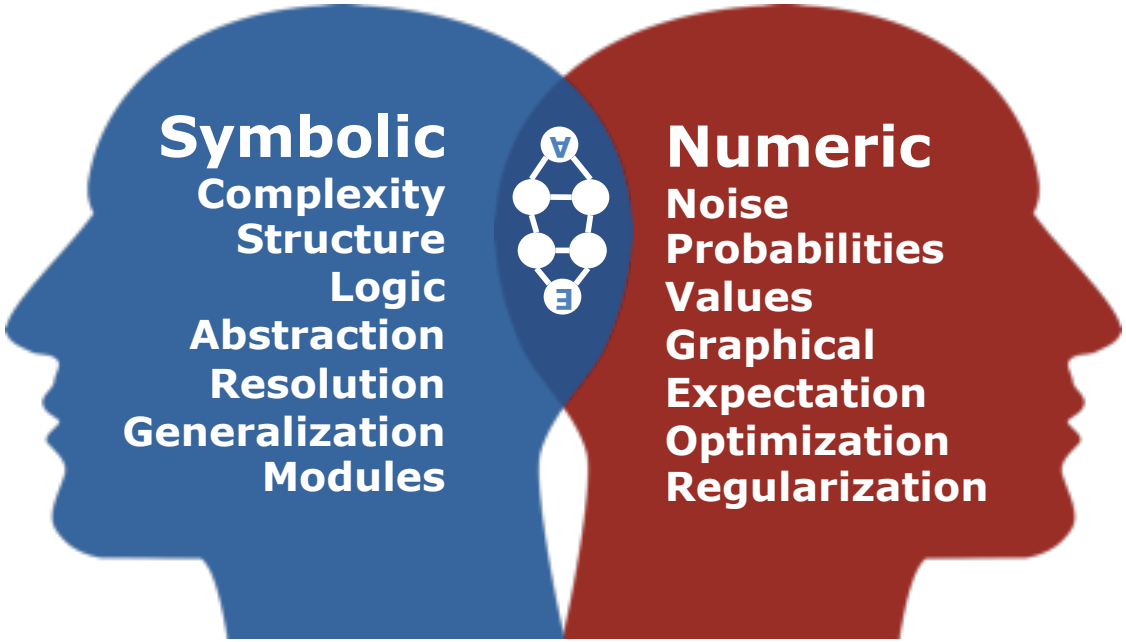


TECHNISCHE UNIVERSITÄT DARMSTADT

Lifted Belief Propagation [Jaimovich-UAI07, Singla-AAAI08, Kersting-UAI09]

Lifted Bisimulation/Mini-buckets [Sen-VLDB08, Sen-UAI09]

Lifted Importance Sampling [Gogate-UAI11, Gogate-AAAI12]

Lifted Relax, Compensate & Recover (Generalized BP) [VdB-UAI12]

Lifted MCMC [Niepert-UAI12, Niepert-AAAI13, Venugopal-NIPS12]

Lifted Variational Inference [Choi-UAI12, Bui-StarAI12]

Lifted MAP-LP [Mladenov-AISTATS14, Apsel-AAAI14] and many more …

Statistical Relational AI

Probabilistic Relational Models - Lifted Approximate Inference

Kristian Kersting

Thanks to Rina Dechter, Luc De Raedt, Pedro Domingos, Peter Flach, Vibhav Gogate, Carlos Guestrin, Daphen Koller, Nir Friedman, Ray Mooney, Sriraam Natarajan, David Poole, Fabrizio Riguzzi, Dan Suciu, Guy van den Broeck, and many others for making their slides publically available

Relational Probabilistic Models also provide opportunities to speed up inference and learning

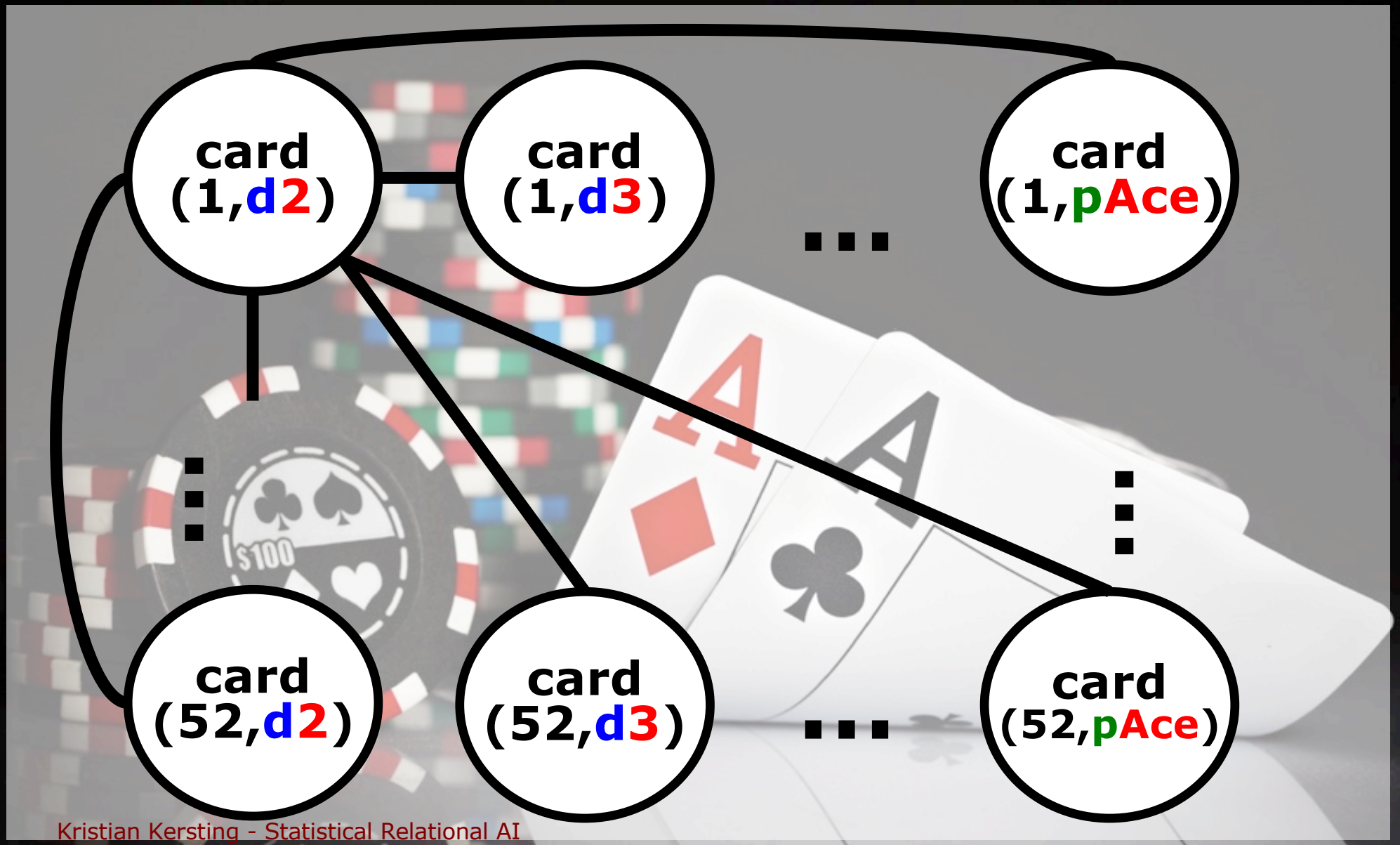
A simple example

Guy van den Broeck
UCLA

What is the probability that the first card of a randomly shuffled deck with 52 cards is an Ace?

