Model Description

Notation

4.1 In this study, we proposed an agent-based model to simulate a blockchain system in which digital coins are traded by agents as currency. Our model consists of multiple *trader* agents, *miner* agents, and one *system* agent. Table 1 summarizes agent-related variables used in our model, where initial values of many exogenous variables are determined arbitrarily.

Agent	Variable	Type[a]	Remark	Value assignment
System	N_n	XV	Number of noisy traders	200
	N_h	XV	Number of herd traders	200
	N_g	XV	Number of game traders	200
	$\mathcal{T}$	XV	Set of all traders	
	M_n	XV	Number of noisy miners	100
	M_h	XV	Number of herd miners	100
	M_g	XV	Number of game miners	100
	$\mathcal{M}$	XV	Set of all miners	
	α	XV	Price sensitivity to the exceed demand	4000
	β	XV	Price sensitivity to the new coins	10
	γ	XV	The weight parameter of miner's stake	0.5
	W_t	XV	New coins added to miners at time t	10
	θ_t	XV	Maximum number of transactions in a block at time t	10
	P_t	NV	Coin price index at time t	$P_0 = 100$
	Δ_t	NV	Change of coin price index at time t	Equation (6)
	B_t	NV	Total demand of coin at time t	Equation (17)
	S_t	NV	Total supply of coin at time t	Equation (18)
	X_t	NV	Number of total transactions at time t	
	G_t	NV	Gini index of system at time t	
	R_t	NV	Average ratio of satisfied coin request at time t	
Trader i	Q_i	XV	Trader i's maximum percentage of tradable coins	0.5
	λ_i	XV	Trader i's sensitivity to price index change	0.05
	$\mathcal{H}_i$	XV	Set of trader i's neighbors in trade network	
	$\deg_i$	XV	Degree of trader i	Equation (10)
	m_i	XV	Number of previous minor decisions of trader i	2
	L_i	XV	Number of considering policies of trader i	100
	$A_{i,t}$	NV	Trader i's cash balance at time t	$A_{i,0} = 10000$
	$C_{i,t}$	NV	Trader i's coin balance at time t	$C_{i,0} = 100$
	$\mathrm{Pr}^B_{i,t}$, $\mathrm{Pr}^S_{i,t}$, $\mathrm{Pr}^H_{i,t}$	NV	The probabilities of trader i's decisions at time t	
	$q_{i,t}$	NV	Trader i's coin trade request at time t	Equation (9)
	$r_{i,t}$	NV	The percentage of satisfied coin request of trader i at time t	Equation (11)
Miner j	U_j	XV	Computation power of miner j	$U(10, 110)$
	$K_{j,t}$	NV	Stake of miner j at time t	Equation (14)
	$V_{j,t}$	NV	Stake-degree value of miner j at time t	Equation (16)

Table 1: Agent-related variables used in the model.

[a] DV: decision variable; NV: endogenous variable; XV: exogenous variable. The values of DVs and NVs are updated at each time step t; while those of XVs remains unchanged after initialization.

4.2 In the following we explain the behavior of each agent type in a static time step t.

Trader's behavior

4.3 Each trader agent (e.g., i) is initialized with two accounts (time step index $t = 0$): a cash account with balance 10000 (denoted by $A_{i,0} = 10000$) and a coin account with balance 100 ($C_{i,0} = 100$). At time t, each trader agent has to make his/her trade decision: to buy, sell or hold coins. In our model, the probabilities that trader i at time t decides to buy, to sell or to hold coins are denoted by $\mathrm{Pr}^B_{i,t}$, $\mathrm{Pr}^S_{i,t}$, $\mathrm{Pr}^H_{i,t}$, respectively. Note that the sum of these three probabilities is always 1.

Making trade decision

4.4 Similar to the classification of stock traders in the study by Liu & Serguieva (2018), there are three groups of traders in our model: noisy traders, herd traders, and game traders. Since it is difficult to obtain the empirical percentages of traders and miners in blockchain systems, we simply let three groups have same traders and miners, i.e., $N_n = N_h = N_g = 200$ and $M_n = M_h = M_g = 100$ as displayed in Table 1.

