

疑似量子計算シミュレーテッド分岐アルゴリズムによる リアルタイム組合せ最適化システム

濱川 洋平¹ 日高 亮¹ 山崎 雅也¹ 辰村 光介¹

概要：組合せ最適化は、社会や産業の様々な場面で頻繁に現れる重要課題である。多くの組合せ最適化問題は、変数が増えると取り得る組合せが膨大となり、解くのが困難となる。イジングマシンは、このような組合せ最適化の近似解を高速に解くことができる専用計算機であり、シミュレーテッド分岐は、一種のイジングマシンを実現する量子インスパイアなアルゴリズムである。高い計算並列度を有するシミュレーテッド分岐を FPGA (Field Programmable Gate Array) に実装したアクセラレータは、特に低レイテンシなイジングマシンを具現化できることが特長である。本稿では、高度な組合せ最適化を利用するリアルタイムシステムに向けた FPGA 実装イジングマシンの応用について述べる。また、汎用イジングマシンを用いたリファレンスデザインを開発し、様々な分野の研究者が誰でも再現可能な形でリアルタイム組合せ最適化システムの実現可能性を示した。

Combinatorial optimization in real-time systems by FPGA-based accelerators for simulated bifurcation

YOHEI HAMAKAWA¹ RYO HIDAKA¹ MASAYA YAMASAKI¹ KOSUKE TATSUMURA¹

1. はじめに

金融取引の最適化や、産業用ロボット動作の最適化、移動経路の最適化など、社会・産業システム上の課題の多くは、膨大な組合せパターンから最良のものを選び出す組合せ最適化問題に帰着する。これは、解の候補数が問題規模の指数関数で増加する、いわゆる組合せ爆発のために、解くことが大変難しい問題として知られている。そのため、組合せ最適化専用計算機の研究開発が国内外で活発に行われている。これらの多くはイジングマシンと呼ばれるもので、磁性体の簡易モデルであるイジングモデルの基底状態を求める問題を解く専用計算機である。多くの組合せ最適化問題がイジング問題に定式化できることから [1]、イジングマシンに高い期待が持たれており、2011 年に D-wave 社が量子アニーラ [2] を商用化して以来、様々な方式のイジングマシンが広範囲に研究されている [3-9]。

これまで、金融や通信、ロボットなどの分野で見られるリアルタイムシステムでは、複雑な組合せ最適化を行うこ

とへの潜在的要求があっても、特定時間内の応答を要するという時間的制約から、それを実現できなかった。近年開発されているイジングマシンは、要求される時間制約の下で選択し得る全ての候補の中から最良の応答を選ぶ可能性を開く。

本稿では、イジング問題を解くための疑似量子アルゴリズムであるシミュレーテッド分岐 [10,11] と、FPGA による高性能なアクセラレータの設計および実装 [12,13]、金融分野での応用事例 [14] について解説した上で、FPGA 実装の汎用イジングマシンを使用した種々のリファレンスデザインを開発し、刻々と変化する状況に対して最適な応答をするリアルタイムシステムへの高度な組合せ最適化の適用可能性を実証する。

2. シミュレーテッド分岐アルゴリズムの FPGA 実装

2.1 シミュレーテッド分岐

シミュレーテッド分岐 (Simulated Bifurcation, 以下, SB と略す) [10,11] は、量子分岐マシンと呼ばれる量子断熱最

¹ 株式会社 東芝
Toshiba Corporation

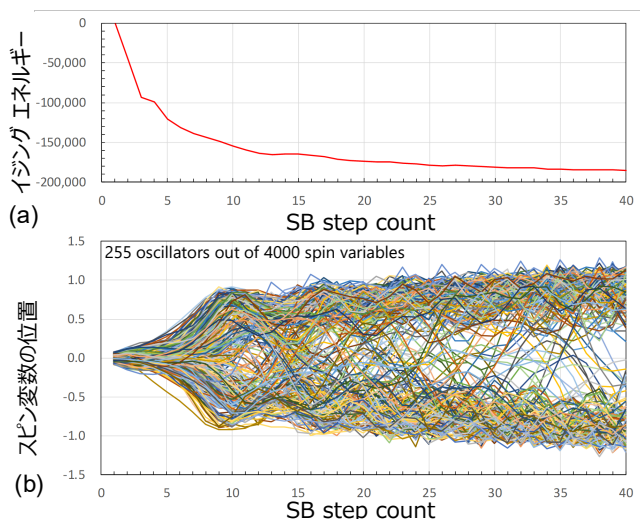


図 1 イジングエネルギー (上) と振動子位置の時間発展 (下) [12]

適化手法 [15,16] に対する古典対応物から導かれた、疑似量子 (量子インスパイアド) 最適化計算アルゴリズムである。SB は分岐現象を示す古典的な非線形振動子ネットワークの断熱時間発展を数値的に模擬すること、つまり、古典計算機による力学系のシミュレーションをベースとしたアルゴリズムであることが特徴的な点である。

