

初期処理を導入した イジング計算機によるスロット配置問題の解法

深田 佳佑¹ マチュー パリジ² 富田 憲範³ 戸川 望⁴

概要: 近年、組合せ最適化問題を効率的に解く計算機としてイジングモデルを用いた様々なイジング計算機が提案されている。組合せ最適化問題とは、ある制約条件のもとで目的関数を最大化または最小化する変数の組合せを探索する問題であり、変数の増加とともに計算量も指数的に増加する場合がある。イジング計算機を用いることで高速かつ高精度で準最適解を得ることが可能である。本稿では、実イジング計算機を用いて効率よく組合せ最適化問題を解法するための初期処理を提案する。初期処理とは組合せ最適化問題に対して、制約条件を満足した上で比較的良好な解を与える初期変数の値を与えるもので、その上で実イジング計算機によりアニーリング処理を実行する。最後にアニーリング処理によって得られた解に対して制約条件を満足するよう補正処理を行う。本稿では、組合せ最適化問題としてスロット配置問題を取り上げ、対交換による初期処理とランダム交換による初期処理を提案する。実イジング計算機を用いた評価実験の結果、初期処理の適用により解が改良されることを確認した。

1. はじめに

近年、組合せ最適化問題を効率的に解く計算機としてイジングモデル [1] を用いた様々なイジング計算機が研究されている [2], [3], [4], [5], [6]。組合せ最適化問題とは、与えられた制約を満たした上で目的関数を最大化または最小化する変数の組合せを探索する問題であり、変数の増加とともに計算量も指数的に増加する場合がある。従来のノイマン型コンピュータでは最適解を得ることが困難であるが、非ノイマン型コンピュータであるイジング計算機を用いることで、高速かつ高精度で準最適解を得ることが可能である。準最適解とは、問題の制約は満たしているが目的関数が厳密に最大化または最小化されていない解である。

イジング計算機とは、統計力学上のモデルの一つであるイジングモデルをベースとした計算機で、組合せ最適化問題を解くことに特化している。組合せ最適化問題を解くために、イジングモデル又はイジングモデルと等価な Quadratic Unconstrained Binary Optimization (QUBO) モデルに組合せ最適化問題をマッピングする。現在では様々な組合せ最適化問題がイジングモデル又は QUBO モデルにマッピングされ、イジング計算機によって計算され

ている [7], [8], [9], [10], [11], [12]。

1.1 スロット配置問題

組合せ最適化問題の一つにスロット配置問題がある [12], [13]。スロット配置問題とは、複数のスロットに一定の配線数で接続された複数の部品を配置する問題であり、論理ブロックの最適配置や回路面積の最小化などに応用できる [14]。スロット配置問題は NP-困難な問題であり [15]、従来は Simulated Annealing (SA) [16] や分枝限定法 [17] を用いて解かれてきた。また、スロット配置問題を QUBO モデルにマッピングしてイジング計算機を用いて解く手法も提案されている [12]。手法 [12] では、イジング計算機の計算後に解釈処理を適用することで、得られる準最適解を改善している。解釈処理とは、イジング計算機の出力を組合せ最適化問題の制約を満たすように解を補正する処理である。しかし、イジング計算機の計算前に特別な処理を実行しておらず、スピンの値を全て 0 としてイジング計算機に入力している。

本稿では、イジング計算機に入力する初期スピンを適当な値に決定する初期処理として、対交換法とランダム交換法の二つを提案する。対交換法とは、元のスロット配置問題のランダムな許容解を生成し、スロットの部品の割り当てにおいて全てのスロット対についてスロットに割り当てられている部品の交換を試行する。そして実際に重み付き総配線長を最小化する部品対を交換する。この処理を繰り返す。

¹ 早稲田大学基幹理工学部情報通信学科

² 富士通研究所/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

³ 富士通研究所

⁴ 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

返し実行し、得られた解を初期スピンとする手法である。許容解とは、制約を満たしている解である。ランダム交換法とは、元のスロット配置問題のランダムな許容解を生成し、スロットの部品の割り当てにおいてランダムに選択したスロット対の交換をある条件のもと一定のイテレーション数だけ繰り返し、得られた解を初期スピンとする手法である。

本稿の実験では、スロット配置問題に対して提案する二つの初期処理を適用し、実イジング計算機の一つであるデジタルアニーラ (DA) [18], [19] を用いて解法し解を評価する。さらに取得した解に対して解釈処理を適用し、初期処理の適用の有無による結果を比較し考察する。

1.2 本稿の貢献

本稿の貢献は以下の三つである。

1. 初期処理として、元のスロット配置問題のランダムな許容解を生成し、スロットの部品の割り当てにおいて全てのスロット対についてスロットに割り当てられている部品の交換を試行する、そして実際に重み付き総配線長を最小化する部品対を交換する。この処理を繰り返し実行し、得られた解を初期スピンとする対交換法を提案した。
2. 初期処理として、元のスロット配置問題のランダムな許容解を生成し、スロットの部品の割り当てにおいてランダムに選択したスロット対の交換をある条件のもと一定のイテレーション数だけ繰り返し、得られた解を初期スピンとするランダム交換法を提案した。
3. 提案した二つの初期処理をスロット配置問題に適用し、実際にイジング計算機を用いて解を取得した。既存手法 [12] と比較して、提案手法では重み付き総配線長の最小値を対交換法でおよそ 2.29%、ランダム交換法でおよそ 2.40% 削減することができた。また、重み付き総配線長の平均値は対交換法でおよそ 5.47%、ランダム交換法でおよそ 5.54% 削減することができた。

