Computer Networks 272 (2025) 111728

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Entangled qubit pricing for quantum networks

Xiaoyu Wang [a,b,c], Yu Jia [d], Yangming Zhao [a,b,e,*], Shouxi Luo [f], Haoze Chen [g], Chen Tian [a], Dong Zhang [h], Bingheng Yan [h]

[a] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210008, China
[b] School of Intelligent Software and Engineering, Nanjing University Suzhou Campus, Suzhou, 215163, China
[c] School of Computer Science and Technology, University of Science and Technology of China, Hefei, 230026, China
[d] China Mobile (Suzhou) Software Technology Company Limited, Suzhou, 215163, China
[e] Hefei National Laboratory, Hefei, 230088, China
[f] School of Computing and Artificial Intelligence, Southwest Jiaotong University, China
[g] CAS Quantum Network Co., Ltd., Shanghai, China
[h] Jinan Inspur Data Co., Ltd., China

ARTICLE INFO

Keywords:
Quantum network
Network pricing
Entangled qubits
Resource allocation

ABSTRACT

In this paper, we propose an approach named Entangled quBit Pricing (EBP) to price the entangled quantum bits (called ebits) in Resources as a Service (RaaS) quantum networks. In such a quantum network, the service provider pursues the maximum payoff by setting the price of ebits over each link, based on which users decide the number of ebits to purchase such that they can obtain the maximum payoff from establishing entanglement connections (ECs). The unique feature of the routing related pricing issue in quantum networks is that a user may purchase a different number of ebits over each link along his entanglement path in order to establish one EC. By taking into consideration the interaction between the service provider and users, EBP can derive a pricing scheme such that each user will automatically choose the entanglement path and purchase ebits following the way to maximize the service provider's payoff. Extensive simulations show that the service provider can obtain 97 % more payoff with EBP compared with simply setting higher prices to ebits that can successfully create entanglements with higher probability.

1. Introduction

Quantum technique is considered as the most promising information solution that can achieve a performance we have never approached with classic technologies. With quantum computing, we can perform high-performance computing more efficiently than with classical computing. For example, we can solve the integer factorization problem, which we still do not have polynomial-time solutions with classic computers, in polynomial time with quantum techniques [1]. In addition, thanks to the superposition feature of quantum bits (called qubits), we need fewer qubits to encode the same information compared with encoding it with classical bits. For example, we need 2^{2n+3} classic bits to encode a $2^n \times 2^n$ grayscale image with an 8-bit depth of its grayscale, while we need only $2n+8$ qubits to encode the same image [2]. Accordingly, quantum communications has a large potential to help build high-throughput communication systems. Furthermore, Quantum Key Distribution (QKD) is able to distribute keys completely secret to communication parties [3,4], and hence it is considered as the ultimate solution to the secure communications.

In a quantum system that connects end nodes (such as quantum computers) via a quantum network, establishing Entanglement Connections (ECs) is the most critical issue. With an established EC, we can perform a remote binary gate operation over two qubits that are placed on two different quantum computers [5,6], deliver one qubit from one quantum computer to another [7,8], or distribute one bit of secret keys between two communication parties [4]. An EC is physically a pair of entangled qubits (called ebits) each of which is hosted by one of the EC ends. However, due to the little energy of a qubit (usually implemented as a photon), it is difficult to directly distribute a pair of ebits to two ends that are far from each other. To address this issue, we have to find out an Entanglement Path (EP) between these two ends and create Entanglement Links (ELs), which can be considered as one-hop ECs, over every link along this EP. Then, we can leverage a technique called quantum swapping to stitch these ELs and establish the end-to-end EC. This problem is referred to as the *entanglement routing* problem.

The key challenge in entanglement routing is that an EL may fail to be created by distributing an ebit pair over a link. Once an EL fails to be created along an EP, the end-to-end EC will also fail to be established

* Corresponding author.
E-mail address: ymzhao@nju.edu.cn (Y. Zhao).

https://doi.org/10.1016/j.comnet.2025.111728
Received 21 April 2025; Received in revised form 17 September 2025; Accepted 17 September 2025
Available online 22 September 2025
1389-1286/

and the created ELs are wasted. To address this issue, we may concurrently distribute multiple ebit pairs over the same link for creating one EL [9,10]. As long as one ebit pair is distributed to both ends of this link, the corresponding EL is successfully created. Since only limited number of ebit pairs that can be concurrently distributed over each link (referred to as the available ebits hereafter), in order to pursue a certain objective, *e.g.*, the maximum network throughput, we have to carefully assign these ebits to establish required ECs.

Conventionally, we consider the entanglement routing problem by assuming that an operator will determine not only the EPs for establishing each EC [9,11–13] but also the number of ebits that will be provisioned over each link to create the desired EL for each EC [9,10,14]. These works can be applied to a self-operated system. However, with the development of the quantum industry, the quantum resources will be provided by some large operators (or service providers), such as T-mobile and AT&T, and users will purchase quantum resources from (multiple) service providers to establish their required ECs. In this case, the number of ebits used to create ELs and establish ECs will no longer be determined by service providers.

In this paper, we will investigate the problem that in future quantum networks, when a service provider abstracts her Resources as a Service (RaaS), how she prices the quantum resources, *e.g.*, ebits, such that she can maximize her payoff from selling (or renting) ebits to users. In the RaaS scenario, as leasing the Internet Services in the classic network, the service provider only ensures that the required amount of resources, *e.g.*, the bandwidth, will be assigned to each user. She does not care about whether or not a user can obtain the expected revenue from the rented resources. For example, when we rent an Internet service of 100 Mbps from AT&T at the price of 79.99 dollars per month, we may experience network congestion and do not obtain the expected services (or revenue) from the rented bandwidth (just as ELs may fail to be created by the purchased ebits in our work). However, this does not impact the amount of money we need to pay.

The key challenge in this work lies in that with different prices, users may choose different EPs and purchase different number of ebits over each link, which directly impacts the service provider's payoff. To address this issue, we propose an EBit Pricing (called EBP) approach for a service provider to determine the price of ebits over different links, such that a rational user will choose its EP and purchase ebits over each link following the way that the service provider desires. As a result, the service provider can maximize her payoff from selling ebits to users.

In EBP, we first treat the ebit prices as parameters and analyze how each user will determine the EP to establish his required ECs and the number of ebits to purchase over each link. Then, given the EP selection and ebit purchase scheme of each user, we leverage a particle swarm optimization algorithm to search a near-optimal pricing scheme for the service provider. Extensive simulations demonstrate that compared with simply setting higher price to ebits that can successfully create entanglements with higher probability, EBP can help the service provider obtain 97 % more payoff.

The main technique contributions of this work include:

- Deeply analyze the number of ebits that a user will purchase over each link given the ebit price (Section 4.2)
- Propose a particle swarm optimization algorithm to search a near-optimal pricing scheme for the service provider (Section 4.3)
- Conduct extensive simulations to show the superior performance of EBP (Section 5)

The remainder of this paper is organized as follows. In Section 2, we briefly review some existing works that are closely related to ours. In Section 3, we present the model of the system studied in this work followed by the design of EBP in Section 4. After that, we conduct extensive simulations to evaluate the performance of EBP in Section 5. At last, we conclude this paper in Section 6.

