

シフトスケジューリング問題における量子アニーリングの適用

浜田 捺希[†] 齋藤 和広[‡] 川島 英之[†][†] 慶應義塾大学総合政策学部 〒252-0882 神奈川県藤沢市遠藤 5322[‡] 株式会社 KDDI 総合研究所 〒356-8502 埼玉県ふじみ野市大原 2 丁目 1 番 15 号[†] 慶應義塾大学環境情報学部 〒252-0882 神奈川県藤沢市遠藤 5322E-mail: [†] {s17681nh, k-saito, river}@sfc.keio.ac.jp, [‡] ku-saitou@kddi-research.jp

あらまし シフトスケジューリング問題はコールセンターのアルバイトなどのシフトの組み合わせ最適化問題である。この問題はナーススケジューリング問題などの必要な人員数が定められ、複数の制約の元で人員を割り当てなければならないスケジューリングを行う現実問題に応用することが可能な問題である。この問題は考慮する人数や期間が増えるほど組合せ数が膨大となり、厳密解を求めることが困難となる。量子アニーリングは量子コンピュータの一種として、このような組み合わせ最適化問題を現実時間で解くことが期待されている。本論文ではアニーリングシミュレータとしてソフトウェア実装をした OpenJij を使い、目的関数や制約式を量子アニーリングに適用できる QUBO 式で定式化し、シフトスケジューリングにおける量子アニーリング適用の効果を評価した。

キーワード シフトスケジューリング問題, 量子アニーリング, 組み合わせ最適化問題

1. はじめに

シフトスケジューリング問題[1]はコールセンターのアルバイトなどのシフトの組み合わせを最適化する組み合わせ最適化問題である。応用例としてナース・スケジューリング問題などがあり、自動化にむけて様々な研究が行われてきた[2][3]。現状として、シフトスケジューリングは手作業で行われていることが多く、パートの希望シフトと照らし合わせた上で、時間帯毎に必要な人員数など、様々な条件を加味してシフトを組む。これらの制約を加味したシフトスケジューリング作業は、手作業で行われていることが多く、時間と労力を要する作業である。従って、シフトスケジューリング問題を自動化できることは重要である。

シフトスケジューリング問題はシフト条件を制約とする組み合わせ最適化問題として定式化が可能である。この問題は NP 完全[4]に分類され、最適解を出すことが困難であり、古典的なコンピュータでは多項式時間で問題を解くことが難しい問題とされている。

近年多項式時間で計算が困難な組み合わせ最適化問題の解決に量子アニーリングの活用が期待されている。量子アニーリングは、二つの量子ビット σ_i, σ_j の相互作用係数 J_{ij} 、及び量子ビット σ_i に影響する外部磁場エネルギー係数 h_i から、ハミルトニアン \mathcal{H} を最小化する量子ビット σ_i のスピン方向 (+1/-1) を決定する。この仕組みはイジングモデルとして以下のように表現できる。

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

組合せ最適化問題も目的関数の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、量子アニーリングによって解くことが可能となる。実問題を解く際には定式化を直感的に行うことが求められるため、以下のように量子ビット σ_i をバイナリ変数 $x_i \in \{0,1\}$ に変換する。

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

量子アニーリング利用時は、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[5][6][7][8]。

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

したがってシフトスケジューリング問題の目的関数、制約を定式化し、数式(3)の QUBO 式に適用することによってシフトスケジューリング問題を量子アニーリングに適用することが可能となる。

そこで本研究ではコールセンターのシフトスケジューリングを想定した基礎的なシフトスケジューリング問題の定式化を行い、QUBO 式を構築して量子アニーリングにおける性能を評価した。評価には、量子アニーリングのシミュレータとしてソフトウェア実装された OpenJij[9] を使用した。その結果、作業員数 6 名、7 日間、3 タームのシフトスケジューリング問題において 200ms で実行可能解が 100% 近辺で、energy 最小値を安定して得ることができた。

