組み合わせ最適化問題向けハードウェアの高速化アーキテクチャー

Accelerator Architecture for Combinatorial Optimization Problems

● 塚本三六　● 高津　求　● 松原　聡　● 田村泰孝

あらまし

社会では，限られた人や時間などの制約のもとで難しい意思決定を迫られる場面，例えば災害復旧の手順を決める場合や，投資ポートフォリオの最適化，経済政策の決定などがしばしば発生する。このような意思決定においては，様々な要因の組み合わせを考慮して評価を行い，最適なものを選択する「組み合わせ最適化問題」を解く必要がある。組み合わせ最適化問題は，考慮する要因の数が増えると組み合わせの数が爆発的に増えるため，現行の汎用ノイマン型プロセッサを用いた単純な数え上げ法では現実的な時間内で解くことが難しい。筆者らはこのような課題を解決するため，1,024ビットの全結合イジングモデルを高速化する手法を開発し，FPGA（Field-Programmable Gate Array）に実装した。組み合わせ最適化問題の例として32都市の巡回セールスマン問題を実際に解き，3.5 GHzのIntel Xeon プロセッサー E5-1620 v3上で同じ処理をした場合に対して約12,000倍の高速化が確認された。

Abstract

In today's world, there are many situations in which difficult decisions must be made under such constraints as a limited number of people and a limited amount of time. These situations include disaster response planning, economic policy decision-making, and investment portfolio optimization. In such situations, it is often necessary to solve a "combinatorial optimization problem," which involves evaluating different combinations of various factors and selecting the optimum combination. Since the number of combinations increases explosively as the number of factors increases, the problem becomes difficult to solve in a realistic amount of time using a von Neumann type processor. To solve such problems, we have developed a scheme to speed up the 1,024-bit Ising model and implemented it in a field-programmable gate array (FPGA). Testing demonstrated that it can solve the 32-city traveling salesman problem 12,000 times faster than a similar program running on a 3.5-GHz Intel Xeon processor E5-1620 v3.

ま え が き

過去50年にわたり，半導体集積回路の性能向上を支えてきたトランジスタのスケーリング則は，あと5年程度で実質的に限界に達すると考えられている。[1] このため，過去と同様に10年で100倍に性能が向上する向上トレンド[2] を維持するためには，システムの性能をスケーリングに頼らず継続的に向上させていく新たな方法が必要となる（**図-1**）。

スケーリングに頼らない性能向上策のうち，ハードウェアに関わる部分では，従来型プロセッサより電力効率と速度が向上した演算方式を用いる方法が注目されている。[3] 例えば，今後5年から10年の期間では，GPGPU（General-Purpose Computing on Graphics Processing Units） やFPGA（Field-Programmable Gate Array）を使った専用ハードウェアによる性能向上が主流になると予想される。

更に，2025年前後から先の性能向上には，メモリアクセスやデータ移動のエネルギーを削減するために，何らかの非ノイマン型のハードウェアを取り入れていく必要がある。これらの非ノイマン型ハードウェアには，アニール型の量子コンピューティング，[4] コヒーレントコンピューティング，[5] ニューラルネットワーク[6, 7] などがある。

アニール型量子コンピューティングやコヒーレントコンピューティングの適用領域の一つに，計算量が膨大になる組み合わせ最適化問題がある。組み合わせ最適化問題は，様々な要素の組み合わせの一つひとつに対して何らかの評価値が決まる場合に，この評価値を最少にする組み合わせを求める問題として定式化できる。要素の数が多くなると組み合わせの数は極めて大きくなるため，単純な数え上げ法では扱うことができない。

