

# イジングマシンによる動的周波数割当問題の評価

齋藤和広<sup>1,a)</sup> 伊神皓生<sup>1</sup> 小野智弘<sup>1</sup>

**概要:** 限られた周波数リソースを効率的に有効活用するため、既存の無線システムで利用されている周波数を通信事業者に割り当てる動的周波数割当 (DSA: Dynamic Spectrum Allocation) が検討されている。DSA として、同じ共用周波数を複数の通信事業者で利用することを想定し、電波干渉を考慮して共用周波数を動的に割り当てる問題を考える。この問題設定は組合せ最適化問題であり、対象となる基地局数、時間軸、周波数チャンネル数に依存して計算量が爆発的に増加する。イジングマシンは、このような組合せ最適化問題の近似解を高速に解くことのできる組合せ最適化問題専用の計算機である。そこで本稿ではイジングマシンによって DSA を高速に解くために、DSA を QUBO (Quadratic Unconstrained Binary Optimization) に定式化した。本評価では、イジングマシンの一種として GPU で高速処理可能なモメンタムアニーリングを用いて、DSA の QUBO 式を評価した。

**キーワード:** イジングマシン, 組合せ最適化問題, 周波数割当問題

## An Evaluation of Dynamic Spectrum Allocation using Ising Machine

KAZUHIRO SAITO<sup>†1</sup> AKIO IKAMI<sup>†1</sup> CHIHIRO ONO<sup>†1</sup>

**Abstract:** Dynamic spectrum allocation (DSA) between different radio systems is a promising method to efficiently use limited frequency resources. The approach of DSA is to allow different mobile network operators to use the same shared channel, without causing interference across base stations. However, existing methods to solve DSA as combinational optimization problem are computationally very expensive in scaling the number of base stations, the number of frequency channels, and time. Ising machine is a fast solver dedicated to solve an approximate solution of combinational optimization problem. In this paper, we formulate the DSA problem as QUBO (Quadratic Unconstrained Binary Optimization) problem to solve it by ising machine. Our evaluation demonstrates that momentum annealing, which can solve a large QUBO problem by GPU as one of ising machine, can obtain high accuracy solutions of the DSA problem in short time comparing to existing method.

**Keywords:** Ising machine, Combinational optimization problem, Dynamic spectrum allocation.

### 1. はじめに

限られた周波数リソースを効率的に有効活用するため、既存の無線システムで利用されている周波数を通信事業者に割り当てる動的周波数割当 (DSA: Dynamic Spectrum Allocation) が検討されている。DSA では、既存の無線システムで時間軸、空間軸、周波数軸から利用されていない共用周波数 (USC: Unused Sharable Channel) を移動通信システムに割り当てて再利用する。異なる無線システム間の周波数管理システムとして、米国の CBRS (the Citizens Broad Radio Service) [1] や欧州の LSA (Licensed Shared Access) [2] などが社会実装に向けて実験を行っている。日本でも同様に政府が 2.3GHz, 26GHz, 及び 38GHz の帯域で DSA を検討している[3]。

近年、電波の周波数効率を高めるために通信事業者 (MNO: Mobile Network Operator) の需要に基づく DSA 手法が提案されている[4][5]。しかしこれらの既存手法は USC を MNO 単位で割り当てているため、ある MNO がチャンネルを利用していない空間が存在しても、他の MNO はそのチャンネルをその空間で利用することができない。従って、

未使用の周波数リソースの空間的な無駄が発生する。伊神ら[6]は基地局間の干渉が一定以下の場合に、その基地局間で同じチャンネルを利用できる手法を提案し、周波数効率を向上させた。しかしこの手法では、基地局数が大きくなることで候補となる組合せ数が膨大となり、最適化の計算が困難となる。

従来のコンピュータでは計算が困難な組合せ最適化問題に、イジングマシンの活用が期待されている。イジングマシンは、イジングモデルの基底状態を探索することに特化した計算機で、組合せ最適化問題のメタヒューリスティックな手法の実装として利用できる。代表的なイジングマシンとして、量子効果を利用してイジングモデルの基底状態を探索する量子アニーリング[7]を、横磁場イジングモデルで実装したハードウェア[8]がある。イジングモデルは、二つのスピン  $\sigma_i, \sigma_j \in \{-1, +1\}$  の相互作用係数  $J_{ij}$ 、及びスピン  $\sigma_i$  に影響する外部磁場エネルギー係数  $h_i$  から、以下のように表現される。

$$\mathcal{H} = \sum_{i,j} J_{ij} \sigma_i \sigma_j + \sum_i h_i \sigma_i \quad (1)$$

イジングマシンは、この基底状態としてハミルトニアン  $\mathcal{H}$

<sup>1</sup> 株式会社 KDDI 総合研究所  
KDDI Research, Inc., Fujimino, Saitama 356-8502, Japan  
a) ku-saitou@kddi-research.jp

を最小化するスピンを探索する。したがって、組合せ最適化問題を最小化問題としてこのイジングモデルに定式化することで、イジングマシンによって良い解を高速に探索することができる。定式化を直感的に行うため、以下のようにスピン $\sigma_i$ をバイナリ変数 $x_i \in \{0,1\}$ に変換する。

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

これを数式(1)のイジングモデルに適用して以下のような QUBO (Quadratic Unconstrained Binary Optimization) 問題とする[9][10]。

$$\mathcal{H} = \sum_{i,j} Q_{ij} x_i x_j \quad (3)$$

本稿では、イジングマシンで DSA を解くために、DSA を組合せ最適化問題として QUBO 問題に定式化した。イジングマシンにおける DSA の QUBO 式を評価するために、イジングモデルを GPU で高速に処理可能な日立 CMOS アニールリングのモメンタムアニールリング[11]を利用し、既存手法と比較する。

