On Fair Designs of Cross-Chain Exchange for Cryptocurrencies via Monte Carlo Simulation

Zini Wang[1], Guangxin Jiang*[2], and Qiang Ye[2]

[1]School of Management, Shanghai University, Shanghai 200444, China
[2]School of Management, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China

March 12, 2021

Abstract

Cryptocurrency is one of the earliest and the most successful applications of blockchain, and it utilizes the distributed ledger, which is a commonly used technique in blockchain, to make a decentralized transaction within the blockchain of a cryptocurrency. However, how to make a decentralized transaction of cryptocurrencies between parties on different blockchains, i.e., the cross-chain exchange, is not well-studied. In this paper, we develop a new method to make cross-chain exchanges based on the classical atomic swap. We first study the optionality embedded into the atomic swap and propose to add a premium into the atomic swap, and then design a new procedure with the premium to guarantee the fairness of the cross-chain exchange. We also provide an algorithm based on the least-squares Monte Carlo method to estimate the premium and analyze the convergence of the algorithm. Moreover, we study the cross-chain exchange with margin trading. We propose an adapted exchange procedure to make a fair cross-chain exchange and an algorithm to estimate the fair premium under the margin trading. Numerical experiments are provided to show the effectiveness of the algorithms.

Keywords: Monte Carlo Simulation; Cryptocurrency and Blockchain; American Option; Atomic Swap; Decentralized Exchange

1 Introduction

In the past decade, blockchain technology has been developed very fast and received a significant amount of attention from both academia and industry. It has been widely studied and used in the areas of finance (Iansiti and Lakhani 2017), supply chain (Kshetri 2018), healthcare (Angraal et al. 2017), and energy (Noor et al. 2018), to name a few. Cryptocurrency or cryptographic currency

*Corresponding author: gxjiang@hit.edu.cn

is one of the earliest and the most successful applications of blockchain. The bitcoin, which is the first cryptocurrency using blockchain, is well-traded around the world. Besides bitcoin, some other cryptocurrencies like ether and litecoin are created to better support real-world applications by using some new blockchain techniques.

The key technology of the cryptocurrency is an open, distributed ledger that can effectively, verifiably and permanently record transactions between two parties, so transactions in a cryptocurrency world do not need trusted third-party intermediaries like banks. Take the bitcoin as an example. The bitcoin blockchain is a public ledger that all nodes on the network have a copy of the ledger. Once a transaction between two bitcoin wallets in the blockchain is committed, it is broadcast to all nodes to be verified. After certain amount of time, all the accepted transactions are packaged in a block, and this block is added to the blockchain. Then the new blockchain is quickly published to all nodes without requiring central oversight. Such a way of transactions without third-party intermediaries is called *decentralized* exchange.

In this paper, we consider how to make a fair decentralized exchange between parties on different blockchains, which is known as *cross-chain exchange*. When the two parties belong to different blockchains, they have no access to each other's ledgers and cannot automatically verify whether an asset is in fact owned by a party and can be transferred. So we need to use some blockchain techniques, e.g., smart contracts, to make a safe and fair cross-chain exchange. In this paper, we consider a new smart contract called *atomic swap* in the decentralized cross-chain exchange. The atomic swap was originated in an online forum (see Nolan 2018, Herlihy 2018), and first applied in transactions between decred and litecoin, which are two different cryptocurrencies. The atomic swap uses hashed time lock contract (HTLC), which is a form of cryptographic escrow, to guarantee the success of the transactions.

The atomic swap provides a doable way to conduct the cross-chain exchange, but there are still some technical details to be addressed. As the party who initiates the transaction in the atomic swap has the right to choose a transaction time in his favor before time lock, there is optionality embedded in the atomic swap. Moreover, the volatilities of cryptocurrencies are generally high, so the optionality cannot be ignored. The initiator may have arbitrage opportunities and benefit from the current procedure of atomic swap. To make the exchange fair, the initiator should pay a certain amount of premium to the counterparty. In this paper, we propose a fair procedure to make

the cross-chain exchange. The most closely related work is Han et al. (2019), which considered a simple cryptocurrency model and a binomial tree method to determine the premium. However, in practice, the price of a cryptocurrency fluctuates heavily, so we need more sophisticated models to describe the evolution of the price of the cryptocurrency. To allow for general pricing models, we use the least-squares Monte Carlo method (Longstaff and Schwartz 2001, Stentoft 2004, Zanger 2013) to estimate the fair premium. Moreover, the new procedure can ensure that the payment of the premium and the exchange contract take effect simultaneously, whereas the procedure in Han et al. (2019) cannot. In summary, we believe our work offers the following contributions:

(1) We formulate the problem of fair cross-chain exchanges as an option pricing problem and propose to add a premium into atomic swap. Based on that, we design a new procedure for the cross-chain exchange with premium by constructing a series of smart contracts. This new procedure can ensure the fairness of the cross-chain exchange, i.e., no one can cheat each other by finding the loophole of the procedure.

(2) Under sophisticated pricing models of cryptocurrencies, we provide an efficient and general-purpose algorithm to estimate the fair premium. The premium is used in the exchange procedure to ensure the fairness of the cross-chain exchange, i.e., there is no arbitrage opportunity in the exchange. We also analyze the convergence of the algorithm.

(3) We introduce margin trading to the cross-chain exchange, i.e., both sides of the exchange deposit a fraction of their funds in the contract and deposit full principal when the exchange is happening. Under this setting, we design a new and fair exchange procedure, which is much more complicated than that without margin, and provide the algorithm to estimate the premium.

Literature Review

Our work is related to two lines of literature. The first one is the cross-chain technique. There are four major cross-chain techniques: notary schemes, sidechains and relays, distributed private key control, and HTLC. The most representative scheme of notary schemes is the interledger protocol (Thomas and Schwartz 2015) that aims to realize the coordination and communication between

different blockchain ledgers. Sidechains and relays technique is referred to Polkadot (2020) and Cosmos (2020). They are new blockchains based on anchoring tokens on the original chain, and have an underlying architecture (Cosmos software development kit and substrate) for developers to build blockchains that can be directly inserted into the corresponding ecosystem. Distributed private key control entrusts the decentralized network to hold the user's private key. The most typical example refers to Wanchain (2020), which has traded bitcoin, ethereum, and other cryptocurrencies across chains. HTLC (Bitcoinwiki 2019b) stipulates that both chains receive the unlock information within a specified time and then transfer assets. The specific mechanism for applying HTLC is atomic swap, see Bitcoinwiki (2019a), Liu (2018), and Han et al. (2019). The advantage of the atomic swap is that it can avoid notary public and require little information about the parties of the transaction. Compared with other techniques, the atomic swap is the most mature, simple, and clear, with the greatest possibility of wide implementation, so we consider it in this paper.

The second related line of literature is about option pricing, especially, the American exchange option pricing. Margrabe (1976) provides a closed-form formula for the price of the European exchange option, and Carr (1988) studies the American exchange option, and proposed a closed-form approximation formula to price it. Armada et al. (2007) propose an improved method based on the model in Carr (1988), and it produces more accurate output. Andrikopoulos (2010) adopts a quadratic approach to price the American exchange option, which provides an alternative function of the option price. Recently, Cheang and Chiarella (2011) have proposed a method to price exchange options under the assumption that the underlying asset price is governed by a jump diffusion process, and Kim and Koo (2016) have considered the credit risk in pricing exchange options.

For options driven by more complicated underlying asset models, Monte Carlo simulation is a commonly used pricing tool. For example, Carr and Madan (2008) propose a framework for pricing options driven by CGMY process and Meixner process models, and Fu (2007) applies Monte Carlo simulation in variance gamma model; see Glasserman (2013) and Schoutens (2003) for comprehensive overviews. Recently, Jiang et al. (2020) and Hong and Jiang (2019) propose a framework called offline simulation online application that can build the formula of option price in the offline simulation stage and apply the formula in real-time price quoting and risk hedging. In

the above literatures, the price of the underlying asset is usually modeled as a continuous stochastic process model. Whereas there are some works that use econometrics methods to model the price of cryptocurrencies. For example, Chen et al. (2017) use ARIMA (autoregressive linear moving average) and GARCH (generalized autoregressive conditional heteroscedasticity) model to conduct an econometric analysis of bitcoin price. Katsiampa (2017) proposes to use the AR-CGARCH (Autoregressive Component GARCH) model to explore the heteroskedasticity of bitcoin prices. To study the prices of correlated cryptocurrencies, we may use a multivariate GARCH method (Ruppert and Matteson 2015).

The paper is organized as follows. In Section 2, we provide a preliminary introduction to the classical atomic swap. Then we analyze the optionality embedded in the classical atomic swap and propose a new decentralized procedure with the premium in Section 3. We propose a method to estimate the fair premium and provide the theoretical properties of the premium estimator in Section 4. In Section 5, we introduce margin trading into the cross-chain exchange and study its practical implementation issues. Numerical results are presented in Section 6, followed by conclusions and discussions in Section 7.

2 Atomic Swap

In the financial market, when two parties want to exchange their currencies or assets, trusted third-party intermediaries like banks and exchanges have to be involved to ensure the fairness and smoothness of the transaction. Otherwise, one of the parties may cheat and do not transfer the promised asset to the other after getting the counterparty's asset. In a cryptocurrency world, the atomic swap, which is a new smart contract technology, is created to enable the exchange of one cryptocurrency for another without using trusted third-party intermediaries. In this section, we introduce the atomic swap protocol. We use the same example in Liu (2018) and Herlihy (2018) to illustrate the procedure of the classical atomic swap. Suppose that there are two parties named Alice and Bob. Alice has α-coins and Bob has β-coins (the currency is usually called the *coin* in the cryptocurrency world). Alice wants to exchange Bob's 1 β-coin with her own 1 α-coin, and Bob is willing to accept this transaction.

The key to the atomic swap is the hashed time lock contract (HTLC). In the HTLC, there are two locks, *hashlock* h, which locks the asset in the contract, and *timelock* t, which locks the effective time of the contract. For the hashlock, if one party provides the correct secret s such that $h = H(s)$, where $H(\cdot)$ is a cryptographic hash function, then the contract is unlocked, and the asset in the contract can be withdrawn. The timelock is used to set time constraints on the contract. The basic steps of the atomic swap are provided in Figure 1. Before introducing the exchange procedure, we first explain some notations in the figure. We call the blockchains of α-coins and β-coins the *α-chain* and *β-chain*, respectively. For simplicity, we use "A" and "B" to represent Alice and Bob, respectively. Since the atomic swap happens on these two blockchains, both Alice and Bob have cryptocurrency wallets on these two blockchains. Specifically, A_α and B_α represent the wallets of Alice and Bob on the *α-chain*, respectively; A_β and B_β represent the wallets of Alice and Bob on the *β-chain*, respectively. "Tx" is short for "transaction", and T and ΔT are preset time in the contract. The procedure of the atomic swap is as follows (also see Nolan 2018).

