HYBRID QUANTUM-CLASSICAL COMPUTING FOR FUTURE NETWORK OPTIMIZATION

Lei Fan[1] and Zhu Han[2],
Thanks to Zhongqi Zhao[1,2], Wenlu Xuan[1,2], and Mingze Li[1,2]

Engineering Technology[1] and Electrical and Computer Engineering[2]

Nov 28 2022

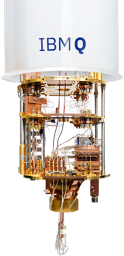
Motivation

Started in the 1980s

Feynman said, maybe we need to use quantum mechanics in our computers. [1]

Many years later

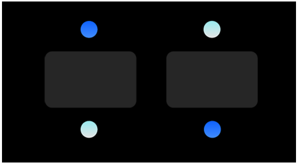
Quantum Computer



...

What is wrong in classic computing?

Dinner Party
But with only **ONE** optimal seating plan



Total combinations:
2

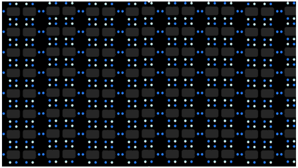
When there are only 2 guests attend,

the total seating plan could be calculated by a permutation equation

$$N^2_{\text{seating plan}} = P^2_2 = 2! = 2.$$

What is wrong in classic computing?

Dinner Party
But with only **ONE** optimal seating plan



Total combinations: 120

There are 5 guests showed up.

The total seating plan could be calculated by a permutation equation

$$N^5_{\text{seating plan}} = P^5_5 = 5! = 120.$$

What is wrong in classic computing?

Dinner Party
But with only **ONE** optimal seating plan

Total combinations: 3,628,800

Now, 10 guests rush in to the party.

The total seating plan could be calculated by a permutation equation

$$N^{10}_{\text{seating plan}} = P^{10}_{10} = 10! = 3,628,800.$$

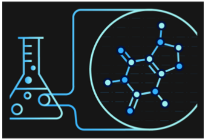
It is hard for us to figure out the optimal solution from a tremendous possible choices.

What is wrong in classic computing?

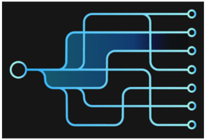
Dinner Party
But with only **ONE** optimal seating plan

Total combinations:
3,628,800

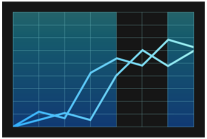
Similar things also happen in these fields.



Life science

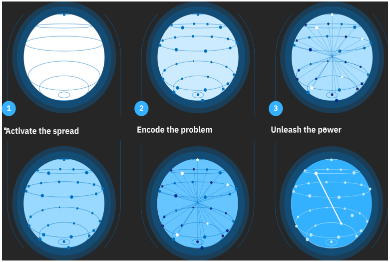


Manufacturing & Logistics



Financial Services

What is wrong in classic computing?

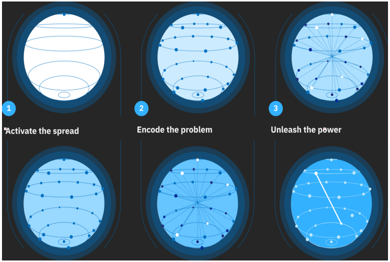


The classical computer has to go through every combination in sequential to sort out the optima.

However, quantum computer can achieve the same in 3 steps.

1. The machine is activated by creating an equal superposition of all 2^n states.
2. The problem is encoded onto the system by applying gates or a magnet field.
3. The machine comes to a solution by using physical principles of interference to magnify the amplitude (possibility) of the correct answer and shrink the incorrect answers. Some problems require iterating steps of 2 and 3

What is wrong in classic computing?



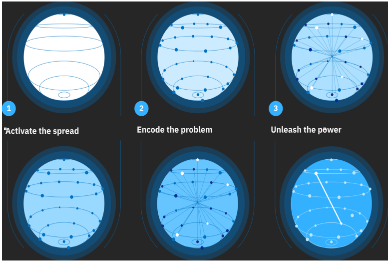
Now, the Party seating plan with 10 guest will use

$$\log_2 3628800 = 21.79 \approx 22 \text{ qubits},$$

to encode the problem and computing the

correct answer in parallel.

What is wrong in classic computing?



Quantum computers
Can

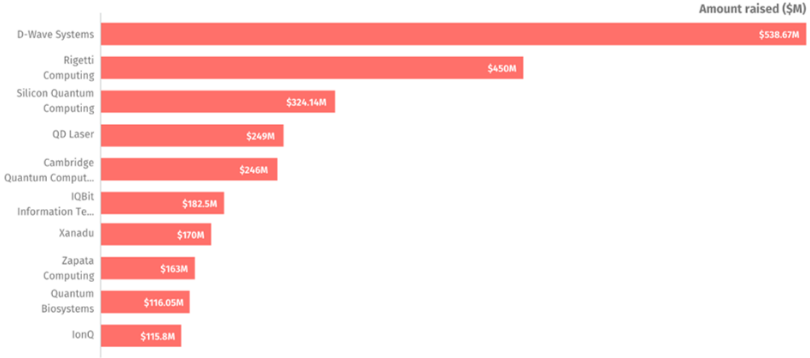
create vast multidimensional spaces to deal with
large problems,

and translate them back into what we can use,

while
classical computers
may have
difficulties
to do the same.

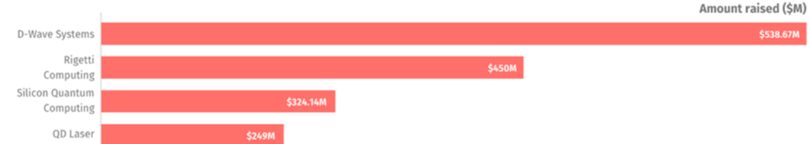
Quantum Computing is Booming

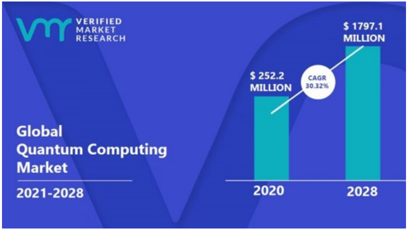
Top Funded Companies



Quantum Computing is Booming

Top Funded Companies





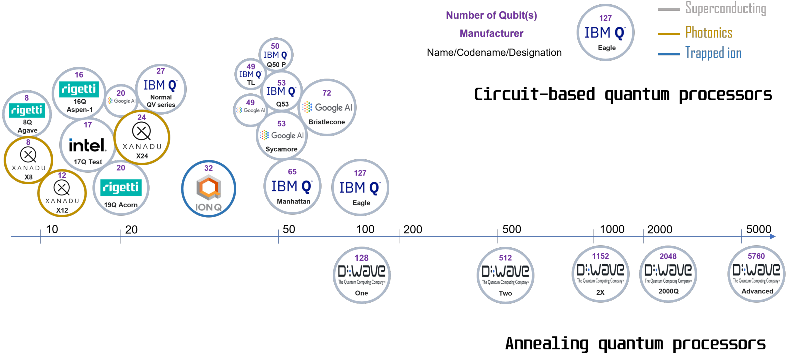
Known Types of Quantum Computing and Their Applications and Generality.

Quantum Annealer	Analog Quantum	Universal Quantum
A very specialized form of quantum computing with unproven advantages over other specialized forms of conventional computing.	The most likely form of quantum computing that will first show true quantum speedup over conventional computing.	The true grand challenge in quantum computing. It offers the potential to be exponentially faster than tradition computers for a number of important applications for science and businesses.

Known Types of Quantum Computing and Their Applications and Generality.

	Quantum Annealer	Analog Quantum	Universal Quantum
Difficulty	⋆	⋆ ⋆ ⋆	⋆ ⋆ ⋆ ⋆ ⋆
Application	Optimization	Chemistry Sampling Quantum Dynamics	Cryptography Searching Securing Computing
Generality	Restrictive	Partial	High

Rank of quantum processors



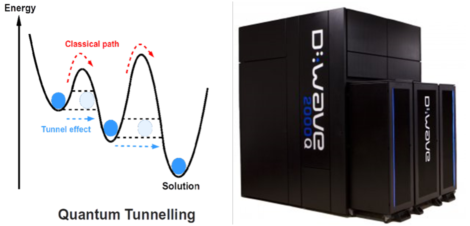
D-WAVE quantum annealer computer fits our problem setting the most.

Classification

Why Quantum Annealing?

Quantum Computer

- Gate Model
- Analog Quantum Model
- Quantum Annealing



Superposition

- A quantum bit $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ in a superposition,
- $||\alpha||^2$: the probability in state $|0\rangle$
- $||\beta||^2$: the probability in state $|1\rangle$
- $||\alpha||^2 + ||\beta||^2 = 1$

Tensor product and multi-qubits

- $|\psi\rangle \otimes |\phi\rangle$ represents the overall state of two quantum bits.
- $$|\psi\rangle \otimes |\phi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle,$$
  $$||\alpha_{00}||^2 + ||\alpha_{01}||^2 + ||\alpha_{10}||^2 + ||\alpha_{11}||^2 = 1.$$
- A general *n*-qubit system:
  - $$|\Phi\rangle = \sum_{i=0}^{n} \alpha_i |i\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \cdots \\ \alpha_{n-1} \\ \alpha_n \end{pmatrix}.$$
- $|i\rangle$ is the i^{th} computational basis of the space, and α_i is the amplitude of the i^{th} computational basis.

Schrödinger Equation

Schrödinger Equation

- Evolution from $t = 0$ to $t = T$
- $i\frac{\partial|\psi(t)\rangle}{\partial t} = H(t)|\psi(t)\rangle$
- $|\psi(t)\rangle$: The actual state of the system
- $H(t)$: a time-dependent Hamiltonian (i.e., Kinetic Energy + Potential Energy)
- $H(t)|\phi_j(t)\rangle = E_j(t)|\phi_j(t)\rangle$
- $|\phi_j(t)\rangle$: the $j-$th instantaneous eigenstate
- $E_j(t)$: the $j-$th instantaneous eigenvalue

Adiabatic Quantum Computing Overview

- A computing approach utilizing the quantum mechanics (e.g., superposition, entanglement).
- Prepare the system in a initial state and transform it to the final state.
- Has the potential to speed up the computing process.
- Polynomial equivalent to circuit model
- Applications: PageRank algorithm, Quadratic Unconstrained Binary Optimization, Machine Learning.

Adiabatic Quantum Computing Algorithm [2][3]

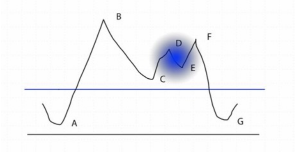
- 1: Encoding the solution
  - 1) Encoding target solution in final state $|\phi_j(T)\rangle, j = 0$.
  - 2) Encoding target function in the final eigenvalue $E_j(T)$.
  - 3) Find the Hamiltonian $H(T) = H_{fin}$ based the encoding rules.
- 2: Prepare the initial Hamiltonian $H(0) = H_{ini}$ and its eigenstates $|\phi_j(0)\rangle, j = 0$.
- 3: Prepare the time dependent Hamiltonian $H(t) = (1 - f(t))H_{ini} + f(t)H_{fin}$, $f(0) = 0, f(1) = 1, 0 \leq f(t) \leq 1, fort \in [0, T]$. function $f(t)$ is at least twice differentiable.
- 4: Evolve the system from $t = 0$ to $t = T$, then observe the final state to obtain the solution.

