

Next-Generation Orchestration: Quantum Computing for Network Services

Engin Zeydan*, Luis Blanco*, Josep Mangués-Befalluy*, Suayb S. Arslan[†], Yekta Turk[‡]

*Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Barcelona, Spain, 08860.

[†] Massachusetts Institute of Technology, MA, USA, 02139.

[‡] Independent Researcher, Istanbul, Turkey, 34906.

Email: {engin.zeydan, luis.blanco, josep.mangués}@cttc.cat,
sarslan@mit.edu, yektaturk@gmail.com.

Abstract—As the demand for high-performance and dynamically adaptable network services continues to grow, the convergence of quantum computing and network orchestration represents a promising development. This paper explores the transformative potential of quantum computing in the orchestration of network services and introduces new paradigms that transcend classical limitations. Using a case study, we propose a workflow for an exemplary link failure process for generative AI (GenAI) and quantum computing-based service orchestration in an edge cloud (EC) network. Later, we address the challenges of integrating services based on quantum computing (using a quantum search algorithm to select the best connections) with GenAI-based topology control and management in an orchestration framework. Our work also introduces multi-solution quantum search for orchestrating services across different edge clouds, emphasizing efficiency and accuracy. Algorithmic considerations, including the impact of the oracle and hybrid approaches, are also thoroughly explored. We conclude the paper with potentials and comparisons of possible GenAI techniques and quantum search algorithms in terms of their advantages and disadvantages.

I. INTRODUCTION

The increasing complexity of modern network infrastructures, combined with ever-evolving landscape of service demands, requires a paradigm shift in the way we orchestrate and manage network services. Quantum computing, with its inherent parallelism and computational power, is a disruptive force capable of reshaping the landscape of network orchestration [1]. One of the first examples of quantum search is provided by the Grover's algorithm [2]. Grover's algorithm specializes in accelerating the search process and demonstrates its ability to solve complex computational problems. It is shown to accelerate brute-force search, reducing the search space from $O(N)$ to $O(\sqrt{N})$ where N is the size of the search space, using quantum parallelism to efficiently find a solution. However, unlike other Quantum algorithms such as Shor's algorithm, the overall search process falls short of polynomial time objective. Generative Artificial Intelligence (GenAI), on the other hand, is a complementary technology that adds another layer of intelligence to the orchestration and management of network services. GenAI uses machine learning techniques to independently create, adapt and optimize network topologies and configurations [3]. By using generative models, it can learn from historical data and react dynamically to changing

network conditions and service requirements. Therefore, the synergy between quantum computing and GenAI [4] has the potential to overcome the complex challenges posed by the dynamic and complicated nature of modern networks.

This paper explores the integration of quantum computing (in particular search algorithms) in conjunction with GenAI techniques into the orchestration of network services, ushering in a new era in optimizing resource utilization, fault tolerance, and rapid adaptation to changing requirements. In our design approach, quantum computing efficiently explores solution spaces for the Edge Cloud (EC) connectivity options and delivers optimized configurations. GenAI, on the other hand, improves adaptability and responsiveness by continuously learning and generating new network configurations (or service paths) that are tuned to the evolving service requirements on the existing topologies to achieve the goal of polynomial time in network orchestration along with quantum search. Together, these technologies form a powerful combination for orchestrating complex network services in a way that is both efficient and adaptable to the dynamic requirements of modern communication infrastructures.

II. CONSIDERED USE CASE AND PROPOSED WORKFLOW

A. Problem Statement over a Case Study

Fig. 1 presents a high-level schematic representation of the problem at hand. During a link failure different network management and orchestration entities (Orchestrator for cloud and Software Defined Networking (SDN) controller for network) operates to optimize the edge-cloud network based on their point of view. Multiple strategies can be explored, each yielding varied EC connection options and consequently, diverse paths to mitigate the impacted traffic [5], [6]. Three distinct strategies can be employed:

(i) *Cloud-aware Approach*: This strategy relies solely on cloud Key Performance Indicators (Key Performance Indicators (KPIs)), with Kubernetes making core selections based on cloud metrics such as Central Processing Unit (CPU), memory, and storage. However, it operates without awareness of underlying transport network KPIs. (ii) *Transport-aware Approach*: In this scenario, core selection is dictated by the transport network, independent of cloud KPIs. While

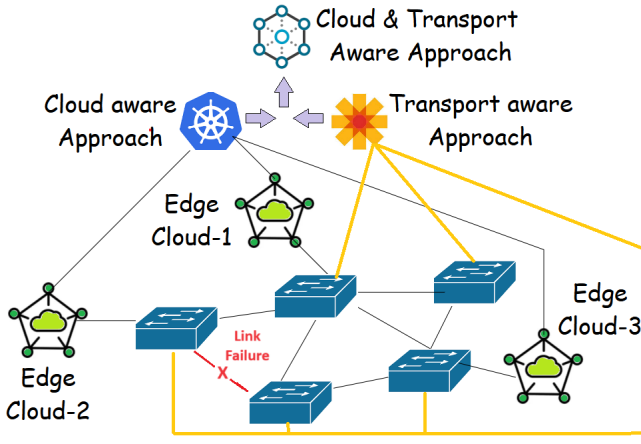


Fig. 1: The illustration of cloud-aware, transport-aware and cloud&transport aware service orchestration in an edge-cloud network.

it addresses routing optimization concerns, it may lead to congestion and service disruptions if the selected core is busy. (iii) *Cloud & Transport-aware Approach*: This strategy integrates both transport network and cloud KPIs to determine EC selection. It comes into play when the initially chosen EC is occupied or unavailable due to various factors.

