Solving Satisfiability Modulo Counting for Symbolic and Statistical AI Integration with Provable Guarantees

Jinzhao Li, Nan Jiang, Yexiang Xue

Department of Computer Science, Purdue University, USA
{li4255, jiang631, yexiang}@purdue.edu

Abstract

Satisfiability Modulo Counting (SMC) encompasses problems that require both symbolic decision-making and statistical reasoning. Its general formulation captures many real-world problems at the intersection of symbolic and statistical Artificial Intelligence. SMC searches for policy interventions to control probabilistic outcomes. Solving SMC is challenging because of its highly intractable nature (NP^{PP}-complete), incorporating statistical inference and symbolic reasoning. Previous research on SMC solving lacks provable guarantees and/or suffers from sub-optimal empirical performance, especially when combinatorial constraints are present. We propose `XOR-SMC`, a polynomial algorithm with access to NP-oracles, to solve highly intractable SMC problems with constant approximation guarantees. `XOR-SMC` transforms the highly intractable SMC into satisfiability problems, by replacing the model counting in SMC with SAT formulae subject to randomized XOR constraints. Experiments on solving important SMC problems in AI for social good demonstrate that `XOR-SMC` outperforms several baselines both in solution quality and running time.

Introduction

Symbolic and statistical approaches are two fundamental driving forces of Artificial Intelligence (AI). Symbolic AI, exemplified by SATisfiability (SAT) and constraint programming, finds solutions satisfying constraints but requires rigid formulations and is difficult to include probabilities. Statistical AI captures uncertainty but often lacks constraint satisfaction. Integrating symbolic and statistical AI remains an open field and has gained research attention recently (cpm 2023; nes 2023; Munawar et al. 2023).

Satisfiability Modulo Counting (SMC) is an umbrella problem at the intersection of symbolic and statistical AI. It encompasses problems that carry out symbolic decision-making (satisfiability) *mixed with* statistical reasoning (model counting). SMC searches for policy interventions to control probabilistic outcomes. Formally, SMC is an SAT problem involving predicates on model counts. Model counting computes the number of models (i.e., solutions) to an SAT formula. Its weighted form subsumes probabilistic inference on Machine Learning (ML) models.

As a motivating SMC application, stochastic connectivity optimization searches for the optimal plan to reinforce the network structure so its connectivity is preserved under stochastic events – a central problem for a city planner who works on securing her residents multiple paths to emergency shelters in case of natural disasters. This problem is useful for disaster preparation (Wu, Sheldon, and Zilberstein 2015), bio-diversity protection (Dilkina and Gomes 2010), internet resilience (Israeli and Wood 2002), social influence maximization (Kempe, Kleinberg, and Tardos 2005), energy security (Almeida et al. 2019), etc. It requires symbolic reasoning (satisfiability) to decide which roads to reinforce and where to place emergency shelters, and statistical inference (model counting) to reason about the number of paths to shelters and the probabilities of natural disasters. Despite successes in many use cases, previous approaches (Williams and Snyder 2005; Conrad et al. 2012; Sheldon et al. 2010; Wu, Sheldon, and Zilberstein 2014) found solutions *lack of certifiable guarantees*, which are unfortunately in need for policy adoption in this safety-related application. Besides, their surrogate approximations of connectivity may overlook important probabilistic scenarios. This results in *suboptimal quality* of the generated plans. As application domains for SMC solvers, this paper considers emergency shelter placement and supply chain network management – two important stochastic connectivity optimization problems.

It is challenging to solve SMC because of their highly intractable nature (NP^{PP}-complete) (Park and Darwiche 2004) – still intractable even with good satisfiability solvers (Biere et al. 2009; Rossi, van Beek, and Walsh 2006; Braunstein, Mézard, and Zecchina 2005) and model counters (Gomes, Sabharwal, and Selman 2006a; Ermon et al. 2013b; Achlioptas and Theodoropoulos 2017; Chakraborty, Meel, and Vardi 2013; Kisa et al. 2014; Cheng et al. 2012; Gogate and Dechter 2012). Previous research on SMC solves either a special case or domain-specific applications (Belanger and McCallum 2016; Welling and Teh 2003; Yedidia, Freeman, and Weiss 2000; Wainwright and Jordan 2008; Fang, Stone, and Tambe 2015; Conitzer and Sandholm 2006; Sheldon et al. 2010). The special case is called the Marginal Maximum-A-Posterior (MMAP) problem, whose decision version can be formulated as a special case of SMC (Marinescu, Dechter, and Ihler 2014; Liu and Ihler 2013; Mauá and de Campos 2012; Jiang, Rai, and III 2011; Lee et al.

2016; Ping, Liu, and Ihler 2015). Both cases are solved by optimizing the surrogate representations of the intractable model counting in variational forms (Liu and Ihler 2012; Kiselev and Poupart 2014), or via knowledge compilation (Choi, Friedman, and den Broeck 2022; Ping, Liu, and Ihler 2015; Mei, Jiang, and Tu 2018) or via sample average approximation (Kleywegt, Shapiro, and Homem-de-Mello 2002; Shapiro 2003; Swamy and Shmoys 2006; Sheldon et al. 2010; Dyer and Stougie 2006; Wu et al. 2017; Xue, Fern, and Sheldon 2015; Verweij et al. 2003).

Nevertheless, previous approaches either cannot quantify the quality of their solutions, or offer one-sided guarantees, or offer guarantees which can be arbitrarily loose. The lack of tight guarantees results in delayed policy adoption in safety-related applications such as the stochastic connectivity optimization considered in this paper. Second, optimizing surrogate objectives without quantifying the quality of approximation leads to sub-optimal behavior empirically. For example, previous stochastic connectivity optimization solvers occasionally produce suboptimal plans because their surrogate approximations overlook cases of significant probability. This problem is amplified when combinatorial constraints are present.

We propose XOR-SMC, ***a polynomial algorithm accessing NP-oracles, to solve highly intractable SMC problems with constant approximation guarantees***. These guarantees hold with high (e.g. $> 99\%$) probability. The strong guarantees enable policy adoption in safety-related domains and improve the empirical performance of SMC solving (e.g., eliminating sub-optimal behavior and providing constraint satisfaction guarantees). The constant approximation means that the solver can correctly decide the truth of an SMC formula if tightening or relaxing the bounds on the model count by a multiplicative constant do not change its truth value. The embedding algorithms allow us to find approximate solutions to beyond-NP SMC problems via querying NP oracles. It expands the applicability of the state-of-the-art SAT solvers to highly intractable problems.

The high-level idea behind XOR-SMC is as follows. Imagine a magic that randomly filters out half of the models (solutions) to an SAT formula. Model counting can be approximated using this magic and an SAT solver – we confirm the SAT formula has more than 2^k models if it is satisfiable after applying this magic k times. This magic can be implemented by introducing randomized constraints. The idea is developed by researchers (Valiant and Vazirani 1986; Jerrum, Valiant, and Vazirani 1986; Gomes, Sabharwal, and Selman 2006b,a; Ermon et al. 2013b,a; Kuck et al. 2019; Achlioptas and Theodoropoulos 2017; Chakraborty, Meel, and Vardi 2013; Chakraborty et al. 2014). In these works, model counting is approximated with guarantees using polynomial algorithms accessing NP oracles. XOR-SMC notices such polynomial algorithms can be encoded as SAT formulae. Hence, SAT-Modulo-Counting can be written as SAT-Modulo-SAT (or equivalently SAT), when we *embed* the SAT formula compiled from algorithms to solve model counting into SMC. The constant approximation guarantee also carries.

We evaluate the performance of XOR-SMC on real-world stochastic connectivity optimization problems. In particular, we consider applied problems of emergency shelter placement and supply chain management. For the shelter placement problem, our XOR-SMC finds high-quality shelter assignments with less computation time and better quality than competing baselines. For wheat supply chain management, the solutions found by our XOR-SMC are better than those found by baselines. XOR-SMC also runs faster than baselines[1].

