On Cross-chain Pathfinding and Bridge Selection for Decentralized Finance

Ajmal Muhammad
Blockchain Research Engineer
Advanced Blockchain Research
ajmal@advancedblockchain.com

Jesper Kristensen
Head of Research
Advanced Blockchain Research
jesper@advancedblockchain.com

***Abstract*—With the current proliferation of L1 and L2 blockchains, the Decentralized Finance (DeFi) space is evolving toward a multi-chain future for the ecosystem. In such an environment, cross-chain bridging solutions are critical for seamless value-transfer across chains, enabling the true vision of DeFi. However, each bridging solution is typically optimized for a specific purpose (e.g., speed, fee, security, etc.) and connects a subset of blockchains while supporting a different set of assets for each chain. Hence, to transfer value between chains, we need to be able to choose the best bridge(s) and route among alternatives based on user-defined requirements. In this paper, we introduce the cross-chain *Pathfinding and Bridge Selection* (PBS) problem as a multi-connected bidirectional graph and formulate an Integer Linear Programming (ILP) solution to jointly optimize the path and bridge selection between source and destination chains. Our ILP formulation aims at transferring value between blockchains in such a way as to minimize the transfer fee or time, or a beneficial trade-off between these two. The ILP results compare favorably against a heuristic algorithm for the same problem and our experiments show a significant reduction in both transfer fee and time, particularly for large cross-chain networks.**

1. Introduction

The Bitcoin whitepaper [1] surfaced in 2008 and laid out the foundation for blockchain technology. Within 12 years, more than 100 active public blockchain networks have been built while numerous others will be launched soon [2]. Holistically, these blockchain networks attempt to strike a balance between the three key elements of *decentralization*, *scalability*, and *security*, which are described by Vitalik Buterin (Ethereum founder) as the "blockchain trilemma" [3]. For instance, the two main blockchain networks (i.e., Bitcoin and Ethereum) excel in decentralization and security while lagging in scalability.

To familiarize the reader with this blockchain trilemma, we define these terms in the context of blockchain networks:

1 **Decentralization**: equally distributing the network power among all the entities (e.g., miners, validators, etc.). No one entity can make the network behave in a certain way without approval from the rest.

2 **Scalability**: achieving higher transactions at a faster rate and lower network fees. So far, scalability is unattainable for Bitcoin and Ethereum base networks (or layer 1) with 5 and 7 transactions per second (tps), respectively.

3 **Security**: preventing networks from being disrupted by malicious attacks. Security is a must since without it blockchain networks would become untrustworthy and malware-infested. Attaining security with scalability is the most challenging task as both oppose each other's functionality.

To overcome the scalability problem, particularly for the Ethereum blockchain network, some solutions, including blockchain layer 2 (L2), have recently emerged [4]. To clarify the terminology, blockchain layer 1 (L1) refers to the base layer, i.e., the peer-to-peer network that brings all the blockchain nodes together into a single system and the underlying consensus mechanisms. On the other hand, L2 refers to the third-party solutions (e.g., protocols or networks) that function on top of a base chain (i.e., L1). For instance, Bitcoin's lightning network [5] or Ethereum's polygon [6], Plasma [7], Arbitrum [8], etc. L2 solutions alleviate the transaction encumbrance on the base chain to an external computing protocol that executes batches of transactions and reports back to the base chain (i.e., L1) the final results for the record.

Considering the different L1 and L2 chains, there is no perfect blockchain for all sectors as each chain is typically optimized for a specific purpose, such as financial transactions, gaming, trading, supply chain, etc. Hence, developers have embraced the uniqueness of each blockchain network for building Decentralized Applications (dApps) [9] and expanding their ecosystem. Fig. 1 shows a list of dApp types that have recently emerged. However, dApps designed for one network operate only within that ecosystem and could not interact with dApps built over other networks, resulting in the limitation of broader adoption of these dApps. Thus, there remains a huge gap in interconnection as most blockchain networks operate in silos and are not fundamentally enabled to interact with each other.

In this context, blockchain bridges are the most successful attempts to unify the blockchain landscape [11]. Blockchain bridges facilitate transfers of information between two or more blockchain networks using different interoperable models while retaining the fundamental princi-

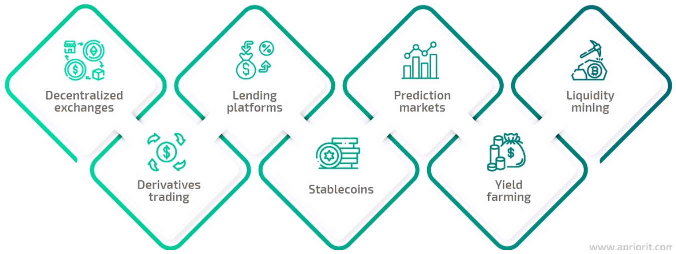

Figure 1. Categorization of existing DeFi protocols into different classes [9]. Decentralized exchanges protocols lead the pack with 386 protocols followed by yield farming protocols with number 312 then lending protocols with number 131 and so on [10].