Procedure of Atomic Swap:

(1) Alice initiates the atomic swap and generates a random secret s for verifying the hashlock.

(2) Alice creates two transactions:

1. Tx1: Pay 1 α-coin to B_α (the Bob's wallet in the α-chain)[1] if the secret s for $H(s)$ is verified and Tx1 is signed by Bob.
2. Tx2: Pay 1 α-coin from Tx1 to A_α, locked time $T + \Delta T$, i.e., this transaction will be effective after time $T + \Delta T$.

Then Alice sends Tx2 to Bob.

(3) Bob signs Tx2 and returns to Alice. That is, Bob agrees and approves Tx2.

(4) Alice broadcasts Tx1 on the α-chain.

(5) Bob creates two transactions:

1. Tx3: Pay 1 β-coin to A_β (the Alice's wallet in the β-chain) if the secret s for $H(s)$ is verified and Tx3 is signed by Alice.

[1]In the transaction of cryptocurrency, one party usually sends cryptographic coins to an address created by the other party in an output, and the other party links this address to his own cryptocurrency wallet. In this paper, we ignore these technical details of recording transactions.

2. Tx4: Pay 1 β-coin from Tx3 to B_β, locked time T, i.e., this transaction will be effective after time T.

Then Bob sends Tx4 to Alice.

(6) Alice signs Tx4 and returns to Bob. That is, Alice agrees and approves Tx4.

(7) Bob broadcasts Tx3 on the β-chain.

(8) Alice uses the secret s to unlock the coin locked in Tx3 before time T, and the secret s is revealed to Bob.

(9) Bob uses the revealed secret s to unlock the coin locked in Tx1 before time $T + \Delta T$.

(10) The atomic swap is finished: Alice gets 1 β-coin and Bob gets 1 α-coin.

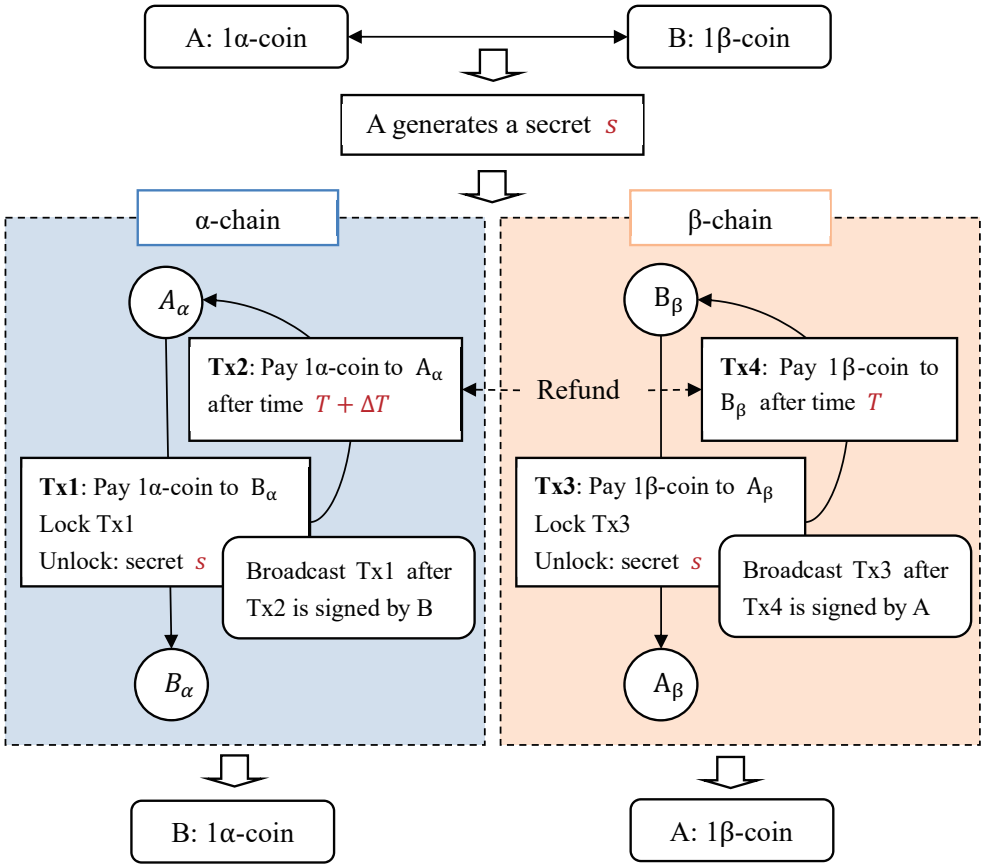

Figure 1: Atomic swap procedure

This procedure ignores the technical details of payments in the blockchain, and lists the sketches of the transaction. Notice that Tx1 and Tx3 are key transactions for exchanging coins, and make

the coins locked in contracts. Since Alice initiates the transaction, she has the secret s, and she can use the secret s to unlock Tx3 to get 1 β-coin. At the same time, the secret s is revealed to Bob. Then Bob can use the secret s to unlock Tx1 to get 1 α-coin, and the transaction is finished. Tx2 and Tx4 are protective transactions to ensure that the coins can be refunded to Alice's and Bob's wallets if the transaction falls through. Notice that only when Tx2 is signed by Bob, Alice then broadcasts Tx1 on the α-chain, and other ledgers of the α-chain can record this transaction. In other words, nothing happens on the α-chain before Tx2 is signed (what happens on β-chain is the same). On the other hand, if the secret s is not revealed before time T, then the 1 β-coin returns to Bob's wallet, and the 1 α-coin returns to Alice's wallet after extra time ΔT.

Remark 1. *The timelock of Tx2 should be longer than the timelock of Tx4, i.e., ΔT should be a proper positive number. For example, ΔT is usually set to be 24 hours in some contracts. This plays a crucial role in preventing the occurrence of cheating. Since Alice initiates the transaction, she has the secret s, and she can unlock Tx3 at any time before T (the β-coin will be refunded to Bob's wallet after T). If the timelock of Tx2 is shorter than the timelock of Tx4, i.e., $\Delta T < 0$, Alice can use her secret s to unlock Tx1 just before $T + \Delta T$, and Bob may not have enough time to use the revealed secret s to unlock Tx1 (since after $T + \Delta T$, the α-coin will be refunded to Alice's wallet), so Alice may trick Bob by getting both 1 β-coin and 1 α-coin. In other words, the extra time ΔT of the Tx2 is used to provide enough time for Bob to unlock Tx1.*

3 New Procedure of Cross-chain Exchange

According to the procedure of atomic swap, Alice has a right, but not an obligation, to reveal the secret s before T to get Bob's 1 β-coin. More specifically, after Alice and Bob signing Tx4 and Tx2, if Alice finds that the value of the α-coin is higher than the value of the β-coin, she may not reveal the secret s, and let the transaction fall through. While if she finds that the value of the α-coin is lower than the value of the β-coin, she may reveal the secret s immediately. However, the β-coin is locked in Tx3, so Bob has no choice but to execute the transaction. That is, *such a procedure gives Alice a (financial) option and is unfair to Bob.* Besides, the cryptocurrencies may fluctuate significantly, which gives Alice a profitable arbitrage opportunity. So we cannot ignore this optionality when using the atomic swap. Liu (2018) and Han et al. (2019) also point out the

optionality embedded in the atomic swap, but they do not conduct an in-depth analysis of this optionality. In this paper, we design a new procedure based on the atomic swap and provide a general and in-depth quantitative analysis of the optionality.

Because of the optionality, Alice, the initiator of the cross-chain exchange, should pay a *premium* p for this atomic swap. This premium p is used to compensate Bob's obligation without the corresponding right, i.e., the obligation that Bob has to execute the transaction once he signs Tx2. In addition, in the cryptocurrency world, almost all the accounts and transactions are anonymous. So that there is no credit binding, and anyone could cheat in the transaction. Thus, the design of the contract should be very careful to ensure that no one can exploit some loopholes of the exchange. Specifically, since there is no third-party intermediary in the cross-chain exchange, synchronization is a critical consideration when we design the exchange procedure. Here the synchronization means that the payment of the premium and the option contract should take effect simultaneously. Otherwise, one of the parties may cheat in the transaction: If the payment of the premium is sent to Bob first, Bob may not sign Tx2 (i.e., enter the option contract), and may get the premium for free; If the option contract takes effect first, Alice may postpone or refuse to pay the premium. In this paper, we design a new procedure that embeds the option into the atomic swap; meanwhile, we use another HTLC to guarantee the synchronization of the payment and the option contract. We summarize the procedure in Figure 2 and provide the specific procedure as follows. We will show the method to determine the premium p in the next section.

Procedure of Atomic Swap with Premium:

(1) Alice initiates the exchange and generates two random secret s and s_0 for verifying the hashlocks.

(2) Alice creates two transactions:

1. Tx1: Pay 1 α-coin to B_α if the secret s for $H(s)$ is verified and Tx1 is signed by Bob.
2. Tx2: Pay 1 α-coin from Tx1 to A_α, locked time $T + \Delta T$, i.e., this transaction will be effective after time $T + \Delta T$.

Then Alice sends Tx2 to Bob.

(3) Bob signs Tx2 and returns to Alice.

(4) Alice broadcasts Tx1 on the α-chain.

(5) Bob creates two transactions:

1. Tx3: Pay 1 β-coin to A_β if the secret s for $H(s)$ is verified and Tx3 is signed by Alice.
2. Tx4: Pay 1 β-coin from Tx3 to B_β, locked time T, i.e., this transaction will be effective after time T.

Then Bob sends Tx4 to Alice.

(6) Alice signs Tx4 and returns to Bob.

(7) Alice creates two transactions:

1. Tx5: Pay p α-coin to B_α if the secret s_0 for $H(s_0)$ is verified and Tx5 is signed by Bob.
2. Tx6: Pay p α-coin to A_α, locked time $T + \Delta T$.

Then Alice sends Tx6 to Bob.

(8) Bob signs Tx6 and returns to Alice.

(9) Alice broadcasts Tx5 on the α-chain.

(10) Bob creates Tx7: Broadcast Tx3 on the β-chain if the secret s_0 for $H(s_0)$ is verified and Tx7 is signed by Alice.

(11) Bob broadcasts Tx7 on the β-chain.

(12) Alice uses the secret s_0 to unlock Tx7 to make Tx3 be broadcast on the β-chain, and the secret s_0 is revealed.

(13) Bob uses the revealed secret s_0 to unlock the p α-coin locked in Tx5 before time $T + \Delta T$.

