Efficient Lifting of MAP LP Relaxations Using k-Locality

Martin Mladenov
TU Dortmund University
{fn.ln}@cs.tu-dortmund.de

Amir Globerson
Hebrew University
amir.globerson@mail.huji.ac.il

Kristian Kersting
TU Dortmund University
{fn.ln}@cs.tu-dortmund.de

Abstract

Inference in large scale graphical models is an important task in many domains, and in particular for probabilistic relational models (e.g,. Markov logic networks). Such models often exhibit considerable symmetry, and it is a challenge to devise algorithms that exploit this symmetry to speed up inference. Here we address this task in the context of the MAP inference problem and its linear programming relaxations. We show that symmetry in these problems can be discovered using an elegant algorithm known as the k-dimensional Weisfeiler-Lehman (k-WL) algorithm. We run k-WL on the original graphical model, and not on the far larger graph of the linear program (LP) as proposed in earlier work in the field. Furthermore, the algorithm is polynomial and thus far more practical than other previous approaches which rely on orbit partitions that are GI complete to find. The fact that k-WL can be used in this manner follows from the recently introduced notion of k-local LPs and their relation to Sherali Adams relaxations of graph automorphisms. Finally, for relational models such as Markov logic networks, the benefits of our approach are even more dramatic, as we can discover symmetries in the original domain graph, as opposed to running lifting on the much larger grounded model.

1 Introduction

Many problems in probabilistic modeling and inference exhibit symmetries. In other words, they consist of certain building blocks that are reused within the model. It has been suggested that such symmetries may be exploited to speed up inference in these models. Indeed, this has motivated an active field of research known as lifted probabilistic inference (e.g., see [1] and references therein). Lifting approaches are especially important and hold considerable promise in the context of relational probabilistic models (RPMs) [2, 3, 4]. These tackle a long standing goal of AI, namely unifying first-order logic (capturing regularities and symmetries) and probability (capturing uncertainty). RPMs often encode large, complex models using only a few logical rules applied to a large class of objects. Thus symmetries and redundancies are likely to abound. Non-lifted inference in these models requires operating on a mostly propositional representation level and does not exploit the additional symmetries. In contrast, lifted inference makes use of symmetries, which have been shown to appear in AI tasks such as citation matchings, information broadcasting, market analysis, tracking of objects, and biomolecular event prediction, among others (e.g., see [5, 6, 7, 8, 9, 10, 11]).

A sensible scheme in devising lifted inference algorithms is to begin with an approximate inference[1] algorithm that provides good performance on the original (non-lifted) model, and then seek an equivalent and faster version of the algorithm, which exploits symmetries. This methodology has been applied successfully to the loopy belief propagation (LBP) algorithm [12, 13, 14, 15]. These works identify variables in the model that send and receive identical messages due to symmetries of the factor graph, and calculate these messages only once, resulting in often considerable savings. However, the downside of this approach is that LBP is theoretically poorly understood and does not always perform well. Another approximate inference approach that has been very successful lately is linear programming (LP) relaxations of the MAP problem (e.g., see [16, 17]). These approximate the MAP problem as an LP with polynomially many constraints, which is therefore tractable. LP relaxations

---

[1]We address approximate inference here, since exact inference is intractable in most of the models of interest.

have several nice properties. First, they result in an upper bound on the MAP value, and can thus be used within branch and bound methods. Second, they provide certificates of optimality, so that one knows when the problem has been solved exactly.[2] Third, the LP can be solved using simple algorithms such as coordinate descent, many of which have a nice message passing structure [18, 19]. Fourth, the LP relaxations can be progressively tightened by adding constraints of a higher order. For example, a kth order MAP-LP will involve constraints that consider k tuples of variables. This is also known as the Sherali-Adams (SA) hierarchy of relaxations. It has been shown that such tightenings can be done gradually, and as a result solve challenging MAP problems [16, 20, 21].

Given the above, it is a natural step to try to lift MAP LP relaxations, and their tightenings. Thus far, two approaches have been proposed to this problem. The first is to identify symmetries using the automorphism group of the graph underlying the model [22, 23, 24]. Unfortunately, this requires calculating a quotient of the graph with respect to its orbit partition and computing the orbit partition is GI-complete (polynomial-time equivalent to graph isomorphism and thus generally intractable). On the other hand, Mladenov *et al.* [10] recently proposed a generic approach to lifting linear programs that takes an LP as input and returns a possibly smaller version of this LP. In principle, this lifted linear programming method can indeed be applied to MAP-LP relaxations by forming the MAP-LP and then applying the method of [10]. However, the MAP-LP contains considerable structure, which [10] is oblivious to. Indeed, the method we propose here offers a faster lifting of MAP-LPs precisely due to this.

Our contribution here is thus to provide a scheme for lifting MAP LP problems of any order. Our approach is tractable, sound, and uses the special structure of these LPs. Specifically, we propose a method that achieves this by using an elegant algorithm known as k^{th} order Weisfeiler-Lehman (k-WL), originally developed for approximating the graph automorphism problem [25]. Our approach works as follows: given a graph G and an order k, we run k-WL on the graph G to identify indistinguishable k^{th} order structures in the graph. We next use these to construct a compact version of various MAP-LP relaxations. We stress that the symmetries are identified on the original graph and not on the MAP-LP itself, which is considerably larger for higher order relaxations. This is in stark contrast to the approach of [10], which would look for symmetries on the MAP-LP. As we will show, the choice of k depends on the structure of the relaxed MAP LP

problem. In particular, we build upon a recent structural characteriztion of general LPs, called k-locality, introduced by Atserias and Maneva [26]. Intuitively, an LP is called k-local if its variables and constraints correspond to k-tuples of vertices of the graph underlying the LP. Atserias and Maneva have shown that k-WL can be used to identify certain invariances of k-local LPs. Here, we show that the invariances can be viewed as symmetries and exploited to speed up MAP LP inference. For example, in Section 3.2, we show that the kth order tightening of a MAP LP is $k + 1$-local and in turn efficiently liftable. In fact, we prove a more general result, namely that any k-local LP can be lifted efficiently using k-WL (see Section 3.1). For the case of RPMs such as Markov Logic Networks (MLNs), k-locality extends even further. As we will show in Section 3.3, MAP LP relaxations of MLNs exhibit additional locality, which can be detected using the appropriate version of the WL algorithm. Finally, before concluding, we present an empirical illustration of our results demonstrating that they can yield significant computational savings.

