Received 3 January 2024; revised 18 January 2024; accepted 19 January 2024. Date of publication 29 January 2024; date of current version 13 February 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3359188

Warm and Cold Start Quantum Annealing for Metaverse Resource Optimization

MAHZABEEN EMU (Graduate Student Member, IEEE), SALIMUR CHOUDHURY (Senior Member, IEEE), AND KAI SALOMAA

School of Computing, Queen's University, Kingston, ON K7L 2N8, Canada

CORRESPONDING AUTHOR: M. EMU (e-mail: 20me21@queensu.ca)

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

ABSTRACT Metaverse refers to the intersection of parallel virtual worlds with their physical counterparts by allowing users to interact with virtual people, objects, and environments. Resource allocation in various aspects of Metaverse domains, called as MetaSlices hereinafter, is a crucial optimization research problem. To serve this purpose, we consider a MetaSlice framework with the notion of sharing resources among common functions and enable placing time-sensitive services at the edge of multi-tier architecture in proximity to users. Unfortunately, the classical Integer Linear Programming is inappropriate for such heavily constrained optimization problem due to the extensive running time and memory. Hence, we model a novel Quadratic Unconstrained Binary Optimization (QUBO) formulation to simultaneously optimize resources and secure Quality of Service for MetaSlices as a paradigm shift towards quantum computing. Furthermore, we propose to employ two hybrid classical-quantum strategies, Warm Start and Cold Start Quantum Annealing to optimize resource under bandwidth uncertainty, offer ultra-low running time, and increase service acceptance rate/scalability in a resource-hungry and dynamic Metaverse system.

INDEX TERMS Quantum annealing, metaverse, combinatorial optimization, warm start, cold start, stochastic optimization.

I. INTRODUCTION

THE CONCEPT of Metaverse initially emerged in a fiction novel called Snow Crash authored by Neal Stephenson in 1992 [1]. Thanks to the recent advancements of cutting-edge technologies (e.g., smart sensors, virtual/augmented reality, next-generation networks), Metaverse has become a buzzword both in academia and industry [2], [3], [4]. Being committed to support the development of Metaverse ecosystem, Facebook re-branded itself as "Meta" in 2021 by taking the spotlight off social media and universalizing multiple virtual worlds [4]. Other tech giants, such as, Microsoft, Amazon are also investing heavily based on acknowledging Metaverse as the successor of Tactile Internet and believing to work towards "moving beyond what's possible today" [2]. Unlike the existing virtual world platforms (e.g., Roblox, Second Life), Metaverse is considered to be a seamless and unique integration of multiple virtual aspects of life, such as, entertainment, education, e-commerce, healthcare, and smart industry [5]. Metaverse users are expected to share and trade their objects and assets by preserving their value across multiple virtual worlds. One of the main driving forces behind Metaverse is the unexpected arrival of COVID-19 that have brought a significant impact on how people socialize, work, interact, and experience. Recently, UC Berkeley organized their graduation ceremony in Minecraft (gaming platform) and the American rapper Lil Nas X performed in an online concert using Roblox platform with over 35 million views [6]. Furthermore, Metaverse is the key to fulfil the dreams of 6G communication and beyond to overcome the tyranny of physical distance in a fully immersive way [7].

A. MOTIVATION

Metaverse ecosystem is expected to demand enormous amount of resources that have never been seen before [2]. Due to the operations of Metaverse, the data usage rate has been predicted to go up by 20% [2]. The success of the Metaverse is anticipated to be driven by its Quality of

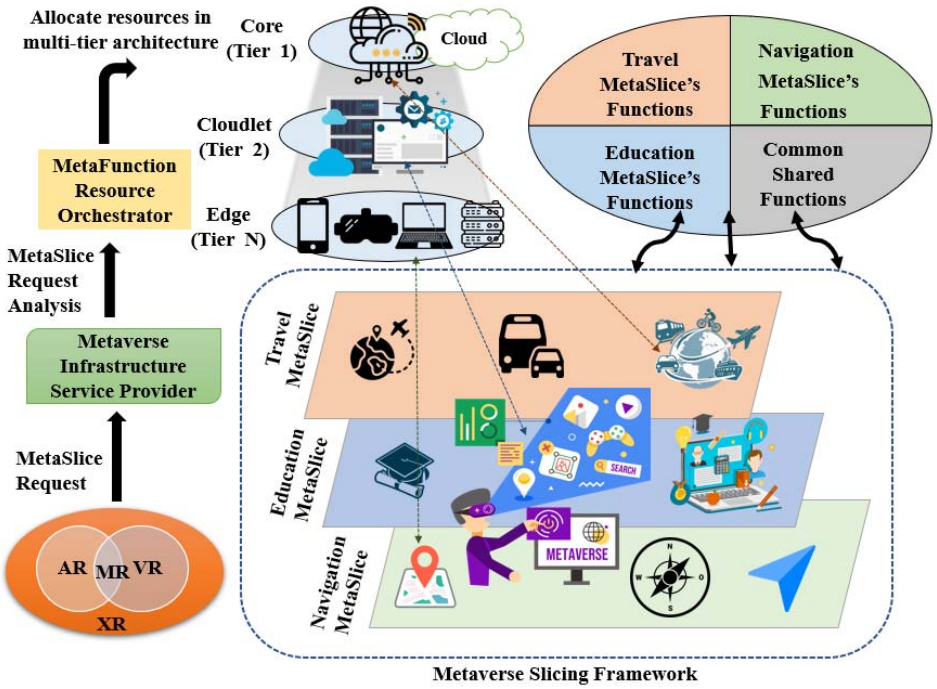


FIGURE 1. Sharing based MetaSlice Resource Allocation.

Service (QoS) and resource sustainability. In order to render 3D objects (augmented reality), 3D environment (virtual reality), and synchronize millions of simultaneous Metaverse applications, shifting to a new paradigm is necessary for optimizing overall resource consumption costs. Failure to ensure real-time Metaverse service delivery can degrade QoS and cause long-term major health issues (e.g., nausea, sever headache, and visual distress) for Metaverse users due to laggy navigation in virtual world with their avatars. In this paper, we propose a sharing based MetaSlice resource allocation in multi-tier architecture framework supported by quantum computing to ensure timely content delivery and optimize resource consumption. Inspired from the analogy of traditional network slice, MetaSlice [6] is defined by a ordered sequences of Meta Network Functions (MNFs) or often termed as Meta Instance. MetaSlices can belong to various application domains, such as, education, tourism, and navigation, as shown in Fig. 1. It is highly likely multiple MetaSlices, such as, navigation and tourism will encounter some common functions (e.g., digital map) [6], [8]. Likewise, Google Map's Application Programming Interface (API) can be shared by multiple Metaverse application for live location sharing, check-in, and routing [9]. The notion of sharing the resources of common MNFs can enable the maximum resource utilization and generate much higher revenue for Metaverse Infrastructure Service Providers (MISPs) [6], [10]. Additionally, the MetaSlices need to be allocated into multi-tier architecture depending upon the Service Level Agreement (SLA) of MISPs and requirements of MNFs [6]. For instance, driving assistance feature and real-time traffic update related MNFs should be placed at the edge of the multi-tier architecture due to their delay-sensitive nature and for ensuring ultra-low latency [6]. On the other hand, digital map that does not need frequent updates may be placed on the cloud relatively far from users [6]. We also consider stochastic nature of multi-tier host architecture (topology) caused by the variable bandwidth flow consumption in MetaSlice resource allocation framework and propose optimization models accordingly. Several reasons, such as, congestion, traffic flow dropping characteristics of MNFs, may cause uncertainties in bandwidth consumption [11], [12]. MetaSlice is more than just a network slicing case for the Metaverse applications. Instead, it is a comprehensive framework for dynamically allocating resources to enhance the QoS tailored to the needs of diverse Metaverse applications [6]. MetaSlice can be considered a platform for deploying Metaverse applications because it provides a framework to manage and optimize the performance of Metaverse applications by allocating different types of resources across various tiers of the computing architecture. MetaSlice incorporates techniques for decomposing Metaverse applications into different functions called Meta Network Functions and shares the resources of the commonly used functions for optimizing resources. The Metaslice framework simultaneously serves the needs of metaverse end users and service providers by optimizing resources. This platform addresses the uncertainty in bandwidth, minimizes running time, and enhances service acceptance rate and scalability in a resource-intensive and dynamic Metaverse system via MetaFunction Resource Orchestrator by leveraging hybrid quantum-classical computing.

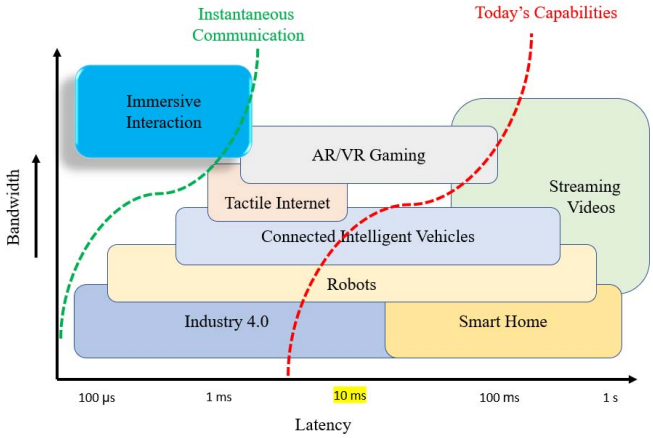


FIGURE 2. Towards quantum computing for metaverse.

As illustrated in Fig. 2, today's network can deliver about 10 milliseconds of latency. However, in order to actualize flawless immersive experience in Metaverse, we need much higher bandwidth and much lower latency. The resource allocation management and optimization schemes directly impact the performance requirements of Metaverse applications. The Metaverse resource allocation infrastructure involves a vast amount of entities, resources, and interactions. Hence, the challenge of solving optimization problems within a reasonable time frame to enhance the interactive virtual experience for users becomes more critical compared to traditional network resource allocation tasks. Along with real-time service dynamics, stochastic aspect of the Metaverse resource allocation requires specialized ILP/MILP optimization models that can not be inherently handled by traditional optimization modelling. Recent studies leverage Integer Linear Programming (ILP), Deep Q Networks (DQN), and heuristics (e.g., simulated annealing) to solve such intricate multi-objective joint optimization problems in communication research avenue [2], [4]. However, several downsides of these classical approaches make these rather unsuitable for Metaverse system that demand real-time characteristics and enormous resources [4]. ILP and classical heuristics often exhibit high running time, while DQN needs frequently prolonged re-training to keep up with the dynamic environment of Metaverse [4], [12]. Quantum computing is believed to unravel the full potentials of Metaverse applications. Hence, the resource allocation and optimization approach in Metaverse foresee to be quantum ready as well [13], [14].

