Efficient Detection of Commutative Factors in Factor Graphs

PGM 2024 – Nijmegen, Netherlands

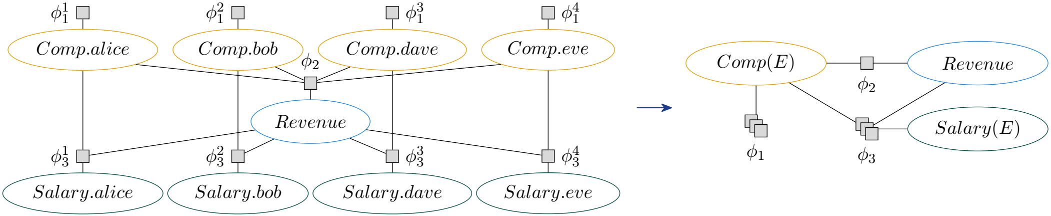
Malte Luttermann[1], Johann Machemer[2], and Marcel Gehrke[3]

[1]German Research Center for Artificial Intelligence (DFKI), Lübeck, Germany
[2]Institute for Software Engineering and Programming Languages, University of Lübeck, Germany
[3]Institute for Humanities-Centered Artificial Intelligence, University of Hamburg, Germany

September 12, 2024

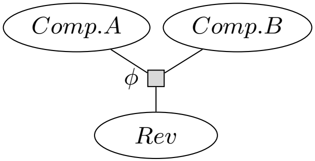
Motivation

- Lifting exploits symmetries in factor graphs to speed up probabilistic inference
- Lifting uses a representative of indistinguishable individuals for computations



Problem Setup I

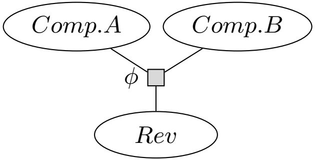
- Goal: Efficiently detect commutative factors
- Commutative factors have symmetries within themselves due to their arguments being exchangeable



$Comp.A$	$Comp.B$	Rev	$\phi(Comp.A, Comp.B, Rev)$
true	true	true	φ_1
true	true	false	φ_4
true	false	true	φ_2
true	false	false	φ_5
false	true	true	φ_2
false	true	false	φ_5
false	false	true	φ_3
false	false	false	φ_6

Problem Setup I

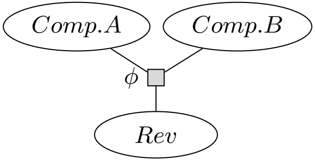
- Goal: Efficiently detect commutative factors
- Commutative factors have symmetries within themselves due to their arguments being exchangeable



$Comp.A$	$Comp.B$	Rev	$\phi(Comp.A, Comp.B, Rev)$
true	true	true	φ_1
true	true	false	φ_4
true	false	true	φ_2
true	false	false	φ_5
false	true	true	φ_2
false	true	false	φ_5
false	false	true	φ_3
false	false	false	φ_6

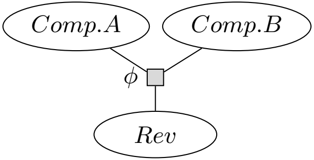
Problem Setup I

- Goal: Efficiently detect commutative factors
- Commutative factors have symmetries within themselves due to their arguments being exchangeable



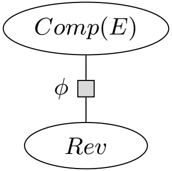
$Comp.A$	$Comp.B$	Rev	$\phi(Comp.A, Comp.B, Rev)$
true	true	true	φ_1
true	true	false	φ_4
true	false	true	φ_2
true	false	false	φ_5
false	true	true	φ_2
false	true	false	φ_5
false	false	true	φ_3
false	false	false	φ_6

