

# 全並列アニーリングのための自律的パラメータ制御

小此木 大輝<sup>1,a)</sup> 神保 聡<sup>1</sup> 安藤 洸太<sup>1</sup> Thiem Van Chu<sup>1</sup> 劉 載勳<sup>1</sup> 本村 真人<sup>1</sup> 川村 一志<sup>1</sup>

**概要:** イジングモデルを用いて組合せ最適化問題を効率的に解くためのアニーリング計算技術に高い関心が寄せられている。SCA (stochastic cellular automata annealing) は SA (simulated annealing) の原理を拡張したアニーリング手法であり、スピン状態を全並列に更新可能であるため、高速に基底状態探索が実現できるものと期待される。SCA ではスピン反転を抑制するパラメータの適切なスケジューリングが必要となるが、そのような制御は解きたい問題によって異なることから、ユーザビリティの低下が懸念される。本稿では、SCA とパラメータ制御機構を一体化した新たな手法として APC-SCA (autonomous pinning effect control SCA) を提案する。APC-SCA は、個々のスピン反転に着目したフィードバック制御によりパラメータの自律制御を可能にする。最大カット問題と N クイーン問題を対象に APC-SCA を評価した結果、問題ごとに細かくパラメータ調整することなく従来の SCA よりも高品質な解を得ることに成功した。

## 1. はじめに

金融、物流、交通など社会の様々な場面に存在する組合せ最適化問題 [1–4] は、一般的に NP 困難な問題として知られている。アニーリング計算は、組合せ最適化問題を効率的に解くことのできる手法として近年注目を浴びている。アニーリング計算では、入力形式としてイジングモデルを用いることで、問題を統一的に表現可能である。イジングモデルの基底状態を探索することが組合せ最適化問題の最適解を探索することに対応する。

様々なアニーリングアルゴリズムとその専用ハードウェアが開発されている。代表的なものとして D-Wave [5] の計算原理となっている量子アニーリング [6] がある。量子アニーリングマシンは、低温状態で動作させる必要があるため、大規模な冷却装置が必要となる。そのような理由から実用が難しいのが現状である。他方、SA (simulated annealing) [7] とその拡張手法 [8–10] も同様に組合せ最適化問題を効率的に解くことができる。拡張手法の中には DA (digital annealing) [8] や SCA (stochastic cellular automata annealing) [10] がある。DA と SCA はそれぞれデジタルアニーラ [8] と STATICA [11] の計算原理となっており、これらは半導体チップ上の常温で動作するため、大規模な装置は必要ない。

SCA はイジングモデルの全スピンを並列更新可能なアルゴリズムであり、高速な基底状態探索を実現できるが、現状ではスピン反転を抑制するパラメータ  $q$  の制御に課題が

ある。 $q$  を適切に制御できない場合、局所解に陥るか、エネルギーの高いスピン状態で安定してしまう。ベンチマーキングで使われるような基本的な組合せ最適化問題については経験的に  $q$  の適切な制御が知られているが、未知の組み合わせ最適化問題については適切な制御を知るために試行錯誤が必要である。これにより SCA のユーザビリティは低下してしまう。

本稿では上記の問題を克服する新たなアニーリング手法として APC-SCA (autonomous pinning effect control SCA) を提案する。提案手法はパラメータ  $q$  をアニーリング中に自律制御することでユーザがパラメータ調節をする手間を大幅に削減可能とする。具体的には、個々のスピンのパラメータ  $q_i$  を持たせ、個々のスピン反転に応じてフィードバック制御することで、自動調節を実現する。最大カット問題と N クイーン問題を用いて評価したところ、本提案手法が SCA において自律的かつ適切に  $q$  を制御可能であることを確認した。

本稿の主な貢献は以下の通りである。

- SCA のユーザビリティを改善する全並列アニーリング手法として APC-SCA を提案した。
- 評価結果から、APC-SCA は実験に使った全ての問題で共通のパラメータを使い、パラメータを手動で最適化した SCA よりも良い解を得られることを示した。
- GPU での評価により、APC-SCA は従来のアルゴリズム (SA ならびに SCA) と比較して短い時間で良い解を得られることを示した。

本稿は以下のように構成される。2 章では SCA におけるパラメータ  $q$  の問題点を明らかにする。3 章では SCA の

<sup>1</sup> 東京工業大学

<sup>a)</sup> okonogi.daiki@artic.iir.titech.ac.jp

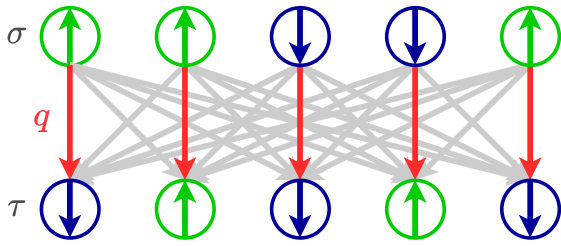


図 1: SCA における並列スピン更新の図.

問題を克服する新たなアニーリング手法として APC-SCA を提案する. 4 章では評価結果を示す. 最後に, 5 章に本稿のまとめを示す.

## 2. SCA とその問題点

### 2.1 イジングモデルとアニーリング計算

イジングモデルは組合せ最適化問題を統一的に表現可能な形式であり, スピン  $\sigma = \{\sigma_i\}_{1 \leq i \leq N}$ , 相互作用  $J = \{J_{ij}\}_{1 \leq i, j \leq N}$ , 外部磁場  $h = \{h_i\}_{1 \leq i \leq N}$  で構成される. 各スピン  $\sigma_i \in \sigma$  は上向きまたは下向きの状態を取り, それぞれが +1 と -1 に対応する.