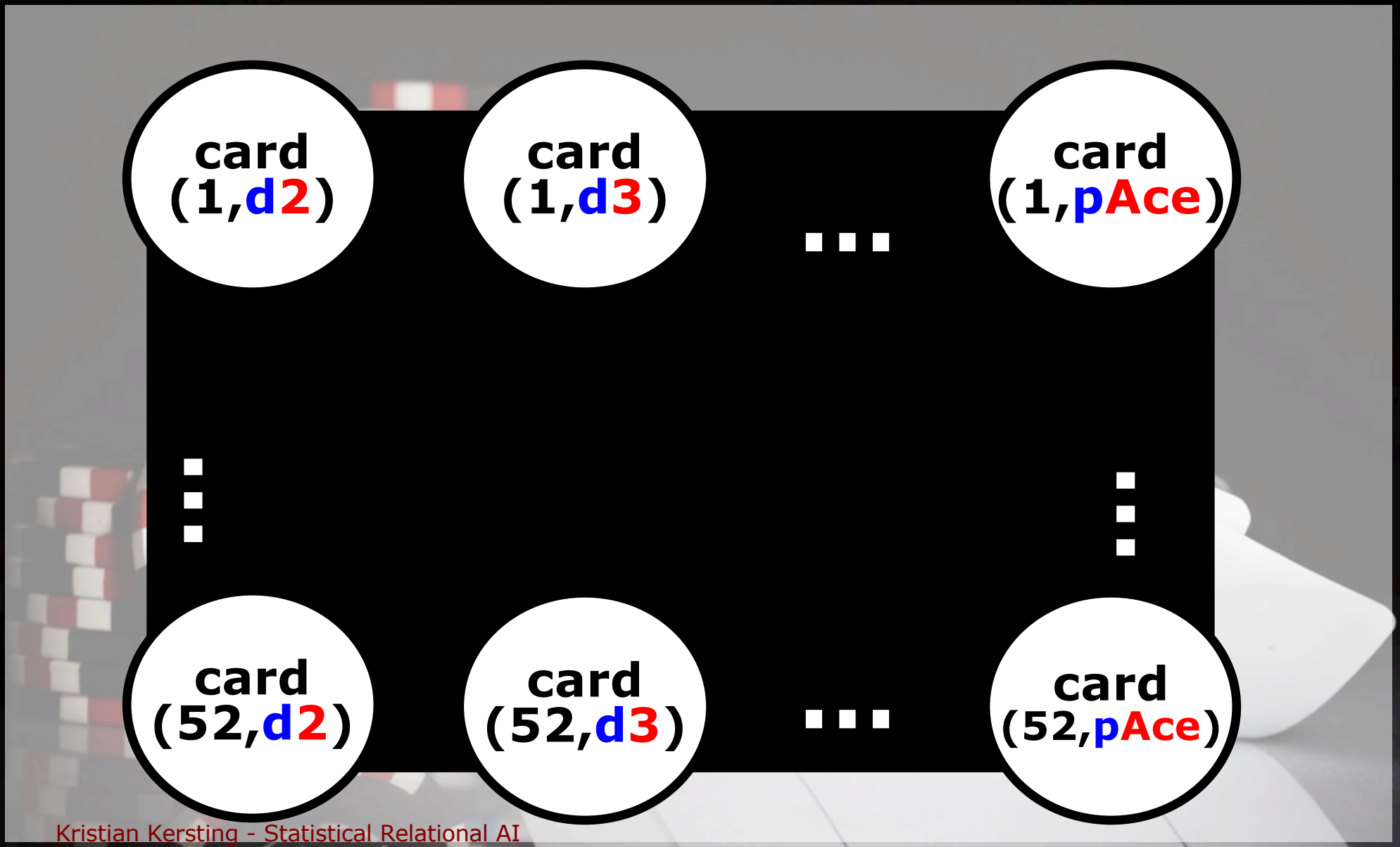
A simple example

Guy van den Broeck
UCLA



A simple example

Guy van den Broeck
UCLA



A simple example

Guy van den Broeck
UCLA

card (1,d2)

card (1,d3)

...

card (1,pAce)

No independencies.
Fully connected.
2^{2704} states

card (52,d2)

card (52,d3)

...

card (52,pAce)

A simple example

Guy van den Broeck
UCLA

card (1,d2)

card (1,d3)

...

card (1,pAce)

A machine will not solve the problem

card (52,d2)

card (52,d3)

...

card (52,pAce)

What are we missing ?

TECHNISCHE UNIVERSITÄT DARMSTADT

Faster modelling

Let's use a high-level language e.g.
Markov Logic Networks (MLNs)

$w1: \forall p,x,y: \text{card}(P,X),\text{card}(P,Y) \Rightarrow x=y$

$w2: \forall c,x,y: \text{card}(X,C),\text{card}(Y,C) \Rightarrow x=y$

and symmetry- and language-aware infernce

Faster inference and learning

Positions and cards are exchangable but the machine is not aware of these symmetries

Lifted Approximate Inference

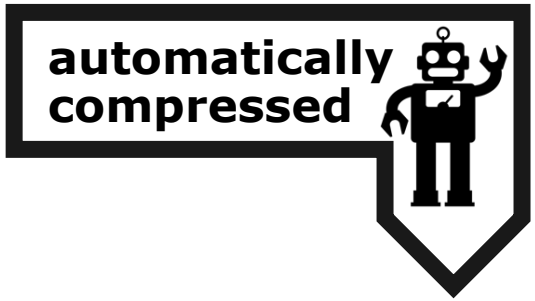
- There are many other, in particular, lifted inference approaches, in particular for exact inference such as
  - Knowledge compilation
  - Lifted recursive conditiong
- Here, we want to take a "group-theoretic" view on approximate lifted inference, also because it connects to Graph Neural Networks
- To do so, we start by lifting (loopy) belief propagation

[Singla, Domingos AAAI'08; Kersting, Ahmadi, Natarajan UAI'09; Ahmadi, Kersting, Mladenov, Natarajan MLJ'13]

Lifted Loopy Belief Propagation

Exploiting computational symmetries

Big Model



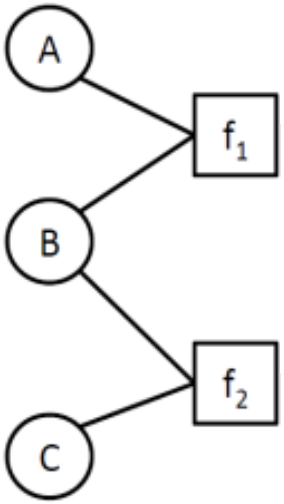
Small Model

Run Loopy Belief Propagation

Run a modified Loopy Belief Propagation

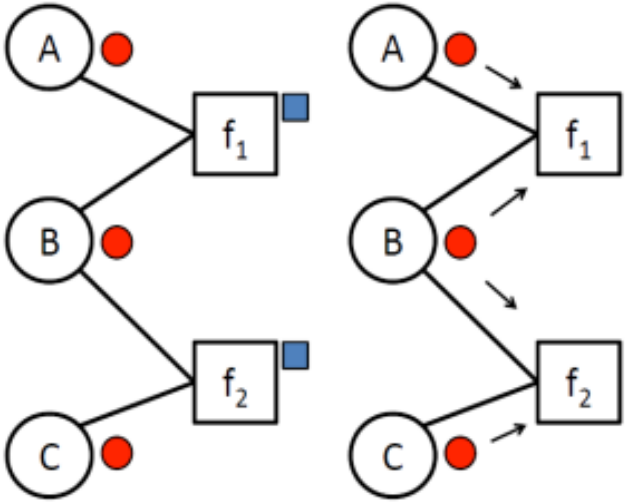
What are symmetries in (loop) belief propagation?

Compression: Coloring the graph



- **Color nodes according to the evidence you have**
  - No evidence, say **red**
  - State „one", say **brown**
  - State „two", say **orange**
  - ...
- **Color factors distinctively according to their equivalences**
  For instance, assuming f_1 and f_2 to be identical and B appears at the second position within both, say **blue**

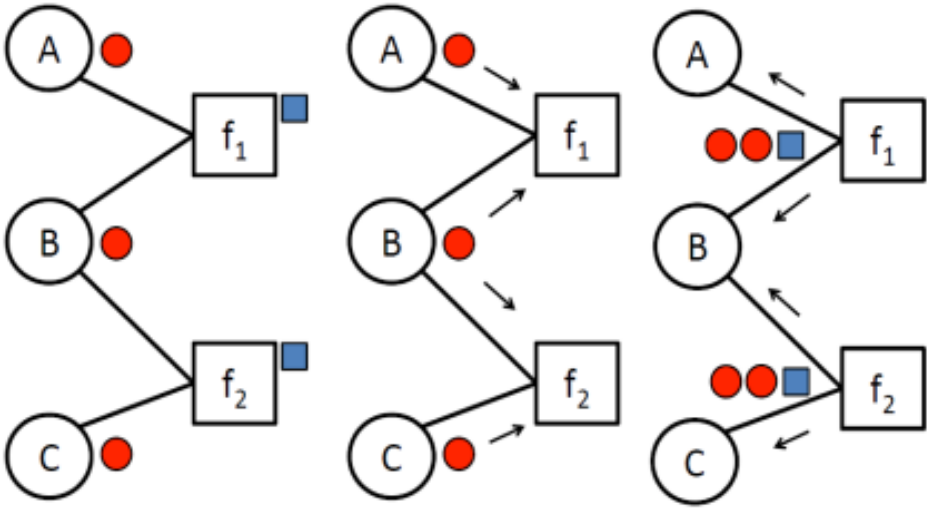
Compression: Pass the colors around



1. Each factor collects the colors of its neighboring nodes

Compression: Pass the colors around*

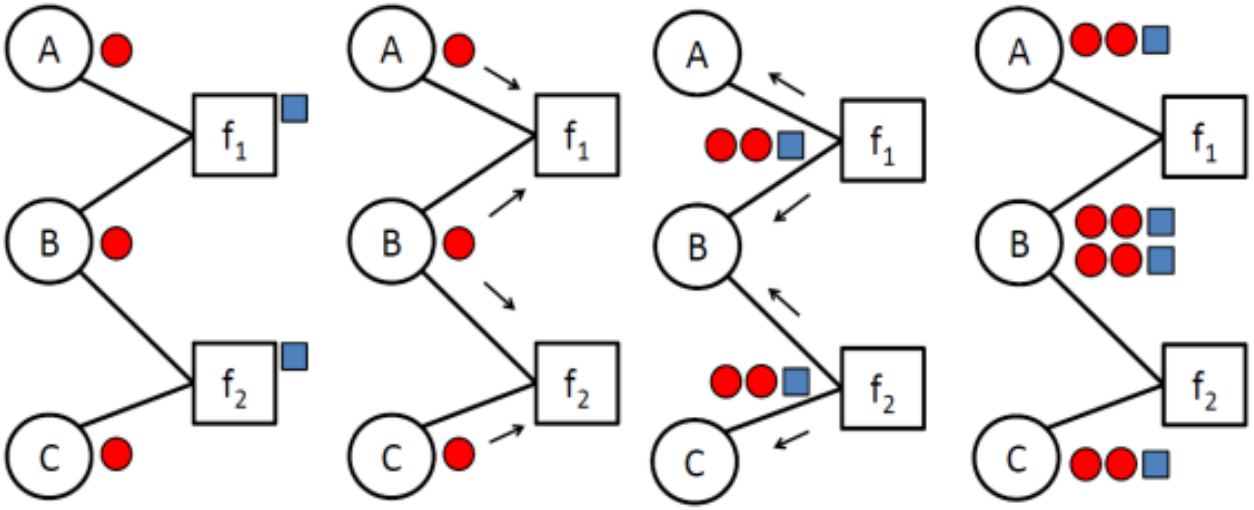
***can also be done at the „lifted", i.e., relational level**



