International Journal of Theoretical Physics (2025) 64:18
https://doi.org/10.1007/s10773-024-05874-7

RESEARCH

An Advanced Collaborative Routing Algorithm for Optimizing Entanglement and Resource Efficiency in Quantum Networks

Zhongrui Huang[1] · Hong Lai[1] · Linchun Wan[1]

Received: 3 December 2024 / Accepted: 26 December 2024 / Published online: 11 January 2025

Abstract
The entanglement routing algorithm facilitates multi-path communication among users in quantum networks via quantum entanglement. While previous research primarily concentrated on maximizing throughput, practical implementations also necessitate consideration of service rates and quantum resource utilization. This paper presents a novel routing algorithm called Collaboratively Optimized Selection of Paths (COSP) that balances expected throughput, service rate and quantum resource utilization. The COSP algorithm integrates resource efficiency as a novel routing metric alongside the Monte Carlo Tree Search method to optimize resource allocation policies. To address high concurrency and conflicting requests, COSP implements a strategic grouping of user requests coupled with a "fail-retransmit" mechanism to ensure fairness. Simulation results reveal that COSP significantly outperforms traditional greedy resource allocation strategies, boosting handling of high concurrency scenarios by 50%, also significantly improving the service rate by up to 55%.

Keywords Quantum network · Entanglement routing · Monte Carlo tree search · Resource allocation · Service rate

1 Introduction

Quantum networks, which include a comprehensive array of hardware and software infrastructures for the generation and transmission of quantum bits and information, have spearheaded advances in transmission efficacy and security. In less than two decades, these networks have significantly influenced both quantum communication and computing. In communications, they have established a groundbreaking paradigm known as Quantum Key Distribution (QKD) [1, 2], which provides a robust framework for facilitating long-range QKD connections among multiple users [3]. This innovation has led to pioneering projects such as the Chinese quantum satellite [4] and the 5GUK test network [5], making long-distance quantum communication a practical reality. Beyond communications, the unique capabilities of quantum networks support sophisticated processes in quantum computing, such as distributed consensus [6], secure remote computation [7], and constructing non-local games [8]. showcasing their transformative potential in advancing the field. In recent years,

✉ Hong Lai
hlai@swu.edu.cn

[1] College of Computer and Information Science, Southwest University, Chongqing 400715, China

the unparalleled advantages demonstrated by quantum communication networks have solidified their transformative impact on the field of quantum information. This has given rise to the concept of the "quantum supremacy of the quantum Internet", a term that highlights the unique properties and capabilities achievable only through quantum networks capabilities that far exceed the limitations of classical networks. These distinctive features underscore the central role quantum networks play, not only in advancing communication and computing but also in defining the future of quantum information science [9].

Entanglement-based quantum networks possess distinctive characteristics. For instance, device-independent QKD offers enhanced security [10, 11], by quantum mechanics such as the "no-cloning theorem" [12] and the "no-communication" theorem [13]. Similarly, quantum teleportation is realized through the physical law of quantum entanglement [14]. Despite these advantages, entanglement-based quantum networks come with their stark challenges: the establishment of entanglement is probabilistic, with success rates decreasing exponentially over distance [15]. Furthermore, entanglement tends to persist for a relatively short duration, approximately 1.46 s [16], posing hurdles to the development of large-scale, long-distance entanglement networks utilizing optical fibers and processors. As research on entangled quantum networks intensifies, current studies discuss issues concerning strong and weak entanglement relationships, network stability, and the effect of noise on quantum channel capacity, all of which affect the development of quantum Internet [17, 18].

Quantum repeaters provide a partial solution by facilitating the extension of entanglement across greater distances, as illustrated in Fig. 1, through performing quantum swapping in repeaters, repeaters act as intermediaries, allowing the establishment of entanglement between users who are far apart [19, 20]. In quantum networks comprising processors and repeaters, the uncertainty and fragility of entanglement render classical routing protocols like Open Shortest Path First [21] and Routing Information Protocol [22] inapplicable. Moreover, the scarcity and preciousness of quantum resources further complicate the establishment of

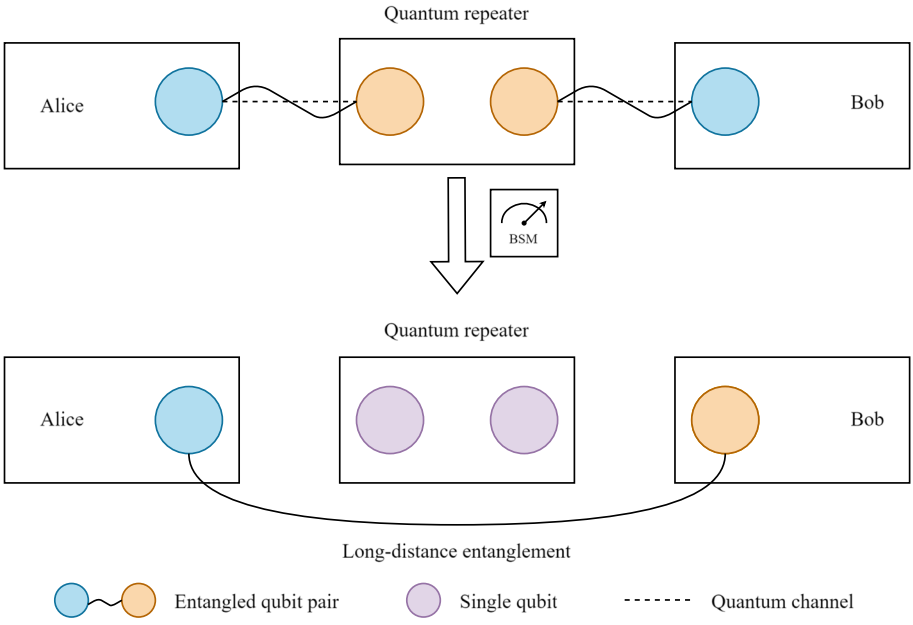

Fig. 1 Entanglement swapping in quantum network. By performing the Bell state measurement(BSM) in the quantum repeater, Alice and Bob share a pair of entangled qubits

effective long-distance entanglement, which is known as entanglement routing. In large-scale networks with high user concurrency, efficiently utilizing quantum resources for entanglement routing poses a significant contemporary challenge.

The study of entanglement routing problems starts with the lattice model introduced by Leone et al. [23], and is later extended by Gyongyosi et al. , who demonstrate the advantages of entanglement routing over fully connected structures [24]. However, these early models were limited by their fixed topologies, which lack flexibility for broader applications. Van's research expands network models to arbitrary structures, yet bandwidth constraints and low service rates persisted [19]. A breakthrough by Chen et al. improved node and path capacities and introduced a time-slot mechanism to enhance concurrency [25], but their focus on single-path throughput neglected fairness and global optimization. Recent advances have expanded the strategies in entanglement routing for tackling the challenges inherent in quantum networks. Linh et al. applied deep reinforcement learning for request sequence [26]. Gyongyosi et al. employ queue theory to define the quantum repeater service rate as a routing metric. Based on network conditions, they then select the routing mode, choosing between deterministic routing and adaptive routing [27]. And Zeng et al. implement linear programming to optimize path selection [28]. However, these methods still fall short in fully addressing issues of resource utilization and fairness. Further progress includes the optimization of entanglement swapping sequences [29], integrating quantum memory [30], characterizing communication capacity [31] and employing innovative routing metrics [32, 33]. Lai et al. have made significant strides in boosting resource efficiency through entanglement compression [34]. Despite these technological gains, persistent issues such as resource competition and managing high concurrency remain formidable challenges. Thus, these ongoing concerns highlight the urgent need for a comprehensive entanglement routing algorithm that encompasses a broad spectrum of resource allocation strategies.

Our research is dedicated to finding an optimal balance between network performance and fairness, aiming to extend services to a broader user base while minimizing the impact on communication throughput and the consumption of quantum resources in routing. We develop an innovative algorithm that incorporates a "fail-retransmit" mechanism through adjustments to the original time slot, and enhances the efficiency of quantum resources by emphasizing resource utilization as a key metric, and promotes fairness by limiting the total bandwidth available to each pair of nodes. This prevents any single request task from monopolizing excessive resources. Furthermore, our algorithm introduces a cooperative resource allocation strategy that optimizes network throughput, thereby ensuring efficient and equitable access to quantum network services.

Our simulation experiments validate the efficacy of our algorithm: in high concurrency scenarios, it achieves a 50% increase in expected throughput compared to the existing algorithm [25], and a 55% improvement in service rates. Additionally, the ungrouped version of our algorithm enhances resource utilization in most cases with low concurrency settings.

Our contributions are summarized as follows:

Enhanced routing algorithm We introduce a new routing algorithm that utilizes a novel metric focused on resource utilization. By expanding the candidate set with Yen's algorithm, we aim to mitigate allocation conflicts and enhance overall performance while maintaining fairness.

Cooperative resource allocation We propose a novel cooperative resource allocation method which transforms the allocation problem into a cooperative game in order to obtain an effective approximate solution using the Monte Carlo Search Tree algorithm.

