

デジタルアニーラのアーキテクチャと将来展望

田村 泰孝, 印 芳, 神田 浩一, 片山 健太郎

富士通デジタルアニーラは、既存の汎用計算機で解くことが困難な組合せ最適化問題を高速に解くための新しいコンピューティング技術である。この技術は、高速性を達成するため DAU (Digital Annealing Unit) と呼ばれる専用プロセッサを用い、制約付き問題を効率的に解くためソフトウェアとハードウェアのハイブリッド技術を使用する。ハイブリッド技術に関しては本特集号の宮澤らの記事で詳しく説明する。本稿では専用のハードウェアアクセラレータである DAU の設計理念、アーキテクチャと将来の拡張性について述べる。

キーワード：イジングマシン, デジタルアニーラ, ハードウェアアクセラレータ

1. はじめに

スケーリング則に頼った CMOS 集積回路の性能向上が事実上終焉した現在、コンピューティングシステムの性能向上にはアーキテクチャの革新が必要である。一般的には CPU などの汎用プラットフォームから CPU+GPU や ASIC などの目的に特化したプラットフォームに移行することで高速化が得られる [1]。しかし、開発コストを回収するためには同一の専用ハードウェアで広い応用範囲をカバーするという矛盾する要求を満たす必要がある。

このような要求を満たす可能性があるとして期待されている問題カテゴリーに最適化がある。最適化問題向けに専用ハードウェアアクセラレータを開発するアプローチは過去にも何度か取り上げられてきた。その最近の例が D-Wave 社の量子アニーラ [2] である。しかし、量子アニーラは 10 mK 程度の極低温を必要としており、変数間の接続係数の個数と精度に制限がある。一方、日立 [3]、東芝 [4]、富士通 [5, 6] による室温での CMOS 技術を用いたアニーラは、十分な性能が得られるなら広く応用される可能性があるとして期待されている。

古典あるいは量子アニーラは二値 (バイナリあるいはスピン) 変数の二次形式のエネルギーを扱うため Ising (イジング) マシンと呼ばれる。エネルギー関数を二値二次形式に限定することは高速化に寄与するが、この限定

が原因で求解に時間がかかる場合がある。たとえば割当問題の場合には、複数のビットのうち 1 個だけが 1 となる 1-hot コーディングを使う場合があり、通常のイジングマシンでは速度低下を招く。不等式制約をスラック変数によって二値二次形式で定式化すると通常のイジングマシンでは求解に時間がかかり実用性がない。コスト関数に 3 次以上の高次項がある場合、高次項を二次形式に変換すると問題サイズが爆発的に増加する問題がある。

さまざまなコスト関数を扱えるようにイジングマシンの機能を拡張できればその応用範囲は広がる。本稿では富士通の第二世代デジタルアニーラに用いる専用プロセッサ DAU (Digital Annealing Unit) にフォーカスし、DAU の動作原理と将来の機能拡張の可能性について述べる。ここで述べる機能拡張は将来の可能性を探るために検討しているものであり、本稿により将来のサービスを約束するものではないことに注意されたい。

本稿の 2 節で DAU の設計方針を説明し、3 節で多値スピンの用いた 1-hot コーディングへの応用、4 節で不等式制約および高次項コスト関数のための補助変数法、5 節で実験結果を示し、6 節で結論を述べる。

2. 基本アーキテクチャ

2.1 設計方針

DAU 設計の基本方針は以下の 4 項目にまとめられる。

- 単純な構成と繰り返し演算
専用化による高速化を得るためできる限りシンプルな構成で単純な繰り返し演算を並列に行う。
- データ移動の最小化
データ移動を最小化する構成とするため演算中のデータの読み書きはオンチップのメモリとの間でのみ発生する方式を用いる。
- 差分計算
演算コストの高い乗算演算を削減するため、2.2 節

たむら ひろたか
株式会社 DXR 研究所
〒223-0066 神奈川県横浜市港北区高田西 4-38-10
tamura.hirota@dxrlab.com
いん ほう, かんた こういち, かたやま けんたろう
富士通株式会社研究本部 ICT システム研究所
〒211-8588 神奈川県川崎市中原区上小田中 4-1-1
fang.yin@fujitsu.com
kouichi.kanda@fujitsu.com
k.kentaro@fujitsu.com

