Engineering Applications of Artificial Intelligence 130 (2024) 107797

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A crowdsourcing logistics solution based on digital twin and four-party evolutionary game

Lingjie Zhang [a], Xiaoding Wang [a], Hui Lin [a,*], Md. Jalil Piran [b,*]

[a] College of Computer and Cyber Security, Fujian Normal University, Engineering Research Center of Cyber Security and Education Informatization, Fujian Province University, Fuzhou, Fujian, 350117, China

[b] Department of Computer Science and Engineering, Sejong University, Seoul 05006, South Korea

ARTICLE INFO

Keywords:
Digital twin
Evolutionary game
Crowdsourcing logistics
Multi-agent
Reinforcement learning
Incentive mechanism
Blockchain

ABSTRACT

With the development of mobile Internet and the popularity of public online consumption, the scale expansion of the e-commerce industry has driven the continuous growth of logistics business, but the increase in order volume has brought pressure and challenges to the end of logistics and distribution, and the emergence of crowdsourcing logistics It provides a new path for alleviating the current logistics and transportation dilemma. The Digital Twin (DT) can virtualize and learn the data of the physical space, and introduce DT into the crowdsourcing logistics. It can iteratively update the crowdsourcing logistics participant strategy by constructing a virtual space, so as to change the corresponding strategy in time. Considering the situation of crowdsourcing logistics workers signing contracts with platforms and colluding with platforms, this paper constructs a four-party evolutionary game model of temporary workers, contract workers, blockchain-based crowdsourcing platforms and task requester, and analyzes the evolution using replication dynamics method stabilization strategy. In the virtual scene of DT, multi-agent reinforcement learning is used to predict the evolution result of the current strategy, and a reward and punishment strategy is given to prevent workers from free-riding and platform false reporting. The simulation results show that the analyzed evolutionary stability strategy can make the crowdsourcing logistics system run continuously and healthily, and the participants can adjust the strategy correctly according to the prediction results of virtual crowdsourcing logistics.

1. Introduction

With the rapid development of computer applications and Internet technology, a large amount of data is generated every second, and we have entered the era of big data. There have been many online data transaction systems in recent years (Jung et al., 2017). Crowdsourcing data trading is a new data trading paradigm that combines mobile crowdsensing. It uses crowds to collect data and solves the scarcity of sales data sources. Crowdsourcing is a new data acquisition mode that combines the idea of crowdsourcing and the perception capability of mobile devices, and it is a form of the Internet of Things (IoT) (Lu et al., 2021). The IoT will provide larger, more complex and more comprehensive perception services through crowdsourcing perception systems, affecting all aspects of life such as disaster monitoring, traffic management, public safety, logistics management, and social services (Chi et al., 2021; Li et al., 2016; Liu et al., 2020; Cai et al., 2021, 2018; Hong et al., 2021; Pu et al., 2021). In this context, some scholars propose crowdsourcing logistics to optimize logistics management. Crowdsourcing logistics is a new type of industry that combines Internet technology and traditional logistics operation models. After outsourcing companies know the user's business requirements, they will find the most suitable solution according to user needs. After finding the corresponding Internet agency, the outsourcing company will assign the corresponding logistics work to the corresponding operation department to serve users to meet their needs.

There are three main stakeholders in a crowdsourcing logistics system, namely task publishers, platforms, and crowdsourcing workers. First, the task publisher publishes the description of the task through the platform to recruit suitable crowdsourcing workers to complete the task. After receiving the task, the workers who are selected by the platform will be given the task. After receiving the task, the workers will go to the designated location to complete the task. Finally, the worker uploads the delivery report to the platform, and if the data meets the requirements through the platform review, the platform sends the delivery result to the requester. In the event that all three parties are honest, the posted task will be successfully completed through the above process.

* Corresponding authors.
E-mail addresses: q1261894969@gmail.com (L. Zhang), wangdin1982@fjnu.edu.cn (X. Wang), linhui@fjnu.edu.cn (H. Lin), piran@sejong.ac.kr (M.J. Piran).

https://doi.org/10.1016/j.engappai.2023.107797
Received 15 December 2022; Received in revised form 19 October 2023; Accepted 21 December 2023
Available online 28 December 2023

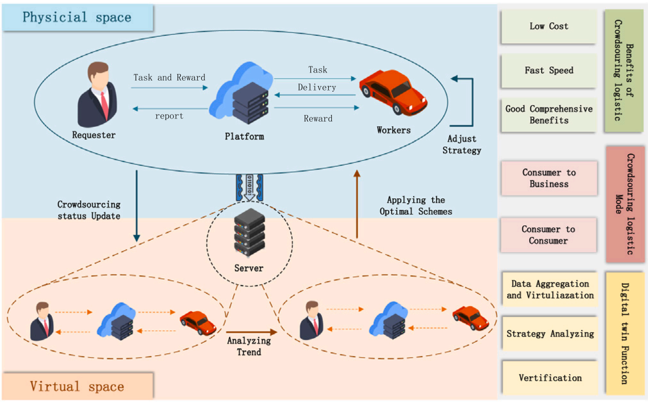


Fig. 1. Architecture of a crowdsourcing logistics systeml applications.

However, in reality, task requesters, platforms, and crowdsourcing workers, who are stakeholders of crowdsourcing systems, are all individuals with bounded rationality, so they all have selfish characteristics. Crowdsourcing workers, as the main performers of tasks in the crowdsourcing system, need to consume resources to complete specific sensing tasks, so it is necessary to carry out appropriate incentives. Due to their selfish nature, they always expect higher rewards with less effort, which leads to overtime of package delivery in crowdsourcing logistics systems, a phenomenon known as free-riding (Zhang and van der Schaar, 2011). The task requester needs to recruit crowdworkers through the platform to complete the task, and pay the crowdsourcing workers who complete the task as an incentive. However, due to the selfish nature of task requesters, they may falsely report the delivery result after receiving the package within the agreed time, thereby reducing payment and producing false reports (Swan, 2015). This will discourage crowdsourcing workers from participating in tasks and is not conducive to the sustainable development of the platform. As a third-party platform in the crowdsourcing system, the platform will issue tasks to requesters, accept delivery reports from crowdsourcing workers, and review and integrate crowdsourcing workers' reports. Platforms and crowdsourcing workers may collude, i.e. platforms lower the standards of monitoring, allowing for looser package delivery times and benefit from requesters. Such collusion will reduce the efficiency of the crowdsourcing logistics system and hinder the sustainable development of the crowdsourcing system.

Greaves describes a digital twin (DT) as an information mirror model (Xie et al., 2023), a digital replica of a physical entity. It enables seamless data transfer by connecting the physical and virtual worlds (Saddik, 2018), allowing virtual entities to coexist with physical entities. DT technology highlights two important characteristics. First, each definition emphasizes the connection between the physical model and the corresponding virtual model or virtual counterpart. Second, connections are established by using sensors to generate real-time data (Negria et al., 2017; Chhetri et al., 2004). As shown in Fig. 1, because of the real-time nature of DT, it can help crowdsourcing participants to understand the development results of the current strategy in time, so that participants can timely discover the risks that may occur in the system in the future, effectively guide them to choose to change the strategy, and promote the crowdsourcing system's healthy development.

The main contributions of this paper include four aspects:

(1) On the basis of the existing three-party evolutionary game (Li et al., 2022), in order to make the game model more consistent with the characteristics of the real crowdsourcing logistics system, we subdivides the crowdsourced workers and establishes a four-party evolutionary game model of temporary workers, contract workers, task requesters and blockchain-based platforms, and analyze their strategy evolution results.
(2) Taking into account the collusion between crowdsourcing workers and the platform and the situation that crowdsourcing workers will sign contracts with the platform, and using replication dynamics to analyze the stability of the strategic equilibrium point in MCS.
(3) A crowdsourcing logistics framework based on DT and evolutionary game is proposed, and blockchain performance is added to the game model to make the prediction results in the DT virtual space more realistic.
(4) The model is simulated in DT virtual space using multi-agent reinforcement learning, and the current strategy development trend of each participant in the crowdsourcing logistics system is predicted, and its effectiveness is verified.

