概要

近年，量子計算と量子アニーラに関連する技術の発展やその研究が注目されている．様々な組み合わせ最適化問題は量子アニーラで迅速に解けることができる．量子アニーラを利用して問題を解決するのに問題自身がQUBO（**Quadratic Unconstrained Binary Optimization**）の形に変更することが必要である．TSP（巡回セールスマン）問題は組み合わせ最適化問題としてよく知られているが，対応するQUBOモデルの二次項数は問題のサイズが大きくなるに連れて爆発的に増えるという問題点がある．本研究では，ドロネー三角形分割とボロノイ図を利用して元のTSP問題の完全グラフから最適巡回路にならない辺を削除することで対応するQUBOモデルの目的関数の二次項数を大幅に削減することができた．実験結果として提案手法のseg法とnei法はそれぞれの削減率は68.59％，56.58％である．

1. はじめに

組み合わせ最適化問題は様々な制約条件の下で数多くの選択肢からある評価関数を最大または最小にする選択肢を選ぶ問題であり，その問題自身がNP困難問題でもある。問題に含まれる変数の個数が多くなるに連れて，問題の解空間が爆発的に拡大していて非常に困難な問題になってしまう．なので，コンピューターの計算性能は成功に問題を解けないかと繋がっている．残念ながら，現在のコンピューターは組み合わせ最適化問題を解くには大量の時間がかかって数年，数十年もかかる場合もある．通常の場合，近似アルゴリズムが多数考案されていて，その中メタヒューリスティックアルゴリズムは一番使われる手法としている．遺伝的アルゴリズム，蟻コロニー最適化アルゴリズム，タブーサーチ，焼きなまし法等は有名なメタメタヒューリスティックアルゴリズムとして知られている．ところがメタヒューリスティックアルゴリズムを用いても得られた解は必ず最適解ということを保証してもらえなくて，なるべく最適解に近い解を手に入れたいという目標で機能している．なお，一般的なメタヒューリスティックアルゴリズムは特定の問題に限らず汎用的に様々な問題が解けるのが，より精度が良い解を得るためにアルゴリズムに関連するパラメータを問題の事前知識で調整する必要がある．

近年，高速発展する量子計算と量子アニーラは組み合わせ最適化問題を解くためのもう一つも手法として注目されている．その中，量子アニーラは量子ビットの性質を利用して短時間で最小エネルギーを持つ解は導出してくれる．そのため，今量子アニーラと関連する研究も多くなっていく．

組み合わせ問題を量子アニーラで最適化するために問題自身がQUBO又はIsingの形に変換することが必要である．本研究ではQUBOモデルを注目している．

1. 関連研究（QUBOモデル，ising，tspモデルの導出、ドロネー、ボロノイー図）

２．１QUBO問題

QUBO（**Quadratic Unconstrained Binary Optimization**）問題とは二次形式の制約なし二値変数最適化問題であり，一般の数学式は：

**QUBOモデル**

その中，ｘはバイナリ変数でその値は０又は１，Ｑiiは一次項の係数，Qijは二次項の係数である．QUBO問題の目標はバイナリ変数xの値を定めてその値がQUBOの数学式を最小化または最大化にする．なお，QUBO問題は上三角行列の形で表現できる．

**QUBO行列**

その中，ｘはバイナリ変数のベクトル，Ｑは一次項と二次項の係数からなる行列で，対角成分Qiiは一次項の係数，非対角成分Ｑijは二次項の係数である．バイナリ変数xの個数が多くなるに連れて解空間の探索範囲は爆発的に拡大していて，なのでコンピューターの計算性能が非常に重要である．

QUBO問題の例を一つ上げる：

このQUBO問題は四つのバイナリ変数があって，x1,x2,x3,x4．

最小化する式は：

**PPT**

その式に対応するQUBO行列は：

**PPT**

x1,x2,x3,x4=0,1,1,1の時，式が最小値-7になる．なので，このQUBO問題の解は0,1,1,1

２．２isingモデル

イジングモデルのハミルトニアンは次の式で与えられる

**Ising**

その中，はスピンで上向き（+1）と下向き（-1）２状態を取れる，ｈは対応するスピンに与える外部磁場，Ｊは異なる二つのスピンの相互作用．イジングモデルとQUBOモデルはほぼ同じであって，次の式で簡単に相互変換できる．

**变换公式**

現実世界の焼きなましと同じ現象を持つイジングモデルは最終的にエネルギーが最小の状態に収束する特徴がある．その特徴を利用して量子アニーラが開発されて様々な問題が迅速に解決できるが，現在の量子アニーラは所有の量子ビット数はまた少ない問題点がある．その一方，量子アニーラが外部からの温度変化が非常に敏感で，少しだけの温度変化でスピンを影響してしまって正確な状態を得られなくなることがある．その故，量子アニーラの稼働状態を保持するために常に非常に低い温度でいなければならない．要するに現在の量子アニーラの稼働はコストが高くて未来の発展が期待される．

