Tackling Random Blind Spots with Strategy-Driven Generation
Matthew Ballance
matt_ballance@mentor.com

Engineer-Directed Tests
- Verify known cases
- Targeted
- Functional Coverage Tracked

Open-Loop Random Tests
- Scatter-shot
- Find lurking bugs
- Generate un-envisioned cases
- Get to edges of constraint space

Open-Loop Random Challenges
- No metrics
- Random resistant corner cases

Unconstrained Random: Even Value Distribution

```
class unconstrained;
  rand bit[3:0] A;
  rand bit[3:0] B;
endclass
```

Key cases likely to be missed!

Unconstrained Random: Uneven Value Distribution

```
class unconstrained;
  rand bit[3:0] A;
  rand bit[3:0] B;
  constraint c {
    if (A > 1) {
      B == 0;
    }
  }
endclass
```


Strategy-Driven Generation
- Automatically hits random-resistant cases
- Identifies high-value tests from constraints
- Identifies key value / variable combinations
- Targets selected values during simulation

Pattern-Based Target Value Selection
- Spreads stimulus across reachable space
- Uniform Ranges
  - Divides reachable domain into N ranges
  - Spreads values across reachable domain
- Edge Ranges
  - Place single-value bins at min/max
  - Min/max values are often corner cases

Constraint-Guided Target Value Selection
- Ensures constraint branches are exercised
- Condition-inferred target values
  - Equality expression in if/implication
  - Inside expression in if/implication
- Body Constraint-inferred target values
  - Equality expression in if/implication body
  - Inside expression in if/implication body

Constraint-Guided Combination Strategy
- Targets constraint-related variables
- Works well with high constraint density
- Less useful with low constraint density

Individual Variable Combination Strategy
- Independently target variables
- Interactions occur randomly
- Enables targeting more value ranges

All-Pairs Variable Combination Strategy
- Approach from the software test domain
- Selects pairs, triples, quads of variables
- Case study data suggests
  - 70% bugs triggered with pairs
  - No bugs required 6 or more to trigger

Proof of Concept Implementation
- Uses an intelligent testbench automation solver
- Constraint-based stimulus description
- Goal-based value generation

Process
- Import SV class fields/constraints
- Analyze variables for reachability
- Identifies strategy-specific goals
- Creates stimulus generator class
- Creates strategy-specific SV covergroup

Results
- Compare Random and Strategy-Driven
  - Random progress stops at 85%
  - Strategy-driven easily hits all cases

Engineer-Directed Tests
- Verify known cases
- Targeted
- Functional Coverage Tracked

Open-Loop Random Tests
- Scatter-shot
- Find lurking bugs
- Generate un-envisioned cases
- Get to edges of constraint space

Open-Loop Random Challenges
- No metrics
- Random resistant corner cases

Unconstrained Random: Even Value Distribution

```
class unconstrained;
  rand bit[3:0] A;
  rand bit[3:0] B;
endclass
```

Key cases likely to be missed!

Unconstrained Random: Uneven Value Distribution

```
class unconstrained;
  rand bit[3:0] A;
  rand bit[3:0] B;
  constraint c {
    if (A > 1) {
      B == 0;
    }
  }
endclass
```


Strategy-Driven Generation
- Automatically hits random-resistant cases
- Identifies high-value tests from constraints
- Identifies key value / variable combinations
- Targets selected values during simulation

Pattern-Based Target Value Selection
- Spreads stimulus across reachable space
- Uniform Ranges
  - Divides reachable domain into N ranges
  - Spreads values across reachable domain
- Edge Ranges
  - Place single-value bins at min/max
  - Min/max values are often corner cases

Constraint-Guided Target Value Selection
- Ensures constraint branches are exercised
- Condition-inferred target values
  - Equality expression in if/implication
  - Inside expression in if/implication
- Body Constraint-inferred target values
  - Equality expression in if/implication body
  - Inside expression in if/implication body

Constraint-Guided Combination Strategy
- Targets constraint-related variables
- Works well with high constraint density
- Less useful with low constraint density

Individual Variable Combination Strategy
- Independently target variables
- Interactions occur randomly
- Enables targeting more value ranges

All-Pairs Variable Combination Strategy
- Approach from the software test domain
- Selects pairs, triples, quads of variables
- Case study data suggests
  - 70% bugs triggered with pairs
  - No bugs required 6 or more to trigger

Proof of Concept Implementation
- Uses an intelligent testbench automation solver
- Constraint-based stimulus description
- Goal-based value generation

Process
- Import SV class fields/constraints
- Analyze variables for reachability
- Identifies strategy-specific goals
- Creates stimulus generator class
- Creates strategy-specific SV covergroup

Results
- Compare Random and Strategy-Driven
  - Random progress stops at 85%
  - Strategy-driven easily hits all cases

Engineer-Directed Tests
- Verify known cases
- Targeted
- Functional Coverage Tracked

Open-Loop Random Tests
- Scatter-shot
- Find lurking bugs
- Generate un-envisioned cases
- Get to edges of constraint space

Open-Loop Random Challenges
- No metrics
- Random resistant corner cases

Unconstrained Random: Even Value Distribution

```
class unconstrained;
  rand bit[3:0] A;
  rand bit[3:0] B;
endclass
```

Key cases likely to be missed!

