# On the Performance of CDCL-based Message Passing Inspired Decimation using $\rho\sigma\text{PMP}^i$

Oliver Gableske[1],
Sven Müelich, Daniel Diepold

[1]Institute of Theoretical Computer Science
Ulm University
Germany

oliver@gableske.net
https://www.gableske.net

Pragmatics of SAT, 08.07.2013

# Outline

# Introduction (1)

- One major topic related to this talk is Message Passing
- Message Passing (MP) is known to the SAT community
  - but not well understood
- To make things even worse, this talk refers to an MP heuristic that has just been developed: $\rho\sigma\text{PMP}^i$
- The talk regarding $\rho\sigma\text{PMP}^i$
  - theory-heavy
  - is substantial
  - is given on Thursday
- Repeating the details here is not possible
  - We have no option but to view $\rho\sigma\text{PMP}^i$ as a "black-box" for this talk

# Introduction (2)

- MP is a class of algorithms
- H $\in$ MP can be understood as a variable and value ordering heuristic in the context of SAT
- The main goal of H is to provide *biases* for the variables in a given CNF F
- $\forall v : \beta_{\mathsf{H}}(v) \in [-1.0, 1.0]$
- The biases are used to guide the search (CDCL or SLS)

# Introduction (2)

- MP is a class of algorithms
- H $\in$ MP can be understood as a variable and value ordering heuristic in the context of SAT
- The main goal of H is to provide *biases* for the variables in a given CNF F
- $\forall v : \beta_{\mathsf{H}}(v) \in [-1.0, 1.0]$
- The biases are used to guide the search (CDCL or SLS)

- What MP heuristics do we currently have?
- What are their respective strengths and weaknesses?

# Introduction (3)

1. **Belief Propagation (BP)**
   - Is not guaranteed to converge
   - Provides biases carelessly (erratic MP behavior)
   - Works comparatively well on small random satisfiable formulas
2. **Survey Propagation (SP)**
   - Is not guaranteed to converge
   - Provides biases carefully (less erratic MP behavior)
   - Works comparatively well on large random satisfiable formulas
3. **EM Belief Propagation Global (EMBPG)**
   - Is guaranteed to converge
   - Provides biases carelessly (erratic MP behavior)
   - Works comparatively well structured (crafted) formulas
4. **EM Survey Propagation Global (EMSPG)**
   - Is guaranteed to converge
   - Provides biases carefully (less erratic MP behavior)
   - Works comparatively well structured (crafted) formulas

# Introduction (4)

The options to introduce MP into a SAT solver look pretty decent.

- Where is the problem?

Introducing MP into a SAT solver requires to choose from the given heuristics.

- A heuristic is better suited to solve specific types of formulas
- A heuristic will not be helpful on the others
- No matter how you choose, you will always choose wrong

**Introducing any of the basic MP heuristics results in a robustness problem.**

# Introduction (4)

The options to introduce MP into a SAT solver look pretty decent.

- Where is the problem?

Introducing MP into a SAT solver requires to choose from the given heuristics.

- A heuristic is better suited to solve specific types of formulas
- A heuristic will not be helpful on the others
- No matter how you choose, you will always choose wrong

**Introducing any of the basic MP heuristics results in a robustness problem.**

- We need more flexible MP heuristics!

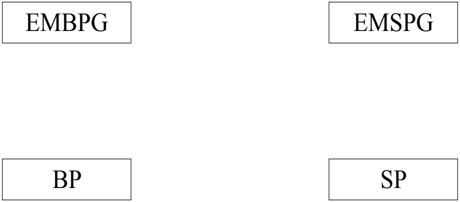
# Introduction (5)

How do we derive more flexible heuristics?

- Interpolation!
- ISI is used to interpolate two given MP heuristics into a new, more general one. Assume
    - We want to interpolate BP and SP
    - Given an interpolation parameter $\rho \in [0.0, 1.0]$
    - Resulting in $\rho\text{SP}^i = \text{ISI}(\text{BP},\text{SP}, \rho)$
- An interpolation can mimic the behavior of what it interpolates
    - Setting $\rho = 0$ results in $\beta_{\text{BP}}(v) = \beta^i_{\rho\text{SP}}(v, 0)$
    - Setting $\rho = 1$ results in $\beta_{\text{SP}}(v) = \beta^i_{\rho\text{SP}}(v, 1)$
- An interpolation can gradually adapt between them
    - Setting $\rho \in (0.0, 1.0)$ adapts the carefulness between BP and SP.

# Introduction (6)

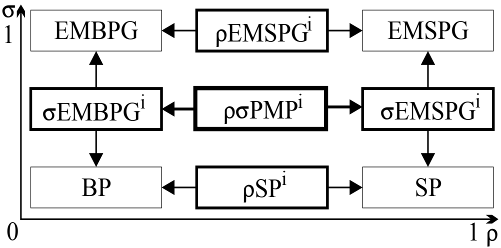
Possible interpolations: the current situation.