Multilevel request queue design To address the challenges of high concurrency and conflicting requests, we develop a multilevel request queue. This system categorizes resources

Table 1 Variables and Functions in the research

Variables and Functions	Descriptions
L_i	Length of link i
$P_{est}(j)$	Entanglement establishment success probability for link j
α	Channel parameter
e	Routing metric
$\mathcal{V}$	Node set in graph G
$\mathcal{E}$	Edge set in graph G
$\mathcal{C}$	Channel set in graph G
p	Single candidate path
$\mathcal{P}$	Candidate path set of the communication source-destination pairs
W_p	Link width of path p
$cvr_{src,dst}$	Coverage rate constraints of the chosen source-destination pairs (src, dst)
$rdn_{src,dst}$	Redundancy rate constraints of the chosen source-destination pairs (src, dst)
Q_n	Number of available qubits at node n
A_j^i	Probability that exactly i links succeed in the j-th hop
B_j^i	Probability that an i-width entanglement path is established over the first j hops.
q	Entanglement swapping success probability in quantum network G
E_p	Expected throughput of path p
U_p	Expected quantum resource utilization of path p
h_p	Hops of path p
x_p	Estimation of quantum resource occupancy in path p
$Q_{e(u,v)}$	Number of qubit available between node pair (u, v)
$\mathcal{N}_s$	Set of source nodes in current sub-queue
$\mathcal{N}_d$	Set of destination nodes in current sub-queue
V	Expected reward value in the whole Markov decision process
$\mathcal{O}$	Other agents' action set
$S(x, a)$	Expected reward function from the state x taking an action a
$R(g, p)$	Reward of allocating resource for path p in the residual graph g
$P(g, p, o, g')$	Probability that other agent adopts strategy o when $g \xrightarrow{p} g'$
$P(x, a, x')$	Probability that an action a taken by the current agent causes $x \xrightarrow{a} x'$
μ_s	Service rate
N_{SD}	Number of total S-D pairs
N_0	Number of S-D pairs that shares 0 entanglement pair in this time slot
C_Q	Capacity of each sub-queue
N_Q	Number of sub-queues
$d(i, j)$	Distance between node i and node j
UCB1_i	UCB1 value of MCTS node i

based on multiple factors, prioritizes requests through a priority queue, and integrates a "fail-retransmit" mechanism to efficiently manage resource competition.

2 Network Model

Mirroring the layer architecture of classical network communication protocols [35], quantum networks similarly employ a hierarchical protocol structure, and the problem of entanglement routing concentrates on the physical and link layers [36, 37]. The behavior of nodes on the physical layer, in terms of Bell measurements and establishing entanglement through channels, is controlled by the classical server's routing algorithm and resource allocation policy on the link layer. Thus, it is critical to improve network performance by implementing a suitable routing algorithm with efficient resource allocation strategies. Previous research has established the methodology, scope, and model architecture of the entanglement routing problem [19, 25].

2.1 Network Components

Quantum processors Quantum processors enable users to access the quantum network. These processors are specifically designed to manage quantum information for various applications, including the storage and measurement of qubits. Within the network, processors are linked through quantum channels extending between them, creating an extensive network that supports the seamless exchange of quantum information.

Entanglement repeaters Due to the difficulty of directly establishing entanglement, repeaters serve as "bridges" in communication [19, 38]. Entanglement swapping is performed in the repeater and controlled by a classical computer. Because of the no-cloning theorem, untrusted repeaters can be used in the quantum network. In our model, processors are equipped with all the functionalities of repeaters, including participation in entanglement swapping. Therefore, both processors and repeaters are collectively referred to as nodes within the network.

Quantum channels The quantum channel functions as a conduit to establish entanglement across nodes within the network. Our model allows for parallel quantum channels linking the same pair of nodes. Each quantum channel has its independent success probability, and the probability is expressed as $P_{est}(i) = e^{-\alpha L_i}$, where L_i denotes the physical length of the channel i and α is a constant that varies based on the material properties of the channel. In real world implementations, these nodes are typically interconnected through optical fibers, providing a practical foundation for the establishment and maintenance of quantum entanglement.

A classical server In our model, a classical server plays a pivotal role in recording the network's topology and orchestrating the paths for requests. We base our discussions on the premise that communications occur between two users, simplifying multi-user interactions into a series of one-to-one communication requests. Even if the classical server is compromised or eavesdropped upon, the security of the quantum network remains intact. This resilience stems from the fact that the classical server has no knowledge of the qubits themselves, ensuring that the confidentiality of the quantum information is preserved regardless of the classical server's security status.

In our approach, a quantum network is defined by the notation $G\langle \mathcal{V}, \mathcal{E}, \mathcal{C} \rangle$, where $\mathcal{V}$ denotes the network nodes. Each node is equipped with a finite qubit memory, indicating a capacity on the number of qubits it can handle simultaneously. $\mathcal{E}$ symbolizes the connections between the nodes, represented as edges within the graph. These edges signify quantum channels linking pairs of nodes, with each connection's capacity inherently limited. $\mathcal{C}$ encompasses all channels

in the quantum network, specifying link utilization and the status of entanglement across the network. This graphical representation facilitates a structured analysis of the architecture and operational dynamics of the quantum network.

2.2 Entanglement establishment process

Given the challenge posed by decoherence, which limits the maintenance time of the entanglement to approximately 1.4s [16], our approach incorporates the use of time slots to establish the entanglement synchronously [25]. As illustrated in Fig. 2, each time slot is comprised of four distinct phases:

Phase 1 (P1) The communicating nodes submit requests to the classical server, specifying source and destination nodes along with their communication priority. The classical server aggregates these requests, storing and assigning them to different time slots in the format of source-destination (S-D) pairs. The classical server maximizes the satisfaction of these communication requests by designing a reasonable schedule of requests.

Phase 2 (P2) This phase focuses on identifying feasible paths for establishing entanglement. By employing an efficient entanglement routing algorithm, the system selects a set of paths expected to yield high availability. Once the resource allocation is completed, nodes attempt to create entanglement with adjacent nodes through quantum channels. Despite the typically low success rate of entanglement for each entanglement establishment attempt, once the routing path is determined, the nodes make repeated attempts to establish entanglement for a specified period of time.

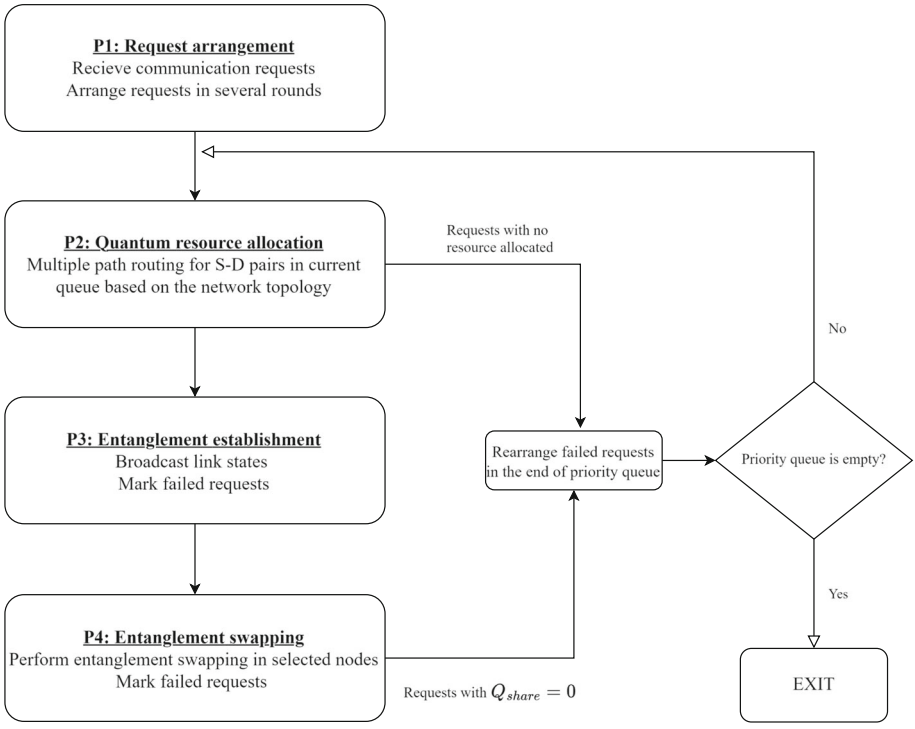

Fig. 2 Flow chart of phases in Q-COSP algorithm

Phase 3 (P3) In this phase, each node broadcasts its link status to adjacent nodes via classical channels. Due to the fleeting nature of entanglement maintenance, node statuses are not shared network-wide. This stage lays the groundwork for the ensuing entanglement swapping process.

Phase 4 (P4) Nodes execute entanglement swapping based on the shared status information. For paths where entanglement is successfully established, long-distance entanglement is achieved through quantum swapping. Conversely, for paths where entanglement fails, quantum swapping offers a secondary opportunity to circumvent the unsuccessful links. By rerouting through alternative, unoccupied edges with successful entanglement, the network can rectify broken paths. This final step solidifies the establishment of end-to-end entanglement.

