

アニメーリングマシンで競技プログラミング の問題を解いてみた

競技プログラミングとは

- ・プログラミングを用いて数学/パズル的な問題を解く競技。
- ・近年ではプログラミング教育において競技型学習が注目されており、学生の採用活動などにも用いられることから参加者が増加している。
日本のサイトでは、コンテスト参加者数がこの8年間で300人→8000人へ
- ・競技プログラミングの問題をアニーリングマシンで解くことで...
 - 何かアイデアが得られるのでは
 - 競技プログラマが興味を持つことで量子アニーリング人材の確保が可能となるのでは

今回使用した問題

AtCoder Regular Contest 001 B問題 リモコン
(https://atcoder.jp/contests/arc001/tasks/arc001_2)

- ・エアコンの設定温度をA度からB度に変更したい($0 \leq A \leq 40$, $0 \leq B \leq 40$)
- ・エアコンのリモコンは1回ボタンを押すことで、 ± 1 度、 ± 5 度、 ± 10 度 設定温度を変更できる
(つまり、6種類の操作)
- ・設定温度をA度からB度に変更するために押すボタンの最小回数を求める

入力:A, B 出力:最小回数

問題例

A=7度, B=34度するとき

- ・10度、10度、5度、1度、1度とリモコンを操作すると5回で設定することができる。
- ・実際の競技プログラミングでは、幅優先探索などのアルゴリズムを用いて答えを求める。

定式化

QUBO

$$H = A \left(\sum_{j=1}^N \sum_{i=1}^R d_i x_{ij} - T \right)^2$$

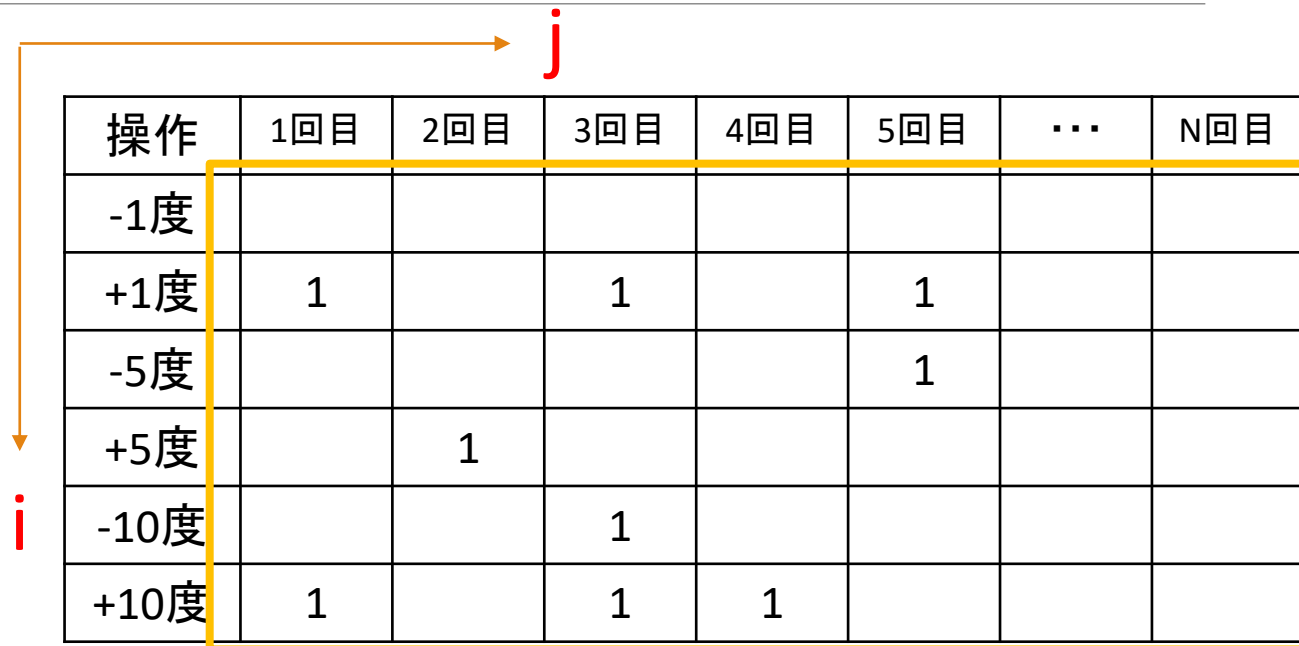
d_i : リモコンの*i*番目の操作による温度変化
 x_{ij} : *j*回目にリモコンの*i*番目の操作をするかどうか