1.3 本稿の構成

本稿は以下のように構成される。2章では、スロット配置問題を定義し、例を用いて問題の概要を説明する。3章では、イジングモデル及び QUBO モデルを紹介し、スロット配置問題を QUBO モデルへマッピングする手法を説明する。4章では、イジング計算機で計算する前に適用する初期処理として対交換法とランダム交換法を提案する。続いて、イジング計算機で計算した後に適用する解釈処理について説明する。5章では、提案した初期処理をスロット配置問題に適用して、イジング計算機による実験より解を取得する。また、取得した解に対して解釈処理を適用し、初期処理の適用の有無による結果を比較し考察する。6章では、本稿を総括する。

2. スロット配置問題

本章では、スロット配置問題を定義および定式化し、例を用いて問題について説明する。

2.1 スロット配置問題の定義

本稿では現実には用いられる組合せ最適化問題の一つとしてスロット配置問題 [14] を取り上げる。スロット配置問題とは、複数の格子状のスロットと、一定の配線数で接続された複数の部品がそれぞれ与えられたとき、各部品間の配線数と各部品が配置されたスロット間の距離の積の総和が最小となるような部品の配置を求める問題である。スロット配置問題は、回路面積の最小化や論理ブロックの最適配置に応用でき、集積回路自動設計や配送管理システムで重要な役割を果たす [14]。現実問題では、短時間でスロット配置問題の準最適解を求めることは、厳密な最適解を求めることと同程度の需要がある。

2.2 スロット配置問題の定式化

m 個の部品集合を $M = \{c_1, c_2, \dots, c_m\}$ 、 p 行 q 列のスロット集合を $S = \{s_1, s_2, \dots, s_t\}$ と定義する。ただし、 $t = p \times q$ とする。スロット s_a が a_1 行 a_2 列に、スロット s_b が b_1 行 b_2 列 ($1 \leq a_1, b_1 \leq p, 1 \leq a_2, b_2 \leq q$) にあるとすると、二つのスロット s_a, s_b 間のマンハッタン距離 $l(s_a, s_b)$ は次の式 (1) のように表せる。

$$l(s_a, s_b) = |a_1 - b_1| + |a_2 - b_2| \quad (1)$$

また、各部品間には部品と部品を接続する非負整数の配線数が与えられ、二つの部品 c_i, c_j ($1 \leq i, j \leq m$) 間の配線数を $w(c_i, c_j)$ とする。なお、 $w(c_i, c_j) = w(c_j, c_i)$ である。部品 c_i が置かれたスロットを $s(c_i)$ 、部品 c_j が置かれたスロットを $s(c_j)$ 、部品 c_i, c_j 間の重み付き配線長を $w(c_i, c_j) \times l(s(c_i), s(c_j))$ と定義すると、全ての部品間の重み付き配線長の総和 L は次の式 (2) のように表せる。

$$L = \sum_{i=1}^{m-1} \sum_{j=i+1}^m w(c_i, c_j) l(s(c_i), s(c_j)) \quad (2)$$

なお、任意の部品はただ一つのスロットに必ず存在するものとし、これを部品重複禁止制約と呼ぶ。また、任意のスロットには高々一つの部品を配置するものとし、これをスロット重複禁止制約と呼ぶ。部品重複禁止制約とスロット重複禁止制約を合わせてスロット配置制約と呼ぶ。以上のものと、スロット配置問題を次のように定義する。

定義 1. m 個の部品と p 行 q 列のスロットがあり、各部品間には部品と部品を接続する非負整数の配線数が与えられている。 L を重み付き総配線長とする。スロット配置問題とは、部品重複禁止制約とスロット重複禁止制約で構成されるスロット配置制約を満たした上で、 L を最小化するよ

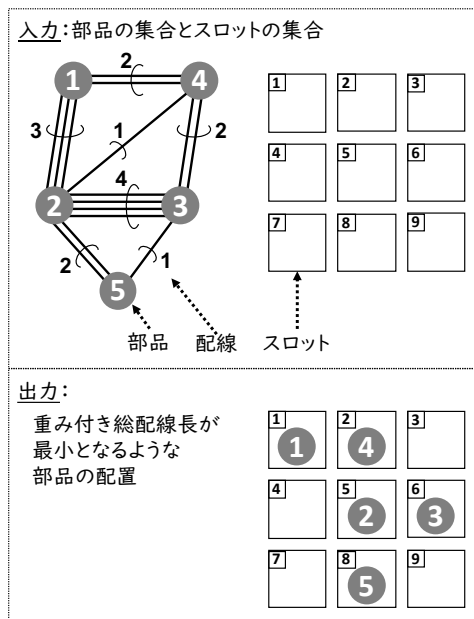


図 1: スロット数 $t = 9$, 部品数 $m = 5$ のスロット配置問題の例.

うな部品の配置を求める問題である.

2.3 スロット配置問題の例

スロット数 $t = 9$ ($p = 3, q = 3$), 部品数 $m = 5$ のスロット配置問題の例を図 1 に示す. 図 1 の入力は 5 個の部品と 9 個のスロットがあり, 各部品間はある配線数で接続されている. 例えば部品 1 と 2 は配線数 3, 部品 1 と 4 は配線数 2 で接続されている. 図 1 の出力は 9 個のスロットに 5 個の部品を配置した問題の解の例である. 例えばスロット 1 には部品 1 が, スロット 2 には部品 4 が配置されている. 図 1 の重み付き総配線長 L は, 次のように計算できる.

$$L = \sum_{i=1}^{m-1} \sum_{j=i+1}^m w(c_i, c_j) l(s(c_i), s(c_j))$$

$$= 3 \times 2 + 2 \times 1 + 4 \times 1 + 1 \times 1 + 2 \times 1 + 2 \times 2 + 1 \times 2$$

$$= 21$$

3. スロット配置問題の QUBO モデルマッピング

本章では, イジングモデル及び QUBO モデルについて説明し, 2 章で定義したスロット配置問題を QUBO モデルにマッピングする手法について説明する.

