Dynamic Scoring Functions with Variable Expressions:

New SLS Methods for Solving SAT

by Dave Tompkins & Holger Hoos

SAT 2010 :: Edinburgh, Scotland

Key Contributions

- Variable Expressions (VEs)
- New Conceptual Model for SLS
- Design Architecture for VEs (DAVE)

Stochastic Local Search (SLS) for SAT

```
randomly initialize all variables
while (formula not satisfied)
    select a variable and “flip” it
```

x_1	x_2	x_3	x_4	x_5

$$(\neg x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4)$$

Stochastic Local Search (SLS) for SAT

```
randomly initialize all variables
while (formula not satisfied)
    select a variable and “flip” it
```

x_1	x_2	x_3	x_4	x_5
T	F	F	T	T

$(\neg x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4)$
$(\neg x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4)$

Stochastic Local Search (SLS) for SAT

```
randomly initialize all variables
while (formula not satisfied)
    select a variable and “flip” it
```

x_1	x_2	x_3	x_4	x_5
T	F	F	T	T

$(\neg x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4)$
$(\neg x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4)$

Stochastic Local Search (SLS) for SAT

```
randomly initialize all variables
while (formula not satisfied)
    select a variable and "flip" it
```

x_1	x_2	x_3	x_4	x_5
T	F	F	T	T
T	F	F	F	T

$(\neg x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4)$
$(\neg x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4)$
$(\neg x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4)$

Stochastic Local Search (SLS) for SAT

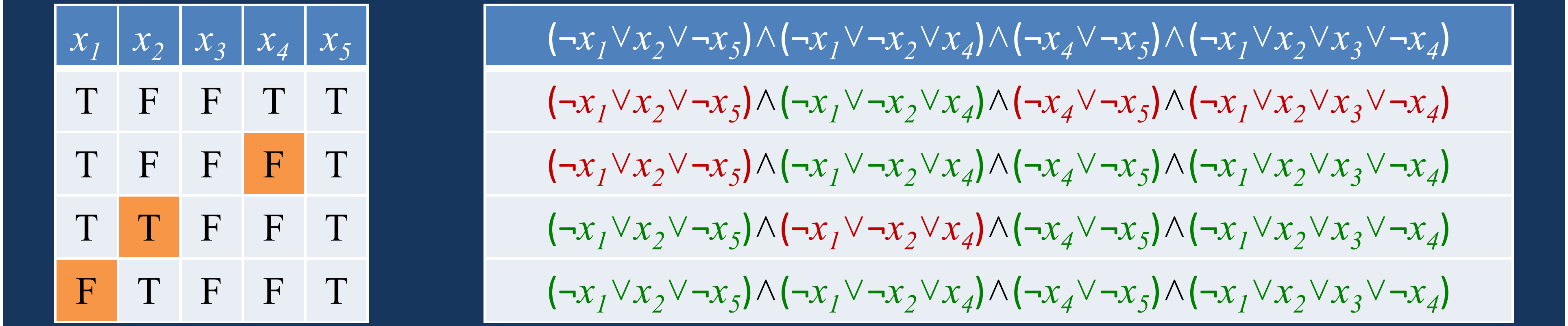
```
randomly initialize all variables
while (formula not satisfied)
    select a variable and "flip" it
```

x_1	x_2	x_3	x_4	x_5
T	F	F	T	T
T	F	F	F	T
T	T	F	F	T

$(\neg x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4)$
$(\neg x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4)$
$(\neg x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4)$
$(\neg x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_4 \vee \neg x_5) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4)$

Stochastic Local Search (SLS) for SAT

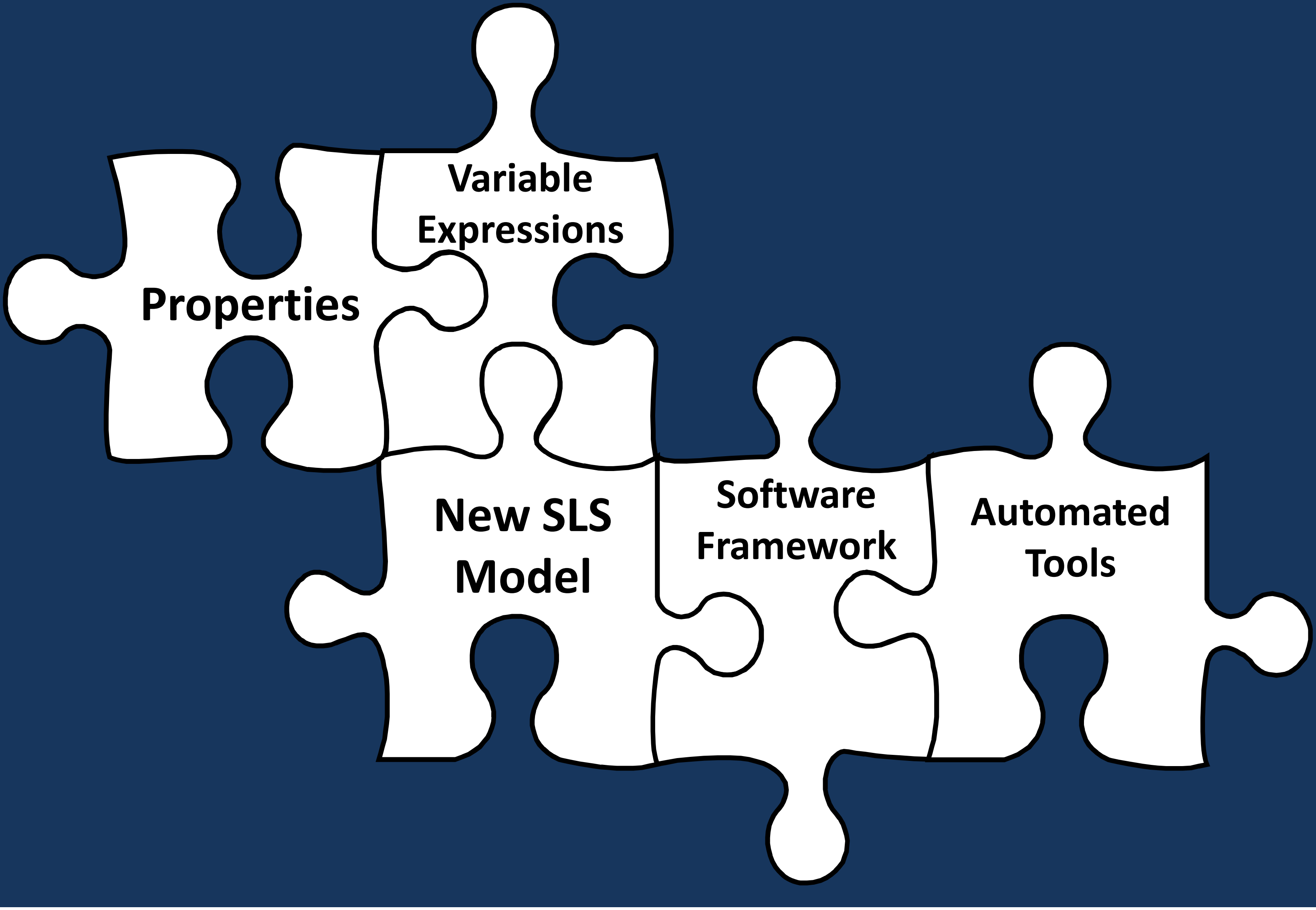
```
randomly initialize all variables
while (formula not satisfied)
    select a variable and "flip" it
```