(14) Alice uses the secret s to unlock the coin locked in Tx3 before time T, and the secret s is revealed.

(15) Bob uses the revealed secret s to unlock the coin locked in Tx1 before time $T + \Delta T$.

(16) The exchange is finished: Alice gets 1 β-coin, and Bob gets 1 α-coin and p α-coin premium.

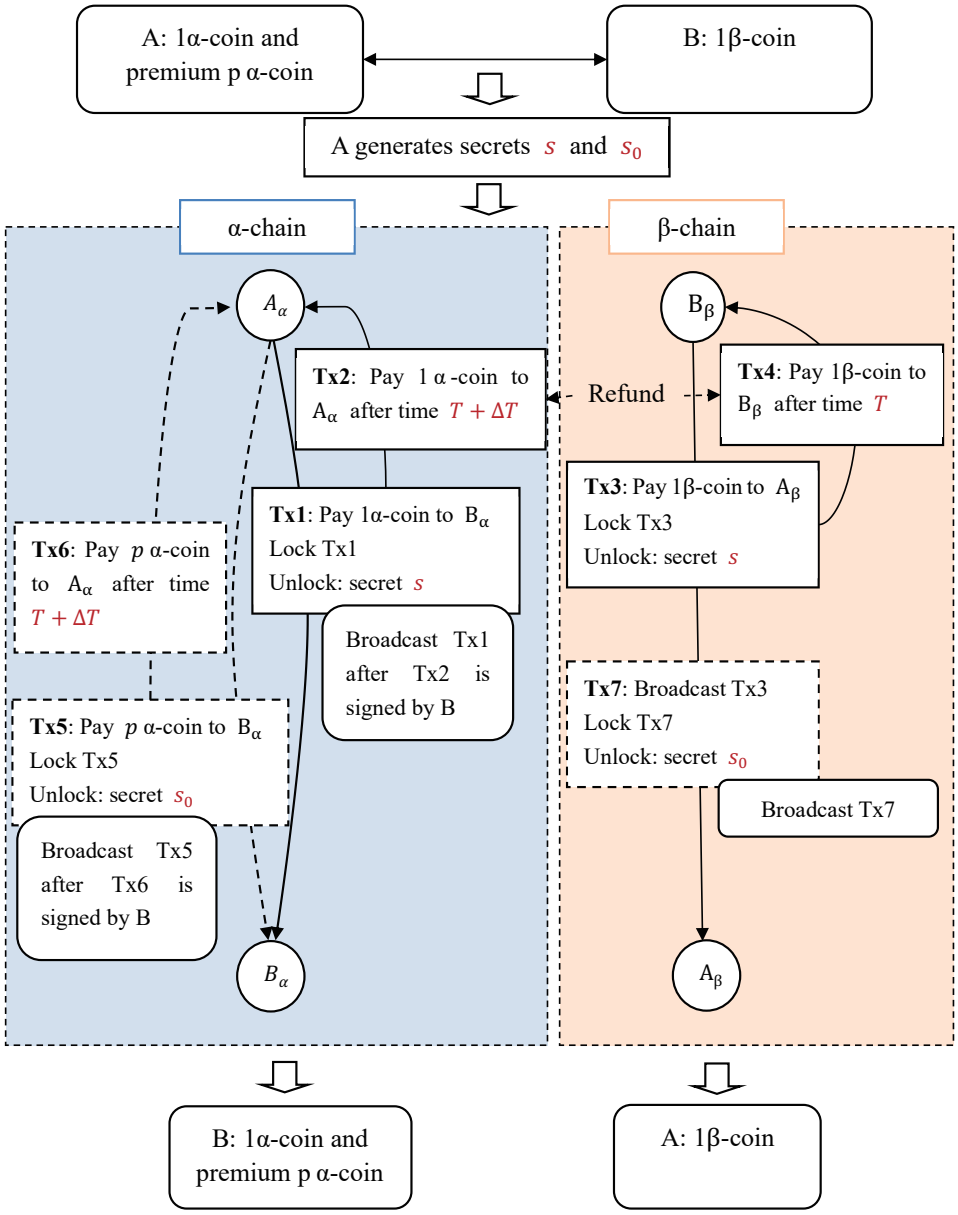

Figure 2: Atomic swap procedure with premium

Most of the steps are the same as the original atomic swap provided in Section 2 but adding another line of transactions (dashed line on the α-chain in Figure 2). This line of transactions aims to pay the premium p to Bob for exchanging the embedded option without a third-party intermediary. First, Alice needs to create another two transactions Tx5 and Tx6. Tx5 pays the premium p to Bob if the secret s_0 is verified, and Tx6 ensures that the premium will be refunded to Alice if Bob does not unlock Tx5. Bob does not unlock Tx5 also means that Bob does not get the premium, so the option will not be effective. Next, Bob creates Tx7 to ensure that the right of the option and the premium are exchanged fairly and no one tricks the other. Tx7 specifies that Tx3 will be broadcast only after Alice using s_0 to unlock Tx7. At that time, s_0 is revealed to Bob

and Bob can use s_0 to unlock Tx5 to get the premium p. If Alice does not reveal s_0, she cannot make Tx3 be broadcast on the β-chain, so the option will not be effective. On the other hand, such transactions can also prevent that Bob does not broadcast Tx3 (option is not effective) but gets the premium.

Notice that, in this paper, the time T and ΔT are idealized as constants. However, in current blockchain transactions (e.g., the bitcoin), the waiting time for the transaction to be added into the blockchain fluctuates, and the transaction cost varies over time, especially during heavy traffics. These technical problems are mainly caused by the low efficiency of the blockchain ledger, and may be fixed as the development of new blockchain techniques. For example, we may use the lightning network (Liu 2018), which can speed up the blockchain payment without worrying about waiting time.

4 Premium Estimation

In this section, we consider how to estimate a proper premium p. Recall that the atomic swap gives the initiator of the transaction (Alice) a right, but not an obligation, to exchange one cryptocurrency to another cryptocurrency from the counterparty (Bob), and the initiator can also choose her favorite transaction time before T. So this option is an American exchange option (Hull 2012), and T is the maturity time.

4.1 The model

We assume a complete probability space $(\Omega, \mathcal{F}, P)$ equipped with a natural filtration $\{\mathcal{F}_t\}_{t\in[0,T]}$. Let $S_\alpha(t)$ and $S_\beta(t)$ be the price[2] of the α-coin and β-coin at time t, respectively. We denote by $Z(t) = e^{-rt}\max(S_\beta(t) - S_\alpha(t), 0), t \in [0, T]$ an adapted payoff process, where r is the risk-free interest rate, and assume that $Z(t)$ is Markovian. Then the fair premium (American exchange option) is given by

$$p = \max_{0\le\tau\le T} \mathbb{E}\left[Z(\tau)\right] = \max_{0\le\tau\le T} \mathbb{E}\left[e^{-r\tau}\max(S_\beta(\tau) - S_\alpha(\tau), 0)\right], \tag{1}$$

[2]For ease of analysis, we assume that both coins have an explicit price just like stocks. We can also represent the value of one coin by the other coin, i.e., we use one of the coins as numeraire.

where τ is a stopping time adapted to the filtration generated by the processes of $Z(t)$.

Notice that a cryptocurrency is not issued by a central bank or qualified commercial banks of a country, and there are no well-traded bonds endorsed by governments, so we do not have a typical risk-free interest rate like real currencies. To determine a risk-free interest rate, we provide two simple ways. Firstly, we can use the cryptocurrency futures to extract the risk-free interest rate. Based on the future price formula in Chapter 5 of Hull (2012), Vojtko and Padyšák (2020) pointed out that the risk-free interest rate can be extracted by

$$r = \frac{1}{t_2 - t_1} \left(\log P(t_2) - \log P(t_1) \right), \tag{2}$$

where $t_1 < t_2$ and $P(t_1)$ and $P(t_2)$ are the bitcoin future prices at different delivery months. Secondly, we may treat a cryptocurrency as a commodity. If there is no place that we can save the cryptocurrency to earn interest, we may just set the risk-free interest rate to be zero, which is the same as the setting of other papers that study the cryptocurrency derivatives, (see Perez 2018 and Hou et al. 2020). In numerical examples, we choose the first way to determine the risk-free interest rate.

According to Margrabe (1976), if there are no dividends for $S_\alpha(t)$ and $S_\beta(t)$, the price of American exchange option equals to the price of its European counterpart. Specifically, assume that the price models are geometric Brownian motions, i.e.,

$$\frac{dS_\alpha(t)}{S_\alpha(t)} = \mu_\alpha dt + \sigma_\alpha dW_\alpha(t),$$

where μ_α and σ_α are the drift and volatility of α-coin, respectively, and W_α is a Brownian motion;

$$\frac{dS_\beta(t)}{S_\beta(t)} = \mu_\beta dt + \sigma_\beta dW_\beta(t),$$

where μ_β and σ_β are the drift and volatility of β-coin, respectively, and W_β is another Brownian motion. Let ρ denote the correlation between $dW_\alpha(t)$ and $dW_\beta(t)$.

Under the risk-neutral framework, we can change the measure to the risk-neutral measure, and

we have $\mu_\alpha = \mu_\beta = r$. Then the closed-form formula of the exchange option price is given by

$$p = S_\beta(0)N(d_1) - S_\alpha(0)N(d_2), \tag{3}$$

where

$$d_1 = \frac{\log\left(\frac{S_\beta(0)}{S_\alpha(0)}\right) + \frac{1}{2}\sigma^2 T}{\sigma\sqrt{T}}, \quad d_2 = d_1 - \sigma\sqrt{T},$$

and $\sigma^2 = \sigma_\beta^2 - 2\sigma_\beta\sigma_\alpha\rho + \sigma_\alpha^2$.

However, the non-dividend condition does not hold in the cryptocurrency world. Recently, there are some discussion on how to earn cryptocurrency dividends in the cryptocurrency industry, and some cryptocurrencies may have dividends through new business models. *Staking* is one of the blockchain business models that allow token holders to obtain rewards or dividends through holding funds in a cryptocurrency wallet to support the security and operations of a blockchain network (Binance 2020). For example in *Bankroll*[3], which is a decentralized finance network that provides rewards through staking, the customer who deposits the cryptocurrency in the wallet of this network can obtain the dividend after a certain amount of time. So if holding the cryptocurrency can bring more benefits than holding the embedded option, then the option should be exercise early, and we cannot use the closed-form formula of the European exchange option price to calculate the premium. In addition, the price model of a cryptocurrency may be complex. According to Figure 5 in Section 6, the probability distribution of daily returns of bitcoin is leptokurtic and heavy-tailed, so geometric Brownian motion is not a suitable model. And we may need more sophisticated models, such as Lévy process models and jump-diffusion models, to describe the evolution of the bitcoin and other cryptocurrencies prices. When sophisticated models are used, closed-form pricing formulas are usually not available, and one may have to resort to Monte Carlo simulation to estimate the premium.