2. Related work

In the past few years, DT have attracted the attention of scholars, who have applied DT to various scenarios for optimization. Zhao et al. (2023) proposed a hierarchical routing scheme in a software-defined vehicle network based on DT to overcome weaknesses in the vehicle network, simulation results showed that the model achieved significant improvements in performance. In order to overcome the difficulties of existing SDVN in building and applying new solutions, Zhao et al. (2020) introduced a new network architecture IDT-SDVN to take advantage of SDVN, and demonstrated the effectiveness of the proposed method through experiments. Zhao et al. (2022) proposed a DT-assisted storage strategy for satellite ground networks, obtained the optimal satellite through a genetic algorithm, and introduced a DT-assisted inter-satellite routing scheme to improve transmission quality.

As a disruptive technology, many scholars applied blockchain technology to scenarios such as the IoT and big data (Wang et al., 2020; Zhang et al., 2020; Wang et al., 2021; Zhang et al., 2021). Some scholars also use theoretical analysis to optimize data collection in IoT, like Sangoleye et al. (2021) use contract theory to motivate data collection of social IoT nodes and jointly optimize the interests of all participants. Recently, some scholars have combined traditional game theory with blockchain to solve problems in crowdsourcing scenarios. Huang et al. (2021) proposed a crowdsourcing data trading system based on Stackelberg game and blockchain, using blockchain as a data transaction middleman, using Stackelberg game to manage the selection of data providers, and using watermarking technology to protect data copyright. Hu et al. (2020) proposed a blockchain-based MCS framework, using blockchain technology to protect privacy, and proposed a three-stage Stackelbreg game-assisted incentive mechanism. Zhang et al. (2019) utilize a dedicated blockchain with a consortium chain to provide decentralized, real, and transparent services for vehicle group sensing, and use Stackelbreg game to solve the scheduling problem of sensing tasks. In order to prevent transaction data in crowdsourcing scenarios, Zhu et al. (2020) designed a blockchain consensus algorithm and designed an incentive mechanism based on game theory to motivate candidate nodes, and gave honest verification results. Feng et al. (2018) used blockchain to build a decentralized platform in the wireless IoT crowdsourcing system, and designed a non-cooperative game model to analyze the competitive situation between sensors. In the drone-assisted mobile crowdsensing scenario, Xie et al. (2021) proposed a new crowdsensing framework based on blockchain, and used Stackelberg game to encourage drones to participate in block creation and provide high-quality services. Different from the above literature, this paper uses a four-party evolutionary game to prevent free-riding and false reporting among crowdsourcing logistics participants, and adds DT to analyze the evolution results of the participants' strategies in real time.

In recent years, there have been many studies using evolutionary game theory to optimize crowdsourcing systems, which mainly provide solutions to problems such as sensing cost, data quality, optimal price determination and incentives (Dasari et al., 2020). For example, Wang et al. (2018) proposed an evolutionary game model to predict the evolutionary trend of mobile crowdsourcing systems and used K-anonymity to protect the information of crowdsourcing workers. Chi et al. (2021) proposed an incentive mechanism based on multi-strategy repeated games to guide workers' strategy choices, and used evolutionary game theory and the Wright-Fisher model to analyze the participants' strategies. Shao et al. (2019) proposed an evolutionary game model and income selection method based on non-cooperative evolutionary game theory to solve the problem of evolutionary stable equilibrium. Li et al. (2022) modeled a three-party evolutionary game model among task issuers, platforms and crowds, and provided strategies to avoid free riding and false reports by analyzing the stability of evolutionary game strategies. On this basis, this paper divides the crowdsourcing workers into temporary workers and contract workers, uses a blockchain-based platform, and considers the situation of crowdsourcing workers colluding with the platform to build a four-party evolutionary game model.

3. System model

Evolutionary games break the assumption of complete rationality in traditional games, which is similar to the concept of survival of the fittest in biology. Since crowdsourcing system participants cannot be perfectly rational, it is more reasonable to use evolutionary games to model crowdsourcing participants. In this section, in order to simulate the decision-making evolution process of the participants of the crowdsourcing logistics system in the virtual space of DT, we subdivide the crowdsourcing logistics workers, and construct the relationship between temporary workers, contract workers, platforms and task requesters. The four-party evolutionary game model maps each real-world crowdsourcing participant to the players participating in the evolutionary game in the DT virtual space.

3.1. Model building

Let W_l be the set of temporary workers, $W_l = \{w_{l1}, w_{l2}, \dots, w_{li}, \dots, w_{ln}\}$, W_h be the set of contract workers, $W_h = \{w_{h1}, w_{h2}, \dots, w_{hi}, \dots, w_{hn}\}$, W is the set of all crowdsourcing workers, $W = W_l \cap W_h$. Define $\phi = \{\varphi_1, \varphi_2, \dots, \varphi_k, \dots, \varphi_m\}$ and $Q = \{q_1, q_2, \dots, q_j, \dots, q_t\}$ as the set of tasks and task publishers. After completing task φ_k, the crowdsourcing worker w_{li} or w_{hi} delivery report will be reviewed by the platform and uploaded to the task publisher q_j. If the crowdworker pays more to deliver the item within the stipulated time, it will pass the platform's review, and if it cannot be delivered, the crowdworker may collude with the platform. Both the crowdsourcing worker and the platform can be rewarded if the crowdsourcing worker and the platform jointly fake the logistics of the package to the requester. This phenomenon is very common in reality, and it is also in line with the selfish characteristics of the players. The disadvantage of collusion is that it will bring reputation damage.

To use evolutionary games to simulate players' strategy choices, we assume that temporary workers, contract workers, platforms, and task issuers have two strategies: honest and fraud, and the probabilities of players choosing the honest strategy are x, y, z, and r, respectively. Obviously, the probabilities of choosing a fraudulent strategy are $1-x$, $1-y$, $1-z$, and $1-r$. x, y, z, $r \in (0,1)$, respectively. For temporary and contract workers, the integrity strategy is to deliver the package within the stipulated time, while the fraud strategy is to delay delivery (the free-rider phenomenon). The honest strategy of the task requester is to give real rewards, while the non-fraud strategy is to give low rewards or even no rewards (false reporting). The platform's integrity strategy is to strictly review the logistics process of the package, which means that there is no collusion between the crowdsourcing workers and the platform, and the fraud strategy is to collude with the crowdsourcing workers to maximize their own interests.

3.2. Payoff matrix

In crowdsourcing logistics, task requesters, platforms, and crowdsourcing workers participate in logistics tasks to earn revenue. In this paper, it is assumed that u_{li} and u_{hi} are the earnings of the crowdworker w_{li} or w_{hi}, respectively, O_j^i is the earnings of the task requester q_j after the crowdsourcing workers w_{li} and w_{hi} complete the logistics task φ_k, and P_i is the payment made by the requester to the crowdworker and the platform delivery fee. We use λ to represent the proportion of the distribution task remuneration received by crowdsourcing workers, $0 \leq \lambda \leq 1$, and u to represent the remuneration ratio of temporary workers, $0 \leq u \leq 0.5$, then $u\lambda P_i$ represents the remuneration received by temporary workers after completing the logistics task, $1-u$ represents the remuneration ratio of contract workers, $(1-u)\lambda P_i$ is the remuneration received by contract workers after completing the logistics task, and $1-\lambda$ represents the remuneration ratio received by the platform, $(1-\lambda)P_i$ is remuneration received by the platform after completing the logistics task. M_p represents the monitoring cost of the platform, and C_p represents the performance of the platform running the blockchain. C_{hi} is the cost for the crowdsourcing workers to deliver the package in time, and C_{li} is the cost for the crowdsourcing workers to deliver the package late, $C_{hi} > C_{li}$. As bounded rational individuals with selfish characteristics, crowdsourcing workers tend to choose deceptive strategies to maximize their benefits, namely the free-rider phenomenon. To avoid this, rewards and penalties are added to the game model. If the crowdsourcing workers deliver the logistics in time, they will receive the corresponding reputation reward R. If the delivery is not completed within the agreed time, there are two situations:

(1) When collusion occurs, crowdsourcing workers need to pay B_t to the platform to seek collusion, which means crowdfunding incurs collusion cost B_t. At this time, the platform will generate a collusion cost C_{pi} for falsifying logistics reports, so $C_{hi} - C_{li} > B_t + tR > C_{pi}$ can be obtained. To encourage participants to choose a credible strategy, we will give the crowdsourcing workers and platform reputation penalty S if collusion occurs. Let t be the reward-penalty ratio of crowdsourcing workers, then the platform's reward-penalty ratio is $1-t$. Let v be the reward-penalty ratio of temporary workers among crowdsourcing workers, then the reward-penalty ratio of contract workers is $1-v$.

