QAOAで組み合わせ最適化問題を解く

株式会社テラスカイ

R&D部
須郷聖也

会社の紹介

TerraSky 株式会社テラスカイ

JPX 東証一部上場　東証一部上場　証券コード3915

設　立：　2006年 3月

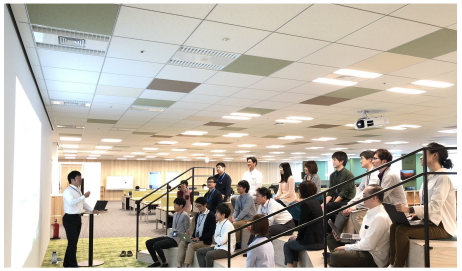
資本金　：　10億2,836.1万円（2018年12月末時点）

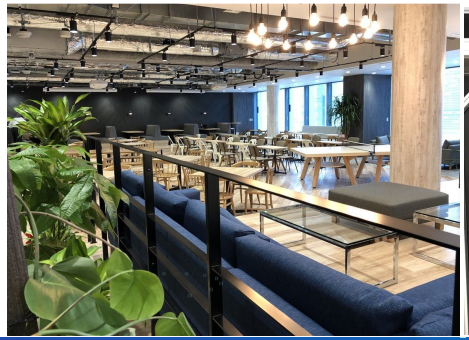
事業　：　クラウドインテグレーション事業

salesforce

powered by aws

導入実績 3,500件以上！





TerraSky

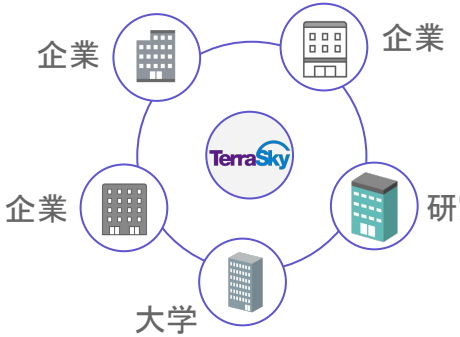
テラスカイの量子コンピューターへの取り組み

① PoC（実証実験）支援

- TerraSky
  - 量子ゲート
  - アニーラー
  - 機械学習

→ 活用提案 PoC支援 → 企業様（課題あり）

② 共同研究



初期教育
共同研究

自己紹介

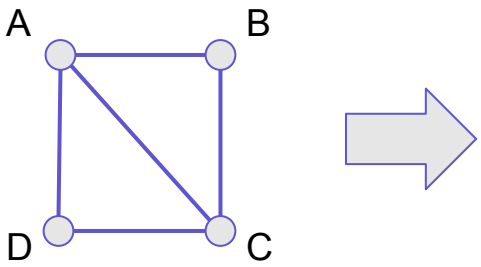
今日の話の流れ

- maxcut問題の紹介
- QAOAの紹介
- qiskitでデモ

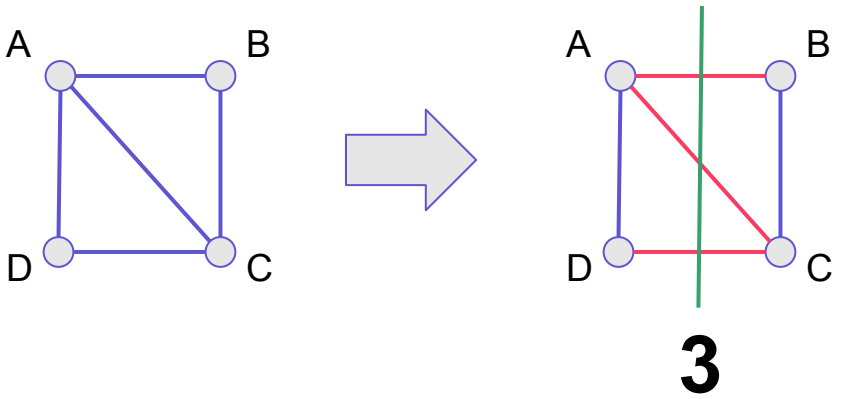
今日の話の流れ

- maxcut問題の紹介
- QAOAの紹介
- qiskitでデモ

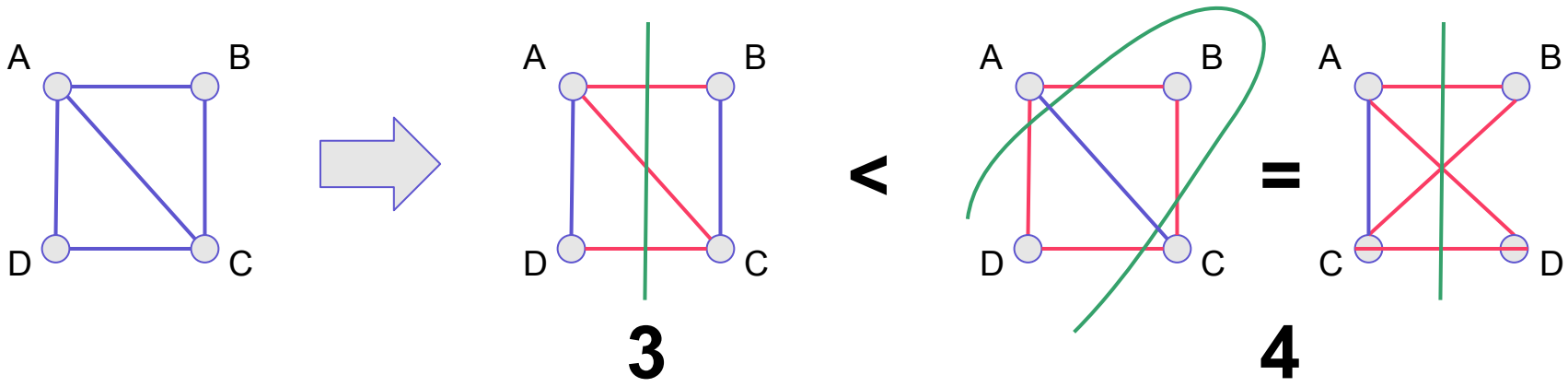