4.5 **Noisy traders**. The noisy traders make random decision on buying/selling/holding coins. Such agents represent the traders who use the blockchain system for payment, instead of investment. Therefore, their trade decisions are independent of coin price changes. Following Cocco & Marchesi (2016) who suggests that "buy and sell orders are always issued with the same probability (by random traders)", the decision probabilities of noisy trader i at time t are:

$$\mathsf{Pr}^B_{i,t} = 0.33, \tag{2}$$

$$\mathsf{Pr}^S_{i,t} = 0.33, \tag{3}$$

$$\mathsf{Pr}^H_{i,t} = 0.34. \tag{4}$$

4.6 **Herd traders.** Herd traders are very sensitive to the fluctuation of coin price index (denoted by P_t), because such agents represent the coin investors. Following Liu & Serguieva (2018), the probability that a herd trader i will hold his/her coins at time t is computed as follows:

$$\mathsf{Pr}^H_{i,t} = \frac{1}{1 + \lambda_i |\Delta_t|}, \tag{5}$$

where λ_i is trader i's sensitivity to price index change, and

$$\Delta_t = P_t - P_{t-1}. \tag{6}$$

Therefore, if the coin price index changes sharply (i.e., $|\Delta_t|$ is a large number), then $\mathsf{Pr}^H_{i,t}$ will be decreased, which means that most of herd traders will buy or sell their coins. The probability that a herd trader i will buy coins at time t is computed as follows:

$$\mathsf{Pr}^B_{i,t} = (1 - \mathsf{Pr}^H_{i,t}) \frac{e^{\Delta_t}}{e^{\Delta_t} + e^{-\Delta_t}}. \tag{7}$$

Therefore, when the coin price index grows, the herd trader i is more likely to buy coins. Consequently, the probability of selling coins is

$$\mathsf{Pr}^S_{i,t} = 1 - \mathsf{Pr}^H_{i,t} - \mathsf{Pr}^B_{i,t}. \tag{8}$$

4.7 **Game traders.** In financial markets, there are always a minority of traders who try to outwit others (Tanaka-Yamawaki & Tokuoka 2006). They buy coins while others are selling and sell coins while others are buying, for the purpose of chasing larger profits than behaving as herd traders. Therefore, game traders make decision based on the previous minor decisions released by the system agent.

4.8 To model the decision-making process of game traders, we assume that game trade i only considers previous $m_i = 2$ steps following Liu & Serguieva (2018). Since there are three possible decisions (Buy, Sell, Hold), the number of possible two-stage minor decisions is $3^2 = 9$, as listed in the first column of Table 2. The number of policies to deal with all possible scenarios is 3^9 = 19,683. For example, the first policy can be all "B", which means that no matter what the minor decisions in previous two time steps are, the game trader using Policy 1st will buy coins at current time step. If Policy 2nd is applied, the game trade will sell coins only if the previous minor decisions are "HH".

4.9 In our model, each game trade is initialized with $L_i = 100$ randomly-selected policies as his/her policy pool because it is infeasible for a trader to examine all 19,683 policies. Besides, a fitness is associate with each selected policy to indicate policy value. Similar to the selection mechanism of genetic algorithm, the game trade in each time step will select and apply a policy selected from the pool according its fitness. At time $t + 1$, the fitness of some policies will be increased if the decision given by these policies is the minor decision at time t, and will be decreased if not. By doing so, the game traders are "smarter" than other traders since the policies are applied dynamically and the trade decisions are made adaptively.

Table 2: Possible two-stage minor decisions and policies for game traders.

Possible two-stage minor decisions	Policy 1st	Policy 2nd	...	Policy 3^9-th
BB	B	B	...	H
BS	B	B	...	H
BH	B	B	...	H
SB	B	B	...	H
SS	B	B	...	H
SH	B	B	...	H
HB	B	B	...	H
HS	B	B	...	H
HH	B	S	...	H

Computing coin trade request

4.10 Once trader i has decided to buy, to sell or to hold coins, his/her coin trade request at time t is expressed as follows:

$$q_{i,t} = \begin{cases} 0, & \text{if hold;} \\ Q_i A_{i,t}/P_{i,t}, & \text{if buy;} \\ -Q_i C_{i,t}, & \text{otherwise.} \end{cases} \tag{9}$$

A positive coin request $q_{i,t}$ means that trader i wants to buy $q_{i,t}$ coins, which is the product of maximum purchasable coins $A_{i,t}/P_{i,t}$ and the percentage limit of coin trade $Q_i = 0.5 \in (0, 1)$. In contrast, a negative coin request means that trader i wants to sell $|q_{i,t}|$ coins, which depends on his/her coin balance $C_{i,t}$ and Q_i.

