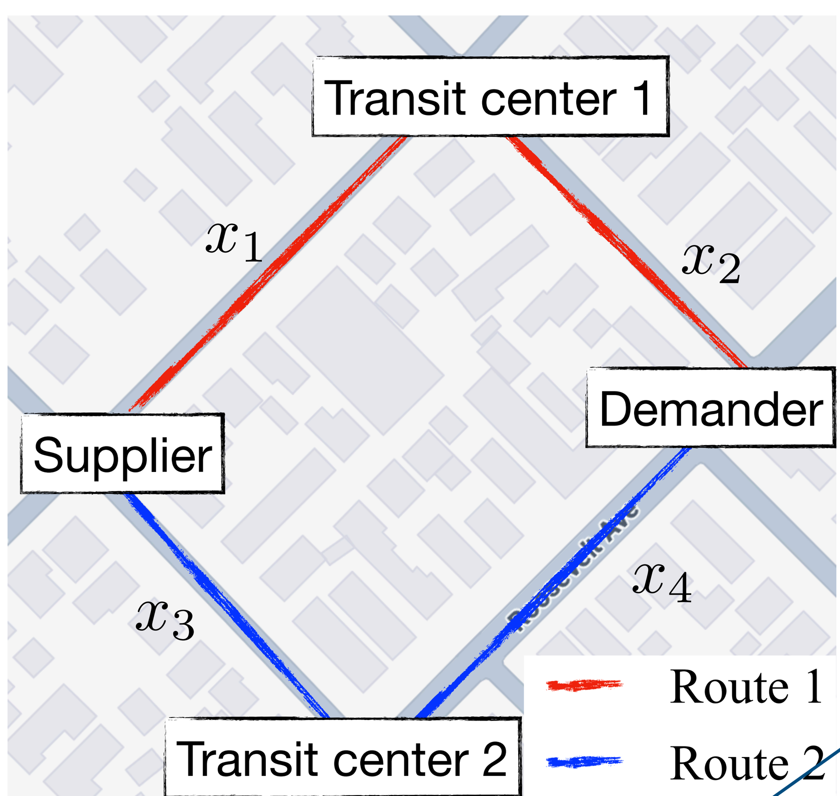
Solving Satisfiability Modulo Counting Exactly with Probabilistic Circuits

Jinzhao Li★, Nan Jiang★, Yexiang Xue. Purdue University

Satisfiability Modulo Counting (SMC) formula: Given a formula $\phi(\mathbf{x},\mathbf{b})$ for Boolean constraints and two sets of weighted functions, $\{f_j\}_{j=1}^{M}$ and $\{g_k\}_{k=1}^{K}$, representing discrete probability distributions, the SMC problem is to determine if the following formula is satisfiable over Boolean variables $\mathbf{x}$ and $\mathbf{b}$:

$$\phi(\mathbf{x},\mathbf{b}), \text{ where } b_j \Leftrightarrow \sum_{\mathbf{y}_j} f_j(\mathbf{x},\mathbf{y}_j) \geq q_j, \quad \text{or } b_j \Leftrightarrow \sum_{\mathbf{y}_j} f_j(\mathbf{x},\mathbf{y}_j) \geq \sum_{\mathbf{z}_k} g_k(\mathbf{x},\mathbf{z}_k), \quad \text{for } j = 1,\ldots,M$$

Motivating example: formulate robust supply chain problem into SMC problem



(a) Pick Route 1 or Route 2?