1. Each factor collects the colors of its neighboring nodes
2. Each factor „signs" ist color signature with its own color

Compression: Pass the colors around*

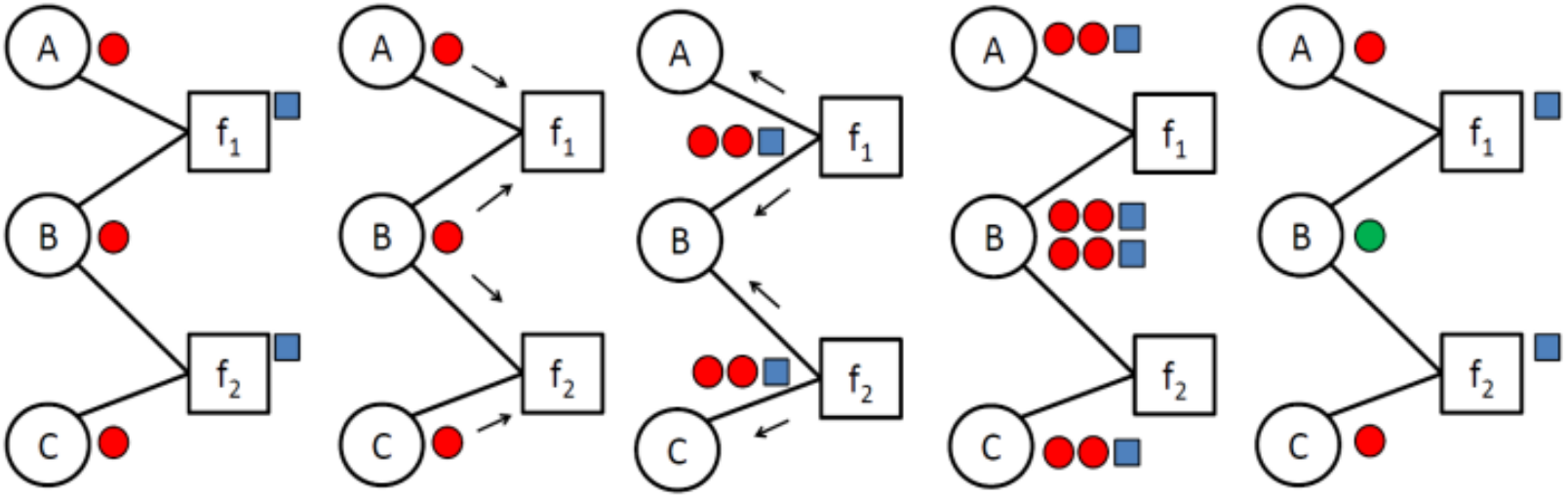
***can also be done at the „lifted", i.e., relational level**



1. Each factor collects the colors of its neighboring nodes
2. Each factor „signs" ist color signature with its own color
3. Each node collects the signatures of its neighboring factors

Compression: Pass the colors around*

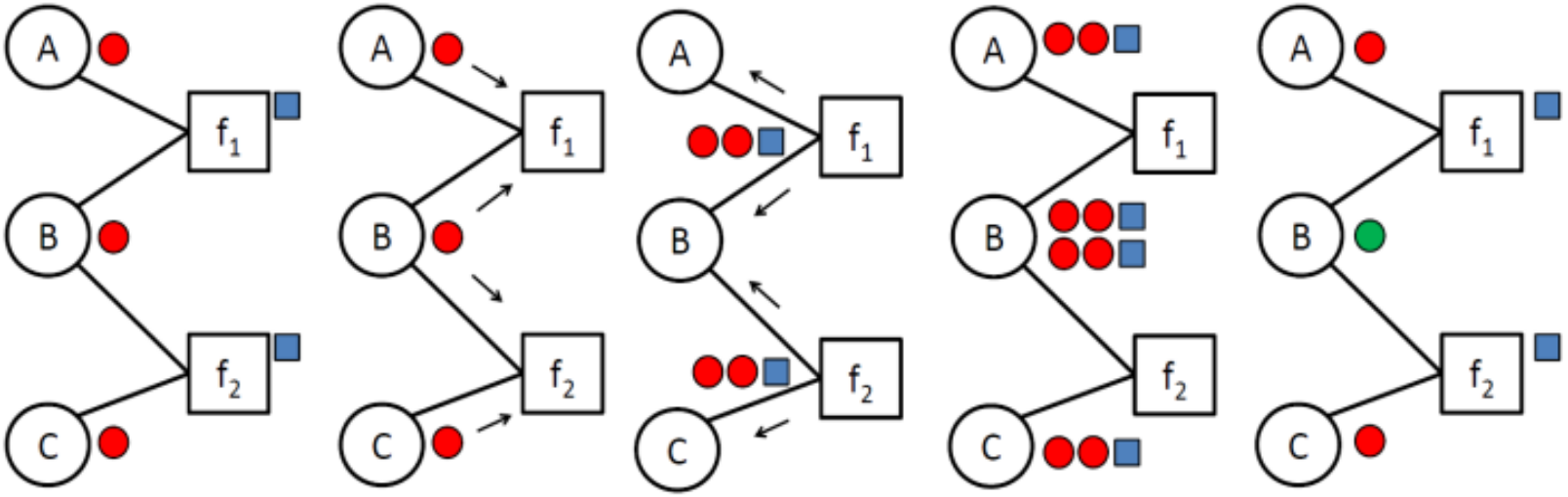
*can also be done at the „lifted", i.e., relational level



1. Each factor collects the colors of its neighboring nodes
2. Each factor „signs" ist color signature with its own color
3. Each node collects the signatures of its neighboring factors
4. Nodes are recolored according to the collected signatures

Compression: Pass the colors around*

*can also be done at the „lifted", i.e., relational level

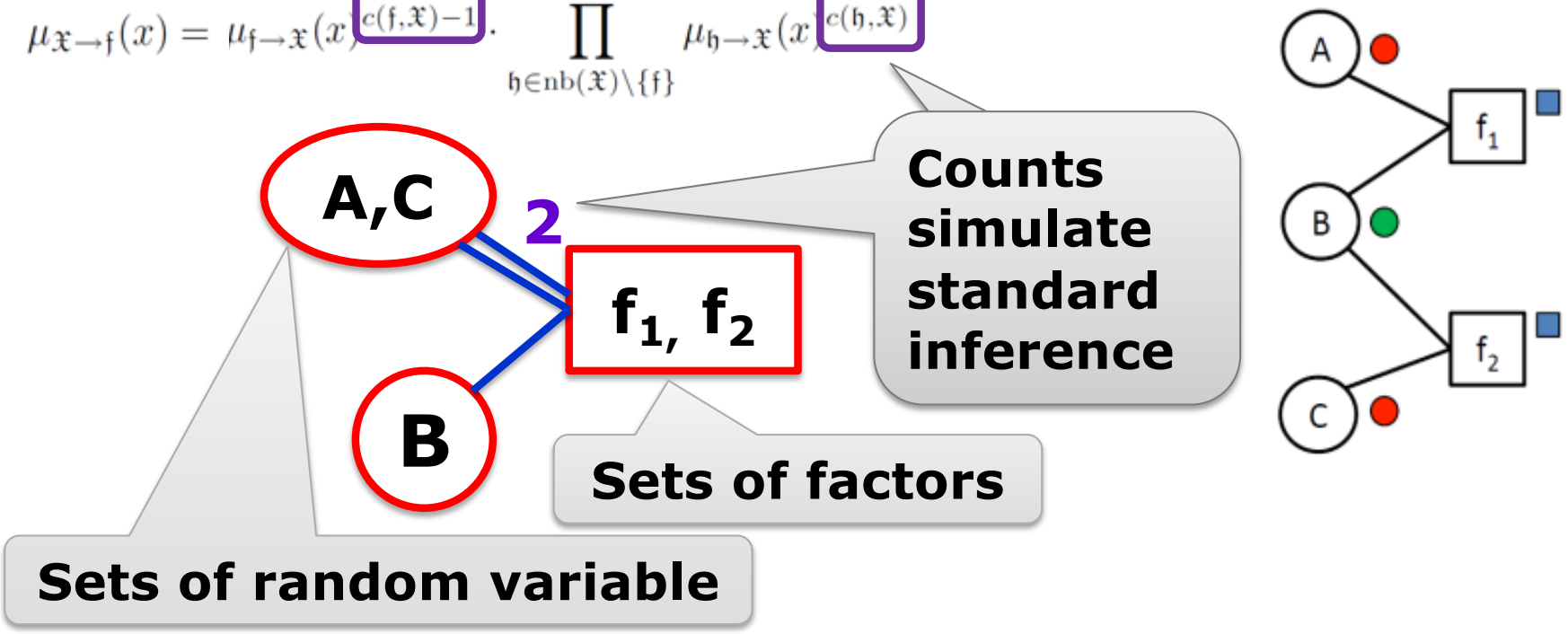


1. Each factor collects the colors of its neighboring nodes
2. Each factor „signs" ist color signature with its own color
3. Each node collects the signatures of its neighboring factors
4. Nodes are recolored according to the collected signatures
5. If no new color is created stop, otherwise go back to 1

Finally, we compute the quotient factor graph and run a modified loopy belief propagation

$$b_i(x_i) = \prod_{\mathfrak{f} \in \mathrm{nb}(\mathfrak{x}_i)} \mu_{\mathfrak{f} \to \mathfrak{x}_i}(x_i)^{c(\mathfrak{f},\mathfrak{x})}$$

