On Computing Minimal Independent Support and Its Applications to Sampling and Counting

Alexander Ivrii[1], Sharad Malik[2], **Kuldeep S. Meel**[3], Moshe Y. Vardi[3]

[1] IBM Research Haifa

[2] Princeton University

[3] Rice University

(Author names are ordered alphabetically by last name)

SAT Sampling and Counting

- Given a propositional logic formula F in CNF
- Generate satisfying assignments uniformly at random
  - Uniform over the space of satisfying assignments
- Count the number of satisfying assignments

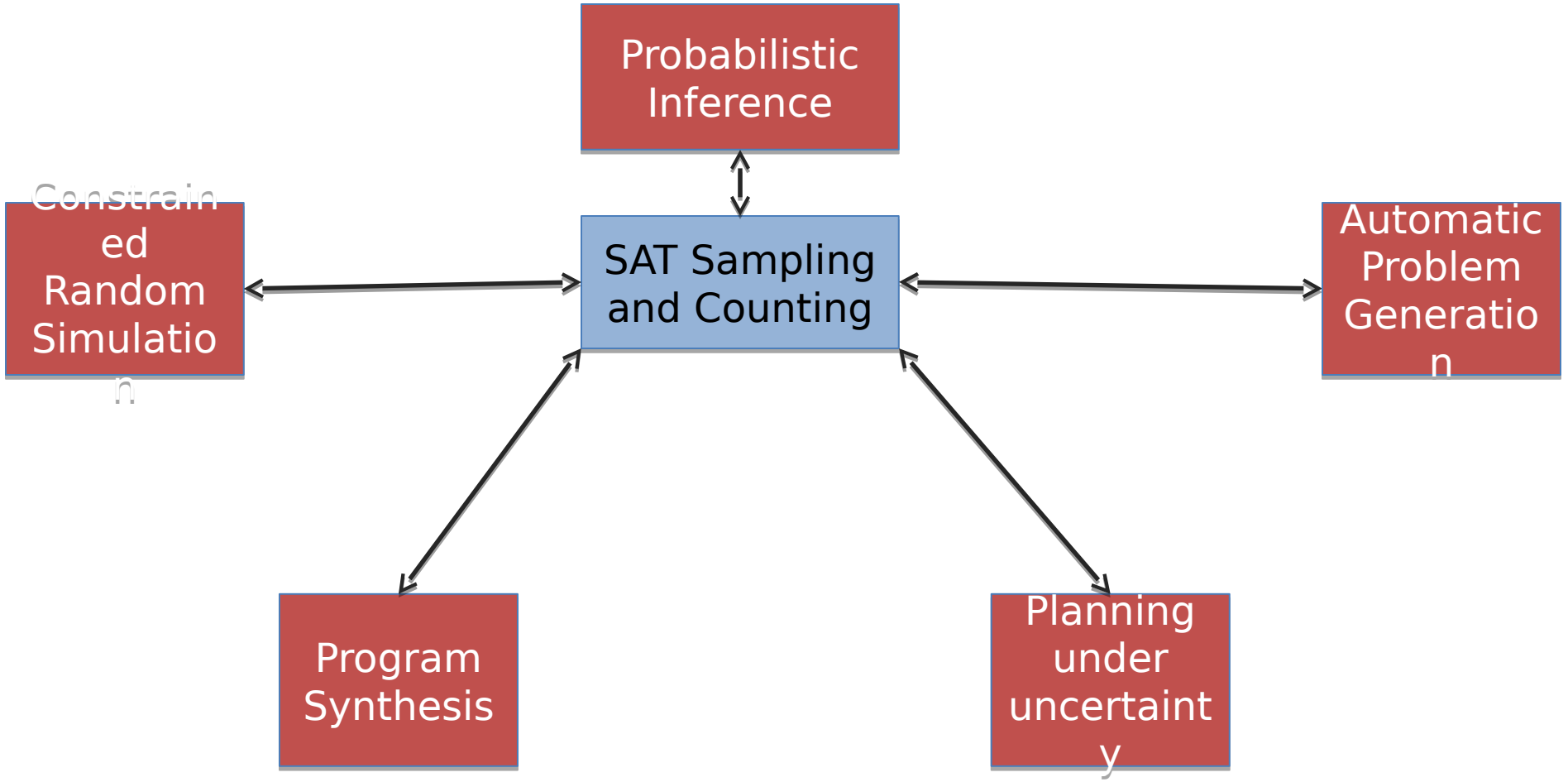
SAT Sampling

- F = (a V b)
- R_F: {(0,1), (1,0), (1,1)}
- G: A uniform generator
- Pr [G(F) = (0,1)] = Pr [G (F) = (1,1)] = …. = 1/3

SAT Counting

- F = (a V b)
- R_F: {(0,1), (1,0), (1,1)}
- #F: 3
- #P: The class of counting problems for decision problems in NP
  - #P-complete (Valiant 1979)

