Adding Unsafe Constraints to Improve Satisfiability Performance

John Franco
Computer Science, University of Cincinnati
based on work of Michal Kouril

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: $(\neg v_1 \vee \neg v_2 \vee v_5 \vee v_9)$

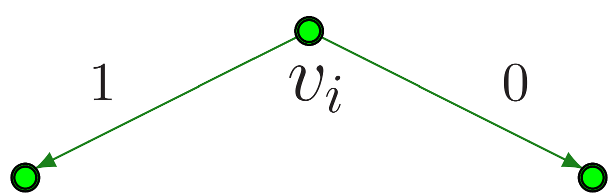
An **instance** of SAT looks like this:

$$(v_1 \vee \neg v_2 \vee v_7) \wedge (\neg v_2 \vee v_6) \wedge (\neg v_2 \vee \neg v_4 \vee \neg v_5) \wedge (v_{10}) \ldots$$

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

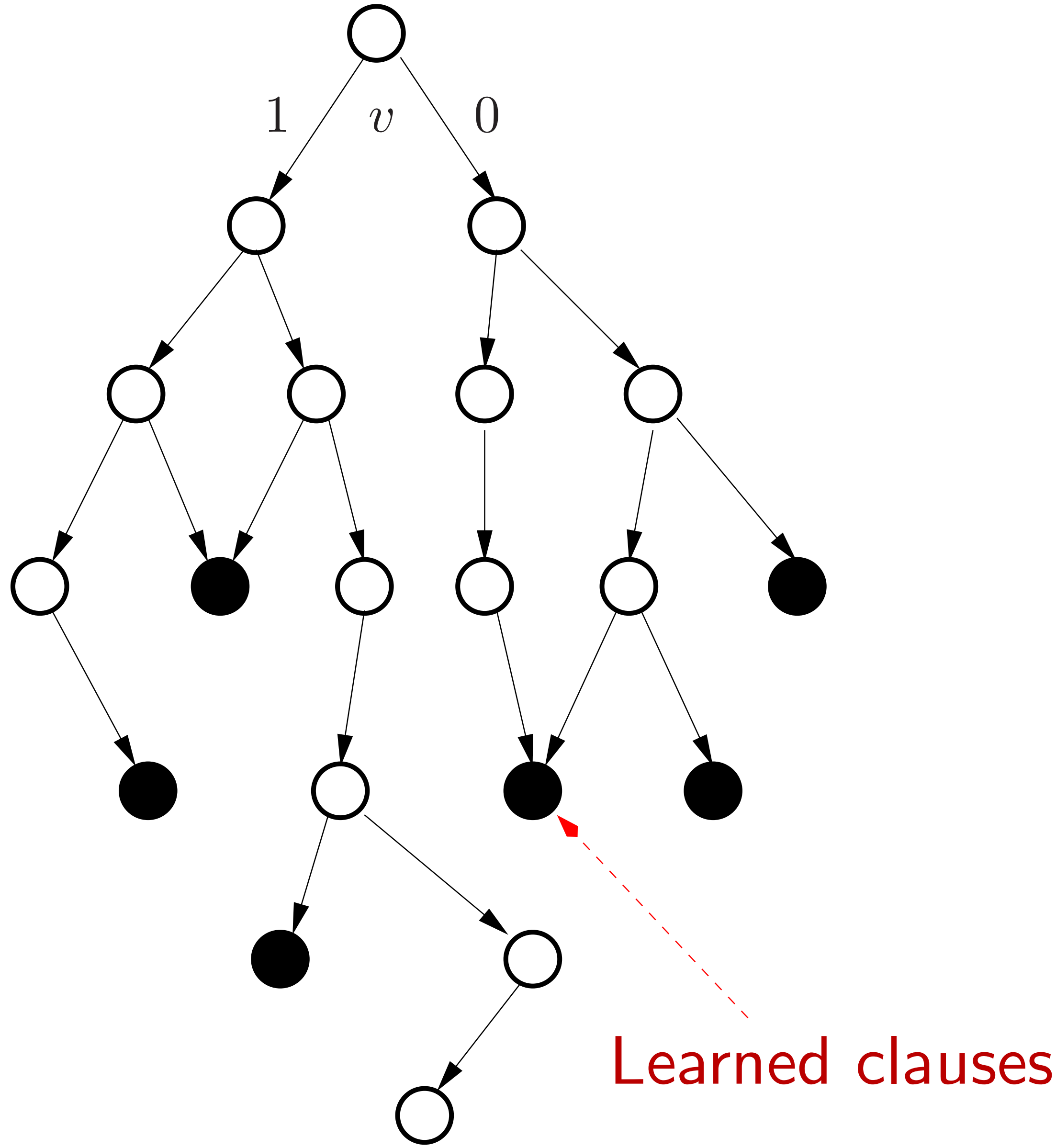
An important **splitting** operation in solvers:

