

イジング型計算機による組合せ最適化のためのハイブリッド計算基盤

LASOLV[®]はNTTが研究開発に取り組む、新しい原理に基づいた計算装置です。LASOLV[®]は組合せ最適化問題を極めて高速に解くことが可能であるため、これまでは解くことができなかった課題の解決が期待されています。一方で実世界の問題をLASOLV[®]で解くためには、LASOLV[®]が受け付ける形式への問題変換や、従来からあるデジタル計算機との連携などの課題があります。本稿では、このような課題の解決手段を幅広いユーザに提供し、LASOLV[®]の効果的な利用を可能にするLASOLV[®]計算システムを紹介します。

あら い じゅんや や ぎ さ と し うちやま ひろゆき
 新井 淳也 /八木 哲志 /内山 寛之
 とみ た きんや みはやら かずひろ ともえ とくま
 富田 均也 /宮原 和大 /巴 徳瑪
 ほりかわ けいたろう
 堀川 桂太郎

NTTソフトウェアイノベーションセンター

イジング型計算機の利用における課題

ムーアの法則の終焉に伴い、計算機の性能を飛躍的に向上させるのは困難になりつつあります。一方で、私たちは今日のデジタル計算機を圧倒するような複雑な問題に依然として直面しています。組合せ最適化問題はそのような問題の1つです。基礎的な組合せ最適化問題としては、指定された複数の都市をもっとも効率的にめぐる訪問順を求める問題(巡回セールスマン問題)や、限られた色数で隣り合う領域が異なる色になるように塗り分ける問題(グラフ彩色問題)が有名です。

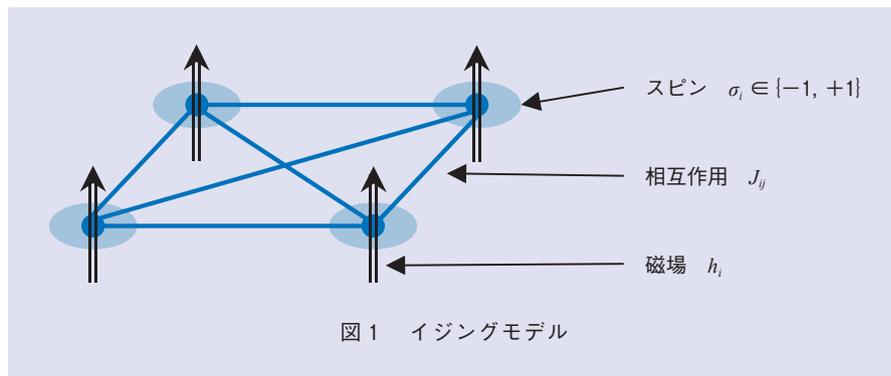
これらの応用の一例としては無線通信の品質改善があります。複数の無線アクセスポイントをイベント会場に配置する際、近傍のアクセスポイントが互いに同じ周波数帯を使用していると、電波の干渉により通信が不安定になります。ここでグラフ彩色問題を応用することで、近傍のアクセスポイントがそれぞれ異なる周波数帯となるような割り当て方を求めることができます。このように、組合せ最適化問題は

実世界の課題に広く内包されています。

組合せ最適化のさまざまな問題は、イジングモデルの基底状態探索(イジング最適化)問題に変換可能であることが知られています。イジングモデルとはスピンのネットワークを表現した統計力学のモデルです(図1)。スピン σ_i はアップ(+1)またはダウン(-1)のいずれかの状態をとります。 J_{ij} と h_i がそれぞれスピン間相互作用と磁場の強さを表すとすると、基底状態とは次式で定義されるイジングハミルトニアン H が最小化された状態を指します。基底状態を満たすスピンの組合せが、基底状態探索問題の解となります。

$$H = - \sum_{i < j} J_{ij} \sigma_i \sigma_j - \sum_i h_i \sigma_i$$

イジング最適化問題、ひいては組合せ最適化問題を効率的に解くため、D-Wave Systemsの量子アニーリングマシンや富士通のデジタルアニーラのようにイジング最適化に特化した計算機が開発されています。本稿ではそのような計算機をイジング型計算機と呼称します。NTTもまたNTT物性科学基礎研究所において光を用いたイジング型計算機であるLASOLV[®]の研究開発を行っています(図2)。これらの計算機の機能は限られた規模のイジング最適化問題を解くことのみですが、イジング型計算機とデジタル計算機を協調的に利用するハイブリッドアルゴリズムの使用により複雑な問題も解くことができます。例えば、大規模なイ