Diverse Applications

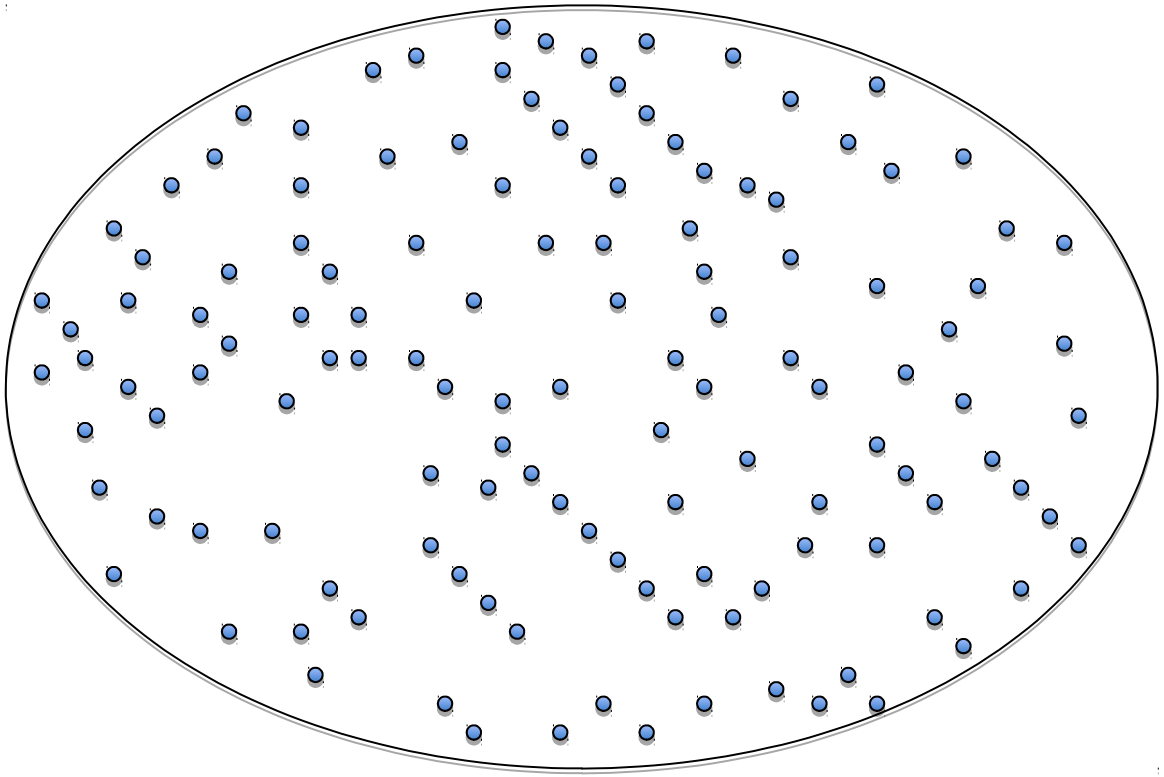


SAT Sampling and Counting: Prior Work

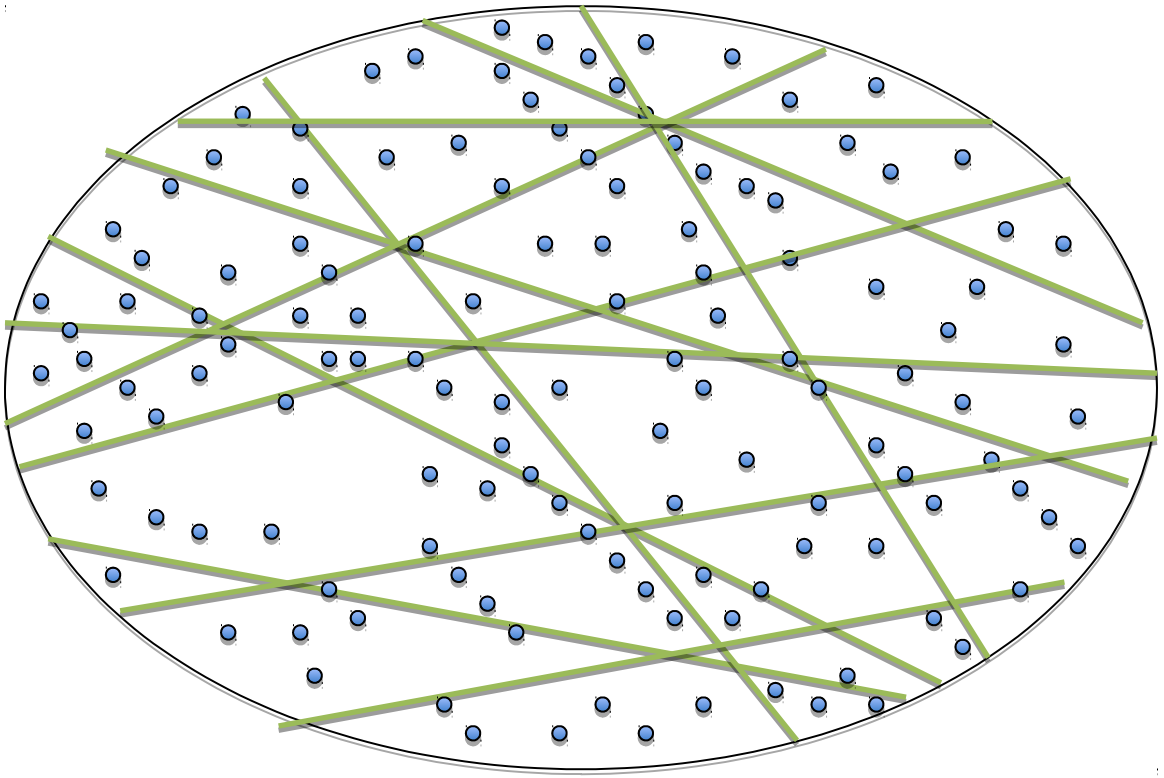
- Prior work (before 2013) either failed to scale or provided very weak guarantees
- (Since 2013) Recent Hashing-based approach to approximate variants of sampling and counting
  - Strong theoretical guarantees
  - Scales to large instances