We would like to stress that our scheme not only results in tractable lifted versions of an important class of inference algorithms. It also unifies the two seemingly disparate works of [10] and [24]. Briefly, [10] uses 1-WL which is adequate for first order LPs, whereas [24] uses exact automorphism (equivalent to full order WL) which is adequate for exact MAP (and unnecessary for lower order LPs).

2   Background

We start off by introducing MAP problems, their LP relaxations, and other relevant background.

2.1   MAP Inference in MRFs

Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ be a set of n discrete-valued random variables and let x_i represent the possible realizations of random variable X_i. Markov random fields (MRFs) compactly represent a joint distribution over $\mathbf{X}$ by assuming that it is obtained as a product of functions defined on small subsets of variables [27]. For simplicity, we will restrict our discussion to a specific subset of MRFs, namely Ising models with arbitrary topology.[3] In an Ising model on a graph $G = (V, E)$, all variables are binary, i.e., $\mathbf{X}_i \in \{0, 1\}$. The model is then given by: $p(x) \propto \exp\left[\sum_{ij \in E} \theta_{ij} x_i x_j + \sum_i \theta_i x_i\right]$. The Maxi-

---

[2]Of course this is not always the case, due to the intractability of MAP in the general case.

[3]The one-to-one correspondence between the parameters of the Ising model and the graph elements simplifies our arguments, sparing us some auxiliary constructions. This restriction does not come with a loss of generality.

mum a-posteriori (MAP) inference problem is defined as finding an assignment maximizing $p(x)$. This can equivalently be formulated as the following LP $\boldsymbol{\mu}^* =$

$$\text{argmax}_{\boldsymbol{\mu} \in \mathcal{M}(G)} \sum_{ij \in E} \mu_{ij} \theta_{ij} + \sum_i \mu_i \theta_i = \boldsymbol{\theta} \cdot \boldsymbol{\mu} \quad (1)$$

where the set $\mathcal{M}(G)$ is known as the marginal polytope [17]. Even though Eq. 1 is an LP, the polytope $\mathcal{M}(G)$ generally requires an exponential number of inequalities to describe, and is NP-complete to maximize over. Hence one typically considers tractable relaxations (outer bounds) of $\mathcal{M}(G)$.

We next describe such outer bounds. The outer bounds we consider are equivalent to the standard local consistency bounds typically considered in the literature (e.g., see [17] Equation 8.32). However, we present them in a slightly different manner, which simplifies our presentation. Define the following set of vectors in $[0,1]^{|V|+|E|}$:

$$P = \left\{ \boldsymbol{\mu} \; \middle| \; \begin{array}{l} 0 \leq \mu_{ij}, \mu_i \leq 1 \; ; \mu_{ij} \leq \mu_j \; ; \\ \mu_{ij} \leq \mu_i ; \mu_i + \mu_j - \mu_{ij} \leq 1 \end{array} \right\}. \quad (2)$$

The vectors with $\{0,1\}$ coordinates in this set are the vertices of the marginal polytope. In other words $\mathcal{M}(G)$ is the convex hull of $P \cap \{0,1\}^{|V|+|E|}$. A standard way of outer bounding such polytopes is via the Sherali-Adams (SA) hierarchy. These are a set of polytopes $P = P^1 \supseteq P^2 \supseteq \ldots \supseteq P^n = \mathcal{M}(G)$ which are progressively tighter bounds on $\mathcal{M}(G)$. The hierarchy is obtained recursively as follows. The variables of P^k correspond to k tuples of V. For instance for $k = 3$ we have variables such as $\mu_{1,2,5}$. To get from P^k to P^{k+1} we multiply all inequalities in P^k by $\mu_1, \ldots, \mu_n, (1 - \mu_1), \ldots, (1 - \mu_n)$ and linearize. In the above example, we will get inequalities with expressions such as $\mu_{1,2,5}\mu_6$. This will be linearized to a new variable $\mu_{1,2,5,6}$. In case of repeating indices like $\mu_{1,2,5}\mu_5$, double indices will be eliminated resulting in $\mu_{1,2,5}$. Thus, the kth level SA relaxation consists of $O(n^k)$ constraints and variables. It has been shown that even the $k = 2$ case results in significant gains, and can be used to solve seemingly hard problems such as protein design [20].

2.2 Graph Automorphisms and Color-passing

As mentioned earlier, lifting involves identifying sets of indistinguishable variables in the LP. Since the LPs we consider are closely related to graphs, it's not surprising that graph automorphisms have recently been identified as an important toolbox for obtaining liftings. We now briefly review relevant background.

Given a graph, its orbit partition (OP) is a partition of the vertices V into $S_1, \ldots, S_p$ such that for every

two vertices $i, j \in S_k$ there is an automorphism that maps i to j. In other words, vertices in S_i are indistinguishable. Indeed the lifting approach of [24] uses such partitions. However, as mentioned earlier, these are generally intractable to compute.

An alternative approach to identifying "similar" vertices is to use color passing procedures. The simplest of these is the "1-dimensional Weisfeiler-Lehman" (or 1-WL. Also known as color passing). It iteratively partitions, or colors, the vertices of a graph according to an iterated degree sequence: initially, all vertices get the same color. Next, at each step, two vertices get again the same color assigned if they have the same histograms of colors among their neighbors. That is, they get different colors if for some color c they have a different number of neighbors of color c. The iteration stops when the partition remains unchanged, i.e., it becomes stable. The resulting partition is known as the *coarsest equitable partition (CEP)* of the graph, which is not an orbit partition and in fact can generally be coarser than the orbit partition [28].

The 1-WL partition has two key advantages in the context of lifting. First, it can be calculated efficiently in time $O((|E| + |V|) \log(|V|))$[29]. Second, it can be used to lift generic LPs via the procedure proposed in [10]. However, 1-WL is not a natural approach to lifting MAP-LPs since it is oblivious to their structure.

In this paper we use a generalized version of the WL algorithm known as the k dimensional WL (or k-WL) [25]. The idea in k-WL is to use not just the immediate neighborhood in deciding whether two nodes are similar but higher order neighborhoods. As an example, if a graph is regular 1-WL will stop after one iteration with all vertices having the same color. On the other hand k-WL may find finer partitions as it considers higher order structure. As we show later k-WL is in fact the appropriate procedure for lifting MAP-LPs.

