# NISQ computing for wireless communications

Olli Apilo • Erik Hieta-aho • Jorma Kilpi • Hannu Reittu • Kari Seppänen

## Introduction

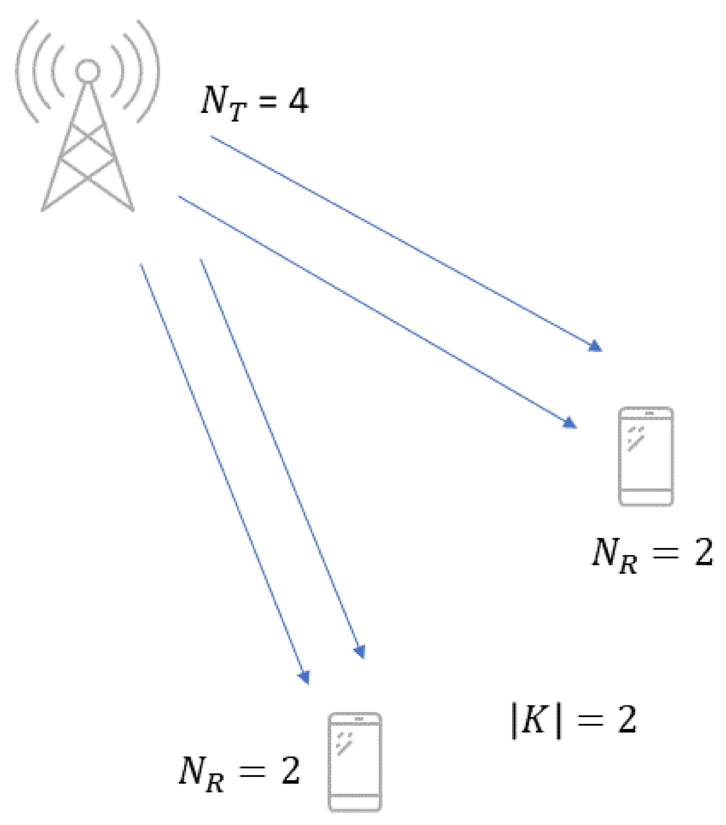
**Noisy Intermediate Scale Quantum (NISQ)** devices have empirical evidence of quantum computational supremacy as scale-up of a search space rather than speed-up performance [3]. In the context of wireless communications, we study combinatorial optimization problems that have a huge search space.

One problem with a huge search space is the **Multi-User Multiple Input Multiple Output (MU-MIMO)** downlink scheduling problem of 5G and future 6G networks. There exist classical algorithms [2] that are very fast in solving the MU-MIMO scheduling problem, but with strong heuristics and parallel computing. A quantum algorithm for the problem may allow to study the full search space and provide insight for even better classical algorithms.

The MU-MIMO radio communication has also other combinatorial problems that may benefit from NISQ computing, including **maximum likelihood based MIMO (ML-MIMO)** signal detection. The lattice based combinatorial problem that relates to ML-MIMO is known as the **shortest vector problem (SVP)**. While this is a quite general problem it impacts many areas of research in particular **post-quantum cryptography (PQC)** and the security of modern encryption systems.

## The MU-MIMO downlink scheduling problem

A base station must share its limited communication capacity fairly between a large number of **user equipments (UEs)**. Communication capacity consist of radio frequency and time.




MU-MIMO downlink scheduling problem.

### System model

The system model includes

- *variables* like the set of $K$ of all UEs that the base station serves,
- *parameters* like the set $B$ of resource blocks, and the set $M$ of possible modulation and coding schemes. A resource block $b \in B$ is the smallest element in scheduling.
- *constraints*, and
- binary *decision variables*.

In order to understand how big the solution space is, we need to consider the following numbers:

| | | |
|---|---|---|
| $\lvert K \rvert$ | The number of UEs | |
| $\lvert B \rvert$ | The number of resource blocks | $\lvert B \rvert \leq 273$ |
| $\lvert M \rvert$ | The number of modulation and coding schemes | $\lvert M \rvert \leq 28$ |
| $N_T$ | Max. number of transmitted data streams from the BS | $N_T = 8$ |
| $N_R$ | The number of antennas at an UE | $N_R < N_T$ |

Important parameters/variables for the MU-MIMO scheduling problem.

