A Flat Histogram Method for Computing the Density of States of Combinatorial Problems*

Stefano Ermon, Carla Gomes, Bart Selman
Department of Computer Science
Cornell University
{ermonste,gomes,selman}@cs.cornell.edu

Abstract

Consider a combinatorial state space $\mathcal{S}$, such as the set of all truth assignments to N Boolean variables. Given a partition of $\mathcal{S}$, we consider the problem of estimating the size of all the subsets in which $\mathcal{S}$ is divided. This problem, also known as computing the density of states, is quite general and has many applications. For instance, if we consider a Boolean formula in CNF and we partition according to the number of violated constraints, computing the density of states is a generalization of both SAT, MAX-SAT and model counting.

We propose a novel Markov Chain Monte Carlo algorithm to compute the density of states of Boolean formulas that is based on a flat histogram approach. Our method represents a new approach to a variety of inference, learning, and counting problems. We demonstrate its practical effectiveness by showing that the method converges quickly to an accurate solution on a range of synthetic and real-world instances.

1 Introduction

Consider a combinatorial state space $\mathcal{S}$, such as the set $\{0,1\}^N$ of all possible truth assignments to N Boolean variables. Given a partition of $\mathcal{S}$ into subsets, we consider the problem of estimating the size of all the subsets in the partition. This problem is also known as computing the density of states. For instance, given a Boolean formula in CNF with m clauses, we can define an "energy" function in terms of the number of violated constraints and partition the set of all possible truth assignments accordingly. In this case, the density of states gives the size of all the subsets defined by the number of violated constraints, i.e., the number of truth assignments that violate exactly k clauses, for $0 \leq k \leq m$. The problem of computing the density of states is a generalization of SAT, MAX-SAT and model counting. The term density of states name is borrowed from statistical physics, where given an energy function the density of states (DOS) for a physical system describes the number of microstates at each energy level that are available to be occupied. In this paper, we will extend this concept to the case of any energy function that defines an interesting partition of the state space.

The information provided by the full density of states distribution is especially useful in the context of probabilistic models defined through combinatorial constraints such as Markov Logic Theories [12]. In fact, the description of the state space can be used to efficiently compute not only the normalization constant of the underlying probabilistic model (also known as the *partition function*), but also its *parameterized version*. This level of abstraction is a fundamental advantage for learning methods because it allows us to reason about the system more abstractly. For example, as we will discuss below, in the case of a Markov Logic Theory, we can parameterize the partition function $Z(w_1, \dots, w_K)$ according to the weights $w_1, \dots, w_K$ of its K first order formulas that define the theory. Upon defining an appropriate energy function and obtaining the corresponding density of states, we can use the information about the partition function to directly compute the model parameters that best fit the training data.

To compute the DOS, we will consider a novel MCMC sampling strategy, inspired by the Wang-Landau method [16], which is a *flat histogram* method from statistical physics. Given a combinatorial space and an energy function, a *flat histogram* method is a sampling strategy based on a Markov Chain that adaptively changes its transition probabilities until it converges to a steady state where it will spend approximately the same amount of time in areas with low-density configurations (usually, low energy states) as in high-density areas of the search space. This condition leads to a flat histogram of the energy levels visited (hence the name of the method).

We empirically demonstrate that our method converges quickly and accurately on a range of instances. Our results are very promising and we expect to see other applications both to counting, learning, and inference problems.

2 Density of states: problem definition

We consider a state space $\{0,1\}^N$ defined as the set of all possible truth assignments to N Boolean variables $x_1, \dots, x_N$. Given such a space and an energy function $E : \{0,1\}^N \to \mathbb{N}$, the *density of states* (DOS) n is a function $n : range(E) \to \mathbb{N}$ that maps energy levels to the number of truth assignments

*This paper is a condensed version of [4].

(configurations) with that energy, i.e.,

$$n(k) = |\{\sigma \in \{0,1\}^N | E(\sigma) = k\}|.$$

So, we are interested in computing the number truth assignments (also called possible worlds in probabilistic reasoning contexts) that satisfy certain properties that are specified using an energy function. For instance, given a Boolean formula in CNF, we might define the *energy* of a configuration $E(\sigma)$ to be the number of constraints that are unsatisfied by σ (this is known as the density of states for unweighted Boolean formulas [4]).