3 COSP Algorithm

3.1 Algorithm Description

Previous research did not adequately consider network capacity during the routing process, leading to concerns over fairness and poor resource utilization in scenarios with high concurrent communications. The routing algorithms presented in prior works adopt a greedy approach that prioritizes paths based on specific routing metrics. While efficient in some cases, this strategy might miss the most effective global path configuration, potentially compromising overall network performance. Moreover, this greedy method can allocate resources to inefficient paths with low expected throughput, further reducing network efficiency. Optimizing total expected throughput and resource utilization while ensuring fairness and minimizing competition is challenging and classified as an NP-Hard problem. Addressing these complexities requires innovative solutions for effective quantum network routing.

This paper proposes a novel approach to address these issues. First, communication requests that might exceed network capacity or conflict are divided into priority queues to avoid initial conflicts. Second, in the resource allocation process, instead of directly selecting the optimal path for each S-D pair, our algorithm selects a set of candidate paths for each request using a path selection algorithm. From these candidates, the algorithm chooses the best combination of paths to maximize network throughput while considering both fairness and capacity. This process involves coordinating and compromising among requests to satisfy as many as possible. We call this algorithm Collaboratively Optimized Selection of Paths (COSP).

3.2 P1 Algorithm of Q-COSP

Previous research [19] has shown that in inadequately resourced entanglement networks, some users experience significantly reduced bandwidth availability or may even fail to complete their communications during peak concurrency periods. To address this issue, our algorithm introduces a multilevel queue in P1. Requests are organized into a multilevel queue based on their priority and the source-destination distance, enhancing fairness in resource distribution. As depicted in Fig. 3, to reduce potential conflicts in resource allocation, the algorithm prioritizes the furthest and closest S-D pairs among requests of the same prior-

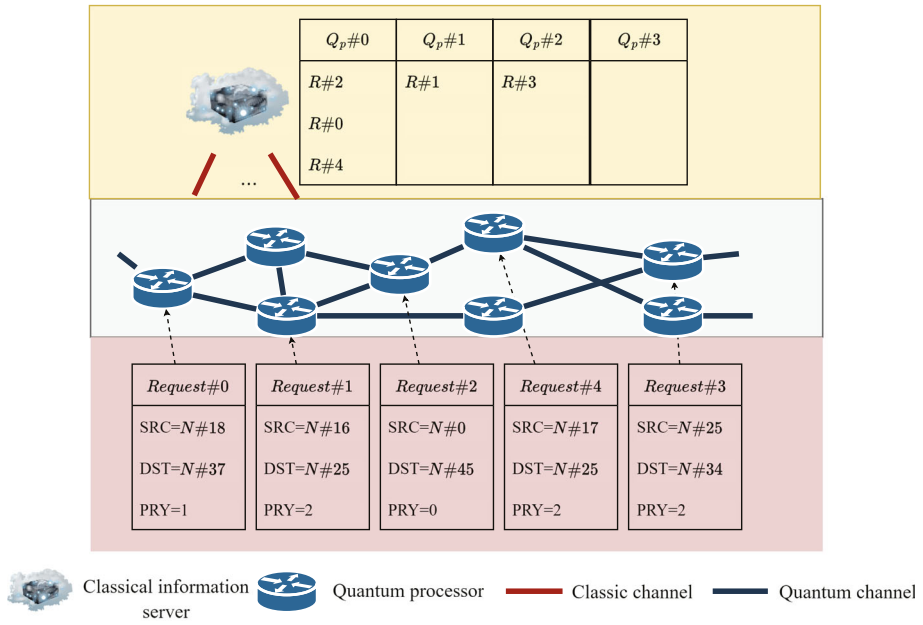

Fig. 3 The request queue in P1. Red part represents the behavior of the user terminal, where communication requests are sent to the classical server. The gray part represents the quantum repeater network, which also serves as a relay between the user and the server during the process. The yellow part represents the classical server, which organizes incoming user requests and schedules them into a multilevel queue, while leaving an empty sub-queue for the implementation of failure retransmission

ity. Each sub-queue has a length limit to alleviate concurrent stress and optimize network throughput efficiently. For details on sub-queue length and level settings, see Appendix A.

Algorithm 1 P1 requests arrangement algorithm.

Input: $srcDstPairs = \text{ListOf}\ \langle src, dst, priority \rangle$
Output: A queue array Q
$srcDstPairs \leftarrow \texttt{SortByPriority}\ (srcDstPairs)$
while $srcDstPairs \neq \varnothing$ **do**
  $P \leftarrow srcDstPairs[0].priority$
  $C \leftarrow \{sdpair \in srcDstPairs | sdpair.priority == P\}$
  $srcDstPairs \leftarrow srcDstPairs - C$
  $C \leftarrow \texttt{SortByDistance}(C)$
  while $C \neq \varnothing$ **do**
    $\texttt{ArrangeRequest}(C.first)$
    $\texttt{RemoveFirst}(C)$
    $\texttt{ArrangeRequest}(C.last)$
    $\texttt{RemoveLast}(C)$
  end
end

The P1 scheduling algorithm seeks to distribute requests with conflicting nodes across different queues. This strategy significantly reduces resource competition and mitigates the risk of starvation, ensuring a fairer distribution of quantum resources. Moreover, this approach promotes balanced consumption of quantum resources across each queue, contributing to more efficient network utilization. Additionally, it ensures an equitable allocation of computational time for the routing process. The pseudocode for the P1 algorithm is presented in Algorithm 1, encapsulating these operational principles.

Procedure ArrangeRequest

```
/* This function is used to put the request into queues.
*/
Input: srcDstPair = ⟨src, dst, priority⟩
for i ← 0 to Q.size do
    A ← ∅
    foreach element e in Q[i] do
        A.add(e.src)
        A.add(e.dst)
    end
    if Q[i] is not full & src and dst nodes are not occupied then
        Q[i].enqueue(srcDstPair)
        return
    end
end
```

3.3 P2 Algorithm of Q-COSP

The P2 algorithm begins by selecting a set of paths for each S-D pair, known as the candidate path set. This set is crucial as subsequent resource allocation in P2 will follow paths from these sets. To devise effective routing strategies, paths in the candidate path set are generated using two algorithms. Firstly, a subset of paths is identified using Dijkstra's algorithm to find optimal path combinations independent of other S-D pairs' candidate path sets. Following this, Yen's algorithm is employed by P2 to generate additional deviation paths within the candidate path set based on the initial paths, aimed at avoiding potential conflicts [39].

During the generation of deviation paths, a step of linear programming is introduced to estimate potential resource consumption accurately. This step involves representing the probability of path selection as a real value between 0 and 1, rather than a Boolean decision variable. This adjustment helps deviation paths identify resource-abundant alternatives, mitigating resource competition.

Finally, the P2 algorithm determines the optimal allocation of quantum resources from each S-D pair's candidate set, thereby identifying the optimal path sequence. Given the immense search space, exhaustive traversal of each sequence is impractical. However, approximation algorithms, such as Monte Carlo Tree Search (MCTS), have proven effective in reducing search complexity. MCTS, renowned for optimizing decisions in game theory, can theoretically be applied to any domain defined by state-action dynamics where simulations predict outcomes. Figure 4 illustrates the general process of the P2 algorithm with an example scenario.