3.1 イジングモデルと QUBO モデル

イジングモデルとは, 物理学の分野の一つである統計力学における基礎モデルである [1]. 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, システム全体でどのような振る舞

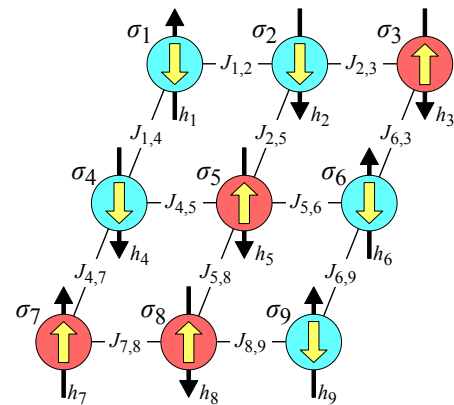


図 2: イジングモデルの例.

いを示すかを調べるためのモデルである. イジングモデルは, スピンが配置される頂点集合 V , 及び頂点間の辺集合 E で構成される無向グラフ $G = (V, E)$ 上で定義される. 二つの頂点 i, j が接続されているとき, $(i, j) \in E$ は頂点 i, j 間の接続辺を表す. 頂点 i に配置されるスピンを σ_i とする. σ_i は ± 1 のいずれかを取り, $+1$ を上向きのスピン, -1 を下向きのスピンとする. 二つのスピン σ_i, σ_j 間に作用する相互作用係数を $J_{i,j} \in \mathbb{R}$ (\mathbb{R} は実数集合), スピン σ_i に作用する外部磁場係数を $h_i \in \mathbb{R}$ と定義すると, イジングモデルのエネルギー関数 \mathcal{H} は, 次の式 (3) のように表せる.

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

\mathcal{H} は値が小さいほど安定な状態であり, 値が最小の状態を基底状態と呼ぶ. イジングモデルの例を図 2 に示す.

イジングモデルのスピン σ_i が ± 1 の 2 値をとる代わりに, 0 または 1 をとるバイナリ変数 n_i を用いたモデルを Quadratic Unconstrained Binary Optimization (QUBO) モデルと呼ぶ. QUBO モデルのエネルギー関数 \mathcal{H} は, 次の式 (4) のように表せる.

$$\mathcal{H} = - \sum_{(i,j) \in E} J'_{i,j} n_i n_j - \sum_{i \in V} h'_i n_i - \text{const} \quad (4)$$

$J'_{i,j}$ は二つのバイナリ変数 n_i, n_j 間に作用する相互作用係数, h'_i はバイナリ変数 n_i に作用する外部磁場係数, const は定数である. バイナリ変数 n_i は, 式 (5) でスピン σ_i から変換できる.

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

3.2 QUBO モデルマッピング

本節では, スロット配置問題を QUBO モデルにマッピングする手法について説明する. 部品数を m , スロット数を t , i 番目の部品を c_i ($1 \leq i \leq m$), a 番目のスロットを s_a ($1 \leq a \leq t$) とし, バイナリ変数 $n_{a,i}$ を次のように定義する.

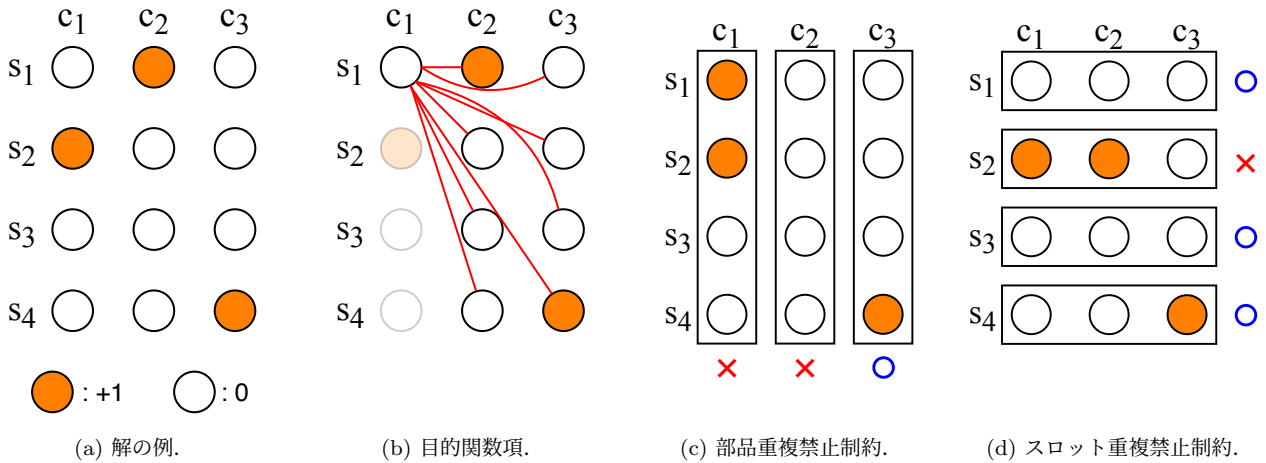


図 3: スロット数 $t = 4$, 部品数 $m = 3$ における解の例と制約条件.

$$n_{a,i} = \begin{cases} 1 & (\text{スロット } s_a \text{ に部品 } c_i \text{ を配置}) \\ 0 & (\text{上記以外}) \end{cases} \quad (6)$$

スロット数 $t = 4$, 部品数 $m = 3$ の場合におけるスロット配置問題の解の例を図 3(a) に示す. オレンジ色の丸はバイナリ変数 $n_{a,i}$ が 1, 白色の丸はバイナリ変数 $n_{a,i}$ が 0 であることを表す.