スピン状態から, イジングモデルのエネルギーを,

$$H(\sigma) = -\frac{1}{2} \sum_{i \neq j} J_{ij} \sigma_i \sigma_j - \sum_i h_i \sigma_i \quad (1)$$

と定義する. 最小のエネルギーになるスピン状態は基底状態と呼ばれる. 組合せ最適化問題をイジングモデルに変換する際には, 最適解がイジングモデルの基底状態に対応するようにエネルギー関数を定義する. 関連研究 [12–15] では, 様々な組合せ最適化問題がイジングモデルに変換できることが示されている.

### 2.2 SCA のアルゴリズム

SCA [10] は並列性により高速なスピン更新可能にしたアニーリングアルゴリズムである. 図 1 のように, SCA は現在のスピン状態  $\sigma$  から次の状態  $\tau$  を同時に生成する. SCA のアルゴリズムをアルゴリズム 1 に示す. SCA は次のエネルギー関数を最小化する.

$$H(\sigma, \tau) = -\frac{1}{2} \sum_{i \neq j} J_{ij} \sigma_i \tau_j - \frac{1}{2} \sum_i h_i (\sigma_i + \tau_i) + q \sum_i (1 - \sigma_i \tau_i). \quad (2)$$

第 3 項に注目すると, スピン状態  $\sigma_i$  が変化するとき, エネルギーは  $2q$  だけ増加することがわかる. この振り舞いからパラメータ  $q$  はピンニングパラメータと呼ばれており, 現状態  $\sigma_i$  と次状態  $\tau_i$  の結びつきの強さを与える.

SCA を使ってイジングモデルの基底状態探索をする際には, 最終的なスピン状態における  $H(\sigma, \tau)$  が  $H(\sigma)$  と等し

## アルゴリズム 1 SCA: Stochastic Cellular Automata annealing

入力:

- スピンの初期状態 ( $\sigma$ )
- スピン間相互作用 ( $J$ ), 外部磁場 ( $h$ )
- モンテカルロ (MC) ステップ数 ( $S$ )
- 擬似温度スケジューリング ( $T(s)$ )
- ピンニングパラメータスケジューリング ( $q(s)$ )

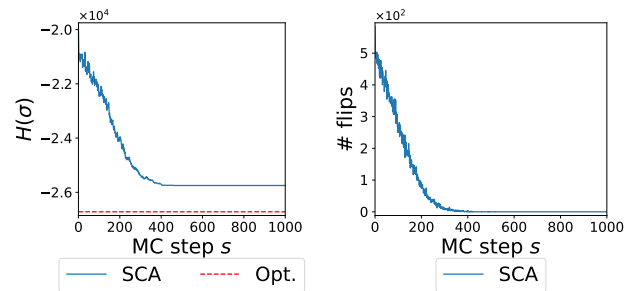
出力:

最適化されたスピン状態 ( $\sigma$ )

```

1: for s = 1 to S do
2:   for i = 1 to N do
3:     局所場の計算
       ▷  $\tilde{h}_i = \sum_j J_{ij} \sigma_j + h_i$ 
4:     スピン反転確率の計算
       ▷  $p_i = \text{sigmoid}(-(\tilde{h}_i \sigma_i + q(s))/T(s))$ 
5:     乱数 rand = [0, 1) の生成
6:     if  $p_i > \text{rand}$  then
7:        $\tau_i = -\sigma_i$ 
8:     else
9:        $\tau_i = \sigma_i$ 
10:   for i = 1 to N do
11:      $\sigma_i = \tau_i$ 

```



(a) イジングエネルギー.

(b) スピン反転の数.

図 2:  $q(s) = \lambda/2$  として SCA で最大カット問題 G22 [16] を解いた時の挙動を示した図.  $(T_{\text{init}}, T_{\text{final}}) = (10, 0.1)$  の指数温度スケジューリング  $T_{\text{exp}}(s)$  を使用している. スピン反転がアニーリング後半 ( $s \geq 600$ ) で止まり, 局所解から脱出できていない.

くなり, 安定していなければならない.  $H(\sigma, \tau) = H(\sigma)$  の条件は  $\sigma = \tau$  であり, 前述の通り  $q$  に大きな値を使用すると容易にこの条件を満たすことができる. 先行研究 [10] により, SCA で基底状態を求めるための十分条件が  $q \geq \lambda/2$  ( $\lambda$  は  $-J$  の最大固有値) で与えられることが数学的に示されている. しかし, この証明では対数関数的に温度を減少させることを想定しており, この場合アニーリングを完了するまでに非現実的な時間を必要とする. 実用的に組合せ最適化問題をアニーリングアルゴリズムで解く場合は通常, MC ステップ数 ( $S$ ) を現実的な長さに制限する. 次に示すような, 指数関数的に温度を下げる指数スケジューリング  $T_{\text{exp}}(s)$  は高温, 中温, および低温のバランスがよく, 実用に適している.

表 1: 本稿で使用したイジングモデルの一覧.