ジング最適化問題を解くため、デジタル計算機上で小規模な部分問題に分解したうえで、個々の部分問題をイジング型計算機で解くという処理を繰り返す発見的手法が存在します。そのため、イジング型計算機は実世界の問題に幅広く適用可能です。

しかし、イジング型計算機の利用にあたってはいくつかの課題があります。例えば、コストや設置場所の問題で多数のイジング型計算機を用意することは困難であるため、複数人で少数のイジング型計算機を共用する仕組みが必要です。そこでいくつかの企業は、クラウドプラットフォーム上のリモートな計算資源としてイジング型計算機を提供しています^{(1),(2)}。プラットフォームの利用者は手元のデジタル計算機でプログラムを実行し、そのプロ

グラムからWeb API (Application Programming Interface) リクエストを発行することでイジング最適化をそのプラットフォームに任せる（オフロードする）ことができます。ところが、イジング型計算機は1ミリ秒オーダーの時間で問題を解くため、相対的にインターネット経由の通信はあまりにも低速です。特に、ハイブリッドアルゴリズムはイジング型計算機とデジタル計算機の間で頻繁に情報をやり取りするため、通信コストによる性能低下が顕著になります。このように、イジング型計算機の利用には依然として課題があります。

これらの課題を解決すべく、NTTソフトウェアイノベーションセンターでは、LASOLV[®]を簡単かつ効率的に利用するためのプラットフォームの研究

開発に取り組んでいます。本稿ではまず、イジング計算プラットフォームに求められる設計上のポイントを整理し、さらにそれらの要求に対する回答として私たちのプラットフォームであるLCS (LASOLV[®] Computing System : LASOLV[®] 計算システム) を紹介します。LCSの大きな特徴は、利用者によって定義されたハイブリッドアルゴリズムをLASOLV[®]と併置されたデジタル計算機上で実行することです。これによりLASOLV[®]とデジタル計算機の間で効率的な通信が可能となります。さらに、LCSは幅広いユースケースにおいてイジング最適化問題への変換を容易にするライブラリを備えています。以降ではこれらについて詳しく説明します。

なお、LASOLV[®]は以前「量子ニューラルネットワーク」と呼ばれていました。紙面の都合でLASOLV[®]自体については説明しませんので、詳しくは過去の特集記事⁽³⁾を参照ください。

プラットフォーム設計における要求

次にイジング計算プラットフォームに求められる設計上の観点を3つ取り上げて説明します。

■効率性

前述したように、インターネット経由の通信はイジング最適化と比較して桁違いに長い時間を要します。さらに、技術の進歩に伴いイジング型計算機が対応するスピンの数は増加します。例えばLASOLV[®]は現状で最大2000スピンに対応していますが、NTT物性科



図2 LASOLV[®]外観

学基礎研究所ではこれを10万スピ
ンまで増加させることをめざしてい
ます。その結果、イジング最適化問
題の規模、すなわち J_{ij} と h_i のサイ
ズはますます増大し、通信効率
が性能により大きな影響を及ぼ
すようになります。したがって、
プラットフォームは通信のオーバ
ーヘッドが最小化されるように
設計されなければなりません。

■ 拡張性

イジング型計算機は進化の途上
にあります。そのため、新しい世代
のハードウェアは従来と異なる
仕様を持つ可能性があり、必然
的にプラットフォームはヘテロ
ジニアス（異種混在）な計算資
源を含むこととなります。計算
資源の使い分けは利用の複雑さ
を増加させるため、これらの資
源を効果的かつ利用しやすい形
態で提供する工夫が必要です。