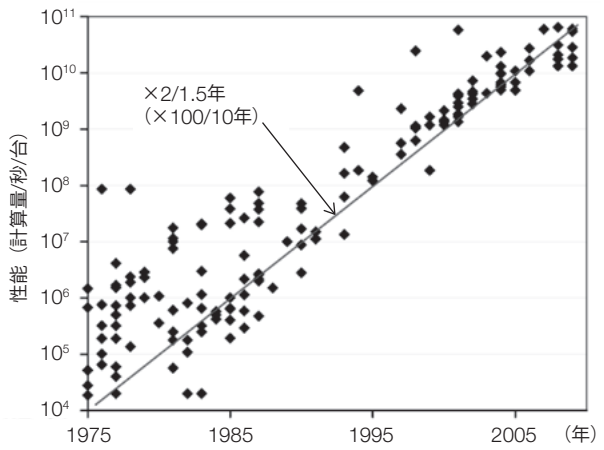


図-1　コンピュータ性能のトレンド

筆者らは，半導体集積回路を用いた専用エンジンにより，組み合わせ最適化問題の近似解を高速に求めることを検討している。これに対して，1,024ビットで組み合わせを表す組み合わせ最適問題の近似解を高速で得ることが可能なアーキテクチャーを開発し，これをAltera社製のFPGA「Arria 10 GX」に実装した。試作したハードウェアを用いて，32都市の巡回セールスマン問題（TSP：Traveling Salesman Problem）を解いた場合，3.5 GHzのIntel Xeon プロセッサー E5-1620 v3上で処理するシミュレーテッドアニーリング（SA）に対して，約12,000倍の高速化が確認された。

将来的には，専用ICによるエンジンを多数実装し，階層的な動作による高速化やアンサンブル交換法[8] などを用いることにより，更に2桁以上の性能改善が可能であると予測している。

本稿では，富士通研究所が開発した高速化手法と，実際にTSPに適用して問題を解いた結果について述べる。

イジング型エネルギー関数による探索

提案するハードウェアは，マルコフ連鎖モンテカルロ法（Markov-Chain Monte-Carlo method，MCMC）に基づく統計的探索を並列に行うことで，以下の式（1）によって示されるイジング型エネルギー関数を高速に最少化する。

$$E(X) = -\sum_{\{i,\,j\}} W_{ij}\, x_i\, x_j - \sum_i b_i x_i \qquad (1)$$

$$x_i \in \{0, 1\} (i = 1, 2, \cdots, N),\ W_{ij} = W_{ji}$$

ここでXはビットの組であり，$X = (x_1, x_2, \cdots x_N)$である。組み合わせ最適化問題に表れる組み合わせは，N個のビット値$x_i (i = 1, 2, \cdots, N)$で表される。$W_{ij}$はビットiとビットj間の結合係数，$b_i$は各ビットに対するバイアス項である（**図-2**）。このハードウェアの動作サイクルは，受け入れ基準を満たすようなビット反転を選ぶ試行フェーズと，選出されたビットを反転させる更新フェーズの二つに分けられる。

試行フェーズでは，現行のビットの組$X = (x_1, x_2,$

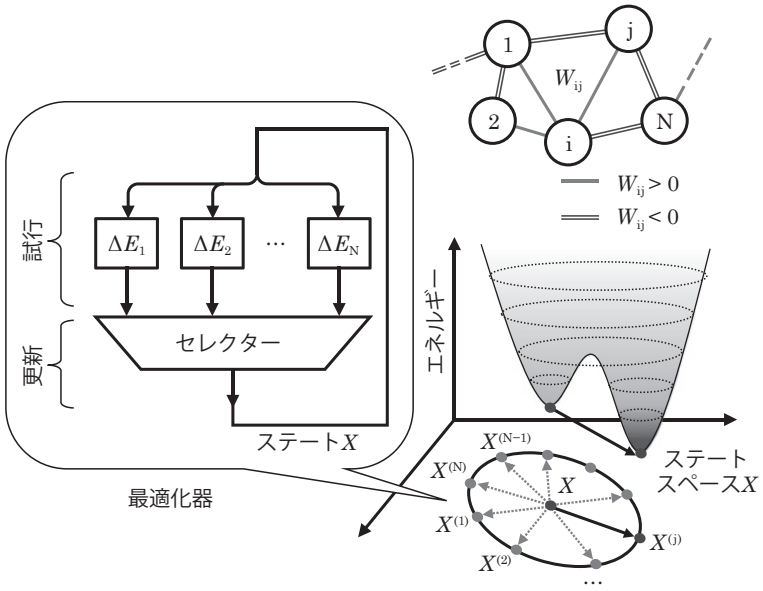


図-2　イジングモデルによる最適化の原理

$\cdots x_N$) から一つのビット値x_iを$1-x_i$に反転させることによって得られるN個の隣接状態$X^{(i)}$に着目する。現在の状態Xから隣接状態$X^{(i)}$に移行することで得られるエネルギー$E(X)$の増加は，以下の式で与えられる。

