Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence

Canadian AI 2026
25-29 May 2026,Vancouver,BC

Execution-Aware A* Search for Cross-Exchange Stablecoin Arbitrage

Kevin Litvin
Simon Fraser University

Abstract

Cross-exchange cryptocurrency arbitrage enables low-risk profit from price discrepancies across exchanges, yet existing approaches employ negative cycle detection that targets opportunity identification rather than execution feasibility. We introduce an execution-aware pathfinding framework using A* search with domain-specific guidance heuristics, applied to stablecoins, a unique asset class exceeding $300 billion in market capitalization that bridges cryptocurrency and fiat currency, offering a novel dataset for arbitrage research. The problem is modelled as a weighted directed graph where nodes represent (exchange, stablecoin) pairs across 12 centralized exchanges and edges encode real-world costs including fees, slippage, gas, transfer delays, and exchange reliability. Three guidance heuristics and a parallelized multi-start baseline are evaluated over 7,200 search instances. Our slippage-aware heuristic h_2 reduces node expansions by 29% relative to Dijkstra while matching its profit, demonstrating that domain-specific heuristics can meaningfully improve execution feasibility in real-time arbitrage planning. Code: https://github.com/kevinl03/Stablecoin-CrossExchange-Arbitrage.

Keywords: stablecoin arbitrage, A* search, cross-exchange trading, execution-aware pathfinding, cryptocurrency, graph algorithms

1. Introduction

We formulate cross-exchange stablecoin arbitrage as a graph-planning problem and develop a route planner that computes executable trade-and-transfer paths using A* search with domain-specific heuristics and live market data via CCXT[1]. Our methodology addresses four research questions: (1) Do certain heuristics consistently outperform others on specific trading-instance classes? (2) How does each heuristic influence A*'s node-expansion behaviour? (3) How does performance change as structural difficulty increases? (4) When heuristics return suboptimal routes, how close are they to the best-known solution?

Stablecoins, with a combined market capitalization exceeding $300 billion[2], serve as the primary liquidity bridges in cryptocurrency markets. Because centralized exchanges operate independently, price discrepancies frequently arise from fragmented liquidity, withdrawal delays, and blockchain congestion. Unlike volatile cryptocurrencies, stablecoins operate within a bounded price band anchored to fiat currency, reframing arbitrage into a constrained execution-optimization problem. The growing institutional relevance of digital assets, underscored by the U.S. Strategic Bitcoin Reserve in 2025[3], further motivates research into this infrastructure.

The four key challenges addressed are: **Liquidity -** order-book depth limits deployable capital, **Slippage -** large orders incur nonlinear price impact, **Latency -** blockchain confirmations may exceed the arbitrage window, and **Reliability -** exchanges may suspend withdrawals or experience API instability. These constraints interact, and our heuristic-guided A* framework models these trade-offs to prioritize realistically executable routes.

2. Related Work

Most prior work models arbitrage as negative-weight cycle detection via Bellman–Ford[4–10], focusing on opportunity identification rather than execution feasibility. Cross-exchange settings introduce blockchain latency, withdrawal fees, and liquidity fragmentation[11, 12], while risk-aware pathfinding[13] and slippage models[14] remain disconnected from executable planning. Our approach integrates these concerns into A* search with domain-specific heuristics.

3. Problem Formulation

3.1. Graph Representation

We formulate the problem as a directed weighted graph $G = (V, E)$ where each node $v \in V$ represents a unique (exchange, stablecoin) pair and each directed edge $e \in E$ corresponds to a trade or a cross-exchange transfer. Every edge has an associated effective exchange rate incorporating all relevant costs, and we assign weight $w(e) = -\log(\mathit{effective\_rate}(e))$, converting multiplicative returns into additive costs for shortest-path algorithms.

3.2. Path Profitability

Given a path $P = (v_0, v_1, \ldots, v_k)$ with initial capital C_0, the final capital is $C_k = C_0 \cdot \prod_{e \in P} r(e)$ where $r(e)$ denotes the effective rate on edge e. A path is profitable when $C_k - C_0 \geq \mathit{min\_profit}$. In additive log space this becomes:

$$\sum_{e \in P} -\log\bigl(r(e)\bigr) \;\leq\; -\log\left(1 + \frac{\mathit{min\_profit}}{C_0}\right).$$

The goal set is $\mathcal{G} = \left\{ v_k \in V \;\middle|\; C_k > C_0 \right\}$. We do *not* require $v_k = v_0$: this **open-path** formulation allows the trader to end on any pair where USD value exceeds initial capital, justified because all assets are USD-pegged stablecoins ($\pm 2\%$ price tolerance).