B. CONTRIBUTIONS

Quantum Annealing (QA) [15] has the strength of a magnetic transverse field to avoid getting stuck into local optimal, unlike the classical counterpart. Quantum tunnelling [16] allows waveforms to overcome barriers by exploiting quantum mechanical phenomena. The potentials of quantum fluctuations and quantum dynamics in combination with quantum entanglement and superposition properties can be considered to overcome the shortcomings of classical approaches [14]. In this regard, we propose to employ quantum computing for optimizing resource allocation in the considered Metaverse framework. The major contributions of this paper have been listed as follows:

- *First of all*, QA only accepts QUBO as quantum compliant format for optimization [17], [18]. Thus, we propose the novel QUBO formulation of sharing-based resource allocation in multi-tier architecture to support MetaSlice services. This contribution is of supreme importance by giving general research guidelines on how to encode communication optimization problems as QUBO to be solvable by quantum computing. Furthermore, to better address the uncertainties in bandwidth allocation, we propose another variant of unified hedging QUBO formulation to optimize resources over a broad range of potential scenarios.
- *Secondly*, we propose a hybrid classical-quantum approach to warm start QA by letting classical simulated annealing seed the initial solution. Later on, our proposed warm start quantum annealing (WSQA) improves upon the candidate solutions within real-time constraints by securing optimal/near-optimal resource allocation in Metaverse.
- *Thirdly*, we propose another hybrid quantum-classical approach, named as cold start quantum annealing (CSQA) that begins optimizing with QA and later uses a classical approach to fine-tune the solutions found by QA.
- *Finally*, extensive simulation studies have been considered to present the superiority of hybrid WSQA and CSQA to optimize resources for Metaverse and futuristic eXtended reality (XR) applications by mutually overcoming the shortcomings from both quantum and classical perspectives. Thus, this study can inspire researchers to solve various other futuristic NP-hard problems by using hybrid quantum-classical approaches, even with the limited availability of qubits (e.g., quantum computing resources).

C. ORGANIZATION

The rest of the paper has been structured as follows. Section II summarizes the relevant existing research studies. Section III presents two novel QUBO formulations for MetaSlice resource allocation. Our proposed hybrid classical-quantum approaches, WSQA and CSQA have been described in Sections IV and V, respectively. Next, the simulation results have been discussed in Section VI. Finally, Section VII concludes the paper.

II. RELATED WORKS

Most of the existing literature propose Mixed Integer Linear Programming (MILP) or ILP for resource allocation in Metaverse [2], [4], [19], [20], [21]. Almost all the resource allocation problems for Metaverse system are NP-hard that demand extensive computing time and memory [6].

Oftentimes, the ILP/MILP models are briefly formulated, while ignoring practical constraints, such as, uncertainties in networks [2], [4], [6]. Some of the research focus on delay minimization, while others stress on resource optimization [2], [4], [6].

In this paper, we strive to make a combined effort for simultaneous optimization of resources and QoS. In doing so, the optimization model become even more complex with several crucial constraints [12]. Some of the literature consider DQN and other reinforcement learning approaches that are not appropriate for Metaverse services due to the requirements of frequent re-training initiated by the highly dynamic application/environment nature [12]. A comprehensive research on resource allocation in next-generation broadband wireless access networks explores traditional approaches to support various features such as energy efficiency, user support, task offloading, and mobility [42]. However, this existing study mainly focuses on novel wireless innovation and ignores the aspect of ultra-low computing capacity to support specialized futuristic Metaverse applications [42]. This research aims to bridge this gap by comprehensively investigating resource allocation strategies that cater specifically to the unique demands of rendering in an immersive environment and managing extensive resources in the context of the Metaverse, where traditional wireless innovations fall short. The promise of quantum era can resolve many of the aforementioned shortcomings related to classical computing. Quantum speedup can guarantee both real-time optimization characteristics and meet up the heavy resource demands of the Metaverse system. Coming out of infancy, researchers are putting enormous effort to integrate Metaverse and quantum from optimization perspective. Chong et al. proposed resource optimization in the education Metaverse domain using two-stage stochastic integer programming (SIP) [2]. Yet, the SIP formulation lack real-time service delivery guarantee. Another resource allocation study for synchronizing Metaverse and physical world has been approached with evolutionary mechanism [4]. A recent research attempted to solve the spectrum resource allocation problem as a discrete Markov decision process (MDP) using Quantum Reinforcement Learning [22]. Apart from Metaverse, a plethora of research studies have been conducted using QA on traditional cellular networks, such as, next-generation quantum-enabled multiple-input multiple-output (MIMO) processing [23], routing [24], scheduling/optimization for Internet of Things (IoT) networks [25], [26], resource optimization for Network Function Virtualization (NFV) [27], and so forth. All the quantum inspired optimization research struggle to simulate large-scale NP-hard problems due to the limited availability of freely available qubits (quantum resources) in today's world [27].

With the intention to bridge the research gap, we propose hybrid quantum-classical WSQA and CSQA to simultaneously offer a marginal optimality gap, quantum speedup, and higher service delivery rate (scalability) for a heavily constrained MetaSlice resource allocation problem with limited availability of qubits. Hybrid quantum-classical optimization is envisioned to attract more research studies on resource optimization for Metaverse and typical wireless domains by resolving the downsides from both classical and quantum perspectives.

III. QUANTUM ANNEALING DRIVEN METASLICE OPTIMIZATION

As illustrated in Fig. 1, we consider a MetaSlice resource allocation architecture, where users across Virtual Reality (VR), Augmented Reality (AR), and XR request for a MetaSlice service that is forwarded to MISPs [6]. The MISPs analyze the request and send it to the orchestrator to enable sharing the resources of common functions from different MetaSlices to optimize resources, whenever possible [6]. Moreover, we allow the resource allocation in multi-tier architecture of hosts to handle three categories of Metaverse applications (e.g., real-time, semi real-time, and non-real time). Unlike traditional network functions, MNFs are expected to facilitate immersive experiences, virtual interactions, and seamless connectivity combining physical and virtual space. These MNFs are supposed to handle virtual environment management, avatar interactions, data synchronization, and other configurations unique to metaverse platforms. To support these unique features, we focus on modelling tailored stochastic muti-tier sharing-based resource allocation and the quantum computing realm for delivering ultra-low latency. In this section, we first describe the QUBO formulation to optimize resources and secure QoS under uncertainty. Next, we discuss the conversion of QUBO into Q matrix and embedding into quantum hardware.

A. UNCERTAINTY IN BANDWIDTH CONSUMPTION

The variability in bandwidth outflow can stem from various factors within the host topology, including congestion, traffic dropping behaviors, and more, as mentioned in the work by [12]. Given the highly dynamic nature of MetaSlice service delivery and network attributes, we address the uncertainties in bandwidth generation through a stochastic optimization approach. Let Ω denote the set of all possible scenarios. The probability of occurring a particular scenario $\lambda_e \in \Omega$ is $P(\lambda_e)$. The historical records can be used to estimate $P(\lambda_e)$. The uncertainty of bandwidth consumption based scenarios can be expressed as follows:

$$\lambda_e = \left[O_i^f(\lambda_e), P(\lambda_e)\right],$$

here, $O_i^f(\lambda_e)$ represents the outflow bandwidth generated by MNF $m_i \in \mathcal{M}$ from $\mathcal{S}_f$ under a scenario $\lambda_e \in \Omega$. For example, $O_i^f(\lambda_e) = 1000$ Mbps and $P(\lambda_e) = 0.4$ indicate that there is 40% probability of the bandwidth consumption occurred by MNF $m_i \in \mathcal{M}$ from $\mathcal{S}_f$ to be 1000 Mbps. Although this is just an assumed scenario, there might be diverse potential demands for bandwidth resources.

TABLE 1. Parameters for MetaSlicing framework.

Parameters of Metaverse resource allocation architecture	
Notation	**Description**
$\mathcal{G} = \{\mathcal{H}, \mathcal{E}\}$	Substrate network topology; where $\mathcal{H}$ and $\mathcal{E}$ represent hosts (nodes) and edges, respectively
$\mathcal{T}_{h_j}$	Tier-level of host $h_j \in \mathcal{H}$
$\mathcal{C}_j$	Remaining CPU capacity of host $h_j \in \mathcal{H}$ (in cores)
$\mathcal{R}_j$	Remaining memory capacity of host $h_j \in \mathcal{H}$ (in GBs)
$\mathcal{B}_{jj'}$	Remaining bandwidth linking two hosts (connected) h_j and $h_{j'}$, where $h_j, h_{j'} \in \mathcal{H}$ and $j \neq j'$ (in Mbps)
$\mathcal{D}_{jj'}$	Communication delay between two hosts (nodes) h_j and $h_{j'}$, where $h_j, h_{j'} \in \mathcal{H}$ and $j \neq j'$
MetaSlice Request and MNF Related Parameters	
Notation	**Description**
$\mathcal{M}$	All possible MNF instances $\{m_1, m_2, ..., m_{\mathcal{M}}\}$
$\mathcal{S}_f$	Requested MetaSlice; $\mathcal{S}_f \subseteq \mathcal{M}$
$\lvert\mathcal{S}_f\rvert$	Length of the requested MetaSlice $\mathcal{S}_f$ equal to the number of required MNFs
cpu_i^f	CPU requirement for new MNF instance $m_i \in \mathcal{M}$ deployment of MetaSlice $\mathcal{S}_f$
ram_i^f	RAM requirement for new MNF instance $m_i \in \mathcal{M}$ deployment of MetaSlice $\mathcal{S}_f$
σ_i	Sharable binary flag of MNF $m_i \in \mathcal{M}$
$\mathcal{I}_i^f$, $\mathcal{O}_i^f$	Inflow and outflow of MNF $m_i \in \mathcal{M}$ from $\mathcal{S}_f$
ρ^f	Priority of MetaSlice $\mathcal{S}_f$
Θ^f	QoS threshold of MetaSlice $\mathcal{S}_f$; where $\Theta^f = f(\rho^f)$
Constants, Auxiliary, and Decision Variables	
Notation	**Description**
$x_{i,j}^f$	Binary decision variable for deployment of new MNF instance $m_i \in \mathcal{M}$ from MetaSlice $\mathcal{S}_f$ in host $h_j \in \mathcal{H}$
$z_{i,j}^f$	Binary decision variable for sharing MNF $m_i \in \mathcal{M}$ from $\mathcal{S}_f$ with similar on-boarded MNF at host $h_j \in \mathcal{H}$
$\chi_{i,j}$	Binary input parameter identifying on-boarded similar MNF instances $m_i \in \mathcal{M}$ at host $h_j \in \mathcal{H}$
$f_{i,j}$	Unconsumed sharable (available) flow of on-boarded MNF $m_i \in \mathcal{M}$ at host $h_j \in \mathcal{H}$
η_c, η_r, η_b	Unit costs associated with CPU, RAM, and bandwidth
$P(\lambda_e)$	Probability of the occurrence of a scenario $\lambda_e \in \Omega$; where Ω is set of all possible scenarios
$\varphi_1, \varphi_2, \varphi_3, \ldots$	Penalty co-coefficients of constraints
ϵ_c	Slack variables for transforming inequality to equality