に述べるように探索のための演算に可能な限り差分値を積算する加減算を用いる。

● 並列性の利用

MCMC (Markov-Chain Monte Carlo 法) による状態空間内の探索を行う場合に探索演算の並列化やレプリカレベルの並列化を利用する。

2.2 DAU のアルゴリズムとアーキテクチャ

DAU が扱うエネルギー (コスト関数) は二値変数の N 次元ベクトル \mathbf{x} の二次形式 $E(\mathbf{x})$ で与えられる:

$$E(\mathbf{x}) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N W_{ij} x_i x_j - \sum_{i=1}^N b_i x_i \quad (1)$$

結合係数 W_{ij} は $W_{ij} = W_{ji}, W_{ii} = 0$ を満たす。本稿は主として変数 x_i が 0/1 バイナリの場合について述べるが、わずかな変更でスピン変数を扱うことができる。

DAU は (1) を制約条件なしで最小化する状態変数 \mathbf{x} を確率的に探索する。そのため最小化したい $E(\mathbf{x})$ を (1) から直接計算せず状態の局所の変化で生じた差分のみを計算する。この差分計算は状態 \mathbf{x} のうち x_i を $x_i + \Delta x_i$ と変化させたものを $\mathbf{x}^{(i)}$ としたときエネルギーの増加分 ΔE_i が

$$\Delta E_i = E(\mathbf{x}^{(i)}) - E(\mathbf{x}) = -\Delta x_i h_i \quad (2)$$

$$h_i = \sum_{j \in D} W_{ij} x_j + b_i \quad (3)$$

となることを利用する。 h_i はニューラルネットの用語ではニューロン x_i への入力総和であり、スピン系の物理の言葉ではスピン i が他のスピンとの相互作用と局所的磁界から受ける局所場 (local field) である。

h_i を演算器近傍に配置したアクセス時間の短いメモリに記憶して参照すればエネルギー差分を素早く求めることが可能となる。なおバイナリ変数 $x_i \in \{0, 1\}$ の場合は $\Delta x_i = 1 - 2x_i$ となり、スピン変数 $x_i \in \{-1, +1\}$ の場合は $\Delta x_i = -2x_i$ となる。

われわれの用いるアーキテクチャは (2) のエネルギー差分に基づき MCMC により状態の遷移を制御する。MCMC をシリアル試行で行う場合は変更する変数 x_j をランダムあるいはインデックス順に指定してエネルギーの差分 ΔE_i を計算し、以下の Metropolis 則 [7] に基づく確率 P_j で状態変化 $x_j \rightarrow x_j + \Delta x_j$ を受け入れる。

$$P_j = \min[1, \exp(-\beta \Delta E_j)] \quad (4)$$

ここで β は逆温度で、温度 T の逆数で与えられる。

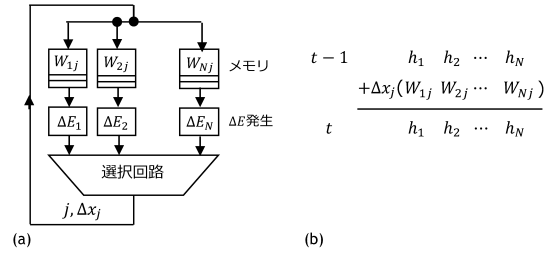


図 1 DAU 動作原理 (a) ブロック図 (b) 差分更新式

局所場 h_i がすべての変数 x_i に対して (3) を満たすよう更新された状態では、現在の状態から Hamming 距離 1 にある複数の近傍から反転する変数を選ぶ並列試行が可能である。並列試行にはさまざまな方式が可能であるが、たとえば Rejection-free (RF) 選択 [8, 9] では r_i を $0 < r_i < 1$ の一様乱数として

$$j = \underset{i}{\operatorname{argmin}} [\max(0, \Delta E_i) + T \log(-\log(r_i))] \quad (5)$$

とする。RF 選択は提案した変数反転は拒絶なしで実行されるため、通常のシリアル選択の MCMC で局所解に留まる回数が非常に多い状況では高速化に寄与する。

(2), (3) の処理はそれぞれの状態変数 x_i ($i = 1, \dots, N$) を計算する回路ブロック (ニューロン) を N 個用意して $O(N)$ の並列度で実行できる (図 1(a))。一方, (5) で示すような選択演算には複数の候補から 1 個を選ぶシリアル処理が必要となる。この選択演算が全体のボトルネックにならないためには、並列試行の並列度や処理回路の構成の最適化が必要である。

