Using Expectation Maximization to Find Likely Assignments for Solving Constraint Satisfaction Problems

Eric Hsu, Matthew Kitching, Fahiem Bacchus, and Sheila McIlraith

University of Toronto

{eihsu, kitching, fbacchus, sheila}@cs.toronto.edu
Constraint Satisfaction Problem (CSP)

- **Inference**: Try to combine overlapping constraints (perhaps partially).

- **Search**: Try different combinations of values for the variables.

Using Expectation Maximization to Find Likely Assignments for Solving CSP's
Motivation: Discovering Structure

- Certain variables are much more important than the rest.
  - Combinatorial cores can be small.
  - Remainder is extraneous.
Motivation: Discovering Structure

- Certain variables are much more important than the rest.
  - Combinatorial cores can be small.
  - Remainder is extraneous.
- Can’t just rely on graph structure in the general case.
Overview

1. Hidden Structure in CSP’s
   - Running Example: Quasigroup with Holes (QWH)
2. Existing Approaches to Finding Cores
   - Heuristics, Random Restarts
   - Computing Surveys Using Belief Propagation (BP)
3. Expectation Maximization BP (EMBP)
   - Guaranteed Convergence via EM
   - Representing the Dual Problem
   - Primal/Dual Semantics: Mixing Search and Inference
4. Experimental Results
5. Conclusions and Future Work
Part 1

Hidden Structure in CSP’s
Using Expectation Maximization to Find Likely Assignments for Solving CSP’s

Quasigroup with Holes (QWH) Problem

- Latin Square:
  - \(d\) rows by \(d\) columns
  - Cell variables \(\in\{1, \ldots, d\}\)
  - No repeats in any row or column: “alldiff” constraint

<table>
<thead>
<tr>
<th>3</th>
<th>4</th>
<th>5</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

\(d = 5\)
Quasigroup with Holes (QWH) Problem

- Latin Square:
  - \(d\) rows by \(d\) columns
  - Cell variables \(\in \{1, \ldots, d\}\)
  - No repeats in any row or column: "alldiff" constraint

- QWH Problem:
  - Remove a percentage of the holes from a valid Latin Square.
(Kautz et al., ’01)
Why QWH is Interesting

- Combinatorial abstraction for real problems
- Small, tightly constrained cores
  - Backdoors/Backbones (Williams et al., '03)
- Heavy-Tailed Behavior
  - High sensitivity to variable ordering
    (Gomes et al., '00; Hulubei et al., '04)

Using Expectation Maximization to Find Likely Assignments for Solving CSP’s
Part 2

Existing Approaches to Hidden Structure
Finding and Exploiting the Highly Constrained Variables

Finding Sets of Highly Constrained Variables

- Guessing
  e.g. (Gomes et al., '98)

- Heuristics / Learning
  e.g. (Boussemart et al., '04)

- Surveys: Belief Propagation (BP) and Survey Propagation (SP)
  e.g. (Kask et al., '04; Braunstein et al., '05)
Finding and Exploiting the Highly Constrained Variables

Exploiting Sets of Highly Constrained Variables within Backtracking Search

Finding Sets of Highly Constrained Variables

- Guessing
  e.g. (Gomes et al., '98)

- Heuristics / Learning
  e.g. (Boussemart et al., '04)

- Surveys: Belief Propagation (BP) and Survey Propagation (SP)
  e.g. (Kask et al., '04; Braunstein et al., '05)

Random Restarts

Variable Ordering toward Solutions

Multiple Options

Using Expectation Maximization to Find Likely Assignments for Solving CSP’s
Surveys: Overview

CSP PROBLEM

List of Solutions

Parameter Estimator (like BP)

“Survey”: Probability for each variable to be fixed a certain way when we choose a solution at random.

\[ P(\Theta_v|SOL) \]
Surveys under Q WH

- **Bias** $\Theta_{a,b}(val)$:
  Probability that the cell at row $a$, column $b$ holds value $v$ in a satisfying configuration.

- **Bias Distribution**, $\Theta_{a,b}()$:
  Probability distribution over a cell’s possible values, comprised of its various biases.