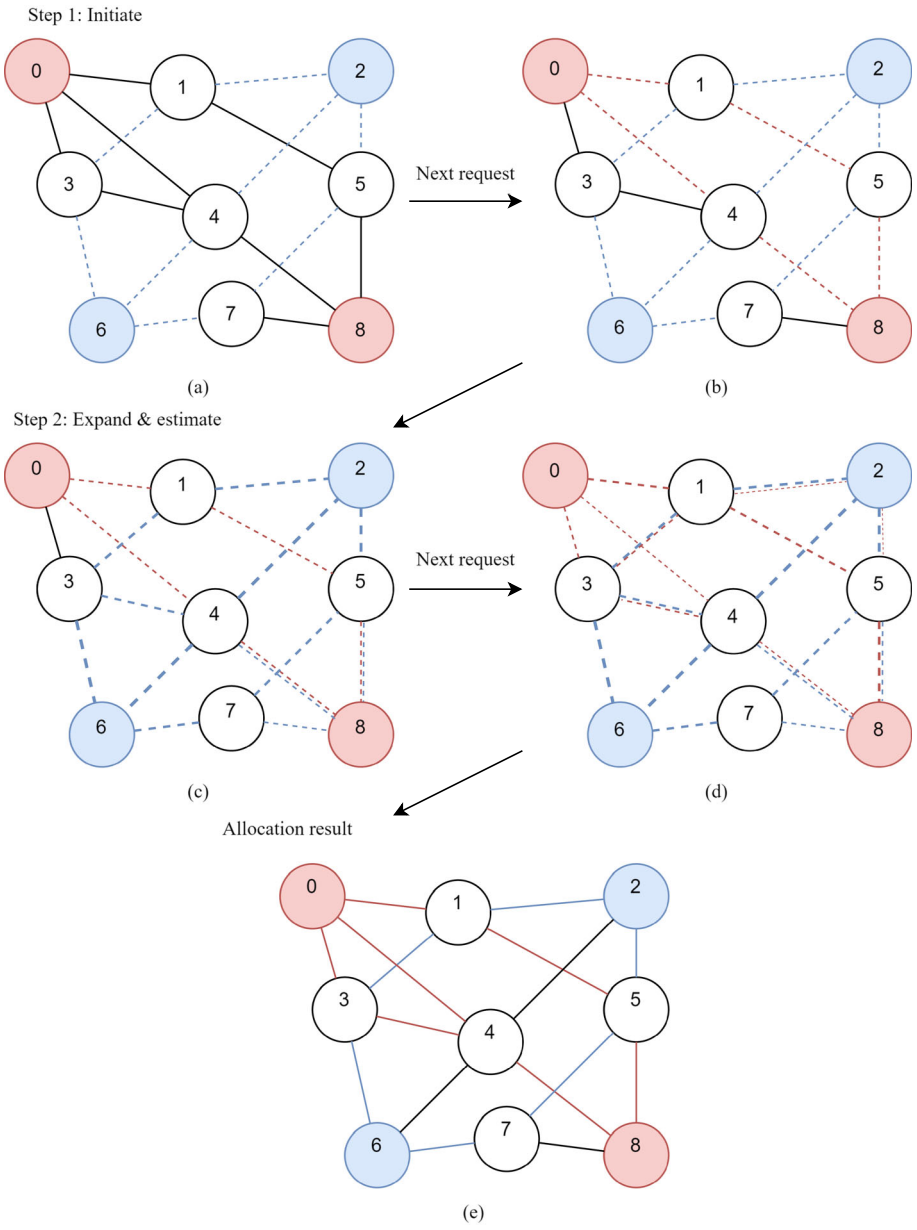

Fig. 4 An example of P2 algorithm. The initial path selection for the S-D pair of (N#0, N#8) and (N#2, N#6) in the graph is represented by Fig. 4a and b respectively. The optimal paths that do not overlap in the graph are selected to form the basis of the candidate path set. Figure 4c and d depict the candidate path set expansion and estimation for (N#0, N#8) and (N#2, N#6) respectively. The dashed lines represent the corresponding edges, and the nodes are covered by the set of candidate paths for the group. The width of the dashed lines is used to represent the estimation of the resource consumption of the paths. Wider dashed lines indicate a higher probability of the path's quantum resources being consumed. Figure 4e denotes the resource allocation scheme. The final resource path is determined following the Monte Carlo Tree Search algorithm, and the classical server will attempt to establish entanglement along this result before the conclusion of Phase 2

3.3.1 Candidate Path Set Selection

The process of selecting candidate paths involves two steps: 1) Within each sub-queue of the multilevel request queue, the algorithm sequentially selects the optimal path for each S-D pair based on a chosen metric e (such as hops, distance or throughput). Once a path is chosen, nodes and edges along it, excluding the source and destination nodes, are not reconsidered in this step. This process continues within the queue until either no new candidate paths can be added for any S-D pair or the cumulative bandwidth of candidate paths reaches a specified control parameter value. 2) The remaining paths in the candidate path set aim to enhance throughput while managing contention and congestion. This approach resembles Yen's algorithm [39], where deviation paths are derived from previously selected paths. Hence, the initial path heap in Yen's algorithm consists of paths chosen in Step 1. In this process, the paths are selected in batches according to the order in the request queue.

Candidate path set size control To ensure efficient resource utilization and prevent exhaustive searches, as described in inequality (1), termination conditions in the aforementioned steps are governed by two parameters: "Coverage" Controls the termination condition in Step 1, ensuring the path selection process continues until sufficient coverage is achieved or the bandwidth limit is reached. "Redundancy" Controls the termination condition in Step 2, limiting additional paths to prevent excessive redundancy and optimize network efficiency. Unlike traditional routing parameters, these controls do not directly dictate the number of paths but manage the total bandwidth allocated to candidate paths. This approach facilitates a more balanced allocation of quantum resources, promoting both efficiency and fairness in quantum network operations. For detailed parameter settings, please refer to Appendix A.

$$\begin{aligned} \sum_{p \in \mathcal{P}_{step1,(src,dst)}} W_p &\geq cvr_{src,dst} \cdot \min(Q_{src}, Q_{dst}) \\ \sum_{p \in \mathcal{P}_{step2,(src,dst)}} W_p &\geq rdn_{src,dst} \cdot \min(Q_{src}, Q_{dst}) \end{aligned} \tag{1}$$

Routing metrics An essential consideration in the routing process is the routing metric e. In this context, quantum resource utilization serves as the routing metric. The utilization of quantum resources on a single path U_p is defined by the expected number of entanglement pairs established between src and dst through that path and the number of qubits consumed on that path. The E_p is the expected number of entanglement pairs established between user pairs, and it can be calculated according to [25]. For a path $p(w, h)$ with bandwidth w and total hops h, the formal definition of utilization $U(p)$ is as follows:

$$\begin{aligned} A_j^i &= \binom{W_p}{i} P_{est}(j)^i (1 - P_{est}(j))^{W-i} \\ B_1^i &= A_1^i \\ B_j^i &= B_{j-1}^i \sum_{k=i}^{W_p} A_j^k + A_j^i \sum_{k=i+1}^{W_p} B_{j-1}^k \\ E_p &= q^h \sum_{i=1}^{W_p} i \cdot B_h^i \\ U_p &= \frac{E_p}{2 \cdot h \cdot W_p} \end{aligned} \tag{2}$$

where i is an integer in$\{1, 2, \ldots, W\}$ and j is an integer in $\{1, 2, \ldots, h\}$. A_j^i denotes the probability that exactly i links succeed in the j-th hop. B_j^i denotes the probability that an i-width entanglement path is established over the first j hops. E_p represents the expected throughput through the path p, and U_p is the utilization of quantum resources on the path p. In conjunction with the above, Algorithm 2 gives the details for the step 1 of the path set selection.

Algorithm 2 P2 path set selecting algorithm in step 1.

Input: Q_c= Queue of $\langle src, dst, priority \rangle$
Output: A set of path $\mathcal{P}$
for $i \leftarrow 1$ **to** Q_c.size **do**
  $q \leftarrow Q_c$.dequeue()
  $cvr[q] \leftarrow (\log_2(1 + N_Q) - 1) \cdot \mathbf{E}(P_{est}) \cdot \frac{d_{\max(G)}}{d_{src,dst}}$
  $rdn[q] \leftarrow 7.0$
  Q_c.enqueue(q)
  Remove $q.src$ and $q.dst$ from G
end
$i \leftarrow 0$
$\mathcal{P} \leftarrow \varnothing$
while $Q_c \neq \varnothing$ **do**
  $q \leftarrow Q_c$.dequeue()
  Add $q.src$ and $q.dst$ to G
  $p \leftarrow$ Dijkstra($q.src, q.dst$)
  if $p \neq \varnothing$ **then**
    foreach e in p **do**
      Remove edge e from G
    end
    $\mathcal{P}$.add(p)
    $cvr[q] \leftarrow cvr[q] - (\frac{W_q}{\min(Q_{q.src}, Q_{q.dst})})$
    if $cvr[q] \leq 0$ **then**
      continue
    end
    else
      Remove $q.src$ and $q.dst$ from G
      Q_c.enqueue(q)
    end
  end
  else
    continue
  end
end

Resource consumption estimation After each S-D pair selects its candidate path, the algorithm must assess the likelihood of each path in the candidate path set being chosen. This evaluation contributes to resource consumption estimation, ensuring that nodes with potentially high resource consumption are avoided as subsequent S-D pairs expand their

candidate path sets. This ensures the integrity of the candidate path set. To address this, linear programming is introduced to solve the problem. Here, we define decision variables $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ where m is the number of paths in the path set $\mathcal{P}src, dst$, and x_i represents the ratio of resource consumption to the maximum bandwidth of the path p_i. The represent resource occupancy as a continuous form without introducing complex calculations. For each path p_i , the objective function is $\sum i = 1^m x_i E_{p_i}$. The objective of linear programming is to maximize this function. Thus, the linear programming formulation for the resource occupancy estimation problem is as follows:

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{m} E_{p_j} x_j \\
\text{s.t.} \quad & \sum_{j:e(u,v)\in p_j} x_j W_j \le Q_{e(u,v)} \\
& \sum_{j:u\in p_j} 2 \cdot x_j W_j + \sum_{i:u\in p_i} x_i W_i \le Q_u \\
& x_j \in [0,1] \\
& \forall e(u,v) \in p, \forall p \in \mathcal{P}_{(src,dst)} \\
& \forall u \in p_j, u \in \mathcal{P} - (\mathcal{N}_s \cup \mathcal{N}_d) \\
& \forall u \in p_i, u \in \mathcal{N}_s \cup \mathcal{N}_d \\
& j = 1, 2, \ldots, m
\end{aligned} \tag{3}
$$

where $\mathcal{N}_s, \mathcal{N}_d$ represent the set of source nodes and the set of destination nodes in all paths respectively. In the subsequent routing process, quantum resources on edges and nodes are deducted as a result of linear programming. Thus, during the routing process, the S-D pair that is closer to the head of the request queue will obtain more adequate resources in the residual graph. The behavior of step 2 can be summarized as shown in Algorithm 3. In the next step, the algorithm searches for the resource allocation scheme with highest expected throughput from the set of candidate paths.

