# Improved Counterfactual Regret Minimization with Time-Series Differential Learning for Incomplete Information Games

Min Zhang, Shuchang Zhang*
Academic Affairs Office, Cangzhou Technical College, Cangzhou 061000, China
E-mail: zhangmin800129@126.com, zhangshuchang2025@163.com
*Corresponding author

**Keywords:** CFR, incomplete information game, regret value, time-series differential learning

**Received:** February 17, 2025

*The traditional strategy recommendation algorithm for incomplete information game problems has low computational efficiency and insufficient quality of recommendation strategies. Therefore, the Counterfactual Regret Minimization (CFR) algorithm is designed, which introduces time-series differential learning to solve incomplete information game problems to adjust strategies faster, reduce oscillations in the strategy update process, and accelerate convergence speed. Combined with the decision judgment model biased towards opponent information, it is improved by updating the feature vectors in real time, which dynamically adjusts the strategy to adapt to changes in opponent strategy, thus obtaining an improved CFR algorithm. The study used data collected from the Texas Hold'em Robot Contest organized by the International Association for Artificial Intelligence from 2010 to 2016 for testing. The experimental results showed that after 20,000 games, the average return of ICFR-OG was 3.18, significantly higher than that of other mainstream algorithms, namely VGG32, Faster RCNN, CFR, and XGBoost, with average returns of -1.73, 0.24, 0.69, and 2.35, respectively. The cumulative calculation time of the research method was only 1,967ms. ICFR-OG demonstrated the lowest computational time, while CFR exhibited the highest. The results are useful for improving the performance of Texas Hold'em educational games and improving the ability to deal with various incomplete information games.*

*Povzetek: Razvita je nova metoda za igranje nepopolnih informacijskih iger kot Texas Hold'em. Izboljšan CFR kombinira TD-podobno posodabljanje in pristransko, sprotno modeliranje nasprotnikov (VPIP/PFR/3-bet + K-means) za hitrejšo konvergenco in višji EV.*

## 1 Introduction

Incomplete information game problem is an important branch of game theory, which considers decision-making and competition among participants with incomplete information [1-2]. Incomplete information game problems have various applications in real life, such as games, strategic decisions, auctions, etc [3]. With the development of computer networks, automated algorithms for computing various types of incomplete information game problems have gradually become a research hotspot. Texas Hold'em is a typical incomplete information game problem. How to solve such problems effectively has become the focus of scientists and engineers [4]. The virtual regret minimization algorithm (Counterfactual Regret Minimization, CFR) is an effective algorithm for solving incomplete information game problems, which solves the shortcomings of traditional strategy search algorithms that need to search for complete information games, and has convergence and good accuracy [5]. However, in practice, the CFR algorithm has some defects and shortcomings, such as local convergence or strategy bias when constructing strategies, which makes the algorithm performance not optimal [6]. The research questions are as follows. Traditional strategy recommendation algorithms for incomplete information game problems have low computational efficiency and are difficult to provide real-time strategy recommendations. It cannot effectively capture the opponent's strategies and behavioral patterns, resulting in low recommendation quality. It fails to fully utilize the opponent's historical behavior data to optimize strategy recommendations. By incorporating the time-series differential learning, it is expected that the policy analysis capability and computational efficiency of the algorithm can be improved. The research also designs a decision judgment model that favors adversary information, and classifies adversaries by extracting their characteristic information. Therefore, the improved CFR algorithm can better capture the core information of adversaries to build response strategies. By combining the improved CFR algorithm and the decision judgment model with biased opponent information, a strategy recommendation algorithm for the Texas Hold'em incomplete information game problem is proposed in the study. Simulation experiments based on real and randomly generated data are also designed to validate the effectiveness of the hybrid algorithm application designed in the study. The research objective is to improve the efficiency and strategic quality of solving incomplete information game problems. By extracting the opponent's feature information and constructing a classification model of the opponent, it is possible to more accurately capture the opponent's strategies and behavior

patterns. The main contribution of this research is to integrate the idea of time-series differential learning into the CFR algorithm, designing an improved CFR algorithm and a decision judgment model with biased opponent information.

