All-or-Nothing Concurrent Entanglement Routing

Ching-Ting Wei[§], Kai-Xu Zhan[§], Po-Wei Huang[‖], Wei-Ting Chang[‖], and Jian-Jhih Kuo[*]
Dept. of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan

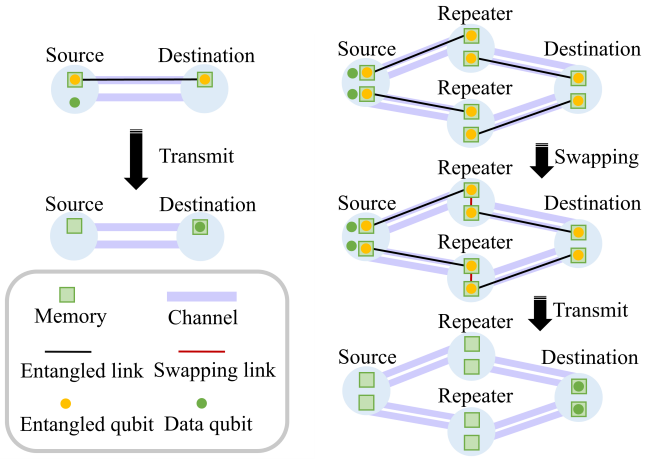
***Abstract*—Establishing entangled links by entangling and swapping can enable end-to-end teleportation, avoiding eavesdropping and facilitating the critical applications, such as quantum key distribution (QKD) service. For these promising applications, a specific number of data qubits should be transmitted (periodically) to meet the quality of service (QoS) requirements for each request. Otherwise, the network provider cannot earn the profit from the request. In this paper, we present a new optimization problem and propose a novel approximation algorithm to maximize the (expected) total profit under the constraint of network resources. Finally, extensive simulation results manifest that our algorithm can outperform the existing approaches by up to 46%.**

I. Introduction

Quantum networks (QNs) are promising to facilitate secure data transmission and foster enormous innovative applications for information and communication technology [1]. In a QN, quantum nodes (e.g., quantum switches and quantum computers) are connected to transmit quantum information in the form of *qubits* [2]. As shown in Fig. 1, each quantum node in the QN has a specific amount of *quantum memory* to store qubits and avoid rapid decoherence [3]. By doing *entangling* to generate an entangled pair through a *quantum channel* (e.g., an optical fiber), any two adjacent nodes can establish an entangled link to transmit a data qubit via quantum teleportation [4], as shown in Fig. 1(a). For any two non-adjacent nodes, multiple consecutive entangled links can be merged into one entangled link between them by conducting entanglement *swapping* at the repeaters, i.e., gluing entangled links with logical links created by swapping [5] (i.e., swapping links), as depicted in Fig. 1(b). This way can prevent a third party from eavesdropping data transmission thanks to secure *end-to-end* teleportation. However, entangling and swapping may fail [6], so an efficient algorithm to find the routing paths for requests is crucial.

Most routing algorithms have been proposed in the literature to maximize network throughput for QNs. For example, in [7], a greedy algorithm is employed to identify the path with the fewest hops for each request. Q-CAST [8] uses the Dijkstra algorithm to find primary and recovery paths, mitigating the impact of entangling failures and enhancing network throughput. REPS [5] partitions the routing problem into two subproblems, which are formulated as two linear programmings (LPs) to arrange entangling and swapping, respectively. However, for many critical applications, such as quantum key distribution (QKD) service [9], a specific number of data qubits should be transmitted successfully to meet the quality of service (QoS) requirements for each request. Otherwise, the network provider cannot earn profit from the request (i.e., *all-or-nothing*). These

§ ‖: equal contributions; *: corresponding author (lajacky@cs.ccu.edu.tw)



(a) Direct transmission for a data qubit (b) Use swapping for two data qubits

Fig. 1. Illustration of end-to-end transmission in quantum networks.

requests are difficult to satisfy by most existing routing algorithms for QNs since those algorithms neglect the all-or-nothing effect. In contrast, the all-or-nothing routing algorithms for traditional networks may not be suitable for QNs because each request's profit does not change over different traditional routing paths. Nevertheless, the (expected) profit usually varies based on the success probability of constructing the entangled link for that request in QNs, complicating the routing problem.

In this paper, therefore, we make the first attempt to explore *all-or-nothing concurrent entanglement routing* in QNs, which introduces the following challenges: 1) *Tradeoff between success probability and load balance*. The successful transmission of a data qubit relies on the successful creation of entangled links. Intuitively, prioritizing the paths with the highest probability can improve throughput. Nonetheless, simply considering probability could cause hot spots and potential congestion. Therefore, it is necessary to strike a balance between success probability and load distribution. 2) *Varying resource consumption*. Allocating resources to more profitable requests can yield significant benefits. However, satisfying such requests may consume plenty of resources if it transmits a lot of data. In addition, it may consume a different amount of resources varying from the selected routing paths. 3) *All-or-nothing effect*. We must ensure allocating a sufficient number of paths to a request to earn profit from it. This effect exacerbates the difficulty since requests may compete for resources more seriously than throughput maximization. Thus, all-or-nothing concurrent entanglement routing is very challenging because it needs to jointly decide whether and how to serve the requests.

To tackle the aforementioned challenges, we first formulate

a new optimization problem, called All-or-nothing Concurrent Entanglement Routing (ACER), to capture the all-or-nothing effect and the QN uniqueness (e.g., the success probabilities of entangling and swapping). Given a set of requests, ACER aims to maximize the expected total profit under the channel and memory limitations. Then, we provide its integer linear programming (ILP) and show its hardness in Section II. Subsequently, Section III introduces a bi-criteria approximation algorithm for the ACER. Particularly, we cleverly devise a separation oracle for the dual LP and employ a primal-dual algorithm (PDA) to obtain a fractional solution with a bounded error to address the possibly exponential number of paths between any source-destination (SD) pair. We also apply the LP rounding technique to achieve the approximation ratio and further refine the solution by a heuristic algorithm. In Section IV, the extensive simulation results show that our algorithm can outperform existing approaches by up to 46%. Finally, Sections V and VI discuss the related works and conclude this paper.