(2) If the crowdsourcing worker does not collude with the platform or the platform rejects the crowdsourcing worker's colluding request, the task requester knows the correct logistics information, the revenue of both the task requester and the platform is 0, while the revenue of the crowdsourcing worker is $-C_{li}$. When crowdsourced workers complete logistics tasks and submit delivery reports, requesters may choose untrusted strategies to maximize their benefits, leading to false reporting problems. To avoid this, after the requester chooses the untrusted policy, the requester is penalized S_q. Therefore, we give the corresponding return matrix, as shown in Table 1, and the definitions of parameters in the model are given in Table 2.

4. Evolutionary game stability strategy analysis

Let U_x be the expected revenue of temporary workers for timely delivery, and U_{1-x} be the expected revenue of delayed delivery. Then, the expected return of temporary workers and the replication dynamics equation of the strategy are as follows:

$$\begin{cases} U_x = [(1-y)z(1-u)+u]r\lambda P_i + (z-yz-1)C_{hi} + vtR, \\ U_{1-x} = (1-z)ru\lambda P_i - (1-z)vtS - (1-z)B_t - C_{li}, \end{cases} \tag{1}$$

$$\begin{aligned} F(x) =dx/dt = x(1-x)\{&[u+(1-y)(1-u)]zr\lambda P_i + vtR \\ &- (1+z-yz)C_{hi} + (1-z)(vtS+B_t) + C_{li},\} \end{aligned} \tag{2}$$

Table 1
Payoff matrix.

	Temporary worker		Requester			
			Honest r		Fraud $1-r$	
			Honest y	Fraud $1-y$	Honest y	Fraud $1-y$
Platform	Honest z	Honest x	$u\lambda P_i + \upsilon tR - C_{hi}$ $(1-u)\lambda P_i + (1-\upsilon)tR - C_{hi}$ $(1-\lambda)P_i - M_p - C_p + (1-t)R$ $O_j^i - P_i + R_q$	$\lambda P_i - 2C_{hi} + \upsilon tR$ $-C_{li}$ $(1-\lambda)P_i - M_p - C_p + (1-t)R$ $O_j^i - P_i + R_q$	$\upsilon tR - C_{hi}$ $(1-\upsilon)tR - C_{hi}$ $-M_p - C_p + (1-t)R$ $O_j^i - S_q$	$\upsilon tR - 2C_{hi}$ $-C_{li}$ $-M_p - C_p + (1-t)R$ $O_j^i - S_q$
		Fraud $1-x$	$-C_{li}$ $\lambda P_i + (1-\upsilon)tR - 2C_{hi}$ $(1-\lambda)P_i - M_p - C_p + (1-t)R$ $O_j^i - P_i + R_q$	$-C_{li}$ $-C_{li}$ 0 0	$-C_{li}$ $(1-\upsilon)tR - 2C_{hi}$ $-M_p - C_p + (1-t)R$ $O_j^i - S_q$	$-C_{li}$ $-C_{li}$ 0 0
	Fraud $1-z$	Honest x	$u\lambda P_i + \upsilon tR - C_{hi}$ $(1-u)\lambda P_i + (1-\upsilon)tR - C_{hi}$ $(1-\lambda)P_i - C_p - (1-t)S$ $O_j^i - P_i + R_q$	$u\lambda P_i + \upsilon tR - C_{hi}$ $(1-u)\lambda P_i - (1-\upsilon)tS - C_{li} - B_t$ $(1-\lambda)P_i - (1-t)S - C_{pi} - C_p + B_t$ $O_j^i - P_i + R_q - A_g$	$\upsilon tR - C_{hi}$ $(1-\upsilon)tR - C_{hi}$ $-(1-t)S - C_p$ $O_j^i - S_q$	$\upsilon tR - C_{hi}$ $-(1-\upsilon)tS - C_{li} - B_t$ $-(1-t)S - C_{pi} - C_p + B_t$ $O_j^i - S_q - A_g$
		Fraud $1-x$	$u\lambda P_i - \upsilon tS - C_{li} - B_t$ $(1-u)\lambda P_i + (1-\upsilon)tR - C_{hi}$ $(1-\lambda)P_i - (1-t)S - C_{pi} - C_p + B_t$ $O_j^i - P_i + R_q - A_g$	$u\lambda P_i - \upsilon tS - C_{li} - B_t$ $(1-u)\lambda P_i - (1-\upsilon)tS - C_{li} - B_t$ $(1-\lambda)P_i - (1-t)S - 2C_{pi} - C_p + 2B_t$ $O_j^i - P_i + R_q - 2A_g$	$-\upsilon tS - C_{li} - B_t$ $(1-\upsilon)tR - C_{hi}$ $-(1-t)S - C_{pi} - C_p + B_t$ $O_j^i - S_q - A_g$	$-\upsilon tS - C_{li} - B_t$ $(1-\upsilon)tS - C_{li} - B_t$ $-(1-\upsilon)tS - 2C_{pi} - C_p + 2B_t$ $O_j^i - S_q - 2A_g$

Table 2
Definition of model parameters.

Parameter	Description
P_i	Reward of requester to crowdsourcing workers and platforms
C_{hi}	Cost of crowdsourcing workers to transport timely
C_{li}	Cost of crowdsourcing workers to delay transport
M_p	Cost of platform supervision
C_p	Cost of running blockchain (blockchain performance)
B_t	Cost of crowdsourcing workers colluding with platform
R	Rewards for crowdsourcing workers and platform
S	Penalties for crowdsourcing workers and platform
C_{pi}	Cost of platform colluding with crowdsourcing worker
O_j^i	Benefits to the requester after the transport is completed
A_g	Losses caused by requester after logistical delays
R_q	Rewards given to requester
S_q	Penalty given to requester
λ	Proportion of the delivery task remuneration received by crowdsourcing workers
u	Proportion of the delivery task remuneration received by temporary workers
t	Reward-penalty ratio of crowdsourcing workers
υ	Reward-penalty ratio of temporary workers

Similarly, let U_y be the expected benefit of timely delivery by contract workers, and U_{1-y} be the expected benefit of delayed delivery. Then the expected return of contract workers and the replication dynamic equation of the strategy are as follows:

$$\begin{cases} U_y = & [(1-x)zu + (1+u)]r\lambda P_i + (z - xz - 1)C_{hi} \\ & + (1-\upsilon)tR, \\ U_{1-y} = & (1-z)r(1-u)\lambda P_i - (1-z)(1-\upsilon)tS \\ & - (1-z)B_t - C_{li}, \end{cases} \tag{3}$$

$$\begin{aligned} F(y) = & dy/dt = y(1-y)\{[(1-u) + (1-y)u]zr\lambda P_i \\ & + (1-\upsilon)tR - (1 + z - xz)C_{hi} \\ & + (1-z)[(1-\upsilon)tS + B_t] + C_{li},\} \end{aligned} \tag{4}$$