名称	スピンの数	相互作用の数	グラフ構造
G22	2 000	19 990	全ての相互作用が $J_{ij} = -1$ のランダムグラフ.
G30	2 000	19 990	全ての相互作用が $J_{ij} \in \{+1, -1\}$ のランダムグラフ.
G32	2 000	4 000	トーラス状に $J_{ij} \in \{+1, -1\}$ の結合を持つグラフ.
G35	2 000	11 778	平面状に $J_{ij} = -1$ の結合を持つグラフ.
32 クイーン	1 024	52 576	縦横斜めの方に負の相互作用を持つ 2 次元グラフ.

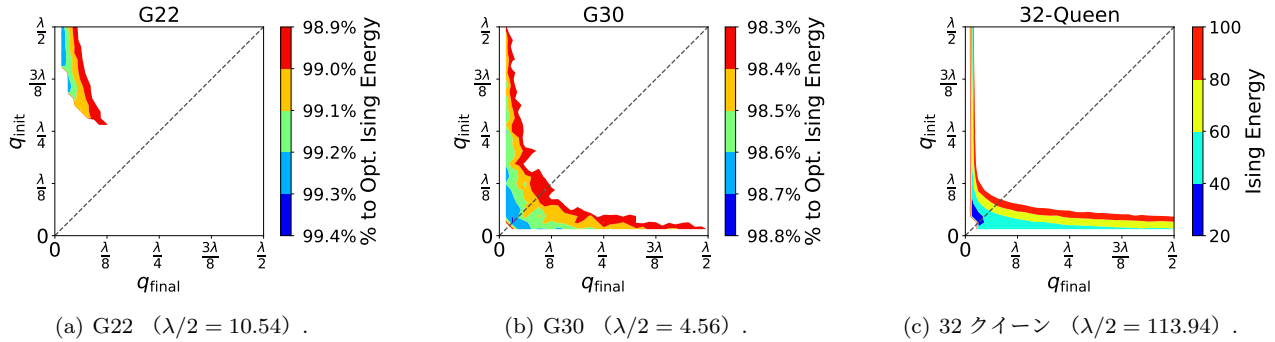


図 3: 2つの最大カット問題 (G22, G30) と 32 クイーン問題を SCA で解いた時のイジングエネルギーのヒートマップ. 式 (4) と式 (5) に基づいてピンニングパラメータのスケジューリングを変化させて実行した. 着色された領域に含まれる  $(q_{\text{init}}, q_{\text{final}})$  を用いると, 比較的良好な解が得られるが, 藍色の部分に含まれる最適なペアを探索することは非常に困難である.

$$T_{\text{exp}}(s) = T_{\text{init}} \times r_T^{s-1}, \quad r_T = \left( \frac{T_{\text{final}}}{T_{\text{init}}} \right)^{\frac{1}{s-1}}. \quad (3)$$

ここで  $T_{\text{init}}$  と  $T_{\text{final}}$  はそれぞれ初期温度と最終温度である. SA においても経験的に良い解を得ることができる一般的なスケジューリングであるが, 指数スケジューリングを使った場合, SCA で  $q = \lambda/2$  を使った時に基底状態に到達する数学的な保証はない.

ベンチマークとして知られる最大カット問題 G22 [16] を  $S = 1000$ ,  $q(s) = \lambda/2$ , および  $(T_{\text{init}}, T_{\text{final}}) = (10, 0.1)$  とした  $T_{\text{exp}}(s)$  で解いた時の SCA の挙動を図 2 に示す. この図から, アニーリングの後半で局所解に陥り, 基底状態探索が機能していないことがわかる. ピニングパラメータのスケジューリング  $q(s)$  を最適化することにより解の質を向上させられる可能性はあるが, 適切なスケジューリングが問題ごとに異なるため, その最適化は難しい.

### 2.3 SCA におけるピンニングパラメータ制御

本節では, いくつかの組合せ最適化問題をピンニングパラメータを変えながら SCA で解くことにより, 適切な  $q(s)$  を求めることが簡単ではなく, かつ非自明であることを示す. ここでは  $q(s)$  として, 次の式で表される指数関数を想定する.

$$q(s) = q_{\text{init}} \times r_q^{s-1}, \quad r_q = \left( \frac{q_{\text{final}}}{q_{\text{init}}} \right)^{\frac{1}{s-1}}, \quad (4)$$

ここで  $q_{\text{init}}$  と  $q_{\text{final}}$  はそれぞれ  $q(1)$  と  $q(S)$  で使用する値

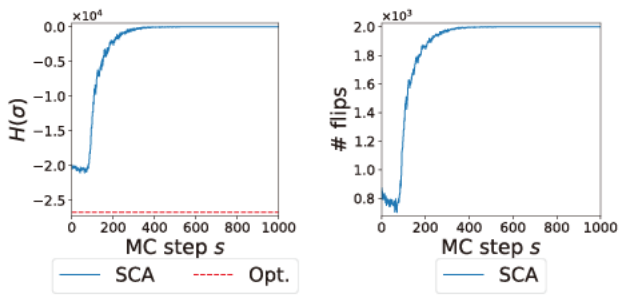
に対応している.  $q_{\text{init}}$  と  $q_{\text{final}}$  を

$$q_{\text{init}} \in \left\{ \frac{\lambda}{64} n \right\}_{1 \leq n \leq 32}, \quad q_{\text{final}} \in \left\{ \frac{\lambda}{64} n \right\}_{1 \leq n \leq 32} \quad (5)$$