シリアル試行にせよ並列試行にせよ更新する変数 x_j が決まったら局所場、エネルギーの更新および状態変数の更新を行う。更新のためには、選択された変数 x_j のインデックス (アドレス) j を用いて係数行列 W_{ij} を読み出し、各ニューロンで以下の差分計算を行う。この差分計算は N 個のニューロンを用いて並列に実行できる (図 1(b)):

$$x_j \leftarrow x_j + \Delta x_j \quad (6)$$

$$h_i \leftarrow h_i + W_{ij} \Delta x_j \quad \text{for } i \neq j \quad (7)$$

$$E \leftarrow E - h_j \Delta x_j \quad (8)$$

なお (3) の局所場 h_i は (1) の二次形式エネルギーの変数 x_i による階差 (の符号を反転したもの) であり、 h_i の変数 x_j による階差が結合係数 W_{ij} で一定値となる。図 1(a) の構成のハードウェアは二階階差 W_{ij} を次々に加減算して任意の二値二次形式のエネルギー値とその階差を計算している。この意味で DAU は 1822 年に Charles Babbage により提案された階差

機関 (Difference Engine) の確率動作版といえる。

階差機関は一次元変数空間の中で一方向に変数を変えながら階差を積算により高次多項式の値を計算した。これに対し DAU は N 次元の二値変数の空間の中を確率的に移動しながら階差を積算する。その結果として二次形式のエネルギーも求められるがこれはむしろ副産物であり、本当の目的はエネルギーの一階階差により確率的探索をガイドしてエネルギーの最小値を求めることである。

2.3 交換モンテカルロ法 [10]

DAU は既知の 3 種類の並列計算技術をハードウェアとして実装した。一つは、局所場の情報を用いて反転する変数 x_j の探索を図 1(a) に示したように並列実行することである。もう一つは局所場の更新演算 (6), (7) をニューロン回路で同時並行的に行うことである。

これらに加えて DAU は複数の独立な探索プロセス (レプリカ) を用いた交換モンテカルロ法を用いる。交換モンテカルロ法は異なる温度で探索を行う複数のレプリカ (独立な探索を行うシステム) を用いることで局所解からの脱出の手段を与える。

交換モンテカルロ法では、独立に MCMC 探索を行う複数のレプリカを用い、それぞれを異なる逆温度に置く。 k 番目の逆温度を β_k とする。それぞれのレプリカで一定回数の時間ステップ (iteration) の MCMC 演算を行ってから、以下で示す確率的ルールで隣接する温度のレプリカ間で温度を交換する。交換は温度空間内で隣合うペア状態に番号を付け、偶数ペア間の交換と奇数ペア間の交換を繰り返す。逆温度 β_k と β_{k+1} にあるレプリカを $\mathbf{x}^k, \mathbf{x}^{k+1}$ としレプリカのエネルギーがそれぞれ E_k, E_{k+1} の場合、レプリカ \mathbf{x}^k と \mathbf{x}^{k+1} の間の温度交換が受け入れられる確率は

$$A = \min [1, \exp [-(\beta_k - \beta_{k+1})(H_{k+1} - H_k)]] \quad (9)$$

で与えられる。なおハードウェア実装では各レプリカの演算は並列に実行される [11]。

3. 多値スピンの使用

この節ではイジングマシンの機能拡張の一つの自然な方向の一つとして、多値スピン表現への拡張を取り上げる。多値スピンとはイジングモデルのスピンの二値のいずれかを取るのに対し、多値 (多色) のスピンの一つを取ることを意味する。

多値スピン表現を用いると、順列 (permutation) や割当 (assignment) を変数とした最適化問題が効率的に定式化できる。以下では多値スピン表現を使って二

次割当問題 (QAP: Quadratic Assignment Problem) と、QSAP: Quadratic Semi-Assignment Problem を解くアルゴリズムを説明する。

QAP [12] は n 個の工場を n 個の場所に割り当てる問題であり順列を変数とする最適化問題である。QAP では集合 $N = \{1, 2, \dots, n\}$ に対するすべての permutation の集合 Φ_n の要素 ϕ のうち以下のコスト $C(\phi)$ を最小化するものを求める。

$$C(\phi) = \sum_{i=1}^n \sum_{j=1}^n f_{ij} d_{\phi_i \phi_j} = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n f_{ij} d_{kl} x_{ik} x_{jl} \quad (10)$$

$$x_{ij} = \begin{cases} 1 & \text{if } \phi_i = j \\ 0 & \text{otherwise} \end{cases}, \quad (x_{ij}) \in \{0, 1\}^{n \times n} \quad (11)$$

ここで $F = (f_{ij}) \in \mathbb{R}^{n \times n}$ は工場 i と j の間の物流の大きさを表す F 行列、 $D = (d_{kl}) \in \mathbb{R}^{n \times n}$ は場所 k と l の間の距離を表す D 行列である。

