

QUBO 問題の高次自動変換による量子ビット数削減技術の開発 —量子ビット・ゲート数削減による組合せ最適化の高速化—

1. 背景

近年、半導体の微細化限界が危惧されていることから、組合せ最適化問題に対する量子コンピューティング技術の活用が期待されている。量子アニーリングは二次制約なし二値変数最適化 (quadratic unconstrained binary optimization; QUBO) 問題に対応している。したがって、組合せ最適化問題を解く際は QUBO として定式化するのが一般的である。

そのような現状に対し私は、無線通信分野の組合せ最適化問題である無線資源割当問題を高次制約なし二値変数最適化 (higher-order unconstrained binary optimization; HUBO) 問題として定式化することで、グローバール適応探索 (Grover adaptive search; GAS) を適用した際の必要量子ビット数を削減した。

また HUBO は IBM CPLEX Optimizer など、最先端の数理計画ソルバにも対応していない。したがって HUBO を解く際は、補助変数を追加することで QUBO もしくは整数計画問題として定式化し直すアプローチが主流である。対して、GAS は QUBO に加え HUBO に対しても大域的最適解が得られ、二次的な加速が証明されており、実機での動作を確認されていることが大きな特徴である。

2. 目的

本プロジェクトの目的は、無線資源割当問題のみに対応している量子ビット数削減手法を用いて、GAS を実行する際の量子ビット数を削減する一般的手法を開発することにより、より一層効率的で高速に組合せ最適化問題を解く技術を提供することである。

3. ソフトウェア開発内容

本プロジェクトの調査により、組合せ最適化問題のインデックスを one-hot encoding して QUBO 定式化している組合せ最適化問題に対しては量子ビット数削減手法を適用できることがわかった。ここで one-hot encoding とは、 $1 \rightarrow [1\ 0\ 0\ 0]$ や $3 \rightarrow [0\ 0\ 1\ 0]$ のようにワンホットベクトルによってインデックスを表す方法である。このような組合せ最適化問題に対し、インデックスを binary encoding して HUBO 定式化することで必要量子ビット数を削減できる。ここで binary encoding とは、 $1 \rightarrow [0\ 0]$ や $3 \rightarrow [1\ 0]$ のように二進数によってインデックスを表す方法である。

3.1 量子ビット数削減手法

量子ビット数削減手法について、グラフ彩色問題 (graph coloring problem; GCP) を例に説明する。GCP とは無向グラフ $G = (V, E)$ と色の集合 I が与えられたとき、頂点や辺に色を塗り分ける問題である。今回は隣接する頂点を異なる色で塗り分ける頂点

彩色問題を考える。

HUBO 定式化ではインデックスを二進数で表す。一般に、インデックスが複数個存在する場合は最も小さいインデックスを binary encoding すればよい。GCP は $V > I$ であるため色のインデックス I を binary encoding する。色のインデックスを表すのに必要なビット数 N_B は次式で定義される。

$$N_B = \lceil \log_2 I \rceil \quad (1)$$

二進数を昇順で割当てる HUBO-ASC (HUBO with ascending assignment; HUBO-ASC) の場合、色のインデックスが与えられるとビット列は次式で定義される。

$$[b_{i1} b_{i2} \cdots b_{iN_B}] = [i - 1]_2 \quad (2)$$

ここで、 $[\cdot]_2$ は十進数から二進数への変換を表す。

二進数を降順で割当てる HUBO-DSC (HUBO with descending assignment; HUBO-DSC) の場合、色のインデックスが与えられるとビット列は次式で定義される。

$$[b_{i1} b_{i2} \cdots b_{iN_B}] = [I - i + 1]_2 \quad (3)$$

HUBO-ASC/DSC において、頂点 v を色 i に塗る状態は次式で表すことができる。

$$\delta_{vi}(x) = \prod_{r=1}^{N_B} (1 - b_{ir} + (2b_{ir} - 1)x_{vr}) \quad (4)$$

よって、GCP の HUBO-ASC/DSC は次式で定義される。

$$E_{\text{GCP}}^{\text{HUBO}}(x) = \sum_{u,v \in E} \sum_{i=1}^I \delta_{ui}(x) \delta_{vi}(x) + \lambda' \sum_{v=1}^V \sum_{i=I+1}^{2^{N_B}} \delta_{vi}(x) \quad (5)$$

