

Session5



量子・AIハイブリッドに関する実践的内容

2023年 2月28日

DEVEL株式会社 比嘉恵一朗

今回の目的

企業における量子技術の普及を加速させるため、
企業のビジネス課題への効果的な施策を提案できる量子産業人材の育成。

概要

この時間では量子アニーリングと古典AIについて、技術的な部分を中心とした説明を行う。
説明には簡単な数式までを用いることはしないが、計算の仕組みをイメージしてもらう。

最後は交通最適化の例を元に流れを明確化させる。

目次

[量子とは？](#)

[組み合わせ最適化問題とは？](#)

[機械学習について](#)

[量子アニーリングについて](#)

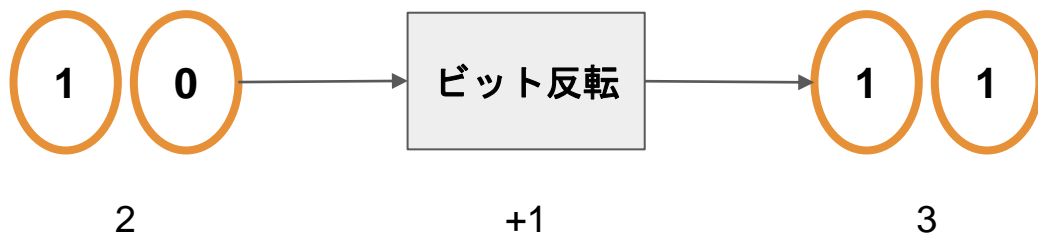
※ 量子アニーリングについてのセッションの際に Google Colab を使います。

量子とは？

計算方法が古典コンピュータ（通常のPC）とは異なる。

古典コンピュータの場合

例） $2+1$ を古典コンピュータ上で行う。



ビット (bit)