3.3.2 Resource Allocation with Monte Carlo Tree Search

The candidate path sets described above effectively narrow down the search space for optimal resource allocation, transforming the abstract problem into a concrete challenge of finding the best combination of candidate paths. To ensure fairness, the allocation process can be likened to a sequential game for S-D pairs. During each turn, an active S-D pair selects at most one path from its path set and allocates quantum resources accordingly. Once a path is chosen, that S-D pair is excluded from subsequent resource allocation processes. It is apparent that this allocation process resembles a finite-horizon Markov Decision Process (MDP) structured like a decision tree. A general MDP framework can be expressed using the following formulas [40]:

$$
\begin{aligned}
V^*(x) &= \max_{a\in\mathcal{A}} S^*(x,a) \\
S^*(x,a) &= R(x,a) + \sum_{x'\in\mathcal{X}} P(x,a,x') V^*(x)
\end{aligned} \tag{4}
$$

Algorithm 3 P2 path set selection algorithm step 2.

Input: $\langle src, dst \rangle$, G, $\mathcal{P}_{src,dst}$
Output: A extended set of path $\mathcal{P}_{src,dst}$
create a priority queue H
$L_p \leftarrow \varnothing$
foreach p in $\mathcal{P}_{src,dst}$ **do**
  $L_p[k] \leftarrow p$
  $rdn[src, dst] \leftarrow rdn[src, dst] - \frac{W_p}{\min(Q_{src}, Q_{dst})}$
end
$k \leftarrow 1$
while $rdn[src, dst] > 0$ **do**
  for 0 **to** $L_p[k].size - 2$ **do**
    $spurNode \leftarrow L_p[k-1][i]$
    $rootPath \leftarrow L_p[0:i]$
    Remove edges and nodes in $rootPath$ before $spurNode$ from G
    $spurPath \leftarrow$ `Dijkstra`$(spurNode, dst)$
    $totalPath \leftarrow rootPath + spurPath$
    if $totalPath.size < 2$ **then**
      break
    end
    if $totalPath \notin H$ **then**
      H.`enqueue`$(totalPath, E_t)$
    end
  end
  $k \leftarrow k + 1$
  if $H = \varnothing$ **then**
    break
  end
  $candidatePath \leftarrow H$.`peek`()
  $\mathcal{P}_{src,dst}$.`add`$(candidatePath)$
  $rdn[src, dst] \leftarrow rdn[src, dst] - \frac{W(candidatePath)}{\min(Q_{src}, Q_{dst})}$
end

where the function $V^*(x)$ gives the maximum expected reward from the initiate state x, and $S^*(x, a)$ is the maximum expected reward function from the state x taking an action a. In the routing problem, for each player (S-D pairs), the common goal is to maximize the total expected throughput. If the decision policy for each S-D pair is (4) is also expressed as follows:

$$V^*(g) = \max_{\pi \in \nu(\mathcal{P}_c)} \max_{o \in \mathcal{O}} \sum_{p \in \mathcal{P}_c} S(g, p, o)\pi_p$$
$$S^*(g, p, o) = R(g, p) + \sum_{g' \in \mathcal{G}} P(g, p, o, g')V^*(g') \tag{5}$$

These formulas transform the objective of a general MDP tree into a solution for the quantum resource allocation problem. In (5), π represents the policy of the current active S-D pair, which is a vector of probabilities. The probability simplex over the path set for the active S-D pair, denoted $\nu(\mathcal{P}_c)$, defines the domain of π. Similarly to a Minimax tree,

our goal is to construct the strategy tree π^* with the highest global value quantity V^* for each agent, where $\mathcal{O}$ represents the opponents' path set, and o is a sequence of paths chosen sequentially by other S-D pairs. However, computing this expectation can be challenging when the path set is large. In such cases, the Monte Carlo Tree Search (MCTS) algorithm serves as a valuable tool for approximating an optimal solution. Thus, the MCTS method is used in improving routing design, which is an incremental iterative heuristic search algorithm for decision tree to progressively approximate the optimal policy within limited resources and time.

MCTS enhances routing design through an incremental iterative heuristic search in decision trees, aiming to progressively approximate the optimal policy within limited resources and time. The transformation from the MDP tree to MCTS is straightforward: The state set in this MCTS process corresponds to the residual graph set, and the action set comprises the candidate paths available to the current S-D pair. Transitioning involves actions of path selection: Selected paths are removed from the candidate path set, and resources are adjusted in the residual graph. The estimated expected throughput serves as the reward function for each action. The MCTS algorithm operates through several key steps: selection, expansion, simulation, and backpropagation. The overarching framework of MCTS for this problem is detailed in Appendix B.

3.4 P4 Algorithm of Q-COSP

The nodes on the path broadcast the link state within K hops, after these nodes have performed entanglement swapping within the node based on the received broadcast information. Similarly to [25], if the recovery path is allowed, the algorithm could calculate a set of paths $\mathcal{R}$ between the failed nodes, but this action will consume more quantum resources. Another method to handle failed communication is therefore simpler: According to the design in Fig. 2, for the failed S-D pairs who were unable to establish any successful entanglement pairs, will have one more chance in the last request queue, which collects all of the failed pairs. This "fail-retransmit" mechanism has improved the availability of the entanglement network.

3.5 Complexity Analysis

For the P1 algorithm, time complexity is $O(N_{SD}\lg(N_{SD}))$. The P2 algorithm has a high time complexity: We assume the max width of a path is W_m, and the max hop limit for a path is h_m. The complexity of the original Dijkstra's algorithm is $O(|\mathcal{V}|\lg(|\mathcal{V}|)+|\mathcal{E}|)$, in this algorithm, calculating new metric has a $O(h_m W_m)$ complexity. So Step 1 in P2 has the complexity of $O(N_Q\lg(N_Q)(|\mathcal{V}|\lg(|\mathcal{V}|)+h_m W_m|\mathcal{E}|))$. Step 2 has a time complexity of $O(N_Q h_m(|\mathcal{V}|\lg(|\mathcal{V}|)+h_m W_m|\mathcal{E}|))$. The final step in P2 has $O(N_Q^2\lg^2(N_Q)I h_m W_m)$, where I is the iteration count in MCTS process (Table 2).

4 Evaluation Result

In this paper, the network model is generated using the Waxman model to simulate the network topology. The graph is generated randomly based on given parameters E_p, E_d, $|V|$, q. In this model, the network facility has a limitation on the number of qubits per node, which is

uniformly distributed between 10 and 14. Similarly, quantum links have capacities independently and uniformly selected from 3 to 7, with the probability of successfully establishing an entanglement pair being $E_p \pm 0.01$.

We focus on evaluating three key metrics: expected throughput, resource utilization rate, and service rate. The expected throughput and resource utilization rate are defined in (2). The service rate is a metric that reflects the fairness of the algorithm, indicating the proportion of requests that have been allocated resources and have shared at least one entangled Bell pair during a communication round. The expression for this metric is as follows:

$$\mu_s = \frac{N_{SD} - N_0}{N_{SD}} \tag{6}$$

where N_0 is the number of the S-D pairs who shares 0 ebit in the previous process. These metrics measure a network's transmission performance, communication efficiency, and availability, respectively. As the number of concurrent S-D pairs increases, we are interested in both the performance of the algorithm and the carrying capacity of the network.

4.1 Methodology

In order to observe the performance of the algorithm under different concurrency scenarios, the simulation is designed as follows: first the topology is fixed and then S-D pairs are randomly selected in the graph. The number of concurrent S-D pairs $N_{S,D}$ increases from 2 to a maximum of 10, and the evaluation metrics are recorded during the process. . Finally, the previous process is repeated with different topologies to analyze algorithm performance. The parameters for the topologies are $|V| \in \{50, 100, 150, 200\}$, $q \in \{0.6, 0.7, 0.8, 0.9\}$, $E_d \in \{6, 7, 8\}$. In the comparison experiments, we compare the Q-COSP algorithm proposed in this paper and Q-COSP-G, which does not use a grouping strategy, with Q-CAST algorithm [25], Q-PASS algorithm [25], and the Greedy algorithm [41], which is based on a distributed greedy strategy, from the original work.

Simulation experiments are conducted using two distinct design approaches:

Concurrent user analysis This approach examines the relationship between the number of concurrent users and network performance within a fixed topology setting.

Topology comparison This approach evaluates the overall comparative performance of various routing algorithms across networks with different topologies while maintaining a consistent number of concurrent user pairs for equitable comparisons.