B. Workflow

Fig. 2 shows the flow of a service orchestration based on quantum computing that combines generative Artificial Intelligence (AI) for topology control and management with a quantum search algorithm for selecting the best connections. The service orchestration problem defined in the previous subsection is based on the network topology, cloud parameters and transport constraints. (i) *Quantum Circuit Design*: To solve this problem, a quantum circuit containing the quantum search algorithm (e.g. Grover's algorithm) must first be designed to efficiently search for optimal solutions. (ii) *Generative AI for Topology Control*: Generative AI techniques can be used to optimize the network topology. This can include the generation of alternative topologies taking into account factors such as link capacities, fault tolerance and geographical aspects. The AI model, which has been trained offline based on training data, can suggest topologies that improve the overall efficiency of the network online. (iii) *Quantum Search Algorithm-Assisted Topology Verification*: Quantum search algorithms can be used to verify the effectiveness of the generated network topologies. Quantum search algorithms can help to quickly evaluate the quality of different topological configurations, taking into account both classical and quantum parameters. (iv) *Encoding Problem Parameters for Quantum Search*: The relevant parameters of the service orchestration problem must be encoded into the quantum state. This includes encoding information about cloud parameters (such as CPU, memory, storage), parameters of the transport network and the specific requirements of the affected Base Station (BS). (v) *Quantum Search Execution*: The executed quantum search

algorithm finds optimal EC connections based on the encoded parameters. The quantum search efficiently explores possible solutions while taking into account the constraints of the cloud and the transport network. (vi) *Integration with Classical Management and Orchestration (MANO) Strategies*: Integrate quantum search results with classical service orchestration strategies. Combine quantum-assisted path selection with the three predefined strategies (cloud-aware, transport-aware and cloud & transport-aware) for comparison purposes. (vii) *Path Selection and Validation*: Evaluate the results of quantum-based path selection for different strategies. Verify that the selected paths fulfil the requirements of the affected BS, taking into account link-level constraints, cloud parameters and identified faults. (viii) *Feedback Loop with Generative AI*: Create a feedback loop between quantum-enabled service orchestration and generative AI for continuous improvement. Generative AI can customise its topology suggestions based on the efficiency and performance of quantum-assisted orchestration. (ix) *Deployment and Monitoring*: Deploy quantum-enabled service orchestration in a controlled environment and monitor its performance in a production-like environment. Implement mechanisms to continuously monitor, collect feedback and adapt to dynamic network conditions.

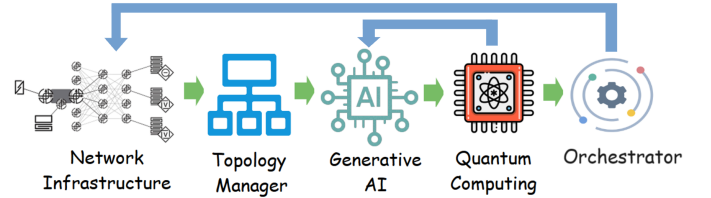


Fig. 2: The flow for quantum computing based service orchestration.

Fig. 3 shows the example link failure and the process for quantum computing based service orchestration in an EC network. An orchestration service based on a recovery process through topology proliferation using generative AI and local/collaborative quantum search (quantum computing) is key to our proposed methodology. 1. *Link Failure Detection*: The orchestration process begins in a first step with the detection of a link failure within the network. This detection can be triggered by various mechanisms, e.g. real-time monitoring, anomaly detection or feedback from network devices indicating a connectivity problem. 2. *Topology Proliferation using GenAI*: After the orchestration service has detected the failure of the connection, it initiates a recovery process. This step is where GenAI comes into play. The existing active connections in the network are used as a basis and the GenAI algorithms independently create a new topology. This topology proliferation is driven by AI models that adapt and optimize the network structure based on historical data, performance metrics and learned patterns. 3. *Service Path Generation with GenAI*: In the next step, all possible service paths are created with the newly generated topology using generative AI. Using the power of AI, the system explores different

routing options, taking into account factors such as traffic patterns, latency requirements and Service Level Agreements (SLAs). This step ensures a comprehensive set of service paths that can potentially be used to reroute the affected traffic.