情報の単位。
ここでは2ビットを用いる。

古典コンピュータ上では各値を2進数に変換して計算を行う。

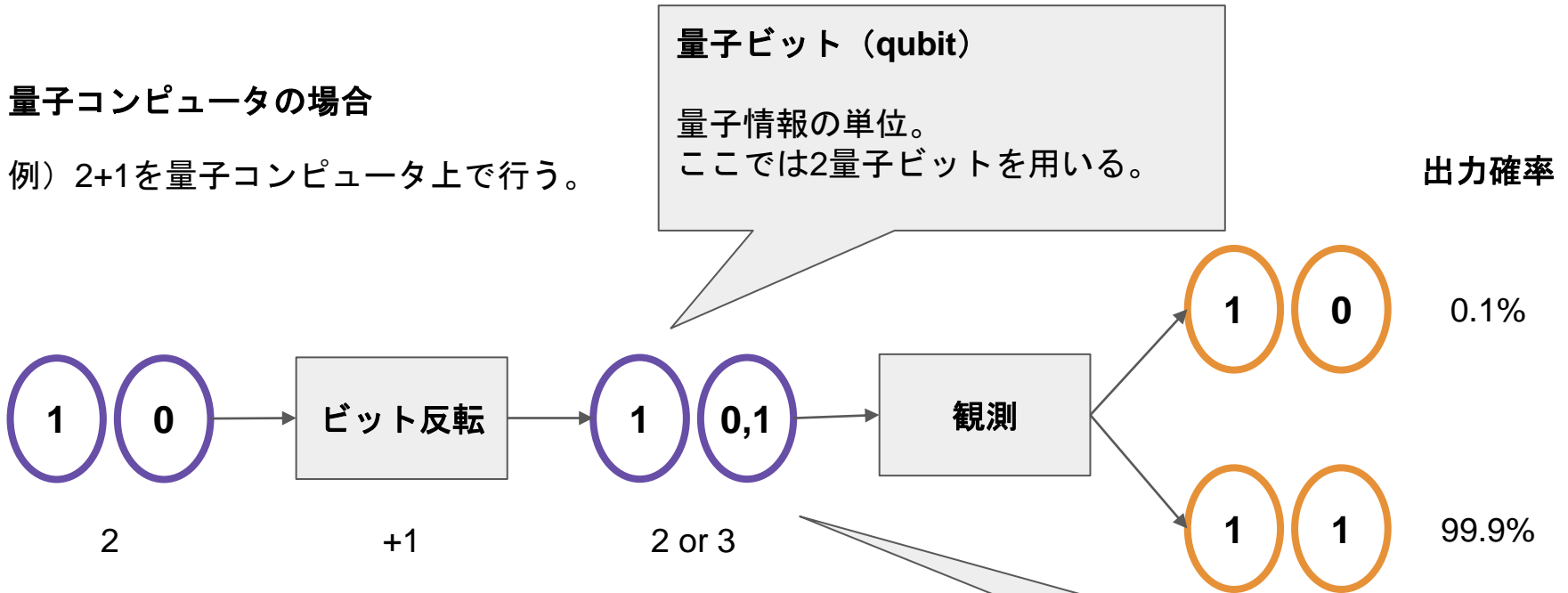
スイッチのように各ビットを切り替えて計算を行う。（ビット反転）

計算時間はこのビット反転に依存する。

量子とは？

量子コンピュータの場合

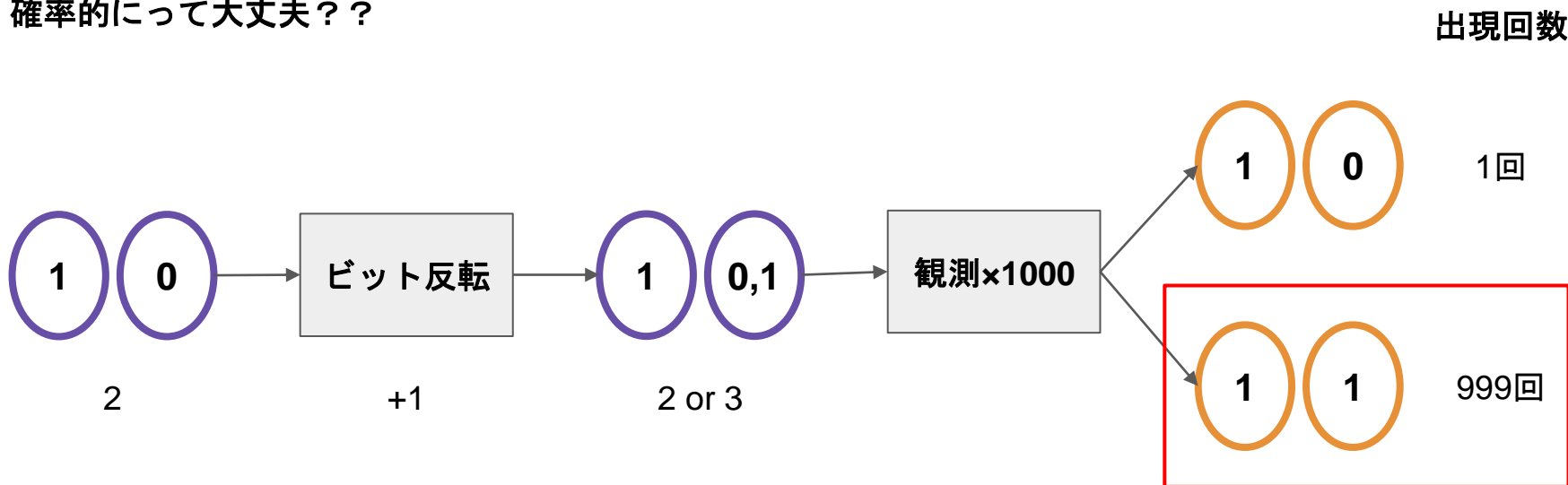
例) 2+1を量子コンピュータ上で行う。



量子コンピュータ上でも各値を2進数に変換して計算を行う。
古典コンピュータとの違いとして、観測という操作が存在する。
計算結果は確率的に出力される。

量子とは？

確率的になって大丈夫??



観測は1回だけ行う訳ではない。通常 1000 ~ 10000 回ほど行う。

複数回観測して最も出現するものを答えとする。

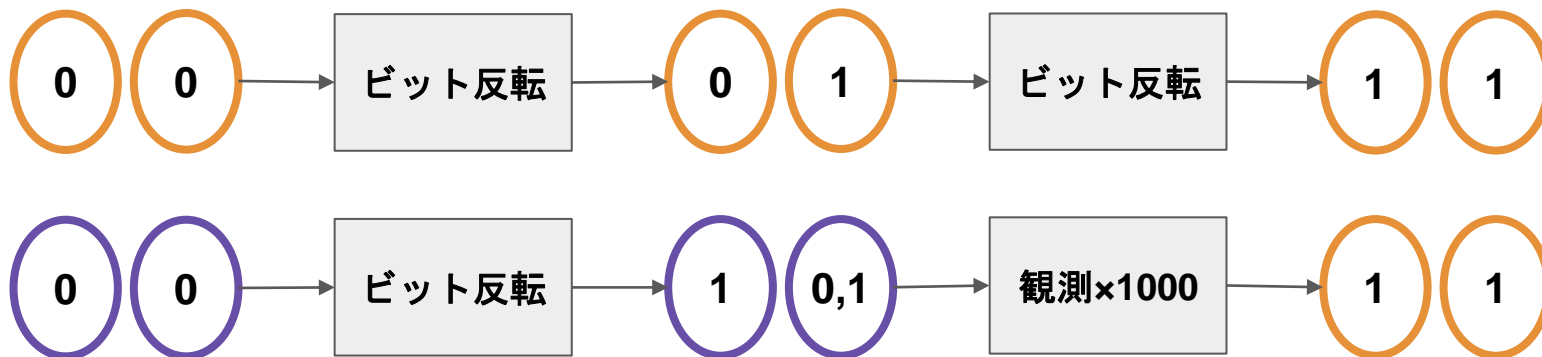
こちらを採用

量子とは？

何が嬉しいの？

例) 3 を作る。

※イメージです。実際の計算とは異なります。



古典コンピュータはスイッチのように一つずつビットを切り替える。

量子コンピュータは複数の量子ビットを一度に切り替える。

ビット反転の回数が量子コンピュータの方が少ない → 高速に処理をすることができる。

量子とは？（詳細）

量子コンピュータは並列処理が可能

古典コンピュータは1ビットに“0”, “1”の状態を持っており、計算する際には“0”, “1”のどちらか一方の状態にしかなることはできない。

一方で量子コンピュータの場合は1量子ビットに“0”, “1”の両方の状態を同時に持つことができる。

従って、“0”の状態の場合と“1”の状態の場合で計算をそれぞれ行いたいとすると...

→古典コンピュータでは**一方の状態しか持たない**ため、2回計算を行わないといけない。

→量子コンピュータでは**両方の状態を同時に持つ**ため、1回で計算を行うことが可能。

組み合わせ最適化問題とは？

一般的に最適化問題とは何かを考える。

例) 100円で10円のチョコと5円のガムを買う。

条件1

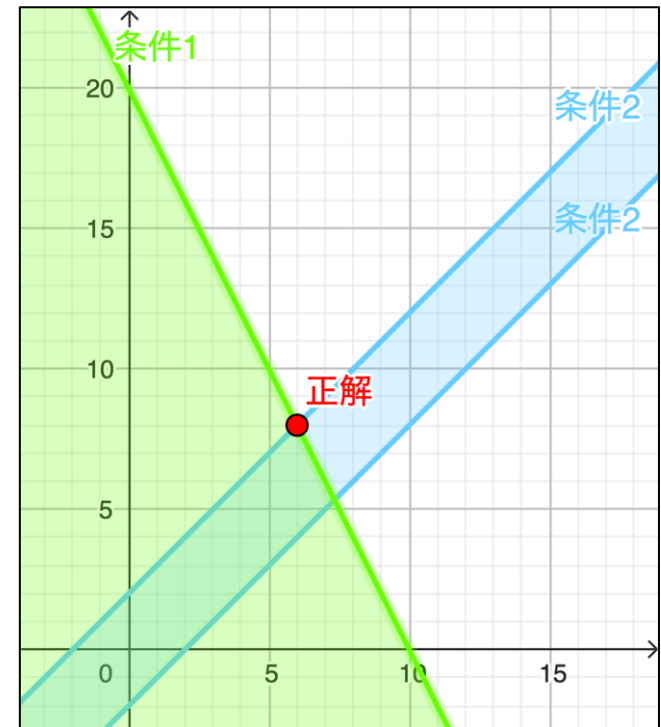
チョコとガムをできるだけ多く買いたい。(最大化)

条件2

チョコとガムの個数は同じくらいほしい。(制約条件)
(個数の差が2個以下)