のように  $(0, \lambda/2]$  の範囲で独立に変化させ, 3つの異なる組合せ最適化問題 (G22, G30, および 32 クイーン) を解いた. G22 と G30 [16] は最大カット問題であり二つの問題は異なるグラフ構造を持つ. 32 クイーン問題は制約充足問題の代表的なベンチマーク問題のひとつである. 各問題の詳細を表 1 に示す. 実験では,  $(q_{\text{init}}, q_{\text{final}})$  の各パラメータ設定に対してアニーリングを 128 回ずつ試行し, イジングエネルギーの平均値を算出している. 図 3(a), 図 3(b), 図 3(c) はそれぞれ G22, G30, および 32 クイーンの結果を表している. それぞれの図では比較的良好な解が得られる  $(q_{\text{init}}, q_{\text{final}})$  の範囲を着色している. 藍色の範囲は探索した中で一番低いエネルギーを得られた  $(q_{\text{init}}, q_{\text{final}})$  の組み合わせを含む. 補助線として  $q_{\text{init}} = q_{\text{final}}$  となる直線を破線で引いた. この実験から, 次の事実が明らかとなった.

- 取りうる  $q_{\text{init}}$  と  $q_{\text{final}}$  の範囲のうち, 有効な組み合わせは限定的である.
- 最適な  $(q_{\text{init}}, q_{\text{final}})$  の組み合わせが G22 では左上角, G30 と 32 クイーンでは左下角である. したがって問題ごとに適切なスケジューリングが異なる.

現状, 未知の組合せ最適化問題に対して適切な  $q$  のスケジューリングを知るためには, 様々な  $q(s)$  を繰り返し実行するしかない. このような状況は, 様々な組合せ最適化問



(a) イジングエネルギー. (b) スピン反転の数.

図 4: 図 2 と同じ最大カット問題 (G22) を  $q(s) = \lambda/6$  で解いた時の SCA の挙動.  $q < \lambda/6$  とすると, スピン状態がイジングエネルギーの高い状態に向かってしまう場合がある.

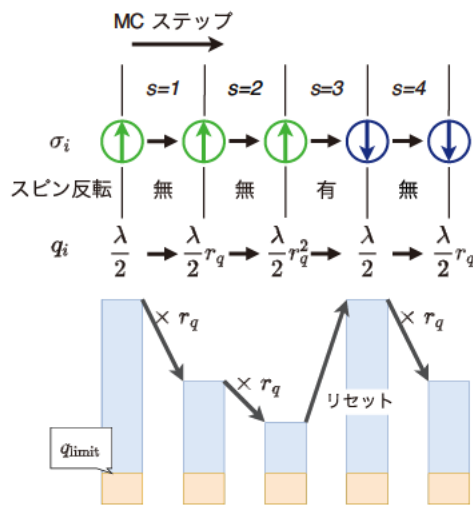


図 5: 自律的ピンニングパラメータ制御の模式図.  $q_i$  は  $i$  番目のスピンの反転しない限り徐々に減少し, スピン反転が起こると  $\lambda/2$  にリセットされる.

問題を効率的かつ統一的に解くというアニーリング計算の目的に反する. したがって,  $q$  を自律的に制御する新たなアプローチが必要である.

### 3. 提案手法

#### 3.1 基本方針

ピンニングパラメータの自律制御を考案するために, 固定された値を  $q$  として使ったときの挙動を観察した. 大きな値 ( $q = \frac{\lambda}{2}$ ) を使った場合, 現状態を維持する効果が大きすぎるため, 図 2 のようにアニーリングの後半ではスピン状態が簡単に局所解に嵌ってしまう. 小さな値 ( $q = \frac{\lambda}{6}$ ) を使った場合, 現状態を維持する効果が小さすぎるため, 図 4 のようにフリップ数が極端に多くなり, イジングエネルギーが高い状態に遷移してしまう. これらの結果から, フリップ数が多いときは  $q$  を大きくするべきであり, フリップ数が少ないときは  $q$  を小さくするべきであることが

### アルゴリズム 2 APC-SCA: Autonomous Pinning effect Control SCA.

入力:

- スピンの初期状態 ( $\sigma$ )
- スピン間相互作用 ( $J$ ), 外部磁場 ( $h$ )
- モンテカルロ (MC) ステップ数 ( $S$ )
- 擬似温度スケジューリング ( $T(s)$ )
- ピンニングパラメータの減少率 ( $r_q$ )
- ピンニングパラメータの下限 ( $q_{limit}$ )

出力:

最適化されたスピン状態 ( $\sigma$ )

```

1: for  $i = 1$  to  $N$  do
2:    $q_i = \lambda/2$ 
3: for  $s = 1$  to  $S$  do
4:   for  $i = 1$  to  $N$  do
5:     局所場の計算
        $\triangleright \tilde{h}_i = \sum_j J_{ij}\sigma_j + h_i$ 
6:     スピン反転確率の計算
        $\triangleright p_i = \text{sigmoid}(-(\tilde{h}_i\sigma_i + q_i)/T(s))$ 
7:     乱数  $\text{rand} = [0, 1)$  の生成
8:     if  $p_i > \text{rand}$  then
9:        $\tau_i = -\sigma_i$ 
10:       $q_i = \lambda/2$ 
11:    else
12:       $\tau_i = \sigma_i$ 
13:       $q_i = \max(q_i \times r_q, q_{limit})$ 
14:   for  $i = 1$  to  $N$  do
15:      $\sigma_i = \tau_i$ 

```

わかる. したがって, 自律制御のためにスピン反転の数に応じて  $q$  の値を増減させることが考えられる.