グラフの頂点を二つのグループに分ける。グループ間の辺の本数(または重みの合計)が最大になるように分ける



グラフの頂点を二つのグループに分ける。グループ間の辺の本数(または重みの合計)が最大になるように分ける

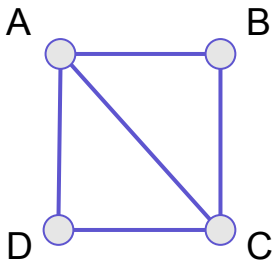


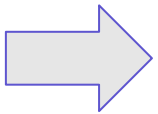
グラフの頂点を二つのグループに分ける。グループ間の辺の本数(または重みの合計)が最大になるように分ける



グラフの構造を表すデータ

- グラフの頂点同士が繋がりを数値にして表現した行列
- 繋がっている場合は1(または繋がりの強さを表す数値)、
  繋がっていない場合は0を入れる



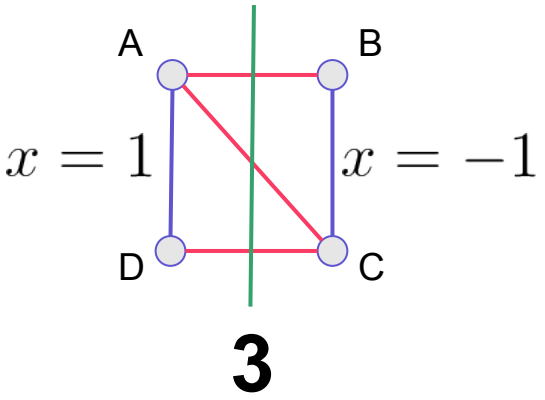


	A	B	C	D
A	0	1	1	1
B	1	0	1	0
C	1	1	0	1
D	1	0	1	0

$= C_{ij}$

イジング型のコスト関数

$$E = -\frac{1}{2}\sum_{i,j} C_{ij}\left(1 - x_i x_j\right) \quad x_i = 1, -1$$



$$E = \frac{-5 + x_A x_B + x_A x_C + x_A x_D + x_B x_C + x_C x_D}{2}$$

$$= \frac{-5 + 1 \times (-1) + 1 \times (-1) + 1 \times 1 + (-1) \times (-1) + (-1) \times 1}{2}$$