Trading with available neighbors

4.11 Due to the absence of central counter party in the blockchain system, trader i has to trade with neighbors in his/her trade network. Each trader agent (e.g., i) has a set of neighbors denoted by $\mathcal{H}_i$, the number of which is the degree of trader i, i.e.,

$$\deg_i = |\mathcal{H}_i|. \tag{10}$$

4.12 Because of agent heterogeneity and different trade decisions, trader i's coin request could not be satisfied. For example, a coin buyer seeks coin sellers from neighbors. He/She will continue seeking until his/her coin request has been fulfilled, or there is no more available seller in neighbors. If all the neighbors are also coin buyers, then trader i's satisfied ratio (denoted by $r_{i,t}$) is 0. Formally, the trader i's satisfied ratio is computed as follows:

$$r_{i,t} = \frac{\min\left\{|q_{i,t}|, \sum_{j\in\mathcal{H}_i} |q_{j,t}| \mathbb{1}(-q_{i,t}q_{j,t})\right\}}{|q_{i,t}|}, \tag{11}$$

where $\mathbb{1}(x)$ is the indicator function which yields 1 when $x > 0$, and yield 0 otherwise. If $-q_{i,t}q_{j,t} > 0$, the trader i and j have made different (and non-hold) trade decisions, and thus they can generate a transaction. Meanwhile, the coin requests of both buyers and sellers will be updated, as well as their cash and coin balances. All the transactions will be collected by the system agent introduced later in this section.

Miner's behavior

4.13 The miner agents are special traders. Hence, miners are also divided into three groups: noisy, herd, and game miners. They have the same endowment (initial cash, coins, neighbors) with traders, same trade behaviors, e.g., making trade decision, computing coin trade request, and trading with available neighbors.

4.14 However, miners have additional attributes (e.g., computation power, stake) because some of them will be selected by the system agent to create blocks and get a certain number of coins as reward. Let Pr(j) denote the probability that miner j will be selected, Assumption H_1 can be expressed as follows:

$$\text{Pr}(j) = \frac{e^{U_j}}{\sum_{l\in\mathcal{M}} e^{U_l}}, \tag{12}$$

where U_j is miner j's exogenous computation power, and $\mathcal{M}$ is the set of all miners.

4.15 Similarly, Assumption H_2 can be expressed as follows:

$$\text{Pr}(j) = \frac{e^{K_j}}{\sum_{l \in \mathcal{M}} e^{K_l}}, \tag{13}$$

where miner j' stake at time t is updated as follows:

$$K_{j,t+1} = K_{j,t} + C_{j,t}. \tag{14}$$

If the miner is selected and rewarded, the stake $K_{j,t+1} := 0$.

4.16 Finally, Assumption H_3 can be expressed as follows:

$$\text{Pr}(j) = \frac{e^{V_j}}{\sum_{l \in \mathcal{M}} e^{V_l}}, \tag{15}$$

where the time-dependent $V_{j,t}$ is defined as the stake-degree value of miner j at time t:

$$V_{j,t} = \gamma \times \frac{K_{j,t}}{\max\{K_{l,t}\}_{l \in \mathcal{M}}} + (1-\gamma) \times \frac{\deg_{j,t}}{\max\{\deg_{l,t}\}_{l \in \mathcal{M}}}, \tag{16}$$

where $\gamma = 0.5 \in (0, 1)$ is the weight parameter of miner's stake.

System's behavior

4.17 The system agent in our model performs many important tasks. First, it creates trader and miner agents and initializes their exogenous variables. Second, if the model meets the stop criterion, the simulation will be terminated and all important data will be saved for further analysis. Third, in each time step, the system agent needs to collect coin supply-demand information, such as the minor trade decision of agents for game traders, the number of total transactions X_t, and the total request of buying coins B_t:

$$B_t = \sum_{i \in \mathcal{T} \cup \mathcal{M}} q_{i,t} \mathbb{1}(q_{i,t}), \tag{17}$$

where $\mathcal{T}$ is the set of all traders. Similarly, the total request of selling coins S_t is computed as follows:

$$S_t = \sum_{i \in \mathcal{T} \cup \mathcal{M}} -q_{i,t} \mathbb{1}(-q_{i,t}). \tag{18}$$

4.18 Following the popular price update rule by LeBaron (2006) who suggested that the price of an artificial financial market is anchored to the previous price and price change is a proportion of demand/supply excess, the coin price index can be updated as follows:

$$P_{t+1} = P_t + \frac{B_t - S_t}{\alpha} - \frac{W_t}{\beta}, \tag{19}$$

where both α and β are positive parameters of price sensitivity. Consequently, the change of coin price index in our model depends on not only the exceed demand (i.e., $B_t - S_t$), but also the new W_t coins generated for incentivizing miners. The coin price index will increase if the exceed demand is large. However, the creation of new coins will suppress the coin price index. In our model, these new coins are equally given to selected miners. In particular, given X_t transactions, $\lceil X_t/\theta_t \rceil$ blocks will be created where θ_t is the maximum number of transactions in a block, and $\lceil x \rceil$ is the ceiling function which returns the smallest integer greater than $x > 0$. Therefore, for each block, the corresponding miner will receive $\frac{W_t}{\lceil X_t/\theta_t \rceil}$ coins.