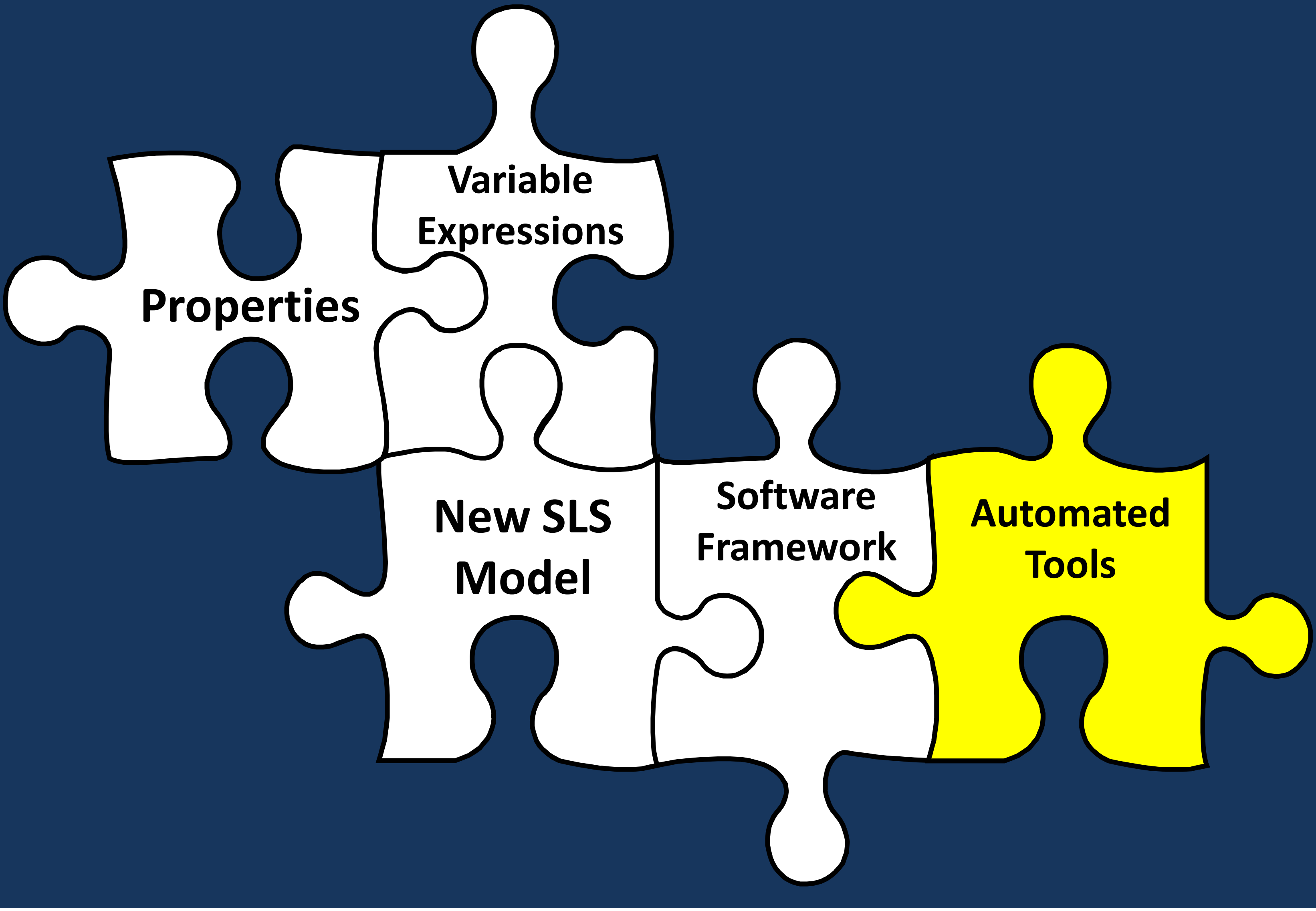
x_1	x_2	x_3	x_4	x_5
T	F	F	T	T
T	F	F	F	T
T	T	F	F	T
F	T	F	F	T

$(\neg x_1 \lor x_2 \lor \neg x_5) \land (\neg x_1 \lor \neg x_2 \lor x_4) \land (\neg x_4 \lor \neg x_5) \land (\neg x_1 \lor x_2 \lor x_3 \lor \neg x_4)$
$(\neg x_1 \lor x_2 \lor \neg x_5) \land (\neg x_1 \lor \neg x_2 \lor x_4) \land (\neg x_4 \lor \neg x_5) \land (\neg x_1 \lor x_2 \lor x_3 \lor \neg x_4)$
$(\neg x_1 \lor x_2 \lor \neg x_5) \land (\neg x_1 \lor \neg x_2 \lor x_4) \land (\neg x_4 \lor \neg x_5) \land (\neg x_1 \lor x_2 \lor x_3 \lor \neg x_4)$
$(\neg x_1 \lor x_2 \lor \neg x_5) \land (\neg x_1 \lor \neg x_2 \lor x_4) \land (\neg x_4 \lor \neg x_5) \land (\neg x_1 \lor x_2 \lor x_3 \lor \neg x_4)$
$(\neg x_1 \lor x_2 \lor \neg x_5) \land (\neg x_1 \lor \neg x_2 \lor x_4) \land (\neg x_4 \lor \neg x_5) \land (\neg x_1 \lor x_2 \lor x_3 \lor \neg x_4)$

