Exploiting Constraint Solving History to Construct Interaction Test Suites

Myra Cohen

Matthew Dwyer

Jiangfan Shi

ESQuaReD

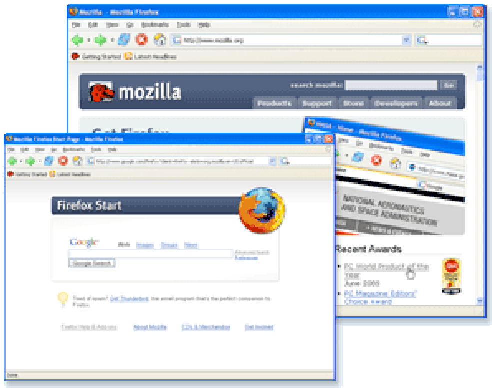
University of Nebraska Lincoln

Highly-Configurable Systems

- Highly-Configurable Systems

Office Applications

Database Servers



Web Browsers

Highly-Configurable Systems

- Highly-Configurable Systems

Software Product Lines

Dynamically Reconfigurable

Highly-Configurable Systems

- Features (bound) added/removed from the system
    - build, compile, run-time
- Module (feature) developers do not control or anticipate all possible feature combinations
- Encompass families of software systems

Phone Software Product Line

Possible Values	Product Line Options (factors)				
	Display	Email Viewer	Camera	Video Camera	Video Ringtones
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8 Million Colors	Text	1 Megapixel	No	No
	Black and White	None	None		

Phone Software Product Line

Possible Values	Product Line Options (factors)				
	Display	**Email Viewer**	**Camera**	**Video Camera**	**Video Ringtones**
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8 Million Colors	Text	1 Megapixel	No	No
	Black and White	None	None		

Phone Software Product Line

Possible Values	Product Line Options (factors)				
	Display	Email Viewer	Camera	Video Camera	Video Ringtones
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8 Million Colors	Text	1 Megapixel	No	No
	Black and White	None	None		

Phone Software Product Line

	Product Line Options (factors)				
Possible Values	Display	Email Viewer	Camera	Video Camera	Video Ringtones
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8 Million Colors	Text	1 Megapixel	No	No
	Black and White	None	None		

Phone Software Product Line

Possible Values	Product Line Options (factors)				
	Display	**Email Viewer**	**Camera**	**Video Camera**	**Video Ringtones**
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8 Million Colors	Text	1 Megapixel	No	No
	Black and White	None	None		

Phone Software Product Line

Possible Values	Product Line Options (factors)				
	Display	**Email Viewer**	**Camera**	**Video Camera**	**Video Ringtones**
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8 Million Colors	Text	1 Megapixel	No	No
	Black and White	None	None		

Phone Software Product Line

	Product Line Options (factors)				
Possible Values	**Display**	**Email Viewer**	**Camera**	**Video Camera**	**Video Ringtones**
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8 Million Colors	Text	1 Megapixel	No	No
	Black and White	None	None		

Testing the Product Line

Possible Values	Product Line Options (factors)				
	Display	Email Viewer	Camera	Video Camera	Video Ringtones
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8 Million Colors	Text	1 Megapixel	No	No
	Black and White	None	None		

Testing the Product Line

Possible Values	Product Line Options (factors)				
	Display	**Email Viewer**	**Camera**	**Video Camera**	**Video Ringtones**
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8 Million Colors	Text	1 Megapixel	No	No
	Black and White	None	None		

Test Case: ***Receive normal length text message***

Testing the Product Line

Possible Values	Product Line Options (factors)				
	Display	**Email Viewer**	**Camera**	**Video Camera**	**Video Ringtones**
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8 Million Colors	Text	1 Megapixel	No	No
	Black and White	None	None		

Test Case: ***Receive normal length text message***

Testing the Product Line

Possible Values	Product Line Options (factors)				
	Display	Email Viewer	Camera	Video Camera	Video Ringtones
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8				No
	Black and White	None	None		

Interaction Fault

Test Case: ***Receive normal length text message***

Testing the Product Line

Possible Values	Product Line Options (factors)				
	Display	**Email Viewer**	**Camera**	**Video Camera**	**Video Ringtones**
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8 Million Colors	Text	1 Megapixel	No	No
	Black and White	None	None		

Test Case: ***Receive normal length text message***

A Combinatorial Problem

	Display	Email Viewer	Camera	Video Camera	Video Ringtones
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8 Million Colors	Text	1 Megapixel	No	No
	Black and White	None	None		

A Combinatorial Problem

3

Display	Email Viewer	Camera	Video Camera	Video Ringtones
16 Million Colors	Graphical	2 Megapixels	Yes	Yes
8 Million Colors	Text	1 Megapixel	No	No
Black and White	None	None		

A Combinatorial Problem

Display	Email Viewer	Camera	Video Camera	Video Ringtones
16 Million Colors	Graphical	2 Megapixels	Yes	Yes
8 Million Colors	Text	1 Megapixel	No	No
Black and White	None	None		

3 x 3

A Combinatorial Problem

Display	Email Viewer	Camera	Video Camera	Video Ringtones
16 Million Colors	Graphical	2 Megapixels	Yes	Yes
8 Million Colors	Text	1 Megapixel	No	No
Black and White	None	None		

3 x 3 x 3

A Combinatorial Problem

	Display	Email Viewer	Camera	Video Camera	Video Ringtones
	16 Million Colors	Graphical	2 Megapixels	Yes	Yes
	8 Million Colors	Text	1 Megapixel	No	No
	Black and White	None	None		

3 x 3 x 3 x 2 x 2

- $3^3 x 2^2$=108 feasible configurations

A Combinatorial Problem

Display	Email Viewer	Camera	Video Camera	Video Ringtones
16 Million Colors	Graphical	2 Megapixels	Yes	Yes
8 Million Colors	Text	1 Megapixel	No	No
Black and White	None	None		

- $3^3 x 2^2$=108 feasible configurations
- 10 factors each with 5 values = 5^{10} or
  9,765,625 configurations

A Combinatorial Problem

Display	Email Viewer	Camera	Video Camera	Video Ringtones
16 Million Colors	Graphical	2 Megapixels	Yes	Yes
8 Million Colors	Text	1 Megapixel	No	No
Black and White	None	None		

- $3^3 x 2^2 = 108$ feasible configurations
- 10 factors each with 5 values = 5^{10} or 9,765,625 configurations
- 4 hours to run test suite = approximately 4,459 years to test

A Combinatorial Problem

Display	Email Viewer	Camera	Video Camera	Video Ringtones
16 Million Colors	Graphical	2 Megapixels	Yes	Yes
8 Million Colors	Text	1 Megapixel	No	No
Black and White	None	None		

GCC 4.1 optimizer configuration space:

199 factors:

- 189 with 2 values
- 10 factors with 3 values

A Combinatorial Problem

Display	Email Viewer	Camera	Video Camera	Video Ringtones
16 Million Colors	Graphical	2 Megapixels	Yes	Yes
8 Million Colors	Text	1 Megapixel	No	No
Black and White	None	None		

GCC 4.1 optimizer configuration space:

199 factors:

189 with 2 values

10 factors with 3 values

4.6×10^{61}

Sampling Configurations for Testing

- Desirable qualities for sampling technique:
  - Systematic
  - Quantifiable

Combinatorial Interaction Testing (CIT)

- Test all *pairs* or *t*-way combinations of factor-values, where *t* is a defined strength of testing
- Use mathematical object - covering array

Combinatorial Interaction Testing (CIT)

- Test all *pairs* or *t*-way combinations of factor-values, where *t* is a defined strength of testing
- Use mathematical object - covering array
- Algorithms for generating CIT samples:
  - Greedy search, meta-heuristic search, mathematical constructions

2-way CIT

	Display	Email Viewer	Camera	Video	V. Ringtones
1	16 Million Colors	None	1 Megapixel	Yes	Yes
2	8 Million Colors	Text	None	No	No
3	16 Million Colors	Text	2 Megapixels	No	Yes
4	Black and White	None	None	No	Yes
5	8 Million Colors	None	2 Megapixels	Yes	No
6	16 Million Colors	Graphical	None	Yes	No
7	Black and White	Text	1 Megapixel	Yes	No
8	8 Million Colors	Graphical	1 Megapixel	No	Yes
9	Black and White	Graphical	2 Megapixels	Yes	Yes