II. System Model and Problem Formulation

A. System Model

A QN comprises multiple quantum nodes. Each node has limited quantum memory and connects to neighboring nodes through a limited number of quantum channels [10]. Quantum nodes can act as sources, repeaters, or destinations for different requests simultaneously. During the entangling process, both two nodes, say, u and v, consume one unit of their quantum memory to generate the entangled pair through the quantum channel (u, v), as shown by the purple bold lines in Fig. 1. The success probability of entangling decreases exponentially with the channel distance due to signal loss. That is, $\Pr(u, v) = e^{-\gamma \cdot l(u,v)}$, where $l(u, v)$ is the length of the quantum channel between u and v, and γ is a constant determined by the optical fiber material [3]. In addition, the swapping process can form a swapping link logically to merge two entangled links (paths) into a single entangled link (path). Swapping also has a different success probability at different node u, denoted as $\Pr(u)$. Thus, the success probability of a path p can be expressed as a formula in Definition 1. For clarity, Fig. 1(b) shows an example where each red swapping link logically connects two entangled links to form an entangled path.

Definition 1. The success probability of constructing an entangled path p, denoted as $\Pr(p)$, is the product of all the entangling success probability of links on path p and all the swapping success probability of the repeaters on path p. That is, $\Pr(p) = \prod_{e \in p} \Pr(e) \prod_{v \in p \setminus \{s,d\}} \Pr(v)$, where $\Pr(e)$ (or $\Pr(v)$) is the success probability of entangling on edge e (or swapping on node v) and s (or d) denotes the source (or the destination) of path p.

Definition 2. Following Definition 1, the success probability of a path set S, denoted as $\Pr(S)$, is the product of all the success probability of the paths in S, i.e., $\Pr(S) = \prod_{p \in S} \Pr(p)$.

Each request needs to send a specific number of data qubits, and its qubits should be transmitted through multiple entangled links (paths) to earn the profit. To coordinate all nodes, we follow [5] to deploy a central controller to periodically collect information about the entangling and swapping outcomes on each link and node. The centralized approach is reasonable since the maximum one-way propagation delay typically ranges in the tens of milliseconds [5], [11]. Overall, how to allocate resources efficiently and maximize the expected total profit is an important optimization problem in QNs.

B. Problem Formulation

We formulate the problem based on the system model. Consider an undirected network $G = (V, E)$ with a channel limit $c(e) \in \mathbb{Z}^+$ and an entangling success probability $\Pr(e) \in (0, 1]$ associated with each edge $e \in E$, as well as a memory limit $m(v) \in \mathbb{Z}^+$ and a swapping success probability $\Pr(v) \in (0, 1]$ associated with each node $v \in V$. Let I be the set of SD pairs for the arrival requests. Each SD pair $i \in I$ requests to send $k(i) \in \mathbb{Z}^+$ qubits periodically.[1] In addition, each pair SD $i \in I$ has its profit $r(i) \in \mathbb{R}^+$.[2] The All-or-nothing Concurrent Entanglement Routing (ACER) aims to allocate the network resources for the SD pairs to maximize the (expected) total profit for all SD pairs, subject to the following constraints:

1) For each quantum node v, the amount of quantum memory used on v does not exceed $m(v)$.
2) For each edge e, the total number of transmitted qubits on e does not exceed $c(e)$.
3) For each SD pair i, a set of $k(i)$ paths can be selected to earn the profit $r(i)$.

Let $P(i)$ be the set of all possible paths for each SD pair $i \in I$, and let $x_S^i \in \{0, 1\}$ denote whether to choose the path set S from $P(i)$, where $|S| = k(i)$, for each SD pair $i \in I$. Let $\binom{P}{k}$ be the set of all k-combinations from the set P with repetition, i.e., each element $p \in P$ can be selected multiple times. Then, the ACER is formulated as an ILP (1a)−(1e).

$$\text{maximize} \quad \sum_{i \in I} \sum_{S \in \binom{P(i)}{k(i)}} \Pr(S) \cdot r(i) \cdot x_S^i \tag{1a}$$

$$\text{subject to} \quad \sum_{i \in I} \sum_{S \in \binom{P(i)}{k(i)}} \sum_{u \in V} \sum_{p \in S : (u,v) \in p} x_S^i \le m(v), \quad \forall v \in V \tag{1b}$$

$$\sum_{i \in I} \sum_{S \in \binom{P(i)}{k(i)}} \sum_{p \in S : e \in p} x_S^i \le c(e), \quad \forall e \in E \tag{1c}$$

$$\sum_{S \in \binom{P(i)}{k(i)}} x_S^i \le 1, \quad \forall i \in I \tag{1d}$$

$$x_S^i \in \{0, 1\} \quad \forall i \in I, \forall S \in \binom{P(i)}{k(i)} \tag{1e}$$

Note that the objective function (1a) aims to maximize the expected total profit for all SD pairs, where $\Pr(S)$ represents

[1] For ease of presentation, we assume that the demand of each SD pair i is much lower than the number of channels of each link e and the memory limit of each node, i.e., $k(i) \ll c(e)$ and $k(i) \ll m(v)$. It is reasonable since a request can typically be served and satisfied by a path in networks.

[2] Each SD pair i may have a different profit that may be set according to the task urgency, SD distance, resource fairness, willingness to pay, etc.

the success probability of path set S, as defined in Definition 2. Constraints (1b) and (1c) guarantee that the total amount of memory used on node v and the total number of channels used on link e do not exceed their respective limits. Constraint (1d) ensures that each SD pair $i \in I$ chooses at most one subset with exact $k(i)$ paths from path set $P(i)$.

The ACER is NP-hard, and the detailed proof is provided in Appendix A of the technical report [12] due to the page limit.

Theorem 1. The ACER is NP-hard.

III. Bi-criteria Approximation Algorithm

In the original formulation of the ILP problem (1a)−(1e), the number of variables may grow exponentially with the size of the input since the number of paths between any two nodes can also be exponential. As a result, solving the relaxed LP (i.e., relaxing the constraints $x_S^i \in \{0,1\}$ to be $x_S^i \in [0,1]$) becomes infeasible for an LP solver (e.g., Gurobi) in polynomial time.

Fortunately, according to the findings in [13], if an LP is in standard form (see Definition 3) with a polynomial number of constraints (excluding non-negativity constraints on variables), the number of variables in its dual LP is also polynomial. In this case, we can solve the dual LP by designing a separation oracle for the dual LP to identify the violated constraint among an exponential number of constraints. Moreover, based on [14], a near-optimal solution for the primal LP can be obtained in polynomial time by a primal-dual algorithm (PDA) if the following conditions are met:

1) In every constraint of the primal LP, except for the non-negativity constraints of the variables, each coefficient on the left-hand side of each inequality is less than or equal to the constant on the right-hand side.
2) In every constraint of the dual LP, except for the non-negativity constraints of the variables, all coefficients on the left-hand side and the constant on the right-hand side of each inequality are positive.
3) A *separation oracle* exists for the dual LP to identify the (most) violated constraint efficiently.