$$(\neg v_1 \vee \neg v_2 \vee v_i) \wedge (\neg v_i \vee v_3) \wedge (\neg v_2 \vee v_4)$$



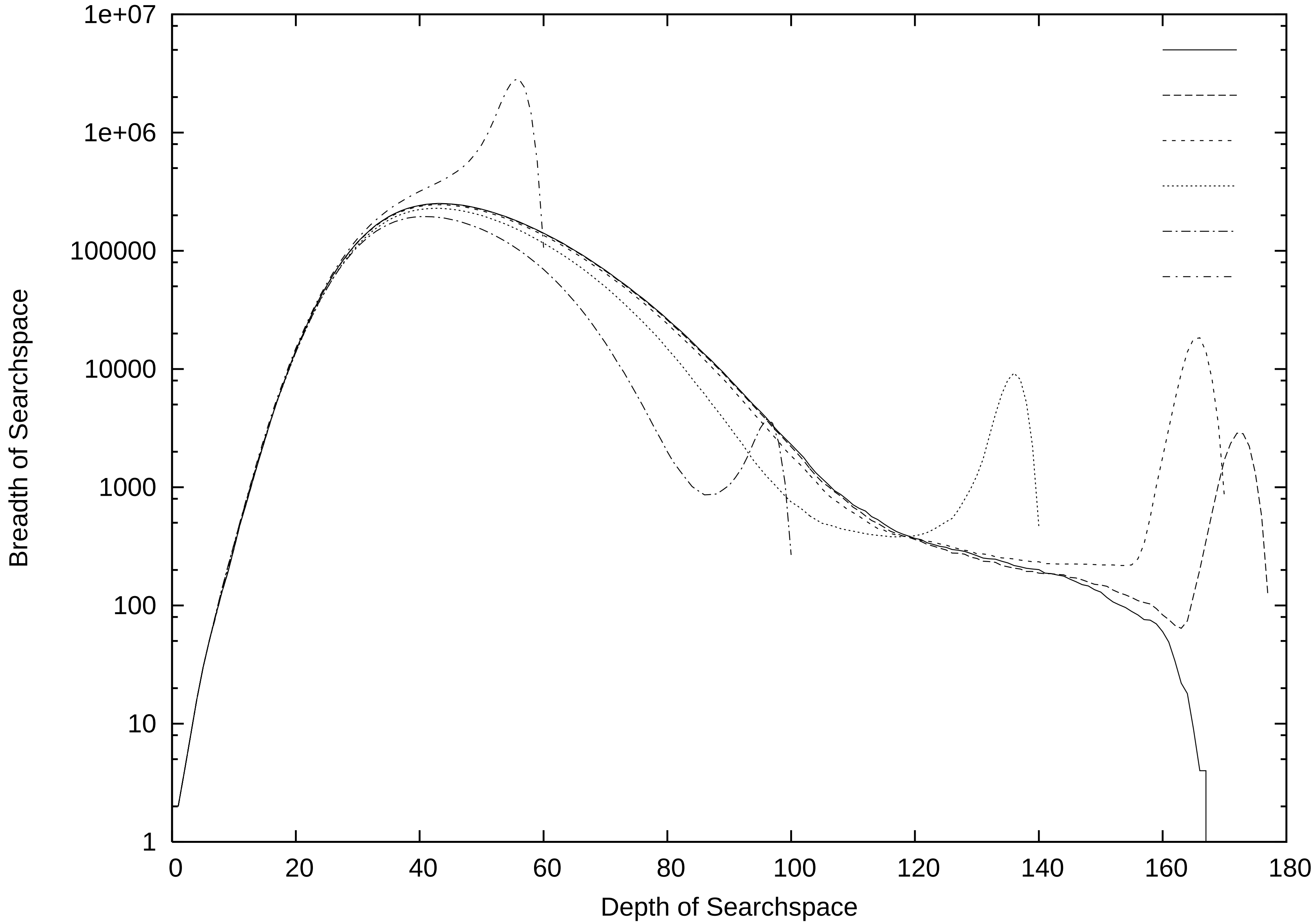
$$(v_3) \wedge (\neg v_2 \vee v_4) \qquad\qquad (\neg v_1 \vee \neg v_2) \wedge (\neg v_2 \vee v_4)$$