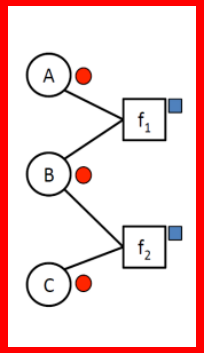
$$\mu_{\mathfrak{x} \to \mathfrak{f}}(x) = \mu_{\mathfrak{f} \to \mathfrak{x}}(x)^{c(\mathfrak{f},\mathfrak{x})-1} \cdot \prod_{\mathfrak{h} \in \mathrm{nb}(\mathfrak{x}) \setminus \{\mathfrak{f}\}} \mu_{\mathfrak{h} \to \mathfrak{x}}(x)^{c(\mathfrak{h},\mathfrak{x})}$$



Lifted Loopy Belief Propagation

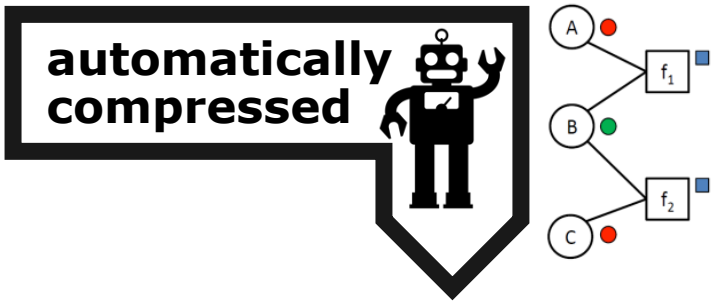
Exploiting computational symmetries

TECHNISCHE UNIVERSITÄT DARMSTADT

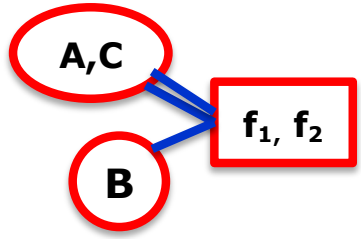


Big Model

quasi-linear time



Small Model



Run Loopy Belief Propagation

Run a modified Loopy Belief Propagation

[Singla, Domingos AAAI'08; Kersting, Ahmadi, Natarajan UAI'09; Ahmadi, Kersting, Mladenov, Natarajan MLJ'13; Van Haaren, Van den Broeck, Meert, Davis MLJ'16]

Compression can considerably speed up inference and training

Probabilistic inference using lifted (loopy) belief propagation

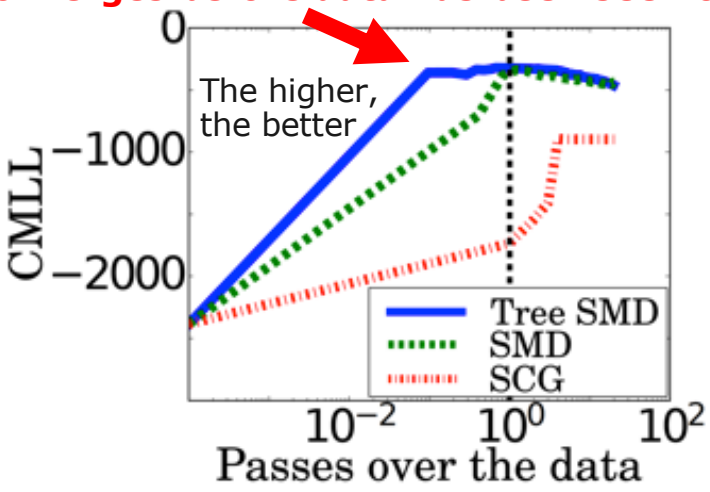
Domain	Time (in seconds) The lower, the better						No. of (Super) Features	
	Construction		BP		Total			
	Ground	Lifted	Ground	Lifted	Ground	Lifted	Ground	Lifted
Cora	263.1	1173.3	12368.4	3997.7	12631.6	5171.1	2078629	295468
UW-CSE	6.9	22.1	1015.8	602.5	1022.8	624.7	217665	86459
Friends & Smokers	38.8	89.7	10702.2	4.4	10741.0	94.2	1900905	58

114x faster

Parameter training using a lifted stochastic gradient

CORA entity resolution

converges before data has been seen once



State-of-the-art

What is going on algebraically?

Can we generalize this to other data science approaches?

WL computes a fractional automorphism of some matrix A

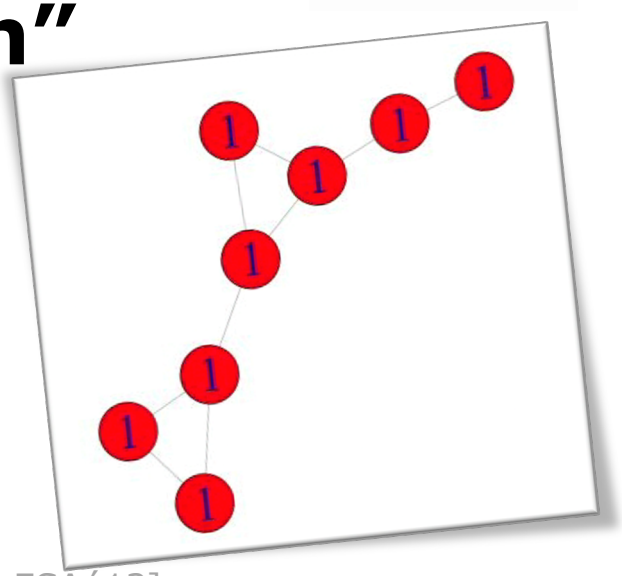
$$\mathbf{X}_Q \mathbf{A} = \mathbf{A} \mathbf{X}_P$$

where X_Q and X_p are doubly-stochastic matrixes (relaxed form of automorphism)

It turns out that color passing is well known in graph theory

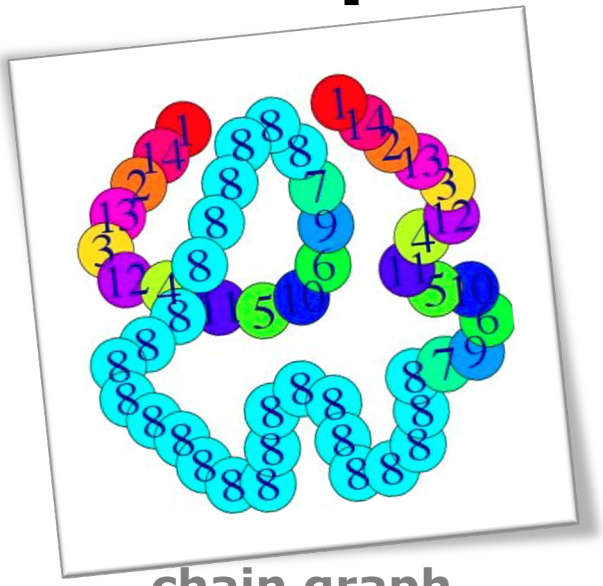
The Weisfeiler-Lehman Algorithm

Weisfeiler-Lehman (WL) Algorithmus aka "naive vertex classification"

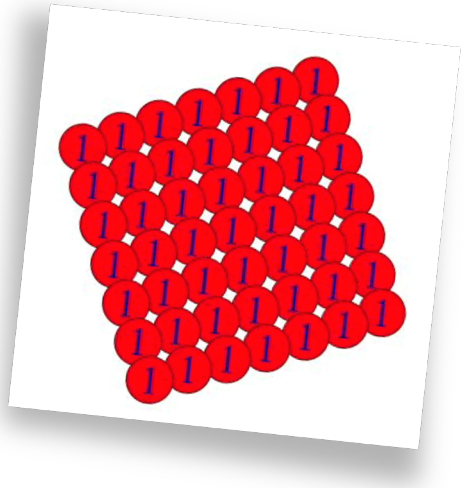


- Basic subroutine for GI testing
- Computes LP-relaxations of GA-ILP, aka. **fractional automorphisms**
- **Quasi-linear** running time $O((n+m)\log(n))$ when using asynchronous updates [Berkholz, Bonsma, Grohe ESA´13]
- **Part of graph tool SAUCY** [See e.g. Darga, Sakallah, Markov DAC´08]
- Has lead to highly performant graph kernels
  [Shervashidze, Schweitzer, van Leeuwen, Mehlhorn, Borgwardt JMLR 12:2539-2561 ´11]
- Can be extended to weighted graphs/real-valued matrices [Grohe, Kersting, Mladenov, Selman ESA´14]
- Actually a Frank-Wolfe optimizer and can be viewed as recursive spectral clustering
  [Kersting, Mladenov, Garnett, Grohe AAAI´14]

