Solving Satisfiability Modulo Counting Problems in Computational Sustainability with Guarantees

Jinzhao Li, Nan Jiang, Yexiang Xue
Department of Computer Science
Purdue University
West Lafayette, Indiana, 47906
{li4255, jiang631, yexiang}@cs.purdue.edu

Abstract

Many real-world problems in computational sustainability require tight integrations of symbolic and statistical AI. Interestingly, Satisfiability Modulo Counting (SMC) captures a wide variety of such problems. 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, replacing the model counting in SMC with SAT formulae subject to randomized XOR constraints. Experiments on solving important SMC problems in computational sustainability demonstrate that `XOR-SMC` finds solutions close to the true optimum, outperforming several baselines which struggle to find good approximations for the intractable model counting in SMC.

1 Introduction

Solving real-world problems in computational sustainability requires tight integrations of symbolic and statistical 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 [25, 6, 45].

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 application in computational sustainability, stochastic connectivity optimization searches for the optimal plan to reinforce the network structure so its connectivity preserves 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 can be formulated as SMC and is useful for disaster preparation [61], bio-diversity protection [19], internet resilience [29], social influence maximization [32], energy security [2], etc. It requires symbolic reasoning (satisfiability) to decide which roads to reinforce and where to place emergency shelters, and statistical inference

Computational Sustainability Workshop at NeurIPS 2023.

(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 [58, 16, 50, 60] 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) [46] – still intractable even with good satisfiability solvers [7, 48, 9] and model counters [27, 22, 1, 11, 33, 13, 26]. Previous research on SMC solves either a special case or domain-specific applications [5, 57, 64, 56, 23, 15, 51]. The special case is called the Marginal Maximum-A-Posterior (MMAP) problem, whose decision version can be formulated as a special case of SMC [42, 40, 43, 31, 37, 47] Both cases are solved by optimizing the surrogate representations of the intractable model counting in variational forms [39, 34], or via knowledge compilation [14, 47, 44] or via sample average approximation [35, 49, 52, 50, 20, 59, 62, 54].

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*** (full version in [38]). 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 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 [53, 30, 28, 27, 22, 21, 36, 1, 11, 10]. 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.

2 Preliminaries

2.1 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 [4]. Over the years, many good SMT solvers are built, such as the Z3 [18, 8] and cvc5 [3]. They play a crucial role in automated theorem proving, program analysis [24], program verification [55], and software testing [17].

2.2 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 [12, 63].

2.3 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:

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

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. 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 [53, 30] and has been developed by a rich line of work [28, 27, 22, 21, 36, 1, 11, 10]. This idea has recently gathered momentum thanks to the rapid progress in SAT solving [41, 9]. The contribution of this proposal extends the success of SAT solvers to problems with even higher complexity, namely, NP^{PP}-complete SMC problems.

3 Problem Formulation

Satisfiability Modulo Counting (SMC) is Satisfiability Modulo Theory (SMT) [4] with model counting as the background theory. A canonical definition of the SMC problem is to determine if there exists $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X} = \{0,1\}^n$ and $\mathbf{b} = (b_1, \ldots, b_k) \in \{0,1\}^k$ that satisfies the formula:

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

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.

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:

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

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.

4 The XOR-SMC Algorithm

The key motivation behind our proposed XOR-SMC algorithm is to notice that the XOR-Counting Algorithm presented in Section 2.3 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 1).

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

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

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

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

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 2.3). 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.

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 the full Algorithm in [38], 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***. Please see the full paper for the algorithm's pseudo-code. We prove XOR-SMC has a constant approximation guarantee in Theorem 1:

Theorem 1. *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 $1 - \eta$.

References

[1] Dimitris Achlioptas and Panos Theodoropoulos. Probabilistic model counting with short xors. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 3–19. Springer, 2017.

[2] R. Almeida, Qinru Shi, Jonathan M. Gomes-Selman, Xiaojian Wu, Yexiang Xue, H. Angarita, N. Barros, B. Forsberg, R. García-Villacorta, S. Hamilton, J. Melack, M. Montoya, Guillaume Perez, S. Sethi, C. Gomes, and A. Flecker. Reducing greenhouse gas emissions of amazon hydropower with strategic dam planning. *Nature Communications*, 10, 2019.

[3] Haniel Barbosa, Clark Barrett, Martin Brain, Gereon Kremer, Hanna Lachnitt, Makai Mann, Abdalrhman Mohamed, Mudathir Mohamed, Aina Niemetz, Andres Nötzli, et al. cvc5: A versatile and industrial-strength smt solver. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 415–442. Springer, 2022.

[4] Clark Barrett, Roberto Sebastiani, Sanjit Seshia, and Cesare Tinelli. *Satisfiability Modulo Theories*, chapter 26, pages 825–885. 2009.

[5] David Belanger and Andrew McCallum. Structured prediction energy networks. In *International Conference on Machine Learning*, pages 983–992, 2016.