Problem Setup II



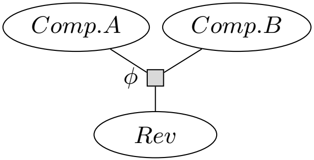
$Comp.A$	$Comp.B$	Rev	$\phi(Comp.A, Comp.B, Rev)$
true	true	true	φ_1
true	true	false	φ_4
true	false	true	φ_2
true	false	false	φ_5
false	true	true	φ_2
false	true	false	φ_5
false	false	true	φ_3
false	false	false	φ_6

$\downarrow$



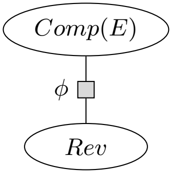
$\#_E[Comp(E)]$	Rev	$\phi(\#_E[Comp(E)], Rev)$
$[2, 0]$	true	φ_1
$[1, 1]$	true	φ_2
$[0, 2]$	true	φ_3
$[2, 0]$	false	φ_4
$[1, 1]$	false	φ_5
$[0, 2]$	false	φ_6

Problem Setup II



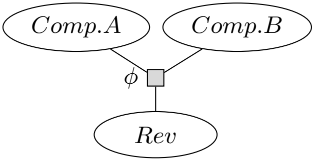
$Comp.A$	$Comp.B$	Rev	$\phi(Comp.A, Comp.B, Rev)$
true	true	true	φ_1
true	true	false	φ_4
true	false	true	φ_2
true	false	false	φ_5
false	true	true	φ_2
false	true	false	φ_5
false	false	true	φ_3
false	false	false	φ_6

$\downarrow$



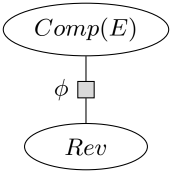
$\#_E[Comp(E)]$	Rev	$\phi(\#_E[Comp(E)], Rev)$
$[2, 0]$	true	φ_1
$[1, 1]$	true	φ_2
$[0, 2]$	true	φ_3
$[2, 0]$	false	φ_4
$[1, 1]$	false	φ_5
$[0, 2]$	false	φ_6

Problem Setup II



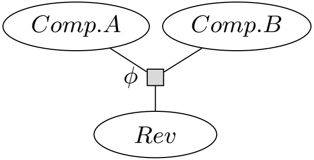
$Comp.A$	$Comp.B$	Rev	$\phi(Comp.A, Comp.B, Rev)$
true	true	true	φ_1
true	true	false	φ_4
true	false	true	φ_2
true	false	false	φ_5
false	true	true	φ_2
false	true	false	φ_5
false	false	true	φ_3
false	false	false	φ_6

$\downarrow$



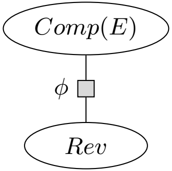
$\#_E[Comp(E)]$	Rev	$\phi(\#_E[Comp(E)], Rev)$
$[2, 0]$	true	φ_1
$[1, 1]$	true	φ_2
$[0, 2]$	true	φ_3
$[2, 0]$	false	φ_4
$[1, 1]$	false	φ_5
$[0, 2]$	false	φ_6

Problem Setup II



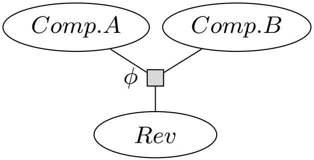
$Comp.A$	$Comp.B$	Rev	$\phi(Comp.A, Comp.B, Rev)$
true	true	true	φ_1
true	true	false	φ_4
true	false	true	φ_2
true	false	false	φ_5
false	true	true	φ_2
false	true	false	φ_5
false	false	true	φ_3
false	false	false	φ_6

$\downarrow$

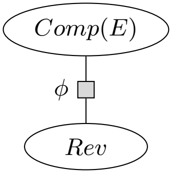


$\#_E[Comp(E)]$	Rev	$\phi(\#_E[Comp(E)], Rev)$
$[2, 0]$	true	φ_1
$[1, 1]$	true	φ_2
$[0, 2]$	true	φ_3
$[2, 0]$	false	φ_4
$[1, 1]$	false	φ_5
$[0, 2]$	false	φ_6