Summary

4.19 Before we start the simulation experiments, the agents' behavior should be scheduled in a time step for implementation in the computer simulation programs. Figure 2 summarizes the sequence of events in our model

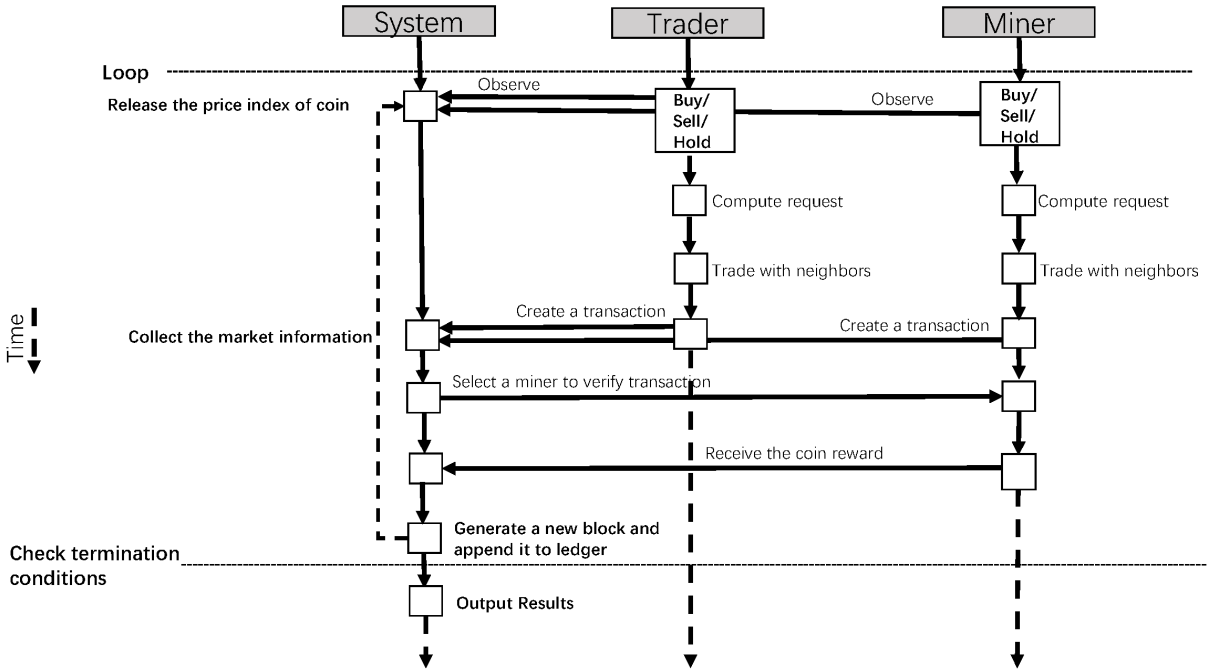

Figure 2: The time sequence diagram of the agent-based blockchain system. The gray rectangles represent involved objects such as agents and the model. White rectangles and horizontal solid lines demonstrate the events and interactions among objects, respectively.

in the form of a unified modeling language behavior diagram, in which parallel vertical lines indicate different processes or objects that live simultaneously, and horizontal arrows represent messages exchanged between them in the order in which they occur. All the components in Figure 2 have been discussed above.

Numerical Simulation

5.1 We first conduct nine experiments — (H_1, H_2, H_3) $\otimes$ (H_4, H_5, H_6) — with different consensus protocols and trade network topologies. Only three trade networks are generated, the node degree distributions of which are presented in Table 3. We develop the model using Python, and perform each experiment 100 times to ensure robust outputs against randomness in initializing the computation power, miner selection, policy selection, and so on. All the 100 independent tests of each experiment can be well compared and reproduced by assigning $\{0, 1, 2, ..., 99\}$ as random seeds, which means that the differences among experiments almost only depend on the configuration of its consensus protocol and trade network topology.

Table 3: The node degree distribution of three generated trade networks.

Trade network topology	Max degree	Min degree	Avg. degree	Median degree
Random	21	1	10.09	10
Small-world	17	6	10.09	10
Scale-free	656	1	4.27	2

5.2 The events presented in Figure 2 were carried out 1000 times for each test, i.e., each simulation stops at $t = 1000$. Therefore, the total computation load is: 9 experiments $\times$ 100 tests with different random seeds $\times$ 1000 time steps. During simulation, we collect necessary data to evaluate the performance of a blockchain, which is measured by three indicators:

- Price index of coin P_t, as updated by Equation (19).
- Request-satisfied ratio R_t is the division of total satisfied coin requests by total coin requests. The larger the R_t is, the higher the trade efficiency of blockchain system is.