Table 2 Time complexity of COSP algorithm

Steps in COSP Algorithm	Time complexity
P1 algorithm	$O(N_{SD} \lg(N_{SD}))$
Preselection algorithm in P2	$O(N_Q \lg(N_Q)(\lvert\mathcal{V}\rvert \lg(\lvert\mathcal{V}\rvert) + h_m W_m \lvert\mathcal{E}\rvert))$
Candidate set selection in P2	$O(N_Q h_m(\lvert\mathcal{V}\rvert \lg(\lvert\mathcal{V}\rvert) + h_m W_m \lvert\mathcal{E}\rvert))$
MCTS algorithm in P2	$O(N_Q^2 \lg^2(N_Q) I h_m W_m)$

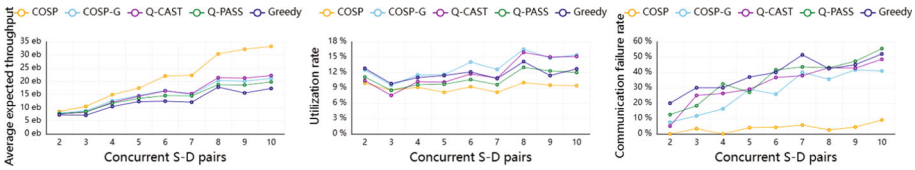

Fig. 5 Performance in the quantum network with $N = 50$, $E_d = 6$, $q = 0.8$

4.2 Result Analysis

Throughput analysis The COSP algorithm demonstrates remarkable performance under conditions of high concurrency. As illustrated in Fig 4, COSP achieves an average expected throughput that is 50% higher compared to previous algorithms when handling 10 concurrent user pairs. This significant improvement is further highlighted in Fig. 6, where COSP's throughput exceeds previous algorithms by 52% of previous work with the same number of user pairs. In contrast, the COSP-G algorithm, which omits the grouping strategy employed by COSP, shows greater sensitivity to varying levels of concurrent user pairs and fails to maintain a substantial throughput advantage over other methods at higher concurrency levels.

This divergence in performance can be attributed to two main factors: 1) In other algorithms, the absence of a fairness constraint results in prioritizing some pairs' requests over others as the number of concurrent S-D pairs increases. This leads to resource starvation for some pairs, hindering their communication opportunities. 2) Conventional algorithms often prioritize immediate path performance metrics, potentially opting for longer-distance paths with lower bandwidth as resources become scarce. This myopic approach reduces overall resource utilization and impedes throughput for a larger number of S-D pairs. Moreover, constraints imposed by quantum resource availability in entangled networks significantly limit throughput in scenarios with multiple concurrent S-D pairs. However, the introduction of grouping within the COSP algorithm effectively mitigates these challenges, resulting in substantially enhanced total throughput when managing multiple concurrent S-D pairs.

Resource utilization rates The COSP-G algorithm demonstrates effective resource utilization across various scenarios compared to other algorithms. As illustrated in Fig. 5, , the resource utilization rate of COSP-G surpasses that of Q-CAST by 20%. While the Greedy algorithm also shows high resource utilization at lower concurrency levels, its throughput consistently falls short compared to other algorithms. In high concurrency scenarios, as depicted in Fig. 6, the COSP-G algorithm's performance reaches 105% of previous methods. In contrast, the COSP algorithm exhibits bad resource utilization, achieving only 63% of Q-CAST's rate at high concurrency. This diminished performance is due to the "fail-

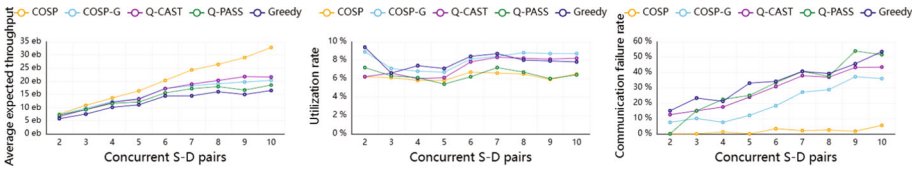

Fig. 6 Performance in the quantum network with $N = 100$, $E_d = 6$, $q = 0.9$

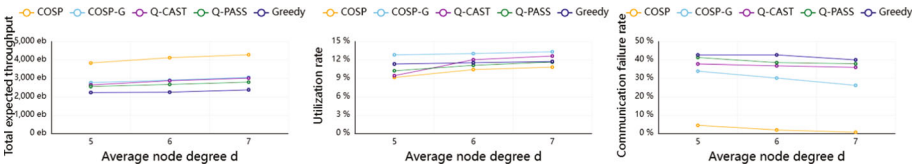

Fig. 7 Performance vs. average node degree

retransmit" mechanism within COSP,which decreases resource utilization each time a user pair fails to establish communication. This mechanism imposes additional overhead on the network, leading to lower average resource utilization by the algorithm. Similarly, introducing grouping and "fail-retransmit" mechanisms into other algorithm-based routing strategies can exacerbate overall resource inefficiency.

Service rates Our research has demonstrated a notable enhancement in the service rate metric. As depicted in Fig. 6, the failure rate with the COSP algorithm is only 19% compared to the Q-CAST algorithm, while the Q-COSP-G algorithm exhibits a failure rate of 84% relative to Q-CAST. This enhanced reliability stems from COSP's strategic approach during the initial candidate path selection phase. Specifically, COSP allocates a subset of user pairs by grouping source and destination nodes within the same group. This allocation strategy effectively reserves node resources for intended pair communications, thereby reducing overall failure rates.

Topology impacts As network size increases, illustrated in Fig. 9, all algorithms lose their distinct performance advantages in networks exceeding 150 nodes. Moreover, in a network of 200 nodes, quantum resource utilization may drop to less than 1%. This observation indicates that in large-scale quantum networks, network performance is more constrained by topology than by the choice of entanglement routing algorithm. Denser nodes and longer distances introduce additional challenges not easily mitigated by routing algorithms alone, leading to increased rates of path entanglement failure and significant impacts on network availability. Although the COSP algorithm mitigates these issues to some extent, the adverse effects of network size on algorithm performance remain pronounced and cannot be ignored.

The parameters E_p and q also significantly influence routing performance. As shown in Figs. 8, and 9, network performance across all metrics shows a linear relationship with both E_p and q. COSP demonstrates notable improvements in throughput and fairness, while the COSP-G algorithm excels in resource utilization without significant advantages in throughput. Additionally, as shown in Fig. 7, the success rate q has a more significant impact on network performance compared to the average node degree in the network. When q is reduced to 0.6, all routing algorithms cannot keep the network available, indicating that in practice it is necessary to offer an effective measurement device for each nodes.

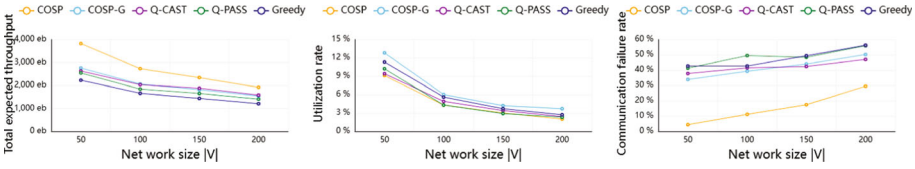

Fig. 8 Performance vs. network scale

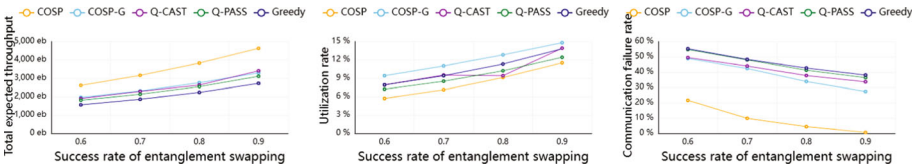

Fig. 9 Performance vs. swapping success rate

5 Conclusion and Future Work

5.1 Discussion

While our COSP algorithm generally enhances network availability, fairness, and throughput, there are areas where it requires further refinement:

Excessive time complexity The COSP algorithm exhibits significantly high computational complexity, particularly noticeable in its second-highest term. A promising solution to this challenge lies in adopting distributed search techniques, which can seamlessly integrate with the MCTS process, potentially reducing computational burdens significantly [42–44].

Routing parameter configuration: The configuration of coverage and redundancy parameters requires an in-depth understanding of the network topology. Figures 8 and 9 suggest that incorporating a broader scope of topological information and queue request details can lead to more precise calculations. In practical network parameter conditions, equipping the algorithm with more detailed topological insights and request information could make the routing process more responsive and effective.

5.2 Conclusion

In this study, we introduced a novel routing algorithm designed for entangled networks. It leveraged innovative routing metrics and the Monte Carlo Tree Search methodology to globally optimize resource allocation policies. Our approach enhanced communication protocols within the current network framework by mitigating the impact of high concurrency on network resources while boosting fairness and availability. Furthermore, the COSP algorithm, distinguished by its grouping policy and "fail-retransmit" mechanism, surpassed current routing algorithms in both expected throughput and service rate, proving especially beneficial for networks of small to medium scale. Comparative analysis reveals that the COSP algorithm significantly improved quantum resource utilization and the service rate of communication by 50% and 55% respectively, without adversely affecting throughput.

