SWAQ

特定の問題にちょっと強くなった量子アニーリングシミュレーター

岡田颯斗

大阪府立四條畷高等学校

目次

自己紹介

どうもこんにちは

- 名前：岡田颯斗（高校 3 年生）
- 趣味・興味：
  - 競技数学
  - 量子コンピュータ

前提

量子アニーリングは組合せ最適化問題を解く手段の一つ

よく扱われる問題の例：

彩色問題

- 隣り合う場所は異なる色で塗分ける

巡回セールスマン問題

- 複数の街を最短経路ですべて訪れる

彩色問題

彩色問題：リンク

だがしかし

量子アニーリングは組合せ最適化問題を解く手段の一つ

彩色問題

- 隣り合う場所は異なる色で塗分ける

巡回セールスマン問題

- 複数の街を最短経路ですべて訪れる

だがしかし

量子アニーリングは組合せ最適化問題を解く手段の一つ

彩色問題

- 隣り合う場所は異なる色で塗分ける

巡回セールスマン問題

- 複数の街を最短経路ですべて訪れる

しかし、量子アニーリングは最適化問題を効率よく解けるかというと...

背景

量子アニーリングは m 個の中から n 個選ぶのが苦手

なぜなら...

- 制約がペナルティ項として目的関数につけられてしまう
- すると問題が非本質な方向へ最適化される

$$\begin{aligned} \text{minimize} \quad & H_{object} \\ \text{subject to} \quad & H_{constraint} = c \end{aligned}$$

量子アニーリングは m 個の中から n 個選ぶのが苦手

なぜなら...

- 制約がペナルティ項として目的関数につけられてしまう
- すると問題が非本質な方向へ最適化される

$$\text{minimize} \quad \boldsymbol{H_{object} + \underbrace{(H_{constraint} - c)^2}_{H_{penalty}}}$$

式で見ると

彩色問題

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j \in Adj} \sum_{k \in color} q_{i,k} q_{j,k} \\
\text{subject to} \quad & \sum_{i \in vertics} q_{i,k} = 1
\end{aligned}
$$

$$\downarrow$$

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j \in Adj} \sum_{k \in color} q_{i,k} q_{j,k} \\
& + \sum_{k \in color} \left(\sum_{i \in Adj} q_{i,k} - 1 \right)^2
\end{aligned}
$$

巡回セールスマン問題

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j \in C} \sum_{k=0}^{n} w_{i,j} q_{i,k} q_{j,k+1} \\
\text{subject to} \quad & \sum_{i \in C} q_{i,k} = 1 \\
& \sum_{k=0}^{n} q_{i,k} = 1
\end{aligned}
$$

$$\downarrow$$

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j \in C} \sum_{k=0}^{n} w_{i,j} q_{i,k} q_{j,k+1} \\
& + \sum_{k=0}^{n} \left(\sum_{i \in C} q_{i,k} - 1 \right)^2 + \sum_{i \in C} \left(\sum_{k=0}^{n} q_{i,k} - 1 \right)^2
\end{aligned}
$$

つまり

量子アニーリングは組合せ最適化問題を解く手段の一つ

彩色問題

- 隣り合う場所は異なる色で塗分ける
- 各地点はちょうど一色で塗られる

巡回セールスマン問題

- 複数の街を最短経路ですべて訪れる
- 各時間に訪れることのできるサイトはちょうど 1 か所
- 各地点ちょうど 1 回訪れる

つまり

量子アニーリングは組合せ最適化問題を解く手段の一つ

彩色問題

- 隣り合う場所は異なる色で塗分ける
- 各地点はちょうど一色で塗られる

巡回セールスマン問題

- 複数の街を最短経路ですべて訪れる
- 各時間に訪れることのできるサイトはちょうど 1 か所
- 各地点ちょうど 1 回訪れる

これらの制約を常に満たすように解を遷移させていけば効率が上がりそう

デモ

彩色問題（再）

エネルギーが下がっていく様子を付けた 20×20 の彩色問題の動画

手法

簡単な例：彩色問題

$q_{i,j}$	場所 1	場所 2	場所 3	場所 4	場所 5
色 1	0	0	1	0	0
色 2	1	0	0	1	0
色 3	0	0	0	0	1
色 4	0	1	0	0	0