Definition 3. An LP is said to be in standard form if it is written as $\max\{c^T x | Ax \leq b, x \geq 0\}$, where x is an $n \times 1$ variable vector, and c, b, and A denote $n \times 1$, $m \times 1$, $m \times n$ constant vectors, respectively. Then, its dual LP is $\min\{b^T y | A^T y \geq c, y \geq 0\}$, where y is an $m \times 1$ variable vector. Note that $x \geq 0$ and $y \geq 0$ are the non-negativity constraints of x and y in the primal and dual LP, respectively.

In order to achieve the desired relaxed primal LP, we begin by replacing constraint (1e) with constraint (2) as follows:

$$x_S^i \geq 0, \qquad \forall i \in I, \forall S \in \binom{P(i)}{k(i)}, \tag{2}$$

which aligns with the non-negativity constraints for variables. Consequently, the primal LP (1a)−(1d), (2) is in standard form with a polynomial number of constraints (excluding the non-negativity constraints for variables) and also meets the first condition. According to Definition 3, we can derive the corresponding dual LP (3a)−(3c). Note that the dual LP satisfies the second condition.

$$\text{minimize} \quad \sum_{v \in V} m(v) \cdot \alpha_v + \sum_{e \in E} c(e) \cdot \beta_e + \sum_{i \in I} \tau_i \tag{3a}$$

$$\text{subject to} \quad \sum_{p \in S} \Big(\sum_{(u,v) \in p} (\alpha_u + \alpha_v) + \sum_{e \in p} \beta_e \Big) + \tau_i \geq \Pr(S) \cdot r(i), \quad \forall i \in I, \forall S \in \binom{P(i)}{k(i)} \tag{3b}$$

$$\alpha_v, \beta_e, \tau_i \geq 0, \quad \forall v \in V, \forall e \in E, \forall i \in I \tag{3c}$$

Later, in Section III-A, we will focus on designing a separation oracle step by step for the dual LP to satisfy the third condition. Then, since $\hat{x}_S^i$ output by the PDA may be fractional, infeasible to the ACER, we propose a bi-criteria approximation algorithm via randomized rounding in Section III-B. Nonetheless, the randomized rounding algorithm may violate some constraints moderately. Therefore, we provide a heuristic algorithm in Section III-C to refine the solution.

Remark that the PDA iteratively selects a path set and increases the corresponding fractional variable $\hat{x}_S^i$. The increase is typically small, and the resources will not be occupied by an SD pair arbitrarily. Thus, later randomized rounding will make all SD pairs have an opportunity to compete for the resources and can effectively address the third challenge in the ACER.

A. The Separation Oracle

Given an arbitrary (fractional) solution (α, β, τ) for the dual LP, we attempt to design a separation oracle to determine whether a violated constraint exists. It is straightforward to find a violated constraint in (3c) in polynomial time if it exists. However, the challenge arises in identifying whether a violated constraint exists in (3b) since constraint (3b) may have an exponential number of inequalities.

Note that since the number of SD pairs is polynomial, our focus can narrow down to finding the most violated constraint for a specific SD pair $i \in I$. Because $r(i)$ in the denominator is a constant, this can be achieved by computing:

$$\min_{S \in \binom{P(i)}{k(i)}} \frac{\sum_{p \in S} \big(\sum_{(u,v) \in p} (\alpha_u + \alpha_v) + \sum_{e \in p} \beta_e \big) + \tau_i}{\Pr(S)}. \tag{4}$$

Nevertheless, computing Eq. (4) for a given SD pair i is a non-trivial task due to the specific coefficient $\frac{1}{\Pr(S)}$ associated with each path set S. It is impractical to examine every path set S since the number of path sets may be exponential.

To this end, we create an auxiliary undirected edge-weighted graph $\mathcal{G}_i$ such that a feasible path set for each SD pair $i \in I$ corresponds to a path from a virtual node $\hat{s}_i$ to another one $\hat{d}_i$ in $\mathcal{G}_i$. First, we clone every node and edge in the network G for $k(i)$ times and then add them to $\mathcal{G}_i$. Then, the cost and success probability of each generated edge e' between the nodes in each copies in $\mathcal{G}_i$ corresponding to the edge e between nodes u and v in G are set by Eqs. (5) and (6), respectively, as follows:

$$\mathcal{W}(e') = \alpha_u + \alpha_v + \beta_e \tag{5}$$

$$\mathcal{P}(e') = \begin{cases} \Pr(e) & \text{if } (u,v) = (s_i, d_i) \\ \Pr(e) \cdot \sqrt{\Pr(v)} & \text{else if } u = s_i \text{ or } d_i \\ \Pr(e) \cdot \sqrt{\Pr(u)} & \text{else if } v = s_i \text{ or } d_i \\ \Pr(e) \cdot \sqrt{\Pr(u) \cdot \Pr(v)} & \text{otherwise.} \end{cases} \quad (6)$$

Then, to make $\mathcal{G}_i$ connected, for each integer $j \in [1, k(i)-1]$, an edge with a cost of $\frac{\tau_i}{k(i)-1}$ and a success probability of 1 is created between the corresponding node of d_i in the j-th copy of G and the corresponding node of s_i in the $(j+1)$-th copy of G. Last, let $\hat{s}_i$ (or $\hat{d}_i$) denote the corresponding node of s_i (or d_i) in the first (or the $k(i)$-th) copy of G, and let $P_{\mathcal{G}_i}(\hat{s}_i, \hat{d}_i)$ represent the set of all possible paths between $\hat{s}_i$ and $\hat{d}_i$ in $\mathcal{G}_i$. In this way, computing Eq. (4) is equivalent to computing:

$$\min_{p \in P_{\mathcal{G}_i}(\hat{s}_i, \hat{d}_i)} \frac{\sum_{e \in p} \mathcal{W}(e)}{\prod_{e \in p} \mathcal{P}(e)}, \quad (7)$$

However, solving Eq. (7) with the classical Dijkstra algorithm is still hard since the criterion lacks the monotonicity property.[3] To address these challenges, we adopt an approach similar to the one described in [15], [16] to take into account the impact of both the numerator and denominator (i.e., bi-objective) in Eq. (7). Based on [15], [16], the bi-objective shortest path can be obtained in polynomial time in expectation if the following properties are satisfied:

1) The first- and second-objective cost functions are additive.[4]
2) The bi-objective total cost function is quasiconcave.