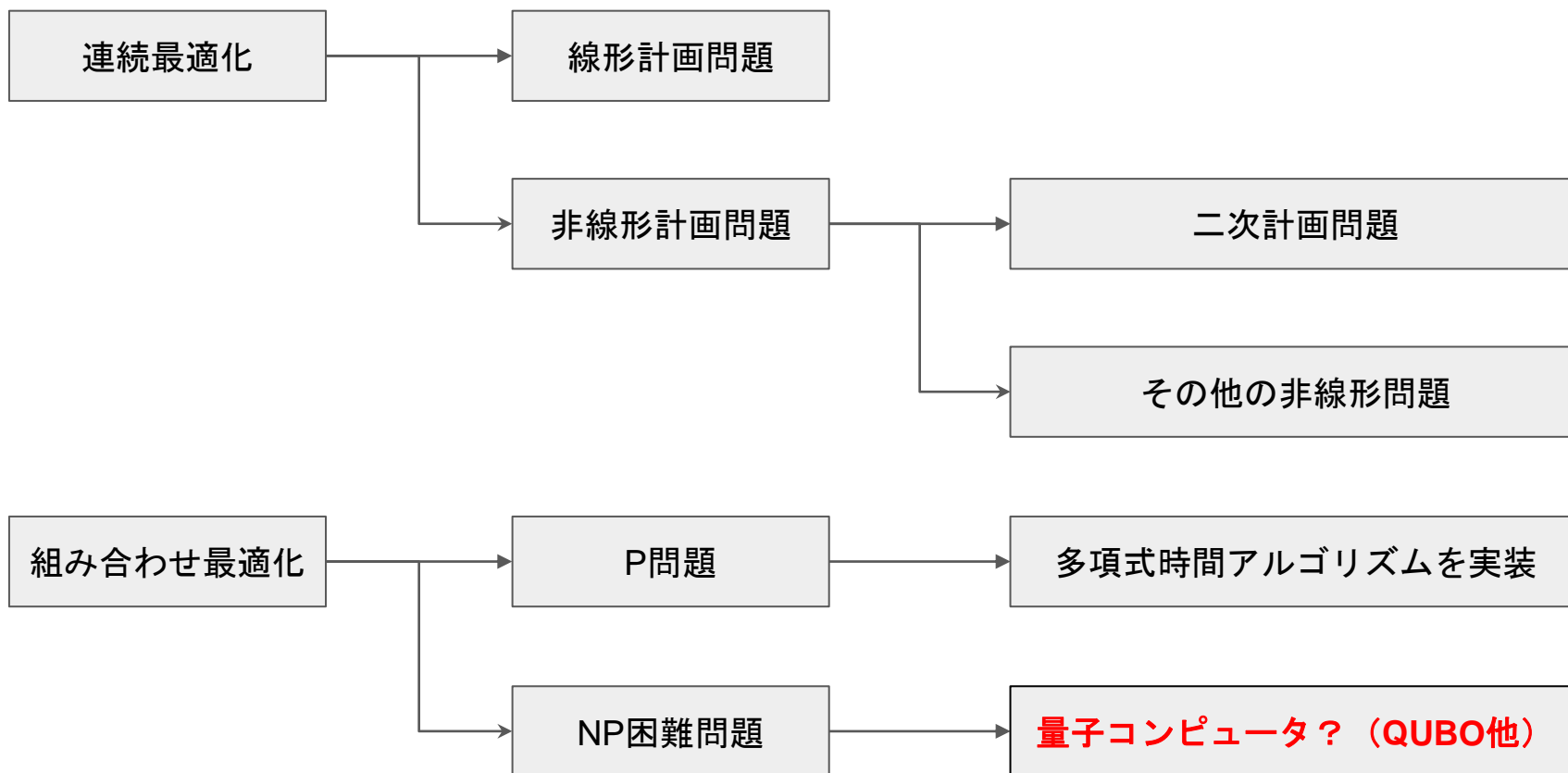
→右のグラフからチョコが6個ガムが8個となる。

このようにある条件下での最大・最小値を求める問題を最適化問題という。



組み合わせ最適化問題とは？

最適化問題の種類を以下にまとめる。



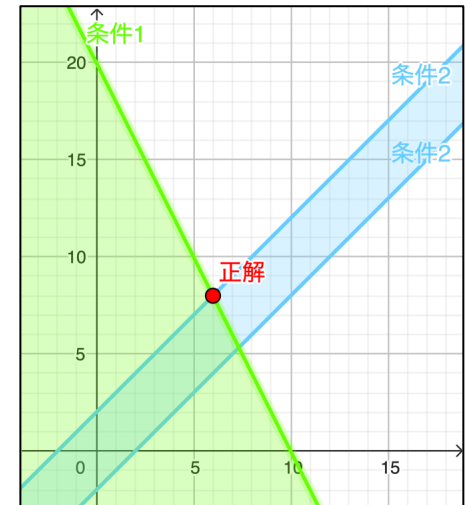
組み合わせ最適化問題とは？

最適化問題は大きく分けて、**連続最適化**と**組合せ最適化**（**離散最適化**）が存在する。

→扱う値が実数（**連続値**）か、整数（**離散値**）かの違い。
前者が連続最適化。後者が離散最適化。

例1) 100円で10円のチョコと5円のガムをできるだけ多く買う。
→買う個数は整数値なので、組み合わせ最適化。

例2) 材料Aと材料Bを数 g ずつ混ぜて一番美味しいチョコを作る。
→材料は連続値なので、連続最適化。



離散最適化

組み合わせ最適化問題とは？

連続最適化と組み合わせ最適化の解き方について

連続最適化の場合に適用可能な微分可能性に基づいたアルゴリズムは組合せ最適化問題には利用することができない。

→一般に組合せ最適化問題は連続最適化問題よりもはるかに難しい。

組合せ最適化問題の通常の見方

具体的な問題まで落とし込んだあとに、その問題に特化したアルゴリズムの実装によって解くことが基本的なアプローチとなる。

ただしこれらの問題に関しては厳密な効率的アルゴリズムは存在しないため、局所探索、近似解法あるいはメタヒューリスティクスな手法等の利用が検討される。

組み合わせ最適化問題とは？

QUBO (Quadratic Unconstrained Binary Optimization) 問題（離散）

量子アニーリングの文脈でよく登場する最適化問題。

目的関数が 2 次形式 (Quadratic) な問題でありかつ変数が 0,1 のいずれかしか取らない 2 値変数 (Binary) であり、かつ制約条件が存在しない (Unconstrained) 最適化問題のこと。

