URL　https://journal.ntt.co.jp/article/23087
DOI　https://doi.org/10.60249/23095002

イジングマシン活用アプリケーションの開発を支援するコンピューティングシステム

CPUのような従来のコンピュータが解くことが難しい問題の１つに，膨大なパターンの組み合わせの中から条件を満たすものを見つけ出す「組み合わせ最適化問題」があります．この問題に特化して，新しい原理に基づいて演算するコンピュータが「イジングマシン」です．しかし，その使いこなしには，実社会の問題をイジングマシンが扱える形式に変換するなど多くの課題があります．本稿では，この課題を解決する，SDK（Software Development Kit）やコンピューティングシステムの研究開発について紹介します．

キーワード：#イジングマシン，#LASOLV®，#組み合わせ最適化問題

宮原　和大（みやはら　かずひろ）
寺本　純司（てらもと　じゅんじ）
NTTコンピュータ＆データサイエンス研究所

組み合わせ最適化問題とイジングマシン

膨大な選択肢からもっとも良いものを探す組み合わせ最適化問題は，現在のコンピュータでも多くの計算時間を必要とする難問です．例えば，指定された複数の訪問先をもっとも効率的に巡る順番を求める「巡回セールス問題」や，限られた色数という制約の下，与えられた地図の隣り合う領域が違う色で塗分けられるようにする「グラフ彩色問題」などは，規模が拡大するほど難しくなる典型的な組み合わせ最適化問題です．

この課題に対し，磁性体の性質を表すモデルを用いて解くという新しいアプローチで解を導き出す「イジングマシン」の研究が進んでいます．イジングマシンは，組み合わせ最適化問題の１つ，イジングモデル基底状態探索問題を解くことに特化したコンピュータです．イジングモデルとは，磁石（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二次多項式のイジングモデル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み合わせが，基底状態探索問題の解となります．

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

組み合わせ最適化問題はさまざまな種類がありますが，このイジングモデル基底状態探索問題に変換可能であることが理論的に証明されています．そのため，D-Wave Systems[1]や富士通[2]などのさまざまな会社がイジングマシンを開発し，それを用いていろいろな組み合わせ最適化問題を解く取り組みを進めています．NTT研究所においても，レーザ光を利用したコヒーレントイジングマシン[3]「LASOLV®」を開発しています．

イジングマシンの利用を容易にするSDK

イジングマシンの活用にあたっては，前述のとおり解こうとする問題をイジングマシンに入力可能なイジングモデルと呼ばれる形式で表現する必要があります．そして，問題をイジングモデルで表現する際には，スピンの取る±１の値と問題の解の意味との間に対応付けを行い，イジングモデルの値が最小となるときが得たい解に対応するように多項式をつくる必要があります．最小となるときと得たい解を対応付けるのは，イジングマシンが入力されたイジングモデルを最小化するようなスピンの値を求める計算機であるためです．このイジングモデルの考案作業は問題の複雑さに比例して煩雑になり，イジングマシン利用の難易度を上げてしまいます．NTTコンピュータ＆データサイエンス研究所では，このイジングモデルの考案作業を大きく簡略化する機能を持つSDK（Software Development Kit）「cimsdk」の開発をしています．ま