The MCMC algorithm we will use to compute the density of states does not make any assumption about the actual values of the energy. In principle, the only thing we need is a partitioning of the state space, where the energy is just providing an index over the subsets that compose the partition.

Probabilistic reasoning. The notion of the density of states idea finds many natural applications in the context of probabilistic reasoning and Markov Logic Networks (MLNs) [12]. MLNs are used to define complex probability distributions over a set of *possible worlds* which are truth assignments to a set of N Boolean variables. The probability is specified through weighted constraints that are represented as (CNF) formulas over the Boolean variables. Such constraints can be either *hard* or *soft*. Hard constraints are such that if a truth assignment $\sigma \in \{0,1\}^N$ violates one of them, it gets assigned probability 0. Otherwise, when σ satisfies all hard constraints, its probability is given by

$$P(\sigma) = \frac{1}{Z(w)} \exp\left(-\sum_{i \in \mathcal{C}} w_i \chi_i(\sigma)\right) \tag{1}$$

where $\mathcal{C}$ is the set of soft constraints, w_i is the weight corresponding to the i-th *soft* constraint and $\chi_i(\sigma) = 1$ if and only if σ violates the i-th constraint, and $Z(w)$ is the normalization constant also referred to as the *partition function*.

In such a framework, many inference (e.g., computing the probability of σ) and learning (e.g., finding the weights that maximize the likelihood of x) tasks require the computation of the normalization constant $Z(w)$. This is in general an intractable problem because $Z(w) = \sum_\sigma \exp\left(-\sum_{i \in \mathcal{C}} w_i \chi_i(\sigma)\right)$ is defined as a sum over exponentially many states.

A key advantage of the density of states idea is that it can be used to efficiently compute not only the partition function, but also its *parameterized version*. For example, in the case of a Markov Logic Theory, we can parameterize the partition function $Z(w_1, \dots, w_K)$ according to the weights $w_1, \dots, w_K$ of its K first order formulas. We can use this parameterized version of $Z(w)$ to directly compute the optimal set of weights that provide the best fit to the training data.

3 Preliminaries on MCMC methods

Sampling from a combinatorial space refers to the process of generating samples from a probability distribution π defined over a large (but finite) set Σ, such as the set of possible configurations of a physical system or the set of all possible truth assignments to a Boolean formula.

Sampling from a combinatorial space has applications to probabilistic inference, learning, and counting. In a counting problem, the goal is to estimate the cardinality of a set Σ or a weighted sum of the form $\sum_{\sigma \in \Sigma} w(\sigma)$, where w is a weight assigned to the elements of Σ. The general idea is that we can often obtain an approximate solution to a counting problem by looking at the statistical properties of a sequence of independent random samples from a distribution π defined over Σ (see [7] for details). The close connection between counting and sampling has been formally studied in a complexity theoretical sense in [8].

The Markov Chain Monte Carlo (MCMC) method is a general approach to combinatorial sampling that has received increasing attention over the past ten years, leading to major results for counting problems in graphs (matchings, independent sets, colorings, and homomorphisms) [7].

The Markov Chain Monte Carlo method solves the sampling problem by constructing a Markov Chain with state space Σ and stationary distribution π. The transition probabilities $p_{\sigma \to \sigma'}$ for $\sigma, \sigma' \in \Sigma$ are designed so that the Markov Chain asymptotically converges to the stationary distribution π, regardless of the initial state [10]. Typically, this is accomplished by enforcing a condition known as *detailed balance*, which defines a reversible Markov Chain such that $\pi(\sigma) p_{\sigma \to \sigma'} = \pi(\sigma') p_{\sigma' \to \sigma}$ for all $\sigma, \sigma' \in \Sigma$ (a probability distribution π that satisfies the detailed balance condition can be proved to be stationary). Moreover, transitions correspond to simple modifications of the elements of Σ (such as variable flipping), so that the chain can be efficiently simulated, such as a random walk on a graph.

Given such a Markov Chain, the sampling process works as follows. Given an arbitrary initial state $\sigma \in \Sigma$, we run (simulate) the chain for a finite number T of steps, and we output the final state. The probability distribution of the output can be made arbitrarily close to π by taking a large enough T.