Ising 問題

QUBO問題の変数の取る値を 0,1 ではなく -1,1 にした問題。

量子アニーリングの文脈で最適化と言った場合、基本的に QUBO 問題を指す。

組み合わせ最適化問題とは？

QUBO 問題は制約条件を変数に要求することはできない。

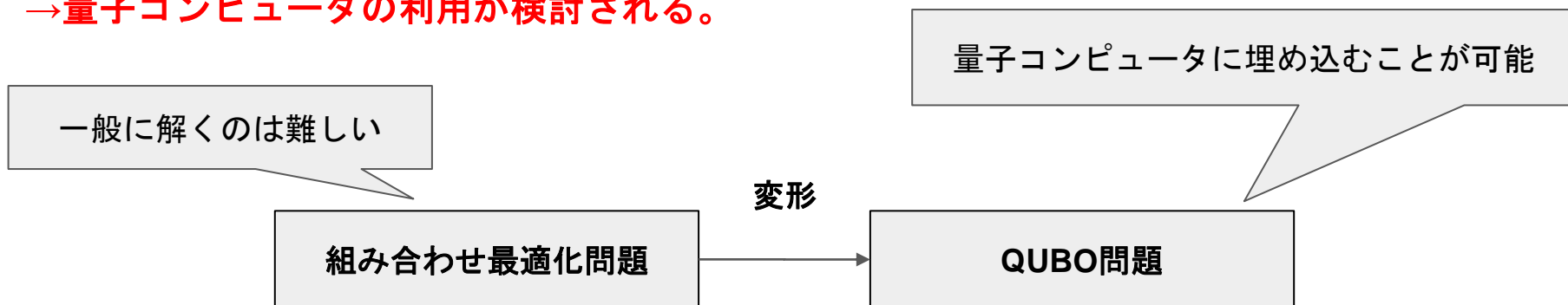
→制約が必要な場合は制約条件自体も目的関数の中に含ませる必要がある。

QUBO 問題として定式化できる社会的、学術的に重要な問題は多く知られている。

一方で変数の値が離散値である最適化問題は一般に解くのが難しい。

また、QUBO 問題として定式化されるいくつかの重要な問題は効率的なアルゴリズムが知られていない。

→量子コンピュータの利用が検討される。



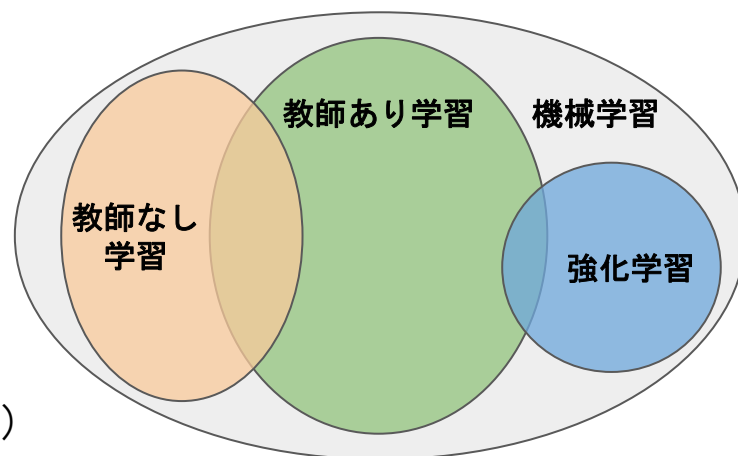
機械学習について

機械学習はデータからパターンを学習し、学習結果を応用する強いツールである。学習結果に基づいて、過去のデータから次の動きを予測したり分類できる。また、過去のデータの整備やデータにおける重要な要因などの分析を行うこともできる。

大まかには以下のものが存在する。

- ・ 教師あり学習（正解のあるデータを用いて学習）
→ 回帰分析、分類分析
- ・ 教師なし学習（正解のないデータを用いて学習）
→ クラスタリング
- ・ 強化学習（システム自身の行動と結果をもとに学習）