According to the payoff matrix, the expected payoffs for the platform to choose the honest strategy and the fraudulent strategy and the replication dynamic equation can be expressed as the following equation:

$$\begin{cases} U_z = & (xr + yr - xyr)(1-\lambda)P_i \\ & + (x + y - xy)[-M_p - C_p + (1-t)R], \\ U_{1-z} = & R(1-\lambda)P_i - (1-t)S \\ & + (2 - x - y)(B_t - C_{pi}) - C_p, \end{cases} \tag{5}$$

$$\begin{aligned} F(z) = & dz/dt = z(1-z)\{(x + y - xy - 1)r(1-\lambda)P_i \\ & - (2 - x - y)(B_t - C_{pi}) \\ & + (x + y - xy)[-M_p + (1-t)R] \\ & - (x + y - xy - 1)C_p + (1-t)S\}, \end{aligned} \tag{6}$$

Let U_r and U_{1-r} denote the expected return of the task requester choosing the honest strategy and the fraud strategy, respectively. Similar to before, the expected revenue of the task requester and the replication dynamic equation can be obtained separately:

$$\begin{cases} U_r = & (xz - xyz + yz - z + 1)(O_j^i - P_i + R_q) \\ & - (2 - x - y)(1-z)A_g, \\ U_{1-r} = & (xz - xyz + yz - z + 1)(O_j^i - S_q) \\ & - (2 - x - y)(1-z)A_g, \end{cases} \tag{7}$$

$$\begin{aligned} F(r) = & dr/dt \\ = & r(1-r)(xz - xyz + yz - z + 1)(R_q + S_q - P_i), \end{aligned} \tag{8}$$

According to the replication dynamic equation of each game subject, the Jacobian matrix of the replication dynamic system is obtained as

$$J = \begin{pmatrix} \partial F(x)/\partial x & \partial F(x)/\partial y & \partial F(x)/\partial z & \partial F(x)/\partial r \\ \partial F(y)/\partial x & \partial F(y)/\partial y & \partial F(y)/\partial z & \partial F(y)/\partial r \\ \partial F(z)/\partial x & \partial F(z)/\partial y & \partial F(z)/\partial z & \partial F(z)/\partial r \\ \partial F(r)/\partial x & \partial F(r)/\partial y & \partial F(r)/\partial z & \partial F(r)/\partial r \end{pmatrix}. \tag{9}$$

Table 3
Eigenvalues of Jacobian matrix.

Equilibrium point	Eigenvalue $\lambda_1, \lambda_2, \lambda_3, \lambda_4$
$E(1,1,1,1)$	$C_{hi} - C_{li} - u\lambda P_i - vtR, C_{hi} - C_{li} - (1-u)\lambda P_i - (1-v)tR, M_p - (1-t)(S+R), P_i - R_q - S_q$
$E(1,1,0,1)$	$C_{hi} - C_{li} - vt(R+S) - B_t, C_{hi} - C_{li} - (1-v)t(R+S) - B_t, (1-t)(S+R) - M_p, P_i - R_q - S_q$
$E(0,1,1,1)$	$u\lambda P_i + vtR - C_{hi} + C_{li}, 2C_{hi} - C_{li} - \lambda P_i - (1-v)tR, M_p - (1-t)(S+R) + B_t - C_{pi}, P_i - R_q - S_q$

Let $F(x) = 0$, $F(y) = 0$, $F(z) = 0$, $F(r) = 0$, we can get that the equation has 2^4 or 16 equilibrium solutions of pure strategies, which are (0,0,0,0), (0,0,0,1), (0,0,1,0), (0,0,1,1), (0,1,0,0), (0,1,0,1), (0,1,1,0), (0,1,1,1), (1,0,0,0), (1,0,0,1), (1,0,1,0), (1,1,0,0), (1,0,1,1), (1,1,0,1), (1,1,1,0), (1,1,1,1). The equilibrium point is brought into the Jacobian matrix, the eigenvalues of each equilibrium point are obtained in turn, and the positive and negative values are judged, and then the stability and stability conditions of the equilibrium point are judged. This paper mainly analyzes the stability of the equilibrium point under the condition that the crowdsourcing logistics system can develop healthily and continuously, namely (1,1,1,1), (1,1,0,1), (0,1,1,1), the corresponding eigenvalues can be obtained in Table 3. To make them stable points, their eigenvalues must all be less than 0, that is, they must satisfy $P_i - R_q - S_q < 0$, which means that the combination of the task requester's reward and punishment is greater than the perceived payment to the crowdsourcing logistics system, and the remaining conditions discussed separately below:

(1) The stable point (1,1,1,1) indicates that temporary workers, contract workers, platforms and task issuers all adopt honest strategies, and the stable condition is $C_{hi} - C_{li} - u\lambda P_i - vtR < 0$, $C_{hi} - C_{li} - (1-u)\lambda P_i - (1-v)tR < 0$, $M_p - (1-t)(S+R) < 0$. At this time, the cost saved by crowdsourcing workers by choosing to delay delivery is less than the sum of the reward and punishment obtained from the task requester, and the monitoring cost of the platform is greater than the sum of the reward and punishment obtained. This situation achieves the purpose of incentivizing all participants to choose a credible strategy, and the crowdsourcing system can continue to operate stably.

(2) The stable point (1,1,0,1) indicates that temporary workers, contract workers, and task issuers all adopt an honest strategy, while the platform adopts a fraud strategy. The stable condition is $C_{hi} - C_{li} - vt(R+S) - B_t < 0$, $C_{hi} - C_{li} - (1-v)t(R+S) - B_t < 0$, $(1-t)(S+R) - M_p < 0$. At this time, the cost saved by crowdsourcing workers by choosing to delay delivery is less than the sum of the rewards and penalties they get and the cost of collusion with the platform, and the monitoring cost of the platform is less than the sum of the rewards and penalties they get. In this case, the crowdworkers always complete the delivery within the stipulated time, the platform chooses not to monitor the data to save costs, and the package of the task requester can also be delivered on time.

(3) The stable point (0,1,1,1) indicates that contract workers, platforms, and task issuers all adopt honest strategies, while temporary workers adopt fraudulent strategies. The stable condition is $u\lambda P_i + vtR - C_{hi} + C_{li} < 0$, $2C_{hi} - C_{li} - (1-u)\lambda P_i - (1-v)tR < 0$, $M_p - (1-t)(S+R) + B_t - C_{pi} < 0$. At this time, the cost saved by the temporary worker overtime delivery is greater than the sum of the reward and the reward obtained from the task requester, and the cost saved by the contract worker by choosing to delay delivery is less than the sum of the reward and the cost of collusion with the platform, and the monitoring cost of the platform is less than the sum of the rewards and penalties earned, and the worker's collusion cost is less than the platform's collusion cost. In this case, although temporary workers choose fraudulent strategies, the platform can still choose contract workers to deliver packages to task issuers at the specified time, ensuring the healthy operation of the crowdsourcing logistics system.

5. Multi-agent reforcment learning

Reinforcement learning can handle incomplete information in dynamic environments and search for optimal policies for agents. As a learning algorithm that does not require its environment model and can be used online, the Q-learning algorithm is very suitable for incomplete information games. In this paper, multi-agent reinforcement learning is used in the DT virtual space to simulate the evolutionary trend of the current crowdsourcing logistics system participant strategies and to verify the analyzed evolutionary stable strategies. As shown in Fig. 2, the crowdsourcing logistics system in the real space transmits the current policy state to the DT virtual space, models the crowdsourcing logistics participants as players in an evolutionary game, and maps them to multi-agent reinforcement learning agents in the virtual space. When the agent chooses a strategy, if the environment gives positive feedback, the probability of the agent choosing the same strategy in the next round will increase; otherwise, it will decrease. Therefore, the decision-making agent will acquire knowledge, learn from the acquired knowledge and the feedback given by the environment, and choose a strategy, so that the strategy choice trend of the participants in the real space can be reflected in the virtual space. Finally, the predicted results are fed back to the real space, and suggestions are provided. Participants can make timely strategic adjustments to avoid losses and can also maintain the sustainable and healthy development of the interim reporting system. The following is the specific implementation of multi-agent reinforcement learning used in this paper.