SB では、1つの振動子 (例えば i 番目の振動子) を実数である位置変数 x_i と運動量変数 y_i によって表現する。SB のステップ処理とは、すべての変数 (x, y) を離散時刻 t_n から次の時刻 $t_n + \Delta t$ へ更新する手続きをいう。所定回数のステップ処理の完了後、連続変数である位置 x_i を二値化したものが、イジングモデルにおける i 番目のスピン (± 1) に相当する。これが、イジング問題への最適解もしくは良解となる。シミュレーテッド分岐は、 N 体の相互作用する振動子の時間発展ダイナミクスを通じて、イジング問題の (良) 解を得るヒューリスティクスである。

図 1 は 4000 スピン規模の問題を対象に、SB ステップが進む (時間発展する) につれ振動子が分岐し、分岐とともにイジングエネルギーが減少する様子を示す。SB ステップが進行するにつれてイジングエネルギーは減少していることから、系全体は基底状態もしくはそれに近い局所安定状態へ近づいていることが分かる。

2.2 基本回路アーキテクチャ

SB アルゴリズムの最大の特長は、アルゴリズムに内在する並列度が高いことにある。これは SB アルゴリズムが並列計算により高速化できることを意味する。また、必要な演算が和と積だけで、特殊な関数の計算が不要なことも、FPGA などによるハードウェア化にとって望ましい。

これらの特長を生かして開発した SB アクセラレータの回路アーキテクチャ [12] を図 2 に示す。SB ステップ処理は、行列ベクトル積 (matrix-vector multiplication: MM)

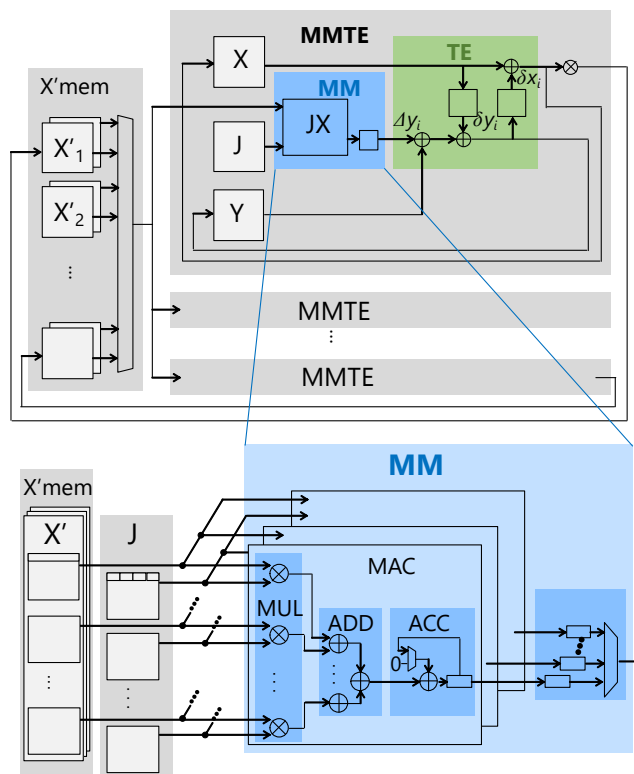


図 2 シミュレーテッド分岐のアクセラレータの基本回路アーキテクチャ [12]

と時間発展 (time evolution: TE) という主たる 2つの部分から構成される (SB の疑似コードについては、文献 [12] を参照)。MM 部は、各振動子について、他の振動子との多体相互作用に基づく運動量の修正量 (Δy_i) を計算する。TE 部は、各振動子について、はじめ運動量を Δy_i だけ更新し、続いて自己の運動量と 1 体ポテンシャルから計算される修正量 δx_i と δy_i だけ、位置と運動量を更新する。

MM 部は、多数の積和演算器を並列に配置し、それらへデータを十分に供給するためのメモリサブシステムを構築することにより高速化した (空間並列化)。TE 部は、逐次処理であるが各粒子について独立であるため、パイプライン回路により高速化した (時間並列化)。他にも、データバスの多重化や MM 処理と TE 処理のオーバーラップ実行といった高速化実装も行っている。

本アーキテクチャをベースに FPGA 実装した第 2 世代 SB アルゴリズムである ballistic SB (bSB) と discrete SB (dSB) は、種々のベンチマーク問題で極めて高い性能を示した [11]。

2.3 スケールアウトアーキテクチャ

前節におけるシングル FPGA 実装の SB アクセラレータは、高速読み書きが可能な FPGA のオンチップメモリ資源を最大限利用することで高速演算および低レイテンシ性 (高速応答性) を実現する。このことは逆に、シングル FPGA 実装の最大計算機サイズ (取り扱い可能な最大問題

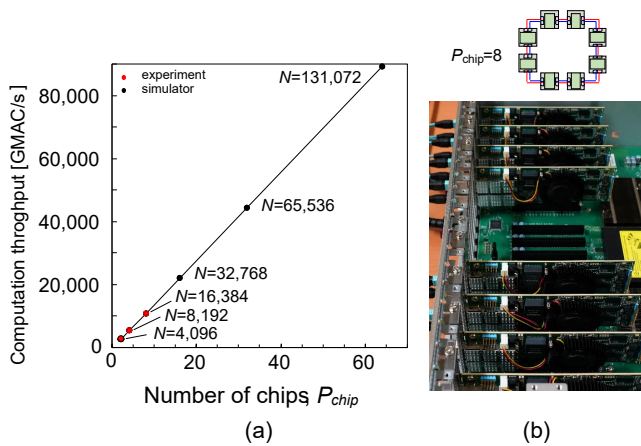


図 3 シミュレーテッド分岐のスケールアウトアーキテクチャ [13]: (a) 効率一定 (weak) スケーリング特性, (b) 双方向リングトポロジによる複数チップの接続 (チップ数 8)

サイズ) は, オンチップメモリ資源によって制限される, と言える.