2. Related work

EBP is an approach that determines not only the EP to establish each EC but also the ebit provisioning over each quantum link along this EP. This is the problem that has to be solved in the entanglement routing problem. Recently, entanglement routing is widely studied. Q-CAST [7] and Multi-R [11] were proposed to find out EPs for each demand such that we can maximize the number of ECs that can be concurrently established. However, they did not consider the EC demand distribution in the network and hence they may establish many ECs that will never be used by some applications, *e.g.*, to teleport data qubits or perform remote quantum logic gates. In addition, they did not consider that the failure of an EL will lead to the failure of establishing the corresponding end-to-end EC. Then, the created ELs will be wasted and it will result in a low quantum resource utilization. To address this issue, REPS [9] and SEE [10] proposed to provision redundant ebits to mitigate the impact of EL creation failures. Q-PATH [15], EFiRAP [16], and [17] took into consideration the fidelity of the established ECs, [18] integrated topology design into the entanglement distribution problem, [19] extended the entanglement distribution problem into satellite-based quantum networks, and [20] studied the problem of co-designing network topology and qubit allocation for distributed quantum computing. All these works assumed the ebit (*i.e.*, qubit) provisioning is determined by the service provider and the service provider only tries to optimize the network performance. However, in practice, a service provider expects obtaining payoff from providing services to users. None of these works considered the payoff obtained by the service provider which is determined by the interaction between the service provider and users.

In order to obtain payoff from operating quantum networks, the service provider has to price the resources she can provide to users. To the best of our knowledge, there are no existing works on the pricing issue in quantum networks. However, there were many existing works [21–23] focusing on the pricing problem in classical networks, though most of them did not consider the routing issue. The representatives that combined routing and pricing were [24–26]. In classical networks, along a routing path, the required bandwidth over each link equals to end-to-end throughput. In contrast, in quantum networks, the required number of ebits over each link along an EP varies. As a result, the works for classic networks on pricing and routing cannot be directly applied in quantum networks.

3. System model

In this section, we introduce the system model we will study.

Network Model. In this work, we consider that a service provider operates a network consisting of some quantum nodes and links. Similar to most previous works [7,9–11,14], we assume quantum networks are operated in a time-slotted manner. The duration of a time slot is determined by network features, such as the lifetime of an entanglement, *e.g.*, several seconds [7,27]. Over each quantum link, there are several ebits continuously generating ELs. Once an EL is created, it will be used to establish an EC or destroyed by the end of the corresponding time slot. The service provider will maximize its payoff by pricing the ebits over each link and in turn controlling the number of ebits provisioned to each user. It should be noted that, in our work, an ebit would be a logic one that is generated by multiple physical ones in order to ensure the fidelity of the created EL. In other words, we in fact assign a group of physical ebits to create one EL. Some of them are used as sacrificial qubits for purification to ensure the fidelity of the created EL. Once the purification fails, the corresponding EL is also treated as fail to be created since we cannot ensure its fidelity. When an EL is successfully created, its fidelity meets requirement.

User and Demand Models. In a quantum network we study, there are a set of users having to establish ECs in order to perform distributed quantum computing or quantum communications. In each time slot, a user has to report his demand information to the service provider, including the source and destination of the required EC and his revenue from an established EC. Then, based on the ebit prices, each user will determine the number of ebits he will purchase over each link in order to establish ECs. For brevity, we assume a user will try to establish only one EC in a time slot. If a user needs multiple ECs concurrently, he can report multiple demands to the service provider.

Resource Assignment Model. In every time slot, the service provider will provision ebits over each link to users according to the number of ebits they would like to purchase. In a specific time slot, each ebit is dedicated to a user in order to ensure the performance isolation among demands. In other words, once an ebit is provisioned to a user, the corresponding created EL can be used to establish the EC for only this user, but cannot be used to establish an EC for other users, even if this EL will not be used to create an EC for the user who paid for the corresponding ebit (it is possible since a user may purchase multiple ebits over the same link to enhance the success probability of establishing an EC. Besides, some ELs along an EP may not be successfully created and consequently, the ELs created over the remaining links along this EP will not be used to establish an EC).

EC Establishing Model. In EBP, the service provider will assign ebit pairs over each link to user demands in order to establish ECs. Without loss of generality, we assume that within a time slot, a user's demand is only for one EC. Even if created ELs can be used to establish multiple ECs, the users can get revenue associated with only one EC. If a user needs to transmit multiple data qubits, this user should report multiple demands.

Trading Model. In EBP, the service provider provisions her ebits to demands via a commercial trade procedure. For a user having to establish ECs, he will first report the source and destination of his required EC along with the revenue of successfully establishing an EC to the service provider.

According to the demand information, the service provider will determine the ebit price over each quantum link (all ebits over the same link have the same price). This is a setting considering the commercial issue. Though an ebit may have different values when it is used to create an EL for different EPs, the service provider cannot announce different prices for homogeneous ebits according to how a user will use it. After that, every user will calculate the number of ebits he would like to purchase over different links in order to maximize his payoff (*i.e.*, revenue minus payment for purchasing ebits). EBP has embedded mechanism to ensure that as long as all users are rational, there will be always enough ebits to fulfill the requirement of all users. Otherwise, the service provider will increase ebit price to reduce ebit requirements.

It should be noted that once the service provider assigns an ebit to a user, she does not care about whether the corresponding EL will be successfully created nor whether the corresponding end-to-end EC will be successfully established. This is a common setting in conventional networks. For example, when we construct an experiment Quantum Key Distribution (QKD) network in China, we rent a dedicate fiber between Jinan and Wuhan, and another dedicate fiber between Jinan and Hefei. If we want to synchronize a set of secret keys between Wuhan and Hefei, we have to first distributed a set of secret keys between Jinan and Wuhan, and another set of secret keys between Jinan and Hefei. Then, we leverage trusted relay at Jinan to synchronize the set of desired secret keys between Wuhan and Hefei. However, as long as these fibers are assigned to us, the service provider does not concern if the secret keys are successfully distributed on these fibers. We can not obtain a set of secret keys between Wuhan and Hefei as long as either set of secret keys fail to be distributed. In addition, whether or not we obtain the set of desired secret keys does not impact the amount of money we have to pay.

Sometimes, the service provider may price the reliable end-to-end data delivery. More specifically, the service provider will announce a price for end-to-end ECs, instead of the ebits over each link. In this case, the service provider should optimize the EP to establish every sold EC in order to engage more users and make a larger revenue. This pricing model is parallel to our work and we will study it in the future.

4. EBP Design

In this section, we will propose an approach named EBP for the service provider to determine ebit price such that her payoff can be maximized. We first present an overview of EBP in Section 4.1. Then, we analyze through which EP a user will establish the desired EC and how many ebits he will purchase over every link along the selected EP Section 4.2. Based on this analysis, we will propose an algorithm for the service provider to determine her pricing scheme in Section 4.3. At last, we show that a user cannot get more payoff via faking his demand information in Section 4.4. For clear presentation, we summarize all notations used in this work in Table 1.

4.1. EBP in a nutshell

Since a quantum network is operated by a single service provider and a rational user will maximize its own payoff, we design EBP as a hybrid system, *i.e.*, the service provider determines the ebit price in a centralized manner, while users determine how many ebits over each link they would like to purchase in a distributed way. Both the service provider and users know the static network information, such as the network topology and the number of ebits that are available over each link.

At the beginning of a time slot, every user having an EC demand reports the source and destination of his desired EC to the service provider along with the revenue that he can obtain from establishing the corresponding EC. According to the demand information from all users, the service provider will centrally determine the price of an ebit over each link and all ebits over the same link will have the same price. This price will be updated to every user. Based on the ebit price, each user himself will determine not only the EP along which he wants his required EC to be established, but also the number of ebits over every link along the selected EP he would like to purchase for creating the required ELs. At last, every user will report his purchase decision to the service provider and the service provider will fulfill every user's ebit requirement. It is worth noting that EBP will ensure that given the price of ebits over each link, there will be always enough ebits to be sold. Otherwise, EBP will increase the price of ebits over some link in order to reduce the ebit demand. This will also help the service provider obtain more revenue from selling ebits.

The most critical issues in EBP are (i) how each user will determine its EP and the number of ebits he will purchase over each link along

Table 1
Notation list.