4 The flat histogram method

We propose a Markov Chain Monte Carlo method to compute the density of states based on the flat histogram idea inspired by recent developments of statistical physics [16] as an alternative to Metropolis sampling.

The idea behind the method is as follows. The goal is to construct a reversible Markov Chain on the space of all truth assignments $\{0,1\}^N$ such that the steady state probability of a truth assignment σ is inversely proportional to the density of states $n(E(\sigma))$. In this way, the stationary distribution is such that all the energy levels are visited equally often (i.e., when we count the visits to each energy level, we see a flat visit histogram). Specifically, we define a Markov Chain with the following transition probability

$$p_{\sigma \to \sigma'} = \begin{cases} \frac{1}{N} \min\left\{1, \frac{n(E(\sigma))}{n(E(\sigma'))}\right\} & d_H(\sigma, \sigma') = 1 \\ 0 & d_H(\sigma, \sigma') > 1 \end{cases} \tag{2}$$

where $d_H(\sigma, \sigma')$ is the Hamming distance between σ and σ'. The probability of a self-loop $p_{\sigma \to \sigma}$ is defined by the normalization constraint $p_{\sigma \to \sigma} + \sum_{\sigma' | d_H(\sigma, \sigma') = 1} p_{\sigma \to \sigma'} = 1$. The detailed balance equation $P(\sigma) p_{\sigma \to \sigma'} = P(\sigma') p_{\sigma' \to \sigma}$ is satisfied by $P(\sigma) \propto 1/n(E(\sigma))$. This means[1] that the Markov Chain will reach a stationary probability distribution P (regardless of the initial state) such that the probability of a truth assignment σ with energy $E = E(\sigma)$ is inversely proportional to the total number of truth assignments with energy E. This leads to an asymptotically flat histogram of the energies of the states visited because $P(E) = \sum_{\sigma:E(\sigma)=E} P(\sigma) \propto n(E)\frac{1}{n(E)} = 1$ (i.e., independent of E).

Algorithm 1 `MCMC-FlatSAT`

```
Start with a guess $g(E) = 1$ for all $E = 1, \ldots, m$
Initialize $H(E) = 0$ for all $E = 1, \ldots, m$
Start with a modification factor $F = F_0$
**repeat**
  Randomly pick a truth assignment $\sigma$
  **repeat**
    Generate a new assignment $\sigma'$ (by flipping a variable)
    Let $E = E(\sigma)$ and $E' = E(\sigma')$
    Set $\sigma = \sigma'$ with probability $\min\left\{1, \frac{g(E)}{g(E')}\right\}$
    Let $E_c = E(\sigma)$ be the current energy level
    Adjust the density $g(E_c) = g(E_c) \times F$
    Update visit histogram $H(E_c) = H(E_c) + 1$
  **until** until $H$ is flat
  Reduce $F$
  Reset the visit histogram $H$
**until** $F$ is close enough to 1
Normalize $g$
**return** $g$ as estimate of $n$
```

Since the density of states is unknown a priori, and computing it is precisely the goal of the algorithm, it is not possible to directly set up a Markov Chain with transition probability (2). However it is possible to start from an initial guess of the DOS and keep changing the current estimate g in a systematic way using a modification factor F (used as a multiplier to adapt the density estimate g) to produce a flat histogram of the energy levels visited and simultaneously make the estimated density of states converge to the true value $n(E)$.

The modification factor F plays a critical role because it controls the trade-off between the convergence rate of the algorithm and its accuracy. Large initial values of F imply a substantial diffusion rate and therefore fast convergence to a rather inaccurate solution. This rough initial estimate is subsequently refined as the value of F decreases until $F \approx 1$, at which point when a flat histogram is produced $g(E)$ has converged to the true density $n(E)$.

While in [4] we generated new states by randomly flipping a variable, greedier strategies can lead to faster convergence rates [3]. In particular, we have shown that introducing a random walk component (i.e., flipping variables from violated clauses with a higher probability) can significantly improve the convergence rates [3].

Due to statistical fluctuations, a perfectly flat histogram occurs with an extremely low probability. Therefore in our implementation we use a flatness parameter; in our experiments it is set so that an histogram is considered flat when all the values are between 90% and 100% of the maximum value, independently of F. The value of F is reduced according to the schedule $F \leftarrow \sqrt{F}$, with an initial value $F_0 = 1.5$. By construction the DOS is obtained only up to a constant factor: we therefore normalize g to ensure that $\sum_E g(E) = 2^N$, where N is the number of variables in the formula.

