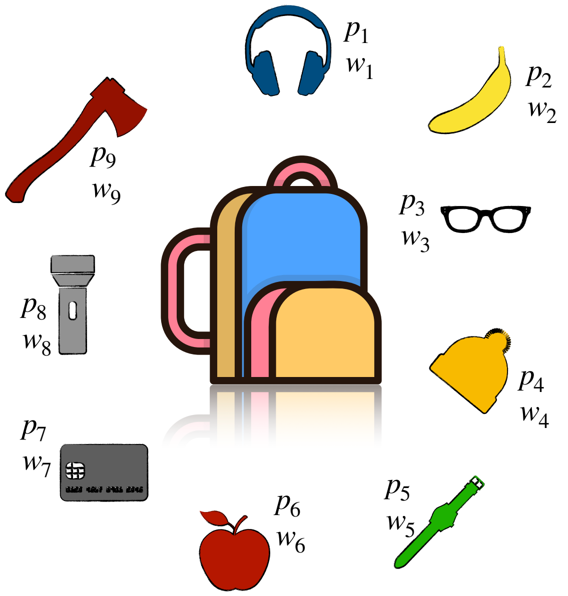
Topic 8: Optimization with tensor networks – The Knapsack Problem

What to expect

In this project, you will delve into modeling combinatorial optimization problems using the QUBO formalism and transforming them into a ground-state search of Ising-like Hamiltonians. You will learn how to obtain these ground states through QAOA. Additionally, you will explore tensor network methods, specifically quantum circuit emulation using Matrix Product State, and parallel computing on clusters, enabling scalability of the circuit to larger problem sizes. A basic understanding of quantum mechanics and Python is necessary, while familiarity with Qiskit and tensor networks is beneficial but not required.

Overview



Many problems of practical relevance can be framed as combinatorial optimization problems over discrete configuration spaces, where the goal is determining the optimal arrangement of a set of discrete variables to maximize or minimize a designated objective function. The Knapsack Problem (KP) is one of the simplest non-trivial binary models with versatile applications to various scenarios, ranging from financial decision-making, planning missions of a group of satellites in the field of Earth Observation and to logistics tasks in cargo transportation. The KP involves selecting a subset of items from a numbered list 1 to n, aiming to maximize total profit while adhering to a capacity constraint. A practical analogy for this problem is akin to preparing a mountaineer's backpack for a hike: among various desired items, each characterized by weight (increasing the backpack load upon selection) and benefit (measuring utility or comfort when carried), the mountaineer seeks to optimize benefit while limiting total weight. The maximum load, denoted by $\mathcal{C}$, sets the capacity of the backpack, representing the mountaineer's need to balance comfort against the constraint of carrying capacity.
Although simple to formulate, KP belongs to the complexity class of NP-hard problems[11], for which no efficient classical algorithms to obtain a solution with polynomial resources are known.

The simplest version of KP is mathematically defined as a Binary Linear Integer Program (BLIP), called the *integer-weight knapsack problem (IWKP)*, and consists in finding the optimal bitstring $\vec{x}_{op}$ that maximizes the linear *profit cost function*

$$\mathcal{V} = \sum_{j=1}^{n} p_j \, x_j \tag{3}$$

while the following *linear capacity constraint* is secured

$$\sum_{j=1}^{n} w_j \, x_j \leq \mathcal{C}. \tag{4}$$

Here, n is the number of items, $p_j > 0$ and $w_j > 0, w_j < \mathcal{C} \; \forall j = 1, \ldots, n$ are positive integers quantifying profit and weight associated to each object respectively; the non-triviality condition follows $\sum_{j=1}^{n} w_j > \mathcal{C}$.

While classical solvers for the KP are available, achieving optimal solutions for large n remains computationally challenging. Recently, there has been increasing interest in utilizing quantum computing techniques to tackle optimization problems like the KP. In this project we map the IWKP into an equivalent ground-state search of a many-body quantum Hamiltonian, making it suitable for quantum algorithms, such as the Quantum Approximate Optimization Algorithm (QAOA). The objective is to explore the potential benefits of this quantum optimization approach compared to state-of-the-art classical algorithms. Furthermore, we use Tensor Network (TN) methods to simulate larger instances of the KP.

Tasks

- Solve the binary linear integer problem (BLIP) brute-force (generating all the possible bitstrings) as benchmark;
- Solve the BLIP using CPLEX (or another classical solver) as benchmark;
- Translate the BLIP to the corresponding Ising spin glass Hamiltonian (QUBO + binary-to-spin);
- Implement the QAOA variational quantum circuit in Qiskit;
- Solve the ground state search with QAOA;
- Scale up the problem to larger instances with Matcha TEA.

[11] H. Kellerer et al., *Knapsack Problems* (Springer Science & Business Media, Mar. 2013).