In the future, we will focus on refining the routing algorithm by incorporating a broader array of request and network topology data, including stability of entangled quantum networks, network fluctuation and expected duration of the user communication, to improve the calculation of coverage and redundancy parameters. We also plan to explore more realistic network models introducing a more refined channel noise model to accurately mirror real-world operations, incorporating comprehensive topology details to enhance the practical applicability of our routing solutions.

Appendix A: Control Parameter Setting

The routing control parameters employed in the experimental procedures presented in this paper are as follows:

Multilevel queue in P1 The capacity of each queue C_Q and the number of queue N_Q for multilevel queue are designed as follows:

$$\begin{aligned} C_Q &= \sqrt{\frac{|V|}{2}} \\ N_Q &= \max(3, \frac{N_{SD}}{C_Q}) \end{aligned} \tag{A1}$$

where $|V|$ and N_{SD} represent the number of nodes in the graph and the number of concurrent S-D pairs respectively.

Coverage and redundancy setting in P2: In order to be able to utilize the information brought by SD pairs and topology network parameters while keeping the calculation of routing parameters from being overly complex, the expressions are as follows:

$$\begin{aligned} cvr(src, dst, N_Q, G) &:= (\log_2(1 + N_Q) - 1) \cdot \mathbf{E}(P_{est}) \cdot \frac{d_{\max(G)}}{d_{src,dst}} \\ rdn(src, dst) &:= 7.0 \end{aligned} \tag{A2}$$

where N_Q is the number of S-D pairs in current queue, and $\mathbf{E}(P_{est})$ is the average value of the probability that a link in graph G will successfully establish an entanglement. Here, d denotes the distance; thus, $d(src, dst)$ and $d(\max(G))$ denote the distance between src and dst and the maximum distance between the nodes in the graph G, respectively.

Appendix B: Monte Carlo Tree Search Process

Selection: In this process, the MCTS algorithm traverses decision tree from the root to the tree node to be explored by an evaluation function, and select the node with the highest function value from child nodes. Repeat this step until the node whose child node has not been fully explored. The Upper Confidence Bound (UCB1) formula is typically used as the evaluation function to traverse the tree [45].The standard UCB1 function is

$$\mathrm{UCB1}_i = \bar{V}(g_i) + \lambda\sqrt{\frac{\ln(t)}{n_i}} \tag{B3}$$

where t represents the total number of simulations, and n_i is the number of times this node has been selected. λ is a constant, usually $\lambda = \sqrt{2}$. However, for the resource allocation problem, the algorithm is more focus on the optimal performance, therefore, instead of using the highest confidence bound for the average reward value in UCB1, as a modification, the highest confidence bound for the maximum reward value is used in the modified UCB1. As

a result, the evaluation function in this MCTS process has modified as follows:

$$\mathrm{UCB1}_i' = \max V(g_i) + \lambda \sqrt{\frac{\ln(t)}{n_i}} \tag{B4}$$

With all S-D pairs working together to identify the action for the highest expected throughput, it strikes a balance between exploration and exploitation. Furthermore, a challenge arises with the normalization of the function $V(g_i)$, which has a range of [0, 1], but does not correspond to the domain of expected throughput. Consequently, it is of vital importance to ensure the valid normalization of EXT values in order to succeed in solving this problem. In order to obtain approximate information about the EXT values, 100 random simulations are performed prior to the search, and the total EXT values are recorded for each simulation. Based on this statistical information, the normalized $V(g)$ is calculated in the following method:

$$V(g_i) = \omega(1 - 2(\frac{V_{max} - E_{g_i}}{V_{max}})^{0.5}) + (1 - \omega)(1 - 2(\exp(\frac{E_{g_i}}{V_{avg}})))$$

This normalized value function consists of two components, and the weights assigned to the two components are controlled through the parameter ω. The first part evaluates the effectiveness of the current strategy by measuring the difference between the current allocation strategy and the maximum value V_{max} from pre-experimental data. The second part addresses situations where the value may exceed the pre-experiment maximum by calculating the ratio to the average value of the pre-experiment V_{avg}, applying an exponential form for normalization. The advantage of this design is that it amplifies the normalized value of superior solutions, ensuring that high-expectation strategies still produce significant differences in value. This approach ultimately enhances the normalized value of the optimal solution.

Expansion In this process, a new child node is added to the tree under the selected node, which means the algorithm chooses a new status to evaluate from the previous choice. The new node is randomly selected from the next unexplored possible states.

Simulation During this process, the algorithm takes random actions from the expanded node until all user pairs can no longer take any actions (known as rollouts or playouts). The total expected throughput is then recorded and the modified UCB1 value of the extended node is updated.

Backpropagation After determining the evaluation value of the newly added node, the evaluation value of the remaining tree nodes will be updated. This results in the evaluation value backpropagating from the new node to the root node: If the simulation results of a new node are greater than its node value, the node value of each ancestor node should be updated.

Algorithm 4 P2 Monte Carlo Tree Search algorithm.

Input: $Q_u, \mathcal{P}, G$
Output: A resource allocation sequence L_p
/* Initialize a resource allocation sequence */
Seq.Initialize($Q_u, \mathcal{P}$)
while $Seq.status \neq 0$ **do**
  $\langle src, dst \rangle \leftarrow Q_u$.dequeue()
  T.Initialize()
  $Seq \leftarrow$ AddNextPath($Seq, \langle src, dst \rangle$)
  /* Initialize a Monte Carlo search tree and assign the current state to it */
  $rootNode \leftarrow T.root$
  $rootNode$.Initialize($Seq, \langle src, dst \rangle$)
  for $i \leftarrow 0$ **to** Max_I **do**
    $Node_p =$ SelectingPromisingNode()
    if $Node_p \neq \{\}$ **then**
      if ($Node_p.Seq.status \neq 0$ & $Node_p$.UnexploredState() $\neq \varnothing$) **then**
        if !$Node_p.Seq.(src, dst)$.IsSkipped() **then**
          $Node_e \leftarrow$ ExpandNode($Node_p$)
        end
        else
          $Node_e \leftarrow$ SkipNode($Node_p$)
        end
      end
    end
    $playoutResult \leftarrow$ RandomSimulate($Node_e$)
    backPropogation($Node_e, playoutResult$)
    $resources$.Initialize()
  end
  $Node_w \leftarrow rootNode$.ChildWithMaxScore()
  $Seq \leftarrow Node_w.Seq$
  Q_u.enqueue($\langle src, dst \rangle$)
end

Acknowledgements This paper has been supported by the National Natural Science Foundation of China (Grant Nos. 61702427, 62301454), the Natural Science Foundation of Chongqing, China (Grant No. CSTB-2022NSCQ-MSX0749, CSTB2023NSCQ-MSX0739), China Scholarship Council (Grant No. 202306990061), and Southwest University's 2022 school-level teaching reform project (Grant No. 2022JY086).