[6] Vaishak Belle, Michael Fisher, Xiaowei Huang, Masoumeh Mansouri, Albert Merono-Penuela, Sriraam Natarajan, and Efi Tsamoura, editors. *Neuro-symbolic AI for Agent and Multi-Agent Systems (NeSyMAS) Workshop at AAMAS-23*, 2023.

[7] Armin Biere, Marijn J. H. Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, February 2009.

[8] Nikolaj Bjørner, Leonardo de Moura, Lev Nachmanson, and Christoph M Wintersteiger. Programming z3. *Engineering Trustworthy Software Systems: 4th International School, SETSS 2018, Chongqing, China, April 7–12, 2018, Tutorial Lectures 4*, pages 148–201, 2019.

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

[10] Supratik Chakraborty, Daniel J. Fremont, Kuldeep S. Meel, Sanjit A. Seshia, and Moshe Y. Vardi. Distribution-aware sampling and weighted model counting for sat. In *AAAI*, 2014.

[11] Supratik Chakraborty, Kuldeep S. Meel, and Moshe Y. Vardi. A scalable and nearly uniform generator of sat witnesses. In *Proceedings of the 25th International Conference on Computer Aided Verification*, 2013.

[12] Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799, 2008.

[13] Qiang Cheng, Feng Chen, Jianwu Dong, Wenli Xu, and Alexander Ihler. Approximating the sum operation for marginal-map inference. In *Conf. on Artificial Intelligence (AAAI)*, pages 1882–1887. July 2012.

[14] YooJung Choi, Tal Friedman, and Guy Van den Broeck. Solving marginal map exactly by probabilistic circuit transformations. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, mar 2022.

[15] Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proc. 7th ACM Conference on Electronic Commerce (EC)*, pages 82–90, 2006.

[16] John Conrad, Carla P. Gomes, Willem-Jan van Hoeve, Ashish Sabharwal, and Jordan F. Suter. Wildlife corridors as a connected subgraph problem. *Journal of Environmental Economics and Management*, 63(1), 2012.

[17] Leonardo De Moura and Nikolaj Bjørner. Efficient e-matching for smt solvers. In *Automated Deduction–CADE-21: 21st International Conference on Automated Deduction Bremen, Germany, July 17-20, 2007 Proceedings 21*, pages 183–198. Springer, 2007.

[18] Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer, 2008.

[19] Bistra Dilkina and Carla P. Gomes. Solving connected subgraph problems in wildlife conservation. In *CPAIOR*, pages 102–116, 2010.

[20] Martin Dyer and Leen Stougie. Computational complexity of stochastic programming problems. *Mathematical Programming*, 106(3):423–432, 2006.

[21] Stefano Ermon, Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Embed and project: Discrete sampling with universal hashing. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[22] Stefano Ermon, Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Taming the curse of dimensionality: Discrete integration by hashing and optimization. In *Proceedings of the 30th International Conference on Machine Learning, ICML*, 2013.

[23] Fei Fang, Peter Stone, and Milind Tambe. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *IJCAI*, 2015.

[24] John Feser, Sam Madden, Nan Tang, and Armando Solar-Lezama. Deductive optimization of relational data storage. *Proceedings of the ACM on Programming Languages*, 4(OOPSLA):1–30, 2020.

[25] Eugene C. Freuder and Barry O'Sullivan, editors. *AAAI-23 Constraint Programming and Machine Learning Bridge Program*, 2023.

[26] Vibhav Gogate and Rina Dechter. Importance sampling-based estimation over and/or search spaces for graphical models. *Artificial Intelligence*, 184-185:38 – 77, 2012.

[27] Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Model counting: A new strategy for obtaining good bounds. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[28] Carla P Gomes, Ashish Sabharwal, and Bart Selman. Near-uniform sampling of combinatorial spaces using xor constraints. In *Advances in Neural Information Processing Systems*, 2007.

[29] Eitan Israeli and R Kevin Wood. Shortest-path network interdiction. *Networks: An International Journal*, 40(2):97–111, 2002.

[30] M.R. Jerrum, L.G. Valiant, and V.V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[31] Jiarong Jiang, Piyush Rai, and Hal Daumé III. Message-passing for approximate MAP inference with latent variables. In *Advances in Neural Information Processing Systems 24*, 2011.

[32] David Kempe, Jon Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *Automata, languages and programming*, pages 1127–1138. Springer, 2005.

[33] Doga Kisa, Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Probabilistic sentential decision diagrams. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2014.

[34] Igor Kiselev and Pascal Poupart. Policy optimization by marginal-map probabilistic inference in generative models. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1611–1612. Citeseer, 2014.

[35] Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12:479–502, February 2002.

[36] Jonathan Kuck, Tri Dao, Shengjia Zhao, Burak Bartan, Ashish Sabharwal, and Stefano Ermon. Adaptive hashing for model counting. In *Conference on Uncertainty in Artificial Intelligence*, 2019.

[37] Junkyu Lee, Radu Marinescu, Rina Dechter, and Alexander T. Ihler. From exact to anytime solutions for marginal MAP. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI*, 2016.