4. Quantum Computing for Optimal Service Path Selection: In the final step, quantum computing is integrated into the orchestration process. Using local or collaborative quantum search algorithms, the system evaluates and selects the best service path from the set generated in the previous step. The quantum computer's ability to efficiently explore solution spaces enables it to quickly determine the optimal route that satisfies the network and service constraints.

Once the optimal service path has been determined by quantum computing, the orchestration service implements the chosen path in the network infrastructure. This includes reconfiguring network devices, updating routing tables and establishing the necessary connections to reroute the affected traffic. Successful implementation results in the services affected by the connection failure being restored. After recovery, the orchestration system continues to monitor the performance of the network and the provision of services. This monitoring serves as valuable feedback for both the generative AI and the quantum computing components. The system adapts to changes in the network environment, continuously learns from new data and refines its orchestration strategies for future resilience.

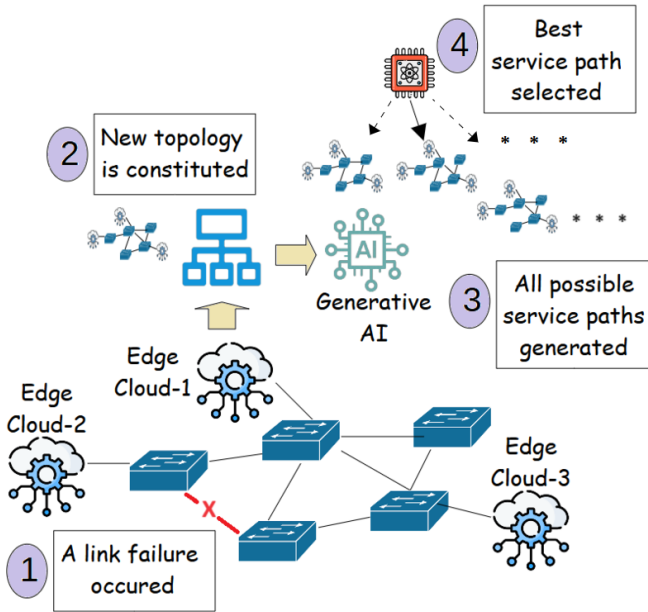


Fig. 3: An example link failure in an edge-cloud network is illustrated.

III. INTEGRATION AND IMPLEMENTATION

The integration of quantum search algorithms into existing network management and orchestration frameworks involves several steps to ensure a seamless integration of quantum capabilities.

(i) *Quantum oracle design:* After defining the problem, a quantum oracle must be designed that represents the search problem in such a way that the quantum search algorithm can efficiently search for solutions. The oracle should be designed in such a way that it evaluates proposed solutions and provides quantum states that correspond to valid solutions. Network management and orchestration frameworks usually have classic components that fulfil different tasks.

(ii) *Interface with classical components:* The interfaces between the quantum algorithm and these classical components should be designed to ensure smooth communication and data exchange via Application Programming Interfaces (APIs), data structures or other integration mechanisms. Later, a quantum computing platform that supports the execution of quantum search algorithms should be selected. This could be the use of a cloud-based quantum computing service or an on-site quantum computer, if available.

(iii) *Quantum algorithm execution:* The quantum algorithm should be adapted to the specific characteristics and limitations of the chosen quantum computing hardware [7].

(iv) *Data preprocessing and postprocessing:* After running the quantum algorithm, the results need to be post-processed to extract meaningful information that can be used by the classical components of the orchestration framework [8]. The input data should be prepared for the quantum algorithm to ensure that it is encoded in a format suitable for quantum computation.

(v) *Hybrid classical-quantum approach:* A hybrid approach in which classical computation is used in conjunction with the quantum algorithm must be considered [9]. For example, certain tasks can be solved more efficiently by classical means, and the results can be combined with the quantum results to form a comprehensive solution.

(vi) *Error mitigation and noise handling:* Since quantum computers are prone to errors and noise, error mitigation and noise handling techniques should be implemented to increase the reliability of the quantum algorithm results [10]. This may include error correction codes or other strategies specific to the quantum computing platform.

(vii) *Security and authentication:* Best security practices should be followed during integration, especially if the network management and orchestration framework contains sensitive data. Authentication mechanisms should also be implemented to secure communication between classical and quantum components [11].

(viii) *Testing and validation:* The integrated solution need to be tested in simulated or controlled environments before deploying it in production. The performance of the quantum algorithm and its impact on the overall network management and orchestration workflow should be validated [12].

(ix) *Monitoring and maintenance:* Monitoring mechanisms must be implemented to track the performance of the integrated quantum solution over time [13]. The system must be regularly updated and maintained to reflect improvements and advancements in quantum computing technology.

A. Challenges and Limitations

The implementation of an integrated approach, in particular the integration of quantum algorithms and classical network management and orchestration systems, is associated with a number of challenges and limitations.

(i) *Quantum Hardware Constraints*: The availability and maturity of quantum hardware is still under development. Current quantum computers have limited qubits, coherence times and error rates. This can impact the scalability and reliability of quantum algorithms for real-world network problems.