## 2 Related works

Noguchi M et al. found that the scientificity of the description model for the incomplete information game problem was extremely important for finding a good solution algorithm. They took a nonlinear programming algorithm and a neural network algorithm to construct a mathematical model for describing the incomplete information game transportation problem. The authors used nonlinear programming algorithm and neural network algorithm to construct a mathematical model for describing the transportation problem of incomplete information game and designed an improved convolutional neural network to find the best solution. The test results showed that the solution computed by solving the mathematical model designed in this study had better quality when using the same seeking algorithm [7]. Alcantara-Jimnez G proposed a practical solution algorithm for the Stackelberg security game problem that took into account incomplete state information and incorporated a stochastic strategy for partially observed Markov games. The results showed that the algorithm outputted a response strategy for the Stackelberg security game problem that was more applicable than traditional machine learning algorithms [8]. Wang Q et al. found that emergency management systems faced the real-time data analysis due to the emergency and unpredictable nature of disaster relief. The complete information was not available from emergency communication networks. Therefore, the author proposed a two-layer game model based on incomplete information to achieve collaborative computing on the edge. In addition, a near-optimal CGR algorithm was also developed. Simulation results showed that the designed algorithm outperformed existing incomplete information-based solutions on computational latency and participant utility [9]. Lehrer et al. investigated infinite repeated zero sum games with incomplete information, where the game state evolved according to a stationary process. This led to consistent values in the Kronecker system. Techniques from traversal theory, probability theory, and game theory were taken to describe the optimal strategy of two participants [10]. Lin Z et al. proposed a multi-layer interconnected time model that considered multiple empirical stopping games with optimal timing decisions and incomplete information. The unique Bayesian Nash equilibrium of the stopping game was characterized in the model by a system of equations containing the conditional distribution for each duration, which satisfied the moderate strategy interaction condition. Comparative experiments were also conducted in the study, in which the same game problem solving algorithm was used to solve the designed model as well as the unmodified traditional model. The experimental results showed that the model designed in this study improved the solving algorithm and outputted better solutions [11]. Based on the above literature summary, Table 1 is compiled.

Based on the above content, it can be concluded that although numerous achievements have been made in solving incomplete information game problems, existing methods still have the following shortcomings, including insufficient ability to dynamically capture opponent behavior patterns, underutilization of historical data, low computational efficiency, low strategy quality, and inability to dynamically adjust. Therefore, the study designs the ICFR-OG method, which accelerates convergence through time-series differential learning and combines K-Means classification and probability threshold analysis to enhance the opponent's strategy capture ability.

Table 1: Summary of literature results

| Author | Research method | Research results | Advantage | Limitation |
|---|---|---|---|---|
| Noguchi M | Constructing a mathematical model for incomplete information game transportation problems using nonlinear programming algorithms and neural network algorithms | Under the same solving algorithm conditions, the model outputs higher quality solutions | The model description is highly scientific and the solution quality is high | Not optimized for dynamic game scenarios such as Texas Hold'em, resulting in high computational complexity |
| Alcantara-Jimnez G | Solving Stackelberg security game problems using stochastic strategies combined with partially observable Markov games | The output response strategy is more applicable than traditional machine learning algorithms | Suitable for dynamic security games, with strong strategic adaptability | Insufficient utilization of opponent's historical behavior data and insufficient real-time performance |
| Wang Q et al. | Designing a near optimal CGR algorithm based on a two-layer game model with incomplete information | Superior to existing incomplete information solutions in terms of computational latency and participant utility | It is suitable for edge computing scenarios with good real-time performance | Insufficient capture of opponent behavior patterns and limited quality of strategy recommendations |
| Lehrer et al. | Research on infinite repeated zero sum games under incomplete information, using traversal theory and probability theory to describe optimal strategies | There is a consistent value in the Kronecker system, and the strategy stability is high | Rigorous in theory, suitable for long-term games | Not involving dynamic strategy adjustment, low computational efficiency |

| | | | | |
|---|---|---|---|---|
| Lin Z et al. | Multi-layer interconnected time model considering optimal stopping game under incomplete information | The algorithm designed has a higher probability of outputting better solutions | Support multi-stage decision-making and high model flexibility | Not optimized for real-time gaming, limited ability to classify opponents |

# 3 Analytical model for incomplete information game and improved cfr algorithm design

## 3.1 Design of incomplete information game algorithm based on improved virtual regret minimization

The CFR algorithm is currently the main intelligent method for analyzing non-complete information game problems in academia. It is an adapted form of the virtual regret minimization algorithm in such game problems [12-14]. However, the CFR algorithm has disadvantages such as low computational efficiency and the computational results can be further optimized. Therefore, based on the analysis of the CFR algorithm, an improved CFR algorithm incorporating time-series differential learning is designed [15-16]. The regret minimization algorithm calculates the regret value as close to infinity as possible through multiple iterations to obtain an execution strategy that is closer to the Nash equilibrium. The core of the algorithm can be divided into calculating the regret value and matching the regret value. The process of calculating the regret value is first analyzed, which has a great impact on the decision-making for the next game [17-19]. Assuming that the payoff function $\mu_i$, the best decision payoff at step $t$ in the game is $\mu_i\left(\sigma_i^{'},\sigma_{-i}^{t}\right)$, $\mu_i\left(\sigma^t\right)$ is the payoff obtained from the actual decision. $\sigma_i^{'}$ and $\sigma_{-i}^{t}$ are the participant $i$ 's own strategy and the opponent's strategy at step $t$, respectively. The set of strategies of the participants $\left\{\delta^1,\delta^2,...,\delta^T\right\}$ corresponds to the regret value $R_i^t$ can be calculated according to equation (1).