$$\Delta E_i = -(1-2x_i)\, h_i \qquad (2)$$

$$h_i = \sum_j W_{ij}\, x_j + b_i \qquad (3)$$

ここで，h_iをニューラルネットの用語を借りてi番目のビットに対する局所場と呼ぶ。i番目のビット値x_iが反転によって増加する場合（0から1になる場合），エネルギー$E(X)$はh_iだけ減少する。逆に，反転によりビットiの値が減少する場合（1から0になる場合）は，エネルギーはh_iだけ増加する。1回の更新フェーズで最大1個のビットが変化し，更新された局所場の値h_i (i=1, 2, $\cdots N$) をレジスタに記憶するアーキテクチャーを用いた（**図-3**）。例えば，更新フェーズでビット値x_jが$1-x_j$に更新（反転）されたとき，h_iの変化分δh_iは式（3）より

$$\delta h_i^{(j)} = W_{ij}\,(1-2x_j) \qquad (4)$$

となる。ビットごとにレジスタに記憶してある局所場の値に，式（4）で与えられる変化分を加算することで，局所場の値を並列に更新する。

今回の設計では，チップ上のビット数Nは1,024であり，これらのビット中の任意のビットiとjが結合係数W_{ij}で結合可能である。結合係数W_{ij}は16ビットの符号付き2進固定小数点数，局所場h_iは27ビットの符号付き2進固定小数点数，バイアスb_iは26ビットの符号付き2進固定小数点数で表現される。バイアスb_iは，局所場h_iとビットiの初期値x_iを式（3）を満たすように設定することで与えられる。

試行フェーズでは，式（5）で与えられるMetropolis-Hastings（M−H），またはGibbsの基準を用いてビット反転の受け入れ可否を決定する。

$$A(\Delta E_i) = \begin{cases} \min[1, \exp(-\beta \Delta E_i)] & \text{(M−H)} \\ 1/[1+\exp(\beta \Delta E_i)] & \text{(Gibbs)} \end{cases} \qquad (5)$$

ここで$A(\Delta E_i)$は，ビットx_iが$1-x_j$に反転された場合のエネルギー変化がΔE_iである場合の反転受け入れ確率であり，β (=1/T) はSA法で使われる温度Tの逆数である。ビットごとに設けた受け入れ決定ブロック（ADB：Acceptance Decision Block）により，各状態変化に対するΔE_iの値と適切な値を取る乱数（ノイズ）との比較を行うことにより，式（5）の受け入れ確率で値1を取る2値フラグを出力する。

このようなノイズを発生するには，0から1の間で均一に分散する一様乱数r_iを発生させ，テーブル参照により$A^{-1}(r_i)$が出力される（図-3）。この2値フラグの値が1であることは，対応するビットの反転を行ってよいことを表す。最終的には，2値フラッグの値が1のものから反転するビット1個を更新セレクターを使って選ぶ（**図-3，4**）。