回帰分析については後半に詳しく説明する。



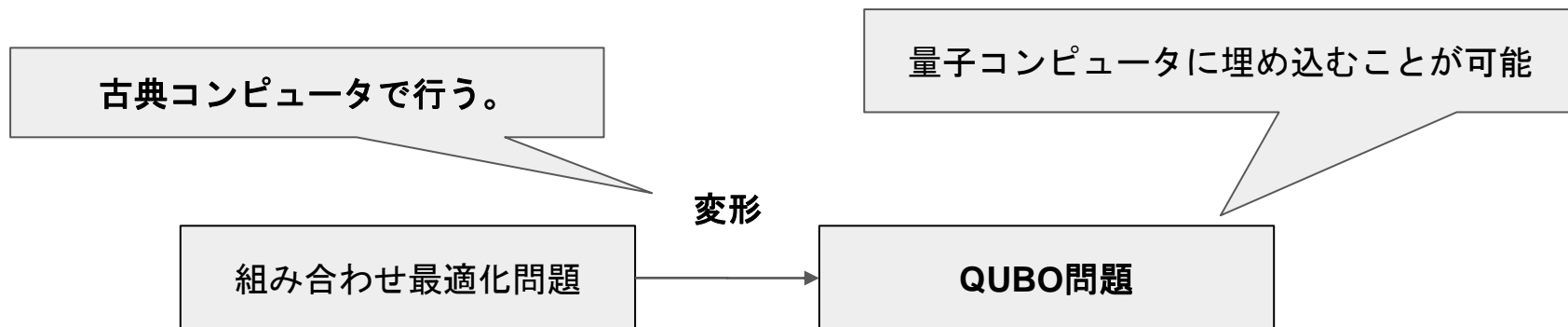
量子アニーリングについて

組み合わせ最適化問題に特化した量子コンピュータ

→基本的な構成要素は量子ゲート方式と同じく量子ビットであるが、実行のプロセスが異なっている。

量子アニーリングを行うために、組み合わせ最適化問題をQUBO問題に変換してやる必要がある。

組み合わせ最適化問題からQUBO問題への変換は古典コンピュータで行う。



量子アニーリングについて

利点

- ・汎用性の高さ

→量子アニーリングでは解きたい組み合わせ最適化問題を QUBO 問題に書き下すことができれば、メモリが許す限りどんな問題でも解析が可能である。

→量子ゲート方式のように回路設計などが不要であり実装が比較的容易。

欠点

- ・組み合わせ最適化問題の定式化が難しい。

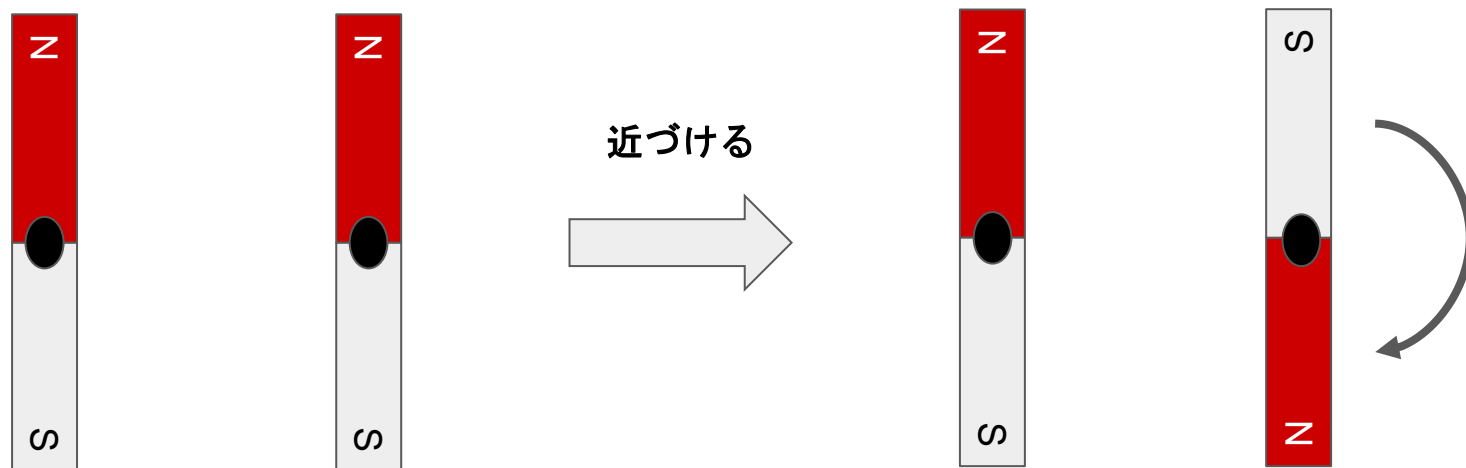
・ソフトウェアやライブラリ開発は進むが、組み合わせ最適化の定式化の知識がないとプログラミングは容易ではない。

