Improving Max-Sum through Decimation to Solve Cyclic Distributed Constraint Optimization Problems

J. Cerquides[⋆] R. Emonet[†] **G. Picard**[‡] J.A. Rodriquez-Aguilar[⋆]

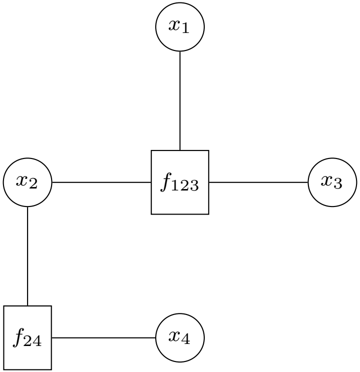
[⋆] IIIA-CSIC, Campus UAB, 08193 Cerdanyola, Catalonia, Spain
{cerquide,jar}@iiia.csic.es

[†] Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d'Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France
remi.emonet@univ-st-etienne.fr

[‡] Mines Saint-Etienne, Univ Lyon, Univ Jean Monnet, IOGS, CNRS, UMR 5516 LHC, Institut Henri Fayol, Departement ISI, F - 42023 Saint-Etienne France
picard@emse.fr

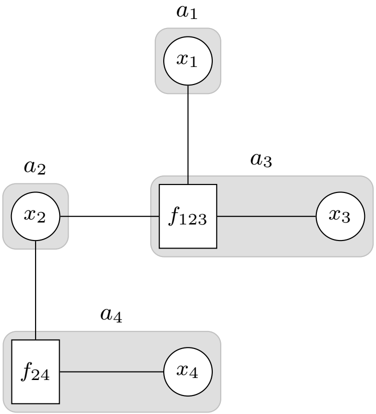
What problems are we dealing with?

Problems represented as factor graphs



What problems are we dealing with?

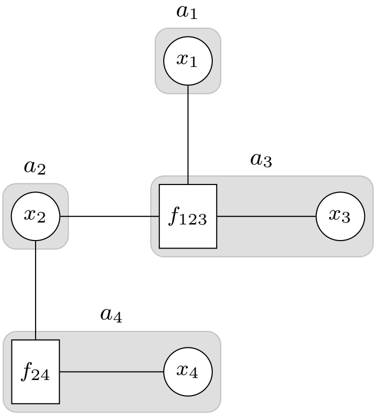
Problems represented as factor graphs



Distributed Constraint Optimization Problems (DCOPs)

What problems are we dealing with?

Problems represented as factor graphs

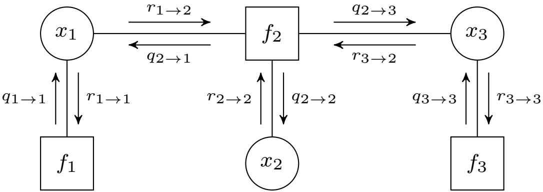


Distributed Constraint Optimization Problems (DCOPs)

One possible and often efficient solution method to find $\max_{\mathcal{X}} \sum_{m=1}^{M} f_m(\mathcal{X}_m)$: Max-Sum [Farinelli et al., 2008]

What's Max-Sum?

Belief-Propagation-based message passing algorithm



Each variable/factor sends messages:

$$q_{n\to m}(x_n) = \alpha_{nm} + \sum_{m'\in\mathcal{V}(n)\setminus m} r_{m'\to n}(x_n) \tag{1}$$

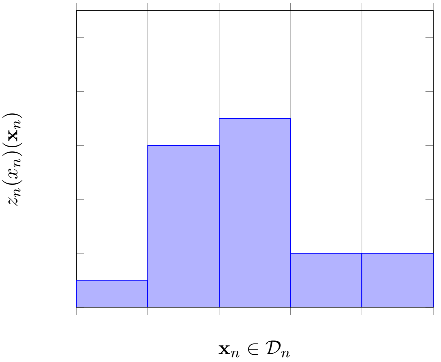
$$r_{m\to n}(x_n) = \max_{\mathcal{X}_m\setminus n}\left(f_m(\mathcal{X}_m) \sum_{n'\in\mathcal{F}(m)\setminus n} q_{n'\to m}(x_{n'})\right) \tag{2}$$

and computes a marginal function:

$$z_n(x_n) = \max_{\mathcal{X}_m\setminus n} \sum_{m=1}^{M} f_m(\mathcal{X}_m) \tag{3}$$