In the next subsection, we provide a simulation-based algorithm, specifically, the least-squares Monte Carlo method, to estimate the price of the American exchange option (i.e., the premium). In the following, we consider a discrete-time valuation framework, i.e., the option can be only exercised at a series of discrete-time points $0 = t_0 < t_1 \leq t_2 \leq \cdots \leq t_K = T$, where K is the

[3]see https://bankroll.network

number of discrete periods. In other words, we regard the American option as a Bermuda option. At each time t_k, Alice needs to determine whether to reveal the secret s immediately or continue to hold until the next time point t_{k+1}.

4.2 Least-squares Monte Carlo method

The least-squares Monte Carlo (LSM) method for pricing American-style options was first proposed by Longstaff and Schwartz (2001), and then developed by other scholars, see Clément et al. (2002), Stentoft (2004), Gerhold (2011), and Zanger (2013). We introduce the algorithm based on the framework in Stentoft (2004). Recall that $Z(t) = e^{-rt}\max\left(S_\beta(t) - S_\alpha(t), 0\right)$. Then, we define a series of optimal stopping times $\{\tau(t_k)\}_{k=0}^K$:

$$\begin{cases} \tau(t_K) &= T, \\ \tau(t_k) &= t_k \mathbf{1}_{\{Z(t_k)\geq \mathbb{E}[Z(\tau(t_{k+1}))|\mathcal{F}_t]\}} + \tau(t_{k+1})\mathbf{1}_{\{Z(t_k)<\mathbb{E}[Z(\tau(t_{k+1}))|\mathcal{F}_t]\}}. \end{cases} \tag{4}$$

Notice that the conditional expectation in the indictor function is the expected value of the option holding to the time t_{k+1} given the information of time t_k. Let

$$\begin{aligned} F(t_k) &\triangleq F(S_\alpha(t_k), S_\beta(t_k)) = \mathbb{E}\left[Z(\tau(t_{k+1}))|\mathcal{F}_{t_k}\right] \\ &= \mathbb{E}\left[e^{-r\tau(t_{k+1})}\max\left(S_\beta(\tau(t_{k+1})) - S_\alpha(\tau(t_{k+1})), 0\right)|\mathcal{F}_{t_k}\right] \\ &= \mathbb{E}\left[e^{-r\tau(t_{k+1})}\max\left(S_\beta(\tau(t_{k+1})) - S_\alpha(\tau(t_{k+1})), 0\right)|S_\alpha(t_k), S_\beta(t_k)\right]. \end{aligned} \tag{5}$$

The key idea of LSM is to use the cross-sectional information in the simulation sample paths and least-squares regression to approximate the conditional expectations $F(t_k), k = 1, \ldots, K-1$. Specifically, we define a class of basis functions $\{\phi_m(\cdot,\cdot)\}_{m=0}^\infty$, and according to Longstaff and Schwartz (2001) and Stentoft (2004),

$$F(t_k) = \sum_{m=0}^{\infty} \phi_m\left(S_\alpha(t_k), S_\beta(t_k)\right) a_m(t_k), \tag{6}$$

where $\{a_m(t_k)\}_{m=0}^\infty$ are corresponding coefficients of the basis functions. This representation is based on the theory of Hilbert space, that is, any functions in the Hilbert space can be presented

by a linear combination of countable basis vectors for this space (see Stentoft 2004). However, in practice, we can only use a finite number of basis functions to approximate $F(t_k)$. Define $F_M(t_k)$ as

$$F_M(t_k) = \sum_{m=0}^{M-1} \phi_m \left(S_\alpha(t_k), S_\beta(t_k)\right) a_m(t_k),$$

so the stopping times are also approximated by

$$\begin{cases} \tau_M(t_K) &= T, \\ \tau_M(t_k) &= t_k \mathbf{1}_{\{Z(t_k) \geq F_M(t_k)\}} + \tau_M(t_{k+1}) \mathbf{1}_{\{Z(t_k) < F_M(t_k)\}}. \end{cases}$$

Suppose that we have N simulation sample paths of the α-coin and β-coin prices, which are denoted by $\{S_\alpha^n(t_k), k = 0, 1, \ldots, K\}_{n=1}^N$ and $\{S_\beta^n(t_k), k = 0, 1, \ldots, K\}_{n=1}^N$, respectively. Here the superscript n is used to denote the index of the simulation sample. Denote the estimates of the coefficients at time t_k as $\{\hat{a}_m^N(t_k)\}_{m=0}^{M-1}$, $k = 0, 1, \ldots, K-1$, and we can use least-squares regression to estimate these coefficients backward. The specific steps refer to Step 9 to Step 13 in Algorithm 1. After obtaining the estimates of the coefficients, the estimated conditional expectations and the optimal stopping times along the nth simulation sample path $(S_\alpha^n(t_k), S_\beta^n(t_k))$ are given by

$$\hat{F}_M^{N,n}(t_k) = \sum_{m=0}^{M-1} \phi_m \left(S_\alpha^n(t_k), S_\beta^n(t_k)\right) \hat{a}_m^N(t_k), \tag{7}$$

and

$$\begin{cases} \hat{\tau}_M^{N,n}(t_K) &= T, \\ \hat{\tau}_M^{N,n}(t_k) &= t_k \mathbf{1}_{\left\{Z(t_k) \geq \hat{F}_M^{N,n}(t_k)\right\}} + \hat{\tau}_M^{N,n}(t_{k+1}) \mathbf{1}_{\left\{Z(t_k) < \hat{F}_M^{N,n}(t_k)\right\}}. \end{cases}$$

Then, we generate another two sets of simulation paths $\{S_\alpha^l(t_k), k = 0, 1, \ldots, K\}_{l=1}^L$ and $\{S_\beta^l(t_k), k = 0, 1, \ldots, K\}_{l=1}^L$, respectively, and use (7) to estimate the optimal stopping times $\hat{\tau}_M^{N,l}(t_k)$ along the new sample paths. At last, we calculate the sample mean $1/L \sum_{l=1}^N Z^l(\hat{\tau}_M^{N,l}(0))$. Notice that we can use different types of price models in LSM as long as we can simulate the sample paths of the prices. Generally, we assume that the parameter sets of the price models for α-coin and β-coin are Θ_α and Θ_β, respectively. In the following algorithm, we assume the parameter sets of the price models are given in advance. In Section 6, we will show how to use maximum likelihood estimation to calibrate the model parameters. Then we summarize the whole procedure in Algorithm 1.

Algorithm 1 LSM Algorithm

Input: The initial price $S_\alpha(0)$ and the model parameters Θ_α of α-coin, the initial price $S_\beta(0)$ and the model parameters Θ_β of β-coin, the maturity T, the number of time periods K, the exercise time $0 = t_0 < t_1 \leq t_2 \leq \cdots \leq t_K = T$, the number of simulation sample paths N for approximating conditional expectation; the number of new simulation paths L to estimate p.

for $n = 1$ to N **do**
  for $k = 1$ to K **do**
    Generate sample path $S_\alpha^n(t_k)$ based on $S_\alpha^n(t_{k-1})$;
    Generate sample path $S_\beta^n(t_k)$ based on $S_\beta^n(t_{k-1})$;
    Set $Z^n(t_k) = e^{-rt_k} \max\left(S_\beta^n(t_k) - S_\alpha^n(t_k), 0\right)$;
  end for
end for
Let $\hat{\tau}_M^{N,n}(t_K) = T$ and $\hat{F}_M^{N,n}(\hat{\tau}_M^{N,n}(t_K)) = Z^n(t_K)$, for $n = 1, 2, \ldots, N$;
for $k = K-1, ..., 1$ **do**
  Generate basis functions $\{\phi_m(S_\alpha^n(t_k), S_\beta^n(t_k))\}_{m=0}^{M-1}$, for $n = 1, 2, \ldots, N$;
  Solve the least-squares problem to obtain the coefficients $\{\hat{a}_m^N(t_k)\}_{m=0}^{M-1}$

$$\min_{\{a_m^N(t_k)\}_{m=0}^{M-1}} \sum_{n=1}^{N} \left(\sum_{m=0}^{M-1} \phi_m(S_\alpha^n(t_k), S_\beta^n(t_k)) a_m^N(t_k) - \hat{F}_M^{N,n}(\tau_M^n(t_{k+1})) \right)^2;$$

  For $n = 1, 2, \ldots, N$, let $\hat{F}_M^{N,n}(t_k) = \sum_{m=0}^{M-1} \phi_m(S_\alpha^n(t_k), S_\beta^n(t_k)) \hat{a}_m^N(t_k)$, and

$$\hat{\tau}_M^{N,n}(t_k) = t_k \mathbf{1}_{\left\{Z^n(t_k) \geq \hat{F}_M^{N,n}(t_k)\right\}} + \hat{\tau}_M^{N,n}(t_{k+1}) \mathbf{1}_{\left\{Z^n(t_k) < \hat{F}_M^{N,n}(t_k)\right\}};$$

end for
for $l = 1$ to L **do**
  for $k = 1$ to K **do**
    Generate new sample path $S_\alpha^l(t_k)$ based on $S_\alpha^l(t_{k-1})$;
    Generate new sample path $S_\beta^l(t_k)$ based on $S_\beta^l(t_{k-1})$;
    Set $Z^l(t_k) = e^{-rt_k} \max\left(S_\beta^l(t_k) - S_\alpha^l(t_k), 0\right)$;
  end for
end for
Set $\hat{\tau}_M^{N,l}(t_K) = T$ and estimate $\hat{\tau}_M^{N,l}(t_k), k = K-1, \ldots, 1$ as Step 12, for $l = 1, 2, \ldots, L$;
Estimate the premium $\hat{p}_M^{N,L} = 1/L \sum_{l=1}^{L} Z^l(\hat{\tau}_M^{N,l}(0))$.

Output: Return $\hat{p}_M^{N,L}$.

Remark 2. *The methods to generate simulation paths of α-coin and β-coin (see Steps 3, 4, 16 and 17 in the algorithm) are adapted to their models under a risk-neutral framework. For example, if the price model of α-coin is a geometric Brownian motion, then its model parameters are the risk-free interest rate r, the dividend q_α, and the volatility σ_α, i.e., $\Theta_\alpha = (r, q_\alpha, \sigma_\alpha)$. We then use the Girsanov's theorem (Musiela and Rutkowski 2005) to change the parameters from the physical measure to the risk-neutral measure, i.e., changing the drift of the geometric Brownian motion as $r - q_\alpha$ and keeping volatility σ_α unchanged, and Steps 3 and 16 are given by*

$$S_\alpha^n(t_k) = S_\alpha^n(t_{k-1}) \exp\left((r - q_\alpha)(t_k - t_{k-1}) + \sigma_\alpha \sqrt{t_k - t_{k-1}} Z_k\right),$$

where Z_k is a standard normal random variable.