２．３tspモデルの導出

Tsp（巡回セールスマン問題）問題は組み合わせ最適化問題としてよく知られている．ｔｓｐは町の座標あるいは距離行列が与えられた時，全ての町をちょうど一度ずつ巡り出発地に戻る巡回路のうちで総移動距離が最小のものを求める組合せ最適化問題である．

例えば：次は町四つあるのtsp問題の完全グラフであり，目標は完全グラフの中から総距離が最小の巡回路を求めることである．町間の距離は距離行列で示す．例のインスタンスで最適巡回路は：１-２-３-４-1．

**ｔｓｐ示例图　　　距离矩阵**

通常のメタヒューリスティックアルゴリズムでtsp問題が解けるが，対応するQUBOモデルに変換して量子アニーラ，QUBOソルバーでも解決できる．

２．4tspのQUBOモデルの導出

TspのQUBOモデルは目的関数と制約条件から変換されたQUBO式，２部分から構成されている．

まずバイナリ変数xを定義する：

**ｘ変数定義**

ｘitはバイナリ変数でその値は１あるいは0，ｘitが１の時町iへt番目に訪れる，０の時町iへt番目に訪れないと定義している，なのでサイズｎのインスタンス（町の個数はｎ）でｎ^2個のバイナリ変数が必要である．

目的関数は次の式で示す．

**目的関数**

その中，ｎは町の個数．ｄijは町iと町jのユークリッド距離（整数）である．目的関数の値は求めた巡回路の総距離である．

TSP問題は二つの制約条件がある．一つ目は

1. 各町は1回しか訪れてはいけない

**PPT图**

**QUBO式子**

二つ目は

1. 同じタイミングに複数の町に行くことはできない

**PPT图**

**QUBO式子**

全ての部分を合わせてtsp問題のQUBOモデルが得られる．

**ｔｓｐQUBO**

その中，はペナルティー係数．通常は距離行列の最大値に設定して制約条件を違反すると式の値を増加させる．

町ｎ個あるインスタンスに対して目的関数の二次項数はn^2\*(n-1)，なのでインスタンスのサイズが大きくなるに連れて目的関数の二次項の個数が爆発的に増えるという問題点がある．本研究の新しいアイデアとしてドロネー三角形分割とボロノイ図を利用して元のTSP問題の完全グラフから最適巡回路にならない辺を削除することで目的関数の二次項数を削減することである．

２．４ドロネー三角分割　と　ボロノイー図

ボロノイ図（Voronoi Diagram）とは，平面上に配置された複数の点（母点）に対して，各点から最も近い領域を決定する方法である．具体的には，各母点からの距離が最小となる点の集合が，その母点に対応するボロノイ領域を形成する．これにより，平面が複数のボロノイ領域に分割される。例えば：下の図はあるtspインスタンスに基づいて生成されたボロノイ図，複数のボロノイ領域から構成されている．青い点は母点（tsp問題の町），オレンジ色の点はボロノイ頂点，黒い辺はボロノイ辺である．

**ボロノイ図**

ボロノイ図の性質

ドロネー三角分割（Delaunay Triangulation）は、与えられた母点集合を三角形に分割する手法で、ボロノイ図と密接に関連している．具体的には、ドロネー三角分割は、ボロノイ図の隣接する領域の母点を結んで得られる三角形から構成される．

このように、ボロノイ図とドロネー三角形分割は双対の関係にあり、ボロノイ図の頂点はドロネー三角形の外接円の中心に対応する．この関係性により、ドロネー三角形分割とボロノイ図は計算幾何学やグラフィックス、地理情報システム（GIS）など、さまざまな分野で応用されている．

1. 提案手法

ドロネー三角形分割の辺はTSP問題の最適巡回路になる可能性が高いが，全ての最適巡回路はドロネー三角形分割に含まれることはないという問題がある．ここで二つの新しい手法を提案して最適巡回路は新しいグラフに含まれてほしいのである．二つの手法は既存のドロネー三角形分割の上で新しい辺を加えてグラフを作る．

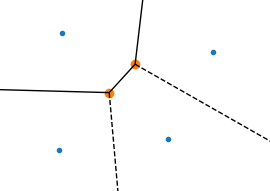
* Nei方法

ドロネー三角形分割を作る一つの手法としてボロノイ図で隣接する二つの領域に属する母点をつなぐとドロネー三角分割になる．提案するNei方法では隣接関係は一回だけではなく三回まで拡張する．つまり，一つのボロノイ領域対して隣の隣の隣の領域まで母点を繋ぐ手法となる．

**Nei方法**

図のように赤色，緑色，青色の領域はぞれぞれ一回，二回，三回の隣接関係を拡張した結果で，提案手法で真中の灰色領域の母点は他の三色領域の母点とつないでいく．これでneiグラフで含まれる辺はおおくなってほとんどの最適巡回路が含まれる可能性が高くなる．