Preliminaries

Satisfiability Modulo Theories

Satisfiability Modulo Theory (SMT) determines the SATisfiability (SAT) of a Boolean formula, which contains predicates whose truth values are determined by the background theory. SMT represents a line of successful efforts to build general-purpose logic reasoning engines, encompassing complex expressions containing bit vectors, real numbers, integers, and strings, etc (Barrett et al. 2021). Over the years, many good SMT solvers are built, such as the Z3 (de Moura and Bjørner 2008; Bjørner et al. 2018) and cvc5 (Barbosa et al. 2022). They play a crucial role in automated theorem proving, program analysis (Feser et al. 2020), program verification (K., Shoham, and Gurfinkel 2022), and software testing (de Moura and Bjørner 2007).

Model Counting and Probabilistic Inference

Model counting computes the number of models (i.e., satisfying variable assignments) to an SAT formula. Consider a Boolean formula $f(\mathbf{x})$, where the input $\mathbf{x}$ is a vector of Boolean variables, and the output f is also Boolean. When we use 0 to represent false and 1 to represent true, $\sum_x f(\mathbf{x})$ computes the model count. Model counting is closely related to probabilistic inference and machine learning because the marginal inference on a wide range of probabilistic models can be formulated as a weighted model counting problem (Chavira and Darwiche 2008; Xue et al. 2016).

Exact approaches for probabilistic inference and model counting are often based on knowledge compilation (Darwiche and Marquis 2002; Kisa et al. 2014; Choi, Kisa, and Darwiche 2013; Xue, Choi, and Darwiche 2012). Approximate approaches include Variational methods and sampling. Variational methods (Wainwright and Jordan 2008; Wainwright, Jaakkola, and Willsky 2003; Sontag et al. 2008; Hazan and Shashua 2010; Flerova et al. 2011) use tractable forms to approximate a complex probability distribution. Due to a tight relationship between counting and sampling (Jerrum, Valiant, and Vazirani 1986), sampling-based approaches are important for model counting. Importance sampling-based techniques such as SampleSearch (Gogate and Dechter 2007) is able to provide lower bounds. Markov Chain Monte Carlo is asymptotically accurate. However, they cannot provide guarantees except for a limited number of cases (Jerrum and Sinclair 1997; Madras 2002). The authors of (Papandreou and Yuille 2010; Hazan and Jaakkola 2012; Balog et al. 2017) transform weighted integration into optimization queries using extreme value distribution, which

[1]The code is available at: https://github.com/jil016/xor-smc. Please refer to https://arxiv.org/abs/2309.08883 for the Appendix.

Algorithm 1: $\texttt{XOR-Binary}\ (f, \mathbf{x}_0, q)$

```
Randomly sample XOR_1(y), ..., XOR_q(y);
if f(x_0, y) ∧ XOR_1(y) ∧ ... ∧ XOR_q(y) is satisfiable then
  | return True;
else
  | return False;
end
```

today is often called the "Gumbel trick" (Papandreou and Yuille 2011; Jang, Gu, and Poole 2017).

XOR Counting

There is an interesting connection between model counting and solving satisfiability problems subject to randomized XOR constraints. To illustrate this, hold $\mathbf{x}$ at $\mathbf{x}_0$, suppose we would like to know if $\sum_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{x}_0, \mathbf{y})$ exceeds 2^q. Consider the SAT formula:

$$f(\mathbf{x}_0, \mathbf{y}) \wedge \texttt{XOR}_1(\mathbf{y}) \wedge \ldots \wedge \texttt{XOR}_q(\mathbf{y}). \quad (1)$$

Here, $\texttt{XOR}_1, \ldots, \texttt{XOR}_q$ are randomly sampled XOR constraints. $\texttt{XOR}_i(\mathbf{y})$ is the logical XOR or the parity of a randomly sampled subset of variables from $\mathbf{y}$. In other words, $\texttt{XOR}_i(\mathbf{y})$ is true if and only if an odd number of these randomly sampled variables in the subset are true.

Formula (1) is likely to be satisfiable if more than 2^q different $\mathbf{y}$ vectors render $f(\mathbf{x}_0, \mathbf{y})$ true. Conversely, Formula (1) is likely to be unsatisfiable if $f(\mathbf{x}_0, \mathbf{y})$ has less than 2^q satisfying assignments. The significance of this fact is that it essentially transforms model counting (beyond NP) into satisfiability problems (within NP). An intuitive explanation of why this fact holds is that each satisfying assignment $\mathbf{y}$ has 50% chance to satisfy a randomly sampled XOR constraint. In other words, each XOR constraint "filters out" half satisfying assignments. For example, the number of models satisfying $f(\mathbf{x}_0, \mathbf{y}) \wedge \texttt{XOR}_1(\mathbf{y})$ is approximately half of that satisfying $f(\mathbf{x}_0, \mathbf{y})$. Continuing this chain of reasoning, if $f(\mathbf{x}_0, \mathbf{y})$ has more than 2^q solutions, there are still satisfying assignments left after adding q XOR constraints; hence formula (1) is likely satisfiable. The reverse direction can be reasoned similarly. The precise mathematical argument of the constant approximation is in Lemma 1.

Lemma 1. *(Jerrum, Valiant, and Vazirani 1986; Gomes, Sabharwal, and Selman 2006a; Ermon et al. 2013b) Given Boolean function $f(\mathbf{x}_0, y)$ as defined above,*

- *If $\sum_{\mathbf{y}} f(\mathbf{x}_0, \mathbf{y}) \geq 2^{q_0}$, then for any $q \leq q_0$, with probability $1 - \frac{2^c}{(2^c-1)^2}$, $\texttt{XOR-Binary}\ (f, \mathbf{x}_0, q-c)$ returns True.*
- *If $\sum_{\mathbf{y}} f(\mathbf{x}_0, \mathbf{y}) \leq 2^{q_0}$, then for any $q \geq q_0$, with probability $1 - \frac{2^c}{(2^c-1)^2}$, $\texttt{XOR-Binary}\ (w, \theta_0, q+c)$ returns False.*

This idea of transforming model counting problems into SAT problems subject to randomized constraints is rooted in Leslie Valiant's seminal work on unique SAT (Valiant and Vazirani 1986; Jerrum, Valiant, and Vazirani 1986) and has been developed by a rich line of work (Gomes, Sabharwal, and Selman 2006b,a; Ermon et al. 2013b,a; Kuck et al. 2019; Achlioptas and Theodoropoulos 2017; Chakraborty, Meel, and Vardi 2013; Chakraborty et al. 2014). This idea has recently gathered momentum thanks to the rapid progress in SAT solving (Maneva, Mossel, and Wainwright 2007; Braunstein, Mézard, and Zecchina 2005). The contribution of this work extends the success of SAT solvers to problems with even higher complexity, namely, NP^{PP}-complete SMC problems.

Problem Formulation

Satisfiability Modulo Counting (SMC) is Satisfiability Modulo Theory (SMT) (Barrett et al. 2009) with model counting as the background theory. A canonical definition of the SMC problem is to determine if there exists $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ and $\mathbf{b} = (b_1, \ldots, b_k) \in \{0,1\}^k$ that satisfies the formula:

$$\phi(\mathbf{x}, \mathbf{b}), b_i \Leftrightarrow \left(\sum_{\mathbf{y}_i \in \mathcal{Y}_i} f_i(\mathbf{x}, \mathbf{y}_i) \geq 2^{q_i} \right), \forall i \in \{1.., k\}. \quad (2)$$

Here each b_i is a Boolean predicate that is true if and only if the corresponding model count exceeds a threshold. Bold symbols (i.e., $\mathbf{x}$, $\mathbf{y}_i$ and $\mathbf{b}$) are vectors of Boolean variables. $\phi, f_1, \ldots, f_k$ are Boolean functions (i.e., their input is Boolean vectors, and their outputs are also Boolean). We use 0 to represent false and 1 to represent true. Hence $\sum f_i$ computes the number of satisfying assignments (model counts) of f_i. The directions of the inequalities do not matter much because one can always negate each f_i. For instance, let $f(\mathbf{x}, \mathbf{y})$ be a Boolean function (output is 0 or 1). $\sum_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{x}, \mathbf{y}) \leq 2^q$ can be converted by negating f and modifying the threshold to $|\mathcal{Y}| - 2^q$, resulting in an equivalent predicate $\sum_{\mathbf{y}\in\mathcal{Y}} (\neg f(\mathbf{x}, \mathbf{y})) \geq |\mathcal{Y}| - 2^q$.