そこで著者らは, 複数の FPGA を接続することで計算機サイズと計算スループットを増大することを可能とする, 全結合イジングマシンのスケールアウトアーキテクチャを開発している [13].

開発したアーキテクチャは, 自律的な同期メカニズムを特徴とし, 各チップは, 独立に起動後, 隣接チップ間でのスピン情報の交換過程を通じて自律的に同期する. この自律的な同期メカニズムが通信オーバーヘッドによる性能の飽和を回避させ, 計算スループットのスケールアップを可能とする.

本アーキテクチャをベースに試作した FPGA クラスタ (8つの FPGA チップを光ファイバで結合) は, 計算効率一定のスケールアップ特性 (問題サイズとチップ数を同じ割合で増大) を実証した (図 3). このことは, 全結合型でありながら, FPGA チップやサーバーを増設することで, FPGA 実装 SB アクセラレータの高速演算と低レイテンシ性と, 規模拡張を両立可能であることを意味する.

3. FPGA 実装版 SB アクセラレータの応用

3.1 裁定取引マシンの実証

最適な応答をするリアルタイムシステムの一例として, 外国為替間の裁定取引をモチーフにしたコンセプト実証機を開発した [14].

裁定取引は同じ (もしくは本質的に類似の) 資産の買いと売りを 2つの異なる市場 (状況) で実行することによって, 価格差に基づく利益を得ることをいう. 通貨裁定機会の検出問題は, 通貨をノードに, 通貨間交換レートをエッジに割り当てた重み付き有向グラフにおける最適経路探索問題として定式化できる. これは典型的な組合せ最適化問題である.

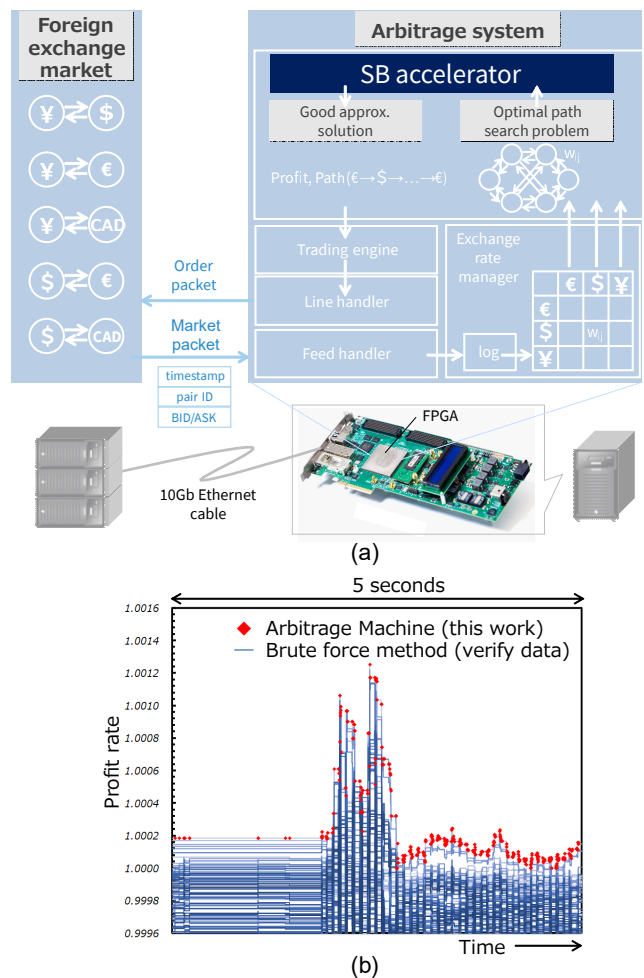


図 4 シミュレーテッド分岐に基づく裁定取引マシン [14]: (a) システムアーキテクチャ, (b) 裁定取引経路の利益率の時間変化

図 4(a) は SB ベース裁定取引マシンのシステムアーキテクチャを示す. これはシングル FPGA に実装されたエンドツーエンドシステムで, その中では, フィードハンドラが不定期に発行される市況パケットを取り込み, SB アクセラレータが経路探索問題を解き最良裁定経路を発見し, ラインハンドラが注文パケットを発行する. 著者らは 8 通貨 15 ペアの外国為替取引の実データ (細粒度データ) を用い, 市況パケットの到着から注文パケットの発行までのシステムワイドレイテンシ (応答時間) を測定した (図 4(b)).

この結果, 刻々と変化する外国為替市場において膨大な通貨の組合せパターンの中から利益率が最大となる裁定取引の機会を 90% 以上の高確率で発見し, 売買注文の発行までを 30 マイクロ秒以内の時間で完了できることを実証した.