本論文の構成は以下の通りである。2 節においてシフトスケジューリング問題の問題設定と、量子アニーリングに適用するための定式化と QUBO 式における定式化を記述し、3 節において OpenJij を用いて各種パラメータをチューニングし、評価する。4 節で関連研究を、5 節で結論を述べる

2. シフトスケジューリング問題の量子アニーリング適用

2.1 問題設定

本論文におけるシフトスケジューリング問題は、コールセンターのシフトスケジューリングを想定する。コールセンターにはブースが存在し、そのブースに作業員を割り当てる。日によって必要なブース数が定められており、そのブース数に近づくようにそれぞれのタームに人を割り振る。

目的関数は二つ存在する。一つ目は、一日のシフトを数ターム（朝番、昼番、夜番など）に分け、そのそれぞれのタームにおいて割り振りたい人数を設定する。その人数と実際に割り振った人数との差を最小化する組み合わせ最適化問題である。二つ目は、あらかじめ指定された日数の中で作業員の希望ターム数を定め、それぞれの作業員の割り振られたターム数と希望ターム数との差を最小化する。これにより、ある特定の作業員にシフトが集中し、他の作業員がシフトに入ることができなくなる状態を回避することができる。

制約は二つ存在し、一つ目は作業員それぞれが具体的にどのタームに入りたいのかという希望シフトを提出してもらい、希望シフトに沿うように割り振る。二つ目は一緒にシフトを組んで欲しい作業員をグループにまとめ、そのグループに属している作業員を全員同じシフトに割り振る。

2.2 最適化問題の定式化

記号の定義を以下で示す

$A = \{\text{作業員 1, 作業員 2, ..., 作業員 } a\}$: 作業員の集合

$D = \{1 \text{ 日, } 2 \text{ 日, ..., } d \text{ 日}\}$: スケジュール対象日の集合

$T = \{\text{ターム 1, ターム 2, ..., ターム } t\}$: 1 日のターム（午前、午後、夕方など）の集合

$S^{dt} = \{s \mid s \text{ は } d \text{ 日のターム } t \text{ ごとに必要なブース数}\}$

$r_{adt} = \{0,1\}$: 作業員 a が d 日のターム t へ割り当て可能を表す集合（割り当て可能なら 1、不可なら 0 とする）

$R_a = N$: 作業員 a の希望割り当てシフト数 ($\sum R_a > \sum S^{dt}$)

$G_{ga} = \{0,1\}$: グループ g に作業員 a が所属していれば 1,

いなければ 0

$x_{adt} = \{0,1\}$: 組み合わせ問題を解いた際に作業員 a が d 日のターム t に割り振られたか否か（割り振られたら 1, 割り振られなかったら 0）

問題設定で説明したように本研究での目的関数は二つ存在する。

$$\text{Minimize } \sum_{d \in D} \sum_{t \in T} \left(\sum_{a \in A} |x_{adt} - S^{dt}| \right) \quad (4)$$

(4)式は d 日目におけるターム t に割り振った x_{adt} を合計し、もともと設定されていた希望ブース数 S^{dt} から引いた値の絶対値の総和である。これにより、それぞれのターム毎に設定された必要な作業員数に近づくように作業員を割り振ることができる。

$$\text{Minimize } \sum_{a \in A} \left(\sum_{d \in D} \sum_{t \in T} |x_{adt} - R_a| \right) \quad (5)$$

(5)式は作業員 a 毎に割り振られた x_{adt} を合計し、設定されたそれぞれの作業員の R_a から引いた差の絶対値の総和である。 R_a はそれぞれの作業員が提出している入りたい合計ターム数であるのでこの総和を最小化することで作業員の希望の通りにシフトを組むことができるようになる。またシフトが特定の作業員に偏ることなく割り振ることができるようになる。

次に制約を定式化する。

$$\begin{aligned} \forall a \in A, \forall d \in D, \forall t \in T \\ x_{adt} = 0, \quad \text{if } r_{adt} = 0 \end{aligned} \quad (6)$$