Problem Setup II

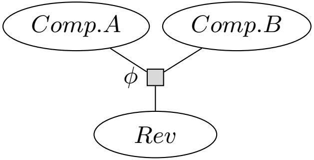


$Comp.A$	$Comp.B$	Rev	$\phi(Comp.A, Comp.B, Rev)$
true	true	true	φ_1
true	true	false	φ_4
true	false	true	φ_2
true	false	false	φ_5
false	true	true	φ_2
false	true	false	φ_5
false	false	true	φ_3
false	false	false	φ_6



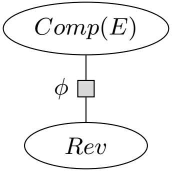
$\#_E[Comp(E)]$	Rev	$\phi(\#_E[Comp(E)], Rev)$
$[2, 0]$	true	φ_1
$[1, 1]$	true	φ_2
$[0, 2]$	true	φ_3
$[2, 0]$	false	φ_4
$[1, 1]$	false	φ_5
$[0, 2]$	false	φ_6

Problem Setup II



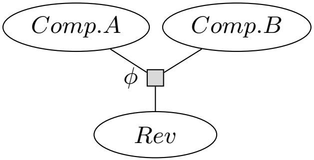
$Comp.A$	$Comp.B$	Rev	$\phi(Comp.A, Comp.B, Rev)$
true	true	true	φ_1
true	true	false	φ_4
true	false	true	φ_2
true	false	false	φ_5
false	true	true	φ_2
false	true	false	φ_5
false	false	true	φ_3
false	false	false	φ_6

$\downarrow$



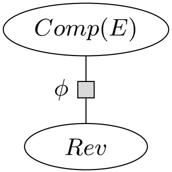
$\#_E[Comp(E)]$	Rev	$\phi(\#_E[Comp(E)], Rev)$
$[2, 0]$	true	φ_1
$[1, 1]$	true	φ_2
$[0, 2]$	true	φ_3
$[2, 0]$	false	φ_4
$[1, 1]$	false	φ_5
$[0, 2]$	false	φ_6

Problem Setup II



$Comp.A$	$Comp.B$	Rev	$\phi(Comp.A, Comp.B, Rev)$
true	true	true	φ_1
true	true	false	φ_4
true	false	true	φ_2
true	false	false	φ_5
false	true	true	φ_2
false	true	false	φ_5
false	false	true	φ_3
false	false	false	φ_6

$\downarrow$



$\#_E[Comp(E)]$	Rev	$\phi(\#_E[Comp(E)], Rev)$
$[2, 0]$	true	φ_1
$[1, 1]$	true	φ_2
$[0, 2]$	true	φ_3
$[2, 0]$	false	φ_4
$[1, 1]$	false	φ_5
$[0, 2]$	false	φ_6

Previous Work and Our Contributions

Previous work (Luttermann et al., 2024):

- Advanced Colour Passing algorithm to construct a lifted representation
- Iterate over all subsets of arguments to find commutative arguments
- Number of iterations for a factor with n arguments is in $O(2^n)$

Malte Luttermann et al. (2024). »Colour Passing Revisited: Lifted Model Construction with Commutative Factors«. *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-2024).* AAAI Press, pp. 20500–20507.

Previous Work and Our Contributions

Previous work (Luttermann et al., 2024):

- Advanced Colour Passing algorithm to construct a lifted representation
- Iterate over all subsets of arguments to find commutative arguments
- Number of iterations for a factor with n arguments is in $O(2^n)$

Our contributions:

- Theoretical guarantees: *Buckets* to avoid iterating over all subsets
- Practical algorithm: Detection of Commutative Factors (DECOR) algorithm

Malte Luttermann et al. (2024). »Colour Passing Revisited: Lifted Model Construction with Commutative Factors«. *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-2024)*. AAAI Press, pp. 20500–20507.

