Quantum Switch Scheduling for Information Qubits

Panagiotis Promponas
Yale University
New Haven, Connecticut, USA

Víctor Valls
IBM Research Europe
Dublin, Ireland

Leandros Tassiulas
Yale University
New Haven, Connecticut, USA

ABSTRACT

This paper studies the problem of designing a scheduling policy for a quantum switch that teleports information qubits. The problem is analogous to scheduling photons in high-speed optical switches, with the difference that an optical switch is modeled as a bipartite graph instead of a complete graph. The paper's contributions are to model the problem of designing a scheduling policy as decomposing a *symmetric* doubly stochastic matrix, and to show that we can use Birkhoff's algorithm to obtain such a decomposition.

1. INTRODUCTION

We study the problem of designing a scheduling policy for a quantum switch that needs to teleport information qubits. In brief, a quantum switch is a star graph connected to clients with dedicated links (see Figure 1), where links are used for creating link-level entanglements (LLEs) between the switch and the clients [1]. The task of the switch is to perform Bell State Measurements (BSMs) with the LLEs, to create end-to-end entanglements that will be used by quantum applications, e.g., quantum key distribution (QKD) or distributed quantum computing (DQC).

The problem we want to solve is the following. For a given arrival process of end-to-end entanglement requests, we want to find an entanglement swapping policy that maximizes the switch utilization. This problem has been addressed in previous work with queue-based scheduling approaches that treat requests as packets in traditional communication networks [1–4]. In particular, requests are accumulated in queues with unlimited storage capacity, and the goal is to design a policy that keeps the queues stable. While these approaches are well-suited when requests for end-to-end entanglements are classical information, they are not ideal when requests are quantum information (i.e., qubits) that has to be teleported. There are two reasons for this. Firstly, storing information qubits for an extended period is generally not possible due to decoherence. Secondly, quantum memory is a scarce resource, and therefore, employing a queuing approach that measures performance based on a queue stability criterion is inadequate.

The problem of scheduling information qubits is analogous to scheduling problems in high-speed optical networks, where photons cannot be stored and must be forwarded directly [5]. The latter is the case in DQC, where the qubits at the output of a circuit must be teleported to a circuit located in another quantum processor.

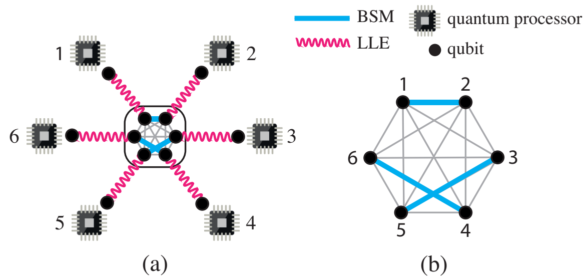

Figure 1: (a) Quantum switch with 6 clients. (b) A matching in a complete graph with 6 nodes. An edge in the matching corresponds to carrying out a BSM with the clients' LLEs.

In this paper, we study the problem of designing a scheduling policy for a quantum switch where requests are information qubits. The problem is similar to designing a scheduling policy for an optical switch [5, 6], with the difference that a quantum switch is modeled as a star graph instead of a bipartite graph. The paper's contributions are to model the problem of designing a scheduling policy as decomposing a *symmetric* doubly stochastic matrix as the weighted sum of *symmetric* permutations, and to show we can use Birkhoff's algorithm to obtain such decomposition. Solving such a problem is not straightforward since, unlike the classical non-symmetric case [7,8], it is not possible to decompose *any* symmetric doubly stochastic as the convex combination of symmetric permutation matrices [9].

2. PROBLEM FORMULATION

2.1 Switch operation

We consider a quantum switch connected to n clients that operates in intervals. At the beginning of each interval, we are given an $n \times n$ traffic demand matrix of the form:

$$D = \begin{bmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 \\ 0 & 1 & 2 & 0 \end{bmatrix}. \tag{1}$$

The demand matrix is symmetric, where $D(a, b)$ with $a, b \in \{1, \dots, n\}$ indicates the number of qubits that need to be teleported between clients a and b. Also, D has zeroes in the diagonal since the origin and destination of an end-to-end entanglement cannot be the same.

During the interval, the switch has to make scheduling decisions to teleport the qubits indicated in the demand matrix D. A scheduling decision correspond to selecting a matching in the complete graph, which indicates the requests/qubits to serve (see Figure 1). Note also that we can write a *maximal* matching as a symmetric permutation matrix when n is even. When the matching is not maximal or n is not even, we can map the matching to a "substochastic" permutation matrix: a symmetric binary matrix where the sum of the columns/rows is equal to zero or one. The first step is to decompose a demand matrix D as the sum of such permutation matrices. For example, we can write the demand matrix D in (1) as

$$D = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2}$$

When the schedules have been computed, we can serve the qubit requests when enough LLEs become available. Once all the scheduling decision have been made, the process is restarted. That is, we get a new demand matrix, decompose it, and generate the corresponding collection of schedules/matchings to teleport qubits.