Our XOR-SMC algorithm obtains the constant approximation guarantee to the following slightly relaxed SMC problems. The problem $\texttt{SMC}(\phi, f_1, \ldots, f_k,\ q_1, \ldots, q_k)$ finds a satisfying assignment $(\mathbf{x}, \mathbf{b})$ for:

$$\phi(\mathbf{x}, \mathbf{b}) \wedge \left[b_1 \Rightarrow \left(\sum_{\mathbf{y}_1 \in \mathcal{Y}_1} f_1(\mathbf{x}, \mathbf{y}_1) \geq 2^{q_1} \right) \right] \cdots \wedge \left[b_k \Rightarrow \left(\sum_{\mathbf{y}_k \in \mathcal{Y}_k} f_k(\mathbf{x}, \mathbf{y}_k) \geq 2^{q_k} \right) \right]. \quad (3)$$

The only difference compared to the full-scale problem in Eq. (2)) is the replacement of $\Leftrightarrow$ with $\Rightarrow$. This change allows us to derive a concise constant approximation bound. We also mention that all the applied SMC problems considered in this paper can be formulated in this relaxed form. We thank the reviewers for pointing out the work of (Fredrikson and Jha 2014), who came up with a slightly different SMC formulation with focused applications in privacy and an exact solver. Their formulation was a little more general than ours, since theirs allows for predicates like $\sum f \geq \sum g$, while ours only allows for $\sum f \geq$ constant. However, our formulation can handle $\sum f \geq \sum g$ by formulating it with $(\sum f \geq \alpha) \wedge (\sum g \leq \alpha)$ and binary searching on α.

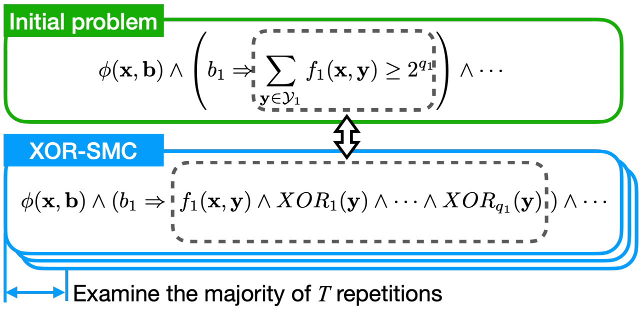

Figure 1: Our XOR-SMC (shown in Algorithm 2) solves the intractable model counting with satisfiability problems subject to randomized XOR constraints and obtains constant approximation guarantees for SMC.

The XOR-SMC Algorithm

The key motivation behind our proposed XOR-SMC algorithm is to notice that Algorithm 1 itself can be written as a Boolean formula due to the Cook-Levin reduction. When we embed this Boolean formula into Eq. (3), the Satisfiability-Modulo-Counting problem translates into a Satisfiability-Modulo-SAT problem, or equivalently, an SAT problem. This embedding also ensures a constant approximation guarantee (see Theorem 2).

To illustrate the high-level idea, let us consider replacing each $\sum_{\mathbf{y}_i\in\mathcal{Y}_i} f_i(\mathbf{x},\mathbf{y}_i) \geq 2^{q_i}$ in Eq. (3) with formula

$$f_i(\mathbf{x},\mathbf{y}_i) \wedge \mathtt{XOR}_1(\mathbf{y}_i) \wedge \ldots \wedge \mathtt{XOR}_{q_i}(\mathbf{y}_i). \qquad (4)$$

We denote the previous equation (4) as $\gamma(f_i,\mathbf{x},q_i,\mathbf{y}_i)$. This replacement results in the Boolean formula:

$$\phi(\mathbf{x},\mathbf{b}) \wedge [b_1 \Rightarrow \gamma(f_1,\mathbf{x},q_1,\mathbf{y}_1)] \wedge \cdots \wedge [b_k \Rightarrow \gamma(f_k,\mathbf{x},q_k,\mathbf{y}_k)]. \qquad (5)$$

We argue that the satisfiability of formula (5) should be closely related to that of formula (3) due to the connection between model counting and satisfiability testing subject to randomized constraints (discussed in Section). To see this, Eq. (5) is satisfiable if and only if there exists $(\mathbf{x},\mathbf{b},\mathbf{y}_1,\ldots,\mathbf{y}_k)$ that render Eq. (5) true (notice $\mathbf{y}_1,\ldots,\mathbf{y}_k$ are also its variables). Suppose $\mathtt{SMC}(\phi,f_1,\ldots,f_k,q_1+c,\ldots,q_k+c)$ is satisfiable (a.k.a., Eq. (3) is satisfiable when q_i is replaced with q_i+c). Let $(\mathbf{x},\mathbf{b})$ be a satisfying assignment. For any $b_i=1$ (true) in $\mathbf{b}$, we must have $\sum_{\mathbf{y}_i\in\mathcal{Y}_i} f_i(\mathbf{x},\mathbf{y}_i) \geq 2^{q_i+c}$. This implies with a good chance, there exists a $\mathbf{y}_i$ that renders $\gamma(f_i,\mathbf{x},q_i,\mathbf{y}_i)$ true. This is due to the discussed connection between model counting and SAT solving subject to randomized constraints. Hence $b_i \Rightarrow \gamma(f_i,\mathbf{x},q_i,\mathbf{y}_i)$ is true. For any $b_i=0$ (false), the previous equation is true by default. Combining these two facts and $\phi(\mathbf{x},\mathbf{b})$ is true, we see Eq. (5) is true.

Conversely, suppose $\mathtt{SMC}(\phi,f_1,\ldots,f_k,q_1-c,\ldots,q_k-c)$ is not satisfiable. This implies for every $(\mathbf{x},\mathbf{b})$, either $\phi(\mathbf{x},\mathbf{b})$ is false, or there exists at least one j such that b_j is true, but $\sum_{\mathbf{y}_j\in\mathcal{Y}_j} f_j(\mathbf{x},\mathbf{y}_j) < 2^{q_j-c}$. The first case implies Eq. (5) is false under the assignment. For the second case, $\sum_{\mathbf{y}_j\in\mathcal{Y}_j} f_j(\mathbf{x},\mathbf{y}_j) < 2^{q_j-c}$ implies with a good chance there is no $\mathbf{y}_j$ to make $\gamma(f_j,\mathbf{x},q_j,\mathbf{y}_j)$ true. Combining these two facts, with a good chance Eq. (5) is not satisfiable.

Algorithm 2: XOR-SMC (ϕ, $\{f_i\}_{i=1}^k$, $\{q_i\}_{i=1}^k$, η, c)

```
T ← ⌈((n+k) ln 2 − ln η) / α(c,k)⌉;
for t = 1 to T do
    for i = 1 to k do
        ψ_i^(t) ← f_i(x, y_i^(t));
        for j = 1, ..., q_i do
            ψ_i^(t) ← ψ_i^(t) ∧ XOR_j(y_i^(t));
        end
        ψ_i^(t) ← ψ_i^(t) ∨ ¬b_i;
    end
    ψ_t ← ψ_1^(t) ∧ ··· ∧ ψ_k^(t);
end
φ* ← φ ∧ Majority(ψ_1, ..., ψ_T);
if there exists (x, b, {y_i^(1)}_{i=1}^k, ..., {y_i^(T)}_{i=1}^k) that satisfies φ* then
    return True;
else
    return False;
end
```

In practice, to reduce the error probability the determination of the model count needs to rely on the majority satisfiability status of a series of equations (4) (instead of a single one). Hence we develop Algorithm 2, which is a little bit more complex than the high-level idea discussed above. The idea is still to ***transform the highly intractable SMC problem into solving an SAT problem of its polynomial size***, while ***ensuring a constant approximation guarantee***. Fig. 1 displays the encoding of Algorithm 2. We can see the core is still to replace the intractable model counting with satisfiability problems subject to randomized constraints. We prove XOR-SMC has a constant approximation guarantee in Theorem 2. We leave the implementation of XOR-SMC in the Appendix B.