Buckets

- Buckets count the occurrences of specific range values in an assignment
- Each potential belongs to *exactly one* bucket
- Each bucket contains *at least one* potential

A	B	R	$\phi(A,B,R)$	b
true	true	true	φ_1	$[3,0]$
true	true	false	φ_4	$[2,1]$
true	false	true	φ_2	$[2,1]$
true	false	false	φ_5	$[1,2]$
false	true	true	φ_2	$[2,1]$
false	true	false	φ_5	$[1,2]$
false	false	true	φ_3	$[1,2]$
false	false	false	φ_6	$[0,3]$

b	$\phi(b)$
$[3,0]$	$\langle\varphi_1\rangle$
$[2,1]$	$\langle\varphi_4,\varphi_2,\varphi_2\rangle$
$[1,2]$	$\langle\varphi_5,\varphi_5,\varphi_3\rangle$
$[0,3]$	$\langle\varphi_6\rangle$

Properties of Buckets I

For any subset $\boldsymbol{S}$ of commutative arguments:

- $|\boldsymbol{S}| \leq \min_{b \in \{b \mid b \in \mathcal{B}(\phi) \wedge |\phi(b)|>1\}} \max_{\varphi \in \phi(b)} \operatorname{count}(\phi(b), \varphi)$, i.e.,
- in each bucket b with $|\phi(b)| > 1$, there is a potential occurring at least $|\boldsymbol{S}|$ times

A	B	R	$\phi(A, B, R)$	b
true	true	true	φ_1	$[3, 0]$
true	true	false	φ_4	$[2, 1]$
true	false	true	φ_2	$[2, 1]$
true	false	false	φ_5	$[1, 2]$
false	true	true	φ_2	$[2, 1]$
false	true	false	φ_5	$[1, 2]$
false	false	true	φ_3	$[1, 2]$
false	false	false	φ_6	$[0, 3]$

b	$\phi(b)$
$[3, 0]$	$\langle \varphi_1 \rangle$
$[2, 1]$	$\langle \varphi_4, \varphi_2, \varphi_2 \rangle$
$[1, 2]$	$\langle \varphi_5, \varphi_5, \varphi_3 \rangle$
$[0, 3]$	$\langle \varphi_6 \rangle$

Properties of Buckets II

For a group of identical potentials in a bucket:

- Intersection of their corresponding assignments yields candidates
- E.g., for φ_2's in $[2, 1]$: $(\text{true}, \text{false}, \text{true}) \cap (\text{false}, \text{true}, \text{true}) = (\emptyset, \emptyset, \text{true})$

A	B	R	$\phi(A, B, R)$	b
true	true	true	φ_1	$[3, 0]$
true	true	false	φ_4	$[2, 1]$
true	false	true	φ_2	$[2, 1]$
true	false	false	φ_5	$[1, 2]$
false	true	true	φ_2	$[2, 1]$
false	true	false	φ_5	$[1, 2]$
false	false	true	φ_3	$[1, 2]$
false	false	false	φ_6	$[0, 3]$

b	$\phi(b)$
$[3, 0]$	$\langle \varphi_1 \rangle$
$[2, 1]$	$\langle \varphi_4, \varphi_2, \varphi_2 \rangle$
$[1, 2]$	$\langle \varphi_5, \varphi_5, \varphi_3 \rangle$
$[0, 3]$	$\langle \varphi_6 \rangle$

The DECOR Algorithm

- Iterate over buckets and compute candidates for commutative arguments

A	B	R	$\phi(A,B,R)$	b
true	true	true	φ_1	$[3,0]$
true	true	false	φ_4	$[2,1]$
true	false	true	φ_2	$[2,1]$
true	false	false	φ_5	$[1,2]$
false	true	true	φ_2	$[2,1]$
false	true	false	φ_5	$[1,2]$
false	false	true	φ_3	$[1,2]$
false	false	false	φ_6	$[0,3]$

b	$\phi(b)$
$[3,0]$	$\langle\varphi_1\rangle$
$[2,1]$	$\langle\varphi_4,\varphi_2,\varphi_2\rangle$
$[1,2]$	$\langle\varphi_5,\varphi_5,\varphi_3\rangle$
$[0,3]$	$\langle\varphi_6\rangle$