### Industrial use case

Our special focus is in an industrial application, with moving robots and their communication in a restricted factory area. The main simplification comes from the small number of UEs, moreover, $|K|$ is a more predictable variable. Assuming a digital twin we can also have more time for the computing.

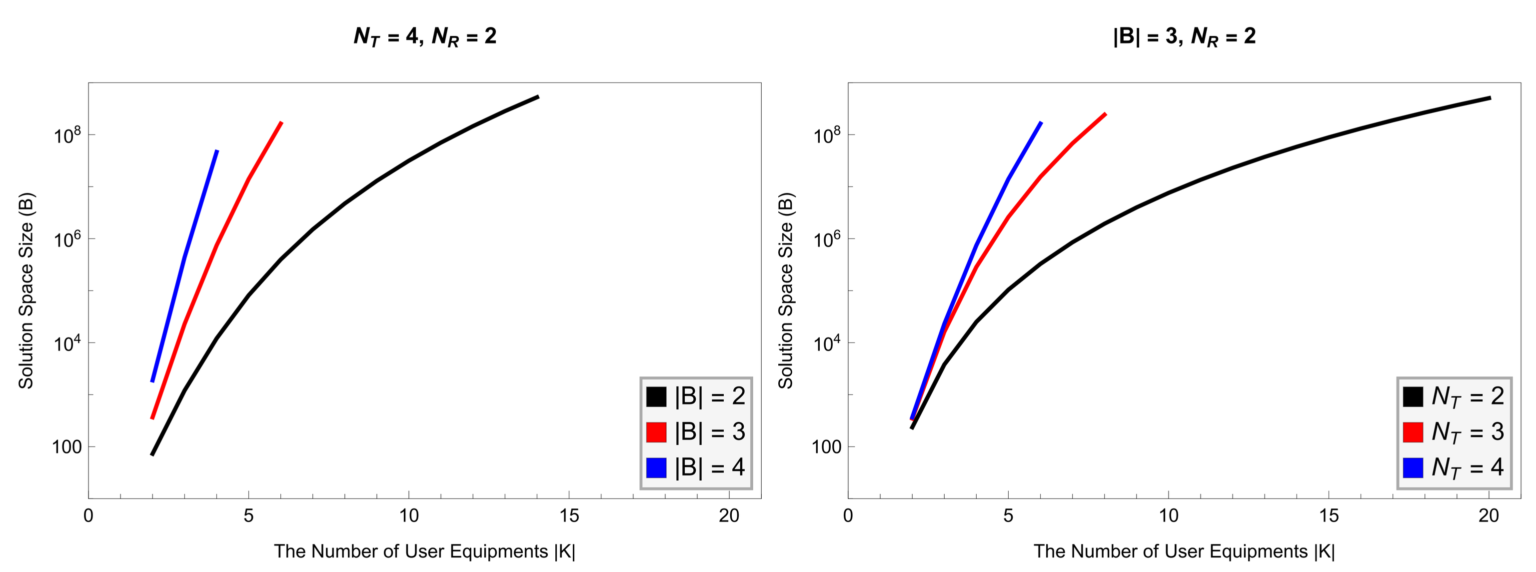
### The size of the search space

The number of scheduling choices can be computed with the formula [2]

$$(|M|N_R)^{|K|}|B| \left[ \binom{|K|}{1} + \cdots + \binom{|K|}{N_T} \right]$$

**General case:** With $|M| = 28$, $N_R = 2$, $|K| = 100$, $|B| = 273$, and $N_T = 8$, the search space of the general problem is approximately the order of $10^{189}$, see [2].