出典：量子コンピュータの概要と動向 株式会社日本総合研究所

<https://www.jri.co.jp/MediaLibrary/file/column/opinion/pdf/11942.pdf>

量子アニーリングについて

計算の原理をイメージするために、磁石を例にして考える。
2つの磁石が縦に固定して並んでいる。



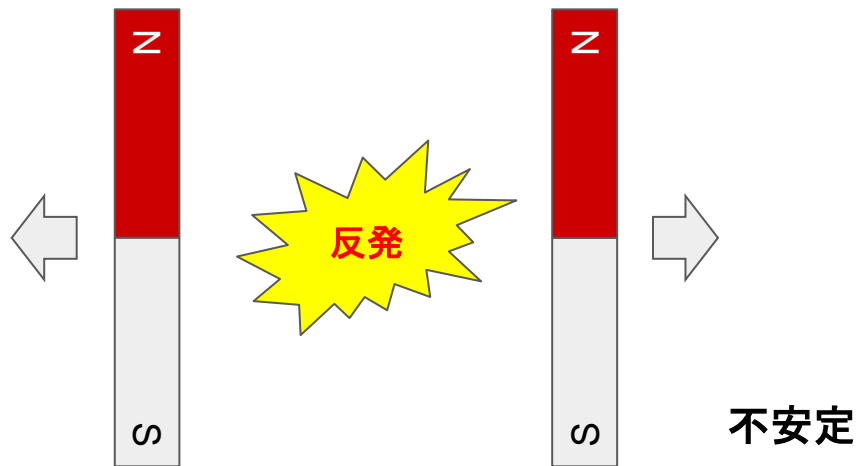
左側の磁石のN極を近づけるとどうなる？
→回転してS極が上になる。

量子アニーリングについて

同じ極同士を近づける。

→互いに反発しあう。

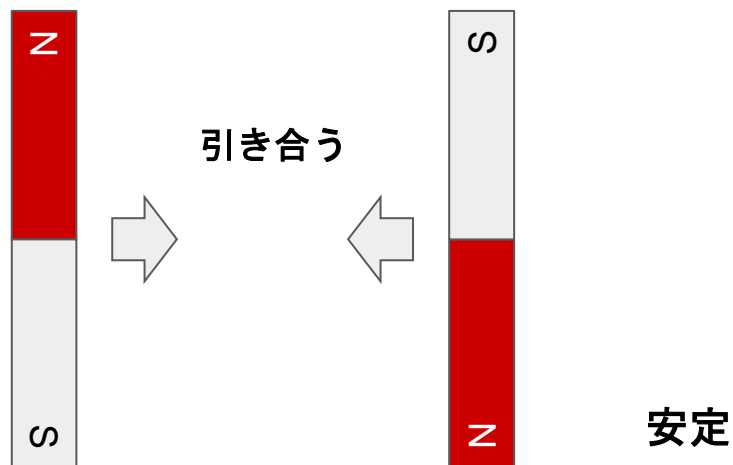
これは不安定な状態。



違う極同士を近づける。

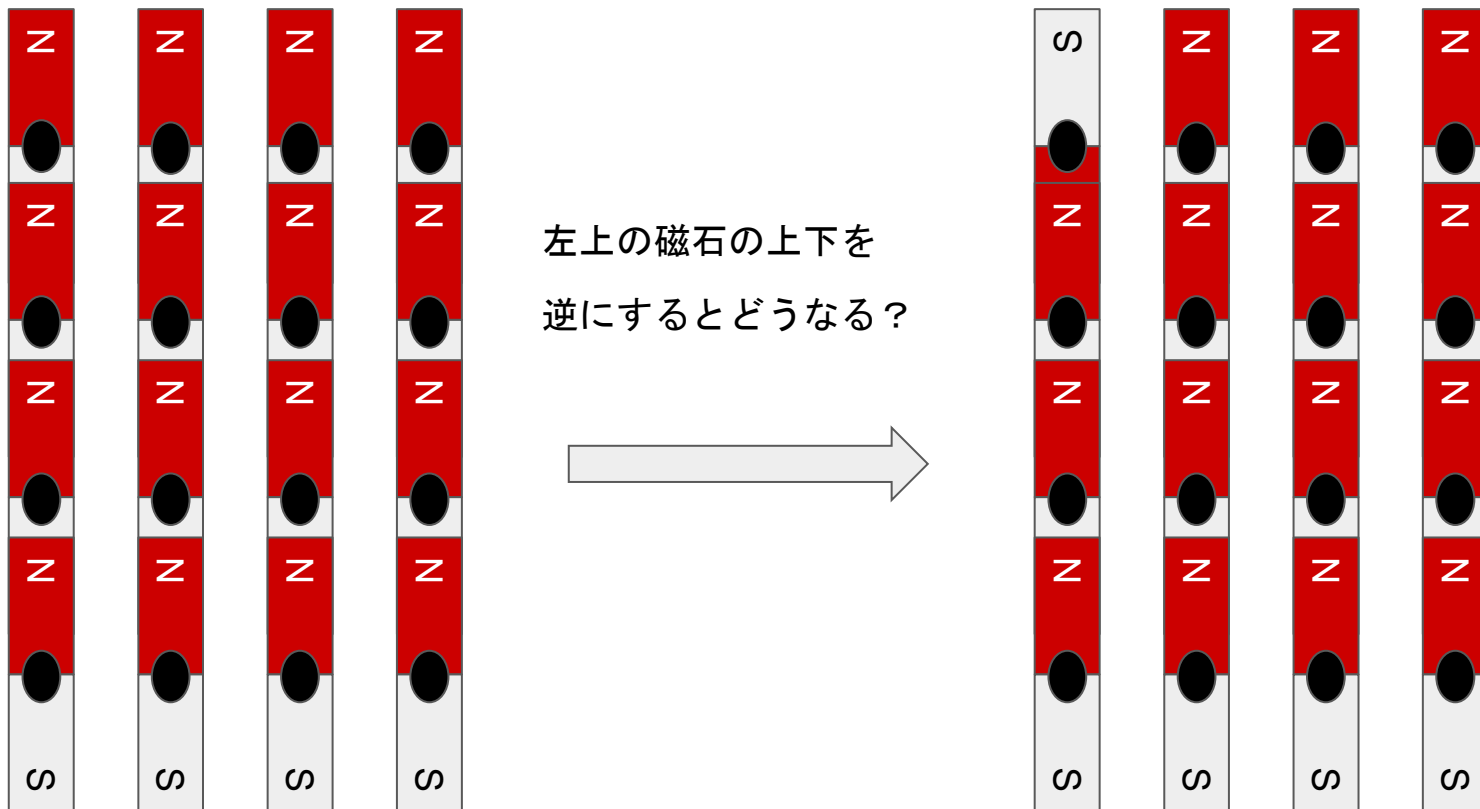
→互いに引き合う。

これは安定した状態。



量子アニーリングについて

これをいっぱい増やしてみる。



量子アニーリングについて

他の磁石が一斉に動く。

→一番安定した状態になろうとする。

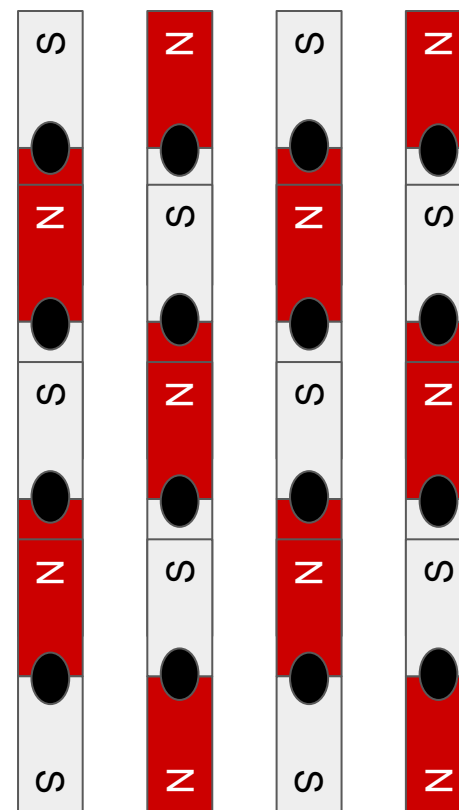
このように、磁石を動かして

安定した状態にさせる処理をアニーリングという。

(実際は磁石ではないのであくまでもイメージ)