(ii) *Algorithmic Complexity*: Quantum algorithms can have a different time complexity compared to their classical counterparts. While some problems show a speedup due to quantum effects, the overhead of quantum operations, initialization and measurement can cancel out the advantages for certain cases.

(iii) *Noise and Error Mitigation*: Quantum computers are susceptible to noise and errors due to factors such as decoherence and gate imperfections. Implementing effective error correction or mitigation techniques is crucial to achieve reliable results, especially in complex network scenarios.

(iv) *Limited Quantum Access*: Access to quantum computers is currently limited, and organizations may have to rely on cloud-based quantum services. This brings with it potential latency, security concerns and reliance on external providers.

(v) *Integration with Classical Systems*: The seamless integration of quantum algorithms and classical network management and orchestration systems requires well-defined interfaces. Bridging the gap between quantum and classical computational models is a challenge and effective communication protocols need to be established.

(vi) *Data Encoding and Preprocessing*: The preparation of classical data for quantum algorithms can include encoding and preprocessing steps. The efficiency of these steps and the compatibility of quantum data representations with classical systems can be a challenge.

(vii) *Security and Trust*: Quantum computing brings with it new security considerations, such as the potential threat to classical encryption algorithms. Ensuring the security and trustworthiness of the integrated approach is crucial, especially when dealing with sensitive network data.

(viii) *Cost Considerations*: Building and maintaining a quantum-enabled infrastructure, including access to quantum computers and expertise, can be costly. Entities need to evaluate the cost-benefit ratio and justify the investment in quantum technologies.

(ix) *Regulatory Compliance*: Regulatory compliance and standards are critical in network environments, and the integration of quantum technologies can bring new challenges.

B. Comparisons with baseline management platform

The evaluation and comparison of an evolved architecture that combines quantum search algorithm with an existing classical network service and management platform requires a systematic approach that takes into account various aspects as listed below:

(i) *Evaluation Metrics*: The metrics that will be used to evaluate the performance of both architectures need to be clearly defined. These metrics may include:

- 1. *Time Efficiency*: Measure the time taken by each architecture to perform specific tasks, such as network optimization or service orchestration.
- 2. *Resource Utilization*: Evaluate the usage of computational resources, including CPU, memory, and storage, by each architecture.
- 3. *Solution Quality*: Assess the quality of solutions provided by each architecture in terms of meeting network and service requirements.
- 4. *Scalability*: Examine how well each architecture scales with the size and complexity of the network.

(ii) *Quantitative Analysis*: Quantitative analyses based on the defined metrics must be performed. For quantum search algorithms, factors such as the number of iterations required for convergence, the probability of success and the impact of quantum parallelism on search speed should be considered.

(iii) *Simulation and Modeling*: Simulation tools or modeling techniques can be used to replicate real-world scenarios. Simulations can help evaluate the performance of each architecture under different conditions and identify potential scalability issues.

(iv) *Benchmarking*: tests can be performed to compare the developed architecture with quantum search algorithm to the baseline network MANO platforms. Benchmarking could be done with different scenarios, including different network sizes, traffic patterns and service requirements.

(v) *Real-world Testing*: Perform real-world testing in a controlled environment to observe how each architecture performs under actual operating conditions. This may mean deploying both architectures on a test network and measuring their performance in a production-like environment.

(vi) *Fault Tolerance and Robustness*: Evaluate the fault tolerance and robustness of each architecture. Introduce simulated faults or unexpected events and observe how well each platform can recover and maintain optimal operation.

(vii) *Cost Analysis*: The cost of each architecture, including the cost of hardware, software, maintenance and additional resources required, should be considered. The total cost of ownership for both the base platform and the platforms supported by quantum search algorithms must be compared.

(viii) *Security Analysis*: A security analysis must be performed to ensure that both architectures meet the necessary security standards and requirements. Assess vulnerabilities and potential risks associated with each platform.

(ix) *Quantum Advantage Assessment*: If quantum search is expected to have a quantum advantage, evaluate the extent of this advantage in terms of speedup and efficiency compared to classical search algorithms in a MANO platform. Consider the impact of quantum noise and error correction mechanisms.

IV. MULTI-SOLUTION QUANTUM SEARCH FOR MULTI-EDGE-CLOUD ORCHESTRATION

Let us suppose that each EC i has its own view of the network with limited communication capabilities. In the event of a connection failure that interrupts the service, a GenAI generates a new set of topologies/service paths locally. The generation technique could, for example, be based on deep autoregressive models [14] for scalability. However, the generated topologies are not necessarily accurate and could solve the link breakage issue for continued service (see Fig. 3) in all cases. Suppose that of $N_i = |\mathcal{N}_i|$ topological solutions, where \mathcal{N}_i is the set of these solutions, $S_i = |\mathcal{S}_i|$ of them are accurate and useful. While the remaining generated topologies exhibit structural accuracy, they may not be able to effectively solve the problem of link failures. Given the operation of m ECs under the orchestrator, quantum search has the task of quickly finding the appropriate topology for each EC. This aims to promptly resolve potential link failures that can have a significant impact on the overall global performance of network connectivity.