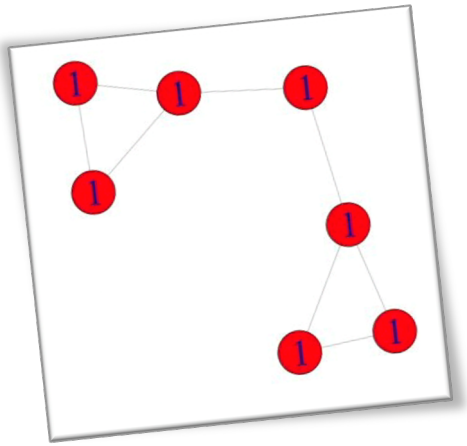
Examples



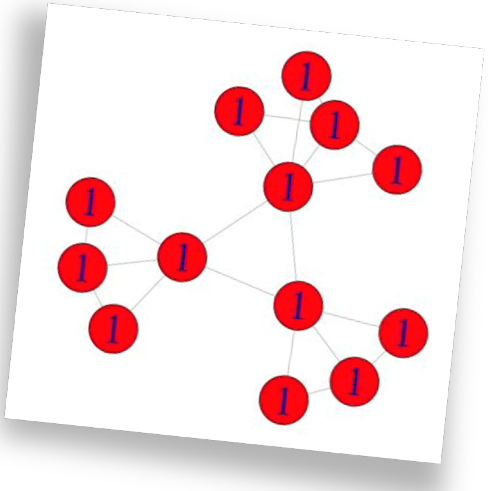
chain graph



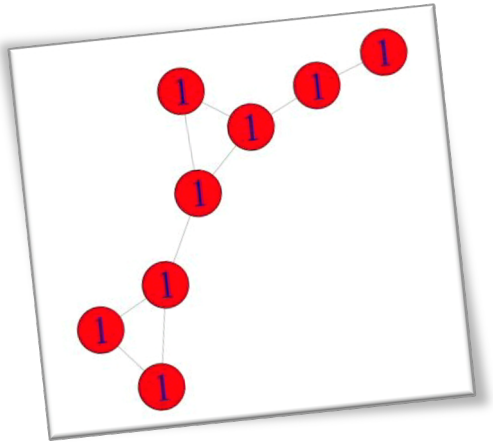
grid graph



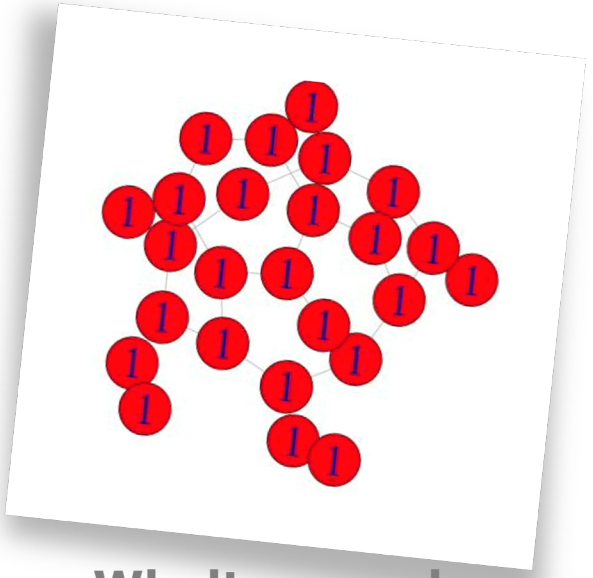
McKay graph



social rules



„walk" graph



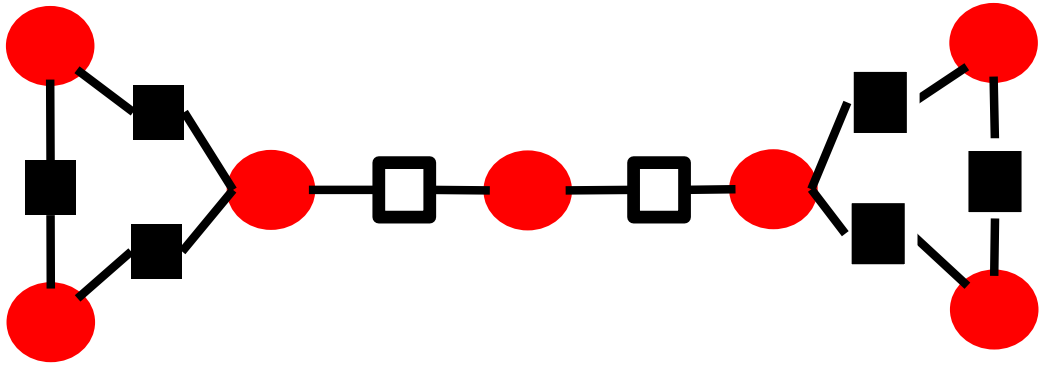
Whalter graph

Why quasi-linear running time?

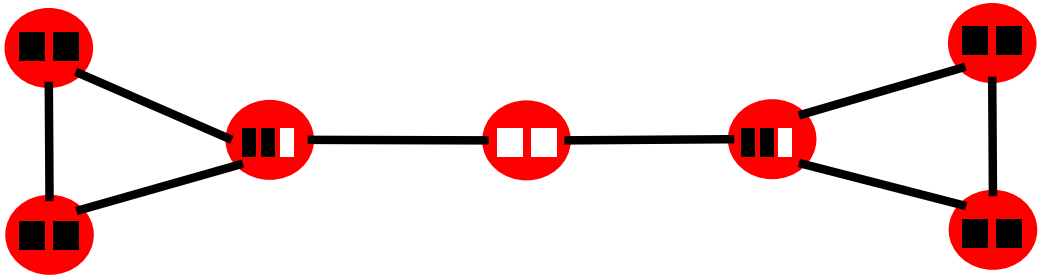
TECHNISCHE UNIVERSITÄT DARMSTADT

- Keep an active set of color classes and send color message asynchronously
  - Select an active color class but not the largest (halfing argument)
  - Send message to the neighboring nodes of the class
  - If a color class refines, make it activ.
- Due to the halfing argument, every node can become active only O(log(n)) times.
- Message update is of order m

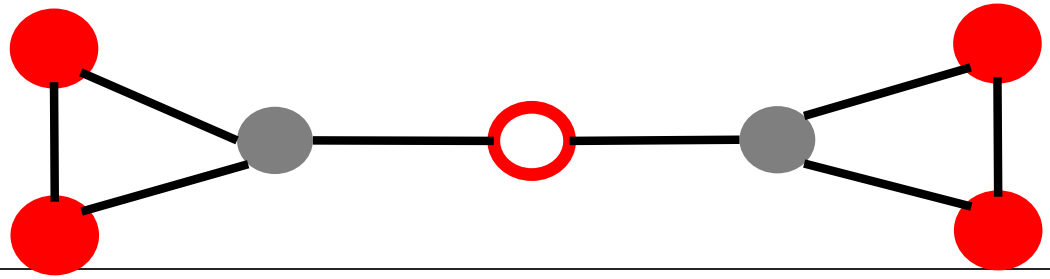
From Factor Graphs to Graphs



- Encode the factor colors into the node colors



- Then run Weisfeiler Lehman / Color Passing just on the graph



… more importantly, it paves the way to an algebraic understanding:

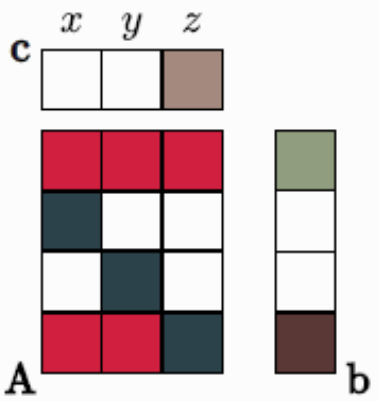
Instead of looking at ML through the glasses of probabilities over possible worlds, let's approach it using optimization

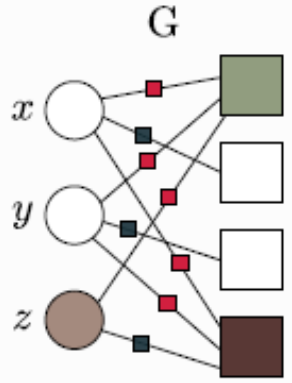
[Mladenov, Ahmadi, Kersting AISTATS´12, Grohe, Kersting, Mladenov, Selman ESA´14,Kersting, Mladenov, Tokmatov AIJ´15]

Lifted Linear Programming

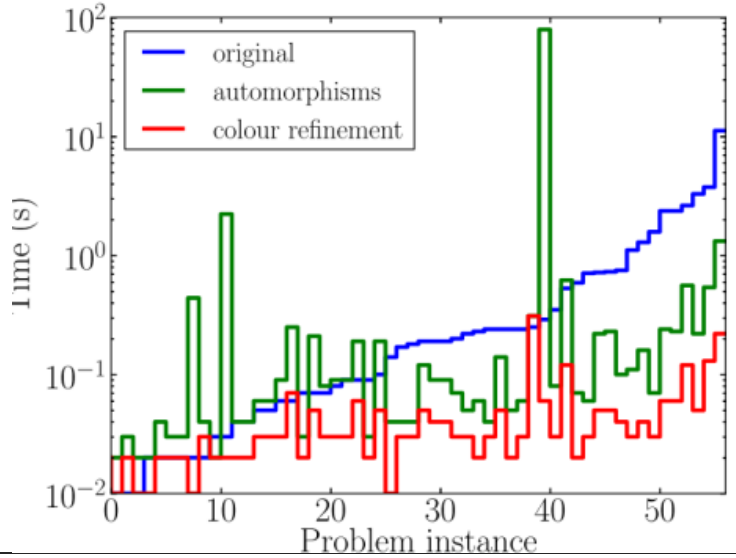
$$\max_{[x,y,z]^T \in \mathbb{R}^3} \quad 0x + 0y + 1z$$