Level 0
BP
SP
EMBPG
EMSPG

# Introduction (7)

Possible interpolations: after applying ISI.



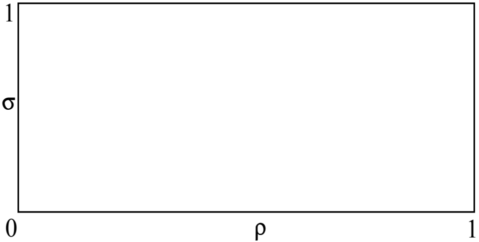

| Level 0 | Level 1 | Level 2 |
|---|---|---|
| BP | $\sigma$EMBPG$^i$ | $\rho\sigma$PMP$^i$ |
| SP | $\sigma$EMSPG$^i$ | |
| EMBPG | $\rho$SP$^i$ | |
| EMSPG | $\rho$EMSPG$^i$ | |

# $\rho\sigma\text{PMP}^i$ (1)

Why is $\rho\sigma\text{PMP}^i$ so special?

- It is the most general product-based MP heuristic.
- It can mimic the behavior of all others.
- It can provide MP behavior that cannot be achieved by any other heuristic.

Each point in the parameter plane $(\rho, \sigma) \in [0.0, 1.0]^2$ characterizes a specific MP behavior.

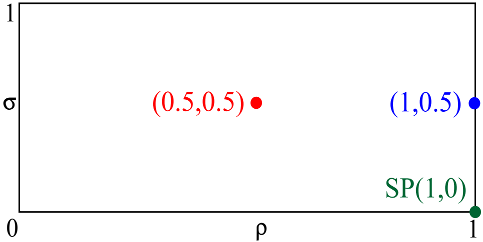


# $\rho\sigma$PMP$^i$ (2)

Why is $\rho\sigma$PMP$^i$ so special?

- It is the most general product-based MP heuristic.
- It can mimic the behavior of all others.
- It can provide MP behavior that cannot be achieved by any other heuristic.

Each point in the parameter plane $(\rho, \sigma) \in [0.0, 1.0]^2$ characterizes a specific MP behavior.




Best behavior given?

Can use:
SP, ρSP$^i$, σEMSPG$^i$, ρσPMP$^i$

Can use:
σEMSPG$^i$, ρσPMP$^i$

Can use:
ρσPMP$^i$

# $\rho\sigma\text{PMP}^i$ (3)

Using $\rho\sigma\text{PMP}^i$ in a SAT solver circumvents the robustness problem.

- Implement only $\rho\sigma\text{PMP}^i$
- The desired MP behavior can be achieved by setting $\rho, \sigma$ accordingly

How to set $\rho$ and $\sigma$ in order to solve a specific formula?

# Goals

1. Deploy the new MP heuristic $\rho\sigma\text{PMP}^i$ within a CDCL solver
2. Keep it simple
3. Determine the preferable MP behavior for different types of CNF formulas: how to set $\rho$ and $\sigma$?
4. Determine how important the flexibility of $\rho\sigma\text{PMP}^i$ is

# Message Passing Inspired Decimation

Common use of MP biases: Message Passing Inspired Decimation (MID).
Assume we have a parameter $p \in (0.0, 1.0]$.

1. Reset the empty assignment $\alpha = \{\}$
2. Compute $\forall v \notin \alpha : \beta_{\mathsf{H}}(v)$
3. Sort the variables according to the largest $|\beta_{\mathsf{H}}(v)|$
4. Assign a variable following that order, the sign of $\beta_{\mathsf{H}}(v)$ determines the assignment (extend $\alpha$ with this decision)
5. Perform unit propagation (extend $\alpha$ with all implications)
6. Determine the result
    - If a solution is found or a conflict occurred, stop
    - Otherwise
        - If $p \cdot n$ variables were assigned, go to 2
        - Otherwise go to 3.

What is CDCL-based MID?

# CDCL-based MID

1. Reset the empty assignment $\alpha = \{\}$
2. Compute $\forall v : \beta_{\mathsf{H}}(v)$
3. Initialize CDCL VSIDS activities with $|\beta_{\mathsf{H}}(v)|$
4. Initialize CDCL variable phases using the sign of $\beta_{\mathsf{H}}(v)$
5. Call CDCL in order to extend $\alpha$
    - Have it assign $p \cdot n$ *new* variables
    - Must not return until this is done
    - Clause learning is done within the CDCL
    - Learned clauses are invisible to MP
6. After CDCL returns we check the result
    - If it returns a solution or returns "unsatisfiable", stop
    - Otherwise, go back to 2.