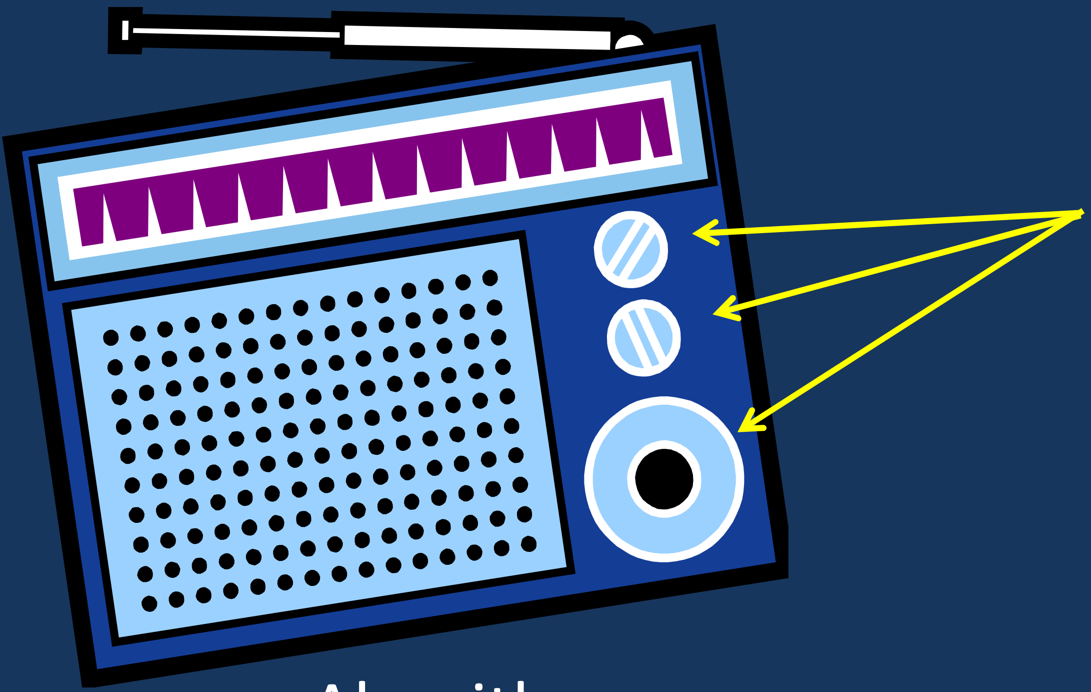
Overview


Variable Expressions
Properties
New SLS Model
Software Framework
Automated Tools

Overview


Variable Expressions
Properties
New SLS Model
Software Framework
Automated Tools

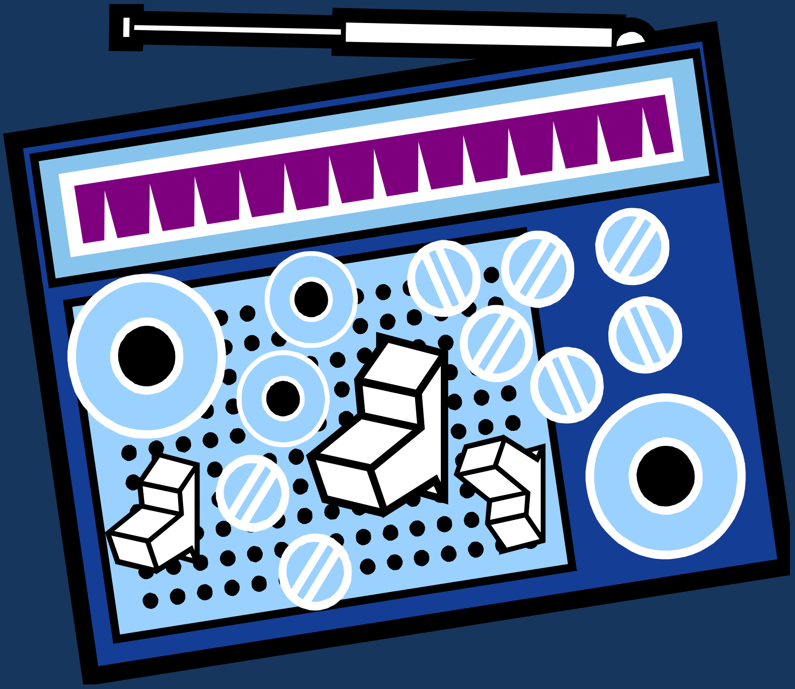
Automated Design Philosophy



Algorithm Parameters

Algorithm

Automated Design Philosophy



- Don’t be afraid to expose more parameters
- Don’t “fix” early design choices

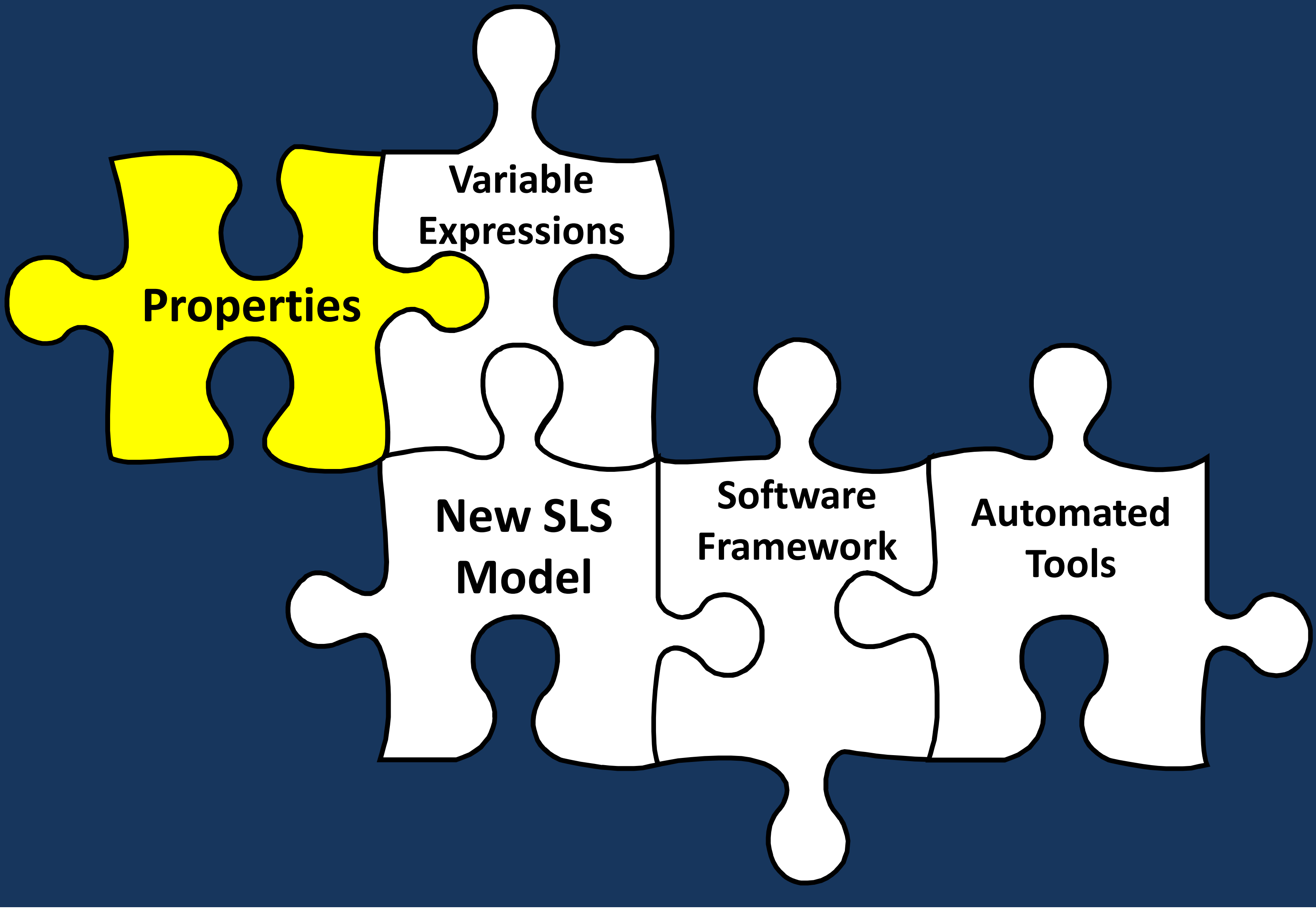
Automated Tools

- We can use automated configurators to determine the optimal algorithm parameters for a target instance set

- ParamILS [Hutter *et al.*, 2007]

- Offload tedious human tasks to machines

Overview


Properties
Variable Expressions
New SLS Model
Software Framework
Automated Tools

Variable Properties

- Scoring Properties

  make = # of clauses that become satisfied if we flip x

  break = ... unsatisfied ...

  score = (make – break) [GSAT, Selman *et al.* 1992]

Variable Properties

- Scoring Properties

  make = # of clauses that become satisfied if we flip x

  break = … unsatisfied …

  score = (make – break) [GSAT, Selman *et al.* 1992]

- Dynamic Properties

  age = # of steps since x was flipped [TABU, Glover 1986]

  flips = # of times x has been flipped [HSAT, Gent & Walsh 1992]

Variable Properties

- Scoring Properties

  make = # of clauses that become satisfied if we flip x

  break = … unsatisfied …

  score = (make – break) [GSAT, Selman *et al.* 1992]

- Dynamic Properties

  age = # of steps since x was flipped [TABU, Glover 1986]

  flips = # of times x has been flipped [HSAT, Gent & Walsh 1992]

- Static Properties

Variable Properties

- Scoring Properties

  make = # of clauses that become satisfied if we flip x

  break = … unsatisfied …

  score = (make – break) [GSAT, Selman *et al.* 1992]

- Dynamic Properties

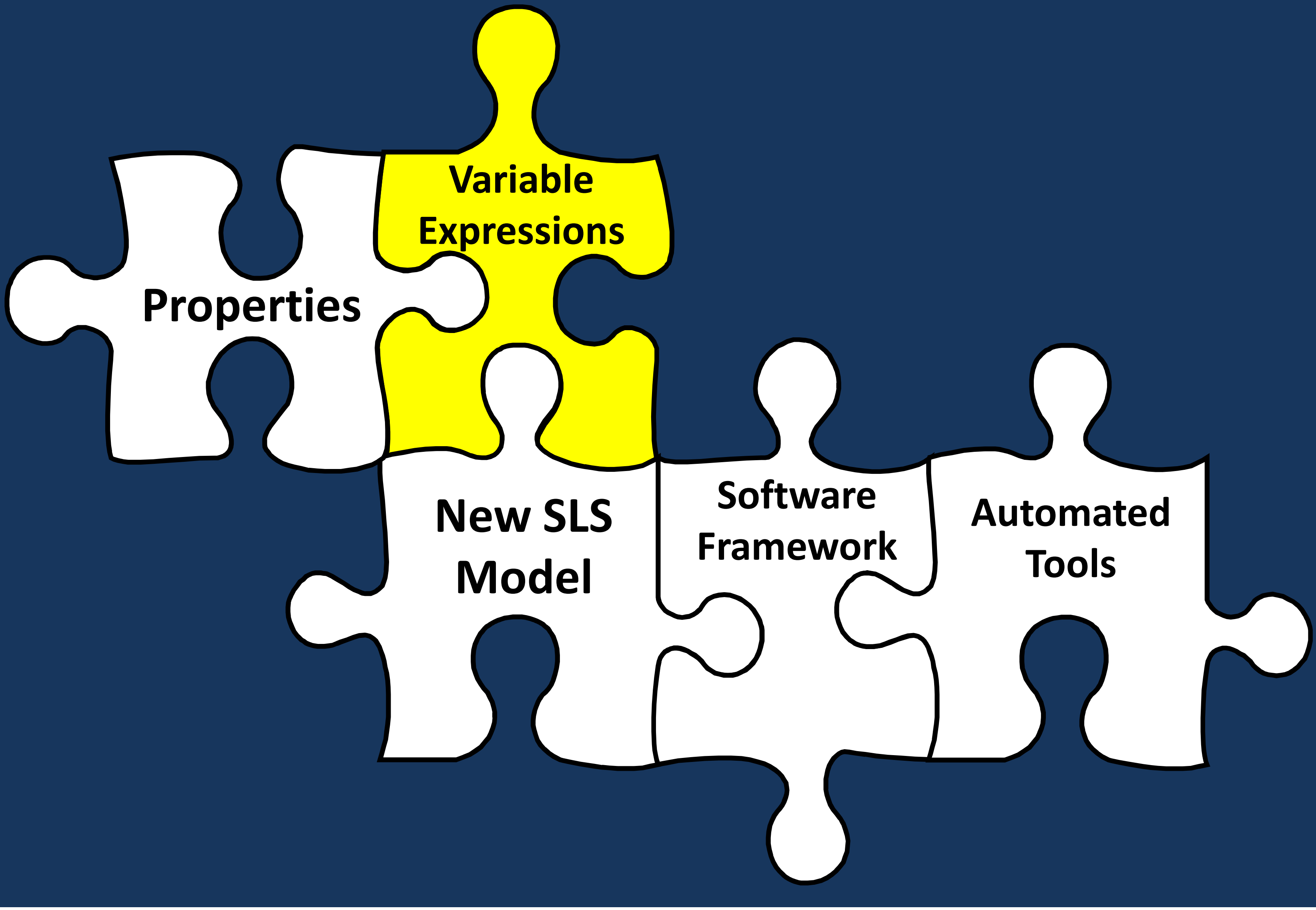
  age = # of steps since x was flipped [TABU, Glover 1986]