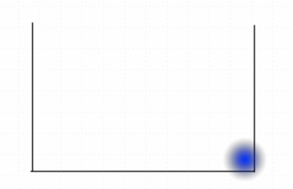
Quantum Annealing

- Annealing a Metal
  - Heat the metal to a temperature
  - Lower the temperature
- Simulated Annealing
  - Heuristic, random search method.
- Quantum Annealing
  - A relaxed QAC approach
  - Work in finite temperature and in open environments.

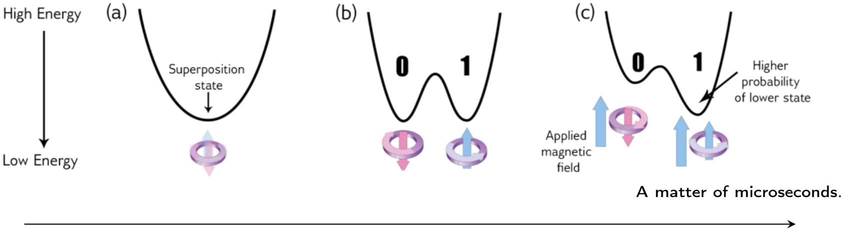
Simulated Annealing VS Quantum Annealing

- Quantum Tunneling
- Enables jumping from one classical state to another
    - Decreases likelihood of getting stuck in a local minimum
- Width of energy barrier is important, but height is not





Energy diagram in Quantum Annealing



- Initial Qubits
- Superposition at $|0\rangle$s and $|1\rangle$s.
- Not yet coupled

- Qubits are entangled
- At state of many possible answers
- Couplers & biases applied

- Inputs' energy are set.
- Lowest energy is at or closes to the optima.
- Energy $\rightarrow$ possibility

Ising Model

Spins interact with applied (external) field

Neighboring spins interact with each other

$$H(t) = \sum_{j} h_j \sigma_z^j + \sum_{\langle i,j \rangle} J_{i,j} \sigma_z^i \sigma_z^j .$$

- QA algorithm use Ising Model as its final Hamiltonian.
- σ_z^j is the Pauli Z operator .
- $J_{i,j}$ represents the coupling strength between qubits i, j.
- h_j is the local bias on qubit i.

QUBO

$$f(x) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} x_i Q_{i,j} x_j \quad x \in \{0,1\}^n.$$

These important optimization problems can be transformed into QUBO model:

- Knapsack Problems
- Assignment Problems
- Task Allocation Problems
- Capital Budgeting Problems
- . . . (NP-hard problem)

→

Quantum Computing:
provide an alternative method to solve some NP-hard problems

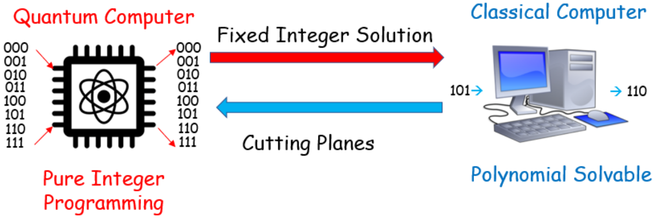
QUBO as Ising Model

Core Idea

- Encoding the objective function $f(x)$ as the eigenvalue of the ground state.
- $H_{fin}|x\rangle = f(x)|x\rangle$, where $|x\rangle = |x_{n-1}\dots x_0\rangle$
- $\sigma_z|0\rangle = |0\rangle$, $\sigma_z|1\rangle = -|1\rangle$
- $\sigma_z|x_j\rangle = (1-2x_j)|x_j\rangle, x_j \in \{0,1\}$
- $\Sigma_z^j = 1^{\otimes n-1-j} \otimes \sigma_z \otimes 1^{\otimes j}$, for $j \in \{0, n-1\}$
- $\Sigma_z^j|x_j\rangle = (1-2x_j)|x_j\rangle, x \in \{0,1\}$

Hamiltonians for QUBO

- $$H_{\text{ini}} = \sum_{j=0}^{n-1} \Sigma_z^j$$
- $$H_{\text{final}} = \sum_{j=0}^{n-1} K_j \Sigma_z^j + \sum_{\substack{i,j=0 \\ i \neq j}}^{n-1} J_{ij} \Sigma_z^i \Sigma_z^j + c 1^{\otimes n}$$
  - $J_{ij} = \frac{1}{4} Q_{ij}$ for $i \neq j$
  - $$K_j = -\frac{1}{4} \sum_{\substack{i,j=0 \\ i \neq j}}^{n-1} (Q_{ij} + Q_{ji}) - \frac{1}{2} Q_{jj}$$
  - $$c = \frac{1}{4} \sum_{\substack{i,j=0 \\ i \neq j}}^{n-1} Q_{ji} + \frac{1}{2} \sum_{0}^{n-1} Q_{jj}$$
- Optimal objective value : eigenvalue of ground state.
- Optimal solution obtained : final ground state.



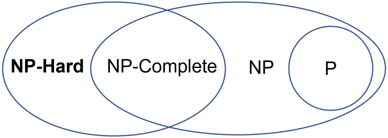
- Divide the mixed-integer convex problem into two parts.
  - Pure integer part: solved by the quantum computer.
  - Polynomial solvable continuous part: convex optimization algorithms.
- Obtain solutions of integer variables from quantum computer.
- Generate cutting planes from classical computer.

Structure of MILP

Mixed-integer Linear Programming is of:

$$\max_{x,y} c^T x + h^T y$$
$$\text{s.t. } Ax + Gy \leq b,$$
$$x \in X, x \in \{0,1\}^n,$$
$$y \in \mathbb{R}^p.$$

- Mixed-Integer linear Programming (MILP) is **NP-Hard**.
- It can't be solved in polynomial time unless P = NP.



Problem type	Example Problem
NP-Hard	Matrix Permanent Turing Halting Problem **MILP**
NP-Complete	Steiner Tree Graph 3-coloring Maximum Clique
NP	Factoring Graph Isomorphism
P	Linear Programming Graph Connectivity

Benders' Decomposition Introduction

Consider a Mixed-integer Linear Programming,

$$\max_{x,y} c^T x + h^T y \quad \text{s.t. } Ax + Gy \leq b, \; x \in X, x \in \{0,1\}^n, \; y \in \mathbb{R}_+^p.$$

$\xrightarrow{z_{LP} \text{ Replacement}}$

$$\max_{x} \; c^T x + z_{LP}(x) \quad \text{s.t.} \; x \in X, \; x \in \{0,1\}^n$$

We denote the value of the best choice for y by $z_{LP}(x)$

$$z_{LP}(x) = \max_{y} \; h^T y \quad \text{s.t.} \; Gy \leq b - Ax, \; y \in \mathbb{R}_+^p.$$

$\xrightarrow{\text{LP Duality}}$

$$z_{LP}(x) = \min_{u} (b - Ax)^T u \quad \text{s.t. } G^T u \geq h, \; u \in \mathbb{R}_+^m.$$

Feasible Region Q

Benders' Decomposition Introduction

$$z_{LP}(x) = \min_{u} (b - Ax)^\top u \quad \text{s.t. } G^\top u \geq h, \quad u \in \mathbb{R}_+^m$$

$$\max_{x} \quad c^\top x + z_{LP}(x) \quad \text{s.t.} \quad x \in X, \; x \in \{0,1\}^n$$

Feasible region Q does not depend on x.

Extreme points: $u^k, \; k \in K$, $\xrightarrow{z_{LP} \text{ bounded}}$ s.t. $(b - Ax)^\top u^k \geq z_{LP}(x)$,

Extreme rays: $r^j, \; j \in J$, $\xrightarrow[z_{LP}=-\infty]{}$ $(b - Ax)^\top r^k \geq 0$,

$$z_{LP}(x) \in \mathbb{R}.$$

Benders' Decomposition Introduction

$$\begin{aligned} \max_{\mathrm{x}} \quad & \mathrm{c}^\intercal \mathrm{x} + z_{LP}(\mathrm{x}) \\ \text{s.t.} \quad & (\mathrm{b} - \mathrm{A}\mathrm{x})^\intercal u^k \geq z_{LP}(\mathrm{x}) \quad \text{for } k \in K, \\ & (\mathrm{b} - \mathrm{A}\mathrm{x})^\intercal r^j \geq 0 \quad \text{for } j \in J, \\ & z_{LP}(\mathrm{x}) \in \mathbb{R},\ \mathrm{x} \in \mathrm{X},\ \mathrm{x} \in \{0,1\}^n . \end{aligned}$$

$$\begin{aligned} \max_{\mathrm{x}} \quad & \mathrm{c}^\intercal \mathrm{x} + z_{LP}(\mathrm{x}) \\ \text{s.t.} \quad & \mathrm{x} \in \mathrm{X},\ \mathrm{x} \in \{0,1\}^n . \end{aligned}$$

feasible region Q does not depend on x.

Extreme points: $u^k,\ k \in K$, $\xrightarrow{z_{LP} \text{ bounded}}$ s.t. $(\mathrm{b} - \mathrm{A}\mathrm{x})^\intercal u^k \geq z_{LP}(\mathrm{x})$,

Extreme rays: $r^j,\ j \in J$, $\xrightarrow[z_{LP}=-\infty]{}$ $(\mathrm{b} - \mathrm{A}\mathrm{x})^\intercal r^k \geq 0$,

$z_{LP}(\mathrm{x}) \in \mathbb{R}$.