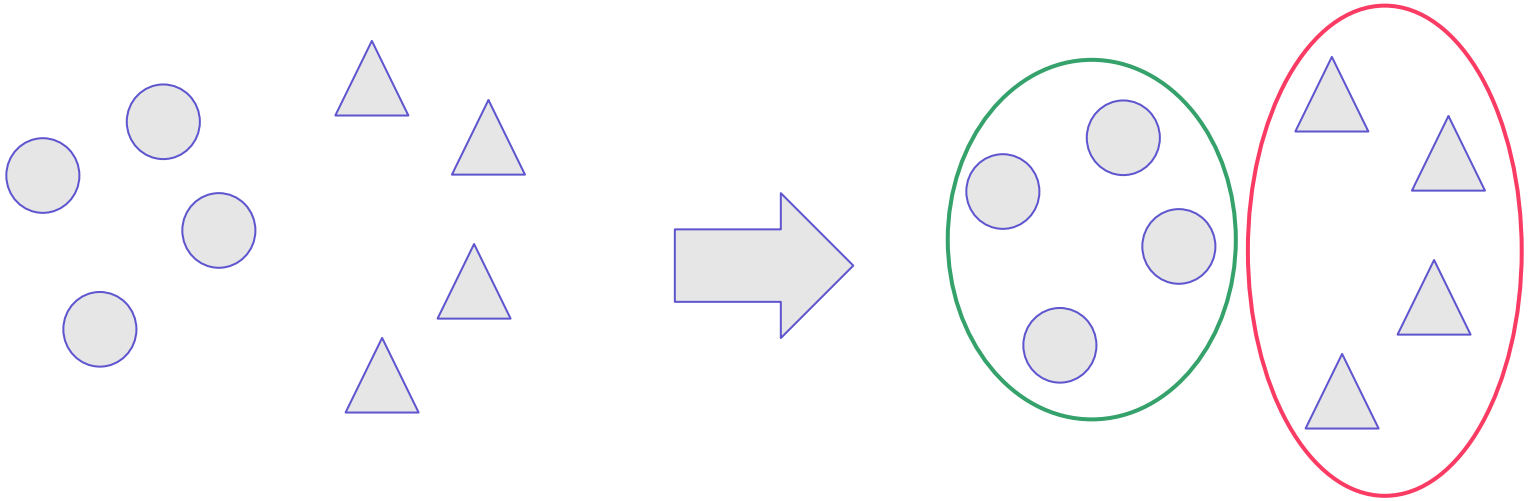
$$= -3$$

クラスタリング

いくつかのデータをある観点で似たもの同士にグループ分けする

→適用例：倉庫部品の部品棚の最適化(富士通)(QAOAではない)

https://www.fujitsu.com/jp/documents/about/resources/publications/magazine/backnumber/vol69-4/paper12.pdf