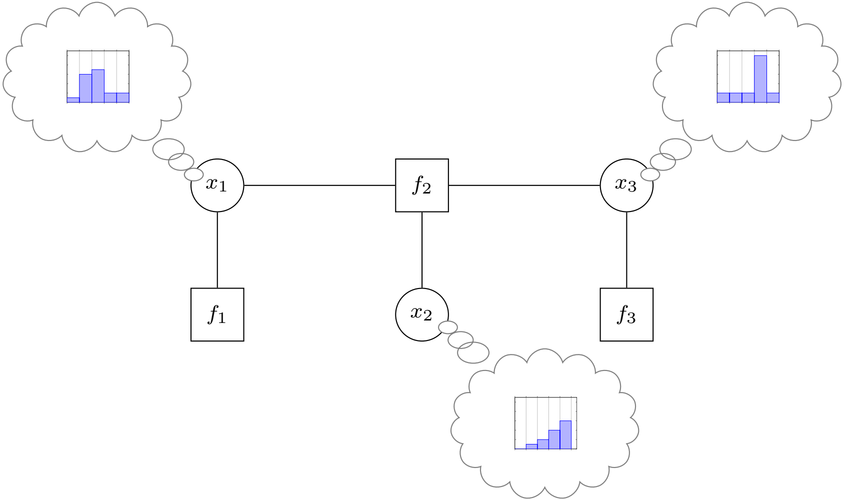
What’s Max-Sum? (cont.)

In the end, each variable acquires belief about its influence on the overall objective $\rightarrow$ decoding to get the solution ($\operatorname{argmax} z_n(x_n)$)



What's Max-Sum? (cont.)

In the end, each variable acquires belief about its influence on the overall objective $\rightarrow$ decoding to get the solution ($\operatorname{argmax} z_n(x_n)$)



What's the problem with Max-Sum (MS)?

- On tree-shaped FGs: MS proven to converge to optimal solutions
- In more general cyclic settings:
  - May converge to non optimal solutions
  - May not converge at all

Here, convergence means the marginal functions do not change for a while...

What's the problem with Max-Sum (MS)?

- On tree-shaped FGs: MS proven to converge to optimal solutions
- In more general cyclic settings:
  - May converge to non optimal solutions
  - May not converge at all

Here, convergence means the marginal functions do not change for a while...

Several approaches to handle loops in MS

- Bounded MS [Rogers et al., 2011]
- Max-Sum_AD_VP [Zivan et al., 2017]

What's the problem with Max-Sum (MS)?

- On tree-shaped FGs: MS proven to converge to optimal solutions
- In more general cyclic settings:
  - May converge to non optimal solutions
  - May not converge at all

Here, convergence means the marginal functions do not change for a while...

Several approaches to handle loops in MS

- Bounded MS [Rogers et al., 2011]
- Max-Sum_AD_VP [Zivan et al., 2017]

But let's also have a look at...

- Decimation [Montanari et al., 2007], coming from statistical physics to solve k-satisfiability loopy problems

What's decimation?

Simple principle= alternating belief-propagation (BP) and assignment of values to some variables depending on their marginal value, until all variables have been assigned a value

What's decimation?

Simple principle= alternating belief-propagation (BP) and assignment of values to some variables depending on their marginal value, until all variables have been assigned a value

Example (implementing [Montanari et al., 2007])

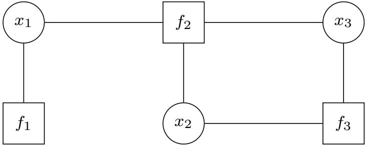
BP $\rightarrow$ choosing a variable randomly $\rightarrow$ sampling the value of the variable depending on its marginal $\rightarrow$ BP $\rightarrow$ ...

What's decimation?

Simple principle= alternating belief-propagation (BP) and assignment of values to some variables depending on their marginal value, until all variables have been assigned a value

Example (implementing [Montanari et al., 2007])

BP $\rightarrow$ choosing a variable randomly $\rightarrow$ sampling the value of the variable depending on its marginal $\rightarrow$ BP $\rightarrow$ ...

