Cullen College of Engineering
UNIVERSITY OF HOUSTON

Quantum Annealing Methods for Large-Scale Optimization in Communication Networks

Ph.D. Candidate: Wenlu Xuan

Committee Chair: Dr. Lei Fan, and Dr. Zhu Han

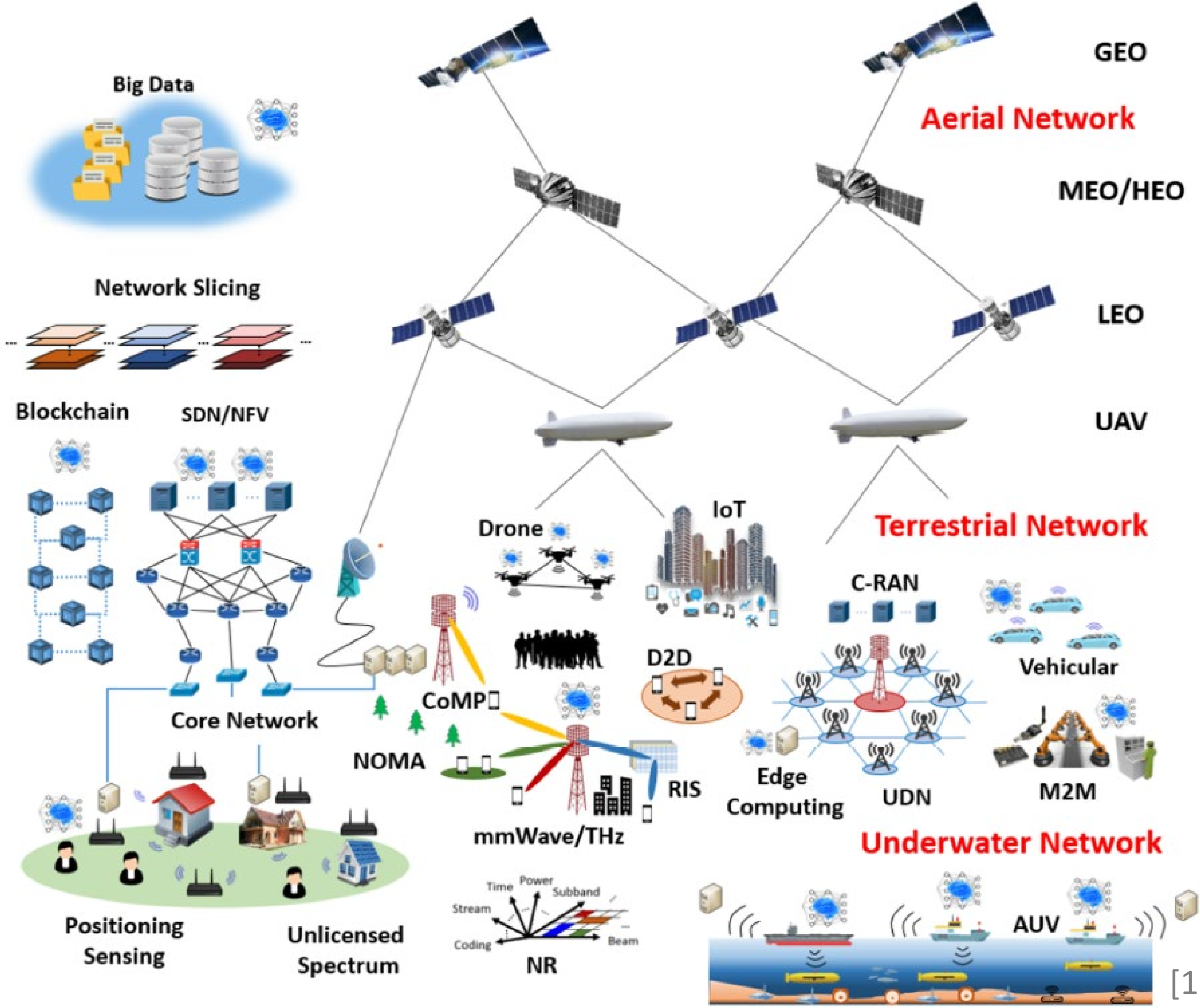
Committee Member: Dr. David Mayerich, Dr. Xiaodi Wu, and Dr. Miao Pan

Department of Electrical and Computer Engineering
University of Houston
Houston, TX

Outline

- **Introduction**
- Work I: Minimizing Delay in Network Function Virtualization with Quantum Computing
- Work II: Lagrangian Relaxation Based Parallelized Quantum Annealing and Its Application in Network Function Virtualization
- Work III: Parallelized Quantum-Inspired Algorithm with Augmented Lagrangian for Data Collection in UAV-enabled IoT Networks
- Future Works and Conclusions

Motivation



[1]

Network Slicing

Network Function Virtualization

MIMO Detection

Resource Allocation

. . .

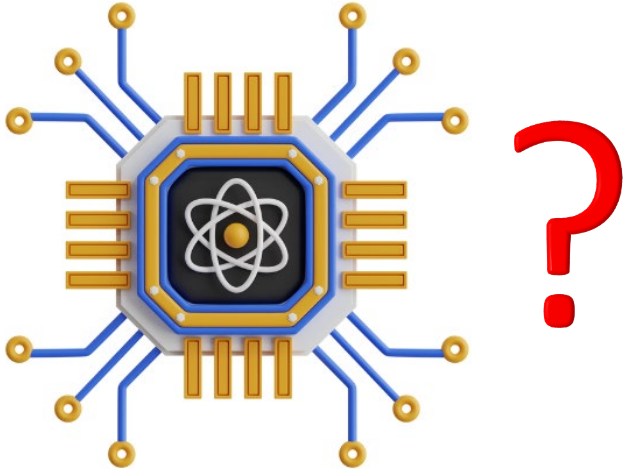
[1] L.H. Shen, K.T. Feng, and L. Hanzo, “Five facets of 6G: Research challenges and opportunities,” *ACM Computing Surveys*, vol. 55, no. 11, pp.1-39, 2023.

Motivation

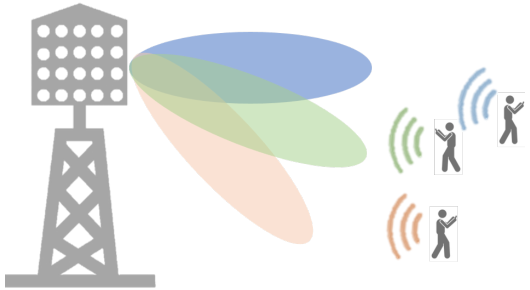
Quantum Computing: use quantum mechanics to solve problems too complex for classical computers.

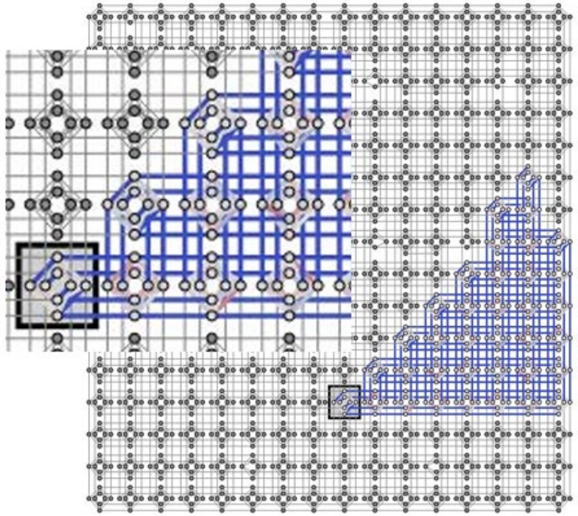
MIMO Detection
Network Slicing
Resource Allocation
. . .

NP-hard Optimization Problems



E. g. Maximum Likelihood (ML) MIMO Detection Using Quantum Computing





[2]

Quantum Computing:

Network Function Virtualization

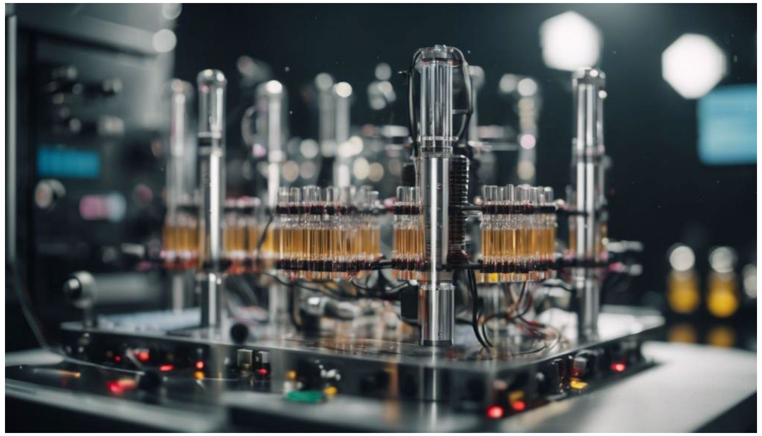
Network Slicing

. . .

[2] M. Kim, D. Venturelli, and K. Jamieson, “Leveraging quantum annealing for large MIMO processing in centralized radio access networks,” in *Proceedings of the ACM Special Interest Group on Data Communication*, Beijing, China, Aug. 2019.

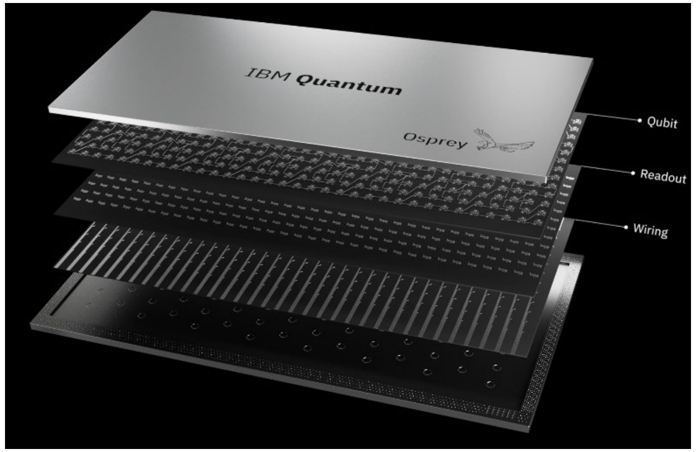
Quantum Computing

Maybe we need to use quantum mechanics in our computers.

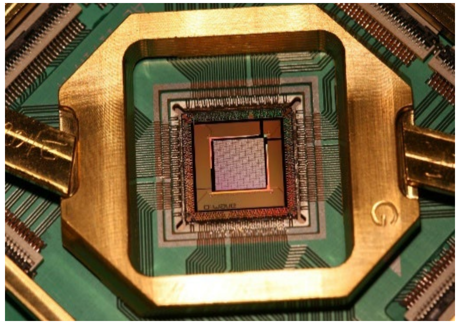


Quantum Computer

- ➢ Gate Model
- ➢ Analog Quantum Model
- ➢ Quantum Annealing



- More Qubits
- Faster than classical solvers
- Fit some optimization problems

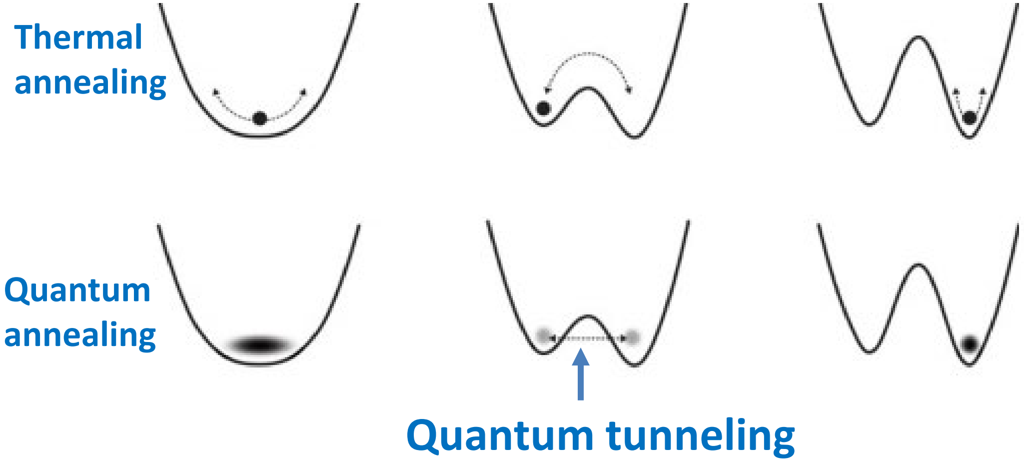


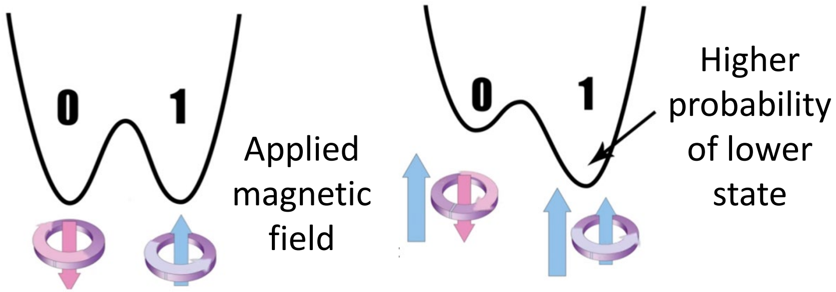
Quantum Annealing

Annealing a Metal
- Heat the metal to a temperature;
- Lower the temperature;

Simulated Annealing
- Heuristic algorithm;
- Random search method;
- **The temperature variable;**

Quantum Annealing





- Apply an external magnetic field: the qubit falls into the 0 or the 1 state;
- At the end of the quantum annealing process, each qubit collapses from a superposition state into either 0 or 1;

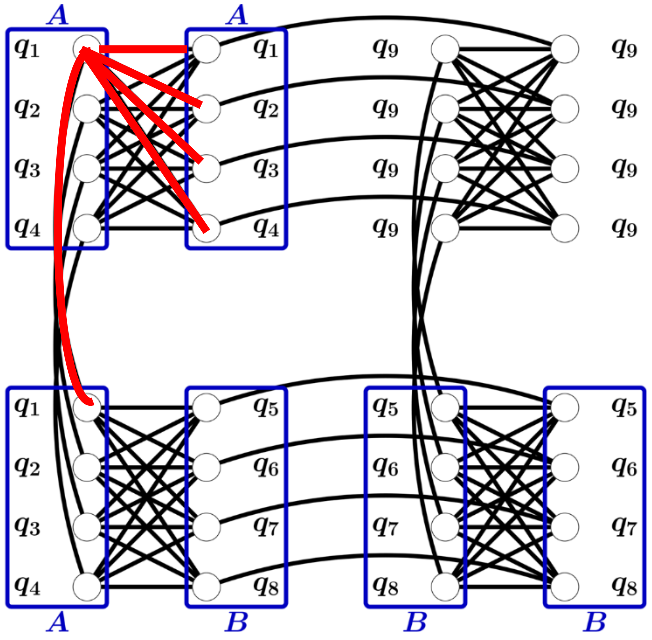
Quantum Annealing

Ising Model

$$H(s) = \sum_i h_i s_i + \sum_i \sum_{i<j} J_{i,j} s_i s_j$$

Spins interact with applied field

Neighboring spins interact with each other



QUBO (Quadratic Unconstrained Binary Optimization)

$$f(x) = \sum_i \boldsymbol{Q}_{i,i} x_i + \sum_i \sum_{i<j} \boldsymbol{Q}_{i,j} x_i x_j$$

$\boldsymbol{Q}$: upper-diagonal matrix

These important optimization problems can be transformed into QUBO model:

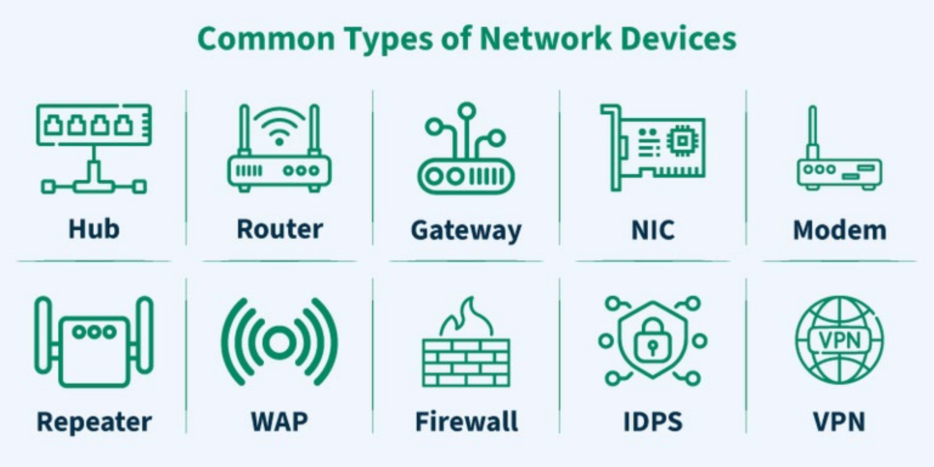
- Knapsack Problems
- Assignment Problems
- Task Allocation Problems
- Capital Budgeting Problems
- … (NP-hard problem)

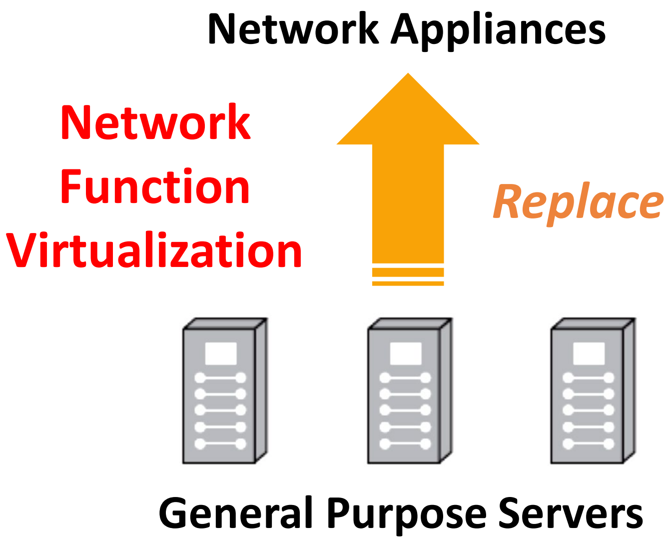
Quantum Computing:
provide an alternative method to solve some NP-hard problems

Outline

- Introduction
- **Work I: Minimizing Delay in Network Function Virtualization with Quantum Computing**
- Work II: Lagrangian Relaxation Based Parallelized Quantum Annealing and Its Application in Network Function Virtualization
- Work III: Parallelized Quantum-Inspired Algorithm with Augmented Lagrangian for Data Collection in UAV-enabled IoT Networks
- Future Works and Conclusions

Background

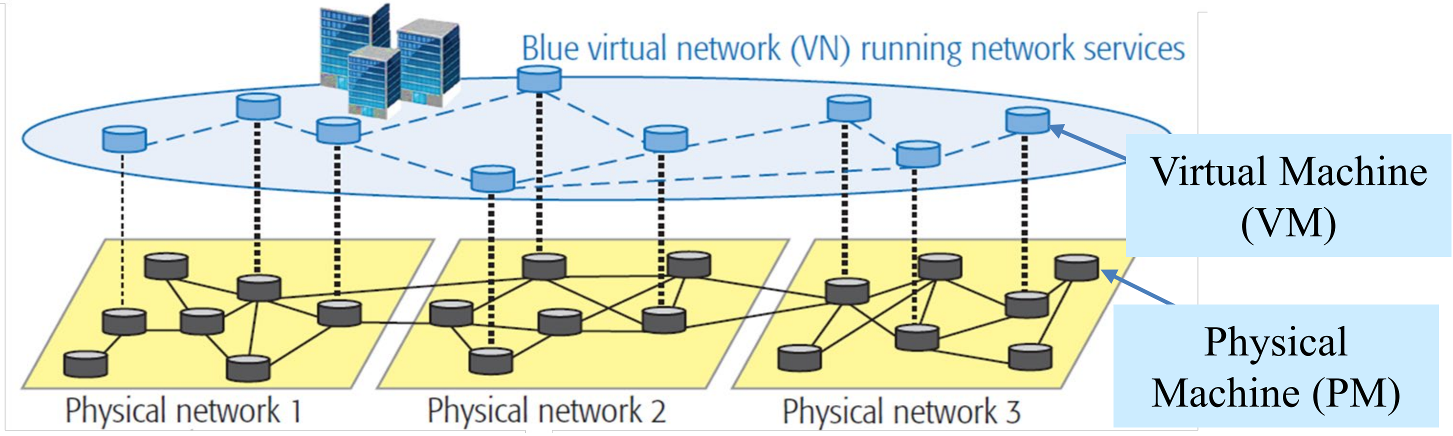


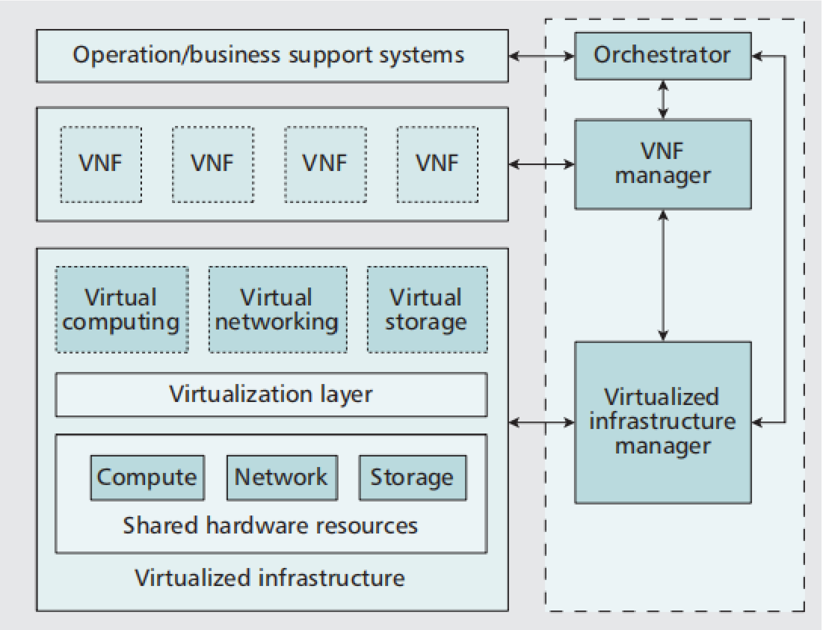

Network Appliances

Network Function Virtualization

Replace

General Purpose Servers





- Vast service chains;
- VNF scheduling problem:

  --- how to deploy VMs to process VNFs;
- Delay minimization;

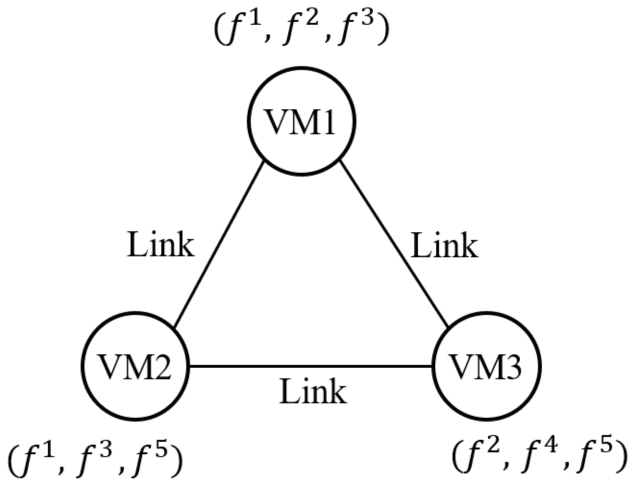
[3]

[3] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, “Network function virtualization: challenges and opportunities for innovations,” IEEE Communications Magazine, vol. 53, no. 2, pp. 90–97, Feb. 2015.

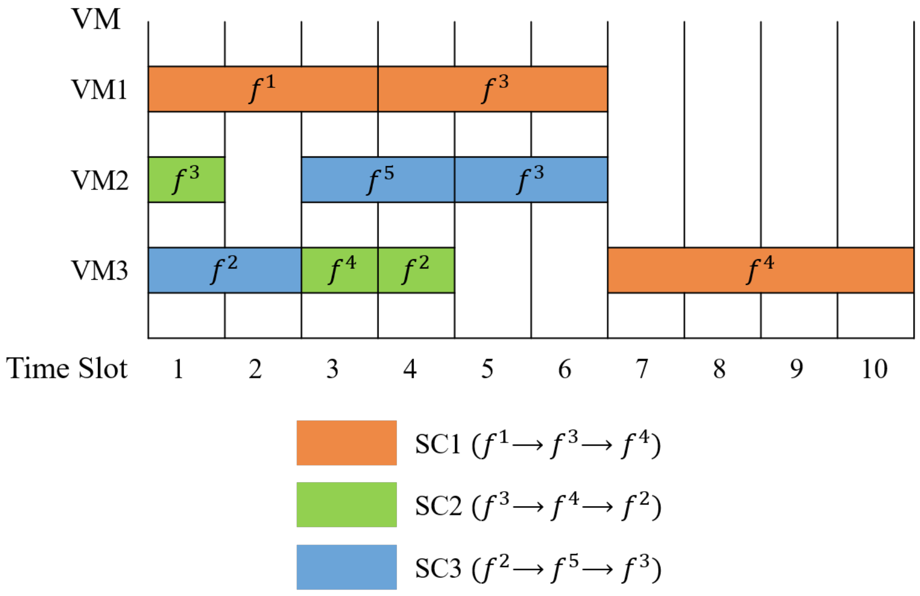
System Model

- All hardware is located in a data center ---- Neglect the transmission delay
- Workload W_{ij} , corresponding to the data package required processing, will be processed on VM m.
- C_m , the computing capability of VM m.
- The processing time t_{ijm} is given by (W_{ij}/C_m) .
- T_{ijm} is the minimum integer that is equal to or larger than $(t_{ijm}/\Delta T)$.

Example



A NFV network



A possible arrangement of service chains

Problem Formulation

ILP Model

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

Objective Function: minimize the total processing delay

$$s_{iJ} = \sum_{m=1}^{M} \sum_{t=1}^{T_{max}} p_{iJmt} \cdot (t-1) \cdot \Delta T, \quad \forall i. \qquad (2)$$

The finish time

$$\sum_{m \in V_{ij}^{k}} x_{ijm} = 1, \quad \forall i, j. \qquad (3)$$

Isolated task conducting

$$x_{ijm} = \sum_{t=1}^{T_{max}} z_{ijmt}, \quad \forall i, j, m. \qquad (4)$$

Logical relationship

$$\sum_{i=1}^{I} \sum_{j=1}^{J} y_{ijmt} \leq 1, \quad \forall m, t. \qquad (5)$$

t: process one function

$$y_{ijmt} \leq x_{ijm}, \quad \forall i, j, m, t. \qquad (6)$$

Logical relationship

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

Problem Formulation

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

(7) Transmission time

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

(8) Task sequence

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

(9) Logical relationship

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

(10) Task sequence

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

(11) Task sequence

$$x_{ijm} = y_{ijmt} = z_{ijmt} = p_{ijmt} = 0, \quad \forall i, j, t; \quad m \notin V_{ij}^{k}.$$

(12) Definition

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

(13) Definition

ILP Model → **QUBO Model** → *Apply quantum annealing*

Problem Formulation

QUBO Model

$$f(x) = \sum_{i} \boldsymbol{Q}_{i,i} x_i + \sum_{i} \sum_{i<j} \boldsymbol{Q}_{i,j} x_i \, x_j$$

$\boldsymbol{Q}$: upper-diagonal matrix

- QUBO: Quadratic Unconstrained Binary Optimization ------ ***No constraint***

How to transfer the ILP model to QUBO model?

- Reformulate all constraints into quadratic penalties: penalize the constraint violation
- Add them to the original objective function;

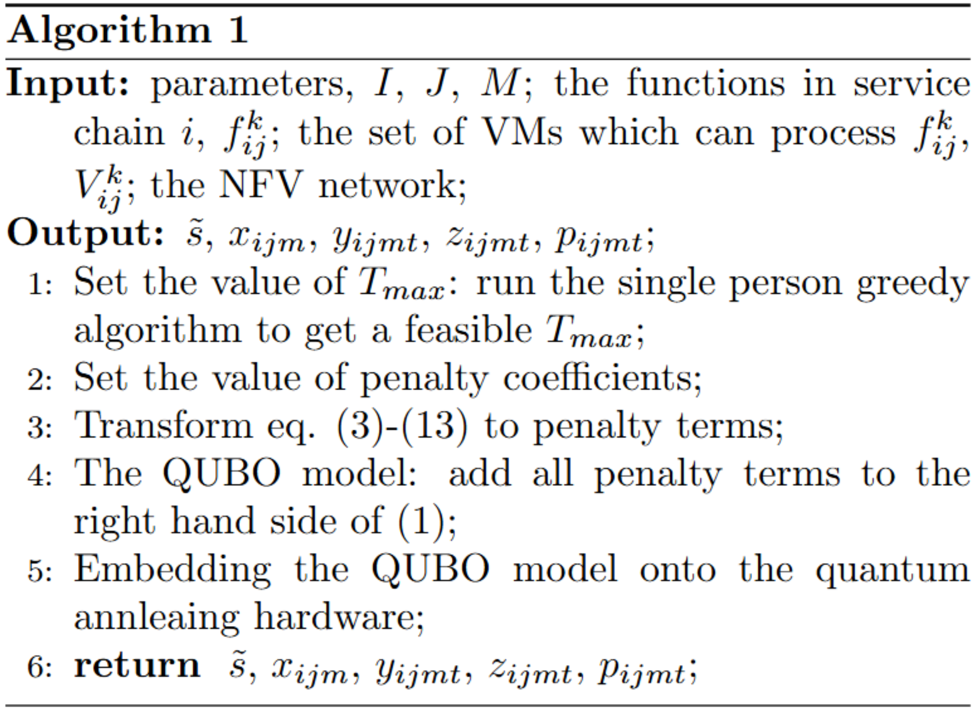
Constraint	**Equivalent Penalty**
$x_1 + x_2 = 1$	$P(x_1 + x_2 - 1)^2$
$x_1 + x_2 + x_3 \leq 1$	$P(x_1 x_2 + x_1 x_3 + x_2 x_3)$
$x_1 + x_2 \leq x_3$	$P(x_1 + x_2 - x_3 + \sum_l a_l r_l)^2$
$x_1 + x_2 = b$	$P(x_1 + x_2 - b)^2$

Penalty coefficient: large positive constant

Binary variable: $x_1 = x_1^2$

Slack variable

Simulation Result



```
Algorithm 1
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: $\tilde{s}$, $x_{ijm}$, $y_{ijmt}$, $z_{ijmt}$, $p_{ijmt}$;
  Set the value of $T_{max}$: run the single person greedy algorithm to get a feasible $T_{max}$;
  Set the value of penalty coefficients;
  Transform eq. (3)-(13) to penalty terms;
  The QUBO model: add all penalty terms to the right hand side of (1);
  Embedding the QUBO model onto the quantum annleaing hardware;
  return $\tilde{s}$, $x_{ijm}$, $y_{ijmt}$, $z_{ijmt}$, $p_{ijmt}$;
```