## 2. DSA の定式化

### 2.1 問題設定

本研究で対象とする問題は、複数の異なる無線システムを対象とした DSA である。これは三つのシステム：既存無線システム、移動通信システム、および周波数管理システムから構成される。既存無線システムは、例えば衛星通信システムや固定無線アクセスシステムなど、事前に許可されたバンド帯が割り当てられていて、その周波数の優先利用権がある。移動通信システムは MNO によって構築され、MNO はあらかじめ定められた時間スロット単位で基地局ごとに必要なチャンネル数を周波数管理システムに要求する。周波数管理システムは、既存無線システムの位置と利用可能な USC を検出し、MNO の要求に対して USC を割り当てる。この USC は、周波数管理システムに事前登録した既存無線システムの周波数利用情報を活用して事前に検出される。ここで USC を割り当てられる基地局の伝送電力は、CBRS システム[1]を参考に既存無線システムと干渉しない値を事前計算する。更に、割当対象となる全ての基地局間の干渉を事前計算し、干渉が発生する基地局同士で同じ時間に同じチャンネルを割り当てない。

本研究における DSA は、上記周波数管理システムが解く組合せ最適化問題として、以下の4つの目的を最適化する。

- MNO の要求に対する割当ての最大化
- 基地局内の時間連続性の最大化
- 基地局内の周波数連続性の最大化
- 空間連続性の最小化

したがって、MNO の要求を満たすように、同基地局内で時間的及び周波数的に連続するようチャンネルを割り当てるが、干渉を防ぐため空間的に隣接した基地局に対しては同じ時

間に同じ周波数を割り当てないようにする。ただし、制約として、干渉関係にある基地局に対して同じ時間に同じチャンネルを割り当てない。

図 1 はこの DSA の問題設定例である。図 1(a)は3つの MNO (事業者 A, 事業者 B, 事業者 C) がそれぞれ保持する計 5 つの基地局に対する USC 要求と、異なる事業者間の基地局間の干渉、および同一事業者間の基地局の空間的隣接を表す。例えば事業者 A の基地局 1 と事業者 B の基地局 1 と事業者 C の基地局 1 は互いに干渉関係にあるため、その三局に対して同じ時刻に同じチャンネルを割り当ててはいけない。また事業者 C の基地局 1 と基地局 2 は隣接関係にあり、同一事業者内での干渉も可能な限り避けるため、同じ時刻に同じチャンネルをできる限り割り当てない。図 1(b)はこの要求に対する DSA の最適化結果例である。

### 2.2 数理モデル

イジングマシン適用に向けて QUBO 式を構築するために、DSA の目的変数はバイナリ変数をとる。ある基地局  $n \in N$  において、あるタイムスロット  $t \in T$  にチャンネル  $f \in F$  の割り当て有無を目的変数として以下で定義する。

$$x_{n,f,t} = \begin{cases} 1 & \text{チャンネル } f \text{ を基地局 } n \text{ に} \\ & \text{タイムスロット } t \text{ で割り当て。} \\ 0 & \text{それ以外} \end{cases} \quad (4)$$

ここで  $N$  は基地局の集合、 $F$  は利用可能な USC の集合、 $T$  は時間スロットの集合を表す。

次に、DSA の目的関数を定式化する。MNO は、基地局  $n \in N$  に対して、時間スロット  $t \in T$  における USC の利用数  $f d_{n,t}$  を要求する。一つ目の目的関数は、この MNO の要求に対する割り当ての割合の最大化である。ここで割り当ての数が要求数を超えないように、その割合を反転させて二乗した値を最小化する。したがって、この目的関数は目的変数  $x_{n,f,t}$  の集合  $X$  に対する最小化として以下で定式化される。

$$\operatorname{argmin}_X \sum_{n \in N} \sum_{t \in T} \left( 1 - \frac{\sum_{f \in F} x_{n,f,t}}{f d_{n,t}} \right)^2. \quad (5)$$

加えて、同基地局内で割り当てるチャンネルの時間連続性の最大化を以下とする。

$$\operatorname{argmax}_X \sum_{n \in N} \sum_{f \in F} \sum_{t \in T} (x_{n,f,t} \cdot x_{n,f,t+1}). \quad (6)$$

同様に、同基地局内で割り当てるチャンネルの周波数連続性の最大化を以下とする。

$$\operatorname{argmax}_X \sum_{n \in N} \sum_{f \in F} \sum_{t \in T} (x_{n,f,t} \cdot x_{n,f+1,t}). \quad (7)$$

