

実イジング計算機による 制約付きスロット配置問題の解法

金丸 翔^{1,a)} 川村一志² 田中 宗^{3,4} 戸川 望¹

概要：イジング計算機は組合せ最適化問題を物理システムにマッピングすることで、組合せ最適化問題の準最適解を高速に得ることができるとして注目されている。スロット配置問題は論理ブロックの最適配置や最適配送決定において重要な役割を果たす組合せ最適化問題である。本研究では、スロット配置問題に制約条件を付加した問題をイジング計算機によって効率よく解法する手法を提案する。イジング計算機でこの問題を解法するため、この問題をイジングモデルで表現する。まず、スロット配置問題をイジングモデルの状態空間で表現する際に満足すべき制約条件が満たされているとき、イジングモデルのエネルギーが最小となる制約項を導入する。この制約をスロット配置制約と呼ぶ。また、今回は付加する制約条件として、特定の部品が一定の距離以内に置かれなければならない、というものを考える。この制約を違反したときにエネルギーが大きくなる付加制約項を導入する。この制約を付加制約と呼ぶ。更に、配置された部品間の配線数とマンハッタン距離の加重和である目的関数項をイジングモデルで表現する。また、イジング計算機によって得られた解に対し、解釈処理なるアイデアを導入することで、スロット配置制約を満たさない解が得られたとしても、スロット配置制約を満たす解に解釈し直す手法を提案する。デジタルアニーラを利用し、単純な SA 手法とデジタルアニーラで比較した結果、制約付きスロット配置問題において、SA と同程度の精度の準最適解を得るのに必要なデジタルアニーラの時間が最大で約 1/50 に圧縮することができた。

キーワード：イジング計算機、組合せ最適化問題、スロット配置問題、二次割り当て問題、イジングモデル、デジタルアニーラ

1. はじめに

1.1 イジング計算機

近年、組合せ最適化問題を効率的に解く計算機として、さまざまなイジング計算機が研究されている [1–5]。組合せ最適化問題とは、与えられた制約を満たした上で評価関数を最大または最小とする決定変数の組合せを探索する問題である。与えられる決定変数の数が増えると解の候補が爆発的に増加し、従来のノイマン型計算機では最適解を現実的な時間で得ることが困難となる。イジング計算機は組合せ最適化問題を物理システムにマッピングすることで、組合せ最適化問題の準最適解を高速かつ高精度で得ることができるとして注目されている。

1.2 制約付きスロット配置問題

組合せ最適化問題の 1 つにスロット配置問題がある。スロット配置問題とは部品と格子状のスロットが与えられたとき、部品間の配線長の総和が最小となる配置を求める問題である。スロット配置問題は論理ブロックの最適配置や回路面積の最小化などに貢献し、LSI 自動設計や配送管理システムにおいて重要な役割を果たす [6]。現実的な問題では特定の部品は一定の距離以内に置かなければならないという制約が存在する。スロット配置問題にこの制約を追

加したものを制約付きスロット配置問題と呼ぶ。一般に制約付きスロット配置問題は NP-困難な問題である二次割り当て問題として定式化できる [7,8]。

1.3 本稿の提案

従来、二次割り当て問題は Simulated Annealing (SA) や分枝限定法を用いて解法されてきた。それに対し本研究では、近年その計算性能が期待されているイジング計算機を用いた解法を検討する。

さまざまな組合せ最適化問題がイジングモデルにマッピングされ、イジング計算機によって解が評価されてきた [9,10]。しかし、制約付きスロット配置問題に対しイジング計算機を適用した研究は著者らの知る限り存在しない。さらに、一般に組合せ最適化問題をイジングモデルにマッピングしてイジング計算機で解を得たとしても、その解は必ずしも元の組合せ最適化問題における実現可能解にはならない。イジング計算機による組合せ最適化問題の解法では、得られた解をいかに解釈するかが重要となる。

1.4 本稿の貢献

本稿の貢献は以下の 3 つである。

- (1) イジングモデルのハミルトニアンとして、まず、スロット配置問題をイジングモデルの状態空間で表現する際に満足すべき制約条件が満たされているとき、イジングモデルのエネルギーが最小となる制約項を導入する。また、今回は付加する制約条件として、特定の部品が一定の距離以内に置かれなければならない、というものを考える。この制約を違反したときにエネ

