is SP BP?

Yongyi Mao

University of Ottawa

November, 2007

Co-Authors

- Ronghui Tu
- Jiying Zhao

is SP BP?

Yongyi Mao

University of Ottawa

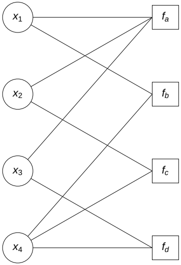
November, 2007

Co-Authors

- Ronghui Tu
- Jiying Zhao

Factor Graphs and Belief Propagation (BP)

[Kschischang, Frey and Loeliger, 2001]



- $F(x_1, x_2, x_3, x_4) = f_a(x_1, x_2, x_3) \cdot f_b(x_1, x_4) \cdot f_c(x_2, x_4) \cdot f_d(x_3, x_4)$
- Codes on graphs: e.g., $f_a(x_1, x_2, x_3) := [x_1 \oplus x_2 \oplus x_3 = 0]$.

Factor Graphs and Belief Propagation (BP)

[Kschischang, Frey and Loeliger, 2001]

Markov Random Field (MRF)

When $F(x_V)$ represents a joint distribution of $X_V := \{X_v : v \in V\}$, the factor graph is a Markov Random Field (MRF).

Computation Objective

For every $u \in V$, determine

$$\hat{x}_u := \arg\max_{x_u} \sum_{\sim x_u} F(x_V)$$

Example

- Statistical inference
- MAP Decoding

Factor Graphs and Belief Propagation (BP)

[Kschischang, Frey and Loeliger, 2001]

Markov Random Field (MRF)

When $F(x_V)$ represents a joint distribution of $X_V := \{X_v : v \in V\}$, the factor graph is a Markov Random Field (MRF).

Computation Objective

For every $u \in V$, determine

$$\hat{x}_u := \arg\max_{x_u} \sum_{\sim x_u} F(x_V)$$

Example

- Statistical inference
- MAP Decoding

Factor Graphs and Belief Propagation (BP)

[Kschischang, Frey and Loeliger, 2001]

Markov Random Field (MRF)

When $F(x_V)$ represents a joint distribution of $X_V := \{X_v : v \in V\}$, the factor graph is a Markov Random Field (MRF).

Computation Objective

For every $u \in V$, determine

$$\hat{x}_u := \arg\max_{x_u} \sum_{\sim x_u} F(x_V)$$

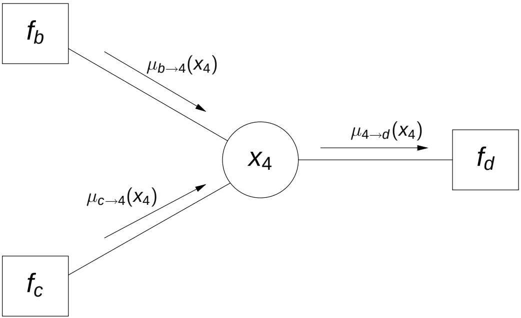
Example

- Statistical inference
- MAP Decoding

Factor Graphs and Belief Propagation (BP)

[Kschischang, Frey and Loeliger, 2001]

BP: variable vertex passing message

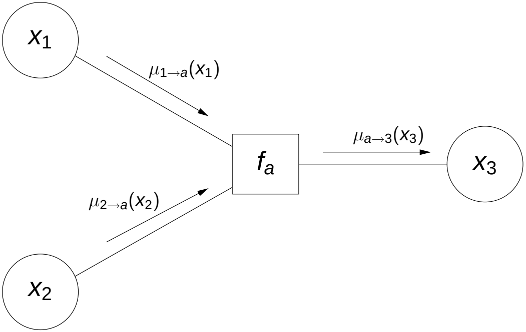


$$\mu_{4\to d}(x_4) = \mu_{b\to4}(x_4)\cdot\mu_{c\to4}(x_4)$$

Factor Graphs and Belief Propagation (BP)

[Kschischang, Frey and Loeliger, 2001]

BP: function vertex passing message

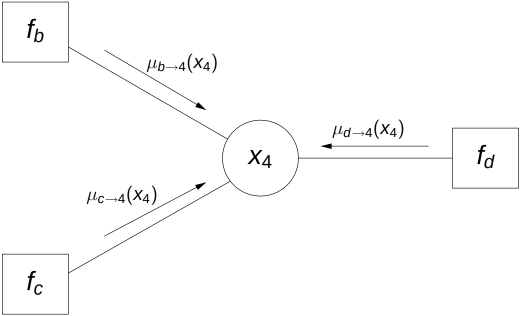


$$\mu_{a\to 3}(x_3) = \sum_{x_1,x_2} f_a(x_1,x_2,x_3)\cdot \mu_{1\to a}(x_1)\cdot \mu_{2\to a}(x_2)$$

Factor Graphs and Belief Propagation (BP)

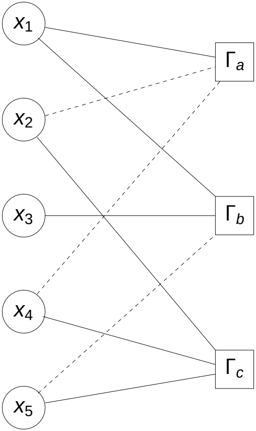
[Kschischang, Frey and Loeliger, 2001]

BP: summary message



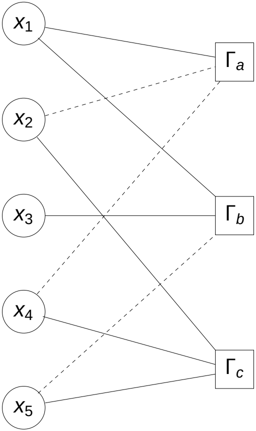
$$\mu_4(x_4) = \mu_{b\to 4}(x_4) \cdot \mu_{c\to 4}(x_4) \cdot \mu_{d\to 4}(x_4)$$

k-SAT Problems



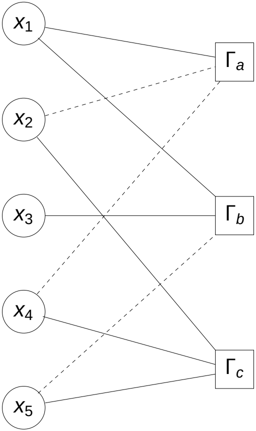
- each variable $x_v \in \{0, 1\}$
- each constraint involves k variables
- find a global configuration of variables satisfying all constraints

k-SAT Problems



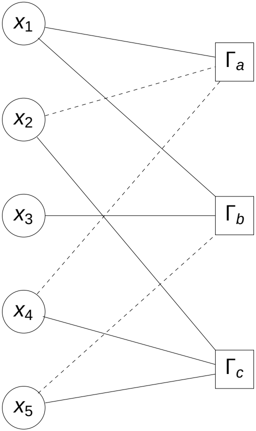
- each variable $x_v \in \{0, 1\}$
- each constraint involves k variables
- find a global configuration of variables satisfying all constraints

k-SAT Problems



- each variable $x_v \in \{0, 1\}$
- each constraint involves k variables
- find a global configuration of variables satisfying all constraints

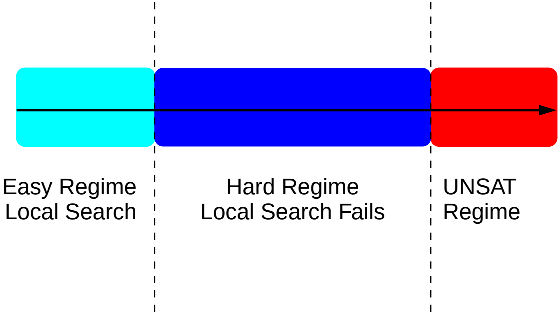
k-SAT Problems



- each variable $x_v \in \{0, 1\}$
- each constraint involves k variables
- find a global configuration of variables satisfying all constraints

Random k-SAT Problems

- Erdos-Renyi random graph ensemble parametrized by density α
- Two thresholds of α



Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

- left message (variable to constraint):
  - distribution of "intention"
  - possible intentions:
    - s: "I will satisfy you!"
    - u: "I will not satisfy you!"
    - $*$: "I can be whatever"
- right message (constraint to variable):
  - distribution of "command"
  - possible commands:
    - `warning`: "Satisfy me!"
    - `no warning`: "Whatever ..."

Survey Propagation (SP) for *k*-SAT

[Mézard, Parisi and Zecchina, 2002]

- left message (variable to constraint):
  - distribution of "intention"
  - possible intentions:
    - s: "I will satisfy you!"
    - u: "I will not satisfy you!"
    - $*$: "I can be whatever"
- right message (constraint to variable):
  - distribution of "command"
  - possible commands:
    - warning: "Satisfy me!"
    - no warning: "Whatever ..."

Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

- left message (variable to constraint):
  - distribution of “intention”
  - possible intentions:
    - s: “I will satisfy you!”
    - u: “I will not satisfy you!”
    - $*$: “I can be whatever”
- right message (constraint to variable):
  - distribution of “command”
  - possible commands:
    - `warning`: “Satisfy me!”
    - `no warning`: “Whatever ...”

Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

- left message (variable to constraint):
  - distribution of "intention"
  - possible intentions:
    - s: "I will satisfy you!"
    - u: "I will not satisfy you!"
    - $*$: "I can be whatever"
- right message (constraint to variable):
  - distribution of "command"
  - possible commands:
    - warning: "Satisfy me!"
    - no warning: "Whatever ..."

Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

- left message (variable to constraint):
  - distribution of "intention"
  - possible intentions:
    - s: "I will satisfy you!"
    - u: "I will not satisfy you!"
    - $*$: "I can be whatever"
- right message (constraint to variable):
  - distribution of "command"
  - possible commands:
    - warning: "Satisfy me!"
    - no warning: "Whatever ..."

Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

- left message (variable to constraint):
  - distribution of "intention"
  - possible intentions:
    - s: "I will satisfy you!"
    - u: "I will not satisfy you!"
    - $*$: "I can be whatever"
- right message (constraint to variable):
  - distribution of "command"
  - possible commands:
    - warning: "Satisfy me!"
    - no warning: "Whatever…"

Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

- left message (variable to constraint):
    - distribution of "intention"
    - possible intentions:
        - s: "I will satisfy you!"
        - u: "I will not satisfy you!"
        - $*$: "I can be whatever"
- right message (constraint to variable):
    - distribution of "command"
    - possible commands:
        - warning: "Satisfy me!"
        - no warning: "Whatever ..."

Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

- left message (variable to constraint):
  - distribution of "intention"
  - possible intentions:
    - s: "I will satisfy you!"
    - u: "I will not satisfy you!"
    - $*$: "I can be whatever"
- right message (constraint to variable):
  - distribution of "command"
  - possible commands:
    - warning: "Satisfy me!"
    - no warning: "Whatever ..."

Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

- left message (variable to constraint):
  - distribution of "intention"
  - possible intentions:
    - s: "I will satisfy you!"
    - u: "I will not satisfy you!"
    - $*$: "I can be whatever"
- right message (constraint to variable):
  - distribution of "command"
  - possible commands:
    - `warning`: "Satisfy me!"
    - `no warning`: "Whatever ..."

Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

- left message (variable to constraint):
  - distribution of "intention"
  - possible intentions:
    - s: "I will satisfy you!"
    - u: "I will not satisfy you!"
    - $*$: "I can be whatever"
- right message (constraint to variable):
  - distribution of "command"
  - possible commands:
    - `warning`: "Satisfy me!"
    - `no warning`: "Whatever ..."

Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

- left message (variable to constraint):
  - distribution of "intention"
  - possible intentions:
    - s: "I will satisfy you!"
    - u: "I will not satisfy you!"
    - $*$: "I can be whatever"
- right message (constraint to variable):
  - distribution of "command"
  - possible commands:
    - `warning`: "Satisfy me!"
    - `no warning`: "Whatever ..."

Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

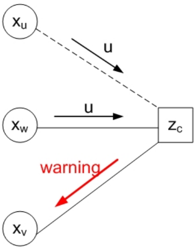
- left message (variable to constraint):
  - distribution of "intention"
  - possible intentions:
    - s: "I will satisfy you!"
    - u: "I will not satisfy you!"
    - $*$: "I can be whatever"
- right message (constraint to variable):
  - distribution of "command"
  - possible commands:
    - `warning`: "Satisfy me!"
    - no `warning`: "Whatever ..."