A Search Space



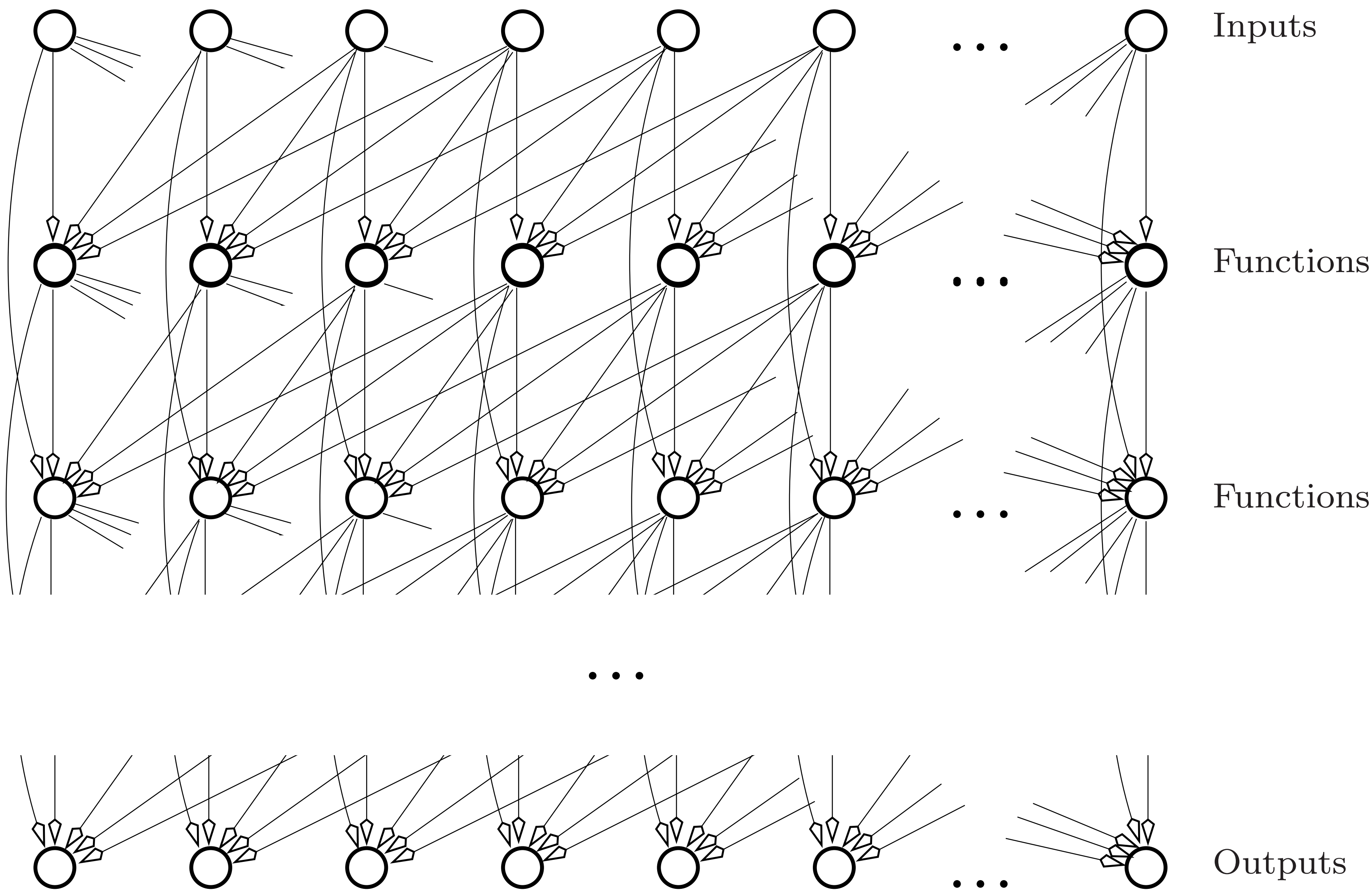
What Makes a Problem Hard?

