モンテカルロ木探索を利用した制約付き分布からの効率的なサンプリング法

Sampling from constrained distributions using Monte Carlo Tree Search

石畠正和 [1] * 佐藤泰介 [1]
Masakazu Ishihata[1] Taisuke Sato[1]

[1] 東京工業大学 大学院情報理工学研究科
[1] Graduate School of Information Science and Engineering, Tokyo Institute of Technology

Abstract: We propose *MC3TS*, which is a Markov chain Monte Carlo method (MCMC) based on Monte Carlo tree search (MCTS), for sampling from constrained distributions. We apply MC3TS to distributions constrained by random SAT problem and the s-t path problem. We experimentally show that MC3TS acheives better perform than existing methods.

1 はじめに

確率モデリングはデータ中の不確実性や曖昧性を表現する方法として広く利用されており、関係データや自然言語、バイオデータなどの複雑な構造を持つ記号データ（離散データ）を効率的に扱うため、論理に基づく確率モデリングが提案されている [6, 4]。確率モデリングにおける重要なタスクとして期待値計算が挙げられる。しかし、離散確率変数上の分布において、期待値計算は一般に指数的な時間を要する。論理に基づく確率モデリングでは、論理制約によって条件付けられた分布である **制約付き分布** を考える場合が多く、その場合は制約を満たす空間のみを考える。しかし、制約を満たす空間を効率よく列挙することは難しく、更に列挙できてもその数が膨大であればやはり期待値計算は効率的に行えない。そこで本研究では制約付き分布からの効率的なサンプリング法を提案する。制約付き分布からのサンプリング法はいくつか提案されているが、どの手法もなんらかの制限が存在する。例えば constraint-based slice sampling (CSS) [17] は制約を CNF 式に限定している。また BDD-based sampling (BBS) [11] は制約を表現する二分決定図 (binary decision diagram; BDD) を必要とし、BDD が構成できない場合は適用できない。SampleSearch [7] は目的の制約付き分布からのサンプリングの代わりに、重要度重み付きサンプリング (importance sampling) を提案しているが、目的分布からのサンプリングが必要な場合はこれは利用できない。

*連絡先: 東京工業大学 大学院情報理工学研究科
〒 152-8552 東京都目黒区大岡山 2-12-1
E-mail: ishihata@mi.cs.titech.ac.jp

本研究では一般的かつ効率的な制約付き分布からのサンプリングを実現するため、一般的なサンプリング法であるマルコフ連鎖モンテカルロ法 (Markov chain Monte Carlo; MCMC) と一般的な探索手法であるモンテカルロ木探索 (Monte Carlo tree seach; MCTS) を組み合わせた MC3TS 法を提案する。

2 準備

本章では本研究においてサンプリングの対象とする制約付き分布を定義する。確率的命題変数 b_i $(1 \le i \le N)$ が値 1 (true) および 0 (false) を取る確率をそれぞれ ψ_{i1}, ψ_{i0} とする。即ち、b_i はパラメータ $\psi_i \equiv (\psi_{i0}, \psi_{i1})$ のベルヌーイ分布 $\mathrm{Ber}(\psi_{i0}, \psi_{i1})$ より得られたとする。このとき $b \equiv (b_1, \ldots, b_N)$ の同時分布はパラメータ $\psi \equiv (\psi_1, \ldots, \psi_N)$ を用いて以下のように計算できる。

$$p(b \mid \psi) = \prod_{i=1}^{N} p(b_i \mid \psi_i) = \prod_{i=1}^{N} \psi_{i0}^{1-b_i} \psi_{i1}^{b_i}$$

f を b_i の論理式、$f(b)$ を f が表す論理関数とする。つまり f は確率事象 $\{b \mid f(b) = 1\}$ を表現する。よって f の確率は以下のように計算できる。

$$p(f \mid \psi) = \sum_{b} p(f, b \mid \psi) = \sum_{b} f(b) p(b \mid \psi)$$

ここで b に対する可能な代入は 2^N 通りあるため、一般的に周辺確率 $p(f \mid \psi)$ の計算量は $O(2^N)$ である。よって以下の条件付き確率の計算量も $O(2^N)$ である。

$$p(b \mid f, \psi) = \frac{f(b) p(b \mid \psi)}{p(f \mid \psi)}$$

上式の条件付き分布は、$f(b)=1$ を満たす b にのみ正の確率を与えることから、これを**制約付き分布** と呼ぶ。

一般にこの制約付き分布からサンプリングは困難である。何故ならこのサンプリング は SAT 問題を包含しているからである。また制約により、確率空間が複雑になることも原因の一つである。例えば制約付き分布 $p(b \mid f, \psi)$ に Gibbs sampling (GS) [5] を適用することを考える。GS を構成するには条件付き確率 $p(b_i \mid b_{-i}, f, \psi)$ を計算する必要があり、以下の様に容易に計算可能である。

$$\begin{aligned} p(b_i=v \mid b_{-i}, f, \psi) &\propto f(b_i=v) p(b_i=v, b_{-i} \mid \psi) \\ &= f(b_i=v, b_{-i}) \psi_{iv} \prod_{i' \neq i} p(b_{i'} \mid \psi_{i'}) \\ &\propto f(b_i=v, b_{-i}) \psi_{iv} \end{aligned}$$