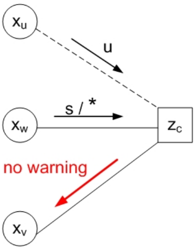
Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

Right Message: Constraint $\to$ Variable

- $\eta_{c\to v} = \prod_{u \in V(c)\setminus\{v\}} \frac{\Pi^{\mathrm{u}}_{u\to c}}{\Pi^{\mathrm{u}}_{u\to c} + \Pi^{\mathrm{s}}_{u\to c} + \Pi^{*}_{u\to c}}$



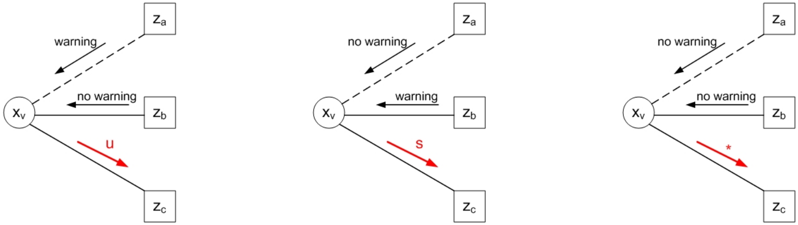


Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

Left Message: Variable $\rightarrow$ Constraint

- $\Pi^{u}_{v\rightarrow c} = \left(1 - \prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})\right)\prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{s}_{v\rightarrow c} = \left(1 - \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})\right)\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{*}_{v\rightarrow c} = \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$



Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

Summary Message

- $\zeta_v^1 = \left(1 - \prod_{b \in C^1(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$
- $\zeta_v^0 = \left(1 - \prod_{b \in C^0(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^1(v)} (1 - \eta_{b \to v})$
- $\zeta_v^* = \prod_{b \in C^1(v)} (1 - \eta_{b \to v}) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$

- $B(v) = \zeta_v^1 - \zeta_v^0$

Survey Propagation (SP) for k-SAT

[Mézard, Parisi and Zecchina, 2002]

- `while (~solvableByLocalSearch(problem))`
  `{`
  - `while ( ~converge || ~reachMaxIteration)`
    `{`
    - `variables.passMessages`
    - `constraints.passMessages`
    - `variables.updateSummaryMessages`

    `}`
  - `problem:=decimation()`

  `}`
- `solution=localSearch(problem)`

k-SAT: From SP to SP(γ)

[Maneva, Mossel and Wainwright, 2005]

SP Right Message: Constraint $\rightarrow$ Variable

- $\eta_{c\rightarrow v} = \prod_{u \in V(c)\setminus\{v\}} \frac{\Pi^{u}_{u\rightarrow c}}{\Pi^{u}_{u\rightarrow c}+\Pi^{s}_{u\rightarrow c}+\Pi^{*}_{u\rightarrow c}}$

SP(γ) Right Message: Constraint $\rightarrow$ Variable

- $\eta_{c\rightarrow v} = \prod_{u \in V(c)\setminus\{v\}} \frac{\Pi^{u}_{u\rightarrow c}}{\Pi^{u}_{u\rightarrow c}+\Pi^{s}_{u\rightarrow c}+\Pi^{*}_{u\rightarrow c}}$

k-SAT: From SP to SP(γ)

[Maneva, Mossel and Wainwright, 2005]

SP Right Message: Constraint $\longrightarrow$ Variable

- $\eta_{c\rightarrow v} = \prod_{u\in V(c)\setminus\{v\}} \frac{\Pi^{\mathrm{u}}_{u\rightarrow c}}{\Pi^{\mathrm{u}}_{u\rightarrow c}+\Pi^{\mathrm{s}}_{u\rightarrow c}+\Pi^{*}_{u\rightarrow c}}$

SP(γ) Right Message: Constraint $\longrightarrow$ Variable

- $\eta_{c\rightarrow v} = \prod_{u\in V(c)\setminus\{v\}} \frac{\Pi^{\mathrm{u}}_{u\rightarrow c}}{\Pi^{\mathrm{u}}_{u\rightarrow c}+\Pi^{\mathrm{s}}_{u\rightarrow c}+\Pi^{*}_{u\rightarrow c}}$

k-SAT: From SP to SP(γ)

[Maneva, Mossel and Wainwright, 2005]

SP Left Message: Variable $\rightarrow$ Constraint

- $\Pi^{u}_{v\rightarrow c} = \left(1 - \prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})\right)\prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{s}_{v\rightarrow c} = \left(1 - \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})\right)\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{*}_{v\rightarrow c} = \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$

SP(γ) Left Message: Variable $\rightarrow$ Constraint

- $\Pi^{u}_{v\rightarrow c} = \left(1 - \gamma\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})\right)\prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{s}_{v\rightarrow c} = \left(1 - \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})\right)\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{*}_{v\rightarrow c} = \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$

k-SAT: From SP to SP(γ)

[Maneva, Mossel and Wainwright, 2005]

SP Left Message: Variable $\rightarrow$ Constraint

- $\Pi^{u}_{v\rightarrow c} = \left(1 - \prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})\right)\prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{s}_{v\rightarrow c} = \left(1 - \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})\right)\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{*}_{v\rightarrow c} = \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$

SP(γ) Left Message: Variable $\rightarrow$ Constraint

- $\Pi^{u}_{v\rightarrow c} = \left(1 - \gamma\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})\right)\prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{s}_{v\rightarrow c} = \left(1 - \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})\right)\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{*}_{v\rightarrow c} = \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$

k-SAT: From SP to SP(γ)

[Maneva, Mossel and Wainwright, 2005]

SP Summary Message

- $\zeta_v^1 = \left(1 - \prod_{b \in C^1(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$
- $\zeta_v^0 = \left(1 - \prod_{b \in C^0(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^1(v)} (1 - \eta_{b \to v})$
- $\zeta_v^* = \prod_{b \in C^1(v)} (1 - \eta_{b \to v}) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$

SP(γ) Summary Message

- $\zeta_v^1 = \left(1 - \gamma \prod_{b \in C^1(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$
- $\zeta_v^0 = \left(1 - \gamma \prod_{b \in C^0(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^1(v)} (1 - \eta_{b \to v})$
- $\zeta_v^* = \gamma \prod_{b \in C^1(v)} (1 - \eta_{b \to v}) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$

k-SAT: From SP to SP(γ)

[Maneva, Mossel and Wainwright, 2005]

SP Summary Message

- $\zeta_v^1 = \left(1 - \prod_{b \in C^1(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$
- $\zeta_v^0 = \left(1 - \prod_{b \in C^0(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^1(v)} (1 - \eta_{b \to v})$
- $\zeta_v^* = \prod_{b \in C^1(v)} (1 - \eta_{b \to v}) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$

SP(γ) Summary Message

- $\zeta_v^1 = \left(1 - \gamma \prod_{b \in C^1(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$
- $\zeta_v^0 = \left(1 - \gamma \prod_{b \in C^0(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^1(v)} (1 - \eta_{b \to v})$
- $\zeta_v^* = \gamma \prod_{b \in C^1(v)} (1 - \eta_{b \to v}) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$

k-SAT: From SP to SP(γ)

[Maneva, Mossel and Wainwright, 2005]

SP(γ)

- $\gamma \in [0, 1]$ providing tunable performance
- $\gamma = 1 \Rightarrow SP(\gamma)$ is SP.
- developed for k-SAT
- lacking probabilistic interpretation
- resulted from BP on a (different) MRF formalism of k-SAT
    - $\Rightarrow$ "SP is BP"
        - $\Rightarrow$ a new look of SP

k-SAT: From SP to SP(γ)

[Maneva, Mossel and Wainwright, 2005]

SP(γ)

- $\gamma \in [0, 1]$ providing tunable performance
- $\gamma = 1 \Rightarrow SP(\gamma)$ is SP.
- developed for k-SAT
- lacking probabilistic interpretation
- resulted from BP on a (different) MRF formalism of k-SAT
  - $\Rightarrow$ "SP is BP"
    - $\Rightarrow$ a new look at SP

k-SAT: From SP to SP(γ)

[Maneva, Mossel and Wainwright, 2005]

SP(γ)

- $\gamma \in [0, 1]$ providing tunable performance
- $\gamma = 1 \Rightarrow SP(\gamma)$ is SP.
- developed for k-SAT
- lacking probabilistic interpretation
- resulted from BP on a (different) MRF formalism of k-SAT
  - $\Rightarrow$ "SP is BP"
    - $\Rightarrow$ a new look at SP

k-SAT: From SP to SP(γ)

[Maneva, Mossel and Wainwright, 2005]

SP(γ)

- $\gamma \in [0, 1]$ providing tunable performance
- $\gamma = 1 \Rightarrow SP(\gamma)$ is *SP*.
- developed for *k*-SAT
- lacking probabilistic interpretation
- resulted from BP on a (different) MRF formalism of *k*-SAT
    - $\Rightarrow$ "SP is BP"
        - $\Rightarrow$ a new look of SP

k-SAT: From SP to SP(γ)

[Maneva, Mossel and Wainwright, 2005]

SP(γ)

- $\gamma \in [0, 1]$ providing tunable performance
- $\gamma = 1 \Rightarrow SP(\gamma)$ is SP.
- developed for k-SAT
- lacking probabilistic interpretation
- resulted from BP on a (different) MRF formalism of k-SAT
  - $\Rightarrow$ “SP is BP”
    - $\Rightarrow$ a new look at SP

k-SAT: From SP to SP(γ)

[Maneva, Mossel and Wainwright, 2005]

SP(γ)

- $\gamma \in [0, 1]$ providing tunable performance
- $\gamma = 1 \Rightarrow SP(\gamma)$ is SP.
- developed for k-SAT
- lacking probabilistic interpretation
- resulted from BP on a (different) MRF formalism of k-SAT
  - $\Rightarrow$ "SP is BP"
    - $\Rightarrow$ a new look of SP

SP(γ)

- $\gamma \in [0, 1]$ providing tunable performance
- $\gamma = 1 \Rightarrow SP(\gamma)$ is SP.
- developed for k-SAT
- lacking probabilistic interpretation
- resulted from BP on a (different) MRF formalism of k-SAT
  - $\Rightarrow$ "SP is BP"
    - $\Rightarrow$ a new look of SP

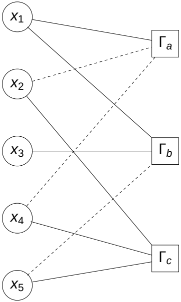
k-SAT: From SP to SP(γ)

[Maneva, Mossel and Wainwright, 2005]

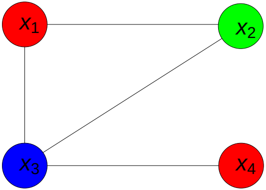
SP(γ)

- $\gamma \in [0, 1]$ providing tunable performance
- $\gamma = 1 \Rightarrow SP(\gamma)$ is SP.
- developed for k-SAT
- lacking probabilistic interpretation
- resulted from BP on a (different) MRF formalism of k-SAT
  - $\Rightarrow$ "SP is BP"
    - $\Rightarrow$ a new look of SP

Success of SP: Constraint-Satisfaction Problems (CSPs)



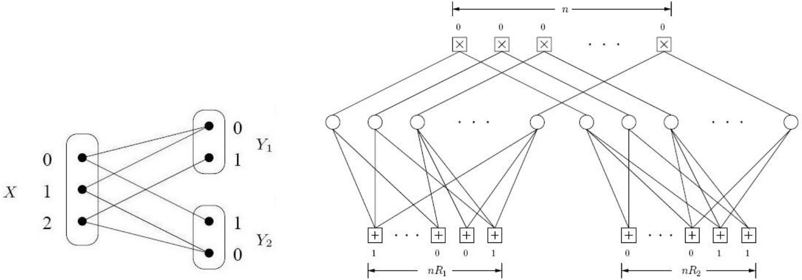
k-SAT



q-COL

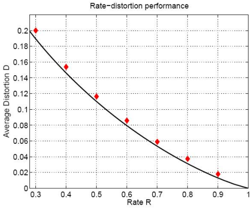
[Mézard, Parisi and Zecchina, 2002] [Braunstein, Mulet, Pagnani, Weigt and Zecchina, 2003]

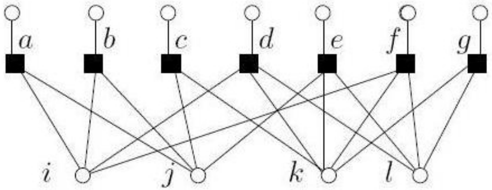
Success of SP: Coding for Blackwell Channel



[Yu and Aleksic, 2005]

Success of SP: Quantization of Bernoulli Source





[Wainwright and Maneva, 2005]

is SP BP ?

"SP is BP"

For $k-$SAT Problems

- SP is BP [Braunstein and Zecchina, 2004]
- SP(γ) is BP [Maneva, Mossel and Wainwright, 2005]

Question

How about for general CSPs?

“SP is BP”

For $k-$SAT Problems

- SP is BP [Braunstein and Zecchina, 2004]
- SP(γ) is BP [Maneva, Mossel and Wainwright, 2005]

Question

How about for general CSPs?

Roadmap

- Generic formulation of CSP?
- Generalizing SP?
  - What is SP?
  - Generalizing $\mathrm{SP}_{ksat}(\gamma)$ for arbitrary CSPs?
- MRF formalism for arbitrary CSP?
- BP-to-SP reduction rule?

Example

- k-SAT
- 3-COL

Roadmap

- Generic formulation of CSP?
- Generalizing SP?
  - What is SP?
  - Generalizing $\mathrm{SP}_{ksat}(\gamma)$ for arbitrary CSPs?
- MRF formalism for arbitrary CSP?
- BP-to-SP reduction rule?

Example

- k-SAT
- 3-COL

Roadmap

- Generic formulation of CSP?
- Generalizing SP?
  - What is SP?
  - Generalizing $\mathrm{SP}_{ksat}(\gamma)$ for arbitrary CSPs?
- MRF formalism for arbitrary CSP?
- BP-to-SP reduction rule?

Example

- k-SAT
- 3-COL

Roadmap

- Generic formulation of CSP?
- Generalizing SP?
  - What is SP?
  - Generalizing $\mathrm{SP}_{ksat}(\gamma)$ for arbitrary CSPs?
- MRF formalism for arbitrary CSP?
- BP-to-SP reduction rule?

Example

- k-SAT
- 3-COL

Roadmap

- Generic formulation of CSP?
- Generalizing SP?
  - What is SP?
  - Generalizing $\mathrm{SP}_{ksat}(\gamma)$ for arbitrary CSPs?
- MRF formalism for arbitrary CSP?
- BP-to-SP reduction rule?

Example

- k-SAT
- 3-COL

Roadmap

- Generic formulation of CSP?
- Generalizing SP?
  - What is SP?
  - Generalizing $\mathrm{SP}_{ksat}(\gamma)$ for arbitrary CSPs?
- MRF formalism for arbitrary CSP?
- BP-to-SP reduction rule?

Example

- k-SAT
- 3-COL

Roadmap

- Generic formulation of CSP?
- Generalizing SP?
  - What is SP?
  - Generalizing $\text{SP}_{ksat}(\gamma)$ for arbitrary CSPs?
- MRF formalism for arbitrary CSP?
- BP-to-SP reduction rule?