また、一部の磁石を固定したり、

磁力の強弱をつけることもできる。



量子アニーリングについて

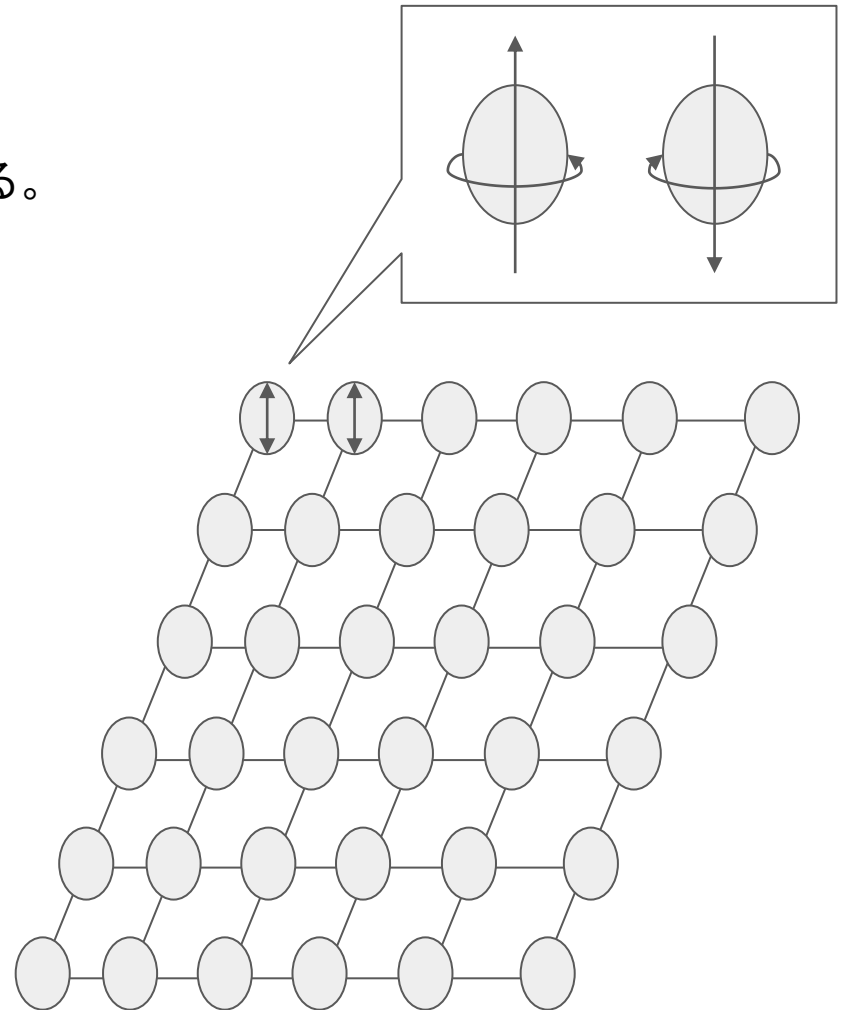
先ほどの話を実際の量子アニーリングで考える。

磁石一つ一つが電子に置き換わっている。

→ただしやっけることは磁石と同じ。

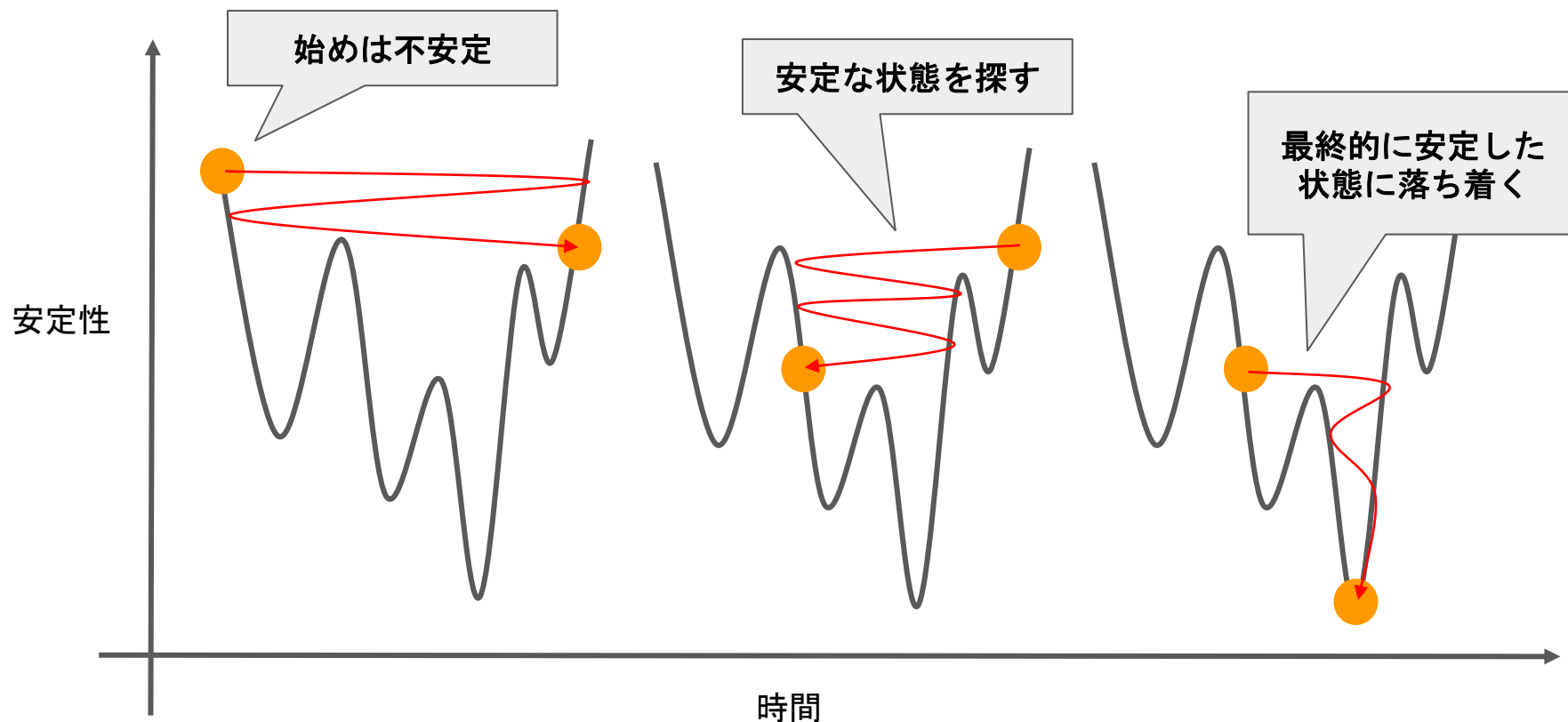
電子一個一個に向きがある。（スピン）

磁石のように安定な状態になろうとする。



量子アニーリングについて

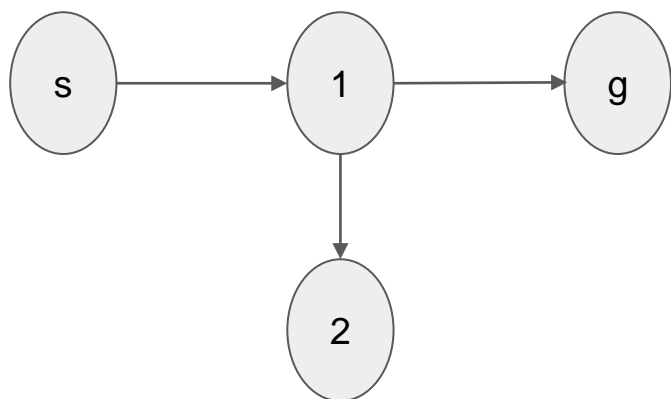
時間をかけて一番良い状態を探していく。



量子アニーリングについて（実践編）

実際問題にどう適用させるのか。

簡単な経路を考える。

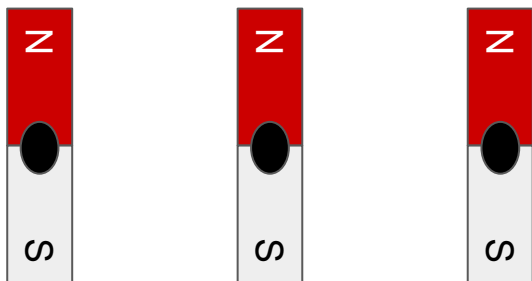


s から1を通過して g にいきたい。（2を通らないようにしたい。）

これを先ほどの磁石の話で問題をとく。（実際の解き方とは異なります。）

量子アニーリングについて（実践編）

3つの磁石を用意する。



1つ目の磁石は $s \rightarrow 1$ 間の道を通るか否か。

2つ目の磁石は $1 \rightarrow 2$ 間の道を通るか否か。

3つ目の磁石は $1 \rightarrow g$ 間の道を通るか否かを判断する。

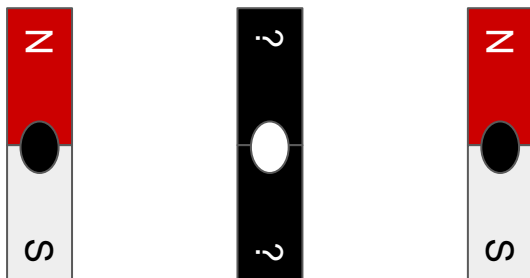
ここで各磁石はN極が上に来たらその道を通る。下に来たら通らないものとする。

量子アニーリングについて（実践編）

次に条件を確認する。

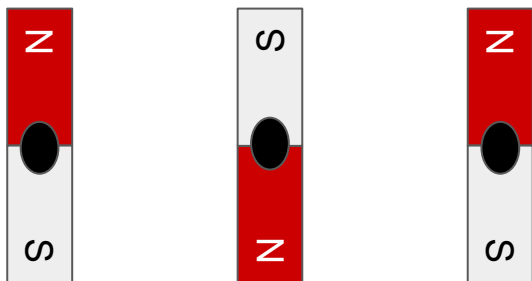
1. $s \rightarrow 1$ は必ず通らなければならない。
2. $s \rightarrow 1$ を通るときは $1 \rightarrow 2$ または $1 \rightarrow g$ を通る。また、逆も成り立つ。
3. $1 \rightarrow g$ は必ず通らなければならない。

つまり、1番目と3番目の磁石は条件から固定される。2番目のみを考えたい。



量子アニーリングについて（実践編）

答え。



この並びのときに一番安定する。

つまり、 $1 \rightarrow 2$ は通らない。

これで冒頭の問題を解くことができた。