In our example scenario (Quantum computing), we are interested in the number of Grover iterations to find the solution for all edge clouds. The oracle $U_f^{(i)} : |x\rangle \rightarrow |f(x)\rangle$ can be designed to evaluate the generated topology for connectivity particularly for testing the paths that use the failed link. In other words, the function $f(\cdot)$ is the topology evaluator that equals 1 whenever topology x can recover the failed link. The oracle is queried for topology evaluations and the service delay will be measured in terms of the queries rather than the running time as they are simpler to reason about and present a lower bound to running time. Note that topologies generated for i -th edge cloud could be a potential solution for j -th edge cloud as well. It holds significance in the absence of immediate interaction between the clouds. This observation will be taken care of using either collaborative and non-collaborative Grover iterations.

If we use the Grover's plain vanilla version of the algorithm [2] for multi-solution search it is previously established that we would need $O(\sqrt{N_i/S_i})$ iterations to converge to one of the accurate solutions for edge cloud i almost surely [15]. Considering some of the generated topologies are identical, the total number of Grover iterations in case the edge clouds work collaboratively can be shown to satisfy

$$O\left(\sqrt{\frac{|\cup_i \mathcal{N}_i|}{|\cup_i \mathcal{S}_i|}}\right) \leq O\left(\sum_i \sqrt{N_i/S_i}\right) \quad (1)$$

where the upper bound assumes that edge clouds operate independent of each other (a union bound).

There are two important issues with the number of generated potential topologies. First, the number of viable/accurate topologies play a significant role in the complexity of the collaborative Grover search i.e., the performance of the GenAI could make the overall search process polynomial time if

$$|\cup_i \mathcal{S}_i| \leq \frac{\sum_i N_i}{\log^2(\sum_i N_i)} \quad (2)$$

which could not be better than quadratic time if the edge clouds generate constant accurate topologies without collaboration.

The second problem is that in practice it is not always practicable to know a priori the number of accurate topologies that would exist in the edge region. An extension of the plain vanilla form of the Grover search algorithm was introduced by Boyer et al. [16] in the form of the so-called Boyer-Brassard-Høyer-Tap (BBHT) algorithm. However, since the number of accurate topologies S_i would be an unknown, we cannot determine the required number of Grover iterations. This technique uses classical processing and a "trial-and-error" approach, resulting in $O(\sqrt{N_i})$ Grover iterations, which makes the overall process take quadratic time. On the other hand, the accuracy rate when training the GenAI can be used to estimate the S_i 's, which in turn has a positive impact on the overall latency performance of the orchestration. As can be seen, the careful design of the GenAI for more accurate topology generations and the subsequent quantum computation (searching and finding the right topology) is quite important to optimize the search time of the system (polynomial number of Grover iterations) and thus the repair of the links to finally restore the network connectivity on time.

V. ALGORITHMIC CONSIDERATIONS

While the Grover algorithm is a well-known quantum search algorithm, there are other quantum algorithms such as Quantum Approximate Optimization Algorithm (QAOA) [17] and Quantum Walks [18] can be adapted or used for search related tasks depending on the specific problem and requirements. Optimizing the Quantum search algorithm (e.g. Grover) for the specific problem of network and service orchestration involves understanding the nature of the problem and adapting the quantum search to efficiently explore the solution space. The optimization process involves a balance between algorithmic improvements and practical considerations of the quantum computing platform. Some suggestions to optimize the Grover search algorithm are as follows:

(i) *Quantum Oracle Design*: Design a quantum oracle that evaluates the suitability of a solution based on transport and cloud parameters. The oracle is a crucial component of the algorithm, as it marks the solution(s) of the search problem. The efficiency of the algorithm depends on how effectively the oracle can identify these solutions. The oracle should be able to verify whether a given solution (combination of paths and EC selection) fulfils the requirements imposed by the constraints of the cloud and the transport network.

(ii) *State Encoding*: Encode the problem state efficiently in qubits. Map the different components of your problem (regions, BS, EC, links, etc.) to qubits in a way that captures the relationships and constraints between them. This encoding should facilitate the exploration of valid solutions.

(iii) *Amplitude Amplification*: Use Grover's amplitude amplification to increase the probability of finding the correct solution. Since the problem involves three different strategies,

the amplitudes of the states corresponding to the optimal solutions according to each strategy should be amplified.

(iv) *Parallelization and Superposition*: Exploit the parallelization capabilities of quantum computers by using superposition. This allows the algorithm to evaluate multiple potential solutions simultaneously. Ensure that the quantum algorithm explores the solution space in a way that efficiently considers different combinations of cloud-aware, transport-aware and cloud & transport-aware strategies.