Technically, `MCMCFlatSAT` is an example of an Adaptive Markov Chain Monte Carlo method. In an Adaptive MCMC scheme, the transition probabilities are adjusted over time in order to achieve some optimality condition, learning the parameters while the chain runs [13]. This approach is closely related to Simulated Annealing [9; 15], Simulated Tempering [11], and Multicanonical Sampling [2]. The main contribution of the flat histogram idea is that it can simultaneously learn the optimal weights and at the same time sample from the re-weighted distribution. However, even though Adaptive MCMC algorithms can significantly improve the performance over standard MCMC methods, it is usually harder to rigorously prove convergence properties.

5 Effectiveness and validation of `MCMC-FlatSat`

Despite the increasing popularity in statistical physics applications, the asymptotic properties of flat histogram methods are not yet fully understood [1]. Under some regularity assumptions (which are generally hard to verify) it is possible to prove certain formal results on the consistency of the method [1; 17], but little is known about the rate of convergence and the efficiency of the method. The goal of this section is to verify the convergence of `MCMC-FlatSat` and to empirically evaluate the accuracy of the solution obtained for energy functions defined through combinatorial constraints.

For all our experiments, we consider formulas F in CNF over a set V of variables with m clauses. We say that a variable assignment σ satisfies a clause C if at least one literal of C is TRUE. A literal is a variable or its negation. We define the energy of a configuration $E(\sigma)$ to be the number of clauses that are unsatisfied when F is evaluated under σ. If $E(\sigma) = 0$, then σ satisfies F and σ is called a model, solution, a ground state or satisfying assignment for F.

We first empirically check the accuracy of the results obtained for small structured formulas, for which we can compute the true density by exact enumeration of the entire (exponentially large) state space. We also test `MCMC-FlatSat` on larger synthetic formulas for which we derive an analytical expression for the true density of states, as well as on random 3-SAT formulas. For larger structured instances, for which no known method can be used to compute the true DOS, we make use of partial consistency checks to validate the results.

When the true DOS is known, we employ several metrics to evaluate the accuracy of the results. We consider the relative error for each data point and two global measures represented by the Kullback-Leibler (K-L) divergence and the Total Variation distance. The Kullback-Leibler divergence is a standard

[1]The chain is finite, irreducible, and aperiodic, therefore ergodic.

Instance	variables	clauses	KL-divergence $D_{KL}(n\|\|g)$	Total Variation $\delta(n,g)$	Max relative error
ram_k3_n7.ra0.cnf	21	70	4.0×10^{-5}	0.0038	2.3 %
ram_k3_n8.ra0.cnf	28	126	1.2×10^{-5}	0.0019	5.1 %
johnson8-2-4.clq.cnf	28	420	4.6×10^{-5}	0.0039	5.5 %
t3pm3-5555.spn.cnf	27	162	1.3×10^{-5}	0.0020	3.1 %
Synth. formula-Uniform	50	100	1.2×10^{-5}	0.0021	3.0 %
Synth. formula-PigeonHole	200	750	1.2×10^{-7}	0.0006	2.2 %

Table 1: Exact and estimated density of states compared in terms of KL-divergence and maximum relative error.

information theoretic non-symmetric measure defined as:

$$D_{KL}(n||g) = \sum_{E=0}^{m} \frac{n(E)}{Z} \log_2 \left(\frac{n(E)}{g(E)} \right)$$

where $Z = 2^n$ is used to normalize the DOS to probability distributions. The Total Variation distance is often used to evaluate the convergence of MCMC methods and is defined as

$$\delta(n,g) = \frac{1}{2}||n-g||_1 = \frac{1}{2}\sum_E |g(E) - n(E)|$$

5.1 Structured problems: exact counts

We compare the true and estimated densities for several small instances (all with less than 28 variables) from the MAXSAT-2007 competition benchmark, which are encodings of three classes of problems (Ramsey Problems, Spin Glasses, and Max Cliques). The true density is computed by exact enumeration.