多値スピンは QAP 類似の割り当て問題である QSAP に応用可能である。QSAP では工場のペアの間での場所交換によるコストの変化 ΔC_{ex}^s に加えて工場 a を現在の場所から場所 l に割り当て直すことによるコスト変化 ΔC_{rel}^s も考える必要がある。詳細は文献 [13, 14] を参照されたい。

イジング型の DAU の基本動作が 1 変数の反転であったのに対し、多値スピンの QAP を解く場合には二つの工場 a, b を選んでその場所 ϕ_a, ϕ_b を交換する動作となる。これを二値スピンの実装すると 4 回のビット反転を必要とする。この交換動作によるエネルギー (コスト) の変化 ΔC_{ex} は次式で与えられる。

$$\Delta C_{ex} = \sum_{i \neq a, b}^n 2(f_{bi} - f_{ai})(d_{\phi_a \phi_i} - d_{\phi_b \phi_i}) \quad (12)$$

QSAP の場合も (12) と同様の差分式が現れる。

多値スピンによる実装で用いるエネルギー差分式 (12) はイジングマシン実装での差分演算 (8) に比べると計算オーバーヘッドが大きい。このため並列試行は行わず一つの基本動作に対するシリアル試行を行う。この方式ではイジングマシンで 4-bit 反転が必要な場所交換や 2-bit 反転が必要な場所の再割り当てによるエネルギー差分が (10)–(12) のように一つの閉じた式で書けることにより大幅な高速化が得られる。

4. 不等式制約および高次項への拡張

不等式制約によるペナルティや 3 次以上の高次項

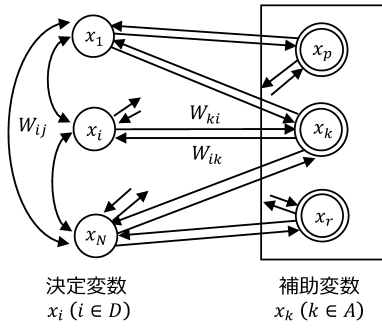


図2 補助変数を用いた拡張

がコスト関数に含まれる場合に、補助的な変数を使うことでこれらのコスト関数を扱うことができる。追加された補助ニューロンは、通常の双方向性ニューロンの間の信号パスの特性を修正して不等式制約や高次エネルギー項を扱うことを可能とする(図2)。本節では不等式制約について説明する。なお本稿で示す例においては、補助変数グループの内部での相互結合はない。

不等式制約をもつ系の総エネルギー H は決定変数(最適化問題の独立変数) \mathbf{x} の関数として以下のように与えられるとする。決定変数のインデックスの集合を D 、決定変数に従属して定まる補助変数のインデックスの集合を A とし、不等式制約は決定変数 x_i の重み付き和に対する上限 $-b_k$ の上限制約とする。

$$H(\mathbf{x}) = E(\mathbf{x}) + \sum_{k \in A} \lambda_k \max(0, h_k) \quad (13)$$

$$h_k = \sum_{i \in D} W_{ki} x_i + b_k \quad (14)$$

(13) は式(1)の二次形式エネルギー E に k 番目の不等式の上限の超過分をペナルティとし正の係数 λ_k を乗じて加算して総エネルギーとしたものである。 h_k ($k \in A$) を局所場(入力総和)とし活性化関数がステップ関数 $u(h)$ の補助変数ニューロンにより補助変数 x_k を用いると(13), (14)は二次形式で表される:

$$H(\mathbf{x}) = E(\mathbf{x}) + \sum_{k \in A} \sum_{i \in D} \lambda_k W_{ki} x_i x_k + \sum_{k \in A} b_k x_k \quad (15)$$

決定変数の一つを変化させた場合、この変化に応じて x_k を1ビットずつ更新して(15)を計算すれば総エネルギーの値が求められる。

決定変数 x_i ($i \in D$) が $x_i + \Delta x_i$ と変化したとき H の変化 ΔH_i は以下で与えられる。

$$\Delta H_i = -h_i \Delta x_i \quad (16)$$

$$h_i = \sum_{j \in D} W_{ij} x_j + b_i - \sum_{k \in A} \lambda_k W_{ki} x_k \quad (17)$$