2-way CIT

	Display	Email Viewer	Camera	Video	V. Ringtones
1	16 Million Colors	None	1 Megapixel	Yes	Yes
2	8 Million Colors	Text	None	No	No
3	16 Million Colors	Text	2 Megapixels	No	Yes
4	Black and White	None	None	No	Yes
5	8 Million Colors	None	2 Megapixels	Yes	No
6	16 Million Colors	Graphical	None	Yes	No
7	Black and White	Text	1 Megapixel	Yes	No
8	8 Million Colors	Graphical	1 Megapixel	No	Yes
9	Black and White	Graphical	2 Megapixels	Yes	Yes

2-way CIT

	Display	Email Viewer	Camera	Video	V. Ringtones
1	16 Million Colors	None	1 Megapixel	Yes	Yes
2	8 Million Colors	Text	None	No	No
3	16 Million Colors	Text	2 Megapixels	No	Yes
4	Black and White	None	None	No	Yes
5	8 Million Colors	None	2 Megapixels	Yes	No
6	16 Million Colors	Graphical	None	Yes	No
7	Black and White	Text	1 Megapixel	Yes	No
8	8 Million Colors	Graphical	1 Megapixel	No	Yes
9	Black and White	Graphical	2 Megapixels	Yes	Yes

2-way CIT

	Display	Email Viewer	Camera	Video	V. Ringtones
1	16 Million Colors	None	1 Megapixel	Yes	Yes
2	8 Million Colors	Text	None	No	No
3	16 Million Colors	Text	2 Megapixels	No	Yes
4	Black and White	None	None	No	Yes
5	8 Million Colors	None	2 Megapixels	Yes	No
6	16 Million Colors	Graphical	None	Yes	No
7	Black and White	Text	1 Megapixel	Yes	No
8	8 Million Colors	Graphical	1 Megapixel	No	Yes
9	Black and White	Graphical	2 Megapixels	Yes	Yes

2-way CIT

	Display	Email Viewer	Camera	Video	V. Ringtones
1	16 Million Colors	None	1 Megapixel	Yes	Yes
2	8 Million Colors	Text	None	No	No
3	16 Million Colors	Text	2 Megapixels	No	Yes
4	Black and White	None	None	No	Yes
5	8 Million Colors	None	2 Megapixels	Yes	No
6	16 Million Colors	Graphical	None	Yes	No
7	Black and White	Text	1 Megapixel	Yes	No
8	8 Million Colors	Graphical	1 Megapixel	No	Yes
9	Black and White	Graphical	2 Megapixels	Yes	Yes

Limitations of CIT

- The theory of CIT, doesn't extend to most real systems

Limitations of CIT

- The theory of CIT, doesn't extend to most real systems
  - Many current algorithms and tools will fail to construct a valid sample in the presence of constraints…..

The *Real* System

Constraints on Valid Configurations:

(1) *Graphical email viewer* **requires** *color display*

(2) 2 Megapixel camera **requires** a *color display*

(3) *Graphical email viewer* **not supported** with *2 Megapixel camera*

(4) 8 Million color display **does not support** a 2 Megapixel camera

(5) *Video camera* **requires** a *camera* **and** a *color display*

(6) *Video ringtones* **cannot occur** with *No video camera*

(7) The combination of *16 Million colors*, *Text* and *2 Megapixel camera* will **not be supported**

The *Real* System

No	Forbidden Tuples	*Derived from*
1	*(Black and white display, Graphical email viewer)*	***1***
2	*(Black and white display, 2 Megapixel camera)*	***2***
3	*(Graphical email viewer, 2 Megapixel camera)*	***3***
4	*(8 Million color display, 2 Megapixel camera)*	***4***
5	*(Video camera=Yes, Camera=No)*	***5***
6	*(Video camera=Yes, Black and white display)*	***5***
7	*(Video ringtones= Yes, Video camera=No)*	***6***
8	*(16 Million colors, Plain text, 2 Megapixel camera)*	***7***

The *Real* System

No	Forbidden Tuples	*Derived from*
1	*(Black and white display, Graphical email viewer)*	***1***
2	*(Black and white display, 2 Megapixel camera)*	***2***
3	*(Graphical email viewer, 2 Megapixel camera)*	***3***
4	*(8 Million color display, 2 Megapixel camera)*	***4***
5	*(Video camera=Yes, Camera=No)*	***5***
6	*(Video camera=Yes, Black and white display)*	***5***
7	*(Video ringtones= Yes, Video camera=No)*	***6***
8	*(16 Million colors, Plain text, 2 Megapixel camera)*	***7***

2-way CIT Sample

	Display	Email Viewer	Camera	Video	V. Ringtones
1	16 Million Colors	None	1 Megapixel	Yes	Yes
2	8 Million Colors	Text	None	No	No
3	16 Million Colors	Text	2 Megapixels	No	Yes
4	Black and White	None	None	No	Yes
5	8 Million Colors	None	2 Megapixels	Yes	No
6	16 Million Colors	Graphical	None	Yes	No
7	Black and White	Text	1 Megapixel	Yes	No
8	8 Million Colors	Graphical	1 Megapixel	No	Yes
9	Black and White	Graphical	2 Megapixels	Yes	Yes

2-way CIT Sample

	Display	Email Viewer	Camera	Video	V. Ringtones
1	16 Million Colors	None	1 Megapixel	Yes	Yes
2	8 Million Colors	Text	None	No	No
3	16 Million Colors	Text	2 Megapixels	No	Yes
4	Black and White	None	None	No	Yes
5	8 Million Colors	[illegible]	[illegible]	[illegible]	[illegible]
6	16 Million Colors	[illegible]	[illegible]	[illegible]	[illegible]
7	Black and White	Text	1 Megapixel	[illegible]	[illegible]
8	8 Million Colors	Graphical	1 Megapixel	No	Yes
9	Black and White	Graphical	2 Megapixels	Yes	Yes

Violates Constraint: “Video Ringtones cannot occur without video camera”

2-way CIT Sample

	Display	Email Viewer	Camera	Video	V. Ringtones
1	16 Million Colors	None	1 Megapixel	Yes	Yes
2	8 Million Colors	Text	None	No	No
3	16 Million Colors	Text	2 Megapixels	No	Yes
4	Black and Whit				Yes
5	8 Million Colors				No
6	16 Million Colo				No
7	Black and White	Text	1 Megapixel	Yes	No
8	8 Million Colors	Graphical	1 Megapixel	No	Yes
9	Black and White	Graphical	2 Megapixels	Yes	Yes

Violates Constraint: “Video camera requires a camera and a color display”

2-way CIT Sample

	Display	**Email Viewer**	**Camera**	**Video**	**V. Ringtones**
1	16 Million Colors	None	1 Megapixel	Yes	Yes
2	8 Million Colors	Text	None	No	No
3	16 Million Colors	Text	2 Megapixels	No	Yes
4	Black and Whit				Yes
5	8 Million Colors				No
6	16 Million Colo				No
7	Black and White	Text	1 Megapixel	Yes	No
8	8 Million Colors	Graphical	1 Megapixel	No	Yes
9	Black and White	Graphical	2 Megapixels	Yes	Yes

Violates Constraint: “Video camera requires a camera and a color display”

2-way CIT Sample

	Display	Email Viewer	Camera	Video	V. Ringtones
1	16 Million Colors	None	1 Megapixel	Yes	Yes
2	8 Million Co	[illegible]	[illegible]	[illegible]	[illegible]
3	16 Million C	[illegible]	[illegible]	[illegible]	[illegible]
4	Black and W	[illegible]	[illegible]	[illegible]	[illegible]
5	8 Million Colors	None	2 Megapixels	Yes	No
6	16 Million Colors	Graphical	None	Yes	No
7	Black and White	Text	1 Megapixel	Yes	No
8	8 Million Colors	Graphical	1 Megapixel	No	Yes
9	Black and White	Graphical	2 Megapixels	Yes	Yes

Case Study Data: 96% of configurations in unconstrained CIT samples violate one or more constraints

Constrained CIT

- Few existing tools handle constraints
- Tools that do handle constraints, often require the user to re-model their system

Constrained CIT

- Few existing tools handle constraints
- Tools that do handle constraints, often require the user to re-model their system
- But: real systems have constraints

Existing Constraint Support

Algorithm	Constraint Handling	Re-Implementable
AETG	Remodel	Partial
DDA	Soft only	Yes
Whitch - CTS	Simple/Expand	No
Whitch - TOFU	Expand	No
IPO	None	--
TestCover	Remodel	No
Simulated Annealing	Soft only	Yes
PICT	Full	Partial
Constraint Solving	None	--