(6)式は作業員が希望していないターム ($r_{adt} = 0$) ではシフトを割り振らないようにするものである。

$$\forall a \in A, \forall d \in D, \forall g$$

$$\sum_a x_{adt} G_{ga} = \left(0 \text{ or } \sum_a G_{ga} \right) \quad (7)$$

(7)式は同じグループに割り振られた作業員全員が同じシフトに割り振られるようにするものである。制約違反を起こす例として、作業員 2 と作業員 3 が同じグループに所属している場合に二人が一緒にのシフトに入っていない状態、 $x_{2dt} = 0, x_{3dt} = 1$ となっていたとする。この時(7)左辺は $\sum_a x_{adt} G_{ga} = 1$, (6)右辺は $\sum_a G_{ga} = 2$ となり、(7)式は満たされない。よって制約違反を表せたこととなる。

2.3 QUBO 式における定式化

量子アニーリングに適用するために QUBO 式を構築する. 2.2 節の(4)~(7)式からハミルトニアン \mathcal{H} を以下に定義する.

$$\mathcal{H} = \lambda_i \mathcal{H}_i + \lambda_j \mathcal{H}_j + W_k \mathcal{H}_k + W_l \mathcal{H}_l \quad (8)$$

$$\mathcal{H}_i = \sum_{d \in D} \sum_{t \in T} \left(\sum_{a \in A} x_{adt} - S^{dt} \right)^2 \quad (9)$$

$$\mathcal{H}_j = \sum_{a \in A} \left(\sum_{d \in D} \sum_{t \in T} x_{adt} - R_a \right)^2 \quad (10)$$

$$\mathcal{H}_k = \sum_{a \in A} \sum_{d \in D} \sum_{t \in T} (1 - r_{adt}) x_{adt} \quad (11)$$

$$\mathcal{H}_l = \sum_d \sum_t \sum_g \left\{ \left(\sum_a G_{ga} - \sum_a x_{adt} G_{ga} \right) \sum_a x_{adt} G_{ga} \right\} \quad (12)$$

(8)式の \mathcal{H} が全体として最小となるよう, アニーリングシミュレータが量子アニーリングで最適解に近い値になるよう演算を行う. (8)式の λ_i, λ_j は目的関数 $\mathcal{H}_i, \mathcal{H}_j$ のそれぞれの係数である. \mathcal{H}_i の値が小さい場合は各タームで割り振られた作業員数が必要ブース数に近いことを表し, \mathcal{H}_j の値が小さい場合は各作業員の割り振りがそれぞれの合計希望シフト数に近いことを表している. 係数 λ_i, λ_j の値を他方と差をつけて設定することで求解における目的関数の優先度を調整することができる. 例えば $\lambda_i = 1, \lambda_j = 2$ と設定した場合, \mathcal{H} 全体を最小化するよう量子アニーリングが実行されるため \mathcal{H}_i に比べて \mathcal{H}_j の値が小さくなるよう解を得られやすくなる.

同様に W_k, W_l は制約項 $\mathcal{H}_k, \mathcal{H}_l$ それぞれのペナルティ係数となっている. $\mathcal{H}_k, \mathcal{H}_l$ は制約項となっており, それぞれの制約を満たしていれば値は 0 になり, 制約違反をすれば値が正の値を取るよう定義されている. よって目的関数項の値を十分超える大きさの W_k, W_l の値を設定することで制約違反が起こらないように量子アニーリングで解を導き出すことができる.

(9)式は前節の(4)式を QUBO 式に適用したものである. d 日目におけるターム t に割り振った x_{adt} を合計し, もともと設定されていた希望ブース数 S^{dt} から引いた値の二乗の総和となる. 二乗することによりそれぞれの差を正の値として受け取ることができる.

(10)は前節の(5)を QUBO 式に適用したものである. 作業員 a 毎に割り振られた x_{adt} を合計し, 設定されたそれぞれの作業員の R_a から引いた差の二乗の総和とな

る. (9)と同様に二乗することでそれぞれの差を正の値として受け取っている.