$$\text{s.t.} \quad \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

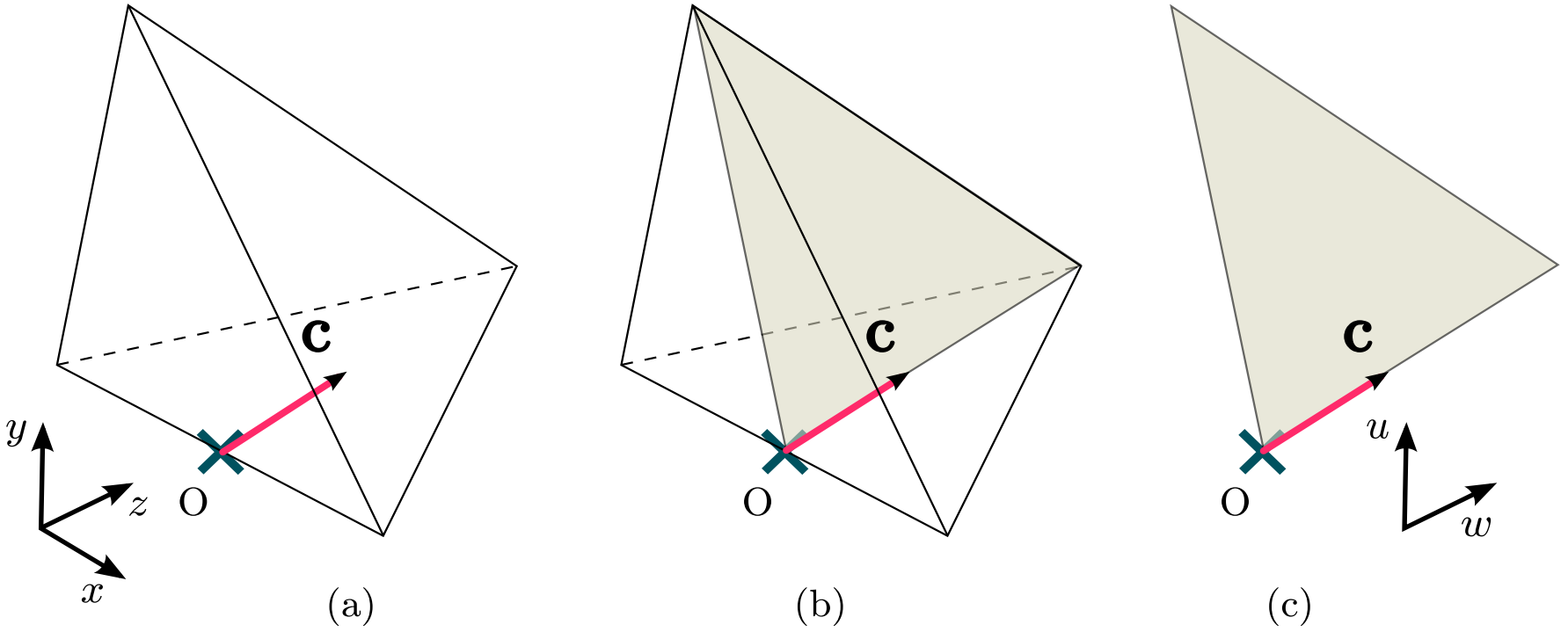




(1) Reduce the LP by running WL on the LP-Graph
(2) Run any solver on the (hopefully) smaller LP



quasi-linear overhead that may result in exponential speed up



Feasible region of LP and the objective vectors

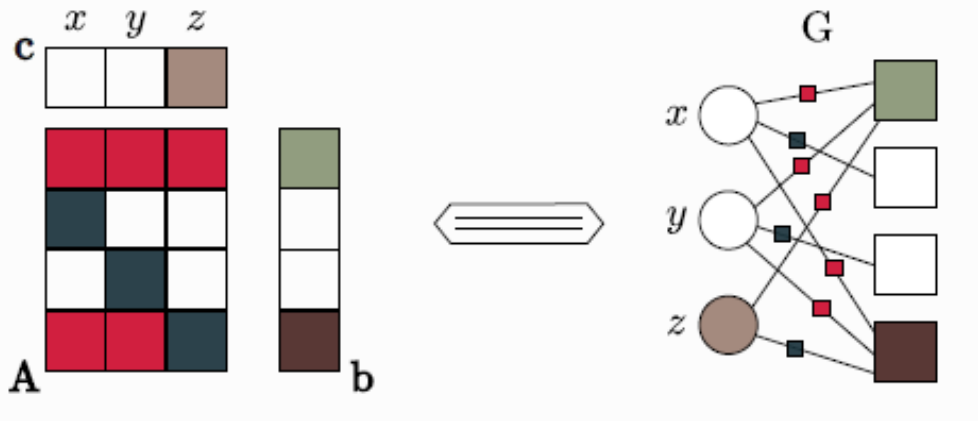
Span of the fractional auto-morpishm of the LP

Projections of the feasible region onto the span of the fractional auto-morphism

Why does this work?

[Mladenov, Ahmadi, Kersting AISTATS´12, Grohe, Kersting, Mladenov, Selman ESA´14, Kersting, Mladenov, Tokmatov AIJ´15]

Compute Equitable Partition (EP) of the LP using WL



$$\mathcal{P} = \{P_1, \ldots, P_p; Q_1, \ldots, Q_q\}$$

Partition of LP variables | **Partition of LP constraints**

Intuitively, we group together variables resp. constraints that interact in the very same way in the LP.

Fractional Automorphisms of LPs

TECHNISCHE UNIVERSITÄT DARMSTADT

The EP induces a fractional automorphism of the coefficient matrix **A**

$$\mathbf{X}_Q \mathbf{A} = \mathbf{A} \mathbf{X}_P$$

where $\mathbf{X}_Q$ and $\mathbf{X}_p$ are doubly-stochastic matrixes (relaxed form of automorphism)

$$(\mathbf{X}_P)_{ij} = \begin{cases} 1/|P| & \text{if both vertices } i,\ j \text{ are in the same } P, \\ 0 & \text{otherwise.} \end{cases}$$

$$(\mathbf{X}_Q)_{ij} = \begin{cases} 1/|Q| & \text{if both vertices } i,\ j \text{ are in the same } Q, \\ 0 & \text{otherwise} \end{cases}$$

Fractional Automorphisms Preserve Solutions

If $\mathbf{x}$ is feasible, then $\mathbf{X}_p\mathbf{x}$ is feasible, too.

By induction, one can show that left-multiplying with a double-stochastic matrix preserves directions of inequalities. Hence,

$$\mathbf{A}\mathbf{x} \leq \mathbf{b} \Rightarrow \mathbf{X}_Q\mathbf{A}\mathbf{x} \leq \mathbf{X}_Q\mathbf{b} \Leftrightarrow \mathbf{A}\mathbf{X}_P\mathbf{x} \leq \mathbf{b}$$

Fractional Automorphisms Preserve Solutions

If $\mathbf{x}^*$ is optimal, then $\mathbf{X}_p\mathbf{x}^*$ is optimal, too.

Since by construction $\mathbf{c}^T\mathbf{X}_P = \mathbf{c}^T$ and hence

$$\mathbf{c}^T(\mathbf{X}_P\mathbf{x}) = \mathbf{c}^T\mathbf{x}$$

What have we established so far?

Instead of considering the original LP

$$(\mathbf{A}, \mathbf{b}, \mathbf{c})$$

It is sufficient to consider

$$(\mathbf{A}\mathbf{X}_P, \mathbf{b}, \mathbf{X}_P{}^T\mathbf{c})$$

i.e. we "average" parts of the polytope.

But why is this dimensionality reduction?

Dimensionality Reduction

TECHNISCHE UNIVERSITÄT DARMSTADT

The doubly-stochastic matrix $\mathbf{X}_P$ can be written as

$$\mathbf{X}_P = \mathbf{B}\mathbf{B}^T$$

$$\mathbf{B}_{iP} = \begin{cases} \frac{1}{\sqrt{|P|}} & \text{if vertex } i \text{ belongs to part } P, \\ 0 & \text{otherwise.} \end{cases}$$

Since the column space of B is equivalent to the span of $\mathbf{X}_P$, it is actually sufficient to consider only

$$(\mathbf{A}\mathbf{B}_P, \mathbf{b}, \mathbf{B}_P^T\mathbf{c})$$

This is of reduced size, and actually we can also drop any constraint that becomes identical