しかしこの条件付き確率は GS が目的分布に収束するための条件であるエルゴード性を満たさない場合がある。例えば $f \equiv b_1 \oplus b_2$ とすれば、制約付き分布 $p(b \mid f, \psi)$ は $(b_1, b_2)=(1,0),(0,1)$ にのみ確率を与える。しかし、この 2 つの割り当て $(1,0),(0,1)$ は b_1 または b_2 の一方を変更するだけでは遷移できず、この f に対する GS はエルゴード性を満たさない。

本研究ではこの制約付き分布からの効率的なサンプリング法を提案する。ここですべての確率変数は 2 値であるとしたが、多値の確率変数を 2 値の確率変数にエンコードする方法が提案されているため [10]、本研究では 2 値の場合のみを考える。

3 既存手法

3.1 Monte Carlo Sampling

制約付き分布 $p(b \mid f, \psi)$ からの最も単純なサンプリング法は Monte Carlo sampling (MCS) である。今、$p(b \mid f, \psi)$ からのサンプリングは困難である一方、$p(b \mid \psi)$ からのサンプリングは非常に容易である。よって以下の棄却サンプリングにより $p(b \mid f, \psi)$ からのサンプリングが実現できる。

1. サンプル候補 $b \sim p(b \mid \psi)$ を得る。
2. $f(b)=1$ なら b を受理、そうでなければ棄却する。

この MCS の受理確率は $p(f \mid \psi)$ である。MCS は非常に簡単かつ高速である。しかし、$p(b \mid f, \psi)$ に関する期待値をサンプル近似する場合、$f(b)=1$ を満たすサンプルを大量に要するが、MCS は b 全体からサンプル候補を得るため、受理確率 $p(f \mid \psi)$ が小さいとき、十分な量のサンプルを得るのに膨大な時間を要する。また、MCS は繰り返し SAT 問題を解いているにもかかわらず、過去の知識を利用しないため、サンプリングを繰り返してもその受理確率は向上しない。

3.2 Constraint-based Slice Sampling

$f(b)=1$ を満たすサンプルを効率的に得る方法として、Constraint-based Slice Sampling (CSS) [17] が提案されている。CSS は Slice Sampling (SS) [14] を制約付き分布 $p(b \mid f, \psi)$ に適用した手法であり、slice よ呼ばれる新たな確率変数 $u \equiv (u_1, \ldots, u_N)$ を以下のように導入する。

$$\begin{aligned} p(f, b, u \mid \psi) &\equiv p(f \mid b)\, p(b, u \mid \psi) \\ p(b, u \mid \psi) &\equiv \prod_{i=1}^{N} p(b_i, u_i, \mid \psi_i) \\ p(b_i, u_i, \mid \psi_i) &\equiv \delta(0 \leq u_i \leq p(b_i \mid \psi_i)) \end{aligned}$$

よって条件付き分布 $p(u \mid b, f, \psi)$, $p(b \mid u, f, \psi)$ は以下のように計算できる。

$$\begin{aligned} p(u \mid b, f, \psi) &= \frac{p(b, u \mid \psi)}{p(b \mid \psi)} \\ &= \prod_{i=1}^{N} \frac{\delta(0 \leq u_i \leq p(b_i \mid \psi_i))}{p(b_i \mid \psi_i)} \\ p(b \mid u, f, \psi) &= \frac{f(b) p(b, u \mid \psi)}{\sum_b f(b) p(b, u \mid \psi)} \\ &= \frac{f(b) \prod_{i=1}^{N} \delta(0 \leq u_i \leq p(b_i \mid \psi_i))}{\sum_b f(b) \prod_{i=1}^{N} \delta(0 \leq u_i \leq p(b_i \mid \psi_i))} \end{aligned}$$

つまり、$b_i=v$ が与えらるとき、u_i は区間 $[0, \psi_{iv}]$ 上の一様分布に従う。一方、u が与えられるとき、b は $f(b)$ を満たし、かつ、各 $b_i=v$ において $u_i \leq \psi_{iv}$ を満たす空間上の一様分布に従う。新たに論理関数 $f_u(b)$ を以下のように定める。

$$f_u(b) \equiv f(b) \wedge \left(\bigwedge_{i \text{ s.t. } \psi_{i0} \leq u_i \leq \psi_{i1}} b_i \right) \wedge \left(\bigwedge_{i \text{ s.t. } \psi_{i1} \leq u_i \leq \psi_{i0}} \neg b_i \right)$$

このとき、CSS の手順は以下となる。

1. $f(b)=1$ なる初期値サンプル b を選ぶ。
2. u_i を区間 $[0, p(b_i \mid \psi_i)]$ 上の一様分布から得る。
3. b を集合 $\{b \mid f_u(b)=1\}$ 上の一様分布から得る。
4. 2.-3. を繰り返す。