上記のような自律制御のためには, フリップ数の大小をどのようにフィードバックするかが問題となる. 全スピンのうちのどの程度のスピンのフリップすれば良いということを事前に知ることはできない. 一方で, 一つ一つのスピンの注目すると, あるスピンがフリップしたならば (しなかったならば), そのスピンに対しては, より大きな (小さな)  $q$  を割り当てても良いということが言える.  $i$  番目のスピンにパラメータ  $q_i$  を持たせ, スピン反転の有無により  $q_i$  を増減させる個別制御が, 本手法の基本方針となる.

#### 3.2 APC-SCA: Autonomous Pinning effect Control SCA

本節では 3.1 節で示した自律制御の基本方針に基づいた新たな全並列アニーリングアルゴリズム APC-SCA を提案する. このアルゴリズムはパラメータ  $q$  を自律的かつスピンごとに制御する機構を SCA に組み込んだものである. 提案する  $q_i$  の制御方式を図 5 に示す.  $q_i$  は  $i$  番目のスピンの反転しない限り徐々に減少し, 反転すると  $q_i$  は  $\lambda/2$  にリセットされる.

アルゴリズム 2 に APC-SCA のアルゴリズムを示す. アルゴリズム 1 との違いを青色でハイライトした. 従来の SCA と異なり,  $N$  個のピンニングパラメータ  $\{q_i\}_{1 \leq i \leq N}$



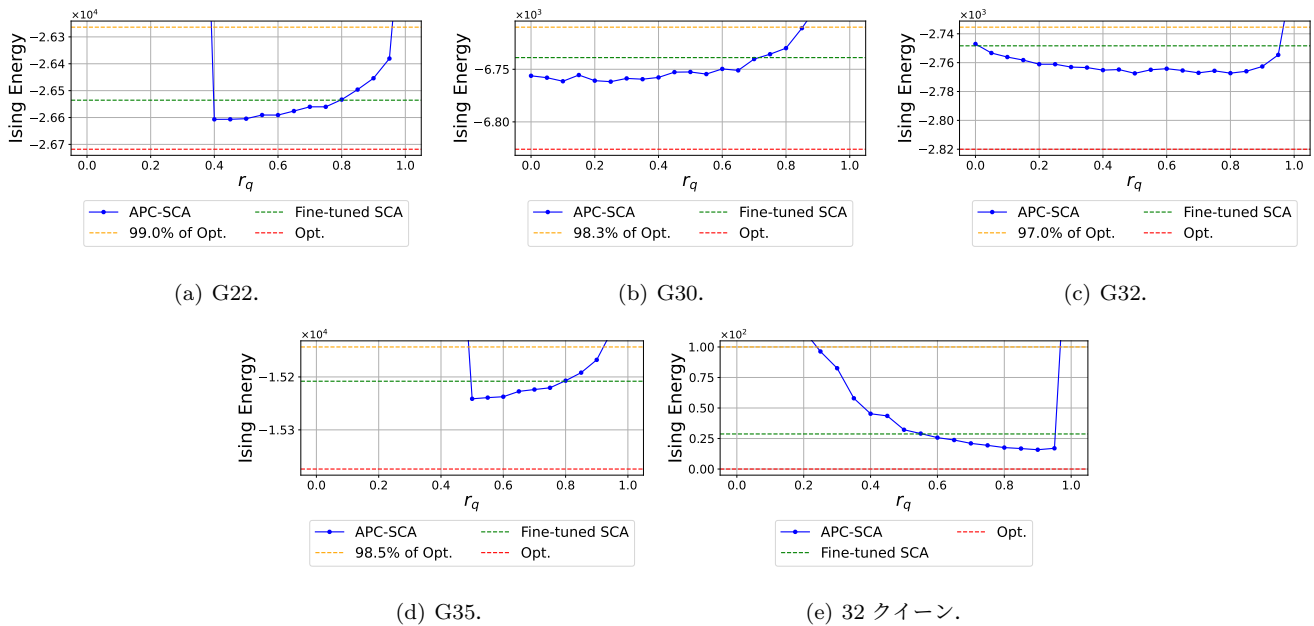


図 6:  $r_q$  を 0.0 から 1.0 まで変化させながら APC-SCA で各問題を解いた時のイジングエネルギーの平均値.  $r_q$  が [0.5, 0.7] の範囲であれば, APC-SCA は調整済 SCA より良い解を得られている.

表 2: 共通のパラメータ設定.

項目 (表記)	使用した値
MC ステップ数 ( $S$ )	1 000
試行回数 ( $M$ )	128
擬似温度の初期値 ( $T_{init}$ )	10.0
擬似温度の最終値 ( $T_{final}$ )	0.1

を用意し,  $\lambda/2$  で初期化する. (1-2 行目), 各 MC ステップで  $i$  番目のフリップ確率  $p_i$  を計算するための動作は, 個別の  $q_i$  を使う以外 SCA と同じである (6 行目). 毎回の MC ステップで,  $i$  番目のスピン反転の有無によって,  $q_i$  を更新する. この更新の時には,  $r_q < 1$  と  $q_{limit} \geq 0$  という 2 つのパラメータを使う.  $r_q$  と  $q_{limit}$  は  $q_i$  に対し, それぞれ減少率と下限を与える.  $i$  番目のスピンがフリップした場合,  $q_i$  は  $\lambda/2$  にリセットされる (10 行目).  $i$  番目のスピンがフリップしなかった場合,  $q_i$  は  $q_i \times r_q$  に減少するか, 下限  $q_{limit}$  にとどまる (13 行目).