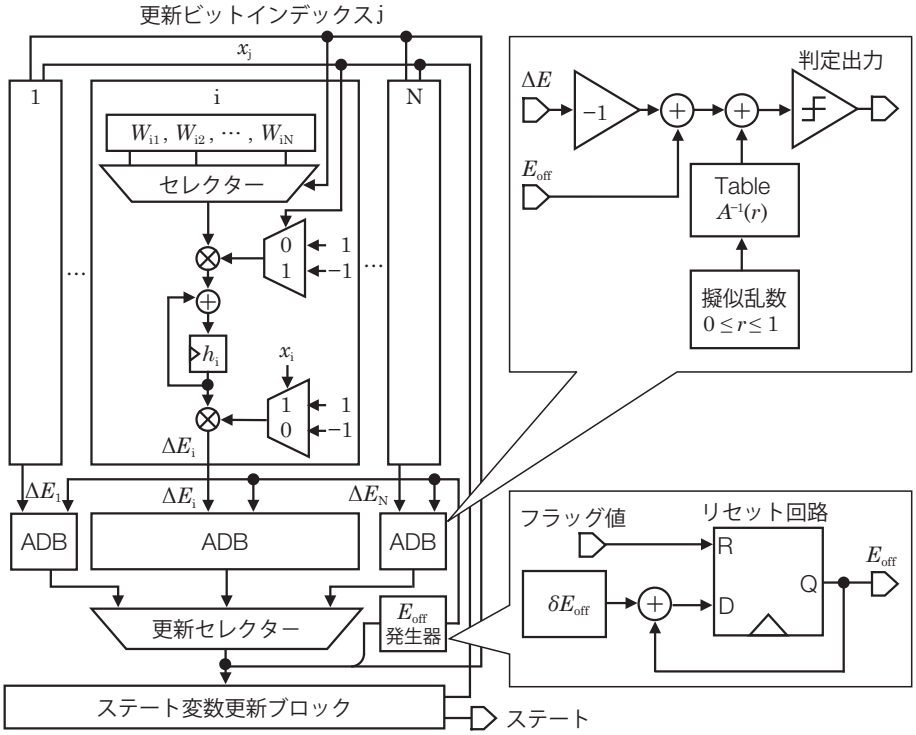


図-3　最適化器の構成

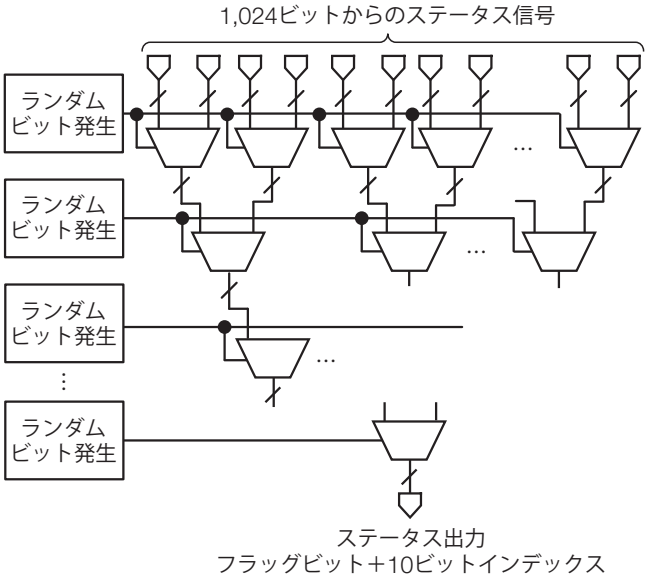


図-4　更新セレクターの構成

高速化手法1：並列試行

更新セレクターは，受け入れ決定ブロックが発生したフラグビットを基に，更新候補ビットのインデックスを発生する。このために，10段の2-1セレクターを用いた（図-4）。もし，2-1セレクターの両方の入力が次の反転の候補である場合，いずれか一方をランダムに選択する。もし更新できるビットの候補が一つもない場合には，最終段のセレクターがフラッグ値0を出力する。

この方式は，系が次の更新フェーズで反転できるビットを並列で探索するため，反転が発生する

確率が高くなり収束の速度が改善される。シミュレーションでは，32都市のTSPを速度評価のためのベンチマークとして用い，巡回距離が最小値に到達するまでのサイクル数が試行の並列数に反比例することが確認された｛**図-5**（a)｝。この手法は，複数のビットを並行して更新する並列試行と異なり，収束性に問題が発生しない利点がある。

高速化手法2：ダイナミックオフセット

系の状態がイジング型エネルギー関数の極小値（Local Minimum）に陥った場合，そこから抜け出して別の状態に移行する確率は，並列試行を用いてもかなり低くなる可能性がある。その場合には，系は何サイクルもの間，同じ極小値にとどまり，収束に時間がかかるようになる。Rejection-free Metropolis手法[9]では，ビット反転が発生する確率を正規化し，受け入れ確率の合計が1になるようにすることで，1サイクルで極小値から抜け出すことができる。