ここで f_u を満たす空間からの一様サンプリングは制約付き分布からのサンプリングと同様、一般的には指数的な時間を要する。しかし、SampleSAT [16] と呼ばれる局所探索アルゴリズムを用いることで、近似的に制約を満たす空間からの一様分布が実現できる。

CSS の問題点はその速度と精度が、SampleSAT のそれに強く依存することである。実際に問題によっては SampleSAT の出力の偏りより、CSS の出力が安定しない場合がある。また、CSS は MCS と同様、繰り返し SAT 問題を解いているにもかかわらず、過去の知識を利用しない。

3.3 BDD-based Sampling

$p(b \mid f,\psi)$ からの正確なサンプリングを実現する方法として Binary Decision Diagram (BDD) [2] を用いた BDD-based sampling (BBS) が提案されている [11]。

今、$v=(v_1,\dots,v_N)$ $(v_i\in\{0,1\})$ とすれば、条件付き確率 $p(b=v \mid f,\psi)$ は以下のように分解可能である。

$$p(b=v|f,\psi)=\prod_{i=1}^{N} p\left(b_i=v_i \mid f\wedge\bigwedge_{j=1}^{i-1} b_j=v_j,\psi\right) \quad (1)$$

よって上式の各条件付き確率が計算可能であれば、制約付き分布 $p(b \mid f,\psi)$ からのサンプリングは逐次的に実行可能である。BBS はこの条件付き確率を効率よく計算するため BDD を利用する。

BDD は論理関数を圧縮表現する有向非循環グラフであり、変数 b_i でラベル付けられた変数節点と $v\in\{0,1\}$ でラベル付けられた v-定数節点からなる。論理関数 $f(b)$ を表現する BDD を Δ_f と書き、n_i を Δ_f 上のトポロジカル順序で i 番目の変数節点、そのラベルを l_i と書く。n_i は v-枝 $v\in\{0,1\}$ と呼ばれる出力枝を持ち、e_{iv} と書く。e_{iv} の指す節点を n_i の v-child と呼び、c_{iv} と書く。e_{iv} は変数 l_i が値 v をとることを表す。よって BDD 上のパスは b （もしくは b の一部）に対する値の割り当てを表す。BDD 上の根節点から始まるパスはすべて定数節点に至り、パスが v 定数節点に至るとき、そのパスが表現する b は $f(b)=v$ を満たす。BDD は再帰的シャノン展開により構成される。論理関数 $f(b)=(b_1,\dots,b_N)$ は常に以下のように分解できる。

$$\begin{aligned} f(b_1,\dots,b_N) &= (\neg b_1\wedge f(1,b_2,\dots,b_N)) \\ &\vee (b_1\wedge f(0,b_2,\dots,b_N)) \end{aligned}$$

上式を f の b_1 に関するシャノン展開、$f(v,b_2,\dots,b_N)$ を f の v-シャノン因子と呼ぶ。論理関数 f の変数 b_i に関する v-シャノン因子を $f_{|b_i=v}$ と書く。シャノン因子 $f_{|b_1=v}$ を表現する BDD $\Delta_{f_{|b_1=v}}$ が与えられたとする。このとき、Δ_f は、b_1 でラベル付けられ、v-child が $\Delta_{f_{|b_1=v}}$ の根を指す変数節点を構成することで得られる。よって、Δ_f の変数節点 n_i 以下の部分 BDD が表現する論理関数を f_{n_i} と書けば、以下成り立つ。

$$\begin{aligned} f_{n_i} &= \left(\neg l_i\wedge f_{n_i|l_i=0}\right)\vee\left(l_i\wedge f_{n_i|l_i=1}\right) \\ &= (\neg l_i\wedge f_{c_{i0}})\vee(l_i\wedge f_{c_{i1}}) \end{aligned}$$

つまり $l_i=b_j$ とすれば、周辺確率 $p(f_{n_i} \mid \psi)$ は以下のように再帰的に計算できる。

$$\begin{aligned} p(f_{n_i} \mid \psi) &= p(\neg l_i\wedge f_{c_{i0}} \mid \psi)+p(l_i\wedge f_{c_{i1}} \mid \psi) \\ &= \psi_{j0}p(f_{c_{i0}} \mid \psi)+\psi_{j1}p(f_{c_{i1}} \mid \psi) \end{aligned}$$

この再起計算を定式化するため、後ろ向き確率 $\mathrm{B}_v[n_i]$ $(v\in\{0,1\})$ を以下のように定義する。

$$\mathrm{B}_1[n_i]=p(f_{n_i} \mid \psi)\,, \qquad \mathrm{B}_0[n_i]=p(\neg f_{n_i} \mid \psi)$$