(17) は二値の補助変数から決定変数に $-\lambda_k W_{ki}$ という係数で局所場への寄与があることを示す。この寄与により決定変数に対して適切な大きさの「復元力」が発生し、系のエネルギーを最小化する方向の遷移が促進される。決定変数の値が更新された後に、これに伴って補助変数が $x_k \leftarrow x_k + \Delta x_k$ と変化した場合の総エネルギー変化 ΔH_k は以下で与えられる。

$$\Delta H_k = \lambda_k h_k \Delta x_k \quad (18)$$

補助変数を出力するニューロンは制約違反状況の検出器として動作しており、違反状態に応じて決定変数に対して帰還信号を発生する。この帰還ループが不等式制約を満たすように決定変数の動作にバイアスを与えていると解釈できる。

エネルギーに高次多項式が含まれる場合には不等式制約項ではなく3次以上の多項式の付加項 $G(\mathbf{x})$ が二次エネルギー $E(\mathbf{x})$ に加算されるとする。ここで変数がバイナリの場合を考え、以下のように $G(\mathbf{x})$ が正負のリテラルの積を含む場合を考える。

$$H(\mathbf{x}) = E(\mathbf{x}) - \sum_{k \in A} Z_k \prod_{j \in J_k} (s_{kj} \oplus x_j) \quad (19)$$

Z_k はインデックス k の高次項のエネルギー、 J_k はこの高次項に含まれる変数インデックス j の集合、 s_{kj} は高次項 k にバイナリ変数 x_j が正リテラル (x_j) として含まれるとき $s_{kj} = 0$ 、負リテラル (\bar{x}_j) として含まれるとき $s_{kj} = 1$ となる変数、 \oplus は EX-OR 演算である。

ここでも(19)を使って直接高次エネルギーを計算することはせず、決定変数 x_i を $x_i + \Delta x_i$ とした場合のエネルギー差分 ΔH_i を求める。以下ではインデックスが k の高次積により決定変数に発生する局所場を求める。複数の高次積がある場合には k に関する和を実行する。

高次の局所場が非ゼロになるのは、高次積の値に対して決定変数 x_i が決定権をもっている (x_i の変化により高次積の値が変化する) 場合である。決定変数 x_i が高次積に対して決定権をもつ状態を検出するには、積の中のリテラルがすべて1の場合、一つだけ0の場合、二つ以上0となる領域間の遷移を検出すればよい。

このため高次積 k の中で値が1となるリテラルの数

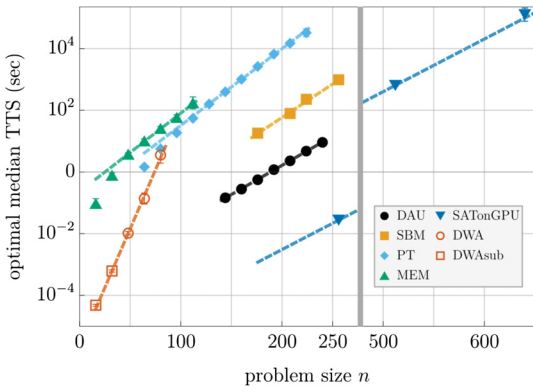


図3 3R-3XORSAT 問題の TTS スケール特性 [15]
 プロセッサ上の交換モンテカルロ法 (=PT: Parallel Tempering), D-Wave の量子アニーラ (=DWA, DWASub: D-Wave Annealer) や CPU/GPU で実装された専用システム (=SBM: Simulated Bifurcation Machine, MEM: memcomputing)

h_k に応じて 3 値をとる補助変数 x_k を用い, この値を用いて高次の局所場の変化分を計算できる. 結果は次式となる.

$$x_k = \begin{cases} 0 & \text{if } h_k \leq |J_k| - 2 \\ 1 & \text{if } h_k = |J_k| - 1 \\ 2 & \text{if } h_k = |J_k| \end{cases} \quad (20)$$

$$\Delta h_i = \begin{cases} \Delta x_k Z_k \sigma_{ki} (\overline{s_{ki}} \oplus x_i) & (x_k: 0 \leftrightarrow 1) \\ \Delta x_k Z_k \sigma_{ki} (s_{mi} \oplus x_i) & (x_k: 1 \leftrightarrow 2) \end{cases} \quad (21)$$

5. 数値実験

5.1 第二世代 DAU

2 節で動作原理を説明した単一変数反転を基本とした DAU (第二世代, ASIC 版) を用いて 3R-3XORSAT 問題を解いた [15]. この問題は三つのリテラルで構成された XOR clause を最大に充足する問題を 2 体結合に変換したものであり, 問題サイズに対して指数関数的に時間がかかることが知られている. TTS (Time-to-Solution: 正解が得られるまでの時間) の問題サイズ依存性を図 3 に示す.