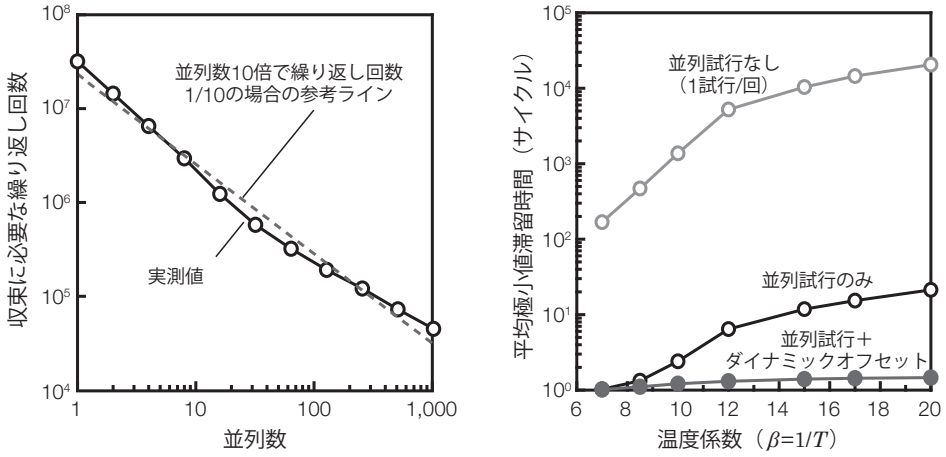
この手法は効果的ではあるが，受け入れ確率を正規化するための計算上のオーバーヘッドが大きい。この計算オーバーヘッドを減らすために，エネルギーの増加分から正のオフセットE_{off}を減算する手段を入れた。これは，共通の因子exp(β・Eoff) >1を状態反転の受け入れ確率に乗じたこととほぼ等価である。これを実行するために，新しい反転ビット候補が見つからない場合に，オフセット発生器を用いて一定の増分値をオフセットに加えていく。

オフセットの増加は，次の状態が見つかるまで続く。この方法により，オフセットの値は次の状態反転先が見つかる確率が1になるようにダイナミックに制御され，極小値にとどまる時間が短縮される｛図-5（b)｝。

ビット間の結合

筆者らのハードウェアでベンチマーク問題であるTSPをイジングモデルを用いて解く場合，巡回経路を表すためには訪れる順番を表す番号（時刻）と，訪れる都市の番号の組み合わせで表される数のビットが必要になる。32都市の場合，都市間の距離を表すには少なくとも10^{-3}程度以上の分解能が必要である。また，都市を1回ずつ順番に回る解のみを許容するペナルティ項を導入するため，多数のビット間の結合が必要となる。したがって，重み係数の分解能が小さいハードウェアや，物理的に近接したビット間の結合しか許されないハードウェアを用いて，TSPを解くのは非常に困難である。筆者らのハードウェアは，各ビットがほかの全てのビットと結合を持つ全結合で，かつ16ビット（65,536諧調）で重み係数の設定が可能であるため，32都市のTSPを解くことができた。

また，組み合わせ最適化問題のベンチマークとして使われることの多い，最大カット問題と呼ばれるグラフ問題の標準的問題では，重み係数の分解能は必要とされないものの，広範囲なビット間の結合が必要とされる。筆者らのハードウェアは