ples of decentralization. The information can include assets, smart contract calls, proofs, or arbitrary data. Currently, there is a plethora of bridging solutions connecting different blockchains [12]. Fig. 2 illustrates a subset of these bridging solutions connecting different blockchain networks. As visible in Fig. 2, existing bridging solutions connect only a few blockchains and support the transfer of a subset of assets between different blockchains. Therefore, to transfer an asset from one blockchain to another that is not directly connected via a bridge, a dApp will need to choose the intermediate blockchains and bridges connecting them.

With the proliferation of new blockchain networks and bridges, it will be challenging to find a cost-efficient (in terms of overall transaction fee) and faster (in terms of transaction finality [13]) path and corresponding bridges across source and destination chains for transferring assets or other information. In other words, it will be more intricate to find the best path and bridge(s) within a complex web of blockchain networks and bridges.

This paper solves the pathfinding and bridge selection problem for inter-blockchain networks. Given a cross-chain network topology with multiple edges (i.e, bridges) having different cost and transaction finality values between any two adjacent nodes (i.e., blockchains), the amount of assets to be transferred from a source to destination nodes, our goal is to choose appropriate bridges and corresponding path created by those bridges with the objective of minimizing the total transaction cost and/or time for transaction completion. For this purpose, we formulate the cross-chain Pathfinding and Bridge Selection (PBS) problem as an Integer Linear Program (ILP) with the objective to minimize the total transaction cost and/or transaction finality time.

The remainder of this paper is structured as follows. An overview of cross-chain interoperability is presented in Section II, whereas, Section III describes PBS and modeling cross-chain networks as a graph network. Section IV presents our ILP formulation for the PBS problem while the heuristic approach for the PBS problem is described in Section V. Numerical results are analyzed in Section VI, with concluding remarks in Section VII.

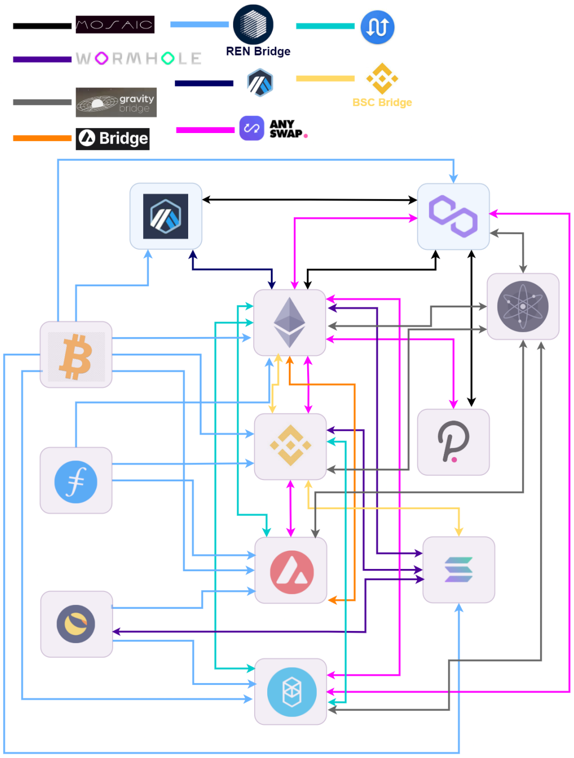

Figure 2. Graph representation of different operational blockchains connected by interoperability/bridging solutions. For instance, Mosaic bridge [14] interconnects Ethereum, Polygon, Arbitrum, and Polkdot blockchains.

2. Cross-chain Interoperability

As described above, interoperability bridges facilitate users to swap or transfer assets from one blockchain to another as well as allow dApps to communicate arbitrary data and messages across blockchains. There can be several reasons behind transferring assets from one chain to another. For instance, the destination/host chain could have a low transaction fee, less block confirmation time (i.e., fast finality), valuable functionality/contract, high liquidity, capital utility, etc. For instance, Bitcoin can only handle about 5 tps and its average cost per transaction is approx. \$1.93 [15], while its energy expenditure per transaction is very high. On the other hand, the Solana blockchain can handle more than 2,000 tps and its average cost per transaction is less than a cent [16]. Similarly, there are only two dApps built on top of the BTC chain with a Total Value Lock (TVL) of less than \$200 million, while there are more than 500 dApps deployed on the Ethereum chain with a TVL of around \$124 Billion [17]. Hence there are more opportunities for putting BTC asset to work on Ethereum chain compared to BTC chain.