Parameters	Description
$\boldsymbol{G}-(\boldsymbol{N},\boldsymbol{L})$	The network graph topology. $\boldsymbol{N}$ is the set of nodes and $\boldsymbol{L}$ is the set of links.
$\boldsymbol{A}-(\boldsymbol{V},\boldsymbol{E})$	The network auxiliary graph topology. $\boldsymbol{V}$ is the set of vertices and $\boldsymbol{E}$ is the set of edges.
R_i	The revenue of establishing an EC for user i.
q_{uv}	The success probability to create an EL between nodes u and v with one pair of ebits.
c_{uv}	The number of ebit pairs available over link (u,v).
w^k_{uv}	The weight of the kth edge over link (u,v) in $\boldsymbol{A}$.
Variables	**Description**
S_i	The success probability to establish an EC for user i.
p_{uv}	The price of ebit pairs over link (u,v).
f^i_{uv}	A binary variable to denote if the user i selects link (u,v).
x^i_{uv}	A integer variable to denote the number of ebit pairs over link (u,v) purchased by the user i.
f^i_{uvk}	A binary variable to denote if the user i selects the kth edge over link (u,v) in $\boldsymbol{A}$.

the selected EP given the ebit price over each link; and (ii) how the service provider will determine the ebit price over different links given the demand information. We will address these two issues in Sections 4.2 & 4.3, respectively.

4.2. Ebit purchase decision

Given the ebit price over each link, users will purchase ebits to create ELs and then ECs. In order to maximize his payoff, a user i can solve the following problem:

$$\text{maximize} \quad S_i R_i - \sum_{(u,v)\in \boldsymbol{L}} p_{uv} x^i_{uv} \tag{1}$$

subject to:

$$\sum_{v:(u,v)\in \boldsymbol{L}} f^i_{uv} - \sum_{v:(v,u)\in \boldsymbol{L}} f^i_{vu} = \begin{cases} 1 & \text{if } u = s_i \\ -1 & \text{if } u = d_i \\ 0, & \text{otherwise} \end{cases} \quad \forall u \in \boldsymbol{N} \tag{1a}$$

$$f^i_{uv} \le x^i_{uv} \quad \forall (u,v) \in \boldsymbol{L} \tag{1b}$$

$$\ln S_i = \sum f^i_{uv} \ln[1-(1-q_{uv})^{x^i_{uv}}] \quad \forall (u,v) \in \boldsymbol{L} \tag{1c}$$

$$f^i_{uv} \in \{0,1\}, x^i_{uv} \in \mathbb{Z} \quad \forall (u,v) \in \boldsymbol{L} \tag{1d}$$

The object is to maximize the payoff of user i, which is the difference between the expected revenue from establishing an EC and the payment to purchase ebits, *i.e.*, $S_i R_i - \sum_{(u,v)\in \boldsymbol{L}} p_{uv} x_{uv}$. Constraint (1a) formulates a feasible EP for user i. Constraint (1b) states user i must purchase at least one ebit over every link along its EP. When user i purchases x^i_{uv} ebits over link (u,v), the success probability of creating an EL will be $1-(1-q_{uv})^{x^i_{uv}}$. Accordingly, constraint (1c) calculates the success probability of establishing the corresponding EC. Constraint (1d) limits the variable to indicate if user i's EP uses link (u,v), *i.e.*, f^i_{uv}, to be a binary variable and the number of ebits purchased by user i on the link (u,v) to be an integer. In (1), we do not formulate the constraint enforced by the number of available ebits over each link, since it is not a responsibility for a user to consider if the service provider has enough resources when he decides the number of ebits to purchase over each link.

It is challenging to solve problem (1) since (i) constraint (1c) is a nonlinear formulation; (ii) f^i_{uv} is a binary variable and x^i_{uv} is an integer variable. To tackle the first challenge, we construct an auxiliary graph $\boldsymbol{A}$ by splitting each link into multiple edges, each of which indicates purchasing a specific number of ebits. Specifically, the nodes in $\boldsymbol{G}$ are same as the vertices in $\boldsymbol{A}$. Each link (u,v) in $\boldsymbol{G}$ constructs c_{uv} parallel edges in $\boldsymbol{A}$. (In the example shown Fig. 1, if there are 3 ebit pairs over link (u,v) in $\boldsymbol{G}$, we can construct 3 edges, *i.e.* $(u,v,1),(u,v,2),(u,v,3)$ in $\boldsymbol{A}$.) In addition, we set the weight of the kth edge over link (u,v) is $\ln[1-(1-q_{uv})^k]$. In this case, the exponent part of constraint (1c) becomes constant and the integer variable x^i_{uv} is eliminated:

$$\text{maximize} \quad \exp\left(\sum_{(u,v,k)\in \boldsymbol{E}} -w^k_{uv} f^i_{uvk}\right) R_i - \sum_{(u,v,k)\in \boldsymbol{E}} k p_{uv} f^i_{uvk} \tag{2}$$

subject to:

$$\sum_{v,k:(u,v,k)\in \boldsymbol{E}} f^i_{uvk} - \sum_{v,k:(v,u,k)\in \boldsymbol{E}} f^i_{vuk} = \begin{cases} 1, & \text{if } u = s_i \\ -1, & \text{if } u = d_i \\ 0, & \text{otherwise} \end{cases} \quad \forall u \in \boldsymbol{V} \tag{2a}$$

$$f^i_{uvk} \in \{0,1\} \quad \forall (u,v,k) \in \boldsymbol{E} \tag{2b}$$

where $w^k_{uv} = \ln[1-(1-q_{uv})^k]$ for all $(u,v,k) \in \boldsymbol{E}$ and f^i_{uvk} denotes if the user i uses edge (u,v,k). When w^k_{uv} approaches 0, the success probability of creating an EL over $(u,v,k) \in \boldsymbol{E}$ approaches 1.

In problem (2), there is an exponential objective function, which makes the problem be a non-convex optimization and difficult to solve. To obtain a large payoff, a user should use links that have a large success probability of creating an EL, *i.e.*, the links (u,v,k) with w^k_{uv} approaches 0. Accordingly, we can infer that in an optimal solution to the problem (2), there would be $-\sum_{(u,v,k)\in \boldsymbol{E}} w^k_{uv} f^i_{uvk} \approx 0$. Then, based on the first order taylor expansion of e^x, *i.e.*, $e^x \approx 1 + x$ when $x \to 0$, we know $\exp(-\sum_{(u,v,k)\in \boldsymbol{E}} w^k_{uv} f^i_{uvk}) \approx 1 - w^k_{uv} f^i_{uvk}$. Consequently, we can reformulate (2) as

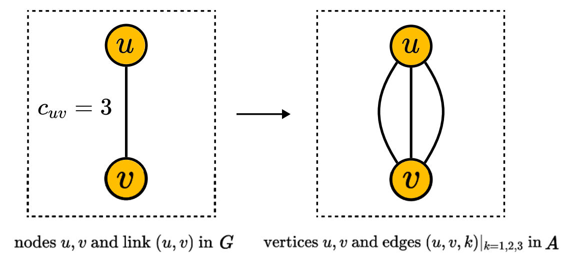


Fig. 1. How to construct an auxiliary graph.

$$\text{maximize} \quad R_i - \sum_{(u,v,k)\in \boldsymbol{E}} (w^k_{uv} R_i + k p_{uv}) f^i_{uvk} \tag{3}$$

subject to: (2a),(2b)

In problem (3), there is still a binary variable f^i_{uvk}, which leads the problem intractable in large scale networks. However, we can show that the solution to problem (3) will not be changed even if we relax the binary constraint, *i.e.*, the solution to problem (3) is the same as that to the following problem:

$$\text{maximize} \quad \left(1 - \sum_{(u,v,k)\in \boldsymbol{E}} w^k_{uv} f^i_{uvk}\right) R_i - \sum_{(u,v,k)\in \boldsymbol{E}} k p_{uv} f^i_{uvk} \tag{4}$$

subject to: (2a)

$$f^i_{uvk} \ge 0, \quad \forall (u,v,k) \in \boldsymbol{E} \tag{4c}$$

To demonstrate the above clarification, we first show following two lemmas:

Lemma 1. *If $\boldsymbol{M}$ is a totally unimodular matrix and $\boldsymbol{b}$ is an integer vector, then all vertices of $P = \{\boldsymbol{x} : \boldsymbol{M}\boldsymbol{x} \le \boldsymbol{b}\}$ are integral.*

Proof. Let $\boldsymbol{v}$ be a vertex of P. There exists a non-singular square submatrix $\boldsymbol{A}'$ of $\boldsymbol{A}$ such that $\boldsymbol{A}'\boldsymbol{v} = \boldsymbol{b}'$. We have $\det(\boldsymbol{A}') = \pm 1$ since $\boldsymbol{A}'$ is nonsingular. By Cramer's Rule [28], we have $\boldsymbol{v}_i = \frac{\det(\boldsymbol{A}'_i|\boldsymbol{b}')}{\det(\boldsymbol{A}')}$, where $\boldsymbol{A}'_i|\boldsymbol{b}'$ is derived by replacing the ith column of $\boldsymbol{A}'$ using $\boldsymbol{b}'$. Therefore, $\boldsymbol{v}_i$ is integral. □

Lemma 2. *The coefficient matrix $\boldsymbol{F} = \{f^i_{uvk}\}$ in constraint (2a) is a totally unimodular matrix.*

Proof. To prove this lemma, we have to show that the determinant of any square submatrix of $\boldsymbol{F}$ is 0, 1, or −1. This can be done via mathematical induction.

Apparently, the determinant of any 1×1 submatrix of $\boldsymbol{F}$ is 0, 1, or −1.

In addition, every column of $\boldsymbol{F}$ has exactly 2 non-zero elements, one is 1 and the other is -1.

Now, suppose the determinant of every $(k-1)\times(k-1)$ submatrix is 0, 1 or −1 and $\boldsymbol{F}'$ is a $k \times k$ submatrix of $\boldsymbol{F}$, we consider the following three cases:

1. If all columns in $\boldsymbol{F}'$ have exactly two non-zero elements, which are 1 and -1, we have $\det(\boldsymbol{F}') = 0$.
2. If there is an all-zero column in $\boldsymbol{F}'$, we have $\det(\boldsymbol{F}') = 0$.
3. If there is a column in $\boldsymbol{F}'$ that has one non-zero element, 1 or -1, suppose this element is at the mth row and nth column, then $|\det(\boldsymbol{F}')| = |det(\boldsymbol{F}'')|$, where $\boldsymbol{F}''$ is the $(k-1)\times(k-1)$ matrix obtained by deleting the mth row and nth column of $\boldsymbol{F}'$. Based on the induction hypothesis, we have $\det(\boldsymbol{F}') \in \{0, \pm 1\}$.

In summary, the determinant of any order square submatrix of $\boldsymbol{F}$ is 0, 1 or −1. In other words, $\boldsymbol{F}$ is a totally unimodular matrix.

□

Based on Lemmas 1 and 2, we derive:

Theorem 1. *The optimal solution to problem (4) is integral.*

Proof. Problem (4) is a linear programming and the optimal solution should be a vertex of its feasible region. Based on Lemmas 1 and 2, we know that all vertices of Problem (4)'s feasible region is integral. Accordingly, the optimal solution to problem (4) is also integral. □

Then, we can further derive the following corollary

Corollary 1. *Problems (3) and (4) have the same solution.*

Proof. Suppose $T^{(3)}$ and $T^{(4)}$ are the objective values of Problems (3) and (4), respectively. At first, (4) is derived by relaxing the integer constraint in (3), we have $T^{(4)} \geq T^{(3)}$. Then, the solution to Problems (4) is integral, and hence it is also a feasible solution to Problems (3). As a result, $T^{(3)} \geq T^{(4)}$. Accordingly, we can conclude that $T^{(3)} = T^{(4)}$. Since the optimal solution to (4) is a feasible solution to (3), the optimal solution to Problem (4) is also the optimal solution to Problem (3) . □

Based on above discussions, we know that given the price of ebits over each link, every user can decide his EP to establish the EC and the number of ebits to purchase over each link by solving the Linear Programming (LP) problem (4). In a network with hundreds of links and every link has tens of available ebits, there are only tens of thousands of variables in this LP problem. With a commodity LP solver, such as CPLEX [29], we can solve it in several milliseconds. For example, in a network with 200 links and each link has 50 available ebits, we can solve (4) in 4 ms on a desktop carrying an Intel Xeon Platinum 8375C CPU and 64 GB memory.

4.3. Ebit pricing approach

Based on the analysis in last subsection, the service provider can maximize its payoff by pricing ebits. The pricing scheme can be derived by solving the following problem:

$$\text{maximize} \quad \sum_{i} \sum_{(u,v)\in \boldsymbol{L}} p_{uv} x^{i}_{uv} \tag{5}$$

subject to:

$$\sum_{i} x^{i}_{uv} \leq c_{uv} \quad \forall (u,v) \in \boldsymbol{L} \tag{5a}$$

$$\boldsymbol{x}^{i} \in \phi^{i}(\boldsymbol{p}) \quad \forall i \tag{5b}$$

where $\boldsymbol{x}^i$ and $\boldsymbol{p}$ are vectors consisting of $\{x^i_{uv}\}$ and $\{p_{uv}\}$, respectively. The objective of (5) is to maximize the payoff obtained by the service provider from selling ebits. Constraint (5a) means the number of ebits purchased by all users cannot exceed the link capacity. By denoting the users' optimal strategies (derived by solving (4)) as $\phi^i(\boldsymbol{p})$, constraint (5b) states that the number of ebits purchased by every user given the ebit price $\boldsymbol{p}$ is determined by solving the problem (4). From the above formulation, especially the constraint (5a), we can observe that in EBP, the service provider ensures that she has enough resources over every link to satisfy the demands from all users. More specifically, if there are not enough ebits over a link to satisfy users' demands, the service provider has to set a higher price to ebits over this link. This is rational since by doing so, the service provider will get more payoff from selling every ebit over this link without selling fewer ebits. Since we cannot get an explicit expression of $\boldsymbol{x}^i$ in constraint (5b), problem (5) is difficult to solve using a classical convex optimization algorithm. Therefore, we resort to efficient heuristics.

Problem (5) is in fact a bi-level optimization problem where the service provider leads the upper problem and each user leads one lower problem. An intuitive method to solve such a bi-level optimization problem is alternatively solving upper and lower problems based on the solution of the other one. However, even a small-scale linear bi-level optimization problem cannot be solved by simple iteration. To tackle this challenge, in EBP, we leverage Particle Swarm Optimization (PSO) [30] to handle this bi-level optimization problem.

More specifically, in EBP, we use $\boldsymbol{pbest}_n$ to record the best position encountered by the particle n in the history and $\boldsymbol{gbest}$ to record the best position encountered by the entire swarm. Every particle adjusts its position and velocity based on $\boldsymbol{pbest}$ and $\boldsymbol{gbest}$. Through sharing $\boldsymbol{gbest}$ among all particles, they will be able to collaborate and find a near-optimal solution quickly.

Algorithm 1: Ebit pricing based on PSO.

