

イジングマシンにおける共通ソフトウェア基盤開発

Research and Development of Common Software Platform for Ising Machines

松田佳希

Yoshiki Matsuda

(株) フィックスターズ
Fixstars Corporation, Japan

1. はじめに

量子アニーリングマシンを始めとして、組合せ最適化問題に特化したイジングマシンが注目を集めており、近年盛んに研究開発が進められている。イジングマシンが注目されるきっかけの一つとなった D-Wave Systems による世界初の商用量子アニーリングマシンを始めとして、最近では量子アニーリングに限らず様々な原理に基づいて動作するイジングマシンが日本国内を中心に発表されている。

一方で、現実的な課題における組合せ最適化問題とイジングマシンのハードウェアには大きな乖離があり、その間を埋めるにはソフトウェア環境の整備が必須である。

本研究では、イジングマシンの適用可能範囲の拡充と、利便性能向上を目指し、ハードウェアとソフトウェアの間層としてミドルウェア群や共通 API 等によって構成された共通ソフトウェア基盤の開発を行う。これにより、ユーザは各々のイジングマシンの詳細仕様を意識することなく使用することが可能となり、イジングマシンを用いた応用探索の加速が期待される。

2. イジングマシンの実行手順

イジングマシンは共通して、イジング模型または Quadratic unconstrained binary optimization (QUBO) と呼ばれる模型を入力形式として持つ。イジング模型では、無向グラフ $G = (V, E)$ においてエネルギー関数が

$$E(\{s_1, \dots, s_n\}) = \sum_{i \in V} h_i s_i + \sum_{(i,j) \in E} J_{ij} s_i s_j \quad (s_i \in \{+1, -1\})$$
として与えられる。ここで V, E はそれぞれ頂点集合と辺集合を表し、 n は頂点数である。頂点上に配置される s_i はイジング変数と呼ばれる組合せ変数であり、 $+1$ または -1 を取る。頂点及び辺に対応する重み h_i, J_{ij} はそれぞれ局所磁場、相互作用と呼ばれ実数をとる。局所磁場と相互作用の 2 種類のパラメータがイジングマシンの入力データに相当する。

イジング模型は統計物理学において確立している模型であり、物理学ではエネルギーが低いほど状態が安定しているという原理がある。これに基づき、あるいはアナロジーとして、イジング模型のエネルギー関数を組合せ最適化問題の目的関数とみなすことで、イジングマシンはイジング模型を入力に持つ組合せ最適化問題に特化したマシンと言える。一方、QUBO は組合せ変数がバイナリとなり、 $0, 1$ を取るビットで表される以外は、目的関数は同一の式で表される。つまりイジング模型からは変数変換の関係にあるため、イジング模型と本質的には等価である。

イジングマシンは、それぞれの動作原理に基づいたヒューリスティクスによって、イジング模型または QUBO のエネルギー最小化問題の近似解を出力する。別の言い方をす

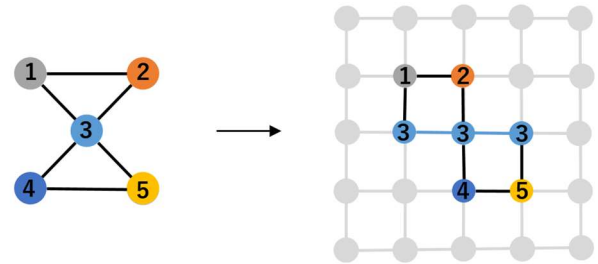


図 1. 5 変数の結合(左)を正方格子(右)に埋め込む例。左の中央にある変数③は、右では 3 変数の集合が担う。この集合の間には同一の値を取る制約が課せられる。

ると、ユーザが任意の組合せ最適化問題に対してイジングマシンを適用するためには、元の組合せ最適化問題に対応するイジング模型または QUBO で表す必要がある。これがイジングマシンを用いる最初のステップであり、様々な組合せ最適化問題のイジング定式化の方法が知られている。中でも充足可能性問題 (SAT) やそれに類する重み付き充足可能性問題 (MaxSAT) のイジング定式化は、NP 完全問題、NP 困難問題がイジングマシンで実行可能という意味で最も基礎的であり、イジングマシンの応用可能性を示唆している。

多くのイジング定式化においては、目的関数の他に制約条件が課せられることが多い。これらの制約条件は、問題の性質から由来する他にも、イジング変数が本質的に 2 値であることに由来する制約もある。例えば、変数の取り得る値が N 値の場合、 N ビットのバイナリベクトルを用いて多値を表現する方法がある。この時、ベクトルを表す変数集合は、one-hot エンコーディング、すなわち 1 つだけ 1 になるビット列であるという制約条件が課せられる。このような制約条件もエネルギー関数で表現する必要があり、目的関数に加算する形でイジング定式化に現れる。

組合せ最適化問題をイジング模型で定式化出来れば全てのマシンで直ぐに実行可能というわけではなく、主にハードウェアの仕様起源する制限への対応が必要になる。これは、イジング模型あるは QUBO で表された論理模型からハードウェアで直接実行可能な物理模型に変換が必要であることを意味している。具体的には、ハードウェアで実行可能なイジング変数の数 (または QUBO におけるビット数) の制限と、全ての変数の間に辺を張ることが出来ない物理グラフの制限が挙げられる。後者については、グラフ埋め込みと呼ばれる処理により、1 つの論理変数に対し複数の物理変数集合を割り当てる方法で解消される (図 1)。しかしながら、この対応関係により実質的にハードウェアで使

用可能なイジング変数の数 (ビット数) が減少することや、物理変数の集合を 1 つの変数とみなすために、集合内で同一の値を持つ制約が必要となることに注意を要する。一方、前者については問題を分割する方法が検討されているが、全ての問題に適用可能な確立された手段には至っていない。