Example

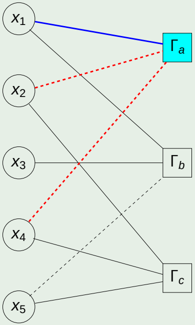
- k-SAT
- 3-COL

Generic Formulation of CSP

- variable $x_v \in \chi_v$
- local constraint $\Gamma_c \subset \prod_{v \in V(c)} \chi_v$

Generic Formulation of CSP

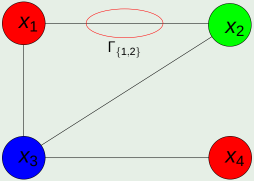
Example (k-SAT)



- $\chi_v := \{0, 1\}$.
- $\Gamma_a :=$
  $(\chi_1 \times \chi_2 \times \chi_4) \setminus \{(0, 1, 1)\}$

Generic Formulation of CSP

Example (3-COL)



- $\chi_v := \{1, 2, 3\}$.
- $\Gamma_{\{1,2\}} :=$
  $(\chi_1 \times \chi_2) \setminus \{(1,1), (2,2), (3,3)\}$

find a solution for

$$\prod_{c \in C} [x_{V(c)} \in \Gamma_c] = 1$$

- Factor graph representation

find a solution for

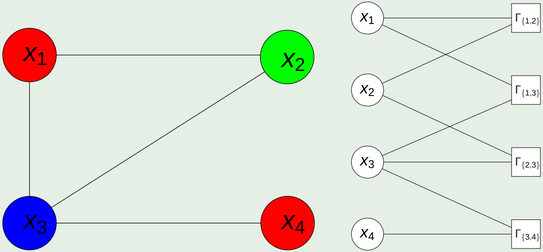
$$\prod_{c \in C} [x_{V(c)} \in \Gamma_c] = 1$$

- Factor graph representation

Generic Formulation of CSP

Example

Factor graph representation of 3-COL



Generalizing SP

- What is SP?
  - How to characterize "intention"/"command"?
- How to interpret $SP_{ksat}(\gamma)$?

Generalizing SP

- What is SP?
  - How to characterize "intention"/"command"?
- How to interpret $SP_{ksat}(\gamma)$?

Generalizing SP

- What is SP?
  - How to characterize “intention”/“command”?
- How to interpret $SP_{ksat}(\gamma)$?

Generalizing SP: Alphabet Extension

Alphabet Extension

For every $v \in V$, define token set of v

$$\chi_v^* := \{t \subseteq \chi_v : t \neq \emptyset\}$$

- "intentions" and "commands": elements of χ_v^*.

Example (symbols in SP_{ksat})

"intentions"

- $\mathtt{s} : \{L\}$
- $\mathtt{u} : \{\bar{L}\}$
- $* : \{0, 1\}$

"commands"

- $\mathtt{no\ warning} : \{0, 1\}$
- $\mathtt{warning} : \{L\}$

Generalizing SP: Alphabet Extension

Alphabet Extension

For every $v \in V$, define token set of v

$$\chi_v^* := \{t \subseteq \chi_v : t \neq \emptyset\}$$

- “intentions” and “commands”: elements of χ_v^*.

Example (symbols in SP_{ksat})

“intentions”

- `s` : $\{L\}$
- `u` : $\{\bar{L}\}$
- $*$: $\{0, 1\}$

“commands”

- `no warning` : $\{0, 1\}$
- `warning` : $\{L\}$

Generalizing SP: Alphabet Extension

Alphabet Extension

For every $v \in V$, define token set of v

$$\chi_v^* := \{t \subseteq \chi_v : t \neq \emptyset\}$$

- "intentions" and "commands": elements of χ_v^*.

Example (symbols in SP_{ksat})

"intentions"

- s : $\{L\}$
- u : $\{\bar{L}\}$
- $*$: $\{0, 1\}$

"commands"

- `no warning` : $\{0, 1\}$
- `warning` : $\{L\}$

Generalizing SP: Alphabet Extension

Alphabet Extension

For every $v \in V$, define token set of v

$$\chi_v^* := \{t \subseteq \chi_v : t \neq \emptyset\}$$

- "intentions" and "commands": elements of χ_v^*.

Example (symbols in SP_{ksat})

"intentions"

- s : $\{L\}$
- u : $\{\bar{L}\}$
- $*$: $\{0, 1\}$

"commands"

- `no warning` : $\{0, 1\}$
- `warning` : $\{L\}$

Generalizing SP: Alphabet Extension

- *speaking of obedience ...*

"obedience"

Intention a obeys command b if

$$a \subseteq b$$

Generalizing SP: Alphabet Extension

- *speaking of obedience ...*

"obedience"

Intention a obeys command b if

$$a \subseteq b$$

Generalizing SP: Alphabet Extension

- *speaking of democracy ...*

"democratic voice (letter)"

A letter $x_v \in \chi_v$ is "democratic" w.r.t. a set of intentions $t_{V(c)\setminus\{v\}} := \{t_u \in \chi_u^* : u \in V(c) \setminus \{v\}\}$ if at least one combination of the letters in the intentions paired with letter x_v makes Γ_c satisfied, namely

$$(x_v, x_{V(c)\setminus\{v\}}) \in \Gamma_c \text{ for some } x_{V(c)\setminus\{v\}} \in \prod_{u \in V(c)\setminus\{v\}} t_u.$$

Generalizing SP: Alphabet Extension

- *speaking of democracy ...*

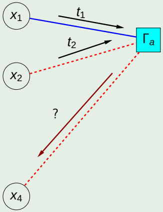
"democratic voice (letter)"

A letter $x_v \in \chi_v$ is "democratic" w.r.t. a set of intentions $t_{V(c)\setminus\{v\}} := \{t_u \in \chi_u^* : u \in V(c) \setminus \{v\}\}$ if at least one combination of the letters in the intentions paired with letter x_v makes Γ_c satisfied, namely

$$\left(x_v, x_{V(c)\setminus\{v\}}\right) \in \Gamma_c \text{ for some } x_{V(c)\setminus\{v\}} \in \prod_{u \in V(c)\setminus\{v\}} t_u .$$

Generalizing SP: Alphabet Extension

Example (k-SAT)



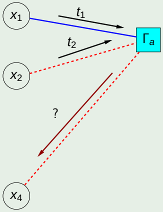
- $t_1 = \{1\}, t_2 = \{1\}$
  - $x_4 = 1$ is democratic
  - $x_4 = 0$ is democratic
- $t_1 = \{0\}, t_2 = \{1\}$
  - $x_4 = 0$ is democratic

"democratic command": Forced Token $F_c\left(\prod_{u \in V(c)\setminus\{v\}} t_u\right)$

The set of all "democratic letters" w.r.t a set of intentions $t_{V(c)\setminus\{v\}} := \{t_u \in \chi_u^* : u \in V(c) \setminus \{v\}\}$ is the "democratic command" w.r.t $t_{V(c)\setminus\{v\}}$.

Generalizing SP: Alphabet Extension

Example (k-SAT)



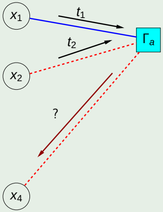
- $t_1 = \{1\}, t_2 = \{1\}$
  - $x_4 = 1$ is democratic
  - $x_4 = 0$ is democratic
- $t_1 = \{0\}, t_2 = \{1\}$
  - $x_4 = 0$ is democratic

"democratic command": Forced Token $F_c\left(\prod_{u \in V(c)\setminus\{v\}} t_u\right)$

The set of all "democratic letters" w.r.t a set of intentions $t_{V(c)\setminus\{v\}} := \{t_u \in \chi_u^* : u \in V(c) \setminus \{v\}\}$ is the "democratic command" w.r.t $t_{V(c)\setminus\{v\}}$.

Generalizing SP: Alphabet Extension

Example (k-SAT)



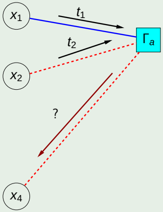
- $t_1 = \{1\}, t_2 = \{1\}$
  - $x_4 = 1$ is democratic
  - $x_4 = 0$ is democratic
- $t_1 = \{0\}, t_2 = \{1\}$
  - $x_4 = 0$ is democratic

“democratic command”: Forced Token $F_c \left(\prod_{u \in V(c) \setminus \{v\}} t_u \right)$

The set of all “democratic letters” w.r.t a set of intentions $t_{V(c)\setminus\{v\}} := \{t_u \in \chi_u^* : u \in V(c) \setminus \{v\}\}$ is the “democratic command” w.r.t $t_{V(c)\setminus\{v\}}$.

Generalizing SP: Alphabet Extension

Example (k-SAT)



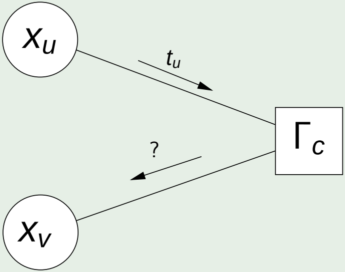
- $t_1 = \{1\}, t_2 = \{1\}$
  - $x_4 = 1$ is democratic
  - $x_4 = 0$ is democratic
- $t_1 = \{0\}, t_2 = \{1\}$
  - $x_4 = 0$ is democratic

"democratic command": Forced Token $F_c\left(\prod_{u \in V(c)\setminus\{v\}} t_u\right)$

The set of all "democratic letters" w.r.t a set of intentions $t_{V(c)\setminus\{v\}} := \{t_u \in \chi_u^* : u \in V(c) \setminus \{v\}\}$ is the "democratic command" w.r.t $t_{V(c)\setminus\{v\}}$.

Generalizing SP: Alphabet Extension

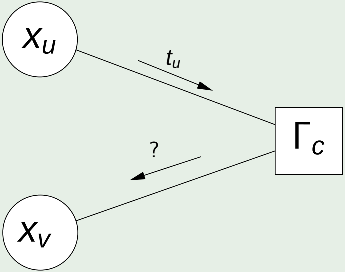
Example (3-COL)



- $t_u = \{1\}$
  - $F_c(t_u) = \{2, 3\}$
- $t_u = \{1, 2\}$
  - $F_c(t_u) = \{1, 2, 3\}$

Generalizing SP: Alphabet Extension

Example (3-COL)



- $t_u = \{1\}$
  - $F_c(t_u) = \{2, 3\}$
- $t_u = \{1, 2\}$
  - $F_c(t_u) = \{1, 2, 3\}$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - having maximal "freedom"
- $t_{v\to c} := \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v\to c}(t) := \sum_{t_{C(v)\setminus\{c\}\to v}} \left[t = \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v} \right] \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(t_{b\to v})$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - having maximal “freedom”
- $t_{v\to c} := \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v\to c}(t) := \sum_{t_{C(v)\setminus\{c\}\to v}} \left[t = \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v} \right] \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(t_{b\to v})$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - having maximal "freedom"
- $t_{v \to c} := \bigcap_{b \in C(v) \setminus \{c\}} t_{b \to v}$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v \to c}(t) := \sum_{t_{C(v)\setminus\{c\} \to v}} \left[t = \bigcap_{b \in C(v)\setminus\{c\}} t_{b \to v} \right] \prod_{b \in C(v)\setminus\{c\}} \rho_{b \to v}(t_{b \to v})$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - having maximal “freedom”
- $t_{v\to c} := \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v\to c}(t) := \sum_{t_{C(v)\setminus\{c\}\to v}} \left[t = \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v} \right] \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(t_{b\to v})$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - having maximal “freedom”
- $t_{v\to c} := \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v\to c}(t) := \sum_{t_{C(v)\setminus\{c\}\to v}} \left[t = \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v} \right] \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(t_{b\to v})$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - having maximal "freedom"
- $t_{v\to c} := \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v\to c}(t) := \sum_{t_{C(v)\setminus\{c\}\to v}} \left[t = \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v} \right] \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(t_{b\to v})$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - having maximal “freedom”
- $t_{v \to c} := \bigcap_{b \in C(v) \setminus \{c\}} t_{b \to v}$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v \to c}(t) := \sum_{t_{C(v) \setminus \{c\} \to v}} \left[t = \bigcap_{b \in C(v) \setminus \{c\}} t_{b \to v} \right] \prod_{b \in C(v) \setminus \{c\}} \rho_{b \to v}(t_{b \to v})$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - having maximal "freedom"
- $t_{v\to c} := \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v\to c}(t) := \sum_{t_{C(v)\setminus\{c\}\to v}} \left[t = \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v} \right] \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(t_{b\to v})$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Right Message

- outgoing command:
  - the "democratic command" w.r.t. incoming intentions from upstream
- $t_{c\to v} := \mathsf{F}_c\left(\prod_{u\in V(c)\setminus\{v\}} t_{u\to c}\right)$
- right message: the distribution of outgoing command
  - conditioned on no conflict in incoming intentions
  - assuming independence of incoming intentions
- $\rho_{c\to v}(t) := \sum_{t_{V(c)\setminus\{v\}\to c}} \left[t = \mathsf{F}_c\left(\prod_{u\in V(c)\setminus\{v\}} t_{u\to c}\right)\right] \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(t_{u\to c})$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Right Message

- outgoing command:
  - the "democratic command" w.r.t. incoming intentions from upstream
- $t_{c \to v} := \mathrm{F}_c\left(\prod_{u \in V(c) \setminus \{v\}} t_{u \to c}\right)$
- right message: the distribution of outgoing command
  - conditioned on no conflict in incoming intentions
  - assuming independence of incoming intentions
- $\rho_{c \to v}(t) :=$

$$\sum_{t_{V(c)\setminus\{v\} \to c}} \left[t = \mathrm{F}_c\left(\prod_{u \in V(c)\setminus\{v\}} t_{u \to c}\right)\right] \prod_{u \in V(c)\setminus\{v\}} \lambda_{u \to c}(t_{u \to c})$$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Right Message

- outgoing command:
  - the "democratic command" w.r.t. incoming intentions from upstream
- $t_{c \to v} := \mathsf{F}_c \left(\prod_{u \in V(c) \setminus \{v\}} t_{u \to c} \right)$
- right message: the distribution of outgoing command
  - conditioned on no conflict in incoming intentions
  - assuming independence of incoming intentions
- $\rho_{c \to v}(t) :=$
  $$\sum_{t_{V(c) \setminus \{v\} \to c}} \left[t = \mathsf{F}_c \left(\prod_{u \in V(c) \setminus \{v\}} t_{u \to c} \right) \right] \prod_{u \in V(c) \setminus \{v\}} \lambda_{u \to c}(t_{u \to c})$$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Right Message

- outgoing command:
  - the "democratic command" w.r.t. incoming intentions from upstream
- $t_{c \to v} := \mathrm{F}_c \left(\prod_{u \in V(c) \setminus \{v\}} t_{u \to c} \right)$
- right message: the distribution of outgoing command
  - conditioned on no conflict in incoming intentions
  - assuming independence of incoming intentions
- $\rho_{c \to v}(t) :=$

$$\sum_{t_{V(c) \setminus \{v\} \to c}} \left[t = \mathrm{F}_c \left(\prod_{u \in V(c) \setminus \{v\}} t_{u \to c} \right) \right] \prod_{u \in V(c) \setminus \{v\}} \lambda_{u \to c}(t_{u \to c})$$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Right Message