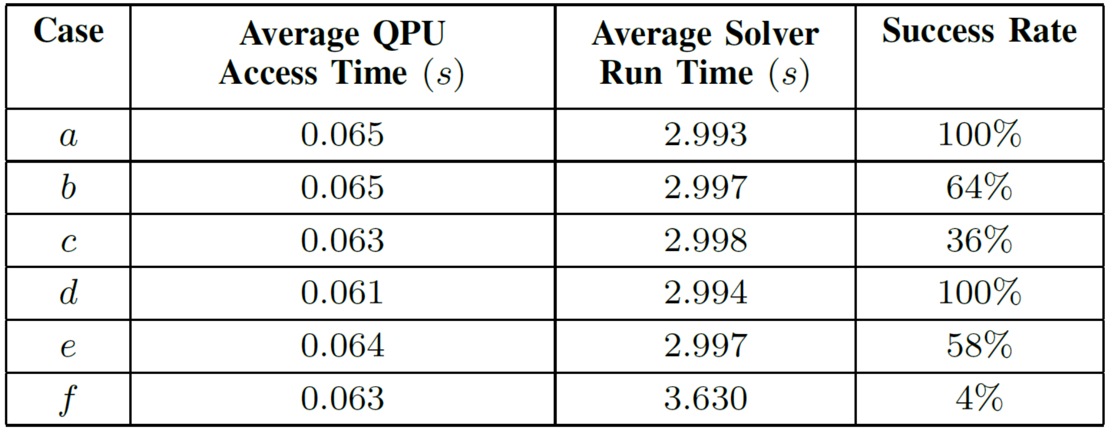
----- *Find a reasonable T_{max}*

----- *Reduce the number of variables*

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

----- *Solve the problem with more variables*

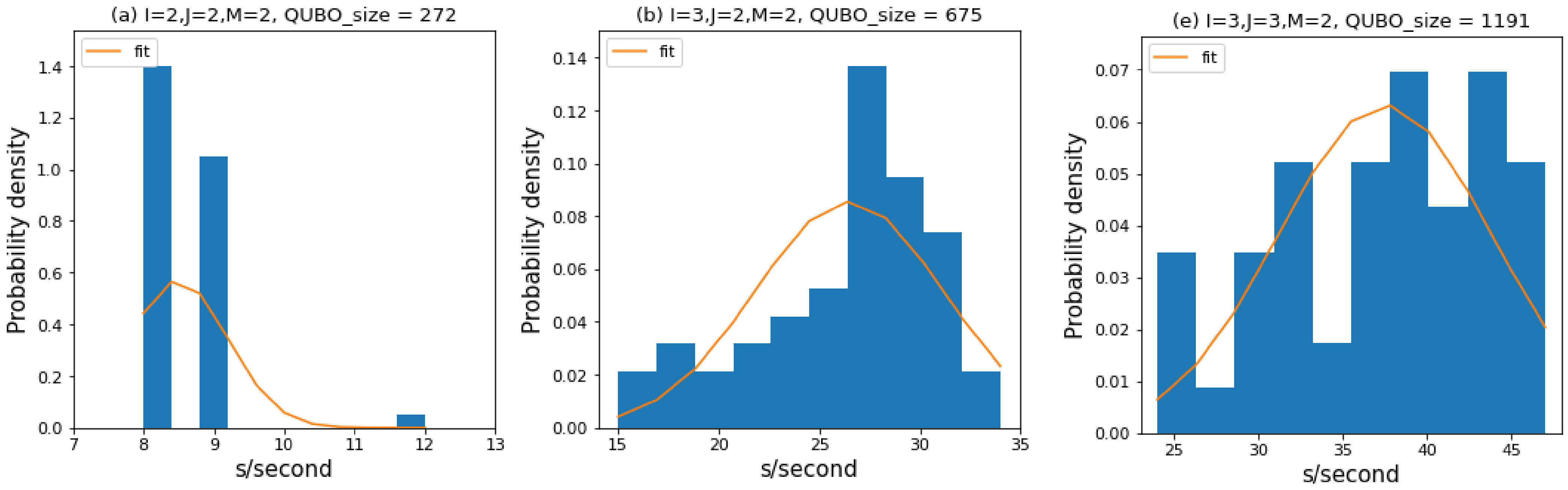
- ➢ D-Wave hybrid solver: use classical computation to assist quantum annealing



Case	**Average QPU Access Time** (s)	**Average Solver Run Time** (s)	**Success Rate**
a	0.065	2.993	100%
b	0.065	2.997	64%
c	0.063	2.998	36%
d	0.061	2.994	100%
e	0.064	2.997	58%
f	0.063	3.630	4%

- Spending a much longer time on finding a feasible solution for *case f* ;
- Matrix *Q* size ↑, the difficulty of finding the optimal solution ↑ .

Simulation Result



➢ Fig. The total processing delays of all service chains

- Size of matrix Q : small ⟶ Distribution: concentrated
- Size of matrix Q : large ⟶ Distribution: dispersed

Summary of Work I:

✓ **solve the NVFs scheduling problem: an ILP model ---- QUBO model;**

✓ **first use quantum computing to solve the optimization problem in NFV;**

Outline

- Introduction
- Work I: Minimizing Delay in Network Function Virtualization with Quantum Computing
- **Work II: Lagrangian Relaxation Based Parallelized Quantum Annealing and Its Application in Network Function Virtualization**
- Work III: Parallelized Quantum-Inspired Algorithm with Augmented Lagrangian for Data Collection in UAV-enabled IoT Networks
- Future Works and Conclusions

Background

Classical Lagrangian Relaxation

Original optimization problem:

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \in X \end{aligned}$$

Lagrangian function:

$$L(\lambda) = \min_{x \in X} \left(c^T x + \lambda^T (Ax - b) \right)$$

Lagrangian Bounding Principle: the value L(λ) is a lower bound on the optimal objective function value of the original problem.

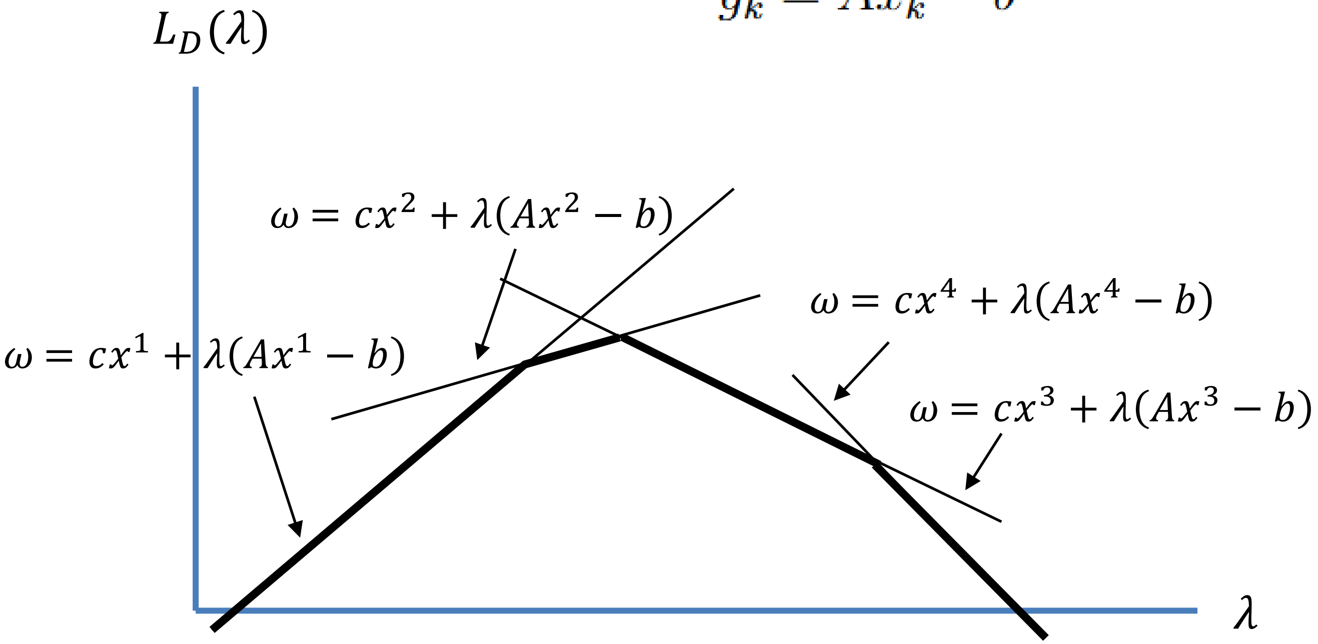
Dual problem:

$$L_D(\lambda) = \max_{\lambda} L(\lambda)$$

$$L_D(\lambda) \longrightarrow \begin{aligned} \max_{x,w} \quad & w \\ \text{s.t.} \quad & w \le c^T x + \lambda^T (Ax - b) \\ & x \in X \\ & \lambda \in \mathbb{R}^n \end{aligned}$$

Subgradient Method:

$$\lambda_{k+1} = \max\{0, \lambda_k + \theta_k g_k\}$$
$$g_k = Ax_k - b$$



Background

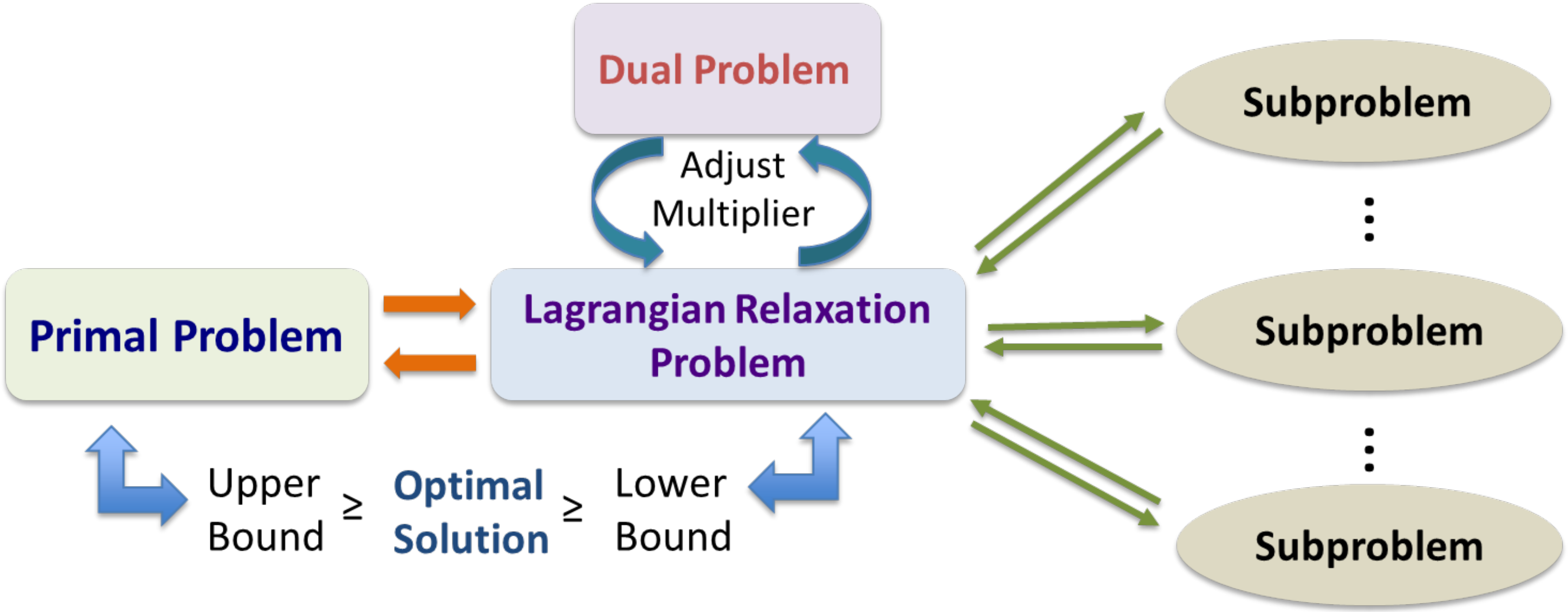
Original model:

binary variable

$$
\begin{aligned}
L_{\mathrm{IP}} = \min_{x_1,\ldots,x_I} \quad & \sum_{i=1}^{I} L_i(x_i) \\
s.t. \quad & a_i x_i \le b_i, \qquad i = 1, \ldots, I, \\
& \sum_{i=1}^{I} c_i x_i \le d,
\end{aligned}
$$

Lagrangian Relaxation

$$
\begin{aligned}
L_{\mathrm{LR}}(\lambda) = \min_{x_1,\ldots,x_I} \quad & \sum_{i=1}^{I} \left(L_i(x_i) + \lambda\Big(\sum_{i=1}^{I} c_i x_i - d\Big) \right) \\
s.t. \quad & a_i x_i \le b_i, \qquad i = 1, \ldots, I.
\end{aligned}
$$



Dual Problem

Adjust Multiplier

Primal Problem

Lagrangian Relaxation Problem

Upper Bound ≥ **Optimal Solution** ≥ Lower Bound

Subproblem

⋮

Subproblem

⋮

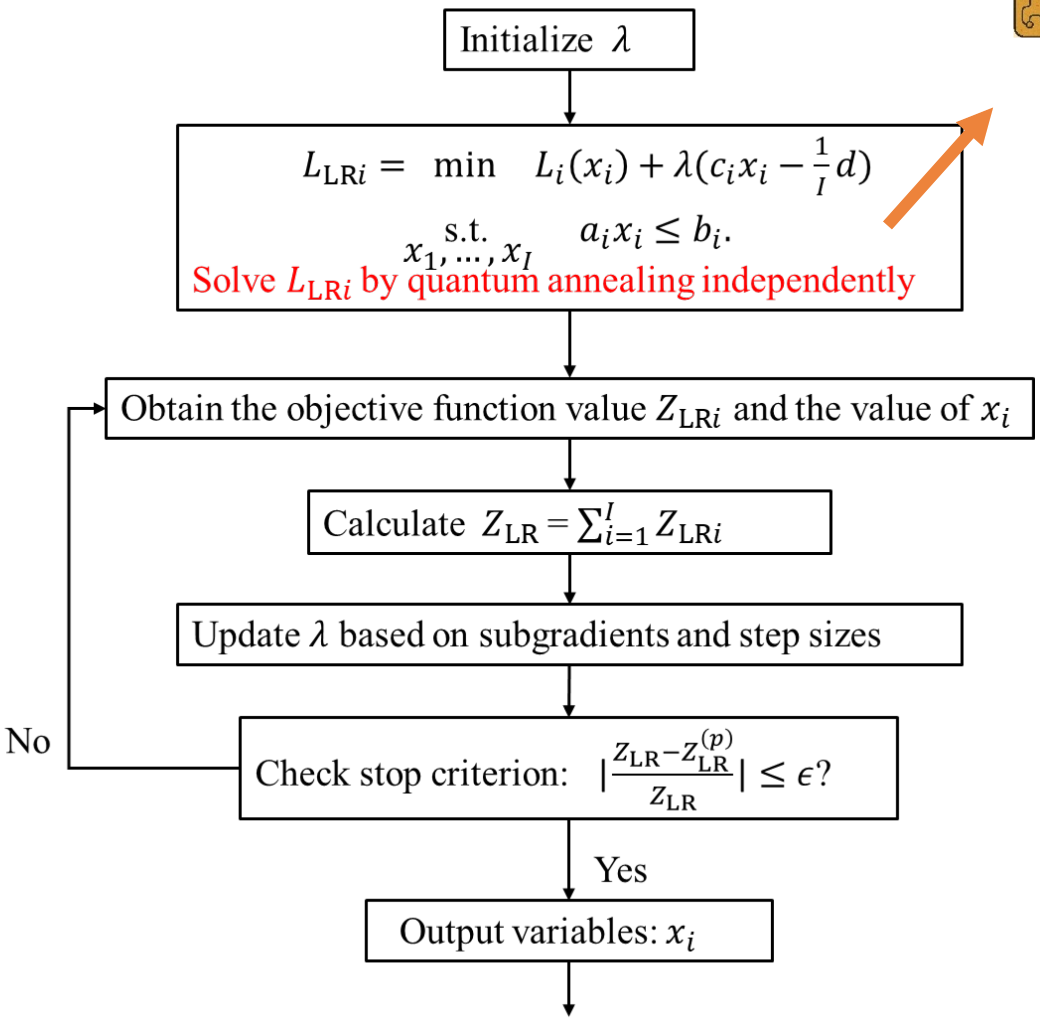
Subproblem

$$
\begin{aligned}
\min \quad & L_1(x_1) + \lambda\Big(c_1 x_1 - \frac{1}{I} d\Big) \\
s.t. \quad & a_1 x_1 \le b_1.
\end{aligned}
$$

$$
\begin{aligned}
\min \quad & L_i(x_i) + \lambda\Big(c_i x_i - \frac{1}{I} d\Big) \\
s.t. \quad & a_i x_i \le b_i.
\end{aligned}
$$

Algorithm

Proposed Lagrangian relaxation based parallelized quantum annealing algorithm



Initialize λ

$$L_{\mathrm{LR}i} = \min_{x_1,\dots,x_I} \quad L_i(x_i) + \lambda(c_i x_i - \tfrac{1}{I}d)$$
$$\text{s.t.} \quad a_i x_i \le b_i.$$