  flips = # of times x has been flipped [HSAT, Gent & Walsh 1992]

- Static Properties

- Potential for New Properties

Overview


Variable Expressions
Properties
New SLS Model
Software Framework
Automated Tools

Variable Expressions (VEs)

- combinations of variable *properties* in mathematical expressions:

  break
  make – break
  (make – break) + 3 $\cdot \log_2$(age) + age/flips

- Most existing SLS algorithms use straightforward VEs
  … we explore more complex VEs

VW2: The Origin of VEs

- Variable Weighting Algorithm VW2 [Prestwich, 2005]

- New Property: w(eight) *(initialized to zero)*

  updated after a flip: $\mathrm{w} := (1-s)\cdot(\mathrm{w}+1) + s\cdot(\text{step \#})$

- New Scoring VE: break + $c\cdot(\mathrm{w} - \mathrm{w}_{\mathrm{avg}})$

Our Interest in VW2

- CBMC software verification instances
  - Challenging for SLS, but
    easy for most DPLL-based solvers
  - VW2 was the best-performing SLS algorithm

    (Optimal performance with no smoothing)

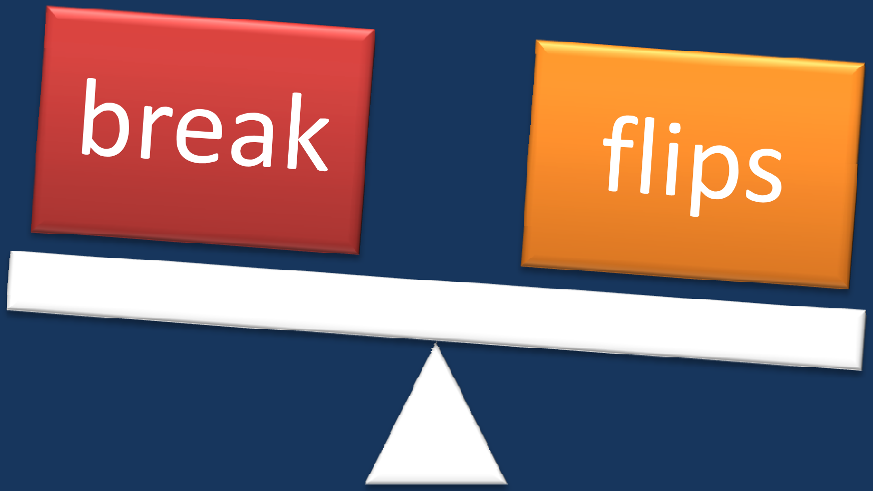
- With no smoothing, the VE becomes:

$$\text{break} + c \cdot \text{flips}$$

Combining Properties

Select variable with minimum value of:

break + c·flips



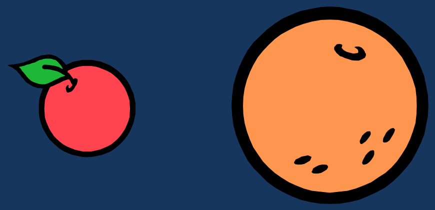
Combining Properties

Select variable with minimum value of:

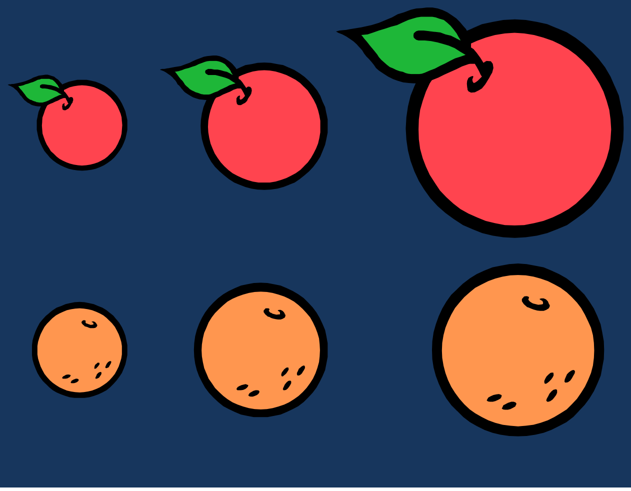
break + $c\cdot$flips

Combining Properties

- Normalize properties values to [0..1] *amongst the "candidate" variables*



- Allow for non-linear normalization



More Normalization

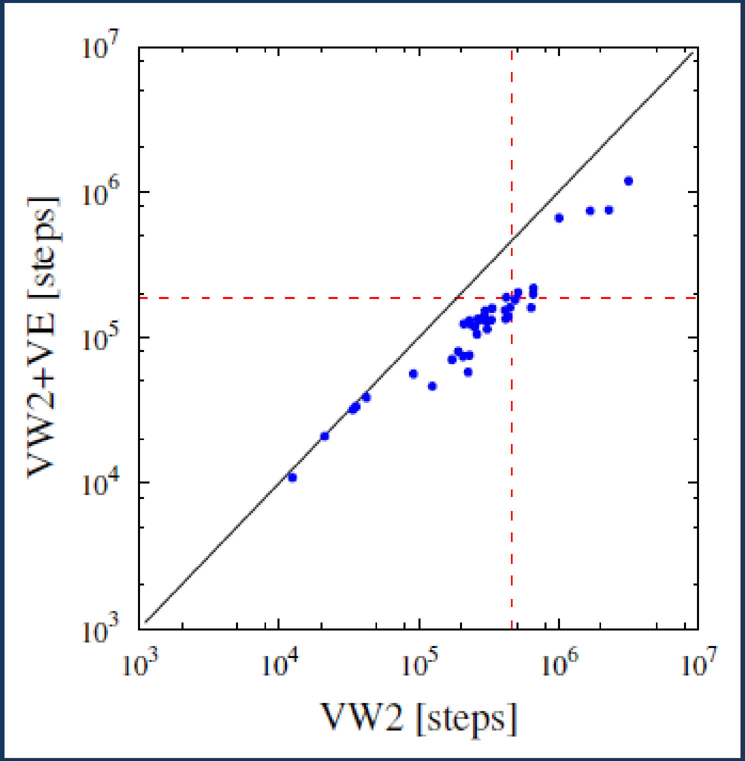
- Consider traditional (make – break)…
  - what about ($c_1 \cdot$ make – $c_2 \cdot$ break) ??
  - *Relative* number of clauses (relMake & relBreak)

- more $\|$normalization$\|$ discussed in paper

Modifying Existing Algorithms with VEs

- VW2:
  $\text{break} + c \cdot \text{flips} \rightarrow \|\text{break}\| + c \cdot \|\text{flips}\|^{a}$

- Speedup factor:
  2.5 (steps)
  2.1 (time)



Modifying Existing Algorithms with VEs

- WalkSAT:

  $c_1 \cdot \|\text{make}\|^{a1} + c_2 \cdot \|\text{relMake}\|^{a2} + c_3 \cdot \|\text{break}\|^{a3} + c_4 \cdot \|\text{relBreak}\|^{a4}$

- Speedup factor:

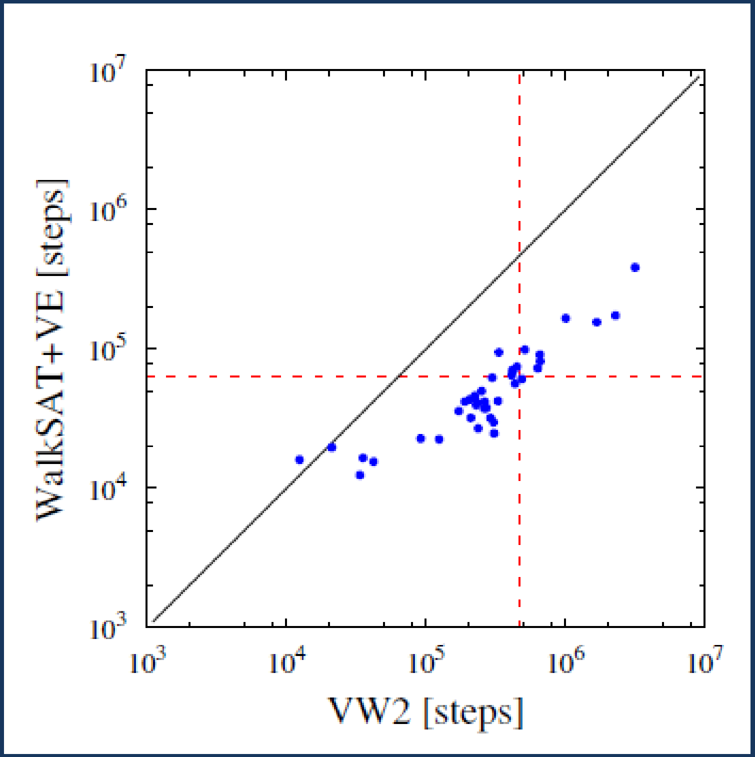
  7.2 (steps)

  3.1 (time)

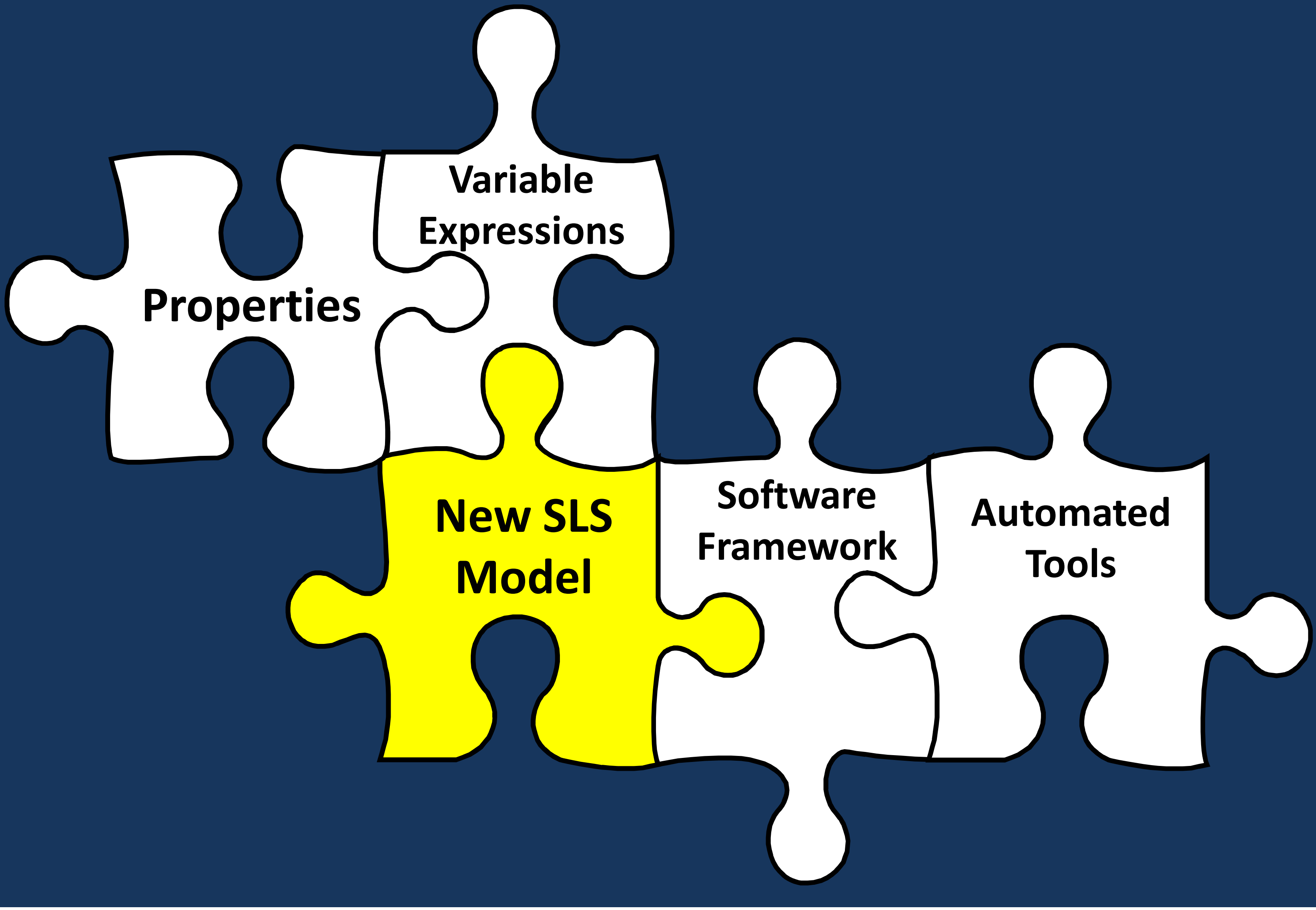
- (compared to original WalkSAT)

  > 4000 (steps)

  > 2000 (time)



Overview



Variable Expressions

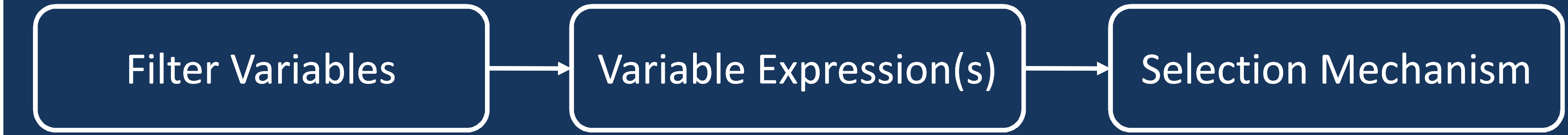
Properties

New SLS Model

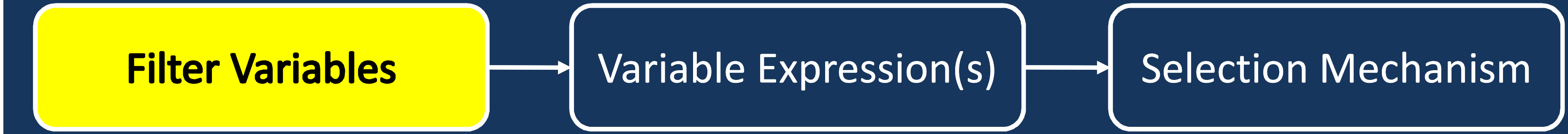
Software Framework

Automated Tools

Our New SLS Model



Variable Filters



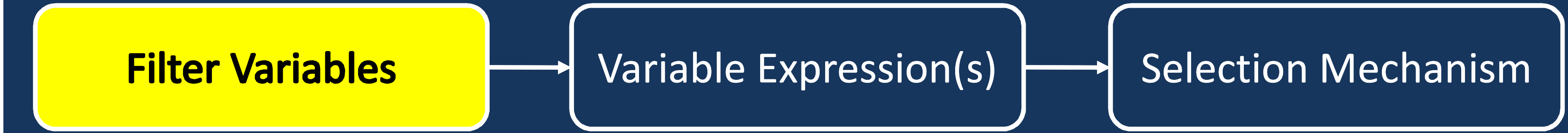
Filter Variables

Variable Expression(s)