3.2.1 目的関数項

目的関数は, 重み付き総配線長である. 例を図 3(b) に示す. バイナリ変数 $n_{a,i}, n_{b,j}$, 二つの部品 c_i, c_j 間の配線数 $w(c_i, c_j)$, 二つのスロット s_a, s_b 間のマンハッタン距離 $l(s_a, s_b)$ を用いて, 目的関数 \mathcal{H}_A は次の式 (7) のように表せる.

$$\mathcal{H}_A = \sum_{a=1}^t \sum_{b=1}^t \sum_{i=1}^{m-1} \sum_{j=i+1}^m w(c_i, c_j) l(s_a, s_b) n_{a,i} n_{b,j} \quad (7)$$

\mathcal{H}_A の最小値は問題の入力に依存する. \mathcal{H}_A の最小値を x_a とする.

3.2.2 部品重複禁止制約

部品重複禁止制約とは, i 番目の部品 c_i はただ一つのスロットにのみ存在するという制約である. 例を図 3(c) に示す. 青い丸は本制約を満たし, 赤いバツは本制約に違反していることを表す. 本制約を定式化すると, 次の式 (8) のように表せる.

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

全ての部品 c_i が式 (8) を満たすときに最小値をとるようなエネルギー関数 \mathcal{H}_B を導入すると, 次の式 (9) のように表せる.

$$\mathcal{H}_B = \sum_{i=1}^m \left(1 - \sum_{a=1}^t n_{a,i} \right)^2 \quad (9)$$

\mathcal{H}_B の最小値は 0 をとる.

3.2.3 スロット重複禁止制約

スロット重複禁止制約とは, a 番目のスロット s_a には高々一つの部品が配置されているという制約である. 例を図 3(d) に示す. 青い丸は本制約を満たし, 赤いバツは本制約に違反していることを表す. 本制約を定式化すると, 次の式 (10) のように表せる.

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

全てのスロット s_a が式 (10) を満たすときに最小値をとるようなエネルギー関数 \mathcal{H}_C を導入すると, 次の式 (11) のように表せる.

$$\mathcal{H}_C = \sum_{a=1}^t \left(\frac{1}{2} - \sum_{i=1}^m n_{a,i} \right)^2 \quad (11)$$

\mathcal{H}_C の最小値は問題の入力に依存する. \mathcal{H}_C の最小値を x_c とする.

3.2.4 QUBO モデルのエネルギー関数

3.2.1 節から 3.2.3 節にかけて紹介した三つのエネルギー関数を重み付き和で表すと, 最終的なエネルギー関数 \mathcal{H} は次の式 (12) のように表せる.

$$\mathcal{H} = \alpha \mathcal{H}_A + \beta \mathcal{H}_B + \gamma \mathcal{H}_C \quad (12)$$

$\alpha, \beta, \gamma (> 0)$ はハイパーパラメータである. また, エネルギー関数 \mathcal{H} は最小値 $\alpha x_a + \gamma x_c$ を取り, このとき基底解となる. 基底解が得られたときのスピンの最適解となる.

4. 初期処理と解釈処理

本章では, スロット配置問題をイジング計算機で計算する前に適用する初期処理として対交換法とランダム交換法を提案する. 続いて, イジング計算機で計算した後に適用する解釈処理 [12] について説明する.

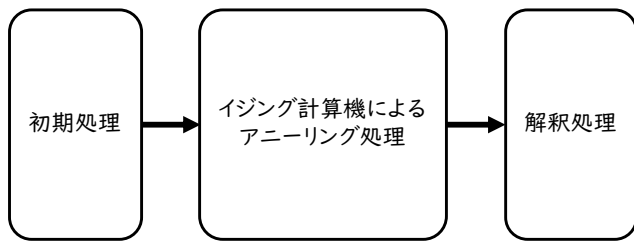


図 4: スロット配置問題を解くフロー.

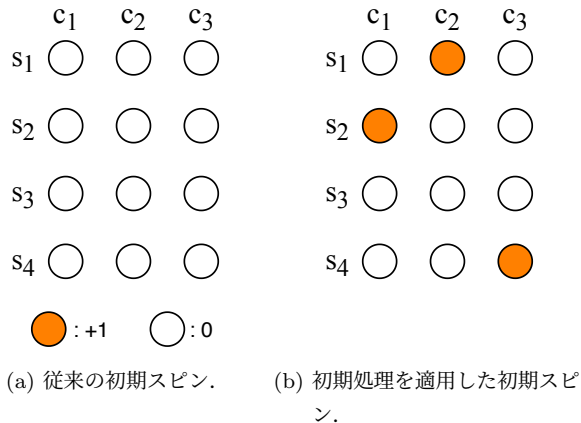


図 5: 従来の初期スピンと初期処理を適用した初期スピンの例.