$\mathrm{B}_v[n_i]$ は以下のように BDD Δ_f 上の動的計画法として bottom-up に計算可能である。ただし $l_i=b_j$ とする。

$$\begin{aligned} \mathrm{B}_v[u] &\equiv \delta(u=v) \\ \mathrm{B}_v[n_i] &\equiv \sum_{u\in\{0,1\}}\psi_{ju}\mathrm{B}_v[c_{iu}] \end{aligned}$$

ここで $\mathrm{B}_v[u]$ は u-定数節点の後ろ確率である。

後ろ向き確率 $\mathrm{B}_v[n_i]$ を用いることで式 (1) からの逐次的なサンプリングは効率的に実行できる。まず、$f_{n_1}=f$ であることを利用すれば、$p(b_1=v_1 \mid f,\psi)$ は以下のように計算できる。

$$\begin{aligned} p(b_1=v_1 \mid f,\psi) &\propto p(b_1=v_1\wedge f_{n_1} \mid \psi) \\ &= p\big(b_1=v_1\wedge f_{n_1|b_1=v_1} \mid \psi\big) \\ &= \psi_{1v_1}\mathrm{B}_1[c_{iv_1}] \end{aligned}$$

つまり、v_1 は以下のベルヌーイ分布よりサンプリング可能である。

$$v_1\sim\mathrm{Ber}\left(\frac{\psi_{10}\mathrm{B}_1[c_{10}]}{\mathrm{B}_1[n_1]},\frac{\psi_{11}\mathrm{B}_1[c_{11}]}{\mathrm{B}_1[n_1]}\right)$$

よって $p(b \mid f,\psi)$ からのサンプリングは、n_i $(l_i=b_j)$ の v_j-child を以下のベルヌーイ分布によって逐次的に選択することで実現できる。

$$v_j\sim\mathrm{Ber}\left(\frac{\psi_{j0}\mathrm{B}_1[c_{i0}]}{\mathrm{B}_1[n_i]},\frac{\psi_{j1}\mathrm{B}_1[c_{i1}]}{\mathrm{B}_1[n_i]}\right) \quad (2)$$

上記の逐次計算ではいくつかの v_i を得られない場合がある。これは BDD の簡略化規則により f の値に影響を与えない変数 b_i は削除されるからである。つまり b_i のサンプル値 v_i が上記の逐次計算で得られなかったとき、以下のベルヌーイ分布から得ればよい。

$$v_i\sim\mathrm{Ber}(\psi_{i0},\psi_{i1}) \quad (3)$$

BBS の全体の手順は以下である。

1. BDD Δ_f を構築する。
2. 各節点の後ろ向き確率 $\mathrm{B}_v[n_i]$ を計算する。
3. $n:=n_1$ とする。
4. $n=n_i$, $l_i=b_j$ において、式 (2) より v_j をサンプルし、$n:=c_{iv_j}$ とする。
5. サンプルされていない v_i を式 (3) より得る。
6. $v=(v_1,\dots,v_N)$ を b のサンプルとする。

BBS は MCMC 法と異なり近似ではない正確なサンプリングである。更に BDD が与えられればその計算量は N に比例する。ただし、後ろ向き確率の計算量は Δ_f の節点数に比例する。しかし Δ_f が構築できない場合、この手法は実行できない。

4 提案手法

本章では制約付き分布 $p(b \mid f, \psi)$ からの新たなサンプリング法として MC3TS を提案する。MC3TS は一般的なサンプリング手法である Markov Chain Monte Carlo (MCMC) 法と一般的な探索手法である Monte Carlo Tree Search (MCTS) [1] を組み合わせたものである。MC3TS は BDD Δ_f の代わりに、MCTS によって構成される探索木を用いてサンプリングを行う。4.1 節で MCTS について簡単に説明し、4.2 節で提案手法である MC3TS について述べる。

4.1 Monte Carlo Tree Search

モンテカルロ木探索 (MCTS) はモンテカルロ法によって計算された評価尺度をもとに探索木を展開する探索手法の総称であり、ゲーム AI 分野を中心に非常に多くの分野で利用されている [1]。

例えば 2 人ゲームの例を考える。探索木の節点はゲームの盤面、枝はプレイヤーの行動に対応する。探索空間は非常に膨大であるため、有望な節点を優先的に展開したい。MCTS では各節点の有望度合いをランダムなモンテカルロシュミレーション（プレイアウト）により計算する。探索木の節点 n におけるプレイアウトの勝敗の数をそれぞれ $N_1[n], N_0[n]$ とする。このとき、最も単純な評価値は勝率 $N_1[n]/(N_1[n] + N_0[n])$ である。