・操作後の温度が $T(=B-A)$ に等しい

ここまでは確定

決定問題(温度をA度からB度にできるか)ならこれで良いが、
最小回数は求まらない。

例えば、+10と-10は打ち消されるので、余分に選ばれる可能性がある



操作	1回目	2回目	3回目	4回目	5回目	...	N回目
-1度							
+1度	1		1		1		
-5度					1		
+5度		1					
-10度			1				
+10度	1		1	1			

1の個数を数える→操作回数

定式化

QUBO

$$H = A \left(\sum_{j=1}^N \sum_{i=1}^R d_i x_{ij} - T \right)^2 + B \sum_{j=1}^N \left(\sum_{i=1}^R x_{ij} - 1 \right)^2 - C \sum_{j=1}^N x_{0j}$$

- ・操作後の温度が $T(=B-A)$ に等しい
- ・1回の操作で1度しか温度を変更しない
- ・0度を選んだらちょっとだけ報酬を与える

操作	1回目	2回目	3回目	4回目	5回目	...	N回目
±0度						1	1
-1度							
+1度			1		1		
-5度							
+5度		1					
-10度							
+10度	1			1			

1の個数を数える→最小回数

0度(操作しない)もQUBOに入れることで、
1度のアニーリング計算によって最適解を求めるアプローチ
(Nは十分大きくとる必要があることに注意)

計算結果

A=7度, B=34度するとき

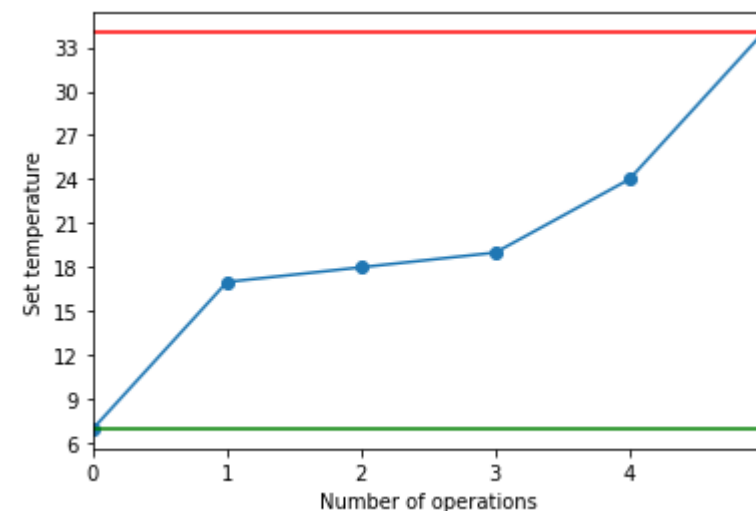
5回の操作で7度→34度
になっている

結果 (0℃変更含む)

[0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1] 0℃変更
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -1℃変更
[0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0] 1℃変更
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -5℃変更
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0] 5℃変更
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -10℃変更
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1] 10℃変更

結果 (0℃変更含まない)

[0, 0, 0, 0, 0] -1℃変更
[0, 1, 1, 0, 0] 1℃変更
[0, 0, 0, 0, 0] -5℃変更
[0, 0, 0, 1, 0] 5℃変更
[0, 0, 0, 0, 0] -10℃変更
[1, 0, 0, 0, 1] 10℃変更