(v) *Dynamic Programming or Quantum Dynamic Programming*: If necessary, consider using dynamic programming techniques to break the problem into smaller subproblems and solve them iteratively. The advantages of quantum dynamic programming in terms of quantum parallelism can be explored.

(vi) *Adaptive Strategies*: Allow the quantum algorithm to adaptively choose strategies based on intermediate results. For this purpose, the oracle or the Grover diffusion operator could be modified to direct the search to more promising solution spaces.

(vii) *Hybrid Classical-Quantum Approach*: Consider a hybrid approach combining classical computations with quantum computations. Classical algorithms can be used to pre- or post-process certain aspects of the problem, making the quantum part of the algorithm more efficient.

(viii) *Noise Mitigation*: Implement noise suppression techniques as quantum computers are prone to errors. This may include error correction codes or other noise suppression strategies [19] to increase the reliability of the quantum algorithms.

(ix) *Experimentation and Iterative Refinement*: Quantum algorithms often benefit from an iterative refinement process. Experiment with different parameters or even consider adjusting the problem formulation based on the insights gained during the execution of the algorithm.

A. Efficiency and Accuracy

To find the optimal network and service configuration, several factors must be taken into accounts.

Search Space Structure: Grover search is particularly effective for unstructured search problems where the solution space is not well organised and offers a quadratic speedup over classical algorithms. QAOA, on the other hand, is designed for optimization problems with structured objective functions. Quantum walks can be applied to a variety of graphs and structured spaces. Therefore, the type of search space in the network and service configuration can influence the choice of algorithm.

Objective Function Optimization: QAOA is specially tailored to solving optimization problems and finding approximate solutions for combinatorial optimization tasks.

Quantum Parallelism and Superposition: Grover's algorithm uses quantum parallelism and superposition to explore multiple possibilities simultaneously. QAOA also uses quantum parallelism, but in the context of optimization landscapes. Quantum walks can exhibit quantum superposition, but their

dynamics are more complex and can involve classically-like random walks.

Oracle Design and Problem Encoding: The design of the quantum oracle that encodes the problem information is crucial. Grover's algorithm uses an oracle to mark the solution space. QAOA requires the encoding of the objective function, and quantum walks require the graph structure to be defined.

Adaptability to Dynamic Changes: Quantum walks can be particularly well adapted to dynamic network structures due to their graph-based nature. The performance of QAOA may depend on the specific optimization landscape.

B. Oracle Impact

Some important considerations about the impact of the oracle on the performance of a quantum search algorithm are as follows:

Oracle Function Definition: The oracle function is responsible for marking the solution states by applying a phase inversion. The design of this function should ensure that it reverses the sign of the amplitude for the target states and leaves the non-target states unchanged.

Completeness of Marking: The oracle must indeed mark all valid solutions to the search problem. If the oracle fails to mark a valid solution or marks invalid states, the efficiency of the search algorithm may be impaired.

Number of Solutions: The quantum search algorithm is particularly efficient when the number of solutions is small compared to the total number of possible states. The design of the oracle should take into account the specific number and structure of solutions in the problem domain.

Quantum Parallelism Exploitation: Effective oracles utilise the principles of quantum parallelism so that the quantum algorithm can explore multiple possibilities simultaneously. This exploitation of parallelism contributes to the quantum acceleration of search algorithms.

Quantum Database Representation: The design of the oracle is influenced by how the quantum database is represented. Efficient encoding and manipulation of data in quantum state space contribute to the overall efficiency of the search algorithm.

Oracle Complexity: The computational complexity of the oracle affects the overall complexity of the quantum search algorithm. Minimizing the computational complexity of the oracle is crucial for speeding up the quantum search.

Amplitude Amplification: The oracle should be designed to facilitate the amplification process of some quantum search algorithms, such as the Grover algorithm, to increase the probability amplitudes of the correct solutions.

C. Hybrid Approaches

Hybrid approaches that combine Grover's search algorithm with other classical or quantum algorithms can be considered to leverage the strengths of each approach. Hybrid algorithms aim to utilize the advantages of classical and quantum methods to address certain challenges more effectively. Below d some

TABLE I
POTENTIALS OF GENAI AND QUANTUM SEARCH ALGORITHMS FOR TOPOLOGY MANAGEMENT IN THE MANAGEMENT AND ORCHESTRATION OF NETWORK SERVICES DOMAIN.