2.2 Abstract mathematical problem

Let $\mathbf{S}_n$ be the convex hull of symmetric permutation matrices with zeroes in the diagonal, representing the set of normalized demand matrices. Also, let $S^\star$ be a matrix in $\mathbf{S}_n$.[1] The mathematical problem we want to solve is the following. For a given matrix $S^\star \in \mathbf{S}_n$, we want to find a collection of *symmetric* permutation matrices $P_1, P_2, \dots, P_k$ and weights $\theta_1, \theta_2, \dots, \theta_k > 0$ with $\sum_{i=1}^k \theta_i = 1$ such that

$$S^\star = \sum_{i=1}^{k} \theta_i P_i. \tag{3}$$

The problem above is similar to the problem addressed by Birkhoff, with the difference that permutation matrices must be symmetric and that $S^\star$ must be in a *subset* of symmetric and doubly stochastic (i.e., $\mathbf{S}_n$). The latter is important because unlike the classical non-symmetric case [7], it is not possible to decompose any symmetric doubly stochastic as the convex combination of symmetric permutation matrices. In particular, the extreme points of the set of symmetric doubly stochastic matrices have the form $\frac{1}{2}(\hat{P} + \hat{P}^T)$ where $\hat{P}$ is a permutation (not necessarily symmetric) [9].

3. ALGORITHM

We show how to use Birkhoff's algorithm (Algorithm 1) to decompose a *symmetric doubly stochastic matrix* $S^\star \in \mathbf{S}_n$ as the weighted sum of *symmetric permutation matrices*.

Algorithm 1 takes as input a matrix $S^\star \in \mathbf{S}_n$ and consists of four main steps. The first step is to find a maximum weighed matching in the complete graph with the weights $S^\star(a,b) - S_{k-1}(a,b)$ while ignoring the edges that have zero weight. The second step is to map the obtained matching to a symmetric permutation matrix. If the matching is not maximal, the symmetric "permutation" matrix will be substochastic. The third and fourth steps are as in Birkhoff's algorithm. We select a weight that is equal to the smallest component of $S_{k-1}(a,b)$ and update the decomposition as indicated in (5). Algorithm 1 obtains a decomposition trivially with at most $n^2/2$ schedules. And when the matchings computed in step 1) of the algorithm are always maximal, we have the following result:

[1] For example, $S^\star = D/3$ where D is as in (1).

Algorithm 1 Birkhoff algorithm for symmetric doubly stochastic matrices

Input: Symmetric doubly stochastic matrix $S^\star$
Set: $k = 1$ and $S_0 = \{0\}^{n \times n}$
while $\|S_{k-1} - S^\star\|_F > 0$ **do**
1) Define $R_k(a,b) := S^\star(a,b) - S_{k-1}(a,b)$ where $a, b \in \{1, \dots, n\}$. Find a maximum weighted matching in the n-complete graph with weights

$$W(a,b) = \begin{cases} R_{k-1}(a,b) & \text{if } R_{k-1}(a,b) > 0 \\ -\infty & \text{otherwise} \end{cases} \tag{4}$$

2) Map the matching to a symmetric permutation P_k
3) Select θ_k equal to the smallest value of $S_k(a,b)$ such that $P_k(a,b) = 1$.
4) Update the decomposition:

$$S_k \leftarrow S_{k-1} + \theta_k P_k \tag{5}$$

5) $k \leftarrow k + 1$
end while
return $(P_1, \dots, P_{k-1})$, $(\theta_1, \dots, \theta_{k-1})$

THEOREM 1. *Suppose that the matchings in step 1) in Algorithm 1 are always maximal. Then, Algorithm 1 obtains a decomposition with at most* $(n-1)^2/2 + 1$ symmetric *permutations matrices.*

PROOF (SKETCH). To start, note that any (symmetric) permutation matrix P satisfies $\|P\|_F^2 = n$, where $\|\cdot\|_F$ is the Frobenious norm. Next, observe that

$$\begin{aligned}
&\|S_k - S^\star\|_F^2 \\
&\quad = \|S_{k-1} + \theta_k P_k - S^\star\|_F^2 \\
&\quad = \|S_{k-1} - S^\star\|_F^2 + \theta_k^2 \|P_k\|_F^2 \\
&\qquad + 2\theta_k \textstyle\sum_{a,b} P_k(a,b)(S_{k-1}(a,b) - S^\star(a,b)) \\
&\quad \le \|S_{k-1} - S^\star\|_F^2 + \theta_k^2 \|P_k\|_F^2 - 2\theta_k^2 \textstyle\sum_{a,b} P_k(a,b) P_k(a,b) \\
&\quad = \|S_{k-1} - S^\star\|_F^2 + \theta_k^2 \|P_k\|_F^2 - 2\theta_k^2 \|P_k\|_F^2 \\
&\quad = \|S_{k-1} - S^\star\|_F^2 - \theta_k^2 \|P_k\|_F^2 \\
&\quad = \|S_{k-1} - S^\star\|_F^2 - \theta_k^2 n
\end{aligned}$$

where the inequality is due to steps 1) and 3) in Algorithm 1. Thus,

$$\|S_k - S^\star\|_F^2 - \|S_{k-1} - S^\star\|_F^2 \le -\theta_k^2 n. \tag{6}$$

Importantly, $\theta_k > 0$ by step 3) since, in step 1), the matching selected does not contain edges that have zero weight.