Useful clauses are not learned early enough:



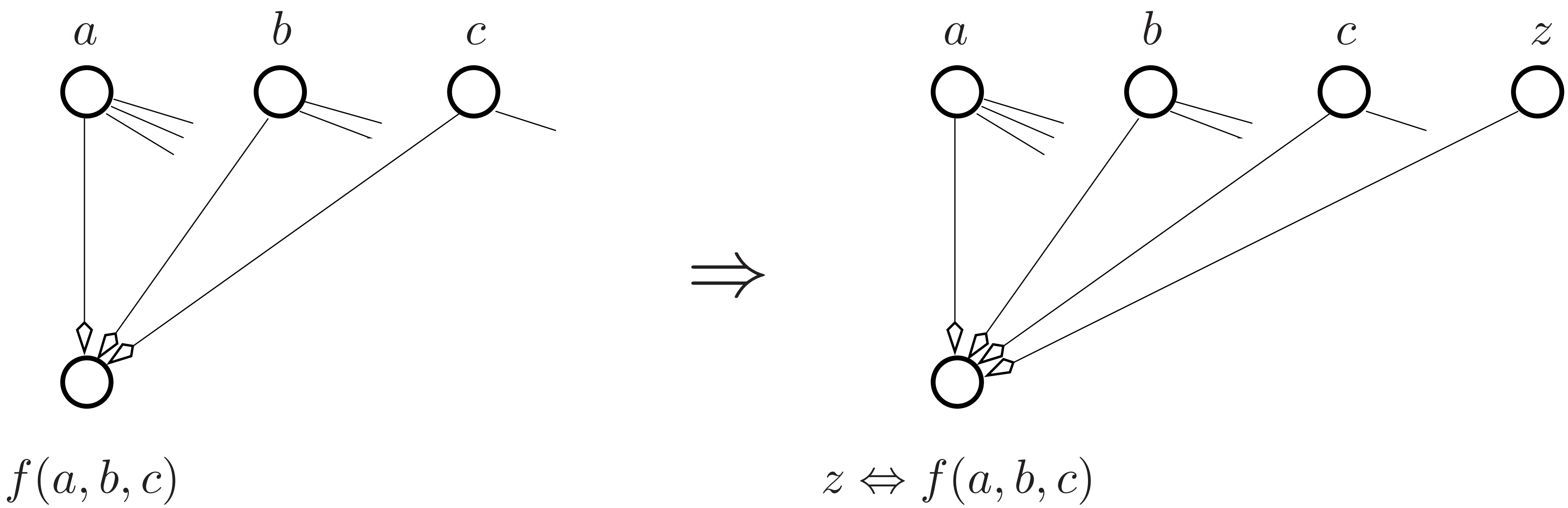
What Makes a Problem Hard?