最後の目的である空間連続性の最小化を定式化するために、隣接する基地局  $r, s \in N$  の組の集合  $B_n$  を以下で定義する。

$$B_n = \{(r, s), \dots\}. \quad (8)$$

この集合を用いて、基地局間における同一チャンネルの空間連続性の最小化を以下とする。

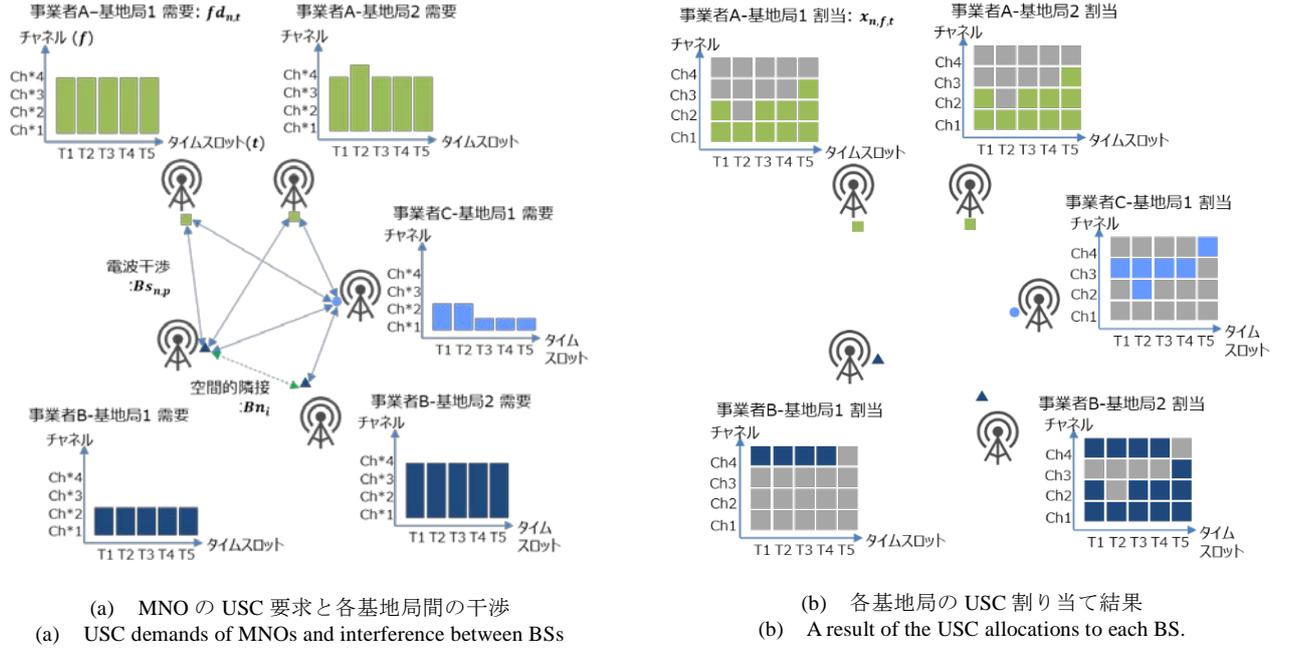


図 1 DSA の問題設定例

Figure 1 An example of our DSA problem

$$\operatorname{argmin}_X \sum_{(r,s) \in Bn} \sum_{f \in F} \sum_{t \in T} (x_{r,f,t} \cdot x_{s,f,t}). \quad (9)$$

制約条件として、干渉関係にある基地局に対して同じチャンネルを同時刻に割り当てない制約を考える。干渉は、事前に計算された特定の複数基地局で同時刻に同じチャンネルを利用することで発生する。この基地局の干渉関係の集合を  $P$  とし、ある基地局間の干渉  $p \in P$  において干渉原因となる基地局  $n \in N$  を以下で定義する。

$$BS_{n,p} = \begin{cases} 1 & \text{基地局 } n \text{ が干渉 } p \text{ の原因となる} \\ 0 & \text{それ以外} \end{cases}. \quad (10)$$

ある干渉  $p$  において、 $BS_{n,p} = 1$  となる全ての基地局に同じチャンネルを同時刻に割り当てることで、干渉が発生する。したがって、あるチャンネル・時刻において、干渉原因となる基地局に割り当てた数を、干渉原因となる基地局数より少なくすることで干渉を回避できる。この干渉制約は以下で定式化できる。

$$\sum_{n \in N} (BS_{n,p} \cdot x_{n,f,t}) \leq \sum_{n \in N} (BS_{n,p}) - 1, \quad \forall p \in P, \forall f \in F, \forall t \in T \quad (11)$$

以上のように、DSA は数式(11)を制約として、四つの目的関数(5), (6), (7), (9)を対象とする多目的最適化問題となる。

### 2.3 QUBO

DSA 問題をイジングマシンで実行できる形式にするため、QUBO 式を構築する。2.2 節で定義した目的関数及び制約からハミルトニアン  $\mathcal{H}$  を以下のように定義する。

$$\mathcal{H} = \alpha_1 \mathcal{H}_{obj1} + \alpha_2 \mathcal{H}_{obj2} + \alpha_3 \mathcal{H}_{obj3} + \alpha_4 \mathcal{H}_{obj4} + \lambda \mathcal{H}_{st}, \quad (12)$$

$$\mathcal{H}_{obj1} = \sum_{n \in N} \sum_{t \in T} \left( 1 - \frac{\sum_{f \in F} x_{n,f,t}}{fd_{n,t}} \right)^2, \quad (13)$$

$$\mathcal{H}_{obj2} = - \sum_{n \in N} \sum_{f \in F} \sum_{t \in T} (x_{n,f,t} \cdot x_{n,f,t+1}), \quad (14)$$

$$\mathcal{H}_{obj3} = - \sum_{n \in N} \sum_{f \in F} \sum_{t \in T} (x_{n,f,t} \cdot x_{n,f+1,t}), \quad (15)$$

$$\mathcal{H}_{obj4} = \sum_{(r,s) \in Bn} \sum_{f \in F} \sum_{t \in T} (x_{r,f,t} \cdot x_{s,f,t}), \quad (16)$$

$$\mathcal{H}_{st} = \sum_{p \in P} \sum_{f \in F} \sum_{t \in T} \begin{cases} f_{bin}(p, f, t), & \text{if } \sum_{n \in N} (BS_{n,p}) = 2 \\ f_{mt}(p, f, t), & \text{otherwise} \end{cases}. \quad (17)$$