Solutions	Characteristics	Advantages	Disadvantages
GenAI-based Techniques	<ul style="list-style-type: none"> — Leverages generative AI to optimize network topologies — Optimize communication paths within the network, reducing latency and improving the overall responsiveness of the system — Refine understanding of network dynamics, leading to ongoing enhancement in the quality of topology management decisions — Adapt topology management strategies to different architectural paradigms — Ensure efficient utilization of bandwidth, computing resources, and other network components, leading to improved performance and cost-effectiveness 	<ul style="list-style-type: none"> — <u>Adaptability and Learning:</u> Enables adaptive topology control based on real-time conditions. — <u>Efficient Resource Utilization:</u> Optimize the allocation of resources within the network — <u>Fault Tolerance:</u> Design network topologies with built-in fault tolerance. — <u>Scaling:</u> Accommodate the growth of network size and complexity based on its optimization strategies — <u>Enhanced Security:</u> designing topologies that incorporate security measures — <u>Diverse Network Architectures:</u> Support traditional on-premises networks, cloud-based architectures, hybrid infrastructures — <u>Service Requirements:</u> customization of network topologies based on specific service requirements — <u>Reduced Human Intervention:</u> reduces the need for extensive manual intervention. 	<ul style="list-style-type: none"> — <u>Computational Complexity:</u> Requiring significant processing power and resources. — <u>Training Data Dependency:</u> Relies on the quality and representativeness of the training data. — <u>Limited Interpretability:</u> Understanding the decisions and reasoning behind the generated network topology can be challenging — <u>Adversarial Attacks:</u> susceptible to adversarial attacks where malicious actors intentionally manipulate input data to deceive the model — <u>Resource Overhead:</u> Introduced by implementing and maintaining generative AI models — <u>Continuous Model Maintenance:</u> Require continuous maintenance to stay relevant and effective
Quantum Search Methods	<ul style="list-style-type: none"> — Utilize quantum search algorithms for efficient path optimization. — Leverage the principle of , superposition allowing to explore multiple possibilities simultaneously — Entanglement can facilitate the representation of interconnected network elements, influencing the exploration of optimal topologies — Exploit the concept of quantum parallelism, enabling them to process exponentially many states in parallel — Provide quadratic speedup in unstructured search problems for topology optimization — Quantum parallel queries explore a large solution space efficiently. 	<ul style="list-style-type: none"> — <u>Exponential Speedup:</u> Provides exponential speedup in search for optimal solutions. — <u>Parallelism and Superposition:</u> enhances the algorithm's ability to consider a vast number of possibilities concurrently. — <u>Reduced Search Complexity:</u> reduce the computational complexity associated with search problems. — <u>Global Optimization:</u> Finding a globally optimal configuration — <u>Hybrid Quantum-Classical Approaches:</u> Offering a versatile and complementary strategy for topology management. 	<ul style="list-style-type: none"> — <u>Quantum Hardware Constraints:</u> Requires quantum hardware, susceptible to noise and errors. — <u>Quantum Error and Noise:</u> Susceptible to errors and noise impacting the accuracy and reliability of quantum search algorithms — <u>Limited Qubit Coherence Time:</u> Topology management tasks may require longer computation times, — <u>Quantum Measurement:</u> managing and interpreting probabilistic results can be challenging. — <u>Complex Algorithmic Design:</u> The algorithms often involve complex mathematical principles

scenarios can be found in which hybrid approaches can be advantageous:

Classical Preprocessing and Postprocessing: Use classical algorithms for pre-processing tasks such as data cleansing, filtering or initial problem definition before applying the quantum search algorithm. Also use classical algorithms for post-processing steps to refine and validate the quantum results.

Quantum Speedup for Specific Subproblems: Identify subproblems within a larger computational problem that can benefit from quantum acceleration. Apply the quantum search algorithm to efficiently solve these specific subproblems, while using classical algorithms for the rest of the computation.

Integration with Heuristic Algorithms: Combine the quantum search algorithm with classical heuristic algorithms (e.g. GenAI) to guide the exploration of the search space. Heuristics

can help narrow down the possibilities and make the quantum search more targeted and efficient.

Quantum-Assisted Optimization: Integrate quantum search algorithm with classical optimization algorithms to improve the search for optimal solutions. The quantum component can be used to explore solution spaces faster, and classical optimization techniques can further refine the results.

Quantum Sampling and Machine Learning: Integrate the quantum search algorithm for quantum sampling or amplitude estimation in machine learning tasks. Quantum components can be used to efficiently sample data or estimate probabilities, complementing classical machine learning algorithms.

Parallel Computation with Quantum and Classical Nodes: Develop a hybrid system in which both quantum and classical nodes operate in parallel. Use the quantum search algorithm

on the quantum nodes for certain computations, while the classical nodes handle other aspects of the problem.

Hybrid Cloud-Quantum Solutions: Implement a hybrid solution where certain tasks are offloaded to a quantum processing unit in the cloud while classical computations are performed locally. This can be particularly useful when access to quantum hardware is limited.

Quantum Reinforcement Learning: Combine quantum search algorithm with classical reinforcement learning techniques to improve decision making in dynamic environments. The quantum component can help with rapid exploration of state spaces.

Error Correction and Fault Tolerance: Integrate classical error correction mechanisms with quantum search algorithm to improve the reliability of quantum computations. This is especially important in real-world scenarios where quantum hardware can be prone to errors.

Dynamic Switching between Classical and Quantum Modes: Develop algorithms that dynamically switch between classical and quantum modes depending on problem characteristics or algorithmic progress. This adaptive approach can optimize the use of quantum resources.