Input: Network topology; the revenue of each user establishing an EC R_i; the success probability to create an EL overe each link q_{uv}; the available number of ebit pairs over each link c_{uv}

1 Initialize the swarm with N particles, each with a random position vector $\boldsymbol{p}_n$ and velocity vector $\boldsymbol{v}_n$;
2 Initialize $\boldsymbol{pbest}_n$ and $fitness^*_n$ for each particle;
3 Initialize $\boldsymbol{gbest}$ and $fitness^*$ for entire swarm;
4 **while** *Stop condition has not been achieved* **do**
5   **for** *each particle n* **do**
6     $\boldsymbol{v}_n = \boldsymbol{v}_n + c_1 r_1 \cdot (\boldsymbol{pbest}_n - \boldsymbol{p}_n) + c_2 r_2 \cdot (\boldsymbol{gbest} - \boldsymbol{p}_n)$;
7     $\boldsymbol{p}_n = \boldsymbol{p}_n + t\boldsymbol{v}_n$;
8   **for** *each user i* **do**
9   Formulate and solve problem(4) and get the vector $\boldsymbol{x}^i$;
10   **if** $\exists (u,v) \in \boldsymbol{L}, \sum_i x^i_{uv} - c_{uv} > 0$ **then**
11       $fitness_n = -\infty$;
12   **else**
13       $fitness_n = \sum_i \boldsymbol{p}_n \cdot \boldsymbol{x}^i$;
14   **if** $fitness_n > fitness^*_n$ **then**
15       $fitness^*_n, \boldsymbol{pbest}_n = fitness_n, \boldsymbol{p}_n$;
16   **if** $fitness_n *> fitness *$ **then**
17       $fitness *, \boldsymbol{gbest} = fitness_n *, \boldsymbol{pbest}_n$;
18 **return** $\boldsymbol{gbest}$

We summarize the algorithm used by the service provider to determine the ebit price in Algorithm 1. The solution to problem (5), *i.e.*, the pricing scheme, is encoded as the position of a particle, *i.e.*, $\boldsymbol{p}_n$, and the fitness of this particle is the payoff of the service provider if she prices ebits following $\boldsymbol{p}_n$. At first, we initialize the swarm with N particles in Line 1 and each particle has two properties: position and velocity, denoted as $\boldsymbol{p}_n$ and $\boldsymbol{v}_n$, respectively. Then, we calculate and initialize $\boldsymbol{pbest}_n$, $fitness^*_n$, $\boldsymbol{gbest}$, and $fitness^*$ accordingly (Lines 2–3). In each iteration, each particle first updates its own velocity and position (Lines 4–7). This update consists of three components: the particle's current velocity ($\boldsymbol{v}_n$), the cognitive component ($r_1 \cdot (\boldsymbol{pbest}_n - \boldsymbol{p}_n)$), and the social component ($r_2 \cdot (\boldsymbol{gbest} - \boldsymbol{p}_n)$). The cognitive component encourages a particle to move towards its own best-known position and the social component pulls the particle towards the global best position found by the entire swarm. There are two knobs, *i.e.*, c_1 and c_2, to control the weights among these three components. Once the velocity is updated, the particle's position is also adjusted accordingly, in order to search a better solution (Line 7). In this step, t is a parameter similar to step size. After that, given the position vector $\boldsymbol{p}_n$, we can get the ebit purchase scheme of each user by solving problem (4) (Lines 8–9). Correspondingly, we update the fitness of each particle. Note that if a particle is associated with an infeasible solution, we will set its fitness as $-\infty$ (Lines 10–13). At last, we will update the best position of each particle and the swarm (Lines 14–17). This iteration process ends when a stopping criterion, such as a maximum number of iterations, meets and $\boldsymbol{gbest}$ will be output as the pricing scheme.

Algorithm running time. The most time consuming part in Algorithm 1 is solving problem (4) in Line 9. As we have discussed at the end of

Section 4.2, even in a network with 200 links and each link carries 50 available ebits, we can solve it in 4 ms on a desktop carrying an Intel Xeon Platinum 8375 CPU and 64 GB memory. On the other hand, the loops for each particle (Lines 5–17) and that for each user (Lines 8–9) can be run in parallel. With enough particles, we only need several iterations (*i.e.*, the while loop in Lines 4–17) before the algorithm converges (see details in Section 5.2). Based on above facts, as long as we have enough (but not too many) computation resources, *e.g.*, hundreds of CPU cores and hundreds of Gigabytes memory, we can complete the computation of Algorithm 1 in tens of milliseconds.

4.4. Discussion on user misreporting behavior

In EBP, we assume implicit that all users will report its true revenue from one established EC, *i.e.*, R_i, to the service provider. Though we cannot prove that EBP is a truth-proof approach, in this section, we will show the following proposition:

Proposition 1. *Without knowing the exact information of all other users, e.g., the number of users, the source and destination of each required EC, and the revenue that each user can get from an established EC, a user i does not have a certain scheme to improve his payoff via reducing or increasing the revenue that he reports to the service provider.*

Proof. At first, we assume the EP of each user is fixed. Then, Problem (1) will be reduced to

$$\text{maximize} \quad S_i R_i - \sum_{(u,v)\in P_i} p_{uv} x^i_{uv} \tag{6}$$

subject to:

$$\ln S_i = \sum \ln[1-(1-q_{uv})^{x^i_{uv}}] \quad \forall (u,v)\in P_i \tag{6c}$$

$$x^i_{uv} \geq 0 \quad \forall (u,v)\in P_i \tag{6d}$$

where P_i is the EP used by user i. Since S_i and $1-(1-q_{uv})^{x^i_{uv}}$ are close to 1, we can enforce the approximations $\ln S_i \approx 1 - S_i$ and $\ln[1-(1-q_{uv})^{x^i_{uv}}] \approx (1-q_{uv})^{x^i_{uv}}$. Then, we will be able to solve (6) and derive

$$x^i_{uv} = \frac{\ln\left(\frac{-p_{uv}}{\ln(1-q_{uv})R_i}\right)}{\ln(1-q_{uv})} \tag{7}$$

Given x^i_{uv} as an explicit function of p_{uv}, we can solve Problem (5) and derive

$$p_{uv} = \begin{cases} \frac{-\ln(1-q_{uv})}{e}\left(\prod_{i:(u,v)\in P_i} R_i\right)^{\frac{1}{N_{uv}}}, \text{if } c_{uv} - \sum_i x^i_{uv} > 0 \\ -\ln(1-q_{uv})(1-q_{uv})^{-\frac{c_{uv}}{N_{uv}}}\left(\prod_{i:(u,v)\in P_i} R_i\right)^{\frac{1}{N_{uv}}}, \\ \qquad \text{if } c_{uv} - \sum_i x^i_{uv} = 0 \end{cases} \tag{8}$$

Based on (7) and (8) , we can derive the payoff of a user if it reports its revenue to the service provider as kR_i via some induction, say it is $\Delta\Pi^i_{uv}$. We have

$$\left.\frac{d(\Delta\Pi^i_{uv})}{dk}\right|_{k=1} = A\left(\frac{c_{uv}}{N_{uv}} - \ln\left(\prod_{j\neq i:(u,v)\in P_j} R_j\right)\right)$$

if $c_{uv} - \sum_i x^i_{uv} = 0$, and

$$\left.\frac{d(\Delta\Pi^i_{uv})}{dk}\right|_{k=1} = B\left(\ln\left(\prod_{j\neq i:(u,v)\in P_j} R_j\right) - 1\right)$$

if $c_{uv} - \sum_i x^i_{uv} > 0$. In both cases, we cannot determine the sign of $\left.\frac{d(\Delta\Pi^i_{uv})}{dk}\right|_{k=1}$ without knowing N_{uv} and $R_{j:j\neq i}$. This shows that a user cannot ensure if his payoff will increase or decrease without reporting it true revenue to the service provider. □

Proposition 1 shows that a user does not have motivation to hide his true revenue from an established EC.

5. Performance evaluation

In this section, we evaluate the performance of EBP through extensive simulations. Simulations involve randomly generated networks with a certain number of ebits available over each link, and a set of users with their demands and revenues. For a given set of parameters, simulations are run 100 trails and the averaged results are shown.

5.1. Simulation methodology

Test Case Generation. For each simulation trail, we first randomly generate a network topology by connecting N quantum nodes with $2N$ quantum links. To generate each user's demand, we randomly pick out its source and destination nodes. The users' revenues from establishing an EC follow a logarithmic normal distribution Log-$\mathcal{N}(7, 0.5^2)$. By default, we assume there are 100 nodes and 100 users, the success probability to create an EL with one ebit varies from 0.8 to 1, and there are 6 ebits available to create ELs over each link.