3.2 汎用イジングマシン

前節のコンセプト実証機は, 裁定取引というアプリケーションに特化し, SB アクセラレータだけではなく (周辺の取引処理, 専用インターフェースなど含む) システム全体を最適化した上で, これらを全て FPGA 実装したソリュー

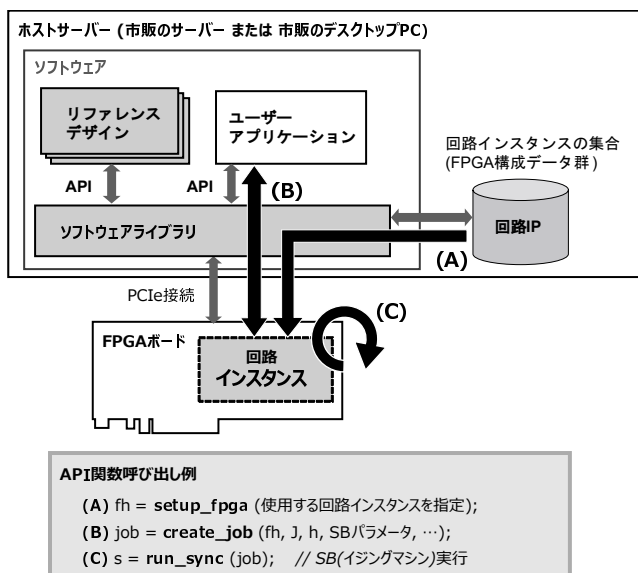


図 5 FPGA 実装 SB アクセラレータによる汎用イジングマシンのシステム構成

ションであった。対象とするアプリケーションが決定し、そのパフォーマンス要求を達成する上で必要な場合は、このようなカスタムメイドのシステム設計を行うことは有望な選択肢の 1 つである。一方、実証実験の初期段階 (アプリケーションの要求事項が曖昧、アプリケーション自体の探索段階) ではより汎用的かつ容易に利用できる試作環境があることが望ましいと考えられる。

そこで、様々なバックグラウンドを持つ研究者が、このような革新的なリアルタイムシステムの研究開発を手軽に手掛けられるよう、SB アクセラレータに汎用的なイジングマシンインタフェースを付加した汎用イジングマシン (ソルバ) を開発した。

開発した汎用イジングマシンのシステム構成を図 5 に示す。ハードウェアは、サーバーまたは PC と、市販の FPGA ボードだけであり、他の特殊な機材を必要としない。ソフトウェアライブラリは、FPGA に対する一連で複雑な処理を隠蔽し、C/C++ 及び Python で利用できる API (Application Programming Interface) を提供する。図 5 下部に示すよう、最小 3 つの極めてシンプルな API 関数群を使って SB アクセラレータを利用することができる。また、SB アクセラレータは、第 2 世代 SB アルゴリズム [11] である bSB と dSB の 2 者に対し、それぞれ精度と規模が異なる複数バリエーションの FPGA 回路構成データ (回路インスタンス) を用意し、ユーザーアプリケーションの用途に応じて (API 経由で) 使い分けが可能である。

これら FPGA 回路構成データとソフトウェアライブラリ、後述のリファレンスデザインを含むパッケージは、オンプレミス版のシミュレーテッド分岐マシン™ として一般のユーザーに提供されている。

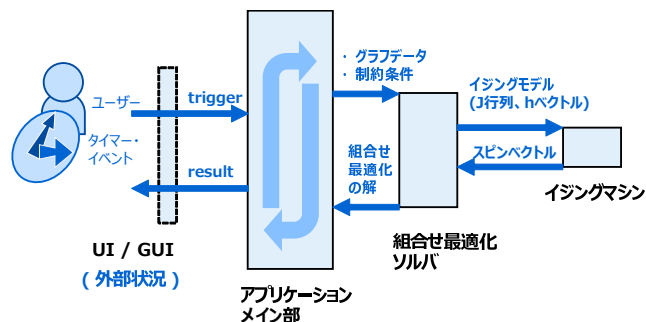


図 6 汎用イジングマシンを用いたアプリケーション構築のための簡易フレームワーク

4. 汎用イジングマシンを用いたリアルタイムシステムの実証

4.1 リファレンスデザイン

前章で説明した汎用イジングマシンは、高度な組合せ最適化を使ったユーザーのリアルタイムシステムの試作に活用されることを期待する。ここで、専門外の様々な技術分野のユーザーを対象とするためには、この汎用イジングマシンで実現できること (アプリケーションのイメージ、速度性能) や、プログラミング方法などを分かりやすく提示する手段が必要である。

ここでは、本マシンのリアルタイム性を実証 (デモンストレーション) し、ユーザーのシステム開発を促進する目的で、リファレンスデザインを開発した。

リファレンスデザインの設計に先立ち、SB アクセラレータによる汎用イジングマシンを使った組合せ最適化処理を、ユーザーのアプリケーションで利用するための簡易的な参照構造として、図 6 に示す共通のソフトウェアフレームワークを定義した。このフレームワークでは、下記の 4 階層を定義する。