- outgoing command:
  - the "democratic command" w.r.t. incoming intentions from upstream
- $t_{c \to v} := \mathsf{F}_c \left(\prod_{u \in V(c) \setminus \{v\}} t_{u \to c} \right)$
- right message: the distribution of outgoing command
  - conditioned on no conflict in incoming intentions
  - assuming independence of incoming intentions
- $\rho_{c \to v}(t) := \sum_{t_{V(c) \setminus \{v\} \to c}} \left[t = \mathsf{F}_c \left(\prod_{u \in V(c) \setminus \{v\}} t_{u \to c} \right) \right] \prod_{u \in V(c) \setminus \{v\}} \lambda_{u \to c}(t_{u \to c})$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Right Message

- outgoing command:
  - the "democratic command" w.r.t. incoming intentions from upstream
- $t_{c \to v} := \mathsf{F}_c \left(\prod_{u \in V(c) \setminus \{v\}} t_{u \to c} \right)$
- right message: the distribution of outgoing command
  - conditioned on no conflict in incoming intentions
  - assuming independence of incoming intentions
- $\rho_{c \to v}(t) :=$
  $$\sum_{t_{V(c)\setminus\{v\} \to c}} \left[t = \mathsf{F}_c \left(\prod_{u \in V(c) \setminus \{v\}} t_{u \to c} \right) \right] \prod_{u \in V(c) \setminus \{v\}} \lambda_{u \to c}(t_{u \to c})$$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Right Message

- outgoing command:
  - the "democratic command" w.r.t. incoming intentions from upstream
- $t_{c \to v} := \mathsf{F}_c \left(\prod_{u \in V(c) \setminus \{v\}} t_{u \to c} \right)$
- right message: the distribution of outgoing command
  - conditioned on no conflict in incoming intentions
  - assuming independence of incoming intentions
- $\rho_{c \to v}(t) :=$
  $$\sum_{t_{V(c)\setminus\{v\} \to c}} \left[t = \mathsf{F}_c \left(\prod_{u \in V(c) \setminus \{v\}} t_{u \to c} \right) \right] \prod_{u \in V(c) \setminus \{v\}} \lambda_{u \to c}(t_{u \to c})$$

Generalizing SP: Interpretation of SP

Probabilistic Token Passing (PTP): Summary Message

- summary intention:
  - obeying all incoming commands from all directions
  - having maximal "freedom"
- $t_v := \bigcap_{b \in C(v)} t_{b \to v}$
- summary message: the distribution of summary intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\mu_v(t) := \sum_{t_{C(v) \to v}} \left[t = \bigcap_{b \in C(v)} t_{b \to v} \right] \prod_{b \in C(v)} \rho_{b \to v}(t_{b \to v})$

Generalizing SP: PTP is SP

Theorem

$PTP_{ksat} = SP_{ksat}$. Specifically

- $\Pi^{s}_{v\to c} \leftrightarrow \lambda_{v\to c}(\{L\})$
- $\Pi^{u}_{v\to c} \leftrightarrow \lambda_{v\to c}(\{\bar{L}\})$
- $\Pi^{*}_{v\to c} \leftrightarrow \lambda_{v\to c}(\{0,1\})$
- $\eta_{c\to v} \leftrightarrow \rho_{c\to v}(\{0\}) + \rho_{c\to v}(\{1\})$

- $\zeta^{0}_{v} \leftrightarrow \mu_v(\{0\})$
- $\zeta^{1}_{v} \leftrightarrow \mu_v(\{1\})$
- $\zeta^{*}_{v} \leftrightarrow \mu_v(\{0,1\})$

Theorem

$PTP_{3COL} = SP_{3COL}$ [Braunstein, Mulet, Pagnani, Weigt and Zecchina, 2003].

Generalizing SP: PTP is SP

Theorem

$PTP_{ksat} = SP_{ksat}$. Specifically

- $\Pi^{s}_{v\to c} \leftrightarrow \lambda_{v\to c}(\{L\})$
- $\Pi^{u}_{v\to c} \leftrightarrow \lambda_{v\to c}(\{\bar{L}\})$
- $\Pi^{*}_{v\to c} \leftrightarrow \lambda_{v\to c}(\{0,1\})$
- $\eta_{c\to v} \leftrightarrow \rho_{c\to v}(\{0\}) + \rho_{c\to v}(\{1\})$

- $\zeta^{0}_{v} \leftrightarrow \mu_{v}(\{0\})$
- $\zeta^{1}_{v} \leftrightarrow \mu_{v}(\{1\})$
- $\zeta^{*}_{v} \leftrightarrow \mu_{v}(\{0,1\})$

Theorem

$PTP_{3COL} = SP_{3COL}$ [Braunstein, Mulet, Pagnani, Weigt and Zecchina, 2003].

Generalizing SP: PTP is SP

Theorem

$PTP_{ksat} = SP_{ksat}$. Specifically

- $\Pi^{s}_{v \to c} \leftrightarrow \lambda_{v \to c}(\{L\})$
- $\Pi^{u}_{v \to c} \leftrightarrow \lambda_{v \to c}(\{\bar{L}\})$
- $\Pi^{*}_{v \to c} \leftrightarrow \lambda_{v \to c}(\{0, 1\})$
- $\eta_{c \to v} \leftrightarrow \rho_{c \to v}(\{0\}) + \rho_{c \to v}(\{1\})$

- $\zeta^{0}_{v} \leftrightarrow \mu_{v}(\{0\})$
- $\zeta^{1}_{v} \leftrightarrow \mu_{v}(\{1\})$
- $\zeta^{*}_{v} \leftrightarrow \mu_{v}(\{0, 1\})$

Theorem

$PTP_{3COL} = SP_{3COL}$ [Braunstein, Mulet, Pagnani, Weigt and Zecchina, 2003].

Generalizing SP: PTP is SP

Theorem

$PTP_{ksat} = SP_{ksat}$. Specifically

- $\Pi^{s}_{v\to c} \leftrightarrow \lambda_{v\to c}(\{L\})$
- $\Pi^{u}_{v\to c} \leftrightarrow \lambda_{v\to c}(\{\bar{L}\})$
- $\Pi^{*}_{v\to c} \leftrightarrow \lambda_{v\to c}(\{0,1\})$
- $\eta_{c\to v} \leftrightarrow \rho_{c\to v}(\{0\}) + \rho_{c\to v}(\{1\})$

- $\zeta^{0}_{v} \leftrightarrow \mu_{v}(\{0\})$
- $\zeta^{1}_{v} \leftrightarrow \mu_{v}(\{1\})$
- $\zeta^{*}_{v} \leftrightarrow \mu_{v}(\{0,1\})$

Theorem

$PTP_{3COL} = SP_{3COL}$ [Braunstein, Mulet, Pagnani, Weigt and Zecchina, 2003].

Generalizing SP: PTP is SP

Theorem

$PTP_{ksat} = SP_{ksat}$. Specifically

- $\Pi^{s}_{v\to c} \leftrightarrow \lambda_{v\to c}(\{L\})$
- $\Pi^{u}_{v\to c} \leftrightarrow \lambda_{v\to c}(\{\bar{L}\})$
- $\Pi^{*}_{v\to c} \leftrightarrow \lambda_{v\to c}(\{0,1\})$
- $\eta_{c\to v} \leftrightarrow \rho_{c\to v}(\{0\}) + \rho_{c\to v}(\{1\})$

- $\zeta^{0}_{v} \leftrightarrow \mu_{v}(\{0\})$
- $\zeta^{1}_{v} \leftrightarrow \mu_{v}(\{1\})$
- $\zeta^{*}_{v} \leftrightarrow \mu_{v}(\{0,1\})$

Theorem

$PTP_{3COL} = SP_{3COL}$[Braunstein, Mulet, Pagnani, Weigt and Zecchina, 2003].

SP messages are distributions of tokens

tokens are
subsets of variable alphabet

Generalizing SP: From $SP_{ksat}(\gamma)$ to $Weighted\text{-}SP_{ksat}$

$SP_{ksat}(\gamma)$ Right Message: Constraint $\rightarrow$ Variable

- $\eta_{c\rightarrow v} = \prod_{u\in V(c)\setminus\{v\}} \frac{\Pi^{u}_{u\rightarrow c}}{\Pi^{u}_{u\rightarrow c}+\Pi^{s}_{u\rightarrow c}+\Pi^{*}_{u\rightarrow c}}$

$Weighted\text{-}SP_{ksat}$ Right Message: Constraint $\rightarrow$ Variable

- $\eta_{c\rightarrow v} = \prod_{u\in V(c)\setminus\{v\}} \frac{\Pi^{u}_{u\rightarrow c}}{\Pi^{u}_{u\rightarrow c}+\Pi^{s}_{u\rightarrow c}+\Pi^{*}_{u\rightarrow c}}$

Generalizing SP: From $SP_{ksat}(\gamma)$ to $Weighted\text{-}SP_{ksat}$

$SP_{ksat}(\gamma)$ Right Message: Constraint $\rightarrow$ Variable

- $\eta_{c\rightarrow v} = \prod_{u\in V(c)\setminus\{v\}} \frac{\Pi^{u}_{u\rightarrow c}}{\Pi^{u}_{u\rightarrow c}+\Pi^{s}_{u\rightarrow c}+\Pi^{*}_{u\rightarrow c}}$

$Weighted\text{-}SP_{ksat}$ Right Message: Constraint $\rightarrow$ Variable

- $\eta_{c\rightarrow v} = \prod_{u\in V(c)\setminus\{v\}} \frac{\Pi^{u}_{u\rightarrow c}}{\Pi^{u}_{u\rightarrow c}+\Pi^{s}_{u\rightarrow c}+\Pi^{*}_{u\rightarrow c}}$

Generalizing SP: From $SP_{ksat}(\gamma)$ to $Weighted\text{-}SP_{ksat}$

$SP_{ksat}(\gamma)$ Left Message: Variable $\rightarrow$ Constraint

- $\Pi^{u}_{v\rightarrow c} = \left(1 - \gamma \prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})\right) \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{s}_{v\rightarrow c} = \left(1 - \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})\right) \prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{*}_{v\rightarrow c} = \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v}) \prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$

$Weighted\text{-}SP_{ksat}$ Left Message: Variable $\rightarrow$ Constraint

- $\Pi^{u}_{v\rightarrow c} = \left(1 - \gamma \prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})\right) \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{s}_{v\rightarrow c} = \left(1 - \gamma \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v})\right) \prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$
- $\Pi^{*}_{v\rightarrow c} = \gamma \prod_{b\in C^{s}_{c}(v)}(1-\eta_{b\rightarrow v}) \prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\rightarrow v})$

Generalizing SP: From $SP_{ksat}(\gamma)$ to $\textit{Weighted-SP}_{ksat}$

$SP_{ksat}(\gamma)$ Left Message: Variable $\rightarrow$ Constraint

- $\Pi^{\mathrm{u}}_{v\to c} = \left(1 - \gamma \prod_{b\in C^{\mathrm{u}}_c(v)} (1-\eta_{b\to v})\right) \prod_{b\in C^{\mathrm{s}}_c(v)} (1-\eta_{b\to v})$
- $\Pi^{\mathrm{s}}_{v\to c} = \left(1 - \prod_{b\in C^{\mathrm{s}}_c(v)} (1-\eta_{b\to v})\right) \prod_{b\in C^{\mathrm{u}}_c(v)} (1-\eta_{b\to v})$
- $\Pi^{*}_{v\to c} = \prod_{b\in C^{\mathrm{s}}_c(v)} (1-\eta_{b\to v}) \prod_{b\in C^{\mathrm{u}}_c(v)} (1-\eta_{b\to v})$

$\textit{Weighted-SP}_{ksat}$ Left Message: Variable $\rightarrow$ Constraint

- $\Pi^{\mathrm{u}}_{v\to c} = \left(1 - \gamma \prod_{b\in C^{\mathrm{u}}_c(v)} (1-\eta_{b\to v})\right) \prod_{b\in C^{\mathrm{s}}_c(v)} (1-\eta_{b\to v})$
- $\Pi^{\mathrm{s}}_{v\to c} = \left(1 - \gamma \prod_{b\in C^{\mathrm{s}}_c(v)} (1-\eta_{b\to v})\right) \prod_{b\in C^{\mathrm{u}}_c(v)} (1-\eta_{b\to v})$
- $\Pi^{*}_{v\to c} = \gamma \prod_{b\in C^{\mathrm{s}}_c(v)} (1-\eta_{b\to v}) \prod_{b\in C^{\mathrm{u}}_c(v)} (1-\eta_{b\to v})$

Generalizing SP: From $SP_{ksat}(\gamma)$ to $Weighted\text{-}SP_{ksat}$

$SP_{ksat}(\gamma)$ Summary Message

- $\zeta_v^1 = \left(1 - \gamma \prod_{b \in C^1(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$
- $\zeta_v^0 = \left(1 - \gamma \prod_{b \in C^0(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^1(v)} (1 - \eta_{b \to v})$
- $\zeta_v^* = \gamma \prod_{b \in C^1(v)} (1 - \eta_{b \to v}) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$

$Weighted\text{-}SP_{ksat}$ Summary Message

- $\zeta_v^1 = \left(1 - \gamma \prod_{b \in C^1(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$
- $\zeta_v^0 = \left(1 - \gamma \prod_{b \in C^0(v)} (1 - \eta_{b \to v})\right) \prod_{b \in C^1(v)} (1 - \eta_{b \to v})$
- $\zeta_v^* = \gamma \prod_{b \in C^1(v)} (1 - \eta_{b \to v}) \prod_{b \in C^0(v)} (1 - \eta_{b \to v})$

Generalizing SP: From $SP_{ksat}(\gamma)$ to $Weighted\text{-}SP_{ksat}$

$SP_{ksat}(\gamma)$ Summary Message

- $\zeta_v^1 = \left(1 - \gamma \prod_{b \in C^1(v)} (1 - \eta_{b\to v})\right) \prod_{b \in C^0(v)} (1 - \eta_{b\to v})$
- $\zeta_v^0 = \left(1 - \gamma \prod_{b \in C^0(v)} (1 - \eta_{b\to v})\right) \prod_{b \in C^1(v)} (1 - \eta_{b\to v})$
- $\zeta_v^* = \gamma \prod_{b \in C^1(v)} (1 - \eta_{b\to v}) \prod_{b \in C^0(v)} (1 - \eta_{b\to v})$

$Weighted\text{-}SP_{ksat}$ Summary Message

- $\zeta_v^1 = \left(1 - \gamma \prod_{b \in C^1(v)} (1 - \eta_{b\to v})\right) \prod_{b \in C^0(v)} (1 - \eta_{b\to v})$
- $\zeta_v^0 = \left(1 - \gamma \prod_{b \in C^0(v)} (1 - \eta_{b\to v})\right) \prod_{b \in C^1(v)} (1 - \eta_{b\to v})$
- $\zeta_v^* = \gamma \prod_{b \in C^1(v)} (1 - \eta_{b\to v}) \prod_{b \in C^0(v)} (1 - \eta_{b\to v})$

Generalizing SP: From $SP_{ksat}(\gamma)$ to $\textit{Weighted-SP}_{ksat}$

Lemma

$\textit{Weighted-SP}_{ksat} = SP_{ksat}(\gamma)$

Proof:

- $\Pi^{s}_{v\to c}$ and $\Pi^{*}_{v\to c}$ alway appear together in the form of $\Pi^{s}_{v\to c} + \Pi^{*}_{v\to c}$.
- In $SP_{ksat}(\gamma)$ and in $\textit{Weighted-SP}_{ksat}$, $\Pi^{s}_{v\to c} + \Pi^{*}_{v\to c}$ has the same parametric form, both equal to $\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\to v})$.