Benders' Decomposition Introduction

$$\begin{aligned} \max_{\mathrm{x}} \quad & \mathrm{c}^\intercal \mathrm{x} + z_{LP}(\mathrm{x}) \\ \text{s.t.} \quad & (\mathrm{b} - \mathrm{A}\mathrm{x})^\intercal u^k \geq z_{LP}(\mathrm{x}) \quad \text{for } k \in K, \\ & (\mathrm{b} - \mathrm{A}\mathrm{x})^\intercal r^j \geq 0 \quad \text{for } j \in J, \\ & z_{LP}(\mathrm{x}) \in \mathbb{R},\ \mathrm{x} \in \mathrm{X},\ \mathrm{x} \in \{0,1\}^n . \end{aligned}$$

$$\begin{aligned} z_{LP}(\mathrm{x}) = \min_{\mathrm{u}} & (\mathrm{b} - \mathrm{A}\mathrm{x})^\intercal \mathrm{u} \\ \text{s.t. } & \mathrm{G}^\intercal \mathrm{u} \geq \mathrm{h}, \\ & \mathrm{u} \in \mathbb{R}^m_+ . \end{aligned}$$

Replace $z_{LP}(\mathrm{x})$ with symbol t.

$$\begin{aligned} \max_{\mathrm{x},\ t} \quad & \mathrm{c}^\intercal \mathrm{x} + t \\ \text{s.t.} \quad & (\mathrm{b} - \mathrm{A}\mathrm{x})^\intercal u^k \geq t \quad \text{for } k \in K, \\ & (\mathrm{b} - \mathrm{A}\mathrm{x})^\intercal r^j \geq 0 \quad \text{for } j \in J, \\ & t \in \mathbb{R},\ \mathrm{x} \in \mathrm{X},\ \mathrm{x} \in \{0,1\}^n . \end{aligned}$$

solution x →

← Feasible region Q
Either extreme ray
or point x

$$\begin{aligned} \min_{\mathrm{u}} \quad & (\mathrm{b} - \mathrm{A}\mathrm{x})^\intercal \mathrm{u} \\ \text{s.t.} \quad & \mathrm{G}^\intercal \mathrm{u} \geq \mathrm{h}, \\ & \mathrm{u} \in \mathbb{R}^m_+ . \end{aligned}$$

Benders' Decomposition Algorithm:

Algorithm:

- Determine (possibly empty) initial sets $\hat{K}$ of extreme points and $\hat{J}$ of extreme rays of Q.
- Solve the (modified) master problem, the relaxation of the Benders reformulation. Obtain solution $\bar{x}$ and corresponding $\bar{t}$.
- Determine $z_{LP}(\bar{x})$ by solving the dual of the subproblem.
- If $\bar{z}_{LP} = -\infty$, an extreme ray of Q has been found. Add the extreme ray to $\hat{J}$ and return to Step 2. (Feasibility Cuts).
- If $z_{LP}(\bar{x}) < \bar{t}$ and finite, Add the extreme point of Q to $\hat{K}$ and return to Step 2. (Optimality Cuts)
- If $z_{LP}(\bar{x}) = \bar{t}$ then $\bar{x}$ solves the original mixed integer program (1), with optimal y equal to the solution to the primal subproblem (2) with $x = \bar{x}$.

$$\begin{aligned}
\max_{\mathbf{x},\ t} \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} + t \\
\text{s.t.} \quad & (\mathbf{b} - \mathbf{A}\mathbf{x})^{\mathsf{T}}\, u^k \geq t \quad \text{for } k \in K, \\
& (\mathbf{b} - \mathbf{A}\mathbf{x})^{\mathsf{T}}\, r^j \geq 0 \quad \text{for } j \in J, \\
& t \in \mathbb{R},\ \mathbf{x} \in \mathbf{X},\ \mathbf{x} \in \{0,1\}^n .
\end{aligned}$$

$\downarrow\uparrow$

$$\begin{aligned}
\min_{\mathbf{u}} \quad & (\mathbf{b} - \mathbf{A}\mathbf{x})^{\mathsf{T}}\, \mathbf{u} \\
\text{s.t.} \quad & \mathbf{G}^{\mathsf{T}}\mathbf{u} \geq \mathbf{h}, \\
& \mathbf{u} \in \mathbb{R}^m_+ .
\end{aligned}$$

Benders' Decomposition to QUBO

Classical Benders' Decomposition (CBD)

$$\max_{x,\, t} \quad c^{\mathsf{T}} x + t$$
$$\text{s.t.} \quad (b - Ax)^{\mathsf{T}} u^k \geq t \quad \text{for } k \in K,$$
$$(b - Ax)^{\mathsf{T}} r^j \geq 0 \quad \text{for } j \in J,$$
$$t \in \mathbb{R},\ x \in X,\ x \in \{0,1\}^n.$$

Solution x →

← Feasible region Q
Either extreme ray
or point x

$$\min_{u} \quad (b - Ax)^{\mathsf{T}} u$$
$$\text{s.t.} \quad G^{\mathsf{T}} u \geq h,$$
$$u \in \mathbb{R}_+^m.$$

QUBO (Quadratic Unconstrained Binary Optimization)

$$Q_{\text{obj}} = \sum_i x_i Q_{i,i} x_i + \sum_i \sum_{i<j} Q_{i,j} x_i x_j.$$

Q_{obj} : Upper triangular matrix x_i : Binary variable

Master problem of **CBD** is

one step away from pure **ILP**
(Integer-linear programming).

The **last** barrier is
the scalar t.

Benders' Decomposition to QUBO

Classical Benders' Decomposition Master Problem

$$\begin{aligned} \max_{\mathrm{x},\ t} \quad & \mathrm{c}^{\intercal}\mathrm{x} + t \\ \text{s.t.} \quad & (\mathrm{b} - \mathrm{Ax})^{\intercal}\, u^{k} \geq t, \quad \text{for } k \in K, \\ & (\mathrm{b} - \mathrm{Ax})^{\intercal}\, r^{j} \geq 0, \quad \text{for } j \in J, \\ & t \in \mathbb{R},\ \mathrm{x} \in \mathrm{X}. \end{aligned}$$

In order to reformulate the master problem into the QUBO formulation,

we use a **binary** vector w with length of $M = \overline{m}_{+} + \overline{m}_{-} + \underline{m} + 1$ bit(s) to replace the continuous variable t.

$$\begin{aligned} t &= \sum_{i=-\underline{m}}^{\overline{m}_{+}} 2^{i} w_{i+\underline{m}} - \sum_{j=0}^{\overline{m}_{-}} 2^{j} w_{j+(1+\underline{m}+\overline{m}_{+})}, \\ &= \overline{t}\,(\mathrm{w})\,. \end{aligned}$$

$\overline{m}_{+}$: # of bits of $\mathbb{N}$ part.
$\underline{m}$: # of bits of the decimal part.
$\overline{m}_{-} + 1$: # of bits of $\mathbb{Z}_{-}$ part.

Benders' Decomposition to QUBO

Classical Benders' Decomposition Master Problem

$$\begin{aligned} \max_{\mathbf{x},\mathbf{w}} \quad & c^{\mathsf{T}}\mathbf{x} + \sum_{i=-\underline{m}}^{\overline{m}_+} 2^i w_{i+\underline{m}} - \sum_{j=0}^{\overline{m}_-} 2^j w_{j+(\mathbf{1}+\underline{m}+\overline{m}_+)} \\ \text{s.t.} \quad & (b - A\mathbf{x})^{\mathsf{T}} u^k \geq \overline{t}(\mathbf{w}), \quad \text{for } k \in \hat{K}, \\ & (b - A\mathbf{x})^{\mathsf{T}} r^j \geq 0, \quad \text{for } j \in \hat{J}, \\ & \mathbf{x} \in X, \quad \mathbf{x} \in \{0,1\}^n, \\ & \mathbf{w} \in W, \quad \mathbf{w} \in \{0,1\}^M. \end{aligned}$$

MILP

$\xrightarrow[t \text{ reformulation}]{}$

Alternative Benders' Decomposition Master Problem

$$\begin{aligned} \max_{\mathbf{x},\mathbf{w}} \quad & c^{\mathsf{T}}\mathbf{x} + \sum_{i=-\underline{m}}^{\overline{m}_+} 2^i w_{i+\underline{m}} - \sum_{j=0}^{\overline{m}_-} 2^j w_{j+(\mathbf{1}+\underline{m}+\overline{m}_+)} \\ \text{s.t.} \quad & (b - A\mathbf{x})^{\mathsf{T}} u^k \geq \overline{t}(\mathbf{w}), \quad \text{for } k \in \hat{K}, \\ & (b - A\mathbf{x})^{\mathsf{T}} r^j \geq 0, \quad \text{for } j \in \hat{J}, \\ & \mathbf{x} \in X, \quad \mathbf{x} \in \{0,1\}^n, \\ & \mathbf{w} \in W, \quad \mathbf{w} \in \{0,1\}^M. \end{aligned}$$

Pure ILP

QUBO can be applied now.

Hybrid Quantum-classical Benders' Decomposition

Alternative Benders' Decomposition Master Problem

$$\begin{aligned}
\max_{\mathrm{x},\mathrm{w}} \quad & c^{\mathsf{T}}\mathrm{x} + \sum_{i=-\underline{m}}^{\overline{m}_+} 2^i w_{i+\underline{m}} - \sum_{j=0}^{\overline{m}_-} 2^j w_{j+(1+\underline{m}+\overline{m}_+)} \\
\text{s.t.} \quad & (b - A\mathrm{x})^{\mathsf{T}} u^k \geq \overline{t}(\mathrm{w}), \quad \text{for } k \in \hat{K}, \\
& (b - A\mathrm{x})^{\mathsf{T}} r^j \geq 0, \quad \text{for } j \in \hat{J}, \\
& \mathrm{x} \in X, \quad \mathrm{x} \in \{0,1\}^n, \\
& \mathrm{w} \in W, \quad \mathrm{w} \in \{0,1\}^M.
\end{aligned}$$

Hybrid Quantum-classical Benders' Decomposition

Hybrid Quantum-classical Benders' Decomposition Master Problem

$$\begin{aligned}
\max_{\mathrm{x},\mathrm{w}} \quad & c^{\mathsf{T}}\mathrm{x} + \sum_{i=-\underline{m}}^{\overline{m}_+} 2^i w_{i+\underline{m}} - \sum_{j=0}^{\overline{m}_-} 2^j w_{j+(1+\underline{m}+\overline{m}_+)} \\
\text{s.t.} \quad & (b - A\mathrm{x})^{\mathsf{T}} u^k \geq \overline{t}(\mathrm{w}), \quad \text{for } k \in \hat{K}, \\
& (b - A\mathrm{x})^{\mathsf{T}} r^j \geq 0, \quad \text{for } j \in \hat{J}, \\
& \mathrm{x} \in X, \quad \mathrm{x} \in \{0,1\}^n, \\
& \mathrm{w} \in W, \quad \mathrm{w} \in \{0,1\}^M.
\end{aligned}$$

Hybrid Quantum-classical Benders' Decomposition

Hybrid Quantum-classical Benders' Decomposition
Master Problem

$$\max_{\mathbf{x},\mathbf{w}} \quad c^{\mathsf{T}}\mathrm{x} + \sum_{i=-\underline{m}}^{\overline{m}_+} 2^i w_{i+\underline{m}} - \sum_{j=0}^{\overline{m}_-} 2^j w_{j+(1+\underline{m}+\overline{m}_+)}$$

$$\text{s.t.} \quad (b - A\mathrm{x})^{\mathsf{T}} u^k \geq \overline{t}(\mathrm{w}), \quad \text{for } k \in \hat{K},$$

$$(b - A\mathrm{x})^{\mathsf{T}} r^j \geq 0, \quad \text{for } j \in \hat{J},$$

$$\mathrm{x} \in X, \quad \mathrm{x} \in \{0,1\}^n,$$

$$\mathrm{w} \in W, \quad \mathrm{w} \in \{0,1\}^M.$$

Constraint	Equivalent Penalty
$x_1 + x_2 = 1$	$P(x_1 + x_2 - 1)^2$
$x_1 + x_2 \geq 1$	$P(1 - x_1 - x_2 + x_1 x_2)^2$
$x_1 + x_2 \leq 1$	$P(x_1 x_2)$
$x_1 + x_2 + x_3 \leq 1$	$P(x_1 x_2 + x_1 x_3 + x_2 x_3)$