Remark 3. *If we assume that the volatility of α-coin or β-coin is 0, i.e., the price of one coin increases with a constant rate, then this exchange option is an American call or put option. Thus, the procedure in Figure 2 can be regarded as a way of decentralized American option trading.*

According to the lemma in Stentoft (2004), we prove the convergence of Algorithm 1 and summarize in Theorem 1. The assumptions and proof are provided in Appendix A.

Theorem 1. *Under Assumptions 1-4, if $M = M(N)$ is increasing in N such that $M \to \infty$ and $M^3/N \to 0$, then $\hat{F}_M^N(t_k) \to F(t_k)$ in probability, for $k = 0, 1, \ldots, K$. In addition, let $N(L)$ is increasing in L such that $N \to \infty$ as $L \to \infty$, then*

$$\sqrt{L}(\hat{p}_M^{N,L} - p) \to N(0, \sigma^2),$$

where $\sigma^2 = \mathrm{Var}(Z(\tau(0)))$.

5 Cross-Chain Exchange with Margin Trading

Based on the procedure of the cross-chain exchange in this paper, once the transaction is established, all the coins are locked in the contract, which may tie up the capital of both sides of the transaction. To fix this issue, we consider a margin trading in the cross-chain exchange, i.e., both sides of the transaction deposit a fraction of their funds in the contract, and then deposit full

principal when the exchange is happening. In this section, we first provide a fair procedure of the cross-chain exchange with both premium and margin, then provide an algorithm to estimate the proper premium under this setting.

5.1 Procedure of cross-chain exchange with margin trading

In the procedure of cross-chain exchange with margin trading, we can use *SIGHASH flags* technique (see Liu 2018), which allows each party in the contract to only sign parts of the contract and other parts to be changed without their involvement. Specifically, we can use *SIGHASH_SINGLE* mode and *SIGHASH_ANYONECANPAY* modifier to sign a transaction contract. The parties of the transaction first deposit a fraction of their coins into the contract, and if one of the parties does not deposit the remaining coins in time, such signatures allow the other party to withdraw its own margin and get the counterparty's margin. One may see Bitcoinwiki (2019c) for more details about SIGHASH flags.

In this paper, we consider a fixed margin. Let m_α and m_β denote the margin level of Alice and Bob, respectively. Notice that $0 < m_\alpha, m_\beta \leq 1$, and the margin case degenerates to the original case as in Section 3 when $m_\alpha = m_\beta = 1$. The whole procedure of the cross-chain exchange with both premium and margin (see Figure 11 in Appendix B) and its description are provided in Appendix B. Here we only list the differences from the cross-chain exchange with premium in Section 3. That is, Step (2) and Step (5) are respectively replaced by

(2) Alice creates two transactions:

1. Tx1: Pay 1 α-coin to B_α (only deposit m_α α-coin first in the contract, and deposit the remaining coin before Tx1 unlocked by Bob) if the secret s is verified and Tx1 is signed by Bob.
2. Tx2: Pay all the α-coin in Tx1 to A_α, locked time $T + \Delta T$, and use the SIGHASH flag signature.[4]

Then Alice sends Tx2 to Bob.

[4]Notice that Tx2 is signed by Bob with SIGHASH, and the amount of the coins in it is preset as the same as the full principal in Tx1 in advance. If Alice does not deposit the remaining α-coin in Tx1, the coin in Tx2 cannot reach the preset amount either, and such a signature allows Bob to switch the payee from Alice to himself, i.e., Bob receives Alice's margin since Alice defaults. The specific steps to achieve this need to construct new transactions in the blockchains, which are beyond the scope of this paper, so we do not discuss these technical details in this paper. Bob's contract Tx4 functions similarly.

(5) Bob creates two transactions:

1. Tx3: Pay 1 β-coin to A_β (only deposit m_β β-coin first in the contract, and deposit the remaining coin before Tx3 unlocked by Alice) if the secret s is verified and Tx3 is signed by Alice.
2. Tx4: Pay all the β-coin from Tx3 to B_β, locked time T, and use the SIGHASH flag signature.

Then Bob sends Tx4 to Alice.

5.2 Premium estimation for margin trading

In the margin trading, if Bob finds that the price of his β-coin increases to a certain level, he may take the initiative to default and will not deposit the whole principal in Tx3. In this situation, Bob loses at most m_β β-coin. For Alice, her gain is at most m_β β-coin. Based on these facts, the premium of the exchange with margin is different from (1) and given by

$$p' = \max_{0\leq\tau\leq T} \mathbb{E}\left[\min(e^{-r\tau}\max(S_\beta(\tau) - S_\alpha(\tau), 0), m_\beta S_\beta(\tau))\right]. \tag{8}$$

According to (1) and (8),

$$\min(e^{-r\tau}\max(S_\beta(\tau) - S_\alpha(\tau), 0), m_\beta S_\beta(\tau)) \leq e^{-r\tau}\max(S_\beta(\tau) - S_\alpha(\tau), 0),$$

so we have the following result.

Proposition 1. *The premium of the cross-chain exchange with margin given by* (8) *is smaller than the premium of the cross-exchange without margin given by* (1)*. That is,*

$$p' \leq p.$$

To estimate p', we can still use the LSM, and the only difference is the payoff function

$$\min(e^{-rt}\max(S_\beta(t) - S_\alpha(t), 0), m_\beta S_\beta(t)).$$

The whole algorithm is provided in Appendix C.

6 Numerical Experiments

In this section, we conduct a sequence of numerical experiments to study the performance of the algorithms for estimating the premium with and without margin. We take the cross-chain exchange between bitcoin (BTC) and litecoin (LTC) as an example, and use geometric Brownian motion (GBM), variance gamma (VG) process, and Merton's jump-diffusion (MJD) process to model their prices. We collect the price data of bitcoin and litecoin at HitBTC[5], and summarize them in a file uploaded to Github (`https://github.com/wangzinishu/pricing-cryptocurrency`). We first analyze the data and provide some summary statistics in Table 1, and show the daily prices and returns of bitcoin and litecoin from February 1, 2019, to August 10, 2020, in Figures 3 and 4.

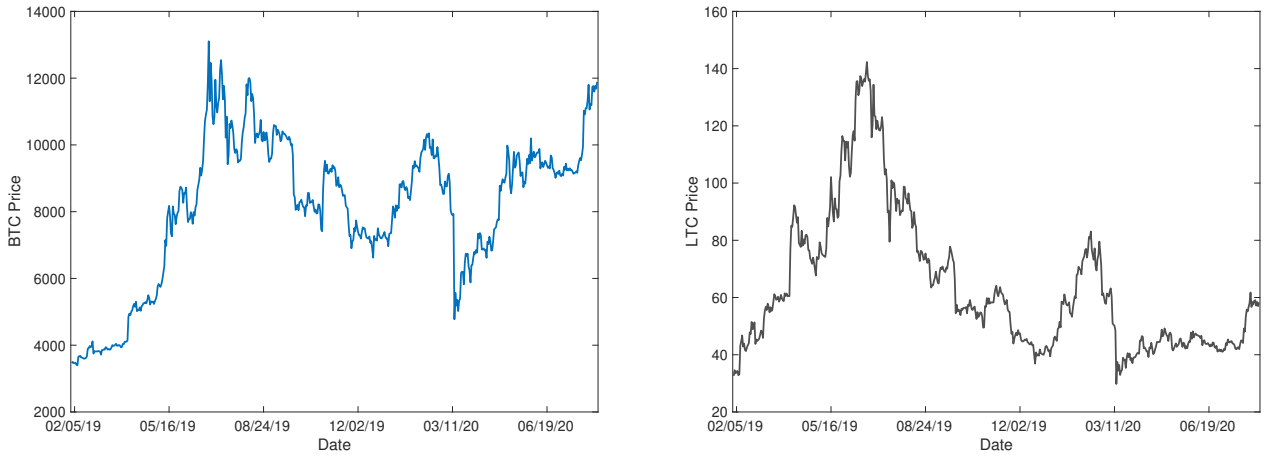

Figure 3: Daily prices of bitcoin (left panel) and litecoin (right panel)

[5]HitBTC is a global cryptocurrencies platform, which founded in 2013 with a venture capital of 6 million euros, supporting more than 100 cryptocurrencies transaction including BTC, ETH and LTC. Data from: `https://cn.investing.com/crypto/currencies`.

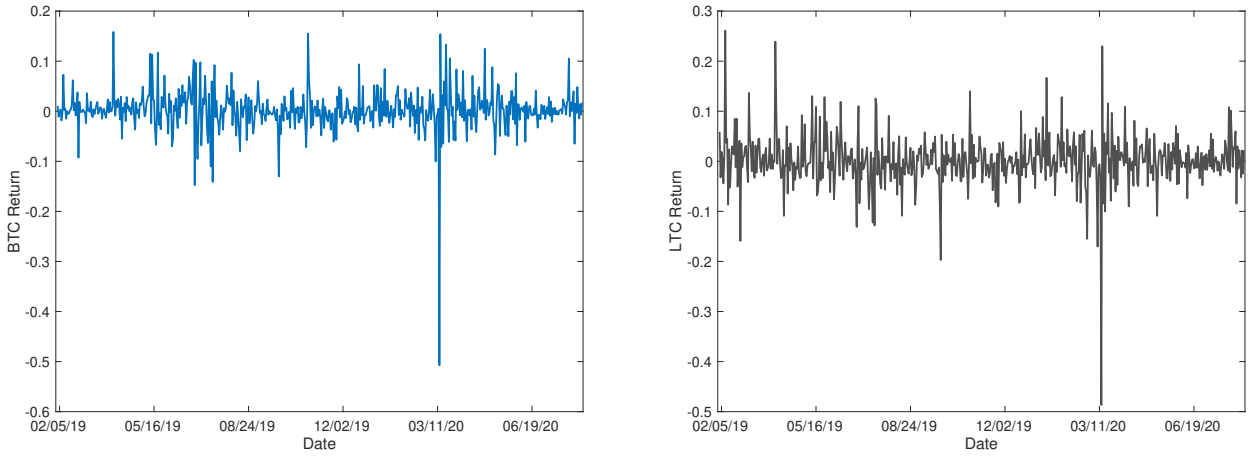

Figure 4: Daily returns of bitcoin (left panel) and litecoin (right panel)

Remark 4. *There is another line of research that models the underling prices as time-series models. Based on the price data, we do further analysis and find that the conditional heteroscedasticity exists in the bitcoin and litecoin data. So a multi-dimensional time-series model, e.g., a bivariate GARCH model, may be more suitable. We will leave this problem for future research.*

6.1 Calibration via maximum likelihood estimation

There are two critical issues in model parameter calibration. The first one is how to choose the data set to do the calibration. In the traditional financial market, when we calibrate the model parameters of an underlying asset to price options, we usually use the prices of another widely traded options for this underlying asset as the data set. Because such price data contains the expectation of the future price of this underlying asset under a risk-neutral world. However, for the cryptocurrencies, we do not have well-traded bitcoin and litecoin options[6], so we use the historical price data to do the calibration. In addition, the maturity of the embedded option in the atomic swap is quite small, so the recent historical data of bitcoin and litecoin may contain some information for the expectation of the future price. We believe that the rapid development of the cryptocurrency option markets will alleviate this issue. In the future, when we have enough data for the cryptocurrency options, we can calibrate the model parameters under the risk-neutral framework.