(11)は前節の(6)の制約条件を QUBO 式として定義した式である. 作業員が希望していないターム ($r_{adt} = 0$) でシフトを割り振る ($x_{adt} = 1$) と制約違反となり, 値が $(1 - r_{adt})x_{adt} = 1$ となり制約項が正の値をとる. それ以外の場合全てにおいて $(1 - r_{adt})x_{adt} = 0$ となるように定義している.

(12)は前節の(7)の制約条件を QUBO 式として定義した式である. 以下で本制約条件の QUBO 式的作用を例で説明する.

図 1

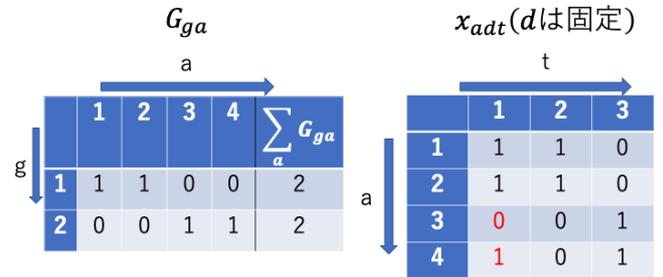


図 1.1 G_{ga} の一例

図 1.2 x_{adt} の一例

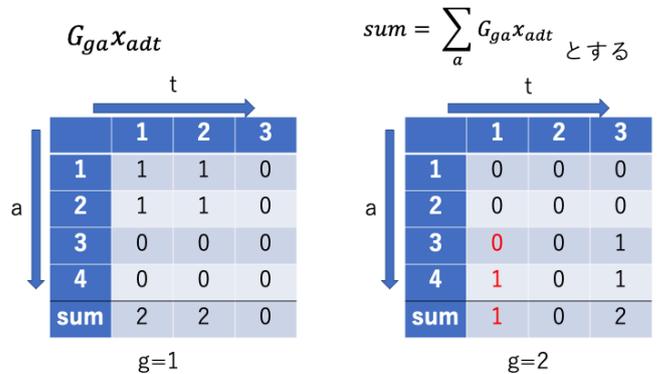


図 1.3 $g=1$ での $G_{ga}x_{adt}$

図 1.4 $g=2$ での $G_{ga}x_{adt}$

図 1.1 は G_{ga} , 作業員 1, 作業員 2 がグループ 1 に属していることを示し, 作業員 3, 作業員 4 がグループ 2 に属していることを表している.

図 1.2 は解として得られた x_{adt} を示す. 同じグループに属している作業員は全く同じシフトを組むことが制約条件である. 図 1.2 ではグループ 1 に属する作業員 1 と作業員 2 が全く同じ解を持ち, 制約に準拠しているが, グループ 2 に属する作業員 3 と作業員 4 は図の赤字で表している $t=1$ において同じシフトが組まれておらず制約違反となる.

その二つの表をそれぞれ対応する a と t で掛け合わせ

たもので制約違反していないグループ1が図 1.3, 制約違反を起こしているグループ2が図 1.4の表となっている。それらの総和と(12)式を比べると, (12)式が制約項として正しく機能していることが確認できる。

3. 評価

3.1 評価内容

本節ではハミルトニアン \mathcal{H} を energy と呼び, 理解しやすいように二つのペナルティ係数 W_k, W_l はそれぞれ w_desire, w_group と呼ぶこととする。

本評価では, 制約を限定した基礎的なシフトスケジューリング問題として, 2 節で述べた問題設定のシフトスケジューリング問題に量子アニーリングを適用した際の実行時間と energy, 実行可能解の関係を評価する。また, 評価の前に, 量子アニーリングで使用されるペナルティ係数をチューニングする。これにより評価において制約違反の解の数と実行時間を大幅に削減する。本評価では, 7 日間の 3 タームを固定し, 作業員数を増やすことで変数の個数を変化させた。表 1 が本評価で利用したデータとそれぞれの変数の個数である。

