Loop Calculus for Satisfiability

LAUR 08-0321

Lukas Kroc
Cornell University
kroc@cs.cornell.edu

Michael Chertkov
Los Alamos National Lab
chertkov@lanl.gov

1. Introduction

- **Problems as Factor Graphs**
  - Many problems can be naturally cast as inference in factor graphs. We focus on estimating the *partition function* $Z = \sum_{\vec{x}} w(\vec{x})$ e.g. number of solutions of a problem.
- **Loopy Belief Propagation as a Heuristic**
  - BP is exact on tree factor graphs, and often provides a surprisingly good approximation on other topologies.
- **Loop Series as Correction to BP**
  - Loop Calculus is a way to express the exact value of Z as an (exponentially long) sum with BP's estimate as a leading term.

2. Main Idea

Incremental improvement to BP's estimate with a **tunable** efficiency/accuracy **tradeoff**.

3. Research Agenda

A) incremental improvement of BP's estimate for number of solutions of a SAT problem

B) efficient partial summation of the loop series

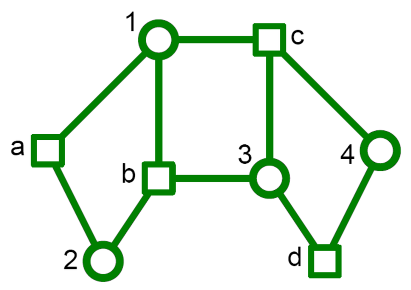
C) empirical tests for applicability of the approach in the SAT domain

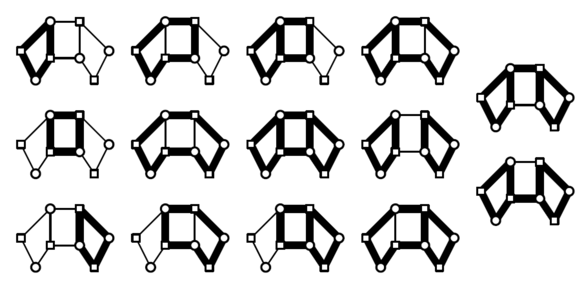
A) Loop Calculus for SAT

- **Exact value** expressed as $Z = Z_0 \left(1 + \sum_C r_C\right)$

  where Z_0 is BP's estimate and r_C a generalized loop contribution (sum of which is **loop series**)

- **Generalized loops** are subgraphs of the factor graph with no degree 1 nodes





- **Loop contributions** are computed entirely from BP's beliefs as

$$r_C = \left(\prod_{i \in \mathcal{C}} \mu_i\right)\left(\prod_{\alpha \in \mathcal{G}} \mu_{\alpha;\mathcal{C}}\right)$$

with

$$\mu_i = \frac{\sum_{\sigma_i} (\sigma_i - m_i)^{q_i} b_i(\sigma_i)}{(1 - m_i^2)^{q_i}}$$

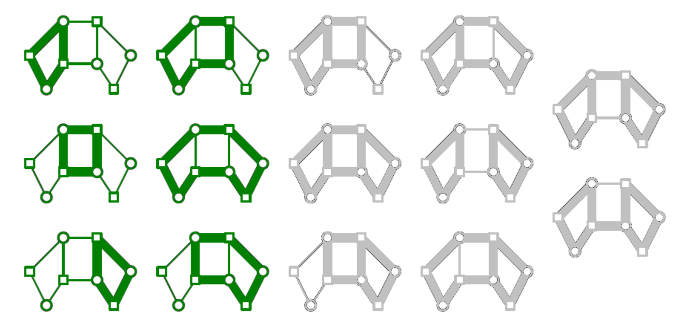
$$\mu_{\alpha;\mathcal{C}} = \sum_{\boldsymbol{\sigma}_\alpha} b_\alpha(\boldsymbol{\sigma}_\alpha) \prod_{i \in \alpha;\mathcal{C}} (\sigma_i - m_i)$$

and magnetization $m_i \equiv \sum_{\sigma_i} \sigma_i b_i(\sigma_i)$ and variable node degree q_i

- **Drawback:** Loop contributions can be both positive and negative, resulting in non monotonic improvements.

B) Summation of Loop Series

- Full summation requires an exponential amount of work ⇒ do **partial summation.**
- Generalized loops with **low node degrees** have in general higher weight than high-degree loops.
- Sum weights of **simple loops**: connected graphs with node degrees exactly 2.



- For simple loops, the loop weights simplify to:

$$r_C = \prod_{\alpha \in C} \tilde{\mu}_{\alpha,ij}$$

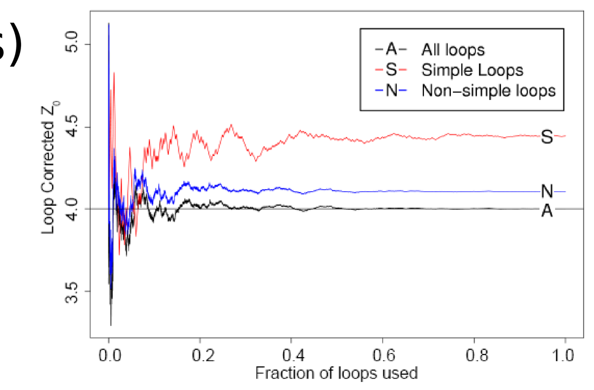
$$\tilde{\mu}_{\alpha,ij} = \frac{\mu_\alpha}{\sqrt{1 - m_i^2}\sqrt{1 - m_j^2}}$$

  for *i,j* adjacent to α in *C*

- Approach based on **A* search algorithm** can efficiently list simple loops in order of decreasing weight (in absolute value):
  - Search for a loop of minimal weight in a derived graph with labeled edges, such that each edge label appears at most once.
  - The algorithm is not polynomial in the worst case, but works well in practice.

C) Empirical Tests

- Tiny formula (4 vars) where all loops can be listed: there are 682 simple loops and 330,000 non-simple loops.



- Larger formulas (100 vars): improvement is possible, but mainly for easier instances.

α	exact count Z	LBP's Z_0	$Z_0\,(1 + \sum r_c)$
3.0	1.97×10^{10}	4.34×10^{10}	2.09×10^{10}
3.5	6.66×10^{8}	2.60×10^{8}	2.46×10^{8}

4. Discussion and Remarks

- Loop Calculus provides a way to incrementally improve on BP's results.
  - It has been shown to improve quality of BP based decoding in information theory, and in a particle tracking problem for learning flows.
- This research focuses on its application to SAT:
  - The problem is more general, the search space is more complicated.
  - Progress made towards loop series summation, but results do not yet show consistent improvement.
- Main future research goal is to identify problem domains with few important loops.
  - Where significant improvement is possible.