5.1. Algorithm description

In this paper, we use the multi-agent reinforcement learning model proposed in bing Liu and Wang (2009) to predict the outcome of evolutionary games, this algorithm has been proven by the author to be successfully used in evolutionary games to find optimal strategies. For a four-party evolutionary game, suppose the strategy form is $\langle S, A^1, A^2, A^3, A^4, r^1, r^2, r^3, r^4, p\rangle$, where S is a set of states, A^1, A^2, A^3, and A^4 are the action space of player 1, player 2, player 3 and player 4 respectively, r^1, r^2, r^3 and r^4 are the reward functions of the players, which can be obtained from the game's payoff matrix, and p is the state transition probability. t is the number of repeated games, and the state at time t is denoted as s_t. Since the set of states in the model is the action composition of the game, the transitions of the states can be depicted by the player's action space. In state s, the player chooses his actions a^1, a^2, a^3 and a^4 to get rewards r^1, r^2, r^3 and r^4, respectively. After that, a new state s' is reached, and the transition probability is $p(s'|s, a^1, a^2, a^3, a^4)$. So far, the decision-making process of the evolutionary game is mapped to the reinforcement learning model, which can be used to simulate the dynamic decision-making in the evolutionary game.

According to the above model, the total expected rewards of players are as follows.

$$\begin{aligned} Q^1(s, a^1, a^2, a^3, a^4) &= r^1(s, a^1, a^2, a^3, a^4) \\ &+ \gamma \sum p(s'|s, a^1, a^2, a^3, a^4) Q^1(s, a^1, a^2, a^3, a^4), \end{aligned} \tag{10}$$

$$\begin{aligned} Q^2(s, a^1, a^2, a^3, a^4) &= r^2(s, a^1, a^2, a^3, a^4) \\ &+ \gamma \sum p(s'|s, a^1, a^2, a^3, a^4) Q^2(s, a^1, a^2, a^3, a^4), \end{aligned} \tag{11}$$

$$\begin{aligned} Q^3(s, a^1, a^2, a^3, a^4) &= r^3(s, a^1, a^2, a^3, a^4) \\ &+ \gamma \sum p(s'|s, a^1, a^2, a^3, a^4) Q^3(s, a^1, a^2, a^3, a^4), \end{aligned} \tag{12}$$

$$\begin{aligned} Q^4(s, a^1, a^2, a^3, a^4) &= r^4(s, a^1, a^2, a^3, a^4) \\ &+ \gamma \sum p(s'|s, a^1, a^2, a^3, a^4) Q^4(s, a^1, a^2, a^3, a^4), \end{aligned} \tag{13}$$

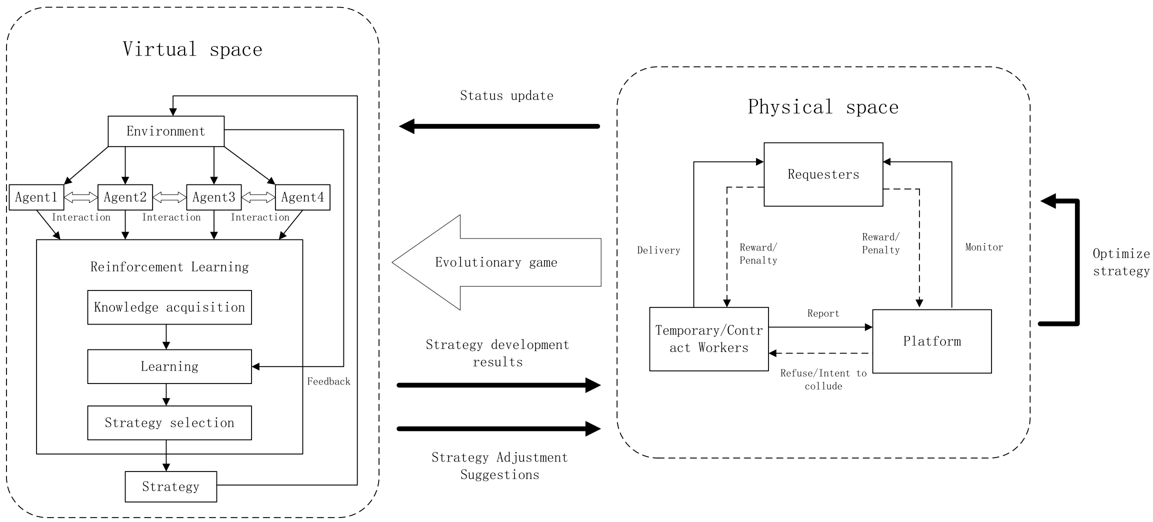


Fig. 2. Framwork of DT based crowdsoucing.

The player updates the Q value according to the following formula

$$Q^1_{t+1}(s,a^1,a^2,a^3,a^4) = (1-\alpha_t)Q^1_t(s,a^1,a^2,a^3,a^4) + \alpha_t[r^1_t + \gamma\delta^1(s')\delta^2(s')\delta^3(s')\delta^4(s')Q^1_t(s')], \tag{14}$$

$$Q^2_{t+1}(s,a^1,a^2,a^3,a^4) = (1-\alpha_t)Q^2_t(s,a^1,a^2,a^3,a^4) + \alpha_t[r^2_t + \gamma\delta^1(s')\delta^2(s')\delta^3(s')\delta^4(s')Q^2_t(s')], \tag{15}$$

$$Q^3_{t+1}(s,a^1,a^2,a^3,a^4) = (1-\alpha_t)Q^3_t(s,a^1,a^2,a^3,a^4) + \alpha_t[r^3_t + \gamma\delta^1(s')\delta^2(s')\delta^3(s')\delta^4(s')Q^3_t(s')], \tag{16}$$

$$Q^4_{t+1}(s,a^1,a^2,a^3,a^4) = (1-\alpha_t)Q^4_t(s,a^1,a^2,a^3,a^4) + \alpha_t[r^4_t + \gamma\delta^1(s')\delta^2(s')\delta^3(s')\delta^4(s')Q^4_t(s')], \tag{17}$$

Among them, $(\delta^1(s),\delta^2(s),\delta^3(s),\delta^4(s))$ and $(\delta^1(s'),\delta^2(s'),\delta^3(s'),\delta^4(s'))$ are the matrix games $(Q^1_t(s),Q^2_t(s),Q^3_t(s),Q^4_t(s))$ and $(Q^1_{t+1}(s'),Q^2_{t+1}(s'),Q^3_{t+1}(s'),Q^4_{t+1}(s'))$ mixed strategy Nash equilibrium respectively.

The process of the policy iteration algorithm is shown in Algorithm 1, and the process is as follows:

Step 1 Initialize $t=0$, $Q^1_t(s,a^1,a^2,a^3,a^4)=0$, $Q^2_t(s,a^1,a^2,a^3,a^4)=0$, $Q^3_t(s,a^1,a^2,a^3,a^4)=0$, $Q^4_t(s,a^1,a^2,a^3,a^4)=0$. $\forall s\in S$, $a^1\in A^1$, $a^2\in A^2$, $a^3\in A^3$, $a^4\in A^4$, and initialize the current state;

Step 2 Calculate the mixed strategy Nash equilibrium $(\delta^1(s),\delta^2(s),\delta^3(s),\delta^4(s)$ through matrix game $(Q^1_t(s),Q^2_t(s),Q^3_t(s),Q^4_t(s)))$, and select an action a^1 according to $\delta^1(s)$, observe the rewards r^1_t,r^2_t,r^3_t,r^4_t, the opponent's actions a^2,a^3,a^4 and the next state s';

Step 3 Calculate the mixed-strategy Nash equilibrium $(\delta^1(s'),\delta^2(s'),\delta^3(s'),\delta^4(s'))$, and update Q^1,Q^2,Q^3,Q^4;

Step 4 Go back to step 2, set $t=t+1$, repeat until all states have been searched.