Existing Constraint Support

Algorithm	Constraint Handling	Re-Implementable
AETG	Remodel	Partial
DDA	Soft only	Yes
Whitch - CTS	Simple/Expand	No
Whitch - TOFU	Expand	No
IPO	None	--
TestCover	Remodel	No
Simulated Annealing	Soft only	Yes
PICT	Full	Partial
Constraint Solving	None	--

Our Original Approach
Constrained CIT

Represent constraints as boolean formulae

Use standard CIT algorithms

Use a SAT solver to:

1. Uncover all implicit constraints
2. Incrementally check for satisfiability as the algorithms proceed

Our Approach

CIT Algorithm

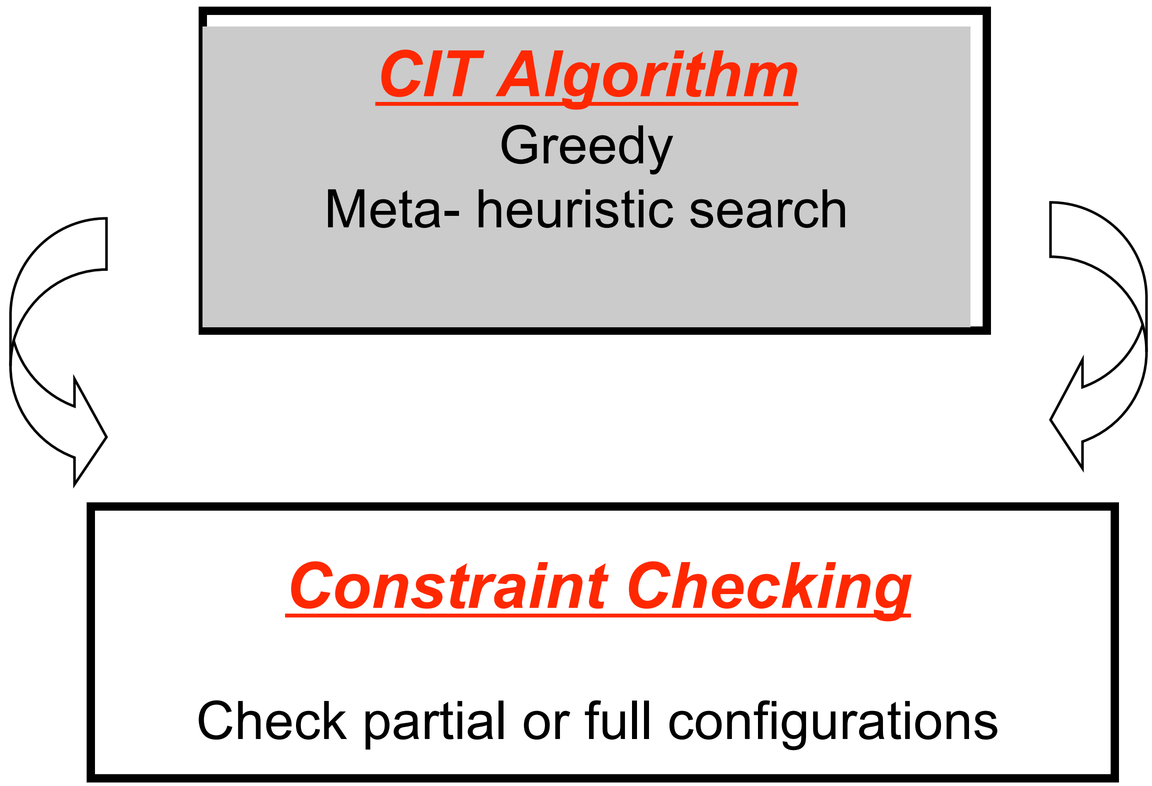
Greedy

Meta- heuristic search

Constraint Checking

Check partial or full configurations

Our Approach



Our Approach

CIT Algorithm

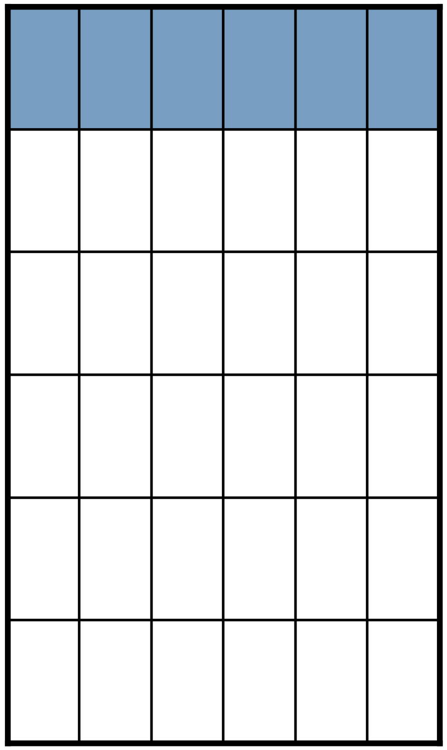
Greedy

Meta- heuristic search

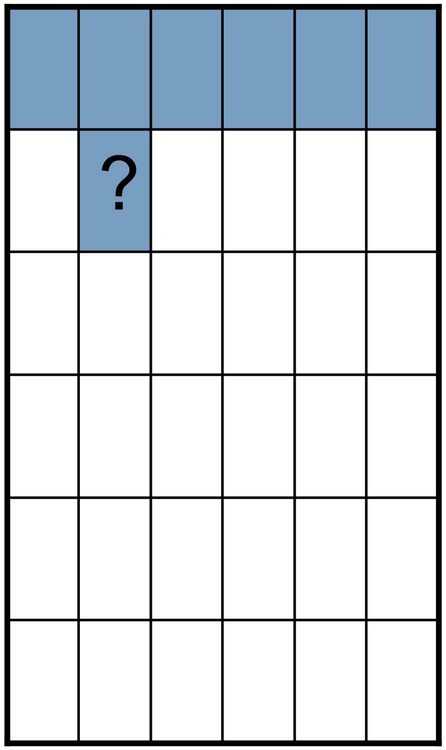
Constraint Checking

Check partial or full configurations

Greedy AETG-like

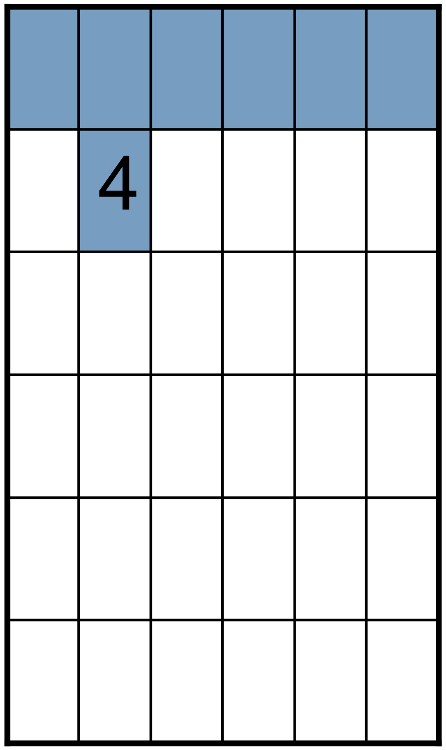


Greedy AETG-like

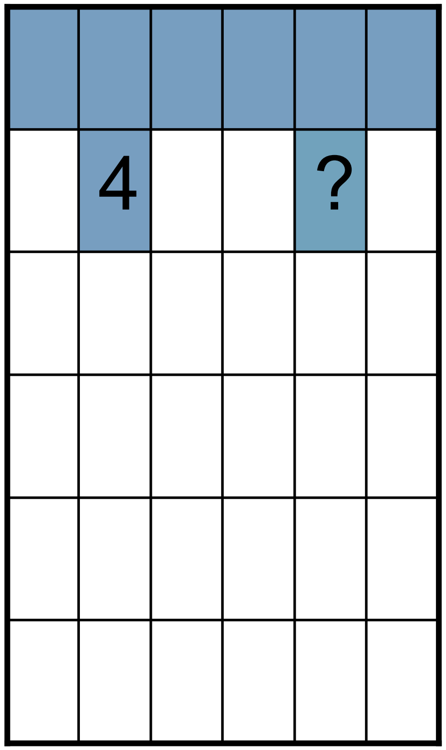


-select the first factor-value that covers most uncovered t-sets

Greedy AETG-like



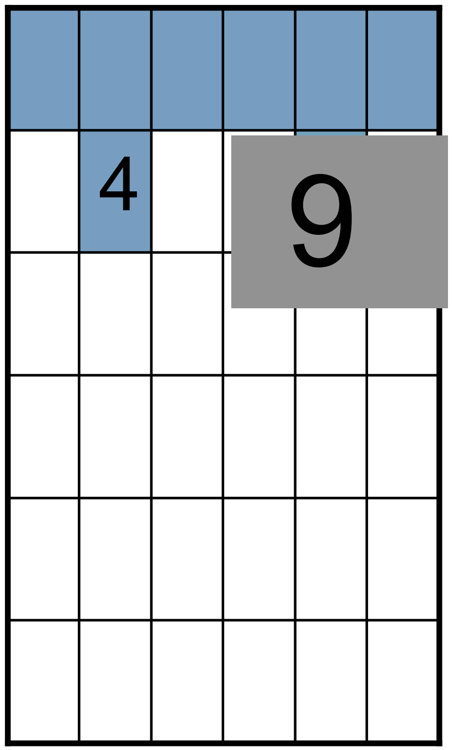
Greedy AETG-like



- Iterate through all values for the factor

- Select value that covers most new t-sets

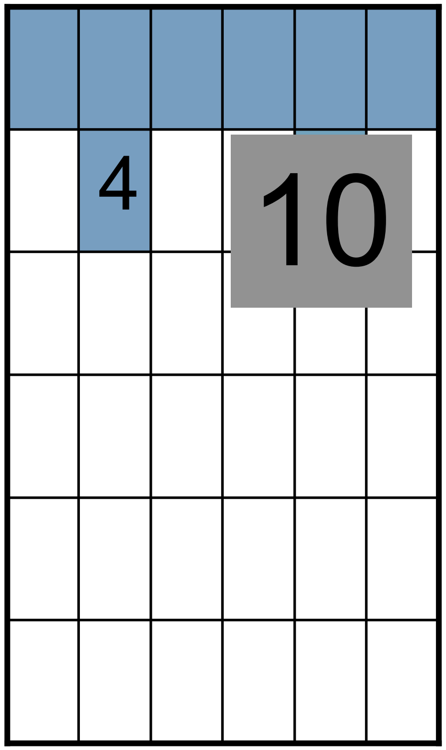
Greedy AETG-like



- Iterate through all values for the factor

- Select value that covers most new t-sets

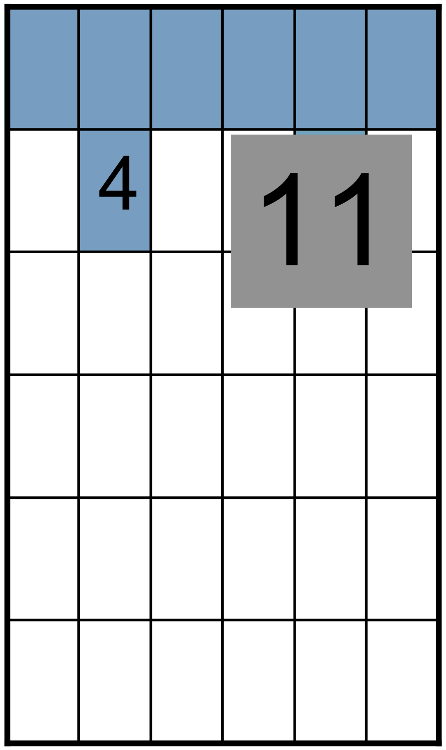
Greedy AETG-like



- Iterate through all values for the factor

- Select value that covers most new t-sets

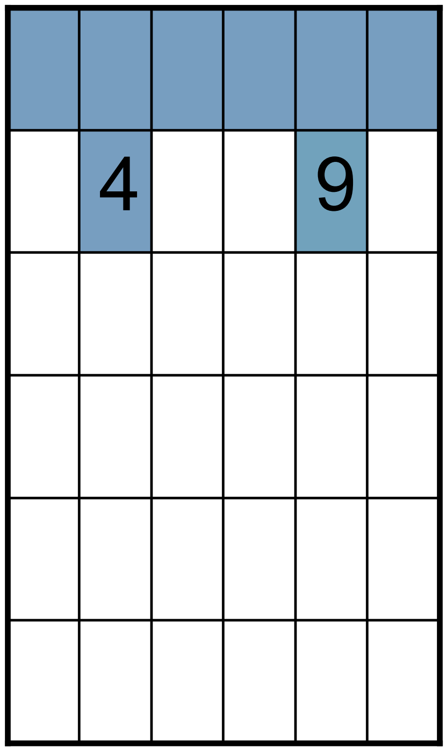
Greedy AETG-like