Although the number of cross-chain bridges has recently grown rapidly [12], in general, these bridges utilize a model called lock-mint-burn [18]. The basic operation principle of the lock-mint-burn model is as follows:

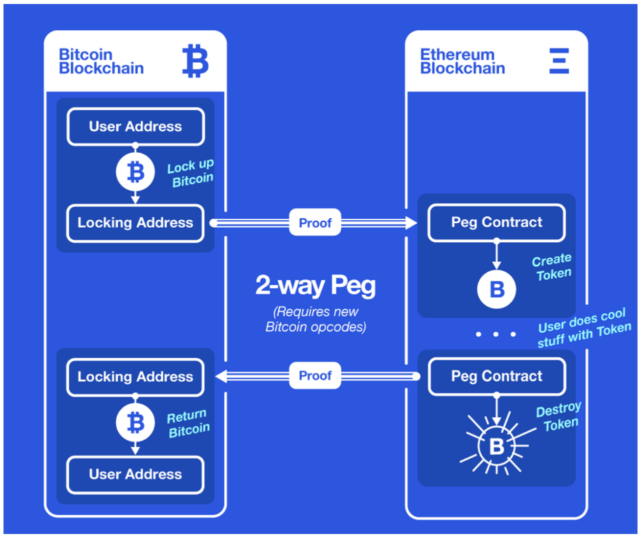

Figure 3. Assets such as BTC are locked on the base layer of the native chain (i.e., BTC blockchain) and unlocked on the host blockchain (i.e., Ethereum) in the form wrapped BTC [19]. Wrapped BTC can be utilized for different purposes such as collateral for borrowing another crypto asset (e.g., USDC, ETH, etc.), lending, trading, etc. on the host blockchain. Later on, the wrapped BTC can be burnt/destroyed on the host chain for claiming the original BTC back on the native chain.

- Users deposit assets into the bridge on native chain, and the bridge locks these assets in a smart contract.
- The bridge mints an equivalent amount of wrapped tokens (1:1 mapping ratio with the number of assets locked on native chain) on destination/host chain to the desired address.
- Later, when users need to withdraw assets back to native chain, users send back the wrapped token to the bridge.
- The wrapped tokens are burned on the destination/host chain while the bridge unlocks the assets on the native chain for users.

In essence, the lock-mint-burn model allows users to place their crypto assets into a digital vault on the native blockchain and creates wrapped versions on the host blockchain. Fig. 3 illustrates an example of the lock-mint-burn model, where the source chain is the Bitcoin network while the host chain is the Ethereum network. Users can transfer their Bitcoin tokens to the Ethereum ecosystem, exploring the different yield farming [20] opportunities there. Normally, to create liquidity and potential utility for the wrapped versions of the non-native assets in an ecosystem/blockchain (e.g., wrapped BTC tokens in Ethereum ecosystem), Liquidity Bootstrapping Pools (LBPs) are initiated on the destination chains. LBPs incentivize users to provide liquidity for the wrapped versions of the non-native assets pools, facilitating trading/swapping of the non-native assets with native assets of the ecosystem. Furthermore, a source blockchain could comprise numerous asset types, e.g., the different types of ERC20 tokens for Ethereum ecosystem [21]. However, each bridge solution targets a specific set of source chain assets to be bridged to the destination chain. For instance, a bridge between two blockchains would support the transfer of $\{s_1, s_2, s_3, s_4\}$ set of assets while another bridge between the same chains would facilitate the transfer of $\{s_2, s_5, s_6, s_7\}$ set of assets.

Next, despite design differences, the bridges can broadly be divided into two types, namely, centralized and decentralized. Centralized cross-chain bridges need users to trust third parties that play a broker role between chains by receiving assets from users in one chain and mint wrapped assets in another chain. For instance, wBTC is a wrapped version of BTC minted by a centralized custodian, namely BitGo [22]. On the other hand, decentralized cross-chain bridges do not require trusted third parties but a group of validators that manage the assets. The larger the number of validators, the more decentralized the bridge. In general, the user deposits assets from a chain into the pool, the validators verify the transaction, and the pool will mint wrapped tokens in another chain. Decentralized cross-chain bridges have a mechanism for authentication, consensus, event listening, etc, which is outside the scope of this work. Also, the transactions must be approved by a minimum number of validators to allow mint wrapped tokens to be on destination chain. More details about these aspects of decentralized cross-chain bridges can be found in [11].