Initial candidates: $\{\{A,B,R\}\}$

The DECOR Algorithm

- Iterate over buckets and compute candidates for commutative arguments

A	B	R	$\phi(A, B, R)$	b
true	true	true	φ_1	$[3, 0]$
true	true	false	φ_4	$[2, 1]$
true	false	true	φ_2	$[2, 1]$
true	false	false	φ_5	$[1, 2]$
false	true	true	φ_2	$[2, 1]$
false	true	false	φ_5	$[1, 2]$
false	false	true	φ_3	$[1, 2]$
false	false	false	φ_6	$[0, 3]$

b	$\phi(b)$
$[3, 0]$	$\langle \varphi_1 \rangle$
$[2, 1]$	$\langle \varphi_4, \varphi_2, \varphi_2 \rangle$
$[1, 2]$	$\langle \varphi_5, \varphi_5, \varphi_3 \rangle$
$[0, 3]$	$\langle \varphi_6 \rangle$

Initial candidates: $\{\{A, B, R\}\}$

$[3, 0]$ $(|\phi(b)| < 2)$: $\{\{A, B, R\}\}$

The DECOR Algorithm

- Iterate over buckets and compute candidates for commutative arguments

A	B	R	$\phi(A,B,R)$	b
true	true	true	φ_1	$[3,0]$
true	true	false	φ_4	$[2,1]$
true	false	true	φ_2	$[2,1]$
true	false	false	φ_5	$[1,2]$
false	true	true	φ_2	$[2,1]$
false	true	false	φ_5	$[1,2]$
false	false	true	φ_3	$[1,2]$
false	false	false	φ_6	$[0,3]$

b	$\phi(b)$
$[3,0]$	$\langle\varphi_1\rangle$
$[2,1]$	$\langle\varphi_4,\varphi_2,\varphi_2\rangle$
$[1,2]$	$\langle\varphi_5,\varphi_5,\varphi_3\rangle$
$[0,3]$	$\langle\varphi_6\rangle$

$$
\begin{aligned}
& (\mathrm{true}, \mathrm{false}, \mathrm{true}) \\
\cap\ & (\mathrm{false}, \mathrm{true}, \mathrm{true}) \\
=\ & (\emptyset, \quad \emptyset, \quad \mathrm{true})
\end{aligned}
$$

Initial candidates: $\{\{A,B,R\}\}$

$[3,0]$ $(|\phi(b)|<2)$: $\{\{A,B,R\}\}$

$[2,1]$: $\{\{A,B\}\}$

The DECOR Algorithm

- Iterate over buckets and compute candidates for commutative arguments

A	B	R	$\phi(A, B, R)$	b
true	true	true	φ_1	$[3, 0]$
true	true	false	φ_4	$[2, 1]$
true	false	true	φ_2	$[2, 1]$
true	false	false	φ_5	$[1, 2]$
false	true	true	φ_2	$[2, 1]$
false	true	false	φ_5	$[1, 2]$
false	false	true	φ_3	$[1, 2]$
false	false	false	φ_6	$[0, 3]$

b	$\phi(b)$
$[3, 0]$	$\langle \varphi_1 \rangle$
$[2, 1]$	$\langle \varphi_4, \varphi_2, \varphi_2 \rangle$
$[1, 2]$	$\langle \varphi_5, \varphi_5, \varphi_3 \rangle$
$[0, 3]$	$\langle \varphi_6 \rangle$

$$
\begin{array}{rl}
 & (\text{true}, \text{false}, \text{false}) \\
\cap & (\text{false}, \text{true}, \text{false}) \\
= & (\emptyset, \quad \emptyset, \quad \text{false})
\end{array}
$$