- Iterate through all values for the factor

- Select value that covers most new t-sets

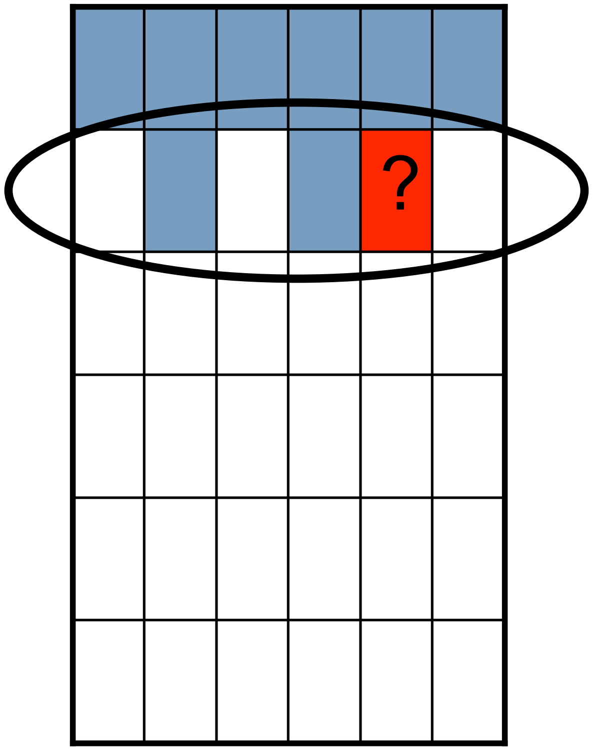
Greedy AETG-like



- Iterate through all values for the factor

- Select value that covers most new t-sets

Constraint Checking



GV ∧ *VC* ∧ *VR*

- If factor involved in constraints
- Pass partial configuration
- Check if SAT

Performance

- Although our method was feasible, the overhead was not insignificant
- This agrees with conventional wisdom in the CIT community:
  - "Large numbers of constraints may significantly increase the time to find a solution"

An Insight

- The most expensive portion of the AETG algorithm is selection of the "best" value at each step
- If we can mine data about infeasible and required values from the SAT solver's history we may be able to decrease the evaluations

Boolean Propagation

Factors:

f: {v1,v2,v3}
g: {v4,v5}
h: {v6,v7,v8}

Constraints:

require(g=v4,h=v6}
forbidden(f=v1,h=v6}

at-most:

{!x1,!x2},{!x1,!x3},{!x2,!x3}
{!x4,!x5}
{!x6,!x7},{!x6!,x8},{!x7,!x8}

at-least:

{x1,x2,x3}
{x4,x5}
{x6,x7,x8}

forbidden:

{!x4,!x7}
{!x4,!x8}
{!x1,!x6}

Boolean Propagation

X1

level 1

Factors:

f: {v1,v2,v3}
g: {v4,v5}
h: {v6,v7,v8}

Constraints:

require(g=v4,h=v6}
forbidden(f=v1,h=v6}

at-most:

{!x1,!x2},{!x1,!x3},{!x2,!x3}
{!x4,!x5}
{!x6,!x7},{!x6!,x8},{!x7,!x8}

at-least:

{x1,x2,x3}
{x4,x5}
{x6,x7,x8}

forbidden:

{!x4,!x7}
{!x4,!x8}
{!x1,!x6}

Boolean Propagation

X1

!X2

!X3

!X6

level 1

Factors:

f: {v1,v2,v3}
g: {v4,v5}
h: {v6,v7,v8}

Constraints:

require(g=v4,h=v6}
forbidden(f=v1,h=v6}

at-most:

{!x1,!x2},{!x1,!x3},{!x2,!x3}
{!x4,!x5}
{!x6,!x7},{!x6!,x8},{!x7,!x8}

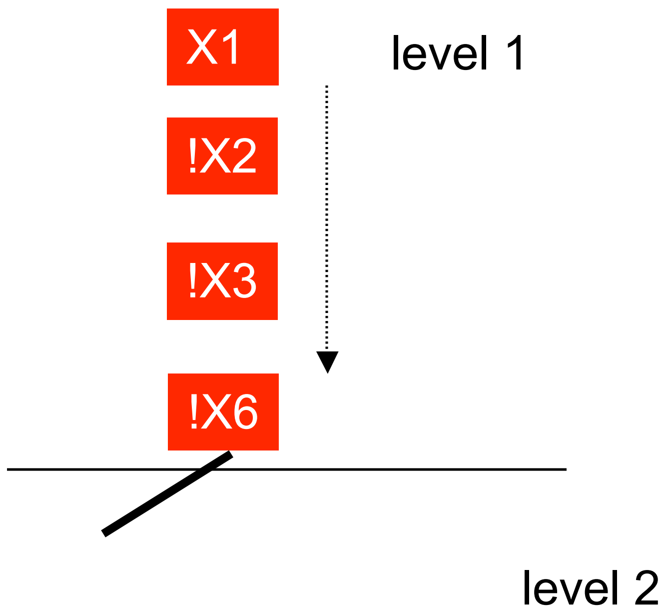
at-least:

{x1,x2,x3}
{x4,x5}
{x6,x7,x8}

forbidden:

{!x4,!x7}
{!x4,!x8}
{!x1,!x6}

Boolean Propagation



Factors:

f: {v1,v2,v3}
g: {v4,v5}
h: {v6,v7,v8}

Constraints:

require(g=v4,h=v6}
forbidden(f=v1,h=v6}

at-most:

{!x1,!x2},{!x1,!x3},{!x2,!x3}
{!x4,!x5}
{!x6,!x7},{!x6!,x8},{!x7,!x8}

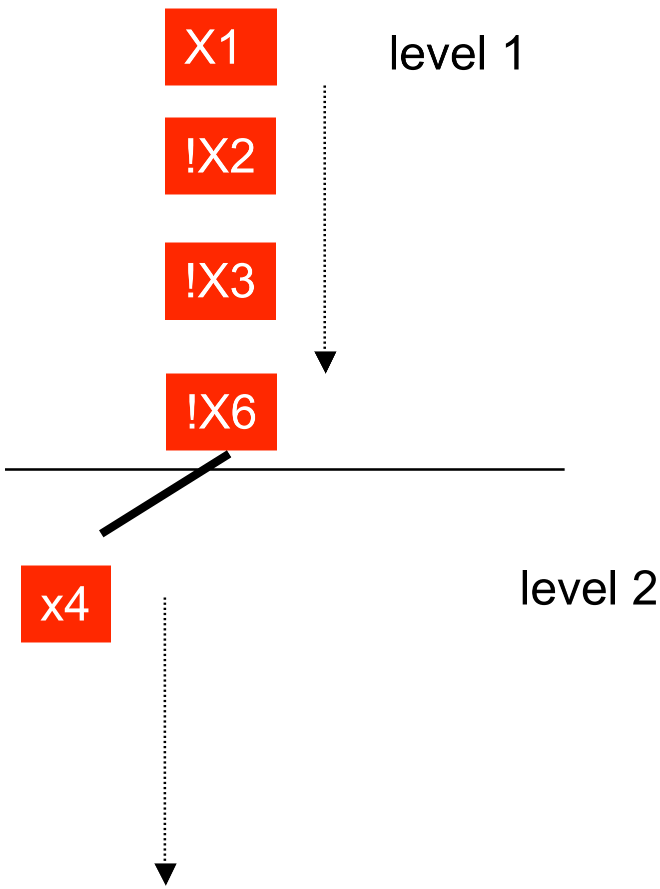
at-least:

{x1,x2,x3}
{x4,x5}
{x6,x7,x8}

forbidden:

{!x4,!x7}
{!x4,!x8}
{!x1,!x6}

Boolean Propagation



Factors:

f: {v1,v2,v3}
g: {v4,v5}
h: {v6,v7,v8}

Constraints:

require(g=v4,h=v6}
forbidden(f=v1,h=v6}

at-most:

{!x1,!x2},{!x1,!x3},{!x2,!x3}
{!x4,!x5}
{!x6,!x7},{!x6!,x8},{!x7,!x8}

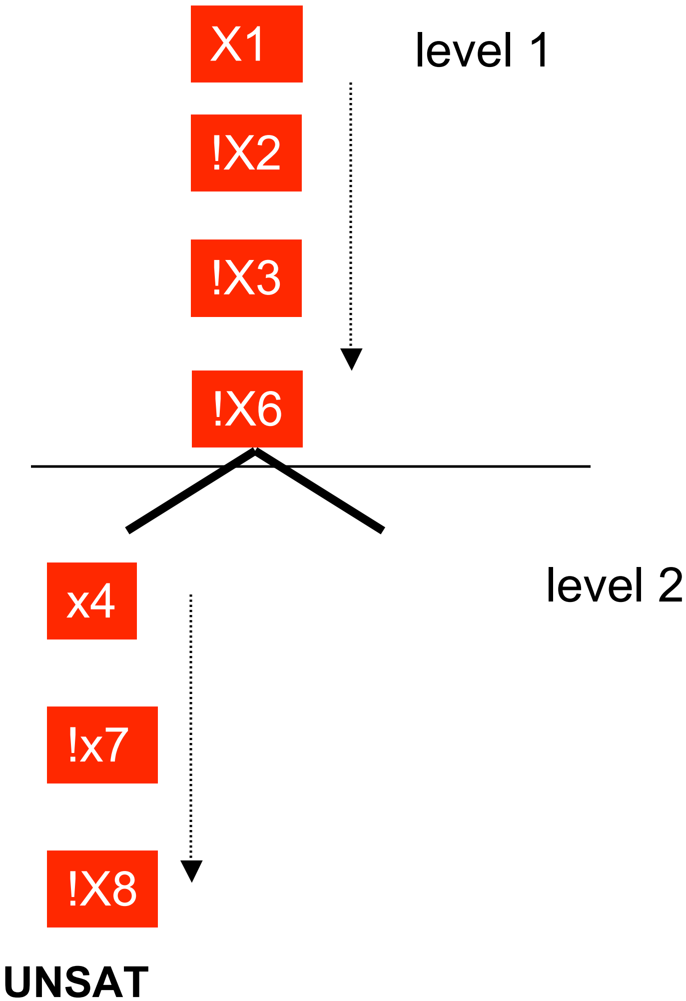
at-least:

{x1,x2,x3}
{x4,x5}
{x6,x7,x8}

forbidden:

{!x4,!x7}
{!x4,!x8}
{!x1,!x6}

Boolean Propagation



Factors:

f: {v1,v2,v3}
g: {v4,v5}
h: {v6,v7,v8}

Constraints:

require(g=v4,h=v6}
forbidden(f=v1,h=v6}

at-most:

{!x1,!x2},{!x1,!x3},{!x2,!x3}
{!x4,!x5}
{!x6,!x7},{!x6!,x8},{!x7,!x8}

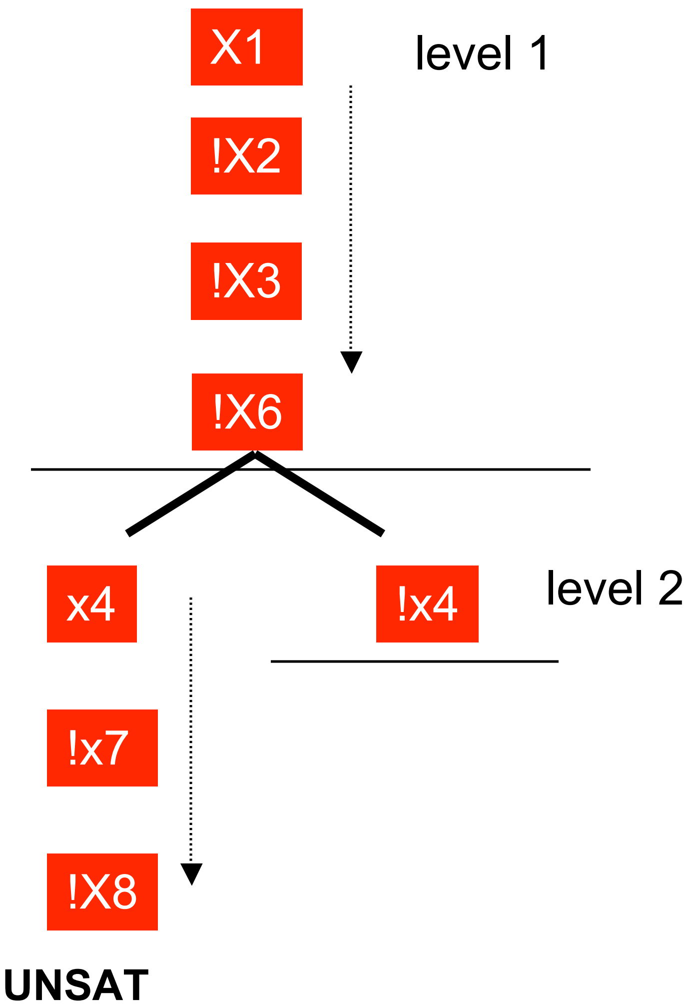
at-least:

{x1,x2,x3}
{x4,x5}
{x6,x7,x8}

forbidden:

{!x4,!x7}
{!x4,!x8}
{!x1,!x6}

Boolean Propagation



Factors:

f: {v1,v2,v3}
g: {v4,v5}
h: {v6,v7,v8}

Constraints:

require(g=v4,h=v6}
forbidden(f=v1,h=v6}

at-most:

{!x1,!x2},{!x1,!x3},{!x2,!x3}
{!x4,!x5}
{!x6,!x7},{!x6!,x8},{!x7,!x8}

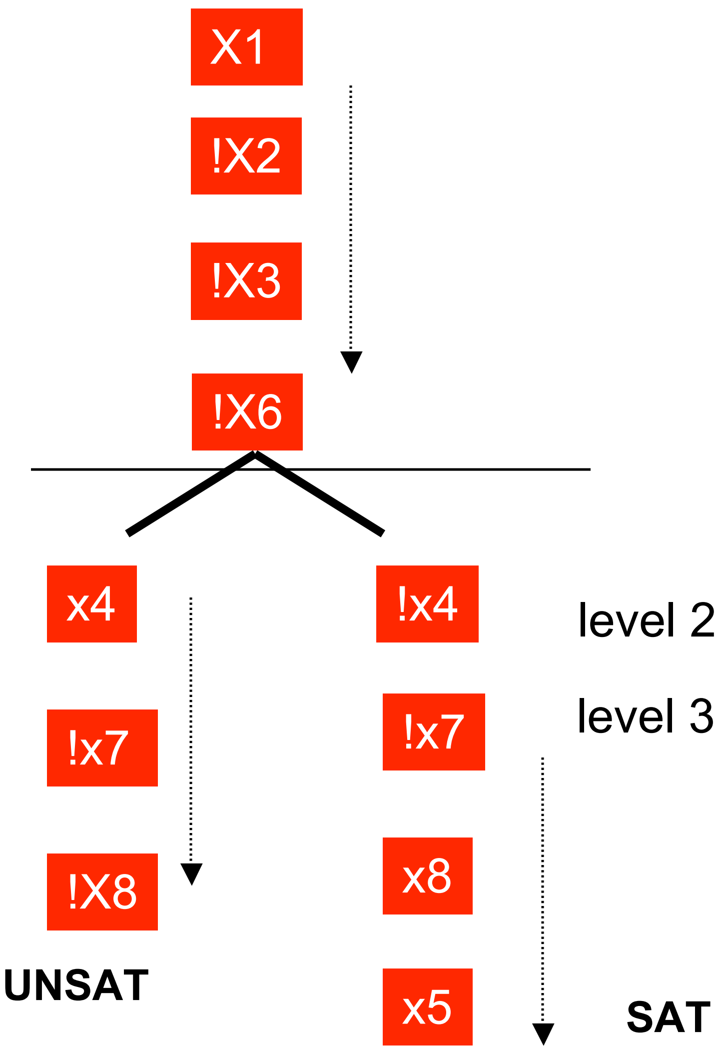
at-least:

{x1,x2,x3}
{x4,x5}
{x6,x7,x8}

forbidden:

{!x4,!x7}
{!x4,!x8}
{!x1,!x6}

Boolean Propagation



Factors:

f: {v1,v2,v3}
g: {v4,v5}
h: {v6,v7,v8}

Constraints:

require(g=v4,h=v6}
forbidden(f=v1,h=v6}

at-most:

{!x1,!x2},{!x1,!x3},{!x2,!x3}
{!x4,!x5}
{!x6,!x7},{!x6!,x8},{!x7,!x8}

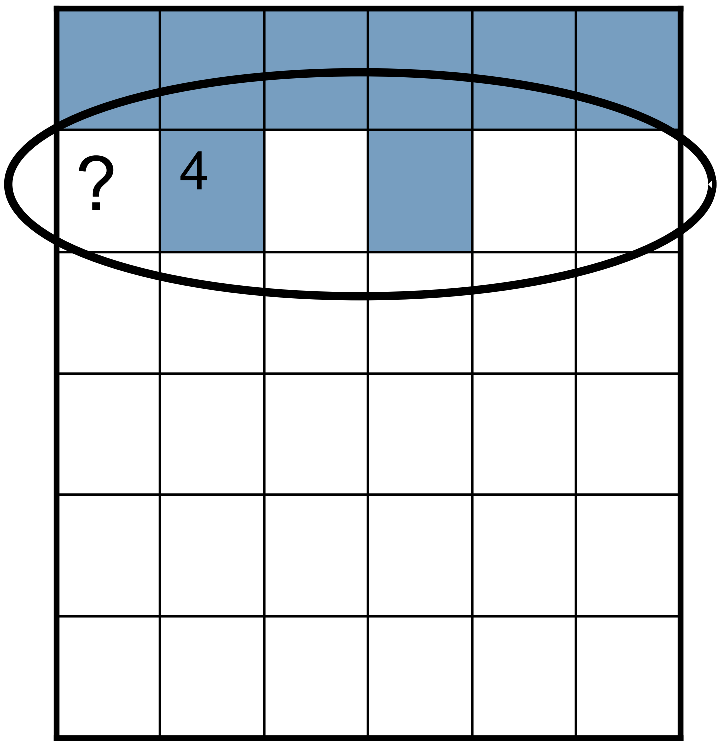
at-least:

{x1,x2,x3}
{x4,x5}
{x6,x7,x8}

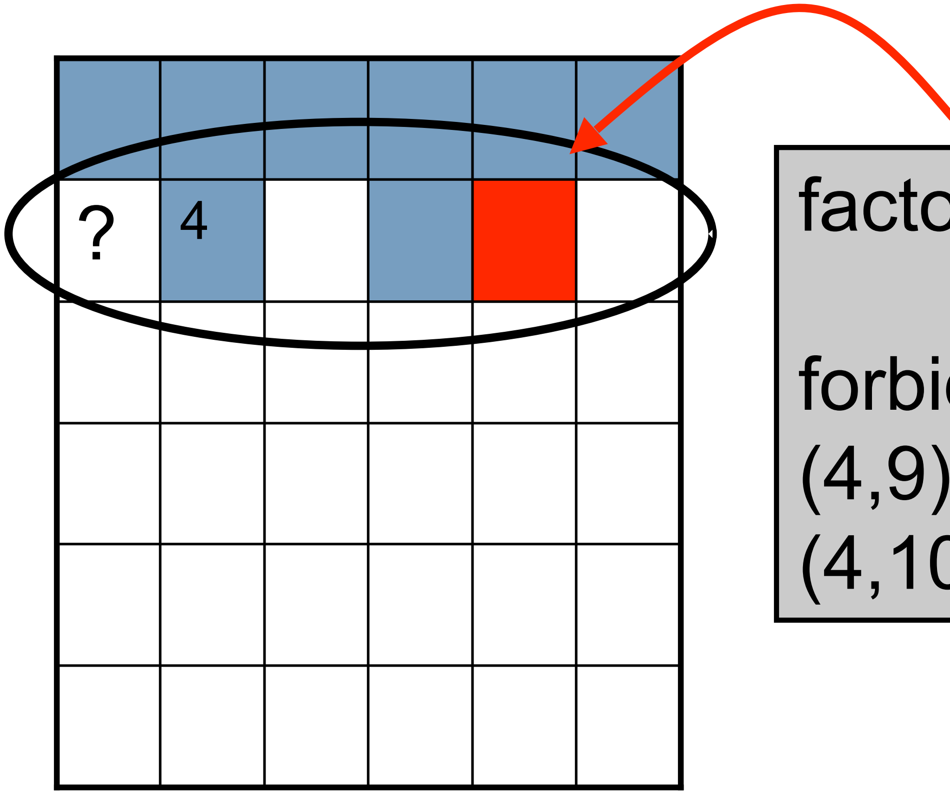
forbidden:

{!x4,!x7}
{!x4,!x8}
{!x1,!x6}

Filling in Factor 1



Filling in Factor 1



factor 5:

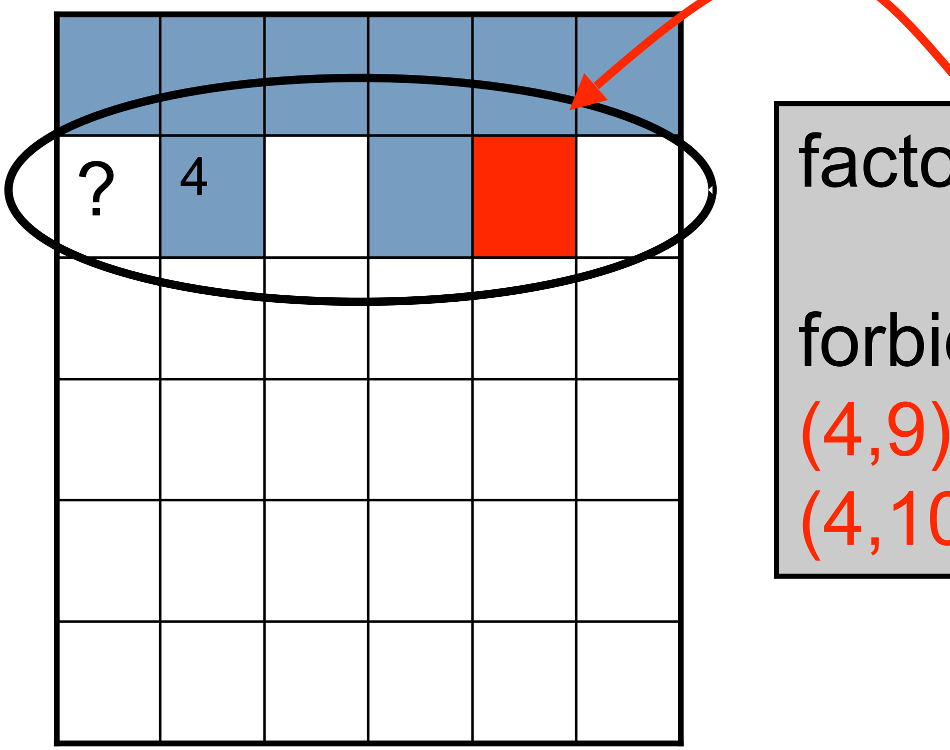
9,10,11

forbidden tuples:

(4,9)

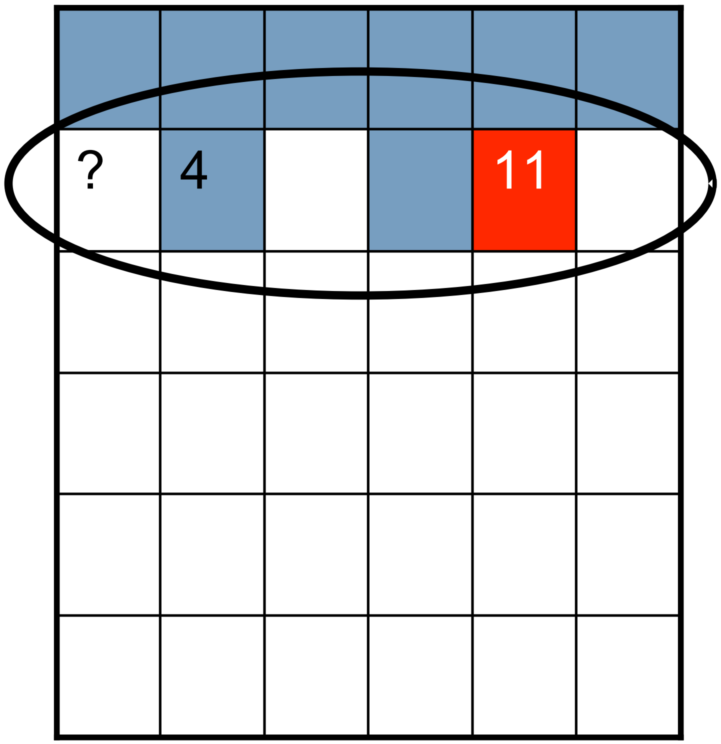
(4,10)

Filling in Factor 1



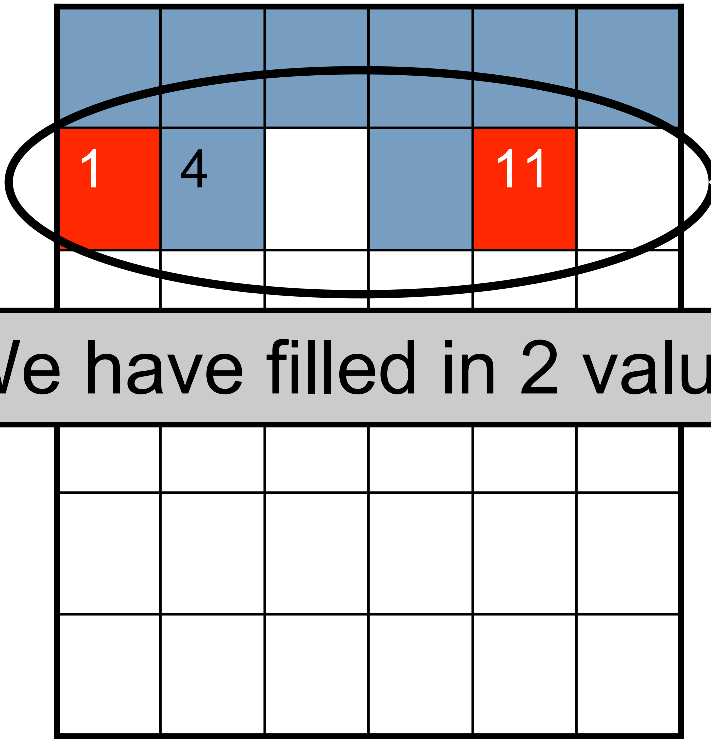
factor 5:
9,10,11
forbidden tuples:
(4,9)
(4,10)

Greedy AETG-like



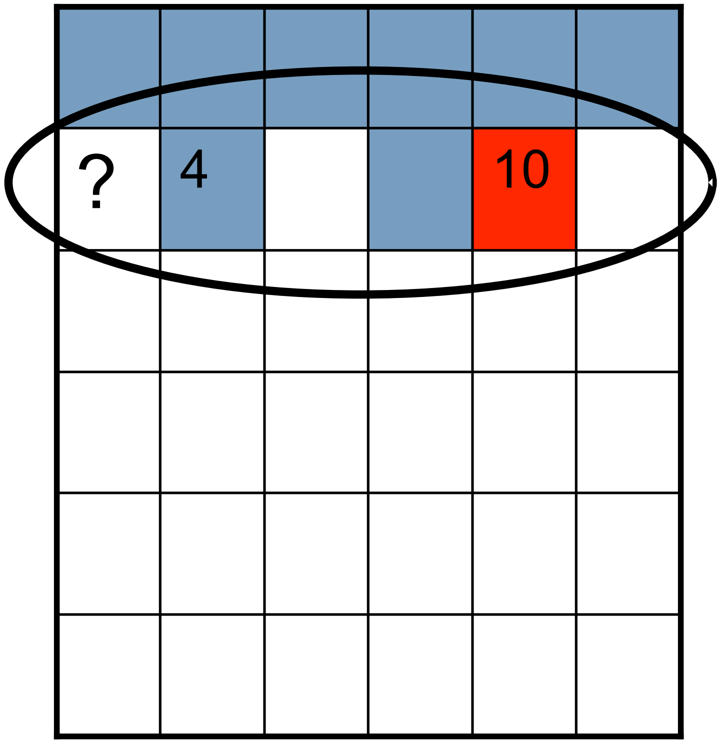
The SAT solver will deduce that factor 5 MUST be 11

Filling in Factor 1



We have filled in 2 values at once

Filling in Factor 1



If instead:

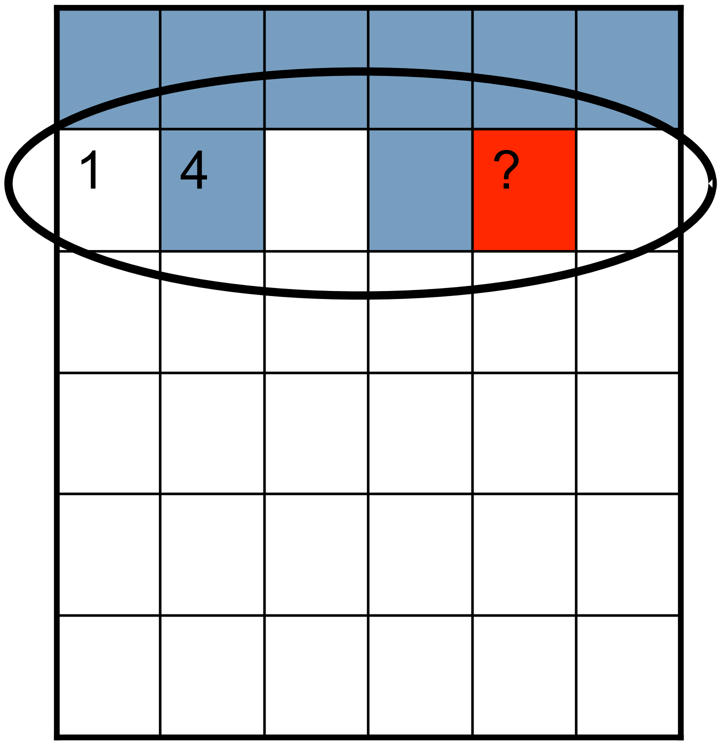
factor 5:

9,10,11

forbidden tuples:

(4,9)

Filling in Factor 1



Have 2 values on a

MAY list

10 or 11

This prunes the space that AETG has to search

History Based Approach

- At each step of the algorithm
  - Mine the MUST and MAY -lists
  - Fill in MUST values
  - Only evaluate MAY values for a factor
  - Continue to check SAT at end of selection

Evaluation

- We evaluated this on 5 case studies we have conducted from the literature
- A set of 30 simulated data sets based on the characteristics our our case studies
- Use MiniSAT: an incremental SAT solver

Case Studies

	Num Factors	**Values**	**Num Cons**	**Factor Involved**	**Cons Arity**
SPINs	18	2-4	13	9	2
SPINv	55	2-4	49	33	2-3
GCC	199	2-3	40	36	2-3
Apache	172	2-6	7	18	2-5
Bugzilla	82	2-4	5	11	2-3

Techniques

- We compared several versions of our algorithms:
  - Base version with no optimization
  - Incremental version - uses incremental SAT solving
  - History version - mines MUST and MAY data

Case Studies: SAT calls

	Average # SAT calls (50 runs)			
	Base SAT	**Inc SAT**	**Hist SAT**	**% Dec**
SPINs	14,557	14,480	7,993	44.8
SPINv	97,379	95,848	37,021	61.4
GCC	57,388	55,432	31,089	43.9
Apac	44,199	40,088	32,687	18.5
Bugz	15,691	15,353	10,609	30.9

Case Studies: SAT calls

	Average # SAT calls (50 runs)			
	Base SAT	**Inc SAT**	**Hist SAT**	**% Dec**
SPINs	14,557	14,480	7,993	44.8
SPINv	97,379	95,848	37,021	61.4
GCC	57,388	55,432	31,089	43.9
Apac	44,199	40,088	32,687	18.5
Bugz	15,691	15,353	10,609	30.9

Case Studies: SAT calls

	Average # SAT calls (50 runs)			
	Base SAT	**Inc SAT**	**Hist SAT**	**% Dec**
SPINs	14,557	14,480	7,993	44.8
SPINv	97,379	95,848	37,021	61.4
GCC	57,388	55,432	31,089	43.9
Apac	44,199	40,088	32,687	18.5
Bugz	15,691	15,353	10,609	30.9

Case Studies: SAT calls

	Average # SAT calls (50 runs)			
	Base SAT	**Inc SAT**	**Hist SAT**	**% Dec**
SPINs	14,557	14,480	7,993	44.8
SPINv	97,379	95,848	37,021	61.4
GCC	57,388	55,432	31,089	43.9
Apac	44,199	40,088	32,687	18.5
Bugz	15,691	15,353	10,609	30.9

Case Studies: Time

	Average Time in Seconds (50 runs)					Avg Size	
	mAETG	**Base SAT**	**Inc SAT**	**Hist SAT**	**% Inc**	**Basic SAT**	**Hist SAT**
SPINs	0.3	1.7	0.4	0.3	8.4	27	27
SPINv	8.2	32.2	11.3	8.5	3.8	43	43
GCC	217.6	320.0	286.9	204.0	**-6.2**	25	25
Apac	278.7	318.6	249.2	244.1	**-12.4**	43	43
Bugz	4.4	7.4	6.2	4.4	**-0.3**	25	25

Case Studies: Time

	Average Time in Seconds (50 runs)					Avg Size	
	mAETG	**Base SAT**	**Inc SAT**	**Hist SAT**	**% Inc**	**Basic SAT**	**Hist SAT**
SPINs	0.3	1.7	0.4	0.3	8.4	27	27
SPINv	8.2	32.2	11.3	8.5	3.8	43	43
GCC	217.6	320.0	286.9	204.0	**-6.2**	25	25
Apac	278.7	318.6	249.2	244.1	**-12.4**	43	43
Bugz	4.4	7.4	6.2	4.4	**-0.3**	25	25

Case Studies: Time

	Average Time in Seconds (50 runs)					Avg Size	
	mAETG	**Base SAT**	**Inc SAT**	**Hist SAT**	**% Inc**	**Basic SAT**	**Hist SAT**
SPINs	0.3	1.7	0.4	0.3	8.4	27	27
SPINv	8.2	32.2	11.3	8.5	3.8	43	43
GCC	217.6	320.0	286.9	204.0	**-6.2**	25	25
Apac	278.7	318.6	249.2	244.1	**-12.4**	43	43
Bugz	4.4	7.4	6.2	4.4	**-0.3**	25	25

Case Studies: Time

	Average Time in Seconds (50 runs)					Avg Size	
	mAETG	**Base SAT**	**Inc SAT**	**Hist SAT**	**% Inc**	**Basic SAT**	**Hist SAT**
SPINs	0.3	1.7	0.4	0.3	8.4	27	27
SPINv	8.2	32.2	11.3	8.5	3.8	43	43
GCC	217.6	320.0	286.9	204.0	**-6.2**	25	25
Apac	278.7	318.6	249.2	244.1	**-12.4**	43	43
Bugz	4.4	7.4	6.2	4.4	**-0.3**	25	25

Case Studies: Time

	Average Time in Seconds (50 runs)					Avg Size	
	mAETG	**Base SAT**	**Inc SAT**	**Hist SAT**	**% Inc**	**Basic SAT**	**Hist SAT**
SPINs	0.3	1.7	0.4	0.3	8.4	27	27
SPINv	8.2	32.2	11.3	8.5	3.8	43	43
GCC	217.6	320.0	286.9	204.0	**-6.2**	25	25
Apac	278.7	318.6	249.2	244.1	**-12.4**	43	43
Bugz	4.4	7.4	6.2	4.4	**-0.3**	25	25

Case Studies: Time

	Average Time in Seconds (50 runs)					Avg Size	
	mAETG	**Base SAT**	**Inc SAT**	**Hist SAT**	**% Inc**	**Basic SAT**	**Hist SAT**
SPINs	0.3	1.7	0.4	0.3	8.4	27	27
SPINv	8.2	32.2	11.3	8.5	3.8	43	43
GCC	217.6	320.0	286.9	204.0	**-6.2**	25	25
Apac	278.7	318.6	249.2	244.1	**-12.4**	43	43
Bugz	4.4	7.4	6.2	4.4	**-0.3**	25	25

Across All Samples

	SAT Calls			
	Base SAT	Inc SAT	Hist SAT	% Dec
Avg	**55,793**	**55,058**	**24,265**	**59.0**

	Time in Seconds				
	mAETG	Base SAT	Inc SAT	Hist SAT	% Inc
Avg	**83.5**	**114.1**	**92.3**	**76.1**	**-9.8**

Conclusions

- Most real industrial problems have non-trivial constraints - existing CIT methods will not scale
- Our method integrates constraints in such a way that reduces CIT solution time over unconstrained solutions without sacrificing quality
- Mining SAT history reduces the number of SAT calls and evaluations by reducing decisions made during the AETG algorithm

Conclusions

- Most real industrial problems have non-trivial constraints - existing CIT methods will not scale
- Our method integrates constraints in such a way that reduces CIT solution time over unconstrained solutions without sacrificing quality
- Mining SAT history reduces the number of SAT calls and evaluations by reducing decisions made during the AETG algorithm
- In new work we have driven down the cost without sacrificing quality even further…

Future Work

- Additional case studies
- Integrating more tightly with meta-heuristic search algorithms

Acknowledgements

This work was supported in part by the Army Research Office through DURIP award W91NF-04-1-0104, and by the National Science Foundation through awards 0429149, 0444167, 0454203, and 0541263.