Solve $L_{\mathrm{LR}i}$ by quantum annealing independently

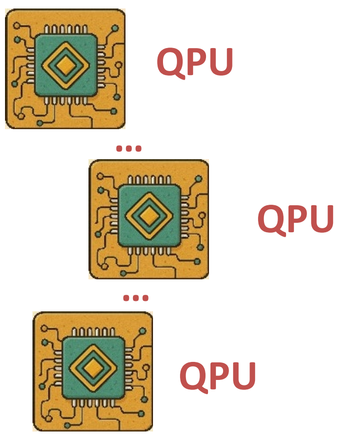
Obtain the objective function value $Z_{\mathrm{LR}i}$ and the value of x_i

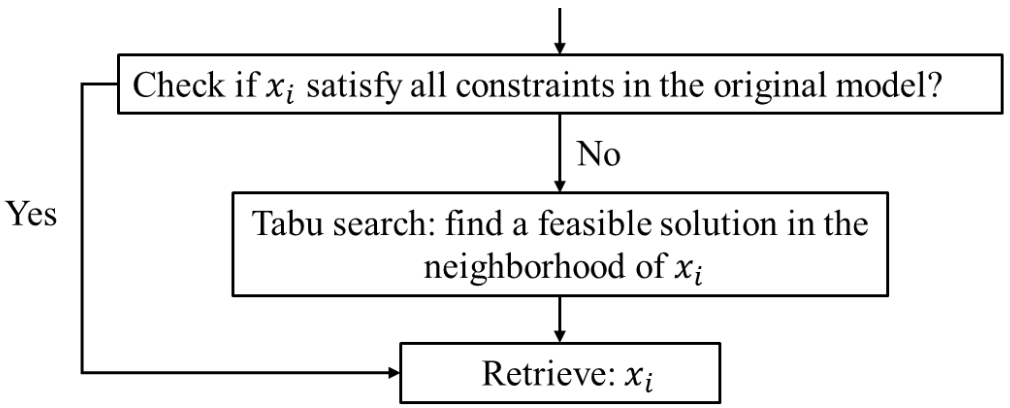
Calculate $Z_{\mathrm{LR}} = \sum_{i=1}^{I} Z_{\mathrm{LR}i}$

Update λ based on subgradients and step sizes

Check stop criterion: $\left|\frac{Z_{\mathrm{LR}} - Z_{\mathrm{LR}}^{(p)}}{Z_{\mathrm{LR}}}\right| \le \epsilon$?

Output variables: x_i





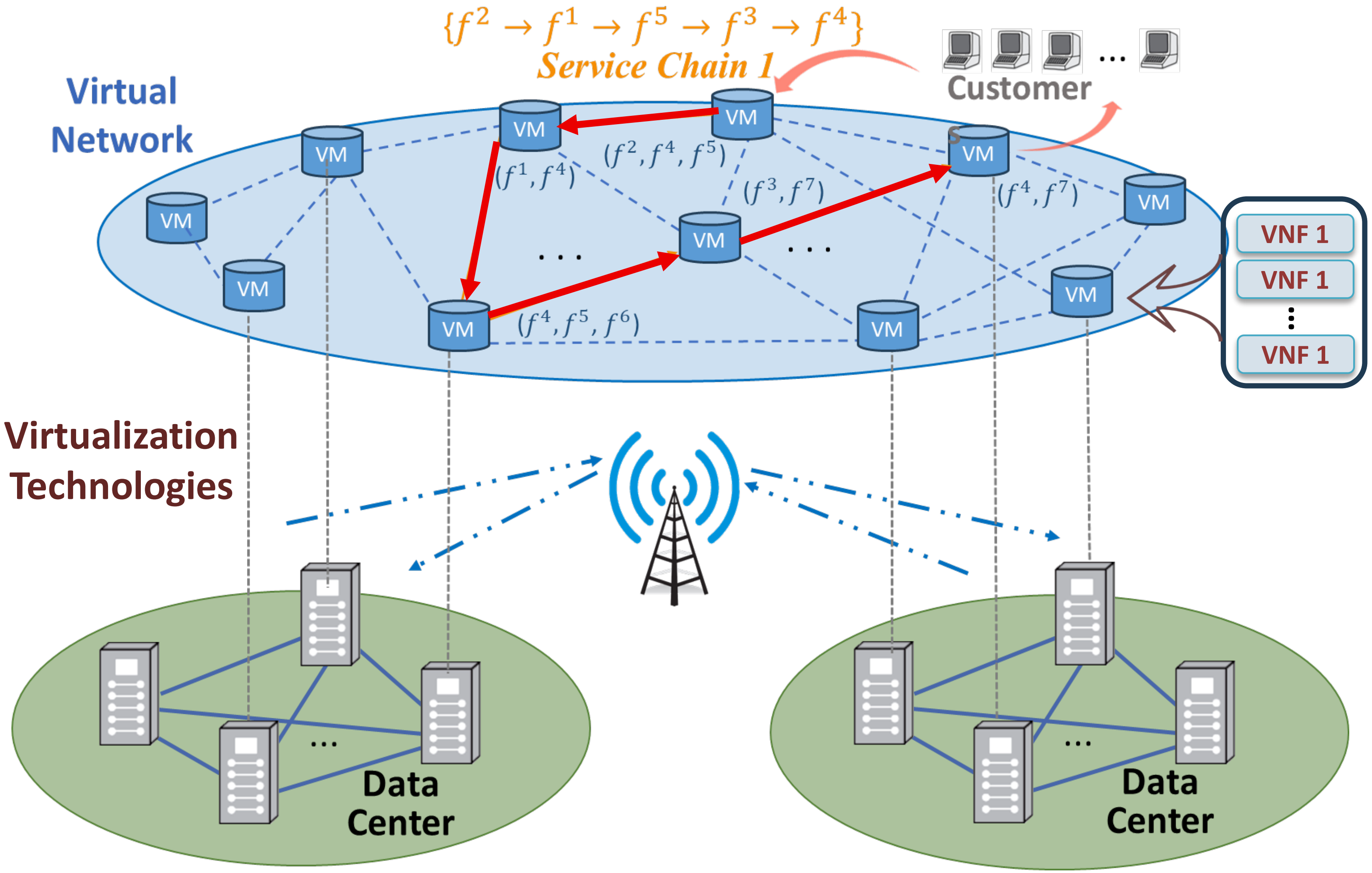
Check if x_i satisfy all constraints in the original model?

Tabu search: find a feasible solution in the neighborhood of x_i

Retrieve: x_i

- ➢ Decompose the integer programming model by Lagrangian relaxations to form several sub-problem models $L_{\mathrm{LR}i}$;
- ➢ Use quantum annealing to solve these sub-problem models $L_{\mathrm{LR}i}$ separately and get the optimal solutions;
- ➢ Update λ : $\boldsymbol{\lambda^{t+1} = \max\{\lambda^t + g^t\gamma^t, 0\}}$
- ➢ Check stop criterion;
- ➢ Tabu Search : converge the solution found by quantum annealing to the global optimum;

System Model



Transmission Delay

+

Processing Delay

- The virtual link between VM m and VM n : $l_{(m,n)}$.
- $T_{ij(m,n)}$ is the number of time intervals occupied by the transmission of f_{ij}^k processing results through the virtual link $l_{(m,n)}$.
- $l_{(m,n)}$ only allows one transmission at the same time.

Problem Formulation

ILP Model

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

Minimize the total delay

Subject to

$$s_{iJ} = \sum_{m=1}^{M} \sum_{t=1}^{T_{max}} p_{iJmt} \cdot (t-1) \cdot \Delta T, \quad \forall i. \qquad (2)$$

The finish time

Constraints (3) – (13)

Processing procedure of f_{ij}^{k}

$$u_{ij(m,n)} \leq x_{ijm}, \quad \forall i, j, m, n. \qquad (14)$$

Logical relationship

$$\sum_{m=1}^{N} u_{ij(m,n)} = \sum_{m'=1}^{N} u_{i(j+1)(n,m')}, \quad \forall i, j; \quad n \in V_{i(j+1)}^{k'}. \qquad (15)$$

Flow in
=
Flow out

$$\sum_{m \in V_{ij}^{k}} \sum_{n \in V_{i(j+1)}^{k'}} u_{ij(m,n)} = 1, \quad \forall i, j. \qquad (16)$$

Occupy one link

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
$u_{ij(m,n)}$	equals to 1, if the virtual link $l_{(m,n)}$ is used to transmit the processing results of f_{ij}^{k}; otherwise, equals to 0
$v_{ij(m,n)t}$	equals to 1, if the virtual link $l_{(m,n)}$ is used to transmit the processing results of f_{ij}^{k} in the time slot t ; otherwise, equals to 0

Problem Formulation

$$(1 - u_{ij(m,n)}) \cdot v_{ij(m,n)t} = 0, \quad \forall i, j, m, n, t. \tag{17}$$

Choose link to do transmission

$$\sum_{i=1}^{I} \sum_{j=1}^{J} v_{ij(m,n)t} + \sum_{i'=1}^{I} \sum_{j'=1}^{J} v_{i'j'(n,m)t} \leq 1. \quad \forall m, n, t. \tag{18}$$

$l_{(m,n)}$ & t : results of one function

$$\sum_{t=1}^{T_{\max}} v_{ij(m,n)t} = T_{ij(m,n)} \cdot u_{ij(m,n)}, \quad \forall i, j; \quad m \in V_{ij}^{k}, \quad n \in V_{i(j+1)}^{k'}. \tag{19}$$

Transmission time

$$\sum_{\alpha=1}^{T_{\max}} p_{ijm(t-\alpha+1)} \geq \sum_{n \in V_{i(j+1)}^{k'}} v_{ij(m,n)t}, \quad \forall i, j, t; \quad m \in V_{ij}^{k}. \tag{20}$$

Task sequence

$$\sum_{\beta=1}^{T_{\max}} z_{i(j+1)n(t+\beta)} \geq \sum_{m \in V_{ij}^{k}} v_{ij(m,n)t}, \quad \forall i, j, t; \quad n \in V_{i(j+1)}^{k'}. \tag{21}$$

Task sequence

$$u_{ij(m,n)} = v_{ij(m,n)t} = 0, \forall i, j, t; \quad m \notin V_{ij}^{k} \quad or \quad n \notin V_{i(j+1)}^{k'}. \tag{22}$$

Definition

ILP Model → **Subproblem Models** → **QUBO Models**

Problem Formulation: Decomposition

Original object function

$$\min_{\mathbf{p}} \sum_{i=1}^{I}\sum_{m=1}^{M}\sum_{t=1}^{T_{\max}} p_{iJmt}\cdot(t-1)\cdot\Delta T.$$

Original model Decompose **Subproblem models**

$$u_{ij(m,n)} \le x_{ijm}, \quad \forall i,j,m,n.$$

$$\sum_{\alpha=1}^{T_{\max}} p_{ijm(t-\alpha+1)} \ge \sum_{n\in V_{i(j+1)}^{k'}} v_{ij(m,n)t}, \quad \forall i,j,t; \quad m\in V_{ij}^{k}.$$

$$\sum_{\beta=1}^{T_{\max}} z_{i(j+1)n(t+\beta)} \ge \sum_{m\in V_{ij}^{k}} v_{ij(m,n)t}, \quad \forall i,j,t; \quad n\in V_{i(j+1)}^{k'}.$$

Lagrangian

Relaxation

$$\begin{aligned} Z_{\mathrm{IP}} = \min_{\mathbf{x},\mathbf{p},\mathbf{z},\mathbf{u},\mathbf{v}} & \sum_{i=1}^{I}\sum_{m=1}^{M}\sum_{t=1}^{T_{\max}} p_{iJmt}\,(t-1)\cdot\Delta T \\ & + \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{m\in V_{ij}^{k}}\sum_{t=1}^{T_{\max}} \lambda_{ijmt}\Big(\sum_{n\in V_{i(j+1)}^{k'}} v_{ij(m,n)t} - \sum_{\alpha=1}^{T_{\max}} p_{ijm(t-\alpha+1)}\Big) \\ & + \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{n\in V_{i(j+1)}^{k'}}\sum_{t=1}^{T_{\max}} \lambda_{ijnt}\Big(\sum_{m\in V_{ij}^{k}} v_{ij(m,n)t} - \sum_{\beta=1}^{T_{\max}} z_{i(j+1)n(t+\beta)}\Big) \\ & + \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{m=1}^{N}\sum_{n=1}^{N} \lambda_{ijmn}\big(u_{ij(m,n)} - x_{ijm}\big), \end{aligned}$$

with

Lagrangian Multiplier

$$\lambda_{ijmn} \ge 0, \quad \forall i,j,m,n,$$

$$\lambda_{ijmt} \ge 0, \quad \forall i,j,t; \quad m\in V_{ij}^{k},$$

$$\lambda_{ijnt} \ge 0, \quad \forall i,j,t; \quad n\in V_{i(j+1)}^{k'}.$$

Decompose it to 2 subproblems

Problem Formulation: Decomposition

Subproblem 1 model

$$Z_{\mathrm{IP1}} = \min_{\mathbf{x},\mathbf{p},\mathbf{z}} \sum_{i=1}^{I}\sum_{t=1}^{T_{\max}}\Big(\sum_{m=1}^{M} p_{iJmt}\cdot(t-1)\cdot\Delta T - \sum_{j=1}^{J}\sum_{m\in V_{ij}^{k}} \lambda_{ijmt}\sum_{\alpha=1}^{T_{\max}} p_{ijm(t-\alpha+1)}$$

$$-\sum_{j=1}^{J}\sum_{n\in V_{i(j+1)}^{k'}} \lambda_{ijnt}\sum_{\beta=1}^{T_{\max}} z_{i(j+1)n(t+\beta)}\Big) - \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{m=1}^{N}\sum_{n=1}^{N} \lambda_{ijmn}x_{ijm}.$$

with constraint (3)-(13). → **QUBO model**

$$P_{1ij}\Big(\sum_{m\in V_{ij}^{k}} x_{ijm} - 1\Big)^2, \quad \forall i,j,$$

$$P_{1ijm}\Big(\sum_{t=1}^{T_{\max}} z_{ijmt} - x_{ijm}\Big)^2, \quad \forall i,j; \quad m\in V_{ij}^{k}.$$

…

Subproblem 2 model

$$Z_{\mathrm{IP2}} = \min_{\mathbf{u},\mathbf{v}} \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T_{\max}}\Big(\sum_{m\in V_{ij}^{k}} \lambda_{ijmt}\sum_{n\in V_{i(j+1)}^{k'}} v_{ij(m,n)t} + \sum_{n\in V_{i(j+1)}^{k'}} \lambda_{ijnt}\sum_{m\in V_{ij}^{k}} v_{ij(m,n)t}\Big)$$

$$+\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{m=1}^{N}\sum_{n=1}^{N} \lambda_{ijmn}u_{ij(m,n)}.$$