Theorem 2. *Let $0<\eta<1$ and $c \geq \log(k+1)+1$. Select $T = \lceil ((n+k)\ln 2 - \ln\eta)/\alpha(c,k) \rceil$, we have*

- *Suppose there exists $\mathbf{x}_0 \in \{0,1\}^n$ and $\mathbf{b}_0 \in \{0,1\}^k$, such that $\mathtt{SMC}(\phi,f_1,\ldots,f_k,q_1+c,\ldots,q_k+c)$ is true. In other words,*

$$\phi(\mathbf{x}_0,\mathbf{b}_0) \wedge \left(\bigwedge_{i=1}^k \left(b_i \Rightarrow \sum_{\mathbf{y}_i} f_i(\mathbf{x}_0,\mathbf{y}_i) \geq 2^{q_i+c} \right) \right),$$

Then algorithm XOR-SMC (ϕ, $\{f_i\}_{i=1}^k$, $\{q_i\}_{i=1}^k$, T) returns true with probability greater than $1-\eta$.

- *Contrarily, suppose $\mathtt{SMC}(\phi,f_1,\ldots,f_k,q_1-c,\ldots,q_k-c)$ is not satisfiable. In other words, for all $\mathbf{x} \in \{0,1\}^n$ and $\mathbf{b} \in \{0,1\}^k$,*

$$\neg\left(\phi(\mathbf{x},\mathbf{b}) \wedge \left(\bigwedge_{i=1}^k \left(b_i \Rightarrow \sum_{\mathbf{y}_i} f_i(\mathbf{x},\mathbf{y}_i) \geq 2^{q_i-c} \right) \right) \right),$$

then XOR-SMC *(ϕ, $\{f_i\}_{i=1}^k$, $\{q_i\}_{i=1}^k$, T) returns false with probability greater than $1-\eta$.*

Proof. **Claim 1:** Suppose there exists $\mathbf{x}_0 = [x_1, \ldots, x_n] \in \{0,1\}^n$ and $\mathbf{b}_0 = [b_1, \ldots, b_k] \in \{0,1\}^k$, such that

$$\phi(\mathbf{x}_0, \mathbf{b}_0) \wedge \left(\bigwedge_{i=1}^{k} \left(b_i \Rightarrow \sum_{\mathbf{y}_i} f_i(\mathbf{x}_0, \mathbf{y}_i) \geq 2^{q_i+c} \right) \right) \quad (6)$$

holds true. Denote k_0 as the number of non-zero bits in $\mathbf{b}_0$. Without losing generality, suppose those non-zero bits are the first k_0 bits, i.e., $b_1 = b_2 = \cdots = b_{k_0} = 1$ and $b_i = 0, \forall i > k_0$. Then Eq. (6) can be simplified to:

$$\phi(\mathbf{x}_0, \mathbf{b}_0) \wedge \left(\bigwedge_{i=1}^{k_0} \left(\sum_{\mathbf{y}_i} f_i(\mathbf{x}_0, \mathbf{y}_i) \geq 2^{q_i+c} \right) \right) \quad (7)$$

Consider the Boolean formula ψ_t defined in the XOR-SMC algorithm (choosing any $t \in \{1, \ldots, T\}$). ψ_t can be simplified by substituting the values of $\mathbf{x}_0$ and $\mathbf{b}_0$. After simplification, we obtain:

$$\psi_t = \left(f_1(\mathbf{x}_0, \mathbf{y}_1^{(t)}) \wedge \mathtt{XOR}_1(\mathbf{y}_1^{(t)}) \cdots \wedge \mathtt{XOR}_{q_1}(\mathbf{y}_1^{(t)}) \right) \wedge \ldots$$
$$\wedge \left(f_{k_0}(\mathbf{x}_0, \mathbf{y}_{k_0}^{(t)}) \wedge \mathtt{XOR}_1(\mathbf{y}_{k_0}^{(t)}) \cdots \wedge \mathtt{XOR}_{q_{k_0}}(\mathbf{y}_{k_0}^{(t)}) \right).$$

Let $\gamma_i = \left(f_i(\mathbf{x}_0, \mathbf{y}_i^{(t)}) \wedge \mathtt{XOR}_1(\mathbf{y}_i^{(t)}) \cdots \wedge \mathtt{XOR}_{q_i}(\mathbf{y}_i^{(t)}) \right)$. Observing that $\sum_{\mathbf{y}_i} f_i(\mathbf{x}_0, \mathbf{y}_i) \geq 2^{q_i+c}, \forall i = 1, \ldots, k_0$. According to Lemma 1, with probability at least $1 - \frac{2^c}{(2^c-1)^2}$, there exists $\mathbf{y}_i^{(t)}$, such that $(\mathbf{x}_0, \mathbf{y}_i^{(t)})$ renders γ_i true. The probability that ψ_t is true under $(\mathbf{x}_0, \mathbf{b}_0, \mathbf{y}_1^{(t)}, \ldots, \mathbf{y}_k^{(t)})$ is:

$$\begin{aligned}
&\mathbb{P}((\mathbf{x}_0, \mathbf{b}_0, \mathbf{y}_1^{(t)}, \ldots, \mathbf{y}_k^{(t)}) \text{ renders } \psi_t \text{ true}) \\
&= \mathbb{P}\left(\bigwedge_{i=1}^{k_0} ((\mathbf{x}_0, \mathbf{y}_i^{(t)}) \text{ renders } \gamma_i \text{ false}) \right) \\
&= 1 - \mathbb{P}\left(\bigvee_{i=1}^{k_0} ((\mathbf{x}_0, \mathbf{y}_i^{(t)}) \text{ renders } \gamma_i \text{ false}) \right) \\
&\geq 1 - \sum_{i=1}^{k_0} \mathbb{P}\left((\mathbf{x}_0, \mathbf{y}_i^{(t)}) \text{ renders } \gamma_i \text{ false} \right) \\
&\geq 1 - \frac{k_0 2^c}{(2^c-1)^2} \geq 1 - \frac{k 2^c}{(2^c-1)^2}.
\end{aligned}$$

Define Γ_t as a binary indicator variable where

$$\Gamma_t = \begin{cases} 1 & \text{if } (\mathbf{x}_0, \mathbf{b}_0, \mathbf{y}_1^{(t)}, \ldots, \mathbf{y}_k^{(t)}) \text{ renders } \psi_t \text{ true}, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore $\mathbb{P}(\Gamma_t = 0) \leq \frac{k2^c}{(2^c-1)^2}$. $\mathbb{P}(\Gamma_t = 0) < \frac{1}{2}$ when $c \geq \log_2(k+1) + 1$. XOR-SMC returns true if the majority of $\psi_t, t = 1, \ldots, T$ are true; that is, $\sum_t \Gamma_t \geq \frac{T}{2}$. Let's define

$$\begin{aligned}
\alpha(c,k) &= D\left(\frac{1}{2} \middle\| \frac{k2^c}{(2^c-1)^2} \right) \\
&= \frac{1}{2} \ln \frac{(2^c-1)^2}{k2^{c+1}} + \left(1 - \frac{1}{2}\right) \ln \frac{2(2^c-1)^2}{(2^c-1)^2 - k2^{c+1}}.
\end{aligned}$$

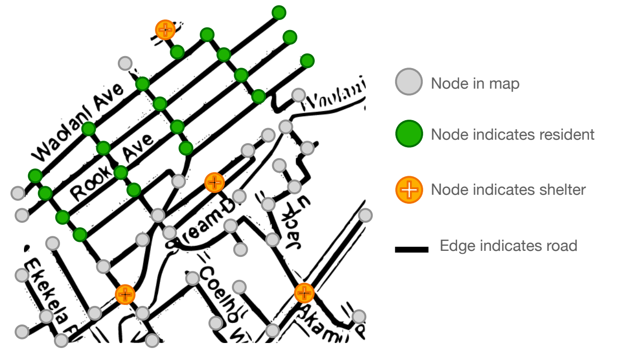

Figure 2: Example assignment of shelters that guarantee sufficient alternative paths from the resident areas, at Hawaii Island. Every orange dot corresponds to shelters and the green dot indicates a resident area.