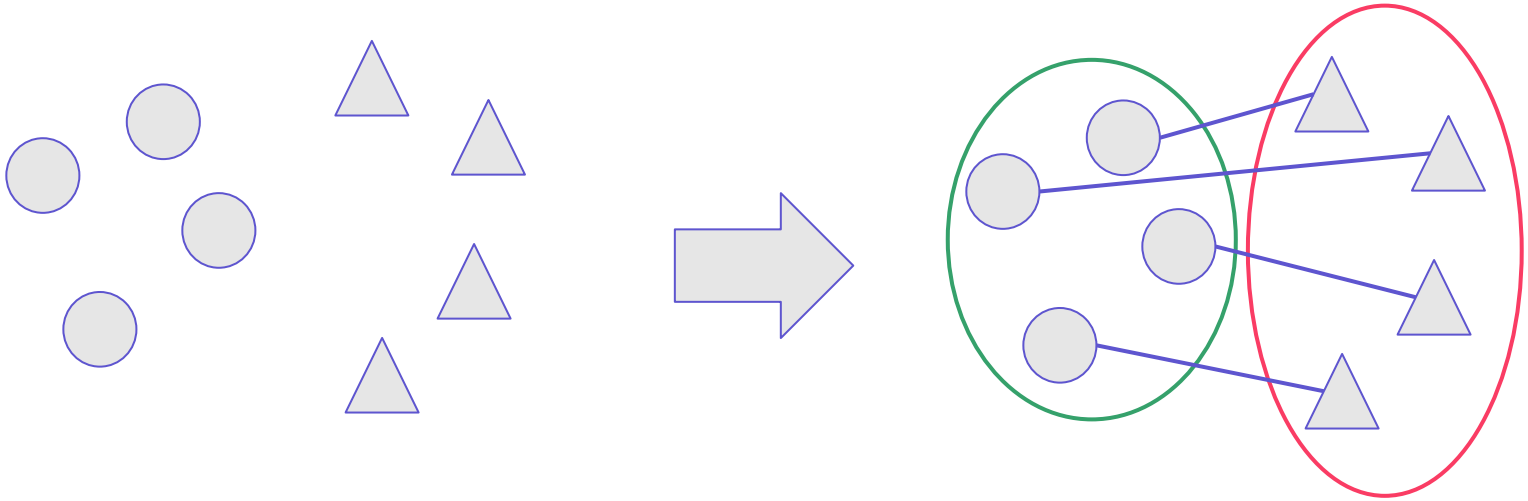


クラスタリング

いくつかのデータをある観点で似たもの同士にグループ分けする

→適用例：倉庫部品の部品棚の最適化(富士通)(QAOAではない)

https://www.fujitsu.com/jp/documents/about/resources/publications/magazine/backnumber/vol69-4/paper12.pdf



今日の話の流れ

- maxcut問題の紹介
- **QAOAの紹介**
- qiskitでデモ

組み合わせ最適化問題を解くためのゲートモデルのアルゴリズム

- 量子断熱計算によって最適な解を求める
- 時間発展をトロッター展開
- 古典・量子の繰り返し計算で最適化

量子断熱計算の形に

組み合わせ最適化の変数はそのままqubitの演算子に変換できる

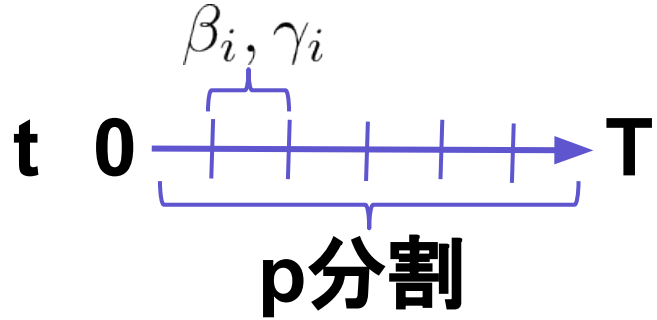
$$E = -\frac{1}{2}\sum_{i,j} C_{ij}\left(1 - x_i x_j\right) \quad \xrightarrow{x_i \to \hat{\sigma}_i^z \to Z_i} \quad H_{cost}$$

時間tの変化に合わせて形の変わる以下のようなハミルトニアンを用いることで、最適解を求める

$$H(t) = (1 - \frac{t}{T})H_{initial} + \frac{t}{T}H_{cost}\ (0 \leq t \leq T)$$

トロッター展開

$$H(t) = (1 - \frac{t}{T})H_{initial} + \frac{t}{T}H_{cost}$$



$$|\gamma, \beta\rangle = U(\beta_p)U(\gamma_p)\cdots U(\beta_0)U(\gamma_0)\,|s\rangle$$

$$\langle \gamma, \beta \,|\, H_{cost} \,|\, \gamma, \beta \rangle$$

古典・量子の繰り返しで最適化

古典

$$\beta_i, \gamma_i$$

optimizerで古典パラメーターを最適化

量子

$$\langle \gamma, \beta \mid H_{cost} \mid \gamma, \beta \rangle$$

実機またはシミュレーターでエネルギーを計算

今日の話の流れ

- maxcut問題の紹介
- QAOAの紹介
- **qiskitでデモ**

qiskit-aqua = qiskitのNISQ用ライブラリー

```python
import numpy as np
from qiskit_aqua.algorithms import QAOA
from qiskit_aqua import get_aer_backend
from qiskit_aqua.translators.ising import maxcut
from qiskit_aqua.components.optimizers import COBYLA
from qiskit_aqua import QuantumInstance

w = np.array([
    [0, 1, 1, 1],
    [1, 0, 1, 0],
    [1, 1, 0, 1],
    [1, 0, 1, 0]
])

p = 1
backend = get_aer_backend('statevector_simulator')
optimizer = COBYLA()
qubitOp, offset = maxcut.get_maxcut_qubitops(w)
qaoa = QAOA(qubitOp, optimizer, p, operator_mode='matrix')
quantum_instance = QuantumInstance(backend)
result = qaoa.run(quantum_instance)
```