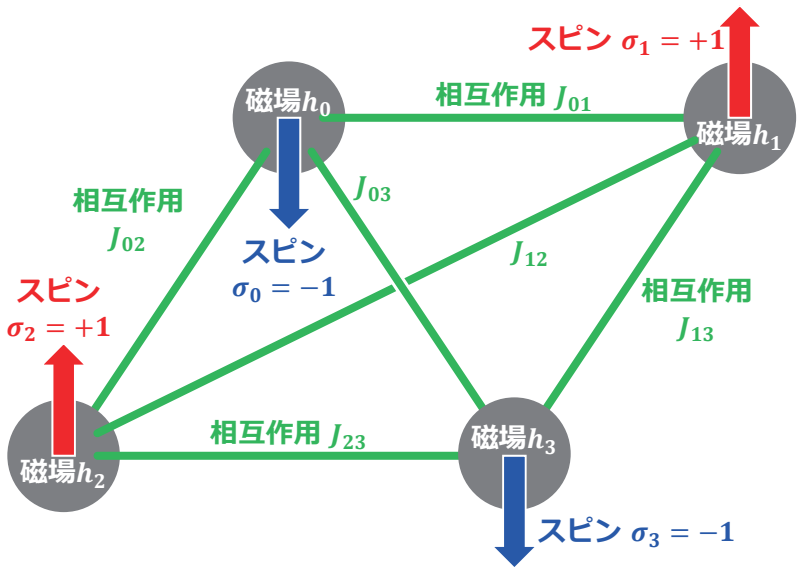


図１　イジングモデル

ずは通常のイジングモデルの考案作業を簡単な問題を例に説明します．

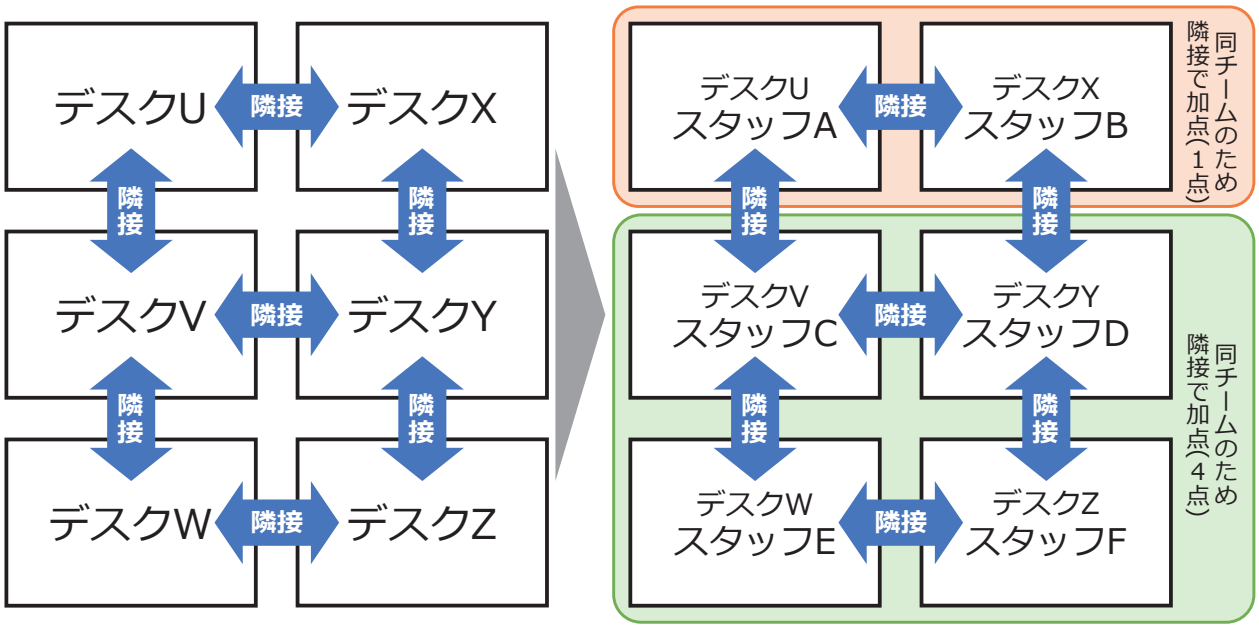
あなたは自部署のスタッフ６人（仮にA，B，C，D，E，Fとします．）に６つのデスク（仮にU，V，W，X，Y，Zとします．）を割り当てることを考えています．６つのデスクは**図２**(a)のような位置関係です．スタッフ１人につきデスクは１つ割り当てる必要があります（前提１）．また，AさんとBさん，CさんとDさんとEさんとFさんは同一業務でチームを組んでいるので，隣接するデスクであることが望ましいです．ここでは同チームで隣接デスクとなったスタッフの組み合わせ１つごとに１点加算され，この点数を最大化することをめざします（前提２）．