B. QUBO FORMULATION

Ultimately, the QUBO format is formed by adding all the constraints ($\mathcal{C}1$ through $\mathcal{C}9$) described below to the objective function (Eq. (1)). The penalty of each constraint has been represented using φ, sub-scripted by the constraint number. The penalty factors are usually set as large values depending on MetaSlice length and host specifics to avoid infeasible solutions and the accumulation of huge penalties. Finally, we propose a hybrid WSQA approach to optimize the QUBO problem as an alternative to standalone QA with greater scalability benefits. All the parameters for the QUBO model has been presented in Table 1.

The core objective function, as defined in Eq. (1), optimizes the total cost of resource allocation for satisfying MetaSlice application requests. The objective function has been designed in such a way that it inspires leasing resources among MNFs rather than creating new meta instance, whenever possible [6], [27]. The first part of the objective function determines the additional costs for CPU and RAM in case of creating a new MNF. The later part of the objective function calculates the costs of sharing bandwidth resources, considering uncertainties in bandwidth consumption.

$$\min_{x_{i,j}^f, z_{i,j}^f \in \{0,1\}^n} \sum_{i=1}^{|\mathcal{S}_f|} \sum_{h_j \in \mathcal{H}} \left\{ \left(\eta_c \times cpu_i^f + \eta_r \times ram_i^f \right) \times x_{i,j}^f - \mathbb{E}\left[\mathcal{Q}\left(\eta_b, \mathcal{O}_i^f, x_{i,j}^f, z_{i,j}^f, \lambda_e \right) \right] \right\}, \tag{1}$$

$$\text{where,} \quad \mathcal{Q}\left(\eta_b, \mathcal{O}_i^f, x_{i,j}^f, z_{i,j}^f, \lambda_e \right) = \sum_{\lambda_e \in \Omega} P(\lambda_e) \times \sum_{i=1}^{|\mathcal{S}_f|} \sum_{h_j \in \mathcal{H}} \mathcal{O}_i^f \times \eta_b \times \left(x_{i,j}^f + z_{i,j}^f \right) \tag{2}$$

Here, Ω indicates a set of scenarios and is represented using the tuple $\mathcal{Q}(\eta_b, \mathcal{O}_i^f, x_{i,j}^f, z_{i,j}^f, \lambda_e)$. The probability of occurring a scenario $\lambda_e \in \Omega$ is $P(\lambda_e)$. The uncertainties of bandwidth outflow may arise due to several reasons in topology of hosts, such as, congestion, traffic dropping characteristics, and so forth [12]. Since MetaSlice service delivery and network characteristics are claimed to be extremely dynamic in nature, we encounter the uncertainties of bandwidth generation as stochastic optimization approach.

$$\mathcal{C}1 : \varphi_1 \times \left(\sum_{h_j \in \mathcal{H}} \left(x_{i,j}^f + z_{i,j}^f \right) - 1 \right)^2, \forall_{i \in [1-|\mathcal{S}_f|]} \tag{3}$$

Constraint $\mathcal{C}1$ ensures the allocation (sharing/deployment) of a MNF m_i from MetaSlice $\mathcal{S}_f$ into a single host $h_j \in \mathcal{H}$. In case the total allocation ($\sum_{h_j \in \mathcal{H}} (x_{i,j}^f + z_{i,j}^f)$) of any MNF belonging to a MetaSlice over all hosts exceeds 1, a penalty will be generated and eventually added to the objective function in Eq. (1). The non-zero penalty is amplified by squaring the term and multiplying with a factor φ_1. Thus, the optimizer will strive to avoid such large penalties by satisfying MNF allocation to exactly one host.

$$\mathcal{C}2 : \varphi_2 \times \left(\sum_{h_j \in \mathcal{H}} \left(x_{i,j}^f + \chi_{i,j} \times \sigma_i \times z_{i,j}^f \right) - 1 \right)^2, \forall_{i \in [1-|\mathcal{S}_f|]} \tag{4}$$

The constraint $\mathcal{C}2$ is imposed upon MNFs to be either deployed as new instance or lease the resources from already on-boarded/running MNFs of other MetaSlice applications. However, a similar MNF with permissible sharable property has to be present in previously deployed MetaSlices to activate leasing/sharing resources instead of creating a new instance. Upon the failure of any above-mentioned condition,

the constraint term will incur a large penalty, unlike zero, which is the expected ideal scenario.

$$\mathcal{C}3 : \varphi_3 \times \left(\mathcal{I}_i^f \times z_{i,j}^f - f_{i,j} + \epsilon_3\right)^2, \forall_{i\in[1-|\mathcal{S}_f|]} \tag{5}$$

Next, constraint $\mathcal{C}3$ expresses that the MNF to be shared (already running instance) must have sufficient available/unconsumed flow ($f_{i,j}$) as per the requirements ($\mathcal{I}_i^f$) of requested MNF. Traditionally, this constraint can be thought of as an inequality. However, to convert it into equality fitted for QUBO format, a slack variable (ϵ_3) has been introduced. The slack variable represents the non-binding point from which candidate/potential solutions are allowed [27]. The rest of the constraints due to the inequality nature also have respective slack variables linked to those.

$$\mathcal{C}4 : \varphi_4 \times \left(\sum_{i=1}^{|\mathcal{S}_f|} cpu_i^f \times x_{i,j}^f - \mathcal{C}_j + \epsilon_4\right)^2, \forall_{h_j\in\mathcal{H}} \tag{6}$$

$$\mathcal{C}5 : \varphi_5 \times \left(\sum_{i=1}^{|\mathcal{S}_f|} ram_i^f \times x_{i,j}^f - \mathcal{R}_j + \epsilon_5\right)^2, \forall_{h_j\in\mathcal{H}} \tag{7}$$

Constraints $\mathcal{C}4$ and $\mathcal{C}5$ validate whether the host devices have enough CPU and memory capacity to deliver a MetaSlice application request. The unavailability of sufficient computing and memory resources will reject MetaSlice service requests by incurring huge penalties to the optimization solver.

$$\mathcal{C}6 : \varphi_6 \times \left(\left(x_{i,j}^f + z_{i,j}^f\right) - \sum_{\{h_j,h_{j'}\}\in\mathcal{E}} \left(x_{i+1,j'}^f + z_{i+1,j'}^f\right) + \epsilon_6\right)^2, \forall_{h_j,h_{j';j'\neq j}\in\mathcal{H}}\forall_{i\in[1-|\mathcal{S}_f|]} \tag{8}$$

Subsequently, constraint $\mathcal{C}6$ is responsible for allocating two ordered MNFs (m_i and m_{i+1}) from the requested MetaSlice $\mathcal{S}_f$ into two directly connected (adjacent) hosts (nodes) in the substrate network/multi-tier topology. Hence, the traffic flow routing through the meta instances is optimized as a by-product of this constraint. It is noteworthy that the slack variable ϵ_6 of this specific constraint can be binary due to only involving decision variables.

$$\mathcal{C}7 : \varphi_7 \times \left(\sum_{h_j\in\mathcal{H}} \sum_{h_{j';j'\neq j}\in\mathcal{H}} \mathcal{O}_i^f \times \left(x_{i,j}^f + z_{i,j}^f\right) \times \left(x_{i+1,j'}^f + z_{i+1,j'}^f\right) - \mathcal{B}_{jj'} + \epsilon_7\right)^2, \forall_{i\in[1-(|\mathcal{S}_f|-1)]} \tag{9}$$

Constraint $\mathcal{C}7$ verifies that two host devices have enough bandwidth capacity ($\mathcal{B}_{jj'}$) to handle the outflow ($\mathcal{O}_i^f$) generated by the allocation of requested MNF instance. In other words, the available bandwidth must be as much as possible equal to the summation of outflow and slack variable for preventing constraint violation penalties.