Baselines. We compare EBP with the following schemes:

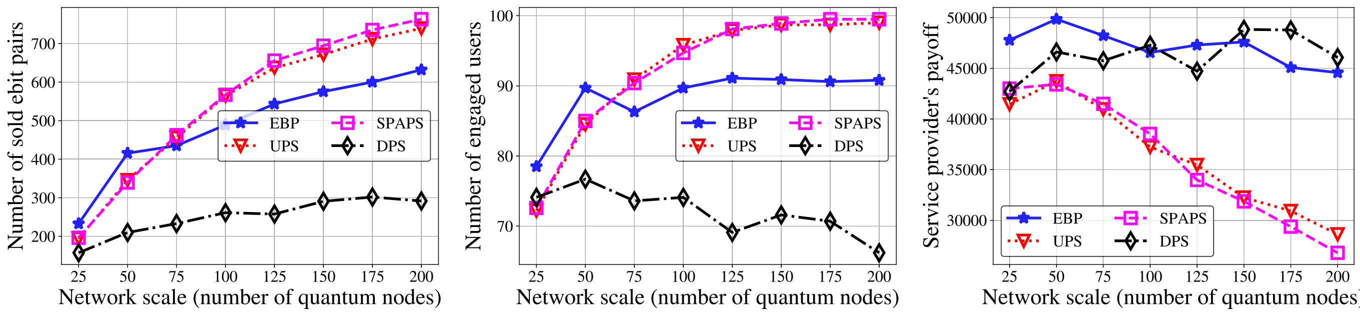
- Universal Pricing Scheme (UPS): In this scheme, we first set the price of all ebits as the minimum revenue of all users over the network diameter. Then, if some links are overloaded, we will increase the price of ebits over it to ensure there are enough ebits to fulfill all demands.
- Success Probability Aware Pricing Scheme (SPAPS): We set the price of an ebit proportional to its success probability to create an EL such that (i) ebits over at least one link are sold out; and (ii) no links are overloaded.
- Differential Pricing Scheme (DPS): In this scheme, the price of an ebit for a user is set to be the user's revenue over the network diameter. If some link is overloaded, we will increase the smallest ebit price over this link to ensure there are enough ebits to fulfill all demands.

Performance Metrics. We use the following three metrics to evaluate the performance of EBP: (i) the number of sold ebits; (ii) the number of engaged users, *i.e.*, the number of users who would like to purchase ebits and establish ECs; and (iii) the payoff of the service provider.

5.2. Evaluation results

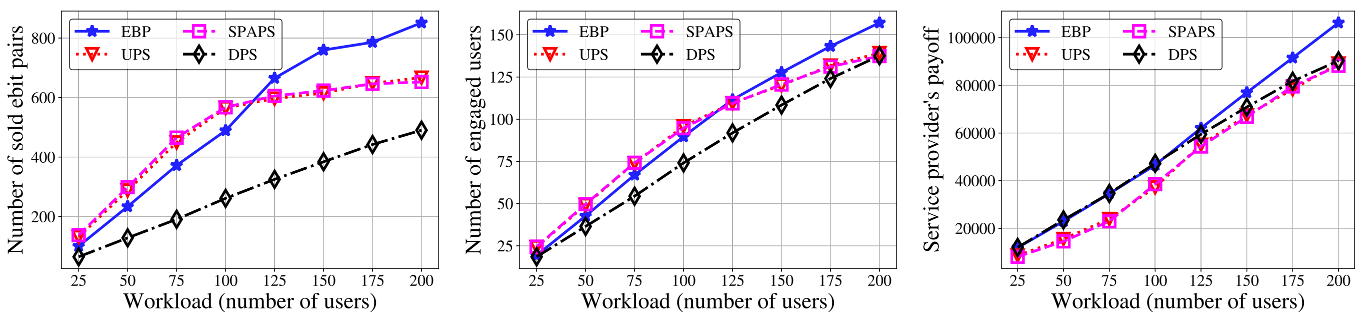
Main observations. Compared with UPS and SPAPS, EBP will sell fewer ebits and engage fewer users. However, with EBP, the service provider can get 113 % and 97 % more payoff compared with that achieved by using UPS and SPAPS, respectively. This is because EBP aims at maximizing the service provider's payoff rather than maximizing network throughput as previous entanglement routing works. Fully utilizing all ebits and maximizing network throughput do not always lead to the maximum payoff to the service provider. In addition, EBP can derive similar or even more payoff to the service provider as DPU, who can ask different prices from different users for ebits over the same link and hence has larger optimization space. This shows the superior performance of EBP.

Impact of Network Scale. We keep the default settings and change the number of quantum nodes in the network from 25 to 200. The simulation results are shown in Fig. 2. From Fig. 2(a) and (b), we can observe that the number of sold ebits and engaged users achieved by EBP, UPS, and SPAPS increase with the network scale. This is because in a larger network, demands will be distributed in a wider area and hence there will be lighter resource contention. As a result, more users can be engaged and correspondingly, more ebits will be sold to users. However, with DPS, the price of ebits over the same link is different for different users, and hence, the average price of ebits also increases. As a result, though more ebit pairs can be sold out in a larger network, fewer users will be engaged.



(a) Effect to the Amount of Sold Resources. (b) Effect to the Number of Engaged Users. (c) Effect to the Service Provider's Payoff.

Fig. 2. Impact of network scale.



(a) Effect to the Amount of Sold Resources. (b) Effect to the Number of Engaged Users. (c) Effect to the Service Provider's Payoff.

Fig. 3. Impact of workload.

In Fig. 2(c), we can see that for both UPS and SPAPS, the payoff of the service provider initially increases and then decreases with the increasing of the network scale. The maximum payoff appears when there are 50 nodes in the network. In a larger network, the price of each ebit will decrease since there is a larger diameter. Then, more users would like to purchase ebits and more ebits will be sold. In a smaller network, this will lead to a payoff increase to the service provider. However, in a large network engaged more users, the price reduction will dominate the benefit brought by selling more ebits, which results in a payoff reduction. When EBP or DPS is adopted, though the price of each ebit also decreases with the network scale increases (since a user needs more ebits to establish its required EC and hence he is not willing to pay that much for each ebit as in a smaller network), setting different prices to ebits over different links or to different users help ask for more payment from users. Accordingly, the price decrease may be offset by selling out more ebits and we cannot observe significant trend from the service provider's payoff when the network scale increases.

Impact of Workload. When the number of users in the network increases from 25 to 200, we derive the simulation results shown in Fig. 3. Regardless of which pricing approach is adopted, the service provider will sell more ebits when there are more users in the network. Correspondingly, more users will be engaged and more payoff that the service provider will obtained from selling ebits. When there are more than 125 users in the network, the number of ebits that can be sold to users will increase at a slower pace. This is due to the limitation of available number of ebits.