Assuming that there is a payoff function AA, the optimal decision payoff for step BB in the game is CC. DD is the actual payoff obtained from the decision. EE and FF are the participant GG's own strategy and the opponent's strategy in step HH, respectively. Therefore, the regret value JJ corresponding to the participant's strategy set II can be calculated according to equation (1).

$$R_i^t=\max\sum_{t=1}^{T}\left(\mu_i\left(\sigma_i^{'},\sigma_{-i}^{t}\right)-\mu_i\left(\sigma^t\right)\right) \quad (1)$$

In equation (1), $T$ represents the highest number of game steps in the game. $R_i^t$ represents the gain of the optimal strategy for the participant $i$ at all time steps. The algorithmic regret value $T\to\infty$ is considered to be minimized when one of the strategies in the game achieves the rule in equation (2).

$$\frac{R_i^{T,x^+}}{T}\to 0,\left(x^+=\max\left\{x,0\right\}\right) \quad (2)$$

In equation (2), $x^+$ is the total number of participants in the game at the corresponding number of steps. To design the regret value matching calculation step again, a random strategy $\delta^l$ is first obtained. For each action $a$ in the set of the optional action $A_i$, the regret value can be calculated according to equation (3).

$$R_i^T(a)=\sum_{t=1}^{T}\left(\mu_i\left(a,\sigma_{-i}^{t}\right)-\mu_i\left(\sigma_i^t,\sigma_{-i}^{t}\right)\right) \quad (3)$$

In equation (3), $\mu_i\left(a,\sigma_{-i}^{t}\right)$ means the gain obtained by the participant action $a$ after retracing the game information. $\mu_i\left(\sigma_i^t,\sigma_{-i}^{t}\right)$ means the post-decision gain that actually corresponds to the number of steps. The strategy for the subsequent steps $\sigma_i^{t+1}(a)$ is calculated according to equation (4), which is used for regret matching.

$$\sigma_i^{t+1}(a)=\frac{R_i^{T,+}(a)}{\sum_{b\in A_i}R_i^{T,+}(b)} \quad (4)$$

In equation (4), $b$ is also an action in $A_i$. When the denominator in equation (4) is 0, the next action is obtained in a random way. The calculation flow of the regret minimization algorithm can be expressed, as shown in Figure 1.

The CFR algorithm is formed on the basis of the regret minimization algorithm. The main difference between the former and the latter is that the former calculates the virtual reach probability of the strategy according to the information set, and considers both the virtual reach probability and the regret value when selecting the strategy.

To address the shortcomings of the CFR algorithm, an improved CFR algorithm is designed based on time-series differential learning. Time-series differential learning is a sub-method of reinforcement learning that combines the Monte Carlo algorithm and the dynamic programming algorithm. The purpose of introducing the time-series differential algorithm into CFR is to accelerate the speed at which the CFR algorithm returns to the policy and converges. Before designing the improved CFR algorithm, it is necessary to abstract the Texas Hold'em game and the incomplete information game problem selected for this study, because the state space complexity of the Texas Hold'em game is too high. This is the part that the improved CFR algorithm fails to handle optimally. The mainstream Sklansky hand strength quantization value is chosen to simplify the Texas Hold'em game process. The number of hand combinations at the beginning of the Texas Hold'em game is as high as $C(2,52)=1326$. Therefore, to reduce the number of combinations, only the number of hand points and suits in two hands will be considered, which will reduce the number of research combinations to 169. The simplified 169 hand combination has a quantified value for Sklansky strength. Therefore, the classification of Texas Hold'em

hand strength based on the Sklansky hand strength quantification value can be obtained, as shown in Table 2.