[Chakraborty et al 13a,13b,14,15, Chakraborty et al. 14, Ermon et al. 13a,14,15, Belle 15, Chistikov 15 and others]

Core Idea

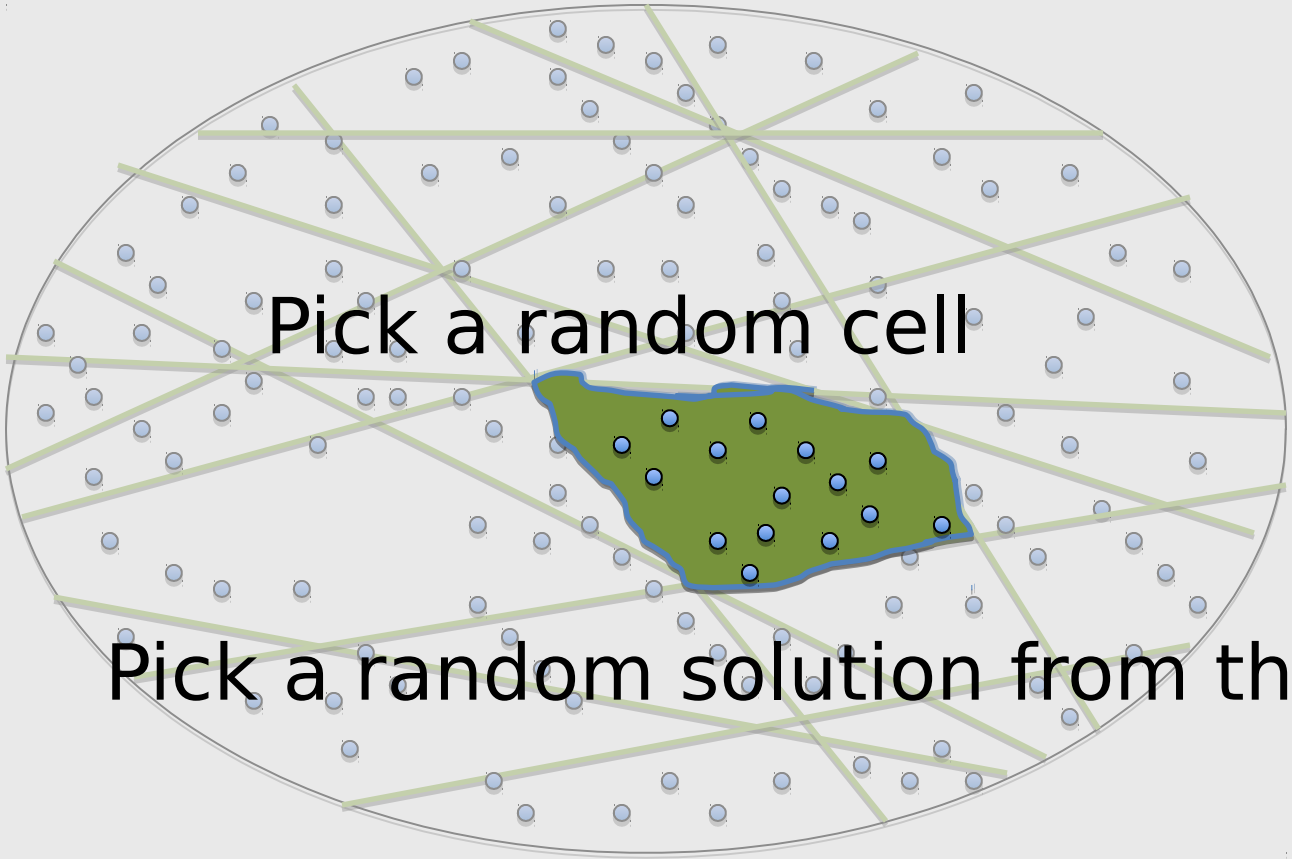


Partitioning into cells

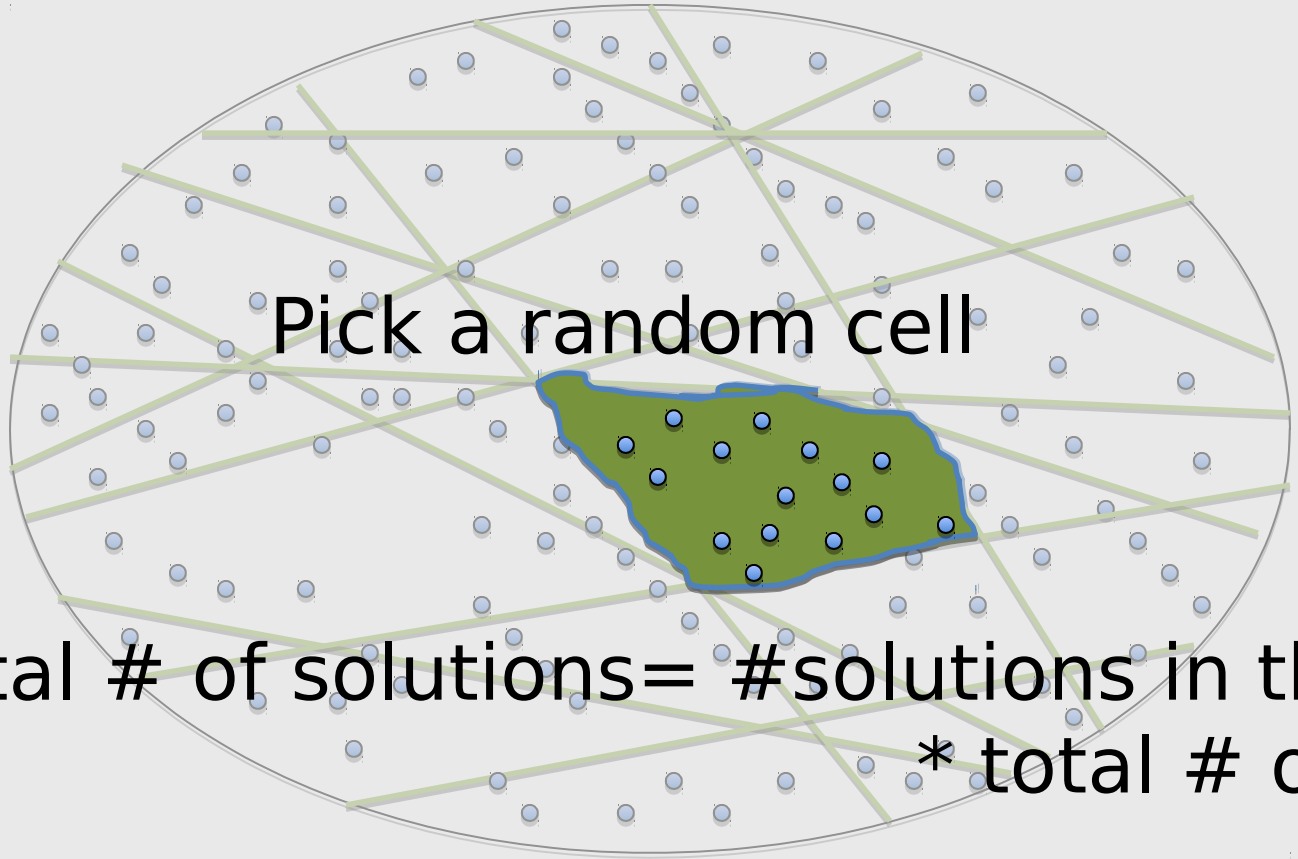


Cells should be roughly equal in size and small enough to enumerate completely

Uniform Sampling



Counting through Partitioning



Total # of solutions= #solutions in the cell
* total # of cells

How to Partition?

How to partition into roughly equal small cells of solutions without knowing the distribution of solutions?

Universal Hashing [Carter-Wegman 1979]

XOR-based Hashing

Variables: $X_1, X_2, X_3, \ldots . X_n$

Pick every variable with probability 1/2 : $X_1, X_3, X_4, \ldots . X_n$

XOR all the picked variables : $X_1 \oplus X_3 \oplus X_4 \oplus \ldots . \oplus X_n$

Equate to 0 or 1 with prob 1/2 : $X_1 \oplus X_3 \oplus X_4 \oplus \ldots . \oplus X_n = 0$

$$\left.\begin{array}{l} X_1 \oplus X_3 \oplus X_4 \oplus \ldots . \oplus X_n = 0 \\ X_2 \oplus X_3 \oplus X_7 \oplus \ldots . \oplus X_{n-2} = 1 \\ . \\ . \\ . \\ X_1 \oplus X_5 \oplus X_6 \oplus \ldots . \oplus X_n = 0 \end{array}\right\} m \text{ XOR constraints} \Rightarrow 2^m \text{ cells}$$

XOR-Based Hashing

- The cell: F ∧ {μ XOR constraints}
- **CryptoMiniSAT**: Efficient for CNF+XOR

- Avg Length : n/2

- Smaller the XORs, better the performance

How to shorten XOR clauses?