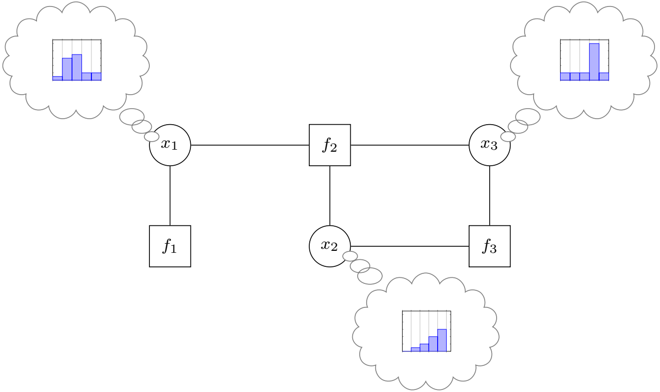


What's decimation?

Simple principle= alternating belief-propagation (BP) and assignment of values to some variables depending on their marginal value, until all variables have been assigned a value

Example (implementing [Montanari et al., 2007])

BP $\rightarrow$ choosing a variable randomly $\rightarrow$ sampling the value of the variable depending on its marginal $\rightarrow$ BP $\rightarrow$ ...

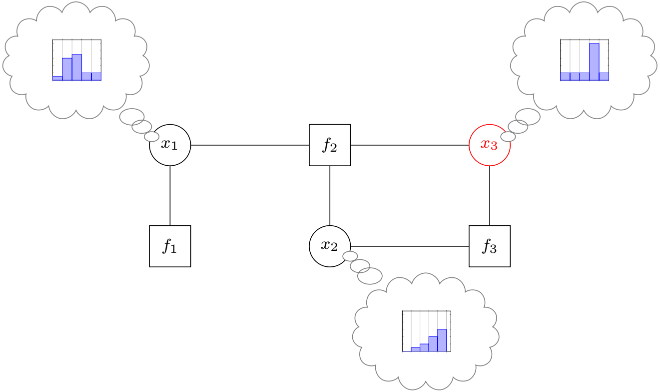


What's decimation?

Simple principle= alternating belief-propagation (BP) and assignment of values to some variables depending on their marginal value, until all variables have been assigned a value

Example (implementing [Montanari et al., 2007])

BP $\rightarrow$ choosing a variable randomly $\rightarrow$ sampling the value of the variable depending on its marginal $\rightarrow$ BP $\rightarrow$ ...

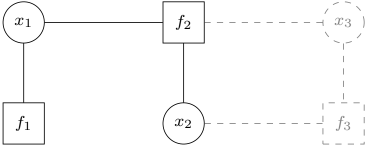


What's decimation?

Simple principle= alternating belief-propagation (BP) and assignment of values to some variables depending on their marginal value, until all variables have been assigned a value

Example (implementing [Montanari et al., 2007])

BP $\rightarrow$ choosing a variable randomly $\rightarrow$ sampling the value of the variable depending on its marginal $\rightarrow$ BP $\rightarrow$ ...

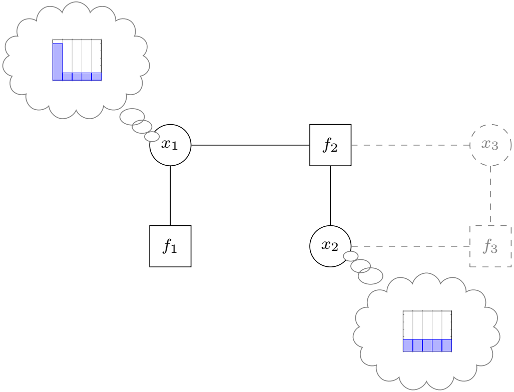


What's decimation?

Simple principle= alternating belief-propagation (BP) and assignment of values to some variables depending on their marginal value, until all variables have been assigned a value

Example (implementing [Montanari et al., 2007])

BP → choosing a variable randomly → sampling the value of the variable depending on its marginal → BP → ...

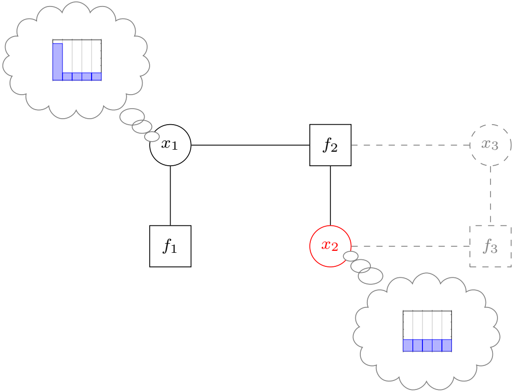


What's decimation?