式 (4) を使ってピンニングパラメータのスケジューリングを与える場合,  $q_{init}$  と  $q_{final}$  という 2 つのパラメータが必要になり, APC-SCA でもパラメータは  $r_q$  と  $q_{limit}$  が必要であるため, パラメータの数は変わらない. 4 章では, APC-SCA を評価しパラメータ設定の容易さが従来の SCA よりも高いことを示す.

#### 4. 評価

アルゴリズム 2 に示した APC-SCA のアルゴリズムを CUDA C++ で実装し, NVIDIA Tesla V100 上で実行・評価した. 従来 SCA の実装は先行研究 [17] のものを利用し,

表 3: SCA にピンニングパラメータのスケジューリングを事前に与える場合の最適化された ( $q_{init}, q_{final}$ ) の組. 問題ごとに式 (4) と式 (5) に基づいてグリッドサーチし, 微調整した結果である.

	G22	G30	G32	G35	32 クイーン
$q_{init}$	$\frac{31}{64}\lambda$	$\frac{2}{64}\lambda$	$\frac{26}{64}\lambda$	$\frac{27}{64}\lambda$	$\frac{4}{64}\lambda$
$q_{final}$	$\frac{1}{64}\lambda$	$\frac{2}{64}\lambda$	$\frac{3}{64}\lambda$	$\frac{1}{64}\lambda$	$\frac{1}{64}\lambda$

これを拡張する形で APC-SCA を実装した. APC-SCA の評価のために, 4 種類の異なる最大カット問題 (G22, G30, G32, および G35) [16] と, 32 クイーン問題を用いた. 各問題の概要と各問題で使用したパラメータを, それぞれ表 1 と表 2 に示す.

比較には, 事前に  $q(s)$  のスケジューリングを決定する従来の SCA を使った. パラメータは式 (4) と式 (5) で規定される  $32 \times 32$  のグリッドサーチの中で最適であった組み合わせを用いる. 使用した  $q_{init}$  と  $q_{final}$  の組を表 3 に示す. 以降, グリッドサーチによって探索した最適なスケジューリングを使用したものを“調整済 SCA (fine-tuned SCA)”と呼ぶことにする.

##### 4.1 $r_q$ を変化させたときの解の品質の評価

$q_i$  の減少率  $r_q$  を 0.0 から 1.0 の間で変化させながら, APC-SCA で 5 つの組合せ最適化問題を解いた結果を図 6(a), 図 6(b), 図 6(c), 図 6(d), および 図 6(e) に示す. 結果は 128 回の平均イジングエネルギーを表している. 下限には  $q_{limit} = 0$  を使用した. 緑色の線で示した破線は調整済

表 4: 実行時間とイジングエネルギーの関係を評価するためのパラメータ設定.

	SA	調整済 SCA	APC-SCA
MC ステップ数	20 480	1 000	1 000
	40 960	2 000	2 000
	81 920	4 000	4 000
	163 840	8 000	8 000
	327 680	16 000	16 000
	655 360	32 000	32 000
	1 310 720	64 000	64 000
	2 621 440	128 000	128 000
	5 242 880	256 000	256 000

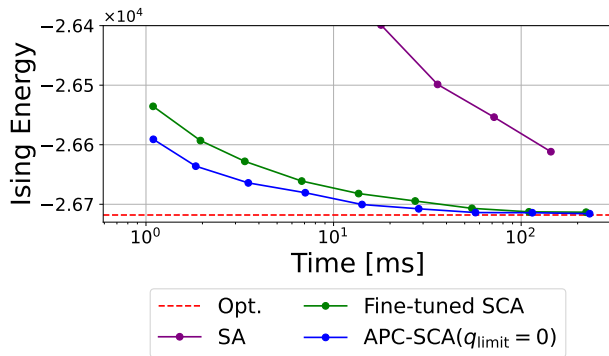


図 7: MC ステップ数を変化させながら G22 を解いた時の平均イジングエネルギーの比較. APC-SCA は SA や調整済 SCA と比較して, 短時間でより良い解が得られることがわかる.

SCA を使って実行したときのエネルギーを表している. 結果から, 今回使用したデータセットでは  $r_q \in [0.5, 0.7]$  の範囲で調整済 SCA よりも低いエネルギーを得られていることがわかる. したがって,  $r_q = 0.6$  と設定すれば, 組合せ最適化問題の種類によらず効率的に解探索可能なアニーリングを実現できる.

#### 4.2 解の品質と実行時間の評価

G22 について,  $r_q = 0.6, q_{\text{limit}} = 0$  というパラメータを使い, 表 4 に示すような複数の MC ステップ数で実行し, その実行時間とイジングエネルギーを評価した. 図 7 は実行時間とエネルギーの関係を表したグラフである. この実行時間は 128 回分のアニーリング計算を可能な限り並列に実行し, 全体にかかった実行時間を 128 で割り, 1 回の試行にかかる時間を算出したものである. 比較のため, APC-SCA の他に SA と調整済 SCA でも表 4 に示すステップ数を使って実行した. SA の実装は先行研究 [17] のものを APC-SCA で使用したものと同一 GPU 上で実行した.