The rest of the proof follows by the standard Birkhoff arguments. In particular, by selecting θ_k equal to the smallest value of $S_{k-1}(a,b)$ such that $R_k(a,b) > 0$, we make at least one entry in S_k equal to zero, i.e., we reduce the dimensionality of the residual by two since the matrix is symmetric.

The algorithm will terminate in at most $(n-1)^2/2+1$ iterations since the set of doubly stochastic matrices (or a subset of it), is embedded into an space of dimension $(n-1)^2$. $\square$

That is, we can obtain the same result of Birkhoff but for a subset of symmetric doubly stochastic matrices, i.e., not any symmetric doubly stochastic matrix. Importantly, the step in 1) does not guarantee that the matching is maximal. Having maximal matchings is important as it allows us to teleport as many qubits in parallel as possible. Note also that, by construction, a matching computed in step 1) of Algorithm 1 contains at least one edge since otherwise $S_{k-1} = S^\star$, i.e., the algorithm would have terminated in the previous iteration.[2]

4. DISCUSSION AND OPEN QUESTIONS

To minimize the time needed to teleport all the qubits in the demand matrix, we need to find a decomposition that uses as few schedules/matching as possible, and that those matching are maximal or near-maximal. However, the decomposition obtained in Algorithm 1 does not guarantee neither of those. An open research question is how to minimize the number of symmetric permutation matrices required in the decomposition, and how to ensure that the corresponding matchings have as many edges as possible. A similar problem has been studied in optical switches [10,11], where the minimization of schedules in the decomposition is important to reduce the switching costs [5]. However, in those problems, the matchings/schedules are always maximal, which we cannot guarantee with Algorithm 1.

5. REFERENCES

[1] Wenhan Dai, Anthony Rinaldi, and Don Towsley. Entanglement swapping in quantum switches: Protocol design and stability analysis. *arXiv preprint arXiv:2110.04116*, 2021.

[2] Nitish K Panigrahy, Thirupathaiah Vasantam, Don Towsley, and Leandros Tassiulas. On the capacity region of a quantum switch with entanglement purification. *arXiv preprint arXiv:2212.01463*, 2022.

[3] Thirupathaiah Vasantam and Don Towsley. A throughput optimal scheduling policy for a quantum switch. In *Quantum Computing, Communication, and Simulation II*, volume 12015, pages 14–23. SPIE, 2022.

[4] Panagiotis Promponas, Víctor Valls, and Leandros Tassiulas. Full exploitation of limited memory in quantum entanglement switching. *arXiv preprint arXiv:2304.10602*, 2023.

[5] He Liu, Matthew K Mukerjee, Conglong Li, Nicolas Feltman, George Papen, Stefan Savage, Srinivasan Seshan, Geoffrey M Voelker, David G Andersen, Michael Kaminsky, et al. Scheduling techniques for hybrid circuit/packet networks. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, pages 1–13, 2015.

[6] Fanny Dufossé and Bora Uçar. Notes on birkhoff–von neumann decomposition of doubly stochastic matrices. *Linear Algebra and its Applications*, 497:108 – 115, 2016.

[2]Also, $S_k(a,b) \leq S^\star(a,b)$ for all $a, b \in \{1, \ldots, n\}$ by construction.

[7] D. Birkhoff. Tres observaciones sobre el algebra lineal. *Universidad Nacional de Tucuman Revista , Serie A*, 5:147–151, 1946.

[8] Richard A. Brualdi. Notes on the birkhoff algorithm for doubly stochastic matrices. *Canad. Math. Bull.*, 25, 1982.

[9] Allan B Cruse. A note on symmetric doubly-stochastic matrices. *Discrete Mathematics*, 13(2):109–119, 1975.

[10] Fanny Dufossé and Bora Uçar. Notes on birkhoff–von neumann decomposition of doubly stochastic matrices. *Linear Algebra and its Applications*, 497:108–115, 2016.

[11] Víctor Valls, George Iosifidis, and Leandros Tassiulas. Birkhoff's decomposition revisited: Sparse scheduling for high-speed circuit switches. *IEEE/ACM Transactions on Networking*, 29(6):2399–2412, 2021.