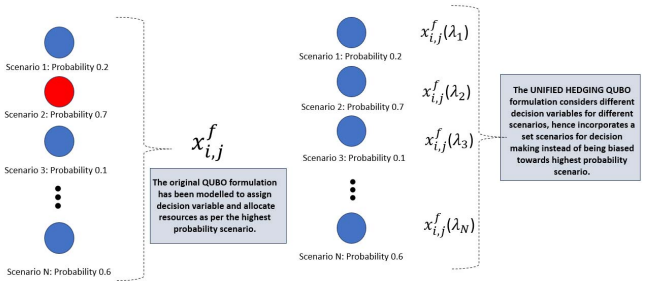


FIGURE 3. Original versus unified hedging QUBO formulation.

$$\mathcal{C}8 : \varphi_8 \times \left(\sum_{i=1}^{|\mathcal{S}_f|-1} \sum_{h_j\in\mathcal{H}} \sum_{h_{j';j'\neq j}\in\mathcal{H}} \mathcal{D}_{jj'} \times \left(x_{i,j}^f + z_{i,j}^f\right) \times \left(x_{i+1,j'}^f + z_{i+1,j'}^f\right) - \Theta^f + \epsilon_8\right)^2 \tag{10}$$

Similarly, constraint $\mathcal{C}8$ prevents the total delay of requested MetaSlice application to exceed beyond a pre-specified threshold Θ^f. This delay limit is usually set by the service providers as per the priority of MetaSlice (ρ^f) and Service Level Agreement (SLA) [27]. The slack variable ϵ_8 allows the cumulative latency of the requested MNF instances to be under QoS threshold of respective MetaSlice service. Since Metaverse services intend to support context-aware latency tolerance, this quadratic constraint is considered quite significant for Metaverse framework [28].

$$\mathcal{C}9 : \varphi_9 \times \left(\left(x_{i,j}^f + z_{i,j}^f\right) \times \rho^f - \mathcal{T}_{h_j} + \epsilon_9\right)^2, \forall_{h_j\in\mathcal{H}}\forall_{i\in[1-|\mathcal{S}_f|]} \tag{11}$$

Finally, constraint $\mathcal{C}9$ ensures that MNFs are placed and shared across multi-tier architecture depending upon the requirements/priority of requested MetaSlice. Thus, the real-time latency-sensitive MNFs are placed on the edge/outer tier hosts, while others are allocated to inner-tiered hosts with comparatively higher latency. Overall, this constraint verifies the allocation of MNFs into hosts situated at equal or higher tiers compared to priority levels of MetaSlice [6].

C. UNIFIED HEDGING QUBO FORMULATION

Unified Hedging is a technique that uses single strategy to optimize over multiple scenarios rather than prioritizing a single scenario [29], as shown in Fig. 3. In the previous subsection, we design the objective function to prioritize the scenario with the highest likelihood and allocate bandwidth accordingly. However, to achieve more conservative decision-making and have better security against potential uncertainty, we modify the objective function in such a way that it explores a diverse range of scenarios and optimizes to minimize the overall worst-case scenario outcome at the

price of risking best-case scenario outcome [30]. For this purpose, the objective function is redefined as follows:

$$\min_{x^f_{i,j}(\lambda_e), z^f_{i,j}(\lambda_e)\in\{0,1\}^n} \sum_{i=1}^{|\mathcal{S}_f|} \sum_{h_j\in\mathcal{H}} \left\{ \left(\eta_c \times cpu^f_i + \eta_r \times ram^f_i\right) \times x^f_{i,j}(\lambda_e) + \mathbb{E}\left[\mathcal{Q}\left(\eta_b, \mathcal{O}^f_i, x^f_{i,j}(\lambda_e), z^f_{i,j}(\lambda_e), \lambda_e\right)\right]\right\}, \quad (12)$$

$$\text{where, } \mathcal{Q}\left(\eta_b, \mathcal{O}^f_i, x^f_{i,j}(\lambda_e), z^f_{i,j}(\lambda_e), \lambda_e\right) = \sum_{\lambda_e\in\Omega} P(\lambda_e) \times \beta_e \times \sum_{i=1}^{|\mathcal{S}_f|} \sum_{h_j\in\mathcal{H}} \mathcal{O}^f_i \times \eta_b \times \left(x^f_{i,j}(\lambda_e) + z^f_{i,j}(\lambda_e)\right) \quad (13)$$

In the above mentioned, Eq. (12), we define decision variables of deploying/sharing MNF instances for every potential scenario $\lambda_e \in \Omega$ so that the model is able to evaluate the system cost of each scenario. In the equation Eq. (13), we introduce β_e that tunes the degree of risk tolerance/conservatism of the solution affordable by the service. As an example, some of the Metaverse services such as observing the virtual environment require less bandwidth, whereas MNFs required to interact with the virtual environment demand much higher bandwidth. Thus, MSPs can tune β_e as per the desirable trade-off between system cost and MetaSlice application risk tolerance limit. The summation of β_e overall all the scenarios should be 1. It is noteworthy that for this model we need to reform the constraints $\mathcal{C}1 - \mathcal{C}9$ by increasing the dimension of the decision variables for each scenario in the QUBO model to make unified hedging decisions favourable across multiple scenarios. Following is another additional constraint of the unified hedging QUBO model:

$$\mathcal{C}10 : \varphi_{10} \times \left(\sum_{\lambda_e\in\Omega} \left(x^f_{i,j}(\lambda_e) + z^f_{i,j}(\lambda_e)\right) - 1\right)^2, \quad \forall_{h_j\in\mathcal{H}}\forall_{i\in[1-|\mathcal{S}_f|]} \quad (14)$$

This additional QUBO constraint in Eq. (14) ensures that the model takes MetaSlice sharing/redeployment decision in favour of the particular uncertain scenario that is able to somewhat optimize across a broad range of situations instead of tailoring to a single potential scenario.

In the worst-case, the computational complexity of both QUBO models is exponential. One of the main driving factors in the computational complexity of these models is the number of potential scenarios affecting the size of the optimization problem significantly. As the number of potential scenarios increases, the optimization problem becomes more computationally challenging to solve.

D. QUANTUM ANNEALING

Quantum Annealing leverages quantum mechanics to explore multiple solutions simultaneously and find optimal or near-optimal solutions more efficiently, unlike the classical counterpart. Typically, quantum annealing involves three major parts that are encoding the problem, tunneling phase, and finding ground state or solution. The first step is to encode the problem as quantum compliant format which is either QUBO or Ising model. The entire formulation creates a energy landscape in quantum system through the qubits. Each solution of the optimization problem refers to different energy states/levels. Once the formulation is embedded into quantum hardware, the tunneling effect allows high energy barrier penetration through quantum energy states which allows the exploration of multiple solutions parallel and faster, compared to classical computing. Finally, the goal is for the quantum system to reach the ground state, which corresponds to the optimal or near-optimal solution of the original optimization problem. The annealing process is designed to drive the system to this state.

E. Q-MATRIX COMPOSITION

The QUBO formulation is the first step of quantum annealing. As per the QUBO formulation, the optimizer model should include only one objective function with no apparent constraints [14]. Hence, all the quadratic constraints, such as equations $\mathcal{C}1$ through $\mathcal{C}9$, are added to the core objective functions of Eq. (1). For the unified hedging QUBO formulation, we add constraints $\mathcal{C}1$ through $\mathcal{C}10$ to the core objective function represented by Eq. (12). Then, an upper triangular matrix is constructed using the linear and quadratic coefficients of the objective function. The linear and quadratic terms are respectively set as diagonal and off-diagonal real numbers of the upper triangular matrix, later called as Q matrix. For the sake of simplicity, we can consider that the quantum computer optimizes as follows:

$$\min_{x\in\{0,1\}^n} x^T Q x;$$

where, Q is the upper triangular matrix, known as Q matrix and x is the vector of binary decision variables. Another alternative is to transform the QUBO model into an Ising model [14]. In such cases, the decision variables are represented using dipole moments of atomic spins (+1 or -1) [14]. Moreover, the linear and quadratic coefficients are considered as biases and coupling strengths for QA to distinguish optimal/near-optimal solutions.

Our research problem includes involves optimization with two sets of decision variables. Considering these decision vectors represented by x and y, the general QUBO formulation can be re-written as follows:

$$\min_{x,y\in\{0,1\}^n} \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}x_i y_j + \sum_{i=1}^{n} Q_{ii}x_i + \sum_{j=1}^{n} Q_{jj}y_j;$$

here, n is the number of decision variables in each of the decision vectors. Q_{ii} and Q_{jj} are the coefficients of the quadratic terms. Q_{ij} are the coefficients for the cross-product terms between variables x_i and y_j. This extended generic QUBO format allows interaction between the decision variables from both vectors.

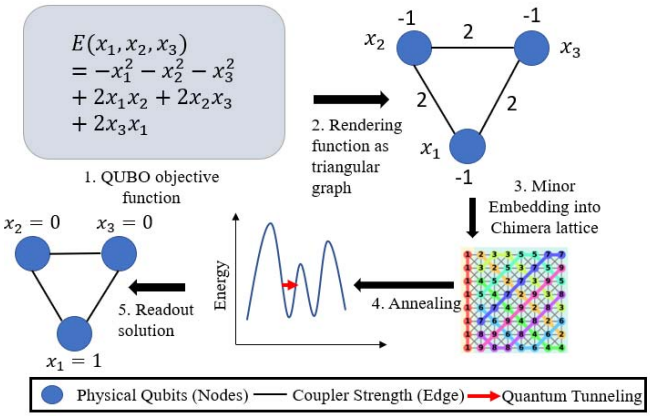


FIGURE 4. Typical workflow of quantum annealer.

F. EMBEDDING QUBO INTO QUANTUM HARDWARE

Once the QUBO reformulation is completed, the next step is to transform the objective function, also known as, energy function into Quantum Processing Unit (QPU). Despite the existence of various topologies, D-wave particularly uses a Chimera Graph Lattice Architecture [27]. In this case, the QPU is constructed using qubits that are partially connected [14].

Fig. 4 portrays the overall workflow of quantum annealing mechanism. The nodes of chimera graph denote the decision variables of the optimization problem. Edges/couple strengths are represented using the respective quadratic coefficients. The bias of the nodes are set according to the linear coefficients. The biases control the magnetic field acting upon the qubits, while the coupler strength determines the interaction power among qubits. Eventually, the standard QUBO input for QA is mapped into physical qubits of the quantum hardware. This process is called minor embedding [14]. Initially, the qubits are entangled with equal superposition of all possible states. The annealer continues to sample the solutions from many candidate solutions, as defined by the energy landscape (QUBO objective function). The QA strives to find the minimum/low energy state (optimal/near-optimal solutions) through quantum tunneling strategy. At the end of the annealing phase, the qubit spins/decision variable values are externally read out and stored using reverse minor embedding process.