Table of Common Constraint-penalty Pairs

(1) Objective Function:

$$c^{\mathsf{T}}\mathrm{x} + \sum_{i=-\underline{m}}^{\overline{m}_+} 2^i w_{i+\underline{m}} - \sum_{j=0}^{\overline{m}_-} 2^j w_{j+(1+\underline{m}+\overline{m}_+)}$$

$$\Longrightarrow$$

$$Q_{obj} = \mathrm{x}^{\mathsf{T}}\mathrm{diag}(c)\,\mathrm{x} + \sum_{i=-\underline{m}}^{\overline{m}_+} w_{i+\underline{m}} 2^i w_{i+\underline{m}} - \sum_{j=0}^{\overline{m}_-} w_{j+(1+\underline{m}+\overline{m}_+)} 2^j w_{j+(1+\underline{m}+\overline{m}_+)}.$$

Hybrid Quantum-classical Benders' Decomposition

Hybrid Quantum-classical Benders' Decomposition

Master Problem

$$\max_{\mathbf{x},\mathbf{w}} \quad c^{\mathsf{T}}\mathbf{x} + \sum_{i=-\underline{m}}^{\overline{m}_+} 2^i w_{i+\underline{m}} - \sum_{j=0}^{\overline{m}_-} 2^j w_{j+(1+\underline{m}+\overline{m}_+)}$$

$$\begin{aligned}
\text{s.t.} \quad & (b - A\mathbf{x})^{\mathsf{T}} u^k \geq \overline{t}(\mathbf{w}), \quad \text{for } k \in \hat{K}, \\
& (b - A\mathbf{x})^{\mathsf{T}} r^j \geq 0, \quad \text{for } j \in \hat{J}, \\
& \mathbf{x} \in X, \quad \mathbf{x} \in \{0,1\}^n, \\
& \mathbf{w} \in W, \quad \mathbf{w} \in \{0,1\}^M.
\end{aligned}$$

(2) Optimality Cuts:

$$\overline{t}(\mathbf{w}) + \left(u^k\right)^{\mathsf{T}} A\mathbf{x} \leq b^{\mathsf{T}} u^k, \text{ for } k \in \hat{K}.$$

$$\Rightarrow P_k \left(\overline{t}(\mathbf{w}) + \left(u^k\right)^{\mathsf{T}} A\mathbf{x} + \sum_{l=-\underline{m}}^{\overline{l}^K} 2^l s_{kl}^K - b^{\mathsf{T}} u^k \right)^2,$$

$$\text{where } \overline{l}^J = \left\lceil \log_2 \left(b^{\mathsf{T}} u^k - \min_{\mathbf{w},\mathbf{x}} \left(\overline{t}(\mathbf{w}) + \left(u^k\right)^{\mathsf{T}} A\mathbf{x} \right) \right) \right\rceil$$

(3) Feasibility Cuts:

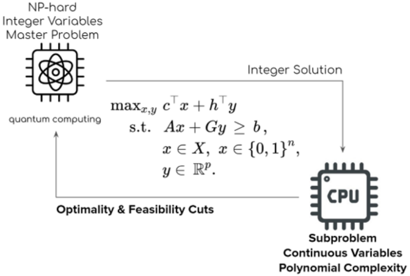
$$\left(r^j\right)^{\mathsf{T}} A\mathbf{x} \leq b^{\mathsf{T}} r^j, \quad \text{for } j \in \hat{J}.$$

$$\Rightarrow P_j \left(\left(r^j\right)^{\mathsf{T}} A\mathbf{x} + \sum_{l=0}^{\overline{l}^J} 2^l s_{kl}^J - b^{\mathsf{T}} r^j \right)^2,$$

$$\text{where } \overline{l}^J = \left\lceil \log_2 \left(b^{\mathsf{T}} r^j - \min_{\mathbf{x}} \left(\left(r^j\right)^{\mathsf{T}} A\mathbf{x} \right) \right) \right\rceil$$

Hybrid Quantum-classical Benders' Decomposition for MILP

Hybrid Quantum-classical Benders' Decomposition for Mixed-integer Linear Programming



$$\mathrm{x}^\dagger = \{\mathrm{w}, \mathrm{x}\},$$

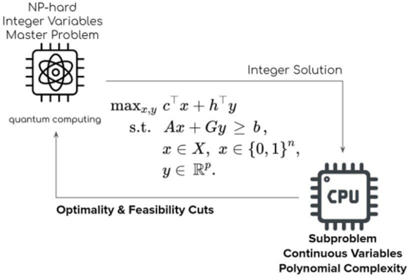
s is the set of slack variables.

$$f(\mathrm{x}') = \mathrm{x}'^\mathsf{T} \mathrm{Q}_{\mathrm{QUBO}} \mathrm{x}',$$

$$\begin{aligned}
\mathrm{Q}_{\mathbf{QUBO}} &= \mathrm{x}^\mathsf{T} \mathrm{diag}(c) \mathrm{x}, \\
&+ \sum_{i=-\underline{m}}^{\overline{m}_+} w_{i+\underline{m}} 2^i w_{i+\underline{m}} - \sum_{j=\mathbf{0}}^{\overline{m}_-} w_{j+(\mathbf{1}+\underline{m}+\overline{m}_+)} 2^j w_{j+(\mathbf{1}+\underline{m}+\overline{m}_+)}, \\
&+ \sum_{k \in K} P_k \left(\overline{t}\,(\mathrm{w}) + \left(u^k\right)^\mathsf{T} A\mathrm{x} + \sum_{l=-\underline{m}}^{\overline{l}^K} 2^l s_{kl}^K - b^\mathsf{T} u^k \right)^{\mathbf{2}}, \\
&+ \sum_{j \in J} P_j \left(\left(r^j\right)^\mathsf{T} A\mathrm{x} + \sum_{l=\mathbf{0}}^{\overline{l}^J} 2^l s_{kl}^J - b^\mathsf{T} r^j \right)^{\mathbf{2}}.
\end{aligned}$$

Hybrid Quantum-classical Benders' Decomposition for MILP

Hybrid Quantum-classical Benders' Decomposition for Mixed-integer Linear Programming



$$\mathrm{x}' = \{\mathrm{w}, \mathrm{x}, \mathrm{s}\},$$

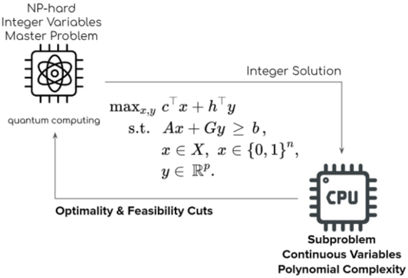
s is the set of slack variables.

$$f(\mathrm{x}') = \mathrm{x}'^{\mathsf{T}} \mathrm{Q}_{\mathrm{QUBO}} \mathrm{x}',$$

$$\begin{aligned}
\mathrm{Q}_{\mathbf{QUBO}} &= \mathrm{x}^{\mathsf{T}} \mathrm{diag}(c)\mathrm{x}, \\
&+ \sum_{i=-\underline{m}}^{\overline{m}_+} w_{i+\underline{m}} 2^i w_{i+\underline{m}} - \sum_{j=0}^{\overline{m}_-} w_{j+(1+\underline{m}+\overline{m}_+)} 2^j w_{j+(1+\underline{m}+\overline{m}_+)}, \\
&+ \sum_{k \in K} P_k \left(\bar{t}\,(\mathrm{w}) + \left(u^k\right)^{\mathsf{T}} A\mathrm{x} + \sum_{l=-\underline{m}}^{\bar{l}^K} 2^l s_{kl}^K - b^{\mathsf{T}} u^k \right)^2, \\
&+ \sum_{j \in J} P_j \left(\left(r^j\right)^{\mathsf{T}} A\mathrm{x} + \sum_{l=0}^{\bar{l}^J} 2^l s_{kl}^J - b^{\mathsf{T}} r^j \right)^2.
\end{aligned}$$

Hybrid Quantum-classical Benders' Decomposition Algorithm

Hybrid Quantum-classical Benders' Decomposition for Mixed-integer Linear Programming



Hybrid Quantum-classical Benders' Decomposition Algorithm [4] [5]

Require: Initial sets $\hat{K}$ of extreme points and $\hat{J}$ of extreme rays of Q

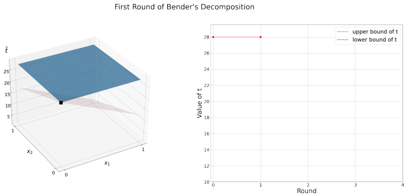
$\bar{t} \leftarrow +\infty$
$\underline{t} \leftarrow -\infty$
while $\mid \bar{t} - \underline{t} \mid \geq \epsilon$ **do**
    P $\leftarrow$ Appropriate penalties numbers or arrays
    Q $\leftarrow$ Reformulate both objective and constraints in the master problem and construct the QUBO formulation by using corresponding rules
    x$'$ $\leftarrow$ Solve the master problem by quantum computers.
    $\bar{t} \leftarrow$ Extract w and replace the $\bar{t}$ with $\bar{t}$(w)
    z_{LP}(x) $\leftarrow$ Solve the sub-problem
    $\underline{t} \leftarrow z_{LP}$(x)
    if $z_{LP}(x) = -\infty$ **then**
        An extreme ray j of Q has been found.
        $\hat{J} = \hat{J} \cup \{j\}$
    else if $z_{LP}(x) < \bar{t}$ **and** $\bar{t} \neq +\infty$ **then**
        An extreme point k of Q has been found.
        $\hat{K} = \hat{K} \cup \{k\}$
return $\bar{t}$, x

Result and Demonstration of HQCBDA

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & -1 \\ -1 & 0 \\ -1 & 0 \\ 0 & -1 \\ 0 & -1 \end{bmatrix}, G = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$b^{\mathsf{T}} = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$h^{\mathsf{T}} = \begin{bmatrix} 8 & 9 & 5 & 6 \end{bmatrix}, \quad c^{\mathsf{T}} = \begin{bmatrix} -15 & -10 \end{bmatrix}.$$



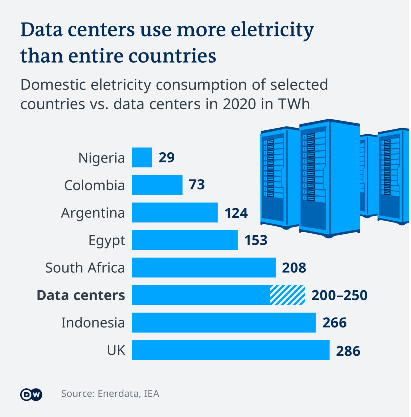
D-Wave hybrid solver: using classical computation to assist quantum annealing.

Scheduling of Datacenter and HVAC Loads with HQCBD

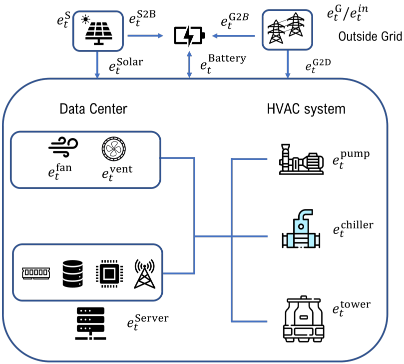
Data center need to manage the power well.