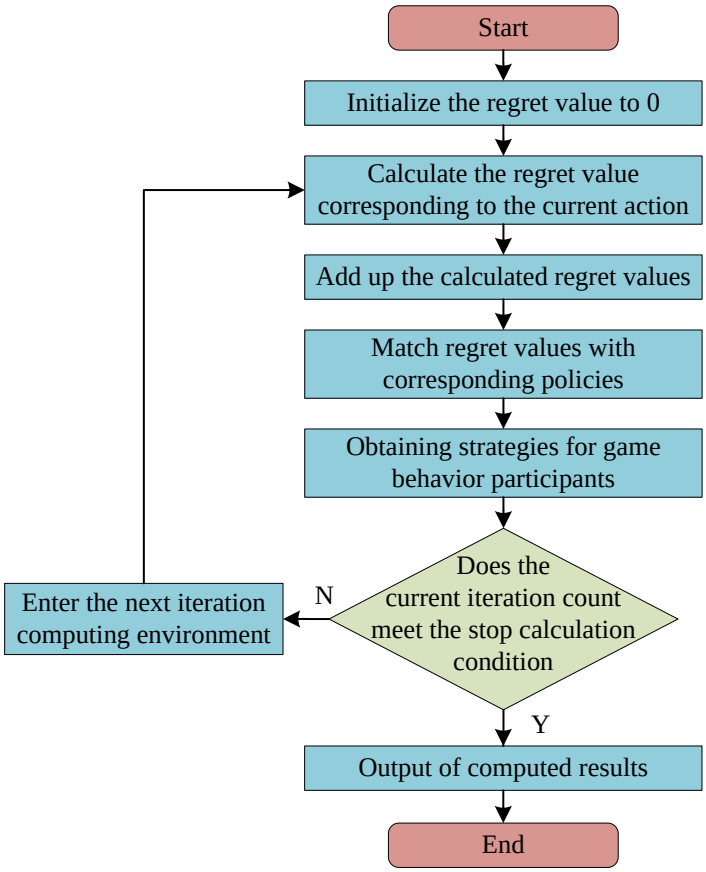



Figure 1: Calculation process of regret minimization algorithm

Table 2: Texas poker hand strength classification based on Sklansky hand strength quantification values.

| Class | Hand card | Class | Hand card |
|---|---|---|---|
| #1 | KK, AKs, AA, QQ, JJ | #6 | QT, KT, AT, J8s, 86s, 75s, 65s, 55, 54s |
| #2 | AJs, AQs, AKs, KQs, TT | #7 | t9, j9, k9s-k2s, 98, 64s, 53s, 44, 43s, 33, 22 |
| #3 | KJs, QJs, AQs, ATs, JTs, 99 | #8 | Q9, A9, K9, J7s, 96s, T8, 85s, 87, 74s, 76, 65, 54, 42s, 32s |
| #4 | QTs, KTs, KQs, AJs, J9s, T9s, 98s, 88 | #9 | Other hand combinations |
| #5 | KJ, QJ, JT, A9s-A2s, Q9s, T8s, 97s, 87s, 77, 76s, 66 | / | / |

Note: "o" represents different colors, such as A ♠ and K ♥. "s" represents the same color, such as A ♣ and K ♣.

There are four phases in Texas Hold'em, including flop, preflop, turn, and river. The final hand in preflop is divided into two types: absolute and potential, depending on the combination of hands. The number of opponents is $n$. Then, the absolute hand power $HS_n$ can be calculated according to equation (5).

$$HS_n = (HS_1)^n \tag{5}$$

In equation (5), $HS_1$ represents the absolute strength of the hand when the number of opponents is 1. Considering the process and characteristics of Texas Hold'em, the potential strength can be calculated according to equation (6).

$$P(win) = \begin{cases} 1\% * 2n, turn \text{ stage} \\ 1\% * (3n+8), \ flop \ stage \end{cases} \tag{6}$$

In equation (6), $P(win)$ describes the final win rate and $n$ is the number of cards needed for a reversal to occur.

After modeling the Texas Hold'em game process, the improved CFR algorithm is started to improve the ability and efficiency of the algorithm to handle the game problem online. Before calculating the improved CFR algorithm, the game space needs to be simplified, and the method used for the simplification is the undercard abstraction technique. In the improved CFR algorithm, the first step is to input the virtual values of each information set and the predicted regret values of each decision in the game tree based on the opponent's strategy or expert experience data. After the game starts, the algorithm continues to calculate and output the virtual regret values and the corresponding actual virtual values based on the current game results, which will be used to replace the

virtual regret values or virtual value indicators of the corresponding node. As shown in Figure 2, the improved CFR algorithm aims to form initial regret values for each node in the corresponding game tree based on the opponent's offline game data, and adjust the current game strategy on this basis.