Considering the fact that most available work on cross-chain interoperability focuses on the design and operation mechanisms of bridging solutions, there is a need for integrating all the available bridges solutions in the form of a network and utilizing the enhanced solution space for seamless value transfer across the blockchains, making possible the true vision of DeFi. This paper aims at filling this gap and considers decentralized cross-chain bridges connecting the blockchains. It is further considered that multiple bridges exist between any two blockchains, where each bridge supports a different set of assets transfer. Also, it is assumed that enough liquidity will be available for the transferred assets on the host chain, and the assets can be further moved to another chain, i.e., the host chain will serve as an intermediate between the source and the actual destination chains. More details on this matter are given in the next section.

3. Modeling Cross-chain Networks

Normally, interoperability/bridging solutions between any two chains support a different set of digital assets as well as charge differently based on their business models. Also, these bridges have different liquidity depths [23] and might be attractive for different types and sizes of users and dApps transactions. Therefore, all this information needs to be incorporated while modeling the cross-chain networks for transferring asset from a source chain to a destination chain.

The model begins with data collection and representation. Specifically, information regarding the blockchains of interest and the current network of bridges connected to them is sourced at a given cadence and a matrix is

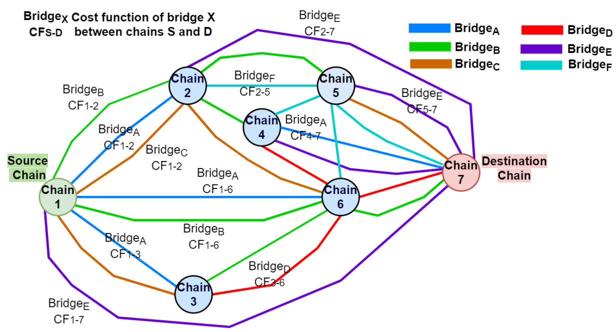

Figure 4. Graph representation of different blockchains connected via multiple bridges with different cost functions. Source and destination chains are highlighted. Nodes represent blockchains while edges represent bridges capable of transferring assets between given networks.

constructed containing information about their fee, speed, liquidity, etc. Based on this, a multi-connected bidirectional graph representation is built as shown in Fig. 4. The graph has nodes and edges, representing different L1/L2 blockchains and bridge solutions, respectively, whereas, the edge cost function values denote the cost (in terms of fee, speed, etc.) of traversing assets through these bridges.

Next, the goal is now to traverse this graph in a way that is most optimal to the user requirements. For instance, a dApp seeks to move the asset from chain 1 to chain 7 (Fig. 4) via a path optimizing an objective defined by the user, which could find the fastest path, or the cheapest path, or a combination of the two. Much like picking a flight via, e.g., the flight aggregator Google Flights. The dApp can utilize historical information as well such as what to expect in terms of speed and fees for a given path. There can be several options for bridges and paths to traverse for transferring funds from source chain to destination chain. Therefore, it is crucial to find the best bridges and path(s) for the transfer of funds from source to destination chain. In other words, enable users and dApps to transfer their assets between different chains at the best available fees and other possible conditions. In some situations, for instance, in a large-size transfer, it would be viable to split the transfer across multiple paths and bridges as there will not be enough liquidity available on a single path to support the transaction between blockchains. Note that in intra-blockchain networks, such splitting of large asset size is performed for asset swap across several Decentralized Exchanges (DEXes) by different dApps, including 1inch [24].

For chains that are not directly connected by bridges, the fund will traverse through intermediate chains before reaching the destination chain. It is very similar to buying an airline ticket. A customer can buy a direct flight, but if there is no direct connection between the cities, the customer will be offered a connecting flight and a specific airline. The same applies to asset transfer between chains. For instance, funds leave one network through a specific bridge, then "transfer" to another network, and through a specific bridge reach the final destination. The user only has to choose a preferable route in terms of cost, speed, effectiveness, or even security, or the dApp will choose one that is best for the users' requirements. Doing it manually will be nonviable, and hence requires an optimization mechanism to instantly provide the best solution based on users' demands.

4. Optimization Model for Cross-chain Pathfinding and Bridge Selection (PBS)

The aim of the cross-chain Pathfinding and Bridge Selection (PBS) problem considered in this paper is to derive the best bridge on each edge and consequently the path constituted by these edges (with best bridges) for transferring asset between source S and destination D blockchains. The problem of PBS is formalized as Integer Linear Programming problem (ILP) with model below:

Input parameters:

- $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{B})$: a directed graph $\mathcal{G}$ with a set of blockchains $\mathcal{V}$, edges $\mathcal{E}$, and bridges $\mathcal{B}$;
- $F_b^{(i,j)}$ Transfer fee for bridge $b \in \mathcal{B}$ operating on edge $(i, j) \in \mathcal{E}$;
- $T_b^{(i,j)}$ Transfer time for bridge $b \in \mathcal{B}$ operating on edge $(i, j) \in \mathcal{E}$;
- C: large constant;