When $c \geq \log_2(k+1) + 1$, observing that $\alpha(c,k) > 0$, we can apply the Chernoff-Hoeffding theorem to obtain:

$$\mathbb{P}\left(\sum_{t=1}^{T} \Gamma_t \geq \frac{T}{2} \right) = 1 - \mathbb{P}\left(\sum_{t=1}^{T} \Gamma_t < \frac{T}{2} \right) \geq 1 - e^{-\alpha(c,k)T}$$

For $T \geq \lceil \frac{((n+k)\ln 2 - \ln \eta)}{\alpha(c,k)} \rceil \geq \frac{-\ln \eta}{\alpha(c,k)}$, it follows that $e^{-\alpha(c,k)T} \leq \eta$. Therefore, with a probability at least $1-\eta$, we have $\sum_t \Gamma_t \geq \frac{T}{2}$. In this scenario, XOR-SMC (ϕ, $\{f_i\}_{i=1}^k$, $\{q_i\}_{i=1}^k$, T) returns true as it discovers $\mathbf{x}_0$, $\mathbf{b}_0$, $(\mathbf{y}_1^{(t)}, \ldots, \mathbf{y}_k^{(t)})$, for which the majority of Boolean formulae in $\{\psi_t\}_{t=1}^T$ are true.

Claim 2: Suppose for all $\mathbf{x} \in \{0,1\}^n$ and $\mathbf{b} \in \{0,1\}^k$,

$$\neg \left(\phi(\mathbf{x}, \mathbf{b}) \wedge \left(\bigwedge_{i=1}^{k} \left(b_i \Rightarrow \sum_{\mathbf{y}_i} f_i(\mathbf{x}, \mathbf{y}_i) \geq 2^{q_i-c} \right) \right) \right)$$

Consider a fixed $\mathbf{x}_1$ and $\mathbf{b}_1$, the previous condition with high probability renders most ψ_t false in Algorithm 2. We prove that the probability is sufficiently low such that XOR-SMC will return false with a high probability after examining all $\mathbf{x}$ and $\mathbf{b}$. The detailed proof is left in Appendix A. □

Experiment 1: Locate Emergency Shelters

Problem Formulation. Disasters such as hurricanes and floods continue to endanger millions of lives. Shelters are safe zones that protect residents from possible damage, and evacuation routes are the paths from resident zones toward shelter areas. To enable the timely evacuation of resident zones, picking a set of *shelter locations* with sufficient routing from resident areas should be considered. Given the unpredictability of chaos during natural disasters, it is crucial to guarantee multiple paths rather than one path from residential areas to shelters. This ensures that even if one route is obstructed, residents have alternative paths to safety areas.

Given a map $G = (V, E)$ where nodes in $V = \{v_1, \ldots, v_N\}$ represent N areas and an edge $e = (v_i, v_j) \in E$ indicates a road from v_i to v_j, N and M denote the number of nodes and edges, respectively. Given a subset of nodes

	Graph Size		
	$N = 121$	$N = 183$	$N = 388$
XOR-SMC (ours)	$\mathbf{0.04}hr$	$\mathbf{0.11}hr$	$\mathbf{0.16}hr$
GibbsSampler-LS	$0.56hr$	$0.66hr$	$6.97hr$
QuickSampler-LS	$0.31hr$	$0.29hr$	$0.62hr$
Unigen-LS	$0.08hr$	$0.17hr$	$0.42hr$

Table 1: XOR-SMC takes less empirical running time than baselines to find shelter location assignments over different graphs. Graph size is the number of nodes in the graph.

$R = \{v_{r_1}, \dots, v_{r_k}\} \subseteq V$ indicates the *residential areas*, the task is to choose at most m nodes as shelters from the rest of the nodes, such that the number of routes that can reach a shelter from each residential area is maximized. Fig. 2 gives an example with $m = 4$ shelters and there are sufficiently many roads connecting the resident area to those shelters.

Current methods (Bayram and Yaman 2018; Amideo, Scaparra, and Kotiadis 2019) considered finding shelter locations that have at least one single path from a residential area. However, those proposed methods cannot be generalized to solve the problem that requires sufficient alternative routes from residential area to shelters, primarily because counting the number of paths is intractable. This complexity makes it difficult to solve large-scale problems of this type.

SMC Formulation. XOR-SMC transforms this optimization problem into a decision problem by gradually increasing the path count threshold q_r. The decision problem decides if there are at least 2^{q_r} paths connecting any residential area with a shelter. The assigned shelters is represented by a vector $= (b_1, \dots, b_n) \in \{0, 1\}^n$, where $b_i = 1$ implies node v_i is chosen as shelter. Let $\phi(\mathbf{b}) = (\sum_{i=1}^{n} b_i) \leq m$ represent there are at most m shelters. Let $f(v_r, v_s, E')$ be an indicator function that returns one if and only if the selected edges E' form a path from v_r to v_s. The whole formula is:

$$\phi(\mathbf{b}), b_i \Rightarrow \left(\sum_{v_s \in S, E' \subseteq E} f(v_r, v_s, E') \geq 2^{q_r} \right) \text{for } 1 \leq i \leq n.$$

We leave the details implementation of $f(v_r, v_s, E')$ in the Appendix C.

Empirical Experiment Analysis

Experiment Setting. We crawl the real-world dataset from the Hawaii Statewide GIS Program website. We extract the real Hawaii map with those major roads and manually label those resident areas on the map. We create problems of different scales by subtracting different sub-regions from the map. 3 major resident areas are picked as R, and set $m = 5$.

In terms of baselines, we consider the local search algorithm with shelter locations as the state and the number of paths between shelters and resident areas as the heuristic. Due to the intractability of path counting in our formulation, the heuristic is approximated by querying sampling oracles. In particular, we consider 1) Gibbs sampling-based (Geman and Geman 1984) Local Search (Gibbs-LS). 2) Uniform SAT sampler-based (Soos, Gocht, and Meel 2020) Local Search (Unigen-LS). 3) Quick Sampler-based (Dutra et al. 2018) Local Search (Quick-LS). Each baseline runs 5 times, and the best result is included. Each run is until the approach finds a local minimum. For our XOR-SMC, we give a time limit of 12 hours. The algorithm repeatedly runs with increasing q_r until it times out. The time shown in Table 1 and the number of paths in Figure 2 correspond to the cumulative time and the best solutions found before the algorithm times out. For the evaluation metrics, we consider 1) the number of paths identified in the predicted plan by each algorithm, and 2) the total running time of the process.

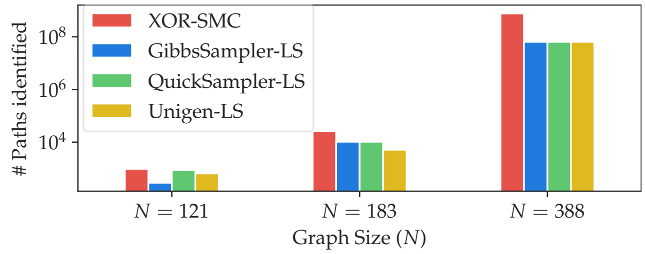

Figure 3: XOR-SMC finds better shelter locations with orders of magnitudes more paths from the resident area to the chosen shelters than competing baselines across different graphs (y-axis in log scale).