以上をまとめると、組合せ最適化問題に対してイジングマシンを適用する手順は次の通りである。

1. 実行したい組合せ最適化問題をイジングモデルまたは QUBO に変換する。どちらかの形式でコスト関数と制約条件を定式化し論理モデルとする
2. 定式化された論理モデルから物理モデルへの変換を行う。必要に応じてグラフ埋め込み処理やこれに伴う制約条件を定式化し物理モデルに加算する
3. ハードウェアの入力仕様に従い、物理モデルに基づいたデータ入力と実行を指示する
4. 上記の逆変換処理を行う。各ステップにおいて全ての制約条件を検査し、条件を満たさない解をフィルタする

3. ミドルウェアの設計

上記手順のうち、ステップ 2, 3, 4 について各々の解決手段の方針を指定さえすればソフトウェアで自動処理が可能である。また、ステップ 1 についてもユーザ補助が可能な余地がある。

本研究開発のミドルウェアはアプリケーション層からハードウェア層に対し次の機能を提供する。

- (1) **組合せ最適化問題のイジング定式化**
典型的な組合せ最適化問題に対して、イジングモデルや QUBO に変換する機能を提供する。最も基礎的な機能として「イジング・パイナリ高次多項式」「論理式 (SAT)」等の入力や、性能指標に使われることの多い「最大カット問題」「巡回セールスマン問題」を始めとした組合せ最適化問題に対応する。
- (2) **定式化支援・数式処理ライブラリ**
ユーザがイジング定式化を自身のプログラムコード上で表現するための補助機能を提供する。具体的には、イジング変数やブール変数で表される多項式の代数法則に特化した数式処理ライブラリや、エネルギー関数に表れる $\sum_{i \in V}, \sum_{(i,j) \in E}$ 等の数学記号に相当する関数を提供する
- (3) **入力-論理、論理-物理変換・逆変換**
入力された最適化問題の変数が論理イジング変数に対してどのように変換されるか、また論理イジング変数が物理イジング変数に対してどのような対応関係になるかを管理する。またマシンの出力から逆変換を行うことで、ユーザにとって意味のある解を提示する。
- (4) **制約条件の抽象化と管理**
イジング変数の間に課せられる制約条件の抽象化と検査を行う。これらは定式化と物理イジングモデルへの変換に現れ、エネルギー関数を用いて制約条件を表現し、目的関数に加算する必要があるが、その処理をミドルウェアが代理する。またマシンの出力の逆変換ステップにおいて制約条件を満たしているか検査を行う。

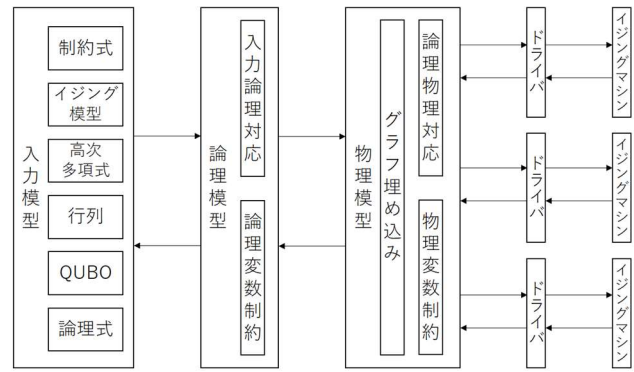


図 2. イジングマシンの共通ソフトウェア基盤ミドルウェアによる入力モデル・論理モデル・物理モデルの変換フロー

(5) 物理イジングモデルへの変換

各イジングマシンのハードウェアグラフに対応した論理イジングモデルから物理イジングモデルへの変換機能 (グラフ埋め込み処理) を提供する

(6) イジングマシンドライバ

物理イジングモデルを各ハードウェアのデータ形式仕様に合わせて入力し実行する。マシンの出力についても、マシンドライバがミドルウェアによって統一された形式で返却する

以上の機能を活用することにより、ユーザはイジングマシンの詳細に立ち入る必要なく、イジングマシンの利活用が可能となる。図 2 は、共通ソフトウェア基盤の提供するミドルウェアによる入力モデルから、イジングマシンへのデータ変換フローを表す。ミドルウェアの入力モデルとして、基礎的なモデルを中心に様々な数学形式を取り扱うことが出来る。ミドルウェアは入力層において入力モデルを受けとる。まず論理層においては、入力を論理モデルに変換し、変数間の制約と変数変換対応を維持したまま下位の物理層に渡す。物理層においても同様に、入力された論理モデルを物理モデルに変換し変数変換と制約を維持する。最後のドライバ層では各イジングマシンの実行命令を呼び出し、ハードウェア出力をラップして上位の層に返却する。ユーザは最終的に入力層において、入力変数に対する解の対応関係を得る。

4. まとめ

本研究はイジングマシンにおける共通ソフトウェア基盤の開発のうち、ハードウェア層とアプリケーション層の間となるミドルウェア層の設計と実装について検討した。ミドルウェアの中核を担う各機能を実現するアルゴリズムの研究は現在精力的に続けられており、それらの成果を適用することで、将来的により高度かつ高精度なミドルウェア機能の提供が期待出来る。また、新たなマシンの対応はドライバ層を追加することで、他のマシンからのシームレスな切り替えが可能となる。

今後は入力可能なモデルや数式の拡充を行うことで、アプリケーション層における共通 API 等の開発を進める予定である。これにより、現実的な課題における組合せ最適化問題に対するイジングマシン適用の加速が期待される。

本研究成果の一部は国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務の結果得られたものです。