Simple principle= alternating belief-propagation (BP) and assignment of values to some variables depending on their marginal value, until all variables have been assigned a value

Example (implementing [Montanari et al., 2007])

BP → choosing a variable randomly → sampling the value of the variable depending on its marginal → BP → ...

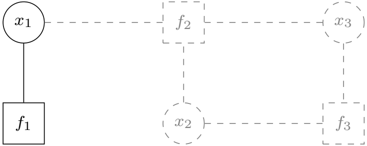


What’s decimation?

Simple principle= alternating belief-propagation (BP) and assignment of values to some variables depending on their marginal value, until all variables have been assigned a value

Example (implementing [Montanari et al., 2007])

BP $\rightarrow$ choosing a variable randomly $\rightarrow$ sampling the value of the variable depending on its marginal $\rightarrow$ BP $\rightarrow$ ...

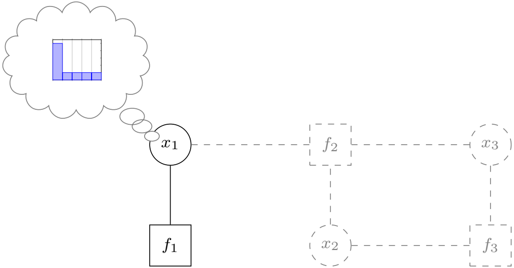


What's decimation?

Simple principle= alternating belief-propagation (BP) and assignment of values to some variables depending on their marginal value, until all variables have been assigned a value

Example (implementing [Montanari et al., 2007])

BP $\rightarrow$ choosing a variable randomly $\rightarrow$ sampling the value of the variable depending on its marginal $\rightarrow$ BP $\rightarrow$ ...

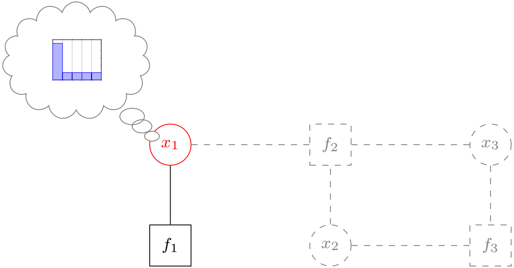


What's decimation?

Simple principle= alternating belief-propagation (BP) and assignment of values to some variables depending on their marginal value, until all variables have been assigned a value

Example (implementing [Montanari et al., 2007])

BP $\rightarrow$ choosing a variable randomly $\rightarrow$ sampling the value of the variable depending on its marginal $\rightarrow$ BP $\rightarrow$ ...

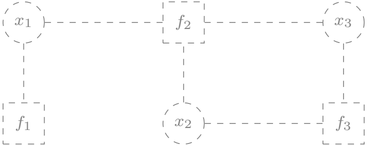


What's decimation?

Simple principle= alternating belief-propagation (BP) and assignment of values to some variables depending on their marginal value, until all variables have been assigned a value

Example (implementing [Montanari et al., 2007])

BP $\rightarrow$ choosing a variable randomly $\rightarrow$ sampling the value of the variable depending on its marginal $\rightarrow$ BP $\rightarrow$ ...



Let's gerenalize and try to use decimation in Max-Sum

To install decimation in a BP-based solution method, we need to identify

1. when decimation should be triggered
   - each time step, each n time steps, once a loop is detected, ...
2. the subset of variables to decimate
   - one variable randomly, one variable with some properties, several variables, ...
3. the values to assign to decimated variable(s)
   - sampling on marginal values, most determined value, ...

We call a decimation policy any combination of (1), (2) and (3)

Our idea here is to apply different decimation policies to Max-Sum

Let’s gerenalize and try to use decimation in Max-Sum

To install decimation in a BP-based solution method, we need to identify

1. when decimation should be triggered
   - each time step, each n time steps, once a loop is detected, ...
2. the subset of variables to decimate
   - one variable randomly, one variable with some properties, several variables, ...
3. the values to assign to decimated variable(s)
   - sampling on marginal values, most determined value, ...