Variables

- $\mathsf{y}_b^{(i,j)} \in \{0, 1\}$- equal to 1 if $b \in \mathcal{B}$ is used on edge $(i, j) \in \mathcal{E}$ that connects chain i with chain j for transferring fund A^{S-D} and 0 otherwise;
- $\mathsf{f}^{S-D} \in F$- denote the total fee spend on transferring fund A^{S-D};
- $\mathsf{t}^{S-D} \in F$- denote the total time spend on transferring fund A^{S-D};

Objective function

$$\text{Minimize}: \quad \alpha \cdot \mathsf{f}^{S-D} + \beta \cdot \mathsf{t}^{S-D} \tag{1}$$

Constraints

$$\mathsf{f}^{S-D} = \sum_{b \in \mathcal{B}} \sum_{(i,j) \in \mathcal{E}} F_b^{(i,j)} \tag{2}$$

$$\mathsf{t}^{S-D} = \sum_{b \in \mathcal{B}} \sum_{(i,j) \in \mathcal{E}} T_b^{(i,j)} \tag{3}$$

$$\sum_{i:(i,j) \in \mathcal{E}} \mathsf{y}_b^{(i,j)} - \sum_{i:(j,i) \in \mathcal{E}} \mathsf{y}_b^{(j,i)} = \begin{cases} 1, i = S \\ -1, i = D \\ 0, \text{otherwise} \end{cases} \tag{4}$$

$$\sum_{b \in \mathcal{B}} \mathsf{y}_b^{(i,j)} \leq 1 \ \forall (i, j) \in \mathcal{E} \tag{5}$$

$$F_b^{(i,j)} - \mathsf{C} \cdot \mathsf{y}_b^{(i,j)} \leq 0 \; \forall b \in \mathcal{B}, \forall (i,j) \in \mathcal{E} \tag{6}$$

$$T_b^{(i,j)} - \mathsf{C} \cdot \mathsf{y}_b^{(i,j)} \leq 0 \; \forall b \in \mathcal{B}, \forall (i,j) \in \mathcal{E} \tag{7}$$

The objective function represents a weighted sum where the choice of α and β allows to select in which degree each of the two terms in (1) should be minimized. The first term in (1) represents the total cost of transferring from source chain S to destination chain D as defined in (2), whereas, the second term in (1) corresponds to the total transfer time as computed in (3).

Constraint (4) is used to establish/compute a path between source chain S and D destination chain. The constraint enforces flow conservation and makes sure that the incoming traffic is equal to the outgoing traffic for the intermediate chains. Constraint (5) inflicts that at most one bridge is selected for each edge of the path between source chain S and destination chain D. Constraint (6) ensures the fee for only used bridges are counted in the total transfer fee for moving funds from source chain S to destination chain D. Finally, constraint (7) makes ensure the transfer time for only used bridges are counted in the total transfer time for moving fund from source chain S to destination chain D.

It is important to highlight that the proposed ILP model assumes that enough liquidity is available on all the bridges that are part of the $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{B})$. In other words, the bridges will have more liquidity than the transaction size. Similarly, the model selects the best path (in terms of input parameters) and is not applicable for situations that require splitting a transaction across multiple paths. However, the proposed model can be extended to consider splitting a transaction across multiple paths, which we leave for future work.

5. Heuristic Algorithm for the PBS Problem: PBS-H

For performance benchmarking of the proposed ILP model, in this section, we present a heuristic algorithm for the PBS problem. The algorithm denoted as PBS-H follows a 2-step approach whose pseudocode is summarised in Fig. 5.

The PBS-H first finds the best bridges for graph edges based on the criteria chosen by the user (i.e., fastest, cheapest, etc.) for a given asset to transfer. In other words, PBS-H first transforms the multi-connected directional graph to a uni-connected directional graph. The transformed graph will appear different for distinct criteria (or input parameters) selected by users. For instance, for the user A input parameters (or selected criteria), the algorithm will transform the multi-connected graph to the one shown in Fig. 6 where each edge represents the best bridge for the criteria opted by user A. On the other hand, for distinct criteria chosen by user B, the algorithm will result in the graph displayed in Fig. 7 with non-identical bridges for some edges compared to Fig. 6.

Algorithm 1 Pathfinding and Bridge Selection Heuristic (PBS-H) algorithm

Input parameters

$\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{B})$: a graph $\mathcal{G}$ with a set of blockchains $\mathcal{V}$, edges $\mathcal{E}$, and bridges $\mathcal{B}$;

F_{S-D}: Amount of fund to be transferred from Chain S to Chain D;

U: Set of input parameters values provided by the user;

Main steps

Step.1-1: Find the best bridge b^e for each $e \in \mathcal{E}$ based on the input parameter values U;

Step.1-2: Set the cost function cf^e of each $e \in \mathcal{E}$ to the cost function cf^b of the bridge selected in Step.1-1;

Step.2-1: Compute shortest path from chain S to D based on cf^e of each $e \in \mathcal{E}$;

Step.2-2: Transfer F_{S-D} over bridges constituting the path selected in Step.2-1;

Figure 5. Pseudocode of the PBS-H algorithm for cross-chain transfer of asset.