Finally, we note an elegant link between k-WL algorithms and orbit partitions. It has recently been shown [26] that the partitions found by k-WL are a solution to a certain k^{th} order fractional relaxation of the graph automorphism problem. Due to this, we refer to the symmetries found by k-WL as fractional symmetries. Furthermore, when $k = n$ the k-WL algorithm will find the exact orbit partition. Thus k-WL spans a spectrum of partitions, the coarsest one obtained with 1-WL and the finest with n-WL.

2.3 Local Linear Programs

Recall that our goal is to lift k-MAP-LPs. These are linear programs whose structure is derived from a graph G. Thus, one would expect that symmetries in the graph are closely related to those of the LP. It

turns out that there is a very elegant connection between the two, facilitated via the recently introduced notion of k-local LP, which we review below. For more details we refer to [26].

Many problems in combinatorial optimization take a graph as input and construct an LP from this graph. For example, maximum bipartite matching for a graph $G = (V, E)$ is a solution to the following problem:

$$\begin{aligned} \max \quad & \textstyle\sum_{ij \in E} z_{ij} \\ \text{s.t.} \quad & \textstyle\sum_i z_{ij} = 1 \;,\; \sum_j z_{ij} = 1 \;,\; \text{and } z_{ij} \geq 0 \;. \end{aligned} \quad (3)$$

The above essentially specifies a procedure that takes G as input and returns an LP. A key question is then when will different graphs result in similar LPs (e.g., in the sense that the solution of one can be converted to a solution of the other). A recent work by Atserias and Maneva [26] gives a elegant answer to this via the notion of k locality.

Definition 1 *Given a graph $G = (V, E)$ and integer K we say that a function $C : V \cup E \to \{1, \ldots, P\}$ is a coloring of G. We assume that there is no edge that shares the same color with a vertex. We then call $G = (V, E, C)$ a colored graph.*

Definition 2 *An LP generating procedure $\mathcal{F}(G)$ is a function from colored graphs $G = (V, E, C)$ to LPs $(\mathbf{A}, \boldsymbol{b}, \boldsymbol{c})$.[4]*

Next, we will be interested in LP generating procedures whose constraints are highly regular with respect to graph structures. To this end, we need additional definitions.

Definition 3 *Given a set of vertices V, we denote by $V^{\leq k}$ all the ordered subsets of V of size k or less, and by V^k all the ordered subsets of exactly size k (i.e., all k-tuples of V).*

Definition 4 *For a k-tuple $\mathbf{t}$ of vertices of $G, \mathbf{t} = (t_1, ..., t_k)$, we define $\text{etp}(\mathbf{t}) = \{(i, j) | t_i = t_j\}$ as the set of pairs of indices of $\mathbf{t}$ whose elements are equal. Furthermore, we define $\text{atp}(\mathbf{t}) = (I^E, I^1, ..., I^P)$, where $I^E = \{(i, j) | (t_i, t_j) \in E\}$ is the set of index pairs whose elements are adjacent in G and $I^p = \{i | C(t_i) = p\} \cup \{(i, j) | C(t_i, t_j) = p\}$. In other words I^p is the set of elements and pairs of elements in t that have the color p in the graph.*

Definition 5 *An LP generating procedure $\mathcal{F}(G)$ is basic k-local if: **(A)** It has one variable and one constraint for each k-tuple of V.[5] **(B)** There exist func-*

tions f, g, h *with values in $\mathbb{R}$ such that:*

$$\mathbf{A_{u,v}} = \begin{cases} f(\text{etp}(\mathbf{uv}), \text{atp}(\mathbf{uv})) & \text{if } |\mathbf{uv}| \leq k, \\ 0 & \text{otherwise}, \end{cases}$$

$\mathbf{b_u} = g(\text{etp}(\mathbf{u}), \text{atp}(\mathbf{u}))$ *, and $\mathbf{c_v} = h(\text{etp}(\mathbf{v}), \text{atp}(\mathbf{v}))$. Here $\mathbf{uv}$ is the concatenation of $\mathbf{u}$ and $\mathbf{v}$, and $|\mathbf{uv}|$ denotes the number of unique elements in $\mathbf{uv}$. A union of basic k-local procedures is called k-local.[6]*

Intuitively, what the above definition means is that a procedure is local if its coefficients matrix only depends on the intersection between the tuple corresponding to the constraint and the one corresponding to the variables. It however does not depend on the specific identity of these tuples. And this "anonymity" of k-tuples is what we would like to exploit within lifted inference. It is akin to the interchangeability of random variables within lifted inference, and the bounded size k ensures that these potential symmetries in the LP can be identified in a local fashion.

3 Lifting k-Local MAP LPs

We will now show how k-locality can be exploited for efficient lifting. To do so, we make use of the k-WL algorithm mentioned earlier. We will first recap k-WL and then show using recent results of Atserias and Maneva [26] how it can be used to compress LPs, whenever fractional symmetry is present.

3.1 Lifting k-local LPs by k-WL

Given a colored graph $G = (V, E, C)$ and $k > 1$, k-WL[7] iteratively assigns colors to k-tuples of V. Denote the color of $\mathbf{u} \in V^k$ at iteration r by $W^r(\mathbf{u})$. Then, initially two tuples $\mathbf{u} = (u_1, \ldots, u_k)$ and $\mathbf{v} = (v_1, \ldots, v_k)$ are assigned the same color, $W^0(\mathbf{u}) = W^0(\mathbf{v})$, iff they satisfy $\text{etp}(\mathbf{u}) = \text{etp}(\mathbf{v})$ and $\text{atp}(\mathbf{u}) = \text{atp}(\mathbf{v})$ (see Def. 4).[8] To compute the color in iteration $W^{r+1}(\mathbf{u})$, we define the operation for each $g \in V$ and $\mathbf{u} \in V^k$:

$$\text{sift}(f, \mathbf{u}, g) = (f(g, u_2, \ldots, u_k), f(u_1, g, \ldots, u_k), \ldots, \\ f(u_1, u_2, \ldots, g)) .$$

Then, $W^{r+1}(\mathbf{u}) = W^{r+1}(\mathbf{v})$ holds iff for every tuple of colors $\mathbf{t}$, $|\{g \in V | \text{sift}(W^r, \mathbf{u}, g) = \mathbf{t}\}| = |\{g' \in V | \text{sift}(W^r, \mathbf{v}, g') = \mathbf{t}\}|$. In other words, the color of a

---

[4]We use $(\mathbf{A}, \boldsymbol{b}, \boldsymbol{c})$ to denote the optimization problem of minimizing $\boldsymbol{c} \cdot \boldsymbol{x}$ subject to $A\boldsymbol{x} \leq \boldsymbol{b}$.

