



令和6年1月25日

ABS2 QUBO ソルバーの GPU エンジン¹を無償提供します ～ 組合せ最適化問題を解くための C++ API の公開 ～

【本研究成果のポイント】

- QUBO 問題の解を効率的に探索する ABS2 QUBO ソルバーを利用するための C++プログラミング言語用 API を開発しました。
- この API を通じて、ユーザーは C++プログラムから ABS2 の GPU エンジンにアクセスでき、GPU を活用して QUBO 問題の解を迅速に探索できます。
- 非商用かつ評価研究目的に限り、ABS2 の GPU エンジン実行バイナリと C++API を無償かつ無保証で公開・提供します。

【概要】

広島大学大学院先進理工系科学研究科の中野浩嗣教授らの研究チームは、株式会社 NTT データグループと共同で、組合せ最適化問題の効率よく探索する計算方式「ABS (Adaptive Bulk Search)」の開発・改良に取り組んできました。ABS は、複数の GPU (Graphics Processing Unit、*1) を用いて QUBO (Quadratic Unconstrained Binary Optimization) 問題の解を並列探索します。QUBO 問題を介した組合せ最適化分野の研究に貢献するため、開発した QUBO ソルバーの GPU エンジンの実行バイナリとそれにアクセスするための C++言語 API (Application Programming Interface、*2) を非商用かつ評価研究目的に限り、無償かつ無保証で提供します。提供するものは、ABS に探索アルゴリズムの動的自動選択機能が追加された ABS2 QUBO ソルバーの GPU エンジンです (発表文献[1] を参照)。

ユーザーは、GPU を搭載した計算機に ABS2 の GPU エンジンをインストールすることで、C++言語 API を介して QUBO 問題の解探索を自由に行えます。組合せ最適化問題を QUBO 問題に変換する部分は、ユーザー自身が C++言語を用いて開発します。これにより、C++言語 API を呼び出し、高速な並列解探索を実現できます。さらに、ユーザーは GPU エンジンの各種パラメータを調整したり、コールバック関数を設定することで、解きたい QUBO 問題に合わせた ABS2 のチューニングや他の QUBO ソルバーとの連携処理も可能です。

【背景】

社会に存在する多くの解決すべき問題は、数多くの選択肢から最適なものを見つける組合せ最適化問題として捉えられます。例えば、最短経路の探索や画像の最適化など、さまざまな組合せ最適化問題を効率的に解くことが要求されています。しかしながら、個々の課題に合わせた専用のソルバーを開発するのは専門知識と莫大な開発コストが必要であり、困難を伴います。一方で、多くの組合せ最適化問題が QUBO 問題に変換可能であることが知られています。この性質を利用し、図1のように、解きたい組合せ最適化問題を QUBO 問題に変換し、QUBO ソルバーで解を見つけ、その解を元の問題の解となるよう逆変換することで、元の問題の解を導き出すことが可能となります。高性能な QUBO ソルバーを利用すれば、組合せ問題を QUBO 問題に変換する処理と解を元の問題の解に変換する処理のみを開発することで、その組合せ最適化問題の高精度な解を得ることができます。このため、多くの大学や企業で GPU

だけでなく、専用の集積回路や量子デバイスを活用した QUBO ソルバーが開発され、その性能競争が繰り広げられています。

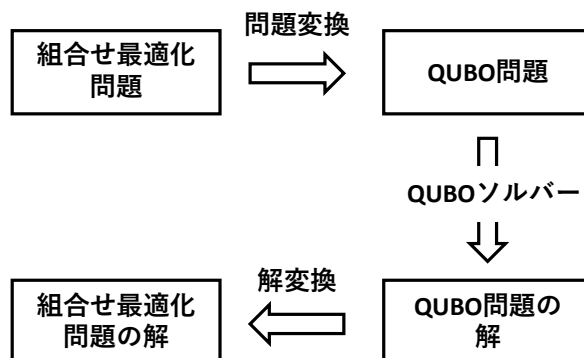


図1：QUBO ソルバーを用いた組合せ最適化問題の解法

【研究成果の内容】

QUBO 問題は、複数の 2 値変数に対する 2 次式を目的関数とし、その関数値を最小化するための 0 または 1 への割り当てを求める問題です。例えば、変数 a、b、c の 2 次式 $f(a, b, c) = 2ab - 2ac - bc + a - b$ を目的関数とする QUBO 問題では、 $a=0$ 、 $b=c=1$ のとき、 $f(0, 1, 1) = -2$ となり、これが最小値となる最適解です。