4.1 初期処理

一般にイジング計算機は、式 (3) や式 (4) で表されるエネルギー関数を小さくするが、イジング計算機によって得られる解は必ずしも基底解とは限らず、式 (3) や式 (4) が最小化されるとは限らない。そのため、元の組合せ最適化問題の制約条件が満足されるとは限らない。そこで、手法 [12] ではスロット配置問題を対象に、イジング計算機でアニーリング処理を実行した後、スロット配置制約が満足されるよう解の補正処理を実行している (これを解釈処理と呼ぶ)。

しかしながら、手法 [12] では、イジング計算機によるアニーリング処理の前に処理は適用されておらず、全てのスピンの初期値を 0 としている。大域的な探索が可能な量子アニーリングでは十分に時間をかければスピンの初期値によらず最適解が得られることが保証されている [20]。実際には計算時間は有限であるため、得られる解はスピンの初期値に依存し、最適解が得られるとは限らない。時間と解の精度はトレードオフの関係にあり、現実の問題では短時間で準最適解を求めることは、厳密な最適解を求めることと同程度の需要がある。

そこで、イジング計算機に入力するスピンの初期値を適切に設定することで、短い実行時間と得られる準最適解の精度の向上の両立を目指す。この処理を初期処理と呼ぶ。組合せ最適化問題には制約条件が存在するため、解空間に多峰性があり大域的な最適解を得るには時間がかかる。初期処理により大域的な最適解に近い初期値を与えることで解の探索範囲を限定し、得られる準最適解の精度の向上や

アルゴリズム 1 対交換法

```

ランダムに部品をスロットに割り当てる；
while 重み付き総配線長の改善が見られる間 do
  全てのスロットの対について部品の交換を試行する；
  重み付き総配線長を最小にする部品対を実際に交換する；
end while
return (この時のスロット配置)
  
```

アルゴリズム 2 ランダム交換法

```

ランダムに部品をスロットに割り当てる；
for (イテレーション数 N) do
  ランダムに二つのスロットについて部品を交換する；
  if (元の重み付き総配線長 < 新しい重み付き総配線長) then
    交換した部品を元に戻す；
  end if
end for
return (この時のスロット配置)
  
```

実行時間の短縮が期待される。

図 4 に初期処理と解釈処理を含むイジング計算のフローを示す。まず初期処理によって、組合せ最適化問題の制約条件を満たし最適解に比較的近い解を与える。その後、イジング計算機によってアニーリング処理を実行する。最後に、解釈処理によって組合せ最適化問題の制約条件を満たすよう解を改変する。

図 5(a) に従来の初期スピンを、図 5(b) に初期処理を適用した初期スピンの例を示す。オレンジ色の丸はスピンの値が 1、白色の丸はスピンの値が 0 であることを表す。図 5(a) は全てのスピンの値が 0 となっているが、図 5(b) はスロット配置制約を満たすようなスピンの配置となっている。

4.1.1 対交換法

スロット配置問題の初期処理として対交換法を提案する。対交換法とは、元のスロット配置問題のランダムな許容解を生成し、スロットの部品の割り当てにおいて全てのスロット対についてスロットに割り当てられている部品の交換を試行する。そして実際に重み付き総配線長を最小化する部品対を交換する。なお、部品が割り当てられていない空のスロットと部品が割り当てられているスロットが選択された場合には、その部品を空のスロットに移動する。この処理を繰り返し実行するのが対交換法である。対交換法のアルゴリズムをアルゴリズム 1 に示す。対交換法を一回実行することで、初期スピンを一つ取得する。

対交換法は、スロット配置制約を満足したランダムな解から始まり、貪欲に二つのスロットを交換し、重み付き総配線長を小さくするものである。局所最適解に陥る可能性があるが、初期解として比較的良い解を与えることができる。一方、重み付き総配線長が改善されなくなるまで処理を繰り返すため、初期処理の実行時間が長くなる可能性がある。

4.1.2 ランダム交換法

スロット配置問題の初期処理としてランダム交換法を提

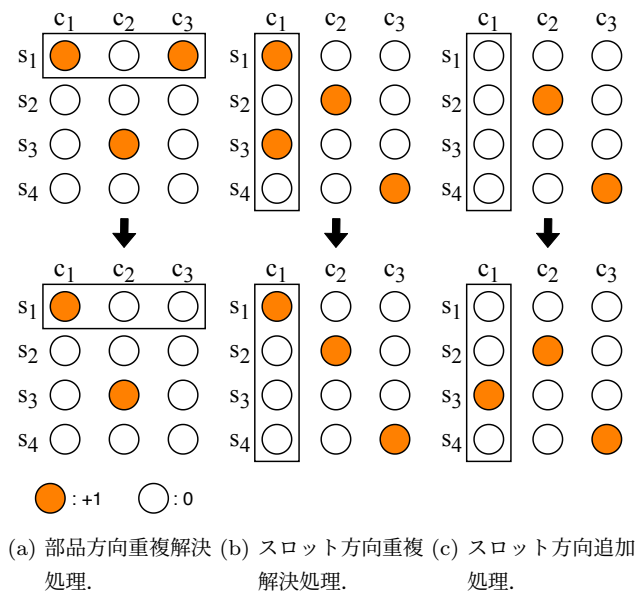


図 6: 解釈処理の例.

案する。ランダム交換法とは、元のスロット配置問題のランダムな許容解を生成し、ランダムに選択したスロット対の交換がある条件のもと一定のイテレーション数だけ繰り返し、得られた解を初期スピンの手法である。ランダム交換法のアルゴリズムをアルゴリズム 2 に示す。イテレーション数は問題の規模に応じて適当に与えるものとする。ランダム交換法を一回実行することで、初期スピンを一つ取得する。

ランダム交換法は、スロット配置制約を満足したランダムな解から始まり、一定回数ランダムに二つのスロットを交換し、重み付き総配線長を小さくするものである。イテレーション数に応じて、初期解として比較的良好な解を与えることができる。またイテレーション数を設定することで、初期処理の実行時間を調整することができる。