3.3. A* Search Formulation

We apply A* search with cost-to-reach $g(n) = \sum_{e \in \text{path}} -\log(r(e))$ and evaluation function $f(n) = g(n) + h(n)$, where $h(n)$ estimates the execution risk of the remaining path given current blockchain state, including factors such as liquidity depth, order-book slippage, chain congestion, and exchange reliability. The algorithm terminates when $C_n > C_0$.

For h_3, we employ **Weighted A***[15] with $f(n) = g(n) + w(n) \cdot h(n)$, where $w(n) = w_{\min} + (w_{\max} - w_{\min}) \cdot \rho(n)$ and $\rho(n) \in [0, 1]$ is a chain kickback risk score. When $w(n) = 1$ for all nodes, this reduces to standard A*. The algorithm is not complete: market data does not guarantee a profitable path from every starting node. Search terminates when a profitable path is found, the frontier is exhausted, or depth/time limits are reached.

4. Novel Heuristics for Arbitrage Search

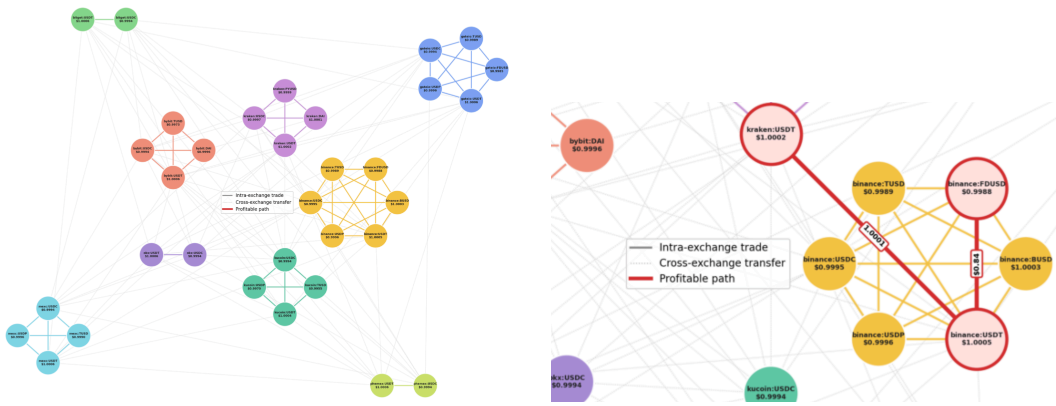
We design three guidance heuristics (h_1, h_2, h_3) to steer A* toward executable arbitrage routes. Each captures a different dimension of execution feasibility via an additive penalty $h(n) \geq 0$ in $f(n) = g(n) + h(n)$. These are *guidance penalties*, not admissible lower bounds. We also evaluate a parallelized multi-start baseline that wraps any heuristic.

4.1. h_1 – Liquidity Heuristic

The liquidity heuristic penalizes low-volume markets that cannot support the required trade size. We compute a liquidity ratio from 24-hour quote volume, the remaining arbitrage window T_{remain}, and order size:

$$\mathit{liquidity\_ratio} = \frac{(\mathit{24h\_vol}/86400) \cdot T_{remain}}{\mathit{order\_size\_usd}}, \quad h_1(n) = \lambda_{liq}\left(1 - \mathit{clamp}\left(\tfrac{\log_{10}(\mathit{liquidity\_ratio})+1}{2}, 0, 1\right)\right).$$

This heuristic favours deep, actively traded markets and is time-aware.



(a) Subgraph of the arbitrage network (9 of 12 exchanges shown).

(b) A discovered profitable path (zoomed).

Figure 1. (a) A subgraph of the stablecoin arbitrage graph across 9 of 12 exchanges: nodes are (exchange, coin) pairs with USD prices; solid edges are intra-exchange trades, light edges are cross-exchange transfers. (b) A profitable path: kraken:USDT $\rightarrow$ kucoin:USDT $\rightarrow$ kucoin:TUSD. The cross-exchange transfer has $r_1 = 1.0001$ and the intra-exchange trade $r_2 = 1.0018$, yielding $\approx 1.0019\times$ return (\$19 on \$10,000). Profitable paths need not form closed cycles. Note: the \$0.84 label visible on binance:FDUSD reflects a fractional stablecoin partially backed by non-USD assets, not a true exchange rate.