数式(12)が組合せ最適化問題として DSA を QUBO 問題に定式化した式である。なおパラメータ  $\alpha_1, \alpha_2, \alpha_3, \alpha_4$  は目的関数の重み係数であり、パラメータ  $\lambda$  は制約違反を回避するためのペナルティ係数である。数式(13), (14), (15), (16) は、それぞれ目的関数である数式(5), (6), (7), (9)を QUBO 式に変換している。イジングマシンは  $\mathcal{H}$  を最小化するが、目的関数の数式(6)と(7)は最大化が目的のため、数式(14)と(15)のように負にすることで最小化問題とした。数式(17)は数式(11)の干渉制約を表し、関数  $f_{bin}(p, f, t)$  と  $f_{mt}(p, f, t)$  は、それぞれ以下のように定義される。

$$f_{bin}(p, f, t) = \left( \sum_{n \in N} (BS_{n,p} \cdot x_{n,f,t}) \right) \left( \sum_{n \in N} (BS_{n,p} \cdot x_{n,f,t}) - 1 \right), \quad (18)$$

$$f_{mt}(p, f, t) = \left( \sum_{n \in N} (BS_{n,p} \cdot x_{n,f,t}) - \sum_{j=0}^{\sum_{n \in N} (BS_{n,p}) - 2} y_{j,p,f,t} \right)^2. \quad (19)$$

なお,  $y_{j,p,f,t}$  はスラック変数で, 一般的な最適化問題を QUBO 問題に変換する際に, 不等式の制約条件式を等式へ変換するために用いられる補助的な目的変数である. 干渉制約の数式(11)は不等式による制約条件のため, QUBO 式に変換すると関数  $f_{mit}(p, f, t)$  のようにスラック変数が必要となる. そのため, スラック変数の削減によって, 求解性能を向上することができるとともに, ハードウェア的に利用可能なスピン数に限りがあるイジングマシンにおいて実行可能な問題の規模を大きくすることができる. そこで, 2 つの基地局間でのみ発生している干渉を数式(18)の関数  $f_{bin}(p, f, t)$  で処理することで, スラック変数を削減する.

### 3. 評価

#### 3.1 評価内容

本評価では, 2 節で構築した QUBO 式による DSA の性能をイジングマシンで評価する. ここでは, 性能の評価軸として, 処理速度と精度を考える. 処理速度は一つの解を得るまでの処理時間とし, 精度は実行可能解が得られた確率とハミルトニアン値 (以下, コスト値) とする. イジングマシンにおける QUBO 式の処理性能は QUBO 式におけるパラメータに依存するため, まずパラメータ評価としてパラメータの値の変化における精度を評価し, パラメータを決定する. そして, 性能評価として実行時間と精度を評価する. これらのパラメータ評価と性能評価を, DSA の問題規模を基地局数とタイムスロット数でそれぞれスケールリングさせて行う.

#### 3.2 評価環境

評価で用いたイジングマシンは, 日立 CMOS アニーリングのモメンタムアニーリング[11] (MA) である. MA は, イジングモデルにおける全結合のスピン間結合を完全二部グラフ状の結合に変換し, 変数更新を GPU で並列処理することで基底状態の探索を高速化している. MA の評価で用いた環境は Xeon Gold 6240 (2.6GHz, 18 cores), NVIDIA Tesla V100 32GB, Ubuntu 18.04.4 LTS である. なお本評価におけるすべての実行結果はアニーリング処理回数 (num\_reads) を 1000 回に設定した. イジングマシンの比較対象は, 伊神らによる DSA の既存手法[6]を利用し, さらに空間連続性の目的関数を追加した. この既存手法は, DSA を 0-1 混合整数線形計画問題として定式化し, タイムウインドウごとに最適化することで高速化している. 本評価ではタイムウインドウを 2 タイムスロットに設定して最適化を実行した. 例えば 10 タイムスロットのデータの場合, 最適化を 5 回実行した. 本評価では既存手法を汎用ソルバーの Gurobi ver9.0.1[14]で実行した. 利用した環境は Xeon Gold 6136 (3.00GHz, 12 cores) x 2 CPU, Windows 10 Pro for Workstations である. イジングマシンと精度を比較するために, 既存手法で出力した解を数式(12)で再計算してコスト値に変換する. なお, 今回評価した DSA の規模にお

表 1 DSA のシミュレーション条件

Table 1 Simulation assumptions of DSA.

周波数帯[1]	3.6 GHz
USC[1]	150 MHz
チャンネル割当単位[1]	10 MHz
タイムスロット[1]	5 分
基地局位置	Random (1 km x 1 km, 100 局は 4km x 4km)
基地局送信電力[1]	20 dBm
アンテナパターン	Omni
パスロス[12]	Uma model (Line of site rate: 50%)
許容干渉量[12]	100dBm/MHz
バーストラフィックによる標準偏差	10MHz
要求するチャンネル $fd_{n,t}$ の設定	London Internet eXchange Point traffic [13]
MNO の平均要求割合	39:28:23:10 (日本の MVNO と MNO の割合)

る最適解は, すべてのデータにおいて現実時間で算出することができなかった.

評価データは, 表 1 で示す DSA の移動通信システムをシミュレーションして生成した. 生成した評価データの規模と, 評価時に利用する QUBO 式のパラメータの値の一覧を表 2 に示す. 基地局スケールリングのデータは基地局数を変化させ, タイムスロットスケールリングではタイムスロット数を変化させた. なお, 基地局数 20, タイムスロット 2 のデータは, 基地局スケールリングとタイムスロットスケールリングで同一である. 2 基地局間の干渉数は, すべての干渉数のうち,  $\sum_{n \in N} (Bs_{n,p}) = 2$  となる干渉  $p \in P$  の数を示している. この干渉は, 制約条件の数式(17)において関数  $f_{bin}(p, f, t)$  を利用するため, 全体の干渉数に対してこの 2 基地局間の干渉数が少ないほど, スラック変数が増加し, 変数の数が増加する. 制約条件のペナルティ係数であるパラメータ  $\lambda$  の決定過程は 3.3 節で述べる. なお本評価では, 目的関数の重み係数であるパラメータ  $\alpha_1, \alpha_2, \alpha_3, \alpha_4$  はすべて 1 で固定した.