Ensure a valid route

For Route 1: $b_1 \Rightarrow x_1 \wedge x_2$

For Route 2: $b_2 \Rightarrow x_3 \wedge x_4$

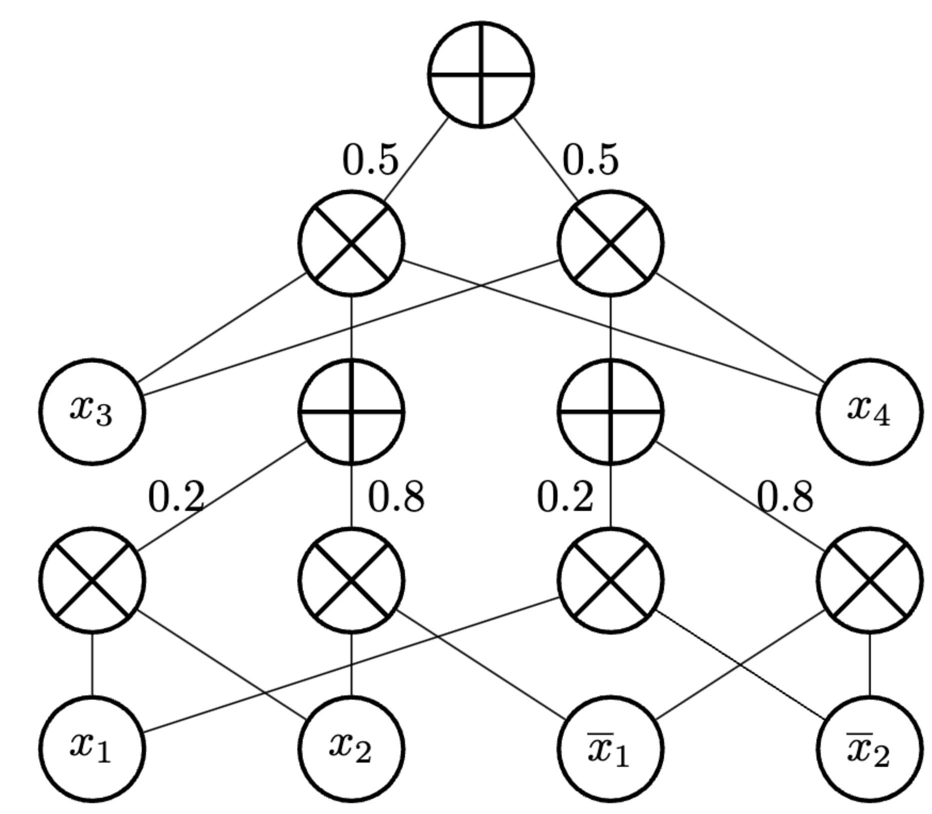
For only one route: $b_1 \oplus b_2$

Ensure sufficient connectivity

For Route 1: $P(b_1 \text{ is accessible}) \geq q$

For Route 2: $P(b_2 \text{ is accessible}) \geq q$

(b) SMC Formulation



(c) Probabilistic circuit for stochastic events

SMC formula

$$\phi(\mathbf{x},\mathbf{b}) = \underbrace{(b_1 \oplus b_2)}_{(a)} \wedge \underbrace{(b_1 \Rightarrow x_1 \wedge x_2)}_{(b)} \wedge \underbrace{(b_2 \Rightarrow x_3 \wedge x_4)}_{(c)}, \text{ where } \underbrace{b_1 \Leftrightarrow \sum_{x_3,x_4} P(x_1,x_2,x_3,x_4) \geq q,}_{(d)} \quad \underbrace{b_2 \Leftrightarrow \sum_{x_1,x_2} P(x_1,x_2,x_3,x_4) \geq q,}_{(e)}$$

Execution steps of baseline exact solvers:

1. Use an SAT solver to solve the Boolean SAT $\phi(\mathbf{x},\mathbf{b})$, e.g.,
   - $x_1 = x_2 = b_1 = \texttt{True}, x_3 = x_4 = b_2 = \texttt{False}$
2. infers the marginal probability:
   - $\sum_{x_3,x_4} P(x_1 = x_2 = \texttt{True}, x_3, x_4) = 0.1 < q$
3. Adds the negated clause $\neg(x_1 \wedge x_2 \wedge b_1)$ to formula ϕ to omit this assignment in the future. Return to step 1.