We call a decimation policy any combination of (1), (2) and (3)

Our idea here is to apply different decimation policies to Max-Sum

$\rightarrow$ A generic decimation framework for Max-Sum, a.k.a DeciMaxSum

DeciMaxSum as an algorithm

Data: A factor graph $FG = \langle \mathcal{X}, \mathcal{C}, E \rangle$, a decimation policy $\pi = \langle \Theta, \Phi, \Upsilon, \Lambda \rangle$
Result: A feasible assignment $\mathcal{X}^*$

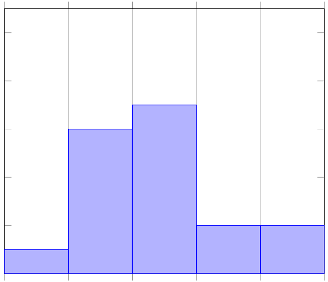
```
initialize BP messages
𝒰 ← ∅
while 𝒰 ≠ 𝒳 do
    run BP until decimation triggers, i.e. Θ(FG^t) = 1
    choose variables to decimate, 𝒳′ = {x_i ∈ Φ(FG^t) | Υ(x_i, FG^t)}
    for x_i ∈ 𝒳′ do
        x_i ← Λ(x_i, FG^t)
        𝒰 ← 𝒰 ∪ {x_i}
        simplify FG^t                                   // remove variables, slice factors
return 𝒳* by decoding 𝒰
```

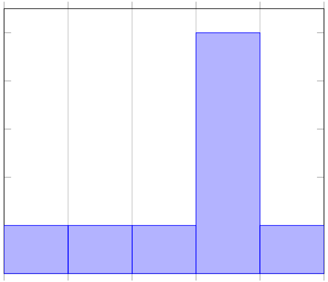
Implementing [Montanari et al., 2007] in DeciMaxSum

1. decimate once BP converges (or halt after some time limit)
2. choose on random variable within the whole set of non decimated variables
3. sampling the value wrt the marginal values

Implementing [Montanari et al., 2007] in DeciMaxSum

1. decimate once BP converges (or halt after some time limit)
2. choose on random variable within the whole set of non decimated variables
3. sampling the value wrt the marginal values





Implementing [Mooij, 2010] in DeciMaxSum

1. decimate once BP converges (or halt after some time limit)
2. choose the most determined variable, i.e. the lowest entropy H on marginal values, within the whole set of non decimated variables

$$H(z_k(x_k)) = -\sum_{d \in \mathcal{D}_k} z_k(x_k)(d) \log(z_k(x_k)(d))$$

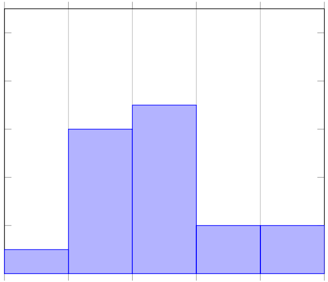
3. choose the value with highest marginal value ($\operatorname{argmax}_{d \in \mathcal{D}_i} z_i(x_i)(d)$)

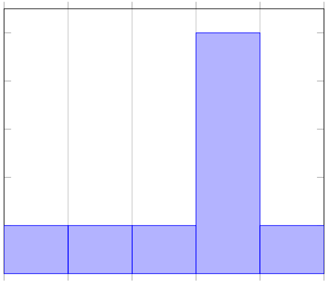
Implementing [Mooij, 2010] in DeciMaxSum

1. decimate once BP converges (or halt after some time limit)
2. choose the most determined variable, i.e. the lowest entropy H on marginal values, within the whole set of non decimated variables

$$H(z_k(x_k)) = -\sum_{d \in \mathcal{D}_k} z_k(x_k)(d) \log(z_k(x_k)(d))$$

3. choose the value with highest marginal value ($\mathrm{argmax}_{d \in \mathcal{D}_i} z_i(x_i)(d)$)





And many more combinations...