[5]Thus, if $k = 3$ and $|V| = 5$ the constraints and variables will be indexed by i_1, i_2, i_3 where $i_m \in \{1, \ldots, 5\}$.

[6]By a union of procedures we mean a concatenation of the constraints in each procedure.

[7][26] present a unified but more complex definition of WL for $k > 0$. For the sake of simplicity we stick to the one presented here and use 1-WL as presented above.

[8]In other words if: **(A)** $u_i = u_j \Leftrightarrow v_i = v_j$, **(B)** $u_i u_j \in E \Leftrightarrow v_i v_j \in E$, and **(C)** $C(u_i) = C(v_i)$ as well as $C(u_i u_j) = C(v_i v_j)$.

tuple in the next iteration is the *multiset*: $W^{r+1}(\mathbf{u}) = \langle \text{sift}(W^r, \mathbf{u}, g) | g \in V \rangle$. This is iterated until the coloring is stable, i.e. the partition induced by the colors does not refine anymore. See [25] for details.

We are now ready to describe our lifting approach. Assume k-WL has ended up with P colors and denote by $W_i \subset V^k$ the subset of tuples that got color i. Denote this partition of V^k by $\mathcal{W}$. Given $\mathcal{W}$ we define the characteristic matrix $\widehat{X} \in \mathbb{R}^{|V^k| \times P}$ as follows:

$$\widehat{X}_{\mathbf{u}i} = 1/\sqrt{|W_i|} \text{ if } \mathbf{u} \in W_i \text{ and } \widehat{X}_{\mathbf{u}i} = 0 \text{ otherwise.} \quad (4)$$

Then, the following holds:

Theorem 1 *Let* $(\mathbf{A}, \mathbf{b}, \mathbf{c}) = \mathcal{F}(G)$ *be a k-local LP and* $\widehat{X}$ *be the characteristic matrix of the k-WL partition* $\mathcal{W}$ *of G. Then* $\mathcal{F}'(G) = (\mathbf{A}\widehat{X}, \mathbf{b}, \widehat{X}^T \mathbf{c})$ *is an LP with* P *variables. Moreover it is equivalent to $\mathcal{F}(G)$ in the following sense:* **(A)** *if y is feasible in $\mathcal{F}'(G)$, then* $\mathbf{x} = \widehat{X}\mathbf{y}$ *is feasible in $\mathcal{F}(G)$;* **(B)** *if $\mathbf{y}^*$ is an optimum of $\mathcal{F}'(G)$, then* $\mathbf{x}^* = \widehat{X}\mathbf{y}^*$ *is an optimum of $\mathcal{F}(G)$.*

Proof See Appendix (A). □

That is, we have a lifted solver for k-local LPs in general as summarized in Alg. 1.

3.2 Lifting k-MAP-LP

As a consequence of Theorem 1, all we need for efficient lifting of tightened MAP-LPs is to show they are local. This is indeed the case, as the following theorem states.

Theorem 2 *The k-level Sherali-Adams tightening of MAP-LP is $(k+1)$-local in G.*

Proof See Appendix (B). □

Consequently, we can compute a solution to the k-level tightening of MAP-LP by **(1)** instantiating the LP for an Ising model G and **(2)** running Alg. 1. Note that WL treats G as an edge- and vertex-colored graph – the pairwise parameters θ_{ij} serve as edge colors, i.e. $C(u_i, u_j) = C(u_k, u_l) \Leftrightarrow \theta_{ij} = \theta_{kl}$, whereas the unary parameters θ_i serve as node colors. Thus parameter symmetry in the model is considered by WL as well.

Theorems 1 and 2 together significantly advance the understanding of lifted inference. First, they provide a complementary view on the spectrum of approximations of lifted inference approaches recently introduces by Van den Broeck *et al.* [15] with a clear mathematical notion of symmetry. At the bottom of the hierarchy is lifted MAP-LP, and exact MAP inference is at the top. In between, there are relaxations whose liftings are equitable partitions of intermediate coarseness. To calculate the k level partition we need to run k-WL

Algorithm 1: Lifted Solver for k-local LPs
Input: A k-local LP $(\mathbf{A}, \mathbf{b}, \mathbf{c}) = \mathcal{F}(G)$
1 Run k-WL(G) to obtain a partition $\mathcal{W}$ of V^k;
2 Calculate $\widehat{X}$ for $\mathcal{W}$ as in Eq. 4;
3 Solve the LP $(\mathbf{A}\widehat{X}, \boldsymbol{b}, \widehat{X}^T \boldsymbol{c})$. Denote solution by $\mathbf{y}^*$;
4 **return** $\mathbf{x}^* = \widehat{X}\mathbf{y}^*$

---

whose complexity is $\mathcal{O}(k^2 n^{k+1} \log(n))$ (e.g., see [25]). Moreover, they generalize the lifted MAP-LPs of [24], which are based on non-fractional automorphisms. To see this, suppose that we know the automorphism group $\text{Aut}(G)$ of a model G. Then, we can partition the $(\leq k)$-tuples of G according to the following rule: $\mathbf{u}, \mathbf{v} \in W_p$ if and only if $u_1, \ldots, u_k = \pi(v_1), \ldots, \pi(v_k)$, for some $\pi \in \text{Aut}(G)$. If we lift k-MAP-LP with this partition in the same way as in Theorem 1, we get LPs identical to the special cases $k = 1, 2$ proposed in [24]. Since, by design k-WL approaches the former partitions with increasing k, our lifted LPs are at least as coarse. Moreover, they can be faster to compute since WL does not rely on GI-completeness.

3.3 Lifting MAP-LPs for Relational Models

The notion of locality results in even more dramatic speedups for the case of relational models. Specifically, let us focus on Markov logic networks (MLNs). See [30] for an introduction. To obtain the graphical model corresponding to an MLN one needs to ground the MLN which can be quite costly. Furthermore, if one then wants to run k-MAP-LP on the resulting model, this requires further compilation efforts. Here we show that symmetries in the MAP-LP of an MLN can be inferred without even grounding, thus resulting in considerable saving. The key to doing this is to show that the MAP-LP is in fact local in a much smaller graph, which we call the domain graph.

