However, we also show that highly structured energy landscapes can actually present new opportunities for solution samplers. Combinatorially defined energy functions have a rich structure embedded, but traditional MCMC methods are very general and therefore treat the energy as a “black-box”, effectively ignoring the underlying structure. On the other hand, modern day constraint solvers use clever heuristic to exploit constraint structure as much as possible, and can solve very large structured industrial problems with millions of variables [10]. However, these solvers cannot be used directly as solution samplers because they tend to oversample certain solutions, as they are designed just to find one satisfying assignment but not to be uniform [8].

In this paper, we propose a novel sampling scheme called SearchTreeSampler, which leverages the reasoning power of a systematic constraint solver while enforcing a uniform exploration of the search space. Constraint solvers have been previously applied by SampleSearch [11, 12, 13] in the context of importance sampling, a framework where the performance is known to heavily depend on the choice of the proposal distribution. In contrast, SearchTreeSampler introduces a new way of exploring the search space that does not rely on a heuristically chosen proposal distribution, and directly provides (approximately) uniform samples. The constraint solver is used as a black-box, so that any systematic solver can be plugged in, with no modifications required. We empirically demonstrate that by leveraging constraint structure, SearchTreeSampler can overcome many of the difficulties encountered by other solution samplers. In particular, we show it can be orders of magnitude faster than competing methods, while providing more uniform samples at the same time. Further, we show that our sampling scheme naturally defines a new technique for approximately counting the number of distinct solutions (model counting), that we empirically show to be very accurate on a range of benchmark problems.

2 Problem Definition

We consider the problem of sampling from a combinatorial search space defined by $n$ Boolean variables and $m$ constraints specified by a Boolean formula $F$ in conjunctive normal form (CNF). A constraint or clause $C$ is a logical disjunction of a set of (possibly negated) variables. A formula $F$ is said to be in CNF form if it is a logical conjunction of a set of clauses $C$.

We define $V$ to be the set of propositional variables in the formula, where $|V| = n$. A variable assignment $\sigma : V \rightarrow \{0, 1\}$ is a function that assigns a value in $\{0, 1\}$ to each variable in $V$. As usual, the value 0 is interpreted as FALSE and the value 1 as TRUE. Let $F$ be a formula in CNF over the set $V$ of variables with $m = |C|$ clauses and let $\sigma$ be a variable or truth assignment. We say that $\sigma$ satisfies a clause $C$ if at least one signed variable of $C$ is TRUE. We say that a truth assignment $\sigma$ is a satisfying assignment for $F$ (also called a model or a solution) if $\sigma$ satisfies all the clauses $C \in C$. Let $S_F$ be the set of solutions of $F$, and let $Z = |S_F|$ be the number of distinct solutions.

Given a Boolean formula $F$, we define a discrete probability distribution $D$ over the set of all possible truth assignments $\{0, 1\}^n$ such that

$$D(\sigma) = \begin{cases} 1/Z & \text{if } \sigma \in S_F \text{ (i.e., } \sigma \text{ is a solution)} \\ 0 & \text{otherwise} \end{cases}$$

In this paper we consider the problem of sampling from $D$. This problem is very hard and in fact simply deciding whether or not the support of $D$ is empty is NP-complete (the CNF-SAT problem). Sampling is however believed to be even harder, as it is closely related to $\#P$-complete problems such as inference and model counting [3, 4]. For instance, SAT solvers cannot be directly used as solution samplers because they tend to oversample certain solutions, as they are designed just to find one satisfying assignment but not to be uniform [8].

3 Background on Solution Sampling

In this section, we briefly describe the main techniques for solution sampling.

3.1 Simulated Annealing

Simulated Annealing is a MCMC algorithm that defines a reversible Markov Chain on the space of truth assignments $\{0, 1\}^n$ to sample from a Boltzmann distribution [14]. The transition probabilities (and the steady state probability distribution) depend only on a property of the truth assignments called “energy”. The energy $E : \{0, 1\}^n \rightarrow \mathbb{N}$ gives the number of clauses violated by a truth assignment $\sigma$ and is defined as follows

$$E(\sigma) = |\{c \in C | \sigma \text{ does not satisfy } c\}|.$$

The Boltzmann steady state probability distribution is given by

$$P_T(\sigma) = \frac{1}{Z(T)} e^{-\frac{E(\sigma)}{T}},$$

where $T$ is a formal parameter called “temperature”, and $Z(T)$ is the normalization constant. Notice that $P_T$ assigns the same probability to all solutions, i.e. $P_T(\sigma) = \alpha$ for all $\sigma \in S_F$, but is not necessarily zero for non-solutions. Given an algorithm $\mathcal{A}$ that produces samples from $P_T$, we can construct an algorithm $\mathcal{B}$ that samples from $D$ (i.e. uniformly from the solution set) using rejection sampling. More specifically, we take samples $s$ produced by $\mathcal{A}$ and we discard all the ones such that $s \notin S_F$. The fundamental tradeoff involved is that we want the probability distribution $P_T$ to be easier to sample from (compared to $D$), but