表 1 評価で利用するデータと変数の個数

作業員数 a	日数 d	ターム数 t	変数の個数
6	7	3	126
10	7	3	210
15	7	3	315
20	7	3	420

3.2 評価環境

本研究では量子アニーリングシュミレータとして OpenJij を利用した。OpenJij は, オープンソースソフトウェアとして, 量子アニーリングをシミュレートする Simulated Quantum Annealing (SQA) の Python 用 API を提供している。本評価ではバージョン 0.1.1 の SQA を利用した。SQA の設定は, $\beta=10, \gamma=1.0, \text{trotter}=10, \text{num_reads}=100$ とした。使用した環境は macOS Catalina バージョン 10.15.7, プロセッサは 2.9 GHz デュアルコア Intel Core i5 メモリは 8 GB 2133 MHz LPDDR3 である。

3.3 ペナルティ係数のチューニング

ここでは, 評価で利用するペナルティ係数の値を決めるために, チューニングを行う。チューニングの流れは, ペナルティ係数の比率決定, ペナルティ係数の値の決定の順で行う。

3.3.1 ペナルティ係数の比率決定

まずは(8)式の W_k, W_l の比率を決定する。 W_k, W_l の

比率を変えることにより, 制約項(11)(12)式がどちらかの式の値だけが大きくなり, バランスをとるようにチューニングした。



図 2.1 変数 126 個



図 2.2 変数 210 個

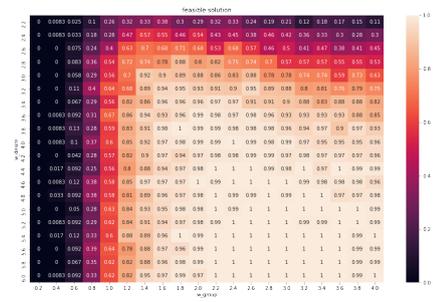


図 2.3 変数 315 個

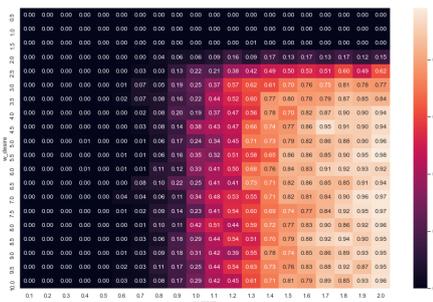


図 2.4 変数 420 個

図 2 w_desire, w_group を変化させた実行可能解の

確率のヒートマップ

図 2 は図 2.1 から図 2.4 の順に変数の個数が 126,210,315,420 で w_{desire} , w_{group} を変化させた際に得られた実行可能解の確率のヒートマップである。これらの結果より w_{desire} と w_{group} で実行可能解の確率が均衡の取れている値を選択する。ここでは、それぞれの変数の個数において、表 2 に示すペナルティ係数の比率を選択した。

表 2 変数の個数毎のペナルティ係数の比率

変数の個数	126	210	315	420
w_{desire} の比率	7.3	7.2	2.8	3
w_{group} の比率	1.2	1.2	1	1

3.3.2 ペナルティ係数の値決定

次に 3.3.1 節で決定したペナルティ係数の比率に掛け合わせる値 base を決定し、ペナルティ係数の値を決定する。制約違反を全く起こさず、かつ energy が低くなる base 値を設定することが目的である。ここでは、 base 値を変化させ、実行可能解の確率と energy を計測した。