# DimetheusMP vs. DimetheusJW

We implemented

- the CDCL-based MID in the solver `DimetheusMP` using $\rho\sigma\text{PMP}^i$
- a twin solver `DimetheusJW` that uses Jeroslow-Wang instead of $\rho\sigma\text{PMP}^i$

Both solvers perform the exact same type of CDCL search. They differ in

- the bias computation to initialize VSIDS/PS: JW vs. MP
- the parameters: $\rho, \sigma, p$ are only present in `DimetheusMP`

The crucial observation is, that `DimetheusMP` has an increased flexibility when it comes to parameter tuning.

How did we proceed to do the tuning?

# Parameter Tuning

We proceeded as follows.

1. Separate all formulas from the SC2012 into different classes
2. Run the solvers on each class (timeout 2000 seconds)
   - `DimetheusJW` (once) in order to determine the base performance
   - `Lingeling` (once) in order to get a SOTA reference performance
   - `DimetheusMP` while tuning $\rho, \sigma, p$ with EDACC/AAC

# Excerpt of the Results (1)

| Benchmark | S/U | Solver Performance | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | DimetheusJW | | DimetheusMP | | | | |
| | | % | PAR10 | % | PAR10 | $\rho$ | $\sigma$ | $p$ |
| battleship | S | 47.4 | 10627.2 | **89.5** | **2130.1** | 0.5002 | 0.0025 | 0.0021 |
| battleship | U | 55.6 | 8919.7 | 55.6 | **8890.4** | 0.4463 | 1.0000 | 0.1256 |
| em-all | S | 75.0 | 5263.7 | **100.0** | **75.4** | 0.8606 | 0.1295 | 0.8903 |
| em-compact | S | 0.0 | 20000.0 | **37.5** | **12728.5** | 0.9229 | 0.7946 | 0.8281 |
| em-explicit | S | 75.0 | 5473.3 | **100.0** | **157.1** | 0.2932 | 0.2698 | 0.0853 |
| em-fbcolors | S | 12.5 | 17723.3 | **37.5** | **12662.9** | 0.0000 | 0.1731 | 0.7672 |
| grid-pebbling | S | 100.0 | 16.5 | 100.0 | **8.0** | 0.9931 | 0.3890 | 0.6449 |
| grid-pebbling | U | 88.9 | 2226.9 | **100.0** | **4.7** | 0.5884 | 0.0035 | 0.2213 |
| sgen1 | S | 16.7 | 16677.7 | **27.8** | **14460.9** | 0.0937 | 0.6563 | 0.4688 |
| k3-r4.200-n40000 | S | 0.0 | 20000.0 | **100.0** | **22.7** | 0.9929 | 0.0004 | 0.0447 |
| k3-r4.237-n18800 | S | 0.0 | 20000.0 | **75.0** | **5026.8** | 0.9961 | 0.0000 | 0.0042 |
| k4-r9.000-n10000 | S | 0.0 | 20000.0 | **100.0** | **10.0** | 0.8592 | 0.0000 | 0.1533 |
| k4-r9.526-n4800 | S | 0.0 | 20000.0 | **100.0** | **5.2** | 0.9530 | 0.0000 | 0.0337 |

# Excerpt of the Results (2)

Most important results are

- The flexibility of $\rho\sigma\text{PMP}^i$ is important (almost always, we have $\rho, \sigma \notin \{0.0, 1.0\}$)
- Using MP can be very helpful to solve crafted formulas (satisfiable *and* unsatisfiable ones)
- Enforcing convergence ($\sigma > 0.0$) is *not helpful on random formulas*

# Issues with the Empirical Study

Reviewer 1: "As submitted, this paper does not show anything."

- Several classes contain only a small set of formulas
  - Robustness of the reported settings for $\rho, \sigma, p$ is questionable
  - We need more formulas, or even better, generators!
  - We cannot use, what isn't there
- Missing test-classes
- No results on application formulas

# Conclusions

The most important conclusions are as follows.

- The empirical study gives a hint that MP can be very helpful to solve random and crafted formulas.
- The flexibility of $\rho\sigma\text{PMP}^i$ is crucial to achieve this performance.

# Thanks

**Thank you for your attention!**

You can send comments and questions to
oliver@gableske.net

# Bibliography


The paper regarding the MP theory:

O. Gableske
On the Interpolation between Product-Based
Message Passing Heuristics for SAT

The paper regarding the practical aspects:

O. Gableske, S. Müelich, D. Diepold
On the Performance of CDCL-based Message
Passing Inspired Decimation using $\rho\sigma\text{PMP}^i$


You can download the papers and the
`Dimetheus` sources from https://www.gableske.net