Execution steps of our Koco-SMC solvers:

1. Suggest a partial assignment for $\phi(\mathbf{x},\mathbf{b})$, e.g.,
   - $x_1 = \texttt{True}$
2. Check the upper bound of the counting part:
   - $\max_{x_2} \sum_{x_3,x_4} P(x_1 = \texttt{True}, x_2, x_3, x_4) = 0.1 < q$.
3. Adds the negated clause $\neg x_1$ to formula ϕ to avoid this assignment in the future. Return to step 1.

Our Koco-SMC saves time by avoiding further assignments to the remaining variables!

Experimental Analysis

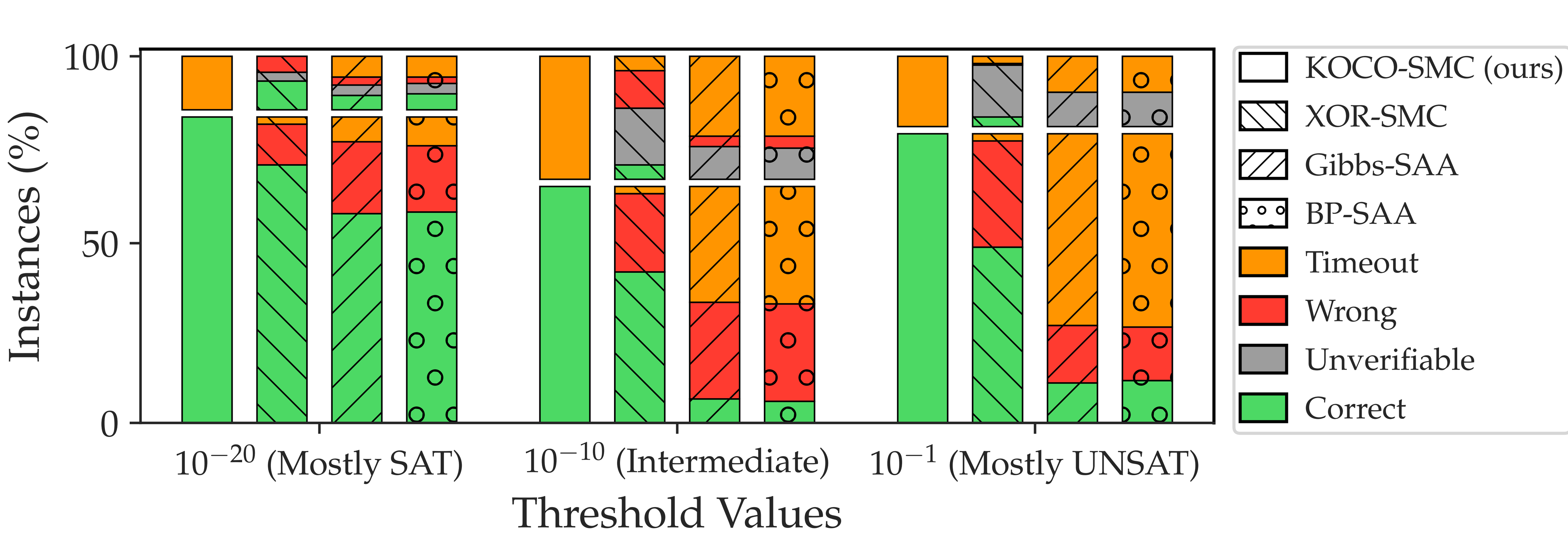


Figure 2. Comparison of KOCO-SMC and approximate solvers on datasets partitioned by threshold values.