- DeciMaxSum (2-periodic, min-entropy, deterministic)
- DeciMaxSum (3-periodic, min-entropy, deterministic)
- DeciMaxSum (4-periodic, min-entropy, deterministic)
- DeciMaxSum (5-periodic, min-entropy, deterministic)
- DeciMaxSum (10-periodic, min-entropy, deterministic)
- DeciMaxSum (20-periodic, min-entropy, deterministic)
- DeciMaxSum (100-periodic, min-entropy, deterministic)

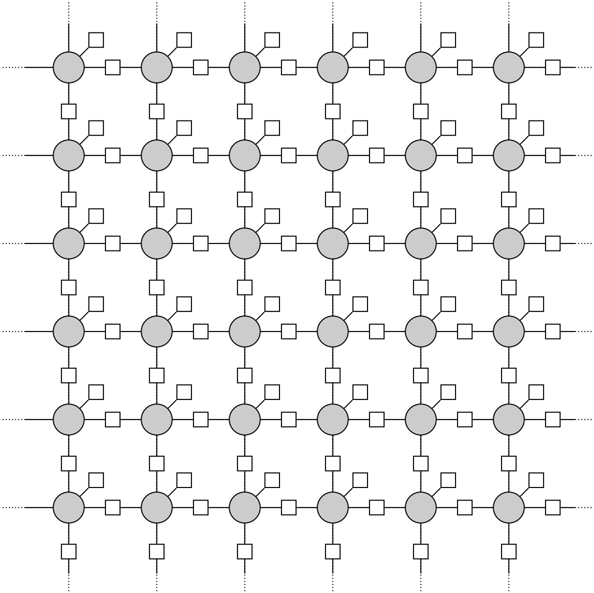
- DeciMaxSum (10-periodic, random, sampling)
- DeciMaxSum (100-periodic, random, sampling)

- DeciMaxSum (periodic, min-entropy, deterministic)

- DeciMaxSum (convergence, min-entropy, deterministic)

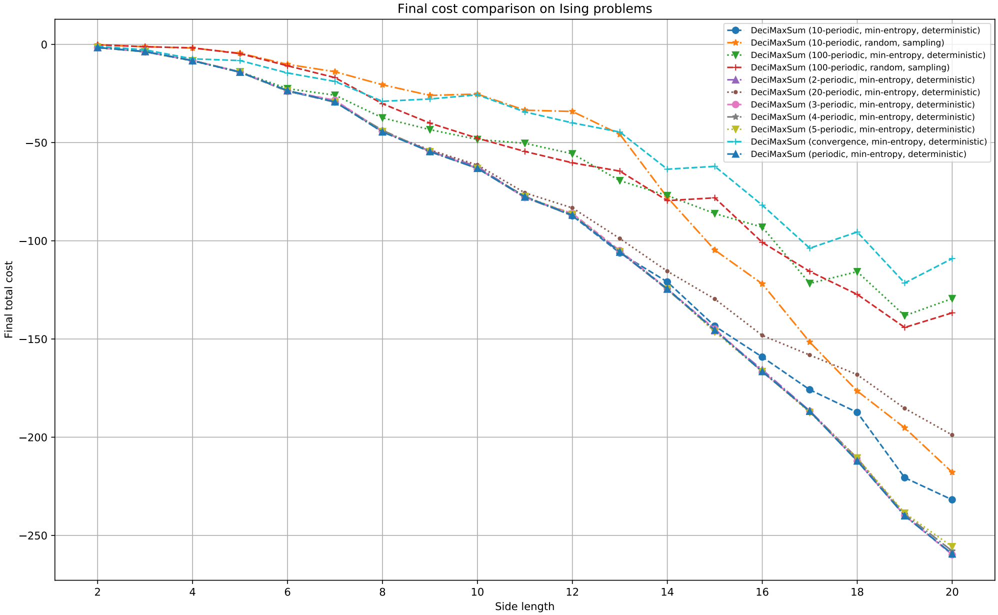
- MaxSum
- Montanari-Decimation
- Mooij-Decimation
- ...