#### 3.3 パラメータ評価

パラメータ  $\lambda$  は制約条件のペナルティ係数であり, 値を大きくすることで制約違反を減らし, 実行可能解の得られる確率を高めることができる. 一方でパラメータ  $\lambda$  を大きくすることで, コスト値の低い解に収束するまでの時間が長くなる. したがって, 性能評価で統計的に有意な数の解を得られ, かつイジングマシンによる QUBO の実行性能を高められるパラメータ  $\lambda$  の値を見つける必要がある. そこ

表 2 評価データの規模と QUBO 式のパラメータ設定  
Table 2 Scales of evaluation data and parameters of our QUBO

規模	基地局スケール				タイムスロットスケール			
基地局数 $ N $	16	20	24	100	20	20	20	20
チャンネル数 $ F $	15	15	15	15	15	15	15	15
タイムスロット数 $ T $	2	2	2	2	2	10	20	30
干渉数 $ P $	76	123	183	710	123	123	123	123
2 基地局間の干渉数	61	103	146	42	103	103	103	103
パラメータ $\alpha_1, \alpha_2, \alpha_3, \alpha_4$	1	1	1	1	1	1	1	1
パラメータ $\lambda$	0.7	0.8	1.2	20.0	1.6	1.6	2.2	3.0
変数の数	990	2,100	3,270	52,200	2,100	10,050	20,100	30,150

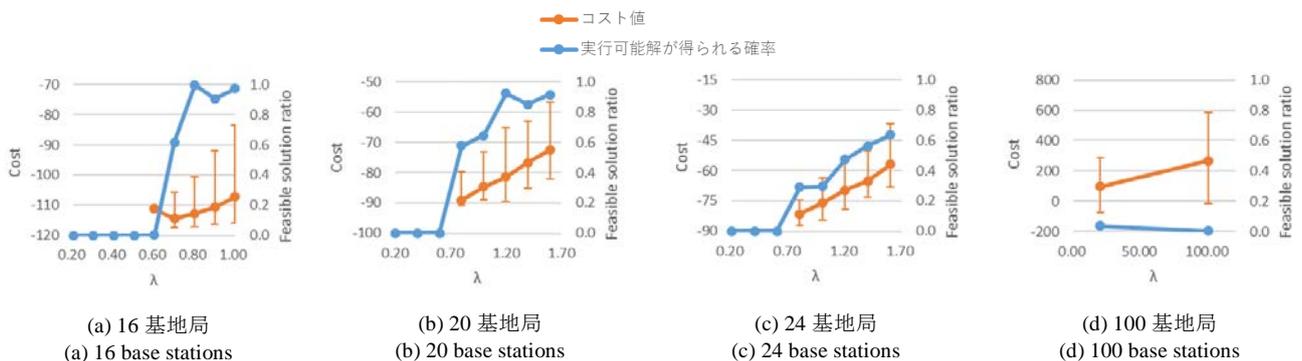


図 2 基地局スケールにおけるパラメータ変化の評価

Figure 2 Penalty parameter evaluation for each setting of scaling up the number of base stations.

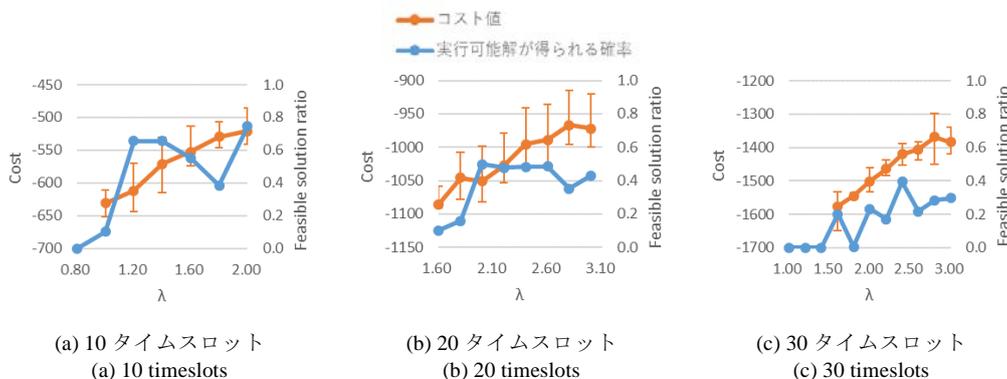


図 3 タイムスロットスケールにおけるパラメータ変化の評価

Figure 3 Penalty parameter evaluation for each setting of scaling up the number of timeslots.

で、性能評価で利用する DSA の QUBO 式のパラメータ $\lambda$ を決定するために、パラメータ $\lambda$ の値を変化させて QUBO 式の精度を評価する。

図 2 は基地局スケールの評価データにおいて、図 3 はタイムスロットスケールでの評価データにおいて、それぞれのデータでパラメータ $\lambda$ の値を変化させて DSA の QUBO 式を MA で実行した結果である。なお、タイムスロットスケールの 2 タイムスロットは、図 2(b)の 20 基地局と同じ評価データのため省略する。アニーリング処理回数 (num\_reads) はすべて 1,000 に設定し、1 回のアニー

リングにおける sweep 数は 1,000 とした。ただし基地局スケールの 100 基地局は解が得られなかったため sweep 数を 2,000,000 とした。それぞれの結果は、得られた 1000 個の解のうち実行可能解が得られた確率と、その実行可能解における平均コスト値を表す。コスト値のエラーバーは、その最大値と最小値を表す。

平均コスト値は、すべての結果において、パラメータ $\lambda$ に比例して大きくなり、その最大・最小値も広がる傾向にあることがわかる。実行可能解を得られた確率は、基本的にパラメータ $\lambda$ と比例しているが、DSA の規模が大きくなり