4.2 解釈処理

イジング計算機で得られる解は必ずしもスロット配置制約を満たしているとは限らない。そのため、得られた解を補正することで、スロット配置制約を満たす解を得る方法を構築する必要がある。本節では、イジング計算機で得られた解がスロット配置制約を満たしていなかった場合、スロット配置制約を満たすように解を補正する方法について説明する [12]。解釈処理には、部品方向重複解決処理、スロット方向重複解決処理、スロット方向追加処理の三つがある。例として、スロット数 $t=4$ 、部品数 $m=3$ の場合の解釈処理を図 6 に示す。図 6 の上半分の図は、スロット配置問題をイジング計算機で解いたときに得られる、スロット配置制約を満たしていない解の典型例で、図 6 の下半分はそれぞれの場合の解釈処理を適用した結果である。

表 1: ハイパーパラメータの値.

m	t	スピンの数	α	β	γ
8	4×4	128	1	50	50
12	4×4	192	1	120	120
16	4×4	256	1	165	165
12	5×5	300	1	165	165
18	5×5	450	1	200	200
25	5×5	625	1	330	330
18	6×6	648	1	300	300
27	6×6	972	1	350	350
36	6×6	1296	1	500	500
24	7×7	1176	1	600	600
36	7×7	1764	1	700	700
49	7×7	2401	1	800	800
32	8×8	2048	1	1000	1000
48	8×8	3072	1	1200	1200
64	8×8	4096	1	1300	1300

4.2.1 部品方向重複解決処理

部品方向重複解決処理とは、スピン行列の横方向に+1のスピンの数が二つ以上存在した時、+1のスピンの一つに決定する処理である。スピン行列の横方向に+1のスピンの数が二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小となるスピンの組合せのみを+1に固定し、それ以外のスピンを0に変化させる。部品方向重複解決処理の例を図 6(a) に示す。

4.2.2 スロット方向重複解決処理

スロット方向重複解決処理とは、スピン行列の縦方向に+1のスピンの数が二つ以上存在した時、+1のスピンの一つに決定する処理である。スピン行列の縦方向に+1のスピンの数が二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小となるスピンの組合せのみを+1に固定し、それ以外のスピンを0に変化させる。スロット方向重複解決処理の例を図 6(b) に示す。

4.2.3 スロット方向追加処理

スロット方向追加処理とは、スピン行列の縦方向に関してスピンの数が全て0の列が存在した時、一つだけ+1のスピンを追加する処理である。スロット行列の縦方向に関してスピンの数が全て0の列が存在する場合、+1が配置可能な場所の中で重み付き総配線長が最小となるスピンの一つだけ+1に変化させる。スロット方向追加処理の例を図 6(c) に示す。

5. 評価実験

本章では、3章で示した手法によって QUBO モデルにマッピングしたスロット配置問題に対して初期処理を適用し、イジング計算機を用いて解を取得する。更に、取得した解に対して解釈処理を適用し、初期処理の適用の有無による結果を比較し考察する。

表 2: スロット配置問題の重み付き総配線長に関する実験結果.

m	t	スピンの数	初期処理なし		対交換法		ランダム交換法	
			最小値	平均値	最小値	平均値	最小値	平均値
8	4 × 4	128	199.94	215.75	199.64 (-0.15%)	215.04 (-0.32%)	199.64 (-0.15%)	215.61 (-0.062%)
12	4 × 4	192	574.48	594.71	572.98 (-0.26%)	583.53 (-1.7%)	573.14 (-0.23%)	584.85 (-1.7%)
16	4 × 4	256	1218.0	1260.2	1206.9 (-0.91%)	1213.3 (-3.7%)	1208.0 (-0.82%)	1217.1 (-3.4%)
12	5 × 5	300	573.51	595.71	573.00 (-0.088%)	587.55 (-1.4%)	572.22 (-0.22%)	587.50 (-1.4%)
18	5 × 5	450	1683.3	1746.3	1658.2 (-1.5%)	1668.2 (-4.5%)	1661.1 (-1.3%)	1671.9 (-4.3%)
25	5 × 5	625	3997.9	4103.8	3884.9 (-2.8%)	3887.0 (-5.3%)	3873.3 (-3.1%)	3875.1 (-5.6%)
18	6 × 6	648	1669.9	1731.8	1655.9 (-0.84%)	1669.8 (-3.6%)	1655.7 (-0.85%)	1667.4 (-3.7%)
27	6 × 6	972	4847.9	5044.8	4713.4 (-2.8%)	4719.1 (-6.5%)	4711.3 (-2.8%)	4717.8 (-6.5%)
36	6 × 6	1296	10432	10712	9997.7 (-4.2%)	10047 (-6.2%)	9986.7 (-4.3%)	10035 (-6.3%)
24	7 × 7	1176	3420.9	3698.0	3381.1 (-1.2%)	3419.0 (-7.5%)	3368.9 (-1.5%)	3398.6 (-8.1%)
36	7 × 7	1764	10099	10640	9673.7 (-4.2%)	9675.8 (-9.0%)	9659.6 (-4.4%)	9665.8 (-9.2%)
49	7 × 7	2401	23254	23755	22147 (-4.8%)	22230 (-6.4%)	22098 (-5.0%)	22178 (-6.6%)
32	8 × 8	2048	7374.6	8130.1	7255.7 (-1.6%)	7286.7 (-10%)	7209.3 (-2.2%)	7240.5 (-11%)
48	8 × 8	3072	21640	22789	20683 (-4.4%)	20683 (-9.2%)	20647 (-4.6%)	20651 (-9.4%)
64	8 × 8	4096	45834	46600	43674 (-4.7%)	43674 (-6.3%)	43777 (-4.5%)	43779 (-6.1%)

表 3: スロット配置問題の実行時間に関する実験結果.