IV. WARM START QUANTUM ANNEALING

The technique of generating initial feasible but not so good solutions is known as warm start [27]. Conventional optimization approaches widely employ warm start to control the combinatorial explosion and computational complexity of NP-hard problems [27]. Motivated from the prior evidences, we propose warm start quantum annealing (WSQA) to secure manifold computing outcomes [27]. The intended aftermaths are to increase the scalability, reduce number of required qubits, and decrease run-time (time to optimal/near-optimal) for the quantum-inspired MetaSlice resource allocation problem. In the following subsections, we propose two different variants of WSQA considering two different classical approaches before initiating QA.

Algorithm 1: MetaSlice Service Allocation by Leveraging WSQA

Input: $\mathcal{G}, \mathcal{S}_f, \mathcal{C}_j, \mathcal{R}_j, \mathcal{D}_{jj'}, \mathcal{B}_{jj'}, \sigma_i, \Theta^f, \chi_{i,j}, f_{i,j}, cpu_i^f, ram_i^f, \mathcal{I}_i^f, \rho^f, \mathcal{O}_i^f, \mathcal{T}_{h_j}, \eta_c, \eta_r, \eta_b, \varphi_c, \epsilon_c, max\_epoch$
Output: $x_{i,j}^f, z_{i,j}^f$

1 $temp \leftarrow temp_0$
2 $state_k \leftarrow state_0$
3 **for** $t = 1$ *to* max_epoch **do**
4 $\quad state_{k+1} \leftarrow neighbour(state_0)$
5 $\quad \Delta E \leftarrow energy(state_{k+1}) - energy(state_k)$
6 $\quad r \leftarrow rand(0, 1)$
7 $\quad$ **if** $min(1, \ e^{-\frac{\Delta E}{temp}}) \geq r$ **then**
8 $\quad\quad state_{k+1} \leftarrow state_k$
9 $\quad temp \leftarrow \alpha^{(t-1)} \times temp_0$
10 Define QUBO and formulate Q matrix
11 Translation of QUBO into logical graph and perform minor embedding into chimera lattice
12 Sample solutions from the energy landscape (search space) through quantum tunneling
13 Readout the sampled solutions by reversing minor embedding

A. WSQA VARIANT WITH SIMULATED ANNEALING

Primarily, we let the classical simulated annealing to construct a valid initial solution. Eventually, the quantum annealer improves on the previously generated solution. Algorithm 1 presents our proposed WSQA methodology to tackle the optimal resource allocation for MetaSlice service delivery. The algorithm takes all relevant MetaSlice request parameters as input and outputs whether a MNF instance should be created ($x_{i,j}^f$) or shared ($z_{i,j}^f$) with the already running similar MNF for resource optimization. Analogous to the physical behaviour of cooling molten metal, we construct an initial solution using the classical simulated annealing heuristic (line number $1 - 9$). First of all, the initial *temp* variable resembling to temperature is initialized with a very high value in line number 1. Next, a candidate solution ($state_0$) is generated and stored as $state_k$ in line number 2. For every single iteration, a new candidate solution is generated by exploiting the neighbourhood search space of the previous solution (line number 4). Subsequently, we compare the quality of both old and new solutions by taking the difference of the energy/objective function value (ΔE) associated with each of the solutions in line number 5. If we find out the quality of the newly generated solution is better than the old one, we accept the candidate solution and continue to generate yet another solution in the next iteration. Otherwise, the newly generated solution of worse

quality may be accepted with a probability of $e^{-\frac{\Delta E}{temp}}$ for diversifying search space. In this case, we generate a random number r between 0 and 1 in line number 6. Then, we check if the acceptance probability is greater than the randomly generated number to proceed with the new deteriorated candidate solution.

Finally, we reformulate the Q matrix by leveraging the initial feasible solution generated by the classical simulated annealing. Then, the QUBO is translated into a triangular logical graph and embedded into the Chimera lattice graph. The quantum annealer continually samples the candidate solutions and strives to find the qubits that produce lowest energy state (optimal solution). Towards the termination of the annealing phase, the qubit status are read out into classical decision variables by reversing the minor embedding procedure. The prototypical stages of quantum annealing have been elaborately discussed in Section III-F.

B. WSQA VARIANT WITH GENETIC PROGRAMMING

We propose another variant of WSQA, where we apply Genetic Algorithm (GA) [31] as a preprocessing step of QA. GA are well-known heuristic approach for approximating solutions to challenging optimization problems [32], [33], [34]. Combining QA with GA is an interesting hybrid approach that can harness the potential strengths of both quantum and classical computing paradigms. The general outline of a GA can be outlined as follows:

- *Initialization:* The algorithm usually starts with a population of randomly generated individual solutions or chromosomes.
- *Fitness Evaluation:* Next, we have to define the fitness function in such a way that it quantifies the quality of each solution/individual in terms of optimizing/solving the problem in hand. In our case, we have directly considered the fitness evaluator as the objective function value of the MetaSlice service allocation optimization problem.
- *Selection:* In this process, some of the individuals are selected based on their fitness score for reproducing offspring solutions. The higher the fitness score or the minimum objective function value, the more likely the solutions will be selected for the reproduction of newer solutions. This concept emulates the *"survival of the fittest"* mechanism.
- *Crossover:* Pairs of individual solutions swap genetic information/part of their solutions to recombine and generate new solutions.
- *Mutation:* Next, some of the individuals undergo random changes/mutations to introduce exploration in the search space. This diversification often becomes necessary to prevent the solutions from being stuck into local minima or homogeneous regions.
- *Replacement:* In this step, the fittest individuals from the combined set of parents and offspring (recombined new solution pool) replace the existing solutions from the previous generation.
- *Termination & Solution Extraction:* The selection, crossover, mutation, and replacement processes are repeated for a fixed number of generations or until a desirable solution is found.

C. COMPUTATIONAL COMPLEXITY

Deriving the computational complexity of the quantum annealing is straightforward. It depends on several factors, such as quantum hardware architecture, interaction of qubits, connectivity, and so forth. Qualitatively, the computational complexity of QA is $O(e^{\sqrt{N}})$, where N is the size of the Q-matrix. Next, if the classical phase is chosen as SA, the additional computational complexity is $O(max\_epoch)$. The time complexity is generally $(max\_epoch \times population\_size)$, where $population\_size$ is the number of individuals in the population. The computational complexity of the overall WSQA algorithm depends on the running time length of the quantum and classical phases. It is noteworthy that the concepts of time complexity and time to solution are not always directly aligned or interchangeable.

V. COLD START QUANTUM ANNEALER

In this section, we explore the strategy of post-processing QA with classical SA. Thus, the cold start quantum annealer is expected to work exactly opposite to the WSQA mechanism. In the realm of computing, The term "cold start" refers to the state when the system starts/reboots beyond normal operation mode [36], [37], [38]. As the MetaSlice deployment system in this proposed approach starts optimization by applying quantum computing first and then makes a transition towards the classical approach, we define this mechanism as the Cold Start Quantum Annealer.

Acknowledging challenges of the MetaSlice deployment problem, such as the extensive search space and need for real-time services, we employ QA to effectively narrow down the search space very quickly and generate an approximate solution. Following the quantum phase, we let classical optimization algorithms to fine-tune the approximate solutions found by QA. The detailed algorithmic steps of CSQA have been mentioned in Algorithm 2.

The algorithm starts by defining our proposed QUBO formulation and generating the Q-matrix. Then, the quantum annealer samples through the energy landscape (search space) and finds an approximate solution. The solution found out by QA is stored as a starting solution in the variable $state_0$. Considering this initial feasible solution, we either choose SA or GA as the classical optimization algorithm and further improve upon the quantum solution (line number 5-19). The process of classical SA and GA has already been elaboratively discussed in Sections IV-A and IV-B, respectively.

Similar to WSQA, the computational complexity of CSQA depends on the running time length of quantum and classical phases. The individual computational complexity

Algorithm 2: MetaSlice Service Allocation by Leveraging CSQA

Input: $\mathcal{G}, S_f, \mathcal{C}_j, \mathcal{R}_j, \mathcal{D}_{jj'}, \mathcal{B}_{jj'}, \sigma_i, \Theta^f, \chi_{i,j}, f_{i,j}, cpu_i^f, ram_i^f, \mathcal{I}_i^f, \rho^f, \mathcal{O}_i^f, \mathcal{T}_{h_j}, \eta_c, \eta_r, \eta_b, \varphi_c, \epsilon_c, max\_epoch$

Output: $x_{i,j}^f, z_{i,j}^f$

Define QUBO and formulate Q matrix
Translation of QUBO into logical graph and perform minor embedding into chimera lattice
Sample solutions from the energy landscape (search space) through quantum tunneling
$state_0 \leftarrow$ Readout the sampled solutions by reversing minor embedding
if *classical_phase* == *"SA"* **then**
  $temp \leftarrow temp_0$
  $state_k \leftarrow state_0$
  for $t = 1$ *to* max_epoch **do**
    $state_{k+1} \leftarrow neighbour(state_0)$
    $\Delta E \leftarrow energy(state_{k+1}) - energy(state_k)$
    $r \leftarrow rand(0, 1)$
    if $min(1, \ e^{-\frac{\Delta E}{temp}}) \geq r$ **then**
      $state_{k+1} \leftarrow state_k$
    $temp \leftarrow \alpha^{(t-1)} \times temp_0$
if *classical_phase* == *"GA"* **then**
  Initialize population
  Evaluate fitness score of individuals
  Apply selection, crossover, and mutation to evolve population
  Replace population, terminate and extract solutions

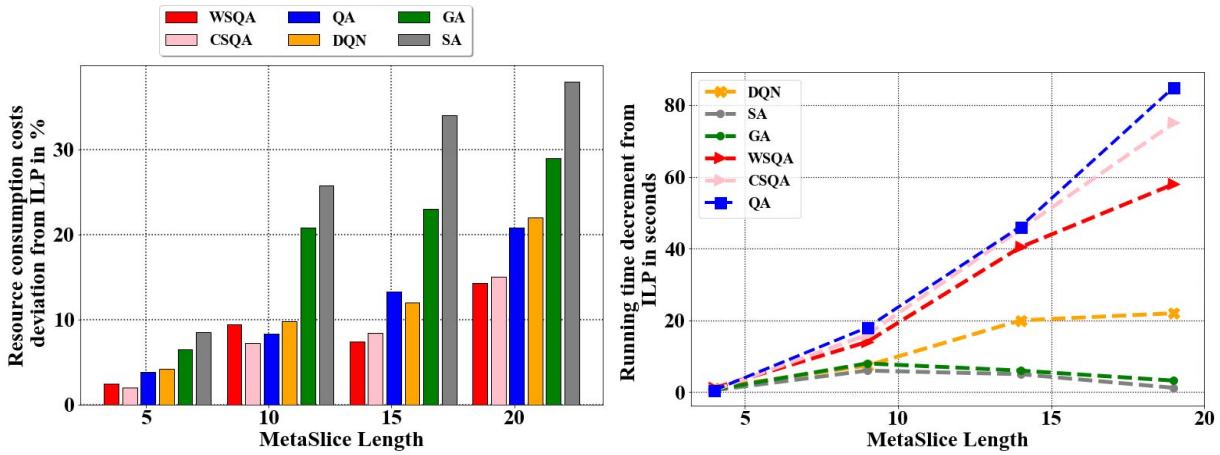
of the quantum and classical phases have been discussed in Section IV-C. The main effect of WSQA and CSQA is not differentiated by the computational complexity, rather noticed by the scalability of the optimization problem in hand.