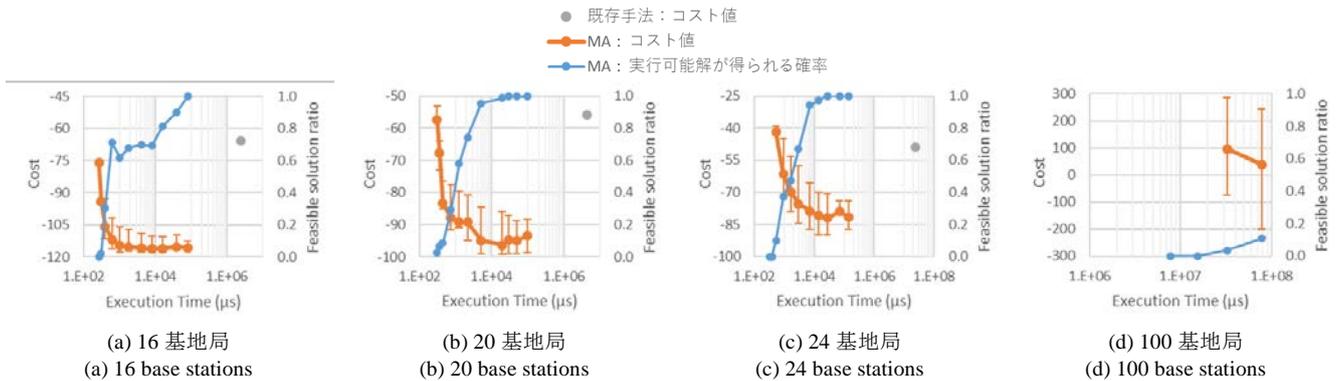


図 4 基地局スケールアップにおける性能評価

Figure 4 Performance evaluation for each setting of scaling up the number of base stations.

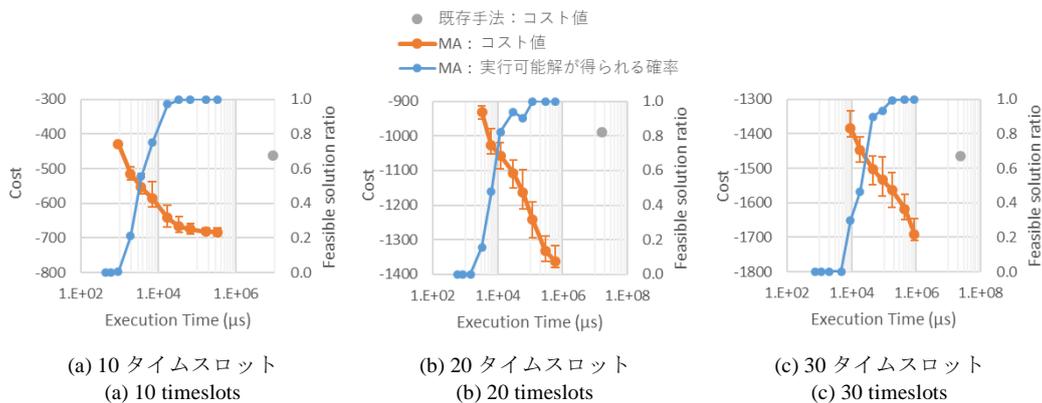


図 5 タイムスロットスケールアップにおける性能評価

Figure 5 Performance evaluation for each setting of scaling up the number of timeslots.

変数の数が多くなるにしたがって上昇率が鈍化している。特に変数の数が 10,000 を超える大きい規模の問題では、実行可能解を得られる確率が一定の値以上にならないことがわかる。そこで、実行可能解を得られた確率が 50%程度となるか、50%未満でもできる限り高い確率となるように、表 2 に示すパラメータ  $\lambda$  を選択した。

### 3.4 性能評価

イジングマシンにおける DSA の QUBO 式を評価するため、表 2 のデータとパラメータを用いて実行した MA と、既存手法を比較する。MA は、アニーリング処理回数 (num\_reads) をすべて 1,000 に設定し、sweep 数を 50 から 100,000 までの 11 通りで実行した。なお、いずれの sweep 数でも基地局スケールアップの 100 基地局は解が得られなかったため、100 基地局に限り、sweep 数を 500,000 から 5,000,000 までの 4 通りで実行した。図 4 が基地局スケールアップのデータを利用した実行結果で、図 5 がタイムスロットスケールアップのデータを利用した実行結果である。3.3 節と同様に、タイムスロットスケールアップの 2 タイムスロットは、図 4(b) の 20 基地局と同じ評価データのため省略する。MA の結果における実行時間は、1 回のアニーリング処理あたりの処理時間であり、既存手法の実行時間は一

つの解が得られるまでの処理時間である。100 局に関しては既存手法で実行することができなかった。

本評価結果から、MA を用いた QUBO 式によって、既存手法では解けなかった 100 基地局における DSA を最短 32 秒で解けることがわかる。一方、実行時間 32 秒において実行可能解が得られる確率は 3%、78 秒においては 11%と、実行可能解が得られる確率は低く、また得られた実行可能解のコスト値のバラつきが大きい。それ以外のデータでは、MA が全ての結果で既存手法より高速に実行可能解を得ている。またコスト値に関しても一定以上の実行時間で既存手法より改善し、実行可能解の得られる確率も 100%に達している。

MA を用いた QUBO 式をより詳細に分析するために、それぞれのデータにおいて、MA の実行結果で最小コスト値が最も小さくかつその中で計算時間の短い結果を既存手法と比較する。図 6 は基地局スケールアップのデータにおける MA と既存手法のコスト値と実行時間の比である。それぞれの基地局数で選択した結果の sweep 数は、16 基地局が 2,000、20 基地局が 5,000、24 基地局が 10,000 である。図 6(a)は、既存手法に対する MA の平均コスト値の比で、エラーバーが最大コスト値と最小コスト値それぞれの比であ

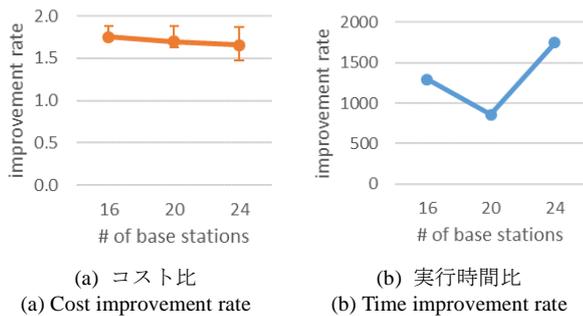


図 6 基地局スケールにおけるイジングマシン(MA)と既存手法の性能比

Figure 6 Performance comparison between Ising machine (MA) and existing method on base station scaling.