Is any particular structure bad?

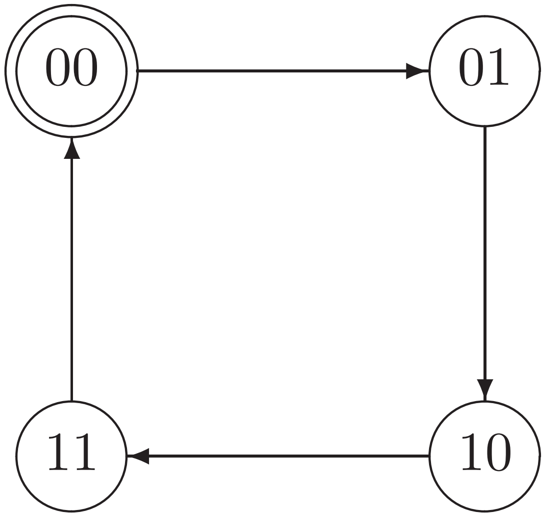


What Makes a Problem Hard?

This can be flattened



Some FV Problems Have This Structure



Variables: at time i,

v_1^i = value of bit 1,

v_2^i = 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:

$$\neg(v_1^0 \wedge v_2^0) \wedge \neg(v_1^1 \wedge v_2^1) \wedge \neg(v_1^2 \wedge v_2^2) \wedge (v_1^3 \wedge v_2^3)$$

Express the starting state:

$$\neg v_1^0 \wedge \neg v_2^0$$

Force legal transitions (repetitions of the transition relation):

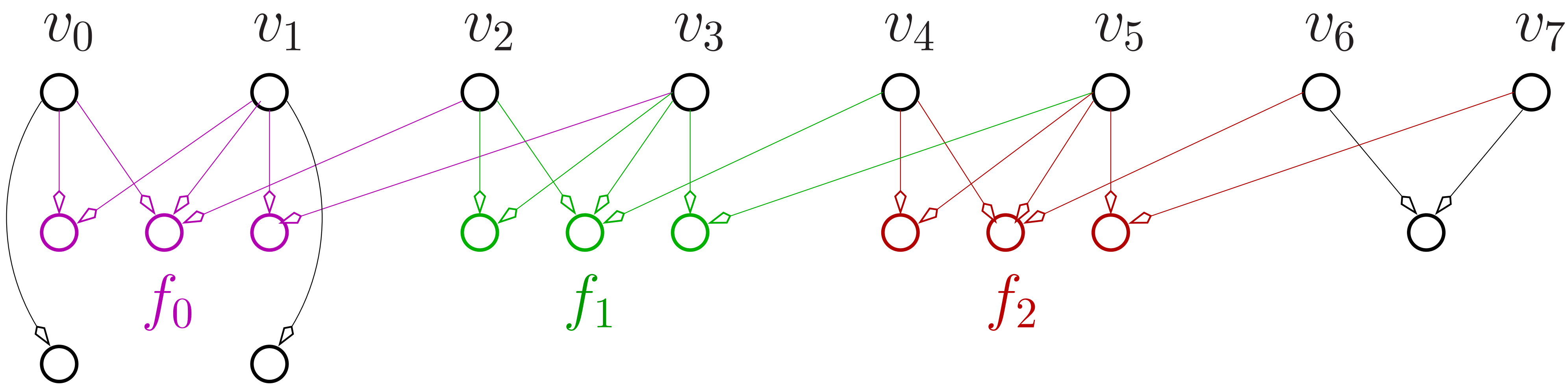
$$(v_2^1 \equiv \neg v_2^0) \;\wedge\; (v_1^1 \equiv v_1^0 \oplus v_2^0) \;\wedge\; (v_2^2 \equiv \neg v_2^1) \wedge$$
$$(v_1^2 \equiv v_1^1 \oplus v_2^1) \;\wedge\; (v_2^3 \equiv \neg v_2^2) \;\wedge\; (v_1^3 \equiv v_1^2 \oplus v_2^2)$$