QUBO 問題の応用例として、視覚的にわかりやすいグレースケール画像の 2 値化処理を題材に説明します（参考文献[2]）。グレースケール画像は、ピクセルの 2 次元配列で表現され、各ピクセルは 0（黒）から 1（白）までの連続した明るさを持ちます。例えば、0.5 は中間の灰色を表します。このグレースケール画像を、0（黒）と 1（白）の 2 値だけを持つピクセルの 2 次元配列、つまり 2 値画像に変換するのが 2 値化処理です。印刷時には、紙の各座標にインクやトナーを「着ける」か「着けない」か（黒か白か）の 2 通りしか扱えません。そのため、印刷時には、グレースケール画像を直接印刷可能な 2 値画像に変換する 2 値化処理が行われます。2 値画像では、元のグレースケール画像のトーンや細部をできるだけ再現することが求められます。また、フルカラー印刷においても、インキ CMYK に対応する 4 つのグレースケール画像ごとに 2 値化処理が行われます。

人間の視覚特性により、2 値画像を目視した際、網膜にはぼかし処理された画像が投影されます。このため、入力されたグレースケール画像とそのぼかし画像との差が小さいほど、2 値画像がより正確にグレースケール画像を再現していると考えられます。このぼかし画像とグレースケール画像の差は、2 値画像のピクセル値の 2 次式として表現することができます。具体的には、グレースケール画像のピクセル値の 2 次元配列 $P=(p_{i,j})$ 、ぼかしフィルタ係数の 2 次元配列 $B=(b_{k,l})$ 、2 値画像のピクセル値の 2 次元配列 $X=(x_{i,j})$ に対して、グレースケール画像とぼかし画像の差を表す目的関数は次の式 $f(X)$ で示されます。各ピクセル座標 (i,j) における明るさの差の 2 乗、つまり 2 乗差の合計が $f(X)$ となります。

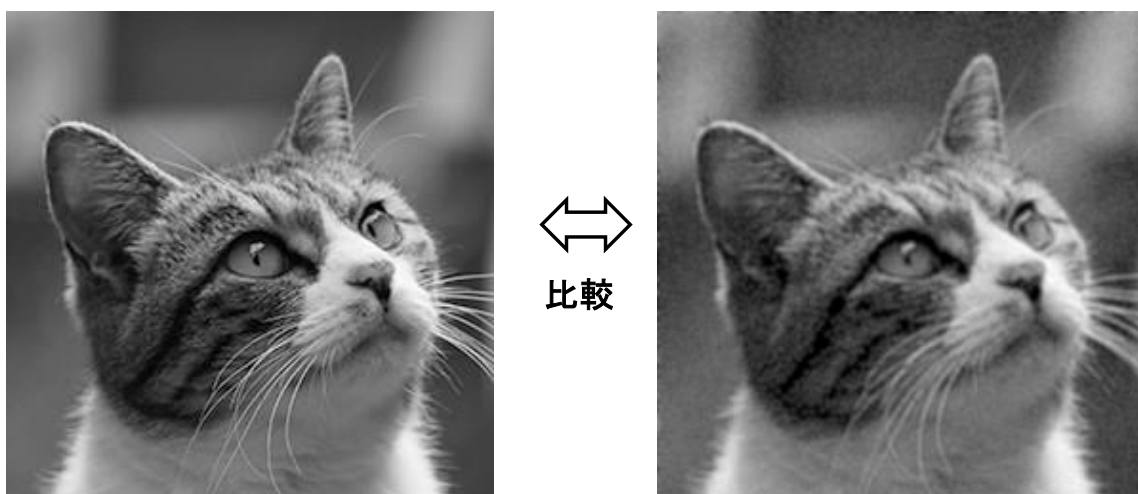
$$f(X) = \sum_i \sum_j \left(p_{i,j} - \sum_k \sum_l b_{k,l} x_{i+k,j+l} \right)^2$$

したがって、人間の視覚特性に基づく 2 値化処理は、関数 $f(X)$ の値を最小化する X を見つける組合せ最適化問題として表すことができます。 $f(X)$ の式を展開すると、 $X=(x_{i,j})$ を変数とする 2 次式になるため、これを QUBO 問題として取り扱うことが可能です（問題変換）。QUBO ソルバーを使用することで、 $f(X)$ を最小化する解 X を見つけることができます。そして、解 X の 2 次元配列をピクセル値として持つ 2 値画像を生成（解変換）することで、入力されたグレースケール画像を最もよく再現する 2 値画像を得ることができます。図 2 では、ABS2 QUBO ソルバーを利用して得られた 2