Our experiments show that the estimate is accurate and that the relative error per count obtained by MCMC-FlatSat compared to the exact count for a given energy level is never greater than 5.5%. The high degree of accuracy obtained is confirmed by the Kullback-Leibler divergences presented in Table 1. The sampling strategy is also very efficient. For instance, for the instance *johnson8-2-4.clq.cnf* it takes about 8×10^6 flips for a search space of size $2^{27} \approx 1.3 \times 10^8$.

5.2 Synthetic formulas: exact analytic counts

Consider a formula F which is the logical conjunction of ℓ copies of another formula ϕ, each one involving a different set of variables $x_1, \dots, x_\ell$:

$$F(x_1, \dots, x_\ell) = \phi(x_1) \wedge \phi(x_2) \wedge \dots \wedge \phi(x_\ell).$$

By noticing that the subformulas in F do not share variables, it is easy to see that the density of states $n_F(E)$ of F can be computed as a nested convolution of n_ϕ:

$$n_F(E) = (n_\phi * \dots * n_\phi)(E), \qquad (3)$$

where $*$ is the convolution operator. This result is analogous to the fact that the probability density function (PDF) of the sum of independent random variables is equal to the convolution of the PDFs of the addends (concentrating the measure on the mean).

We test the effectiveness of MCMC-FlatSat on large synthetic instances, for which exact enumeration would not be possible, by comparing the estimated DOS with the analytical results given by (3). In particular, we use the conjunction of formulas ϕ that are encoding of a Pigeon Hole problem (Synth. formula-PigeonHole), and the conjunction of formulas ϕ that have a uniform density (Synth. formula-Uniform). The ground truth is obtained by computing the density of ϕ by direct enumeration and then carrying over the convolutions according to Equation (3). When comparing the exact density with the estimate obtained by running MCMC-FlatSat directly on the large formula, we get a relative error that is never greater than 3%, as confirmed by the small Kullback-Leibler divergences reported in Table 1. The sampling strategy defined by MCMC-FlatSat is very efficient, since it needs about 2×10^7 samples to collect fine grained statistics about a much larger the state space ($2^{50} \approx 10^{15}$ truth assignments) .

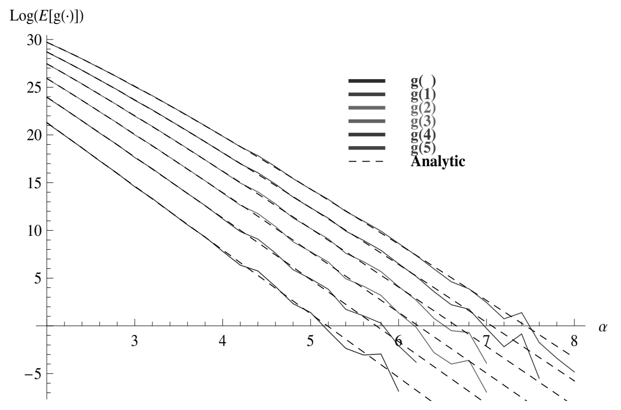
5.3 Random formulas

In this section we present the results of MCMC-FlatSat for random 3-SAT formulas as a function of the ratio clauses to variables α. In particular, we compute the average DOS over 1000 random instances for each value of α in the range considered.

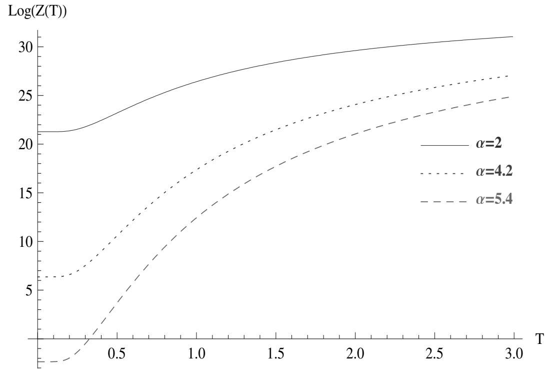
Notice that the average DOS ($\mathbb{E}[g(i)]$) for random k-SAT formulas is given by

$$\mathbb{E}[g(i)] = \binom{m}{i} \left(\frac{1}{2^k}\right)^i \left(1 - \frac{1}{2^k}\right)^{m-i} 2^n \qquad (4)$$