必要な機能をインポート

グラフを隣接行列で準備

古典のoptimizerと量子計算をするbackendを準備

グラフをゲートの演算子に変換し、
QAOAを行うクラスを準備

実行

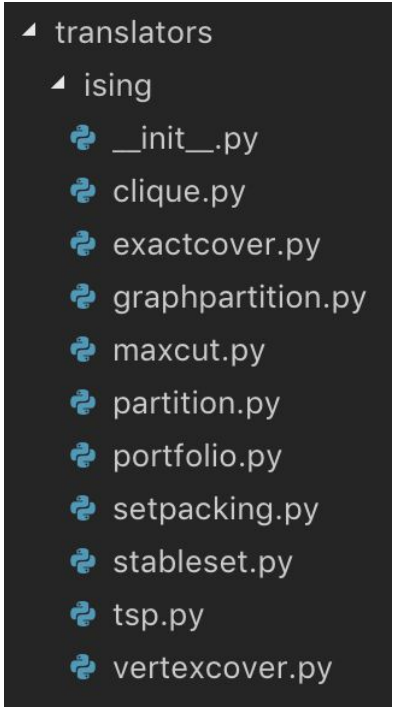
maxcut問題用のクラスを用いてqubit演算子に変換

- 隣接行列をゲートのオペレーターに変換する用クラス
- 他の様々な問題についても同様のものがる

```
from qiskit_aqua.translators.ising import maxcut
qubitOp, offset = maxcut.get_maxcut_qubitops(w)
print(qubitOp.print_operators())
```

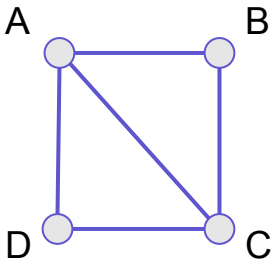
```
IIZZ    0.5
IZIZ    0.5
IZZI    0.5
ZIIZ    0.5
ZZII    0.5
```

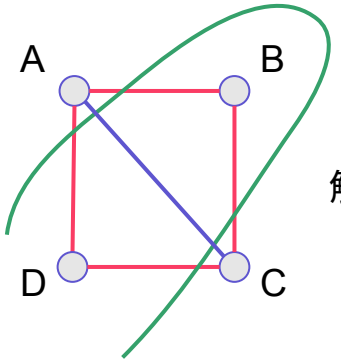
```
w = np.array([
    [0, 1, 1, 1],
    [1, 0, 1, 0],
    [1, 1, 0, 1],
    [1, 0, 1, 0]
])
```



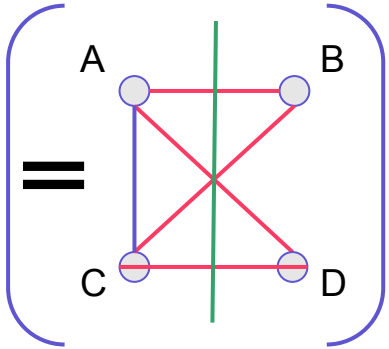
2つの問題を解きます

Q1

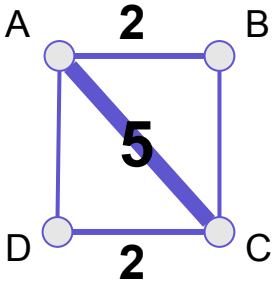


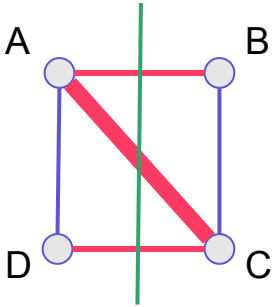


解は”4”



Q2





解は”9”

まとめ

- ゲートモデルでの組み合わせ最適化用アルゴリズムQAOAを紹介
- qiskit-aquaには便利なクラス・サンプルがあるので利用してください
- 簡単な問題でも古典パラメーターの最適化手法によって差が出る