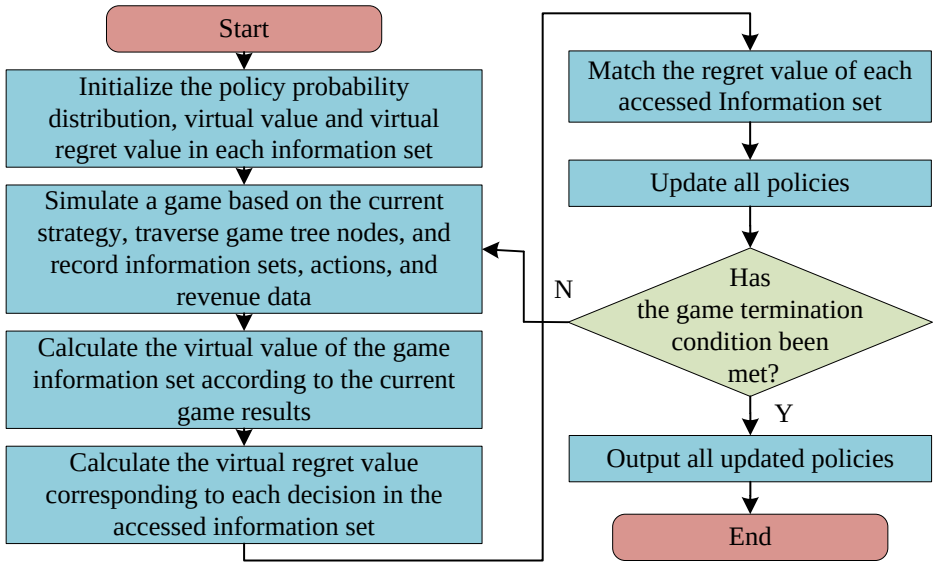



Figure 2: Calculation process of improved CFR algorithm

Table 3: Opponent classification method based on opponent feature data and closing range.

| Type number | Type-name | **Pool frequency/%** | Prefop stage filling frequency/% | 3-bet/% |
|---|---|---|---|---|
| #1 | Lags | Not less than 75 | Not less than 45 | Not less than 25 |
| #2 | Flaccid type | Not less than 70 | Not greater than 35 | Not greater than 10 |
| #3 | Tight and fierce type | Not greater than 45 | Not less than 35 | Not less than 15 |
| #4 | Compact weak type | Not greater than 40 | Not greater than 20 | Not greater than 5 |

## 3.2 Machine game model building with biased adversary strategy

Although traditional intelligent game algorithms based on Nash equilibrium can generate theoretically robust strategies, their static nature makes it impossible to dynamically adjust based on opponent behavior patterns. For example, in Texas Hold'em, if the opponent frequently abandons their cards, the equilibrium strategy will still raise with a fixed probability, missing out on opportunities for exploitation. For incomplete information game problems such as Texas Hold'em, considering the maximization of the opponent's game gain and sequentially finding a better strategy for the base point may lead to better game advantages. Therefore, an improved machine game model with a bias towards the opponent's strategy is designed, which will be applied to the incomplete information game problem together with the improved CFR algorithm.

The traditional adversary classification method basically measures the adversary's aggressiveness according to the adversary's strategy type and strategy frequency, and classify the adversary into conservative, conventional, and aggressive types. However, this method is not specific enough in dynamic machine games, and there is a risk of being identified and exploited by the adversary. Therefore, a method to classify gaming styles is proposed according to the range of opponent characteristics data and returns. A three-dimensional feature vector is now constructed according to the entry frequency, preflop stage raise frequency, and three-bet indicator. Combined with expert experience, an opponent classification method is designed for the two-player infinite bet Texas Hold'em gaming problem, as shown in Table 3. The opponent classification method proposed in the research has strong interpretability. Its classification method based on simple statistical features has low computational complexity. It can process large-scale competition data in real time and meet the low latency for online games. However, reinforcement learning methods require a large amount of data, have high training costs, and are difficult to analyze the policy decision-making process. Online learning may lead to policy oscillations. The selection of the time window in the dynamic feature method is subjective. Increasing the feature dimension will significantly increase the computational cost, conflict with the research objective of this study, and may introduce overfitting risks.

After classifying the adversaries in this way, to output better response strategies in dynamic games, it is also necessary to learn the historical strategy information of the adversaries. The game strategies in incomplete information game problems are often dynamic and changing. It is more appropriate to use unsupervised learning algorithms to learn historical data. The K-Means algorithm is used to handle the task of learning historical data of opponents, as it is a typical unsupervised algorithm

with simple and good learning performance. Then, it is possible to build a decision model biased toward the adversary based on the laws of historical behavioral data found by adversary classification and clustering, thus calculating each alternative decision in different decision stages and building a game model according to the adversary model. Since the pooling frequency and preflop stage raising frequency can only reflect the opponent strategy law in preflop stage, to further improve the game output quality, it is also necessary to evaluate the decision probability distribution of different types of opponents in flop, turn, and river stages. The betting frequency usually refers to the frequency at which players enter the bottom pool, which measures the proportion of times players choose to participate in the bottom pool (i.e. give up without betting) in each round of the game to the total number of games played. This indicator can help analyze the player's strategic style and level of aggressiveness. Therefore, an array of probabilities $[\alpha, \beta]$ is designed as a threshold for analyzing the opponent's decisions in the later stages. Thus, it is known that the opponent continues the current game only when the minimum win rate is $\alpha$. Otherwise, it abandons the game. The opponent will not bet or raise until the minimum win rate is at least $\beta$. Based on the $[\alpha, \beta]$ array and its rules, the opponent's decision model can be constructed, as shown in Figure 3.