ここで、 λ' は制約条件を満たすためのペナルティ係数を表す。

3.2 目的関数を展開せずに量子回路を構成することによるゲート数削減手法

binary encoding による HUBO 定式化では、0 を $(1-x)$ 、1 を x と対応づけて定式化するため、目的関数に $(1-x)$ と x が頻出する。 $x = 1$ のとき必ず $(1-x) = 0$ となり、 $x = 0$ のとき必ず $(1-x) = 1$ となる。この関係性を用いて量子回路を構成することでゲート数を削減できる。従来の GAS は目的関数を展開してそれに対応する量子回路を構成していた。本プロジェクトでは上述の関係性を用いて、目的関数を展開せずに、因数分解した各項と量子回路を一対一対応させることでゲート数を削減する手法 (HUBO with polynomial factorization; HUBO-PF) を提案する。

HUBO-PF は、0 を表す項 $(1-x)$ の前後に X ゲートを挿入することで実現できる。

3.3 目的関数の最大次数を半減する定式化によるゲート数削減手法

1 が偶数個含まれる二進数にのみインデックスを割当てることで、量子ビット数が微増する代わりに、目的関数の最大次数を半減させる手法 (HUBO with order reduction; HUBO-OR) を提案する。

HUBO-OR について GCP を例に説明する。二進数に含まれる 1 が偶数個のときのみインデックスを割当てる、色のインデックスを表すのに必要なビット数 N'_B は次式

で定義される。

$$N'_B = \lceil \log_2 I \rceil + 1 \quad (6)$$

HUBO-OR において、頂点 v を色 i に塗る状態は次式で表すことができる。

$$\delta'_{vi}(x) = \prod_{r=1}^{N'_B} (1 - b_{ir} + (2b_{ir} - 1)x_{vr}) \quad (7)$$

よって、GCP の HUBO-OR は次式で定義される。

$$E_{\text{GCP}}^{\text{HUBO-OR}}(x) = \sum_{u,v \in E} \prod_{r=1}^{N'_B} (1 - x_{ur} - x_{vr}) + \lambda_1 \sum_{v=1}^V \sum_{i=1}^I \frac{(-1)^{i+1} + 1}{2} \delta'_{vi}(x) + \lambda_2 \sum_{v=1}^V \sum_{i=I+1}^{2N'_B} \delta'_{vi}(x) \quad (8)$$

ここで、 λ_1 、 λ_2 は制約条件を満たすためのペナルティ係数を表す。

4. 新規性・優位性

各手法の目的関数から、GCP を例に必要な量子ビット・ゲート数を代数的に解析し、オーダを導出した。

図 1 に量子ビット数の比較結果を示す。従来の QUBO と比較し、HUBO-ASC/DSC は必要量子ビット数を削減できた。また HUBO-OR は、HUBO-ASC/DSC と比較し量子ビット数が微増しているが、従来の QUBO と比較すると大幅に量子ビット数を削減している。

図 2 に GAS の状態準備演算子一つの構成に必要な T ゲート数の比較結果を示す。HUBO-ASC と比較し、HUBO-DSC と HUBO-OR はほとんどの場合で T ゲート数を削減できた。また問題サイズが大きい場合、QUBO よりも HUBO-PF の方が T ゲート数が少ない。このことから、組合せ最適化問題の問題サイズが大きい場合、HUBO-PF が量子ビット数と T ゲート数が最も少なく最良の手法であることがわかる。

5. 期待されるユーザー価値と社会へのインパクト

本プロジェクトのユーザーは、組合せ最適化問題を解くあらゆる人が該当するが、特に配送経路の最適化や無線通信のチャネル割当の最適化など、大規模な最適化を行う企業がユーザーになると想定している。

ユーザーが成果を利用することにより、組合せ最適化問題の探索空間が縮小し、より早く解を求められるようになる。よって、高速かつ大容量な通信の実現や、無駄のない配送計画の実現によりこれまでより早く荷物が届くなどの価値を享受できると考えている。また、本プロジェクトでは量子ビット数に加えゲート数も削減することができたため、従来手法より実機での利用を早い時期に実現できる可能性もある。

本プロジェクトの適用範囲は非常に広い。古典コンピュータにより QUBO を解くのと、量子コンピュータにより HUBO を解くのとでは、量子の方が必要な問合せ回数が二次的に少なく、さらに変数の数は量子の方が対数オーダで少ない。したがって、大規模な誤り耐性万能量子コンピュータが実現した場合、実用的な求解時間は量子の方がとても速い可能性が高く、これは社会全体にとって大きなインパクトである。