- **イジングマシン:** FPGA 実装 SB アクセラレータによる汎用イジングマシン。
- **組合せ最適化ソルバ:** 組合せ最適化問題をイジングモデルに変換し、API 経由でイジングマシンに問題を投入した後、イジングマシンの解として得られたスピネクトルを元の組合せ最適化問題の解に復元し、アプリケーションメイン部に返す。このとき、アプリケーションの時間制約が許せば、複数回イジングマシンを起動し、複数解から最良のものを 1 つ選んで返す、などの処理を行っても良い。
- **アプリケーションメイン部:** アプリケーションのメイン処理を行いながら、組合せ最適化問題に帰着される部分を、グラフや行列データおよび制約条件の形式で整理し、問題処理を組合せ最適化ソルバに委譲する。組合せ最適化ソルバからの解を用いてアプリケーションのメイン処理を継続する。
- **UI (User Interface) / GUI (Graphical User**

Interface): デモンストレーションにおいて、ユーザーに対するグラフィカルな画面表示、インタラクティブな操作のインターフェースを提供する。実際のリアルタイムシステムにおいては、ユーザー操作やタイマーによる割り込み、センサ値の変化や各種イベントによる「外部状況」に相当する階層であると考えても良い。

今回、3つのリファレンスデザインを開発したが、これらはいずれもこのフレームワークに沿って設計した。また、各階層の要求事項に応じて異なるプログラミング言語を使用した。例えば、高速処理を期待する組合せ最適化ソルバ部はC/C++、複雑な処理を担うアプリケーションメイン部はPython、GUIにおいては関連ライブラリが充実しているPythonやJavaScriptなど使い分けている。

4.2 組合せ最適化ソルバの設計 (イジングモデルへの定式化)

イジングマシンは、次の式で表現されるイジングモデルのエネルギー $E(s)$ を最小化するスピンを探索する。

$$E(s) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N J_{ij} s_i s_j + \sum_{l=1}^N h_l s_l \quad (1)$$

ここで、 s_i は i 番目のスピン、 N はスピンの総数、 J_{ij} はスピン間の相互作用の強さ、 h_i は s_i にかかる外部磁場の強さを表す。

イジングマシンを利用して組合せ最適化問題を解くためには、元の組合せ最適化問題において最適化すべき目的変数をスピン変数 $s_i \in \{-1, 1\}$ で表現し、最小化(または最大化)すべき目的関数をイジングモデル、つまり J_{ij} , h_i に変換(定式化)する必要がある。典型的な組合せ最適化問題をイジングモデルに変換する手法については文献 [1] が詳しく、本稿のリファレンスデザインの一部でも採用している。

また、イジングモデルと等価な問題表現に QUBO (Quadratic Unconstrained Binary Optimization) と呼ばれるものがある。これは、 $b_i \in \{0, 1\}$ で表すバイナリ変数を目的変数とし、以下の式で定義されるコスト関数を最小にする問題である。

$$H = \sum_{i=1}^N \sum_{j=1}^N Q_{ij} b_i b_j \quad (2)$$

QUBO はイジングモデルを線形変換したものであり、スピン変数 s_i とバイナリ変数 b_i の対応関係は次式で表現される。

$$b_i = \frac{s_i + 1}{2} \quad (3)$$

元の組合せ最適化問題によっては、バイナリ変数 b_i を用

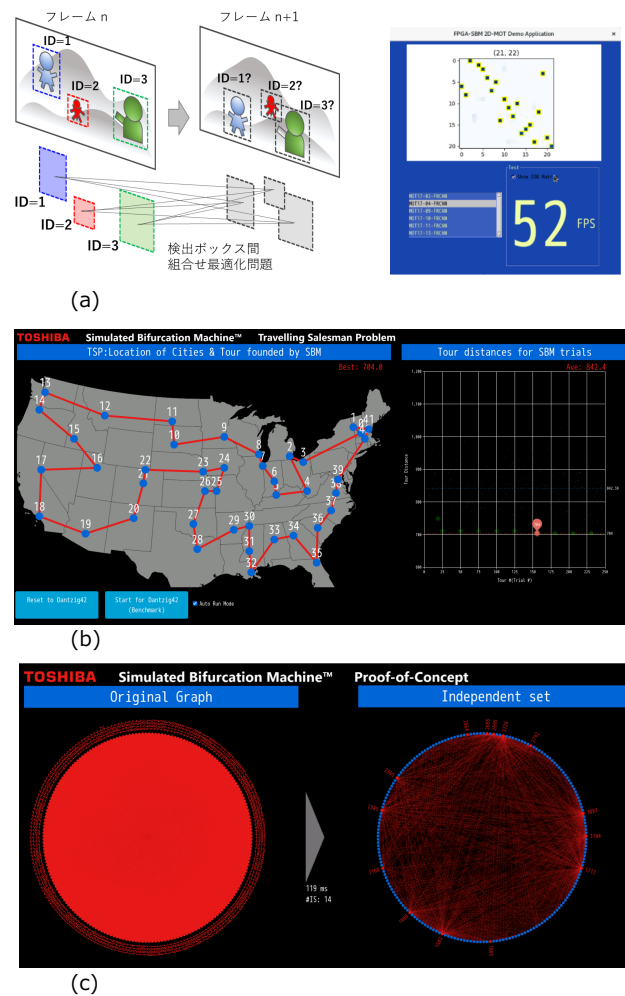


図 7 リファレンスデザイン: (a) マルチオブジェクトトラッキング, (b) インタラクティブ巡回セールスマン問題, (c) ストリームデータ処理型最大独立集合検出