■ 生産性

各種の組合せ最適化問題をイ
ジング

最適化問題へ変換する作業は特
殊な技能を要します。一方で、
変換の方法はあまりにも多様
であるため、それぞれの組合せ
最適化問題に対応する変換アル
ゴリズムの提供は非現実的です。
幅広い利用目的において高い生
産性を実現する、利便性と汎用
性を兼ね備えたプログラミング
インタフェースが求められます。

LASOLV[®] 計算システム

NTTソフトウェアイノベーション
センターでは、LASOLV[®]のため
の計算プラットフォームとして
LCSの開発に取り組んでいます。
具体的には、LCSは独自のPy
thonライブラリ、およびジョ
ブスケジューラなどのミドル
ウェアを備えた計算機クラスタ
です。前述した設計上の要求を
満たすため、LCSは次の3つの
特徴を持ちます。

■ LASOLV[®]とデジタル計算機の統合

LASOLV[®]とデジタル計算機の
間の

効率的な通信を可能にするため
、LASOLV[®]とデジタル計算機は
LCSにおいて1つのクラスタに
統合されています（図3）。LAS
OLV[®]が含まれることを除けば
、HPC（High-Performance
Computing：高性能計算）分
野において標準的なクラスタの
構成です。利用者はSSH接続
によるコマンド操作、または
ブラウザ上のJupyter Notebook
からプログラムの実行をジョ
ブとして投入することができます
（図4）。ジョブスケジューラ
とライブラリの協調により、プ
ログラムのうちデジタル計算機
を使用する部分は計算ノードで
実行され、イジング最適化は
LASOLV[®]にオフロードされま
す（図5）。計算ノードとLAS
OLV[®]の間の通信はクラスタ内
に閉じるため、インターネットに
比べて広帯域・低遅延の通信が
可能です。また、CPUコアおよ
びLASOLV[®]を複数のジョブが
同時に使用することがないよう
に実行されるため、複数ユー
ザの同時使用が可能です。

■ グループディスパッチ

LCSはスケールアウト可能な
設計となっています。今後異
なる仕様を持つLASOLV[®]が
登場し得ることから、それら
を管理しやすくするためLCS
は互換性のあるLASOLV[®]を
グループとして管理します。利
用者はイジング最適化のオフ
ロード先をグループ単位で指
定することが可能です（図5）。
ジョブスケジューラはそのグ
ループ内