る。この結果から、MA の最小コスト値は、いずれの基地局数でも 1.8 倍コスト値が改善している。これは、四つの目的関数が平均して 1.8 倍改善していることを意味する。ただし平均コスト値で見ると、基地局数増加によって 0.05 ポイントずつ改善率が低下している。図 6(b)は、既存手法に対する MA の実行時間の比である。16 基地局で 1292 倍、20 局で 858 倍、24 局で 1748 倍と、既存手法と比較して大幅に高速化している。

同様にタイムスロットスケールにおける MA と既存手法のコスト値と実行時間の比を図 7 に示す。それぞれのタイムスロット数で選択した結果の sweep 数は、10 タイムスロット数が 50,000、20 タイムスロットが 100,000、30 タイムスロットが 100,000 である。図 7(a)の既存手法に対する MA の最小コスト値の比は、2 タイムスロットで 1.8 倍であったが、10 タイムスロットで 1.5 倍、20 タイムスロットで 1.4 倍、30 タイムスロットで 1.2 倍と改善しているが、タイムスロット数の増加とともにその改善率は低下している。図 7(b)の既存手法に対する MA の実行時間の比は、2 タイムスロットで 858 倍であった高速化が、それ以上のタイムスロット数で大幅に減少している。図 7(c)は 10 タイムスロット以上に焦点をあてた実行時間比である。10 タイムスロットで 53 倍、20 タイムスロットで 28 倍、30 タイムスロットで 27 倍高速化しているが、タイムスロット数の増加とともにその割合は減少している。

#### 4. 議論

3 章の評価から、QUBO 式で構築した DSA をイジングマシンで実行することで、既存手法より高速かつ高精度な解を導出でき、より大きい規模の問題が解けることを示した。しかしイジングマシンの実行時間の解釈には注意が必要である。イジングマシンは実行可能解を保証できないため、商用アプリケーションとしてイジングマシンを利用する場合は、実行時間として実行可能解を保証するための仕組みも考慮する必要がある。実行可能解を保証する方法として、

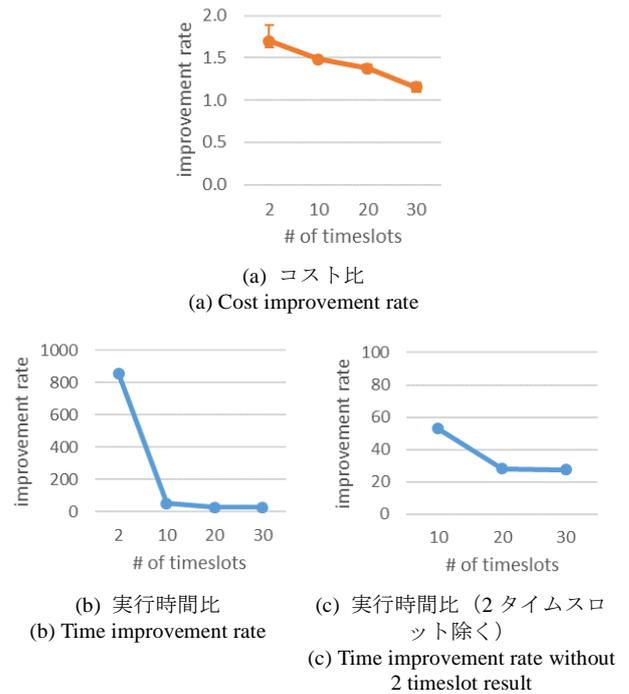


図 7 タイムスロットスケールにおけるイジングマシン(MA)と既存手法の性能比

Figure 7 Performance comparison between Ising machine (MA) and existing method on timeslot scaling.

実行可能解が得られるまで繰り返しイジングマシンを実行する仕組みをアプリケーションに取り入れる方法が考えられる。この方法は実行可能解を確実に保証できるが、イジングマシンの実行に伴い実行回数分のオーバーヘッド (API 呼び出しやアニーリングに関する初期化処理など) が発生する。また、一回のイジングマシン実行における実行可能解が得られる確率が低い場合、繰り返しの回数が実行ごとに変わり、アプリケーションの実行時間が不安定となる。この実行時間は、一回のイジングマシン実行におけるアニーリング処理回数を増やして実行可能解が得られる確率を高めることで安定させることができる。加えて、イジングマシン実行時のオーバーヘッドも削減することができる。したがって、アニーリング処理の回数を十分増やし、保険としてイジングマシンの繰り返し実行を備えることで、アプリケーション実行時間を安定させつつ実行可能解を保証することができる。商用アプリケーションでイジングマシンを利用するときの実行時間は、一回のアニーリング処理における実行時間に加えて、これらの処理を複合的に考える必要がある。