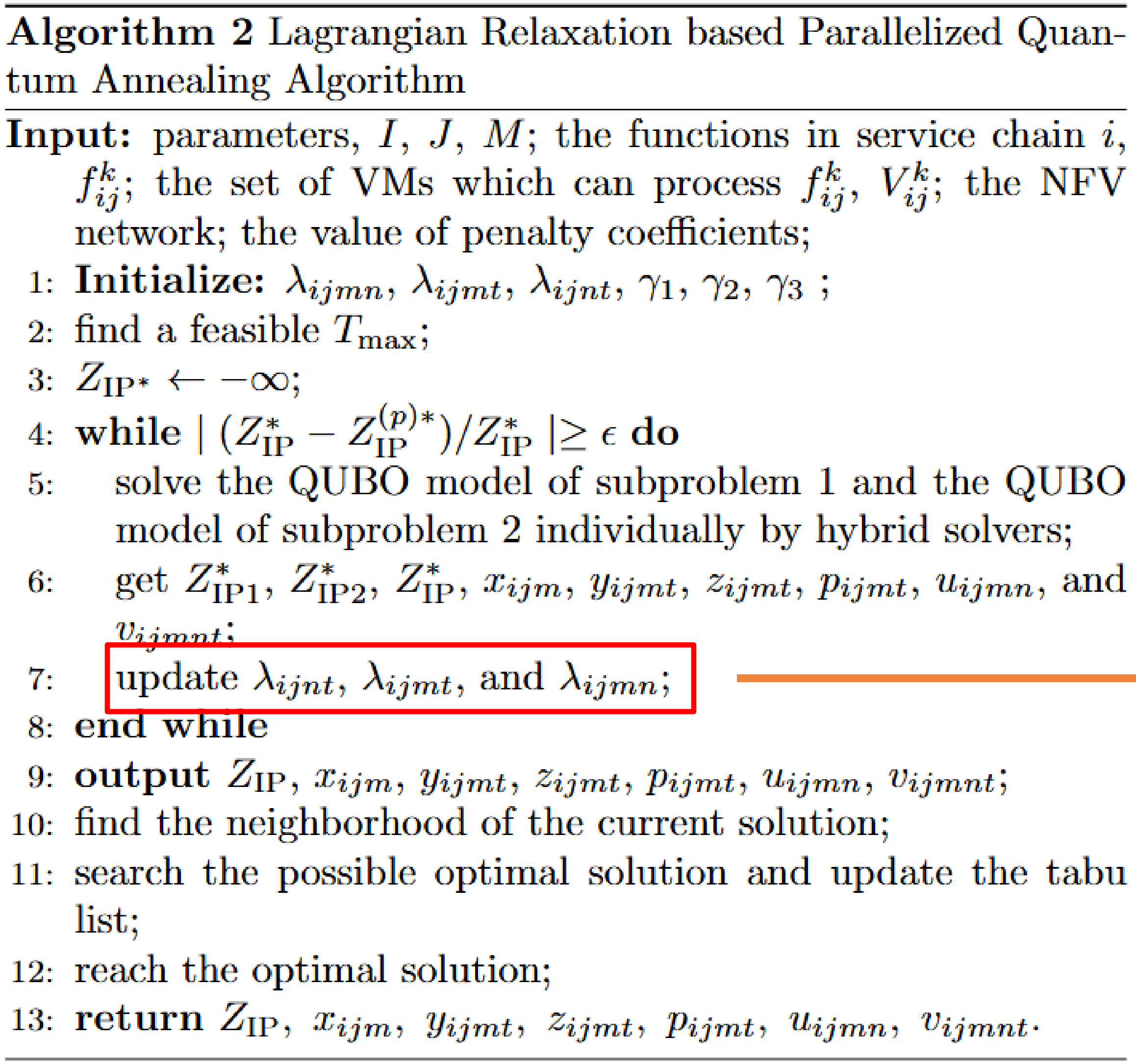
with constraint (15)-(19), (22). → **QUBO model**

$$P_{1ijn}\Big(\sum_{m=1}^{N} u_{ij(m,n)} - \sum_{m'=1}^{N} u_{i(j+1)(n,m')}\Big)^2, \quad \forall i,j; \quad n\in V_{i(j+1)}^{k'}.$$

$$P_{2ij}\Big(\sum_{m\in V_{ij}^{k}}\sum_{n\in V_{i(j+1)}^{k'}} u_{ij(m,n)} - 1\Big)^2, \quad \forall i,j,$$

…

Algorithm



Algorithm 2 Lagrangian Relaxation based Parallelized Quantum Annealing Algorithm

```
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; the value of penalty coefficients;
Initialize: $\lambda_{ijmn}$, $\lambda_{ijmt}$, $\lambda_{ijnt}$, $\gamma_1$, $\gamma_2$, $\gamma_3$ ;
find a feasible $T_{\max}$;
$Z_{\mathrm{IP}^*} \leftarrow -\infty$;
while $| (Z_{\mathrm{IP}}^* - Z_{\mathrm{IP}}^{(p)*})/Z_{\mathrm{IP}}^* | \geq \epsilon$ do
    solve the QUBO model of subproblem 1 and the QUBO
    model of subproblem 2 individually by hybrid solvers;
    get $Z_{\mathrm{IP}1}^*$, $Z_{\mathrm{IP}2}^*$, $Z_{\mathrm{IP}}^*$, $x_{ijm}$, $y_{ijmt}$, $z_{ijmt}$, $p_{ijmt}$, $u_{ijmn}$, and
    $v_{ijmnt}$;
    update $\lambda_{ijnt}$, $\lambda_{ijmt}$, and $\lambda_{ijmn}$;
end while
output $Z_{\mathrm{IP}}$, $x_{ijm}$, $y_{ijmt}$, $z_{ijmt}$, $p_{ijmt}$, $u_{ijmn}$, $v_{ijmnt}$;
find the neighborhood of the current solution;
search the possible optimal solution and update the tabu
    list;
reach the optimal solution;
return $Z_{\mathrm{IP}}$, $x_{ijm}$, $y_{ijmt}$, $z_{ijmt}$, $p_{ijmt}$, $u_{ijmn}$, $v_{ijmnt}$.
```

Subgradient Method

$$g_1^{(k)} = u_{ij(m,n)}^{(k)} - x_{ijm}^{(k)},$$

$$\lambda_{ijmn}^{(k+1)} = \max\left(0, \lambda_{ijmn}^{(k)} + g_1^{(k)}\gamma_1^{(k)}\right).$$

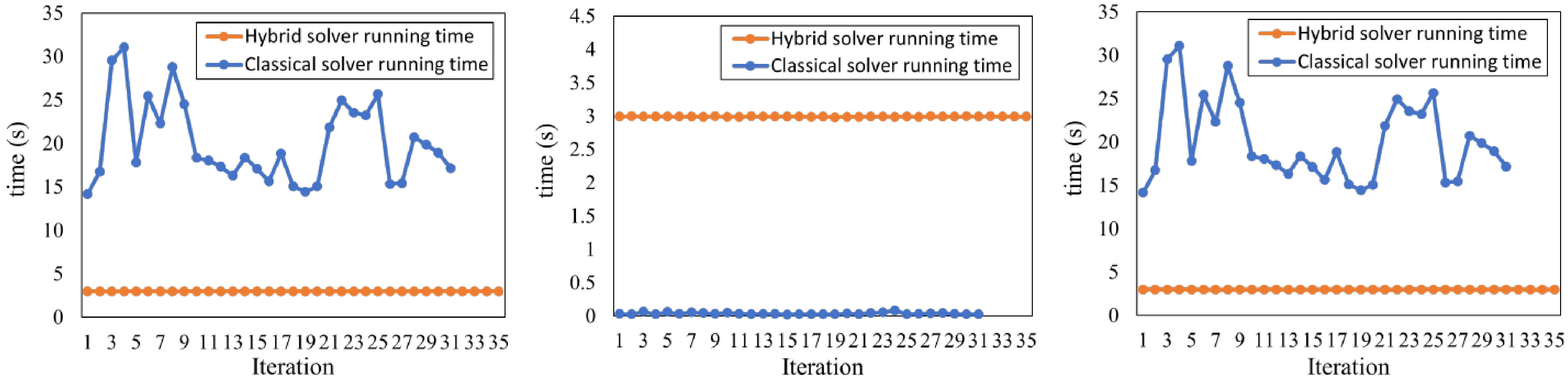
$$g_2^{(k)} = \sum_{n \in V_{i(j+1)}^{k'}} v_{ij(m,n)t}^{(k)} - \sum_{\alpha=1}^{T_{\max}} p_{ijm(t-\alpha+1)}^{(k)},$$

$$\lambda_{ijmt}^{(k+1)} = \max\left(0, \lambda_{ijmt}^{(k)} + g_2^{(k)}\gamma_2^{(k)}\right),$$

$$g_3^{(k)} = \sum_{m \in V_{ij}^{k}} v_{ij(m,n)t}^{(k)} - \sum_{\beta=1}^{T_{\max}} z_{i(j+1)n(t+\beta)}^{(k)},$$

$$\lambda_{ijnt}^{(k+1)} = \max\left(0, \lambda_{ijnt}^{(k)} + g_3^{(k)}\gamma_3^{(k)}\right).$$

Simulation Result

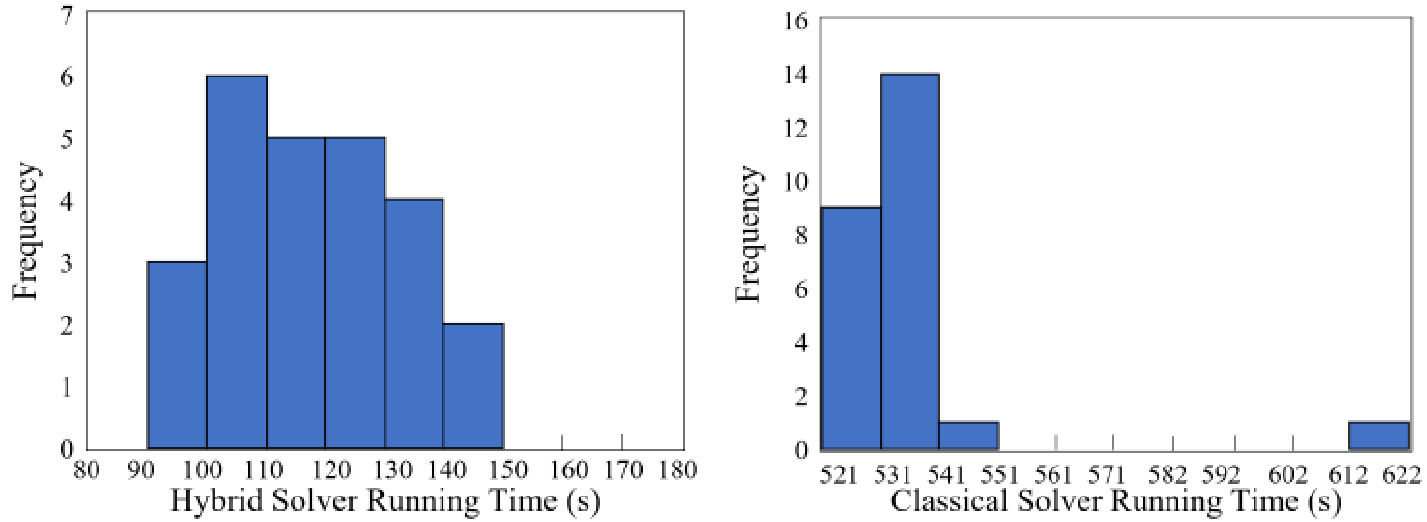


(a) Case a: sub-problem 1 (b) Case a: sub-problem 2 (c) Case a: every iteration

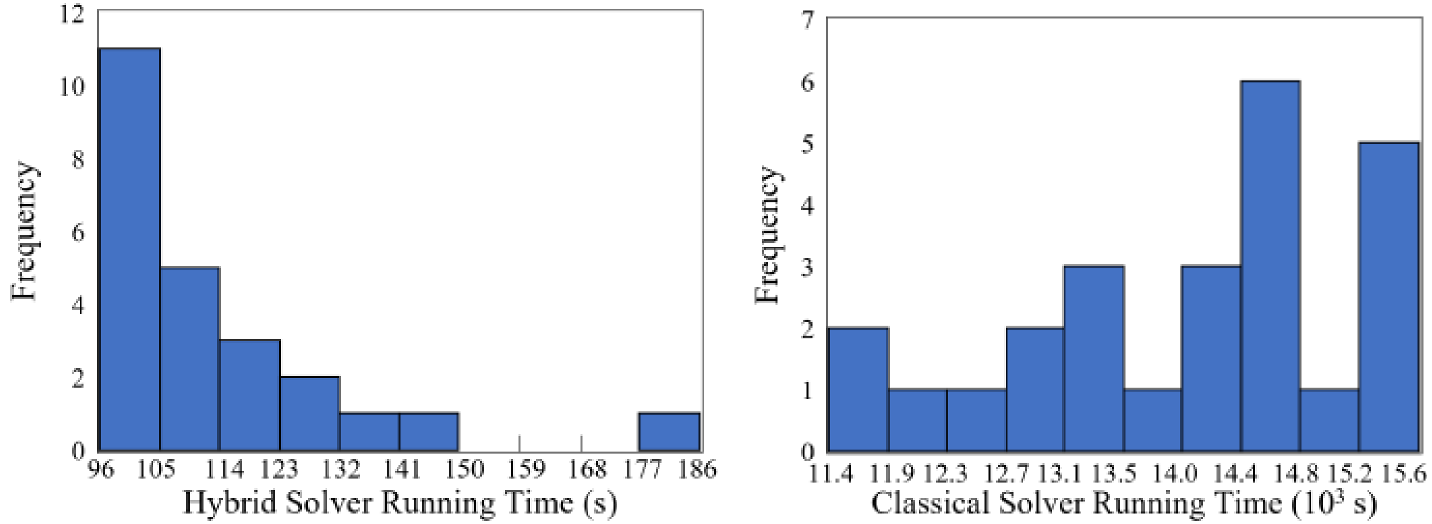
Fig. The solver running time.

- Sub-problem 1: hybrid solvers spend **much shorter time**
- Sub-problem 2: hybrid solvers may spend more time ---- *run serval times to reach the optimal solution*

Simulation Result



(a) Case a (b) Case a



(c) Case b (d) Case b

Fig. the total running time for hybrid solvers and classical solvers in case a and b.

- hybrid solvers:

Distribution: concentrated

- Classical solvers:

Distribution: dispersed

Summary of Work II

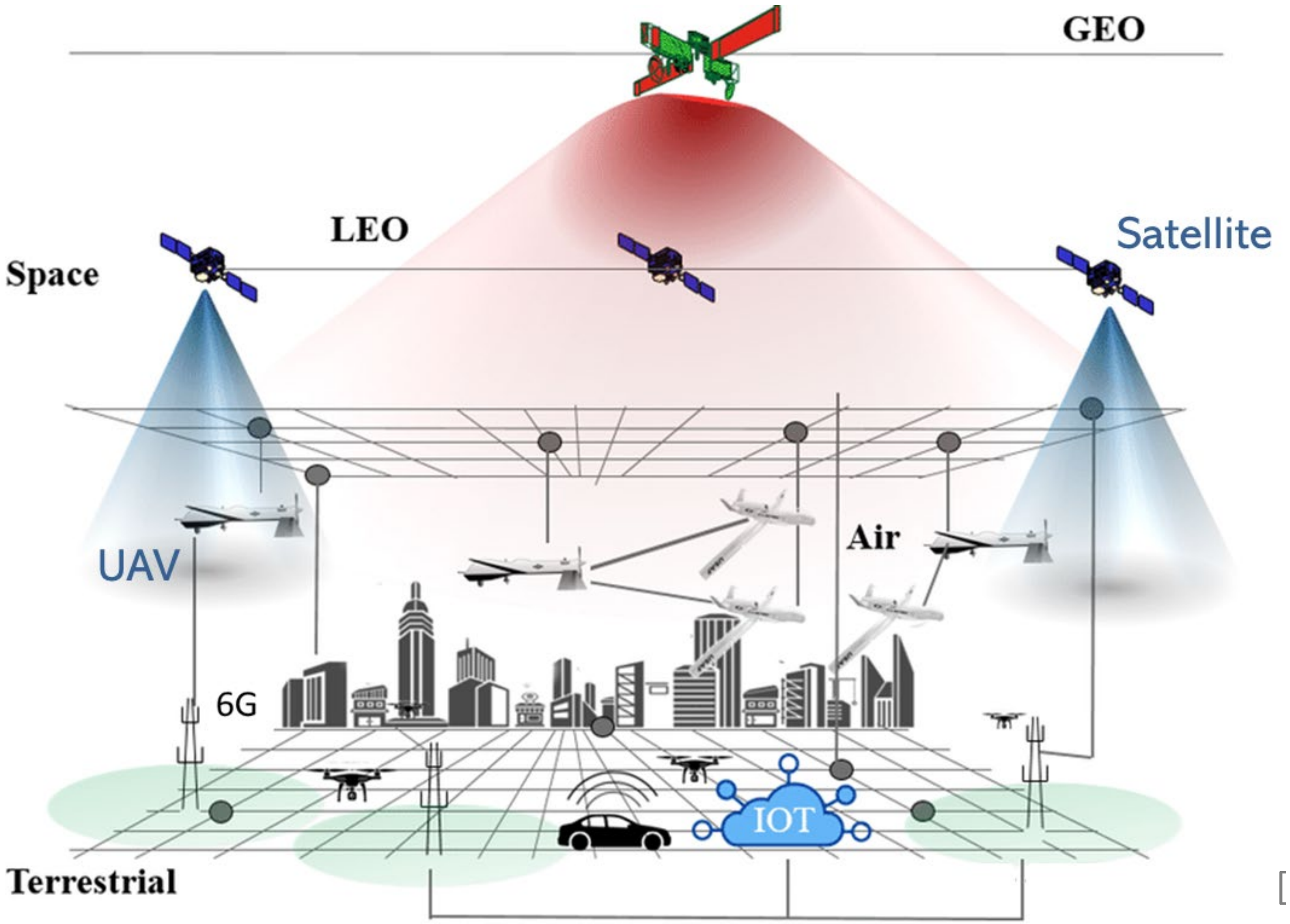
- A novel parallelized quantum annealing algorithm:
  - ✓ Lagrangian relaxation
  - ✓ quantum annealing
- An ILP model of the VNFs scheduling problem:
  - ✓ processing delay
  - ✓ transmission delay
- The superiority of the proposed algorithm:
  - ✓ shorter solving time
  - ✓ stable

Quantum
Advantage?

