G)$. We will construct an instance Ψ of #1P1NSAT such that the number of satisfying assignments to Ψ is equal to the number of list H-colourings of $(G, \mathbf{S})$.

The variable set of Ψ is $\mathbf{x} = \{x_i^u : u \in V(G) \text{ and } 0 \le i \le q\}$. For each vertex $u \in V(G)$ we introduce the clauses (x_0^u) and $(\neg x_q^u)$. Also, for each $j \in \{1, \ldots, q\}$ we introduce the clause $\mathrm{IMP}(x_j^u, x_{j-1}^u)$. Denote by $\Psi_V(\mathbf{x})$ the formula obtained by taking the conjunction of all these clauses.

We will interpret the assignment to the variables in $\mathbf{x}$ as an assignment σ of colours to the vertices of G according to the following rule. If $u \in V_1(G)$ then $x_i^u = 1$ if and only if $\sigma(u) = r_j$ for some $j > i$. If $u \in V_2(G)$ then $x_i^u = 1$ if and only if $\sigma(u) = c_j$ for some $j > i$. Note that there is a one-to-one correspondence between assignments to $\mathbf{x}$ that satisfy the clauses in $\Psi_V(\mathbf{x})$ and assignments σ of colours to the vertices of G.

We now introduce further clauses to enforce the constraint on colours received by adjacent vertices. For each edge $\{u, v\} \in E(G)$ with $u \in V_1(G)$ and $v \in V_2(G)$, and for each $i \in \{1, \ldots, q\}$, we add the clauses $\mathrm{IMP}(x_{i-1}^u, x_{\alpha_i-1}^v)$ and $\mathrm{IMP}(x_{\beta_i}^v, x_i^u)$. Denote by $\Psi_E(\mathbf{x})$ the formula obtained by taking the conjunction of all of these clauses.

We next argue that there is a bijection between H-colourings of G and satisfying assignments to $\Psi_V(\mathbf{x}) \wedge \Psi_E(\mathbf{x})$. In one direction, suppose σ is an H-colouring of G. We wish to show that all clauses in $\Psi_E(\mathbf{x})$ are satisfied. Consider an edge $\{u, v\} \in E(G)$ with $u \in V_1(G)$ and $v \in V_2(G)$ and the corresponding clause $\mathrm{IMP}(x_{i-1}^u, x_{\alpha_i-1}^v)$. The clause is satisfied unless $x_{i-1}^u = 1$, so suppose $x_{i-1}^u = 1$. Then by the interpretation of assignments, $\sigma(u) = r_j$ for some $j \ge i$. Since σ is an H-colouring, this implies that $\sigma(v) = c_k$ for some $k \ge \alpha_i$. But by the interpretation of assignments, this means that $x_{\alpha_i-1}^v = 1$, so the clause is satisfied. The argument for the other clause $\mathrm{IMP}(x_{\beta_i}^v, x_i^u)$ coresponding to the edge $\{u, v\}$ is similar – see the full version.

In the other direction, suppose $\Psi_V(\mathbf{x}) \wedge \Psi_E(\mathbf{x})$ is satisfied. Consider an edge $\{u, v\} \in E(G)$ with $u \in V_1(G)$ and $v \in V_2(G)$ and suppose that $\sigma(u) = r_i$. In the corresponding assignment $x_{i-1}^u = 1$ so by the clause $\mathrm{IMP}(x_{i-1}^u, x_{\alpha_i-1}^v)$ we have $x_{\alpha_i-1}^v = 1$ so $\sigma(v) = c_k$ for some $k \ge \alpha_i$. In the corresponding assignment $x_i^u = 0$ so by the clause $\mathrm{IMP}(x_{\beta_i}^v, x_i^u)$, $x_{\beta_i}^v = 0$, so $\sigma(v) = c_k$ for some $k \le \beta_i$. We conclude that the colours $\sigma(u)$ and $\sigma(v)$ are adjacent in H. This holds for every edge, so σ is an H-colouring of G.

Finally, we add clauses to deal with lists. A colour assignment $\sigma(u) = r_i$ with $u \in V_1(G)$ is uniquely characterised by $x^u_{i-1} = 1$ and $x^u_i = 0$. So we can eliminate the possibility of $\sigma(u) = r_i$ by introducing the clause $\mathrm{IMP}(x^u_{i-1}, x^u_i)$. A similar clause will forbid a vertex $v \in V_2(G)$ to receive colour c_j. Let $\Psi_L(\mathbf{x})$ be the conjunction of all such clauses, arising from the lists in $\mathbf{S}$. Let $\Psi(\mathbf{x}) = \Psi_V(\mathbf{x}) \wedge \Psi_E(\mathbf{x}) \wedge \Psi_L(\mathbf{x})$. Then the list H-colourings of $(G, \mathbf{S})$ are in bijection with the satisfying assignments to $\Psi(\mathbf{x})$, as required ◀

5 A counterexample

The situation that we have studied in this paper is characterised by having hard interactions between pairs of adjacent spins (a pair is either allowed or it is disallowed) and hard constraints on individual spins (again, a spin is either allowed at a particular vertex or it is disallowed). Our results apply both in the degree-bounded case and in the unbounded-degree case. In the unbounded case, earlier work treated the situation with weighted interactions and weighted spins. The characterisations derived in these weighted scenarios (see, e.g. [11, Thm 1]) have a similar feel to the trichotomy that we have presented in Theorem 1. We may wonder whether, in the unbounded case, at least, there is a common generalisation. That is, in the unbounded case, does the trichotomy of [11] survive if weights on spins are replaced by lists? The answer is no. There are examples of weighted spin systems with just $q = 2$ spins whose partition function is #SAT-hard to approximate with vertex weights but efficiently approximable (in the sense that there is an FPRAS) with lists instead of weights.