次に、アニーリング処理回数を増やした場合の実行可能解が得られる確率について考える。1 回のアニーリング処理で実行可能解が得られる確率  $r_1$  に対して、 $m$  回のアニーリング処理で実行可能解が得られる確率  $r_m$  は以下となる。

$$r_m = 1 - (1 - r_1)^m. \quad (20)$$

この数式(20)より、確率 $r_m$ で実行可能解を得るために必要なアニーリング処理の回数  $m$  は以下となる。

$$m = \left\lceil \frac{\log(1 - r_m)}{\log(1 - r_1)} \right\rceil. \quad (21)$$

例えば実行可能解が得られる確率が50%のアニーリング処理の場合、99%以上の確率で実行可能解を得るためにはアニーリング処理を7回実行する必要がある。100基地局のDSAをイジングマシンで実行した結果で、アニーリング処理時間32秒における実行可能解が得られる確率3%では、99%で実行可能解を得るために152回のアニーリング処理が必要であり、この実行時間が4864秒となる。一方、78秒における実行可能解が得られる確率11%の場合、99%で実行可能解を得るには40回のアニーリング処理が必要で、実行時間は3120秒となり、アニーリング処理時間32秒と比較して短い。このように、商用アプリケーションからイジングマシンを利用する場合、一回のアニーリング処理時間と実行可能解が得られる確率から、アニーリング処理を行う回数についても分析し、適切なアニーリング処理時間と処理回数を選択する必要がある。

## 5. おわりに

本稿では、DSAをQUBOに定式化し、イジングマシンのMAを用いて基地局数とタイムスロット数をそれぞれ変化させたDSAによる評価を行った。その結果、イジングマシンを用いることで従来では解けなかった15チャンネルを100基地局に2タイムスロットで割り当てるDSAを最短34秒で解けることがわかった。既存手法と比較できる規模のDSAにおいては、既存手法より大幅な性能改善を確認できた。一方で、タイムスロット数が大きくなることで、既存手法に対する性能向上は大きく減少することがわかった。今後、タイムスロットを分割してイジングマシンで最適化する手法について検討したい。また、本稿ではQUBOにおける目的関数の重み係数を固定とし、四つの目的関数の内訳について分析していないため、多目的最適化問題として目的関数の値を分析し、重みパラメータを最適化することで精度向上を目指したい。最後に、イジングマシンを商用アプリケーションで活用するために、適切なアニーリング処理回数に関する議論を行った。DSAにおけるイジングマシンの評価にて得られた結果からこのアニーリング処理回数について詳細を分析し、イジングマシンの商用利用における実行形態の決定方法について体系化していきたい。

**謝辞** 本研究は、総務省「異システム間の周波数共用技術の高度化に関する研究開発」(JPJ000254)における委託研究の一環として実施されたものである。本稿におけるモメタムアニーリングを用いた評価をサポートして頂いた株式会社日立製作所の関係者の皆様に深く感謝いたします。

## 参考文献

- [1] OFR, 47 CFR 96 CITIZENS BROADBAND RADIO SERVICE, Jun. 2020.
- [2] ETSI, TS 103 379, V1.1.1, Jan. 2017.
- [3] Ministry of Internal Affairs and Communications in Japan “Frequency Reorganization Action Plan (revised version for FY 2020),” [https://www.soumu.go.jp/main\\_sosiki/joho\\_tsusin/eng/presentation/pdf/20200513Frequency\\_Reorganization\\_Action\\_Plan.pdf](https://www.soumu.go.jp/main_sosiki/joho_tsusin/eng/presentation/pdf/20200513Frequency_Reorganization_Action_Plan.pdf).
- [4] M. G. Kibria, G. P. Villardi, K. Ishizu, F. Kojima and H. Yano, “Resource allocation in shared spectrum access communications for operators with diverse service requirements,” EURASIP Journal on Advances in Signal Processing.
- [5] M. M. Butt, I. Macaluso, C. Galiotto, N. Marchetti, “Fair Dynamic Spectrum Management in Licensed Shared Access Systems,” IEEE Systems Journal 2019, Volume 13, Issue 3.
- [6] Akio Ikami, Takahiro Hayashi and Yoshiaki Amano, “Interoperator Channel Management for Dynamic Spectrum Allocation between Different Radio Systems,” IEICE Communications Express, Vol. 9, No. 10, pp. 512-518, 2020.
- [7] T. Kadowaki and H. Nishimori, “Quantum annealing in the transverse Ising model,” Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip Top., vol. 58, no. 5, pp. 5355–5363, 1998.
- [8] M. W. Johnson and M. H. S. Amin and S. Gildert and T. Lanting and F. Hamze and N. Dickson and R. Harris and A. J. Berkley and J. Johansson and P. Bunyk and E. M. Chapple and C. Enderud and J. P. Hilton and K. Karimi and E. Ladizinsky and N. Ladizinsky and T. Oh and I. Perminov and C. Rich and M. C. Thom and E. Tolkacheva and C. J. S. Truncik and S. Uchaikin and J. Wang and B. Wilson and G. Rose, “Quantum annealing with manufactured spins,” Nature, vol. 473, no. 7346, pp. 194–198, 2011.
- [9] E. G. Rieffel, D. Venturelli, B. O’Gorman, M. B. Do, E. M. Prystay, and V. N. Smelyanskiy, “A case study in programming a quantum annealer for hard operational planning problems,” Quantum Inf. Process., vol. 14, no. 1, 2014.
- [10] A. Lucas, “Ising formulations of many NP problems,” Front. Phys., vol. 2, no. 5, pp. 1–14, 2014.
- [11] T. Okuyama, T. Sonobe, K. Kawarabayashi, and M. Yamaoka, “Binary optimization by momentum annealing,” Phys. Rev. E 100, 012111, 2019.
- [12] ITU-R, “Guidelines for evaluation of radio interface technologies for IMT-2020,” Report ITU-R M.2412-0 (10/2017), Geneva, 2017.
- [13] P. Stubbings, “What can internet use tell us about our society and the economy?” Data Science Campus, Sep. 2019.
- [14] Gurobi Optimizer, Version 9.0.1. Gurobi Optimization Inc., Houston, USA, 2020. <https://www.gurobi.com/products/gurobi-optimizer/>.