[6]Even though some bitcoin derivatives are traded in some exchanges, the transaction volume is relatively small, and the quality of the data is not good.

The second issue is how to determine a risk-free interest rate. As discussed in Section 4, we use the bitcoin future prices in the Chicago Mercantile Exchange to calibrate the risk-free interest rate via formula (2). Specifically, we use the settlement price of the futures on December 30th, 2020, to calculate the interest rate. There are four contracts with different delivery months, i.e., JAN 2021, FEB 2021, MAR 2021, and APR 2021, and the settlement prices are 27210, 27510, 27810, and 28100. So according to the formula above, we can calculate three daily interest rates, 0.036%, 0.039%, and 0.034%. Then we use the averaged value as the daily risk-free interest rate, i.e., 0.036%. How to determine the risk-neutral model parameters as well as how to determine a reasonable risk-free interest rate for cryptocurrencies are very important but difficult problems, which are left for future research.

Based on the data, we use the maximum likelihood ratio method to calibrate the parameters in BM[7], VG and MJD. Specifically, we have the probability density functions of the process $X_{\mathrm{BM}}(t)$, $X_{\mathrm{VG}}(t)$, and $X_{\mathrm{MJD}}(t)$, which are given by

$$f_{\mathrm{BM}}(x,t;\mu,\sigma)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu t)^2}{2\sigma^2 t}\right),$$

and

$$f_{\mathrm{VG}}(x,t;\sigma,\nu,\theta)=\frac{2\exp\left(\theta x/\sigma^2\right)}{\nu^{t/\nu}\sqrt{2\pi}\sigma\Gamma\left(\frac{t}{\nu}\right)}\left(\frac{x^2}{2\sigma^2/\nu+\theta^2}\right)^{\frac{t}{2\nu}-\frac{1}{4}}K_{\frac{l}{v}-\frac{1}{2}}\left(\frac{1}{\sigma^2}\sqrt{x^2\left(2\sigma^2/v+\theta^2\right)}\right),$$

where K is the modified Bessel function of the second kind (see Madan et al. 1998), and

$$f_{\mathrm{MJD}}(x,t;\mu_D,\sigma_D,\lambda,\mu_J,\sigma_J)=\sum_{k=0}^{\infty}\left[p_k(\lambda t)f_{\mathrm{BM}}\left(x,1;\left(\mu_D-\frac{\sigma_D^2}{2}\right)t+\mu_J k,\sigma_D^2 t+\sigma_J^2 k\right)\right].$$

Notice that $f_{\mathrm{MJD}}(x,t;\mu_D,\sigma_D,\lambda,\mu_J,\sigma_J)$ involves an infinite number of summation. In the numerical procedure, we use the first 100 terms to approximate the real probability density function f_{MJD}. We set $t=1$ (one day), then we can construct log-likelihood functions for BM, VG, and MJD, and obtain the model parameters, which are provided in Table 2.

In Section 6.2, the prices of bitcoin and litecoin are assumed to be GBMs. In Subsection 6.3, the returns of bitcoin and litecoin are VG or MJD processes. VG process is a pure jump Lévy

[7]The price is assumed to be GBM, so the return is BM.

process to model the return of asset prices, and it can better capture the characteristics of the daily return of bitcoin and litecoin, such as leptokurtic and heavy-tail. MJD is another asset model that can capture leptokurtic and heavy-tail of the returns, and it is suitable for sudden price changes in very short periods.

Based on the real data, we provide the estimated probability density functions of daily returns for BM, VG and MJD in Figure 5 and the Kullback-Leibler (KL) divergences between the estimated kernel distribution and the estimated normal (GBM)/estimated VG/estimated MJD distribution in Table 3. Both the figure and the table indicate that VG and MJD are superior to BM in modeling the returns of bitcoin and litecoin.

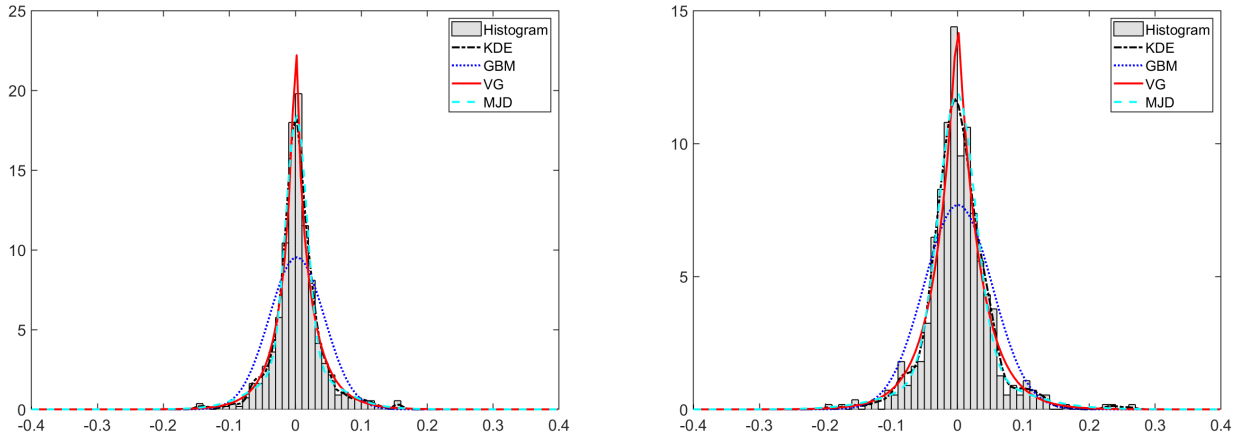

Figure 5: Histogram and estimated densities for the daily returns of bitcoin (left panel) and litecoin (right panel)

6.2 Example 1: Geometric Brownian motion

We first consider that the prices of bitcoin and litecoin are geometric Brownian motions (GBM). We can use the Girsanov's theorem to generate sample paths under the risk-neutral framework (see Remark 2). Let $S_\alpha(0) = S_\beta(0) = 1$, $\rho = 0.8124$, the risk-free interest rate $r = 0.00036$, the number of discrete periods $K = 100$, and $T = 1$. Let the dividends for both coins be 0. Then we use the European exchange option price, which is given by (3), as a benchmark to study the convergence of the premium. We compare three choices of basis functions:

(i) The complete set of the second degree polynomials (denoted by $d = 2$)

$$(1, S_\alpha, S_\beta, S_\alpha^2, S_\alpha S_\beta, S_\beta^2);$$

(ii) The complete set of the third degree polynomials (denoted by $d = 3$)

$$(1, S_\alpha, S_\beta, S_\alpha^2, S_\alpha S_\beta, S_\beta^2, S_\alpha^3, S_\alpha^2 S_\beta, S_\alpha S_\beta^2, S_\beta^3);$$

(iii) The complete set of the forth degree polynomials (denoted by $d = 4$)

$$(1, S_\alpha, S_\beta, S_\alpha^2, S_\alpha S_\beta, S_\beta^2, S_\alpha^3, S_\alpha^2 S_\beta, S_\alpha S_\beta^2, S_\beta^3, S_\alpha^4, S_\alpha^3 S_\beta, S_\alpha^2 S_\beta^2, S_\alpha S_\beta^3, S_\beta^4).$$

Then we obtain the boxplots of premiums (see Figure 6) for different N based on 50 macro-replications.

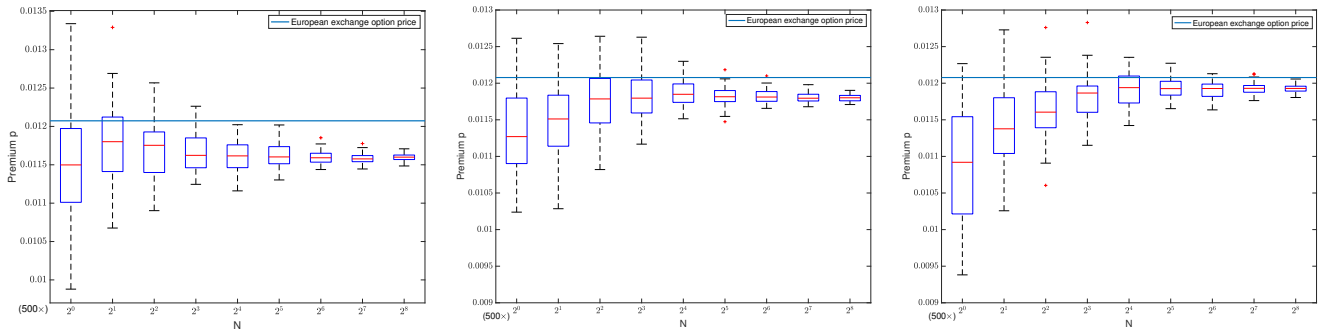

Figure 6: Boxplots of premiums in Example 1 (left panel $d = 2$; middle panel $d = 3$; right panel $d = 4$)

Figure 6 indicates that the LSM algorithm converges. Due to the discretization and the basis function approximation of the conditional expectation, the estimated premium is low-biased to its European counterpart. But when the complexity of the basis function is large, i.e., d is large, the bias can be reduced. This result can also be concluded from Table 4, i.e., when the number of simulation samples are large enough, using higher order polynomial as basis functions can have a better accuracy.

We consider the effect of dividends. We fix the dividend of α-coin as 0.0005, and change the

dividend of β-coin from 0.0005 to 0.0010. All the other parameters keep unchanged. Then we obtain Figure 7, which indicates that the American-style premium is different from the European-style if the dividend for α-coin and β-coin are different. That is, the dividend is indeed a critical issue so that we cannot use the European-style exchange option price formula to determine the premium in the atomic swap.