¹ 早稲田大学大学院基幹理工学研究科 情報理工・情報通信専攻

² 早稲田大学基幹理工学部情報通信学科

³ 早稲田大学グリーン・コンピューティング・システム研究機構

⁴ 科学技術振興機構さきがけ

^{a)} sho.kanamaru@togawa.cs.waseda.ac.jp

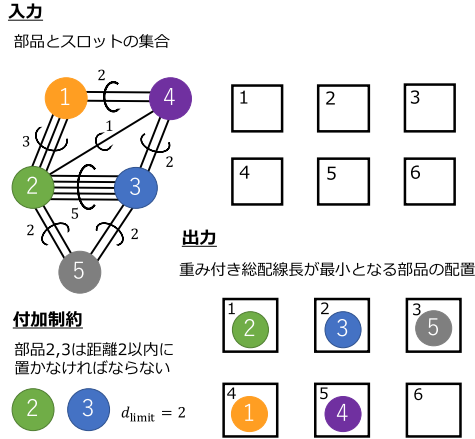


図 1: $o = 2, p = 3, m = 5$ の制約付きスロット配置問題.

ギーが大きくなる付加制約項を導入する. 更に, 配置された部品間の配線数とマンハッタン距離の加重和である目的関数項をイジングモデルで表現する. これらを導入することで, 制約付きスロット配置問題をイジングモデルとしてマッピングした.

- (2) イジング計算機によって得られた解がスロット配置問題を成立させる制約を満たさない場合について, 制約を満たす解に補正する方法を提案した. これにより, 問題の規模の大きさに関わらず, 必ずスロット配置問題を成立させる解を得ることが可能になる.
- (3) デジタルアニーラ (DA) を利用し解を評価した. 単純な SA 手法と DA で比較した結果, 制約付きスロット配置問題において, SA と同程度の精度の準最適解を得るのに必要な DA の時間が最大で約 1/50 に圧縮することができた.

1.5 本稿の構成

本稿の構成を以下に示す. 2 章で, 制約付きスロット配置問題を定式化する. 3 章で, 制約付きスロット配置問題をイジングモデルにマッピングする手法を提案する. 4 章で, アニーリング後の制約条件満足のための補正アルゴリズムを提案する. 5 章で, DA を用いてスロット配置問題の解を求める計算機実験の結果を示し, 考察を述べる. 6 章で, 結論を述べる.

2. 問題の定式化

m 個の部品の集合を $M = \{c_1, c_2, \dots, c_m\}$, o 行 p 列のスロットの集合を $S = \{s_1, s_2, \dots, s_t\}$ と定義する. ただし, $t = o \times p$ とする.

スロット s_a が a_1 行 a_2 列 ($1 \leq a_1 \leq o, 1 \leq a_2 \leq p$) に, スロット s_b が b_1 行 b_2 列 ($1 \leq b_1 \leq o, 1 \leq b_2 \leq p$) にあるとする. 2つのスロット $s_a, s_b \in S$ のマンハッタン距離は $l_{s_a, s_b} = |a_1 - b_1| + |a_2 - b_2|$ (1)

で定義される. また, それぞれの部品間には配線数が与えられている. 2つの部品 $c_i, c_j \in M$ ($1 \leq i \leq m, 1 \leq j \leq m$) の配線数を r_{c_i, c_j} とする. ここで r_{c_i, c_j} は非負整数であり, また $r_{c_i, c_j} = r_{c_j, c_i}$ である. ここで, 部品 c_i が置かれたスロットを明示的に表すため, $s_{a(i)}$ と書くことにする. また後の記法の都合から, 部品 c_j が置かれたスロットを $s_{b(j)}$ と書くことにする. 部品 c_i と部品 c_j の間の重み付き配線

長を, $r_{c_i, c_j} \times l_{s_{a(i)}, s_{b(j)}}$ と定義する. すると, 全ての部品の重み付き総配線長は,

$$L = \sum_{i=1}^m \sum_{j=1}^m r_{c_i, c_j} l_{s_{a(i)}, s_{b(j)}} \quad (2)$$

と定義する. ここで, 1つの部品はただ1つのスロットに必ず存在するものとする. これを部品重複禁止制約と呼ぶ. また, 各スロットには高々1つの部品を置くものとする. これをスロット重複禁止制約と呼ぶ. これらを合わせてスロット配置制約と呼ぶ.

さらに, 特定の部品は一定の距離以内に置かなければならない場合が存在する. これを付加制約と呼ぶ. ここで, 付加制約で用いるマンハッタン距離の制限値を d_{limit} と置く. **定義 1.** 制約付きスロット配置問題とは, m 個の部品の集合 M と o 行 p 列の格子状のスロットの集合 S が与えられたとき, スロット配置制約と付加制約を満足したうえで, 全ての部品の重み付き総配線長が最小となる配置を求める問題である.

$o = 2, p = 3, m = 5, d_{\text{limit}} = 2$ の制約付きスロット配置問題を図 1 に示す.