Generalizing SP: From $SP_{ksat}(\gamma)$ to $Weighted\text{-}SP_{ksat}$

Lemma

$Weighted\text{-}SP_{ksat} = SP_{ksat}(\gamma)$

Proof:

- $\Pi^{s}_{v\to c}$ and $\Pi^{*}_{v\to c}$ alway appear together in the form of $\Pi^{s}_{v\to c} + \Pi^{*}_{v\to c}$.
- In $SP_{ksat}(\gamma)$ and in $Weighted\text{-}SP_{ksat}$, $\Pi^{s}_{v\to c} + \Pi^{*}_{v\to c}$ has the same parametric form, both equal to $\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\to v})$.

Generalizing SP: From $SP_{ksat}(\gamma)$ to *Weighted*-SP_{ksat}

Lemma

Weighted-$SP_{ksat} = SP_{ksat}(\gamma)$

Proof:

- $\Pi^{s}_{v\to c}$ and $\Pi^{*}_{v\to c}$ alway appear together in the form of $\Pi^{s}_{v\to c} + \Pi^{*}_{v\to c}$.
- In $SP_{ksat}(\gamma)$ and in *Weighted*-SP_{ksat}, $\Pi^{s}_{v\to c} + \Pi^{*}_{v\to c}$ has the same parametric form, both equal to $\prod_{b\in C^{u}_{c}(v)}(1-\eta_{b\to v})$.

Generalizing SP: From $SP_{ksat}(\gamma)$ to $Weighted\text{-}SP_{ksat}$

Lemma

$Weighted\text{-}SP_{ksat} = SP_{ksat}(\gamma)$

Proof:

- $\Pi^{s}_{v\to c}$ and $\Pi^{*}_{v\to c}$ alway appear together in the form of $\Pi^{s}_{v\to c} + \Pi^{*}_{v\to c}$.
- In $SP_{ksat}(\gamma)$ and in $Weighted\text{-}SP_{ksat}$, $\Pi^{s}_{v\to c} + \Pi^{*}_{v\to c}$ has the same parametric form, both equal to $\prod_{b \in C^{u}_{c}(v)}(1 - \eta_{b\to v})$.

Generalizing SP: Interpretation of *Weighted-SP*$_{ksat}$

Modified PTP Interpretation of Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - not necessarily having maximal "freedom"
  - every subset of the common command $\bigcap_{b \in C(v) \setminus \{c\}} t_{b \to v}$ allowed.
  - depending on the common command probabilistically via a "conditional" $\omega(a|b)$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v \to c}(t) := \sum_{t_{C(v) \setminus \{c\} \to v}} \omega\left(t \,\middle|\, \bigcap_{b \in C(v) \setminus \{c\}} t_{b \to v}\right) \prod_{b \in C(v) \setminus \{c\}} \rho_{b \to v}(t_{b \to v})$

Generalizing SP: Interpretation of *Weighted-SP*$_{ksat}$

Modified PTP Interpretation of Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - not necessarily having maximal “freedom”
  - every subset of the common command $\bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}$ allowed.
  - depending on the common command probabilistically via a “conditional” $\omega(a|b)$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $$\lambda_{v\to c}(t) := \sum_{t_{C(v)\setminus\{c\}\to v}} \omega\left(t \,\middle|\, \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}\right) \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(t_{b\to v})$$

Generalizing SP: Interpretation of *Weighted-SP*$_{ksat}$

Modified PTP Interpretation of Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - not necessarily having maximal "freedom"
  - every subset of the common command $\bigcap_{b \in C(v) \setminus \{c\}} t_{b \to v}$ allowed.
  - depending on the common command probabilistically via a "conditional" $\omega(a|b)$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v \to c}(t) := \sum_{t_{C(v)\setminus\{c\} \to v}} \omega\left(t \,\middle|\, \bigcap_{b \in C(v)\setminus\{c\}} t_{b \to v}\right) \prod_{b \in C(v)\setminus\{c\}} \rho_{b \to v}(t_{b \to v})$

Generalizing SP: Interpretation of *Weighted-SP*$_{ksat}$

Modified PTP Interpretation of Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - not necessarily having maximal "freedom"
  - every subset of the common command $\bigcap_{b \in C(v) \setminus \{c\}} t_{b \to v}$ allowed.
  - depending on the common command probabilistically via a "conditional" $\omega(a|b)$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v \to c}(t) := \sum_{t_{C(v) \setminus \{c\} \to v}} \omega\left(t \,\middle|\, \bigcap_{b \in C(v) \setminus \{c\}} t_{b \to v}\right) \prod_{b \in C(v) \setminus \{c\}} \rho_{b \to v}(t_{b \to v})$

Generalizing SP: Interpretation of *Weighted-SP*$_{ksat}$

Modified PTP Interpretation of Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - not necessarily having maximal "freedom"
  - every subset of the common command $\bigcap_{b \in C(v)\setminus\{c\}} t_{b\to v}$ allowed.
  - depending on the common command probabilistically via a "conditional" $\omega(a|b)$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v\to c}(t) := \sum_{t_{C(v)\setminus\{c\}\to v}} \omega\left(t \,\middle|\, \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}\right) \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(t_{b\to v})$

Generalizing SP: Interpretation of *Weighted-SP*$_{ksat}$

Modified PTP Interpretation of Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - not necessarily having maximal "freedom"
  - every subset of the common command $\bigcap_{b \in C(v)\setminus\{c\}} t_{b\to v}$ allowed.
  - depending on the common command probabilistically via a "conditional" $\omega(a|b)$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v\to c}(t) := \sum_{t_{C(v)\setminus\{c\}\to v}} \omega\left(t \middle| \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}\right) \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(t_{b\to v})$

Generalizing SP: Interpretation of *Weighted-SP*$_{ksat}$

Modified PTP Interpretation of Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - not necessarily having maximal "freedom"
  - every subset of the common command $\bigcap_{b \in C(v)\setminus\{c\}} t_{b\to v}$ allowed.
  - depending on the common command probabilistically via a "conditional" $\omega(a|b)$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v\to c}(t) := \sum_{t_{C(v)\setminus\{c\}\to v}} \omega\left(t \,\middle|\, \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}\right) \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(t_{b\to v})$

Generalizing SP: Interpretation of *Weighted-SP*$_{ksat}$

Modified PTP Interpretation of Left Message

- outgoing intention:
  - obeying all incoming commands from upstream
  - not necessarily having maximal "freedom"
  - every subset of the common command $\bigcap_{b \in C(v)\setminus\{c\}} t_{b\to v}$ allowed.
  - depending on the common command probabilistically via a "conditional" $\omega(a|b)$
- left message: the distribution of outgoing intention
  - conditioned on no conflict in incoming commands
  - assuming independence of incoming commands
- $\lambda_{v\to c}(t) := \sum_{t_{C(v)\setminus\{c\}\to v}} \omega\left(t \,\middle|\, \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}\right) \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(t_{b\to v})$

Generalizing SP: Interpretation of *Weighted-SP$_{ksat}$*

“conditional” $\omega(a|b) : \chi_v^* \times (\chi_v^* \cup \{\emptyset\}) \to \mathbb{R}_+$

$$
\begin{aligned}
\omega(a|\{0,1\}) &:= \begin{cases} \gamma, & \text{if } a = \{0,1\} \\ 1-\gamma, & \text{if } a \subset \{0,1\} \end{cases} \\
\omega(a|\{0\}) &:= [a = \{0\}] \\
\omega(a|\{1\}) &:= [a = \{1\}] \\
\omega(a|\emptyset) &:= 0
\end{aligned}
$$

Theorem

Replacing $[a = b]$ in *PTP$_{ksat}$* with $\omega(a|b)$ results in *Weighted-SP$_{ksat}$*.

Generalizing SP: Interpretation of *Weighted-SP$_{ksat}$*

"conditional" $\omega(a|b) : \chi_v^* \times (\chi_v^* \cup \{\emptyset\}) \to \mathbb{R}_+$

$$
\begin{aligned}
\omega(a|\{0,1\}) &:= \begin{cases} \gamma, & \text{if } a = \{0,1\} \\ 1-\gamma, & \text{if } a \subset \{0,1\} \end{cases} \\
\omega(a|\{0\}) &:= [a = \{0\}] \\
\omega(a|\{1\}) &:= [a = \{1\}] \\
\omega(a|\emptyset) &:= 0
\end{aligned}
$$

Theorem

Replacing $[a = b]$ in *PTP$_{ksat}$* with $\omega(a|b)$ results in *Weighted-SP$_{ksat}$*.

Generalizing SP: Weighted-PTP

Weighted-PTP messages

$$\lambda_{v\to c}(t) = \sum_{t_{C(v)\setminus\{c\}\to v}} \omega_v\left(t \,\middle|\, \bigcap_{b\in C(v)\setminus\{c\}} t_{b\to v}\right) \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(t_{b\to v})$$

$$\rho_{c\to v}(t) = \sum_{t_{V(c)\setminus\{v\}\to c}} \left[t = \mathrm{F}_c\left(\prod_{u\in V(c)\setminus\{v\}} t_{u\to c}\right)\right] \times \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(t_{u\to c}).$$

$$\mu_v(t) = \sum_{t_{C(v)\to v}} \omega_v\left(t \,\middle|\, \bigcap_{b\in C(v)} t_{b\to v}\right) \prod_{c\in C(v)} \rho_{c\to v}(t_{c\to v})$$

Generalizing SP: Weighted-PTP

Condition of $\omega_v(a|b) : \chi_v^* \times (\chi_v^* \cup \{\emptyset\}) \rightarrow \mathbb{R}_+$

$\omega_v(a|b) = 0$ if

- $b = \emptyset$ or
- $a \not\subseteq b$

intention may depend on commands

functionally (PTP)

or

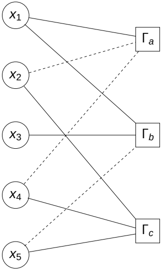
probabilistically (Weighted-PTP)

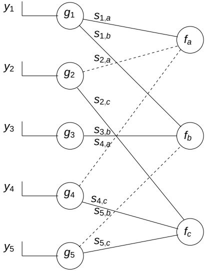
Weighted-PTP

is the most general form

of SP

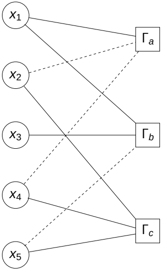
Generalizing MRF

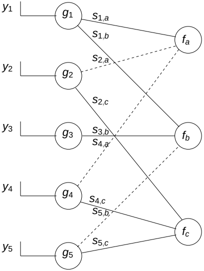




- Forney graph ([Forney, 2001])
- $x_v \to y_v \in \chi_v^*$
- $s_{v,c} = (s_{v,c}^L, s_{v,c}^R) \in \chi_v^* \times \chi_v^*$
  - left state: "intention"
  - right state: "command"

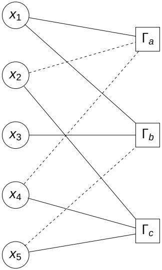
Generalizing MRF

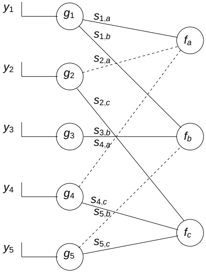




- Forney graph ([Forney, 2001])
- $x_v \rightarrow y_v \in \chi_v^*$
- $s_{v,c} = (s_{v,c}^L, s_{v,c}^R) \in \chi_v^* \times \chi_v^*$
  - left state: "intention"
  - right state: "command"

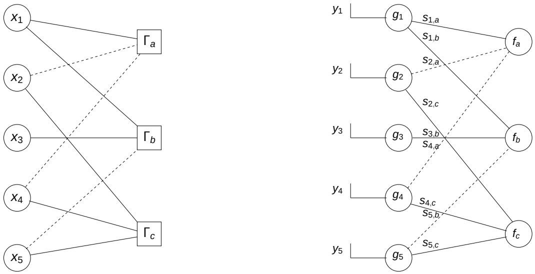
Generalizing MRF





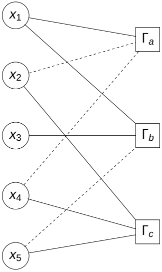
- Forney graph ([Forney, 2001])
- $x_v \longrightarrow y_v \in \chi_v^*$
- $s_{v,c} = (s_{v,c}^L, s_{v,c}^R) \in \chi_v^* \times \chi_v^*$
  - left state: "intention"
  - right state: "command"