Independent Support

- Set of variables such that assignments to these uniquely determine assignments to rest of variables for formula to be true
- If σ_1 and σ_2 agree on I then $\sigma_1 = \sigma_2$
- c ⟷ (a V b) ; Independent Support (I): {a, b}
- Hash only on the independent variables

[Chakraborty et al. DAC 2014]

Independent Support

- Hash only on the Independent Support
- Average size of XOR: n/2 to I/2

- Ad-hoc (and often wrong) estimation of Independent support
- No procedure to determine Independent Support

Contributions

- MIS: The first algorithmic procedure to determine minimal Independent Support
- Scales to formulas with tens of thousands of variables
- Speeds up the state of the art sampling and counting techniques by 1-2 orders of magnitude

Key Idea

Input Formula: F, Solution space: R_F

$\forall \sigma_1, \sigma_2 \in R_F$, If σ_1 and σ_2 agree on I, then $\sigma_1 = \sigma_2$

$$F(x_1, \ldots, x_n) \wedge F(y_1, \ldots, y_n) \wedge \bigwedge_{i | x_i \in I} (x_i = y_i) \implies \bigwedge_{j} (x_j = y_j)$$

where $F(y_1, \ldots, y_n) = F(x_1 \to y_1, \ldots, x_n \to y_n)$

Key Idea

$$F(x_1,\ldots,x_n) \wedge F(y_1,\ldots,y_n) \wedge \bigwedge_{i|x_i \in I} (x_i = y_i) \implies \bigwedge_{j} (x_j = y_j)$$

$$Q_{F,I} = F(x_1,\ldots,x_n) \wedge F(y_1,\ldots,y_n) \wedge \bigwedge_{i|x_i \in I} (x_i = y_i) \wedge \neg \left(\bigwedge_{j} (x_j = y_j) \right)$$

Theorem: $Q_{F,I}$ is unsatisfiable if and only if I is independent support

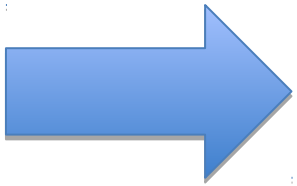
Key Idea

$$H_1 = \{x_1 = y_1\},\ \ldots,\ H_n = \{x_n = y_n\}$$

$$\Omega = F(x_1, \ldots, x_n) \wedge F(y_1, \ldots, y_n) \wedge (\neg \bigwedge_j (x_j = y_j))$$