Selection Mechanism

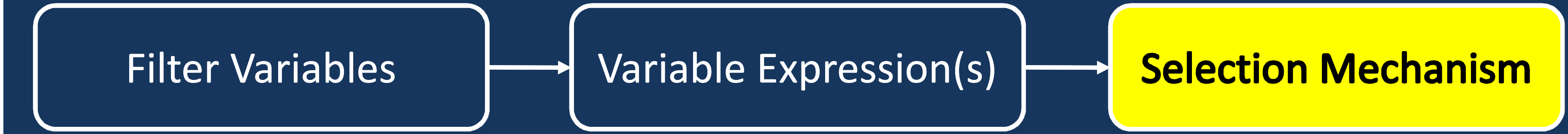
Variable Filters

- Popular Filters:
  - All Variables (GSAT, Selman *et al.*, 1992)
  - Select a random clause (WalkSAT, Selman *et al.*, 1994)
  - Promising variables (G^2WSAT, Li & Huang, 2005)



- Potential for new filters: (examples)
  - Oldest unsatisfied clause
  - Most frequently unsatisfied clause
  - Two random unsatisfied clauses
  - 5 oldest variables

Our New SLS Model



Filter Variables

Variable Expression(s)

Selection Mechanism

Separation of:
VEs & Selection Mechanism

- Novelty Algorithm [McAllester, Selman & Kautz, 1997]

- Select "best" variable with maximum of:

  (make – break)

  breaking ties by

  (age)

- If the best variable has the minimum

  (age)

  then, with probability p, select 2nd best var.

Separation of: VEs & Selection Mechanism

- Novelty Algorithm [McAllester, Selman & Kautz, 1997]

- Select “best” variable with maximum of:

    (VE_1)

  breaking ties by

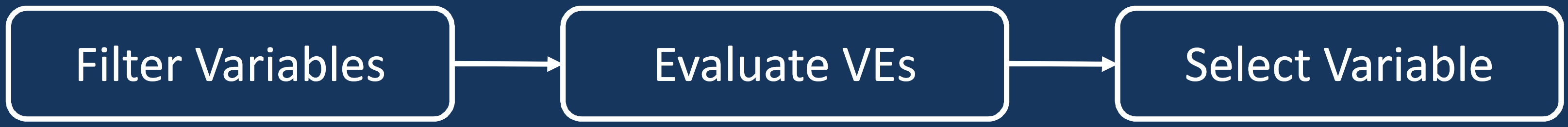
    (VE_2)

- If the best variable has the minimum

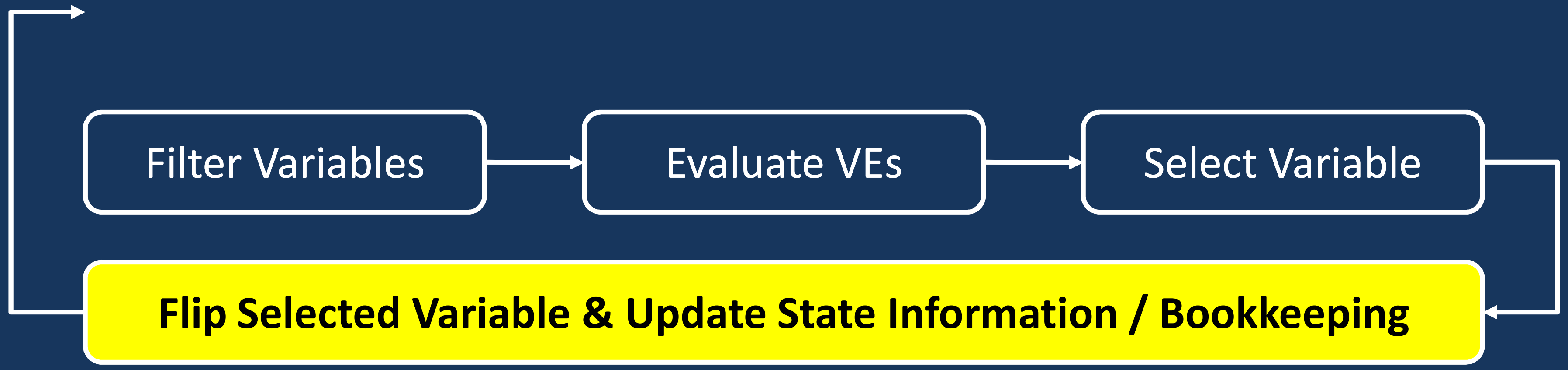
    (VE_3)

  then, with probability p, select 2nd best var.

