Using Quantum Annealing
For Blockchain Acceleration Technology

Kota Mikami
Qubit chain
https://qubit-chain.info/

The current Bitcoin blockchain is facing a scalability problem. Because of this scalability problem, the Bitcoin blockchain is unable to meet the trading needs of the real world. The Lightning Network was developed to solve this problem. We have developed a method to efficiently search for routes in the Lightning Network using an annealing machine.

Background

The current Bitcoin blockchain faces a scalability problem. Due to this scalability problem, the Bitcoin blockchain is unable to meet the transaction needs of the real world. For example, while the number of transactions processed by the credit card company VISA is said to be approximately 4,000 (per second), the number of transactions processed by Bitcoin is approximately 7 (per second). Off-chain technology has been proposed as a solution to the scalability problem described above. One of the networks that uses this off-chain technology is the Lightning Network (hereafter referred to as LN).

By using an annealing machine, we will develop a more efficient method for LN pathfinding.

Method

We have developed a method for pathfinding of LN transactions by annealing machines. The developed method is divided into two stages. First, three candidate routes are created for each transaction. Next, an annealing machine is used to select the best route from among these candidates. We will now explain the method of selecting the optimal path using the annealing machine. First, the following two conditions must be satisfied by the transaction path in LN.

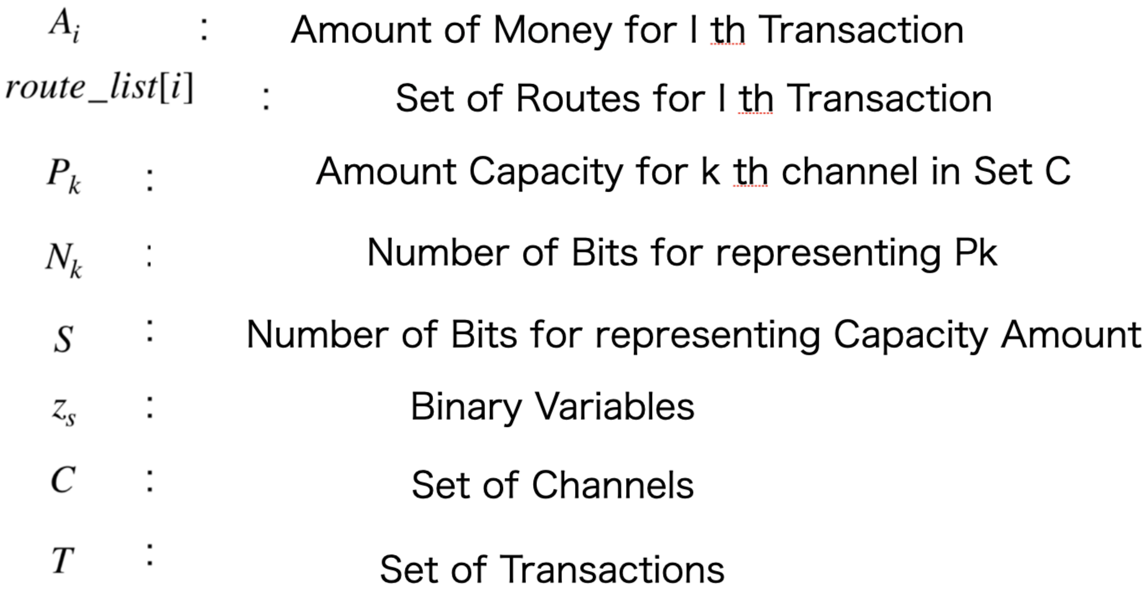
(1) The channel has capacity and no more transactions of any amount can pass through it

(2) There is only one path from the sending node to the receiving node

The unique computers participating in the network of 2 LNs are usually called nodes, and the transfer of money between these nodes is called a transaction. The path that this transaction follows is called a transaction. The path that this transaction follows is the channel that is connected between each node. The volume of transactions that a channel can handle is its capacity.