最も有名な MCTS 法は Upper Confidence Bound for Trees (UCT) algorithm [13] である。この手法は節点 n の子節点 n' を以下の UCT 値を用いて評価する。

$$UCT(n') = \frac{N_1[n']}{N[n']} + 2C\sqrt{\frac{2\ln N[n]}{N[n']}}$$

ここで $N[n] \equiv N_1[n]+N_0[n]$、$C > 0$ は定数である。UCT の第一項は勝率、第二項はそのバリアンスを表す。

本研究ではこの MCTS をサンプリングに利用する。ここで MCTS の本来の目的は最適化であり、サンプリングではないことに注意する。

4.2 MC3TS

MC3TS は MCMC 法を構成する最も有名な方法である Metropolis-Hastings Sampling (MHS) [8] を制約付き分布 $p(b \mid f, \psi)$ に適用したものである。MHS はサンプリングが困難である目的分布の代わりに、サンプリングが容易な提案分布からのサンプリングを行い、得られたサンプルを適切な確率で受理することで、目的の分布からのサンプリングを達成する。MC3TS はこの提案分布を MCTS を用いて構成する。提案分布は BDD の代わりに MCTS により得られた探索木を用いてサンプリングを行う。BBS では BDD を用いて後ろ向き確率 $\mathrm{B}_v[n_i]$ を計算したが、ここでは $\mathrm{B}_v[n_i]$ の近似的な値 $\mathrm{B}_v^{\mathrm{MCTS}}[n_i]$ を MCTS により計算し、これを用いてサンプリングを実行する。

MCTS を $f(b) = 1$ なる b を発見する問題に適用することを考える。このとき探索木は $f(b)$ の二分決定木 (binary decision tree; BDT) に対応する。BDT の各節点は BDD の節点と同様、b_i でラベル付けられており、1-枝と 0-枝を持つ。MCTS では評価値の高い節点を決定的に選択し、BDT を構成する。この決定的な選択を確率的な選択に置き換えることで MCTS をサンプリング手法に拡張する。今、節点 n_i においてその子 c_{iv} $(v \in \{0, 1\})$ を選ぶことを考える。$l_i = b_j$ としたとき、c_{iv} を確率 ψ_{jv} で選べば、この MCTS は MCS と一致する。これに対し、c_{iv} を式 (2) によって選べば、これは BBS と一致する。しかし、$\mathrm{B}_v[n_i]$ を計算するには n_i 以下の節点をすべて展開する必要がある。そこで $\mathrm{B}_v[n_i]$ の近似的な値を考える。今、$N_v[n_i]$ を n_i からサンプリングで v に至った回数とする。$\mathrm{B}_v[n_i]$ は節点 n_i から v 定数節点に至る確率に対応することから、後ろ向き確率 $\mathrm{B}_v[n_i]$ は $N_v[n_i]/N[n_i]$ により近似できる。ここで $\mathrm{B}_v^{\mathrm{MCTS}}[n_i]$ を以下の様に定める。

$$\mathrm{B}_v^{\mathrm{MCTS}}[n_i] \equiv \begin{cases} 1 & n_i \text{ が未展開} \\ \dfrac{N_v[n_i] + c}{N[n_i] + 2c} & n_i \text{ が展開済みかつ未展開の子孫を持つ} \\ \mathrm{B}_v[n] & n_i \text{ 以下がすべて展開済み} \end{cases}$$

$\mathrm{B}_v^{\mathrm{MCTS}}[n_i]$ は $\mathrm{B}_v[n_i]$ の近似的な量である。ここで c は $\mathrm{B}_v[n_i] > 0$ にもかかわらず $\mathrm{B}_v^{\mathrm{MCTS}}[n_i] = 0$ となることを防ぐ擬似カウントである。

MC3TS は BBS の式 (2) において、$\mathrm{B}_v[n_i]$ の代わりに $\mathrm{B}_v^{\mathrm{MCTS}}[n_i]$ を用いた手法を提案分布とする MHS である。この提案分布は探索初期は MCS のように振る舞い、探索終了後は BBS と完全に一致する。つまり、MHS の受理確率は 100% となる。探索中盤では $\mathrm{B}_v[n_i]$ の近似値として、これまでのサンプリングより計算される経験的な後ろ向き確率を用いる。

MC3TS は木探索を用いるため、枝刈りの利用することでより効率的なサンプリングが可能となる。例えば f が CNF (conjunctive normal form) で与えられる場合、SAT ソルバーなどで利用される単位伝播 (unit propagation) [3] を枝刈りに利用できる。また別の枝刈り方法として、フロンティア法 (frontier based search; FBS) [12] がある。FBS は木探索において、展開パターンが一致する節点 (状態) を検知する手法である。FBS によって検知された等価な節点を共有することで、無駄な探索の繰り返しを回避できる。

5 実験

提案手法の有用性を示すため、2つの実験を行った。5.1 節では提案手法と既存手法をランダム SAT 問題に適用し、それらの精度と速度を比較した。5.2 節では提案手法を s-t パス問題に適用し、提案手法が BDD を構成するのが困難であるような複雑な問題に対しても適用可能であることを確認した。

5.1 ランダム SAT 問題

ランダム SAT 問題はランダムに生成された k-CNF 式の充足可能性を判定する問題である。本節ではランダムに生成された 3-CNF 式 f により定義される制約付き分布 $p(b \mid f, \psi)$ に対し各手法を適用し、その速度と精度を比較した。具体的な実験の設定は以下である。