Result Analysis. In terms of running time (in Table 1), XOR-SMC takes less empirical running time than baselines for finding shelter location assignments over different graphs. In Fig. 3, we evaluate the quality of the predicted shelters by counting the number of connecting paths from residents to the shelters. The path is counted by directly solving the counting predicate by SharpSAT-TD (Korhonen and Järvisalo 2021) with given shelter locations. The shelter locations selected by our XOR-SMC lead to a higher number of paths than those found by the baselines.

Experiment 2: Robust Supply Chain Design

Problem Formulation. Supply chain management found its importance in operations research and economics. The essence of supply chain management is to integrate the flow of products and finances to maximize value to the consumer. Its importance is underscored by the increasing complexity of business environments, where minor inefficiencies by random disasters result in significant extra costs.

Given a supply chain of N suppliers, they form a supply-demand network (V, E) where each node $v \in V$ represents a supplier and edges $e \in E$ represent supply-to-demand trades. Each supplier v acts as a vendor to downstream suppliers and also as a buyer from upstream suppliers. We assume a conservation of production conditions for each node, i.e., the input-output ratio is 1. To guarantee substantial production, supplier v should order necessary raw materials from upstream suppliers in advance. Denote the cost of trade between vendor u, and buyer v as $c(u, v)$. Let the amount of goods of the trade between u and v be $f(u, v)$ and $B(v)$ be the total budget of to get all his materials ready. Due to unpredictable natural disasters and equipment failure, the trade between u and v may fail. Denote

	Small Supply Network Disaster Scale (% of affected edges)			Real-world Supply Network Disaster Scale (% of affected edges)		
	10%	20%	30%	10%	20%	30%
XOR-SMC (ours)	$\mathbf{95.0(0.67s)}$	$\mathbf{96.0(1.19s)}$	$\mathbf{119.3(17.63s)}$	$\mathbf{232.2(57.8s)}$	$\mathbf{210.2(75.5s)}$	$\mathbf{147.8(96.9s)}$
BP-SAA	$86.5(1.17s)$	$82.7(3.76s)$	$118.8(11.48s)$	$187.2(2hr)$	$143.7(2hr)$	$130.5(2hr)$
Gibbs-SAA	$88.0(1.24s)$	$95.4(4.48s)$	$113.3(15.61s)$	$155.0(2hr)$	$147.1(2hr)$	$111.0(2hr)$
IS-SAA	$80.5(0.98s)$	$76.7(3.29s)$	$109.6(21.74s)$	$157.4(2hr)$	$123.5(2hr)$	$132.9(2hr)$
LoopyIS-SAA	$88.0(0.84s)$	$95.4(2.96s)$	$112.3(17.95s)$	$200.2(2hr)$	$149.5(2hr)$	$130.5(2hr)$
Weighted-SAA	$86.5(1.11s)$	$95.4(3.57s)$	$117.0(11.80s)$	$170.5(2hr)$	$169.0(2hr)$	$130.3(2hr)$
Best possible values	100.0	104.0	128.0	237.0	226.0	235.0

Table 2: Empirical average of the total production (tons) and running time on two supply networks. Our XOR-SMC finds better solutions (higher production) with less time usage compared to baselines.

$\boldsymbol{\theta} = (\theta_1, \dots, \theta_L) \in \{0,1\}^L$ as the state of L different stochastic events, where $\theta_l = 1$ indicates event l occurs. The objective is to design a global trading plan maximizing the expected total production output, accounting for stochastic influences. This is measured by the output of final-tier suppliers, who produce end goods without supplying others.

Existing works in supply chain optimization often gravitate towards mathematical programming approaches. However, when delving into more complex scenarios involving stochastic events—such as uncertain demand or supply disruptions—the task becomes considerably more intricate. Specifically, formulating counting constraints, like guaranteeing a certain amount of supplies across multiple vendors under stochastic events, is intractable. These complexities necessitate innovative approaches that can capture the randomness and dynamism inherent in real-world supply chain systems without sacrificing optimality.

SMC Formulation. Similarly, we transform the maximization into a decision problem by gradually increasing the threshold of the production. Denote the trading plan as $\mathbf{x} = (x_{u,v}, \dots) \in \{0,1\}^{|E|}$, where $x_{u,v} = 1$ indicates that v purchases raw material from u. The decision problem can be formulated into an SMC problem as follows:

$$\phi(\mathbf{x}), \sum_{v \in D} \mathbb{E}_\theta \left[\sum_{(u,v) \in E_{\mathbf{x},\boldsymbol{\theta}}} f(u,v) \right] \geq 2^q \qquad (8)$$

where D represents final-tier supplier nodes, 2^q is a minimum production level to be guaranteed, $E_{\mathbf{x},\boldsymbol{\theta}}$ is the remaining sets of edges after applying trading plan $\mathbf{x}$ and under the stochastic events of disasters $\boldsymbol{\theta}$. $\phi(\mathbf{x})$ encodes all essential constraints, e.g., adherence to budget limits, output not exceeding input at each node, etc. We leave the formulation details in Appendix C.

Empirical Experiment Analysis

Experiment Setting. The dataset is the wheat supply chain network from (Zokaee et al. 2017). The dataset only provides the cost of trade, the capacity of transportation, and raw material demand. We further generate stochastic events of disasters (see Appendix C) over different portions of supply-demand edges. The random disasters make the expectation computation intractable. For a better comparison of running time, a small-scale synthetic network is included, in which the cost, budget, and capacities are randomly generated. For the small supply network, the number of nodes in each layer is $[4, 4, 5, 5]$. For the real-world supply network, the number of nodes in each layer is $[9, 7, 9, 19]$.

For the baseline, we utilize Sample Average Approximation (SAA)-based methods (Kleywegt, Shapiro, and Homem-de-Mello 2002). These baselines employ Mixed Integer Programming (MIP) to identify a trading plan that directly maximizes the average production across networks impacted by 100 sampled disasters. The average over samples serves as a proxy for the actual expected production. For the sampler, we consider Gibbs sampling (Gibbs-SAA), belief propagation (BP-SAA), importance sampling (IS-SAA), loopy-importance sampling, and weighted sampling (Weighted-SAA). For a fair comparison, we imposed a time limit of 30 seconds for the small-sized network and 2 hours for the real-world network. The time shown in Table 2 for SAA approaches is their actual execution time. Our SMC solver again executes repeatedly with increasing q until it times outs. The time shown is the cumulative time it finds the best solution (last one) before the time limit.

To evaluate the efficacy of a trading plan, we calculate its empirical average production under $10,000$ i.i.d. disasters, sampled from the ground-truth distribution. The production numbers are reported in Table 2. This method is adopted due to the computational infeasibility to calculate expectations directly. For SAA approaches that exceed the time limit, the production numbers in Table 2 are for the best solutions found within the time limit.

Result Analysis. Table 2 shows the production and running times of plans derived from various methods. For small networks, while SAA-based methods can complete MIP within the time limit, they remain sub-optimal as they optimize a surrogate expectation derived from sampling, which deviates from the true expectation. In the case of larger networks, these methods further struggle due to the poor scalability of the large MIP formulation. They fail to find good solutions within the 2-hour time limit. In contrast, XOR-SMC, by formulating as an SMC problem, directly optimizes the intractable expectation using XOR counting, and yields superior solution quality and less running time.

Conclusion

We presented XOR-SMC, an algorithm with polynomial approximation guarantees to solve the highly intractable Satisfiability Modulo Counting (SMC) problems. Solving SMC problems presents unique challenges due to their intricate nature, integrating statistical inference and symbolic reasoning. Prior work on SMC solving offers no or loose guarantees and may find suboptimal solutions. XOR-SMC transforms the intractable SMC problem into satisfiability problems by replacing intricate model counting with SAT formulae subject to randomized XOR constraints. XOR-SMC also obtains constant approximation guarantees on the solutions obtained. SMC solvers offer useful tools for many real-world problems at the nexus of symbolic and statistical AI. Extensive experiments on two real-world applications in AI for social good demonstrate that XOR-SMC outperforms other approaches in solution quality and running time.

Acknowledgments

We thank all the reviewers for their constructive comments. This research was supported by NSF grant CCF-1918327.