5.2. Feasibility analysis

According to the above description, the learning agent needs to maintain n Q-functions, and each agent in the system has a Q-function. These Q-functions are maintained internally by the learning agent, assuming it can observe the behaviors and rewards of other agents.

The learning agent updates $(Q^1,\dots,Q^n)$, where each $Q^j, j=1,2,\dots,n$, for all s, $a^1,\dots,a^n$ consists of $Q^j(s,a^1,\dots,a^n)$. Let $|S|$ be the number of states, and let $|A^i|$ be the size of the action space A^i of agent i. Suppose $|A^1|=\cdots=|A^n|=|A|$, the total number of entries in Q^k is $|S|\cdot|A|^n$. Since our learning agent must maintain n Q-tables, the total space requirement is $n|S|\cdot|A|^n$.

The learning algorithm that we used, in terms of space complexity, is linear in the number of states, polynomial in the number of actions, and exponential in the number of agents. The running time of the algorithm mainly depends on the calculation of the mixed strategy Nash equilibrium used in the Q-function update. For n-player games, approximation methods are usually used. In our proposed game model, the learning time for 5000 rounds is less than 3 s, which shows that it can be implemented in a near real-time manner.

Algorithm 1 Policy iteration

Initialize:
1: Let $t=0$, get the initial state s_0.
2: Let the learning agent be indexed by i, $i=1,2,3,4$.
3: For all $s\in S$ and $a^i\in A^i$, let $Q^i_t(s,a^1,a^2,a^3,a^4)=0$.
Ensure:
4: **repeat**
5: Choose action a^i_t.
6: Compute mixed strategy Nash equilibrium $(\delta^1(s),\delta^2(s),\delta^3(s),\delta^4(s)$ through matrix game $(Q^1_t(s),Q^2_t(s),Q^3_t(s),Q^4_t(s)))$.
7: Observe r^1_t,r^2_t,r^3_t,r^4_t, a^1_t,a^2_t,a^3_t,a^4_t and $s_{t+1}=s'$.
8: Compute $(\delta^1(s'),\delta^2(s'),\delta^3(s'),\delta^4(s'))$ through $(Q^1_{t+1}(s),Q^2_{t+1}(s),Q^3_{t+1}(s),Q^4_{t+1}(s)))$.
9: Update Q^1,Q^2,Q^3,Q^4.
10: Let $t=t+1$.
11: **until** all states have been searched.

Cost considerations are inevitable when integrating digital twins into model-based systems engineering processes (West and Blackburn, 2017). The cost of a digital twin is the number of components in the system, the interfaces and dependencies between components, the complexity of the algorithms used to implement specific functions, and the functions required to build the digital twin. While digital twins require a larger upfront investment, the addition of digital twins is expected to provide significant return on investment over the life of the system, and building reusable digital twins can further reduce costs. This means that although our method is more costly when deployed, it can amortize the cost through continuous operation and reduce potential error correction costs in the future. It can also reduce deployment costs by using a common digital twin framework.

6. Experimental verification

We apply a multi-agent reinforcement learning model to an evolutionary game where each player has two strategies: honesty (H) and fraud (F).

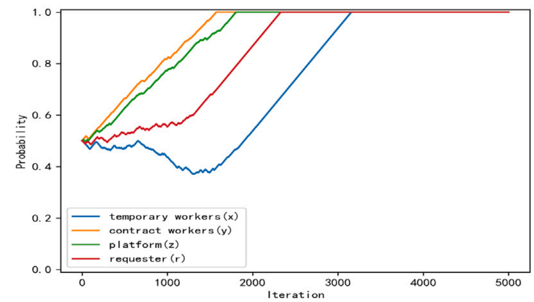
In evolutionary games, players seek optimal strategies through reinforcement learning models, and the states in the reinforcement learning models are regarded as the players' strategy combinations. In this model, the state space is $S = \{\langle HHHH\rangle, \langle HHHF\rangle, \langle HHFH\rangle, \langle HFHH\rangle, \langle HFFH\rangle, \langle HFHF\rangle, \langle HHFF\rangle, \langle HFFF\rangle, \langle FHHH\rangle, \langle FHHF\rangle, \langle FHFH\rangle, \langle FFHH\rangle, \langle FFFH\rangle, \langle FFHF\rangle, \langle FHFF\rangle, \langle FFFF\rangle\}$, the player's strategy set is $A = \{a^1, a^2, a^3, a^4\} = \{H, F\}$, the instantaneous reward of can be determined by the payoff matrix. Therefore, the player's total expected payoff can be denoted by Q. Through the Q-value table, the agent can choose the optimal strategy according to the corresponding state to maximize the expected return. In the experiments, we set the multi-agent reinforcement learning model parameters as $\gamma = 0.7$, $\alpha_t = 0.2$ to simulate the bounded rationality of groups in evolutionary games. We also compare the simulation results of strategy selection based on multi-agent reinforcement learning with the numerical simulation results based on matlab.

We introduce the blockchain performance into the proposed four-party game model, and we use Geth to test the performance of the blockchain when it records transaction information and logistics information under the simulated crowdsourcing logistics scenario of Ethereum. In this article, Geth uses the default configuration and connects to the public test network to simulate real-life conditions. Substitute the obtained real data into the C_p parameters in the model to verify the stability of the evolution equilibrium point with the addition of real blockchain performance data.

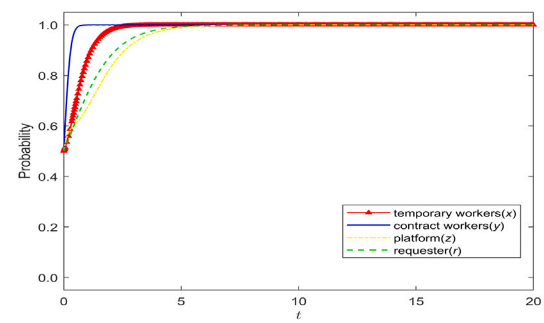
According to the local stability of the equilibrium point in Table 3, experiments are first performed to verify the existence of the stable point (1,1,1,1). We set the parameters in the four-party evolutionary game $\lambda = 0.5$, $t = 0.5$, $u = 0.3$, $v = 0.3$, $P_i = 40$, $C_{hi} = 12$, $R = 15$, $C_{li} = 15$, $S = 15$, $B_t = 4$, $M_p = 14$, $C_{pi} = 2$, $C_p = 0$, $O_j^i = 50$, $A_g = 5$, $S_q = 26$, $R_q = 15$, assuming that temporary workers, contract workers, platforms and task publishers all have an initial selection rate of integrity of 0.5. The simulation results are shown in Fig. 3(a), which are consistent with our inference results, that is, the policy combination of temporary workers, contract workers, platform and task requester is (Honest, Honest, Honest, Honest). As can be seen from Fig. 3(a) and (b) that although the intermediate process is somewhat different, the evolution results based on multi-agent reinforcement learning are consistent with the simulation results of matlab, which shows that the correct policy evolution results can be obtained through this multi-agent reinforcement learning model. Setting z to 0.8 indicates that the platform chooses an honest strategy, and uses Matlab to simulate the evolution process of different initial strategies of temporary workers, contract workers, and task publishers in three-dimensional space. The results are shown in Fig. 3(c). It can be seen that in the crowdsourcing participants, when there is no particular tendency to commit fraud at the beginning, the crowdsourced workers complete the logistics tasks in a timely manner, the platform monitors the logistics information, and the requester provides real rewards, which effectively avoids the problems of free riders and false reports.