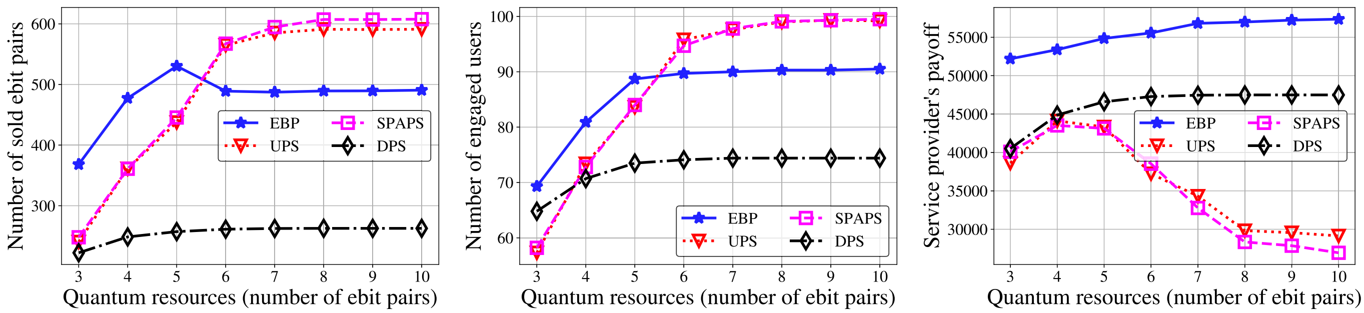
Impact of Available Ebits. To investigate how the number of available ebits over each link affects the performance of different comparison schemes, we vary the available ebits over each link from 3 to 10 and derive the simulation results shown in Fig. 4. First, we observe the performance of EBP. When there are very few ebits available over each link, the more ebits there are available over each link, the more ebits will be sold to users and the more users will be engaged. In this phrase, the average price will decrease in order to serve more users and establish more ECs, which will lead to larger payoff for the service provider. However, when there are more than 5 ebits available over each link, decreasing price to attract more users will not bring more payoff to the service provider. Accordingly, all metrics, *i.e.*, the number of sold ebits, the number of engaged users, and the payoff of the service provider will keep stable when the number of available ebits over each link increases.

When UPS or SPAPS is adopted, the service provider will sell more ebits and engage more users when there are enough ebits in the network. However, the payoff of the service provider decreases with the increase of the network scale. This is because in order to sell all ebits over at least one link, the ebit price is too low. This also demonstrates that EBP can maximize the service provider's payoff at the cost of under-utilizing quantum resources.

Impact of the Success Probability of EL Creation. By increasing the minimum success probability of creating an EL with one ebit from 0.55 to 0.95, we derive the simulation results shown in Fig. 5. When we can create an EL with larger probability, a user does not need to purchase more ebits over each link along his EP for backup purpose. Accordingly, with EBP, UPS, or SPAPS, fewer ebits will be sold with the increases of the success probability of EL creation (as shown in Fig. 5(a)). Since each user needs fewer ebits, as shown in Fig. 5(b), the service provider can engage more users accordingly.

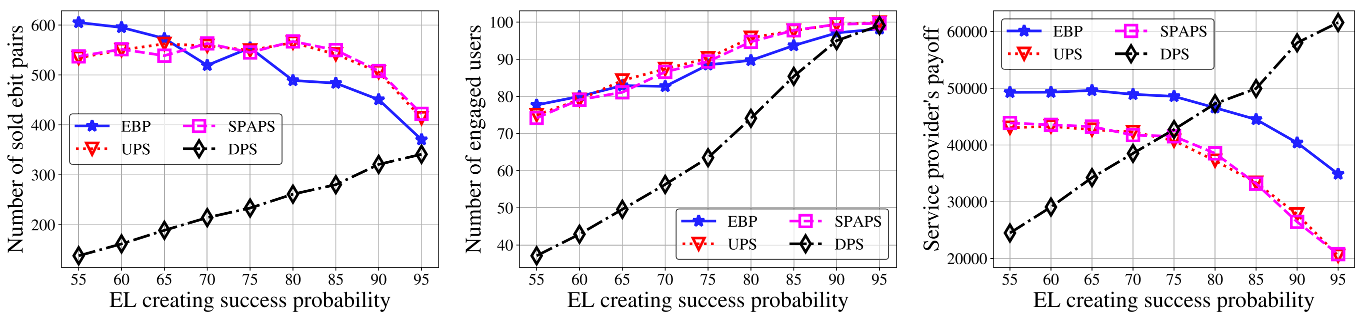
However, with DPS, we can observe an exception. Since the service provider asks for different prices to different users, she can sell more ebits by reducing prices to users having less revenue from an established EC. Due to the same reason, the number of engaged users derived by DPS increases much faster than the other three comparison schemes when an EL can be successfully created with higher probability using one ebit.

In Fig. 5(c), we can observe that the payoff of the service provider will decrease with the increase of success probability to create each EL if EBP, UPS, or SPAPS is adopted. This is mainly because each user will need fewer ebits to establish his EC. However, if DPS is adopted, the service provider will reduce the price of an ebit for users having less revenue from an established EC. As a result, with a larger success



(a) Effect to the Amount of Sold Resources.
(b) Effect to the Number of Engaged Users.
(c) Effect to the Service Provider's Payoff.

Fig. 4. Impact of quantum resources.



(a) Effect to the Amount of Sold Resources.
(b) Effect to the Number of Engaged Users.
(c) Effect to the Service Provider's Payoff.

Fig. 5. Impact of creating success probability.

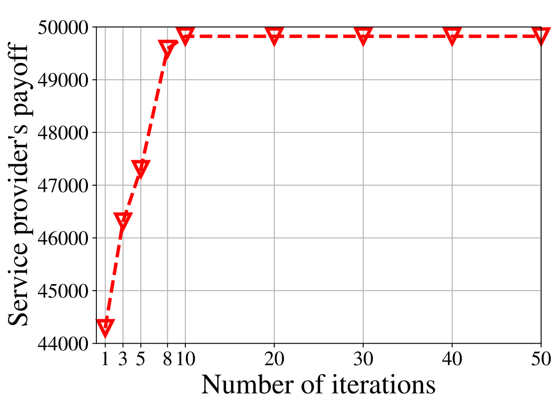


Fig. 6. Convergence rate of the Algorithm 1.

probability to create EL with one ebit, the service provider can ask for larger average price for each ebit which brings her more payoff.

Algorithm Convergence Rate. The number of iterations in the Algorithm 1 (*i.e.*, Lines 4–17) we have to perform determines the time the service provider needs to calculate the price of ebits over each link. Fig. 6 shows how the service provider's revenue changes with the number of iterations that have been performed in Algorithm 1 under the default setting. From this figure, we can observe that the Algorithm 1 converges in 10 iterations. Consider that we need only 4 ms to perform an iteration with enough (but not too many) computation resources, the service provider can complete the price calculation in tens of milliseconds.

6. Conclusions

This paper have proposed an EBit Pricing (called EBP) approach for a service provider to price its quantum resources, *e.g.*, ebits, such that she can maximize her payoff from selling resources to users. The salient feature of EBP is that by setting the ebit price over each link derived by EBP, users will purchase ebits following the way that can maximize the service provider's payoff when all users pursue their maximum payoff. Extensive simulations have been conducted to show the superior performance of EBP.

CRediT authorship contribution statement

Xiaoyu Wang: Writing – review & editing, Writing – original draft, Conceptualization; **Yu Jia:** Resources, Data curation; **Yangming Zhao:** Software, Project administration, Funding acquisition; **Shouxi Luo:** Formal analysis; **Haoze Chen:** Writing – review & editing; **Chen Tian:** Writing – review & editing; **Dong Zhang:** Visualization, Validation; **Bingheng Yan:** Software, Data curation.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants 62572233 and 62272428, and in part by Innovation Program for Quantum Science and Technology under Grant 2021ZD0300705.