Author Contributions Zhongrui Huang designed the algorithm, wrote the main manuscript text and prepared figures. Hong Lai and Linchun Wan assisted in revising the manuscript.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Fang, K., Zhao, J., Li, X., Li, Y., Duan, R.: Quantum network: from theory to practice. Sci. China Inf. Sci. **66**(8) (2023). https://doi.org/10.1007/s11432-023-3773-4
2. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. Theoret. Comput. Sci. **560**, 7–11 (2014). https://doi.org/10.1016/j.tcs.2014.05.025. Theoretical Aspects of Quantum Cryptography - celebrating 30 years of BB84
3. Tomamichel, M., Leverrier, A.: A largely self-contained and complete security proof for quantum key distribution. Quantum. **1**, 14 (2017). https://doi.org/10.22331/q-2017-07-14-14
4. Yin, J., Cao, Y., Li, Y., Liao, S., Zhang, L., Ren, J., Cai, W., Liu, W., Li, B., Dai, H., Wang, J., Pan, J.: Satellite-based entanglement distribution over 1200 kilometers. Science **356**(6343), 1140–1144 (2017). https://doi.org/10.1126/science.aan3211
5. Tessinari, R.S., Bravalheri, A., Hugues-Salas, E., Collins, R., Aktas, D., Guimaraes, R.S., Alia, O., Rarity, J., Kanellos, G.T., Nejabati, R., Simeonidou, D.: Field trial of dynamic DV-QKD networking in the SDN-controlled fully-meshed optical metro network of the Bristol city 5GUK test network. In: 45th European Conference on Optical Communication (ECOC 2019), pp. 1–4 (2019). https://doi.org/10.1049/cp.2019.1033
6. Denchev, V.S., Pandurangan, G.: Distributed quantum computing: A new frontier in distributed systems or science fiction? SIGACT News. **39**(3), 77–95 (2008). https://doi.org/10.1145/1412700.1412718
7. Fitzsimons, J.F.: Private quantum computation: an introduction to blind quantum computing and related protocols. npj Quantum Inform. **3**(1), 23 (2017). https://doi.org/10.1038/s41534-017-0025-3
8. Luo, M.-X.: A nonlocal game for witnessing quantum networks. npj Quantum Inf. **5**(1), 91 (2019). https://doi.org/10.1038/s41534-019-0203-6
9. Gyongyosi, L., Imre, S.: Advances in the quantum internet. Commun. ACM **65**(8), 52–63 (2022). https://doi.org/10.1145/3524455
10. Acín, A., Brunner, N., Gisin, N., Massar, S., Pironio, S., Scarani, V.: Device-independent security of quantum cryptography against collective attacks. Phys. Rev. Lett. **98**, 230501 (2007). https://doi.org/10.1103/PhysRevLett.98.230501
11. Ekert, A.K.: Quantum cryptography based on Bell's theorem. Phys. Rev. Lett. **67**, 661–663 (1991). https://doi.org/10.1103/PhysRevLett.67.661
12. Park, J.L.: The concept of transition in quantum mechanics. Found. Phys. **1**(1), 23–33 (1970). https://doi.org/10.1007/BF00708652
13. Horodecki, R., Horodecki, P., Horodecki, M.: Violating Bell inequality by mixed spin-12 states: necessary and sufficient condition. Phys. Lett. A **200**(5), 340–344 (1995). https://doi.org/10.1016/0375-9601(95)00214-N
14. Riebe, M., Häffner, H., Roos, C.F., Hänsel, W., Benhelm, J., Lancaster, G.P.T., Körber, T.W., Becher, C., Schmidt-Kaler, F., James, D.F.V., Blatt, R.: Deterministic quantum teleportation with atoms. Nature **429**(6993), 734–737 (2004). https://doi.org/10.1038/nature02570
15. Dynes, J.F., Takesue, H., Yuan, Z.L., Sharpe, A.W., Harada, K., Honjo, T., Kamada, H., Tadanaga, O., Nishida, Y., Asobe, M., Shields, A.J.: Efficient entanglement distribution over 200 kilometers. Opt. Express **17**(14), 11440–11449 (2009). https://doi.org/10.1364/OE.17.011440
16. Kómár, P., Kessler, E.M., Bishof, M., Jiang, L., Sørensen, A.S., Ye, J., Lukin, M.D.: A quantum network of clocks. Nat. Phys. **10**(8), 582–587 (2014). https://doi.org/10.1038/nphys3000
17. Gyongyosi, L.: Dynamics of entangled networks of the quantum internet. Sci. Rep. **10**(1), 12909 (2020). https://doi.org/10.1038/s41598-020-68498-x
18. Gyongyosi, L., Imre, S., Nguyen, H.V.: A survey on quantum channel capacities. IEEE Communications Surveys & Tutorials. **20**(2), 1149–1205 (2018). https://doi.org/10.1109/COMST.2017.2786748
19. Meter, R.V., Touch, J.: Designing quantum repeater networks. IEEE Commun. Mag. **51**(8), 64–71 (2013). https://doi.org/10.1109/MCOM.2013.6576340
20. Pant, M., Krovi, H., Towsley, D., Tassiulas, L., Jiang, L., Basu, P., Englund, D., Guha, S.: Routing entanglement in the quantum internet. npj Quantum Inf. **5**(1), 25 (2019). https://doi.org/10.1038/s41534-019-0139-x
21. Ferguson, D., Lindem, A., Moy, J.: OSPF for IPv6. RFC Editor (2008). https://doi.org/10.17487/RFC5340 . https://www.rfc-editor.org/info/rfc5340
22. Routing Information Protocol. RFC Editor (1988). https://doi.org/10.17487/RFC1058 . https://www.rfc-editor.org/info/rfc1058
23. Leone, H., Miller, N.R., Singh, D., Langford, N.K., Rohde, P.P.: QuNet: Cost vector analysis & multi-path entanglement routing in quantum networks (2021)
24. Gyongyosi, L., Imre, S.: Resource prioritization and balancing for the quantum internet. Sci. Rep. **10**(1), 22390 (2020). https://doi.org/10.1038/s41598-020-78960-5

25. Shi, S., Qian, C.: Concurrent entanglement routing for quantum networks: Model and designs. In: Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication. SIGCOMM '20, pp. 62–75. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3387514.3405853
26. Le, L., Nguyen, T.N., Lee, A., Dumba, B.: Entanglement routing for quantum networks: A deep reinforcement learning approach. In: ICC 2022 - IEEE International Conference on Communications, pp. 395–400 (2022). https://doi.org/10.1109/ICC45855.2022.9839240
27. Gyongyosi, L., Imre, S.: Routing space exploration for scalable routing in the quantum internet. Sci. Rep. **10**(1), 11874 (2020). https://doi.org/10.1038/s41598-020-68354-y
28. Zeng, Y., Zhang, J., Liu, J., Liu, Z., Yang, Y.: Multi-entanglement routing design over quantum networks. In: IEEE INFOCOM 2022 - IEEE Conference on Computer Communications, pp. 510–519 (2022). https://doi.org/10.1109/INFOCOM48880.2022.9796810
29. Chang, A., Xue, G.: Order matters: On the impact of swapping order on an entanglement path in a quantum network. In: IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1–6 (2022). https://doi.org/10.1109/INFOCOMWKSHPS54753.2022.9798254
30. Pouryousef, S., Panigrahy, N.K., Towsley, D.: A quantum overlay network for efficient entanglement distribution. In: IEEE INFOCOM 2023 - IEEE Conference on Computer Communications, pp. 1–10 (2023). https://doi.org/10.1109/INFOCOM53939.2023.10228944
31. Jiang, J.-L., Luo, M.-X., Ma, S.-Y.: Quantum network capacity of entangled quantum internet. IEEE J. Sel. Areas Commun. **42**(7), 1900–1918 (2024). https://doi.org/10.1109/JSAC.2024.3380091
32. Li, Z., Li, J., Xue, K., Wei, D.S.L., Li, R., Yu, N., Sun, Q., Lu, J.: Swapping-based entanglement routing design for congestion mitigation in quantum networks. IEEE Trans. Netw. Serv. Manage. **20**(4), 3999–4012 (2023). https://doi.org/10.1109/TNSM.2023.3275815
33. He, B., Zhang, D., Loke, S.W., Lin, S., Lu, L.: Building a hierarchical architecture and communication model for the quantum internet. IEEE J. Sel. Areas Commun. pp. 1–1 (2024). https://doi.org/10.1109/JSAC.2024.3380103
34. Lai, H., Pieprzyk, J., Pan, L.: Novel entanglement compression for QKD protocols using isometric tensors. SCIENCE CHINA Inf. Sci. **66**(8), 180510 (2023). https://doi.org/10.1007/s11432-022-3680-9
35. Day, J.D., Zimmermann, H.: The OSI reference model. Proc. IEEE **71**(12), 1334–1340 (1983). https://doi.org/10.1109/PROC.1983.12775
36. Tessinari, R.S., Woodward, R.I., Shields, A.J.: Software-defined quantum network using a QKD-secured SDN controller and encrypted messages. In: 2023 Optical Fiber Communications Conference and Exhibition (OFC), pp. 1–3 (2023). https://doi.org/10.1364/OFC.2023.W2A.38
37. Dahlberg, A., Skrzypczyk, M., Coopmans, T., Wubben, L., Rozpundefineddek, F., Pompili, M., Stolk, A., Pawełczak, P., Knegjens, R., Oliveira Filho, J., Hanson, R., Wehner, S.: A link layer protocol for quantum networks. In: Proceedings of the ACM Special Interest Group on Data Communication. SIGCOMM '19, pp. 159–173. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3341302.3342070
38. Briegel, H.J., Dür, W., Cirac, J.I., Zoller, P.: Quantum repeaters: The role of imperfect local operations in quantum communication. Phys. Rev. Lett. **81**, 5932–5935 (1998). https://doi.org/10.1103/PhysRevLett.81.5932
39. Yen, J.Y.: An algorithm for finding shortest routes from all source nodes to a given destination in general networks. Q. Appl. Math. **27**, 526–530 (1970)
40. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: Cohen, W.W., Hirsh, H. (eds.) Machine Learning Proceedings 1994, pp. 157–163. Morgan Kaufmann, San Francisco (CA) (1994). https://doi.org/10.1016/B978-1-55860-335-6.50027-1 . https://www.sciencedirect.com/science/article/pii/B9781558603356500271
41. Chakraborty, K., Rozpedek, F., Dahlberg, A., Wehner, S.: Distributed Routing in a Quantum Internet (2019)
42. Yoshizoe, K., Kishimoto, A., Kaneko, T., Yoshimoto, H., Ishikawa, Y.: Scalable distributed Monte-Carlo tree search. In: Symposium on Combinatorial Search (2011). https://api.semanticscholar.org/CorpusID:7526115
43. Schaefers, L., Platzner, M.: Distributed Monte Carlo tree search: A novel technique and its application to computer Go. IEEE Transactions on Computational Intelligence and AI in Games. **7**(4), 361–374 (2015). https://doi.org/10.1109/TCIAIG.2014.2346997
44. Coulom, R.: Efficient selectivity and backup operators in Monte-Carlo tree search. CG'06, pp. 72–83. Springer, Berlin, Heidelberg (2006)
45. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Mach. Learn. **47**(2), 235–256 (2002). https://doi.org/10.1023/A:1013689704352