Next, we verify the existence of the stable point (1,1,0,1). Reset $S = R = 12$, according to Table 3, the Jacobian eigenvalues corresponding to the equilibrium point (1,1,0,1) are all negative, and the stable equilibrium point of the system is (Honest, Honest, Fraud, Honest). The simulation results of multi-agent reinforcement learning are shown in Fig. 4(a). The results are consistent with those in Fig. 4(b) and are consistent with our inference results. Similarly, set $z = 0$ to indicate that the platform selects a fraud strategy, and simulate the evolution process of different initial strategies of the other three parties in three-dimensional space. The simulation results are shown in Fig. 4(c). In this case, crowdsouring workers always complete logistics tasks in time, and task requesters always give real logistics rewards, Therefore, platforms tend not to monitor to save costs.

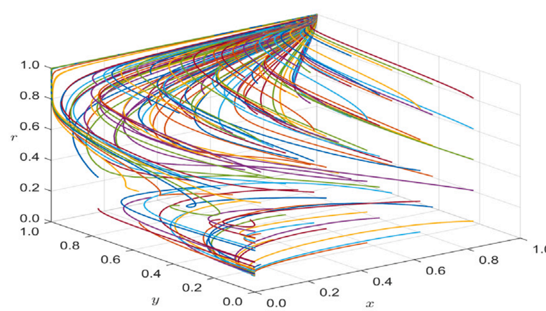
Finally, verify the existence of the stable point (0,1,1,1). According to Table 3, in order to make the equilibrium point (0,1,1,1) reach a stable state, we need to reset $u = v = 0.2$, $C_{hi} = 14$, $M_p = 5$, $S = 8$, at this



(a) Evolution results of multi-agent reinforcement learning.



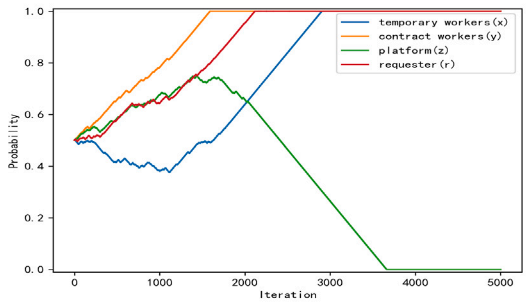
(b) Evolution results of matlab numerical simulation.



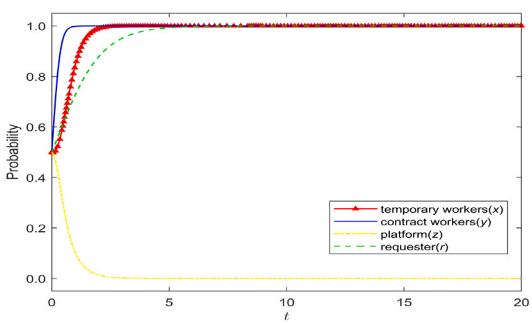
(c) Evolution results of the three parties x, y, and r under different initial values.

Fig. 3. Evolution of four-party under (1,1,1,1).

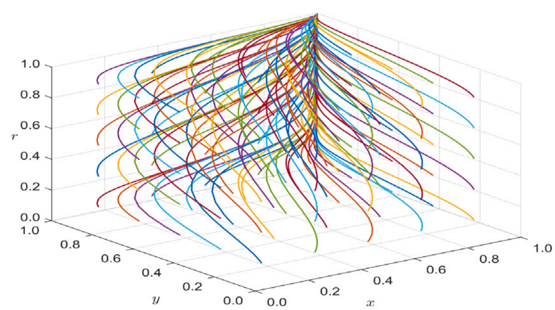
time, the corresponding Jacobian matrix eigenvalues are all negative, and the simulation results based on multi-agent reinforcement learning and matlab simulation are shown in Fig. 5(a) and (b) respectively, which are consistent with our reasoning at this time. We let $z = 0.8$ represent the platform chooses the integrity strategy, and the simulation results of the evolution process of temporary workers, contract workers and task issuers under different initial strategies are shown in Fig. 5(c). Let $x = 0$ indicate that the temporary worker chooses a fraud strategy, and simulate the evolution process of the other three different initial strategies, and the results are shown in Fig. 5(d). It can be seen that if the initial behavior of the crowdsourcing participants tends to be honest, although the temporary workers choose to delay delivery, since the contract workers and the platform respectively deliver and supervise the logistics information in a timely manner, the requester will give the contract workers real logistics rewards. At the same time, it also allows packages to be delivered in a timely manner without suffering losses.



(a) Evolution results of multi-agent reinforcement learning.

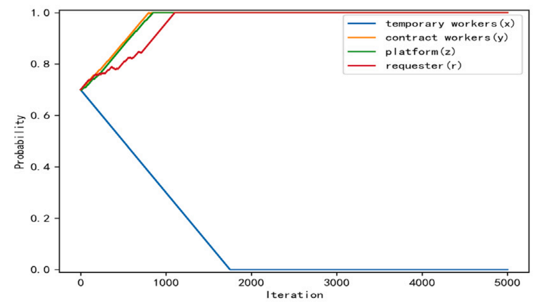


(b) Evolution results of matlab numerical simulation.

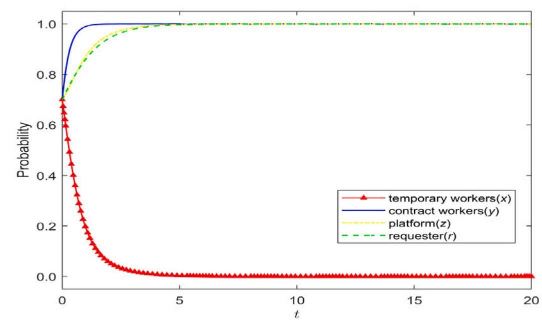


(c) Evolution results of the three parties x, y, and r under different initial values.

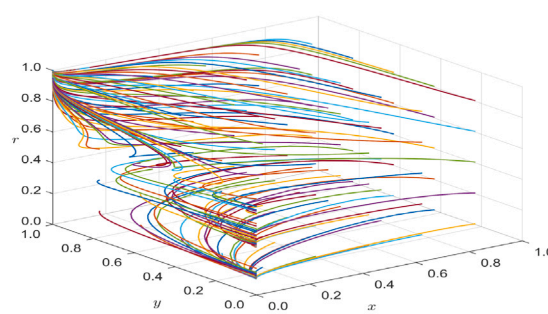
Fig. 4. Evolution of four-party under (1,1,0,1).



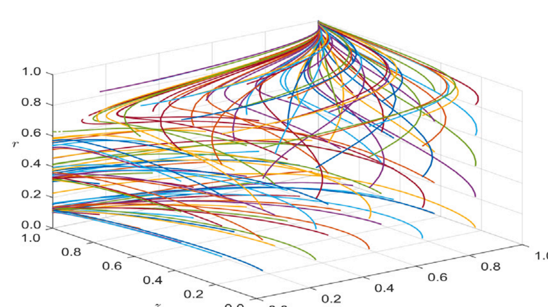
(a) Evolution results of multi-agent reinforcement learning.



(b) Evolution results of matlab numerical simulation.



(c) Evolution results of the three parties x, y, and r under different initial values.



(d) Evolution results of the three parties y, z, and r under different initial values.

Fig. 5. Evolution of four-party under (0,1,1,1).

7. Conclusion

This paper proposed a four-party evolutionary game model in a crowdsourcing logistics scenario in which players include temporary workers, contract workers, a blockchain-based platform and task requesters, and adopts a replication dynamics approach to analyze evolutionary stability strategies. Through the proposed game model, the real crowdsourcing participants are mapped to the players in the DT virtual space, and the multi-agent reinforcement learning method is used to analyze the evolution trend of the current players' strategies, and predict the development results of the crowdsourcing participants' strategies. We also added real blockchain performance data to the proposed game model to make the evolution results more realistic. As far as we know, this paper built a four-party game model in the crowdsourcing logistics scenario for the first time, and adds DT to optimize the crowdsourcing logistics system. We believe that the research results in this paper can stimulate more research on the combination of crowdsourcing logistics system and DT, which will help better understand the behavior of each participant in a real-world crowdsourcing logistics system.