An inner look of a data center

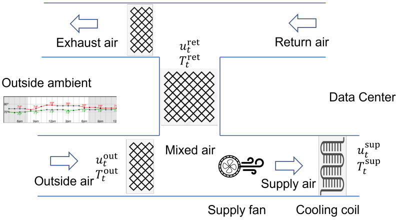
A Google data center in Council Bluff, Iowa



Scheduling of Datacenter and HVAC Loads with HQCBD



A general picture of a data center



The detail of HVAC system in data center

Problem Formulation

- 1/0 (ON/OFF) Binary Decision Variables

u_t^{dis}	Battery discharging state at time t
u_t^{chr}	Battery charging state at time t
$x_{j,t}^{\text{chiller}}$	Chiller j working state at time t
$x_{j,t}^{\text{tower}}$	Cooling tower j working state at time t

- Continuous Decision Variables

p_t^{dis}	Battery discharging power at time t
p_t^{chr}	Battery charging power at time t
$E_t^{\text{B,state}}$	Battery status at time t
$T_{i,t}^{\text{Zone}}$	Temperature in zone i of data center at time t
$T_{i,t}^{\text{sup}}$	AC Temperature in zone i of data center at time t
v_t^{vent}	Ventilation wind speed at time t

Problem Formulation

The list of problem parameters, index, and sets.

Index and Set

$t \in T$	The time range of the problem
$i \in I^{\textbf{Zone}}$	The zones in the data center
$j \in J^{\textbf{Chiller}}$	The available chiller in the data center
$j \in J^{\textbf{tower}}$	The available cooling tower in the data center
$i' \in \mathcal{N}(\cdot)$	Adjacent zones of zone $\cdot$

Binary decision variable set
$\mathrm{x} = \left\{u_t^{\textbf{dis}}, u_t^{\textbf{chr}}, x_{j,t}^{\textbf{chiller}}, x_{j,t}^{\textbf{tower}}\right\}$
Continuous decision variable set
$\mathrm{y} = \left\{p_t^{\textbf{dis}}, p_t^{\textbf{chr}}, E_t^{\textbf{B,state}}, T_{i,t}^{\textbf{Zone}}, T_{i,t}^{\textbf{sup}}, v_t^{\textbf{vent}}\right\}$

Parameters

χ_i	Temperature weight for zone i
$\eta_t^{\textbf{dis}}$	Battery discharging efficiency
$\eta_t^{\textbf{chr}}$	Battery charging efficiency
$\underline{\xi^{\textbf{B}}}$	Battery upper-bound capacity
$\overline{\xi^{\textbf{B}}}$	Battery lower-bound capacity
$\beta^{\textbf{sup}}$	Coefficient for cooling air-flow power rate $v^{\textbf{sup}}$
$\beta_0^{\textbf{vent}}$	1st Coefficient for ventilation power rate $v^{\textbf{vent}}$
$\beta_n^{\textbf{pump}}$	nth Coefficient for pump power rate
$\beta_{n,j}^{\textbf{chiller}}$	nth Coefficient for chiller j power rate
$\beta_{n,j}^{\textbf{tower}}$	nth Coefficient for tower j power rate

Problem Formulation

Parameters

$\theta_{i,t}$	Internal heat generation in zone i
$c^{\textbf{air}}$	The specific heat capacity of water
$c^{\textbf{water}}$	The specific heat capacity of water
$C_i^{\textbf{heat}}$	The heat capacity of room i
$e_t^{\textbf{sup}}$	The eletricity consumed by supply air-flow
$E_t^{\textbf{Server}}$	The eletricity consumed by servers
$E_{init}^{\textbf{B,state}}$	Battery initial power reserve
$E_t^{\textbf{Solar}}$	The eletricity produced by solar system
$\dot{m}_i$	Air mass flow into the zone i
$m_{j,t}^{\textbf{chiller}}$	Mass of water that chiller j can process
$m_{j,t}^{\textbf{tower}}$	Mass of water that tower j can process

$R_{i',i}^{\textbf{Zone}}$	Resistance between i & adjacent node i'
$T^{\textbf{chwr}}$	The return chilled water temperature
$T^{\textbf{chws}}$	The supply chilled water temperature
$T^{\textbf{conwr}}$	The return condense water temperature
$T^{\textbf{conws}}$	The supply condense water temperature
$T_t^{\textbf{out}}$	The outside air temperature at t
$T_{i,init}^{\textbf{Zone}}$	Zone i's initial temperature
$T_i^{\textbf{Zone},+}$	Upper-bound temperature of Zone i
$T_i^{\textbf{Zone},-}$	Lower-bound temperature of Zone i
$T_i^{\textbf{sup},+}$	Maximal AC temperature in Zone i
$T_i^{\textbf{sup},-}$	Minimal AC temperature in Zone i

Problem Formulation

Parameters

$v^{\mathbf{out}}$	The outside air flow rate at t
$v^{\mathbf{return}}$	The return air flow rate at t
$v^{\mathbf{sup}}$	The supply air flow rate at t
$\underline{v}^{\mathbf{vent}}$	The minimal ventilation wind speed

Intermediate Variables

$\Delta E_t^{\mathbf{B}}$	The change of battery power reserve
$e_t^{\mathbf{chiller}}$	Electricity required by chillers
$E_t^{\mathbf{dc,in}}$	Grid electricity required by the data center.
$E_t^{\mathbf{HVAC}}$	Electricity required by the HVAC
$e_t^{\mathbf{pump}}$	Electricity required by pumps
$e_t^{\mathbf{tower}}$	Electricity required by cooling tower
$e_t^{\mathbf{vent}}$	Electricity required by ventilation system
$L_t^{\mathbf{heat}}$	The total thermal load
$m_t^{\mathbf{chw}}$	Chilled water amount required by cooling tower
$m_t^{\mathbf{conw}}$	Condense water amount required by cooling tower

Problem Formulation

MILP Model

$$\min_{\substack{u_t^{\mathbf{dis}},\ u_t^{\mathbf{chr}}, p_t^{\mathbf{dis}},\ p_t^{\mathbf{chr}}, x_{j,t}^{\mathbf{chiller}},\\ x_{j,t}^{\mathbf{tower}}, T_{i,t}^{\mathbf{Zone}}, T_{i,t}^{\mathbf{sup}},\ v_t^{\mathbf{vent}}}} \sum_{t=0}^{T} p_t^{\mathbf{e,g}} e_t^{\mathbf{dc,in}}.$$

The objective function: minimize the total cost of electricity imported from the grid.

$$e_t^{\mathbf{dc,in}} = E_t^{\mathbf{HVAC}} + E_t^{\mathbf{Server}} + \Delta E_t^{\mathbf{B}} - E_t^{\mathbf{Solar}}, \ \forall t.$$

The sum of every energy sources and consumers

$$E_t^{\mathbf{HVAC}} = e_t^{\mathbf{sup}} + e_t^{\mathbf{vent}} + e_t^{\mathbf{chiller}} + e_t^{\mathbf{pump}} + e_t^{\mathbf{tower}}, \ \forall t.$$

The sum of every parts' energy consumption.

$$\Delta E_t^{\mathbf{B}} = p_t^{\mathbf{chr}} \eta^{\mathbf{chr}} - p_t^{\mathbf{dis}} \cdot (\eta^{\mathbf{dis}})^{-1}, \ \forall t.$$

Battery's (dis)charging law

$$E_{t+1}^{\mathbf{B,state}} = E_t^{\mathbf{B,state}} + \Delta E_t^{\mathbf{B}}, \ \forall t.$$

Battery status at time t.

$$\underline{\xi^{\mathbf{B}}} \leq E_{t+1}^{\mathbf{B,state}} \leq \overline{\xi^{\mathbf{B}}}, \ \forall t.$$

Battery status requirements at time t.

$$E_{\mathbf{0}}^{\mathbf{B,state}} = E_{\mathbf{init}}^{\mathbf{B,state}}.$$

Battery initial configuration

Problem Formulation

MILP Model

$$p_t^{\mathbf{chr}} \leq \overline{p_t^{\mathbf{chr}}} \cdot u_t^{\mathbf{chr}}, \ \forall t.$$

The upper bound requirement of battery charging

$$p_t^{\mathbf{dis}} \leq \overline{p_t^{\mathbf{dis}}} \cdot u_t^{\mathbf{dis}}, \ \forall t.$$

The upper bound requirement of battery discharging

$$u_t^{\mathbf{chr}} + u_t^{\mathbf{dis}} \leq 1, \ \forall t.$$

The battery cannot be in charge and discharge mode at the same time interval

$$T_{i,\mathbf{0}}^{\mathbf{Zone}} = T_{i,\mathbf{init}}^{\mathbf{Zone}}.$$

The initial configuration for every zone in data center

$$T_{i,t+\mathbf{1}}^{\mathbf{Zone}} = T_{i,t}^{\mathbf{Zone}} + \sum_{i' \in \mathcal{N}(i)} \left(\frac{T_{i',t}^{\mathbf{Zone}} - T_{i,t}^{\mathbf{Zone}}}{C_i^{\mathbf{heat}} R_{i'i}^{\mathbf{Zone}}} \right) + \frac{\dot{m}_i c_p^{\mathbf{air}} \left(T_{i,t}^{\mathbf{sup}} - T_{i,t}^{\mathbf{Zone}} \right) + \theta_{i,t}}{C_i^{\mathbf{heat}}}, \ \forall i, t.$$

DC RC network temperature linear state space model

$$T_{i,t}^{\mathbf{Zone},-} \leq T_{i,t}^{\mathbf{Zone}} \leq T_{i,t}^{\mathbf{Zone},+}, \ \forall i, t.$$

The upper and lower bound requirement of room temperature.

$$T_{i,t}^{\mathbf{sup},-} \leq T_{i,t}^{\mathbf{sup}} \leq T_{i,t}^{\mathbf{sup},+}, \ \forall i, t.$$

The upper & lower bound of room AC temperature.

Problem Formulation

MILP Model

$$v_t^{\textbf{vent}} + v_t^{\textbf{out}} \geq \underline{v}_t^{\textbf{vent}}, \ \forall t.$$

The minimum ventilation air flow speed

$$v_t^{\textbf{sup}} = v_t^{\textbf{out}} + v_t^{\textbf{return}}, \ \forall t.$$

The air flow speed that comes out of the AC

$$\sum_{j \in \mathbf{J}^{\textbf{chiller}}} x_{j,t}^{\textbf{chiller}} \, m_{j,t}^{\textbf{chiller}} \geq m_t^{\textbf{chw}}, \ \forall t.$$

The min capacity of chiller water that needs to handle.

$$\sum_{j \in \mathbf{J}^{\textbf{tower}}} x_{j,t}^{\textbf{tower}} \, m_{j,t}^{\textbf{tower}} \geq m_t^{\textbf{conw}}, \ \forall t.$$

The min capacity of condense water that needs to handle.

$$L_t^{\textbf{heat}} = \left(T_t^{\textbf{out}} - \sum_{i \in \mathbf{I}^{\textbf{Zone}}} \chi_i T_{i,t}^{\textbf{sup}} \right) \cdot v_t^{\textbf{out}} \, c_p^{\textbf{air}} + \sum_{i \in \mathbf{I}^{\textbf{Zone}}} \chi_i \left(T_{i,t}^{\textbf{Zone}} - T_{i,t}^{\textbf{sup}} \right) \cdot v_t^{\textbf{return}} c_p^{\textbf{air}}, \ \forall t.$$

The sum of heat load in data center

$$m_t^{\textbf{chw}} = \frac{L_t^{\textbf{heat}}}{\left(T_t^{\textbf{chwr}} - T_t^{\textbf{chws}} \right) \cdot c_p^{\textbf{water}}}, \ \forall t.$$

The min amount of chiller water to take away the heat.

$$m_t^{\textbf{conw}} = \frac{L_t^{\textbf{heat}}}{\left(T_t^{\textbf{conwr}} - T_t^{\textbf{conws}} \right) \cdot c_p^{\textbf{water}}}, \ \forall t.$$

The min amount of condense water to take away the heat.

