Discrete optimization: Limitations of existing quantum algorithms

Stefan Creemers[1,2], Luis Fernando Pérez[1]

[1] IÉSEG School of Management
{s.creemers, l.perezarmas}@ieseg.fr
[2] KU Leuven (ORSTAT)
stefan.creemers@kuleuven.be

Keywords : *quantum computing, discrete optimization.*

1 Introduction

We investigate the limitations of existing quantum algorithms to solve discrete optimization problems. First, we discuss the quantum counting algorithm of Brassard et al. (1998), and show that it has performance that is equivalent to that of a brute-force approach when approximating the number of valid solutions. In addition, we show that a straightforward application of Grover's algorithm (referred to as `GUM` by Creemers and Pérez (2023b)) dominates any quantum counting algorithm when verifying whether a valid solution exists. Next, we discuss the nested quantum search algorithm of Cerf et al. (2000), and show that it is dominated by a classical nested search that uses an approach such as `GUM` to find (partial) solutions to (nested) problems. Last but not least, we also discuss amplitude amplification (a procedure that generalizes Grover's algorithm), and show (once more) that it may not be possible to outperform `GUM`.

2 Procedure `GUM` and its use to solve discrete optimization problems

`GUM` can be used as a subroutine in a procedure that solves a discrete optimization problem. For instance, Creemers and Pérez (2023a) use `GUM` in a binary search procedure (hereafter referred to as `BSP`) that solves the binary knapsack problem. Procedure `BSP` uses a binary search procedure that performs L iterations, where L is a logarithmic function of the difference between $V_{\min}$ and $V_{\max}$ (i.e., the upper and lower bounds on optimal solution value V^*, respectively). In each iteration, `BSP` uses `GUM` to assess whether a valid solution exists that satisfies the constraints of the optimization problem, and that has a value of at least $V = \lfloor 0.5(V_{\min} + V_{\max}) \rfloor$ (note that, to keep things simple, we assume integer solution values). If a solution $\mathbf{x}$ is found for which $f_{\mathbf{x}} = 1$, we update the best-found solution, and let $V_{\min} = (V + 1)$. If `GUM` is unable to find a valid solution, we let $V_{\max} = (V - 1)$. This process continues until $V_{\min} > V_{\max}$.

3 Counting algorithm of Brassard et al. (1998)

We discuss the quantum counting algorithm of Brassard et al. (1998; hereafter referred to as `QCB`). The goal of a quantum counting algorithm is to approximate the number of valid solutions (m) in a set of 2^n solutions. An (accurate) approximation of m is particularly useful when using Grover's algorithm: using m, we can determine the number of iterations (required by Grover's algorithm) that maximizes the probability to measure any of the m valid solutions. In addition, quantum counting algorithms can also be used to evaluate whether a solution exists (i.e., to approximate $P(m > 0)$). We show, however, that `QCB` requires 2^n Grover iterations to accurately approximate m (this results in a performance equivalent to that of a brute-force classical approach). In addition, we show that any quantum counting algorithm (i.e., not only `QCB`) is dominated by `GUM` when assessing whether $m > 0$.

4 Nested quantum search algorithm of Cerf et al. (2000)

We discuss the nested quantum search algorithm of Cerf et al. (2000; hereafter referred to as `NQC`). `NQC` is the quantum equivalent of a classical nested search algorithm, and allows to find a valid solution in a set of 2^n solutions using only $O(\sqrt{2^{\gamma n}})$ Grover iterations, where γ is some number less than or equal to 1 that depends on the structure of the problem. We have used `NQC` to solve a discrete optimization problem, and find that: (1) there may only be a low probability to find an optimal solution, and (2) it may not be possible to impose all constraints when performing a nested search. Unfortunately, we show that these problems cannot be resolved, and that a nested quantum search is dominated by a classical nested search that uses e.g., `GUM` to find (partial) solutions to (nested) problems.

5 Amplitude amplification

Last, but not least, we also discuss amplitude amplification. Amplitude amplification is a generalization of Grover's algorithm that tries to reduce the number of iterations (and hence operations) required to measure a valid solution. Whereas Grover's algorithm starts with a "uniform superposition" of qubits (i.e., a superposition where each qubit has an equal probability to collapse into either $|0\rangle$ or $|1\rangle$; a superposition where every solution has equal probability) and requires $\pi 4^{-1}\sqrt{m^{-1}2^n}$ iterations, amplitude amplification uses a superposition that allows to measure a valid solution using (far) less iterations. A "good" superposition uses information of the optimization problem in order to reduce the number of iterations required to measure a valid solution. Finding such a good superposition, however, may not be that easy. In fact, we show that, following logical rules, it is possible to end up with a bad superposition that performs worse than a uniform superposition.

References

[1] Brassard, G., Høyer, P., and Tapp, A. 1998. Quantum counting. In Larsen, K.G., Skyum, S., and Winksel, G. (Eds.), *Automata, Languages and Programming*, 820–831. Springer Berlin Heidelberg, Berlin, Heidelberg.

[2] Cerf, N. J., Grover, L. K., and Williams, C. P. 2000. Nested quantum search and structured problems. *Phys. Rev. A*, **61**(3), 1–14.

[3] Creemers, S., Pérez, L. 2023a. Discrete optimization: A quantum revolution (Part I), unpublished preprint at https://dx.doi.org/10.2139/ssrn.4198077.

[4] Creemers, S., Pérez, L. 2023b. Discrete optimization: A quantum revolution (Part II), unpublished preprint at https://dx.doi.org/10.2139/ssrn.4494506.

[5] Grover, L. K. 1996. A fast quantum mechanical algorithm for database search. *Proc. Annu. ACM Symp. Theory Comput.*, 212–219.