- **Survey**:
  Set of bias distributions, one for each unassigned cell in a problem.
Using Surveys with Unit Decimation

- Variable/Value-Ordering heuristic in a regular backtracking CSP Solver.
  - Set the most highly constrained variables first.
  - Set them to their most likely values.
- Turn off after setting the most constrained variables.
Limitations of Propagation Methods

- **BP might not converge.**
  - Only proper for problems whose constraint graphs are trees.
  - Random problems are full of “loose” loops, structured problems are worse.
    - Parity, Local Interactions, Covariance Dependencies
    - Initialization of Biases

- **Surveys are only approximations.**
  - Local maximum in likelihood

- **Loopy semantics are not well-understood.**
  - Characterizing non-convergence, likely settings
Part 3

EMBP:
Expectation Maximization
Belief Propagation
Problem: Find $\Theta$ to maximize $P(Y, Z | \Theta)$. 
- $Y$: observed data
- $\Theta$: model parameters
- $Z$: unobserved data
- Can’t marginalize on $Z$, or else it would be easy.
The EM Algorithm

- Solution: Artificial estimate $Q(Z)$ to match $P(Z | Y, \Theta)$
  - E-Step: Set $Q(Z)$ to estimate $P(Z | Y, \Theta)$
  - M-Step: Set $\Theta$ to maximize $P(Y, Z | \Theta)$

- Important: Use logs of all probabilities, to exploit Jensen’s Inequality—EM always converges!
Using EM to Compute CSP Surveys

- The basic intuition is to deceive EM.
  - Observations \( Y \):
    “We saw that each of the constraints was satisfied...”
    i.e., “We saw a solution to the CSP...”
  - Hidden Variables \( Z \):
    “...but we didn’t get to see how exactly the constraints were satisfied by the variables in their domains.”
  - Parameters \( \Theta \):
    “Please figure out how the variables were likely set (by hypothesizing likely Z’s representing how the constraints were satisfied.)”

- Parameters are just the variables’ biases.
Final Result: EMBP Update Rule

- Ultimately we derive a simple update rule for setting $\Theta$ given the current $Q()$:

$$\theta_{\text{var}}(val) = \frac{1}{N} \sum_{Z: \text{var}=val} Q(Z)$$

- This is the M-Step; we can substitute in our choice of E-Step representations for $Q()$.

- If you did the whole thing in BP and rearranged, you would get almost the same formula!
  - Change the sum into a product.
Using EM to Solve QWH

\[ \Theta_{2,4}(1) = 0.081 \]
\[ \Theta_{2,4}(2) = 0.074 \]
\[ \Theta_{2,4}(3) = 0.653 \]
\[ \Theta_{2,4}(4) = 0.078 \]
\[ \Theta_{2,4}(5) = 0.114 \]

Probability that cell 2,4 contains 5

Probability of an extension to the 2nd row constraint where 5 is in 4th position

Probability of extension to the 4th column constraint where 5 is in 2nd position
Representing the Dual (Z) for QWH

- **EMBP-e (exact)**
  - Expensive: represent each of the $d!$ possible permutations in the extension of a row/column constraint.
  - Dynamic programming allows $2^d$ complexity.
  - Represents “global consistency.”

- **EMBP-a (approximate)**
  - Inexact: a cell can hold a particular value if nobody else in its row/column does. (Doesn’t represent relationship between other cells within the same constraint.)
  - Linear time complexity on $d$.
  - Represents “arc consistency”.

Using Expectation Maximization to Find Likely Assignments for Solving CSP’s
Primal/ Dual Semantics

- E-Step (Dual)
  - Use the biases to update Q(Z).
  - Inference—variable settings determine the possible extensional values of the constraints.

- M-Step (Primal)
  - Use Q(Z) to update the biases.
  - Search—constraint extensions filter the possible values of the variables.

- “Optimal” alternation between steps, to guaranteed convergence.
Part 4

Experiments
### Basic Performance: Total Runtime, Excluding Timeouts

<table>
<thead>
<tr>
<th>QWH 17x17</th>
<th>EMBP-e</th>
<th>EMBP-a</th>
<th>BP-e</th>
<th>BP-a</th>
<th>dom/wdeg</th>
<th>dom/deg</th>
</tr>
</thead>
<tbody>
<tr>
<td>QWH(45%)</td>
<td>3663</td>
<td>172</td>
<td>5063</td>
<td>8383</td>
<td>208</td>
<td>4429</td>
</tr>
<tr>
<td>QWH(50%)</td>
<td>15767</td>
<td>2065</td>
<td>13080</td>
<td>4471</td>
<td>3710</td>
<td>27586</td>
</tr>
<tr>
<td>QWH(55%)</td>
<td>12780</td>
<td>4338</td>
<td>15839</td>
<td>3850</td>
<td>4813</td>
<td>31610</td>
</tr>
<tr>
<td>QWH(60%)</td>
<td>4885</td>
<td>4708</td>
<td>28034</td>
<td>1951</td>
<td>7376</td>
<td>30501</td>
</tr>
<tr>
<td>QWH(65%)</td>
<td>5191</td>
<td>1909</td>
<td>29616</td>
<td>1927</td>
<td>4266</td>
<td>22026</td>
</tr>
<tr>
<td>QWH(Total)</td>
<td>42287</td>
<td>13217</td>
<td>91129</td>
<td>20585</td>
<td>20375</td>
<td>116154</td>
</tr>
</tbody>
</table>

100 successful runs on random 17x17 problems, in seconds
### Basic Performance: Number of Timeouts