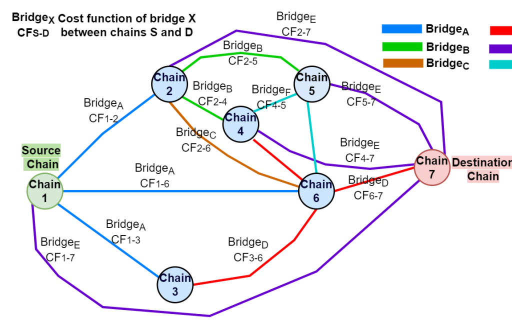

Figure 6. Uni-connected directional graph representation of different blockchains connected via the best bridges that are computed during step.1 of PBS-H based on the specific requirements of user A.

Once the graph is transformed to a uni-connected directional network, the next step is to find the best path based on the chosen criteria (e.g., fastest, cheapest, or a weighted sum of these parameters) by the users. For this purpose, PBS-H computes the shortest path (in terms of weighted sum of transfer fee and speed) from source to destination chain using the transformed graph. The PBS-H chooses the path and updates the information about the available liquidity on each edge of the chosen path accordingly.

6. Numerical Results

In this section, the performance of the proposed optimization model is evaluated and compared to the heuristic algorithm for the PBS problem. For this purpose, a discrete event-driven simulator was built in Python, while the ILP formulation is solved by running a commercially available ILP solver, i.e. CPLEX [25]. Similarly, for the *shortest pathfinding*, the PBS-H algorithm utilized *Dijkstra's* algorithm. The value of input parameters used in the simulations is given in Table I. For the 12-node and 20-node cross-chain networks, topologies were randomly generated, while the

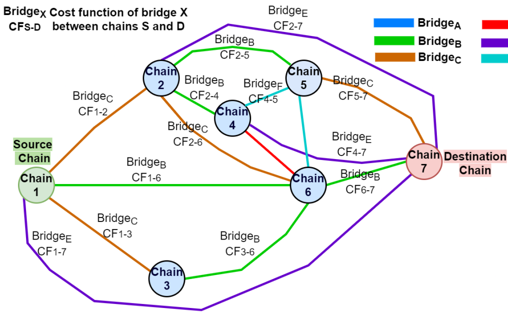

Figure 7. Uni-connected directional graph representation of different blockchains connected via the best bridges that are computed during step.1 of PBS-H based on the specific requirements of user B.

TABLE 1. PARAMETERS AND THEIR VALUES

Parameter	Value
Min (max) number of bridges	3 (7)
Number of blockchains	12 and 20
Min (max) number of edges	3 (6)
Max (min) amount of fee per bridge	2 (8)
Max (min) amount of transaction time per bridge	2 (8)

source and destination nodes/chains for asset transfer were chosen using uniform random distributions. All the results were collected until a confidence interval of 4% or less was reached with a confidence level of 95%.

Fig. 8 and Fig. 9 present results obtained for 12-node cross-chain topologies using CPLEX solver and heuristic algorithm. Note that PBS-ILP can be transformed to `Mini_fee`, `Trade_off`, `Mini_speed` by setting the value of (α, β) to (1,0), (1,1), and (0,1), respectively. Fig. 8 shows the average transaction fee of PBS-H and PBS-ILP for all three optimization strategies. Several insight can be gained from the results displayed in Fig. 8. First, the performance of PBS-ILP is significantly better than PBS-H for all the cases, i.e., it can achieve an average reduction of 24% in transaction fee w.r.t PBS-H. This is because PBS-ILP concurrently solves the bridges selection and pathfinding using a larger solution space compared to PBS-H that employs a 2-step approach of finding bridges and path separately. Second, the strategy that aimed at minimizing the transaction fee (i.e., `Mini_fee`) shows a substantial decrease in fee compared to the one that aimed at minimizing the transaction speed (`Mini_speed`). Similarly, Fig. 9 illustrates the average transaction speed of PBS-H and PBS-ILP for the optimization strategies. In general, the results confirm the same trend observed in Fig. 9. Furthermore, by comparing these results with the ones in Fig. 8, it can be seen that `Trade_off` strategy (i.e., α and β both set to 1) demonstrates a beneficial tradeoff between fee and speed.