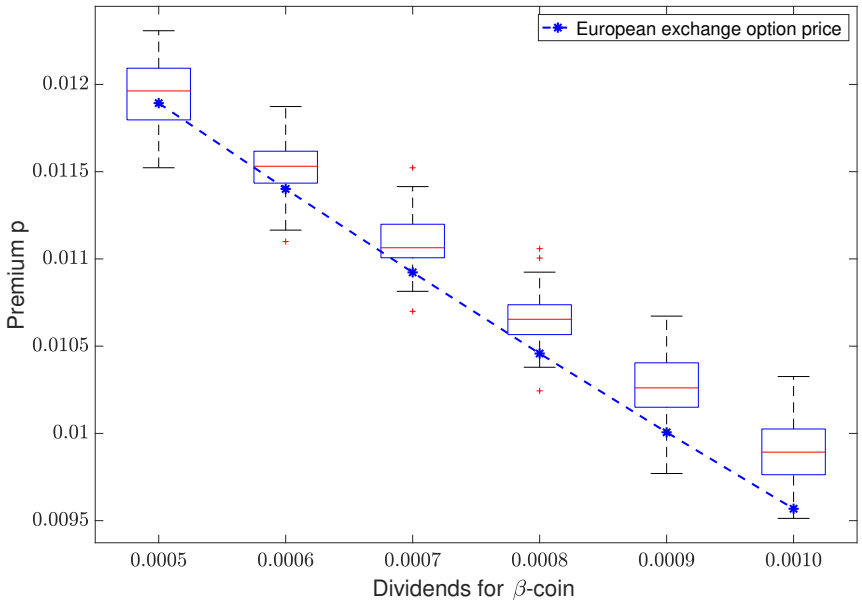

Figure 7: Boxplots of premiums with non-zero dividends in Example 1

Next, we consider the effect of margin. Let $m_\alpha = m_\beta = 0.1$ and $S_\alpha(0) = 0.85$ and $S_\beta(0) = 1$. The other parameters are the same as the case without margin. Notice that we do not have the true premium under this case, so we use estimated premium with $N = 5 \times 10^6$ and $d = 4$ as the true premium (0.10001) to calculate RMSEs. The results are presented in Table 5, which also indicates that the LSM algorithm works well. And when the number of simulation samples is large enough, we should choose the complete set of higher degree polynomials as basis functions.

In addition, we study the relationship between the premium and the margin level. Let $m_\alpha = m_\beta = m = \{0.02, 0.04, \ldots, 0.30\}$, and use the basis function set $d = 3$. Then, we obtain Figure 8. As the increasing of the margin level m, the term $e^{-rt}\max(S_\beta(t) - S_\alpha(t), 0)$ will dominate the term $m_\beta S_\beta(t)$ in the minimizing operation of (8), so the premium should be closer and closer to the price of the premium without margin, which has been shown in Figure 8. Based on the result, we may conclude that if Alice wants to make a cross-chain exchange with Bob, the margin trading

is a good choice since it costs less premium than that without margin.

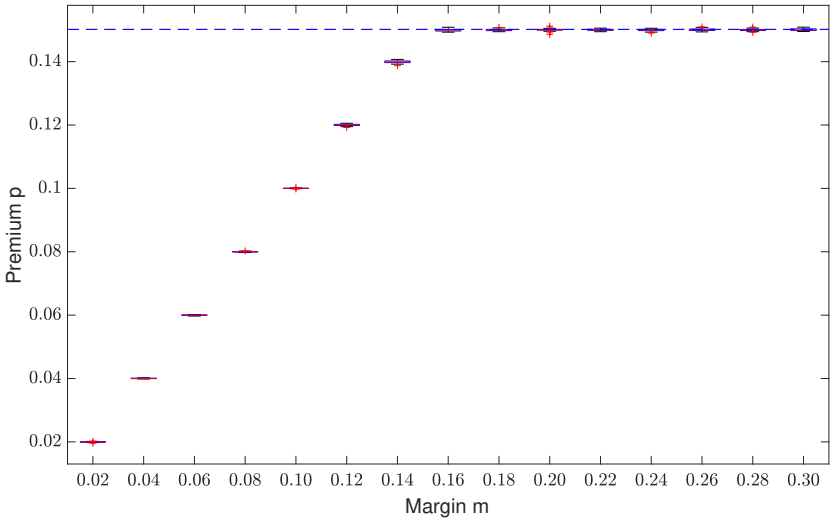

Figure 8: Boxplots of premiums with margin in Example 1

6.3 Example 2: Variance gamma and Merton's jump-diffusion process

In this subsection, we consider that the returns of bitcoin and litecoin are VG and MJD processes, respectively. Let $K = 20$, $S_\alpha(0) = S_\beta(0) = 1$, and dividends are both 0.0005. The samples of the VG process can be simulated via a time-changed Brownian motion, and Appendix D provides the way to simulate the price of VG under a risk-neutral measure. According to the simulation method, we first use Gaussian copula to generate correlated Gamma random variables, and then plug into the time-changed Brownian motion. The parameter in Gaussian copula is determined by a stochastic root-finding procedure such that the correlation between $S_\alpha(t)$ and $S_\alpha(t)$ equals $\rho = 0.8124$. The samples of the MJD can be simulated based on the geometric Brownian motion with a compound Poisson process, and Appendix D provides the way to simulate the price of MJD under a risk-neutral measure. To generate correlated MJD processes, we also use Gaussian copula to generate correlated Poisson random variable, and then calculate the parameter in Gaussian copula via a stochastic root-finding procedure such that the correlation between $S_\alpha(t)$ and $S_\alpha(t)$ equals $\rho = 0.8124$. For more details about the simulation of VG, MJD, and other Lévy process model, readers are referred to see Schoutens (2003) and Glasserman (2013).

The true premiums of VG and MJD are 0.00983 and 0.00899, respectively, which are calculated with $N = 5 \times 10^6$ and basis functions $d = 4$. Similar to the previous subsection, we obtain the

boxplots of premiums (see Figure 9) for different N based on 50 macro-replications. Figure 9 indicates that the estimated premiums for both VG and MJD converge to the true premium. Then, we consider the effect of margin. Let $S_\alpha(0) = 0.85$ and $S_\beta(0) = 1$, and other setting is the same as last subsection, and we obtain Figure 10.

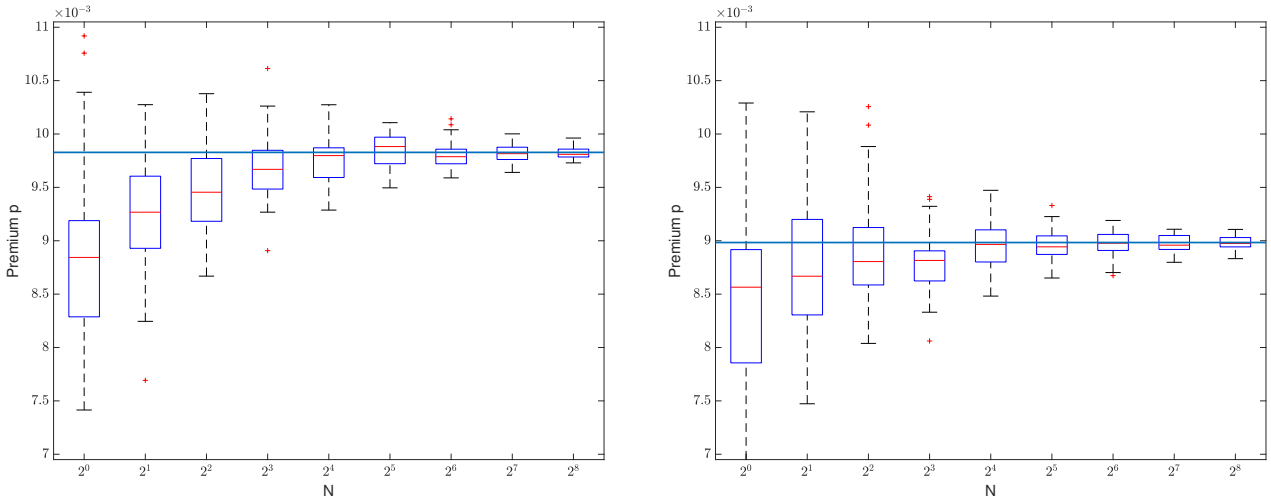

Figure 9: Boxplots of premiums for VG (left panel) and MJD (right panel) in Example 2

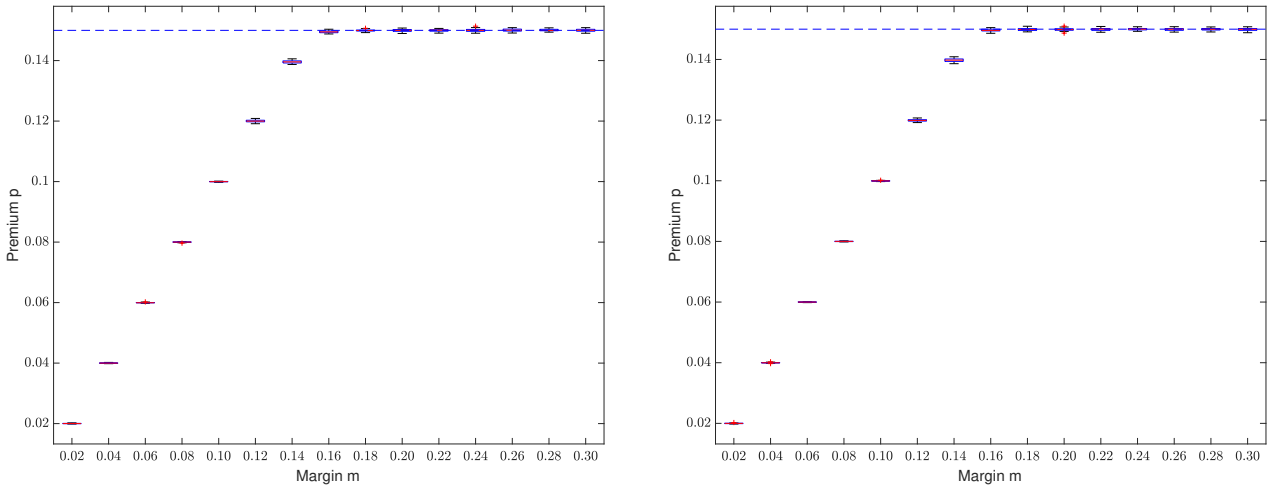

Figure 10: Boxplots of premiums with margin for VG (left panel) and MJD (right panel) in Example 2

7 Conclusions

In this paper, we propose fair procedures to conduct the cross-chain exchange of cryptocurrencies under different settings. We first study the optionality embedded in the classical atomic swap, which is one of the commonly used cross-chain exchange techniques, and propose a new cross-chain exchange procedure to account for this optionality. We develop a least-squares Monte Carlo algorithm to estimate the premium of the optionality, and prove the convergence of the algorithm. Then, we study the cross-chain exchange with margin trading. An adapted procedure for exchange and an adapted algorithm to estimate the premium are provided in the margin trading case. Numerical experiments demonstrate the good performance of the proposed algorithms under geometric Brownian motion, variance gamma process, and Merton's jump-diffusion process. In future research, we have several research directions. Firstly, we may consider the multilateral cross-chain exchange procedures, i.e., more than two parties in a cross-chain exchange, and study its pricing algorithm. Secondly, we may consider the implementation issues and provide the reference scripts or packages of our procedures in some cryptocurrency platforms. Thirdly, we may extend the application of atomic swap to other decentralized transactions of financial products, and consider a "mark-to-market" margin trading regime.