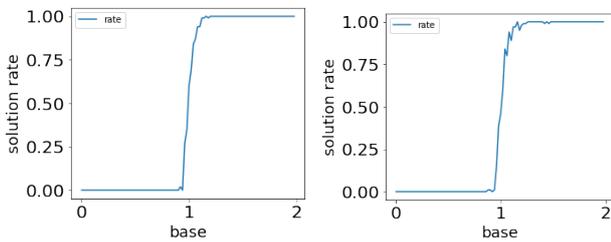


図 3.1 変数 126 個

図 3.2 変数 210 個

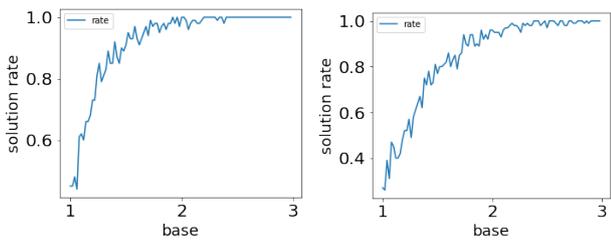


図 3.3 変数 315 個

図 3.4 変数 420 個

図 3 base 値と実行可能解の確率のグラフ

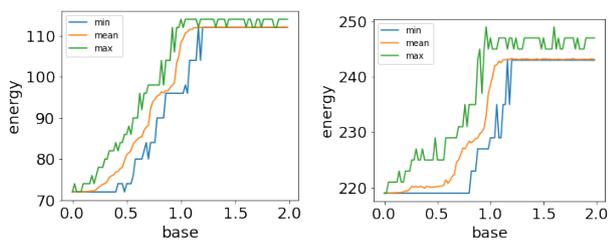


図 4.1 変数 126 個

図 4.2 変数 210 個

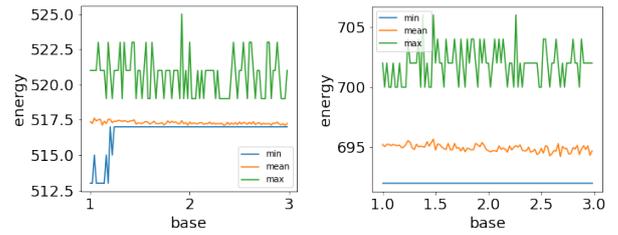


図 4.3 変数 315 個

図 4.4 変数 420 個

図 4 base 値と実行可能解の energy のグラフ

図 3 は図 3.1 から図 3.4 の順に変数が 126,210,315,420 個の際の base 値と実行可能解の確率のグラフであり、図 4 は図 4.1 から図 4.4 の順に変数が 126,210,315,420 個の際の base 値と実行可能解の energy のグラフである。実行可能解の確率が 1 付近で安定し、かつ energy が安定している base 値を選択する。本評価では表 3 で示す base 値に決定した。併せて、表 3 に、3.3.1 節で決定した w_{desire} と w_{group} の比率に base 値を掛け合わせペナルティ係数の値を示す

表 3 変数の個数に対するペナルティ係数の値

変数の個数	126	210	315	420
base 値	1.16	1.28	2.3	2.74
w_{desire}	8.468	9.216	6.44	8.22
w_{group}	1.392	1.536	2.3	2.74

3.4 実行時間の評価

本研究で使用しているアニーリングシミュレータの OpenJij ではアニーリング中のパラメータ(横磁場)を下げていくときの分割数を指す sweeps 数が存在する。 sweeps 数を増やすとアニーリングの精度を高くすることができるが、実行時間が長くなる。以下では 3.3.2 節で決定したペナルティ係数の下で sweeps 数を変化させ、量子アニーリングを実行した結果を示す。