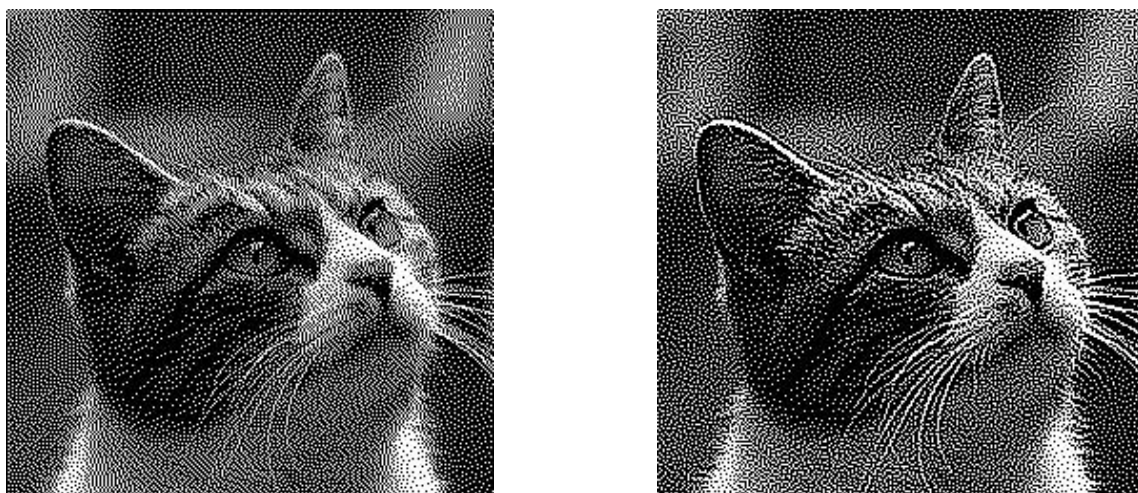
値画像と、一般的な誤差拡散法による 2 値画像を示しています。視覚特性に基づく 2 値画像の方がより鮮明で、細部がより再現されていることが確認できます。

2 値変数への値の割り当ての全組み合わせについて目的関数 f を計算し、最小となるものを求めれば、QUBO 問題の最適解を得ることができます。例えば、3 変数では、 $2^3=8$ 通りです。変数の個数が多くなると、組み合わせの数が膨大になります。一般に変数の個数を n とすると、 2^n 通りになります。2 値化処理の例では、ピクセル数、つまり 2 値変数の個数が 10000 の場合、 $2^{10000} \approx 10^{3010}$ 通りとなり、全通りの計算は不可能です。そこで、最適解に近いであろう解を探索する手法が用いられます。ABS2 は、さまざまな探索手法から、最適な探索手法を動的に選択し、GPU の高い計算能力を用いて短時間に膨大な解探索を行い、最適もしくは最適に近い近似解を求めます。

入力グレースケール画像 $P = (p_{ij})$ 2 値画像のぼかし画像 $\sum_k \sum_l b_{k,l} x_{i+k,j+l}$



↑ ぼかし処理



参考：誤差拡散法による 2 値画像

ABS2により求めた 2 値画像 $X = (x_{ij})$

図 2：人間の視覚特性にもとづいたグレースケール画像の 2 値化処理：

ABS2 QUBO ソルバーの GPU エンジンの実行バイナリと、C++言語用の API を非商用で評価研究目的に限り無償・無保証で提供します。ユーザーは、GPU を搭載した Linux OS の計算機に ABS2 QUBO ソルバーをインストールし、C++言語の API を利用して GPU を最大限に活用しながら QUBO 問題を解くことができます。

さらに、ABS2はコールバック機能（*3）を用いて他のQUBOソルバーと連携することも可能です。たとえば、商用の数値最適化ソルバーであるGurobi Optimizerは2次の目的関数をサポートしており、QUBO問題を解くことができます。ABS2 GPUエンジンとGurobi Optimizerのコールバック機能を組み合わせることで、両者が探索中に得た良い解を共有し合うことができます。これにより、それぞれのソルバーが持つ長所を生かした解探索が可能となります。

図2の2値化処理における2乗差の合計 $f(X)$ を最小化するQUBO問題を例に説明します。図3では、ABS2ソルバーとGurobi Optimizerそれぞれを単独で使用した場合と、両方を同時に利用した場合の、得られた最良解の時間推移が示されています。ABS2ソルバーを利用すると、探索の過程で $f(X)$ の値がより小さな解 X を発見し、解が徐々に改善されていく様子が見られます。一方で、Gurobi Optimizerを使用すると、比較的良い解が十数秒で得られますが、約960秒後に少しの改善した解を見つけた後、さらなる良い解を得ることができませんでした。この結果から、このQUBO問題において、短時間での実行ではGurobiの解探索性能が優れている一方で、長時間の解改善においてはABS2ソルバーの方が優れていると言えます。そこで、両方のソルバーを同時に実行し、それぞれのコールバックを利用して得られた解を共有することで、Gurobiが短時間で得た良い解を基にABS2ソルバーが探索を行い、良い解をより効率的に探索できるようになりました。