1. 10 変数、C 節から成る 3-CNF f をランダムに生成する。
2. BDD Δ_f を構築し、周辺確率 $p(b_i \mid f, \psi)$ を厳密に計算する。
3. 各手法により $p(b \mid f, \psi)$ より 1,000 サンプルを生成し、周辺確率 $p(b_i \mid f, \psi)$ を推定する。
4. 3. を 100 回繰り返し、平均二乗誤差 (mean squared error; MSE) を計算する。
5. 1.-4. を各 $C = 10, 20, 30, 40, 50$ について 100 回ずつ実行する。

表 1 は生成された BDD の平均サイズ、平均構築時間および $p(f \mid \psi)$ の平均である。表 2,3 はそれぞれ各手法の平均 MSE と 1,000 サンプル得るのに要した平均時間である。MCS は正確なサンプリングであるため、非常に小さい MSE を達成しているのに対し、$p(f \mid \psi)$ が小さくなるに連れ、サンプリング時間が急増している。CSS は MCS と異なり、$f(b)=1$ なる b を直接サンプリングするため、$p(f \mid \psi)$ が小さくなっても MCS ほどの速度低下は起こらない。しかし、CSS は MCMC と SampleSAT の2つの近似の影響で MSE が他の手法に比べて非常に大きい。これに対して提案手法である MC3TS は MCS より高速、CSS とほぼ等速でより高精度なサンプリングを達成している。BBS は最も高速で高精度である。よって BDD が構築できる問題においては BBS が最も優れているといえる。

5.2 s-t パス問題

s-t パス問題とは、与えられたグラフの2点 s, t を結ぶパスをすべて列挙する問題である。例えば図 1 は 3×3 格子グラフに対する全 s-t パスである。Knuth はこの s-t パス問題を高速に解く手法として Simpath algorith を提案した [12]。Simpath は全 s-t パスの集合を表現する ZDD (Zero-suppressed BDD) [9] を効率的に構築するフロンティア法の一例である。近年、この手法は電力網のロス最小化問題などに応用されている [15]。

C	Size [# nodes]	Time [s]	$p(f \mid \psi)$
10	$1.927e{+}02$	$2.064e{-}03$	$3.383e{-}01$
20	$1.942e{+}02$	$2.165e{-}03$	$9.036e{-}02$
30	$1.263e{+}02$	$2.797e{-}03$	$3.068e{-}02$
40	$6.064e{+}01$	$2.820e{-}03$	$8.584e{-}03$
50	$3.552e{+}01$	$2.961e{-}03$	$4.793e{-}03$

表 1: BDD サイズ、構築時間および $p(f \mid \psi)$.

C	MCS	CSS	BBS	MC3TS
10	$1.185e{-}02$	$5.782e{-}01$	$1.172e{-}02$	$3.081e{-}01$
20	$1.091e{-}02$	$2.165e{+}00$	$1.086e{-}02$	$3.144e{-}01$
30	$9.224e{-}03$	$3.535e{+}00$	$9.484e{-}03$	$3.380e{-}01$
40	$6.027e{-}03$	$5.349e{+}00$	$6.163e{-}03$	$1.530e{-}02$
50	$4.464e{-}03$	$4.288e{+}00$	$4.581e{-}03$	$1.834e{-}02$

表 2: 各手法の MSE

$E \equiv \{e_1, \ldots, e_N\}$ を s-t パスを列挙したいグラフの辺の集合とする。ここで e_i は同時に命題変数であるとし、$e \equiv (e_1, \ldots, e_N)$ は辺の部分集合 $\{e_i \mid e_i = 1\}$ を表すとする。論理関数 $f(e)$ を e が s-t パスであれば 1 を、そうでなければ 0 を返す関数とする。更に $P \equiv \{e \mid f(e) = 1\}$ とすれば、s-t パス問題はこの P を求める問題である。しかし多くの場合、この P を求めること自体が目的ではなく、P の統計的性質や関数 $F(e)$ の P 上の期待値を計算することが目的となる。e_i をベルヌーイ分布 $\mathrm{Ber}(\psi_{i0}, \psi_{i1})$ に従う確率的命題変数とすれば、関数 F の P に関する期待値 $\mathrm{E}_P[F]$ は以下のように定義される。

$$\mathrm{E}_P[F] = \sum_{e \in P} F(e) p(e \mid f, \psi) \tag{4}$$

Simpath により P を表現する ZDD を構築できても、$|P|$ 自体は非常に巨大な数となり、P に関する期待値 $\mathrm{E}_P[F]$ や P の統計量の計算は容易ではない。例えば表 4 は $n{\times}n$ 格子グラフの $|E|$, $|P|$ および $\psi_{iv}=0.5$ としたときの $p(f \mid \psi)$ である。本実験では式 (4) をサンプリングにより近似することを目的とする。

今、関数 $L(e)$ を $L(e) \equiv |\{e_i \mid e_i = 1\}|$ と定め、すべての i について $\psi_{i1} = p$ とする。すると、制約付き分