プロセッサ上の交換モンテカルロ法, D-Wave の量子アニーラや CPU あるいは GPU で実装された専用システムに対して DAU は速度で 2 桁から 4 桁程度の優位性がある.

5.2 多値スピンをを用いた QAP/QSAP ソルバー

表 1 は従来の最速ソルバ ParEOTS [16] と 4 節の QAP/QSAP アルゴリズムとで正解が知られているイ

表 1 QAP (100 × 100 以下) に対する TTS 比較

Instance	ParEOTS	This work (sec)
dre42	0.7	0.1 ± 0.01
dre56	5.6	0.99 ± 0.07
dre72	26	5.23 ± 0.31
Inst50	17	3.58 ± 0.27
Inst60	67	2 ± 0.16
Inst70	127	6.24 ± 0.98
Inst80	116	19.36 ± 1.75
sko81	22.4	0.13 ± 0.01
sko90	92	0.26 ± 0.01
sko100a	69	0.27 ± 0.01
sko100b	45	0.19 ± 0.01
sko100c	56	0.29 ± 0.02
sko100d	37	0.38 ± 0.02
sko100e	47	0.21 ± 0.01
sko100f	57	0.24 ± 0.01
tai60b	46	0.14 ± 0.01
tai80b	53	0.43 ± 0.01
tai100b	71	0.36 ± 0.02
wil100	97	0.46 ± 0.02
Speedup	1.00×	57.09

ンスタンス (5 分間のタイムアウト時間内に 100% の成功率で解くことができたもの) に対する TTS の比較である. 提案アルゴリズムはメモリ 128 GB 搭載の 64-core AMD Threadripper 3990X で実行した.

なお 3 節の方式の実装にはいくつかのバリエーションがあるが代表的なものを記載した. 詳細は文献 [13, 14] を参照して欲しい.

TTS の 99% 信頼区間と平均値は乱数シードをランダム変えた 100 回の独立した計算より求めた. ParEOTS (128-core CPU system) の TTS は論文から直接引用した. なお ParEOTS の論文 [16] では TTS のばらつきの測定値は発表されていない. 表 1 から多値スピンをを用いた QAP/QSAP ソルバーは従来の BKS :Best-Known Solver に比較して幾何平均で 57 倍高速であることが判る.

正解が知られていない大きなサイズ (125 × 125 から 729 × 729) の QAP 問題インスタンスを解いた. 紙面の都合でサイズが 343 × 343 と 729 × 729 のサイズを表 2 に示した. 計算環境は表 1 と同じである. イジングマシンで定式化すると 15.6 k bit から 531 k bit のサイズとなる (F, D 行列を使うため, 通常のイジングマシンより大きなサイズを扱うことができることに注意).

提案する方法では $n = 125, 175$ の問題に対してカットオフ時間を 10 秒, $n = 343, 729$ の問題に対して 30 秒に設定し測定回数を 20 回として TTS を平均した. ほ

表 2 大規模 QAP での性能比較

Instance	BKS	This work (Best)	This work (Avg.)
343e01	145862	141086	142744.8
343e02	154018	148992	151511
343e03	144,278	141,554	143,970.6
343e04	162,092	154,426	156,751.8
343e05	142,110	141,154	142,817.5
343e06	144,274	141,036	142,987.3
343e07	154,776	143,920	147,737.2
343e08	133,770	132,362	134,268.3
343e09	143,018	139,180	141,779.3
343e10	152,828	147,396	149,856.6
343e11	146,446	144,464	147,992.8
343e12	162,954	155,362	158,808.1
343e13	137,836	130,316	132,892.4
343e14	150,428	145,432	147,836.2
343e15	156,682	148,698	151,749.4
343e16	154,264	150,036	151,375.4
343e17	136,650	129,994	132,600.4
343e18	136,694	132,950	135,916.1
343e19	150,486	145,110	147,864.4
343e20	151,552	146,414	148,606.6
729e01	448,014	422,118	433,695.5
729e02	445,356	436,644	442,884.1
729e03	427,076	414,192	418,740.6
729e04	431,496	419,260	424,187.5
729e05	444,704	422,122	432,819.3
729e06	442,248	425,206	434,288
729e07	428,644	412,706	423,293.8
729e08	425,694	411,066	418,731
729e09	395,656	385,582	392,344.2
729e10	431,938	421,430	428,517.8

とんどの問題インスタンスで従来の BKS 値を更新し、表 2 の範囲で BKS を更新できなかったのは背景を灰色とした値だけであった。BKS が報告されている諸文献については Bagherbeik の論文 [14] を参照してほしい。