Our New SLS Model



Our New SLS Model



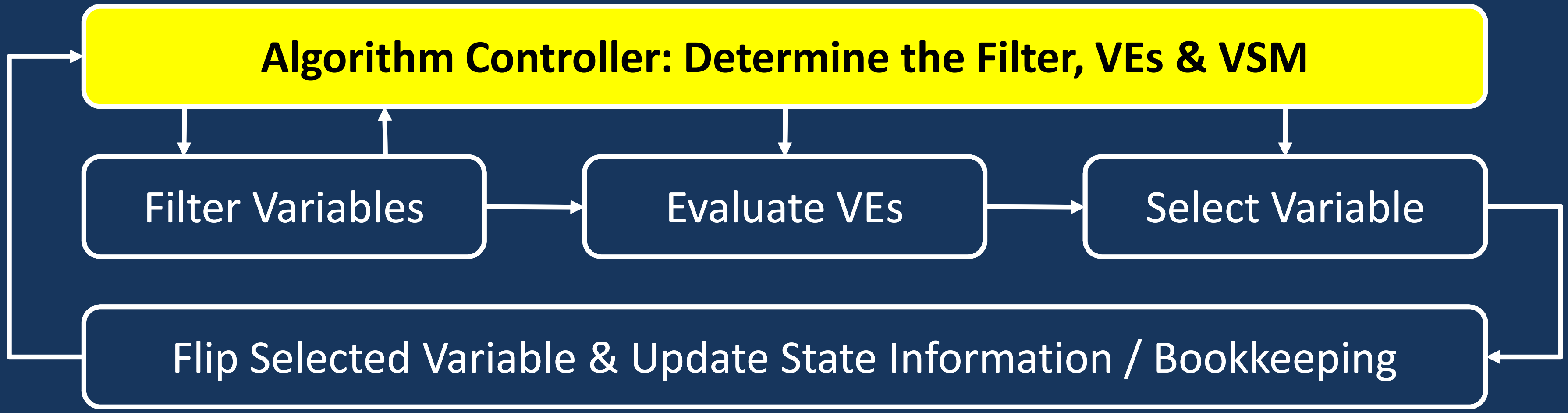
Filter Variables

Evaluate VEs

Select Variable

Flip Selected Variable & Update State Information / Bookkeeping

Our New SLS Model



Algorithm Controller: Determine the Filter, VEs & VSM

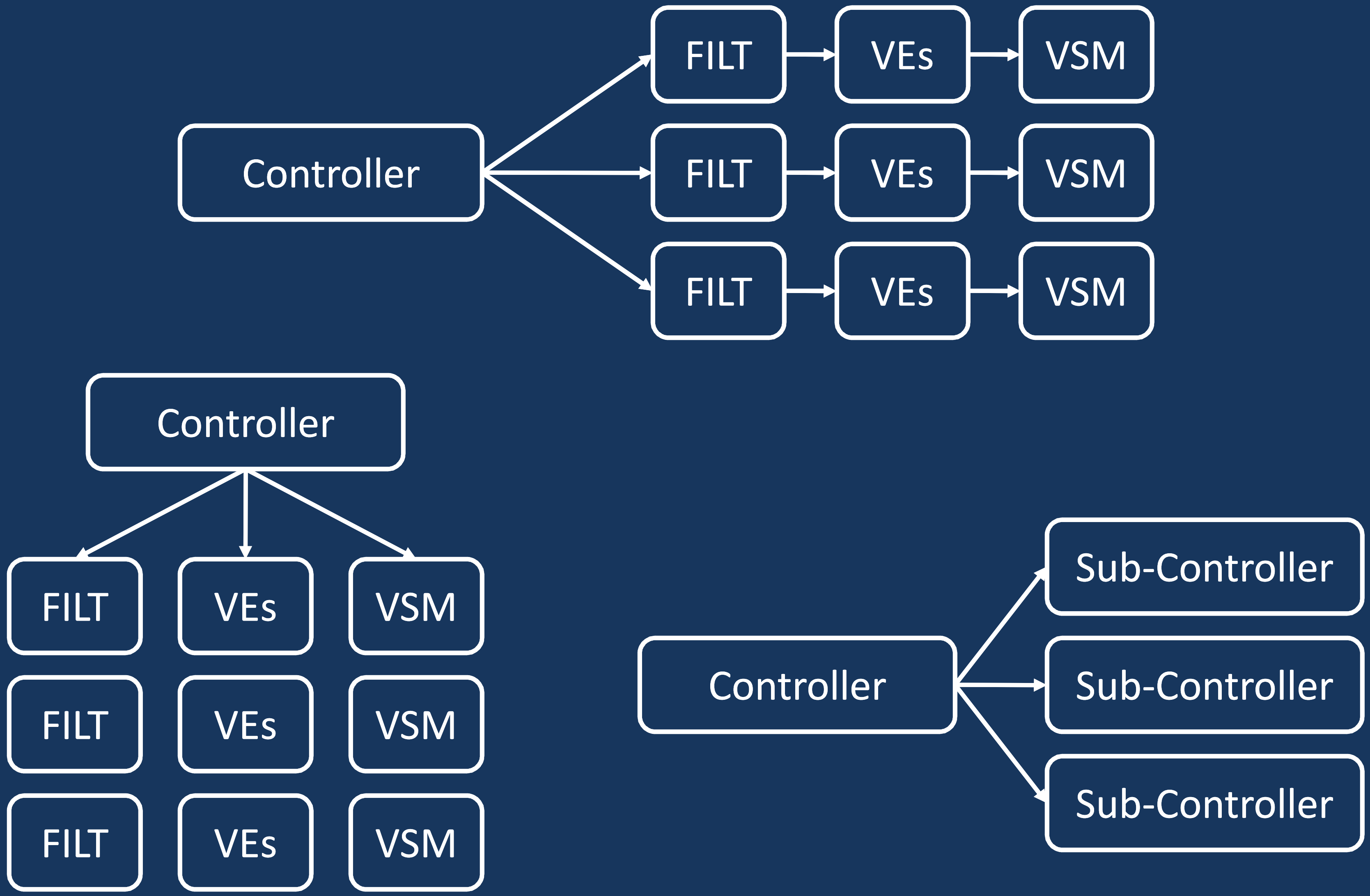
Filter Variables

Evaluate VEs

Select Variable

Flip Selected Variable & Update State Information / Bookkeeping

Algorithm Controllers



Controller

FILT VEs VSM

FILT VEs VSM

FILT VEs VSM

Controller

FILT VEs VSM

FILT VEs VSM

FILT VEs VSM

Controller

Sub-Controller

Sub-Controller

Sub-Controller

Algorithm Controllers

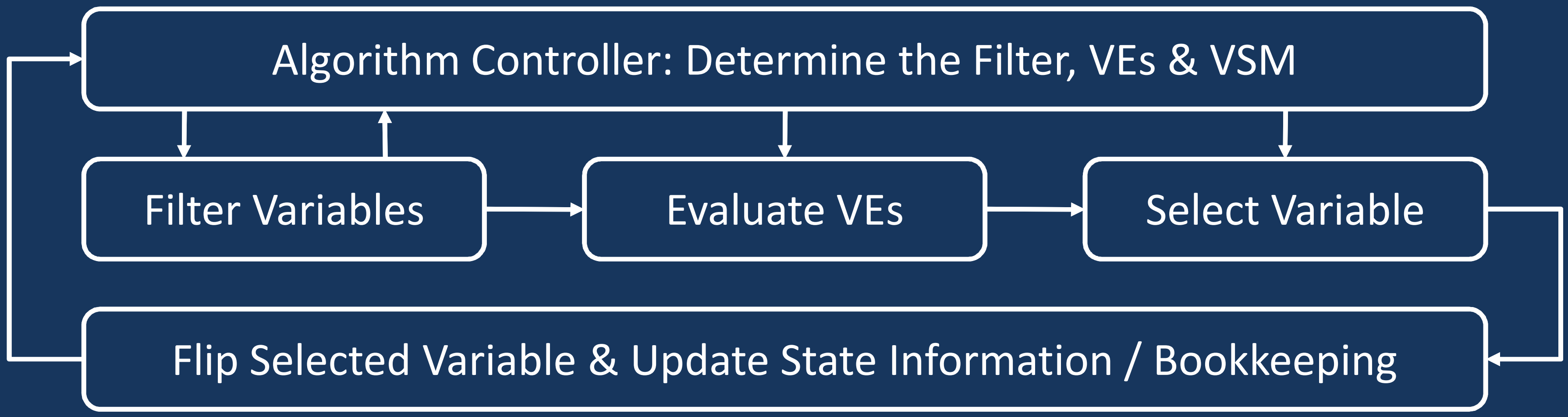
- Randomized hybrids:

  20%: algorithm A
  70%: algorithm B
  10%: filter from A, selection mechanism from B

- “Smart” controllers:

  If condition 1 is true, use algorithm C,
  if condition 2 is true, use algorithm D,
  otherwise, use the above randomized algorithm