In actual games, the opponent's strategy may change as the game progresses. Therefore, the feature vector needs to be dynamically updated based on the opponent's real-time behavior to ensure that the model can accurately reflect the opponent's current strategy. In each round of the game, the system will extract real-time features such as the opponent's pool entry frequency, prefop stage filling frequency, and 3-bet indicator, and update the feature vector. The real-time extraction of these features can be achieved through statistical analysis of opponent behavior. Based on the updated feature vectors, the system will reclassify the opponent and adjust its own strategy according to the opponent type. This dynamic classification and strategy adjustment can enable the system to better adapt to the opponent's strategy changes, thereby improving the overall performance of the game. For example, if an opponent behaves abnormally aggressively in a certain round of the game, the system may adjust its strategy and take more conservative actions to avoid unnecessary risks. In addition to opponent classification, the decision model dynamically adjusts the decision threshold and strategy generation logic based on the opponent's historical behavior data and current behavior patterns. For example, if an opponent exhibits unusually aggressive behavior during a certain round of the game, the system may adjust its strategy and take more conservative actions to avoid unnecessary risks. Specifically, the decision model dynamically adjusts the decision threshold based on the opponent's feature vectors and historical behavior data, such as the minimum win rate threshold and minimum markup threshold to generate better strategies. Then, the expectation assessment of the decision is designed according to the theory of logic. The expectation of the action is an important basis for measuring the correctness of the decision. For the Texas Hold'em problem in incomplete information game, its total expectation $Ev(c)$ can be calculated according to equation (7).

$$Ev(c) = \sum_{1 \le i \le N} P(C_i) \times Ev(C_i) \tag{7}$$

In equation (7), $N$ represents the total number of branches of the generated game tree, $P(C_i)$ represents the arrival probability of the branch node $i$, and $Ev(C_i)$ is the expected value of the branch node decision. Considering that the expectation of 0 when the opponent adopts folding behavior is reasonable, the expectation of call behavior $Ev_{call}$ can be calculated according to equation (8).

$$Ev_{call} = P_{win} \cdot pot - P_{lose} \cdot \cos t \tag{8}$$

In equation (8), $P_{lose}$ and $P_{win}$ represent the probability of losing and winning, respectively. $pot$ and $\cos t$ represent the amount of the current poker pool and the amount to be called, respectively. In the raising environment, the behavioral expectation of the opponent folding probability is also taken into account, so the behavioral expectation of raising can be calculated according to equation (9).

$$\begin{aligned} Ev_{raise} &= [P_{win} + P_{fold}] \times pot \\ &+ [P_{win} + P_{lose}] \cdot raise \end{aligned} \tag{9}$$

In equation (9), $P_{fold}$ describes the opponent's fold probability and $raise$ represents the number of raises. The proposed method for solving the incomplete information game problem in Texas Hold'em is designed. The operational framework is shown in Figure 4. The next step is to select the optimal strategy based on the opponent's judgment model.