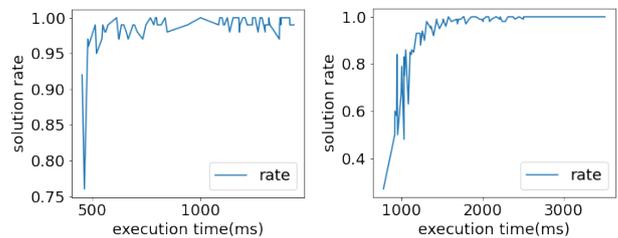


図 5.1 変数 126 個

図 5.2 変数 210 個

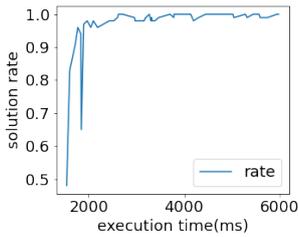


図 5.3 変数 315 個

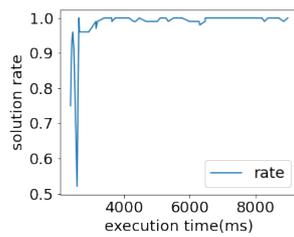


図 5.4 変数 420 個

図 5 実行時間と実行可能解の確率のグラフ

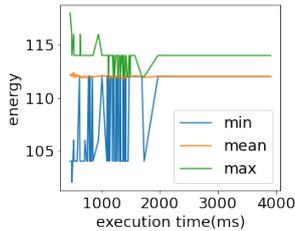


図 6.1 変数 126 個

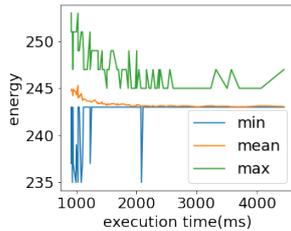


図 6.2 変数 210 個

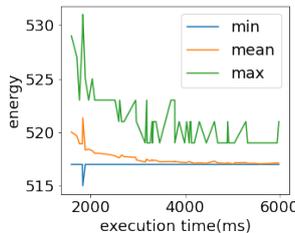


図 6.3 変数 315 個

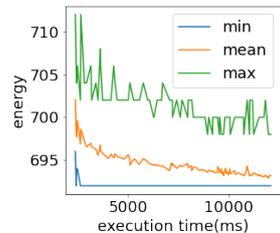


図 6.4 変数 420 個

図 6 実行時間と実行可能解の energy のグラフ

図 5 は図 5.1 から図 5.4 の順に sweeps 数を変化させた時の変数が 126,210,315,420 個の際の実行時間と実行可能解の確率のグラフであり、それぞれのグラフの sweeps 数は以下の表のとおりである。

表 4 図 5 における sweeps 数を変化させた範囲

変数の個数	126	210	315	420
sweeps 数の範囲	6~150	6~150	6~180	6~180

図 6 は図 6.1 から図 6.4 の順に sweeps 数を変化させた時の変数が 126,210,315,420 個の際の実行時間と実行可能解の energy のグラフであり、それぞれのグラフの sweeps 数を変化させた範囲は以下の表の通りである。

表 5 図 6 における sweeps 数を変化させた範囲

変数の個数	126	210	315	420
sweeps 数の範囲	8~180	8~180	9~180	9~300

変数が 126 個の際は図 5.1 より、実行時間が 500ms 以下では実行可能解が安定して得られないが、500ms 以上では 90%以上で得られるようになっている。また

図 6.1 より、実行可能解の energy の最小値は 2000ms を超えると 112 を保たれている。

変数が 210 個の際は図 5.2 より、実行時間が 2000ms 近辺で実行可能解は 100%に近い確率で安定して得られている。また図 6.2 より、1400ms を超えると energy の最小値は 243 を保っている。

変数が 315 個の際は図 5.3 より、実行時間が 2000ms を超えると実行可能解を 100%付近で安定して得られるようになり、図 6.3 より、2000ms 以上では energy の最小値が 517 を保ち、3000ms 以上で平均値も最小値付近で得られている。

図 5.4 より、変数が 420 個の際は実行時間が 4000ms 以上で 90%より高い確率で実行可能解が得られるようになる。また図 6.4 より、1000ms 以上で energy の最小値は 692 を維持し、7500ms 以上で平均値は最小値近辺で安定する結果が得られた。

4 関連研究

シフトスケジューリング問題の応用例としてナーススケジューリングがある。本研究ではコールセンターのシフトスケジューリングをモデルにしたのに対し、ナーススケジューリングは看護師の勤務表作成がモデルとなっている。看護師のスキルや希望を考慮し、かつ制約を設けたスケジューリングは手作業では時間がかかってしまう。特に制約は勤務回数の上限と下限、スキルレベルに合わせたチーム構成を加味したグループ分けなど制約は多岐にわたり、全てを満たすことは非常に難しい。