C	MCS	CSS	BBS	MC3TS
10	$1.195e{-}03$	$3.204e{-}03$	$1.236e{-}03$	$7.521e{-}03$
20	$1.110e{-}02$	$7.023e{-}03$	$1.223e{-}03$	$1.136e{-}02$
30	$5.453e{-}01$	$1.510e{-}02$	$2.008e{-}03$	$1.280e{-}02$
40	$1.943e{+}00$	$3.318e{-}02$	$1.955e{-}03$	$1.037e{-}02$
50	$2.377e{+}01$	$6.972e{-}02$	$1.972e{-}03$	$1.161e{-}02$

表 3: 各手法のサンプリング時間 [s]

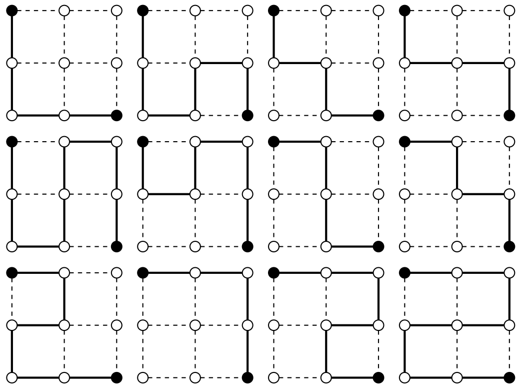

図 1: 3×3 格子グラフの全 s-t パス

布 $p(b \mid f, \psi)$ は以下のような分布となる。

$$p(e \mid f, \psi) \propto \left(\frac{p}{1-p}\right)^{L(e)}$$

ここで $L(e)$ は s-t パス e の長さに対応する。 本実験では $\mathrm{E}_P[L]$ を BBS および MC3TS により近似し、その精度と速度を比較する。ただし $p = 0.5$ とした。ここで $p(f \mid \psi)$ が非常に小さいため、 MCS は適用できない。また我々の知る限り $f(b)$ は CNF で表現でききたにため、 CSS もまた適用できない。ZDD の構築には Simpath を利用し、MC3TS の枝刈りにも Simpath を利用した。具体的な実験手順は以下である。

1. $n \times n$ 格子グラフに対する P を表現する ZDD を Simpath により構築する。
2. 得られた ZDD を用いて $\mathrm{E}_P[L]$ を厳密計算する。
3. 各手法を用い、$p(b \mid f, \psi)$ よりサンプルを T 個生成し、$\mathrm{E}_P[L]$ を近似する。
4. 3. を各 $T = 100, 1,000, 10,000$ について 100 回ずつ実行する。

表 6 は $\mathrm{E}_P[L]$ の厳密値と近似値を示し、表 5 はそれらを得るのに要した計算時間を示す。ここでサンプリング時間は $T = 10,000$ のときの時間であり、BBS の時間は ZDD の構築時間を含まないことに注意する。表中の TO はタイムアウトを意味し、計算が 30 分以内に終了しなかったことを示す。表 6 より、 n が小さい場合は、 BBS と MC3TS の両手法において非常によい近似値が得られるが、n が大きくなるに連れて MC3TS の近似精度が低下していることが分かる。しかし、$n = 11$ のとき、BBS はサンプリング不可能であるのに対し、MC3TS は 10,000 サンプルをおよそ 30 秒で生成した。このように MC3TS は BBS がスケールする範囲では BBS より低速で低精度であるが、BBS がスケールしないような問題にも適用可能であり、これは非常に大きなアドバンテージである。

n	$\lvert E\rvert$	$\lvert P\rvert$	$p(f \mid \psi)$
2	4	2	$1.250e{-}01$
3	12	12	$2.930e{-}03$
4	24	184	$1.097e{-}05$
5	40	8512	$7.741e{-}09$
6	60	1262816	$1.095e{-}12$
7	84	575780564	$2.977e{-}17$
8	122	789360053252	$1.520e{-}22$
9	144	3266598486981642	$1.465e{-}28$
10	180	41044208702632496804	$2.678e{-}35$
11	220	1568758030464750013214100	$9.310e{-}43$

表 4: $n \times n$ 格子グラフの $|E|$, $|P|$ および $p(f \mid \psi)$

n	Simpath	Exact	BDD	MC3TS
3	$1.240e{-}04$	$2.861e{-}06$	$1.811e{-}02$	$4.555e{-}02$
4	$8.080e{-}04$	$3.099e{-}05$	$3.621e{-}02$	$8.942e{-}02$
5	$5.009e{-}03$	$1.578e{-}03$	$6.158e{-}02$	$1.519e{-}01$
6	$2.542e{-}02$	$2.067e{-}01$	$9.342e{-}02$	$2.479e{-}01$
7	$1.302e{-}01$	$9.268e{+}01$	$1.361e{-}01$	$4.674e{-}01$
8	$6.735e{-}01$	TO	$2.133e{-}01$	$1.143e{+}00$
9	$4.668e{+}00$	TO	$3.309e{-}01$	$3.412e{+}00$
10	$7.364e{+}01$	TO	$4.691e{-}01$	$1.077e{+}01$
11	TO	TO	TO	$3.035e{+}01$