この前提１，前提２を踏まえると，図２(b)のような割当てが最大点数となる解の１つです．この例題について，イジングモデルによる表現を考えていきます．

まず，±１のスピンと問題の解の意味の間の対応付けを考えます．今回はスタッフ６人とデスク６つの組み合わせ36通りについて１つずつスピンを用意し，「＋１であればそのスタッフがそのデスクに割り当てられた，－１であれば割り当てられなかった．」という意味として対応付けてみます．これらのスピンを今回は$x_{A,U}$のようにスタッフ名とデスク名を添えたスピンで表現することとしましょう．例えば$x_{B,X}$であれば「＋１であればスタッフBがデスクXに割り当てられた，－１であれば割り当てられなかった．」という意味で解釈することになります．このように用意した36個のスピンについて前提１と前提２の表現を考えます．

前提１を先ほどのスピンに関する表現としてみましょう．前提１は「スタッフ１人につきデスクは１つ」という内容でした．すなわち「各スタッフについて６つのデスクのうち１つだけが割り当てられる」かつ「各デスクについて６人のスタッフのうち１人だけが割り当てられる」ということです．このうち「スタッフAについて６つのデスクのうち１つだけが割り当てられる」を考えてみると，$(\sum_{desk\in\{U,V,W,X,Y,Z\}} x_{A,desk}+4)^2$のような二次多項式で表現することが可能です．この式は$x_{A,U}$～$x_{A,Z}$のうち１つだけが＋１，残りの５つが－１となったときに最小となる式であり，まさに１つのデスクだけが割り当てられることが式の値が最小となることに対応します．他のスタッフ，デスクについても同様の二次多項式で割当てを１つとすることで表現が可能です．

前提２についても考えていきましょう．前提２は「同チームのスタッフが隣接するごとに１点加点」というものでした．例としてスタッフAとBのチームを考えましょう．スタッフAとBはデスクUとXに割り当てられれば１点加点です．これを二次多項式で表現すると$(x_{A,U}+1)(x_{B,X}+1)/4$となります．実際に$x_{A,U}$と$x_{B,X}$に±１を代入してみると，両方とも＋１のときのみこの二次多項式は＋１となり，他のときは０となります．このままでは逆のデスク割り当てや他のスタッフ，隣接デスクの組み合わせで加点されなくなってしまいますので，