$$\mathbf{X}_Q \mathbf{A} = \mathbf{A} \mathbf{X}_P$$

TECHNISCHE UNIVERSITÄT DARMSTADT

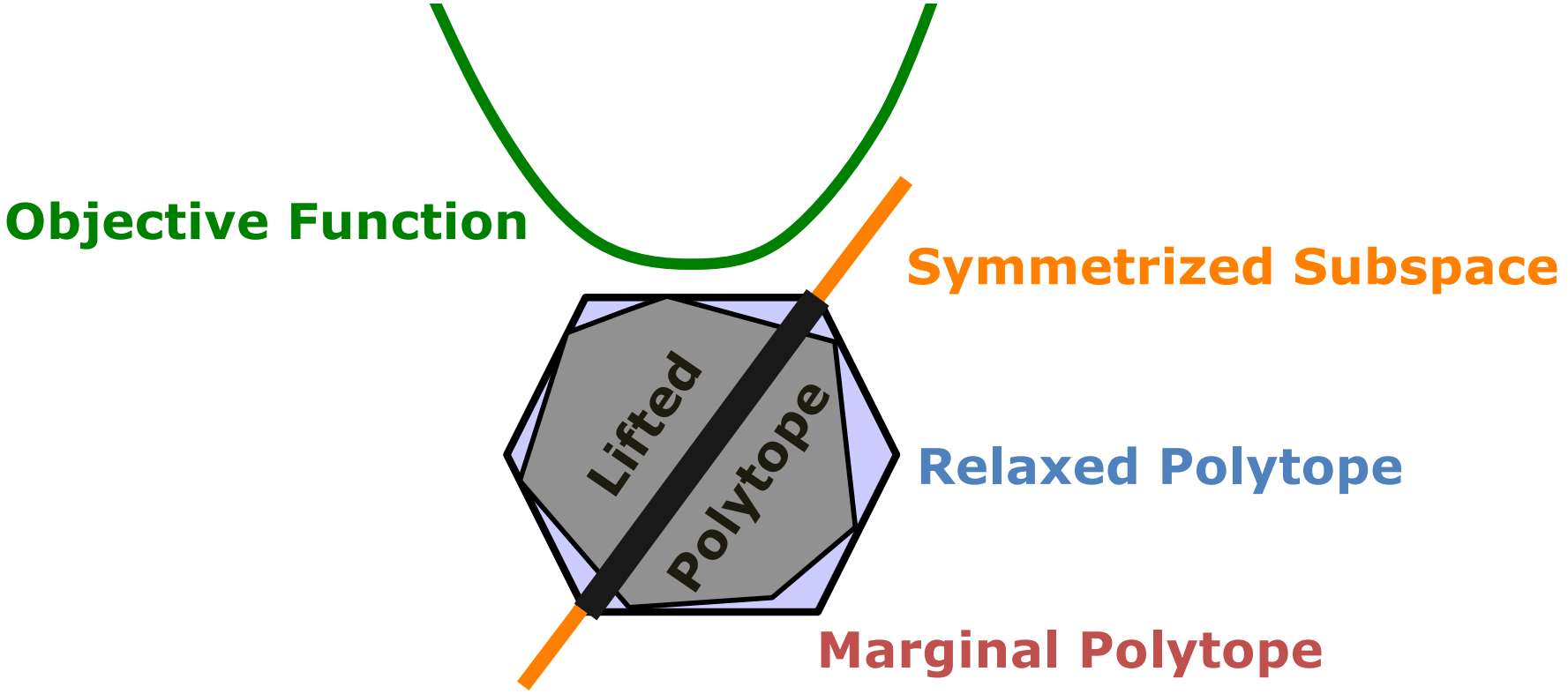
Fractional automorphisms provide an algebraic tool to study lifted inference

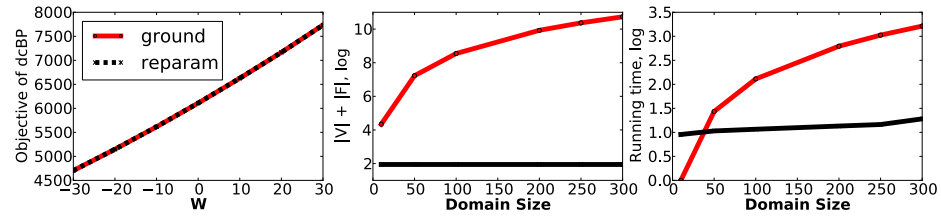
Actually, there is a whole body of work on (fractional) automorphisms for probabilistic inference, see the book, and we have focused here on the arguably simplest view.

This has resulted in a number of insights …

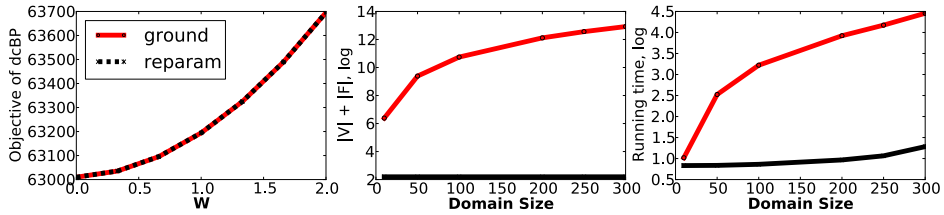
Approximate inference in graphical models is closely connected to linear programs

$$\hat{\mathbf{x}} \in \arg\max_{\mathbf{x}\in\mathcal{X}^N} \left\{ \sum_{s\in V} \theta_s(x_s) + \sum_{(s,t)\in E} \theta_{st}(x_s, x_t) \right\}$$

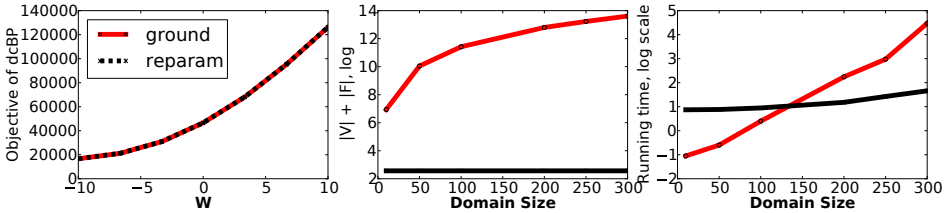




(a) Complete Graph MLN.



(b) Clique-Cycle MLN.



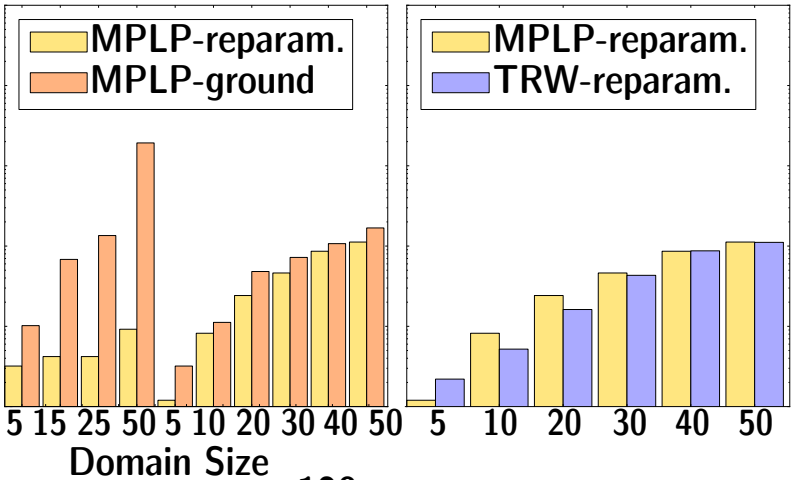
(c) Friends-smokers MLN.

Marginals

Any concave free energy is liftable

MAP

Any MAP-LP message-passing approach is liftable

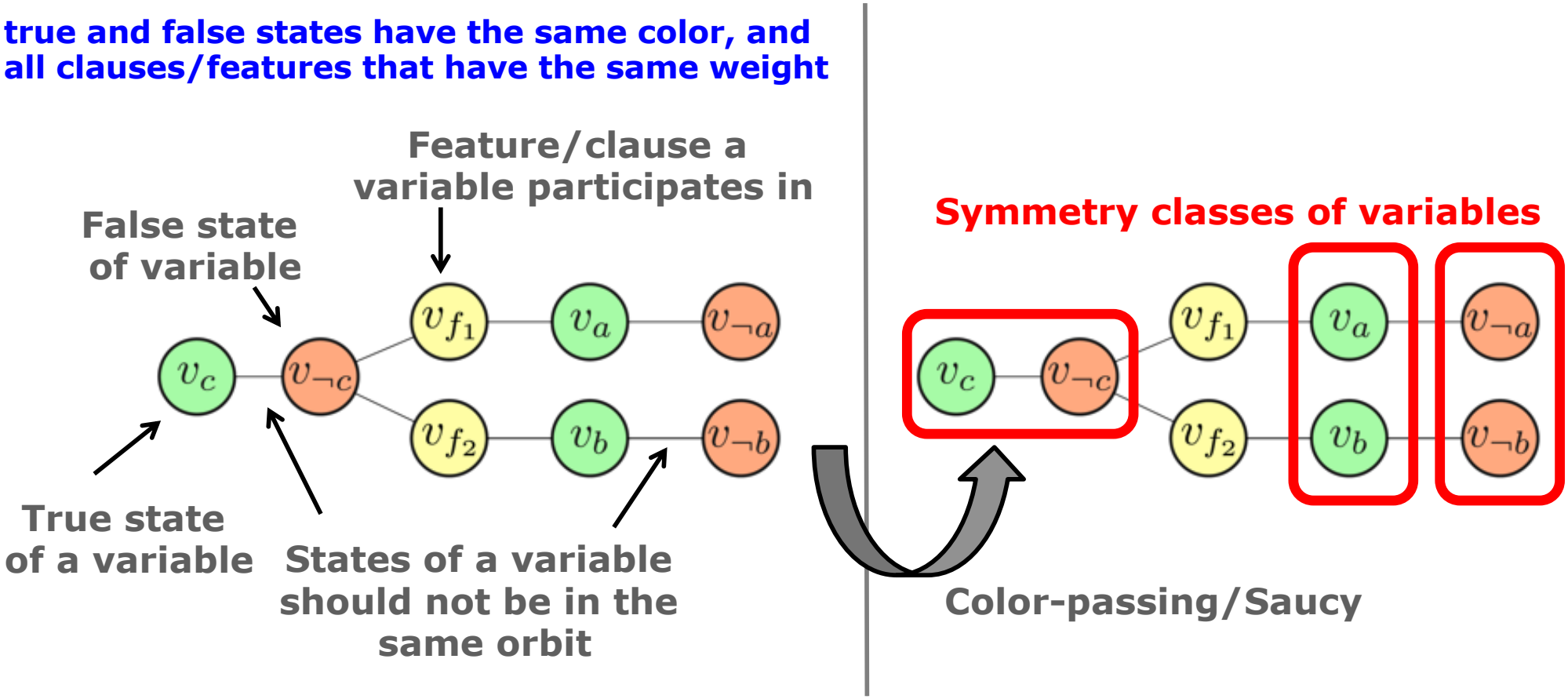


SYMMETRIES CAN ALSO BE EXPLOITED TO SPEED UP SAMPLING

Orbital Markov Chain Monte Carlo

[Niepert UAI 2012, Van den Broeck, Niepert AAAI 2015]

true and false states have the same color, and all clauses/features that have the same weight