* Seg方法

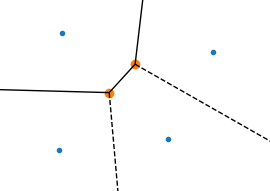


10

9

3

4



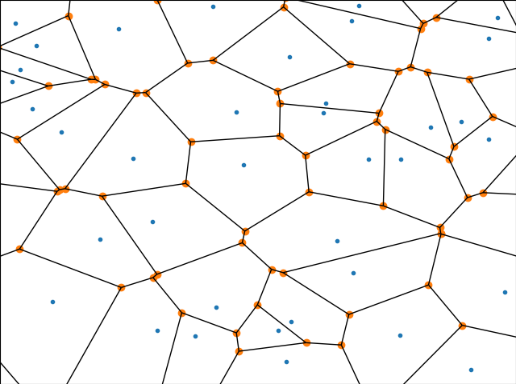
10

9

3

4

下の図のように母点９と４に属するボロノイ領域は隣接するのでドロネー三角形分割で母点９と４が繋がっている．母点１と３は繋がっていないが，真ん中のボロノイ辺は短くて母点９と４からなる線分の長さは母点１と３からなる線分の長さ近いであるので，それは最適巡回路になる可能性も高いというアイデアで母点１と３を繋ぐ．それに基づいてseg方法を提案した．



図のように，赤色の一つのボロノイ辺に対応する両端の赤色の点線で囲まれた母点をつなぐ．それに加えて，緑色の二つ連続するボロノイ辺と青色の三つ連続するボロノイ辺に対応する両端の母点をつなぐ．これでneiグラフと同じようにsegグラフで含まれる辺は最適巡回路になる可能性が高くなる．

1. 実験

実験で用いるインスタンスはまずランダムに町の座標を生成してから四捨五入した距離行列を計算してから上三角行列の形でTSP問題の専用ソルバーLKHに入力する．なお，ひとつのインスタンスに対して同じパラメータで10回繰り返し解いてその中の最短巡回路はインスタンスの最適解とする．実験用のインスタンスのサイズ（町の個数）が５から200，総196個である．

4.1　辺の個数に関する実験

ドロネー三角形分割，seg方法で計算されたグラフ，nei方法で計算されたグラフと完全グラフぞれぞれ辺の個数を統計すると結果は下の図で示す．

**统计边数**

図が示すように横軸はインスタンスのサイズ，縦軸は辺の個数を示している．辺の個数が一番多いのはもちろん完全グラフであり，その次はnei方法で計算されたグラフとseg方法で計算されたグラフ，一番少ない辺を持つのはドロネー三角形分割．明らかにインスタンスが小さいときどの計算方法でも得られたグラフの辺の個数が近くて，サイズが大きいとき提案されたseg方法とnei方法で得られたグラフは完全グラフと比べて大幅に辺を削除できた．よりおおい辺を完全グラフから削除するとそのあとのQUBOモデルで目的関数の二次項数がより少なくなるのである．

４．２提案されたsegグラフとneiグラフには最適巡回路が含まれるか

上記のように各インスタンスの最適巡回路はTSP問題の専用ソルバーLKHで得られた．一つのインスタンスは複数の最適巡回路が存在する可能性があるが，LKHでTSP問題を解くとき一つだけの最適巡回路が得られる．今その最適巡回路は提案されたsegグラフとneiグラフに含まれるかどうかを確認したところ全てのインスタンスに対して提案されたseg方法とnei方法で得られたグラフでLKHで出力された最適巡回路が含まれることが確認できた．そのため，元の完全グラフに基づくTSP問題と提案された方法で得られたグラフで基づくTSP問題が同じ問題になってより少ない辺を持つため制限されたTSP問題も簡単になるのである．

４．３LKHで制限されたグラフ（segグラフとneiグラフ）のTSP問題を解く

一つのインスタンスに対して複数の最適巡回路が存在する可能性があるのがLKHで一つだけの最適巡回路が出力することができる．LKHで制限されたグラフのTSP問題を解くことで他の最適巡回路が存在するかどうかを確認できる．

その手順としてLKHに入力する距離行列に制限されたグラフで存在しない辺（完全グラフと比べる）のところの距離は元の距離行列の最大値で書き換えることで（その辺は必ず最適巡回路にならないという意味）制限するTSP問題の距離行列を作ることができる．

**如何换的**

制限する距離行列をLKHに入力して同じパラメータで解いてから得られた解の総距離と巡回路を比べることで複数の最適巡回路が存在するかどうかを確認できる．

結果として全てのインスタンスに対して制限なし（完全グラフ）のTSP問題の最適巡回路の総距離は制限する（segグラフあるいはneiグラフ）TSP問題の最適巡回路の総距離が全て一致することが確認できた．対応する二つの巡回路の町順番を確認すると多数のインスタンスは制限するTSP問題の最適巡回路と制限なしのTSP問題の最適巡回路が実は違う巡回路になることが確認できた．なので多数のインスタンスは複数の最適巡回路が存在しているが，提案された方法で作られたグラフで少なくとも一つの最適巡回路が存在する．

5.目的関数の二次項数を削減する

第４章で行われた実験のもとで制限されたグラフに存在しない辺は最適巡回路にならないので元のTSP問題のQUBOモデルを作るときそれらの考慮をしなくても良いということで目的関数の二次項数を削減できる．

1. まとめ