Consider the classical Smokers-Friends (SF) MLN shown in Fig. 1. Now assume we want to eventually solve the MAP-LP on the grounding of this model. For that sake of simplicity,[9] consider a class of MLNs with one unary predicate $\mathbf{u}(X)$ and one symmetric binary predicate $\mathbf{b}(X, Y)$ only, ; e.g. the first clause of the SF MLN where $\mathbf{u} = \mathbf{sm}$ and $\mathbf{b} = \mathbf{fr}$. In this case, we can directly apply Eq. 2 to the grounded network and obtain the following LP (nonnegativity constraints are omitted for clarity): $P(M) =$

$$\left\{ \begin{array}{l} \forall C \in \text{clauses}(M): \quad \forall (d_1, d_2) \in D \times D \\ \mu^C_{d_1, d_2} \leq \mu^{\mathbf{b}}_{d_1, d_2}, \mu^C_{d_1, d_2} \leq \mu^{\mathbf{u}}_{d_1}, \\ \mu^{\mathbf{b}}_{d_1, d_2} + \mu^{\mathbf{u}}_{d_1} - \mu^C_{d_1, d_2} \leq 1 \end{array} \right\} . \quad (5)$$

---

[9]Our arguments generalize to arbitrary pairwise MLNs. Every MLN can be converted to a pairwise one [31].

$D = \{d_1, ..., d_5\}, X, Y \in D$

$\theta_1 : \mathtt{fr}(X,Y) \wedge \mathtt{sm}(X)$ (C_1)

$\theta_2 : \mathtt{sm}(X) \Rightarrow \mathtt{ca}(X)$ (C_2)

$T : \mathtt{fr}(d_1, d_2), \mathtt{fr}(d_3, d_4), \mathtt{sm}(d_2), \mathtt{sm}(d_4),$

$F : \neg\mathtt{ca}(d_5),$ (E)
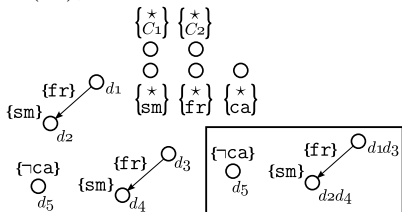
Figure 1: (Top) The MLN SF. Here D is a given set of people, $\mathtt{fr}$ encodes friendship relationships, $\mathtt{sm}$ whether a person smokes, and $\mathtt{ca}$ whether a person has cancer. E.g. the second rule says that smoking causes cancer to a certain extent θ_2. The sets T, F encode our evidence (E) about $\mathtt{ca}$ and $\mathtt{fr}$. (Bottom) The domain graph of SF and the resulting lifting (Box).

Here, the variable $\mu_{d_1,d_2}^{\mathtt{b}}$ corresponds to the pseudo-marginal of the ground atom $\mathtt{b}(d_1, d_2)$. Additionally, we get the evidence constraints $\mu_{d_1,d_2}^{\mathtt{b}} = 1$ if $\mathtt{b}(d_1, d_2)$ is observed true and $\mu_{d_1,d_2}^{\mathtt{b}} = 0$ if the atom is observed false (and likewise for the unary predicate). It is not difficult to see that Eq. 5 is a local LP and hence can be lifted using Theorem 1. However, instead of applying the LP generating procedure to the graph induced by the grounding of the MLN, we can equivalently consider the following directed graph, which we call the domain graph[10] $G_D(M)$ of M.

The vertices of the domain graph $G_D(\text{M})$ consist of the domain elements of M (called the domain vertices) and, for every clause and predicate name, we add a single vertex (called a name vertex). In our running example, $G_D(\text{SF}) = \{d_1, ..., d_5, \mathtt{sm}, \mathtt{ca}, \mathtt{fr}, C_1, C_2\}$. The connectivity and coloring of this graph are determined by the evidence. For every unary predicate $\mathtt{u}$, we color a domain vertex d_i with $\mathtt{u}$ or $\neg\mathtt{u}$ if $\mathtt{u}(d_i)$ or $\neg\mathtt{u}(d_i)$ (recall that we allow more than one color per vertex). We add a directed edge from d_i to d_j if for some binary predicate $\mathtt{b}$, either $\mathtt{b}(d_i, d_j)$ or $\neg\mathtt{b}(d_i, d_j)$ is observed; and we color the edge accordingly. Finally, we color the name vertices with the predicate/clause name together with a unique color (e.g. $\star$) to separate them from the domain vertices and let them remain disconnected. The resulting complete domain graph $G_D(\text{SF})$ is shown in Fig. 1. Its size is inde-

pendent of the number of ground clauses and grows only with the size of the evidence since it consists of $|D| + |\text{clauses}(M)| + |\text{predicates}(M)|$ many vertices and $|\text{evidence}|$ many edges.

We note that $P(M)$ is exactly 3-local in $G_D(\text{M})$. This implies that a partition of (not necessarily all) indistinguishable variables of MAP-LP for MLNs can be computed by 3-WL executed on $G_D(\text{M})$. To accommodate for predicates of higher arity, the domain graph becomes a colored oriented hyper-graph (i.e. a colored collection of tuples)[11]. The locality of the corresponding MAP-LP is then given by the maximum number of different domain elements that can be encountered in a ground clause plus one (for name vertices). We upper-bound this by $v + 1$, where v is the maximum number of free variables in a clause over all clauses. Hence, we have the following Theorem:

Theorem 3 $P(M)$ *is $(v+1)$-local in $G_D(M)$.*

Thm. 3 has several important consequences. First of all, we do not have to ground anymore. We can compute a lifting based on the structure of the MLN and the given evidence, as incorporated in $G_D(M)$ only. Second, for a fixed v, lifting is polynomial in the number of domain elements. Domain elements that do not participate in evidence need not be considered, since they are indistinguishable disconnected nodes of the domain hyper-graph. This results in a speed-up. Third, we know 1-MAP-LP of a propositional graph is 2-local, and the ground MLN contains $\mathcal{O}(n^v)$ ground atoms with $n = |D|$. Recall further that k-WL runs in time $\mathcal{O}(k^2 m^{k+1} \log m)$ for a graph of m vertices. Hence, the cost of running 1-WL on the grounded MAP-LP (the approach of [10]) is then $\mathcal{O}((n^v)^2 \log(n^v)) = \mathcal{O}(vn^{2v} \log(n))$ in terms of domain elements. Running $(v + 1)$-WL on the domain hyper-graph would cost $\mathcal{O}(v^2 n^{v+2} \log(n))$. Assuming $v << n$, lifting via the domain graph has a clear advantage. Moreover, by virtue of composing the above construction with the generation procedure for k-MAP-LP, we have an end-to-end locality of $k(v+1)$, if we see k-MAP-LP as a function of the domain graph. This results in even further speed-ups as opposed to either grounding, tightening and lifting or grounding and applying Section 3.2. Finally, the locality of MLN MAP-LPs does not depend on the number of clauses *per se*, as k-local LPs are closed under unions. Hence, adding clauses with the same number of free variables will not increase the ability to distinguish between domain elements and refine the partition.