Here is one such example. Following Li, Lu and Yin [16], define the interaction matrix $A = (a_{ij} : 0 \le i, j \le 1)$ by $A = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 2 \end{smallmatrix}\right)$, and the partition function associated with an instance G by $Z_A(G) = \sum_{\sigma: V(G) \to \{0,1\}} \prod_{\{u,v\} \in E(G)} a_{\sigma(u),\sigma(v)}$. This is the partition function of a variant of the independent set model, which instead of defining the interaction between spin 1 and itself (two vertices that are out of the independent set) to be 1, defines this interaction weight to be 2.

Li, Lu and Yin [16, Theorem 21] show that Weitz's self-avoiding walk algorithm [22] gives an FPTAS for $Z_A(G)$. Also, Weitz's correlation decay algorithm [22] can accommodate lists. Indeed, the construction of the self-avoiding walk tree relies on being able to "pin" colours at individual vertices. So the partition function remains easy to approximate (in the sense that there is an FPTAS) even in the presence of lists. In contrast, the approximation problem becomes #SAT-hard if arbitrary weights are allowed. Indeed, by weighting spin 0 at each vertex $u \in V(G)$ by $2^{d(u)}$, where $d(u)$ is the degree of u, we recover the usual independent set partition function, which is #SAT-equivalent (Lemma 6). (The same fact can be read off from general results in many papers, including [11, Thm 1].) Thus, even in the unbounded case, the dichotomies presented in [11, Thm 1] and [1, Thm 6] do not hold with lists in place of weights. So even in the unbounded-degree case, it was necessary to explicitly analyse list homomorphisms in order to derive precise characterisations quantifying the problem of approximately counting these.

References

1 Xi Chen, Martin Dyer, Leslie Ann Goldberg, Mark Jerrum, Pinyan Lu, Colin McQuillan, and David Richerby. The complexity of approximating conservative counting CSPs. *J. Comput. System Sci.*, 81(1):311–329, 2015.

2 Martin Dyer, Ann Leslie Goldberg, Catherine Greenhill, and Mark Jerrum. The relative complexity of approximate counting problems. *Algorithmica*, 38(3):471–500, 2003.

3 Martin Dyer and Catherine Greenhill. The complexity of counting graph homomorphisms. *Random Structures Algorithms*, 17(3-4):260–289, 2000.

4 Tomas Feder and Pavol Hell. List homomorphisms to reflexive graphs. *Journal of Combinatorial Theory, Series B*, 72(2):236 – 250, 1998.

5 Tomás Feder, Pavol Hell, and Jing Huang. List homomorphisms and circular arc graphs. *Combinatorica*, 19(4):487–505, 1999.

6 Tomas Feder, Pavol Hell, and Jing Huang. Bi-arc graphs and the complexity of list homomorphisms. *J. Graph Theory*, 42(1):61–80, 2003.

7 Andreas Galanis, Leslie Ann Goldberg, and Mark Jerrum. Approximately counting H-colourings is #BIS-hard. *SIAM Journal on Computing (to appear)*, 2016.

8 Andreas Galanis, Daniel Štefankovič, and Eric Vigoda. Inapproximability of the partition function for the antiferromagnetic Ising and hard-core models. *Combinatorics, Probability and Computing (to appear)*, 2016.

9 T. Gallai. Transitiv orientierbare Graphen. *Acta Math. Acad. Sci. Hungar*, 18:25–66, 1967.

10 Tibor Gallai. A translation of T. Gallai's paper: "Transitiv orientierbare Graphen" [Acta Math. Acad. Sci. Hungar. **18** (1967), 25–66; MR0221974 (36 #5026)]. In *Perfect graphs*, Wiley-Intersci. Ser. Discrete Math. Optim., pages 25–66. Wiley, Chichester, 2001. Translated from the German and with a foreword by Frédéric Maffray and Myriam Preissmann.

11 Leslie Ann Goldberg and Mark Jerrum. A complexity classification of spin systems with an external field. *Proceedings of the National Academy of Sciences*, 112(43):13161–13166, 2015.

12 Pavol Hell and Jing Huang. Interval bigraphs and circular arc graphs. *J. Graph Theory*, 46(4):313–327, 2004.

13 Zygmunt Jackowski. A new characterization of proper interval graphs. *Discrete Math.*, 105(1-3):103–109, 1992.

14 Steven Kelk. *On the relative complexity of approximately counting H-colourings.* PhD thesis, Warwick University, 2003.

15 Ekkehard G. Koehler. *Graphs without asteroidal triples.* PhD thesis, Technische Universität Berlin, 1999.

16 Liang Li, Pinyan Lu, and Yitong Yin. Approximate counting via correlation decay in spin systems. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 922–940, 2012.

17 F.S Roberts. *Representations of indifference relations.* PhD thesis, Stanford University, Stanford, CA, 1968.

18 Allan Sly. Computational transition at the uniqueness threshold. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 287–296, 2010.

19 Jeremy Spinrad, Andreas Brandstädt, and Lorna Stewart. Bipartite permutation graphs. *Discrete Appl. Math.*, 18(3):279–292, 1987.

20 Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theor. Comput. Sci.*, 47(3):85–93, 1986.

21 Gerd Wegner. *Eigenschaften der Nerven homologisch-einfacher Familien im $\mathbb{R}^n$.* PhD thesis, Universität Göttingen, Göttingen, Germany, 1967.

22 Dror Weitz. Counting independent sets up to the tree threshold. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 140–149, 2006.