図 7 より, APC-SCA が調整済 SCA と SA のどちらよりも低いエネルギーを求められていることがわかる. ま

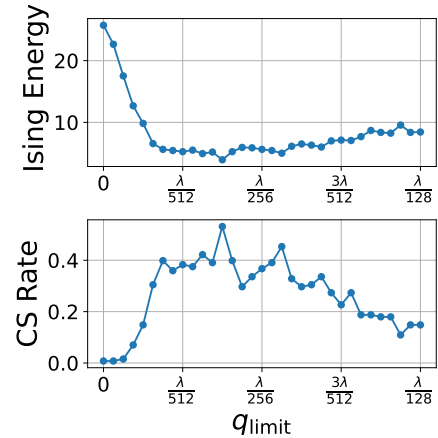


図 8:  $q_{\text{limit}}$  を 0 から  $\lambda/128$  まで変化させながら, APC-SCA で 32 クイーンを解いた時の平均イジングエネルギー (上図) と制約充足率 (下図).  $q_{\text{limit}}$  として正の小さな値を用いることで, 解の質を大幅に改善することができる.

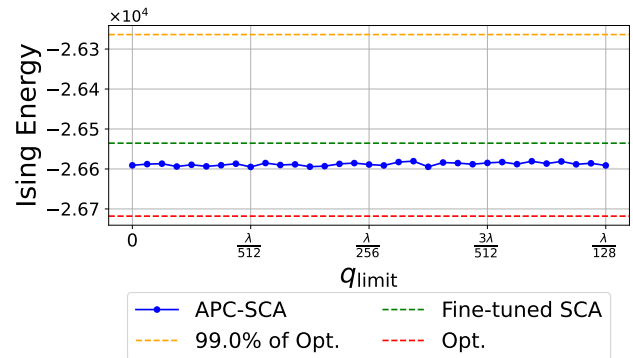


図 9:  $q_{\text{limit}}$  を 0 から  $\lambda/128$  まで変化させながら, APC-SCA で G22 を解いた時の平均イジングエネルギー. この問題では, 小さな  $q_{\text{limit}}$  を使用しても解の質にはほとんど影響がない.

た, 調整済 SCA と APC-SCA の実行時間を比較すると, APC-SCA におけるピンニングパラメータの自律制御が少ないオーバーヘッドで実現できていることがわかる. これは, 各  $q_i$  が独立して更新されるために, その更新手順 (アルゴリズム 2 の 10-13 行目) を容易に並列化できるためである.

#### 4.3 更なる解の品質の向上

上記の実験では, APC-SCA の評価を  $q_{\text{limit}} = 0$  で行ったが,  $q_{\text{limit}} > 0$  とすることで, 解の質が向上する可能性がある. 図 8 は  $r_q = 0.6$  で  $q_{\text{limit}}$  を 0 から  $\lambda/128$  の範囲で変化させて APC-SCA で 32 クイーンを解いた結果, 図 9 は同じ条件で G22 を解いた結果である. 図 8 の上段と下段はそれぞれ平均イジングエネルギーと制約充足率である. この制約充足率は問題を  $M = 128$  回解いたうち, 最適解 ( $H(\sigma) = 0$ ) を得られた割合である. N クイーン問

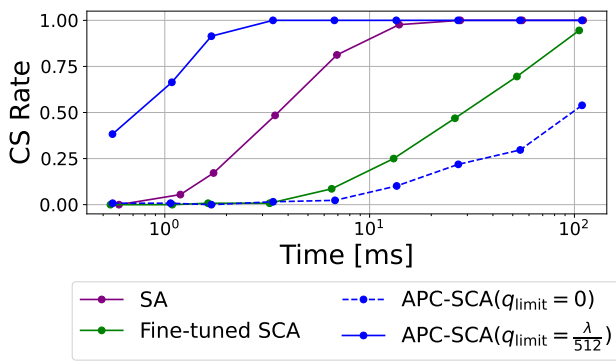


図 10: MC ステップ数を変化させながら、32 クイーンを解いた場合の制約充足率の比較.  $q_{\text{limit}} = \lambda/512$  の APC-SCA は SA や調整済 SCA よりも短時間で良質な解が得られることがわかる.

題では  $q_{\text{limit}} > 0$  を用いることにより解の質が向上する一方で、G22 では小さな  $q_{\text{limit}}$  を与えても解品質にほとんど影響がない.

4.1 節の実験から、 $r_q = 0.6$  の APC-SCA は、 $q_{\text{limit}} = 0$  を使ったとき 5 つの問題全てで調整済 SCA よりも優れた結果を示すことが確認された. したがって、APC-SCA はピンニングパラメータを適切かつ自律的に制御するという目的を既に達成している. 一方、特定の問題 (32 クイーンなど) では、 $q_{\text{limit}}$  として小さな正の値を用いることで、APC-SCA の性能を向上させることができる. 実用的には、 $q_{\text{limit}} = \lambda/512$  が 32 クイーン以外の問題 (図 9 で示した G22 など) の解品質を損なわないため、APC-SCA の共通設定として使用することができる.

図 10 は、 $r_q = 0.6$ 、 $q_{\text{limit}} = 0$  または  $\lambda/512$  で 32 クイーンを APC-SCA で解いた場合の実行時間と解の品質の関係を示したものである. 比較のため、同じ問題を SA と調整済 SCA でも解いた. ここでも、MC ステップ数を図 4 のように変化させた. APC-SCA を使った 2 つの結果を比較すると、 $q_{\text{limit}} = \lambda/512$  を使用することで解の質を向上させることがわかる. 結果から、APC-SCA の解品質は SA や調整済 SCA よりも優れていることがわかった.