Satisfied *only* by:

$$v_1^0 = 0, v_2^0 = 0, v_1^1 = 1, v_2^1 = 0, v_1^2 = 0, v_2^2 = 1, v_1^3 = 1, v_2^3 = 1$$

The Propositional Formula

Three repetitions of a function



$$f_i = \neg v_{i+1} \wedge v_{i+3} \wedge (v_i \equiv v_{i+2})$$

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

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: $k = 2$, $l = 3$, $n = 9$

$\{\{1, 2, 3, 4, 5\}\{\mathbf{6}, \mathbf{7}, \mathbf{8}, 9\}\}$,$\{\{1, 3, 4, 7\}\{\mathbf{2}, \mathbf{5}, 6, \mathbf{8}, 9\}\}$

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.

$k \setminus l$	3	4	5	6
2	9	35	178	1132
3	27			
4	76			

Previous Bounds

$k \setminus l$	3	4	5	6	7
2	9	35	178	>341	>614
3	27	>193	>676	>2236	
4	76	>416			
5	>125	>880			

Formulas $\uparrow$ Analysis $\downarrow$

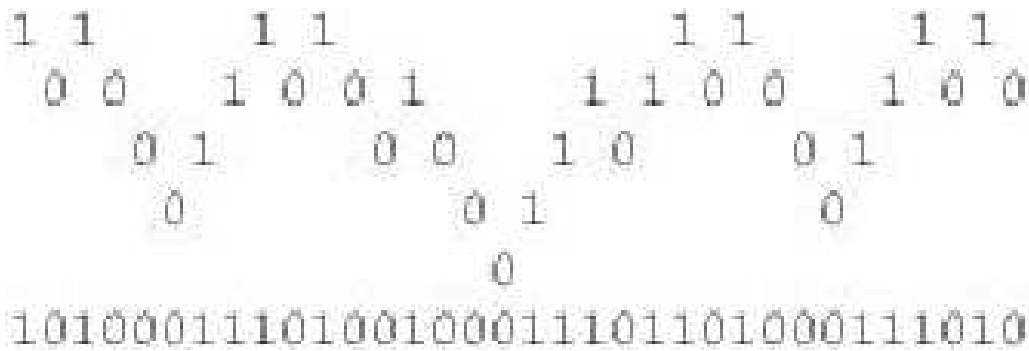
$k \setminus l$	3	4	5	6
2	9	35	178	>695
3	27	>291	>1209	>8885
4	76	>1047	>10436	>90306
5	>125	>2253	>24044	>177955

Formula for W(2,6)

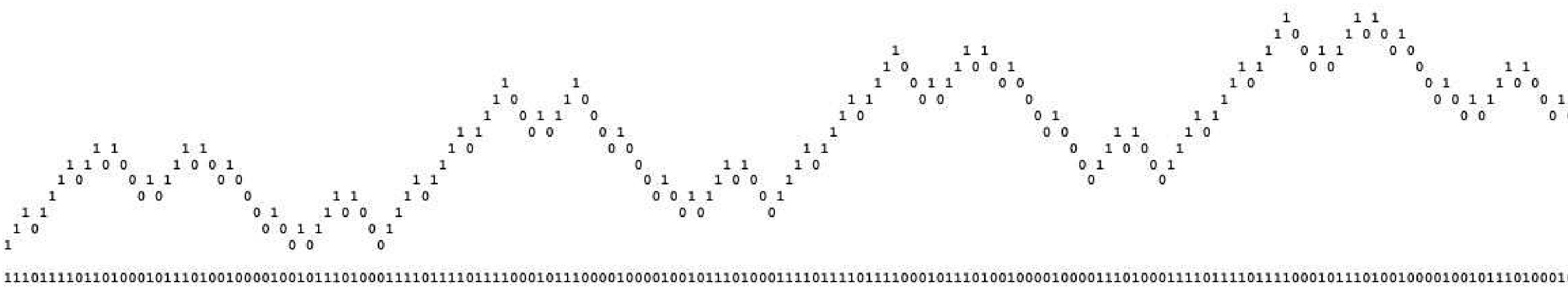
Variables	Meaning
v_i	$v_i \equiv 1$ if $i + n/2 \in C_1$ $v_i \equiv 0$ if $i + n/2 \in C_2$