Initial candidates: $\{\{A, B, R\}\}$

$[3, 0]$ $(|\phi(b)| < 2)$: $\{\{A, B, R\}\}$

$[2, 1]$: $\{\{A, B\}\}$

$[1, 2]$: $\{\{A, B\}\}$

The DECOR Algorithm

- Iterate over buckets and compute candidates for commutative arguments

A	B	R	$\phi(A, B, R)$	b
true	true	true	φ_1	$[3, 0]$
true	true	false	φ_4	$[2, 1]$
true	false	true	φ_2	$[2, 1]$
true	false	false	φ_5	$[1, 2]$
false	true	true	φ_2	$[2, 1]$
false	true	false	φ_5	$[1, 2]$
false	false	true	φ_3	$[1, 2]$
false	false	false	φ_6	$[0, 3]$

b	$\phi(b)$
$[3, 0]$	$\langle \varphi_1 \rangle$
$[2, 1]$	$\langle \varphi_4, \varphi_2, \varphi_2 \rangle$
$[1, 2]$	$\langle \varphi_5, \varphi_5, \varphi_3 \rangle$
$[0, 3]$	$\langle \varphi_6 \rangle$

Initial candidates:	$\{\{A, B, R\}\}$
$[3, 0]$ $(\lvert\phi(b)\rvert < 2)$:	$\{\{A, B, R\}\}$
$[2, 1]$:	$\{\{A, B\}\}$
$[1, 2]$:	$\{\{A, B\}\}$
$[0, 3]$ $(\lvert\phi(b)\rvert < 2)$:	$\{\{A, B\}\}$

Theoretical Guarantees of the DECOR Algorithm

- Avoids »naive« iteration over all 2^n subsets of arguments
- Time complexity is upper-bounded depending on the number of groups of identical potentials in the buckets

b	$\phi(b)$
$[3,0]$	$\langle \varphi_1 \rangle$
$[2,1]$	$\langle \varphi_4, \varphi_2, \varphi_2 \rangle$
$[1,2]$	$\langle \varphi_5, \varphi_5, \varphi_3 \rangle$
$[0,3]$	$\langle \varphi_6 \rangle$

Theoretical Guarantees of the DECOR Algorithm

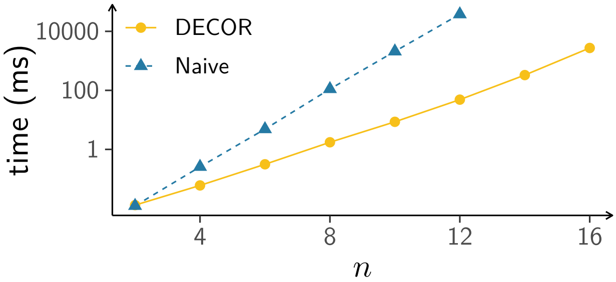
- Avoids »naive« iteration over all 2^n subsets of arguments
- Time complexity is upper-bounded depending on the number of groups of identical potentials in the buckets
- In practice, there are two possible scenarios:

b	$\phi(b)$
$[3,0]$	$\langle \varphi_1 \rangle$
$[2,1]$	$\langle \varphi_4, \varphi_2, \varphi_2 \rangle$
$[1,2]$	$\langle \varphi_5, \varphi_5, \varphi_3 \rangle$
$[0,3]$	$\langle \varphi_6 \rangle$

1. A factor contains commutative arguments
   - Potential values are likely to contain a single group of duplicates
2. A factor contains no commutative arguments
   - Potential values are likely to be distinct

Experiments

- Comparison of run times of DECOR and the »naive« approach
- Average over factors with $k \in \{0, 2, \lfloor \frac{n}{2} \rfloor, n-1, n\}$ commutative arguments
- Timeout after five minutes per instance



Summary

- Problem of efficiently detecting commutative factors solved
- Upper bound on the number of iterations depending on the number of groups of identical potential values within the buckets of a factor
- The DECOR algorithm effectively exploits this upper bound in practice