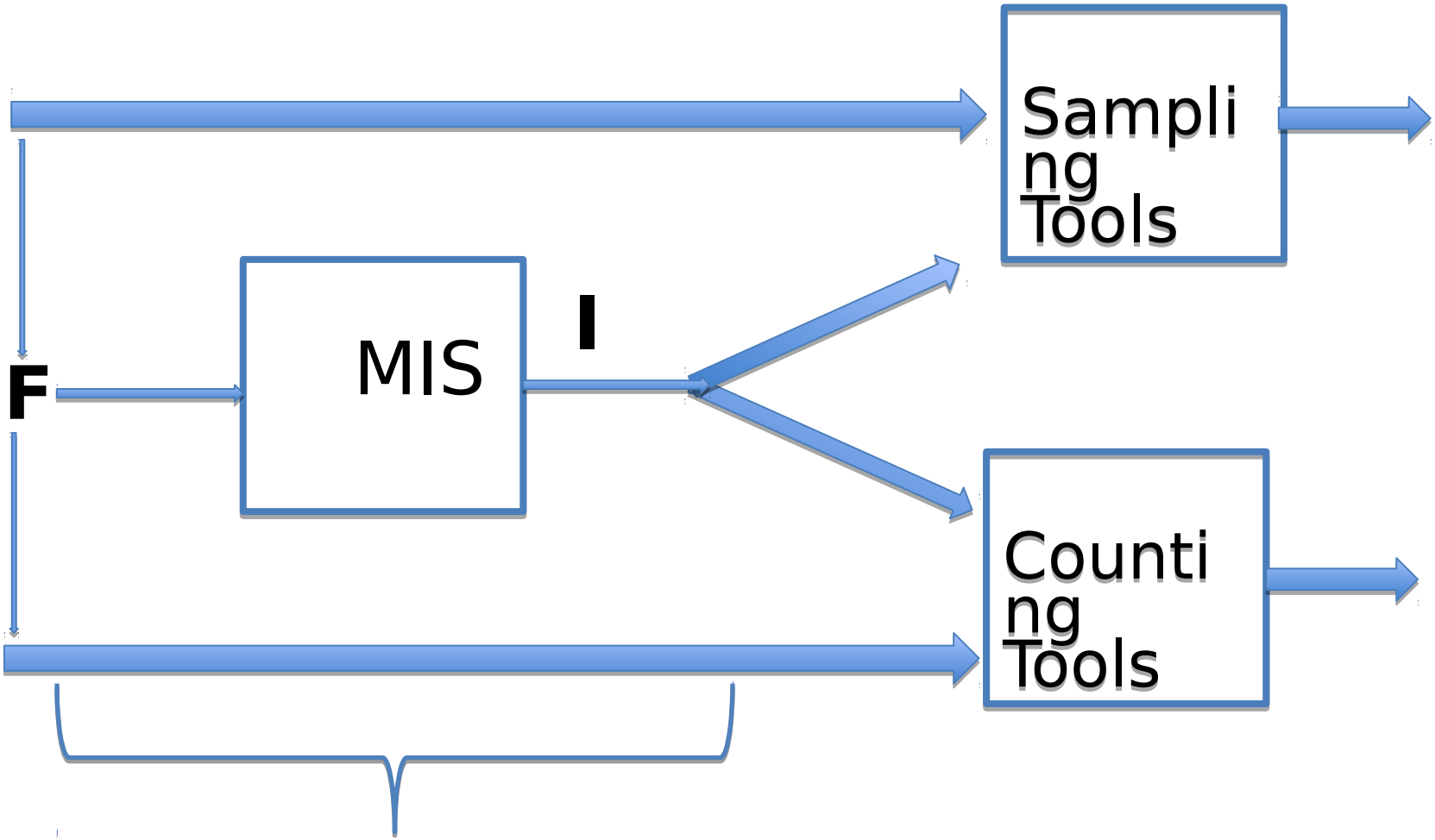
I is minimal independent support iff
$H^I \wedge \Omega$ is unsatisfiable where $H^I = \{H_i | x_i \in I\}$