いることで直感的に定式化できるものもあり、そのような場合には一度 QUBO 形式へ定式化した後でイジングモデルに変換する手続きをとることもある。なお、次節以降のリファレンスデザイン解説においては、全て QUBO 形式への変換例を示す。

4.3 マルチオブジェクトトラッキング

既存の複数物体追跡手法 [17] をベースに、動画像中の複数物体追跡を実装したアプリケーション例である (図 7(a))。追跡処理の一部は、フレーム間の物体の対応関係が最も尤もらしい組合せを求める 2 部グラフ最大マッチング問題に帰着される。この最適化問題をイジング問題に定式化し、1 フレーム毎に解を求める。

本アプリケーションにおける組合せ最適化ソルバ部 (2 部グラフ最大マッチング問題ソルバ) の実装のうち、QUBO 形式への定式化方法を以下に示す。

ここではまず、2つのフレーム (それぞれ過去フレーム、現フレームと呼ぶことにする) のうち、過去フレーム内の p 番目のオブジェクトと現フレーム内の c 番目のオブジェ

クトに対するマッチング有無を QUBO の目的変数として以下で定義した。

$$b_{p,c} = \begin{cases} 1 & (p \text{ 番目と } c \text{ 番目のオブジェクトが一致}) \\ 0 & (\text{それ以外}) \end{cases} \quad (4)$$

ここで N , M はそれぞれ過去フレーム内のオブジェクトの数, 現フレーム内のオブジェクトの数を表す. これらの変数を使って QUBO 問題に定式化したものが数式 (5) である. なお, A , B はそれぞれ制約項, 目的関数の重み係数である.

$$H = BH_{cost} + A(H_{st1} + H_{st2}), \quad (5)$$

$$H_{cost} = - \sum_{p=1}^N \sum_{c=1}^M \text{IOU}(p, c) b_{p,c}, \quad (6)$$

$$H_{st1} = \begin{cases} \sum_{p=1}^N \left(\sum_{c=1}^M b_{p,c} - 1 \right)^2 & (M \geq N) \\ \sum_{p=1}^N \sum_{c \neq c'}^M b_{p,c} b_{p,c'} & (M < N), \end{cases} \quad (7)$$

$$H_{st2} = \begin{cases} \sum_{p \neq p'}^N \sum_{c=1}^M b_{p,c} b_{p',c} & (M > N) \\ \sum_{c=1}^M \left(\sum_{p=1}^N b_{p,c} - 1 \right)^2 & (M \leq N). \end{cases} \quad (8)$$

数式 (6) の H_{cost} はイジングマシンが最小化すべき目的関数である. ここで $\text{IOU}(p, c)$ はオブジェクト p とオブジェクト c の領域の重なり具合を示す指標 (Intersection Over Union) である. H_{cost} が最小となるのは, N 対または M 対の p と c の組合せに対する IOU の総和が最大になるときであり, このときの p と c の組合せの集合が尤もらしいマッチングであると判断する.

次に制約条件を定式化する. 今回のアプリケーションにおいては, 問題を簡単にするため, 過去フレームからオブジェクトがフレームアウトする事象と, 現フレームにオブジェクトがフレームインする事象は同時には起きない, という前提を置くことにする. このとき, N と M の大小関係に応じて下記のことが言える.

- $N = M$ のとき, 現フレーム中のオブジェクトと, 過去フレーム中のオブジェクトは 1 対 1 の関係を持つ
- $M > N$ のとき, 現フレーム中の $M - N$ 個のオブジェクトは, 全て新規にフレームインするものであり, その他の N 個のオブジェクトは過去フレーム中のオブジェクトと 1 対 1 の関係を持つ
- $M < N$ のとき, 過去フレーム中の $N - M$ 個のオブジェクトは, 全てフレームアウトするものであり, その他の M 個のオブジェクトは現フレーム中のオブジェ

クトと 1 対 1 の関係を持つ

これら制約条件を QUBO のペナルティ項として定義したものが数式 (7) と数式 (8) である. 例えば数式 (7) の上段は, 過去フレーム中の N 個の各オブジェクトは, 現フレーム中のオブジェクトのいずれかに必ず 1 つだけ割り当てられる (ときに最小となる) ことを表し, 下段は, 過去フレーム中の各オブジェクトが現フレーム中のオブジェクトに割り当てられる数が 1 以下 (1 または 0) である条件を示したものである. 数式 (8) は, 数式 (7) 中の主語を (過去フレームから現フレームへ) 入れ換えたものである.

なお, この問題に対してイジングマシンに必要なスピンの総数は NM である.

今回, 上記の組合せ最適化ソルバを含むアプリケーションを実装し, 速度性能を評価した. 結果として, 組合せ最適化ソルバにおけるイジングモデル生成からイジングマシン (SB アクセラレータ) での求解, その他追跡処理から出画処理まで含むシステム全体で, 秒間 30 フレーム以上の処理速度を達成しており, リアルシステムへの組込み可能性を端的に示すデモンストレーションを実現できたと言える.