m	t	スピンの数	初期処理の平均実行時間 [s]			DA の平均実行時間 [s]			解釈処理の平均実行時間 [10^{-2} s]		
			初期処理なし	対交換法	ランダム交換法	初期処理なし	対交換法	ランダム交換法	初期処理なし	対交換法	ランダム交換法
8	4 × 4	128	0	0.098873	0.78755	11.299	12.706	10.985	0.31547	0.31349	0.313698
12	4 × 4	192	0	0.20243	1.4542	11.291	12.150	10.685	0.023392	0.012835	0.011022
16	4 × 4	256	0	0.34823	2.3628	11.145	12.524	10.740	0.086644	0.010652	0.015337
12	5 × 5	300	0	0.43087	1.6485	11.083	10.965	10.895	0.15840	0.12951	0.134569
18	5 × 5	450	0	1.4339	3.1476	10.876	10.507	10.664	0.33178	0.034869	0.042381
25	5 × 5	625	0	3.2167	5.5576	10.734	10.370	10.554	0.31736	8.1530×10^{-3}	6.3720×10^{-3}
18	6 × 6	648	0	3.0049	3.4772	10.688	10.444	10.458	0.28681	0.040451	0.037603
27	6 × 6	972	0	10.335	6.8420	10.362	10.243	10.266	3.4712	0.054976	0.068643
36	6 × 6	1296	0	22.402	11.464	21.870	21.443	21.466	6.9117	0.30097	0.3018
24	7 × 7	1176	0	14.626	6.2603	24.577	24.396	24.538	6.0629	0.73456	0.61938
36	7 × 7	1764	0	51.060	12.481	24.041	23.502	24.104	17.804	0	0
49	7 × 7	2401	0	122.09	21.604	72.300	71.115	70.817	38.183	0.64612	0.62209
32	8 × 8	2048	0	61.283	10.570	23.555	24.080	24.183	25.878	1.0285	1.07439
48	8 × 8	3072	0	217.94	21.878	71.439	69.870	69.998	61.003	0	0
64	8 × 8	4096	0	495.07	36.374	69.574	68.562	68.902	82.321	2.0600×10^{-3}	2.0600×10^{-3}

5.1 実験内容

本実験は、富士通が提供するデジタルアニーラ (DA) Unit2, Parallel Tempering (PT) モードを使用し、初期処理を適用しなかった場合、対交換法を適用した場合、ランダム交換法 (イテレーション数 $N = 10,000$ とした) を適用した場合の三通りで実験し、初期処理の有無にかかわらず解釈処理を適用した。DA の PT モードでは、一つの問題に対して 128 個の解を出力する。128 個の解のうちスロット配置制約を満たしていないものについて解釈処理を適用し、その後、スロット配置制約を満たした 128 個のうち、式 (12) で表されるエネルギーが最も小さいものをそのときの解とする。DA のイテレーション数は 1,000,000 回とする。

5.2 実験環境

部品数を m とする。部品は任意の i, j ($1 \leq i, j \leq m$) について $w(c_i, c_j) \in [0, 10]$ の範囲の配線数を持つように設定する。スロットが 4×4 から 6×6 では、一つのスピン規模に対して 100 通りの問題をランダムに入力し、スロットが $7 \times 7, 8 \times 8$ では一つのスピン規模に対して 10 通りの問題をランダムに入力した。 p 行 p 列 ($4 \leq p \leq 8$) のスロットに対し、部品数 $m = \lfloor t/2 \rfloor, \lfloor 3t/4 \rfloor, \lfloor t \rfloor$ ($t = p \times p$) とした。問題の最大規模は、スロットが 8 行 8 列で部品数が 64 個、スピンの数が 4096 である。初期処理と DA の使用言語は Python、解釈処理の使用言語は C である。初期処理と解釈処理を実行した計算機環境は、OS は CentOS Linux 7.7.1908, CPU は Intel Xeon Gold 6148 CPU プロセッサである。式 (12) のハイパーパラメータ α, β, γ は、文献 [12] を参考に

表 4: 解釈処理の実行回数に関する実験結果.

m	t	スピン 数	解釈処理の平均実行回数		
			初期処理なし	対交換法	ランダム交換法
8	4 × 4	128	70.53	69.88	70.06
12	4 × 4	192	8.590	5.280	4.020
16	4 × 4	256	33.98	4.290	6.010
12	5 × 5	300	52.07	42.20	43.57
18	5 × 5	450	32.85	3.690	4.430
25	5 × 5	625	59.05	2.200	1.740
18	6 × 6	648	20.00	8.660	8.070
27	6 × 6	972	74.57	2.530	3.230
36	6 × 6	1296	128.0	36.30	36.97
24	7 × 7	1176	114.8	64.70	52.10
36	7 × 7	1764	19.60	0.000	0.000
49	7 × 7	2401	128.0	39.70	40.10
32	8 × 8	2048	118.4	52.60	50.70
48	8 × 8	3072	33.80	0.000	0.000
64	8 × 8	4096	123.9	0.100	0.100

表 1 のように設定した.

5.3 実験結果

重み付き総配線長に関する実験結果を表 2 に示す. 表 2 中の最小値, 平均値はスロットが 4×4 から 6×6 では一つのスピン規模に対して 100 通り, スロットが 7×7 , 8×8 では一つのスピン規模に対して 10 通りの問題における計算機実験から得られた重み付き総配線長 L の最小値と平均値を表す. また, 対交換法とランダム交換法の項目に記載されているカッコ中の数字は初期処理なしと比較した場合の差分の割合を表し, カッコ中の数字が小さいほど重み付き総配線長が削減されていることを意味する. 表 2 より, 初期処理なしと比較して, 最小値は対交換法でおよそ 2.29%, ランダム交換法でおよそ 2.40% 改善することができた. また, 平均値は対交換法でおよそ 5.47%, ランダム交換法でおよそ 5.54% 改善することができた.