Acknowledgement

The authors are grateful to the associate editor and anonymous referees for their useful comments. The research reported in this paper is supported by National Natural Science Foundation of China [Grants 71801148, 71850013, 91846301, 71532004].

References

Andrikopoulos, A. (2010). On the valuation of American exchange options: An analytical approximation. *Applied Economics Letters*, 17(14):1429–1435.

Angraal, S., Krumholz, H. M., and Schulz, W. L. (2017). Blockchain technology: Applications in health care. *Circulation: Cardiovascular Quality and Outcomes*, 10(9):1–3.

Armada, M. R., Kryzanowski, L., and Pereira, P. J. (2007). A modified finite-lived American exchange option methodology applied to real options valuation. *Global Finance Journal*, 17(3):419–438.

Binance (2020). What is staking? Retrieved 2 August, 2020 from `https://academy.binance.com/blockchain/what-is-staking`.

Bitcoinwiki (2019a). Atomic cross-chain trading. Retrieved 16 March, 2020 from `https://en.bitcoin.it/wiki/Atomic_swap`.

Bitcoinwiki (2019b). Hashed timelock contracts. Retrieved 16 March, 2020 from `https://en.bitcoin.it/wiki/Hash_Time_Locked_Contracts`.

Bitcoinwiki (2019c). Sighash_flags. Retrieved 16 March, 2020 from `https://en.bitcoin.it/wiki/Contract#SIGHASH_flags`.

Carr, P. (1988). The valuation of sequential exchange opportunities. *The Journal of Finance*, 43(5):1235–1256.

Carr, P. and Madan, D. B. (2008). Representing the CGMY and meixner Lévy processes as time changed Brownian motions. *Journal of Computational Finance*, 2:27–47.

Cheang, G. and Chiarella, C. (2011). Exchange options under jump-diffusion dynamics. *Applied Mathematical Finance*, 18(3):245–276.

Chen, S., Chen, C. Y. H., Lee, T. M., and Ong, B. (2017). Econometric analysis of a cryptocurrency index for portfolio investment. In Chuen, D. L. K. and Deng, R., editors, *Handbook of Digital Finance and Financial Inclusion: Cryptocurrency, FinTech, InsurTech, and Regulation*, pages 175–206. Academic Press, Elsevier, London, UK.

Clément, E., Lamberton, D., and Protter, P. (2002). An analysis of the Longstaff-Schwartz algorithm for American option pricing. *Finance and Stochastics*, 6(4):449–471.

Cosmos (2020). Cosmos. Retrieved 16 March, 2020 from `https://cosmos.network`.

Fu, M. C. (2007). Variance-gamma and Monte Carlo. In Fu, M. C., Jarrow, R. A., Yen, J. Y., and Elliott, R. J., editors, *Advances in Mathematical Finance*, pages 21–34. Birkhäuser, Boston, MA.

Gerhold, S. (2011). The Longstaff-Schwartz algorithm for Lévy models: Results on fast and slow convergence. *The Annals of Applied Probability*, 21(2):589–608.

Glasserman, P. (2013). *Monte Carlo Methods in Financial Engineering*. Springer, New York, NY.

Han, R., Lin, H., and Yu, J. (2019). On the optionality and fairness of atomic swaps. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies October*, pages 62–75, New York, NY. ACM.

Herlihy, M. (2018). Atomic cross-chain swaps. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, pages 245–254, New York, NY. ACM.

Hong, L. J. and Jiang, G. (2019). Offline simulation online application: A new framework of simulation-based decision making. *Asia-Pacific Journal of Operational Research*, 36(6):No.1940015.

Hou, A. J., Wang, W., Chen, C. Y. H., and Härdle, W. K. (2020). Pricing cryptocurrency options. *Journal of Financial Econometrics*, 18(2):250–279.

Hull, J. (2012). *Options, Futures, and Other Derivatives*. Pearson Prentice Hall, London, UK, 9th edition.

Iansiti, M. and Lakhani, K. R. (2017). The truth about blockchain. *Harvard Business Review*, 95:118–127.

Jiang, G., Hong, L. J., and Nelson, B. L. (2020). Online risk monitoring using offline simulation. *INFORMS Journal on Computing*, 32(2):356–375.

Katsiampa, P. (2017). Volatility estimation for bitcoin: A comparison of GARCH models. *Economics Letters*, 158:3–6.

Kim, G. and Koo, E. (2016). Closed-form pricing formula for exchange option with credit risk. *Chaos, Solitons and Fractals*, 91:221–227.

Kshetri, N. (2018). 1 blockchain's roles in meeting key supply chain management objectives. *International Journal of Information Management*, 39:80–89.

Liu, J. A. (2018). Atomic swaptions: Cryptocurrency derivatives. Woring paper, University of California, Irvine, `https://arxiv.org/abs/1807.08644`.

Longstaff, F. A. and Schwartz, E. S. (2001). Valuing American options by simulation: A simple least-squares approach. *Review of Financial Studies*, 14(1):113–147.

Madan, D. B., Carr, P., and Chang, E. C. (1998). The variance gamma process and option pricing. *European Finance Review*, 2:79–105.

Margrabe, W. (1976). Alternative investment performance fee arrangements and implications for SEC regulatory policy: comment. *The Bell Journal of Economics*, 7(2):716–718.

Musiela, M. and Rutkowski, M. (2005). *Martingale Methods in Financial Modelling*. Springer-Verlag, Berlin, German.

Nolan, T. (2018). Atomic swaps using cut and choose. Retrieved 16 March, 2020 from `https://bitcointalk.org/index.php?topic=1364951`.

Noor, S., Yang, W., Guo, M., van Dam, K. H., and Wang, X. (2018). Energy demand side management within micro-grid networks enhanced by blockchain. *Applied Energy*, 228:1385–1398.

Perez, I. (2018). *Graphical User Interface for pricing Cryptocurrency Options under the Stochastic Volatility with Correlated Jumps Model*. Master thesis, School of Business and Economics, Humboldt Universiät zu Berlin, Berlin, Germany. `https://edoc.hu-berlin.de/bitstream/handle/18452/20281/master_perez_ivan.pdf?sequence=3&isAllowed=y`.

Polkadot (2020). Polkadot. Retrieved 16 March, 2020 from `https://polkadot.network`.

Ruppert, D. and Matteson, D. S. (2015). *Statistics and Data Analysis for Financial Engineering*. Springer, New York, NY.

Schoutens, W. (2003). *Lévy Processes in Finance: Pricing Financial Derivatives*. John Wiley Sons, Chichester, UK.

Stentoft, L. (2004). Convergence of the least squared Monte Carlo approach to American option valuation. *Management Science*, 50(9):1193–1203.

Thomas, S. and Schwartz, E. (2015). A protocol for interledger payments. Retrieved 16 March, 2020 from `http://blockchainlab.com/pdf/interledger.pdf`.

Vojtko, R. and Padyšák, M. (2020). What is the bitcoin's risk-free interest rate? Working paper, Retrieved 1 January, 2021 from `https://quantpedia.com/what-is-the-bitcoins-risk-free-interest-rate`.

Wanchain (2020). Wanchain. Retrieved 16 March, 2020 from `https://wanchain.org`.

Zanger, D. Z. (2013). Quantitative error estimates for a least-squares Monte Carlo algorithm for American option pricing. *Finance and Stochastics*, 17:503–534.

Table 1: Summary statistics of bitcoin and litecoin

Summary Statistics	BTC	LTC
Mean	0.0022	0.010
Median	0.0015	-0.0014
Standard Deviation	0.0419	0.0519
Standard Error	0.0018	0.0022
Excess Kurtosis	40.1325	16.4520
Skewness	-2.9529	-1.0099
Minimum	-0.5074	-0.4866
Maximum	0.1584	0.2609
Count	556	556
Correlation	0.8124	

Table 2: Model parameters of GBM and VG

	GBM		VG			MJD				
	μ	σ	θ	σ	ν	μ_D	σ_D	λ	μ_J	σ_J
BTC	0.0022	0.0419	0.0022	0.0370	1.4754	0.0018	0.0159	0.4578	0.0012	0.0515
LTC	0.0010	0.0519	0.0010	0.0478	1.0253	-0.0001	0.0278	0.2897	0.0052	0.0769

Table 3: KL divergence for bitcoin and litecoin

	Bitcoin	Litecoin
Normal	0.2031	0.1491
VG	0.0342	0.0283
MJD	0.0200	0.0201

Table 4: Estimated premiums without margin and RMSE in Example 1

N	d=2		d=3		d=4	
	p	RMSE	p	RMSE	p	RMSE
1000	0.0116	0.00049	0.0115	0.00049	0.0113	0.00053
2000	0.0116	0.00037	0.0118	0.00048	0.0116	0.00039
4000	0.0115	0.00026	0.0118	0.00030	0.0118	0.00030
8000	0.0115	0.00021	0.0119	0.00019	0.0119	0.00023
16000	0.0115	0.00015	0.0118	0.00015	0.0119	0.00014
32000	0.0114	0.00009	0.0118	0.00009	0.0118	0.00011
64000	0.0114	0.00007	0.0118	0.00008	0.0119	0.00008
128000	0.0115	0.00005	0.0118	0.00005	0.0119	0.00005

Table 5: Estimated premiums with margin and RMSE in Example 1

N	d=2		d=3		d=4	
	p	RMSE	p	RMSE	p	RMSE
1000	0.09998	10.9E-05	0.10002	9.31E-05	0.10000	9.22E-05
2000	0.10003	8.14E-05	0.10001	6.27E-05	0.10001	5.63E-05
4000	0.10002	4.80E-05	0.10000	3.95E-05	0.10000	4.04E-05
8000	0.10000	3.25E-05	0.10001	2.98E-05	0.10001	2.87E-05
16000	0.10002	2.02E-05	0.10001	2.01E-05	0.10001	1.99E-05
32000	0.10001	1.56E-05	0.10001	1.58E-05	0.10001	1.30E-05
64000	0.10001	1.29E-05	0.10001	1.21E-05	0.10001	1.17E-05
128000	0.10001	1.02E-05	0.10001	0.95E-05	0.10001	0.91E-05