---

[10]This is similar to Bui *et al.*'s [24] renaming permutations but is different in that we have to encode the predicates to conform to the theory.

[11]All our previous arguments carry over to colored hyper-graphs. The concept of isomorphism type can be extended and colored, oriented hyper-graph can be uniquely converted to a colored simple graph preserving all topological information by the addition of linearly many extra vertices.
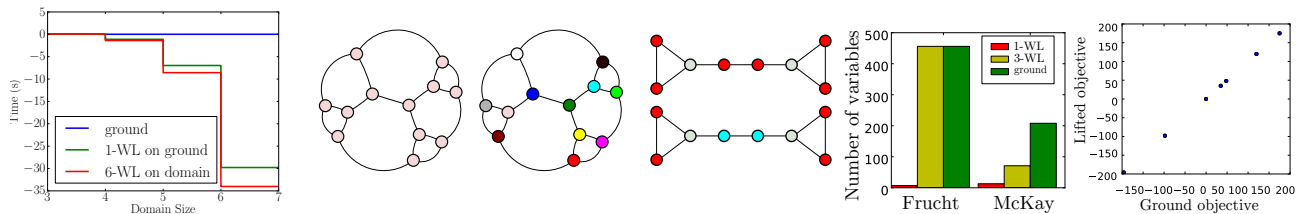
Figure 2: (Left) End-to-end benefits in running times on the SF MLN without evidence (the lower, the faster). (Middle) Frucht and McKay graphs used to encode different social networks as evidence. The colors indicate the node partitions (of these graphs but not for the corresponding MAP-LPs) yielded by 1-WL and 3-WL, which in this case coincide with the CEP resp. OP. (Right) Statistics on the SF MLN using the social networks as evidence. (a) Sizes of ground and lifted LPs. (b) Achieved objectives. (Best viewed in color)

4   Experimental Illustration

To illustrate our results experimentally, we conducted experiments on SF MLN.

The first experiment investigates whether domain lifting can be faster. To compare the speed for inference, we considered the 2-MAP-LP for the MLN with no evidence with up to 7 people. This induced ground LPs with up to 85400 variables and 670036 constraints. We compared ground inference (i.e., no lifting), 1-WL on the grounded MLN following [10], and domain lifting (i.e., 6-WL on the domain graph). Fig. 2(left) summarizes the end-to-end benefit in running times, i.e., inference time minus the time for ground inference minus (including grounding, lifting, and solving the LP). The results clearly show that domain lifting is beneficial and it can speed up inference considerably.

The second experiment illustrates the difference between symmetries based on automorphisms and based on fractional automorphisms. We used the so-called Frucht (among 12 people) and McKay (among 8 people) graphs, see Fig. 2(middle), to encode social networks among people as evidence in the SF MLN. As one can see, 1-WL clusters all nodes of the Frucht graph together. This is fine for 1-MAP-LP. As soon as we want to use domain lifting or move up the SA hierarchy, we have to use at least 3-WL, which automatically separates all nodes. In contrast, the OP resulting from using standard automorphisms to identify symmetries [24] separates all nodes already for 1-MAP-LP and hence results in finer lifting for 1-MAP-LP compared to using fractional automorphisms. For McKay, even 3-WL still identifies symmetries, the same as the OP in this case. Overall, this shows that fractional symmetries can yield coarser liftings for MAP-LPs compared to using automorphisms. This is confirmed by Fig. 2(right,a), which summarizes the sizes of the corresponding lifted LPs. Finally, for all experiments we recorded the achieved objective as shown in Fig. 2(right,b); they always coincided.

5   Conclusions

Shifting focus towards fractional symmetries, we developed a sound and tractable lifting for a large class of MAP LP inference algorithms. The approach is well-grounded in a graphical property of the inference task at hand, namely its locality, recently introduced for LPs in general. For LP relaxations of MAP, we showed that the way to exploit symmetries is independent of the particular relaxations and — as long as they are k-local — is realized by the k-WL algorithm. Both the time for and the amount of lifting depends on k. More local inference algorithms yield larger compression in less time but might be less tight. Thus, locality contributes to a deeper understanding of the interaction between symmetries and the complexity of probabilistic inference. Moreover, the overlap with lifted BP suggests that the notion of locality goes beyond LPs. Exploring this is an exciting avenue for future work. Finally, it would be interesting to design message passing algorithms for solving the lifted MAP-LPs discovered by our approach.

6   Appendix

(A)   Proof of Theorem 1.   The first step towards k-WL lifting of LPs is to note that it is commonly used as a heuristic for the graph isomorphism problem. To do so, take a pair of graphs, G and H, for which isomorphism must be determined and run k-WL on both. Next, compare the set of colors produced on both graphs. If they are different the graphs are certainly not isomorphic. If they are the same, we call the two graphs k-WL equivalent, $G \equiv_{\mathrm{WL}}^{k} H$. The $\equiv_{\mathrm{WL}}^{k}$ equivalence relation is identical to isomorphism on

many graph classes, but not in general, as counterexamples have been found. Whenever $G \equiv_{\mathrm{WL}}^k H$, we can build a symmetric compatibility matrix $X^{G \mapsto H} \in \mathbb{R}^{|V^{\leq k}| \times |U^{\leq k}|}$. In other words, a matrix whose columns are indexed by the $(\leq k)$-tuples of G and rows by the $(\leq k)$-tuples of H (or vice versa since X is symmetric). We do this by setting $X_{\mathbf{uv}}^{G \mapsto H} = 1/|W_m|$ if $\mathbf{u}, \mathbf{v} \in W_m$ for some m and 0 otherwise (*). We will call X the *flat matrix* of the partition $\mathcal{W}$.

Next, we note that k-local LPs cannot distinguish k-WL isomorphic graphs.