VI. SIMULATION RESULTS

In order to demonstrate the performance improvement by quantum approaches, we consider deep Q-networks (DQN), ILP (classical optimal), and simulated annealing (SA) as baselines. We simulate the classical approaches on DELL ALIENWARE m15 R3 machine of Intel core i7-10750H CPU @2.6 GHz equipped with 16 GB RAM and Windows 10 Home. The exact classical optimal results of ILP has been retrieved using GUROBI optimization solver [39]. In order to train DQN, we leverage TensorFlow and several other python packages, while following the model design from literature [12]. We create the multi-tier host architecture for MetaSlice service deployment using NetworkX [40]. Experiments of our proposed quantum approaches (WSQA and QA) for MNF resource allocation have been performed by leveraging D-wave's leap hybrid solver [14]. We establish a simulation environment for substrate IoT networks utilizing NetworkX, incorporating a dynamic node count ranging from 700 to 1000. The resultant topology is characterized by randomly assigned resource capacities spanning 8 to 64 CPU cores, 16 to 128 GB RAM, and 100 to 1000 Mbps bandwidth. The QoS threshold is calculated based on the number of MNFs and their aforementioned bandwidth ranges. Additionally, propagation delays are chosen at random from 50 to 1000 ms. MetaSlice lengths are then set to random values within the range of 5 to 25. Moreover, we define the resource and flow requirements for MNFs of MetaSlices. These requirements are drawn randomly from predefined ranges, encompassing 2 to 8 cores for CPU, 4 to 16 GB for RAM, and determining the maximum flow as a function of CPU and RAM. The inflow is established as a range between 0.15 times the maximum flow and the maximum flow in Mbps. In instances where a MNF exhibits dropping characteristics, the outflow is determined to be between 0.4 times the inflow and the inflow itself. Otherwise, it mirrors the inflow precisely. Unit costs for CPU, RAM, and bandwidth are standardized at 2.5, 1.7, and 2, respectively. For the genetic algorithm in WSQA variant, we consider crossover probability as 0.8, a random mutation probability of 0.2, and the Roulette Wheel selection mechanism. The probability of occurring a scenario varies randomly from 0.1-0.9. The summation of probabilities associated with all the scenarios have to be less than or to 1. All the system parameters have been adopted from literature [12], [27]. The results have been averaged over 20 different simulation runs.

A. CLASSICAL VERSUS QUANTUM PERFORMANCE COMPARISON

Several key performance indicators (KPIs) have been taken into consideration to justify the superiority of quantum approaches over classical ones. We consider the resource consumption costs deviation from ILP as the first KPI to express how far a solution quality is from ILP (optimal). The closer an approach can generate solutions near optimal ones in terms of minimizing the objective function, the smaller is the optimality gap. Fig. 5(a) illustrates the total resource cost consumption deviation from ILP for classical (e.g., SA, GA, DQN) and quantum approaches (e.g., QA, WSQA, CSQA) based on varying MetaSlice length. The classical variant of GA and SA on hybrid quantum-classical WSQA and CSQA approaches have negligible impact. Hence, we demonstrate the results of WSQA and CSQA averaging over two different classical variants SA and GA to distinguish the effects of pre-processing and post-processing quantum annealing with classical approaches. SA is the worst performing algorithm by exhibiting higher optimality gap/resource consumption regardless of MetaSlice length. GA performs equally well as SA. DQN stands out as the best MetaSlice deployment/sharing scheme among standalone classical approaches. DQN is the most used algorithm to address the dynamics and uncertainty of the system environment [6]. The choice of DQN was primarily motivated by its ability to learn and adapt in the Metaverse environment where bandwidth resource



(a) Optimality gap comparison (the lower, the better)

(b) Running time comparison (the higher, the better)

FIGURE 5. Comparison of classical, quantum, and hybrid approaches for QUBO formulation with varying MetaSlice length.

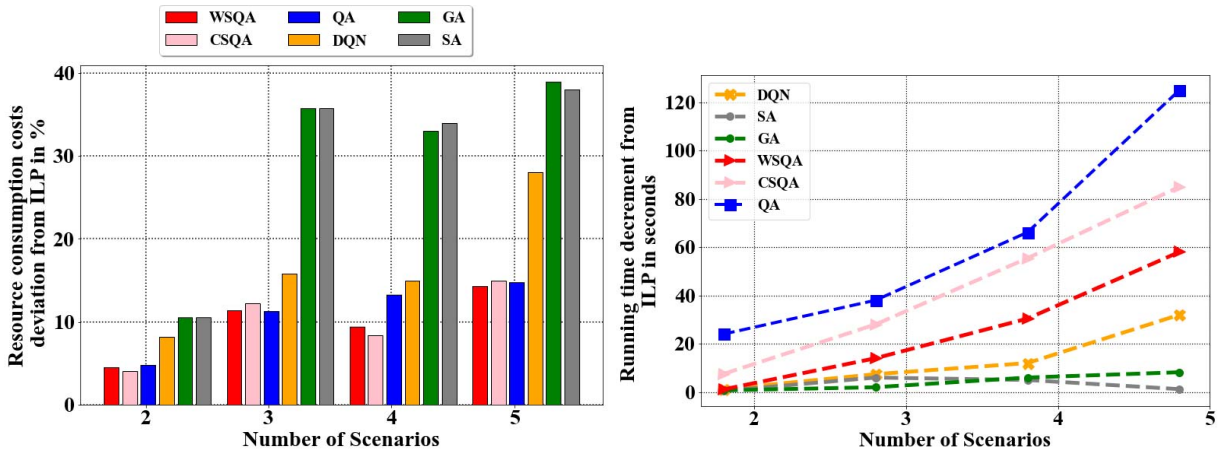
parameters are uncertain. It is noteworthy that in the training phase of the DQN model, we consider the bandwidth allocation of the system being unknown and let the model learn based on Q-value prediction. The DQN is a robust baseline approach to learn policies, particularly effective in stochastic Metaverse resource environments with unknown bandwidth, even where the state (MetaSlice system input parameters) and action spaces (deployment/optimizatiion decisions) are large and complex. However, the stochastic nature of resource allocation in Metaverse applications enforce frequent training for DQN models and the training time takes nearly 10 hours to converge for this model. Due to the aforementioned shortcomings, DQN becomes ineligible to support real-time Metaverse resource allocation problems. Although the quantum and hybrid approaches (QA, WSQA, and CSQA) perform really close for smaller MetaSlice lengths, there is a sudden variability in optimization gap from moderate length of MetaSlice. The hybrid quantum-classical WSQA and CSQA minimize up to 20% resource consumption costs compared to standalone QA. The apparently small percentages of resource costs optimization make a significant impact for Metaverse infrastructure [2], [27].

Fig. 5(b) demonstrates another KPI on how quickly an approach can extend its services to Metaverse users. This metric has been used to evaluate the trade-off between sacrifice in optimality and savings in terms of running time. With the increasing length of MetaSlice, SA and GA continue to perform worse and can not save much time compared to ILP. DQN's inference time is quite better, yet not as close as quantum/hybrid approaches. In our study, DQN requires some additional time to check the feasibility of the solutions, including the computational effort of checking if all the constraints have been satisfied. Hence, we have included the feasibility checking time apart from inference time to calculate the total running time of DQN. This metric reflects the reduction in computational time achieved by DQN or any other model compared to the optimal ILP solution. Specifically, it quantifies the efficiency gained by DQN for inference in terms of the time saved when compared to the computationally expensive ILP approach that always guarantees optimal (best) solutions. Although hybrid WSQA and CSQA are relatively slower compared to standalone QA, we must contemplate the performance improvement by these hybrid approaches. Among the hybrid methods introduced in this paper, CSQA outperforms WSQA in terms of running time. The trade-off between resource costs optimization and running time reduction depends on the requirements and nature of MetaSlice. The real-time Metaverse applications must opt-in for QA by sacrificing some resource costs optimization, since standalone QA saves the highest running time, as shown in Fig. 5(b). On the contrary, the resource-intensive and somewhat non-delay sensitive Metaverse applications should choose hybrid WSQA/CSQA for saving on extra resource costs as much as possible.

Fig. 6(a) and Fig. 6(b) report similar trends in performance evaluation for the unified hedging QUBO formulation. In this case, we vary the number of potential scenarios that determine the problem complexity. Although the performance does not vary much from the original QUBO formulation, Fig. 6(a) demonstrates overall higher optimality gap compared to Fig. 5(a) as the unified hedging QUBO formulation intends to optimize across multiple scenario by sacrificing the outcome for best case scenario. Furthermore, due to the increasing number of potential scenarios, the unified hedging QUBO formulation becomes for computationally complex to solve. Thus, Fig. 6(b) illustrates overall higher running time complexity for all quantum, classical or hybrid strategies with increasing number of scenarios.