Similarly, Fig. 10 and Fig. 11 display results obtained for 20-node cross-chain topologies. Since the topologies are 66% larger than 12-node topologies, both the values of average transaction fee and speed raised around 10 to 12% w.r.t the results for 12-node topologies. Moreover, the performance of PBS-ILP strategies further improved by

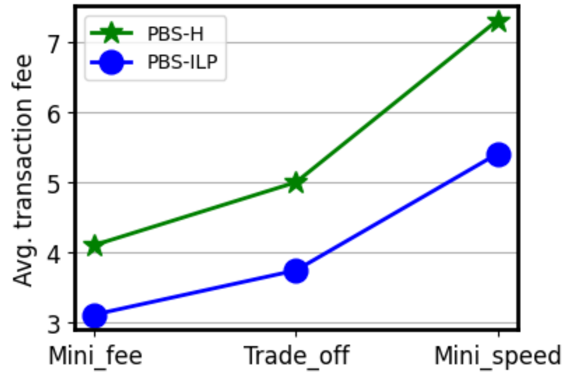

Figure 8. Average transaction fee vs. optimization strategies for the 12-node cross-chain networks.

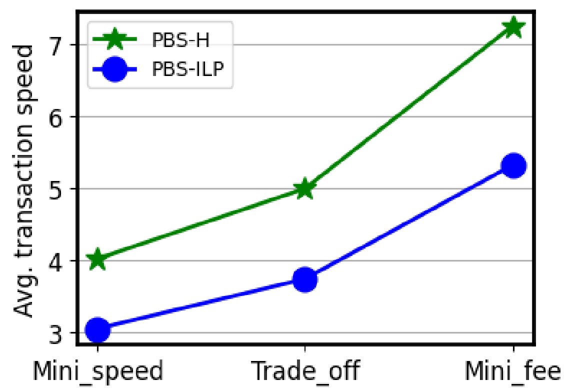

Figure 9. Average transaction speed vs. optimization strategies for the 12-node cross-chain networks.

around 5 to 7% w.r.t PBS-H strategies due to relatively large solution space. However, overall, the pattern of the results is the same as that for the 12-node cross-chain topologies.

Finally, to assess the feasibility of the proposed model in terms of network scalability, we measured the computation time of these optimization strategies for PBS-ILP. For the 12-node topologies, the average computation time for `Trade_off`, `Mini_fee`, and `Mini_speed` is 2.975 μs, 2.156 μs, and 2.252 μs, respectively. On the other hand, for 20-node topologies, the average computation time becomes 4.253μs, 3.031μs, 3.381μs, respectively. The computation time for the PBS-H strategies was also measured, however, they were drastically smaller than the values reported for the PBS-ILP strategies.

7. Conclusion

The paper proposes and studies the pathfinding and bridge selection problem for cross-chain networks, which is crucial to solve for seamless value transfer in a multi-chain future for DeFi. The problem is first modeled as a multi-connected bidirectional graph and then formulated as an ILP. This ILP formulation provides a framework that can be flexibly adapted to optimize the transfer of value across

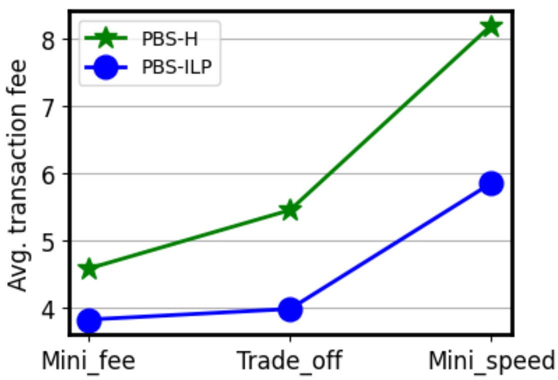

Figure 10. Average transaction fee vs. optimization strategies for the 20-node cross-chain networks.

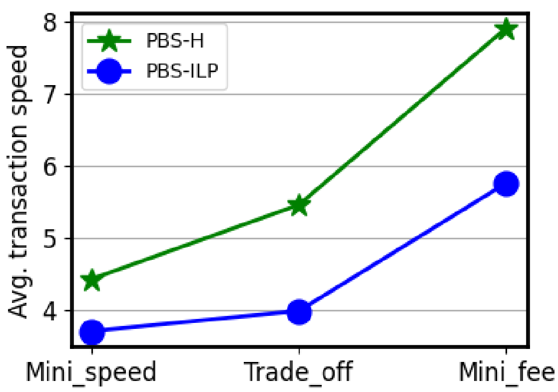

Figure 11. Average transaction speed vs. optimization strategies for the 20-node cross-chain networks.

blockchains in terms of fee or speed. The ILP formulation is solved for different objective functions and sizes of cross-chain networks, and the results are compared with a 2-step heuristic algorithm that computes the bridge and pathfinding separately.