* 現時点ではNotebook全体が一括で
ジョブ実行されます。今後の改良
でインタラクティブな実行もサポ
ートする予定です。

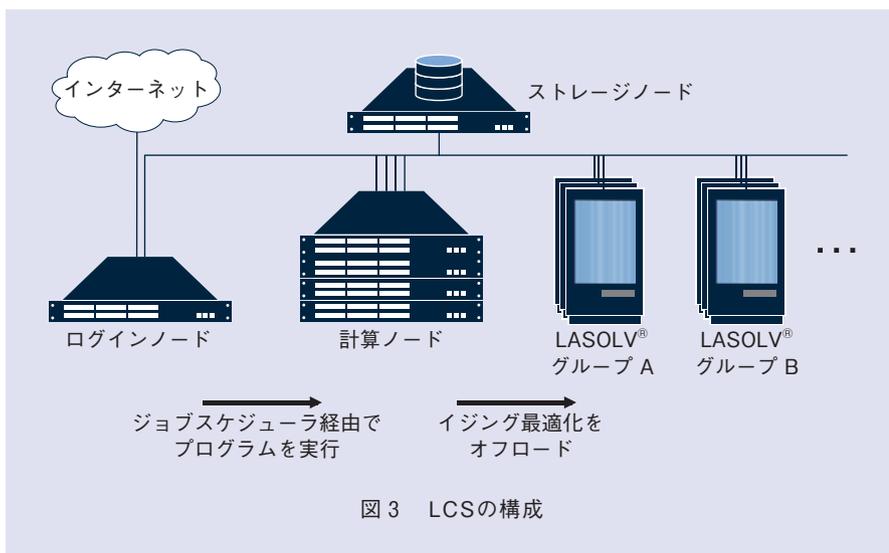


図3 LCSの構成

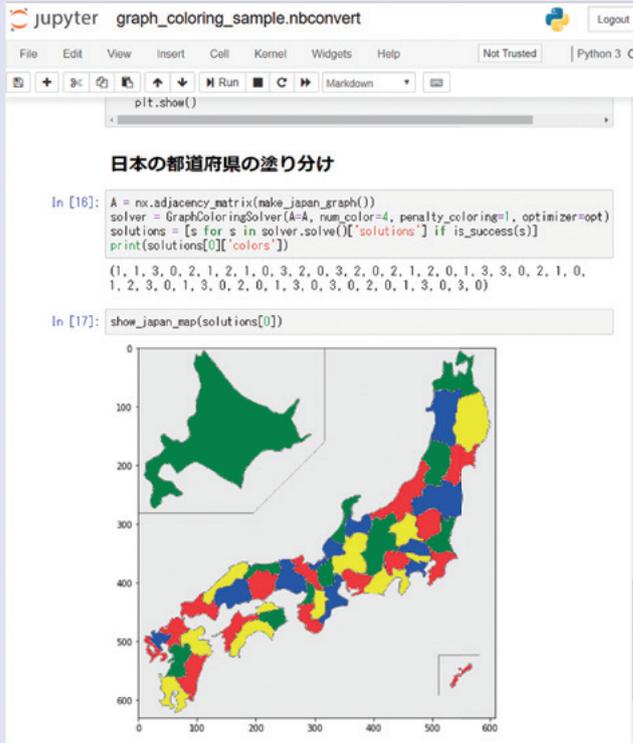


図4 Jupyter Notebookの利用イメージ

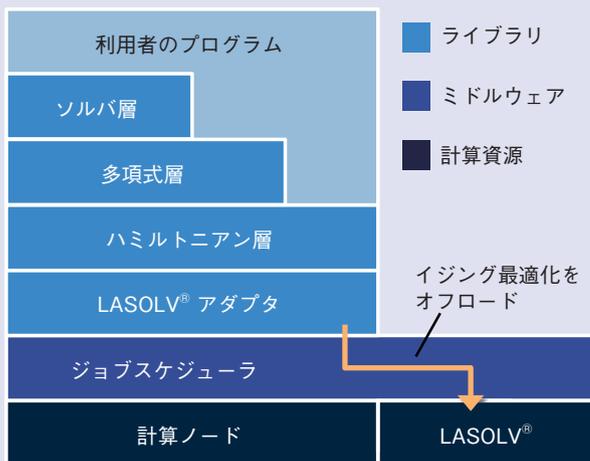


図5 LCSスタック

で処理を振り分けることにより負荷を分散し、効果的にLASOLV®を使用します。なお、与えられたイジング最適化問題の性質（スピン数など）に基づいて必要な仕様を満たすグループを自動的に決定する機能も将来的に実装される予定です。

■階層的な問題変換

利用者は次の3層からなるプログラミングインタフェースを用途に応じて使い分けることができます。

- ① ソルバ層：巡回セールスマン問題やグラフ彩色問題のような広く知られたNP困難問題を多項式、またはイジング最適化問題として表現された目的関数へ変換するアルゴリズム群
- ② 多項式層：多項式として表現された目的関数をイジング最適化問題へ変換するための内部DSL (domain-specific language：ドメイン特化言語)
- ③ ハミルトニアン層：イジング最適化問題 (J_{ij} と h_i) を直接入力する生のインタフェース