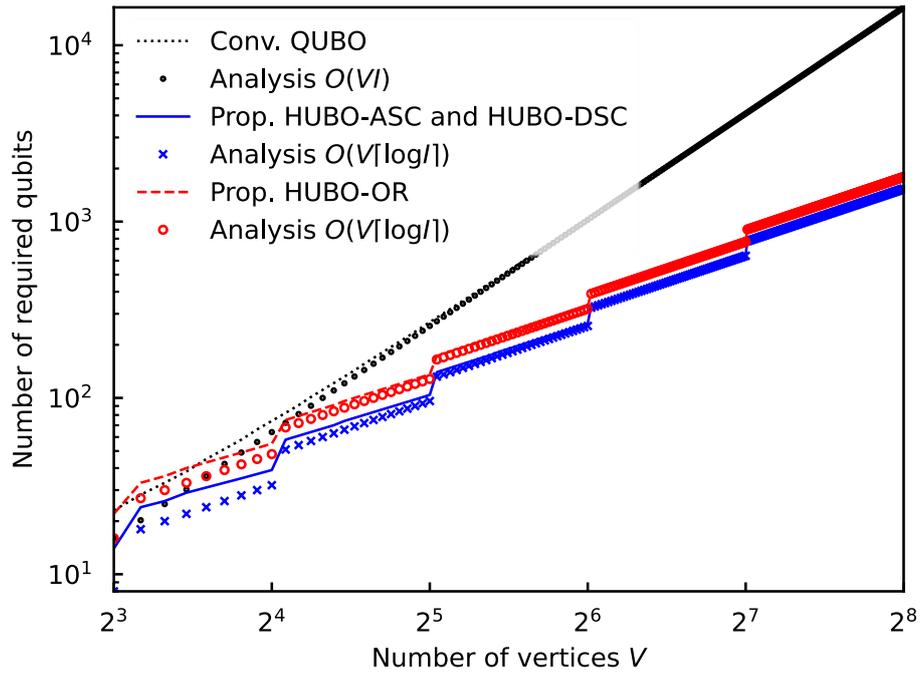


図1 量子ビット数の比較

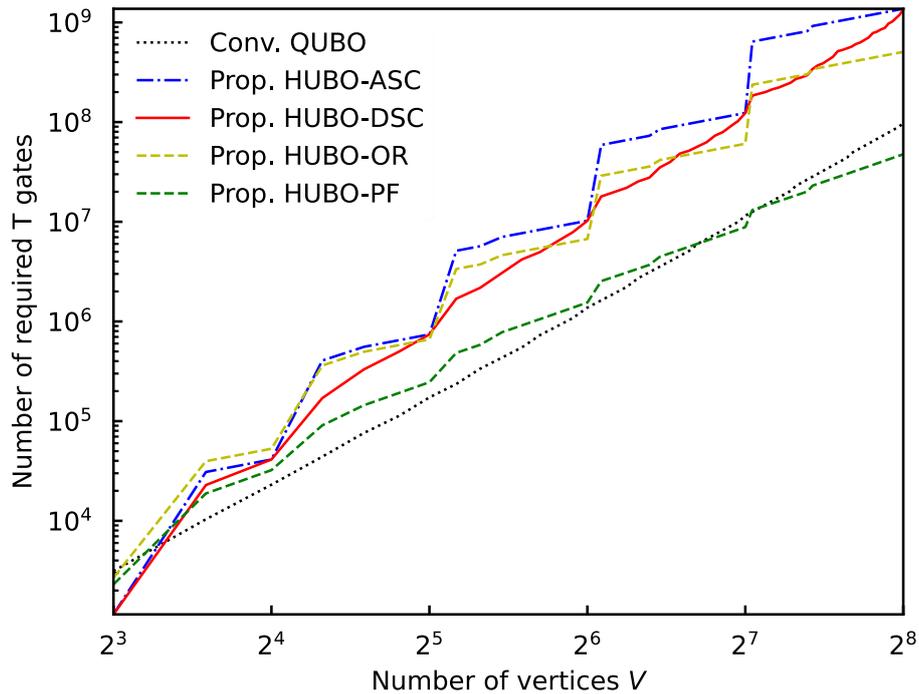


図2 GASの状態準備演算子一つの構成に必要なTゲート数の比較

6. 氏名 (所属)

佐野 友貴 (横浜国立大学 大学院理工学府)

(参考) 関連 URL

<https://github.com/YS-39/GCPToHUBO>