4.4 巡回セールスマン問題

巡回セールスマン問題 (TSP) は, 都市の集合において全ての都市を 1 度だけ訪れる巡回路のうち総距離が最小である経路を求める, 有名な組合せ最適化問題である. 本アプリケーションは, 42 都市の TSP を対象とし, 図 7(b) に示す GUI から, マウスのドラッグ&ドロップ操作によりユーザーが都市の位置を任意に指定できるものとしている.

本アプリケーションの組合せ最適化ソルバ部では, 文献 [1] で提案されている手法に基づき, 以下のように QUBO 表現への定式化を行う.

まず, QUBO の目的変数を下記のように定義する.

$$b_{k,l} = \begin{cases} 1 & (1 \text{ 番目の都市を } k \text{ 番目に訪れる}) \\ 0 & (\text{それ以外}) \end{cases} \quad (9)$$

都市数を n とするとき, イジングマシンに必要なスピンの総数は n^2 となる. ここで, QUBO 問題に定式化したものが数式 (10) である. なお, A , B はそれぞれ制約項, 目的関数の重み係数であり, $d_{l,l'}$ は l 番目の都市と l' 番目の都市の間の距離を示すものとする.

$$H = BH_{cost} + A(H_{st1} + H_{st2}), \quad (10)$$

$$H_{cost} = \sum_{p=1}^n \sum_{l=1}^n \sum_{l'=1}^n d_{l,l'} b_{k,l} b_{k+1,l'}, \quad (11)$$

$$H_{st1} = \sum_{l=1}^n \left(\sum_{k=1}^n b_{k,l} - 1 \right)^2, \quad (12)$$

$$H_{st2} = \sum_{k=1}^n \left(\sum_{l=1}^n b_{k,l} - 1 \right)^2. \quad (13)$$

本アプリケーションでは、マウスのドロップ操作直後から即座に経路探索を開始し、複数回の探索を繰り返しながら更新された最小経路を順次画面表示する。このようなインタラクティブなインタフェースにより、SB アクセラレータによる高速処理や低レイテンシ性を視覚的に訴える手段を実現した。

本リファレンスデザインから想起される展開先として、位置が変化し続けるロボット群の移動パターンや通信経路の最適化などへの応用も考えられる。

4.5 最大独立集合問題

株式データから、市場グラフと呼ばれる、銘柄間の値動きの相関を表すグラフを生成し、最大独立集合問題を解くことで多様化ポートフォリオの構成候補となる銘柄セットを見つけるアプリケーション例である (図 7(c))。最大独立集合問題とは、ある集合から互いに相関がない (最大の) 部分集合を求める、NP 困難に属する組合せ最適化問題である。得られる銘柄セットは互いに値動きの相関が低いため、個別の銘柄の値動きに左右されない安定した株式ポートフォリオを構築できる。

組合せ最適化ソルバ部に相当する最大独立集合問題の QUBO 表現は、文献 [1] で提案されている手法に基づき設計した。

まず、QUBO の目的変数を下記のように定義する。

$$b_i = \begin{cases} 1 & (\text{頂点 } i \text{ は独立集合の要素である}) \\ 0 & (\text{頂点 } i \text{ は独立集合の要素ではない}) \end{cases} \quad (14)$$

このとき、問題対象のグラフを $G = (V, E)$ (V は頂点の集合、 E はエッジの集合) とすると、イジングマシンに必要なスピンの総数は頂点数 $|V|$ に等しい。また、QUBO 表現を数式 (15) に示す。ここでも、 A 、 B はそれぞれ制約項、目的関数の重み係数を表す。

$$H = BH_{cost} + AH_{st}, \quad (15)$$

$$H_{cost} = - \sum_i b_i, \quad (16)$$

$$H_{st} = \sum_{ij \in E} b_i b_j. \quad (17)$$

開発したアプリケーションでは銘柄数 (上限は 2048) に依らず約 120 ms で良解を返すことができ、刻一刻と変化するストリームデータ処理への適用可能性を実例として示した。

5. むすび

リアルタイムシステムにおいて大規模組合せ最適化を可能とする FPGA 向け SB アクセラレータを開発し、その応用事例を示した。また、既に一般ユーザーに提供開始している SB アクセラレータによる汎用イジングマシン (オンプレミス版のシミュレーテッド分岐マシン™) を用いたリファレンスデザインを開発し、高度な組合せ最適化処理を組込んだリアルタイムシステムの実現可能性を実証した。なお、本稿では言及しなかったが、リアルタイムシステムに限らず、SB アクセラレータの高速演算と低レイテンシ性を活かした独自の応用事例も報告されている [18]。また、本稿で言及した bSB、dSB [11] の登場以降も SB アルゴリズムの性能向上が報告されている [19]。本研究および本研究における汎用イジングマシンが、様々な分野で革新的なシステムを実現する道を開くことを期待する。