この3段階は、手でイジング最適化問題へ変換する場合において一般的なプロセスに対応しています。すなわち、形式的な問題定義、多項式関数の最小化問題への変換、イジングハミルトニアンへの変形です。多項式層は解こうとする問題にソルバ層の実装が適さなかった場合や、手動でチューニングを行うための柔軟性が必要となる場合に有用です。例として、グラフ G に含まれる n 個の頂点を、隣接頂点がそれぞれ

```
f = sum(A (u, v) * sum(x (u, i) * x(v, i) for i in range(c))
        for u, v in combinations(range(n), 2))
    + p * sum((sum(x(v, i) for i in range(c)) - 1) ** 2
              for v in range(n))
solution = PolynomialMetasolver(f).solve()
```

図6 多項式層のサンプルコード

- (3) 武居・稲垣・稲葉・本庄：“複雑な組合せ最適化問題を解く量子ニューラルネットワーク,” NTT技術ジャーナル, Vol.29, No.5, pp.11-14, 2017.
- (4) A. Lucas：“Ising formulations of many NP problems,” Frontiers in Physics, Vol.2, No.5, pp.1-15, 2014.
- (5) <https://github.com/Blueqat/Wildqat/>
- (6) <https://ocean.dwavesys.com/>
- (7) <https://github.com/OpenJij/>
- (8) <https://github.com/recruit-communications/pyqubo/>

れ異なる色になるように c 色で塗り分けるグラフ彩色問題を考えます。 A を G の隣接行列, p をハイパーパラメータとし, 変数 $x_{v,i} \in \{0, 1\}$ を用いると, グラフ彩色問題は次式の最小化問題に変換されます⁽⁴⁾。

$$f(x) = \sum_{u < v} A_{u,v} \sum_{i=1}^c x_{u,i} x_{v,i} + p \sum_{v=1}^n \left(\sum_{i=1}^c x_{v,i} - 1 \right)^2$$

この式を展開すると変数 $x_{v,i}$ に関する多項式が導出されますが, 多項式層のDSLを用いるとPythonプログラマにとって直感的な記述でこの問題を解くことができます (図6)。

また, ハミルトニアン層はLASOLV[®]が対応するスピン数を超えて大規模なイジング最適化問題を解くためのハイブリッドアルゴリズムを搭載しています。これはいくつかの既存手法を基に新しく開発された発見的手法です。このような研究開発の成果を素早く利用者に届けるために私たちは独自のライブラリを開発しており, ユーザはこのライブラリで動作するプログラムを実装するだけで成果を利用可能です。

なお, 他のイジング計算ライブラリ^{(5)~(8)}でイジング最適化問題へ変換

し, LCSライブラリのハミルトニアン層に直接入力することで問題を解くという使い方も容易に実現できます。したがって利用者の自由を制限することはありません。また逆にLCSライブラリの機能の多くは汎用性があるので, イジング最適化問題への変換結果をLASOLV[®]以外のイジング型計算機の入力として使用することも可能です。

今後の展開

本稿ではまずプラットフォームの設計における要求事項を整理し, さらにLCSがそれらをどのように達成したかを紹介しました。

現在, LCSは各種機能を兼務するハイエンドサーバ2台とLASOLV[®]1台を中心とするスモールスタートの構成で共同研究者に提供されています。LASOLV[®]を用いた研究開発を推進するため, 私たちは利用状況とフィードバックに基づくLCSの評価と改良を継続的に実施していきます。

■参考文献

- (1) <https://cloud.dwavesys.com/leap/>
- (2) <https://www.fujitsu.com/global/digitalannealer/services/>



(後列左から) 富田 均也/ 巴 徳瑪/
新井 淳也/ 内山 寛之
(前列左から) 八木 哲志/ 堀川 桂太郎/
宮原 和太

CPUとGPUの併用と同じように, LASOLV[®]のような新原理に基づく計算機と従来型のデジタル計算機という視点でも適材適所の併用が重要になりつつあります。そのような研究の一助となるよう, 引き続きLCSの改良に取り組みます。

◆問い合わせ先

NTTソフトウェアイノベーションセンター
分散処理基盤技術プロジェクト
TEL 0422-59-6011
FAX 0422-59-3739
E-mail sic-lasolv-p-ml@hco.ntt.co.jp