↑ ↑ ↑ ↑ ↑

各列 1 はちょうど 1 つ

簡単な例：彩色問題

$q_{i,j}$	場所 1	場所 2	場所 3	場所 4	場所 5
色 1	0	0	1	0	0
色 2	1	0	0	1	0
色 3	0	0	0	0	1
色 4	0	1	0	0	0

↑ ↑ ↑ ↑ ↑

各列 1 はちょうど 1 つ

その次くらいに簡単な例：巡回セールスマン

$q_{i,j}$	場所 1	場所 2	場所 3	場所 4	場所 5
順番 1	0	0	1	0	0
順番 2	1	0	0	0	0
順番 3	0	0	0	0	1
順番 4	0	0	0	1	0
順番 5	0	1	0	0	0

↑ ↑ ↑ ↑ ↑

各列 1 はちょうど 1 つ

その次くらいに簡単な例：巡回セールスマン

$q_{i,j}$	場所 1	場所 2	場所 3	場所 4	場所 5
順番 1	0	0	1	0	0
順番 2	1	0	0	0	0
順番 3	0	0	0	0	1
順番 4	0	0	0	1	0
順番 5	0	1	0	0	0

← 各
← 行
← も
← 同
← 様

その次くらいに簡単な例：巡回セールスマン

$q_{i,j}$	場所 1	場所 2	場所 3	場所 4	場所 5
順番 1	0	0	1	0	0
順番 2	1	0	0	0	0
順番 3	0	0	0	0	1
順番 4	0	0	0	1	0
順番 5	0	1	0	0	0

比較

比較

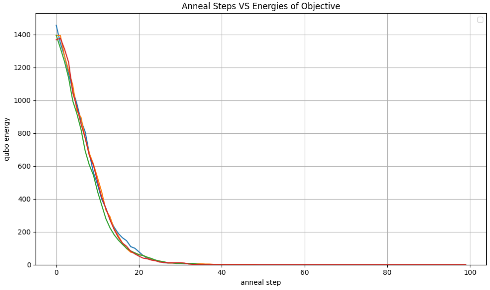


図 1: Swap based の QAS (lowest = 0)

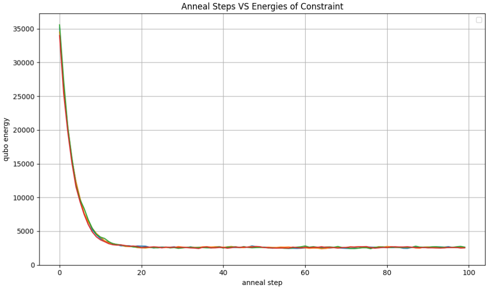
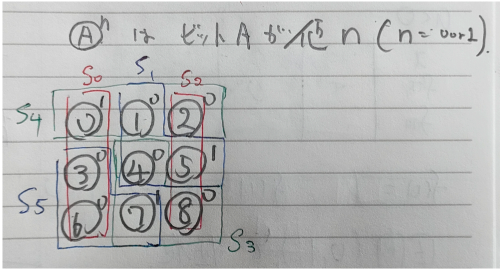


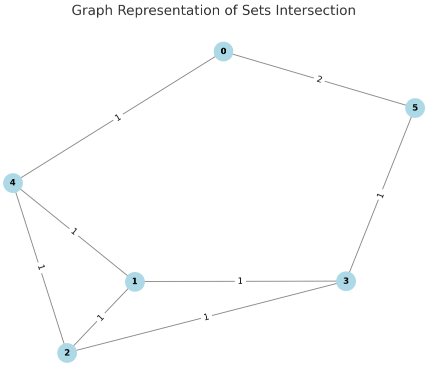
図 2: 普通の QAS (lowest = 2630)

※縦軸の目盛りが異なることに注意

展望

一般化をしたい





こんな感じの図をきれいにつくる

参考文献