Adding Unsafe Constraints to Improve Satisfiability Performance

John Franco
Computer Science, University of Cincinnati
**Constraints are Clauses**

A **variable** looks like this: $v_9$, takes a value from \{0, 1\}

A **positive literal**: $v_9$, a **negative literal**: $\neg v_9$

A **clause** looks like this: \((\neg v_1 \lor \neg v_2 \lor v_5 \lor v_9)\)

An **instance** of SAT looks like this:

\[
(v_1 \lor \neg v_2 \lor v_7) \land (\neg v_2 \lor v_6) \land (\neg v_2 \lor \neg v_4 \lor \neg v_5) \land (v_{10}) \ldots
\]

Clause **width**: \# literals; **k-SAT**: fixed width $k$

An important **splitting** operation in solvers:

\[
(\neg v_1 \lor \neg v_2 \lor v_i) \land (\neg v_i \lor v_3) \land (\neg v_2 \lor v_4)
\]

\[
(v_3) \land (\neg v_2 \lor v_4) \quad (\neg v_1 \lor \neg v_2) \land (\neg v_2 \lor v_4)
\]
A Search Space

Learned clauses
What Makes a Problem Hard?

Useful clauses are not learned early enough:
What Makes a Problem Hard?

Is any particular structure bad?
What Makes a Problem Hard?

This can be flattened

\[ f(a, b, c) \quad \iff \quad z \iff f(a, b, c) \]
Some FV Problems Have This Structure

Variables: at time $i$,

\[ v^i_1 = \text{value of bit 1}, \]
\[ v^i_2 = \text{value of bit 2} \]

Does the 2-bit counter above reach state 11 in exactly 3 time steps?
The Propositional Formula

Force the property to hold:

\[ \neg(v_0 \land v_0) \land \neg(v_1 \land v_1) \land \neg(v_1 \land v_2) \land (v_1 \land v_2) \]

Express the starting state:

\[ \neg v_0 \land \neg v_2 \]

Force legal transitions (repetitions of the transition relation):

\[
(v_2 \equiv \neg v_0) \land (v_1 \equiv v_0 \oplus v_0) \land (v_2 \equiv \neg v_2) \land
(v_1 \equiv v_1 \oplus v_2) \land (v_3 \equiv \neg v_2) \land (v_3 \equiv v_1 \oplus v_2)
\]

Satisfied only by:

\[ v_0 = 0, v_0 = 0, v_1 = 1, v_1 = 0, v_2 = 0, v_1 = 0, v_2 = 1, v_3 = 1, v_3 = 1 \]
The Propositional Formula

Three repetitions of a function

\[ f_i = \neg v_{i+1} \land v_{i+3} \land (v_i \equiv v_{i+2}) \]
How Can We Make the Problem Easier?

Install the inferred constraints early

Install safe, uninferred constraints that are obtained from an analysis of the problem
  - for example, take advantage of problem symmetry

Install unsafe, uninferred constraints that are obtained from an analysis of solutions to smaller problems in the family
  - run the search to some depth past the hump
  - retract the unsafe constraints and search deeper
Example - Van der Waerden Numbers

Let $S_n = \{1, \ldots, n\}$.

Let proposition $P_{n,k}(l)$ be true if and only if all partitions of $S_n$ into $k$ classes contain at least one arithmetic progression of length $l$ in at least one class.

Then $W(k, l)$ is the minimum $n$ for which $P_{n,k}(l)$ is true.

Example, all do: $k = 2, l = 3, n = 9$

\[
\{\{1, 2, 3, 4, 5\}\{6, 7, 8, 9\}\}, \{\{1, 3, 4, 7\}\{2, 5, 6, 8, 9\}\}
\]

Example, one does not: $k = 2, l = 3, n = 8$

\[
\{\{1, 2, 5, 6\}\{3, 4, 7, 8\}\}
\]
Example - Van der Waerden Numbers

There is no known closed form expression for $W(k, l)$

Table shows all known Van der Waerden numbers. $W(2, 6)$ determined in 2007, all others before 1979.

<table>
<thead>
<tr>
<th>$k \setminus l$</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>9</td>
<td>35</td>
<td>178</td>
<td>1132</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>76</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Previous Bounds

<table>
<thead>
<tr>
<th>k \ l</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>9</td>
<td>35</td>
<td>178</td>
<td>&gt;341</td>
<td>&gt;614</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>&gt;193</td>
<td>&gt;676</td>
<td>&gt;2236</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>76</td>
<td>&gt;416</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>&gt;125</td>
<td>&gt;880</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>k \ l</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>9</td>
<td>35</td>
<td>178</td>
<td>&gt;695</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>&gt;291</td>
<td>&gt;1209</td>
<td>&gt;8885</td>
</tr>
<tr>
<td>4</td>
<td>76</td>
<td>&gt;1047</td>
<td>&gt;10436</td>
<td>&gt;90306</td>
</tr>
<tr>
<td>5</td>
<td>&gt;125</td>
<td>&gt;2253</td>
<td>&gt;24044</td>
<td>&gt;177955</td>
</tr>
</tbody>
</table>
## Formula for $W(2,6)$

<table>
<thead>
<tr>
<th>Variables</th>
<th>Meaning</th>
</tr>
</thead>
</table>
| $v_i$     | $v_i \equiv 1$ if $i + n/2 \in C_1$
|           | $v_i \equiv 0$ if $i + n/2 \in C_2$ |