3. スロット配置問題のイジングモデルマッピング

3.1 イジングモデル

イジングモデルとは物理学の一分野である統計力学における基礎モデルの一つである. イジングモデルは, スピンと呼ばれるミクロな要素が複数集まり, スピン間の相互作用と各々のスピンの働く磁場によって, システム全体としてどのような振る舞いを示すかを調べるための理論モデルの一つである. イジングモデルは無向グラフ $M = (V, E)$ 上に定義される. V, E はそれぞれ, スピンが配置される頂点集合及び頂点間の辺集合である. 2つの頂点 $i \in V, j \in V$ が接続されているとき, $(i, j) \in E$ は頂点 i と j の間の接続辺を表す. 頂点 i に配置されたスピンを σ_i で表す. スピンは ± 1 の 2 値のいずれかを取る. $+1$ を上向きスピン, -1 を下向きスピンと呼ぶ. ここで, 2つのスピン σ_i, σ_j の間の相互作用係数を J_{ij} , スピン σ_i に作用する外部磁場を h_i と定義する. ここで, J_{ij} 及び h_i は実数である. イジングモデルのエネルギー関数 \mathcal{H} は

$$\mathcal{H} = - \sum_{(i,j) \in E} J_{ij} \sigma_i \sigma_j - \sum_{i \in V} h_i \sigma_i \quad (3)$$

と定義される. エネルギーが低ければ低いほど安定な状態であり, 最低エネルギー状態を基底状態と呼ぶ.

スピン σ_i の代わりに, 0 または 1 を取るバイナリ変数 n_i を考える. バイナリ変数の 2 次まででエネルギー関数が書ける場合, これを QUBO (Quadratic Unconstrained Binary Optimization) と呼ぶ. QUBO のエネルギー関数 \mathcal{H} は一般に

$$\mathcal{H} = - \sum_{(i,j) \in E} v_{ij} n_i n_j - \sum_{i \in V} w_i n_i - \text{const} \quad (4)$$

と書ける. ここで, v_{ij} は 2つのバイナリ変数 n_i, n_j の間の相互作用係数, w_i はバイナリ変数 n_i に作用する外部磁場である. ここで, const は定数を表す. バイナリ変数 $n_i = 0, 1$ とスピン $\sigma_i = -1, 1$ は以下の式で互いに変換することができる.

$$n_i = \frac{\sigma_i + 1}{2} \quad (5)$$

3.2 イジングモデルへのマッピング

本節では、制約付きスロット配置問題をイジングモデルにマッピングする。

イジングモデルにマッピングするために、バイナリ変数 $n_{i,a}$ ($1 \leq i \leq m$, $1 \leq a \leq t$) を以下のように定義する。

$$n_{i,a} = \begin{cases} 1 & (\text{部品 } c_i \text{ をスロット } s_a \text{ に配置するとき}) \\ 0 & (\text{上記以外}) \end{cases} \quad (6)$$

このように表現することによって、バイナリ変数を制約付きスロット配置問題に導入することができる。ここで、バイナリ変数 $n_{i,a} = 0, 1$ とスピン $\sigma_{i,a} = -1, 1$ は式 (5) より互いに変換することができる。

$$n_{i,a} = \frac{\sigma_{i,a} + 1}{2} \quad (7)$$

3.2.1 部品重複禁止制約

1つの部品はただ1つのスロットに必ず存在する。部品 c_i について、この制約を式で表すと以下ようになる。

$$\sum_{a=1}^t n_{i,a} = 1 \quad (1 \leq i \leq m) \quad (8)$$

以下のように \mathcal{H}_A を設定することにより、式 (8) を満たすときに \mathcal{H}_A は最小値 0 を取る。

$$\begin{aligned} \mathcal{H}_A &= \sum_{i=1}^m \left(1 - \sum_{a=1}^t n_{i,a} \right)^2 \\ &= \frac{1}{4} \sum_{i=1}^m \sum_{a=1}^t \sum_{b=1}^t \sigma_{i,a} \sigma_{i,b} + \frac{t-2}{2} \sum_{i=1}^m \sum_{a=1}^t \sigma_{i,a} \\ &\quad + \text{const.} \end{aligned} \quad (9)$$

3.2.2 スロット重複禁止制約

各スロットには高々1つの部品が置かれている。部品 c_i について、この制約を式で表すと以下ようになる。

$$\sum_{i=1}^m n_{i,a} = 1 \quad \text{or} \quad \sum_{i=1}^m n_{i,a} = 0 \quad (1 \leq a \leq t) \quad (10)$$