問題文の範囲 ($0 \leq A \leq 40$, $0 \leq B \leq 40$) において答えを求めることができる

計算結果

A=-39度, B=38度するとき

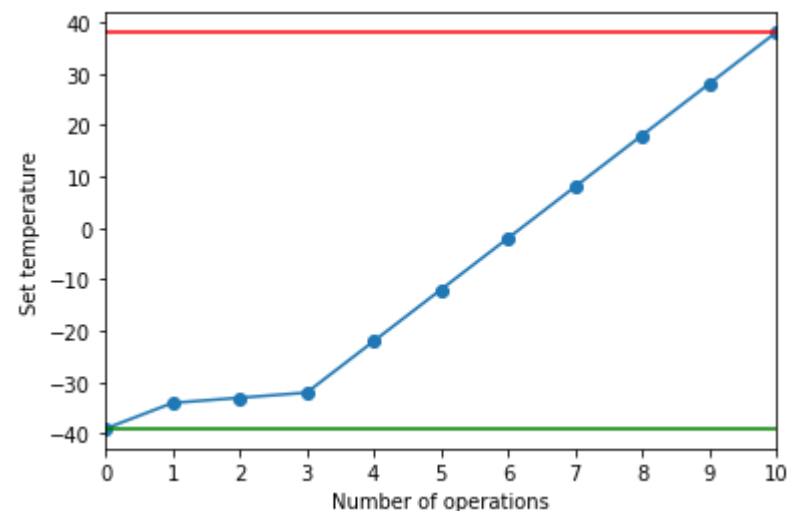
10回の操作で-39度→38度
になっている

結果 (0℃変更含む)

[0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0] 0℃変更
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -1℃変更
[0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0] 1℃変更
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -5℃変更
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] 5℃変更
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -10℃変更
[0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1] 10℃変更

結果 (0℃変更含まない)

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -1℃変更
[0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0] 1℃変更
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -5℃変更
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] 5℃変更
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] -10℃変更
[0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1] 10℃変更



問題文の範囲より少し広くても大丈夫

工夫した点・苦労した点

・工夫した点

Nを十分大きく取ることによって、1回のアニーリング計算で決定問題が求まるようにした
0度の操作を追加し、0度を選んだことに対し報酬を与えることによって、
操作の最小回数が求まるようにした

・苦労した点

とにかく環境構築 WSL環境にてinstallしたもののimportでエラーが...

```
~: $ sudo python3 test.py
Traceback (most recent call last):
  File "test.py", line 10, in <module>
    from amplify import gen_symbols, BinaryPoly
ModuleNotFoundError: No module named 'amplify'
```

環境構築

・どこがおかしかったか

Amplifyのインストール先が、
他のモジュールと異なっていた



他のモジュールと同じフォルダに
移すことで事なきを得た...(2021/3/27)

```
~$ pip show numpy
Name: numpy
Version: 1.20.1
Summary: NumPy is the fundamental package for array computing with Python.
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email: None
License: BCD
Location: /usr/local/lib/python3.8/dist-packages
Requires:
Required-by: matplotlib

~$ pip show amplify
Name: amplify
Version: 0.4.1
Summary: Middleware for Quantum and Classical Annealing Machines
Home-page: https://amplify.fixstars.com
Author: Yoshiki Matsuda
Author-email: y_matsuda@fixstars.com
License: UNKNOWN
Location: /home/~/local/lib/python3.8/site-packages
Requires:
Required-by:
```

その他の競技プログラミング問題

AtCoder Beginner Contest 073 D問題 joisino's travel
(https://atcoder.jp/contests/abc073/tasks/abc073_d)

指定されたR個の点を必ず通る全点对最短経路問題

→隣接していない2点間の距離を非常に大きい値にして巡回セールスマンと同様にする

(ワーシャルフロイド法のQUBO化)

AtCoder Beginner Contest 184 F問題 Programming Contest
(https://atcoder.jp/contests/abc184/tasks/abc184_f)

解くのに $A_i (1 \leq i \leq N)$ 時間かかる問題N問の中から、解くのにかかる時間の総和がT分以下になるように数問選択し、解くのにかかる時間の総和の最大値を求める。

(ナップサック問題の応用: 制約が大きい時に、QUBOの補助変数の係数として 2^i を用いるテクニック)

まとめ

- ・競技プログラミングの問題をQUBOに帰着させることにより、アニーリングマシンで解くことが可能となった。
- ・本来、本問題はアルゴリズム(幅優先探索やワーシャルフロイド法)により解くことを想定されている問題だが、温度以外にも変数が増えてリモコンで操作できる種類が増えた場合など、拡張されて複雑化した場合には量子アニーリングが威力を発揮する。
- ・本問題の「目標の数値になるまでリモコン操作により数値を調整する」という作業は工学的に頻出(ex.温度・座標)であり応用先が無数に存在する。

さいごに

競技プログラミングの問題には実問題を数学的に抽出したものがある

→アイデアの宝庫

拡張・複雑化すると量子アニーリングが力を発揮する可能性

両方を学ぶことによって相乗効果で実問題の解決が進展する

・競技プログラミングから量子アニーリングを見る→アルゴリズムの効率化

相乗効果 

・量子アニーリングから競技プログラミングを見る→QUBO化の工夫