Minimal Independent Support



Group-oriented Minimal Unsatisfiable subset

Impact on Sampling and Counting Techniques



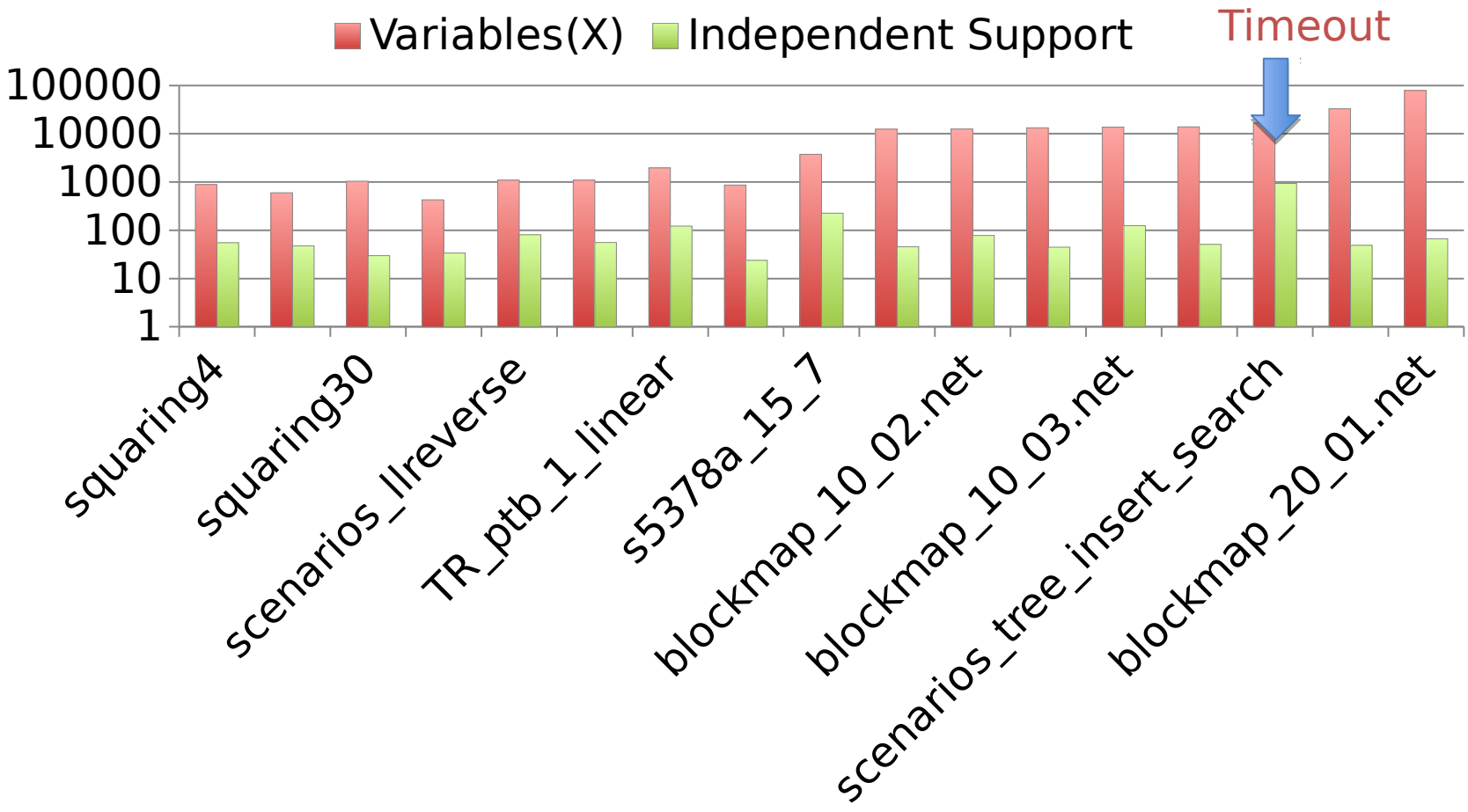
Anytime Algorithm

Experimental Results

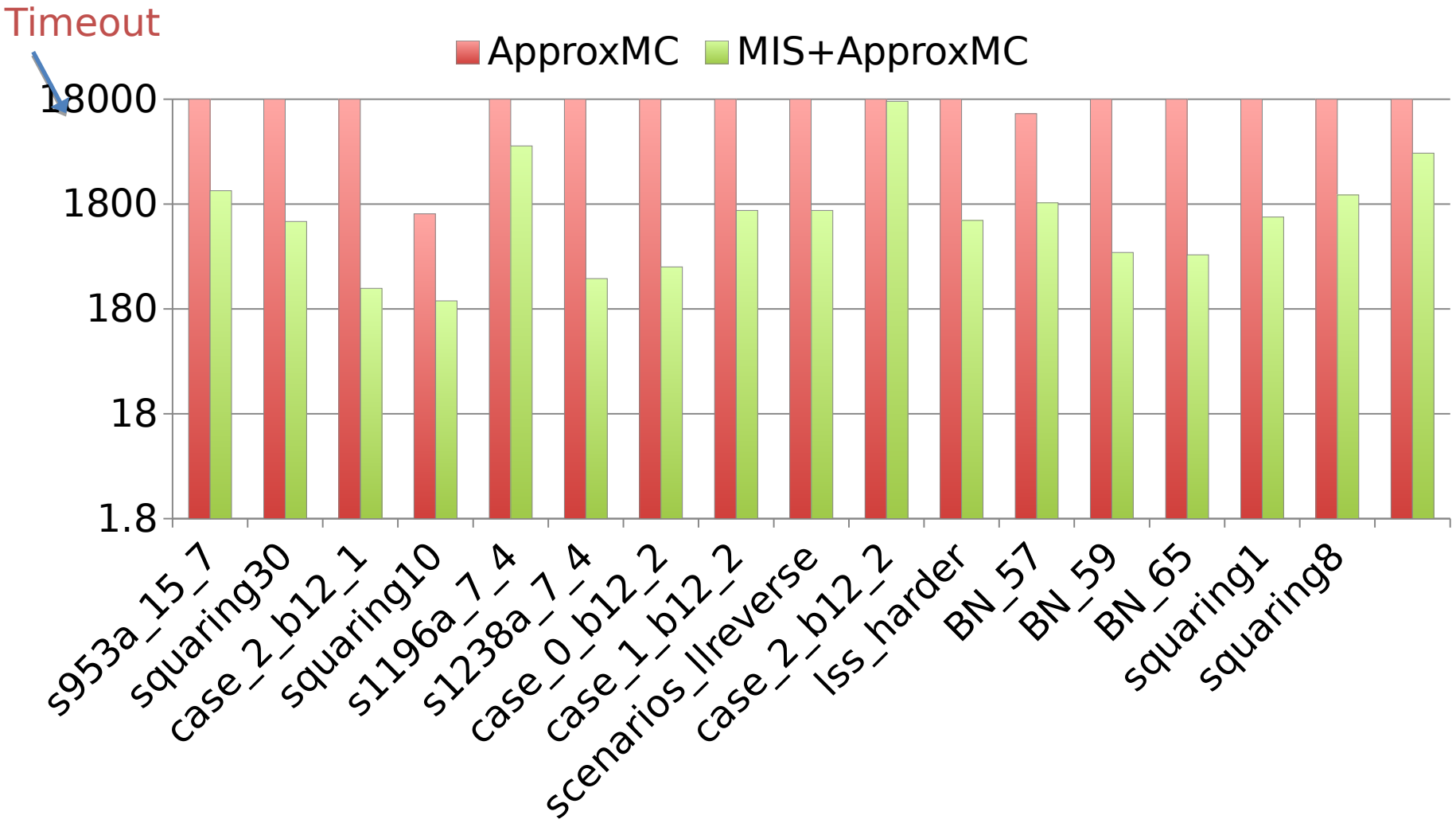
- Prototype implementation (MIS)
  - Employs Muser2 for MUS computation