以下のように \mathcal{H}_B を設定することにより、式 (10) を満たすときに \mathcal{H}_B は最小値 $t/4$ を取る。

$$\begin{aligned} \mathcal{H}_B &= \sum_{a=1}^t \left(\frac{1}{2} - \sum_{i=1}^m n_{i,a} \right)^2 \\ &= \frac{1}{4} \sum_{a=1}^t \sum_{i=1}^m \sum_{j=1}^m \sigma_{i,a} \sigma_{j,a} \\ &\quad + \frac{m-1}{2} \sum_{a=1}^t \sum_{i=1}^m \sigma_{i,a} + \text{const.} \end{aligned} \quad (11)$$

3.2.3 付加制約項

特定の部品を一定の距離以内に置かなければならない場合が存在する。この制約を違反したときにイジングモデルのエネルギーが大きくなる項を導入するために、バイナリ変数 $q_{a,b,i,j}$ を以下のように定義する。

$$q_{a,b,i,j} = \begin{cases} 1 & (\text{部品 } c_i, c_j \text{ が置かれたスロット } s_{a(i)}, s_{b(j)} \text{ のマンハッタン距離が } d_{\text{limit}} \text{ を超えるとき}) \\ 0 & (\text{上記以外}) \end{cases} \quad (12)$$

このように表現することによって、付加制約項は以下のように定義できる。

$$\begin{aligned} \mathcal{H}_C &= \sum_{a=1}^t \sum_{b=1}^t \sum_{i=1}^m \sum_{j=1}^m q_{a,b,i,j} n_{i,a} n_{j,b} \\ &= \frac{1}{4} \sum_{a=1}^t \sum_{b=1}^t \sum_{i=1}^m \sum_{j=1}^m q_{a,b,i,j} \sigma_{i,a} \sigma_{j,b} \\ &\quad + \frac{1}{2} \sum_{a=1}^t \sum_{b=1}^t \sum_{i=1}^m \sum_{j=1}^m q_{a,b,i,j} \sigma_{i,a} + \text{const.} \end{aligned} \quad (13)$$

3.2.4 目的関数項

重み付き総配線長をコストとすると、式 (6) と r_{c_i, c_j} , l_{s_a, s_b} を用いて目的関数は以下のように定義できる。

$$\begin{aligned} \mathcal{H}_D &= \sum_{a=1}^t \sum_{b=1}^t \sum_{i=1}^m \sum_{j=1}^m r_{c_i, c_j} l_{s_a, s_b} n_{i,a} n_{j,b} \\ &= \frac{1}{4} \sum_{a=1}^t \sum_{b=1}^t \sum_{i=1}^m \sum_{j=1}^m r_{c_i, c_j} l_{s_a, s_b} \sigma_{i,a} \sigma_{j,b} \\ &\quad + \frac{1}{2} \sum_{a=1}^t \sum_{b=1}^t \sum_{i=1}^m \sum_{j=1}^m r_{c_i, c_j} l_{s_a, s_b} \sigma_{i,a} + \text{const.} \end{aligned} \quad (14)$$

3.2.5 最終的なエネルギー関数

最終的に得られるエネルギー関数は、各エネルギー関数 (9), (11), (13), (14) の重み付き和で以下のように表される。

$$\mathcal{H} = \alpha \mathcal{H}_A + \beta \mathcal{H}_B + \gamma \mathcal{H}_C + \delta \mathcal{H}_D \quad (15)$$

ただし、 $\alpha, \beta, \gamma, \delta (> 0)$ はハイパーパラメータである。

4. イジング解の解釈処理

4.1 解釈処理

現実的にはイジング計算機で得られる解は準最適解である。従って、イジング計算機で得られた解は制約条件を必ずしも満たさない。そのため、イジング計算機で得られた解を補正することで、制約条件を満たす解を得る方法を構築する必要がある。本章では、イジング計算機によって得られた解がスロット配置制約を満たさない場合について、スロット配置制約を満たす解に補正する方法を提案する。これを解釈処理と呼ぶ。

例として、部品数3個、スロット数4個のときのスロット配置問題をイジング計算機で解いたとしよう。図2はイジング計算機で得られた、制約を満たさない解の典型的な例と対応する解釈処理である。図2はスピン $\sigma_{i,a}$ ($1 \leq i \leq m$, $1 \leq a \leq t$) を並べたものであり、これは式 (9) と式 (11) が最小化されたとき、スロット方向に +1 のスピンのみ、部品方向に +1 のスピンのみ、これはそれぞれ部品重複禁止制約とスロット重複禁止制約を満たしていることに相当する。しかし実際は、式 (9) や式 (11) を最小化する解が得られない場合がある。制約を満たさないときのスピン配置は大きく分けて以下の3つの場合が考えられる。それぞれの場合に対応する解釈処理を以下に示す。

部品方向重複解決処理 (図2(a))