Problem Formulation

MILP Model

$$e_t^{\textbf{chiller}} = \sum_{j \in \mathbf{J}^{\textbf{chiller}}} x_{j,t}^{\textbf{chiller}} \left(\beta_{\mathbf{0},j}^{\textbf{chiller}} + \beta_{\mathbf{1},j}^{\textbf{chiller}} m_{j,t}^{\textbf{chiller}} \right), \ \forall t.$$

The upper bound requirement of battery charging

$$e_t^{\textbf{tower}} = \sum_{j \in \mathbf{J}^{\textbf{tower}}} x_{j,t}^{\textbf{tower}} \left(\beta_{\mathbf{0},j}^{\textbf{tower}} + \beta_{\mathbf{1},j}^{\textbf{tower}} m_{j,t}^{\textbf{tower}} \right), \ \forall t.$$

The upper bound requirement of battery discharging

$$e_t^{\textbf{pump}} = \beta_{\mathbf{0}}^{\textbf{pump}} + \beta_{\mathbf{1}}^{\textbf{pump}} m_t^{\textbf{pump}}, \ \forall t.$$

The battery cannot be in charge and discharge mode at the same time interval

$$e_t^{\textbf{sup}} = \beta^{\textbf{sup}} v_t^{\textbf{sup}}, \ \forall t.$$

The upper bound requirement of battery charging

$$e_t^{\textbf{vent}} = \beta_{\mathbf{0}}^{\textbf{vent}} \left(v_t^{\textbf{vent}} - \underline{v}^{\textbf{vent}} \right), \ \forall t.$$

The upper bound requirement of battery discharging

$$v_t^{\textbf{vent}} \geq \underline{v}^{\textbf{vent}}, \ \forall t.$$

The battery cannot be in charge and discharge mode at the same time interval

Problem Formulation

The original problem.

$$\min_{x,y} \ c^T x + h^T y$$
$$\text{s.t.} \ Ax + Gy \geq b,$$
$$Dx \geq b',$$
$$x \in X, x \in \{0,1\}^n,$$
$$y \in \mathbb{R}^p_+.$$

$\xrightarrow{z_{LP} \text{ Replacement}}$

The master problem.

$$\min_{x} \quad c^T x + z_{LP}(x)$$
$$\text{s.t.} \quad Dx \geq b',$$
$$x \in X, \ x \in \{0,1\}^n.$$

By applying Benders' Decomposition, we yield the sub-problem and its dual-problem.

$$z_{LP}(x) = \min_{y} \quad h^T y$$
$$\text{s.t.} \quad Gy \geq b - Ax,$$
$$y \in \mathbb{R}^p_+.$$

The sub-problem.

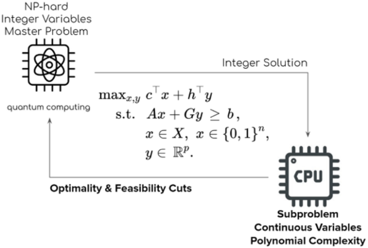
$\xrightarrow{\text{LP Duality}}$

$$z_{LP}(x) = \max_{u} (b - Ax)^T u$$
$$\text{s.t.} \ G^T u \leq h,$$
$$u \in \mathbb{R}^m_+.$$

The dual problem of the sub-problem.

DC Hybrid Quantum-classical Benders' Decomposition Algorithm

Hybrid Quantum-classical Benders' Decomposition for Mixed-integer Linear Programming



DC Hybrid Quantum-classical Benders' Decomposition Algorithm

Require: Initial (Empty) sets of extreme points $\hat{K}$ and rays $\hat{J}$

$\bar{t} \leftarrow +\infty$, $\underline{t} \leftarrow -\infty$

while $\frac{|\bar{t}-\underline{t}|}{|\bar{t}|} \geq \epsilon$ **do**

  P $\leftarrow$ Appropriate penalties numbers or arrays

  Q $\leftarrow$ Reformulate both objective and constraints in master problem and construct the QUBO formulation by using corresponding rules

  $X' = \{x'_1, x'_2, \ldots, x'_N\} \leftarrow$ Solve the master problem by quantum computers and get N feasible solutions.

  $\bar{t} \leftarrow$ Extract w and replace the $\bar{t}$ with $\bar{t}(w)$

  for $x \in X'$ **do**

    $z_{LP}(x) \leftarrow$ Solve the sub-problem

    $\underline{t} \leftarrow z_{LP}(x)$

    if $z_{LP}(x) = -\infty$ **then**

      An extreme ray j of Q has been found.

      $\hat{J} = \hat{J} \cup \{j\}$

    else if $z_{LP}(x) < \bar{t}$ **and** $\bar{t} \neq +\infty$ **then**

      An extreme point k of Q has been found.

      $\hat{K} = \hat{K} \cup \{k\}$

      break

return $\bar{t}$, x

Experiment Set-up

Values for some important parameters in the algorithm.

Symbol	Definition	Value
$\underline{m}$	The bits assigned to decimal part	14
$\overline{m}_+$	The bits assigned to positive integer part	16
$\overline{m}_-$	The bits assigned to negative integer part	0
N	The number of feasible solutions selected from the master problem	6
$\lvert T \rvert$	The length of each time interval (minutes)	10
ϵ	The threshold of gap between $\overline{t}$ and $\underline{t}$	10^{-4}

Experiment Results

Iterations for different case set-up

	Set-up	Binary Variable #	Iterations of CBD	Iteration of HQCBD		
Case 1	$T = 3$, $J^{\text{chiller}} = 1$, $J^{\text{tower}} = 1$	12	84	49	46	47
Case 2	$T = 3$, $J^{\text{chiller}} = 2$, $J^{\text{tower}} = 2$	18	62	36	35	35
Case 3	$T = 3$, $J^{\text{chiller}} = 4$, $J^{\text{tower}} = 5$	33	117	66	74	65
Case 4	$T = 4$, $J^{\text{chiller}} = 2$, $J^{\text{tower}} = 2$	24	217	120	125	127

Experiment Results

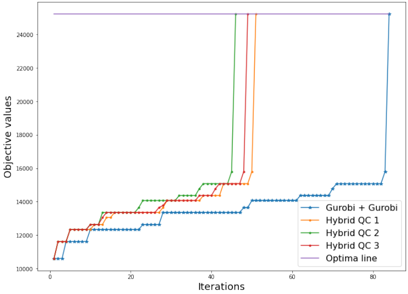
Iterations for different case set-up

	Set-up	Binary Variable #	Iterations of CBD	Aver. iter. of HQCBD	Progress
Case 1	$T = 3$ $J^{\text{chiller}} = 1$ $J^{\text{tower}} = 1$	12	84	48.67	-42.06%
Case 2	$T = 3$ $J^{\text{chiller}} = 2$ $J^{\text{tower}} = 2$	18	62	35.33	-43.01%
Case 3	$T = 3$ $J^{\text{chiller}} = 4$ $J^{\text{tower}} = 5$	33	117	68.33	-41.60%
Case 4	$T = 4$ $J^{\text{chiller}} = 2$ $J^{\text{tower}} = 2$	24	217	127.33	-41.32%

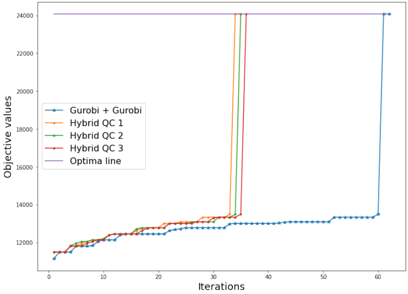
The hybrid quantum-classical Benders' decompsition could save more than 40% iterations than classical Benders decomposition.

Experiment Results

Iteration comparison



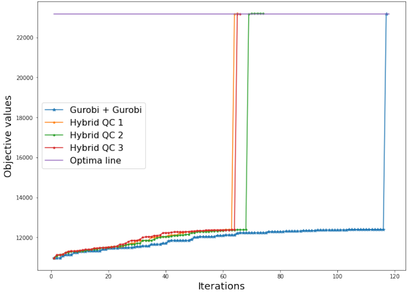
Case 1: $T = 3$, $J^{\text{chiller}} = 1$, $J^{\text{tower}} = 1$



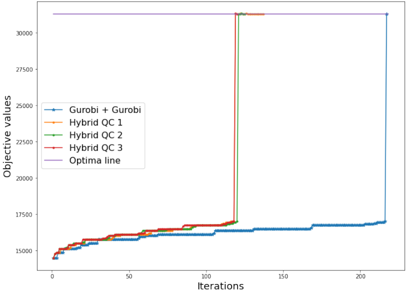
Case 2: $T = 3$, $J^{\text{chiller}} = 2$, $J^{\text{tower}} = 2$

Experiment Results

Iteration comparison



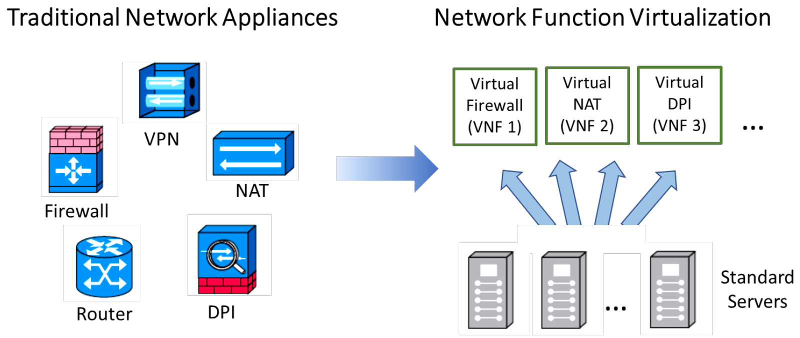
Case 3: $T = 3,\ J^{\textbf{chiller}} = 4,\ J^{\textbf{tower}} = 5$



Case 4: $T = 4,\ J^{\textbf{chiller}} = 2,\ J^{\textbf{tower}} = 2$

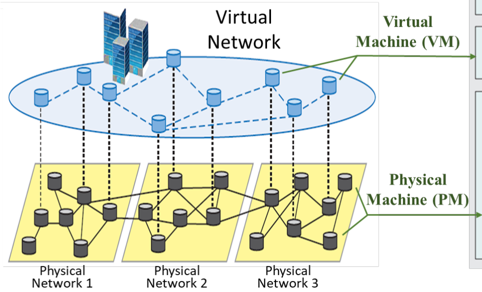
The hybrid quantum-classical Benders' decompsition takes the lead from beginning and wins the comparison safe and sound.