Our New SLS Model



Algorithm Controller: Determine the Filter, VEs & VSM

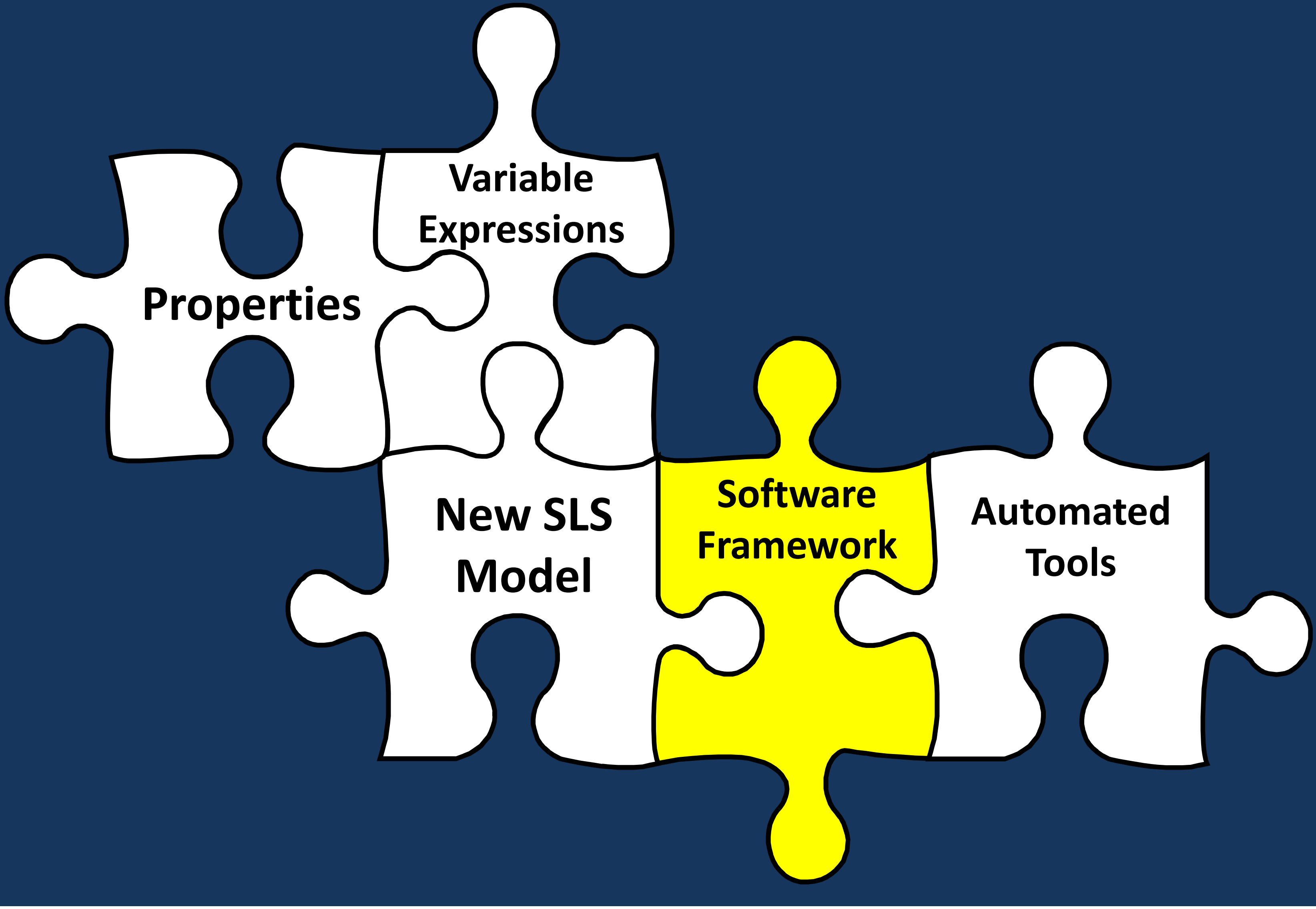
Filter Variables

Evaluate VEs

Select Variable

Flip Selected Variable & Update State Information / Bookkeeping

Overview



Variable Expressions

Properties

New SLS Model

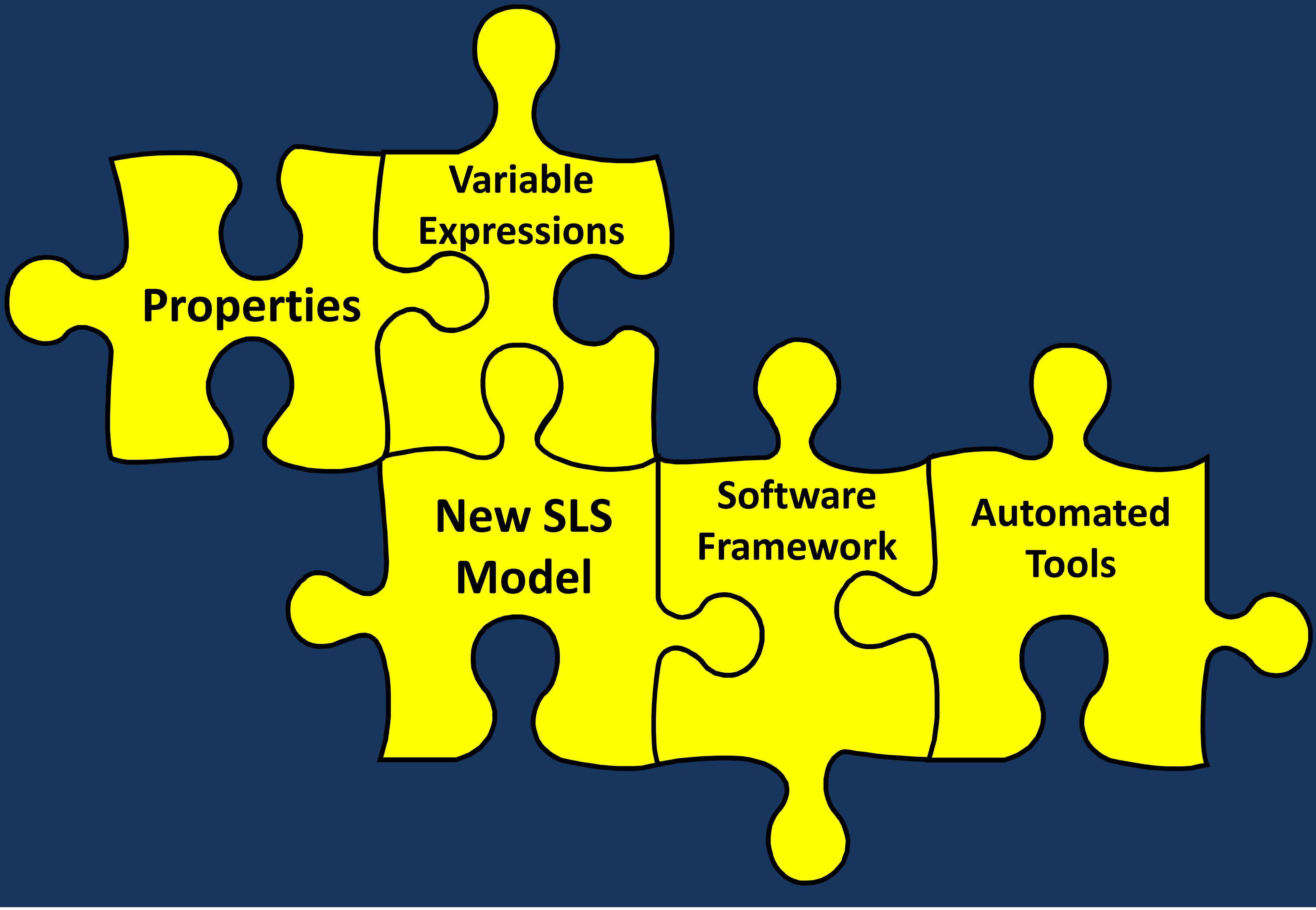
Software Framework

Automated Tools

Software Implementation

- Design Architecture for Variable Expressions (DAVE)
  - Entire algorithm specified at runtime
    - Controllers, filters, VEs, selection mechanisms
  - Arbitrary complex VEs (interpreted)
  - Sophisticated macro system
    - Aids the use of automated configurators
- Extension of UBCSAT (2.0)

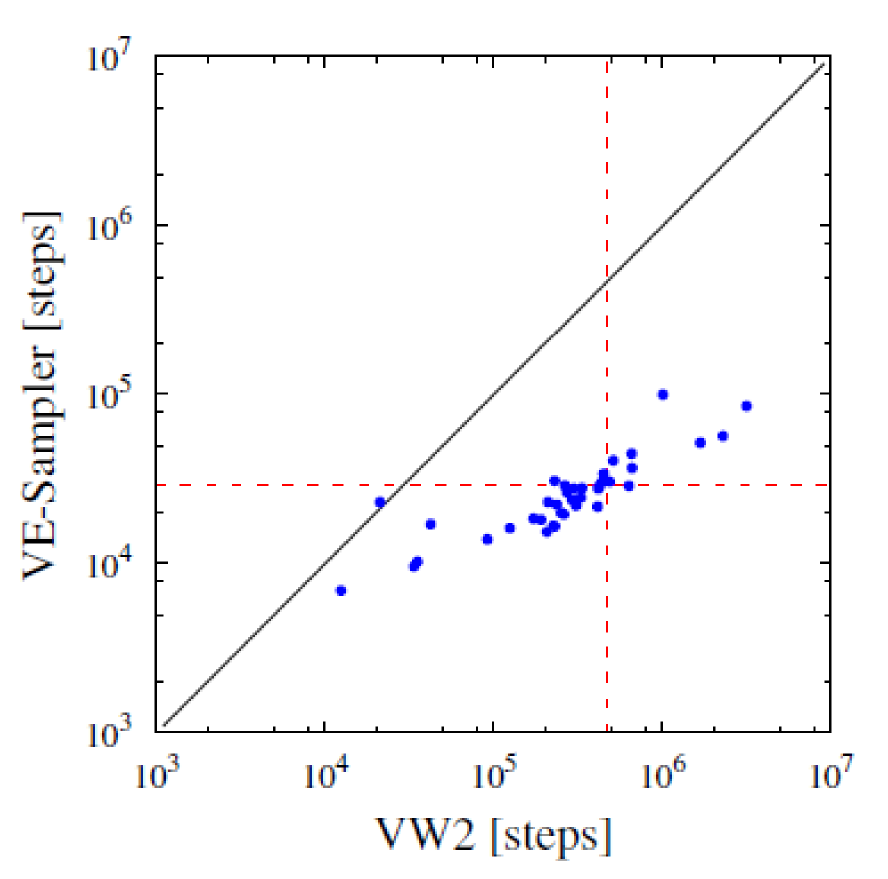
Our Methods in Practice



Our Methods in Practice: VE-Sampler

- Randomized algorithm controller
  - Selects between 5 different VEs in the form: $\|\mathrm{p}_1\|^{a1} + c \cdot \|\mathrm{p}_2\|^{a2}$
- Properties p_1 and p_2 are configurable from amongst 21 different properties (& property ratios)
- Automated configurator (ParamILS)
  - Over 10^{50} possible configurations

VE Sampler Results



- Speedup factor:
  16.2 (steps)
  9.0 (time)

Future Work

- New variable properties
- New VE constructions
  - Better normalizations
- New selection mechanisms
- New algorithm controllers

- Continue to use automated tools to test and evaluate all of the all of the above

Key Contributions

- New conceptual model for SLS
  - Separate flters, VEs & selection mechanisms
  - Algorithm controllers for robust hybrid algorithms
- Variable Expressions (VEs)
  - Complex combinations of variable properties
  - Advanced normalization methods
- Design Architecture for VEs (DAVE)
  - Very flexible
  - Well suited for automated tools