CRediT authorship contribution statement

Lingjie Zhang: Conceptualized the study, Designed the research framework, Development of the Digital Twin integration in crowdsourcing logistics. **Xiaoding Wang:** Implemented the multi-agent reinforcement learning in the virtual scene of Digital Twin and contributed to the design of the reward and punishment strategy. **Hui Lin:** Conducted the analysis of the four-party evolutionary game model, including the formulation and application of the replication dynamics method. **Md. Jalil Piran:** Provided critical insights, Supervised the project, Overall manuscript preparation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

Cai, Z., He, Z., Guan, X., Li, Y., 2018. Collective data-sanitization for preventing sensitive information inference attacks in social networks. IEEE Trans. Dependable Secure Comput. 15, 577–590.
Cai, Z., Xiong, Z., Xu, H., Wang, P., Li, W., Pan, Y., 2021. Generative adversarial networks: A survey towards private and secure applications.
Chhetri, M.B., Krishnaswamy, S., Loke, S.W., 2004. Smart virtual counterparts for learning communities. In: WISE Workshops.
Chi, C., Wang, Y., Li, Y., Tong, X., 2021. Multistrategy repeated game-based mobile crowdsourcing incentive mechanism for mobile edge computing in internet of things. Wirel. Commun. Mob. Comput. 2021, 6695696:1–6695696:18.
Dasari, V.S., Kantarci, B., Pouryazdan, M., Foschini, L., Girolami, M., 2020. Game theory in mobile CrowdSensing: A comprehensive survey. Sensors (Basel Switzerland) 20.
Feng, S., Wang, W., Niyato, D.T., Kim, D.I., Wang, P., 2018. Competitive data trading in wireless-powered internet of things (IoT) crowdsensing systems with blockchain. In: 2018 IEEE International Conference on Communication Systems (ICCS). pp. 289–394.
Hong, H., Hu, B., Sun, Z., 2021. An efficient and secure attribute-based online/offline signature scheme for mobile crowdsensing. Hum.-Centric Comput. Inf. Sci. 11, 26.
Hu, J., Yang, K., Wang, K., Zhang, K., 2020. A blockchain-based reward mechanism for mobile crowdsensing. IEEE Trans. Comput. Soc. Syst. 7, 178–191.
Huang, Z., Zheng, J., Xiao, M., 2021. Privacy-enhanced crowdsourcing data trading based on blockchain and stackelberg game. In: 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS). pp. 621–626.
Jung, T., Li, X., Huang, W., Qian, J., Chen, L., Han, J., Hou, J., Su, C., 2017. AccountTrade: Accountable protocols for big data trading against dishonest consumers. In: IEEE INFOCOM 2017 - IEEE Conference on Computer Communications. pp. 1–9.
Li, J., Cai, Z., Yan, M., Li, Y., 2016. Using crowdsourced data in location-based social networks to explore influence maximization. In: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications. pp. 1–9.
Li, F., Wang, Y., Gao, Y., Tong, X., Jiang, N., Cai, Z., 2022. Three-party evolutionary game model of stakeholders in mobile crowdsourcing. IEEE Trans. Comput. Soc. Syst. 9, 974–985.
bing Liu, W., Wang, X., 2009. Dynamic decision model in evolutionary games based on reinforcement learning. Syst. Eng. - Theory Pract. 29, 28–33.
Liu, T., Wang, Y., Li, Y., Tong, X., Qi, L., Jiang, N., 2020. Privacy protection based on stream cipher for spatiotemporal data in IoT. IEEE Internet Things J. 7, 7928–7940.
Lu, Z., Wang, Y., Tong, X., Mu, C., Chen, Y., Li, Y., 2021. Data-driven many-objective crowd worker selection for mobile crowdsourcing in industrial IoT. IEEE Trans. Ind. Inform. 19, 531–540.
Negria, E., Fumagallia, L., Macchia, M., 2017. Flexible Automation and Intelligent Manufacturing , FAIM 2017 , 27-30 June 2017 , Modena , Italy A review of the roles of Digital Twin in CPS-based production systems.
Pu, B., Li, K., Li, S., Zhu, N., 2021. Automatic fetal ultrasound standard plane recognition based on deep learning and IIoT. IEEE Trans. Ind. Inform. 17, 7771–7780.
Saddik, A.E., 2018. Digital twins: The convergence of multimedia technologies. IEEE MultiMedia 25, 87–92.
Sangoleye, F., Irtija, N., Tsiropoulou, E.-E., 2021. Data acquisition in social internet of things based on contract theory. In: ICC 2021 - IEEE International Conference on Communications. pp. 1–6.
Shao, Z., Wang, H., Feng, G., 2019. PUEGM: A method of user revenue selection based on a publisher-user evolutionary game model for mobile crowdsensing. Sensors (Basel Switzerland) 19.
Swan, M., 2015. Blockchain: Blueprint for a new economy.
Wang, Y., Li, Y., Chi, Z., Tong, X., 2018. The truthful evolution and incentive for large-scale mobile crowd sensing networks. IEEE Access 6, 51187–51199.
Wang, J., Wei, B., Zhang, J., Yu, X., Sharma, P.K., 2021. An optimized transaction verification method for trustworthy blockchain-enabled IIoT. Ad Hoc Netw. 119, 102526.
Wang, J., Yang, Y., Wang, T., Sherratt, R.S., Zhang, J., 2020. Big data service architecture: A survey. J. Internet Technol. 21, 393–405.
West, T.D., Blackburn, M.A., 2017. Is digital thread/digital twin affordable? A systemic assessment of the cost of DoDs latest manhattan project. Procedia Comput. Sci. 114, 47–56.
Xie, L., Su, Z., Chen, N., Xu, Q., 2021. Secure data sharing in UAV-assisted crowdsensing: Integration of blockchain and reputation incentive. In: 2021 IEEE Global Communications Conference (GLOBECOM). pp. 1–6.
Xie, H., Xiao, L., Ming, X.G., Tan, S., Bao, Y., 2023. Driving intelligent manufacturing: An application study on digital twin in factory digitalization. In: Proceedings of the 2023 10th International Conference on Industrial Engineering and Applications.
Zhang, J., Huang, X., Ni, W., Wu, M., Yu, R., 2019. VeSenChain: Leveraging consortium blockchain for secure and efficient vehicular crowdsensing. In: 2019 Chinese Control Conference (CCC). pp. 6339–6344.
Zhang, Y., van der Schaar, M., 2011. Reputation-based incentive protocols in crowdsourcing applications. In: 2012 Proceedings IEEE INFOCOM. pp. 2140–2148.
Zhang, J., Siqi, Z., Wang, T., Chao, H.-C., Wang, J., 2020. Blockchain-based systems and applications: A survey. J. Internet Technol. 21, 1–14.
Zhang, J., Zhong, S., Wang, J., Yu, X., Alfarraj, O., 2021. A storage optimization scheme for blockchain transaction databases. Comput. Syst. Sci. Eng. 36, 521–535.
Zhao, L., Bi, Z., Hawbani, A., Yu, K., Zhang, Y., Guizani, M., 2023. ELITE: An intelligent digital twin-based hierarchical routing scheme for softwarized vehicular networks. IEEE Trans. Mob. Comput. 22, 5231–5247.
Zhao, L., Han, G., Li, Z., Shu, L., 2020. Intelligent digital twin-based software-defined vehicular networks. IEEE Netw. 34, 178–184.
Zhao, L., Wang, C., Zhao, K., Tarchi, D., Wan, S., Kumar, N., 2022. INTERLINK: A digital twin-assisted storage strategy for satellite-terrestrial networks. IEEE Trans. Aerosp. Electron. Syst. 58, 3746–3759.
Zhu, X., Li, Y., Fang, L., Chen, P., 2020. An improved proof-of-trust consensus algorithm for credible crowdsourcing blockchain services. IEEE Access 8, 102177–102187.