部品方向に +1 のスピンのみ2つ以上存在した場合、+1 のスピンの中で重み付き総配線長が最も小さくなる場所のみ +1 と固定する。それ以外のスピンを -1 に変化させる。この処理を部品方向重複解決処理と呼ぶ。

スロット方向重複解決処理 (図2(b))

スロット方向に +1 のスピンのみ2つ以上存在した場合、

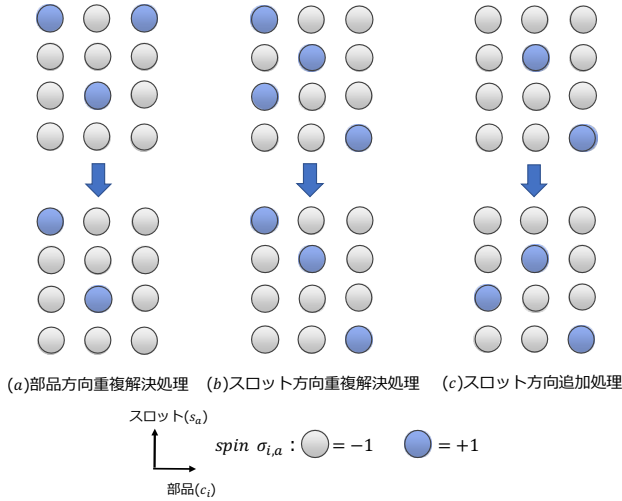


図 2: 制約を満たさなかった場合のパターンと解釈処理。

+1 のスピンの中で重み付き総配線長が最も小さくなる場所のみを +1 と固定する。それ以外のスピンを -1 に変化させる。この処理をスロット方向重複解決処理と呼ぶ。

スロット方向追加処理 (図 2(c))

スロット方向に関してスピンの全て -1 の列の存在する場合、+1 が配置可能な場所の中で重み付き総配線長が最も小さくなる場所を +1 に変化させる。この処理をスロット方向追加処理と呼ぶ。

4.2 提案アルゴリズム

m 個の部品の集合を $M = \{c_1, c_2, \dots, c_m\}$, o 行 p 列のスロットの集合を $S = \{s_1, s_2, \dots, s_t\}$ と定義する。ただし, $t = o \times p$ とする。また, イジングモデルが定義されるグラフの頂点集合を V とする。

+1 のスピンの乗る頂点の集合 V_{all}^+ を以下のように定義する。

$$V_{\text{all}}^+ = \{(i, a) | \sigma_{i,a} = +1, (i, a) \in V\} \quad (16)$$

ただし, $(i, a) \in V$ より, $1 \leq i \leq m, 1 \leq a \leq t$ である。

また, 各部品 c_i に対応する頂点のうち, スピンの向きが +1 である頂点の集合 V_i^+ 及び, 各スロット s_a に対応する頂点のうち, スピンの向きが +1 である頂点の集合 V_a^+ をそれぞれ以下のように定義する。

$$V_i^+ = \{(i, a) | \sigma_{i,a} = +1, (i, a) \in V, 1 \leq a \leq t\} \quad (17)$$

$$V_a^+ = \{(i, a) | \sigma_{i,a} = +1, (i, a) \in V, 1 \leq i \leq m\} \quad (18)$$

提案アルゴリズムをリスト 1 に示す。

5. 評価実験, 考察

3.2 節で提案した制約付きスロット配置問題のイジングモデルマッピングに対して, デジタルアニーラ (DA) [3] を用い計算機実験を行い, 制約付きスロット配置問題の解を求める。この際, スロット配置制約を満たさない解が得られたときには, 4.2 節で提案した解釈処理を用いて制約条件を満たす解を求める。本章では 5.1 節で提案手法と SA による制約付きスロット配置問題の解法手法の実験環境を説明する。5.2 節で実験結果, 考察を述べる。

リスト 1 解釈処理の提案アルゴリズム。

Step 1: 部品重複禁止制約の改良

```
for a = 1 : t do
  do while  $|V_a^+| \geq 2$ 
    部品方向重複解決処理を実行する。
  end
end
```

Step 2: スロット重複禁止制約の改良 (1)

```
for i = 1 : m do
  do while  $|V_i^+| \geq 2$ 
    スロット方向重複解決処理を実行する。
  end
end
```

Step 3: スロット重複禁止制約の改良 (2)

```
for i = 1 : m do
  if  $|V_i^+| = 0$  then
    スロット方向追加処理を実行する。
  end
end
```