To this end, we introduce a bi-objective total cost function $U_{\mathcal{G}_i}(x(p), y(p))$ to subtly calculate the total cost of a path p in $\mathcal{G}_i$, where $x(p)$ and $y(p)$ are the first- and second-objective cost functions of a path p in $\mathcal{G}_i$, respectively. After that, we reformulate Eq. (7) as the minimization problem $\min_{p \in P_{\mathcal{G}_i}(\hat{s}_i, \hat{d}_i)} U_{\mathcal{G}_i}(x(p), y(p))$. Specifically, the cost function is defined cleverly in Definition 4 to meet the above properties.

Definition 4. The bi-objective cost function of path p is

$$U(x(p), y(p)) = x(p) \cdot e^{y(p)}, \quad (8)$$

where $x(p)$ and $y(p)$ are the functions to get the two costs of path p by summing the corresponding costs of its edges in $\mathcal{G}_i$:

$$x(p) = \sum_{e \in p} \mathcal{W}(e) \text{ and } y(p) = \sum_{e \in p} -\ln \mathcal{P}(e). \quad (9)$$

In this way, the cost functions $x(p)$ and $y(p)$ are both additive, meeting the first property. Moreover, Lemma 1 guarantees that $U(x, y)$ is quasiconcave under the condition $x, y > 0$, satisfying the second property. The proof of Lemma 1 is presented in Appendix B of [12] due to the page limit. Thus, the path $p \in P_{\mathcal{G}_i}(\hat{s}_i, \hat{d}_i)$ that minimizes the cost function $U(x, y)$ can be obtained and transformed to the path set $S \in \binom{P(i)}{k(i)}$ that violates the constraint mostly for i in (3b). Remark that, the cost $x(p)$ reflexes the memory load, channel load, and demand satisfication, while the cost $y(p)$ exhibits the probability $\Pr(S)$. Thus, using a path set with lower corresponding $U(x, y)$ for each request can achieve better resources allocation to address the first and second challenges of ACER simultaneously.

Lemma 1. $U(x, y)$ is quasiconcave when $x, y > 0$.

Remark that the second property implies that the bi-objective shortest path must be one of extreme paths (i.e., the convex hull of all possible combinations of the first- and second-objective costs, each of which corresponds to a possible path). Finding the extreme paths one by one can be achieved by iteratively invoking Dijkstra algorithm [15], [16]. Since the number of extreme paths is $O(|V|^{0.5})$ in expectation based on [15], finding the bi-objective shortest path by examining every extreme path can be done in polynomial time in expectation.[5]

B. Randomized Rounding Algorithm

We obtain the (integral) solution of the ACER by rounding the near-optimal (fractional) solution $\hat{x}_S^i$ in a randomized manner. Let $\mathbb{S}(i)$ denote the collection of path sets with $\hat{x}_S^i > 0$ for each SD pair $i \in I$. It is worth noting that the size of $\mathbb{S}(i)$ is polynomial since the PDA terminates in polynomial time [14]. We then interpret $\hat{x}_S^i$ as the probability of selecting a path set for each SD pair i. Consider an example with $\mathbb{S}(i) = \{S_1, S_2, S_3\}$ for an SD pair i. Assume that the given solution is $\hat{x}_{S_1}^i = 0.3$, $\hat{x}_{S_2}^i = 0.4$, and $\hat{x}_{S_3}^i = 0.2$. Thus, the probabilities of selecting path sets S_1, S_2, and S_3 are 0.3, 0.4, and 0.2, respectively, while the probability of selecting no path set for SD pair i is $1 - 0.3 - 0.4 - 0.2 = 0.1$.

Theorem 2 shows that the algorithm is a $(O(1), O(\log |V|))$-approximation algorithm for the ACER, and the detailed proof is provided in Appendix C of [12] due to the page limit.

Theorem 2. The proposed randomized algorithm for the ACER is a $(O(1), O(\log |V|))$-approximation algorithm.

C. Heuristic Algorithm for Solution Improvement

Note that the rounded solution by randomized rounding may violate the memory and channel limits. First, we sort the nodes and links in non-increasing order of their load. Then, for each node (or link) in this order, we iteratively remove the path set with the lowest profit until the limit of the node (or link) is not violated. Second, we sort the SD pairs with no allocated path in non-increasing order of their profit. For each pair in this order, we examine whether there is a path set S with variable $\hat{x}_S^i > 0$ and S can be accommodated in the residual graph. If so, then the pair is served by the path set S with the maximum $\Pr(S)$. The heuristic algorithm will stop if no such pair exists.

IV. Performance Evaluation

A. Simulation Settings

We compare our algorithm (simply called Ours in the following) with GREEDY [7], Q-CAST [8], and REPS [5]. Same

[3] In shortest path problems, the monotonicity criterion means that if p is a shortest path, then the subpaths of p must also be shortest paths.

[4] A cost function for a path is additive stating that the cost of a path is the sum of the cost of its edges.

[5] To bound the worst-case complexity, we can further generate a polynomial-sized set of ϵ-approximated extreme paths by [17] to get an approximated bi-objective shortest path for the PDA at a slightly higher bounded error [14].

as [5], [8], we use the Waxman model to build the networks for the following simulations. In each simulation, we only vary one parameter as the independent variable. The default setting is set as follows. We distribute 70 nodes randomly in each graph, which is an area with a size of 1000 km $\times$ 2000 km. To distribute those nodes to the graph, the average length of an edge $l(u, v)$ between any two nodes (u, v) is set to 600 km and the edge would exist with the probability $\delta e^{-l(u,v)/\epsilon L}$, where L is the farthest distance between any two nodes in the graph. The values of δ and ϵ are set to 0.85 and 0.4 respectively in the Waxman model. For each node, we randomly assign a value between 10 and 14 as the memory limit. As for each edge, the channel limit is assigned randomly between 4 and 8. Besides, γ is set to 0.0002, and thus the average success probability of entangling is 0.94, while the success probability of swapping $\mathcal{Q}(v)$ is between 0.8 and 1 for each node.