4.2. h_2 – Slippage Heuristic

The slippage heuristic models execution price impact using real-time order-book depth. Given mid-price P_{mid} and the VWAP required to fill order size Q:

$$h_2(n) = \lambda_{slip} \cdot \max\{0,\ slippage_{bps} - \theta\}, \quad slippage_{bps} = \frac{VWAP(Q) - P_{mid}}{P_{mid}} \times 10^4,$$

where θ is a configurable tolerance. Edge costs encode static taker and withdrawal fees; h_2 captures *dynamic* order-book slippage, so there is no double-counting with $g(n)$.

4.3. h_3 – Chain Congestion and Exchange Reliability

h_3 penalizes transfer latency and operational risk using pre-computed chain transfer times and static reliability scores $s(n) \in [0, 1]$:

$$h_3(n) = \lambda_{chain}\left(\frac{t_{\min}}{T_{\text{remain}}}\right) + \lambda_{exchange}(1 - s(n))\,.$$

4.4. Parallelized Multi-Start Baseline

The parallelized baseline runs k independent A* searches in parallel from randomly sampled starting nodes, each using a base heuristic, and returns the most profitable result.

5. Experiments and Evaluation

5.1. Setup

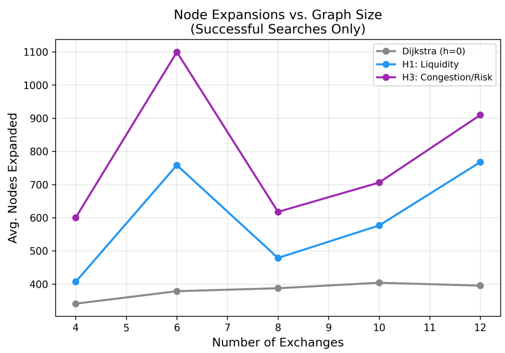
We gather live market data from twelve major exchanges via CCXT[1], constructing graphs of up to 41 nodes and 864 edges. We evaluate via cached-graph heuristic comparison, graph scaling (4–12 exchanges), an 8-hour overnight campaign, and quote staleness.

Table 1. Heuristic comparison at $10k (30 starts).

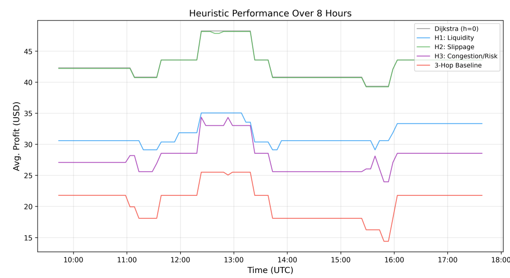
Heuristic	**Succ.**	**Profit**	**Exp.**	**Δ**
Dijkstra	56.7%	$10.01	359	—
h_1 (liquidity)	56.7%	$6.72	568	+58%
h_2 (slippage)	56.7%	$9.91	256	−29%
h_3 (latency)	56.7%	$7.06	673	+87%

Table 2. Node expansions by graph size (10 trials).

Ex.	$\|V\|$	$\|E\|$	**Dijk.**	$\mathbf{h_1}$	$\mathbf{h_3}$
4	19	161	341	407	600
6	28	368	378	758	1100
8	35	619	387	479	618
10	39	760	404	577	706
12	41	864	395	768	910



(a) Node expansions vs. graph size.



(b) Overnight comparison (1,440 runs each).

Figure 2. (a) Dijkstra expansions stay flat as graphs grow; penalty heuristics scale worse. (b) Profit, success, and expansions across the 8-hour campaign.

5.2. Results

Tables 1 and 2 summarize heuristic performance. On the cached graph (30 start nodes), all A*-based methods achieve identical 56.7% success; Bellman–Ford, 1-hop, and 2-hop baselines fail entirely. h_2 reduces expansions by **29%** relative to Dijkstra while matching its profit within 1%, suggesting slippage-aware guidance follows the same optimal routes with less wasted exploration. h_1 and h_3 expand 58–87% *more* nodes and find 30–33% less profit, indicating their risk penalties overcompensate and divert search from structurally optimal paths. The parallelized multi-start baseline (k=3 parallel starts from random nodes) achieves comparable success rates but does not reduce per-search node expansions; its benefit is robustness to unlucky starting positions.

Over an 8-hour overnight campaign (**7,200 instances**), every A*-based run finds a profitable path. h_2 matches Dijkstra-level profit across all order sizes while consistently expanding fewer nodes (Figure 2). Re-evaluating paths after 5–300 s delays, 99.6% remain profitable up to 120 s.