5.1 実験環境

本実験は, o 行 p 列 ($4 \leq o \leq 6$) のスロットに対し, 部品数 $m = t/2, 3t/4, t$ について検討した。ただし, DA が 1024 個のスピンの, スピン間の相互作用が全結合であることから, 一度に問題を入力することが可能なサイズの範囲で検討した。最大の問題規模は, スロット 6 行 6 列, 部品 27 個である。それぞれの部品数, スロット数に対して, シード値を変えて 100 回シミュレーションを実行した。また, 任意の i, j について, $r_{c_i, c_j} \in [0, 10]$ の範囲を持つように設定した。さらに, 3.2.3 節で導入した付加制約は任意の部品間にかかるものとし, 付加制約の個数は $\lceil m/2 \rceil$ 個, マンハッタン距離の制限値は $d_{\text{limit}} = 3$ とする。

5.1.1 DA で提案手法を実行した場合

3.2 節で定式化したイジングモデルを DA を用いて実験した。DA の環境は OS が CentOS 7.5, CPU が Intel Xeon 2.4GHz, メモリが 187GB である。

本実験では, 3.2.5 節で導入したエネルギー関数 (15) を用いて計算機実験を行なった。

DA の計算機実験で使用するパラメータの設定項目を表 1 に示す。表 1 において, 開始温度は十分大きな温度からアニーリングを開始するために, 各問題の相互作用係数の最大値に 1000 を乗算した値を初期値として設定した。終了温度は問題の相互作用係数に比べて十分小さい温度まで下がるように 0.1 とした。アニーリング回数は 50000 から 100000 の範囲で問題ごとに変更して設定した。開始温度 t_{start} , 終了温度 t_{end} , アニーリング回数 iteration としたとき, 減衰率 t_{decay} は以下の式で計算し設定した。

$$t_{\text{decay}} = \left(\frac{t_{\text{end}}}{t_{\text{start}}} \right)^{\frac{1}{\text{iteration}}} \quad (19)$$

式 (15) で表されるエネルギー関数のハイパーパラメータは, $m = 8, o = 4$ のとき $\alpha = 30, \beta = 30, \gamma = 100, \delta = 2$ に設定し, 部品数, スロット数を増やすにつれて, γ, δ の値を小さくした。ただし, α, β は全ての部品数, ス

表 1: DA で使用したパラメータ設定項目.

項目	値
スピン初期値	ランダム
開始温度	$\max(\text{weight}) \times 1000$
最終温度	0.1
アニーリング回数	50000 ~ 100000
減衰率	上記から計算

weight : 各問題の相互作用係数.

ロット数で固定である. いま想定している 3 つの問題は, $t \sim m$ である. 制約項 (9) 式は $O(t^2m)$, 制約項 (11) 式は $O(tm^2)$ である. 一方, 付加制約項 (13), 目的関数項 (14) は $O(t^2m^2)$ であるため, 問題の規模が大きくなるにつれて付加制約項, 目的関数項が支配的になる. それを防ぐために γ, δ の値を小さくした.

5.1.2 SA で実行した場合

DA の結果と比較するために, 制約付きスロット配置問題をイジングモデルに変換することなく Simulated Annealing (SA) で解いた. SA の環境は OS が CentOS 6.9, CPU が Intel Xeon 2.5GHz, メモリが 264GB であり, C 言語で実装した. また, o 行 o 列のスロット ($t = o \times o$), m 個の部品の問題において, 付加制約を満足しなかった場合, 式 (1) と式 (12) を用いて, 以下の項をエネルギー関数にコストとして足し合わせるものとする.

$$10m(o+o) \sum_{a=1}^t \sum_{b=1}^t \sum_{i=1}^m \sum_{j=1}^m q_{a,b,i,j} l_{s_a,s_b} \quad (20)$$

[12] に基づき, SA で使用したパラメータ設定項目を表 2 に示す.

SA による制約付きスロット配置問題の解法手法を以下に示す.

- (1) 初期状態をランダムに設定する.
- (2) ランダムに 2 つの部品を交換し, Metropolis 法を実行する. エネルギー関数が小さくなる場合は受理, エネルギー関数が大きくなる場合は確率 $p = \exp(-\Delta E/T)$ で受理する. ここで, T は温度, ΔE は現在の解と新しい解候補との差を表す.
- (3) Inner loop の回数だけ (2) を試行したら温度を下げる. すなわち, 現在のステップ数を k , 冷却率を r とすると, $T(k+1) = rT(k)$ となる.
- (4) Outer loop の回数だけ (2) - (3) を繰り返す.

5.2 実験結果

5.2.1 DA による計算機実験

DA による解釈処理を適用しなかった場合と適用した場合の結果を表 3 に示す. 解釈処理を適用しなかった場合, スロット配置制約を満たさない場合がいくつか存在するが, 解釈処理を適用することで, 必ずスロット配置制約を満足する.