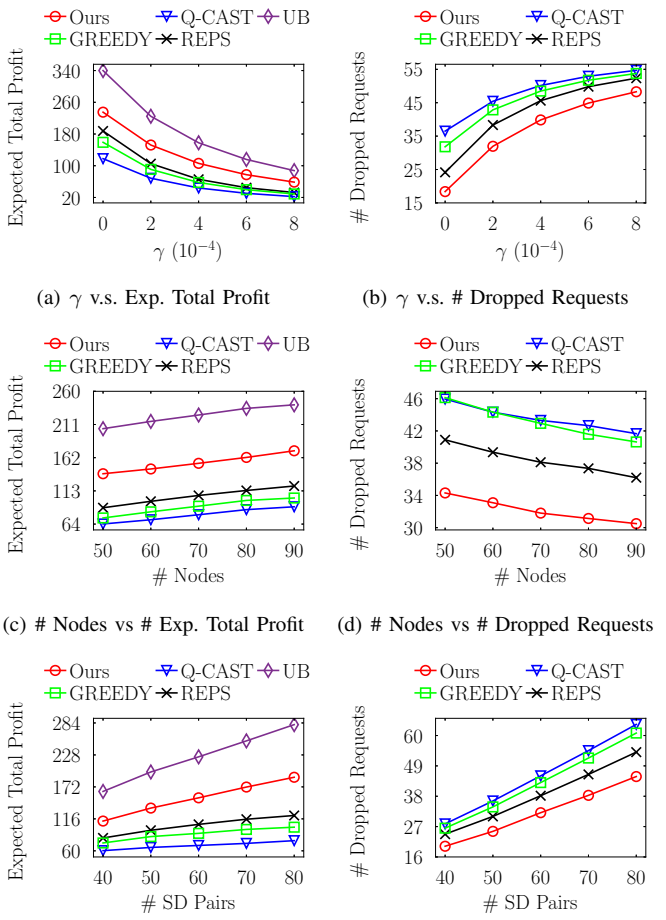
Then, we select 60 pairs of nodes randomly to be the SD pairs and assign its profit based on their willingness to pay. The users' willingness to pay for the completed tasks can be represented by the resource utility [18]. Hence, the willingness to pay is assigned by the number of the swapping and entangling the shortest path of the SD pair needs. However, users may compete for the resources due to the resource rarity. According to auction theory [19], auction prices of customers typically follow normal distribution. Therefore, we set the ratio of willingness to pay randomly between 1 and 3 by the right half part of the normal distribution model with the median of 1 and standard deviation of 1.5. Finally, the key metric "Expected Total Profit" is calculated by the expected profit sum of requests that we have allocated sufficient resources. Each result is averaged over 150 trials.

B. Numerical Results

Overall, Ours outperforms all the other algorithms, as shown in Figs. 2 and 3. Note that UB denotes the upper bound of the ACER, which is derived by solving the primal LP via the PDA [14]. By observation, Ours can improve the expected total profit and alleviate starvation with superior resource efficiency.

1) Effect of Probability and Resource: Figs. 2(a)−2(b) show the effect of success probability of entangling. As the success probabilities grow (i.e., lower γ), the expected total profit and request satisfaction increase because all entangled paths have a higher success probability of construction. Fig. 2(c)−2(d) show the effect of different numbers of nodes on expected total profit and request satisfaction. The network including more nodes within the same area has a higher density. Thus, more repeaters can be chosen such that the success probability of entangling also increases, leading to higher expected total profit and request satisfaction for all algorithms. Note that Ours can get the most appropriate path with the smallest U (i.e., the path with a lower load and a higher probability) by the separation oracle in Section III-A. By doing so, it overcomes the first challenge and outperforms GREEDY, Q-CAST, and REPS by up to 70%, 116%, and 46%, respectively.

2) Effect of Number of SD pairs: Fig. 2(e) illustrates the expected total profit with different numbers of SD pairs. More



(a) γ v.s. Exp. Total Profit (b) γ v.s. # Dropped Requests

(c) # Nodes vs # Exp. Total Profit (d) # Nodes vs # Dropped Requests

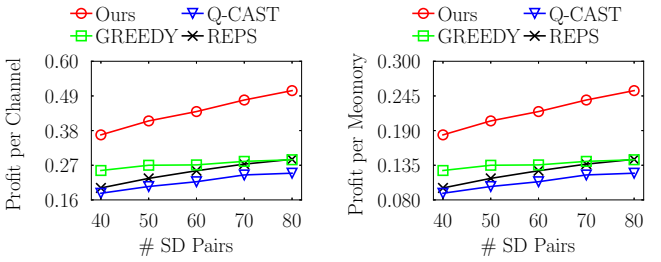
(e) # SD Pairs v.s. Exp. Total Profit (f) # SD Pairs v.s. # Dropped Requests

Fig. 2. Expected Total Profit and Dropped Requests with different parameters.

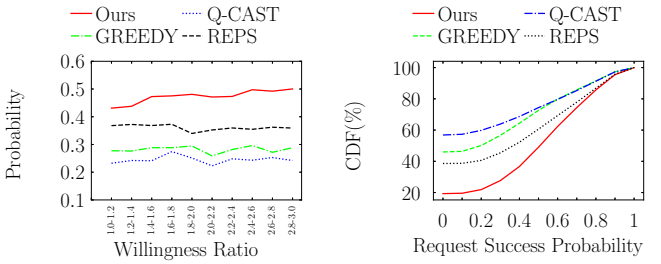
SD pairs result in more resource competition. Ours performs better since it has an overview given by the fractional solution in LP to choose the path set with a moderate success probability and more profitable request, and can circumvent hot points. Thus, Ours allocates resources to proper requests yield the most expected total profit to address third challenge. In contrast, the others may pour massive resources into the most likely saturated SD pair. Fig. 2(f) depicts the number of dropped request, showing that GREEDY, Q-CAST, and REPS sacrifice more SD pairs, so Ours has the most expected total profit.

3) Resource Efficiency: Figs. 3(a)−3(b) show the average resource efficiency, which is measured by dividing the expected total profit by the total number of used channels or memory units. More SD pairs provide more resource allocation selections, leading to more opportunities to increase resource efficiency. Ours performs better because it does not exhaust resources for specific pairs due to the consideration of profit, addressing the second challenge, as described in Section III-A.