Capacity Hamiltonian

$$H_{capacity\_cost} = \sum_{k\in C}\left(\sum_{s\leq N_k} z_s 2^s - \sum_{i\in T}\sum_{j\in route\_list[i]} A_i x_{ij}\right)^2$$



A_i	:	Amount of Money for I th Transaction
$route\_list[i]$	:	Set of Routes for I th Transaction
P_k	:	Amount Capacity for k th channel in Set C
N_k	:	Number of Bits for representing Pk
S	:	Number of Bits for representing Capacity Amount
z_s	:	Binary Variables
C	:	Set of Channels
T	:	Set of Transactions

The total amount of transactions that can pass through the channel is determined at the time the bi-directional payment channel is created, and no more than this total amount can be transferred. In addition, a transaction can only follow one route from start to finish. In order to use an annealing machine to search for transaction routes, we formulated the above two constraints using binary variables to create a Hamiltonian. The Hamiltonian is a binary matrix of the number of transactions x the number of candidate routes, where =1 when the second transaction uses the second route, and =0 otherwise.

The constraint of（1）is expressed by the following Hamiltonian For all channels, a penalty is incurred if the capacity through the channel is greater than the capacity. We also used logarithmic encoding to express the capacity, which is a positive integer, in the smallest number of bits.

The following is the Hamiltonian of the route limit (2), where only one route is selected for each transaction. (2) The following is a Hamiltonian of route restriction, where only one route is selected for each transaction.

$$H_{const} = \sum_{i \in T} \left(\sum_{j \in route\_list[i]} x_{ij} - 1 \right)^2$$

T : Set of Transactions

$route\_list[i]$: Set of Routes for I th Transaction

Since a transaction is charged a fee each time it passes through a node, it is desirable to choose the shortest route possible to reduce the fee charged. We introduce a Hamiltonian to reduce the distance of the transaction path. The transaction distance referred to here refers to the number of channels to be traversed.

Based on the Hamiltonian of capacity cost, the Hamiltonian of route restrictions, and the Hamiltonian of distance cost, the following Hamiltonians were generated.

Hamiltonian of Distance Cost

$$H_{distance\_cost} = \sum_{i \in T} \sum_{j \in route\_list[i]} D_j x_{ij}$$

T : Set of Transactions

$route\_list[i]$: Set of Routes for I th Transaction

D_j : Distance of j th Route for Transaction

In this study, we used Toshiba's SBM to solve the routing problem defined by these Hamiltonians by searching for transaction paths in a simulated graph. The graph structure was generated using the Python library NetworkX2.2. We set the number of nodes to 2000 and the number of channels to 20000. The capacity of each channel was set to be in the range of 200 to 900, so that each node would have approximately 15 to 30 channels in random order.

The number of transactions was set to 4, and the range of the amount of each transaction was set to 200-600. For each transaction, we prepared three different route candidates. We used the Dijkstra method to find the shortest path in a graph in which the channel fee was multiplied by a random number with an average of 0.1 and a standard deviation of 1, and this was used as the candidate route. The parameter α was set to 2, and the parameter β was set to the square of the average transaction amount multiplied by 100 in order to have symmetry, since the Hamiltonian of capacity cost includes the amount of money for the square of the transaction amount and the output may become large.

$$H = H_{capacity\_cost} + \alpha H_{distance\_cost} + \beta H_{const}$$

As a result of calculating the above route search problem using the Toshiba Annealer's Solver mode number_iterations1000000, number_replices100, the solution was obtained in 10 seconds, including the communication time. In addition to the constraints described in (1) and (2), we were able to obtain a path with a short distance as a solution. The number of bits required was 361 bits.

We were able to use an annealing machine to solve a graph problem that mimics the LN graph. Although the number of transactions handled was small, we were able to gain a foothold in solving a graph that is about twice the size of the network of the LND development team, which is the top runner in the development of LNs, and we were able to demonstrate the applicability of the annealing machine to future LN routing problems.

The Lightning Network website[1] states that it is capable of processing millions or billions of simultaneous transactions, but considering the possibility that transactions within the network may become active and concentrate on specific channels, it is necessary to increase the probability of successful transactions and maximize the number of transactions in the network.

However, considering the possibility that transactions in the network will become more active and that transactions will be concentrated in certain channels, it is necessary to aim to increase the probability of successful transactions and maximize the number of transactions in the network. The fact that we were able to demonstrate the applicability of the annealing machine to the routing problem of LNs, which is said to become a problem as the volume of data increases in the future, has a significant impact.