Outline

- Introduction
- Work I: Minimizing Delay in Network Function Virtualization with Quantum Computing
- Work II: Lagrangian Relaxation Based Parallelized Quantum Annealing and Its Application in Network Function Virtualization
- **Work III: Parallelized Quantum-Inspired Algorithm with Augmented Lagrangian for Data Collection in UAV-enabled IoT Networks**
- Future Works and Conclusions

Motivation



[3]

- Global communication environment
- Internet of Things (IoT) in remote areas
- Wide-area coverage in remote and underserved regions: **Satellites, UAVs**
- **UAVs**: improve network flexibility and responsiveness

Quantum computing:
UAV Trajectory Optimization

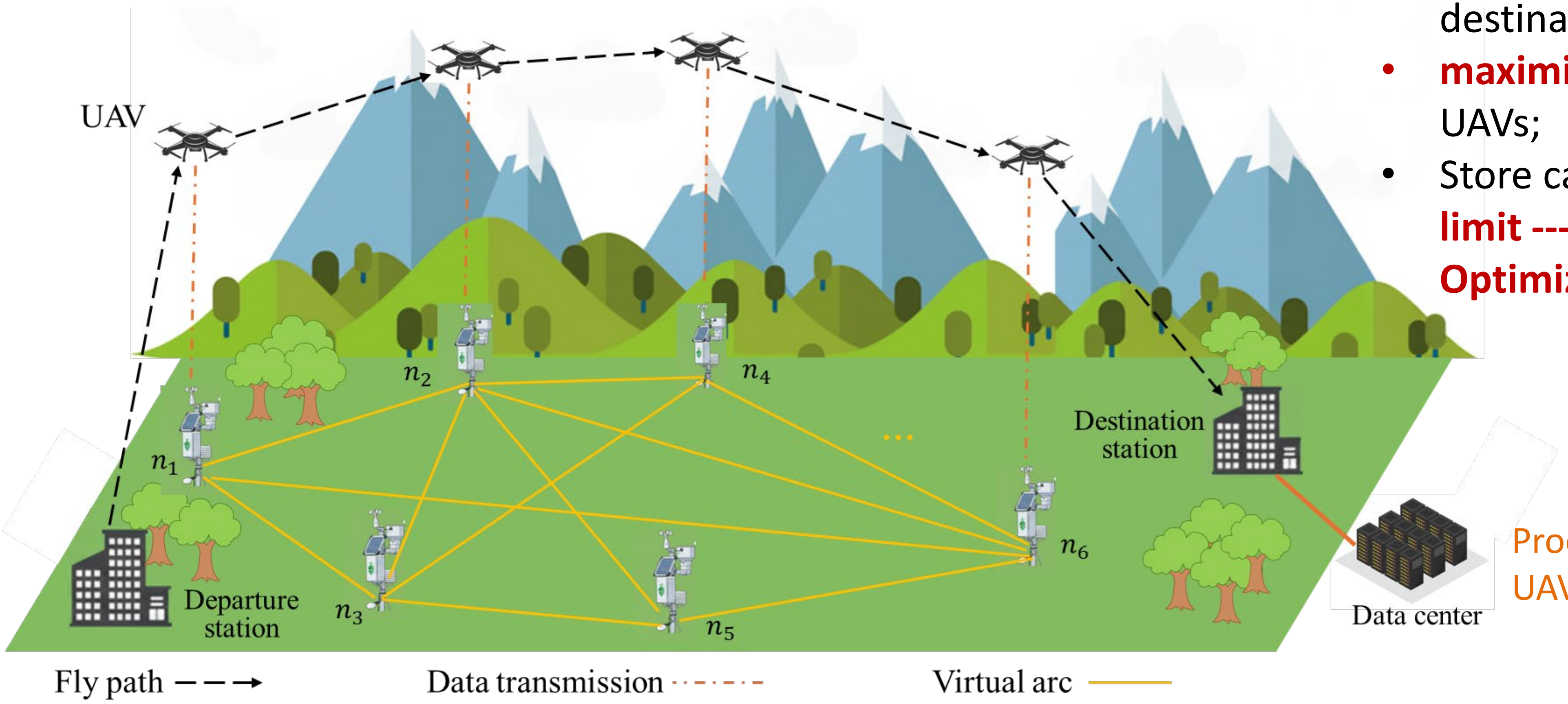
[3] N. Hosseini, et al. "UAV Command and Control, Navigation and Surveillance: A Review of Potential 5G and Satellite Systems," in *IEEE Aerospace Conference*, pp. 1-10, Big Sky, MT, Mar. 2019.

System Model

IoT in remote areas: relay on UAVs to transmit information that can tolerate delays.

Integrated sensors: monitor soil parameters …

- UAVs **collect** data from the IoT nodes and **carry** it to the destination station;
- **maximize** the data collected by all UAVs;
- Store capacity limit and **Energy limit ---- UAV Trajectory Optimization**



Process data collected by UAVs to do analysis.

Formulation: Energy Cost

The total energy cost of the UAV is composed of trajectory, takeoff and landing.

$$E_{\text{traj}}^{u} = E_{f}^{u} + E_{\text{ta}}^{u} + E_{\text{la}}^{u}$$

Constant: $E_{\text{s}}^{u} = \frac{W^{3/2}}{\sqrt{2\rho S}}$

$$E_{f}^{u} = T^{u} \cdot \left(\kappa_1 v_u^3 + \frac{\kappa_2}{v_u} \right)$$

[4] **Carry data and fly to the destination station**

$$T^{u} = \frac{\sum_{i \in \mathcal{I}'} \sum_{j \in \mathcal{I}''} y_{i,j}^{u} d_{i,j}}{v_u}$$

Flight time

$$E_{f}^{u} = \sum_{i \in \mathcal{I}'} \sum_{j \in \mathcal{I}''} y_{i,j}^{u} \boxed{d_{i,j} \cdot \left(\kappa_1 v_u^2 + \frac{\kappa_2}{v_u^2} \right)}$$

Parameter: $K_{i,j}^{u}$

For each UAV: energy capacity limitation

$$\sum_{i \in \mathcal{I}'} \sum_{j \in \mathcal{I}''} y_{i,j}^{u} d_{i,j} \cdot \left(\kappa_1 v_u^2 + \frac{\kappa_2}{v_u^2} \right) + kE_{\text{s}}^{u} \leq \theta \cdot E^{u}$$

$$\boxed{\sum_{i \in \mathcal{I}'} \sum_{j \in \mathcal{I}''} K_{i,j}^{u} y_{i,j}^{u} \leq E_{r}^{u}}$$

[4] Z. Jia, M. Sheng, J. Li, D. Niyato and Z. Han, "LEO-Satellite-Assisted UAV: Joint Trajectory and Data Collection for Internet of Remote Things in 6G Aerial Access Networks," in *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9814-9826, Sep. 2020.

Problem Formulation

ILP Model

x_i^u	equals to 1, if UAV u collects the data from sensor i ; otherwise, equals to 0
$y_{i,j}^u$	equals to 1, if UAV u flies along the path from i to j ; otherwise, equals to 0
$z_{i,j}^u$	equals to 1, if UAV u flies over sensor i and sensor j ; otherwise, equals to 0

$$\max_{\mathbf{x}} \sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}} \phi_i x_i^u \qquad (23)$$

subject to

$$\boxed{\sum_{u\in\mathcal{U}} x_i^u \le 1 \quad \forall i \in \mathcal{I}.} \qquad (24)$$

$$\sum_{i\in\mathcal{I}} \phi_i x_i^u \le D^u \quad \forall u \in \mathcal{U}. \qquad (25)$$

Storage Capacity limitation

$$x_j^u \le \sum_{i\in\mathcal{I}'} y_{i,j}^u \quad \forall u \in \mathcal{U},\ j \in \mathcal{I}. \qquad (26)$$

$$\sum_{i\in\mathcal{I}'}\sum_{j\in\mathcal{I}''} K_{i,j}^u y_{i,j}^u \le E_r^u \qquad (27)$$

Energy Capacity limitation

$$\sum_{j\in\mathcal{I}''} y_{o,j}^u = 1 \quad \forall u \in \mathcal{U}. \qquad (28)$$

$$\sum_{i\in\mathcal{I}'} y_{i,d}^u = 1 \quad \forall u \in \mathcal{U}. \qquad (29)$$

$$\sum_{i\in\mathcal{I}'} y_{i,j}^u = \sum_{i'\in\mathcal{I}''\vee(i'\neq i)} y_{j,i'}^u \quad \forall u \in \mathcal{U},\ j \in \mathcal{I}. \qquad (30)$$

$$y_{i,j}^u + y_{j,i}^u \le 1 \quad \forall u \in \mathcal{U} \quad i,j \in \mathcal{I}, i \neq j. \qquad (31)$$

$$y_{i,j}^u \le z_{i,j}^u \quad \forall u \in \mathcal{U} \quad i \in \mathcal{I}', j \in \mathcal{I}'', i \neq j. \qquad (32)$$

$$z_{i,k}^u + z_{k,j}^u - z_{i,j}^u \le 1$$
$$\forall u \in \mathcal{U} \quad i \in \mathcal{I}', j \in \mathcal{I}'', k \in \mathcal{I}, i \neq j \neq k. \qquad (33)$$

$$z_{i,j}^u + z_{j,i}^u \le 1 \quad \forall u \in \mathcal{U} \quad i,j \in \mathcal{I}, i \neq j. \qquad (34)$$

Eliminate Sub-Loop

Problem Formulation

For each block: Isolated UAV trajectory

QUBO Model

$$P_{1,u}\left(\sum_{i\in\mathcal{I}}\phi_i x_i^u - D^u + \sum_l 2^l r_{1,l}\right)^2 . \quad (35)$$

$$P_{1,u,j}\left(\sum_{i\in\mathcal{I}'} y_{i,j}^u - x_j^u + \sum_l 2^l r_{2,l}\right)^2 \quad \forall j\in\mathcal{I}. \quad (36)$$

$$P_{2,u}\left(\sum_{i\in\mathcal{I}'}\sum_{j\in\mathcal{I}''} K_{i,j}^u y_{i,j}^u - E_r^u + \sum_l 2^l r_{3,l}\right)^2 . \quad (37)$$

$$P_{1,u}\left(\sum_{j\in\mathcal{I}''} y_{o,j}^u - 1\right) . \quad (38)$$

$$P_{2,u}\left(\sum_{j\in\mathcal{I}'} y_{i,d}^u - 1\right)^2 . \quad (39)$$

x_i^u	equals to 1, if UAV u collects the data from sensor i ; otherwise, equals to 0
$y_{i,j}^u$	equals to 1, if UAV u flies along the path from i to j ; otherwise, equals to 0
$z_{i,j}^u$	equals to 1, if UAV u flies over sensor i and sensor j ; otherwise, equals to 0

$$P_{2,u}\left(\sum_{i\in\mathcal{I}'} y_{i,j}^u - \sum_{i\in\mathcal{I}''\vee(i\neq i')} y_{j,i'}^u\right)^2 , \forall j. \quad (40)$$

$$P_{1,u,i,j}\left(y_{i,j}^u \cdot y_{j,i}^u\right), \quad \forall i,j\in\mathcal{I}, i\neq j \quad (41)$$

$$P_{2,u,i,j}\left(y_{i,j}^u - y_{i,j}^u z_{i,j}^u\right) \quad \forall i\in\mathcal{I}', j\in\mathcal{I}'', i\neq j. \quad (42)$$

$$P_{u,i,j,k}\left(z_{i,k}^u + z_{k,j}^u - z_{i,j}^u + r_1 + 2r_2\right)^2 \quad (43)$$
$$\forall i\in\mathcal{I}', j\in\mathcal{I}'', k\in\mathcal{I}, i\neq j\neq k.$$

Slack variable

$$P_{3,u,i,j}\left(z_{i,j}^u \cdot z_{j,i}^u\right), \quad \forall i,j\in\mathcal{I}, i\neq j \quad (44)$$

Algorithm

Augmented Lagrangian Relaxation

- ❑ The original model with the block structure:

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{v} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{v} \le \mathbf{b}, \\ & \mathbf{B}_u \mathbf{v}_u \le \mathbf{d}_u, \quad u \in U, \\ & \mathbf{v}_u \in \{0,1\}^{n_u}. \end{aligned}$$

- ❑ Every block:

$$\begin{aligned} & \mathbf{B}_u \mathbf{v}_u \le \mathbf{d}_u, \quad u \in U, \\ & \mathbf{v}_u \in \{0,1\}^{n_u}. \end{aligned}$$

Augmented Lagrangian Relaxation

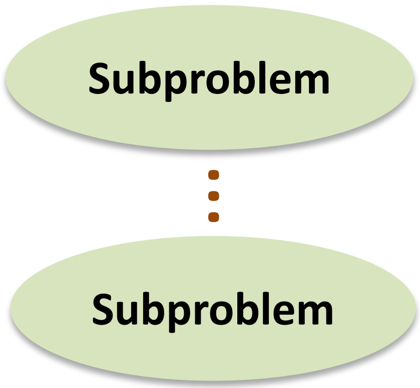
$$L(\mathbf{v},\lambda,\rho) = \sum_{u\in U} \mathbf{c}_u^\top \mathbf{v}_u + \lambda^\top \underbrace{\left(\sum_{u\in U} \mathbf{A}_u \mathbf{v}_u - \mathbf{b}\right)}_{\lambda \in \mathbb{R}^m_+} + \frac{\rho}{2} \underbrace{\left\| \left(\sum_{u\in U} \mathbf{A}_u \mathbf{v}_u - \mathbf{b}\right)_+ \right\|^2}_{\substack{\textit{Penalty term} \\ \rho > 0}}$$

- ❑ update λ and ρ :

$$\lambda^{k+1} = \left(\lambda^k + \gamma^k (\mathbf{A}\mathbf{v}^{k+1} - \mathbf{b})\right)_+$$

$$\rho^{k+1} = \rho^k + \frac{\gamma^k}{2} \left\| (\mathbf{A}\mathbf{v}^{k+1} - \mathbf{b})_+ \right\|^2$$

- ❑ Decompose original model to subproblem models



$$\begin{aligned} & \min L_u^t(\mathbf{v}_u,\lambda,\rho) \\ \text{s.t.} \quad & \mathbf{B}_u \mathbf{v}_u \le \mathbf{d}_u, \quad u \in U, \\ & \mathbf{v}_u \in \{0,1\}^{n_u}. \end{aligned}$$

Algorithm

Customized Augmented Lagrangian Method (CALM)