Clauses	Meaning
$(\neg v_i \vee \neg v_{i+1} \vee \ldots \vee \neg v_{i+5})$ $(\neg v_i \vee \neg v_{i+2} \vee \ldots \vee \neg v_{i+10})$ $\ldots$ $(\neg v_i \vee \neg v_{i+t} \vee \ldots \vee \neg v_{i+5t})$	no arithmetic progression of length 6 in C_1 $-n/2 < i \leq n/2 - 5$ $t = \lfloor (n/2 - i)/5 \rfloor$
$(v_i \vee v_{i+1} \vee \ldots \vee v_{i+5})$ $(v_i \vee v_{i+2} \vee \ldots \vee v_{i+10})$ $\ldots$ $(v_i \vee v_{i+t} \vee \ldots \vee v_{i+5t})$	no arithmetic progression of length 6 in C_2 $-n/2 < i \leq n/2 - 5$ $t = \lfloor (n/2 - i)/5 \rfloor$

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 $W(2,l)*(l-2)/(l-1)$.

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

Clauses	Meaning
$(v_{-i} \vee v_{i+1})$ $\; 0 \leq i < s/2$ $(\neg v_{-i} \vee \neg v_{i+1})$	force $v_{-i} \equiv \bar{v}_{i+1}$.

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.

Clauses	**Filters**
$(v_i, \neg v_{i+t}, v_{i+2t}, \neg v_{i+3t}, v_{i+4t}, \neg v_{i+5t})$	010101
$(\neg v_i, v_{i+t}, \neg v_{i+2t}, v_{i+3t}, \neg v_{i+4t}, v_{i+5t})$	101010
$(v_i, v_{i+t}, \neg v_{i+2t}, \neg v_{i+3t}, v_{i+4t}, \neg v_{i+5t}, \neg v_{i+6t}, v_{i+7t})$	00110110
$(\neg v_i, \neg v_{i+t}, v_{i+2t}, v_{i+3t}, \neg v_{i+4t}, v_{i+5t}, v_{i+6t}, \neg v_{i+7t})$	11001001
$(v_i, \neg v_{i+t}, \neg v_{i+2t}, v_{i+3t}, \neg v_{i+4t}, \neg v_{i+5t}, v_{i+6t}, v_{i+7t})$	01101100
$(\neg v_i, v_{i+t}, v_{i+2t}, \neg v_{i+3t}, v_{i+4t}, v_{i+5t}, \neg v_{i+6t}, \neg v_{i+7t})$	10010011
$(v_i, v_{i+t}, \neg v_{i+2t}, \neg v_{i+3t}, \neg v_{i+4t}, v_{i+5t}, v_{i+6t}, \neg v_{i+7t})$	00111001
$(\neg v_i, \neg v_{i+t}, v_{i+2t}, v_{i+3t}, v_{i+4t}, \neg v_{i+5t}, \neg v_{i+6t}, v_{i+7t})$	11000110
$(\neg v_i, v_{i+t}, v_{i+2t}, \neg v_{i+3t}, \neg v_{i+4t}, \neg v_{i+5t}, v_{i+6t}, v_{i+7t})$	10011100
$(v_i, \neg v_{i+t}, \neg v_{i+2t}, v_{i+3t}, v_{i+4t}, v_{i+5t}, \neg v_{i+6t}, \neg v_{i+7t})$	01100011

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