[38] Jinzhao Li, Nan Jiang, and Yexiang Xue. Solving satisfiability modulo counting for symbolic and statistical ai integration with provable guarantees. *ArXiv, https://arxiv.org/abs/2309.08883*, 2023.

[39] Qiang Liu and Alexander Ihler. Belief propagation for structured decision making. In *Uncertainty in Artificial Intelligence (UAI)*, pages 523–532. AUAI Press, Corvallis, Oregon, August 2012.

[40] Qiang Liu and Alexander T. Ihler. Variational algorithms for marginal MAP. *Journal of Machine Learning Research*, 14, 2013.

[41] E. Maneva, Elchanan Mossel, and M. Wainwright. A new look at survey propagation and its generalizations. *ArXiv*, cs.CC/0409012, 2005.

[42] Radu Marinescu, Rina Dechter, and Alexander T. Ihler. AND/OR search for marginal MAP. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI*, 2014.

[43] Denis Deratani Mauá and Cassio Polpo de Campos. Anytime marginal MAP inference. In *Proceedings of the 29th International Conference on Machine Learning, ICML*, 2012.

[44] Jun Mei, Yong Jiang, and Kewei Tu. Maximum A posteriori inference in sum-product networks. In *AAAI*, pages 1923–1930. AAAI Press, 2018.

[45] Asim Munawar, Jon Lenchner, Francesca Rossi, Lior Horesh, Alexander Gray, and Murray Campbell, editors. *IBM Neuro-Symbolic AI Workshop 2023 – Unifying Statistical and Symbolic AI*, 2023.

[46] James D. Park and Adnan Darwiche. Complexity results and approximation strategies for map explanations. *J. Artif. Int. Res.*, 2004.

[47] Wei Ping, Qiang Liu, and Alexander T. Ihler. Decomposition bounds for marginal MAP. In *Advances in Neural Information Processing Systems 28*, 2015.

[48] Francesca Rossi, Peter van Beek, and Toby Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., New York, NY, USA, 2006.

[49] Alexander Shapiro. Monte carlo sampling methods. *Handbooks in operations research and management science*, 10:353–425, 2003.

[50] Daniel Sheldon, Bistra Dilkina, Adam Elmachtoub, Ryan Finseth, Ashish Sabharwal, Jon Conrad, Carla Gomes, David Shmoys, William Allen, Ole Amundsen, and William Vaughan. Maximizing the spread of cascades using network design. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 517–526, 2010.

[51] Daniel Sheldon, Bistra N. Dilkina, Adam N. Elmachtoub, Ryan Finseth, Ashish Sabharwal, Jon Conrad, Carla P. Gomes, David B. Shmoys, William Allen, Ole Amundsen, and William Vaughan. Maximizing the spread of cascades using network design. In *UAI*, 2010.

[52] Chaitanya Swamy and David B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. In *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science*, pages 5–19, 2006.

[53] L. G. Valiant and V. V. Vazirani. Np is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.

[54] Bram Verweij, Shabbir Ahmed, Anton J. Kleywegt, George Nemhauser, and Alexander Shapiro. The sample average approximation method applied to stochastic routing problems: A computational study. *Computational Optimization and Applications*, 24(2-3):289–333, 2003.

[55] Hari Govind VK, Sharon Shoham, and Arie Gurfinkel. Solving constrained horn clauses modulo algebraic data types and recursive functions. *Proc. ACM Program. Lang.*, 6(POPL):1–29, 2022.

[56] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, January 2008.

[57] Max Welling and Yee Whye Teh. Approximate inference in boltzmann machines. *Artificial Intelligence*, 143(1):19–50, 2003.

[58] Justin C. Williams and Stephanie A. Snyder. Restoring habitat corridors in fragmented landscapes using optimization and percolation models. *Environmental Modeling and Assessment*, 10(3):239–250, 2005.

[59] Xiaojian Wu, Akshat Kumar, Daniel Sheldon, and Shlomo Zilberstein. Robust optimization for tree-structured stochastic network design. In *Proceedings of the 31st Conference on Artificial Intelligence*, 2017.

[60] Xiaojian Wu, Daniel Sheldon, and Shlomo Zilberstein. Stochastic network design in bidirected trees. In *Advances in Neural Information Processing Systems*, pages 882–890, 2014.

[61] Xiaojian Wu, Daniel Sheldon, and Shlomo Zilberstein. Fast combinatorial algorithm for optimizing the spread of cascades. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2655–2661, 2015.

[62] Shan Xue, Alan Fern, and Daniel Sheldon. Scheduling conservation designs for maximum flexibility via network cascade optimization. *J. Artif. Intell. Res. (JAIR)*, 2015.

[63] Yexiang Xue, Zhiyuan Li, Stefano Ermon, Carla P. Gomes, and Bart Selman. Solving marginal map problems with np oracles and parity constraints. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS)*, 2016.

[64] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Generalized belief propagation. In *Advances in neural information processing systems*, pages 689–695, 2001.