Lemma 6 *Let $G = (V, E)$ and $H = (U, F)$ be graphs such that $G \equiv_{WL}^k H$ and let $X^{G \mapsto H}$ be as above. Then for a k-local LP generating procedure $\mathcal{F}$, $\mathcal{F}(G)$ is feasible iff $\mathcal{F}(H)$ is feasible. If the vector $\mathbf{x}^* = (\mathbf{x_v})_{\mathbf{v} \in V^{\leq k}}$ is a solution to $\mathcal{F}(G)$, then $\mathbf{y}^* = (\mathbf{y_u})_{\mathbf{u} \in U^{\leq k}}$ with $\mathbf{y_u} = \sum_{\mathbf{v} \in V^{\leq k}} X_{\mathbf{vu}}^{G \mapsto H} \mathbf{x_v}$ is a solution to $\mathcal{F}(H)$.*

Proof This directly follows the two main results in [26]: (i) X as above solves the k-th level SA tightening of the Fractional Isomorphism (FI) problem (Lemma 6)[12]. (ii) Solutions of k-local LPs are preserved by solutions of k-SA-FI (Thm. 2). $\square$

Now we have everything at hand to lift local LPs using k-WL. Note that the above result gives a way to compute a solution of $\mathcal{F}(H)$ from a solution of $\mathcal{F}(G)$, given that $G \equiv^k H$. We will now show that this method also yields the fractional symmetries of a single graph and the related $\mathcal{F}(G)$.

Clearly, for any graph G, $G \equiv_{WL}^k G$ regardless of k. If the coloring produced by k-WL on the tuples of G is trivial (i.e. every tuple is uniquely colored), then $X^{G \mapsto G}$ (from now on we will call this simply X) is the identity matrix. However, suppose the coloring is not trivial. Then, applying Lemma 6 yields that the image of C contains a solution to $\mathcal{F}(G)$. I.e., if $\mathbf{x}^*$ solves $\mathcal{F}(G)$, then so does $X\mathbf{x}^*$. This implies that restricting the search space of $\mathcal{F}(G) = (\mathbf{A}, \mathbf{b}, \mathbf{c})$ to the span of X (i.e. solving $\mathcal{F}(G)|_X = \operatorname{argmax}_{\mathbf{x}} \mathbf{c}^T(X\mathbf{x})$ s.t. $\mathbf{A}(X\mathbf{x}) \leq \mathbf{b}$ resp. $\mathcal{F}(G)|_X = (\mathbf{A}X, \mathbf{b}, X^T\mathbf{c})$) yields at least one solution to $\mathcal{F}(G)$.

Observe that the flat matrix X of any partition $\mathcal{P} = \{P_1, ..., P_p\}$ (regardless of whether it has special properties) admits a factorization as follows: $X = \hat{X}\hat{X}^T$, where $\hat{X}_{im} = 1/\sqrt{P_m}$ if $i \in P_m$ and $\hat{X}_{im} = 0$ otherwise. Now, X computed according to (*) is the flat matrix of the k-WL partition $\mathcal{W} = \{W_1, .., W_P\}$ of tuples of G having P classes. Hence, it admits the factorization $X = \hat{X}\hat{X}^T$, with $\hat{X}_{\mathbf{u}m} = 1/\sqrt{|W_m|}$ if tuple $\mathbf{u}$ is in color class W_m and 0 otherwise. Here, W_m is

---

[12]This result appears more explicitly in [32]. Note, however, that the $k-1$ of their convention is the k of ours.

the set of all nodes having the same color m. Because of this factorization and the above argument, we have that at least one solution of $\mathcal{F}(G)$ can be expressed as $\hat{X}\hat{X}^T\mathbf{x}^*$. Finally, note that $\operatorname{rank}(\hat{X}^T) = \operatorname{rank} X = P$, hence $\{X^{G \mapsto G}\mathbf{x} | \mathbf{x} \in \mathbb{R}^{|V^{\leq k}|}\} = \{\hat{X}\mathbf{y} | \mathbf{y} \in \mathbb{R}^P\}$. This allows us to re-express our restricted LP as $\mathcal{F}(G)|_X = (\mathbf{A}\hat{X}, \mathbf{b}, \hat{X}^T\mathbf{c})$. However, $\mathcal{F}(G)|_X$ is now a problem in only w variables! Thus, whenever G has a nontrivial k-WL partition, restricting the LP to $\hat{X}$ may yield significant reductions in size. $\square$

(B) Proof of Theorem 2. The argumentation follows the structure of the locality proofs in [26], and we refer the reader to [26] for a more detailed discussion of locality proofs. We will first restate an observation of [26]. Let L be a linear program whose variables and constraints are indexed by $(\leq k)$-tuples of V. Let us fix the constraint indexed by the tuple $\mathbf{u}$ and a subset $P \subseteq V, |P| \leq k$. Now, if for all $\mathbf{v}$, we have that $\mathbf{A_{uv}}$ is nonzero *only if* $\mathbf{v} \in P^{(\leq k)}$ and all elements of $\mathbf{u}$ are also elements of $\mathbf{v}$, then this LP is k-local. This special case is called *bounded k-local* in [26].

Recall that we produce the k-MAP-LP by multiplying every constraint of MAP-LP by the expression $\prod_{i \in I} \mu_i \prod_{j \in J}(1 - \mu_j)$ for all $I, J \subseteq V$ with $|I \cup J| \leq t - 1$ and then linearizing. We let $S(I, J) = \sum_{J' \subseteq J}(-1)^{|J'|} \mu_{I \cup J'}$ be the linearization of $\prod_{i \in I} \mu_i \prod_{j \in J}(1 - \mu_j)$ (note that here $|I \cup J| \leq t - 1$ does not necessarily hold). Observe that $S(I, J) = 0$ if I and J are not disjoint. Then, the constraints of k-MAP-LP become: $\forall \{u, v\} \in E$:

$$S(I \cup \{u, v\}, J) \leq S(I \cup \{u\})(A)$$
$$S(I \cup \{u\}) + S(I \cup \{v\}, J) - S(I \cup \{u, v\}, J) \leq S(I, J)$$
$$S(I \cup \{u, v\}, J) \geq 0$$
$$S(I \cup \{u\}, J) \geq 0$$

We treat every line of the above as a separate LP. Note the variables are indexed by sets instead of tuples, hence, we replace every set-indexed variable μ_I, $I = \{v_1, ..., v_p\}$ by the tuple-indexed variable $\mu_{\mathbf{a}}$, $\mathbf{a} = (v_1, ..., v_p)$. Since this introduces a dependency on the order of the variables which was not there before, we must add the additional constraints $\mu_{\mathbf{a}} = \mu_{(\pi \circ \mathbf{a})}$, for every permutation $\pi : \{1, ..., p\} \to \{1, ..., p\}$. Observe that the indices of the variables and constraints range over $P = I \cup J \cup E$. Since $|I \cup J| \leq k - 1$, we have $|P| \leq k + 1$. Hence, we are exactly in the case described at the beginning of the proof, implying all constraints are bounded $k + 1$-local LPs. Finally, we encode the edge weights of the Ising model as edge colors. Suppose there are m distinct values, $r_1, ..., r_m$ among all $\theta'_{ij}s$. We say $\mathrm{C}(ij) = p$ if $\theta_{ij} = r_i$. Then, the objective of the LP is generated as $\mathbf{c_v} = \mathrm{C}(\mathbf{v})$ if v is a 2-tuple and 0 otherwise. $\square$

References

[1] K. Kersting. Lifted probabilistic inference. In *Proceedings of ECAI-2012*. IOS Press, 2012.