Finally, Table I summarises the potentials of GenAI and quantum search algorithms for topology management in the domain of network service management and orchestration.

VI. CONCLUSIONS

In this paper, we explore the transformative landscape of quantum computing in conjunction with GenAI techniques for network service orchestration. Considering the ever-increasing complexity of network services, we have proposed a workflow that addresses the adaptation of GenAI-powered quantum search design for topology management and service requests under the constraints of cloud and transport networks. The introduction of multi-solution quantum search for orchestrating services across multiple edge clouds has the potential for optimizing service paths. Algorithmic considerations, including the impact of the oracle and hybrid approaches, were also explored. At the end of the paper, we have provided a comparative analysis with insights into the potential and comparisons of various GenAI techniques and quantum search algorithms in the context of network service orchestration.

ACKNOWLEDGMENT

This work was partially funded by Grant PID2021-126431OB-I00 funded by MCIN/AEI/10.13039/501100011033 and by “ERDF A way of making Europe”, Generalitat de Catalunya grant 2021 SGR 00770 and Spanish MINECO - Program UNICO I+D (grants TSI-063000-2021-54 and -55).

REFERENCES

- [1] C. Bernhardt, *Quantum computing for everyone*. Mit Press, 2019.
- [2] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 212–219, 1996.
- [3] A. Karapantelakis, P. Alizadeh, A. Alabassi, K. Dey, and A. Nikou, “Generative ai in mobile networks: a survey,” *Annals of Telecommunications*, pp. 1–19, 2023.
- [4] S. Pise, A. A. Agarkar, and S. Jain, “Unleashing the power of generative ai and quantum computing for mutual advancements,” in *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, pp. 1–7, IEEE, 2023.
- [5] E. Zeydan, J. Manges-Bafalluy, and Y. Turk, “Intelligent service orchestration in edge cloud networks,” *IEEE Network*, vol. 35, no. 6, pp. 126–132, 2021.
- [6] C. R. De Mendoza, B. Bakhshi, E. Zeydan, and J. Manges-Bafalluy, “Near optimal vnf placement in edge-enabled 6g networks,” in *2022 25th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, pp. 136–140, IEEE, 2022.
- [7] S. S. Gill, A. Kumar, H. Singh, M. Singh, K. Kaur, M. Usman, and R. Buyya, “Quantum computing: A taxonomy, systematic review and future directions,” *Software: Practice and Experience*, vol. 52, no. 1, pp. 66–114, 2022.
- [8] A. Savvas, M. S. Lizarralde, and P. T. Marsoit, “Development of quantum neural networks for complex data classification,” *Journal of Computer Science and Research (JoCoSiR)*, vol. 1, no. 4, pp. 132–139, 2023.
- [9] A. Chalumuri, R. Kune, and B. Manoj, “A hybrid classical-quantum approach for multi-class classification,” *Quantum Information Processing*, vol. 20, no. 3, p. 119, 2021.
- [10] J. S. Nelson and A. D. Baczewski, “An assessment of quantum phase estimation protocols for early fault-tolerant quantum computers,” *arXiv preprint arXiv:2403.00077*, 2024.
- [11] J. Jiang and D. Wang, “Qpase: Quantum-resistant password-authenticated searchable encryption for cloud storage,” *IEEE Transactions on Information Forensics and Security*, 2024.
- [12] B. Weder, U. Breitenbücher, F. Leymann, and K. Wild, “Integrating quantum computing into workflow modeling and execution,” in *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, pp. 279–291, IEEE, 2020.
- [13] B. Kantsepolsky, I. Aviv, R. Weitzfeld, and E. Bordo, “Exploring quantum sensing potential for systems applications,” *IEEE Access*, 2023.
- [14] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec, “Graphrnn: Generating realistic graphs with deep auto-regressive models,” in *International conference on machine learning*, pp. 5708–5717, PMLR, 2018.
- [15] G. Chen, S. A. Fulling, H. Lee, and M. O. Scully, “Grover’s algorithm for multiobject search in quantum computing,” in *Directions in Quantum Optics: A Collection of Papers Dedicated to the Memory of Dan Walls Including Papers Presented at the TAMU-ONR Workshop Held at Jackson, Wyoming, USA, 26–30 July 1999*, pp. 165–175, Springer, 2001.
- [16] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, “Tight bounds on quantum searching,” *Fortschritte der Physik: Progress of Physics*, vol. 46, no. 4-5, pp. 493–505, 1998.
- [17] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, “Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices,” *Physical Review X*, vol. 10, no. 2, p. 021067, 2020.
- [18] S. E. Venegas-Andraca, “Quantum walks: a comprehensive review,” *Quantum Information Processing*, vol. 11, no. 5, pp. 1015–1106, 2012.
- [19] A. Sayar, S. S. Arslan, and T. Çakar, “SSQEM: Semi-Supervised Quantum Error Mitigation,” in *2022 7th International Conference on Computer Science and Engineering (UBMK)*, pp. 474–478, 2022.