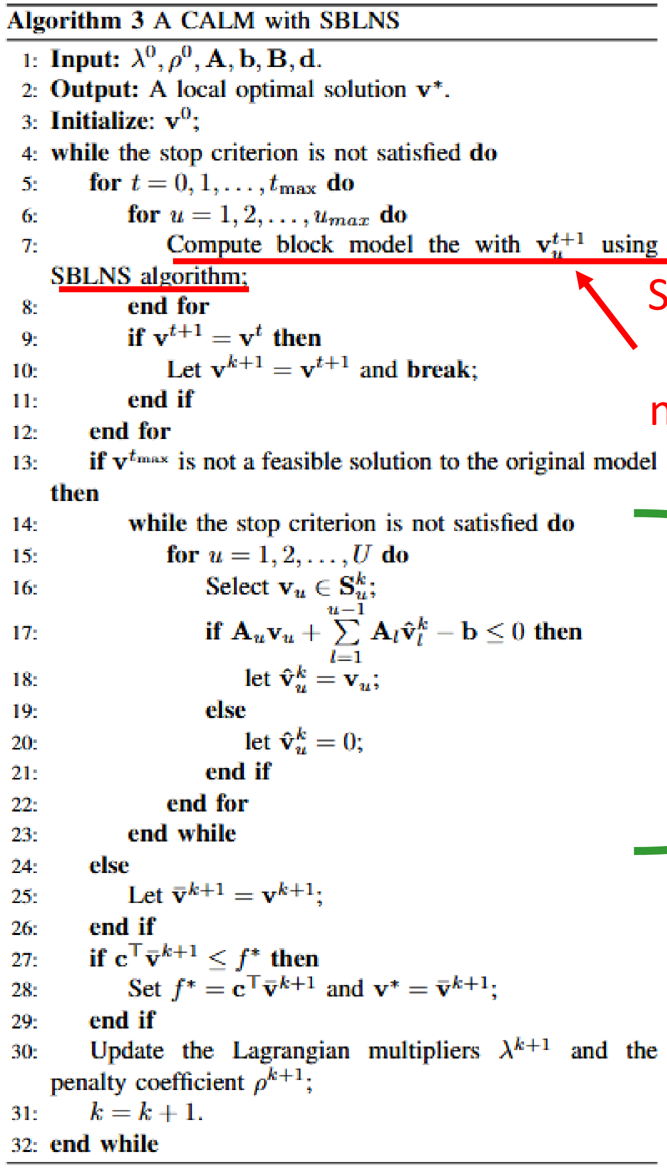
❑ Block Coordinate Descent (BCD) Method

- Iteratively solve the function L in the order of $\mathbf{v}_1, \ldots, \mathbf{v}_u, \ldots$;
- BCD iteratively optimizes over each block $\mathbf{v}_u$ keeping the others fixed. During each iteration

$$L_u^t(\mathbf{v}_u, \lambda, \rho) = L(\mathbf{v}_1^{t+1}, \ldots, \mathbf{v}_{u-1}^{t+1}, \mathbf{v}_u, \mathbf{v}_{u+1}^t, \ldots, \lambda, \rho)$$

- Simplify the objective function of block model u :

$$\mathbf{v}_u^{t+1} \in \arg\min L_u^t(\mathbf{v}_u; \lambda, \rho)$$
$$= \mathbf{v}_u \left(\mathbf{c}_u + \mathbf{A}_u^\top \lambda + \rho \mathbf{A}_u^\top \left(\sum_{l \neq u}^{p} \mathbf{A}_l \mathbf{v}_l^t(u) - \frac{1}{2} \right) \right)_+$$



Algorithm 3 A CALM with SBLNS

```
Input: $\lambda^0, \rho^0, \mathbf{A}, \mathbf{b}, \mathbf{B}, \mathbf{d}$.
Output: A local optimal solution $\mathbf{v}^*$.
Initialize: $\mathbf{v}^0$;
while the stop criterion is not satisfied do
    for $t = 0, 1, \ldots, t_{\max}$ do
        for $u = 1, 2, \ldots, u_{max}$ do
            Compute block model the with $\mathbf{v}_u^{t+1}$ using SBLNS algorithm;
        end for
        if $\mathbf{v}^{t+1} = \mathbf{v}^t$ then
            Let $\mathbf{v}^{k+1} = \mathbf{v}^{t+1}$ and break;
        end if
    end for
    if $\mathbf{v}^{t_{\max}}$ is not a feasible solution to the original model then
        while the stop criterion is not satisfied do
            for $u = 1, 2, \ldots, U$ do
                Select $\mathbf{v}_u \in \mathbf{S}_u^k$;
                if $\mathbf{A}_u \mathbf{v}_u + \sum_{l=1}^{u-1} \mathbf{A}_l \hat{\mathbf{v}}_l^k - \mathbf{b} \leq 0$ then
                    let $\hat{\mathbf{v}}_u^k = \mathbf{v}_u$;
                else
                    let $\hat{\mathbf{v}}_u^k = 0$;
                end if
            end for
        end while
    else
        Let $\bar{\mathbf{v}}^{k+1} = \mathbf{v}^{k+1}$;
    end if
    if $\mathbf{c}^\top \bar{\mathbf{v}}^{k+1} \leq f^*$ then
        Set $f^* = \mathbf{c}^\top \bar{\mathbf{v}}^{k+1}$ and $\mathbf{v}^* = \bar{\mathbf{v}}^{k+1}$;
    end if
    Update the Lagrangian multipliers $\lambda^{k+1}$ and the penalty coefficient $\rho^{k+1}$;
    $k = k + 1$.
end while
```

Simulated bifurcation with adaptive large neighborhood search (SBALNS)

Select values from the solution pool to generate a feasible solution

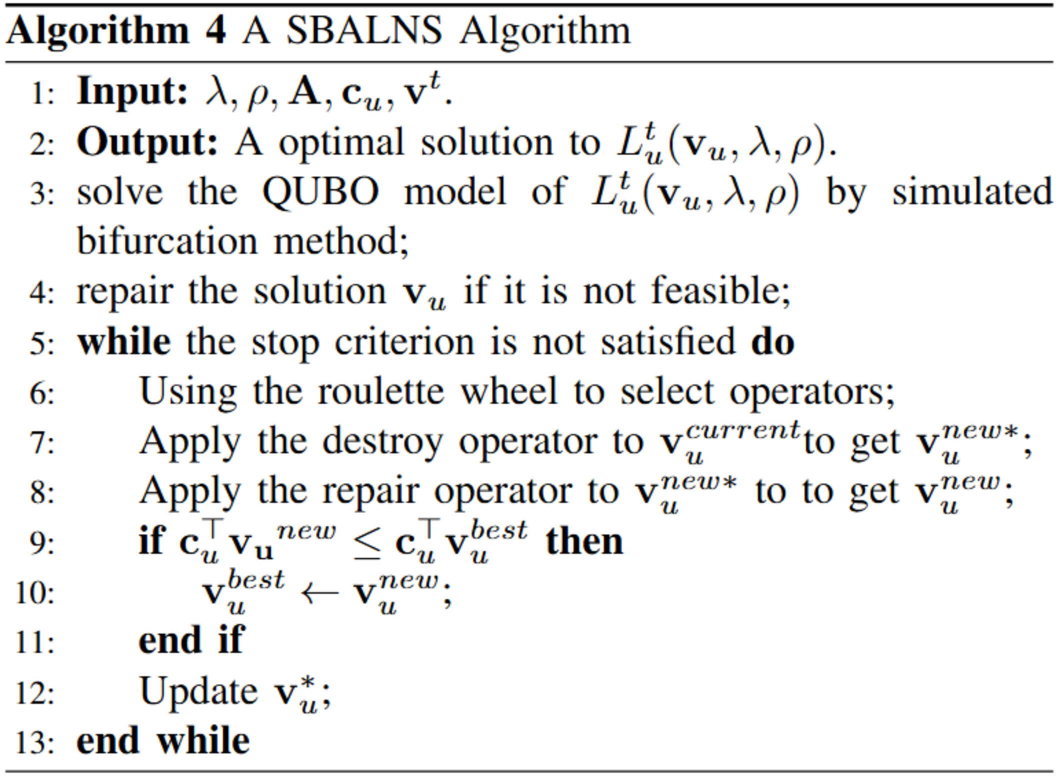
Algorithm

SBALNS Algorithm

- ❑ Simulated Bifurcation
  - A quantum-inspired algorithm;
  - Simulate the adiabatic evolution of a quantum Hamiltonian system;
  - Follow adiabatic and ergodic evolutions of classical nonlinear Hamiltonian systems;
  - Solve QUBO model:

$$f(x) = \sum_{i} \boldsymbol{Q}_{i,i} x_i + \sum_{i} \sum_{i<j} \boldsymbol{Q}_{i,j} x_i \, x_j + c$$

  - Evolve several spin vectors (agent) in parallel;



Algorithm 4 A SBALNS Algorithm

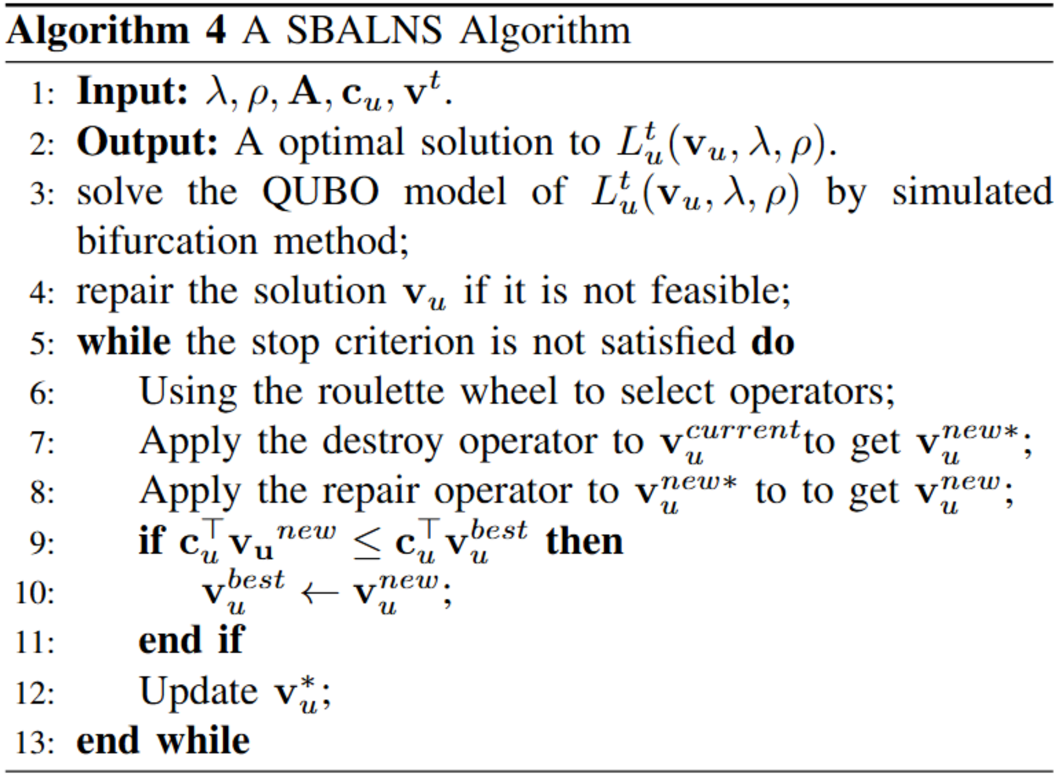
```
Input: $\lambda, \rho, \mathbf{A}, \mathbf{c}_u, \mathbf{v}^t$.
Output: A optimal solution to $L_u^t(\mathbf{v}_u, \lambda, \rho)$.
solve the QUBO model of $L_u^t(\mathbf{v}_u, \lambda, \rho)$ by simulated bifurcation method;
repair the solution $\mathbf{v}_u$ if it is not feasible;
while the stop criterion is not satisfied do
    Using the roulette wheel to select operators;
    Apply the destroy operator to $\mathbf{v}_u^{current}$to get $\mathbf{v}_u^{new*}$;
    Apply the repair operator to $\mathbf{v}_u^{new*}$ to to get $\mathbf{v}_u^{new}$;
    if $\mathbf{c}_u^\top \mathbf{v_u}^{new} \leq \mathbf{c}_u^\top \mathbf{v}_u^{best}$ then
        $\mathbf{v}_u^{best} \leftarrow \mathbf{v}_u^{new}$;
    end if
    Update $\mathbf{v}_u^*$;
end while
```

Algorithm

SBALNS Algorithm

❑ Adaptive large Neighborhood Search

- A heuristic algorithm;
- Need an initial solution;
- Operators: destroy operators and repair operators;
- Roulette wheel selection: updates operator weights ---- decide which operators are used to generate the new candidate solution;
- Evaluate the new candidate solution;
- Acceptance criterion: Record-to-record travel ---- accept solutions when the improvement meets the threshold.



Algorithm 4 A SBALNS Algorithm

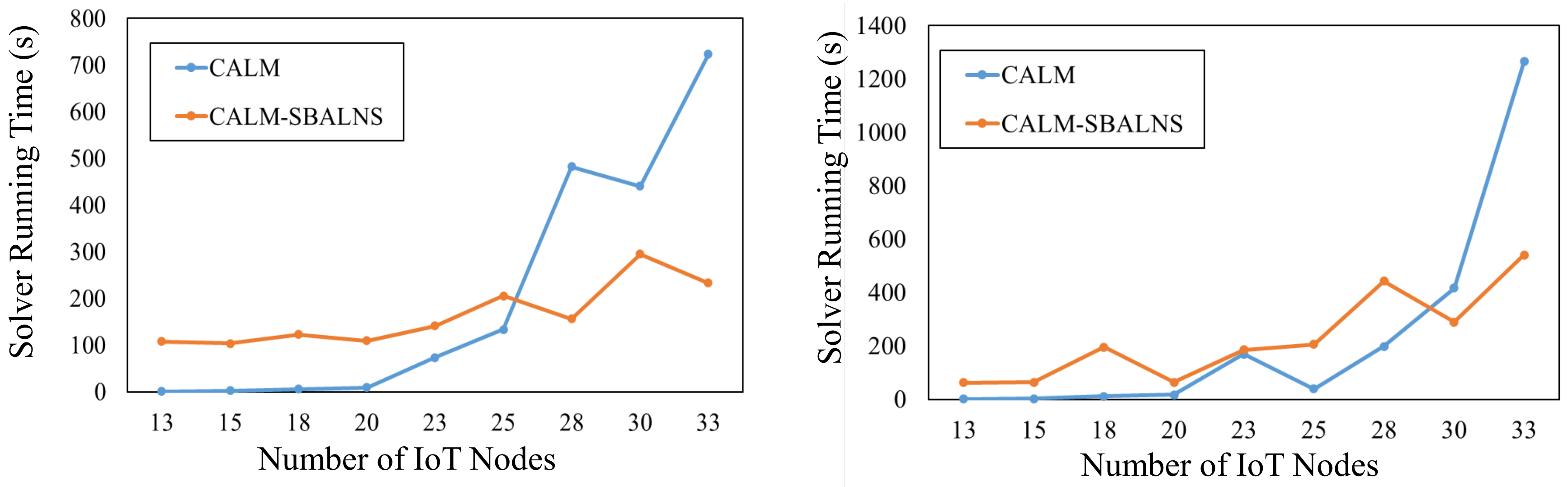
```
**Input:** $\lambda, \rho, \mathbf{A}, \mathbf{c}_u, \mathbf{v}^t$.
**Output:** A optimal solution to $L_u^t(\mathbf{v}_u, \lambda, \rho)$.
solve the QUBO model of $L_u^t(\mathbf{v}_u, \lambda, \rho)$ by simulated bifurcation method;
repair the solution $\mathbf{v}_u$ if it is not feasible;
**while** the stop criterion is not satisfied **do**
    Using the roulette wheel to select operators;
    Apply the destroy operator to $\mathbf{v}_u^{current}$to get $\mathbf{v}_u^{new*}$;
    Apply the repair operator to $\mathbf{v}_u^{new*}$ to to get $\mathbf{v}_u^{new}$;
    **if** $\mathbf{c}_u^\top \mathbf{v_u}^{new} \leq \mathbf{c}_u^\top \mathbf{v}_u^{best}$ **then**
        $\mathbf{v}_u^{best} \leftarrow \mathbf{v}_u^{new}$;
    **end if**
    Update $\mathbf{v}_u^*$;
**end while**
```

Simulation Results

- **Cplex**: use the Cplex solver to solve the original model;
- **CALM**: use the CALM to solve the Augmented Lagrangian model by using Cplex solvers to computer the block models;
- **CALM-SBALNS**: Run simulated bifurcation algorithm on GPU and other parts are using CPU;