[2] L. Getoor and B. Taskar, editors. *An Introduction to Statistical Relational Learning*. MIT Press, 2007.

[3] L. De Raedt, P. Frasconi, K. Kersting, and S.H. Muggleton, editors. *Probabilistic Inductive Logic Programming*, volume 4911 of *Lecture Notes in Computer Science*. Springer, 2008.

[4] L. De Raedt. *Logical and Relational Learning*. Springer, 2008.

[5] P. Sen, A. Deshpande, and L. Getoor. Exploiting Shared Correlations in Probabilistic Databases. In *Proc. of the Intern. Conf. on Very Large Data Bases (VLDB-08)*, 2008.

[6] K. Kersting, Y. El Massaoudi, B. Ahmadi, and F. Hadiji. Informed lifting for message–passing. In *Proc. of the 24th AAAI Conf. on Artificial Intelligence (AAAI–10)*, 2010.

[7] J. Choi, A. Guzman-Rivera, and E. Amir. Lifted relational kalman filtering. In *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence(IJCAI)*, pages 2092–2099, 2011.

[8] A. Nath and P. Domingos. Efficient lifting for online probabilistic inference. In *Proc. of the 24th AAAI Conf. on Artificial Intelligence (AAAI)*, 2010.

[9] C. Kiddon and P. Domingos. Coarse-to-fine inference and learning for first-order probabilistic models. In *Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI 2011)*, 2011.

[10] M. Mladenov, B. Ahmadi, and K. Kersting. Lifted linear programming. In *15th Int. Conf. on Artificial Intelligence and Statistics (AISTATS 2012)*, pages 788–797, 2012. Volume 22 of JMLR: W&CP 22.

[11] B. Ahmadi, K. Kersting, M. Mladenov, and S. Natarajan. Exploiting symmetries for scaling loopy belief propagation and relational training. *Machine Learning Journal*, 92:91–132, 2013.

[12] P. Singla and P. Domingos. Lifted First-Order Belief Propagation. In *Proc. of the 23rd AAAI Conf. on Artificial Intelligence (AAAI-08)*, pages 1094–1099, Chicago, IL, USA, July 13-17 2008.

[13] K. Kersting, B. Ahmadi, and S. Natarajan. Counting Belief Propagation. In *Proc. of the 25th Conf. on Uncertainty in Artificial Intelligence (UAI–09)*, 2009.

[14] B. Ahmadi, K. Kersting, and S. Sanner. Multi-Evidence Lifted Message Passing, with Application to PageRank and the Kalman Filter. In *Proc. of the 22nd Int. Joint Conf. on AI (IJCAI–11)*, 2011.

[15] G. Van den Broeck, A. Choi, and A. Darwiche. Lifted relax, compensate and then recover: From approximate to exact lifted probabilistic inference. In *Proc. of the 28th Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2012.

[16] D. Sontag and T. Jaakkola. New outer bounds on the marginal polytope. In *Proc. of the 21st Annual Conference on Neural Information Processing Systems (NIPS-2007)*, 2007.

[17] M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, January 2008.

[18] O. Meshi, T. Jaakkola, and A. Globerson. Convergence rate analysis of MAP coordinate minimization algorithms. In *Advances in Neural information Processing Systems 25*. MIT Press, 2012.

[19] D. Sontag, A. Globerson, and T. Jaakkola. Introduction to dual decomposition for inference. In S. Sra, S. Nowozin, and S. J. Wright, editors, *Optimization for Machine Learning*, pages 219–285. MIT Press, 2011.

[20] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening LP relaxations for map using message passing. In *Proc. of the 24th Conference in Uncertainty in Artificial Intelligence (UAI-2008)*, pages 503–510, 2008.

[21] D. Sontag, D.K. Choe, and Y. Li. Efficiently searching for frustrated cycles in map inference. In *Proc. of the 28th Conference on Uncertainty in Artificial Intelligence (UAI-2012)*, pages 795–804, 2012.

[22] M. Niepert. Markov chains on orbits of permutation groups. In *Proc. of the 28th Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2012.

[23] J. Noessner, M. Niepert, and H. Stuckenschmidt. Rockit: Exploiting parallelism and symmetry for map inference in statistical relational models. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA*, 2013.

[24] H.H. Bui, T.N. Huynh, and S. Riedel. Automorphism groups of graphical models and lifted variational inference. In *Proc. of the 29th Conference on Uncertainty in Artificial Intelligence (UAI-2013)*, 2013.

[25] J.-Y. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12:389–410, 1992.

[26] A. Atserias and E. Maneva. Sherali-Adams relaxations and indistinguishability in counting logics. *SIAM Journal of Computation*, 42(1):112–137, 2013.

[27] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[28] G. Hahn and G. Sabidussi. *Graph symmetry: algebraic methods and applications.* Springer, 1997.

[29] C. Berkholz, P. Bonsma, and M. Grohe. Tight lower and upper bounds for the complexity of canonical colour refinement. In *Proc. of the 21st Annual European Symposium on Algorithms, Sophia Antipolis, France, September 2-4, 2013.*, pages 145–156. Springer, 2013.

[30] M. Richardson and P. Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[31] D. Fierens, K. Kersting, J. Davis, J. Chen, and M. Mladenov. Pairwise markov logic. In *Postproc. of the 22nd International Conference on Inductive Logic Programming (ILP-2012).*, 2012.

[32] M. Grohe and M. Otto. Pebble Games and Linear Equations. In *Proceedings of the 21st Annual Conference of the European Association for Computer Science Logic (CSL*, volume 16, pages 289–304, Dagstuhl, Germany, 2012.