実行時間に関する実験結果を表 3 に示す. 表 3 より, 対交換法およびランダム交換法ともに全てのスピン規模において, 初期処理なしと比較して解釈処理の平均実行時間が減少したことが確認できる.

解釈処理の実行回数に関する実験結果を表 4 に示す. 表 4 より, 対交換法およびランダム交換法の全てのスピン規模において, 初期処理なしと比較して解釈処理の平均実行回数が減少したことが確認できる.

以上, 表 2 から表 4 より, 初期処理の適用は有効であると考えられる. なお, 初期処理を適用しなかった場合と適用した場合の両方で, 全ての問題でスロット配置制約を満たすことができた.

6. おわりに

本稿では, スロット配置問題を対象に, イジング計算機で計算する前に適用する初期処理を提案した. 計算機実験

の結果, 初期処理なしの場合と比較して重み付き総配線長の最小値を対交換法ではおよそ 2.29%, ランダム交換法ではおよそ 2.40% 削減することができた. また, 重み付き総配線長の平均値は対交換法ではおよそ 5.47%, ランダム交換法ではおよそ 5.54% 削減することができた.

今後の課題として, 更に大きなスピン規模で実験すること, ハイパーパラメータを探索することが挙げられる.

参考文献

- [1] E. Ising, "Beitrag zur Theorie des Ferromagnetismus," *Zeitschrift für Physik*, vol. 31, no. 1, pp. 253–258, Feb. 1925.
- [2] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, "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, 2015.
- [3] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahan, T. Umeki, K. Enbutsu *et al.*, "A coherent Ising machine for 2000-node optimization problems," *Science*, vol. 354, no. 6312, pp. 603–606, 2016.
- [4] A. Marandi, Z. Wang, K. Takata, R. L. Byer, and Y. Yamamoto, "Network of time-multiplexed optical parametric oscillators as a coherent Ising machine," *Nature Photonics*, vol. 8, no. 12, pp. 937–942, 2014.
- [5] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, "Evidence for quantum annealing with more than one hundred qubits," *Nature physics*, vol. 10, no. 3, pp. 218–224, 2014.
- [6] R. Hamerly, T. Inagaki, P. L. McMahan, D. Venturelli, A. Marandi, T. Onodera, E. Ng, C. Langrock, K. Inaba, T. Honjo *et al.*, "Experimental investigation of performance differences between coherent Ising machines and a quantum annealer," *Science advances*, vol. 5, no. 5, p. eaau0823, 2019.
- [7] A. Lucas, "Ising formulations of many np problems," *Frontiers in Physics*, vol. 2, pp. 1–15, 2014.
- [8] K. Terada, D. Oku, S. Kanamaru, S. Tanaka, M. Hayashi, M. Yamaoka, M. Yanagisawa, and N. Togawa, "An Ising model mapping to solve rectangle packing problem," in *Proceedings of 2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pp. 1–4, 2018.
- [9] S. Tanaka, R. Tamura, and B. K. Chakrabarti, *Quantum spin glasses, annealing and computation*. Cambridge University Press, 2017.
- [10] H. Ushijima-Mwesigwa, C. F. Negre, and S. M. Mniszewski, "Graph partitioning using quantum annealing on the D-Wave system," in *Proceedings of the Second International Workshop on Post Moores Era Supercomputing*, pp. 22–29, 2017.
- [11] N. Yoshimura, M. Tawada, S. Tanaka, J. Arai, S. Yagi, H. Uchiyama, and N. Togawa, "Efficient Ising model mapping for induced subgraph isomorphism problems using Ising machines," in *Proceedings of 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*, pp. 227–232, 2019.
- [12] S. Kanamaru, K. Kawamura, S. Tanaka, Y. Tomita, H. Matsuoka, K. Kawamura, and N. Togawa, "Mapping constrained slot-placement problems to Ising models and

- its evaluations by an Ising machine,” in *Proceedings of 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*, pp. 221–226, 2019.
- [13] M. Hanan and J. M. Kurtzberg, “A review of the placement and quadratic assignment problems,” *Siam Review*, vol. 14, no. 2, pp. 324–342, 1972.
- [14] S. Kanamaru, D. Oku, M. Tawada, S. Tanaka, M. Hayashi, M. Yamaoka, M. Yanagisawa, and N. Togawa, “Efficient Ising model mapping to solving slot placement problem,” in *Proceedings of 2019 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–6, 2019.
- [15] M. Rajeswari, S. Jaiganesh, P. Sujatha, T. Vengattaraman, and P. Dhavachelvan, “A study and scrutiny of diverse optimization algorithm to solve multi-objective quadratic assignment problem,” in *2016 International Conference on Communication and Electronics Systems (ICCES)*, pp. 1–5. IEEE, 2016.
- [16] P. J. Van Laarhoven and E. H. Aarts, “Simulated annealing,” in *Simulated annealing: Theory and applications*. Springer, 1987, pp. 7–15.
- [17] E. L. Lawler and D. E. Wood, “Branch-and-bound methods: A survey,” *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.
- [18] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, and H. G. Katzgraber, “Physics-inspired optimization for quadratic unconstrained problems using a digital annealer,” *Frontiers in Physics*, vol. 7, p. 48, 2019.
- [19] M. Sao, H. Watanabe, Y. Musha, and A. Utsunomiya, “Application of digital annealer for faster combinatorial optimization,” *Fujitsu Scientific and Technical Journal*, vol. 55, no. 2, pp. 45–51, 2019.
- [20] S. Morita and H. Nishimori, “Convergence theorems for quantum annealing,” *Journal of Physics A: Mathematical and General*, vol. 39, no. 45, pp. 13 903–13 920, oct 2006.