池上 [2]は目的関数を満たす条件をシフト拘束条件、制約を満たす条件をナース拘束条件とし部分問題として条件を満たすように組み上げ、組み合わせの自由度を残したまま、残りのシフトを手作業で行う部分問題軸アプローチの手法をとるなどが考えられている。組み合わせ問題を解く際にはシミュレーテッドアニーリングやタブー探索法を用いた制約充足問題のためのソルバーである wcsp を用いている。

Miller et al.[3] は数理計画法を用い、独自のアルゴリズムでナーススケジューリングを解いた。

組み合わせ最適化問題に特化したソルバーとして、青野ら[10] はアメーバが中枢神経系のような中央集権型の情報処理ユニットをもたないにもかかわらず、全体として高度な情報処理機能を発揮することができる点に着目したアメーバ計算機を考案した。アメーバの周囲に餌を適当に配置するとアメーバは栄養吸収量を最大化するべくすべての足を伸ばしたいが、光照射されると縮退を促される性質を利用する。アメーバが被照射リスクを最小化できる組合せの足だけを伸ばそうと試行錯誤する過程で、組合せ最適化問題の近似解

を探索する。この性質を用いて、巡回セールスマン問題で都市数 n が増大してもその質を劣化させることなく、探索に要する時間の増大を n のほぼ線形関数に抑えながら近似解を発見している。

5 結論

本論文では制約を限定した基礎的なシフトスケジューリング問題の QUBO 式を構築した。その後、量子アニーリングのアニーリングシュミレータの OpenJij を用いて小規模、中規模でのシフトスケジューリング問題のペナルティ係数をチューニングし、実行時間を変化させた際の実行可能解を得られる確率、energy の推移を示した。変数が小規模の 126 個の際には約 200ms で実行可能解が 100% 近辺で、energy 最小値を安定して得られ、中規模の 420 個の際には約 6000ms で実行可能解が 90% 以上の energy 最小値を安定して得られた。

今後は、現実問題を想定した変数が 1000 個以上の大規模なスケジューリング問題の評価も行いたいと考えている。また今回評価に利用した OpenJij はアニーリングシミュレータであるため、実際に量子効果を利用した量子アニーリングマシンでも評価を行いたい。その後、既存手法との比較を行い、量子アニーリングのシフトスケジューリング問題に対する評価を行いたい。

謝辞

This work is partially supported by JP19H04117 and Project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

参考文献

- [1] 茨木俊秀. "組合せ最適化とスケジューリング問題: 新解法とその動向." 計測と制御 34.5 (1995): 340-346.
- [2] 池上敦子. "ナース・スケジューリング-調査・モデル化・アルゴリズム." 統計数理第 53 巻第 2 号 (2005): 231-259.
- [3] Miller, Holmes E., William P. Pierskalla, and Gustave J. Rath. "Nurse scheduling using mathematical programming." Operations Research 24.5 (1976): 857-870.
- [4] Ullman, Jeffrey D. "NP-complete scheduling problems." Journal of Computer and System sciences 10.3 (1975): 384-393.
- [5] 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.
- [6] V. Choi, "Minor-embedding in adiabatic quantum computation: I. the parameter setting problem," Quantum Inf. Process., vol. 7, no. 5, pp. 193–209, 2008.
- [7] Z. Bian, F. Chudak, R. Israel, B. Lackey, W. G. Macready, and A. Roy, "Discrete optimization using quantum annealing on sparse Ising models," Front. Phys., vol. 2, no. 56, pp. 1–10, 2014.
- [8] A. Lucas, "Ising formulations of many NP problems," Front. Phys., vol. 2, no. 5, pp. 1–14, 2014.
- [9] OpenJij, <https://github.com/OpenJij/OpenJij>
- [10] 青野真士, 鯨井悠生, and 野崎大幹. "サイバー空間とフィジカル空間を癒合するアメーバ計算パラダイム." 人工知能 33.5 (2018): 561-569.