- Experimented with over 200+ benchmarks to study impact on sampling and counting tools
  - UniGen2: Almost uniform sampler
  - ApproxMC: Approximate model counter

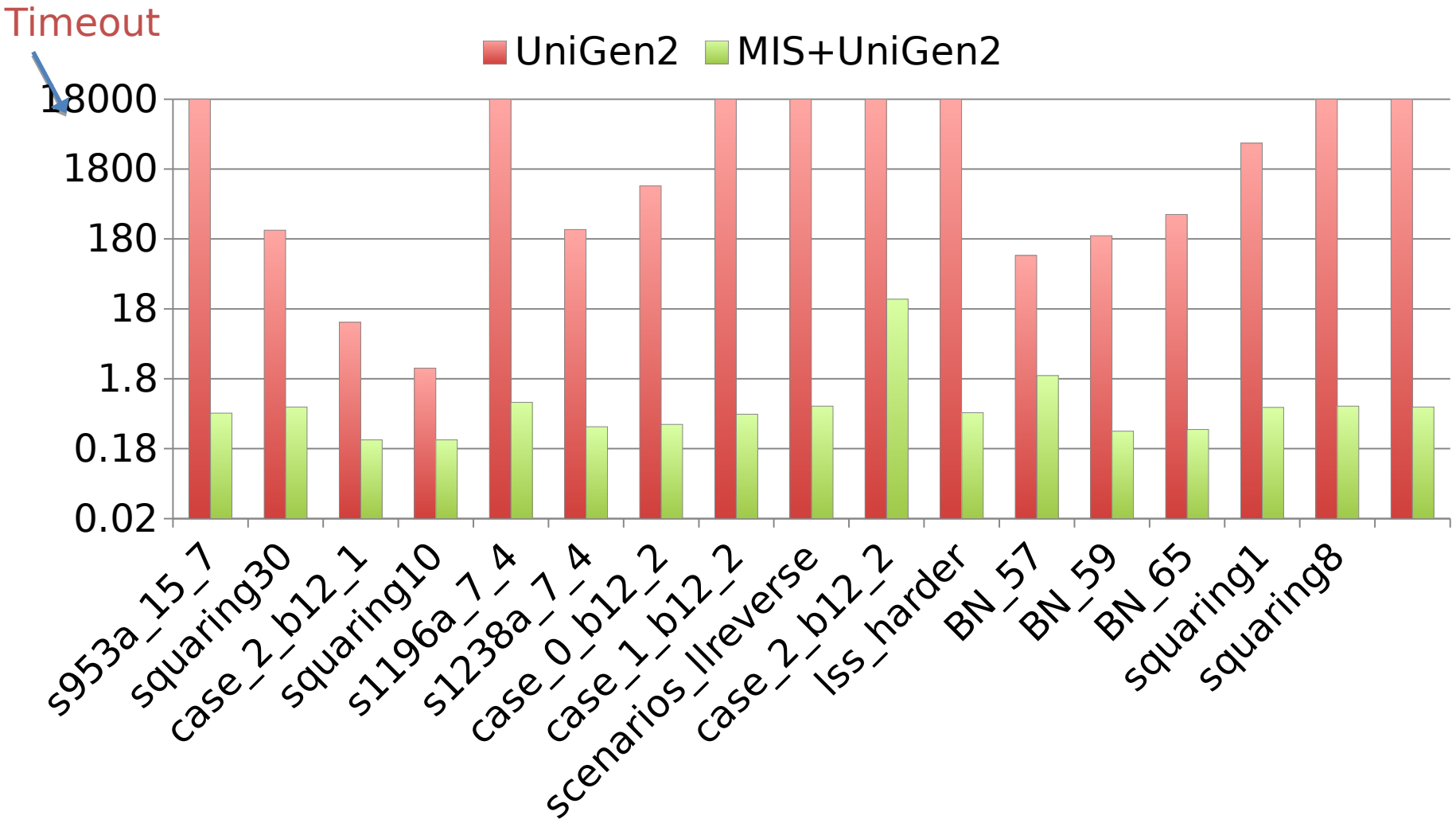
Size of Minimal Independent Supports



Performance Impact on Approximate Model Counting



Performance Impact on Uniform Sampling



Conclusion

- Sampling and counting are fundamental problems with wide variety of applications
- Independent support is key to scalability of the recent techniques
- MIS: First algorithmic procedure to determine independent support
- Provides 1-2 orders of performance improvement in the state-of-art sampler and counters