Jump between symmetric states uniformly

Orbital MCMC Sampling

[Niepert UAI 2012, Van den Broeck, Niepert AAAI 2015]

TECHNISCHE UNIVERSITÄT DARMSTADT

In each sampling iteration:

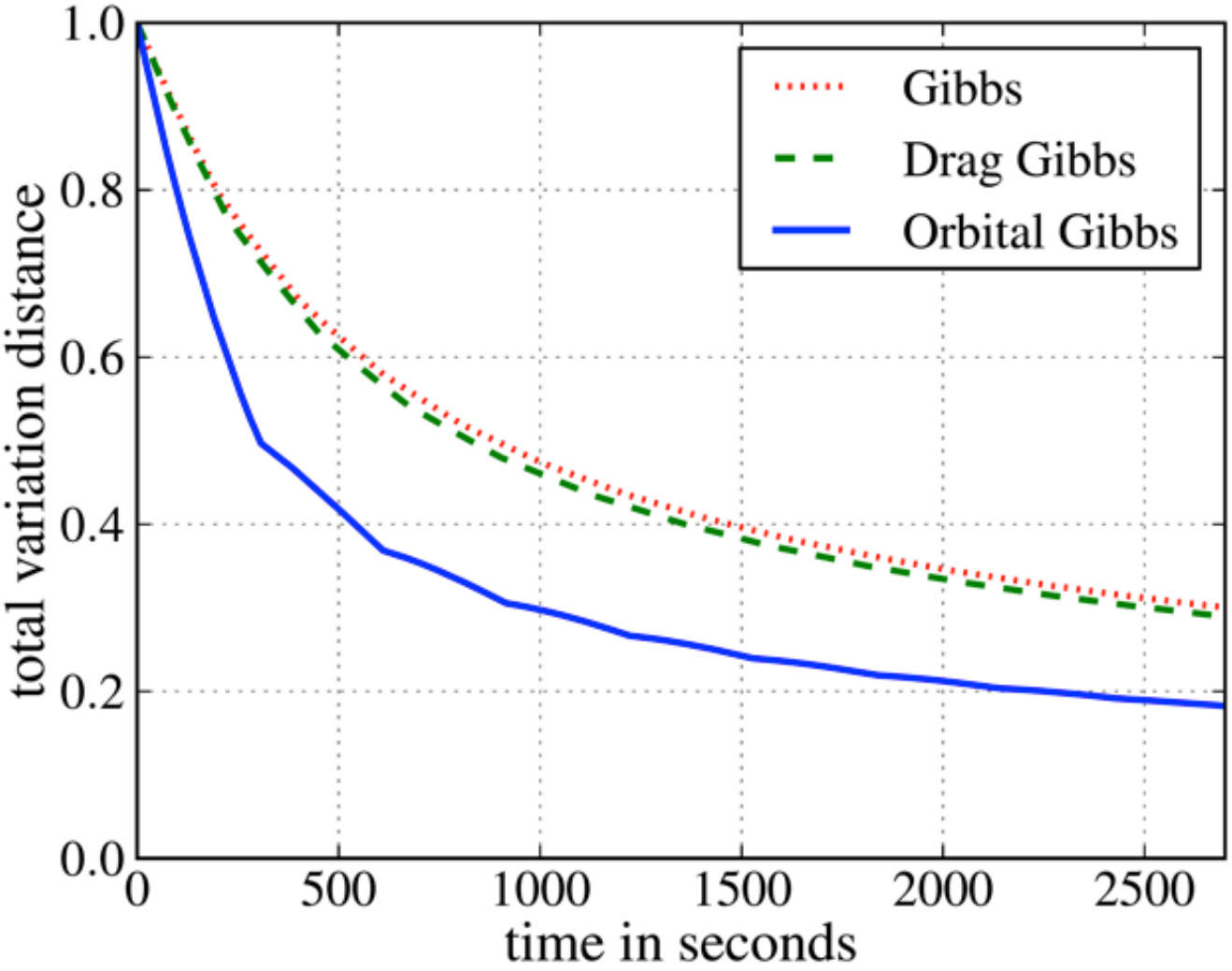
1. **run a step of a traditional MCMC chain TM first and then**

2. **sample the state of M at the next time uniformly at random from the orbit of the state of the original chain TM at time t, i.e., select an equivalent state uniformly at random**

[Niepert UAI 2012, Van den Broeck, Niepert AAAI 2015]

TECHNISCHE UNIVERSITÄT DARMSTADT

Orbital MCMC on a 6x6 Ising grid



[Niepert UAI 2012, Van den Broeck, Niepert AAAI 2015]

TECHNISCHE UNIVERSITÄT DARMSTADT

Lifted Metropolis-Hastings

Given an orbital Metropolis chain A:

- Given symmetry group G (approx. symmetries)
- Orbit $\mathbf{x}^G$ contains all states approx. symm. to **x**
- In state **x**:

Color-passing /Saucy

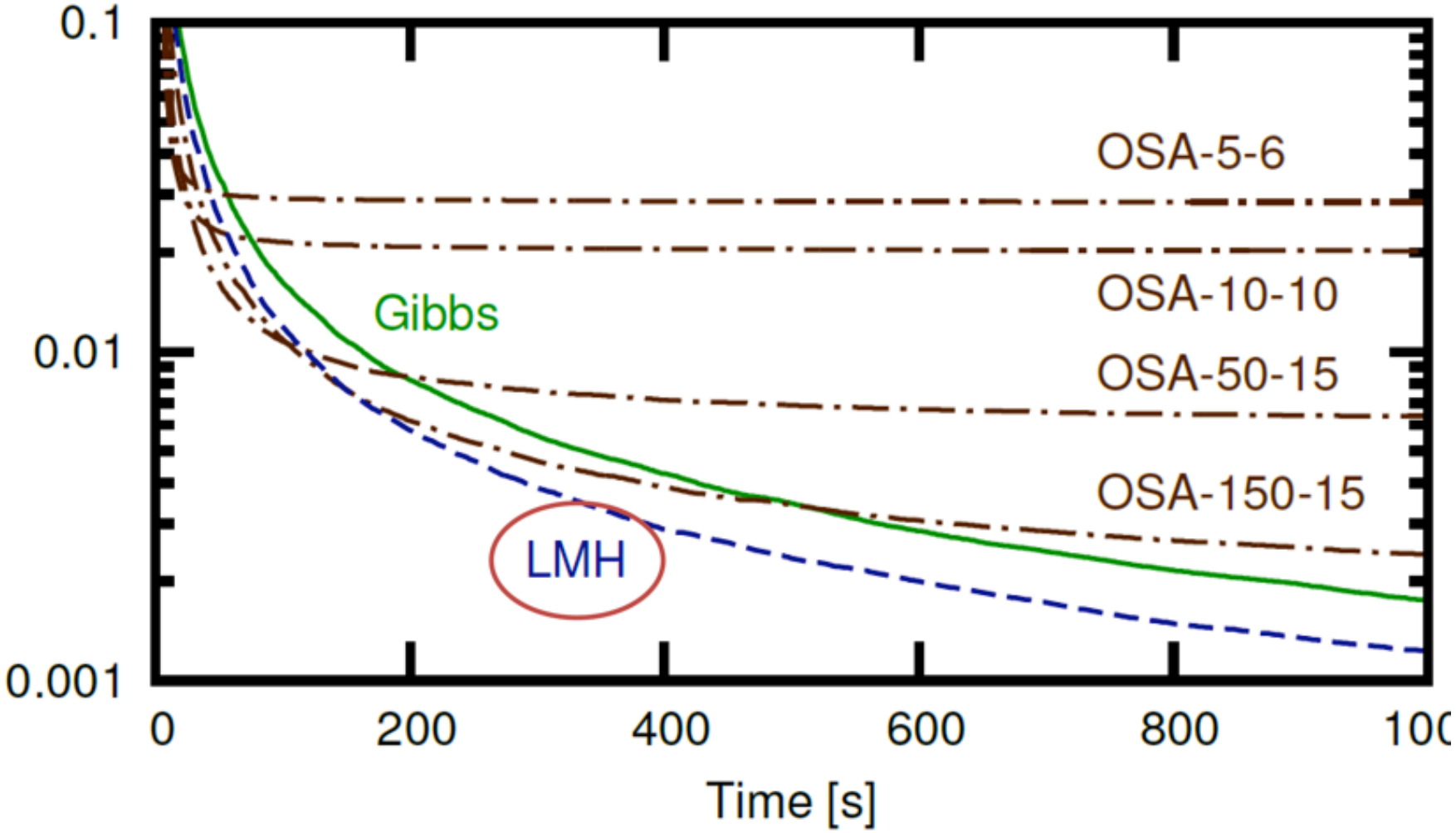
1. Select **y** uniformly at random from $\mathbf{x}^G$
2. Move from **x** to **y** with probability $\min\left(\frac{\Pr(y)}{\Pr(x)}, 1\right)$
3. Otherwise: stay in **x** (reject)
4. Repeat

and an ordinary (base) Markov chain B, with prob. α follow B and with (1-α) follow A

This can also account for evidence that may break symmetries, using e.g. approx. symmetries

Lifted Metropolis-Hastings on WebKB

[Niepert UAI 2012, Van den Broeck, Niepert AAAI 2015]



What have we learnt?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Lifted inference exploits (fractional) symmetries
- Fractional symmetries can be computed in quasi-linear time
- Symmetries allow one to study lifted inference in an algebraic way, i.e., independent of the underlying algorithm
- Essentially, the whole family of approximate inference methods is liftable