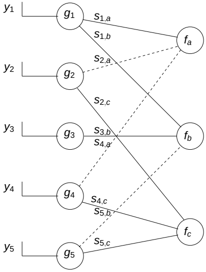
Generalizing MRF



- Forney graph ([Forney, 2001])
- $x_v \longrightarrow y_v \in \chi_v^*$
- $s_{v,c} = (s_{v,c}^L, s_{v,c}^R) \in \chi_v^* \times \chi_v^*$
  - left state: "intention"
  - right state: "command"

Generalizing MRF

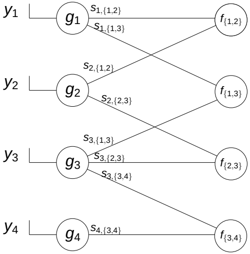




- Forney graph ([Forney, 2001])
- $x_v \longrightarrow y_v \in \chi_v^*$
- $s_{v,c} = (s_{v,c}^L, s_{v,c}^R) \in \chi_v^* \times \chi_v^*$
  - left state: "intention"
  - right state: "command"

Generalizing MRF

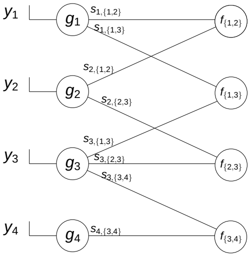
$$F(y_V, s_{V,C}) := \prod_{v \in V} g_v(y_v, s_{v,C(v)}) \prod_{c \in C} f_c(s_{V(c),c})$$



- $f_c(s_{V(c),c}) := \prod_{v \in V(c)} [s^R_{v,c} = \mathrm{F}_c\left(s^L_{V(c)\setminus\{v\},c}\right)]$
- $g_v(y_v, s_{v,C(v)}) := \omega_v\left(y_v \,\middle|\, \bigcap_{c \in C(v)} s^R_{v,c}\right) \prod_{c \in C(v)} [s^L_{v,c} = y_v]$

Generalizing MRF

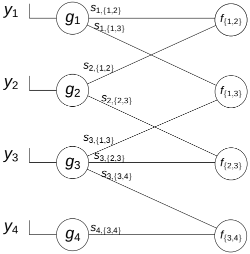
$$F(y_V, s_{V,C}) := \prod_{v \in V} g_v(y_v, s_{v,C(v)}) \prod_{c \in C} f_c(s_{V(c),c})$$



- $f_c(s_{V(c),c}) :=$
  $$\prod_{v \in V(c)} [s^R_{v,c} = \mathrm{F}_c\left(s^L_{V(c)\setminus\{v\},c}\right)]$$
- $g_v(y_v, s_{v,C(v)}) :=$
  $$\omega_v\left(y_v \,\middle|\, \bigcap_{c \in C(v)} s^R_{v,c}\right) \prod_{c \in C(v)} [s^L_{v,c} = y_v]$$

Generalizing MRF

$$F(y_V, s_{V,C}) := \prod_{v \in V} g_v(y_v, s_{v,C(v)}) \prod_{c \in C} f_c(s_{V(c),c})$$



- $f_c(s_{V(c),c}) :=$
  $$\prod_{v \in V(c)} [s^R_{v,c} = \mathrm{F}_c\left(s^L_{V(c)\setminus\{v\},c}\right)]$$
- $g_v(y_v, s_{v,C(v)}) :=$
  $$\omega_v\left(y_v \,\middle|\, \bigcap_{c \in C(v)} s^R_{v,c}\right) \prod_{c \in C(v)} [s^L_{v,c} = y_v]$$

Generalizing MRF

BP Messages on the MRF

$$\lambda_{v\to c}(s^L_{v,c}, s^R_{v,c}) = \sum_{s^R_{v,C(v)\setminus\{c\}}} \omega_v\left(s^L_{v,c} \,\middle|\, \bigcap_{b\in C(v)} s^R_{v,b}\right) \times \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(s^L_{v,c}, s^R_{v,c})$$

$$\rho_{c\to v}(s^L_{v,c}, s^R_{v,c}) = \sum_{s^L_{V(c)\setminus\{v\},c}} \left[s^R_{v,c} = \mathrm{F}_c(s^L_{V(c)\setminus\{v\},c})\right] \times \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(s^L_{u,c}, \mathrm{F}_c(s^L_{V(c)\setminus\{u\},c}))$$

$$\mu_v(y_v) = \sum_{s^R_{v,C(v)}} \omega_v\left(y_v \,\middle|\, \bigcap_{c\in C(v)} s^R_{v,c}\right) \prod_{c\in C(v)} \rho_{c\to v}(y_v, s^R_{v,c}).$$

Generalizing MRF

Example (k-SAT)

$$\begin{aligned}
\lambda_{v\to c}(\mathbf{LL}) &= \prod_{b\in C_c^{\mathrm{u}}(v)} \rho_{b\to v}(\bar{\mathbf{L}}*) \prod_{b\in C_c^{\mathrm{s}}(v)} (\rho_{b\to v}(\mathbf{LL}) + \rho_{b\to v}(\mathbf{L}*)) \\
\lambda_{v\to c}(\bar{\mathbf{L}}*) &= \prod_{b\in C_c^{\mathrm{s}}(v)} \rho_{b\to v}(\bar{\mathbf{L}}*) \left(\prod_{b\in C_c^{\mathrm{u}}(v)} (\rho_{b\to v}(\mathbf{L}*) + \rho_{b\to v}(\mathbf{LL})) - \gamma \prod_{b\in C_c^{\mathrm{u}}(v)} \rho_{b\to v}(\mathbf{L}*) \right) \\
\lambda_{v\to c}(\mathbf{L}*) &= \prod_{b\in C_c^{\mathrm{u}}(v)} \rho_{b\to v}(\bar{\mathbf{L}}*) \left(\prod_{b\in C_c^{\mathrm{s}}(v)} (\rho_{b\to v}(\mathbf{L}*) + \rho_{b\to v}(\mathbf{LL})) - \gamma \prod_{b\in C_c^{\mathrm{s}}(v)} \rho_{b\to v}(\mathbf{L}*) \right) \\
\lambda_{v\to c}(**) &= \gamma \prod_{b\in C_c^{\mathrm{u}}(v)\cup C_c^{\mathrm{s}}(v)} \rho_{b\to v}(**)
\end{aligned}$$

Generalizing MRF

Example (*k*-SAT)

$$
\begin{aligned}
\rho_{c\to v}(\mathbf{LL}) &= \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\bar{\mathbf{L}}*) \\
\rho_{c\to v}(\bar{\mathbf{L}}*) &= \prod_{u\in V(c)\setminus\{v\}} (\lambda_{u\to c}(\mathbf{L}*) + \lambda_{u\to c}(**) + \lambda_{u\to c}(\bar{\mathbf{L}}*)) - \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\bar{\mathbf{L}}*) \\
&\quad + \sum_{u\in V(c)\setminus\{v\}} (\lambda_{u\to c}(\mathbf{LL}) - \lambda_{u\to c}(\mathbf{L}*) - \lambda_{u\to c}(**)) \prod_{w\in V(c)\setminus\{u,v\}} \lambda_{w\to c}(\bar{\mathbf{L}}*) \\
\rho_{c\to v}(\mathbf{L}*) &= \prod_{u\in V(c)\setminus\{v\}} (\lambda_{u\to c}(\mathbf{L}*) + \lambda_{u\to c}(**) + \lambda_{u\to c}(\bar{\mathbf{L}}*)) - \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\bar{\mathbf{L}}*) \\
\rho_{c\to v}(**) &= \prod_{u\in V(c)\setminus\{v\}} (\lambda_{u\to c}(\mathbf{L}*) + \lambda_{u\to c}(**) + \lambda_{u\to c}(\bar{\mathbf{L}}*)) - \prod_{u\in V(c)\setminus\{v\}} \lambda_{u\to c}(\bar{\mathbf{L}}*)
\end{aligned}
$$

Generalizing MRF

Example (k-SAT)

$$
\begin{aligned}
\mu_v(\mathbf{0}) &= \prod_{c \in C^1(v)} \rho_{c \to v}(\bar{\mathbf{L}}*) \left(\prod_{c \in C^0(v)} \left(\rho_{c \to v}(\mathbf{LL}) + \rho_{c \to v}(\mathbf{L}*)\right) - \gamma \prod_{c \in C^0(v)} \rho_{c \to v}(\mathbf{L}*) \right) \\
\mu_v(\mathbf{1}) &= \prod_{c \in C^0(v)} \rho_{c \to v}(\bar{\mathbf{L}}*) \left(\prod_{c \in C^1(v)} \left(\rho_{c \to v}(\mathbf{LL}) + \rho_{c \to v}(\mathbf{L}*)\right) - \gamma \prod_{c \in C^1(v)} \rho_{c \to v}(\mathbf{L}*) \right) \\
\mu_v(*) &= \gamma \prod_{c \in C(v)} \rho_{c \to v}(**)
\end{aligned}
$$

Generalizing MRF

Example (3-COL)

$$
\begin{aligned}
\lambda_{v\to c}(\mathbf{i},\mathbf{ij}) &= \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{i},\mathbf{ij}) + \rho_{b\to v}(\mathbf{i},\mathbf{ik}) + \rho_{b\to v}(\mathbf{i},\mathbf{ijk})) \\
&\quad - \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{i},\mathbf{ij}) + \rho_{b\to v}(\mathbf{i},\mathbf{ijk})) \\
\lambda_{v\to c}(\mathbf{i},\mathbf{ijk}) &= \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{i},\mathbf{ij}) + \rho_{b\to v}(\mathbf{i},\mathbf{ik}) + \rho_{b\to v}(\mathbf{i},\mathbf{ijk})) \\
&\quad - \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{i},\mathbf{ij}) + \rho_{b\to v}(\mathbf{i},\mathbf{ijk})) \\
&\quad - \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{i},\mathbf{ik}) + \rho_{b\to v}(\mathbf{i},\mathbf{ijk})) + \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(\mathbf{ijk},\mathbf{ijk}) \\
\lambda_{v\to c}(\mathbf{ij},\mathbf{ij}) &= \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{ij},\mathbf{ij}) + \rho_{b\to v}(\mathbf{ij},\mathbf{ijk})) \\
\lambda_{v\to c}(\mathbf{ij},\mathbf{ijk}) &= \prod_{b\in C(v)\setminus\{c\}} (\rho_{b\to v}(\mathbf{ij},\mathbf{ij}) + \rho_{b\to v}(\mathbf{ij},\mathbf{ijk})) - \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(\mathbf{ijk},\mathbf{ijk}) \\
\lambda_{v\to c}(\mathbf{ijk},\mathbf{ijk}) &= \prod_{b\in C(v)\setminus\{c\}} \rho_{b\to v}(\mathbf{ijk},\mathbf{ijk})
\end{aligned}
$$

Generalizing MRF

Example (3-COL)

$$
\begin{aligned}
\rho_{c \to v}(\mathbf{i}, \mathbf{ij}) &= \lambda_{V(c)\setminus\{v\} \to c}(\mathbf{k}, \mathbf{jk}) \\
\rho_{c \to v}(\mathbf{i}, \mathbf{ijk}) &= \lambda_{V(c)\setminus\{v\} \to c}(\mathbf{jk}, \mathbf{jk}) \\
\rho_{c \to v}(\mathbf{ij}, \mathbf{ij}) &= \lambda_{V(c)\setminus\{v\} \to c}(\mathbf{k}, \mathbf{ijk}) \\
\rho_{c \to v}(\mathbf{ij}, \mathbf{ijk}) &= \lambda_{V(c)\setminus\{v\} \to c}(\mathbf{ij}, \mathbf{ijk}) + \lambda_{V(c)\setminus\{v\} \to c}(\mathbf{jk}, \mathbf{ijk}) + \lambda_{V(c)\setminus\{v\} \to c}(\mathbf{ik}, \mathbf{ijk}) \\
&\quad + \lambda_{V(c)\setminus\{v\} \to c}(\mathbf{ijk}, \mathbf{ijk}) \\
\rho_{c \to v}(\mathbf{ijk}, \mathbf{ijk}) &= \lambda_{V(c)\setminus\{v\} \to c}(\mathbf{ij}, \mathbf{ijk}) + \lambda_{V(c)\setminus\{v\} \to c}(\mathbf{jk}, \mathbf{ijk}) + \lambda_{V(c)\setminus\{v\} \to c}(\mathbf{ik}, \mathbf{ijk}) \\
&\quad + \lambda_{V(c)\setminus\{v\} \to c}(\mathbf{ijk}, \mathbf{ijk})
\end{aligned}
$$

Generalizing MRF

Example (3-COL)

$$
\begin{aligned}
\mu_v(\mathbf{i}) &= \prod_{c \in C(v)} (\rho^*_{c \to v}(\mathbf{i}, \mathbf{ij}) + \rho_{c \to v}(\mathbf{i}, \mathbf{ik}) + \rho_{c \to v}(\mathbf{i}, \mathbf{ijk})) \\
&\quad - \prod_{c \in C(v)} (\rho_{c \to v}(\mathbf{i}, \mathbf{ij}) + \rho_{c \to v}(\mathbf{i}, \mathbf{ijk})) \\
&\quad - \prod_{c \in C(v)} (\rho_{c \to v}(\mathbf{i}, \mathbf{ik}) + \rho_{c \to v}(\mathbf{i}, \mathbf{ijk})) + \prod_{c \in C(v)} \rho_{c \to v}(\mathbf{i}, \mathbf{ijk}) \\
\mu_v(\mathbf{ij}) &= \prod_{c \in C(v)} (\rho_{c \to v}(\mathbf{ij}, \mathbf{ij}) + \rho_{c \to v}(\mathbf{ij}, \mathbf{ijk})) - \prod_{c \in C(v)} \rho_{c \to v}(\mathbf{ij}, \mathbf{ijk}) \\
\mu_v(\mathbf{ijk}) &= \prod_{c \in C(v)} \rho_{c \to v}(\mathbf{ijk}, \mathbf{ijk})
\end{aligned}
$$

Generalizing BP-to-SP Reduction

- Wait a second, change heading ...