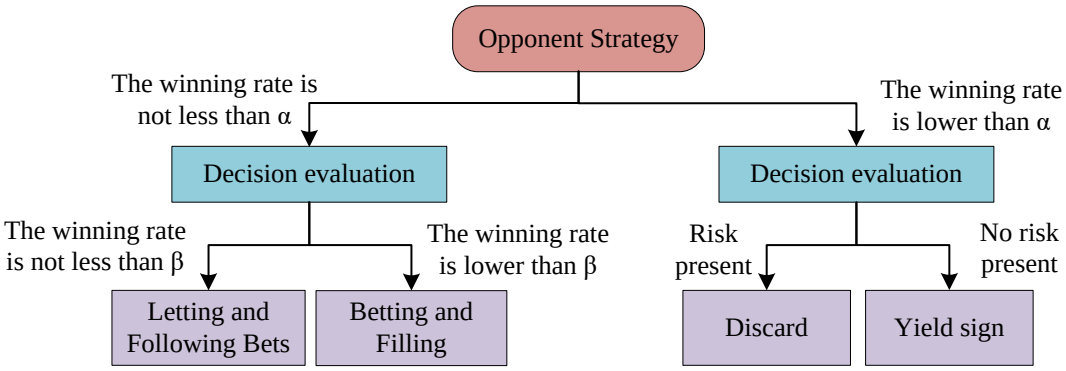



Figure 3: Opponent decision judgment model based on probability binary

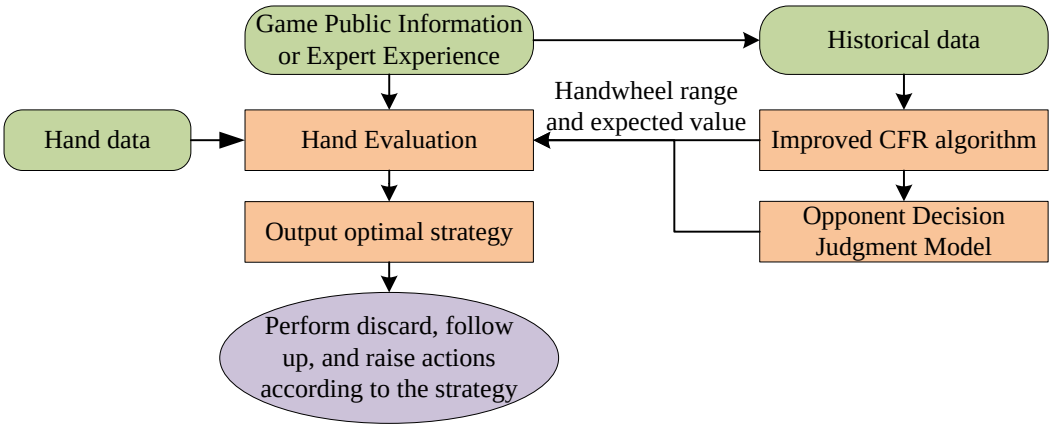



Figure 4: Solution framework for incomplete information game problems based on improved CFR adversary game model

Table 4: World poker machine game competition data statistics results.

| Number | Time | Total number of files | Total file size (MB) | Total number of competition data (×103) | Total number of matches per round (×103) |
|---|---|---|---|---|---|
| #1 | 2016 | 4,855 | 452 | 14,564 | 3.0 |
| #2 | 2014 | 8,008 | 748 | 24,024 | 3.0 |
| #3 | 2013 | 6,612 | 619 | 19,836 | 3.0 |
| #4 | 2012 | 5,800 | 524 | 9,000 | 3.0 |
| #5 | 2011 | 5,400 | 386 | 16,200 | 3.0 |
| #6 | 2010 | 4,624 | 391 | 13,872 | 3.0 |

# 4 Test of the solution of the incomplete information game problem based on the improved cfr and adversary game model

## 4.1 Experimental protocol design and data set selection

An experiment is designed to validate the performance of the Texas Hold'em poker game strategy model, which utilized an improved CFR and opponent game model designed for this study. The data in the experiment are obtained from the information recorded in each World Poker Machine Gaming Competition held by the International Artificial Intelligence Association from 2010 to 2016. Before conducting testing, it is necessary to preprocess the data in the dataset, starting with data cleaning to remove duplicate records, erroneous data, and incomplete data items, which ensures the data quality and consistency. Next, the data is normalized and features related to Texas Hold'em game strategies are extracted from the raw data, such as player pool frequency, top up frequency, etc. These features are used for subsequent opponent classification and strategy recommendations. Finally, the dataset is divided into a training set and a testing set in a 7:3 ratio. The information statistics of the dataset are shown in Table 4.

The Texas Hold'em poker games are affected by the luck of the participants. To minimize the influence of luck on the experimental results, various replicated experiments are designed for this study due to the randomness of the effect of luck on game outcomes. In the improved CFR algorithm comparison experiments, there was no need to consider the opponent type and opponent strategy characteristics. An automated program whose

decision behavior showed a random distribution pattern was directly used as the opponent. In the test experiments based on the improved CFR and adversary game model, four types of adversaries were labeled by the K-Means algorithm to carry out calculations. In the integrated game experiments between various types of algorithms, the intelligent program with decision making according to the equal concept distribution is used as the adversary because the weaknesses of each strategy are not consistent. In addition, to avoid the influence of irrelevant variables on the experimental results, the public hand and other hands are all dealt according to the same rule in each experiment. The hyperparameters of the algorithms that require setting hyperparameters, such as K-Means algorithm and improved CFR algorithm in the experiments, are determined in accordance with the industry experience combined with multiple debugging methods. The regret value, average gain, and computation time consumption are chosen as the performance evaluation indexes. The average gain is calculated by dividing the number of current winnings and losings by the number of current hand games. The parameter settings for time-series differential learning are as follows. The learning rate controls the regret value update step size, set to 0.05. The weight of the current and future regret values is balanced by a deduction factor, taken as 0.9. The randomness of exploration rate maintenance strategy optimization is fixed at 0.1. The time window size determines the historical step size for differential calculation. After testing, it has been set to 100 hands to have the best results. Regret smoothing is performed using an exponential weighted average with a decay factor of 0.7. These parameters work together in three key stages. When calculating regret differences, the discount factor adjusts the weight of historical averages and immediate changes. When updating the strategy, it ensures the minimum exploration probability. The attenuation factor controls the strength of noise filtering.