References

[1] P.W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, in: Proceedings 35th Annual Symposium on Foundations of Computer Science, 1994, pp. 124–134.
[2] M.G. Amankwah, D. Camps, E.W. Bethel, R. Van Beeumen, T. Perciano, Quantum pixel representations and compression for N-dimensional images, Sci. Rep. 12 (1) (2022) 7712.
[3] C.H. Bennett, G. Brassard, Quantum cryptography: public key distribution and coin tossing, Theor. Comput. Sci. 560 (2014) 7–11. Theoretical aspects of quantum cryptography - celebrating 30 years of BB84.
[4] A.K. Ekert, Quantum cryptography based on Bell's theorem, Phys. Rev. Lett. 67 (6) (1991) 661–663.
[5] J. Eisert, K. Jacobs, P. Papadopoulos, M. Plenio, Optimal local implementation of nonlocal quantum gates, Phys. Rev. A 62 (2000) 052317.
[6] A. Yimsiriwattana, J. Lomonaco, Generalized GHZ states and distributed quantum computing, (2004) arXiv preprint arXiv:quant-ph/0402148.
[7] S. Shi, C. Qian, Concurrent entanglement routing for quantum networks: model and designs, in: Proceedings of the ACM SIGCOMM, 2020.
[8] M. Riebe, H. Häffner, C.F. Roos, W. Hänsel, J. Benhelm, G.P.T. Lancaster, T.W. Körber, C. Becher, F. Schmidt-Kaler, D.F.V. James, R. Blatt, Deterministic quantum teleportation with atoms, Found. Phys. 429 (2004) 734–737.
[9] Y. Zhao, C. Qiao, Redundant entanglement provisioning and selection for throughput maximization in quantum networks, in: Proceedings of the IEEE INFOCOM, 2021.
[10] G. Zhao, J. Wang, Y. Zhao, H. Xu, C. Qiao, Segmented entanglement establishment for throughput maximization in quantum networks, in: IEEE ICDCS, 2022.
[11] Y. Zeng, J. Zhang, J. Liu, Z. Liu, Y. Yang, Multi-entanglement routing design over quantum networks, in: Proceedings IEEE INFOCOM, 2022.
[12] L. Yang, Y. Zhao, L. Huang, C. Qiao, Asynchronous entanglement provisioning and routing for distributed quantum computing, in: Proceedings of the IEEE INFOCOM, 2023.
[13] T.N. Nguyen, K.J. Ambarani, L. Le, I. Djordjevic, Z.-L. Zhang, A multiple-entanglement routing framework for quantum networks, (2022). arXiv preprint arXiv:2207.11817.
[14] L. Yang, Y. Zhao, H. Xu, C. Qiao, Online entanglement routing in quantum networks, in: IEEE/ACM IWQoS, 2022.
[15] J. Li, M. Wang, K. Xue, R. Li, N. Yu, Q. Sun, J. Lu, Fidelity-guaranteed entanglement routing in quantum networks, IEEE Trans. Commun. 70 (10) (2022) 6748–6763.
[16] Y. Zhao, G. Zhao, C. Qiao, E2E fidelity aware routing and purification for throughput maximization in quantum networks, in: IEEE INFOCOM, 2022, pp. 480–489.
[17] J. Liu, X. Zhang, X. Wei, et al., Joint swapping and purification optimization for entanglement distribution in quantum networks, in: IEEE/ACM IWQoS, 2025.
[18] J. Liu, X. Liu, X. Wei, Y. Wang, Topology design with resource allocation and entanglement distribution for quantum networks, in: IEEE International Conference on Sensing, Communication, and Networking (SECON), 2024, pp. 1–9.
[19] X. Wei, J. Liu, L. Fan, Y. Guo, Z. Han, Y. Wang, Optimal entanglement distribution problem in satellite-based quantum networks, IEEE Netw. 39 (1) (2025) 97–103.
[20] J. Liu, L. Fan, Y. Guo, Z. Han, Y. Wang, Co-design of network topology and qubit allocation for distributed quantum computing, in: International Conference on Quantum Communications, Networking, and Computing (QCNC), 2025, pp. 355–362.
[21] Q. Lin, L. Duan, J. Huang, Personalized pricing through strategic user profiling in social networks, IEEE/ACM Trans. Netw. (2024).
[22] N. Ding, L. Gao, J. Huang, Joint participation incentive and network pricing design for federated learning, in: IEEE INFOCOM 2023-IEEE Conference on Computer Communications, IEEE, 2023, pp. 1–10.
[23] C. Chen, R.A. Berry, M.L. Honig, V.G. Subramanian, Pricing, bandwidth allocation, and service competition in heterogeneous wireless networks, IEEE/ACM Trans. Netw. 28 (5) (2020) 2299–2308.
[24] S.H. Low, D.E. Lapsley, Optimization flow control. I. basic algorithm and convergence, IEEE/ACM Trans. Netw. 7 (6) (1999) 861–874.
[25] T. Basar, R. Srikant, Revenue-maximizing pricing and capacity expansion in a many-users regime, in: Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, 1, IEEE, 2002, pp. 294–301.
[26] N. Jin, S. Jordan, On the feasibility of dynamic congestion-based pricing in differentiated services networks, IEEE/ACM Trans. Netw. 16 (5) (2008) 1001–1014.
[27] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpundefineddek, M. Pompili, A. Stolk, P. Pawelczak, R. Knegjens, J. de Oliveira Filho, R. Hanson, S. Wehner, A link layer protocol for quantum networks, in: ACM SIGCOMM, 2019, pp. 159–173.
[28] D. Poole, Linear algebra: a modern introduction, Cengage Learning, 2014.
[29] "IBM", User's manual for CPLEX (2022). https://www.ibm.com/docs/en/icos/22.1.1?topic=optimizers-users-manual-cplex.
[30] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, 4, ieee, 1995, pp. 1942–1948.

Xiaoyu Wang is a visiting student at school of intelligent software and engineering, Nanjing University - Suzhou Campus. He is pursuing his master degree at school of computer science and technology, University of Science and Technology of China. He received the BS degree at school of computer science and technology, University of Electronic Science and Technology of China in 2023. Her research interests include network optimization and quantum networks.

Yu Jia, Expert of China Mobile's "Ten-Hundred-Thousand Talent Program", Senior Technical Researcher and R&D Director at China Mobile Cloud Computing Center. Member of the Mobile Cloud Architecture Expert Group and Network Expert Group, responsible for the evolution of the virtualized network architecture of Mobile Cloud and the research on cutting-edge technologies. He hatched the first major strategic technology-to-product achievement transformation of Mobile Cloud. He has undertaken and participated in 2 national-level projects, publicly published 6 papers in the field, applied for more than 60 domestic patents, and obtained authorization for more than 20 patents.

Yangming Zhao is an associate professor at School of Intelligent Software and Engineering, Nanjing University Suzhou Campus. Before that, he was a research professor at school of computer science and technology, University of Science and Technology of China. Before that, he was a research scientist with University at Buffalo. He received the BS degree in communication engineering and the PhD degree in communication and information system from University of Electronic Science and Technology of China in 2008 and 2015, respectively. His research interests include network optimization, quantum networks, edge computing and machine learning.

Shouxi Luo received the bachelor's degree in communication engineering and the PhD degree in communication and information systems from the University of Electronic Science and Technology of China, China, in 2011 and 2016, respectively. He is currently an Associate Professor with %the School of Computing and Artificial Intelligence, Southwest Jiaotong University. His research interests include data center networks, software-defined networking, and networked systems.

Dong Zhang as the chairman of Jinan Inspur Data Technology Co., Ltd., has led the development of the world's highest computing and storage density rack server, the first China UNIX operating system. He has made creative contributions in areas such as converged architecture and high-end system software, earning one national award, and eleven provincial-level awards.

Chen Tian received the BS, MS, and PhD degrees from the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, China, in 2000, 2003, and 2008, respectively. He is currently a Professor with the State Key Laboratory for Novel Software Technology, Nanjing University, China. His research interests include data center networks, network function virtualization, distributed systems, internet streaming, and urban computing.

Bingheng Yan received his PhD at Xi'an JiaoTong University in 2010. He is broadly interested in the area of operating system, virtualization, and cloud computing. He is the cloud R&D director of Jinan Inspur Data Co., Ltd., where he has led a wide range of virtualization research projects, and the development of Inspur's server virtualization product InCloud Sphere, which break the global world record of SpecVirt.

Haoze Chen, a PhD from the University of Science and Technology of China (USTC). He is currently serving as Deputy Chief Engineer at CAS Quantum Network Co., Ltd., with research interests in quantum communication, quantum communication networks, and cryptographic technologies