Generalizing BP-to-PTP Reduction

- the k-SAT special case

Theorem

Under the following conditions

- normalizing $\lambda^{(\text{BP})}_{v\to c}(\mathbf{L}*) + \lambda^{(\text{BP})}_{v\to c}(\mathbf{L}*) + \lambda^{(\text{BP})}_{v\to c}(\mathbf{L}*) = 1$
- Initializing $\rho^{(\text{BP})}_{c\to v}(\mathbf{L}*) = \rho^{(\text{BP})}_{c\to v}(\overline{\mathbf{L}}*) = \rho^{(\text{BP})}_{c\to v}(**)$

BP_{ksat}=$Weighted\text{-}PTP_{ksat}$. Specifically,

- $\lambda^{(\text{BP})}_{v\to c}(\mathbf{L}*) \leftrightarrow \lambda^{(\text{PTP})}_{v\to c}(\mathbf{L})$
- $\lambda^{(\text{BP})}_{v\to c}(\overline{\mathbf{L}}*) \leftrightarrow \lambda^{(\text{PTP})}_{v\to c}(\overline{\mathbf{L}})$
- $\lambda^{(\text{BP})}_{v\to c}(**) \leftrightarrow \lambda^{(\text{PTP})}_{v\to c}(*)$
- $\rho^{(\text{BP})}_{c\to v}(\mathbf{L}*) \leftrightarrow \rho^{(\text{PTP})}_{c\to v}(*)$
- $\rho^{(\text{BP})}_{c\to v}(\mathbf{LL}) \leftrightarrow \rho^{(\text{PTP})}_{c\to v}(\mathbf{0}) + \rho^{(\text{PTP})}_{c\to v}(\mathbf{1})$
- $\mu^{(\text{BP})}_{v} \leftrightarrow \mu^{(PTP)}_{v}$

Generalizing BP-to-PTP Reduction

- the k-SAT special case

Theorem

Under the following conditions

- normalizing $\lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) + \lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) + \lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) = 1$
- Initializing $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) = \rho^{(\mathrm{BP})}_{c\to v}(\overline{\mathbf{L}}*) = \rho^{(\mathrm{BP})}_{c\to v}(**)$

BP_{ksat}=$Weighted\text{-}PTP_{ksat}$. Specifically,

- $\lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(\mathbf{L})$
- $\lambda^{(\mathrm{BP})}_{v\to c}(\overline{\mathbf{L}}*) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(\overline{\mathbf{L}})$
- $\lambda^{(\mathrm{BP})}_{v\to c}(**) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(*)$
- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) \leftrightarrow \rho^{(\mathrm{PTP})}_{c\to v}(*)$
- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{LL}) \leftrightarrow \rho^{(\mathrm{PTP})}_{c\to v}(\mathbf{0}) + \rho^{(\mathrm{PTP})}_{c\to v}(\mathbf{1})$
- $\mu^{(\mathrm{BP})}_{v} \leftrightarrow \mu^{(PTP)}_{v}$

Generalizing BP-to-PTP Reduction

- the k-SAT special case

Theorem

Under the following conditions

- normalizing $\lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) + \lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) + \lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) = 1$
- Initializing $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) = \rho^{(\mathrm{BP})}_{c\to v}(\overline{\mathbf{L}}*) = \rho^{(\mathrm{BP})}_{c\to v}(**)$

BP_{ksat}=$Weighted\text{-}PTP_{ksat}$. Specifically,

- $\lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(\mathbf{L})$
- $\lambda^{(\mathrm{BP})}_{v\to c}(\overline{\mathbf{L}}*) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(\overline{\mathbf{L}})$
- $\lambda^{(\mathrm{BP})}_{v\to c}(**) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(*)$
- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) \leftrightarrow \rho^{(\mathrm{PTP})}_{c\to v}(*)$
- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{LL}) \leftrightarrow \rho^{(\mathrm{PTP})}_{c\to v}(\mathbf{0}) + \rho^{(\mathrm{PTP})}_{c\to v}(\mathbf{1})$
- $\mu^{(\mathrm{BP})}_{v} \leftrightarrow \mu^{(PTP)}_{v}$

Generalizing BP-to-PTP Reduction

- the k-SAT special case

Theorem

Under the following conditions

- normalizing $\lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) + \lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) + \lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) = 1$
- Initializing $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) = \rho^{(\mathrm{BP})}_{c\to v}(\overline{\mathbf{L}}*) = \rho^{(\mathrm{BP})}_{c\to v}(**)$

BP_{ksat}=$Weighted\text{-}PTP_{ksat}$. Specifically,

- $\lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(\mathbf{L})$
- $\lambda^{(\mathrm{BP})}_{v\to c}(\overline{\mathbf{L}}*) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(\overline{\mathbf{L}})$
- $\lambda^{(\mathrm{BP})}_{v\to c}(**) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(*)$
- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) \leftrightarrow \rho^{(\mathrm{PTP})}_{c\to v}(*)$
- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{LL}) \leftrightarrow \rho^{(\mathrm{PTP})}_{c\to v}(\mathbf{0}) + \rho^{(\mathrm{PTP})}_{c\to v}(\mathbf{1})$
- $\mu^{(\mathrm{BP})}_{v} \leftrightarrow \mu^{(PTP)}_{v}$

Generalizing BP-to-PTP Reduction

- the k-SAT special case

Theorem

Under the following conditions

- normalizing $\lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) + \lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) + \lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) = 1$
- Initializing $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) = \rho^{(\mathrm{BP})}_{c\to v}(\overline{\mathbf{L}}*) = \rho^{(\mathrm{BP})}_{c\to v}(**)$

BP_{ksat}=$Weighted\text{-}PTP_{ksat}$. Specifically,

- $\lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(\mathbf{L})$
- $\lambda^{(\mathrm{BP})}_{v\to c}(\overline{\mathbf{L}}*) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(\overline{\mathbf{L}})$
- $\lambda^{(\mathrm{BP})}_{v\to c}(**) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(*)$
- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) \leftrightarrow \rho^{(\mathrm{PTP})}_{c\to v}(*)$
- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{LL}) \leftrightarrow \rho^{(\mathrm{PTP})}_{c\to v}(\mathbf{0}) + \rho^{(\mathrm{PTP})}_{c\to v}(\mathbf{1})$
- $\mu^{(\mathrm{BP})}_{v} \leftrightarrow \mu^{(PTP)}_{v}$

Generalizing BP-to-PTP Reduction

- the k-SAT special case

Theorem

Under the following conditions

- normalizing $\lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) + \lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) + \lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) = 1$
- Initializing $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) = \rho^{(\mathrm{BP})}_{c\to v}(\overline{\mathbf{L}}*) = \rho^{(\mathrm{BP})}_{c\to v}(**)$

BP_{ksat}=$Weighted\text{-}PTP_{ksat}$. Specifically,

- $\lambda^{(\mathrm{BP})}_{v\to c}(\mathbf{L}*) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(\mathbf{L})$
- $\lambda^{(\mathrm{BP})}_{v\to c}(\overline{\mathbf{L}}*) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(\overline{\mathbf{L}})$
- $\lambda^{(\mathrm{BP})}_{v\to c}(**) \leftrightarrow \lambda^{(\mathrm{PTP})}_{v\to c}(*)$
- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) \leftrightarrow \rho^{(\mathrm{PTP})}_{c\to v}(*)$
- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{LL}) \leftrightarrow \rho^{(\mathrm{PTP})}_{c\to v}(\mathbf{0}) + \rho^{(\mathrm{PTP})}_{c\to v}(\mathbf{1})$
- $\mu^{(\mathrm{BP})}_{v} \leftrightarrow \mu^{(PTP)}_{v}$

Generalizing BP-to-PTP Reduction

- the k-SAT special case

Theorem

Under the following conditions

- normalizing $\lambda_{v\to c}^{(\mathrm{BP})}(\mathbf{L}*) + \lambda_{v\to c}^{(\mathrm{BP})}(\mathbf{L}*) + \lambda_{v\to c}^{(\mathrm{BP})}(\mathbf{L}*) = 1$
- Initializing $\rho_{c\to v}^{(\mathrm{BP})}(\mathbf{L}*) = \rho_{c\to v}^{(\mathrm{BP})}(\overline{\mathbf{L}}*) = \rho_{c\to v}^{(\mathrm{BP})}(**)$

BP_{ksat}=$Weighted\text{-}PTP_{ksat}$. Specifically,

- $\lambda_{v\to c}^{(\mathrm{BP})}(\mathbf{L}*) \leftrightarrow \lambda_{v\to c}^{(\mathrm{PTP})}(\mathbf{L})$
- $\lambda_{v\to c}^{(\mathrm{BP})}(\overline{\mathbf{L}}*) \leftrightarrow \lambda_{v\to c}^{(\mathrm{PTP})}(\overline{\mathbf{L}})$
- $\lambda_{v\to c}^{(\mathrm{BP})}(**) \leftrightarrow \lambda_{v\to c}^{(\mathrm{PTP})}(*)$
- $\rho_{c\to v}^{(\mathrm{BP})}(\mathbf{L}*) \leftrightarrow \rho_{c\to v}^{(\mathrm{PTP})}(*)$
- $\rho_{c\to v}^{(\mathrm{BP})}(\mathbf{LL}) \leftrightarrow \rho_{c\to v}^{(\mathrm{PTP})}(\mathbf{0}) + \rho_{c\to v}^{(\mathrm{PTP})}(\mathbf{1})$
- $\mu_v^{(\mathrm{BP})} \leftrightarrow \mu_v^{(PTP)}$

Generalizing BP-to-PTP Reduction

- the k-SAT special case

Theorem

Under the following conditions

- normalizing $\lambda_{v\to c}^{(\mathrm{BP})}(\mathbf{L}*) + \lambda_{v\to c}^{(\mathrm{BP})}(\mathbf{L}*) + \lambda_{v\to c}^{(\mathrm{BP})}(\mathbf{L}*) = 1$
- Initializing $\rho_{c\to v}^{(\mathrm{BP})}(\mathbf{L}*) = \rho_{c\to v}^{(\mathrm{BP})}(\overline{\mathbf{L}}*) = \rho_{c\to v}^{(\mathrm{BP})}(**)$

BP_{ksat}=$Weighted$-PTP_{ksat}. Specifically,

- $\lambda_{v\to c}^{(\mathrm{BP})}(\mathbf{L}*) \leftrightarrow \lambda_{v\to c}^{(\mathrm{PTP})}(\mathbf{L})$
- $\lambda_{v\to c}^{(\mathrm{BP})}(\overline{\mathbf{L}}*) \leftrightarrow \lambda_{v\to c}^{(\mathrm{PTP})}(\overline{\mathbf{L}})$
- $\lambda_{v\to c}^{(\mathrm{BP})}(**) \leftrightarrow \lambda_{v\to c}^{(\mathrm{PTP})}(*)$
- $\rho_{c\to v}^{(\mathrm{BP})}(\mathbf{L}*) \leftrightarrow \rho_{c\to v}^{(\mathrm{PTP})}(*)$
- $\rho_{c\to v}^{(\mathrm{BP})}(\mathbf{LL}) \leftrightarrow \rho_{c\to v}^{(\mathrm{PTP})}(\mathbf{0}) + \rho_{c\to v}^{(\mathrm{PTP})}(\mathbf{1})$
- $\mu_{v}^{(\mathrm{BP})} \leftrightarrow \mu_{v}^{(PTP)}$

Generalizing BP-to-PTP Reduction

Proof Sketch:

- The initialization condition holds for all iterations after
- Simplification

Generalizing BP-to-PTP Reduction

Proof Sketch:

- The initialization condition holds for all iterations after
- Simplification

Generalizing BP-to-PTP Reduction

Proof Sketch:

- The initialization condition holds for all iterations after
- Simplification

Question

What does the initialization condition in k-SAT mean?

- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) = \rho^{(\mathrm{BP})}_{c\to v}(\overline{\mathbf{L}}*) = \rho^{(\mathrm{BP})}_{c\to v}(**)$

Answer

Right message should

- only depend on the right state (outgoing command)
- not depend on the left state (incoming intention)

- Recall this is the case in PTP/Weighted-PTP.

Generalizing BP-to-PTP Reduction

Question

What does the initialization condition in k-SAT mean?

- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) = \rho^{(\mathrm{BP})}_{c\to v}(\overline{\mathbf{L}}*) = \rho^{(\mathrm{BP})}_{c\to v}(**)$

Answer

Right message should

- only depend on the right state (outgoing command)
- not depend on the left state (incoming intention)

- Recall this is the case in PTP/Weighted-PTP.

Generalizing BP-to-PTP Reduction

Question

What does the initialization condition in k-SAT mean?

- $\rho^{(\mathrm{BP})}_{c \to v}(\mathbf{L}*) = \rho^{(\mathrm{BP})}_{c \to v}(\overline{\mathbf{L}}*) = \rho^{(\mathrm{BP})}_{c \to v}(**)$

Answer

Right message should

- only depend on the right state (outgoing command)
- not depend on the left state (incoming intention)

- Recall this is the case in PTP/Weighted-PTP.

Generalizing BP-to-PTP Reduction

Question

What does the initialization condition in k-SAT mean?

- $\rho^{(\mathrm{BP})}_{c \to v}(\mathbf{L}*) = \rho^{(\mathrm{BP})}_{c \to v}(\overline{\mathbf{L}}*) = \rho^{(\mathrm{BP})}_{c \to v}(**)$

Answer

Right message should

- only depend on the right state (outgoing command)
- not depend on the left state (incoming intention)

- Recall this is the case in PTP/Weighted-PTP.

Generalizing BP-to-PTP Reduction

Question

What does the initialization condition in k-SAT mean?

- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) = \rho^{(\mathrm{BP})}_{c\to v}(\overline{\mathbf{L}}*) = \rho^{(\mathrm{BP})}_{c\to v}(**)$

Answer

Right message should

- only depend on the right state (outgoing command)
- not depend on the left state (incoming intention)

- Recall this is the case in PTP/Weighted-PTP.

Generalizing BP-to-PTP Reduction

Question

What does the initialization condition in k-SAT mean?