References

2023. *AAAI-23 Constraint Programming and Machine Learning Bridge Program.*

2023. *Neuro-symbolic AI for Agent and Multi-Agent Systems (NeSyMAS) Workshop at AAMAS-23.*

Achlioptas, D.; and Theodoropoulos, P. 2017. Probabilistic Model Counting with Short XORs. In *SAT*, volume 10491, 3–19. Springer.

Almeida, R.; Shi, Q.; Gomes-Selman, J. M.; Wu, X.; Xue, Y.; Angarita, H.; Barros, N.; Forsberg, B.; García-Villacorta, R.; Hamilton, S.; Melack, J.; Montoya, M.; Perez, G.; Sethi, S.; Gomes, C.; and Flecker, A. 2019. Reducing greenhouse gas emissions of Amazon hydropower with strategic dam planning. *Nature Communications*, 10.

Amideo, A. E.; Scaparra, M. P.; and Kotiadis, K. 2019. Optimising shelter location and evacuation routing operations: The critical issues. *Eur. J. Oper. Res.*, 279(2): 279–295.

Balog, M.; Tripuraneni, N.; Ghahramani, Z.; and Weller, A. 2017. Lost Relatives of the Gumbel Trick. In *ICML*, volume 70, 371–379. PMLR.

Barbosa, H.; Barrett, C. W.; Brain, M.; Kremer, G.; Lachnitt, H.; Mann, M.; Mohamed, A.; Mohamed, M.; Niemetz, A.; Nötzli, A.; Ozdemir, A.; Preiner, M.; Reynolds, A.; Sheng, Y.; Tinelli, C.; and Zohar, Y. 2022. cvc5: A Versatile and Industrial-Strength SMT Solver. In *TACAS*, volume 13243, 415–442. Springer.

Barrett, C.; Sebastiani, R.; Seshia, S.; and Tinelli, C. 2009. *Satisfiability Modulo Theories*, chapter 26, 825–885.

Barrett, C. W.; Sebastiani, R.; Seshia, S. A.; and Tinelli, C. 2021. Satisfiability Modulo Theories. In *Handbook of Satisfiability*, volume 336, 1267–1329. IOS Press.

Bayram, V.; and Yaman, H. 2018. Shelter Location and Evacuation Route Assignment Under Uncertainty: A Benders Decomposition Approach. *Transp. Sci.*, 52(2): 416–436.

Belanger, D.; and McCallum, A. 2016. Structured Prediction Energy Networks. In *ICML*, volume 48, 983–992.

Biere, A.; Heule, M. J. H.; van Maaren, H.; and Walsh, T., eds. 2009. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications.*

Bjørner, N. S.; de Moura, L.; Nachmanson, L.; and Wintersteiger, C. M. 2018. Programming Z3. In *SETSS*, volume 11430, 148–201. Springer.

Braunstein, A.; Mézard, M.; and Zecchina, R. 2005. Survey propagation: an algorithm for satisfiability. *Random Struct. Algorithms*, 27: 201–226.

Chakraborty, S.; Fremont, D. J.; Meel, K. S.; Seshia, S. A.; and Vardi, M. Y. 2014. Distribution-Aware Sampling and Weighted Model Counting for SAT. In *AAAI*, 1722–1730.

Chakraborty, S.; Meel, K. S.; and Vardi, M. Y. 2013. A Scalable and Nearly Uniform Generator of SAT Witnesses. In *CAV*, volume 8044, 608–623. Springer.

Chavira, M.; and Darwiche, A. 2008. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7): 772–799.

Cheng, Q.; Chen, F.; Dong, J.; Xu, W.; and Ihler, A. 2012. Approximating the Sum Operation for Marginal-MAP Inference. In *AAAI*, 1882–1887.

Choi, A.; Kisa, D.; and Darwiche, A. 2013. Compiling Probabilistic Graphical Models Using Sentential Decision Diagrams. In *ECSQARU*, volume 7958, 121–132. Springer.

Choi, Y.; Friedman, T.; and den Broeck, G. V. 2022. Solving Marginal MAP Exactly by Probabilistic Circuit Transformations. In *AISTATS*, volume 151, 10196–10208. PMLR.

Conitzer, V.; and Sandholm, T. 2006. Computing the optimal strategy to commit to. In *EC*, 82–90. ACM.

Conrad, J.; Gomes, C. P.; van Hoeve, W.-J.; Sabharwal, A.; and Suter, J. F. 2012. Wildlife corridors as a connected subgraph problem. *Journal of Environmental Economics and Management*, 63(1).

Darwiche, A.; and Marquis, P. 2002. A Knowledge Compilation Map. *J. Artif. Int. Res.*

de Moura, L. M.; and Bjørner, N. S. 2007. Efficient E-Matching for SMT Solvers. In *CADE*, volume 4603 of *Lecture Notes in Computer Science*, 183–198. Springer.

de Moura, L. M.; and Bjørner, N. S. 2008. Z3: An Efficient SMT Solver. In *TACAS*, volume 4963, 337–340. Springer.

Dilkina, B.; and Gomes, C. P. 2010. Solving Connected Subgraph Problems in Wildlife Conservation. In *CPAIOR*, 102–116.

Dutra, R.; Laeufer, K.; Bachrach, J.; and Sen, K. 2018. Efficient sampling of SAT solutions for testing. In *ICSE*, 549–559. ACM.

Dyer, M. E.; and Stougie, L. 2006. Computational complexity of stochastic programming problems. *Math. Program.*, 106(3): 423–432.

Ermon, S.; Gomes, C. P.; Sabharwal, A.; and Selman, B. 2013a. Embed and Project: Discrete Sampling with Universal Hashing. In *NIPS*, 2085–2093.

Ermon, S.; Gomes, C. P.; Sabharwal, A.; and Selman, B. 2013b. Taming the Curse of Dimensionality: Discrete Integration by Hashing and Optimization. In *ICML*, volume 28, 334–342.

Fang, F.; Stone, P.; and Tambe, M. 2015. When Security Games Go Green: Designing Defender Strategies to Prevent Poaching and Illegal Fishing. In *IJCAI*.

Feser, J. K.; Madden, S.; Tang, N.; and Solar-Lezama, A. 2020. Deductive optimization of relational data storage. *Proc. ACM Program. Lang.*, 4(OOPSLA): 170:1–170:30.

Flerova, N.; Ihler, E.; Dechter, R.; and Otten, L. 2011. Mini-bucket elimination with moment matching. In *NIPS Workshop DISCML*.

Fredrikson, M.; and Jha, S. 2014. Satisfiability modulo counting: a new approach for analyzing privacy properties. In *CSL-LICS*, 42:1–42:10. ACM.

Geman, S.; and Geman, D. 1984. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6): 721–741.

Gogate, V.; and Dechter, R. 2007. SampleSearch: A Scheme that Searches for Consistent Samples. In *AISTATS*, volume 2, 147–154.

Gogate, V.; and Dechter, R. 2012. Importance sampling-based estimation over AND/OR search spaces for graphical models. *Artif. Intell.*, 184-185: 38–77.

Gomes, C. P.; Sabharwal, A.; and Selman, B. 2006a. Model Counting: A New Strategy for Obtaining Good Bounds. In *AAAI*, 54–61.

Gomes, C. P.; Sabharwal, A.; and Selman, B. 2006b. Near-Uniform Sampling of Combinatorial Spaces Using XOR Constraints. In *NIPS*, 481–488. MIT Press.

Hazan, T.; and Jaakkola, T. S. 2012. On the Partition Function and Random Maximum A-Posteriori Perturbations. In *ICML*.

Hazan, T.; and Shashua, A. 2010. Norm-Product Belief Propagation: Primal-Dual Message-Passing for Approximate Inference. *IEEE Trans. Inf. Theory*, 56(12): 6294–6316.

Israeli, E.; and Wood, R. K. 2002. Shortest-path network interdiction. *Networks: An International Journal*, 40(2): 97–111.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparameterization with Gumbel-Softmax. In *ICLR (Poster)*.