## 5. まとめと今後の課題

SCA における自律的なピンニングパラメータ制御を可能とするアニーリングアルゴリズムとして、APC-SCA を考案した. APC-SCA は  $i$  番目のスピンのフリップ情報に基づいて個々のピンニングパラメータ  $q_i$  を更新することで、自律的な制御を実現する. この独立した  $q_i$  の更新により、APC-SCA は従来の SCA に対して大幅な変更を加えることなく GPU 上に実装することができた. 最大カット問題や  $N$  クイーン問題での実験を通して、APC-SCA はピンニングパラメータ制御の設定が全問題で共通して使用で

きることを確認した. これにより、ピンニングパラメータを個々の問題に対してあらかじめ最適化する必要のある従来の SCA よりも優れていることが確認できた. また、最大カット問題や 32 クイーン問題において、APC-SCA が SA や SCA よりも短時間で良質な解を得ることができた.

今後の方向性として、提案手法をより多くの組合せ最適化問題で評価し、今回得られた知見がどの程度一般的であるかを確認する必要がある. さらに、APC-SCA を動作原理とする並列アニーリングハードウェアアーキテクチャを設計することも、意義のある研究方針であると考えられる.

SCA のような並列スピン更新アルゴリズムは、組合せ最適化問題を効率的に解く際には非常に強力なアニーリング手法である. しかし、それらのアルゴリズムで得られる並列性を代償として、本来の SCA におけるピンニングパラメータ  $q$  のように、基底状態への収束に向けてスピン反転を制御下に置くための何らかの手段を持たなければならない. 個々のスピン反転に対する局所的なフィードバックという APC-SCA のキーコンセプトは、今後登場するであろう他の並列スピン更新アニーリング手法にも有用であると考えられる.

**謝辞** 本研究の一部は JST CREST JPMJCR18K3 の支援を受けて実施したものである.

## 参考文献

- [1] L. Wei, Z. Zhang, D. Zhang, and S. C. Leung, "A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints," *European Journal of Operational Research*, vol. 265, no. 3, pp. 843–859, 2018.
- [2] A. M. Fathollahi-Fard, K. Govindan, M. Hajiaghayi-Keshmeli, and A. Ahmadi, "A green home health care supply chain: New modified simulated annealing algorithms," *Journal of Cleaner Production*, vol. 240, pp. 1–19, 2019.
- [3] W. Zhang, A. Maleki, M. A. Rosen, and J. Liu, "Optimization with a simulated annealing algorithm of a hybrid system for renewable energy including battery and hydrogen storage," *Energy*, vol. 163, pp. 191–207, 2018.
- [4] M. C. Aguitoni, L. V. Pavão, and M. Antonio da Silva Sá Ravagnani, *Energy*.
- [5] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk *et al.*, "Quantum annealing with manufactured spins," *Nature*, vol. 473, no. 7346, pp. 194–198, 2011.
- [6] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Physical Review E*, vol. 58, pp. 5355–5363, Nov 1998.
- [7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [8] 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, no. 48.

- [9] T. Okuyama, T. Sonobe, K. Kawarabayashi, and M. Yamaoka, “Binary optimization by momentum annealing,” *Phys. Rev. E*, vol. 100, pp. 1–9, Jul 2019. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevE.100.012111>
- [10] B. H. Fukushima-Kimura, S. Handa, K. Kamakura, Y. Kamijima, and A. Sakai, “Mixing time and simulated annealing for the stochastic cellular automata,” *arXiv preprint arXiv:2007.11287*, 2021.
- [11] K. Yamamoto, K. Kawamura, K. Ando, N. Mertig, T. Takemoto, M. Yamaoka, H. Teramoto, A. Sakai, S. Takamaeda-Yamazaki, and M. Motomura, “STATICA: A 512-spin 0.25M-weight annealing processor with an all-spin-updates-at-once architecture for combinatorial optimization with complete spin–spin interactions,” *IEEE Journal of Solid-State Circuits*, vol. 56, no. 1, pp. 165–178, 2021.
- [12] A. Lucas, “Ising formulations of many NP problems,” *Frontiers in Physics*, vol. 2, no. 5.
- [13] D. Venturelli, D. Marchand, and G. Rojo, “Job shop scheduling solver based on quantum annealing,” in *Proc. of ICAPS-16 Workshop on Constraint Satisfaction Techniques for Planning and Scheduling (COPLAS)*, 2016, pp. 25–34.
- [14] K. Tamura, T. Shirai, H. Katsura, S. Tanaka, and N. Togawa, “Performance comparison of typical binary-integer encodings in an Ising machine,” *IEEE Access*, vol. 9, pp. 81 032–81 039, 2021.
- [15] K. Ikeda, Y. Nakamura, and T. S. Humble, “Application of quantum annealing to nurse scheduling problem,” *Scientific Reports*, vol. 9, no. 1, pp. 1–10, 2019.
- [16] Y. Ye, “Index of / yyye/yyye/Gset,” 2003. [Online]. Available: <https://web.stanford.edu/~yyye/yyye/Gset/>
- [17] K. Kawamura, K. Okawa, G. Gutmann, T. V. Chu, J. Yu, and M. Motomura, “GPU-based acceleration of fully parallel annealing algorithm for combinatorial optimization,” in *Proc. of 34th IEEE International System-on-Chip Conference (SOCC)*, 2022, pp. 1–6.