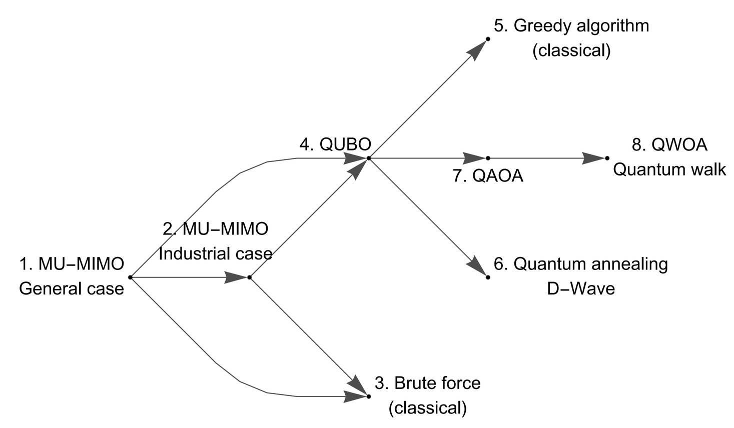
**Industrial use case:** With $|M| = 28$, $N_R = 4$, $|K| = 10$, $|B| = 273$, and $N_T = 8$, the search space of the general problem is approximately the order of $8.6 \cdot 10^{25}$.




Empirical complexity study with our brute force algorithm.

## Our approach

Our approach to solve the MU-MIMO scheduling problem has started with the following steps:




Our overall approach for the MU-MIMO scheduling problem.

1. We started with the existing MU-MIMO downlink scheduling problem formulation from [2].
2. We have specified a simpler scheduling problem in industrial use cases.
3. We have written a classical brute force algorithm for solving small instances of the problem. This algorithm is important for verification and benchmarking purposes.
4. We have formulated the scheduling problem in a **Quadratic Unconstrained Binary Optimization (QUBO)** form, both in general case and in our industrial use case. The constraints of the original problem are formulated as penalties for the QUBO.
5. A greedy algorithm for QUBOs has been developed and tested (around 30K binary variables).

## Our next steps

Our plan is to study the MU-MIMO scheduling problem in several different ways including the following:

6. Quantum annealing with D-Wave. We already have some experience in using D-Wave [4]. In addition to the scheduling problem, we also test the QUBO formulation of SVP [1] on D-Wave with a focus on determining the security levels of lattice-based PQC algorithms.
7. **Quantum Approximate Optimization Algorithm (QAOA)**. To begin we will estimate how many qubits we should have in a gate based quantum computer. We will also use quantum computing (Helmi) and quantum simulation (Kvasi) facilities through CSC.
8. **Quantum Walk assisted QAOA (QWOA)**. To be done in collaboration with quantum walk specialist(s).
9. We will test our classical greedy algorithm for the QUBO of the MU-MIMO scheduling problem (see item 5. above). This approach is interesting itself and it gives us further possibilities for benchmarking and comparisons against quantum computing as well as finding boundaries for quantum advantage.

Our study is funded by Business Finland, see `https://www.cohqca.fi/` for further information. Companies in the project steering group are Nokia Bell Labs, Unitary Zero Space and Cumucore.

Unitary Zero Space

Cumucore

VTT

## Summary

- **We are focusing on MU-MIMO downlink scheduling problem, which has small input, small output, and an enormous search space.**
- **We are interested both in general case scheduling problem and in the industrial use case problem. The latter is assumed to be significantly easier due to smaller number of UEs, predictability of robots' movements and communication, and the relaxed time bound for the computation due to digital twin.**
- **After the QUBO formulation of the MU-MIMO scheduling problem we have several ways to proceed with purely quantum, or quantum inspired classical algorithms.**
- **Similarly with the SVP QUBO formulation we will test the feasibility of solving SVP by adjusting parameters.**

## References

[1] Martin R. Albrecht, Miloš Prokop, Yixin Shen, and Petros Wallden. Variational quantum solutions to the shortest vector problem. Cryptology ePrint Archive, Paper 2022/233, 2022. `https://eprint.iacr.org/2022/233`.

[2] Yongce Chen, Yubo Wu, Y. Thomas Hou, and Wenjing Lou. mCore+: A real-time design achieving $\sim$ 500 $\mu$s scheduling for 5G MU-MIMO systems. *IEEE Transactions on Mobile Computing*, 22(12):7249–7265, December 2023.

[3] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout van den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, and Abhinav Kandala. Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618(7965):500–505, 2023.

[4] Hannu Reittu, Ville Kotovirta, Lasse Leskelä, Hannu Rummukainen, and Tomi Räty. Towards analyzing large graphs with quantum annealing. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2457–2464, 2019.

**www.vttresearch.com**

**Contact:** Kari Seppänen, Senior Scientist
Jorma Kilpi, Senior Scientist
Firstname.Lastname@vtt.fi

**beyond the obvious**