Unconstrained Random: Uneven Value Distribution

```
class unconstrained;
  rand bit[3:0] A;
  rand bit[3:0] B;
  constraint c {
    if (A > 1) {
      B == 0;
    }
  }
endclass
```


Strategy-Driven Generation
- Automatically hits random-resistant cases
- Identifies high-value tests from constraints
- Identifies key value / variable combinations
- Targets selected values during simulation

Pattern-Based Target Value Selection
- Spreads stimulus across reachable space
- Uniform Ranges
  - Divides reachable domain into N ranges
  - Spreads values across reachable domain
- Edge Ranges
  - Place single-value bins at min/max
  - Min/max values are often corner cases

Constraint-Guided Target Value Selection
- Ensures constraint branches are exercised
- Condition-inferred target values
  - Equality expression in if/implication
  - Inside expression in if/implication
- Body Constraint-inferred target values
  - Equality expression in if/implication body
  - Inside expression in if/implication body

Constraint-Guided Combination Strategy
- Targets constraint-related variables
- Works well with high constraint density
- Less useful with low constraint density

Individual Variable Combination Strategy
- Independently target variables
- Interactions occur randomly
- Enables targeting more value ranges

All-Pairs Variable Combination Strategy
- Approach from the software test domain
- Selects pairs, triples, quads of variables
- Case study data suggests
  - 70% bugs triggered with pairs
  - No bugs required 6 or more to trigger

Proof of Concept Implementation
- Uses an intelligent testbench automation solver
- Constraint-based stimulus description
- Goal-based value generation

Process
- Import SV class fields/constraints
- Analyze variables for reachability
- Identifies strategy-specific goals
- Creates stimulus generator class
- Creates strategy-specific SV covergroup

Results
- Compare Random and Strategy-Driven
  - Random progress stops at 85%
  - Strategy-driven easily hits all cases

Engineer-Directed Tests
- Verify known cases
- Targeted
- Functional Coverage Tracked

Open-Loop Random Tests
- Scatter-shot
- Find lurking bugs
- Generate un-envisioned cases
- Get to edges of constraint space

Open-Loop Random Challenges
- No metrics
- Random resistant corner cases

Unconstrained Random: Even Value Distribution

```
class unconstrained;
  rand bit[3:0] A;
  rand bit[3:0] B;
endclass
```

Key cases likely to be missed!

Unconstrained Random: Uneven Value Distribution

```
class unconstrained;
  rand bit[3:0] A;
  rand bit[3:0] B;
  constraint c {
    if (A > 1) {
      B == 0;
    }
  }
endclass
```


Strategy-Driven Generation
- Automatically hits random-resistant cases
- Identifies high-value tests from constraints
- Identifies key value / variable combinations
- Targets selected values during simulation

Pattern-Based Target Value Selection
- Spreads stimulus across reachable space
- Uniform Ranges
  - Divides reachable domain into N ranges
  - Spreads values across reachable domain
- Edge Ranges
  - Place single-value bins at min/max
  - Min/max values are often corner cases

Constraint-Guided Target Value Selection
- Ensures constraint branches are exercised
- Condition-inferred target values
  - Equality expression in if/implication
  - Inside expression in if/implication
- Body Constraint-inferred target values
  - Equality expression in if/implication body
  - Inside expression in if/implication body

Constraint-Guided Combination Strategy
- Targets constraint-related variables
- Works well with high constraint density
- Less useful with low constraint density

Individual Variable Combination Strategy
- Independently target variables
- Interactions occur randomly
- Enables targeting more value ranges

All-Pairs Variable Combination Strategy
- Approach from the software test domain
- Selects pairs, triples, quads of variables
- Case study data suggests
  - 70% bugs triggered with pairs
  - No bugs required 6 or more to trigger

Proof of Concept Implementation
- Uses an intelligent testbench automation solver
- Constraint-based stimulus description
- Goal-based value generation

Process
- Import SV class fields/constraints
- Analyze variables for reachability
- Identifies strategy-specific goals
- Creates stimulus generator class
- Creates strategy-specific SV covergroup

Results
- Compare Random and Strategy-Driven
  - Random progress stops at 85%
  - Strategy-driven easily hits all cases