表 3 において, 「最小値」, 「平均値」は 100 回の計算機実験における重み付き総配線長 L の最小値, 平均値を表す. 「確率 (スロット配置制約)」は, 100 回の計算機実験でスロット配置制約を満たした確率, 「確率 (付加制約)」は, スロット配置制約を満たした解のうち, 付加制約を全て満たした確率を表す.

表 3 から, 解釈処理を適用しなかった場合でも全ての問

表 2: SA で使用したパラメータ設定項目.

項目	値
初期解	ランダム
開始温度	$m \times t/10$
冷却率	0.95
Inner loop	1000
Outer loop	100

題で準最適解を得ることができたが, スロットに対して密に部品が配置されている場合, スロット配置制約を満足する確率が低くなっていることがわかる.

5.2.2 実行時間の比較

次に, DA と SA の実行時間の比較を表 4 に示す. それぞれの実行時間を以下のように定義する.

提案手法 (解釈処理あり): 表 4 の「提案手法 (解釈処理あり)」における「アニーリング時間」, 「解釈処理時間」, 「合計」を以下のように定義する. 「アニーリング時間」は 100 回の計算機実験のうち, 1 回のアニーリングにかかった時間の平均値を示している. 「解釈処理時間」はスロット配置制約を満たさなかったスピンに対して解釈処理を適用した実行時間の平均値を示している. 「合計」は「アニーリング時間」と「解釈処理時間」を足し合わせた時間であり, これを「提案手法 (解釈処理あり)」の実行時間とする.

SA: 表 4 の「SA」を以下のように定義する. 「SA」は SA による制約付きスロット配置問題の解が表 4 における DA による解の最小値より小さくなった回数が Inner loop の回数の過半数 (500 回) に到達するまで実行時間である. 表 4 における DA による解の最小値に到達しなかった場合, SA の実行時間は 100000 ステップ (Inner loop の上限 \times Outer loop の上限) にかかった時間とする.

さらに, スピン数に対する DA と SA の実行時間を図 3 に示す.

表 4, 図 3 より, 提案手法を DA で解いた場合, SA と同程度の精度の準最適解を得るのに必要な DA の時間が最大で約 1/50 に圧縮することができた.

6. おわりに

本稿では, 「制約付きスロット配置問題」をイジングモデルへマッピングする手法を提案し, デジタルアニーラを用いて計算機実験を行った. デジタルアニーラで計算機実験をした場合, 最大でスロット数 6 行 6 列, 部品数 27 個の問題で制約付きスロット配置問題の準最適解を得ることができた. また, SA と同精度の制約付きスロット配置問題の準最適解を最大で約 1/50 倍の実行時間で得ることができた.

今後の課題として, より精度の高い準最適解を得るために, エネルギー関数のハイパーパラメータやデジタルアニーラのより良いパラメータの設定方法の提案が挙げられる. さらに, 現実的な問題に近づくために, より規模の大きな問題を解く必要があると考えられる.

謝辞 本研究の実施にあたり, 貴重なご議論・ご助言をいただきました富士通研究所・富田憲範研究員ほか, 同研究所のみなさまに感謝申し上げます.

表 3: DA を利用した場合の実験結果.

m	t	スピン数	提案手法 (解釈処理なし)				提案手法 (解釈処理あり)		
			最小値	平均値	確率 (スロット配置制約)	確率 (付加制約)	最小値	平均値	確率 (付加制約)
8	4×4	128	188	201.0	100.0	100.0	188	201.0	100.0
12	4×4	192	650	688.3	100.0	99.0	650	688.3	99.0
16	4×4	256	1319	1362.7	44.0	93.2	1319	1381.9	93.0
12	5×5	300	652	693.2	100.0	99.0	652	693.2	99.0
18	5×5	450	1618	1679.2	100.0	100.0	1618	1679.2	100.0
25	5×5	625	3921	4031.3	77.0	92.2	3921	4043.1	86.0
18	6×6	648	1601	1681.3	100.0	99.0	1601	1681.3	99.0
27	6×6	972	4637	4753.5	24.0	100.0	4637	4834.5	87.0

表 4: 実行時間の比較.