- $\rho^{(\mathrm{BP})}_{c\to v}(\mathbf{L}*) = \rho^{(\mathrm{BP})}_{c\to v}(\overline{\mathbf{L}}*) = \rho^{(\mathrm{BP})}_{c\to v}(**)$

Answer

Right message should

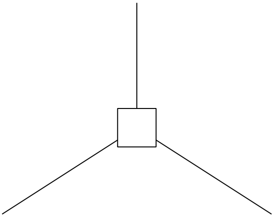
- only depend on the right state (outgoing command)
- not depend on the left state (incoming intention)

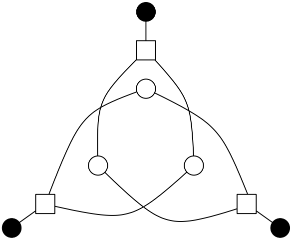
- Recall this is the case in PTP/Weighted-PTP.

Generalizing BP-to-PTP Reduction

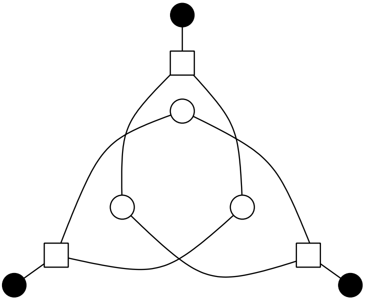
Recall right function in the MRF

$$f_c(s_{V(c),c}) := \prod_{v \in V(c)} \left[s^R_{v,c} = \mathrm{F}_c\left(s^L_{V(c)\setminus\{v\},c} \right) \right]$$





Generalizing BP-to-PTP Reduction



Generalizing BP-to-PTP Reduction

State Decoupling (SD) Condition of BP Message

For all $(s^L_{v,c}, s^R_{v,c})$ in the support of $\rho_{c\to v}(\cdot)$,

$$\rho_{c\to v}(s^L_{v,c}, s^R_{v,c}) = \rho_{c\to v}(s^R_{v,c}, s^R_{v,c}).$$

Generalizing BP-to-PTP Reduction

Test on 3-COL

Lemma

For 3-COL, if SD condition is satisfied in iteration l, it is not satisfied in iteration $l+1$.

What Went Wrong?

- SP is *NOT* BP?
- SD condition is *NOT* the right one?

Generalizing BP-to-PTP Reduction

Test on 3-COL

Lemma

For 3-COL, if SD condition is satisfied in iteration l, it is not satisfied in iteration $l + 1$.

What Went Wrong?

- SP is *NOT* BP?
- SD condition is *NOT* the right one?

Generalizing BP-to-PTP Reduction

Test on 3-COL

Lemma

For 3-COL, if SD condition is satisfied in iteration l, it is not satisfied in iteration $l + 1$.

What Went Wrong?

- SP is *NOT* BP?
- SD condition is *NOT* the right one?

Generalizing BP-to-PTP Reduction

Test on 3-COL

Lemma

For 3-COL, if SD condition is satisfied in iteration l, it is not satisfied in iteration $l + 1$.

What Went Wrong?

- SP is *NOT* BP?
- SD condition is *NOT* the right one?

Generalizing BP-to-PTP Reduction

Test on 3-COL

Lemma

For 3-COL, if SD condition is satisfied in iteration l, it is not satisfied in iteration $l + 1$.

What Went Wrong?

- SP is *NOT* BP?
- SD condition is *NOT* the right one?

Generalizing BP-to-PTP Reduction

"State-Decoupled BP" (SDBP)

- $\rho = BP(\lambda)$
- $\rho^*_{c \to v}(s^L_{v,c}, s^R_{v,c}) := \rho_{c \to v}(s^R_{v,c}, s^R_{v,c})$.
- $\lambda = BP(\rho^*)$

Generalizing BP-to-PTP Reduction

“State-Decoupled BP” (SDBP)

- $\rho = BP(\lambda)$
- $\rho^*_{c \to v}(s^L_{v,c}, s^R_{v,c}) := \rho_{c \to v}(s^R_{v,c}, s^R_{v,c})$.
- $\lambda = BP(\rho^*)$

Generalizing BP-to-PTP Reduction

"State-Decoupled BP" (SDBP)

- $\rho = BP(\lambda)$
- $\rho^*_{c \to v}(s^L_{v,c}, s^R_{v,c}) := \rho_{c \to v}(s^R_{v,c}, s^R_{v,c})$.
- $\lambda = BP(\rho^*)$

Generalizing BP-to-PTP Reduction

Theorem

$SDBP_{3COL}$ = $Weighted\text{-}PTP_{3COL}$. Specifically

$$\rho^{*(\mathrm{SDBP})}_{c \to v}(t, t) \leftrightarrow \rho^{(\mathrm{PTP})}_{c \to v}(t)$$

SP is not BP

Generalizing BP-to-PTP Reduction

Question

SDBP $=$ *Weighted-PTP* in general?

Generalizing BP-to-PTP Reduction

Locally Compatible Constraint

For any $(c - v)$, we say that a token $t_v \in (\chi^*)^V$ is *forceable* by Γ_c if there exists a rectangle $\prod_{u \in V(c) \setminus \{v\}} t_u$ supported by $V(c) \setminus \{v\}$ such that $\mathrm{F}_c\left(\prod_{u \in V(c) \setminus \{v\}} t_u\right) = t_v$. We will denote by $\mathcal{F}_c(v)$ the set of all tokens on v that are forceable by Γ_c. Let $\mathcal{A}_c(v) := \bigcup_{t \in \mathcal{F}_c(v)} t$.
For any $(c - v)$, let $\mathcal{A}_{\sim c}(v)$ be defined by

$$\mathcal{A}_{\sim c}(v) := \bigcap_{b \in C(v) \setminus c} \mathcal{A}_b(v).$$

A constraint Γ_c is said to be locally compatible if for any $v \in V(c)$, any forceable token $t_v \in \mathcal{F}_c(v)$, any rectangle $t' \in \mathrm{F}_c^{-1}(t_v)$ supported by $V(c) \setminus \{v\}$, and any $u \in V(c) \setminus \{v\}$, it holds that

$$\mathcal{A}_{\sim c}(u) \subseteq \mathrm{F}_c\left(t_v \times t'_{:V(c) \setminus \{u,v\}}\right).$$

Generalizing BP-to-PTP Reduction

Locally Compatible Constraint

For any $(c - v)$, we say that a token $t_v \in (\chi^*)^V$ is *forceable* by Γ_c if there exists a rectangle $\prod_{u \in V(c) \setminus \{v\}} t_u$ supported by $V(c) \setminus \{v\}$ such that $\mathsf{F}_c \left(\prod_{u \in V(c) \setminus \{v\}} t_u \right) = t_v$. We will denote by $\mathcal{F}_c(v)$ the set of all tokens on v that are forceable by Γ_c. Let $\mathcal{A}_c(v) := \bigcup_{t \in \mathcal{F}_c(v)} t$.
For any $(c - v)$, let $\mathcal{A}_{\sim c}(v)$ be defined by

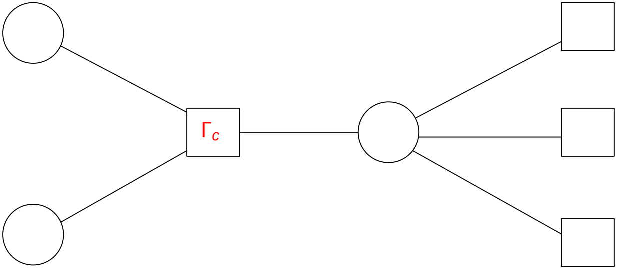
$$\mathcal{A}_{\sim c}(v) := \bigcap_{b \in C(v) \setminus c} \mathcal{A}_b(v).$$

A constraint Γ_c is said to be locally compatible if for any $v \in V(c)$, any forceable token $t_v \in \mathcal{F}_c(v)$, any rectangle $t' \in \mathsf{F}_c^{-1}(t_v)$ supported by $V(c) \setminus \{v\}$, and any $u \in V(c) \setminus \{v\}$, it holds that

$$\mathcal{A}_{\sim c}(u) \subseteq \mathsf{F}_c \left(t_v \times t'_{:V(c) \setminus \{u,v\}} \right).$$

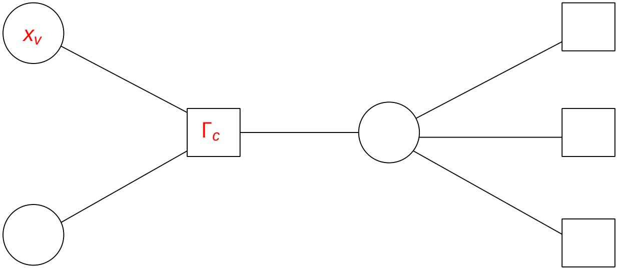
Generalizing BP-to-PTP Reduction

- Locally Compatible Condition



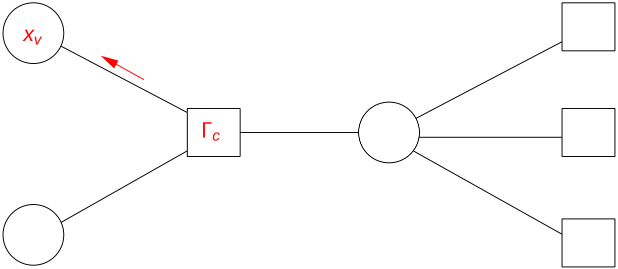
Generalizing BP-to-PTP Reduction

- Locally Compatible Condition



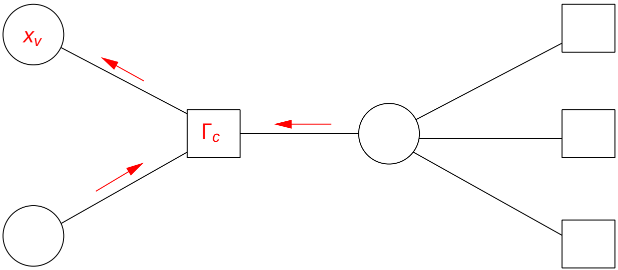
Generalizing BP-to-PTP Reduction

- Locally Compatible Condition



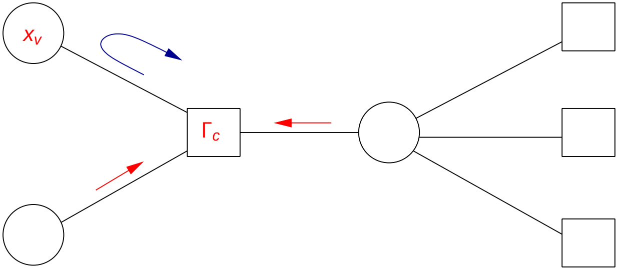
Generalizing BP-to-PTP Reduction

- Locally Compatible Condition



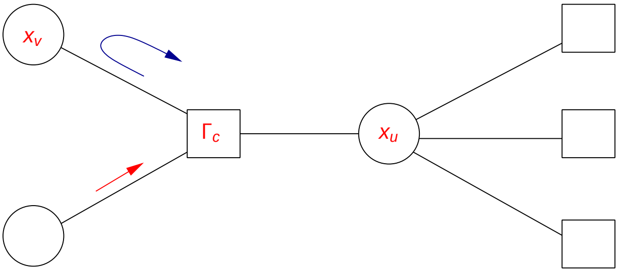
Generalizing BP-to-PTP Reduction

- Locally Compatible Condition



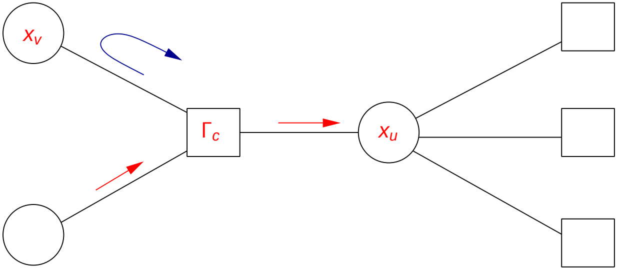
Generalizing BP-to-PTP Reduction

- Locally Compatible Condition



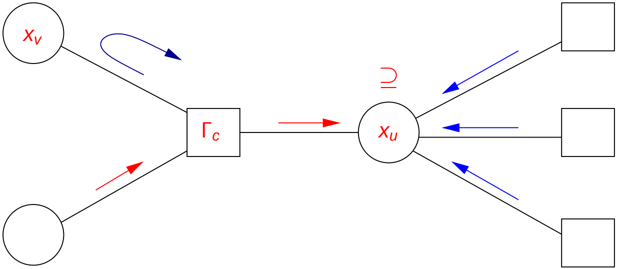
Generalizing BP-to-PTP Reduction

- Locally Compatible Condition



Generalizing BP-to-PTP Reduction

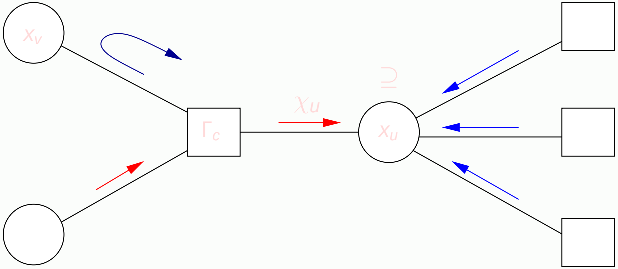
- Locally Compatible Condition



Theorem

SDBP $=$ *Weighted-PTP* if and only if every constraint is locally compatible.

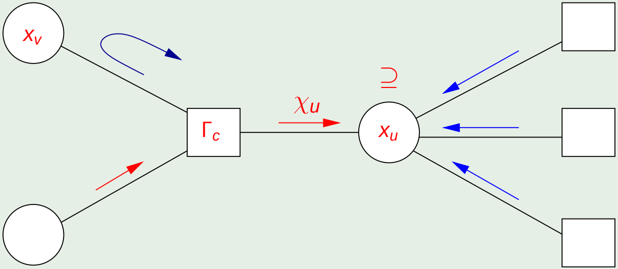
Example (k-SAT and 3-COL)



Theorem

SDBP $=$ *Weighted-PTP* if and only if every constraint is locally compatible.

Example (k-SAT and 3-COL)



is SP BP?

why should SP be BP?