4) Effect of Willingness Ratio and Request Success Probability: Fig. 3(c) shows the average success probability of requests with a different willingness ratio. Ours aims to maximize the expected total profit and thus tends to fulfill those requests with a high willingness ratio. Therefore, requests with



(a) # SD Pairs v.s. Profit per Channel (b) # SD Pairs v.s. Profit per Memory



(c) Willingness Ratio v.s. Probability (d) CDF of Request Success Prob.

Fig. 3. Effect of different parameters on different metrics.

a willingness ratio between 2.4 and 3.0 have a higher success probability than the others, while between 1.0 and 1.4 have the lowest probability. In contrast, the other algorithms have nearly the same success probability on every willingness ratio since they do not consider it, thus spending the memory and channels on less worthy requests. Fig. 3(d) depicts the CDF of success probability of requests, showing that the other algorithms select the path sets with a lower success probability, while Ours chooses the ones with a higher success probability. Therefore, the advantage of using Ours becomes more apparent, as it can strike a better balance between resource allocation and request satisfaction, leading to lower wastage and higher network efficiency in QNs.

V. Related Work

Elliott *et al.* introduced QNs to establish secure communications [20]. Meter *et al.* proposed an extensive QN architecture employing layered recursive repeaters, where repeaters may not trust each other [21], and devised new protocol layers to support quantum sessions. Pirandola *et al.* discussed the limitations of quantum communications without repeaters [6]. Caleffi *et al.* designed a routing protocol to ensure the highest end-to-end entanglement rate between nodes [22]. While many researches concentrated on specific network topologies, such as chain, diamond, ring, and star [22]–[25], some related researches delved into general topologies as follows. Pant *et al.* introduced a greedy algorithm for establishing minimum-hop entangled paths [7]. Shi *et al.* developed a routing method termed Q-CAST based on the Dijkstra algorithm to identify primary paths and recovery paths to mitigate the impact of entanglement failures [8]. Zhao *et al.* proposed an LP-based algorithm known as REPS to maximize throughput in QNs with a central controller [5]. However, none of these works take into account the all-or-nothing effect in QNs.

VI. Conclusion

This paper formulates a novel problem formulation (i.e., the ACER) to maximize the (expected) total profit earned by satisfying the all-or-nothing requests for all SD pairs under the memory and channel limits in a QN. To deal with the problem, we use the primal-dual algorithm and cleverly design the separation oracle for the constraints of the dual problem. Then, we devise a randomized rounding and prove that it is a $(O(1), O(\log |V|))$-approximation algorithm for the ACER. We also provide a heuristic algorithm to refine the constraint violations. Finally, simulation results show that our algorithm can outperform the existing approaches by up to 46%.

References

[1] S. Wehner, D. Elkouss, and R. Hanson, "Quantum internet: A vision for the road ahead," *Science*, vol. 362, no. 6412, 2018.

[2] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information*. American Association of Physics Teachers, 2002.

[3] N. Sangouard *et al.*, "Quantum repeaters based on atomic ensembles and linear optics," *Rev. Mod. Phys.*, vol. 83, p. 33, 2011.

[4] A. S. Cacciapuoti *et al.*, "When entanglement meets classical communications: Quantum teleportation for the quantum internet," *IEEE Trans. Comm.*, vol. 68, pp. 3808–3833, 2020.

[5] Y. Zhao and C. Qiao, "Redundant entanglement provisioning and selection for throughput maximization in quantum networks," in *IEEE INFOCOM*, 2021.

[6] S. Pirandola *et al.*, "Fundamental limits of repeaterless quantum communications," *Nat. Commun.*, vol. 8, 2017.

[7] M. Pant *et al.*, "Routing entanglement in the quantum internet," *NPJ Quantum Inf.*, vol. 5, pp. 1–9, 2019.

[8] S. Shi and C. Qian, "Concurrent entanglement routing for quantum networks: Model and designs," in *ACM SIGCOMM*, 2020.

[9] C. H. Bennett *et al.*, "Experimental quantum cryptography," *J. Cryptol.*, vol. 5, pp. 3–28, 1992.

[10] H. J. Kimble, "The quantum internet," *Nature*, vol. 453, pp. 1023–1030, 2008.

[11] I. Parvez *et al.*, "A survey on low latency towards 5G: RAN, core network and caching solutions," *IEEE Commun. Surv. Tutor.*, vol. 20, pp. 3098–3130, 2018.

[12] C.-T. Wei *et al.*, "All-or-nothing concurrent entanglement routing (full version)," Nov 2023. [Online]. Available: https://reurl.cc/K4lYKm

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*. The MIT Press, 2009.

[14] N. Garg and J. Könemann, "Faster and simpler algorithms for multicommodity flow and other fractional packing problems," *SIAM J. Comput.*, vol. 37, pp. 630–652, 01 2007.

[15] M. I. Henig, "The shortest path problem with two objective functions," *Eur. J. Oper. Res.*, vol. 25, no. 2, pp. 281–291, 1986.

[16] A. Sedeño-Noda and A. Raith, "A dijkstra-like method computing all extreme supported non-dominated solutions of the biobjective shortest path problem," *Comput Oper Res.*, vol. 57, pp. 83–94, 2015.

[17] S. Vassilvitskii and M. Yannakakis, "Efficiently computing succinct trade-off curves," *Theor. Comput. Sci.*, vol. 348, no. 2, pp. 334–356, 2005.

[18] Y. Lee *et al.*, "Quantum network utility: A framework for benchmarking quantum networks," *arXiv preprint arXiv:2210.10752*, 2022.

[19] V. Krishna, *Auction theory*. Academic press, 2009.

[20] C. Elliott, "Building the quantum network," *New J. Phys.*, vol. 4, p. 46, 2002.

[21] R. V. Meter and J. Touch, "Designing quantum repeater networks," *IEEE Commun. Mag.*, vol. 51, pp. 64–71, 2013.

[22] M. Caleffi, "Optimal routing for quantum networks," *IEEE Access*, vol. 5, pp. 22 299–22 312, 2017.

[23] S. Pirandola, "End-to-end capacities of a quantum communication network," *Commun. Phys.*, vol. 2, p. 51, 2019.

[24] E. Schoute *et al.*, "Shortcuts to quantum network routing," *arXiv preprint arXiv:1610.05238*, 2016.

[25] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, "On the stochastic analysis of a quantum entanglement distribution switch," *IEEE Trans. Quantum Eng.*, vol. 2, pp. 1–16, 2021.