Benchmarking on a very cyclic problem: the Ising model

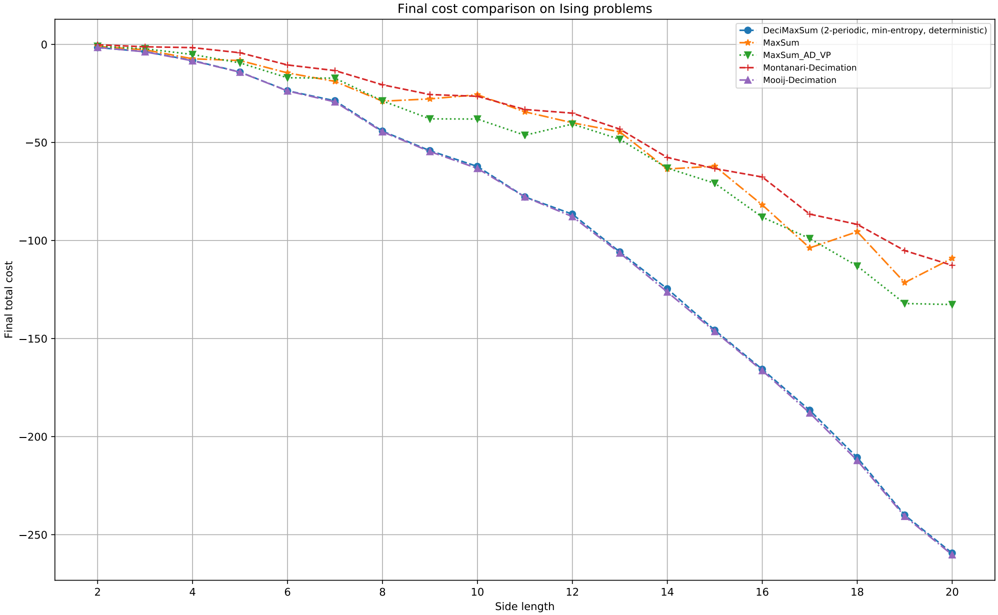


- toroidal grid structure
- boolean variables x_i's
- unary costs r_i's
- binary constraints r_{ij}'s