(a)　デスク配置イメージ　　(b)　前提を踏まえたデスク配置最適解

図２　例題のデスク配置イメージと最適解の例

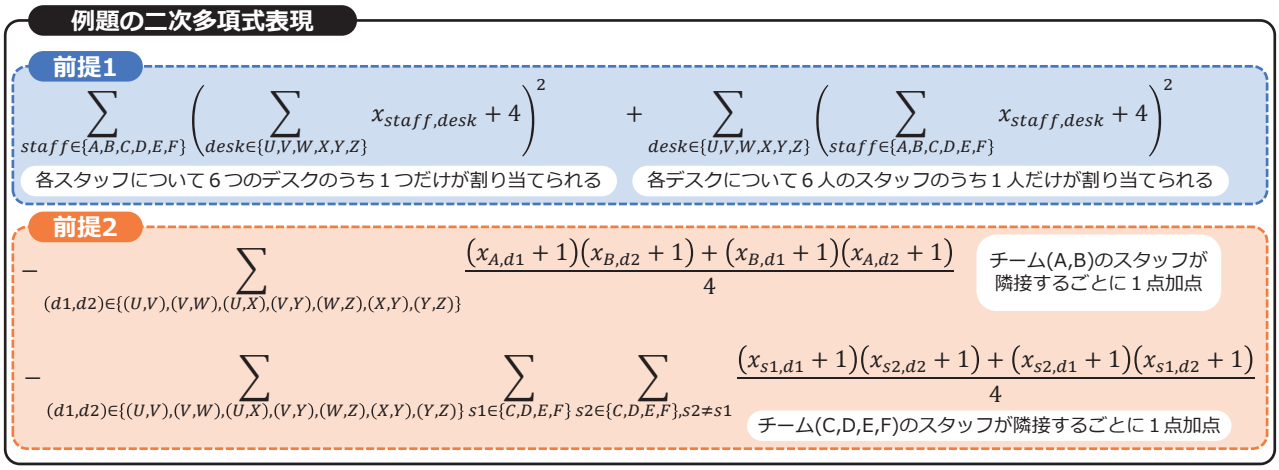


図３ 今回の例題を表現するイジングモデル

すべての組み合わせについてこの式を足し合わせることで，加点される点数を二次多項式で表現することができます．ここで注意するべきは，イジングマシンがイジングモデルを最小とする解を導出するという点です．つまり，隣接による加点が大きいほど，イジングモデルの取る値が小さくなるようにする必要があるため，この二次多項式は正負を反転させて足し合わせることとなります．

ここまでの内容を踏まえ，考案した多項式を足し合わせると，今回の例題のイジングモデルの全体像は**図３**のようになります．このようにシンプルな例題であっても，イジングモデルの数式は比較的複雑なものとなります．

対して，cimsdkによる機能を用いた場合のイジングモデルの構築イメージは**図４**のようになります．図３の数式を考案する際に必要となった，各多項式がどのようなときにどんな値をとるのか，という点を全く考慮せずにイジングモデルを生成できることが分かります．

実用上の問題を考える場合には，より複雑な前提条件をより多くのスピンに対して考えることとなり，イジングモデルの考案作業はとても煩雑なものとなります．実用問題であれば整数を扱うこともあり，複数の±１を取るスピンを組み合わせて１つの整数として扱う必要も生じます．また，同時に３つ以上のスピンに関するような条件を表現しようとすると三次以上の多項式を扱わなければなりませんが，イジングモデルでは二次多項式しか扱えないので，うまく次数を削減するような取り組みも必要となります．

cimsdkでは，このような複雑になりやすいイジングモデル考案の作業について，整数をはじめとした各種スピンとそれらに関する条件を自動的にイジングモデル変換する機能を提供しています．本技術によりイジングモデルおよびイジングマシンに関する専門知識を持たない人でも，容易にイジングマシンによる高速な求解という恩恵を得られるようになります．特にcimsdkでは他社類似技術に比べて，複数の条件を無制限に再帰的に組み合わせることが可能であり，数学的に定義可能な論理構造をそのまま表現可能である点，任意の連想配列から対応するイジングモデルを自動生成できる点が独自の強みとなっています．今後は自動変換の対象となるスピンと条件を拡充していくことにより，対応可能な実用問題を広げていきます．

LASOLV® Computing Systemによるアプリケーション開発と実行

イジングマシンを使いこなすうえでのもう１つの課題は，「イジングモデル基底状態探索問題を解くことに特化したコンピュータ」という特性から，それ単体ではアプリケーションを完結させることが難しいという点です．前述したSDKを利用したデータの前処理・後処理や，古典コンピュータが有利な演算などを組み合わせるために，多種演算装置を組み合わせたヘテロジニアスなシステムが必要です．さらに，イジングマシンはPCや一般的なサーバなどと比べるとコストや設置場所の問題があり，多数のマシンを用意することは困難です．そのため，スーパーコンピュータと同様に，複数人で少数のイジングマシンを共用する手段が必要になります．そこでNTTコンピュータ＆データサイエンス研究所では，LASOLV®などのイジングマシンを効率良く利用するプラットフォームとしてLCS（LASOLV® Computing System）の開発に取り組んでいます[4]．前述したSDK等，アプリケーション開発を容易にする開発環境に加え，ユーザのジョブを制御し，CPU等で計算を行うノードやイジングマシンで計算を行うノードを効率良く連携利用しつつ，複数ユーザのジョブが重複しな

```
staffs = ["A", "B", "C", "D", "E", "F"]
teams = [{"A", "B"}, {"C", "D", "E", "F"}]
desks = ["U","V", "W", "X", "Y", "Z"]
neighbor_desks = [
    {"U", "V"}, {"V", "W"}, {"X", "Y"}, {"Y", "Z"},
    {"U", "X"}, {"V", "Y"}, {"W", "Z"},
]
x = {}
for s, d in itertools.product(staffs, desks):
    # ±１を取る変数を宣言
    x[s, d] = cimsdk.boolean((s, d))

# イジングモデルを足し合わせるための変数
input = 0

# 前提１はchoose_one関数を宣言すれば自動的にイジングモデルがつくられる
for s in staffs:
    input += cimsdk.choose_one(x[s, "U"], x[s, "V"], x[s, "W"], x[s, "X"], x[s, "Y"], x[s, "Z"])
for d in desks:
    input += cimsdk.choose_one(x["A", d], x["B", d], x["C", d], x["D", d], x["E", d], x["F", d])

for t in teams:
    for s1, s2 in itertools.combinations(t, 2):
        for d1, d2 in itertools.product(desks, 2):
            # 前提２は値を連想配列で宣言すれば自動的にイジングモデルがつくられる
            input -= cimsdk.translate_dict(
                (x[s1, d1], x[s2, d2]),
                {
                    (-1, -1): 0,
                    (-1, +1): 0,
                    (+1, -1): 0,
                    (+1, +1): +1 if {d1, d2} in neighbor_desks else 0,
                })
```

