Quantum computing and combinatorial optimization approach for solving Unsplittable Multi-commodity Flow Problem

Miguel Pineda Martín[1], Sébastien Martin[2]

[1] Huawei Technologies & co. 20 Quai du Point du Jour, Boulogne Billancourt, France
{ miguelpinedamartin}@gmail.com

[2] Huawei Technologies & co. 20 Quai du Point du Jour, Boulogne Billancourt, France
{sebastien.martin}@huawei.com

Mots-clés : *Quantum computing, Quantum Approximate Optimization Algorithm, Cutting Plane, Lagrangian Relaxation, Multi-Commodity Flow.*

Introduction

Quantum computing is a new interesting field with promising performance to tackle optimisation problems. However, the potential for a quantum advantage of these algorithms over highly-optimized classical solvers, such as CPLEX, must be explored empirically. But nowadays, despite of its actual fast development, quantum computing hardware has a limited memory (Qubits) and a limited quality,i.e. the size of the circuit that can be faithfully executed (Quantum Volume) [3]. Due to these limitations only small instances can be solved.

In the following we consider the Quantum Approximate Optimization Algorithm (QAOA), proposed on [1], based on the Quadratic Unconstrained Binary Optimization (QUBO) model. A typical approach to QAOA is mapping a combinatorial optimization problem into a binary integer linear program (BILP) and map it into a QUBO. For problems, like maximum cut problem, where this process can be easily done, QAOA has shown a good performance. However, QAOA doesn't have such a good performance on problems with highly restrictive constraints, indeed it usually returns unfeasible solutions. The main contribution of this paper is to show how to solve successively small QUBO problem to converge on a good feasible solution. We show some results on the unsplittable multi-commodity flow problem (UMCFP).

Unsplittable multi-commodity flow problem (UMCFP)

Let consider a graph $G = (V, A)$ where V is a set of vertices and A the set of arcs where for each arc $a \in A$ a weight w_a and a capacity c_a are considered. We also consider a set of commodities K where for each commodity $k \in K$, s_k represents the source, t_k the destination, b_k the size of the commodity and r_k the reward of the commodity. The unsplittable multi-commodity flow problem consists in maximizing the reward of accepted commodity minus the cost to route the commodity on the network such that for each accepted commodity a path is considered from the source to the destination and all arc capacities are respected.

Path based model

Using integer linear program we can model the UCMFP using path variables. This model has an exponential number of variables but thanks to the column generation algorithm we are able to solve the linear relaxation.

Path based model :

$$
\begin{aligned}
\min \quad & \sum_{k\in K}\sum_{p\in P_k}\left(\sum_{a\in A}\omega_a\right)\cdot x_p^k - \sum_{k\in K} r_k y_k && (1)\\
\text{s.t.} \quad & \sum_{p\in P_k} x_p^k = y_k && \forall k\in K \quad (2)\\
& \sum_{k\in\mathcal{K}}\sum_{p\in P_k : a\in p} b_k x_p^k \le c_a && \forall a\in A \quad (3)\\
& x_p^k, y_k \in \{0,1\} && \forall k\in K, p\in P_k, \quad (4)
\end{aligned}
$$

where P_k is the set of all paths for the commodity k, x_p^k is a binary variable equals to 1 if the path p is selected for the commodity k, 0 otherwise and y_k is a binary variable equals to 1 if the commodity k is accepted, 0 otherwise.

Quadratic Unconstrained Binary Optimization model
The classical transformation [2] of an integer linear program (ILP) in a QUBO model is given by

$$
\begin{aligned}
\min \quad & c^T x + M(Ax-b)^T(Ax-b)\\
\text{s.t.} \quad & x\in\{0,1\}^n,
\end{aligned}
$$

where additional slack variables are added to the original ILP to consider only equalities. Due to the number of variables, slack variables and constraints this transformation is not scalable.

Improvements
We propose some improvements to solve bigger instances.

- **Dedicated transformation :** the first improvement consists in removing inequalities $\sum_{p\in P_k} x_p^k = y_k$ for each commodity and considering the following objective function $\min \sum_{k\in K}\sum_{p\in P_k}\left(\sum_{a\in A}\omega_a\right)\cdot y_k x_p^k - \sum_{k\in K} r_k y_k x_p^k$. This preliminary transformation allows reducing the number of iterations for the QAOA to converge on a feasible solution.
- **Cutting plane method :** the second improvement focus on the size of the QUBO model. Reducing the size of the QUBO model allow using less QuBit and thus we can solve bigger instances. The goal is to add dynamically capacity constraints as cut-and-branch algorithm.
- **Lagrangian relaxation :** we propose an alternative QUBO reformulation based on the Lagrangian relaxation.
- **Pricing without linear relaxation :** thanks to the Lagrangian relaxation we can drive the column generation without considering the linear relaxation and the value of dual variables. This pricing procedure is a heuristic approaches since the QAOA produce a feasible (integer) solution.

We present some results to show the efficiency of our method to tackle the UMCFP using QAOA.

Références

[1] Edward Farhi, Jeffrey Goldstone, Sam Gutmann (2014). A Quantum Approximate Optimization Algorithm. `https://doi.org/10.48550/arXiv.1411.4028`

[2] Fred Glover, Gary Kochenberger, Yu Du. A Tutorial on Formulating and Using QUBO Models. `arXiv:1811.11538v6`

[3] Andrew Wack, Hanhee Paik, Ali Javadi-Abhari, Petar Jurcevic, Ismael Faro, Jay M. Gambetta, Blake R. Johnson. *Quality, Speed, and Scale : three key attributes to measure the performance of near-term quantum computers.* `arXiv:2110.14108`