<table>
<thead>
<tr>
<th>( QWH \ 17 \times 17 )</th>
<th>EMBP-e</th>
<th>EMBP-a</th>
<th>BP-e</th>
<th>BP-a</th>
<th>dom/wdeg</th>
<th>dom/deg</th>
</tr>
</thead>
<tbody>
<tr>
<td>QWH(45%)</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>QWH(50%)</td>
<td>5</td>
<td>2</td>
<td>17</td>
<td>6</td>
<td>6</td>
<td>48</td>
</tr>
<tr>
<td>QWH(55%)</td>
<td>1</td>
<td>7</td>
<td>13</td>
<td>5</td>
<td>7</td>
<td>58</td>
</tr>
<tr>
<td>QWH(60%)</td>
<td>6</td>
<td>8</td>
<td>18</td>
<td>3</td>
<td>13</td>
<td>57</td>
</tr>
<tr>
<td>QWH(65%)</td>
<td>8</td>
<td>3</td>
<td>53</td>
<td>2</td>
<td>7</td>
<td>42</td>
</tr>
<tr>
<td>QWH(Total)</td>
<td>20</td>
<td>20</td>
<td>102</td>
<td>22</td>
<td>34</td>
<td>207</td>
</tr>
</tbody>
</table>

100 runs on random 17x17 problems, timeout = 500 seconds
Survey Techniques and Problem Topography

- Techniques can reach different local maxima in likelihood from the same random initialization.
  - BP/EMBP both follow gradient, but non-convergent techniques like BP take bigger steps that can overshoot local maxima.

Using Expectation Maximization to Find Likely Assignments for Solving CSP’s
Comparing Dual Representations:

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>QWH(50%)</th>
<th>QWH(55%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Nodes</td>
</tr>
<tr>
<td>EMBP-e</td>
<td>12508</td>
<td>2.6*10^6</td>
</tr>
<tr>
<td>EMBP-a</td>
<td>1047</td>
<td>7.0*10^6</td>
</tr>
</tbody>
</table>

100 runs on random 17x17 problems, timeout = 500 seconds
## Number of Backtracks: Gambling on Satisfiable Subtrees

<table>
<thead>
<tr>
<th>Version</th>
<th>EMBP-a</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Surveys</td>
<td>Backtrack</td>
</tr>
<tr>
<td>QWH(45%)</td>
<td>497</td>
<td>33</td>
</tr>
<tr>
<td>QWH(50%)</td>
<td>502</td>
<td>21</td>
</tr>
<tr>
<td>QWH(55%)</td>
<td>526</td>
<td>61</td>
</tr>
<tr>
<td>QWH(60%)</td>
<td>557</td>
<td>97</td>
</tr>
<tr>
<td>QWH(65%)</td>
<td>500</td>
<td>15</td>
</tr>
<tr>
<td>QWH(Total)</td>
<td>2582</td>
<td>227</td>
</tr>
</tbody>
</table>

Total from 100 successful runs on random 17x17 problems
Part 5

Conclusions and Future Work
How Can I Try EMBP on My Problem?

- **General Approach:**
  - Bias search over values that score high over (arithmetic) average of supporting extensions.
  - Q(Z): For each type of constraint, you need to choose a level of approximation for inference and implement appropriate data structures.

- **Experiments Suggest:**
  - Best suited for structured problems, with non-random interactions between variables of differing importance.
  - Convergence requires relatively few iterations.
    - Time complexity depends mainly on choice of $Q(Z)$.
    - Dimensionality of optimization = number of values.
Basic Advantages of EMBP

- Semantics, Optimality in Local Search
- Convergence
- Potentially “Better” Optima
- Isolation of Dual Approximation $Q(Z)$
Future Work

- More experimentation: comparing various survey techniques/parameter settings on different problem types
- Explicit comparison of the cores identified by EMBP with known backdoors
- Integrating learning techniques
- Connections to model counting

Thanks!

Eric and Matthew

www.cs.toronto.edu/~{eihsu,kitching}
EXTRA SLIDES
Using Expectation Maximization to Find Likely Assignments for Solving CSP's
Factor Graph Representation for QWH

- Variable node for each cell in the Latin Square.
- Constraint node for each row \textit{alldiff}.
Factor Graph Representation for QWH

- Variable node for each cell in the Latin Square.
- Constraint node for each row `alldiff`.
- Constraint node for each column `alldiff`.
Message Passing in the Factor Graph

Variable to Constraint:
Current bias distribution.
(i.e. $\Theta_{2,4}()$: the probability that variable 2,4 holds each possible value)
Constraints re-weight possible extensions.
Message Passing in the Factor Graph

**Constraint to Variable:**
Level of support for each possible value.
(i.e. the probability that Row 2 realizes each possible permutation, or some approximation)
Variables update biases.