図４　cimsdkによるイジングモデルの構築イメージ

いようにスケジューリングする機能を有しています．

このLCS上で，cimsdkを用いた開発を行うことで，LASOLV®の実力を活かした応用開拓にも取り組んでいます．例を挙げますと，三菱重工業株式会社との共同実証において，人員リソース計画にイジングマシンを用いる取り組みを実施しました(5)．同社には，世界中の発電プラント等の点検・補修工事を行うという業務があります．これに対し，さまざまな点検項目に合わせたスキルを有する作業員を派遣する必要がありますが，その人数，連続勤務日数，残業時間など複雑なパラメータも考慮した人員計画を立案しなければなりません．従来は，点検対象プラント数26カ所，作業員141人，日数64日間の期間で，保有スキルなどのさまざまな制約条件を満たす人員計画の解候補を，熟練者が人手で１カ月程度かけて計画を策定していました．これは組み合わせ総数3.48×10^{13329}通りという途方もない規模になりますが，ここに５万スピンを一気に計算できる最新版のLASOLV®を用いることで，非常に短期間で得られることを実証しました．その高速性を用いて，複数パターンの人員計画策定や，感染症対策等で必要になる計画の随時見直しなどが可能になると期待されています．

今後の展開

本稿では，イジングマシンの特性を踏まえつつ，アプリケーション開発を容易にするSDKの取り組みや，コンピューティングシステム開発の取り組みについて紹介しました．今後，イジングマシンに加え，量子コンピュータの実用化が進む中，その特性を踏まえたSDK等ライブラリの重要性，よりヘテロジニアス性の増したコンピューティングシステムの必要性が増していきます．NTTコンピュータ＆データサイエンス研究所では，量子コンピュータなどの先端的な技術を生かすコンピューティングシステムアーキテクチャのあるべき姿を定義し，研究を発展させていきます．

■参考文献

(1) https://www.dwavesys.com/solutions-and-products/systems/
(2) https://www.fujitsu.com/jp/digitalannealer/
(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) 新井・八木・内山・富田・宮原・巴，堀川：“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，” NTT技術ジャーナル，Vol. 31，No. 11，pp. 27-31, 2019.
(5) https://group.ntt/jp/newsrelease/2022/04/25/220425a.html


（左から）宮原　和大／ 寺本　純司

イジングマシンや量子コンピュータのような新原理に基づく計算機は，今後アプリケーション開発や利用の容易さが一層重要視されることになるでしょう．将来に向け，私たちはソフトウェアとコンピューティングシステムアーキテクチャを進化する研究を推進します．

◆問い合わせ先

NTTコンピュータ&データサイエンス研究所
企画担当
TEL　046-859-4003
FAX　046-855-1149
E-mail　cd-koho-ml @ ntt.com