Case	Parameter		Cplex		CALM		CALM-SBALNS	
	Number of UAVs	Number of IoT Nodes	Objective Value	Total Solver Running Time (sec)	Objective Value	Total Solver Running Time (sec)	Objective Value	Total Solver Running Time(sec)
Case a	3	13	51.33	13.09	49.25	1.03	49.57	107.95
Case b	3	20	51.55	670.21	51.27	9.02	49.85	109.61
Case c	3	25	51.55	1761.78	50.72	133.18	49.69	205.09
Case d	3	30	--	--	48.12	440.61	47.36	294.85
Case e	5	30	--	--	86.50	416.83	83.40	288.39
Case f	3	33	--	--	49.67	723.29	44.50	233.40
Case g	5	33	--	--	99.80	1265.10	98.70	540.56
Case h	7	33	--	--	135.93	3132.56	138.28	537.21

Simulation Results

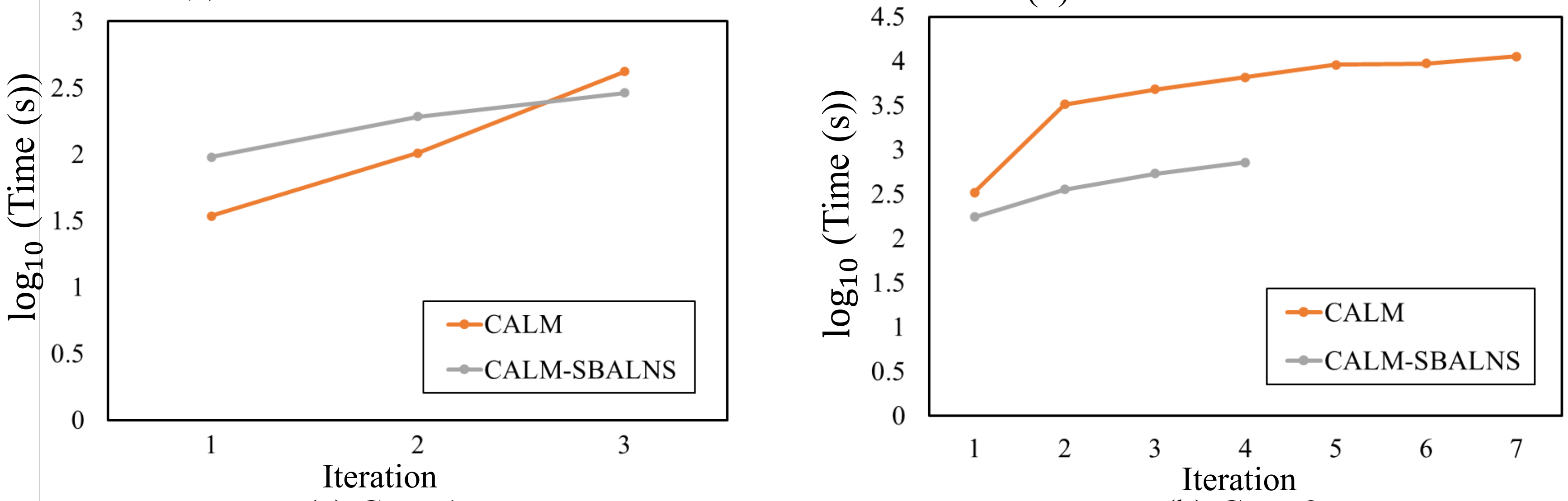


(a) Number of UAV $U = 3$

(b) Number of UAV $U = 5$

Fig. The solver running time.

- ➢ As the number of IoT nodes increases, the solver running time of CALM-SBALNS increases slowly.



(a) Case 1

(b) Case 2

Fig. The accumulated solver running time.

- ➢ In every iteration, the solver running time of CALM-SBALNS keeps approximately stable.

Summary of Work III

- A novel parallelized quantum-inspired algorithm:
  - ✓ Augmented Lagrangian relaxation
  - ✓ BCD method
  - ✓ SB + ALNS
- An ILP model of the UAV trajectory optimization problem:
  - ✓ maximize the data collection
- The superiority of the CALM-SBALNS algorithm:
  - ✓ shorter solving time
  - ✓ Increase slower

Quantum Advantage?

XXX Quantum stock price

Outline

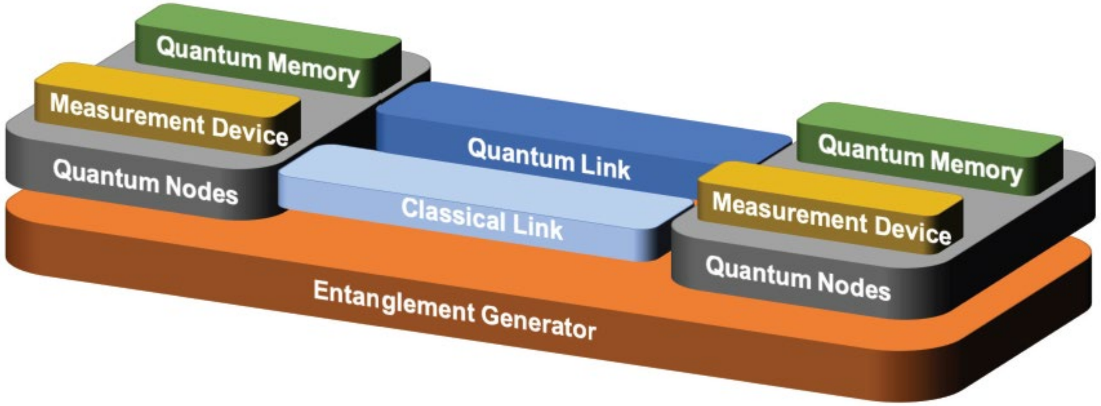
- Introduction
- Work I: Minimizing Delay in Network Function Virtualization with Quantum Computing
- Work II: Lagrangian Relaxation Based Parallelized Quantum Annealing and Its Application in Network Function Virtualization
- Work III: Parallelized Quantum-Inspired Algorithm with Augmented Lagrangian for Data Collection in UAV-enabled IoT Networks
- **Future Works and Conclusions**

Future Works

Quantum Communication & Networking

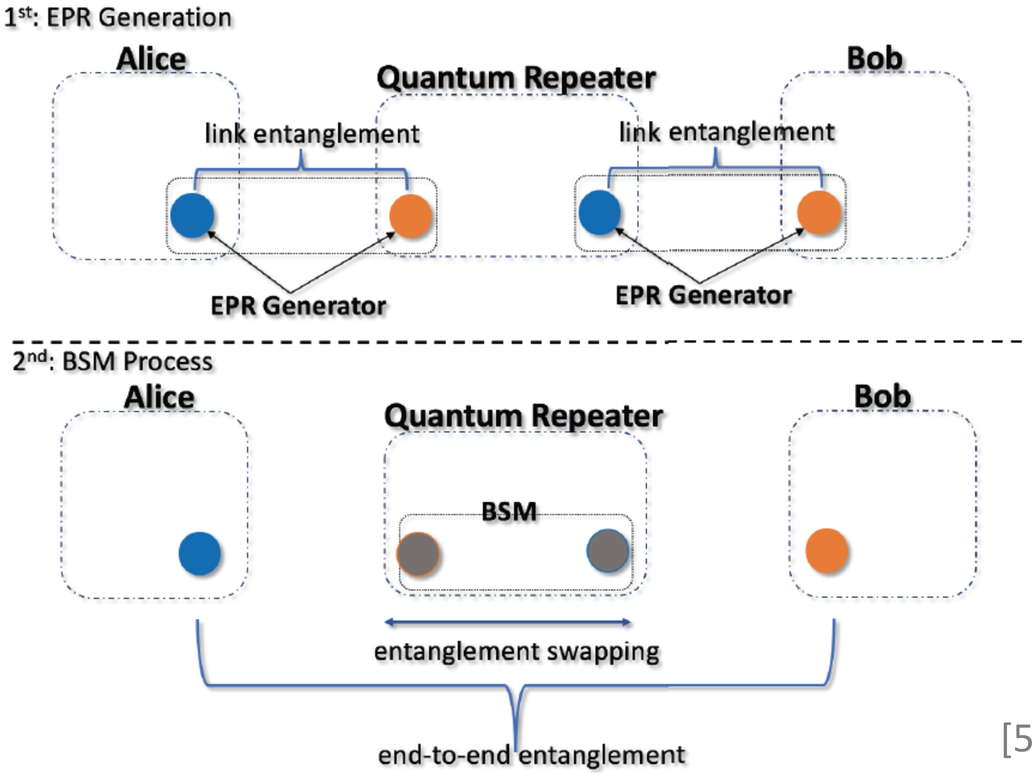
Quantum Internet Architecture:

- Quantum Links: support the transmission of photonic qubits
- Quantum Memories: Storage qubits that hold quantum states temporarily
- …



[5]

Quantum devices: entanglement and qubit distribution ---- **optimization**

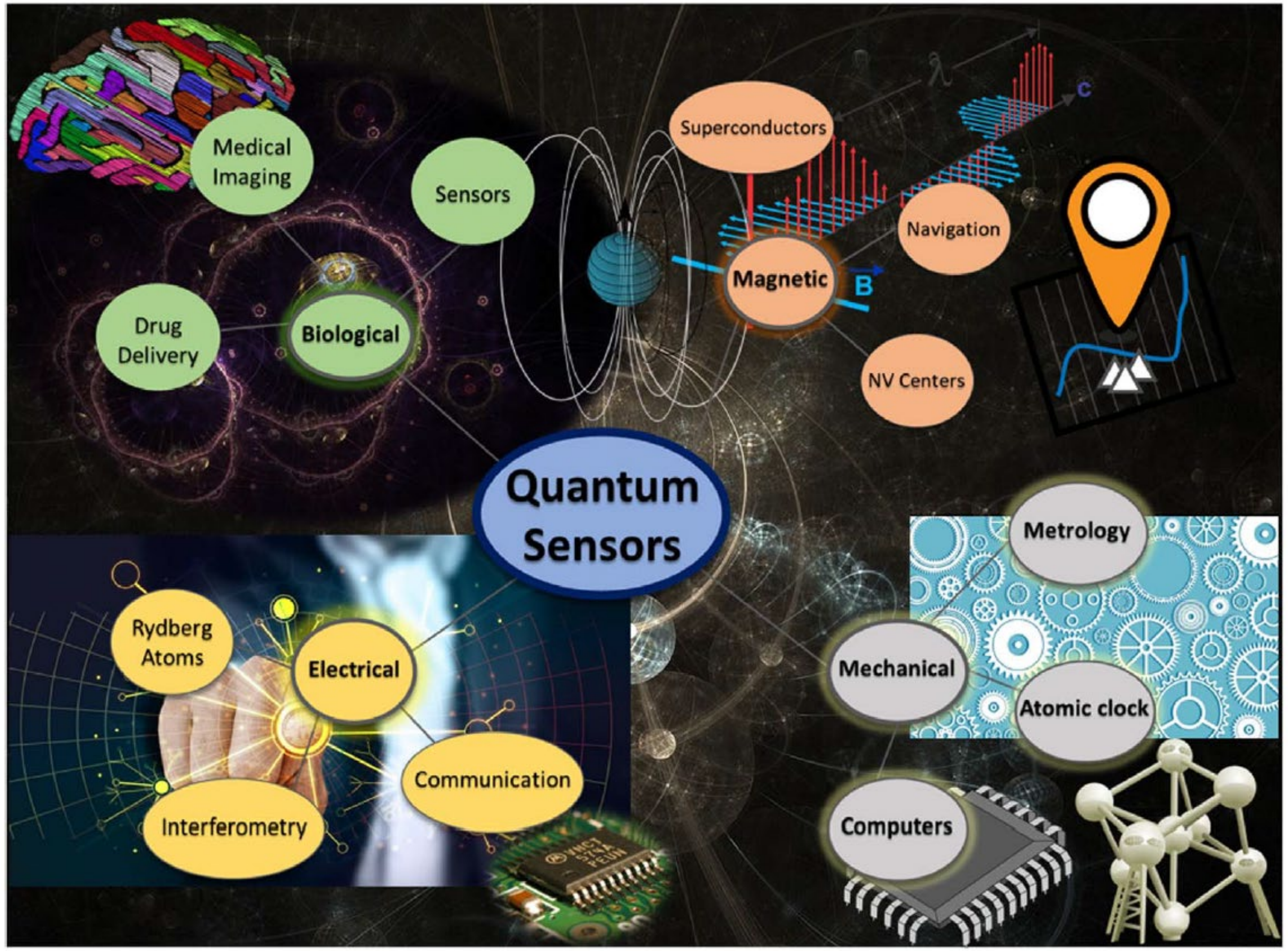


[5]

[5] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini and G. Bianchi, "Quantum Internet: Networking Challenges in Distributed Quantum Computing," *IEEE Network*, vol. 34, no. 1, pp. 137-143, Nov. 2019.

Future Works

Quantum Sensors

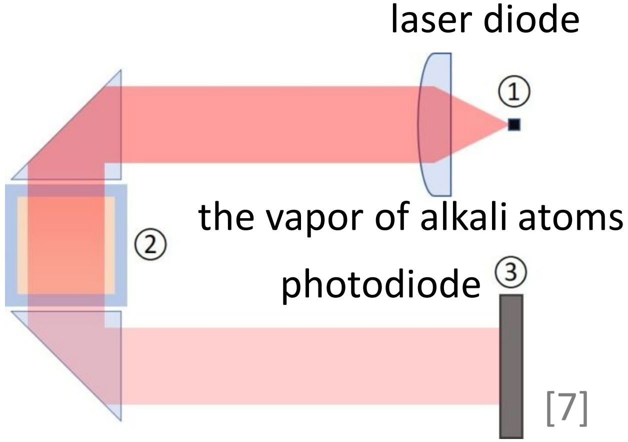


[6]

Signal processing:

- Quantum sensors are sensitive to environmental perturbations;
- Filter noise and enhance SNR;

E.g. Zero-Field OPM (Optically Pumped Magnetometer)



[7]

"cross-talk" effects

Signal processing

[6] https://quspin.com/products-qzfm-gen2-arxiv/zero-field-magnetometer-description/

[6] V. Chugh, A. Basu, A. Kaushik, and A.K. Basu, "Progression in quantum sensing/bio-sensing technologies for healthcare," ECS Sensors Plus, vol. 2, no. 1, pp. 015001, 2023.

Conclusions

- **Work I** Minimizing Delay in Network Function Virtualization with Quantum Computing.
  - ✓ first use quantum computing to solve the optimization problem in NFV
- **Work II** Parallelized Quantum Annealing and Its Application in Network Function Virtualization.
  - ✓ Compared with classical solver: shorter solving time and more stable
- **Work III** Parallelized Quantum-Inspired Algorithm for Data Collection in UAV-enabled IoT Networks.
  - ✓ Compared with classical CLAM algorithm: shorter solving time and increases slower as the problem size increases
- **In conclusion:** quantum advantages in solving large-scale optimization problems in communication networks.

Publications

- Journal
  - **Wenlu Xuan**, Zhongqi Zhao, Lei Fan, and Zhu Han, “Lagrangian Relaxation Based Parallelized Quantum Annealing and Its Application in Network Function Virtualization,” IEEE Open Journal of the Communications Society, vol. 5, p.p. 4260 – 4274, July 2024.
  - **Wenlu Xuan**, Lei Fan, and Zhu Han, “Parallelized Quantum-Inspired Algorithm with Augmented Lagrangian Relaxation for UAV trajectory optimization in the Internet of Things”, IEEE Transactions on Network Science and Engineering, ongoing.
  - **Wenlu Xuan**, Lei Fan, and Zhu Han, “ Novel QUBO-Transforming Method for Network Slicing in Open-RAN”, in preparation.
- Conferences
  - **Wenlu Xuan**, Zhongqi Zhao, Lei Fan, and Zhu Han, “Minimizing Delay in Network Function Virtualization with Quantum Computing”. in Proc. IEEE 18th Int. Conf. Mobile Ad Hoc Smart Syst. (MASS), Dec. 2021, pp. 108–116.

Should I approve Wenlu’s Defense?

Approve

Still

Approve

Thank you for your attention.