B. METASLICE DEPLOYMENT ON MULTI-TIER ARCHITECTURE VERIFICATION

Next, Fig. 7 illustrates the validation of constraint $\mathcal{C}9$ imposed upon the multi-tier architecture of MetaSlice framework. It can be observed that the time-critical MetaSlices are deployed or shared within the edge hosts that



(a) Optimality gap comparison (the lower, the better)

(b) Running time comparison (the higher, the better)

FIGURE 6. Comparison of classical, quantum, and hybrid approaches for Unified Hedging QUBO formulation.

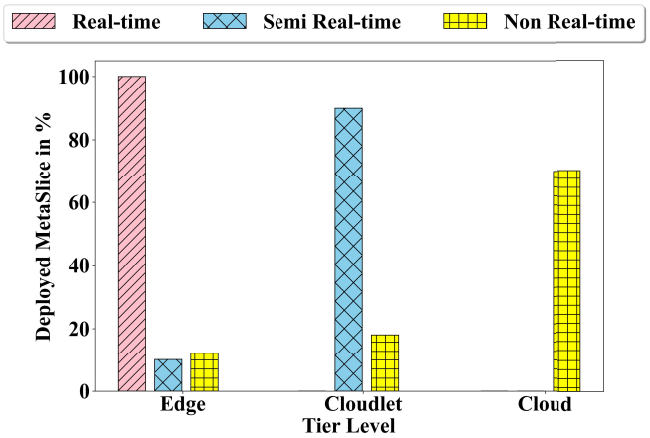


FIGURE 7. MetaSlice deployment on multi-tier architecture.

are in closer proximity to users for seamless service delivery. Similarly, the semi-critical MetaSlice services are allocated mostly on cloudlets (tier-2) hosts, while the non real-time MNFs are placed on the cloud data centers at the core of the Internet. This allocation strategy remains somewhat similar regardless of any undertaken methodology.

C. QUBIT SCALABILITY ANALYSIS

In Fig. 8, we interpolate QUBO objective function/energy landscape for varying MetaSlice length of 5, 10, and 15. Fig. 8(a) suggests that with the increasing number of qubits the energy value decreases, implying that the quantum annealer is capable of accessing higher-quality near-optimal solutions. However, Fig. 8(b) and Fig. 8(c) suggest that as the MetaSlice length increases, higher energy spectrum (lower quality solutions) are more prevalent. Therefore, to encode complex QUBO formulations more efficiently and attain optimal solutions within low energy spectrum in quantum annealer machines, more qubits are needed.

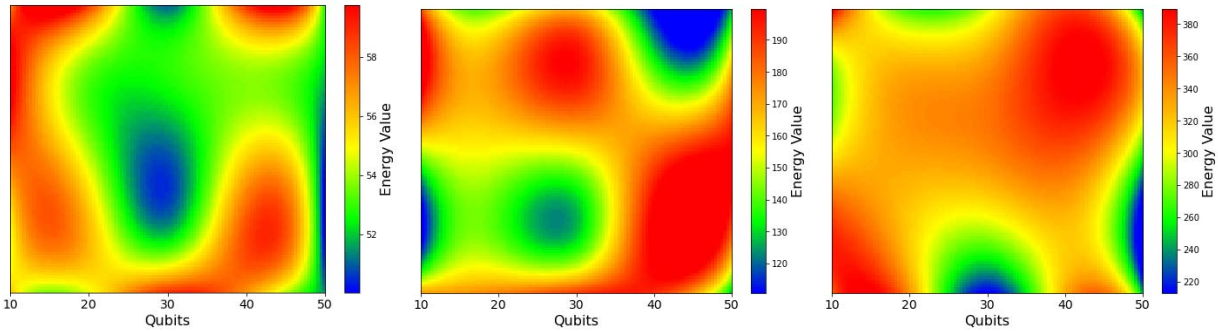
Since there is shortage of physical qubits (quantum resources) for computing in today's world [41], we make an effort to analyze the scalability of quantum methodologies to solve this MetaSlice allocation based combinatorial optimization problem. For this purpose, we study the effect on the Metaverse system caused by multi-tier topology and connectivity probability of nodes (hosts). Thus, we simulate three different cases for sampling available number of hosts for MetaSlice allocation, as shown in Fig. 9. The three different cases refer to various substrate topology sizes, such as, small (10-50 hosts), medium (30-90 hosts), and large (60 -120 hosts).

Next, we configure various connectivity probabilities for each of the aforementioned cases. The probability of two hosts being connected using an edge in the multi-tier graph is termed as connectivity probability in this paper. The intuition for carrying out this experiment is to determine the scalability (Metaverse service acceptance) rate for both of the quantum approaches. This Metaverse service acceptance rate is considered as another KPI to highlight the capability of solving relatively large-scale problem sizes, considering the limited availability of qubits.

Table 2 refers that WSQA can increase the service acceptance rate by up to 30% in comparison with standalone QA. Since the warm start approach shrinks the Q matrix size and required number of qubits, the WSQA approach performs better at solving larger MetaSlice allocation problem instances. On the other hand, CSQA, directly starting with QA needs to manipulate a relatively larger Q-matrix and generate lower scalability rate, almost as comparable as standalone QA. One intriguing observation is that using GA as the classical algorithm instead of using SA in the quantum-classical hybrid strategy frequently results in a higher MetaSlice service delivery rate.

VII. CONCLUSION

By laying the foundation of digital human interaction with avatars, Metaverse demands heavy resource, and thereby



(a) MetaSlice Length = 5 (b) MetaSlice Length = 10 (c) MetaSlice Length = 15

FIGURE 8. Energy landscape visualization.

TABLE 2. MetaSlice service delivery rate comparison.

Case	Connectivity Probability	Acceptance Rate (WSQA - SA)	Acceptance Rate (WSQA - GA)	Acceptance Rate (CSQA - SA)	Acceptance Rate (CSQA - GA)	Acceptance Rate (QA)
A (small)	0.3	100%	100%	100%	100%	100%
	0.6	100%	100%	75%	75%	90%
	1	75%	75%	50%	65%	65%
B (medium)	0.3	85%	75%	50%	75%	65%
	0.6	70%	65%	50%	50%	50%
	1	35%	45%	5%	15%	15%
C (large)	0.3	60%	65%	35%	35%	45%
	0.6	35%	25%	0%	5%	5%
	1	25%	25%	0%	5%	5%

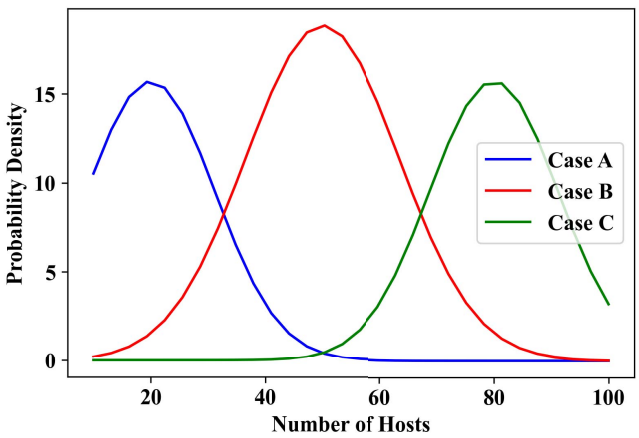


FIGURE 9. Probability distribution of sampling hosts.

sophisticated optimization strategy to maintain the resources. This paper presses on the needs to go beyond the realm and propose quantum computing instead of classical approaches to improve optimization criteria and running time. However, the limited availability of qubits (quantum computing units) can hinder the applicability of QA on large-scale MetaSlice systems. Thus, we propose a hybrid WSQA, basically QA warm started by classical SA, to shrink the Q matrix and energy landscape to be sampled by QA. Consequently, WSQA enable solving comparatively larger MetaSlice resource optimization instances, even with the limited qubits. Furthermore, we propose another hybrid quantum-classical CSQA strategy that initiates QA and makes transition towards the classical approach for fine-tuning solutions. While the CSQA strategy may not improve scalability compared to WSQA, it offers greater advantages in terms of reduced running time. The dream for Metaverse success and next-generation communication is highly dependent on the application of quantum computing. Thus, resource allocation along with other optimization problems for Metaverse need to be quantum-ready for now on. In future, with further availability of Quantum Processing Units (QPUs), which is very much likely due to the recent advancements of qubits quality, the performance of quantum computing can be accelerated. Moving far forward, QPUs can be placed at the edge of the network to decline communication overhead/delay between physical and virtual world.

REFERENCES

[1] Y. Wang et al., "A survey on metaverse: Fundamentals, security, and privacy," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 319–352, 1st Quart., 2023, doi: 10.1109/COMST.2022.3202047.

[2] W. C. Ng, W. Y. B. Lim, J. S. Ng, Z. Xiong, D. Niyato, and C. Miao, "Unified resource allocation framework for the edge intelligence-enabled metaverse," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2022, pp. 5214–5219, doi: 10.1109/ICC45855.2022.9838492.

[3] K. Li et al., "When Internet of Things meets metaverse: Convergence of physical and cyber worlds," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 4148–4173, Mar. 2023, doi: 10.1109/JIOT.2022.3232845.

[4] Y. Han, D. Niyato, C. Leung, C. Miao, and D. I. Kim, "A dynamic resource allocation framework for synchronizing metaverse with IoT service and data," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2022, pp. 1196–1201.

[5] Y. Han et al., "A dynamic hierarchical framework for IoT-assisted digital twin synchronization in the metaverse," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 268–284, Jan. 2023, doi: 10.1109/JIOT.2022.3201082.

[6] N. H. Chu et al., "MetaSlicing: A novel resource allocation framework for metaverse," *IEEE Trans. Mobile Comput.*, early access, Jun. 21, 2023, doi: 10.1109/TMC.2023.3288085.

[7] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6G: A comprehensive survey," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 334–366, 2021, doi: 10.1109/OJCOMS.2021.3057679.

[8] M.A. Habibi et al., "Toward an open, intelligent, and end-to-end architectural framework for network slicing in 6G communication systems," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 1615–1658, 2023, doi: 10.1109/OJCOMS.2023.3294445.

[9] H. Zhang, S. Mao, D. Niyato, and Z. Han, "Location-dependent augmented reality services in wireless edge-enabled metaverse systems," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 171–183, 2023, doi: 10.1109/OJCOMS.2023.3234254.