QKP/QSAP ソルバーは TSP を解くこともできるため参考までに TSPLIB [17] の 51 都市問題 (eli51) と 76 都市問題 (pr76) を解いたときの正解率と求解時間の関係を図 4 に示す。乱数シードを変えた 100 回のシミュレーションを行うと eli51 で 0.02 sec, pr76 で 0.2 sec から正解が得られ始め、それぞれ 1.1 sec, 4.2 sec 計算を行うとすべてのシード値で正解に到達する。

5.3 不等式制約および高次項を扱う補助変数法

4 節で説明した補助変数法により Pisinger の knapsack 問題インスタンス [18] を解いた。slack 変数を用いてこれらの knapsack 問題を二値二次形式エネルギーの最小化として定式化した場合は正解が得られないケースが多いが提案した方法では正解を得ることが

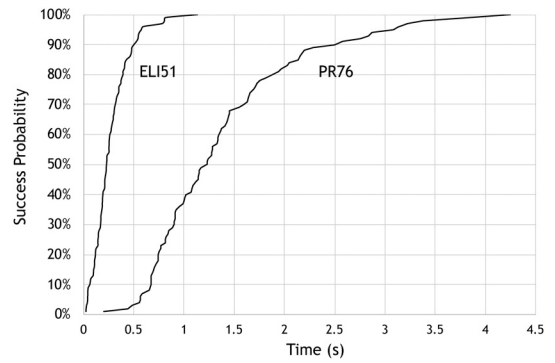


図 4 QAP/QSAP ソルバによる TSP の求解特性

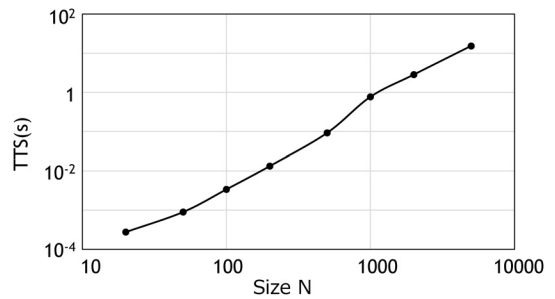


図 5 Knapsack 問題の TTS

表 3 SAT に対する TTS

Instance	リテラル数	節の数	TTS (sec)
Uf20-01	20	91	0.0012
Uf50-01	50	218	0.0019
Uf75-01	75	325	0.0025
Uf100-01	100	430	0.0033
Uf150-01	150	645	0.0053
Uf200-01	200	860	0.05
Uf250-01	250	1,065	0.08

できる (図 5)。ここでの計算環境は 10-core Intel Core i9-10850K CPU@3.60 GHz であった。

バイナリ変数の高次積を補助変数として用いる方法で SATLIB [19] の 3-SAT 問題インスタンス (clause 数が 91–1,065) を解いた結果を表 3 に示す。3-SAT 以上の SAT もイジングマシンで解くのは困難であるが、補助変数法では求解ができています。提案した方法により最適化問題の中に SAT 的な制約条件を入れることが可能になり、問題の定式化が楽になる局面があると期待できる。この高次項問題についてはメモリ 12 GB の Titan V GPU で TTS を求めた。

6. 終わりに

本稿では、二値二次形式のエネルギーを最小化する

ためのコンピューティング方式である DAU とその拡張について述べた。DAU は二値二次形式のエネルギーの最小化に特化し、差分計算と並列化により高速化が達成される。しかし実用性の観点からは二値二次形式への限定は大きな制約となる。この制約を解消する技術として、スピンを二値のイジングモデルから多値スピンのモデルに拡張する技術と、不等式制約や高次のエネルギーなど二次形式に変換が難しい部分を補助変数により実現する方式を紹介した。

世の中の社会課題の求解では、ハードウェアの規模を超える問題を扱うことや、多数の制約条件を扱うことが必要となる。これらの課題に対するハードウェア・ソフトウェアのハイブリッド技術に関しては本特集号の宮澤らによる論文 [20] を参照されたい。

今後、このような専用計算システムが発展するためにはビジネス的に存在意味を主張できる性能を許容しうる開発コストで実現する必要がある。これは決して簡単ではないが、研究者・開発者の創意により解決していくことを期待する。