Jerrum, M.; and Sinclair, A. 1997. *The Markov chain Monte Carlo method: an approach to approximate counting and integration*, 482–520. Boston, MA, USA.

Jerrum, M.; Valiant, L. G.; and Vazirani, V. V. 1986. Random Generation of Combinatorial Structures from a Uniform Distribution. *Theor. Comput. Sci.*, 43: 169–188.

Jiang, J.; Rai, P.; and III, H. D. 2011. Message-Passing for Approximate MAP Inference with Latent Variables. In *NIPS*, 1197–1205.

K., H. G. V.; Shoham, S.; and Gurfinkel, A. 2022. Solving constrained Horn clauses modulo algebraic data types and recursive functions. *Proc. ACM Program. Lang.*, 6: 1–29.

Kempe, D.; Kleinberg, J.; and Tardos, É. 2005. Influential nodes in a diffusion model for social networks. In *Automata, languages and programming*, 1127–1138. Springer.

Kisa, D.; den Broeck, G. V.; Choi, A.; and Darwiche, A. 2014. Probabilistic Sentential Decision Diagrams. In *KR*.

Kiselev, I.; and Poupart, P. 2014. Policy optimization by marginal-map probabilistic inference in generative models. In *AAMAS*, 1611–1612. IFAAMAS/ACM.

Kleywegt, A. J.; Shapiro, A.; and Homem-de-Mello, T. 2002. The Sample Average Approximation Method for Stochastic Discrete Optimization. *SIAM J. Optim.*, 12(2): 479–502.

Korhonen, T.; and Järvisalo, M. 2021. Integrating tree decompositions into decision heuristics of propositional model counters (short paper). In *27th International Conference on Principles and Practice of Constraint Programming (CP 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Kuck, J.; Dao, T.; Zhao, S.; Bartan, B.; Sabharwal, A.; and Ermon, S. 2019. Adaptive Hashing for Model Counting. In *UAI*, volume 115, 271–280. AUAI Press.

Lee, J.; Marinescu, R.; Dechter, R.; and Ihler, A. 2016. From Exact to Anytime Solutions for Marginal MAP. In *AAAI*, 3255–3262.

Liu, Q.; and Ihler, A. 2012. Belief Propagation for Structured Decision Making. In *UAI*, 523–532. AUAI Press.

Liu, Q.; and Ihler, A. T. 2013. Variational algorithms for marginal MAP. *Journal of Machine Learning Research*, 14.

Madras, N. 2002. *Lectures on Monte Carlo Methods*. American Mathematical Society.

Maneva, E. N.; Mossel, E.; and Wainwright, M. J. 2007. A new look at survey propagation and its generalizations. *J. ACM*, 54(4): 17.

Marinescu, R.; Dechter, R.; and Ihler, A. T. 2014. AND/OR Search for Marginal MAP. In *UAI*.

Mauá, D. D.; and de Campos, C. P. 2012. Anytime Marginal MAP Inference. In *ICML*.

Mei, J.; Jiang, Y.; and Tu, K. 2018. Maximum A Posteriori Inference in Sum-Product Networks. In *AAAI*, 1923–1930.

Munawar, A.; Lenchner, J.; Rossi, F.; Horesh, L.; Gray, A.; and Campbell, M., eds. 2023. *IBM Neuro-Symbolic AI Workshop 2023 – Unifying Statistical and Symbolic AI*.

Papandreou, G.; and Yuille, A. L. 2010. Gaussian sampling by local perturbations. In *NIPS*, 1858–1866.

Papandreou, G.; and Yuille, A. L. 2011. Perturb-and-MAP random fields: Using discrete optimization to learn and sample from energy models. In *ICCV*, 193–200.

Park, J. D.; and Darwiche, A. 2004. Complexity Results and Approximation Strategies for MAP Explanations. *J. Artif. Int. Res.*

Ping, W.; Liu, Q.; and Ihler, A. 2015. Decomposition Bounds for Marginal MAP. In *NIPS*, 3267–3275.

Rossi, F.; van Beek, P.; and Walsh, T., eds. 2006. *Handbook of Constraint Programming*, volume 2. Elsevier.

Shapiro, A. 2003. Monte Carlo sampling methods. *Handbooks in operations research and management science*, 10: 353–425.

Sheldon, D.; Dilkina, B.; Elmachtoub, A. N.; Finseth, R.; Sabharwal, A.; Conrad, J.; Gomes, C. P.; Shmoys, D. B.; Allen, W.; Amundsen, O.; and Vaughan, W. 2010. Maximizing the Spread of Cascades Using Network Design. In *UAI*, 517–526.

Sontag, D.; Meltzer, T.; Globerson, A.; Jaakkola, T.; and Weiss, Y. 2008. Tightening LP Relaxations for MAP using Message Passing. In *UAI*, 503–510.

Soos, M.; Gocht, S.; and Meel, K. S. 2020. Tinted, Detached, and Lazy CNF-XOR Solving and Its Applications to Counting and Sampling. In *CAV*, volume 12224 of *Lecture Notes in Computer Science*, 463–484. Springer.

Swamy, C.; and Shmoys, D. B. 2006. Approximation Algorithms for 2-Stage Stochastic Optimization Problems. In *FSTTCS*, volume 4337, 5–19. Springer.

Valiant, L. G.; and Vazirani, V. V. 1986. NP is as Easy as Detecting Unique Solutions. *Theor. Comput. Sci.*, 47(3): 85–93.

Verweij, B.; Ahmed, S.; Kleywegt, A. J.; Nemhauser, G. L.; and Shapiro, A. 2003. The Sample Average Approximation Method Applied to Stochastic Routing Problems: A Computational Study. *Comput. Optim. Appl.*, 24(2-3): 289–333.

Wainwright, M. J.; Jaakkola, T. S.; and Willsky, A. S. 2003. Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudo-moment matching. In *AISTATS*. Society for Artificial Intelligence and Statistics.

Wainwright, M. J.; and Jordan, M. I. 2008. Graphical Models, Exponential Families, and Variational Inference. *Found. Trends Mach. Learn.*, 1(1-2): 1–305.

Welling, M.; and Teh, Y. W. 2003. Approximate inference in Boltzmann machines. *Artif. Intell.*, 143(1): 19–50.

Williams, J. C.; and Snyder, S. A. 2005. Restoring Habitat Corridors in Fragmented Landscapes using Optimization and Percolation Models. *Environmental Modeling and Assessment*, 10(3): 239–250.

Wu, X.; Kumar, A.; Sheldon, D.; and Zilberstein, S. 2017. Robust Optimization for Tree-Structured Stochastic Network Design. In *AAAI*, 4545–4551.

Wu, X.; Sheldon, D.; and Zilberstein, S. 2014. Stochastic Network Design in Bidirected Trees. In *NIPS*, 882–890.

Wu, X.; Sheldon, D.; and Zilberstein, S. 2015. Fast Combinatorial Algorithm for Optimizing the Spread of Cascades. In *IJCAI*, 2655–2661.

Xue, S.; Fern, A.; and Sheldon, D. 2015. Scheduling Conservation Designs for Maximum Flexibility via Network Cascade Optimization. *J. Artif. Intell. Res.*, 52: 331–360.

Xue, Y.; Choi, A.; and Darwiche, A. 2012. Basing Decisions on Sentences in Decision Diagrams. In *AAAI*, 842–849.

Xue, Y.; Li, Z.; Ermon, S.; Gomes, C. P.; and Selman, B. 2016. Solving Marginal MAP Problems with NP Oracles and Parity Constraints. In *NIPS*, 1127–1135.

Yedidia, J. S.; Freeman, W. T.; and Weiss, Y. 2000. Generalized Belief Propagation. In *NIPS*, 689–695. MIT Press.

Zokaee, S.; Jabbarzadeh, A.; Fahimnia, B.; and Sadjadi, S. J. 2017. Robust supply chain network design: an optimization model with real world application. *Annals of Operations Research*, 257: 15–44.