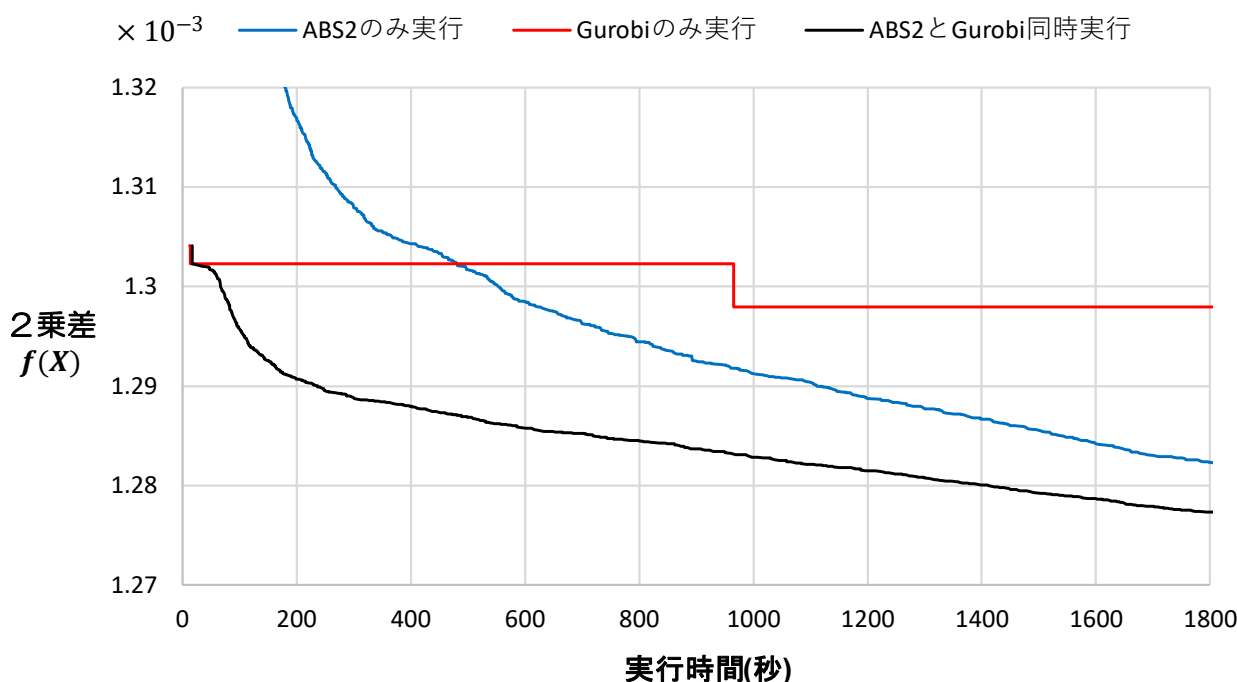


図3：(1)ABS2のみ実行、(2)Gurobiのみ実行、(3)両方を同時実行のそれぞれの場合の2乗差の合計 $f(X)$ の推移。GPU：NVIDIA A100（80GB）×8、CPU：Intel Xeon Gold 6338 CPU（2GHz）×2、メモリ：1TBを計測に使用。

【今後の展開】

ABS2 QUBOソルバーの性能向上を継続的に行うとともに、他のソルバーとの連携や、さまざまな組合せ最適化問題や実アプリケーションへの適用を目指します。

【用語解説】

*1 GPU (Graphics Processing Unit)

グラフィックス処理のための専用集積回路であるが、汎用計算が可能のように拡張されており、多数のプロセッサコアを用いた並列処理を行うことができるため、現在の多くのスーパーコンピュータに搭載されている。

*2 API (Application Programming Interface)

アプリケーションが他のソフトウェアと情報をやり取りするときに外部に提供する機能やデータへのアクセス方法を規定する仕様。

*3 コールバック機能

実行中に特定の条件を満たしたときにユーザーの設定した処理を行う機能。

【論文発表情報】

[1] Diverse Adaptive Bulk Search: A Framework for Solving QUBO Problems on Multiple GPUs. IPDPS Workshops 2023: 314-325

<https://doi.org/10.1109/IPDPSW59300.2023.00060>

[2] A benchmark QUBO problem inspired by digital halftoning based on the human visual system. CANDAR 2022: 56-65

<https://doi.org/10.1109/CANDAR57322.2022.00015>

【詳細情報】

ABS2 QUBO Solver: <https://abs2.cs.hiroshima-u.ac.jp/>

【お問い合わせ先】

大学院先進理工系科学研究科 中野 浩嗣

Tel : 082-424-5363 FAX : 082-424-5363

E-mail : nakano@hiroshima-u.ac.jp

発信枚数 : A4版 5枚 (本票含む)