<table>
<thead>
<tr>
<th>Clauses</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\neg v_i \lor \neg v_{i+1} \lor \ldots \lor \neg v_{i+5})$</td>
<td>no arithmetic progression of length 6 in $C_1$</td>
</tr>
</tbody>
</table>
| $(\neg v_i \lor \neg v_{i+2} \lor \ldots \lor \neg v_{i+10})$ | $-n/2 < i \leq n/2 - 5$
| ... | $t = \lfloor (n/2 - i)/5 \rfloor$
| $(\neg v_i \lor \neg v_{i+t} \lor \ldots \lor \neg v_{i+5t})$ | no arithmetic progression of length 6 in $C_2$ |
| $(v_i \lor v_{i+1} \lor \ldots \lor v_{i+5})$ | $-n/2 < i \leq n/2 - 5$
| $(v_i \lor v_{i+2} \lor \ldots \lor v_{i+10})$ | $t = \lfloor (n/2 - i)/5 \rfloor$
| ... | ...
Analyze Solutions to Smaller Instances

Analysis of solutions to $W(2, 4)$ and $W(2, 5)$

Limited length patterns of reverse symmetry
Unsafe Constraints 1

For $W(2, l)$ formula there exists at least one solution with a reflected pattern of length $W(2, l)/(2(l - 1))$ with the middle positioned somewhere between $W(2, l)/(l - 1)$ and $W(2, l) \times (l - 2)/(l - 1)$.

Design a filter for variable assignment patterns that are not reverse symmetric.

<table>
<thead>
<tr>
<th>Clauses</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$((v_i \lor v_{i+1}) \quad 0 \leq i &lt; s/2$</td>
<td>force $v_i \equiv \bar{v}_{i+1}$.</td>
</tr>
<tr>
<td>$(-v_i \lor \neg v_{i+1})$</td>
<td></td>
</tr>
</tbody>
</table>
Unsafe Constraints 2

Some small assignment patterns do not occur in solutions. Construct constraints to filter them.

This action is opposite to that of forcing patterns to occur which is the objective of unsafe constraints 1.

<table>
<thead>
<tr>
<th>Clauses</th>
<th>Filters</th>
</tr>
</thead>
<tbody>
<tr>
<td>((v_i, \neg v_i+t, v_i+2t, \neg v_i+3t, v_i+4t, \neg v_i+5t))</td>
<td>010101</td>
</tr>
<tr>
<td>((\neg v_i, v_i+t, \neg v_i+2t, v_i+3t, \neg v_i+4t, v_i+5t))</td>
<td>101010</td>
</tr>
<tr>
<td>((v_i, v_i+t, \neg v_i+2t, \neg v_i+3t, v_i+4t, \neg v_i+5t, \neg v_i+6t, v_i+7t))</td>
<td>00110110</td>
</tr>
<tr>
<td>((\neg v_i, \neg v_i+t, v_i+2t, v_i+3t, \neg v_i+4t, v_i+5t, v_i+6t, \neg v_i+7t))</td>
<td>11000101</td>
</tr>
<tr>
<td>((v_i, \neg v_i+t, \neg v_i+2t, v_i+3t, \neg v_i+4t, \neg v_i+5t, v_i+6t, \neg v_i+7t))</td>
<td>01101100</td>
</tr>
<tr>
<td>((\neg v_i, v_i+t, v_i+2t, \neg v_i+3t, v_i+4t, v_i+5t, v_i+6t, \neg v_i+7t))</td>
<td>10010011</td>
</tr>
<tr>
<td>((v_i, v_i+t, \neg v_i+2t, \neg v_i+3t, \neg v_i+4t, v_i+5t, v_i+6t, \neg v_i+7t))</td>
<td>00111001</td>
</tr>
<tr>
<td>((\neg v_i, \neg v_i+t, v_i+2t, v_i+3t, v_i+4t, \neg v_i+5t, \neg v_i+6t, v_i+7t))</td>
<td>11000110</td>
</tr>
<tr>
<td>((v_i, v_i+t, \neg v_i+2t, v_i+3t, v_i+4t, \neg v_i+5t, v_i+6t, \neg v_i+7t))</td>
<td>10011100</td>
</tr>
<tr>
<td>((\neg v_i, \neg v_i+t, \neg v_i+2t, v_i+3t, v_i+4t, \neg v_i+5t, v_i+6t, v_i+7t))</td>
<td>01100011</td>
</tr>
</tbody>
</table>
Unsafe Constraints 3

Analytic solutions to $W(2, 6)$ formulas have been found for various values of $n$ including 565 and 695.

Take solution for $n = 565$ and re-index the assigned variables

$v_{-282}, \ldots, v_0, v_1, \ldots v_{282}$

to

$v_{-564}, v_{-562}, \ldots, v_0, v_2, \ldots, v_{562}, v_{564}$

and add free variables

$v_{-565}, v_{-563}, \ldots, v_1, \ldots, v_{563}, v_{565}$.

This does not introduce any arithmetic progression among the even indexed variables.

Constraints are the assignment to the even indexed variables.
Results

Any of the constraint sets works

- improves performance of off-the-shelf solvers by orders of magnitude
- all constraint sets give the same result:
  a bound of 1132 on $W(2, 6)$
- These ideas were later used to get the exact number $W(2, 6) = 1132$
How to Apply This to FV?

If a solution is found - use it

If no solution is found - need to build confidence
  Try several different unsafe constraint sets
  Gradually remove some of the constraints
  Retract unsafe constraints earlier
  Parallelization helps realize search breadth

If applied to an optimization problem - approximation?
What Are the Problems?

This is too ad-hoc at the moment

A good confidence measure is needed

Need to know when to retract unsafe constraints