where i is the number of violated constraints and m is the total number of clauses. This is because given a random truth assignment σ, the probability of σ violating a clause of size k is $1/2^k$. Therefore, by the linearity of expectation we obtain formula (4). The comparison with the analytic result (4) in Figure 1(a) confirms the good accuracy of MCMC-FlatSat. Moreover, as shown in figure 1(b), the density of states gives not only the number of models (i.e., $n(0)$), but also the the log-partition function $\log Z(T) = \log\left(\sum_\sigma e^{-\frac{1}{T}E(\sigma)}\right)$ *for all the values* of T. The temperature T can be thought as the *softness* of the constraints or, alternatively, we can define it in terms of the *hardness* $w = 1/T$ with the notation used in (1). In the limit $\lim_{T \to 0} Z(T)$, $Z(T)$ counts the number of models (i.e., as the constraints become hard we recover the traditional number of satisfying assignments). However, the fact that we can compute it as a function of an external parameter (in this case the *softness* T) turns out to be fundamental for probabilistic reasoning applications, such as weight learning in Markov Logic Networks [3].



(a) Average DOS.



(b) Log partition function $\log Z(T)$.

Figure 1: Average DOS (a) and log-partition function as a function of the temperature parameter T (b). The number of variables is $n = 50$ (see pdf for color version of figures).

5.4 Large structured instances

In this section we present the results obtained on large structured formulas for which the exact DOS is unknown and direct enumeration is not feasible. Given that we are not aware of any complete solver that is able to compute the exact DOS, we need to resort to partial consistency checks to assess the accuracy of `MCMC-FlatSat`. In particular, when it is possible, we compare $g(0)$ with the exact model count given by a complete solver such as Cachet ([14]).

A further consistency check can be obtained by looking at the moments of the DOS. Intuitively, the moments represent a quantitative measure of the shape of a set of points and therefore they can be used to check that the probability mass is concentrated in the right regions. The k-th order moment is defined as:

$$M(k) = \sum_E E^k \frac{g(E)}{Z}$$

where $Z = 2^n$ is again used to normalize to a probability distribution. For example, $M(1)$ is the average number of violated clauses by a random assignment. This value is compared with the *sample k-th moment*

$$M_s(k) = \frac{1}{\ell} \sum_{i=1}^{\ell} E(X_i)^k$$

where $X_1, X_2, ..., X_\ell$ are samples drawn uniformly from the set of all truth assignments $\{0,1\}^N$.

The results presented in Table 2 show a good agreement with exact model counters and *sample k-th moments*. The sampling strategy is again quite efficient. For instance, in the case of *bw_large.a* it can sample a very large state space (2^{459} truth assignments) in just a few hours.

5.5 Model counting

To further demonstrate the scalability and the accuracy of our method, we test its effectiveness as a method to compute the number of models of a formula. In terms of the density of states, this is equivalent to the problem of computing the value of $n(0)$. We compare the performance of `MCMC-FlatSat` with two state-of-the-art approximate model counters: SampleCount [6] and SampleMiniSATExact [5]. The instances used are taken from the benchmark used in [5; 6]. The results in Table 3 show that `MCMC-FlatSat` almost always obtains more accurate solution counts, and is often significantly faster, especially for random instances.

6 Conclusions and Future Work

We described `MCMC-FlatSat`, a Markov Chain Monte Carlo technique based on the flat histogram method to estimate the density of states for combinatorial energy functions defined on Boolean hypercubes. The density of states provides a deep characterization of the state space. In fact, it gives not only the partition function of the model (e.g., number of models), but also its parameterized version (e.g., for all the levels of hardness of the constraints). This type of information is crucial in many probabilistic reasoning applications, both for learning and inference tasks [3].

We demonstrated the effectiveness of `MCMC-FlatSat`, both in terms of convergence and accuracy, on a broad range of structured and synthetic instances. Because of the generality and the effectiveness of the flat histogram idea, we expect that this approach will find many other applications both in counting and probabilistic inference applications.

7 Acknowledgments

This research is funded by NSF Expeditions in Computing grant 0832782.

References

[1] Y.F. Atchadé and J.S. Liu. The Wang-Landau algorithm in general state spaces: applications and convergence analysis. *Statistica Sinica*, 20:209–233, 2010.