（a）並列化の効果　　（b）極小値滞留時間

図-5　32都市TSPでの高速化手法の効果

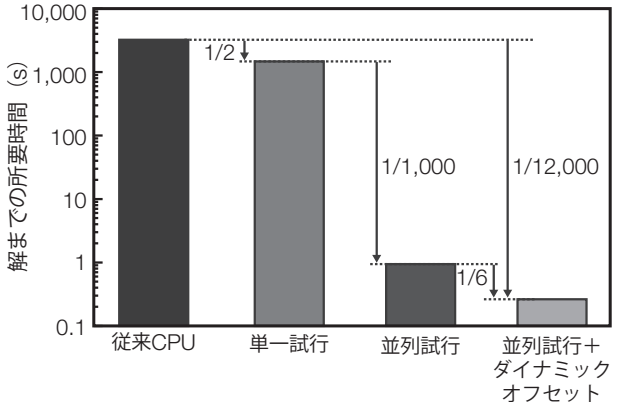


図-6　32都市TSPでの評価結果

この要求を満たしているため，最大カット問題も解くことができる。

性　能　評　価

提案されたハードウェアを，Altera社製の評価ボードArria 10 GX（1 GバイトDDR4 SDRAM付き）に実装した。今回の設計では，更新フェーズと試行フェーズを実行するのに合計5クロックサイクルかかる。同じ処理を従来のCPU上で実行すると，約350クロックサイクルとなる。今回試作したArria 10のクロック周波数は100 MHzであり，比較対象となる従来のCPUは3.5 GHzのIntel社製Xeon E5-1620 v3プロセッサである。このため，探索の並列化を行わない場合には，筆者らのハードウェアの速度はプロセッサ比で約2倍になると予想される。

解が得られる時間として，内製のSAのコードを使い32都市TSPで99％の確率で正解を得られるまでの時間で比較した。筆者らのハードウェアを並列試行を行わずに動作させた場合は，予想どおりプロセッサの約2倍の速度が得られた。一方，1,024並列の並列試行を使うと，並列試行を使わない場合に比べて速度は約1,000倍になった。また，これにダイナミックオフセット手法を併用すると，更にこの6倍に高速化され，全体としてプロセッサ比で約12,000倍に高速化した（**図-6**）。

む　す　び

本稿では，全結合イジングモデルを最適化するハードウェアアーキテクチャーについて報告した。このアーキテクチャーは，確率的な並列探索技術とダイナミックオフセット技術を用いて，32都市のTSPを従来のCPUで同じアルゴリズムを用いて解いた場合と比較して，約12,000倍の高速化を達成した。本アーキテクチャーは，ビット間が全結合で結合重みが16ビット（65,536諧調）であるため，様々な問題の解決に適用が可能である。今後は，実際の社会問題への適用に向けて研究を進める。

参考文献

（1）R. Colwell：The Chip Design Game at the End of Moore's Law. Hot Chips 27, 2015.

（2）J. G. Koomey et al.：IEEE Annals of the History of Computing, July-Sep. p.46-54, 2011.

（3）M. Horowitz：Computing's Energy Problem：(and what we can do about it). ISSCC2014, 1-1, 2014, (referring to Markovic, EE292 Class, Stanford, 2013).

（4）P. Bunyk et al.：Architectural Considerations in the Design of a Superconducting Quantum Annealing Processor. IEEE Trans. Applied Superconductivity, Vol.24, No.4, 2014.

（5）S. Utsunomiya：Mapping of Ising models onto injection-locked laser systems. OPTICS EXPRESS, Vol.19, No.19, Sep. 2011.

（6）P. A. Merolla et al.：A million spiking-neuron integrated circuit with a scalable communication network and interface. Science 8 August 2014, Vol.345, No.6197, p.668-673.

（7）Y. Chen et al.：DaDianNao：A Machine-Learning Supercomputer. 47th IEEE/ACM Int. Symp. on Microarchitecture, p.609-622, 2014.

（8）K. Hukushima and K. Nemoto：Exchange Monte Carlo Method and Application to Spin Glass Simulations. J. Phys. Soc. Jpn. Vol.65, p.1604-1608, 1996.

（9）H. Zhu et al.：Boltzmann Machine with Non-rejective Move. IEICE Trans. Fundamentals of Electronics, Vol.E85-A, p.1229-1235, Jun. 2002.

著者紹介

塚本三六（つかもと　さんろく）
コンピュータシステム研究所
次世代コンピューティングプロジェクト
組み合わせ最適化問題を解くための，ソフトウェアを含めたシステムに関する研究に従事。

高津　求（たかつ　もとむ）
コンピュータシステム研究所
次世代コンピューティングプロジェクト
組み合わせ最適化問題を解くための，ソフトウェアを含めたシステムに関する研究に従事。

松原　聡（まつばら　さとし）
コンピュータシステム研究所
次世代コンピューティングプロジェクト
組み合わせ最適化問題を解くための，ソフトウェアを含めたシステムに関する研究に従事。

田村泰孝（たむら　ひろたか）
コンピュータシステム研究所
次世代コンピューティングプロジェクト
組み合わせ最適化問題を解くための，ソフトウェアを含めたシステムに関する研究に従事。