参考文献

- [1] Andrew Lucas. Ising formulations of many np problems. *Frontiers in Physics*, Vol. 2, , 2014.
- [2] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose. Quantum annealing with manufactured spins. *Nature*, Vol. 473, No. 7346, pp. 194–198, May 2011.
- [3] Takahiro Inagaki, Yoshitaka Haribara, Koji Igarashi, Tomohiro Sonobe, Shuhei Tamate, Toshimori Honjo, Alireza Marandi, Peter L. McMahon, Takeshi Umeki, Koji Enbutsu, Osamu Tadanaga, Hirokazu Takenouchi, Kazuyuki Aihara, Ken ichi Kawarabayashi, Kyo Inoue, Shoko Utsunomiya, and Hiroki Takesue. A coherent ising machine for 2000-node optimization problems. *Science*, Vol. 354, No. 6312, pp. 603–606, 2016.
- [4] D. Pierangeli, G. Marcucci, and C. Conti. Large-scale photonic ising machine by spatial light modulation. *Phys. Rev. Lett.*, Vol. 122, p. 213902, May 2019.
- [5] Takashi Takemoto, Kasho Yamamoto, Chihiro Yoshimura, Masato Hayashi, Masafumi Tada, Hiroaki Saito, Mayumi Mashimo, and Masanao Yamaoka. 4.6 a 144kb annealing system composed of 9×16 kb annealing processor chips with scalable chip-to-chip connections for large-scale combinatorial optimization problems. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, 第 64 巻, pp. 64–66, 2021.
- [6] Satoshi Matsubara, Motomu Takatsu, Toshiyuki Miyazawa, Takayuki Shibasaki, Yasuhiro Watanabe, Kazuya Takemoto, and Hirotaka Tamura. Digital annealer for high-speed solving of combinatorial optimization problems and its applications. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 667–672, 2020.
- [7] Kasho Yamamoto, Kazushi Kawamura, Kota Ando, Norman Mertig, Takashi Takemoto, Masanao Yamaoka, Hiroshi Teramoto, Akira Sakai, Shinya Takamaeda-Yamazaki, and Masato Motomura. Statica: A 512-spin 0.25m-weight annealing processor with an all-spin

- updates-at-once architecture for combinatorial optimization with complete spin-spin interactions. *IEEE Journal of Solid-State Circuits*, Vol. 56, No. 1, pp. 165–178, 2021.
- [8] Fuxi Cai, Suhas Kumar, Thomas Van Vaerenbergh, Xia Sheng, Rui Liu, Can Li, Zhan Liu, Martin Foltin, Shimeng Yu, Qiangfei Xia, J. Joshua Yang, Raymond Beausoleil, Wei D. Lu, and John Paul Strachan. Power-efficient combinatorial optimization using intrinsic noise in memristor hopfield neural networks. *Nature Electronics*, Vol. 3, No. 7, pp. 409–418, Jul 2020.
- [9] Antik Mallick, Mohammad Khairul Bashar, Daniel S. Truesdell, Benton H. Calhoun, Siddharth Joshi, and Nikhil Shukla. Using synchronized oscillators to compute the maximum independent set. *Nature Communications*, Vol. 11, No. 1, p. 4689, Sep 2020.
- [10] Hayato Goto, Kosuke Tatsumura, and Alexander R. Dixon. Combinatorial optimization by simulating adiabatic bifurcations in nonlinear hamiltonian systems. *Science Advances*, Vol. 5, No. 4, p. eaav2372, 2019.
- [11] Hayato Goto, Kotaro Endo, Masaru Suzuki, Yoshisato Sakai, Taro Kanao, Yohei Hamakawa, Ryo Hidaka, Masaya Yamasaki, and Kosuke Tatsumura. High-performance combinatorial optimization based on classical mechanics. *Science Advances*, Vol. 7, No. 6, p. eabe7953, 2021.
- [12] Kosuke Tatsumura, Alexander R. Dixon, and Hayato Goto. Fpga-based simulated bifurcation machine. In *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 59–66, 2019.
- [13] Kosuke Tatsumura, Masaya Yamasaki, and Hayato Goto. Scaling out ising machines using a multi-chip architecture for simulated bifurcation. *Nature Electronics*, Vol. 4, No. 3, pp. 208–217, Mar 2021.
- [14] Kosuke Tatsumura, Ryo Hidaka, Masaya Yamasaki, Yoshisato Sakai, and Hayato Goto. A currency arbitrage machine based on the simulated bifurcation algorithm for ultrafast detection of optimal opportunity. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2020.
- [15] Hayato Goto. Bifurcation-based adiabatic quantum computation with a nonlinear oscillator network. *Scientific Reports*, Vol. 6, No. 1, p. 21686, Feb 2016.
- [16] Hayato Goto. Quantum computation based on quantum adiabatic bifurcations of kerr-nonlinear parametric oscillators. *Journal of the Physical Society of Japan*, Vol. 88, No. 6, p. 061015, 2019.
- [17] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Uppcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, 2016.
- [18] Nasa Matsumoto, Yohei Hamakawa, Kosuke Tatsumura, and Kazue Kudo. Distance-based clustering using qubo formulations. *Scientific Reports*, Vol. 12, No. 1, p. 2669, Feb 2022.
- [19] Taro Kanao and Hayato Goto. Simulated bifurcation assisted by thermal fluctuation, 2022, to be published in *Communications Physics*.