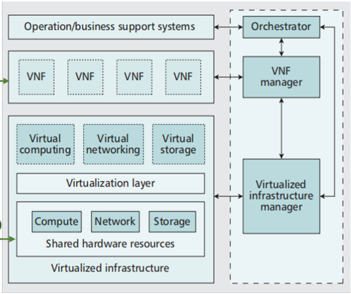
Network Function Virtualization



NFV reduces the difficulty of hardware configuration and improves the flexibility of a network.

Network Function Virtualization

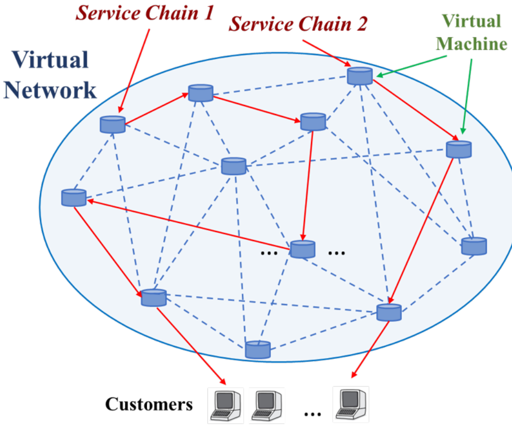




NFV Architecture

The virtual network functions (VNFs) are implemented in virtual machines by software and virtual environment.

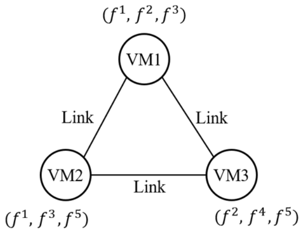
Network Function Virtualization



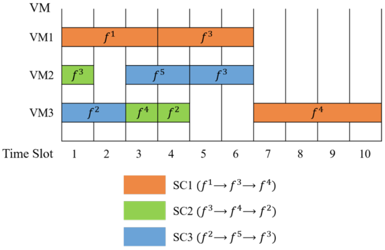
- Vast service chains;
- VNF scheduling problem: how to deploy VMs to process VNFs;
- Delay minimization;

System Model

- All hardware is located in a data center
  — Neglect the transmission delay
- T_{ijm}: the minimum integer that is equal to or larger than $(t_{ijm}/\Delta T)$.



A NFV network



A possible arrangement of service chains

Problem Formulation

- Decision Variables

x_{ijm}	equals to 1, if VM m is used to process f_{ij}^{k}; otherwise, equals to 0
y_{ijmt}	equals to 1, if VM m is used to process f_{ij}^{k} in the time slot t ; otherwise, equals to 0
z_{ijmt}	equals to 1, if VM m starts to process f_{ij}^{k} at the beginning of the time slot t ; otherwise, equals to 0
p_{ijmt}	equals to 1, if VM m finishes processing f_{ij}^{k}at the beginning of the time slot t ; otherwise, equals to 0

- Others

f_{ij}^{k}	the j^{th} function in service i belongs to the k^{th} type of functions
V_{ij}^{k}	the set of VMs which can serve f_{ij}^{k}
T_{ijm}	the number of time slots occupied by processing f_{ij}^{k} on VM m

Problem Formulation

ILP Model

$$\min_{s_{iJ}} \quad \tilde{s} = \sum_{i=1}^{I} s_{iJ}$$

The objective function: minimize the total delay of all service chains in the network.

$$s_{iJ} = \sum_{m=1}^{M} \sum_{t=1}^{T_{max}} p_{iJmt} \cdot (t-1) \cdot \Delta \quad \forall i.$$

Calculate the finish time of any service chain.

$$\sum_{m \in V_{ij}^k} x_{ijm} = 1, \quad \forall i, j.$$

Any function f_{ij}^k can be processed on only one VM.

$$x_{ijm} = \sum_{t=1}^{T_{max}} z_{ijmt}, \quad \forall i, j, m.$$

If and only if f_{ij}^k is allocated to VM m, this VM can start processing f_{ij}^k at some point.

$$\sum_{i=1}^{I} \sum_{j=1}^{J} y_{ijmt} \leq 1, \quad \forall m, t.$$

Each VM can process at most one function in one time slot.

$$y_{ijmt} \leq x_{ijm}, \quad \forall i, j, m, t.$$

The relationship between x_{ijm} and y_{ijmt}.

Problem Formulation

$$\sum_{t=1}^{T_{max}} y_{ijmt} = T_{ijm} \cdot x_{ijm}, \quad \forall i, j; \quad m \in V_{ij}^{k}.$$

Required total time T_{ijm} for processing function f_{ij}^{k} must be satisfied.

$$z_{ijmt} + p_{ijmt} \leq 1, \quad \forall i, j, m, t.$$

p_{ijmt} and z_{ijmt} cannot be equal to 1 at the same time.

$$y_{ijm(t-1)} - y_{ijmt} + z_{ijmt} - p_{ijmt} = 0, \quad \forall i, j, m, t.$$

The logical relationship between y_{ijmt}, z_{ijmt} and p_{ijmt}.

$$\sum_{\alpha=1}^{T_{ijm}} z_{ijm(t-\alpha+1)} \leq y_{ijmt}, \quad \forall i, j, t; \quad m \in V_{ij}^{k}.$$

Once the VM starts processing the function f_{ij}^{k}, the VM must process it for required time.

$$\sum_{m \in V_{ij}^{k}} \sum_{\beta=1}^{T_{max}} p_{ijm(t-\beta+1)} \geq z_{i(j+1)m't},$$

$$\forall i, j, t; \quad m' \in V_{i(j+1)}^{k'}.$$

The next function of the service chain must be processed after the processing of the one before it.

$$x_{ijm} = y_{ijmt} = z_{ijmt} = p_{ijmt} = 0,$$

$$\forall i, j, t; \quad m \notin V_{ij}^{k}.$$

x_{ijm}, y_{ijmt}, z_{ijmt} and p_{ijmt} must be equal to 0 if the VM cannot process the function f_{ij}^{k}.

$$\sum_{m \in V_{ij}^{k}} \sum_{t=1}^{T_{max}} z_{ijmt} = \sum_{m \in V_{ij}^{k}} \sum_{t=1}^{T_{max}} p_{ijmt} = 1, \quad \forall i, j.$$

For any function f_{ij}^{k}, only one z_{ijmt} and one p_{ijmt} can be equal to 1.

Problem Formulation

QUBO (Quadratic Unconstrained Binary Optimization)

$$f(x) = \sum_i Q_{i,i}x_i + \sum_i \sum_{i<j} Q_{i,j}x_i x_j, \quad \text{Q:upper-diagonal matrix.}$$

- No constraint

How to transfer the ILP model to QUBO model? [6]

- Reformulate all constraints into quadratic penalties;
- Add them to the original objective function;

Constraint	Equivalent Penalty
$x_1 + x_2 = 1$	$P(x_1 + x_2 - 1)^2$
$x_1 + x_2 + x_3 \leq 1$	$P(x_1x_2 + x_1x_3 + x_2x_3)$
$x_1 + x_2 \leq x_3$	$P(x_1 + x_2 - x_3 + \sum_l a_l r_l)^2$
$x_1 + x_2 = b$	$P(x_1 + x_2 - b)^2$

- x_1, x_2 and x_3: binary variable
- P: penalty coefficient(large positive constant)
- r_l: slack variable

Algorithm

Proposed Algorithm [7]

Input: parameters, I, J, M; the functions in service chain i, f_{ij}^k; the set of VMs which can process f_{ij}^k, V_{ij}^k; the NFV network;
Output: $\tilde{s}$, x_{ijm}, y_{ijmt}, z_{ijmt}, p_{ijmt};
1: Set the value of T_{max}: run the greedy algorithm to get a feasible T_{max};
2: Set the value of penalty coefficients;
3: Transform all constraints to equivalent penalties;
4: The QUBO model: add all terms in equivalent penalties to the right-hand side of the objective function;
5: Embedding the QUBO model onto the quantum annealing hardware;
6: **return** $\tilde{s}$, x_{ijm}, y_{ijmt}, z_{ijmt}, p_{ijmt};

— Find a reasonable T_{max}
— Reduce the number of variables

- The greedy algorithm: rearrange all VNFs in service chains to a service chain

— Solve the problem with more variables

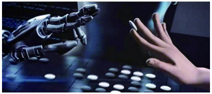
- D-Wave hybrid solver: use classical computation to assist quantum annealing

Study Case Results

Case	Parameters	Matrix Q Size	Average QPU Access Time (s)	Average Solver Run Time (s)	Success Rate
a	$I = 2, J = 2, M = 2$	$(272, 272)$	0.065	2.993	100%
b	$I = 3, J = 2, M = 2$	$(675, 675)$	0.065	2.997	64%
c	$I = 2, J = 3, M = 2$	$(600, 600)$	0.063	2.998	36%
d	$I = 2, J = 2, M = 3$	$(462, 462)$	0.061	2.994	100%
e	$I = 3, J = 3, M = 2$	$(1191, 1191)$	0.064	2.997	58%
f	$I = 3, J = 3, M = 3$	$(1303, 1303)$	0.063	3.630	4%

- Spending a much longer time on finding a feasible solution for *case f* ;
- For *case f* , the success rate is very low (because of too many variables);
- Matrix Q size increases — the difficulty of finding the optimal solution increases.

Quantum Machine Learning



Machine Learning + **Quantum Computing** $\Rightarrow$ **QML**

Classical learning technicues facing complex tasks

Quadradic speed-up

- Quantum Machine Learning
- Machine learning and *fast*
- Not only fast

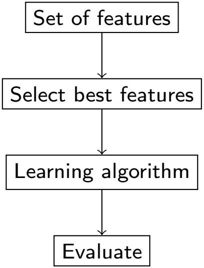
Feature Selection

Need for feature selection

With feature selection, we can optimize our model in some way

- Huge data to train
- Relevant feature
- Irrelevant feature
- Redundant feature

- Prevent overfitting
- Improve accuracy
- Reduce training time



BILP Model

Objective function: maximize relevant feature while minimizing redundant feature [8]

$$\min_{\mathbf{x}} \left\{ \frac{1-\alpha}{k(k-1)/2} \sum_{i} \sum_{j<i} I\left(f_i, f_j\right) x_i x_j - \frac{\alpha}{k} \sum_{i} I\left(f_i, y\right) x_i \right\}, \forall i, j \in S.$$

bias

"influence"

number of feature

independence

x: decision variable

Many Constraints

$$\sum x_i = K_j, \quad \forall i \in S_j \quad j \in \{1, \ldots, z\}.$$

Basic	1	duration
	2	protocol_type
	3	service
	4	Flag
	5	src_bytes
	6	dst_bytes
	7	Land
	8	wrong_fragment
	9	urgent
Content	10	Hot
	11	num_failed_logins
	12	logged_in
	13	num_compromised
	14	root_shell
	15	su_attempted
	16	num_root
	17	num_file_creations
	18	num_shells
	19	num_access_files
	20	num_outbound_cmds
	21	is_host_login
Content	22	is_guest_login
	23	count
	24	srv_count
	25	serror_rate
	26	srv_serror_rate
	27	rerror_rate
	28	srv_rerror_rate
	29	same_srv_rate
	30	diff_srv_rate
	31	srv_diff_host_rate
Traffic	32	dst_host_count
	33	dst_host_srv_count
	34	dst_host_same_srv_rate
	35	dst_host_diff_srv_rate
	36	dst_host_same_src_port_rate
	37	dst_host_srv_diff_host_rate
	38	dst_host_serror_rate
	39	dst_host_srv_serror_rate
	40	dst_host_rerror_rate
	41	dst_host_srv_rerror_rate
	42	Class