APPENDIX A
PROOF OF THEOREM 1

We prove the theorem by reducing the decision version of integer minimum-capacity multicommodity flow problem (IMCMFP) to the ACER. In the decision version of IMCMFP, the input data includes an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a set of SD pairs $\mathcal{I}$, and the link load limit $\mathcal{L}$. It aims to decide whether it is possible to find a simple path for each SD pair $i \in \mathcal{I}$, while ensuring the max link load is no greater than $\mathcal{L}$.

We show how to construct a corresponding instance $T = \{G = (V, E), I, k(i), r(i), m(v), c(e), \Pr(v), \Pr(e)\}$ of the ACER in polynomial time for any given instance $\mathcal{T} = \{\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{I}, \mathcal{L}\}$ of the IMCMFP. First, we create a quantum node v, add it to V, and set its memory limit $m(v)$ to a sufficiently large value (e.g., $2|\mathcal{I}|$) for each node in $\mathcal{V}$. Then, we create an edge e to connect the corresponding quantum nodes u and v, add it to E, and set its channel limit $c(e)$ to $\mathcal{L}$ for each edge in $\mathcal{E}$ that connects the nodes u and v in $\mathcal{G}$. The entangle probability $\Pr(e)$ of every edge $e \in E$ and the swapping probability $\Pr(v)$ of every node $v \in V$ are set to one. Finally, we create an SD pair i with a demand $k(i) = 1$ and a weight $r(i) = 1$ and add it to I for each SD pair $i \in \mathcal{I}$. Clearly, the above reduction is in polynomial time.

We then demonstrate that the maximum load in the instance $\mathcal{T}$ is not greater than $\mathcal{L}$ if and only if the maximum expected profit in T is equal to the number of SD pairs (i.e., $|I|$). Suppose there is a solution with the maximum load no greater than $\mathcal{L}$ for the instance $\mathcal{T}$. In this case, each SD pair in T can have a corresponding path to transmit one qubit with a probability of 1. Consequently, a total profit of $|I|$ can be acquired through the same paths as in $\mathcal{T}$ of the IMCMFP. Conversely, assume that the maximum load in the instance $\mathcal{T}$ is greater than $\mathcal{L}$. Since the number of channels on each edge is set to $\mathcal{L}$, it is impossible to satisfy the demands of all SD pairs in T without violating the channel limit. Therefore, the total expected profit must be less than $|I|$. It completes the reduction, and the theorem follows.

APPENDIX B
PROOF OF LEMMA 1

It suffices to show that given any $x_1, y_1, x_2, y_2 > 0$ such that $U(x_1, y_1) \geq U(x_2, y_2)$, the inequality $U'_x(x_2, y_2)(x_1 - x_2) + U'_y(x_2, y_2)(y_1 - y_2) \geq 0$ always holds, where $U'_x(x, y) = \frac{\partial U(x,y)}{\partial x} = e^y$ and $U'_y(x, y) = \frac{\partial U(x,y)}{\partial y} = x \cdot e^y$. Then,

$$\begin{aligned} &U'_x(x_2, y_2)(x_1 - x_2) + U'_y(x_2, y_2)(y_1 - y_2) \\ &= e^{y_2}(x_1 - x_2) + x_2 e^{y_2}(y_1 - y_2) \\ &= x_2 e^{y_2}(\frac{x_1}{x_2} - 1 + y_1 - y_2) \geq x_2 e^{y_2}(\frac{x_1}{x_2} - 1 + \ln \frac{x_2}{x_1}) \geq 0. \end{aligned}$$

The second to last inequality is correct because $U(x_1, y_1) \geq U(x_2, y_2)$ implies $x_1 e^{y_1} \geq x_2 e^{y_2}$, leading to $y_1 - y_2 \geq \ln \frac{x_2}{x_1}$. Plus, the last one holds since $x + \ln \frac{1}{x} - 1 \geq 0$ as $x > 0$.

APPENDIX C
PROOF OF THEOREM 2

It is clear that our algorithm chooses at most one path set for each SD pair by randomized rounding, meeting constraint (1d). To prove Theorem 2, we have to guarantee the probability of violating the memory or channel limit (i.e., constraints (1b) and (1c)) within a bounded factor is sufficiently large. Thus, we employ the Chernoff bound (i.e., Theorem 3) to bound the violation. Subsequently, we analyze the time complexity and approximation ratio of our algorithm to complete the proof.

Theorem 3 (Chernoff bound)**.** There is a set of n independent random variables $x_1, ..., x_n$, where $x_i \in [0, 1]$ for each $i \in [1, n]$. Let $\mu = \mathbb{E}[\sum_{i=1}^{n} x_i]$. Then,

$$\Pr\left[\sum_{i=1}^{n} x_i \geq (1 + \epsilon)\mu\right] \leq e^{\frac{-\epsilon^2 \mu}{2+\epsilon}}. \tag{10}$$

1) Channel limit constraint: We first bound the extent to which the channel limits are violated. For each edge $e \in E$, we define a random variable z_e^i to denote the number of paths visiting e in the path set $S \in \binom{P(i)}{k(i)}$ selected for SD pair i after randomized rounding. Therefore, it can be written as:

$$z_e^i = \begin{cases} \sum_{p \in S: e \in p} 1 & \text{with } \Pr = \hat{x}_S^i, \ \forall S \in \binom{P(i)}{k(i)}; \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, the random variables z_e^i are independent for each edge $e \in E$. Note that their sum $Z_e = \sum_{i \in I} z_e^i$ represents the exact number of qubits planned to be transmitted through edge e after randomized rounding. Then, we prove the claim that the probability of violating any edge's channel limit by more than a factor of $(1 + 6 \ln |V|)$ is at most $\frac{1}{|V|^2}$, which is negligible.