量子アニーリングについて（実践編）

実際にプログラムで書いてみる。

- ・ ソースコード `annealing.ipynb`

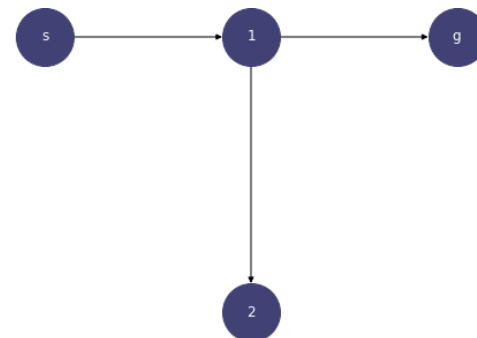
リンクはこちら↓

<https://colab.research.google.com/drive/14IPDy6d7z3paacHcERqpl7ETmsmmlP7j?usp=sharing>

流れは以下の通り。

1. 古典：問題をQUBOに変換する。
2. **量子**：QUBOを元に量子アニーリングを行う。
3. 古典：結果から答えを求める。

量子アニーリングについて（実践編）



Step1 古典：問題をQUBOに変換する。

今回扱う簡単な経路問題を量子アニーリングで解けるようにQUBOの式に変換する。

- ・ 入力データには量子アニーリングの設定をしている。
→ DWAVE_TOKEN には D-wave Leap(<https://cloud.dwavesys.com/leap/>) から取得したトークンを入力。
(トークンを取得できなかった場合はそのままでも動きますので大丈夫です。)
- ・ グラフデータは今回扱う問題をグラフに起こしている。(右上の図)
- ・ QUBO変換では起こしたグラフを元にQUBOの式を作成。

量子アニーリングについて（実践編）

Step2 量子 : QUBOを元にアニーリングを行う。

DWave社の量子アニーリングを用いて QUBO の解を見つける。

Step1で DWAVE_TOKEN を入力している場合は実際の量子コンピュータが動く。
入力していない場合はシミュレーションが動く。

ここでは実行したデバイスと最終結果が表示される。

通らない経路は0、通る経路は1となる。

実行結果↓

デバイス: `SimulatedAnnealingSampler`

結果

`{'1->2': 0, '1->g': 1, 's->1': 1}`

Energy: `-1998.0`

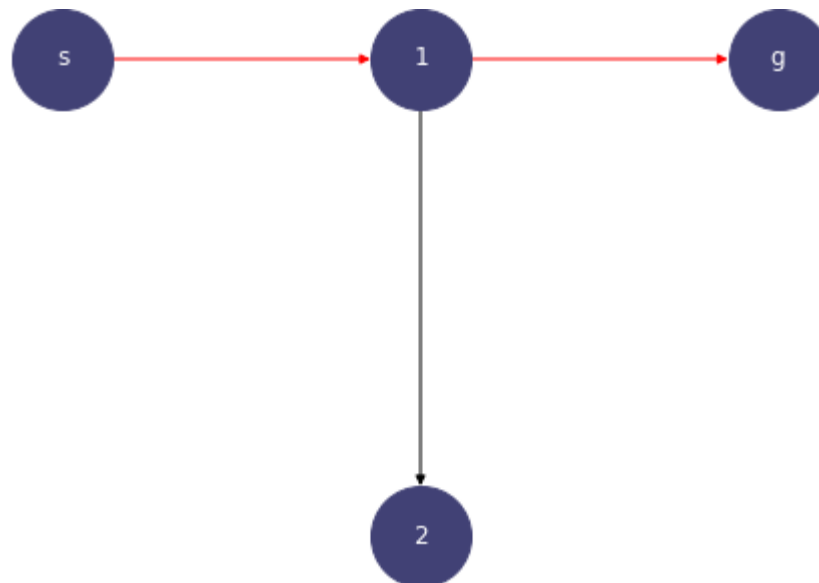
s→1 と 1→g の経路を通る。
1→2 の経路は通らない。

量子アニーリングについて（実践編）

Step3 結果から答えを求める。

Step2で求めた量子アニーリングの結果から実際の経路を求める。

経路は赤色で表示される。
（右図）



磁石で考えたものと同じものが出力される。