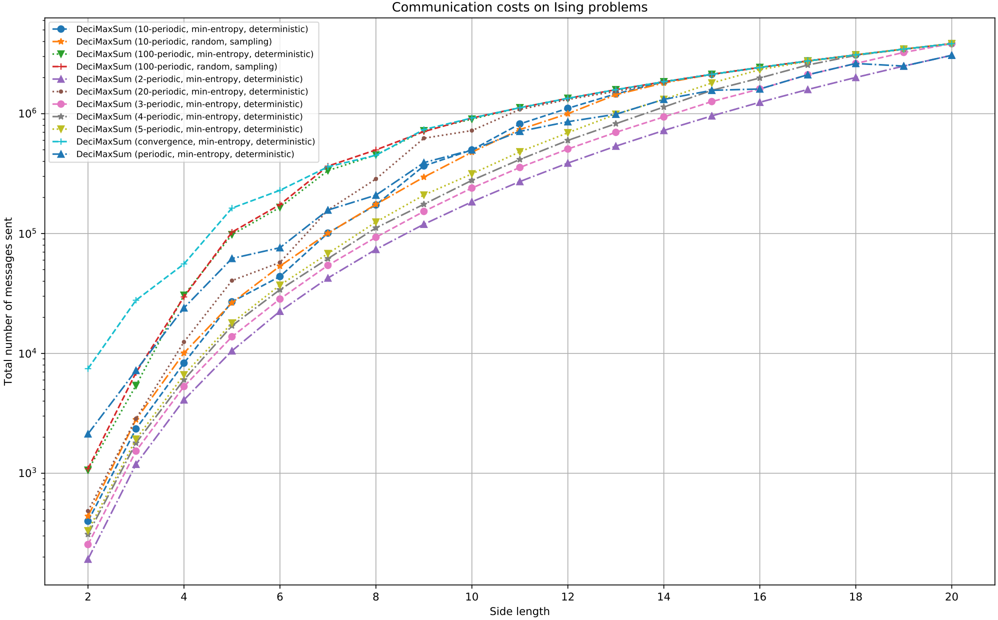
Quality of solutions



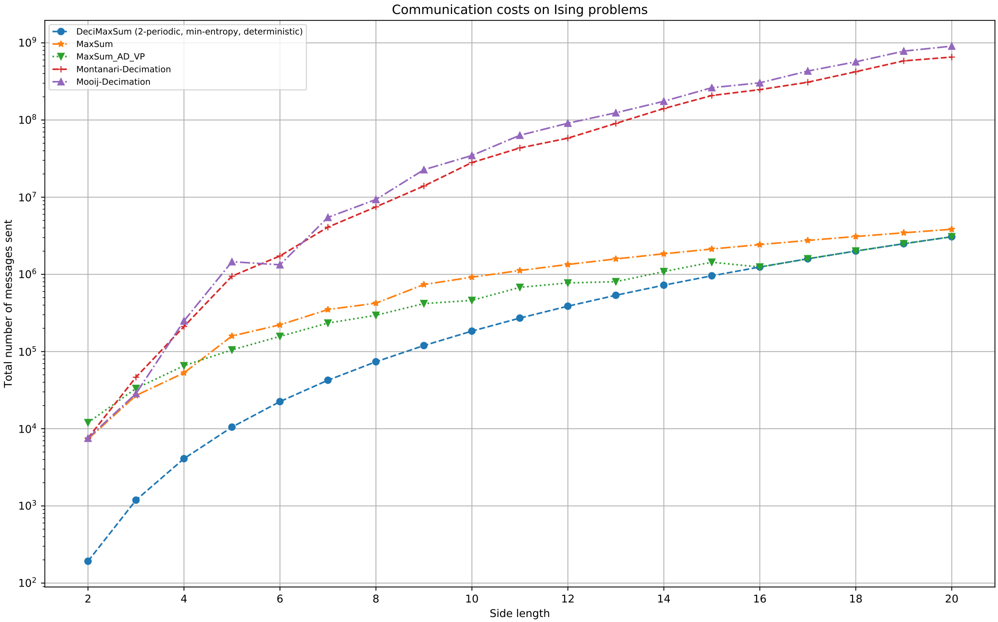
Quality of solutions (cont.)



Communication load



Communication load (cont.)



Conclusions

To sum up

- We have proposed a generic framework to integrate decimation mechanism into MaxSum/BP algorithms
- On very cyclic problems (Ising model), fast decimation based on marginal function entropy and deterministic value assignment showed very good quality solutions, with many less messages
- Decimation $\equiv$ decoding at runtime?

Many ways to go

- Many more policies are possible
  - ex: decimation once a loop is detected
  - ex: alternating deterministic and non deterministic value assignment
- How does decimation behave on less cyclic but less regular problems?
- How does decimation behave on non boolean settings?

Improving Max-Sum through Decimation to Solve Cyclic Distributed Constraint Optimization Problems

J. Cerquides[*] R. Emonet[†] **G. Picard**[‡] J.A. Rodriquez-Aguilar[*]

[*] IIIA-CSIC, Campus UAB, 08193 Cerdanyola, Catalonia, Spain
`{cerquide,jar}@iiia.csic.es`

[†] Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d'Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France
`remi.emonet@univ-st-etienne.fr`

[‡] Mines Saint-Etienne, Univ Lyon, Univ Jean Monnet, IOGS, CNRS, UMR 5516 LHC, Institut Henri Fayol, Departement ISI, F - 42023 Saint-Etienne France
`picard@emse.fr`

References

FARINELLI, A., A. ROGERS, A. PETCU, and N. R. JENNINGS (2008). "Decentralised Coordination of Low-power Embedded Devices Using the Max-sum Algorithm". In: *International Conference on Autonomous Agents and Multiagent Systems (AAMAS'08)*, pp. 639–646. ISBN: 978-0-9817381-1-6. URL: http://dl.acm.org/citation.cfm?id=1402298.1402313.

MONTANARI, A., F. RICCI-TERSENGHI, and G. SEMERJIAN (2007). "Solving Constraint Satisfaction Problems through Belief Propagation-guided decimation". In: *CoRR* abs/0709.1667. URL: http://arxiv.org/abs/0709.1667.

MOOIJ, Joris M. (2010). "libDAI: A Free and Open Source C++ Library for Discrete Approximate Inference in Graphical Models". In: *Journal of Machine Learning Research* 11, pp. 2169–2173. URL: http://www.jmlr.org/papers/volume11/mooij10a/mooij10a.pdf.

ROGERS, A., A. FARINELLI, R. STRANDERS, and N.R. JENNINGS (2011). "Bounded approximate decentralised coordination via the max-sum algorithm". In: *Artificial Intelligence* 175.2, pp. 730 –759. ISSN: 0004-3702. DOI: http://dx.doi.org/10.1016/j.artint.2010.11.001. URL: http://www.sciencedirect.com/science/article/pii/S0004370210001803.

ZIVAN, Roie, Tomer PARASH, Liel COHEN, Hilla PELED, and Steven OKAMOTO (2017). "Balancing exploration and exploitation in incomplete Min/Max-sum inference for distributed constraint optimization". In: *Autonomous Agents and Multi-Agent Systems* 31.5, pp. 1165–1207. ISSN: 1573-7454. DOI: 10.1007/s10458-017-9360-1. URL: https://doi.org/10.1007/s10458-017-9360-1.