謝辞 コード作成およびデータ取得に協力いただいた埼玉大学大学院理工学研究科の古江 友樹氏に感謝します。

参考文献

- [1] M. Horowitz, “Computing’s energy problem (and what we can do about it),” *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14, 2014.
- [2] P. I. Bunyk, E. M. Hoskinson, M. W. Johnson, E. Tolkacheva, F. Altomare, A. J. Berkley, R. Harris, J. P. Hilton, T. Lanting, A. J. Przybysz and J. Whitaker, “Architectural considerations in the design of a superconducting quantum annealing processor,” *IEEE Transactions Applied Superconductivity*, **24**, 1700110, 2014.
- [3] T. Takemoto, K. Yamamoto, C. Yoshimura, M. Hayashi, M. Tada, H. Saito, M. Mashimo and M. Yamaoka, “4.6 a 144kb annealing system composed of 9×16kb annealing processor chips with scalable chip-to-chip connections for large-scale combinatorial optimization problems,” *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, **64**, pp. 64–66, 2021.
- [4] K. Tatsumura, A. R. Dixon and H. Goto, “FPGA-based simulated bifurcation machine,” In *Proceedings of 2019 29th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 59–66, 2019.
- [5] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura and H. G. Katzgraber, “Physics-inspired optimization for quadratic unconstrained problems using a digital annealer,” *Frontiers in Physics*, **7**, p. 48, 2019.
- [6] S. Matsubara, M. Takatsu, T. Miyazawa, T. Shibasaki, Y. Watanabe, K. Takemoto and H. Tamura, “Digital annealer for high-speed solving of combinatorial optimization problems and its applications,” In *Proceedings of 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 667–672, 2020.
- [7] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, **21**, pp. 1087–1092, 1953.
- [8] A. Bortz, M. Kalos and J. Lebowitz, “A new algorithm for monte carlo simulation of ising spin systems,” *Journal of Computational Physics*, **17**, pp. 10–18, 1975.
- [9] J. S. Rosenthal, A. Dote, K. Dabiri, H. Tamura and A. Sheikholeslami, “Jump Markov chains and rejection-free metropolis algorithms,” *Computational Statistics*, **36**, pp. 2789–2811, 2021.
- [10] K. Hukushima and K. Nemoto, “Exchange monte carlo method and application to spin glass simulations,” *Journal of the Physical Society of Japan*, **65**, pp. 1604–1608, 1996.
- [11] K. Dabiri, M. Malekmohammadi, A. Sheikholeslami and H. Tamura, “Replica exchange mcmc hardware with automatic temperature selection and parallel trial,” *IEEE Transactions on Parallel and Distributed Systems*, **31**, pp. 1681–1692, 2020.
- [12] E. L. Lawler, “The quadratic assignment problem,” *Management Science*, **9**, pp. 586–599, 1963.
- [13] M. Bagherbeik, P. Ashtari, S. F. Mousavi, K. Kanda, H. Tamura and A. Sheikholeslami, “A permutational boltzmann machine with parallel tempering for solving combinatorial optimization problems,” *International Conference on Parallel Problem Solving from Nature*, T. Bäck, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich and H. Trautmann (eds.), Springer, pp. 317–331, 2020.
- [14] M. Bagherbeik and A. Sheikholeslami, “Caching and vectorization schemes to accelerate local search algorithms for assignment problems,” *2021 IEEE Congress on Evolutionary Computation*, pp. 2549–2558, 2021.
- [15] M. Kowalsky, T. Albash, I. Hen and D. A. Lidar, “3-regular three-XORSAT planted solutions benchmark of classical and quantum heuristic optimizers,” *Quantum Science and Technology*, **7**, 025008, 2022.
- [16] D. Munera, D. Diaz and S. Abreu, “Hybridization as cooperative parallelism for the quadratic assignment problem,” *Hybrid Metaheuristics*, M. J. Blesa, C. Blum, A. Cangelosi, V. Cutello, A. Di Novo, M. Pavone and E. G. Talbi (eds.), Springer, pp. 47–61, 2016.
- [17] G. Reinelt, TSPLIB, <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/> (2022 年 4 月 23 日閲覧)
- [18] D. Pisinger, <http://hjemmesider.diku.dk/~pisinger/codes.html> (2022 年 4 月 23 日閲覧)
- [19] H. H. Hoos and T. Stützle, “SATLIB: An online resource for research on SAT,” *SAT 2000*, I. P. Gent, H. van Maaren and T. Walsh (eds.), IOS Press, pp. 283–292, 2000.
- [20] 宮澤俊之, 小山純平, Matthieu Parizy, “第三世代デジタルアニーラーハイブリッドソルバ技術とその性能,” オペレーションズ・リサーチ: 経営の科学, **67**, pp. 312–319, 2022.