Instance	var	clauses	$g(0)$	# models	$M_s(1)$	$M(1)$	$M_s(2)$	$M(2)$
brock400_2.clq.cnf	40	1188	0	0	297.0	297.0	88366	88372
spinglass5_10.pm3.cnf	125	750	0	0	187.5	187.5	35249	35247
MANN_a27.clq.cnf	42	1690	0	0	422.5	422.5	178709	178703
bw_large.a.cnf	459	4675	1	1	995.3	995.3	996349	996634
hole10.cnf	110	561	0	0	137.5	137.5	19621	19643
sw100-1.cnf	500	3100	8.04×10^{27}		753.1	753.1	571718	571863

Table 2: Comparison of the moments. Sample moments estimated with $\ell = 10^6$ uniformly sampled truth assignments. Exact model counting is done with Cachet.

Instance	n	m	Exact #	SampleCount		SampleMiniSAT		MCMC-FlatSat	
				Models	Time	Models	Time	Models	Time
2bitmax_6	252	766	2.10×10^{29}	$\geq 2.40 \times 10^{28}$	29	2.08×10^{29}	345	1.96×10^{29}	1863
wff-3-3.5	150	525	1.40×10^{14}	$\geq 1.60 \times 10^{13}$	240	1.60×10^{13}	145	1.34×10^{14}	393
wff-3.1.5	100	150	1.80×10^{21}	$\geq 1.00 \times 10^{20}$	240	1.58×10^{21}	128	1.83×10^{21}	21
wff-4-5.0	100	500		$\geq 8.00 \times 10^{15}$	120	1.09×10^{17}	191	8.64×10^{16}	189
ls8-norm	301	1603	5.40×10^{11}	$\geq 3.10 \times 10^{10}$	1140	2.22×10^{11}	168	5.93×10^{11}	2693

Table 3: Comparison with model counters. Timings for SampleCount and SampleMiniSATExact are taken from the respective papers. `MCMC-FlatSat` timings are obtained on a comparable $3GHz$ machine.

[2] B.A. Berg and T. Neuhaus. Multicanonical ensemble: A new approach to simulate first-order phase transitions. *Physical Review Letters*, 68(1):9–12, 1992.

[3] S. Ermon, C. Gomes, A. Sabharwal, and B. Selman. A flat histogram method for inference with probabilistic and deterministic constraints. *NIPS Workshop on Monte Carlo Methods for Modern Applications*, 2010.

[4] S. Ermon, C. Gomes, and B. Selman. Computing the density of states of Boolean formulas. In *Proc. of CP-2010*, 2010.

[5] V. Gogate and R. Dechter. Approximate counting by sampling the backtrack-free search space. In *Proc. of AAAI-07*, pages 198–203, 2007.

[6] C.P. Gomes, J. Hoffmann, A. Sabharwal, and B. Selman. From sampling to model counting. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, 2007.

[7] Mark Jerrum and Alistair Sinclair. *The Markov chain Monte Carlo method: an approach to approximate counting and integration*, pages 482–520. PWS Publishing Co., Boston, MA, USA, 1997.

[8] 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.

[9] S. Kirkpatrick, CD Gelatt Jr, and MP Vecchi. Optimization by simulated annealing. *Science (New York, NY)*, 220(4598):671, 1983.

[10] N.N. Madras. *Lectures on Monte Carlo Methods*. Amer Mathematical Society, 2002.

[11] E. Marinari and G. Parisi. Simulated tempering: a new Monte Carlo scheme. *EPL (Europhysics Letters)*, 19:451, 1992.

[12] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1):107–136, 2006.

[13] G.O. Roberts and J.S. Rosenthal. Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2):349–367, 2009.

[14] T. Sang, F. Bacchus, P. Beame, H. Kautz, and T. Pitassi. Combining component caching and clause learning for effective model counting. In *Proc. of SAT*, 2004.

[15] D. Štefankovič, S. Vempala, and E. Vigoda. Adaptive simulated annealing: A near-optimal connection between sampling and counting. *Journal of the ACM (JACM)*, 56(3):18, 2009.

[16] F. Wang and DP Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*, 86(10):2050–2053, 2001.

[17] Chenggang Zhou and R. N. Bhatt. Understanding and improving the wang-landau algorithm. *Phys. Rev. E*, 72(2):025701, Aug 2005.