[10] M. Sharma, M. Moonen, Y. Lefevre, and P. Tsiaflakis, "Resource sharing strategies for point-to-multipoint distribution in next-generation DSL networks," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 2697–2716, 2023, doi: 10.1109/OJCOMS.2023.3319993.

[11] Y. Zhou, X. Liu, and L. Lei, "Multi-objective optimization for bandwidth-limited federated learning in wireless edge systems," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 954–966, 2023, doi: 10.1109/OJCOMS.2023.3266389.

[12] M. Emu and S. Choudhury, "DSO: An intelligent SFC orchestrator for time and resource intensive ultra dense IoT networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Montreal, QC, Canada, 2021, pp. 1–6.

[13] D. Volpe, G. A. Cirillo, M. Zamboni, and G. Turvani, "Integration of simulated quantum annealing in parallel tempering and population annealing for heterogeneous-profile QUBO exploration," *IEEE Access*, vol. 11, pp. 30390–30441, 2023, doi: 10.1109/ACCESS.2023.3260765.

[14] C. C. McGeoch, R. Harris, S. P. Reinhardt, and P. I. Bunyk, "Practical annealing-based quantum computing," *Computer*, vol. 52, no. 6, pp. 38–46, Jun. 2019, doi: 10.1109/MC.2019.2908836.

[15] A. Guillaume et al., "Deep space network scheduling using quantum annealing," *IEEE Trans. Quantum Eng.*, vol. 3, pp. 1–13, Aug. 2022, doi: 10.1109/TQE.2022.3199267.

[16] Z. Ye, X. Qian, and W. Pan, "Quantum topology optimization via quantum annealing," *IEEE Trans. Quantum Eng.*, vol. 4, pp. 1–15, Apr. 2023, doi: 10.1109/TQE.2023.3266410.

[17] S. Heng, D. Kim, T. Kim, and Y. Han, "How to solve combinatorial optimization problems using real quantum machines: A recent survey," *IEEE Access*, vol. 10, pp. 120106–120121, 2022, doi: 10.1109/ACCESS.2022.3218908.

[18] J. R. Jiang and C. W. Chu, "Classifying and benchmarking quantum annealing algorithms based on quadratic unconstrained binary optimization for solving NP-hard problems," *IEEE Access*, vol. 11, pp. 104165–104178, 2023, doi: 10.1109/ACCESS.2023.3318206.

[19] M. Emu, S. Choudhury, and K. Salomaa, "Quantum neural networks driven stochastic resource optimization for metaverse data marketplace," in *Proc. IEEE 9th Int. Conf. Netw. Softwarizat. (NetSoft)*, Madrid, Spain, 2023, pp. 242–246, doi: 10.1109/NetSoft57336.2023.10175433.

[20] Z. Meng, C. She, G. Zhao, and D. De Martini, "Sampling, communication, and prediction co-design for synchronizing the real-world device and digital model in metaverse," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 288–300, Jan. 2023, doi: 10.1109/JSAC.2022.3221993.

[21] M. Emu, S. Choudhury, and K. Salomaa, "Quantum computing empowered metaverse: An approach for resource optimization," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Rome, Italy, 2023, pp. 2412–2418.

[22] Y. Ren, R. Xie, F. R. Yu, T. Huang, and Y. Liu, "Quantum collective learning and many-to-many matching game in the metaverse for connected and autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 12128–12139, Nov. 2022, doi: 10.1109/TVT.2022.3190271.

[23] J. C. De Luna Ducoing and K. Nikitopoulos, "Quantum annealing for next-generation MU-MIMO detection: Evaluation and challenges," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2022, Seoul, South Korea, pp. 637–642, doi: 10.1109/ICC45855.2022.9839195.

[24] S. Harwood, C. Gambella, D. Trenev, A. Simonetto, D. Bernal, and D. Greenberg, "Formulating and solving routing problems on quantum computers," *IEEE Trans. Quantum Eng.*, vol. 2, pp. 1–17, Jan. 2021, doi: 10.1109/TQE.2021.3049230.

[25] F. Vista F, G. Iacovelli, and L. A. Grieco, "Quantum scheduling optimization for UAV-enabled IoT networks," in *Proc. CoNEXT Student Workshop*, 2021, pp. 19–20.

[26] M. Bhatia and S. K. Sood, "Quantum computing-inspired network optimization for IoT applications," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5590–5598, Jun. 2020, doi: 10.1109/JIOT.2020.2979887.

[27] M. Emu, S. Choudhury, and K. Salomaa, "Resource optimization of SFC embedding for IoT networks using quantum computing," in *Proc. IEEE Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, Paris, France, 2022, pp. 83–88.

[28] S. M. Park and Y. G. Kim, "A Metaverse: Taxonomy, components, applications, and open challenges," *IEEE Access*, vol. 10, pp. 4209–4251, 2022, doi: 10.1109/ACCESS.2021.3140175.

[29] J. Wang, M. Zhou, X. Jin, X. Guo, L. Qi, and X. Wang, "Variance minimization hedging analysis based on a time-varying Markovian DCC-GARCH model," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 2, pp. 621–632, Apr. 2020, doi: 10.1109/TASE.2019.2938673.

[30] T. Mamalis, D. Stipanović, and P. Voulgaris, "Stochastic learning rate with memory: Optimization in the stochastic approximation and online learning settings," *IEEE Control Syst. Lett.*, vol. 7, pp. 419–424, 2023, doi: 10.1109/LCSYS.2022.3186896.

[31] K. Shen, S. Safapourhajari, T. De Pessemier, L. Martens, W. Joseph, and Y. Miao, "Optimizing the focusing performance of non-ideal cell-free mMIMO using genetic algorithm for indoor scenario," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8832–8845, Oct. 2022, doi: 10.1109/TWC.2022.3170433.

[32] C. Madapatha, B. Makki, A. Muhammad, E. Dahlman, M.-S. Alouini, and T. Svensson, "On topology optimization and routing in integrated access and Backhaul networks: A genetic algorithm-based approach," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 2273–2291, 2021, doi: 10.1109/OJCOMS.2021.3114669.

[33] O. Adekoya and A. Aneiba, "An adapted nondominated sorting genetic algorithm III (NSGA-III) with repair-based operator for solving controller placement problem in software-defined wide area networks," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 888–901, 2022, doi: 10.1109/OJCOMS.2022.3172551.

[34] B. Raj, I. Ahmedy, M. Y. I. Idris, and R. M. Noor, "A hybrid sperm swarm optimization and genetic algorithm for unimodal and multimodal optimization problems," *IEEE Access*, vol. 10, pp. 109580–109596, 2022, doi: 10.1109/ACCESS.2022.3208169.

[35] G. M. Almeida, C. Camilo-Junior, S. Correa, and K. Cardoso, "A genetic algorithm for efficiently solving the virtualized radio access network placement problem," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Rome, Italy, 2023, pp. 1874–1879, doi: 10.1109/ICC45041.2023.10279334.

[36] X. Yang, H. Zhang, H. Ji, and X. Li, "Hybrid cooperative caching based IoT network considering the data cold start," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Nanjing, China, 2021, pp. 1–6, doi: 10.1109/WCNC49053.2021.9417118.

[37] S. Pan, H. Zhao, Z. Cai, D. Li, R. Ma, and H. Guan, "Sustainable serverless computing with cold-start optimization and automatic workflow resource scheduling," *IEEE Trans. Sustain. Comput.*, early access, May 25, 2023, doi: 10.1109/TSUSC.2023.3311197.

[38] Q. Li, L. Huang, W. Liu, X. Chen, and L. Feng, "Cold-start satellite signal acquisition aided by an antenna array in the presence of outliers," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 5847–5861, May 2023, doi: 10.1109/TVT.2022.3230543.

[39] (Gurobi Optimization, Beaverton, OR, USA). *Gurobi 11.0: Every Solution, Globally Optimized*. 2023. [Online]. Available: https://www.gurobi.com

[40] A. Hagberg and D. Conway. *Networkx: Network Analysis With Python*. Networkx. 2020. [Online]. Available: https://networkx.github.io
[41] X. Lin, Z. Wei, and P. Yao, "Quantum and classical hybrid generations for classical correlations," *IEEE Trans. Inf. Theory*, vol. 68, no. 1, pp. 302–310, Jan. 2022, doi: 10.1109/TIT.2021.3123401.
[42] S. Chetna and D. Swades, *Resource Allocation in Next Generation Broadband Wireless Access Networks*. Pennsylvania, PA, USA: IGI Global, 2017.

MAHZABEEN EMU (Graduate Student Member, IEEE) received the B.Sc. degree in computer science and engineering from the Ahsanullah University of Science and Technology in 2017, and the M.Sc. degree in computer science from Lakehead University. She is currently pursuing the Ph.D. degree with the School of Computing, Queen's University, ON, Canada. Her research interests include optimization and artificial intelligence in the field of networking. She was the recipient of Vector Institute AI Scholarship in 2019, OGS in 2020 and 2021, OGS 2021-22, OGS 2022-23, Mitacs Accelerate, and Mitacs Business Intelligence Grant. She has also received the 2021 Governor General Gold Medal Award. Recently, she has been awarded the prestigious Vanier Canada Government Scholarship.

SALIMUR CHOUDHURY (Senior Member, IEEE) is an Associate Professor with the School of Computing, Queen's University. He leads the Global Optimization, Analytics, and Learning Lab, Queen's School of Computing. Earlier, he was an Associate Professor and the Graduate Co-Coordinator with the Department of Computer Science and used to lead the Optimization Research Group, Lakehead University. He has published more than 50 peer-reviewed publications and received grants from various government sectors and industries as well. He is the co-founder of the conference, SGIoT. His primary research focus on network optimization.

KAI SALOMAA received the Ph.D. degree from the University of Turku in 1989, supervised by R. V. Book and M. Steinby. He has been a Professor with Queen's University, Canada, since 2000. He has published over 120 papers in refereed journals. His research lies in the theory of computation, including areas, such as algorithms and descriptional complexity. He is known for his numerous contributions to the state complexity of finite automata. His highly cited 1994 joint paper laid the foundations of the area.