Numerical results indicate the benefits (in terms of minimizing the transfer fee and time) of the proposed ILP model that jointly considers pathfinding and bridge selection. In future work, we will extend the model and algorithms to multi-pathfinding, where a transfer order can be split across multiple paths (between source and destination chains) in case enough liquidity is unavailable on a single path. Similarly, multiple bridges (connecting two blockchains) can be considered if it is not possible to transfer value via a single bridge due to a lack of required liquidity.

References

[1] S. Nakamoto, “A peer-to-peer electronic cash system,” Oct 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] An overview of the Layer 1 Blockchain Ecosystem. (2022, March 21). [Online]. Available: https://blockchain-comparison.com/blockchain-protocols/

[3] A. Hafid, A. S. Hafid, and M. Samih, “Scaling blockchains: A comprehensive survey,” *IEEE Access*, vol. 8, 2020.

[4] C. Sguanci, R. Spatafora, and A. M. Vergani, “Layer 2 blockchain scaling: a survey,” Jun 2021. [Online]. Available: https://arxiv.org/pdf/2107.10881.pdf

[5] J. Poon and T. Dryja, “The bitcoin lightning network: Scalable off-chain instant payments,” Jan 2016. [Online]. Available: https://lightning.network/lightning-network-paper.pdf

[6] “Polygon lightpaper: Ethereum’s internet of blockchains,” Feb 2021. [Online]. Available: https://polygon.technology/lightpaper-polygon.pdf

[7] K. Nelaturu, “Plasma — layer 2 scaling protocol,” Medium blog, Jul 2018. [Online]. Available: https://medium.com/coinmonks/plasma-layer-2-scaling-protocol-5a1263d86bc8

[8] H. A. Kalodner, S. Goldfeder, X. Chen, S. M. Weinberg, and E. W. Felten, “Arbitrum: Scalable, private smart contracts,” in *USENIX Security Symposium*, 2018.

[9] E. Semeniak, “Decentralized finance (defi) solutions: Benefits, challenges, and best practices to build one,” Apriorit, Oct 2021. [Online]. Available: https://www.apriorit.com/dev-blog/749-blockchain-decentralized-finance-defi-solutions

[10] Protocol Categories. (2022, March 29). [Online]. Available: https://defillama.com/categories

[11] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, “A survey on blockchain interoperability: Past, present, and future trends,” *ACM Comput. Surv.*, vol. 54, no. 8, oct 2021.

[12] D. Berenzon, “Blockchain bridges: Building networks of cryptonetworks,” Medium blog, Sep 2021. [Online]. Available: https://medium.com/1kxnetwork/blockchain-bridges-5db6afac44f8

[13] A. Gauba, “Finality in blockchain consensus,” Medium blog, Aug 2018. [Online]. Available: https://medium.com/mechanism-labs/finality-in-blockchain-consensus-d1f83c120a9a

[14] Mosaic Bridge. (2022, March 29). [Online]. Available: https://mosaic.composable.finance/

[15] Bitcoin Average Transaction Fee. (2022, March 29). [Online]. Available: https://ycharts.com/indicators/bitcoin_average_transaction_fee

[16] Solana Blockchain. (2022, March 29). [Online]. Available: https://solana.com/

[17] Total Value Locked All Chains. (2022, March 31). [Online]. Available: https://defillama.com/chains

[18] Mochilab, “Cross-chain - the next trend of blockchain,” Medium blog, Feb 2022. [Online]. Available: https://medium.com/mochilab/cross-chain-the-next-trend-of-blockchain-f2a08d13c217

[19] ConsenSys, “Taking stock: Bitcoin and ethereum,” Medium blog, Dec 2015. [Online]. Available: https://medium.com/@ConsenSys/taking-stock-bitcoin-and-ethereum-4382f0a2f17

[20] “What is yield farming in decentralized finance (defi)?” Binance Academy, Aug 2020. [Online]. Available: https://academy.binance.com/en/articles/what-is-yield-farming-in-decentralized-finance-defi

[21] T. Falk, “What is an erc20 token?” Finder, Oct 2021. [Online]. Available: https://www.finder.com/erc20-tokens

[22] WBTC. (2022, March 09). [Online]. Available: https://www.bitgo.com/

[23] Cross Chain TVL Rankings. (2022, March 09). [Online]. Available: https://defillama.com/protocols/Cross%20Chain

[24] S. Kunz, “What are dex aggregators? a deep dive by 1inch,” coinmarketcap, 2021. [Online]. Available: https://coinmarketcap.com/alexandria/article/what-are-dex-aggregators-a-deep-dive-by-1inch

[25] ILOG CPLEX. (2022, March 10). [Online]. Available: https://www.ibm.com/analytics/cplex-optimizer