表 5: Simpath と 厳密計算の計算時間および、BBS と MC3TS のサンプリング時間 [s]

6 まとめと今後の課題

本研究では制約付き分布 $p(b \mid f, \psi)$ からの効率的なサンプリング法として MC3TS を提案した。MC3TS は MCMC 法と MCTS を組み合わせた手法であり、MCS と BBS の中間的なサンプリングととらえることができる。探索が終了すれば MC3TS は完全に BBS と一致し、受理確率が 100% ととなる。MC3TS は BBS がスケールしない複雑な問題に対しても適用可能であり、サンプリングの結果として BDD の近似的な構造を生成する点が新しい。

今後の課題としては MC3TS により得られた近似的な構造を用いた、近似的な確率計算や確率学習が考えられる。

参考文献

[1] C.B. Browne, E. Powley, D. Whitehouse, S.M. Lucas, P.I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *Computational Intelligence and AI in Games, IEEE Transactions on*, 4(1):1 –43, march 2012.

[2] Randal E. Bryant. Graph-Based Algorithms for Boolean Function Manipulation. *IEEE Trans. Computers*, 35(8):677–691, 1986.

n	Exact	BBS			MC3TS		
		100	1,000	10,000	100	1,000	10,000
3	5.333e+00	5.357e+00	5.337e+00	5.333e+00	5.322e+00	5.336e+00	5.333e+00
4	1.039e+01	1.042e+01	1.039e+01	1.039e+01	1.079e+01	1.044e+01	1.040e+01
5	1.744e+01	1.745e+01	1.743e+01	1.744e+01	1.828e+01	1.771e+01	1.747e+01
6	2.603e+01	2.604e+01	2.603e+01	2.603e+01	2.729e+01	2.701e+01	2.619e+01
7	3.603e+01	3.604e+01	3.602e+01	3.603e+01	3.787e+01	3.781e+01	3.699e+01
8	TO	4.762e+01	4.759e+01	4.758e+01	5.033e+01	4.977e+01	4.903e+01
9	TO	6.068e+01	6.076e+01	6.076e+01	6.438e+01	6.388e+01	6.301e+01
10	TO	7.545e+01	7.555e+01	7.554e+01	7.996e+01	7.951e+01	7.911e+01
11	TO	TO	TO	TO	9.728e+01	9.715e+01	9.685e+01

表 6: $\mathrm{E}_P[L]$ の厳密値及び近似値

[3] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960.

[4] Luc De Raedt, Paolo Frasconi, Kristian Kersting, and Stephen Muggleton, editors. *Probabilistic Inductive Logic Programming - Theory and Applications*, volume 4911 of *Lecture Notes in Computer Science*. Springer, 2008.

[5] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, 1984.

[6] Lise Getoor. An Introduction to Probabilistic Graphical Models for Relational Data. *IEEE Data Eng. Bull.*, 29(1):32–39, 2006.

[7] Vibhav Gogate and Rina Dechter. Samplesearch: Importance sampling in presence of determinism. *Artif. Intell.*, 175(2):694–729, 2011.

[8] W. K. HASTINGS. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[9] Shin ichi Minato. Zero-suppressed bdds for set manipulation in combinatorial problems. In *DAC*, pages 272–277, 1993.

[10] Masakazu Ishihata, Yoshitaka Kameya, Taisuke Sato, and Shin-ichi Minato. An EM Algorithm on BDDs with Order Encoding for Logic-based Probabilistic Models. *Journal of Machine Learning Research - Proceedings Track*, 13:161–176, 2010.

[11] Masakazu Ishihata and Taisuke Sato. Bayesian inference for statistical abduction using Markov chain Monte Carlo. *Journal of Machine Learning Research - Proceedings Track*, 20:81–96, 2011.

[12] Donald Knuth. *The Art of Computer Programming, Volume 4, Fascicle 1;*. Addison-Wesley Professional: Boston, MA, USA.

[13] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *ECML*, pages 282–293, 2006.

[14] Radford M Neal. Slice sampling. *Annals of Statistics*, 31(3):705–767, 2003.

[15] Takayuki Watanabe Jun Kawahara Ryo Yoshinaka Akihiro Kishimoto Koji Tsuda Shin-ichi Minato Takeru Inoue, Keiji Takano and Yasuhiro Hayashi. Loss minimization of power distribution networks with guaranteed error bound. *Hokkaido University, Division of Computer Science, TCS Technical Reports, TCS-TR-A-12-59*, 2012.

[16] Wei Wei, Jordan Erenrich, and Bart Selman. Towards efficient sampling: Exploiting random walk strategies. In *AAAI*, pages 670–676, 2004.

[17] Masahiro Yamaguchi, Masakazu Ishihata, and Taisuke Sato. Probabilistic inference based on slice sampling and sat technologies. In *The 25th Annual Conference of the Japanese Society for Artificial Inteligence*, 2011.