We first derive the upper bound of the expectation of the transmitted qubits over an edge e for randomized rounding.

$$\begin{aligned} \mathbb{E}[Z_e] &= \mathbb{E}\left[\sum_{i \in I} z_e^i\right] = \sum_{i \in I} \mathbb{E}[z_e^i] \\ &= \sum_{i \in I} \sum_{S \in \binom{P(i)}{k(i)}} \sum_{p \in S: e \in p} \hat{x}_S^i \leq c(e). \end{aligned} \tag{11}$$

Note that the last inequality directly follows the channel limit constraint (1c). Subsequently, we find the upper bound of the probability of violating $c(e)$ of any edge e after rounding by the Chernoff bound as follows.

$$\begin{aligned} &\Pr(\exists e \in E : Z_e \geq (1 + 6 \ln |V|) \cdot c(e)) \\ &\leq \sum_{e \in E} \Pr(Z_e \geq (1 + 6 \ln |V|) \cdot c(e)) \\ &= \sum_{e \in E} \Pr(\sum_{i \in I} \frac{z_e^i}{c(e)} \geq 1 + 6 \ln |V|) \\ &\leq |V|^2 \cdot e^{\frac{-(6 \ln |V|)^2}{2 + 6 \ln |V|}} \leq |V|^2 \cdot e^{-4 \ln |V|} = |V|^{-2}. \end{aligned}$$

Note that the inequality $\frac{z_e^i}{c(e)} \leq 1$ holds to ensure every random variable is within the range of $[0, 1]$, meeting the Chernoff bound's requirement since it is reasonable to assume that the

demand of each pair i is much lower than the number of channels of each link e, i.e., $k(i) \ll c(e)$. In addition, the third inequality holds when $|V| \geq 3$. Thus, the claim holds.

2) Memory limit constraint: We then consider the memory limit constraint and bound the probability that the amount of quantum memory planned to be used in node v exceeds $m(v)$. Similarly, for each node v, a random variable z_v^i is defined to denote the amount of quantum memory of v used by the paths in the path set $S \in \binom{P(i)}{k(i)}$ selected for SD pair i after randomized rounding. In this way, it can be written as:

$$z_v^i = \begin{cases} \sum_{u \in V} \sum_{p \in S:(u,v) \in p} 1 & \text{with } \Pr = \hat{x}_S^i, \ \forall S \in \binom{P(i)}{k(i)}; \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, the random variables z_v^i are independent for each node $v \in V$. Note that their sum $Z_v = \sum_{i \in I} z_v^i$ is exactly the amount of memory planned to transmit data qubits after randomized rounding. Next, we prove the claim that the probability of violating any node's memory limit by more than a factor of $(1 + 5 \ln |V|)$ is at most $\frac{1}{|V|^2}$.

We first acquire the upper bound of the expectation of the occupied memory at node v after rounding is:

$$\mathbb{E}\left[Z_v\right] = \mathbb{E}\left[\sum_{i \in I} z_v^i\right] = \sum_{i \in I} \mathbb{E}[z_v^i] \leq m(v).$$

Then, similarly, the probability of violating the memory limit of any node v by more than a factor of $(1 + 5 \ln |V|)$ after rounding by the Chernoff bound can be expressed as:

$$\begin{aligned} & \Pr(\exists v \in V : Z_v \geq (1 + 5 \ln |V|) \cdot m(v)) \\ \leq \ & \Pr(Z_v \geq (1 + 5 \ln |V|) \cdot m(v)) \\ = \ & \sum_{v \in V} \Pr(\sum_{i \in I} \frac{z_v^i}{m(v)} \geq 1 + 5 \ln |V|) \\ \leq \ & |V| \cdot e^{\frac{-(5 \ln |V|)^2}{2 + 5 \ln |V|}} \leq |V| \cdot e^{-3 \ln |V|} = |V|^{-2}. \end{aligned}$$

Similarly, $\frac{z_v^i}{m(v)} \leq 1$ since in most cases the demand of each pair i is much lower than the amount of memory of each node v, i.e., $k(i) \ll m(v)$. Also, the third inequality holds as $|V| \geq 3$. Therefore, the claim also holds.

3) Time complexity: We analyze the time complexity of the proposed algorithm. According to [14], the PDA executes $O(\omega^{-2} m_1 \log m_1)$ times of separation oracle and thus outputs $O(\omega^{-2} m_1 \log m_1)$ path sets for all SD pairs. Note that ω is the user-defined error bound of primal LP solution and m_1 is the number of constraints (except for the non-negativity constraints of variables) in the primal LP, i.e., $m_1 = (|V| + |E| + |I|)$. In addition, since $k(i)$ for each request $i \in I$ is a small constant, each time of separation oracle takes $m_2 = O(|I| \cdot |V|^{0.5}(|E| + |V| \log |V|))$ in expectation based on [15], [16]. Then, in our algorithm, the PDA takes $O(\omega^{-2} m_1 m_2 \log m_1)$, and the randomized rounding spends $O(\omega^{-2} m_1 \log m_1)$. The overall time complexity is $O(\omega^{-2} m_1 m_2 \log m_1)$. Note that the worst-case time complexity can be further bounded if ϵ-approximated extreme paths [17] are used.

So far, it suffices to focus on the value of the objective function of the primal LP.

4) Approximation ratio on the value of objective: Let OPT and OPT_{PL} be the optimum values of the ACER and our primal LP, respectively. Clearly, $OPT \leq OPT_{PL}$ since our primal LP is a relaxation of our ILP. In addition, let $\hat{x}_S^i$ denote the solution output by the primal-dual algorithm, and let $\hat{X} = \sum_{i \in I} \sum_{S \in \binom{P(i)}{k(i)}} \Pr(S) \cdot r(i) \cdot \hat{x}_S^i$. Then, let $x_S'^i$ be the solution after randomized rounding, and let $X' = \sum_{i \in I} \sum_{S \in \binom{P(i)}{k(i)}} \Pr(S) \cdot r(i) \cdot x_S'^i$ be the value of rounded solution. The expected value of the solution after randomized rounding is equal to the solution of the primal LP derived by the PDA. In other word,

$$\begin{aligned} \mathbb{E}[X'] &= \mathbb{E}\Big[\sum_{i \in I} \sum_{S \in \binom{P(i)}{k(i)}} \Pr(S) \cdot r(i) \cdot x_S'^i\Big] \\ &= \sum_{i \in I} \sum_{S \in \binom{P(i)}{k(i)}} \Pr(S) \cdot r(i) \cdot \mathbb{E}\big[x_S'^i\big] \\ &= \sum_{i \in I} \sum_{S \in \binom{P(i)}{k(i)}} \Pr(S) \cdot r(i) \cdot \hat{x}_S^i \\ &= APP_{PL} \geq \frac{OPT_{PL}}{1 + \omega} \geq \frac{OPT}{1 + \omega} = O(1) \cdot OPT, \end{aligned}$$

where APP_{PL} is the solution value of primal LP acquired by the PDA and $\omega > 0$ is a user-defined constant. Note that the first inequalities hold according to [14].