6. Conclusions

We presented an execution-aware A* framework for cross-exchange stablecoin arbitrage, evaluated on a unique centralized exchange (CEX) stablecoin dataset spanning 12 exchanges. Our novel heuristic, h_2 (slippage), reduces expansions by 29% while matching Dijkstra's profit; h_1 and h_3 overcompensate risk, expanding more nodes for less profit. Future work includes asynchronous order-book pre-fetching and extension to DEX/AMM markets.

Acknowledgments

We thank Hang Ma for guidance on an earlier draft of this paper and Tania Pocrnjic for continuous feedback and review.

References

[1] CCXT. *CryptoCurrency eXchange Trading Library*. Accessed: 2025. 2024. URL: https://github.com/ccxt/ccxt.

[2] CoinMarketCap. *Top Stablecoin Tokens by Market Capitalization*. Accessed: 2025. 2025. URL: https://coinmarketcap.com/view/stablecoin/.

[3] The White House. *Establishment of the Strategic Bitcoin Reserve and United States Digital Asset Stockpile*. Accessed: 2025. 2025. URL: https://www.whitehouse.gov/presidential-actions/2025/03/establishment-of-the-strategic-bitcoin-reserve-and-united-states-digital-asset-stockpile/.

[4] R. Bellman. "On a routing problem". In: *Quarterly of Applied Mathematics* 16.1 (1958), pp. 87–90. DOI: 10.1090/qam/102435. URL: https://www.ams.org/journals/qam/1958-16-01/S0033-569X-1958-0102435-2/.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. 3rd. MIT Press, 2009.

[6] R.-C. Oanţă and A. Coroiu. "Crypto Advisor: A Web Application for Spotting Cross-Exchange Cryptocurrency Arbitrage Opportunities". In: *Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023), Volume 1*. INSTICC. SciTePress, 2023, pp. 238–246. ISBN: 978-989-758-641-5. DOI: 10.5220/0011850400003470. URL: https://www.scitepress.org/Link.aspx?doi=10.5220%2F0011850400003470.

[7] L. Zhou, K. Qin, A. Cully, B. Livshits, and A. Gervais. "On the Just-In-Time Discovery of Profit-Generating Transactions in DeFi". In: *2021 IEEE Symposium on Security and Privacy (SP)*. 2021. URL: https://www.ieee-security.org/TC/SP2021/downloads/poster/poster27.pdf.

[8] Y. Zhang, T. Yan, J. Lin, B. Kraner, and C. Tessone. "An Improved Algorithm to Identify More Arbitrage Opportunities on Decentralized Exchanges". In: *arXiv preprint arXiv:2406.16573* (2024). URL: https://arxiv.org/abs/2406.16573.

[9] B. Luo, J. Jiang, Y. Chen, J. Hou, C. J. Tey, Z. Qiu, B. He, S. Xiao, D. Ong, and W. H. Ang. "RICH: Real-Time Identification of Negative Cycles for High-Efficiency Arbitrage". In: *Proceedings of the VLDB Endowment* 18.11 (2025), pp. 4081–4094.

[10] F. Fang, C. Ventre, M. Basios, L. Kanthan, D. Martinez-Rego, F. Wu, and L. Li. "Cryptocurrency trading: a comprehensive survey". In: *Financial Innovation* 8.1 (2022), pp. 1–59. DOI: 10.1007/978-981-96-6839-7_4. URL: https://doi.org/10.1007/978-981-96-6839-7_4.

[11] A. Obadia, A. Salles, L. Sankar, T. Chitra, V. Chellani, and P. Daian. "Unity is Strength: A Formalization of Cross-Domain Maximal Extractable Value". In: *arXiv preprint arXiv:2112.01472* (2021). URL: https://arxiv.org/abs/2112.01472.

[12] Y. Hafsi and E. Vittori. "Optimal Execution with Reinforcement Learning". In: *arXiv preprint arXiv:2411.06389* (2024). URL: https://arxiv.org/abs/2411.06389.

[13] E. Nikolova and D. R. Karger. "Route Planning under Uncertainty: The Canadian Traveller Problem Revisited". In: *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI)*. 2008, pp. 969–974.

[14] R. Almgren and N. Chriss. "Optimal execution of portfolio transactions". In: *Journal of Risk* 3.2 (2001), pp. 5–39. DOI: 10.21314/JOR.2001.041. URL: https://doi.org/10.21314/JOR.2001.041.

[15] I. Pohl. "Heuristic search viewed as path finding in a graph". In: *Artificial Intelligence* 1.3-4 (1970), pp. 193–204.

Statement on the Use of AI

Claude (Anthropic) was used to assist with LaTeX formatting of author-written text. Technical content, experimental design, and analysis are the work of the author.