m	t	実行時間 [sec]			
		SA	提案手法 (解釈処理あり)		
			アニーリング時間	解釈処理時間	合計
8	4×4	4.0×10^{-2}	4.2×10^{-3}	0.0	4.2×10^{-3}
12	4×4	5.3×10^{-2}	4.2×10^{-3}	0.0	4.2×10^{-3}
16	4×4	6.0×10^{-2}	4.2×10^{-3}	2.6×10^{-5}	4.2×10^{-3}
12	5×5	6.1×10^{-2}	4.2×10^{-3}	0.0	4.2×10^{-3}
18	5×5	1.2×10^{-1}	4.8×10^{-3}	0.0	4.8×10^{-3}
25	5×5	1.8×10^{-1}	4.8×10^{-3}	3.7×10^{-5}	4.8×10^{-3}
18	6×6	1.4×10^{-1}	4.8×10^{-3}	0.0	4.8×10^{-3}
27	6×6	2.7×10^{-1}	5.1×10^{-3}	2.2×10^{-4}	5.3×10^{-3}

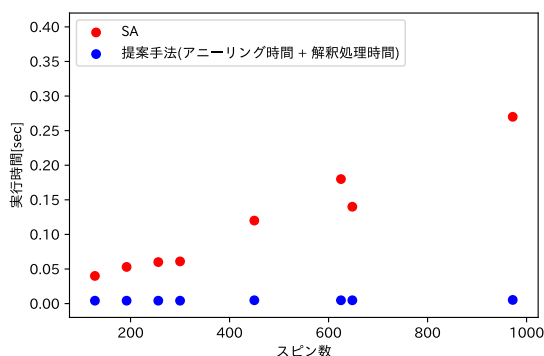


図 3: DA と SA の実行時間.

参考文献

- [1] Johnson, M., Amin, M., Gildert, S., Lanting, T., Hamze, F., Dickson, N., Harris, R., Berkley, A., Johansson, J., Bunyk, P., Chapple, E., Enderud, C., Hilton, J., Karimi, K., Ladizinsky, E., Ladizinsky, N., Oh, T., Perminov, I., Rich, C., Thom, M., Tolkacheva, E., Truncik, C., Uchaikin, S., Wang, J., Wilson, B. and Rose, G.: Quantum annealing with manufactured spins, *Nature*, Vol. 473, No. 7346, pp. 194–198 (2011).
- [2] Yamaoka, M., Yoshimura, C., Hayashi, M., Okuyama, T., Aoki, H. and Mizuno, H.: A 20k-spin Ising chip to solve combinatorial optimization problems with CMOS annealing, *IEEE Journal of Solid-State Circuits*, Vol. 51, No. 1, pp. 303–309 (2016).
- [3] Tsukamoto, S., Takatsu, M., Matsuba, S. and Tamura, H.: An accelerator architecture for combinatorial optimization problems, *Fujitsu Scientific & Technical Journal*, Vol. 53, No. 5, pp. 8–13 (2017).
- [4] Utsunomiya, S., Takata, K. and Yamamoto, Y.: Mapping of Ising models onto injection-locked laser systems, *Opt. Express*, Vol. 19, No. 19, pp. 18091–18108 (2011).
- [5] Yoshimura, C., Hayashi, M., Okuyama, T., and Yamaoka M.: FPGA-based annealing processor for Ising model, *Proc. 2016 Fourth International Symposium on Computing and Networking (CANDAR)*, pp. 436–442 (2016).
- [6] Kanamaru, S., Oku, D., Tawada, M., Tanaka, S., Hayashi, M., Yamaoka, M., Yanagisawa, M., Togawa, N.: Efficient Ising Model Mapping to Solving Slot Placement Problem, *Proc. IEEE International Conference on Consumer Electronics*, pp. 481–486 (2019).
- [7] Srinivasan, S., Kamakoti, V., Bhattacharya, A.: Towards Quick Solutions for Generalized Placement Problem, *Electronic System Design (ISED), 2011 International Symposium on*, pp. 19–21 (2011).
- [8] Rajeswari, M., Jaiganesh, S., Sujatha, P., Vengattaraman, T., Dhavachelvan, P.: A study and scrutiny of diverse optimization algorithm to solve multi-objective quadratic assignment problem, *Communication and Electronics Systems (ICCES), International Conference on*, pp. 21–22 (2016).
- [9] Tanaka, S., Tamura, R., Chakrabarti, B. K.: Quantum Spin Glasses, Annealing and Computation, *Cambridge University Press*, (2017).
- [10] Lucas, A.: Ising formulations of many NP problems, *Frontiers in Physics*, Vol. 2, pp. 5 (2014).
- [11] Terada, K., Tanaka, S., Hayashi, M., Yamaoka, M., Yanagisawa, M., and Togawa, N.: A fully-connected Ising model mapping method and its evaluation for 20K-spin CMOS annealing machines (in Japanese), *Proc. of Design Automation Symposium 2017*, pp. 163–168 (2017).
- [12] Selim, S. Z., Alsultan, K.: A simulated annealing algorithm for the clustering problem, *Pattern Recognition*, Vol. 24, No 10, pp. 1003–1008 (1991).