Set K is decided by a wrapper method

Constraints

$$\sum_{i \in C_g} x_i \leq 1, \quad \forall C_g \in C.$$

Some are conflict

Feature in C_g may have the same information,and are selected at most once

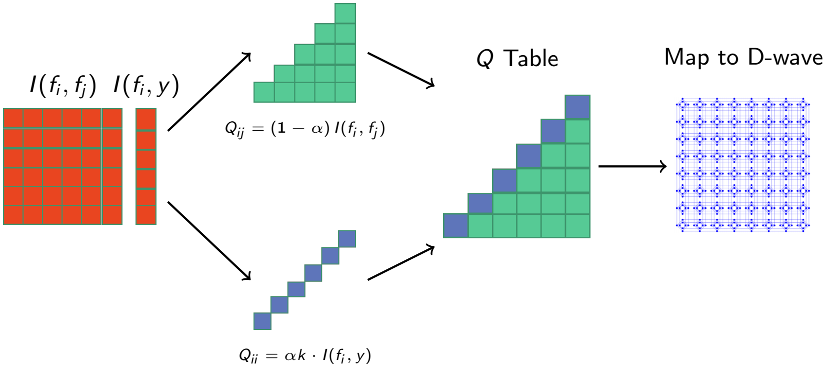
Some rely on others

$$x_i - x_j \leq 0, \quad \forall (i,j) \in \mathbb{E}.$$

$$\sum x_i = T, \quad \forall i \in \mathbb{D}.$$

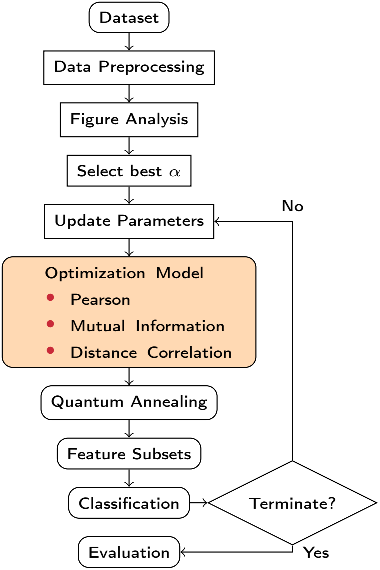
Some are essential and have priority

Reformulate to QUBO



- Similarity between features $\rightarrow Q_{ij}$
- Similarity between features and labels $\rightarrow Q_{ii}$

Algorithm and Experiments



- Pearson

$$p = \frac{\mathbf{Cov}(X, Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$$

  - $y = +1$ indicates that X and Y are totally positively correlated.
  - $y = 0$ implies that X is not correlated to Y at all.

- Mutual Information:

$$I(x; y) = \sum_{i=1}^{n} \sum_{j=1}^{n} p(x_i, y_j) \log \left(\frac{p(x_i, y_j)}{p(x_i)p(y_j)} \right)$$

Mutual information is zero when x and y are statistically independent.

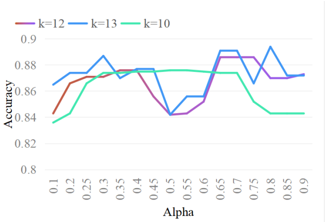
Algorithm and Experiments

In order to search the best α before feature selection we proposed an algorithm.

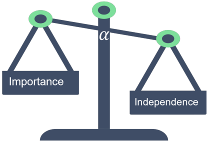
Algorithm for searching α [1]. Given the information matrix I, find the α that maximize the accuracy of classification method $SVC(x)$

Input: I: Common information matrix.
s: step size.
Output: α: weight balance parameter.

```
α ← 0.05;
m_acc ← 0;
acc ← SVC(α, I);
while α < 1 do
    if m_acc < acc then
        m_acc ← acc;
        α ← α;
    α ← α + s;
    acc ← SVC(α, I);
return α.
```



QUBO feature selection with weight balance parameter



α is to balance 2 factors on the two plates

[1] This part could be done by classical solvers or quantum annealers.

Search the Best K

Algorithm: Search the best K

Input: $T^{\mathbf{2}}$, $\mathrm{K}^{\mathbf{3}}$, $\mathrm{I}^{\mathbf{4}}$, $\alpha^{\mathbf{5}}$, $m^{\mathbf{6}}$
Output: The final set K^*

$\mathrm{x}^* \leftarrow QFS(\mathrm{K}, \alpha, \mathrm{I})$;
$Max\_acc \leftarrow SVC(\mathrm{x}^*)$;
$\mathrm{K}^* \leftarrow \mathrm{K}$;
while iteration $< m$ **do**
  if sum(K^*) $\leq T$ **then**
    for $K_i \in \mathrm{K}^*$ **do**
      $K_i^* := K_i + 1$;
      $\mathrm{x}_i^* \leftarrow QFS(K_i^*, \alpha, \mathrm{I})$;
      Update x^* by x_i^*;
      accuracy $a \leftarrow SVC(\mathrm{x}^*)$;
      Insert a to vector A.
    if $Max(\mathrm{A}) > Max\_acc$ **then**
      $Max\_acc \leftarrow Max(\mathrm{A})$;
      Update K^*.
  if sum(K^*) $> T$ **then**
    for $K_i \in \mathrm{K}^*$ **do**
      $K_i^* := K_i - 1$;
      $\mathrm{x}_i^* \leftarrow QFS(K_i^*, \alpha, \mathrm{I})$;
      Update x^* by x_i^* ;
      accuracy $a \leftarrow SVC(\mathrm{x}^*)$;
      Insert a to vector A.
    if $Max(\mathrm{A}) > Max\_acc$ **then**
      $Max\_acc \leftarrow Max(\mathrm{A})$;
      Update K^*.
return K^*.

m: max iteration.

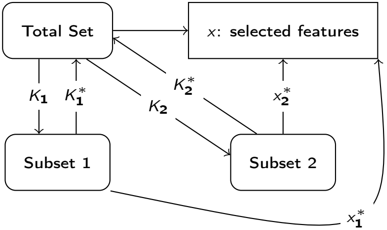
K: Set of initial parameters.

I: Information matrix.

α: weight balance parameter.

T: Number of total selected features.

Sometimes the number of features is too large and D-wave can't handle it. We divide it into small subsets.



After subset 1 calculation, $x = \mathrm{x}_1^* \cap \mathrm{x}_2 \cap \ldots \cap \mathrm{x}_p$

Evaluation Method

	Relevant	Not Relevant
Retrieved	True Positives (TP)	False Positives (FP)
Not Retrieved	True Negatives (TN)	False Negaitives (FP)

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN}.$$

Accuracy is the percentage of accurately predicted samples to all samples that were forecasted

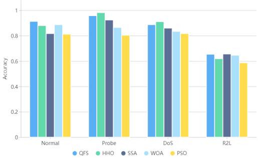
$$precision = \frac{TP}{TP+FP}.$$

Precision is the capability of a classification model to identify only the relevant data points.

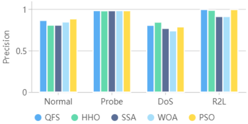
$$recall = \frac{TP}{TP+FN}.$$

Detection estimates the ability of a model to discover all the relevant data points.

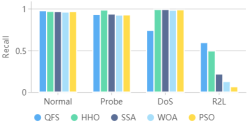
Experiments and Results



Accuracy on different algorithms



Precision on different algorithms



Recall on different algorithms

Control Group:

HHO: Harris hawk optimization (2019)

SSA: Salp swarm algorithm (2017)

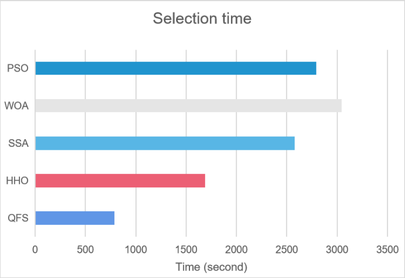
WOA: Whale optimization algorithm (2016)

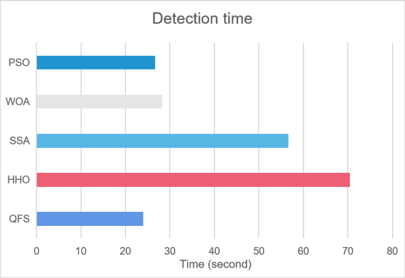
PSO: Particle swarm optimization (2018)

Our result is at least comparable to other algorithms.

The recall is low for R2L attacks because the sample size is small and we did not train a separate model for it.

Experiments and Results





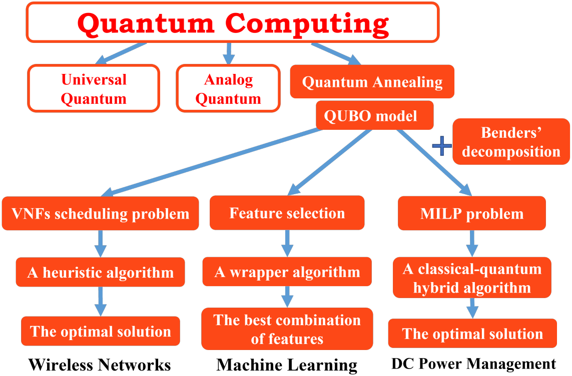
Feature selection and detection time of different algorithms

- Selection time: Time spent removing irrelevant and redundant features.
- Detection time: Time spent training the model with the selected features.

The Summary of the Experiment

- For the evaluation methods accuracy, precision, and call, our QFS's performance is comparable with other classical algorithms
- Compared with other classical algorithms, the QFS algorithm can find the optimal solution at a higher speed while maintaining accuracy. At the same time, the QFS algorithm removes redundant features, which considerably increases the detection speed.

Conclusion



Collaboration

More Applications . . .

References and Bibliography

[1] R. P. Feynman, "Simulating physics with computers," in *Feynman and computation*. CRC Press, 2018, pp. 133–153.

[2] E. K. Grant and T. S. Humble, "Adiabatic quantum computing and quantum annealing," in *Oxford Research Encyclopedia of Physics*, 2020.

[3] W. Scherer, *Mathematics of quantum computing*. Springer, 2019.

[4] Z. Zhao, L. Fan, and Z. Han, "Hybrid quantum benders' decomposition for mixed-integer linear programming," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, may 2022, pp. 2536–2540.

[5] L. Fan and Z. Han, "Hybrid quantum-classical computing for future network optimization," in *IEEE Network*, oct 2022.

[6] F. Glover, G. Kochenberger, R. Hennig, and Y. Du, "Quantum bridge analytics i: a tutorial on formulating and using qubo models," *Annals of Operations Research*, pp. 1–43, 2022.

[7] W. Xuan, Z. Zhao, L. Fan, and Z. Han, "Minimizing delay in network function visualization with quantum computing," in *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, dec 2021, pp. 108–116.

[8] M. Li, H. Zhang, L. Fan, and Z. Han, "A quantum feature selection method for network intrusion detection," in *2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 2022.

References and Bibliography

Thanks!