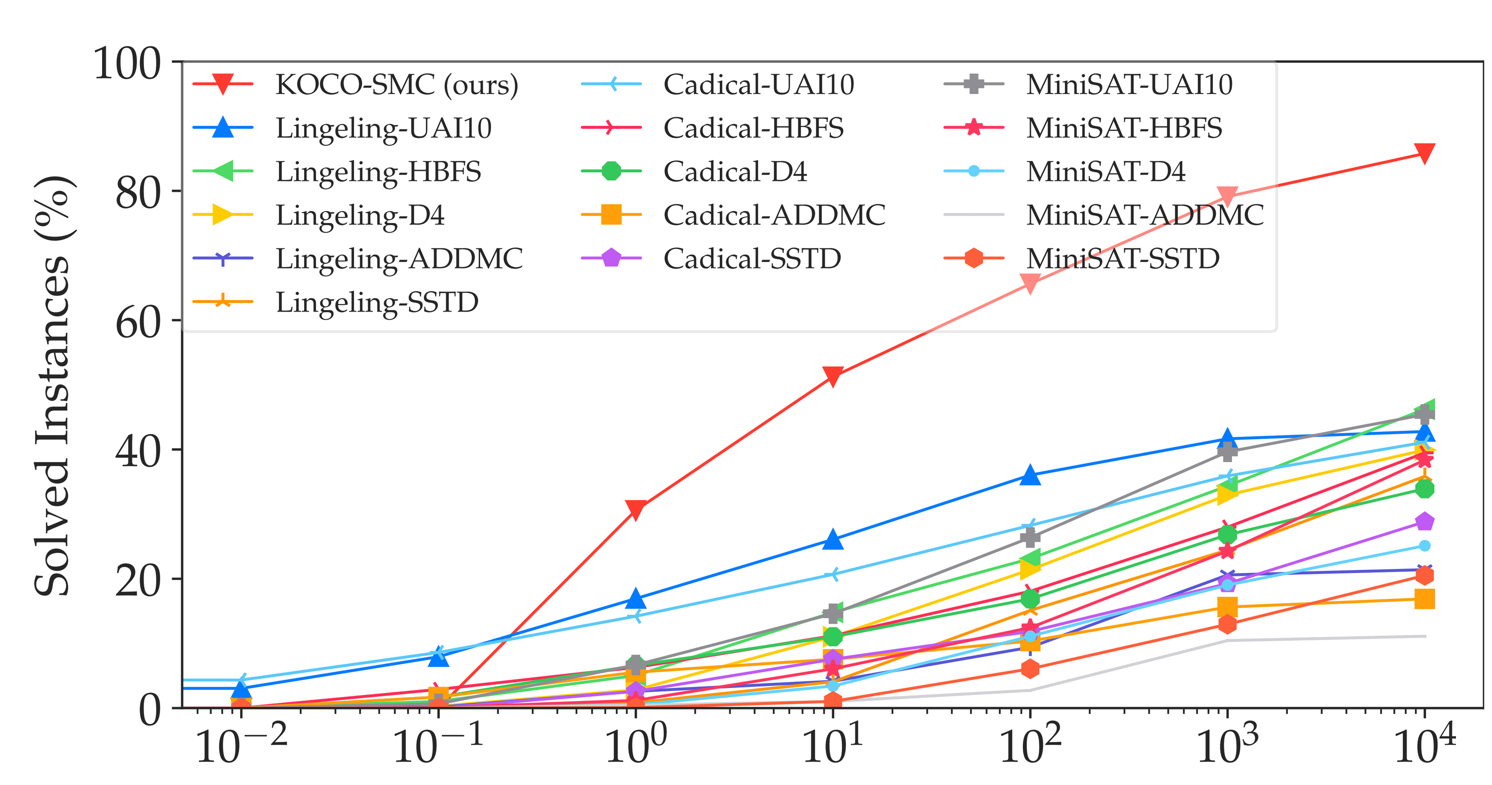


Figure 3. The percentage of instances from the entire dataset solved within the time limit.

Implementation:
https://github.com/jil016/koco-smc

Acknowledgement: This research was supported by NSF grant CCF-1918327, NSF Career Award IIS-2339844, DOE – Fusion Energy Science grant: DE-SC0024583.