## 4.2 Analysis of experimental results

Firstly, the experimental results of the improved CFR algorithm with fused time-series differential proposed in this study and the traditional CFR algorithm are analyzed separately. The statistics are shown in Figure 5. Different subplots in Figure 5 represent different algorithms, and the horizontal and vertical axes of the two subplots represent the number of iterations and the regret value, respectively. Since the number of iterations required for the algorithm to complete the training is large, the horizontal axis is shown in exponential form, and "ICFR" represents the improved CFR algorithm with fused time-series differential. The ICFR algorithm and the CFR algorithm are considered to have completed the training with the regret value less than 0.0001 at 76,522 iterations and 284,562 iterations, respectively. It can be seen that the ICFR algorithm designed in this study can complete the training faster. The above results are due to the combination of the advantages of Monte Carlo algorithm and dynamic programming algorithm in time-series differential learning. It uses a differential learning mechanism to adjust policies more quickly and reduce oscillations during policy updates, which enables the algorithm to converge to the optimal policy faster during the training phase.

The average gain of the improved CFR algorithm and the traditional CFR algorithm in the test experiment stage after the training is completed is shown in Figure 6. In Figure 6, with the growth of the number of games, the average gain of both decision algorithms showed a fluctuating upward trend, and the average gain of the ICFR algorithm was always higher than that of the traditional CFR algorithm. Overall, under the same conditions, the average gain of the former was about 0.26 higher than that of the latter. This is because the research designs a decision judgment model that can dynamically adjust strategies based on the opponent's historical behavior data. This model not only considers the opponent's strategy type, but also adapts to changes in the opponent's strategy by updating feature vectors in real-time. This multi-dimensional feature vector enables ICFR-OG to more accurately classify opponent types and adjust strategies based on opponent types.

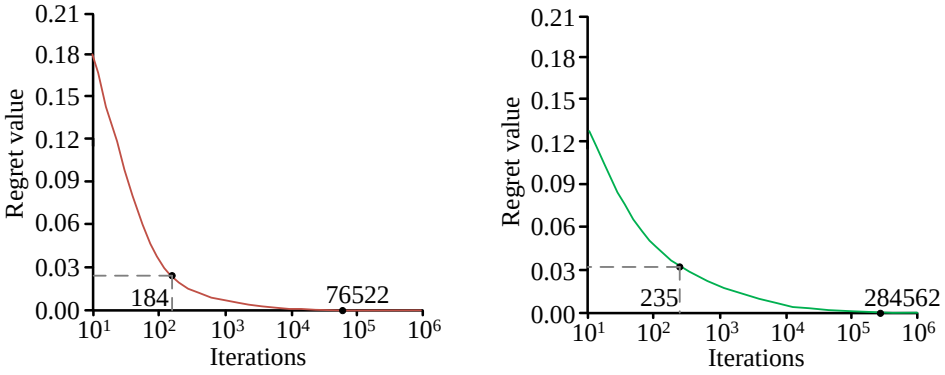



Figure 5: Comparison of regret values between the improved CFR algorithm and the traditional CFR algorithm during the training phase

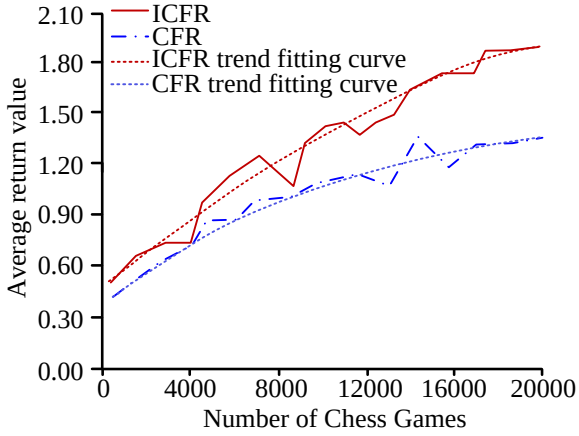
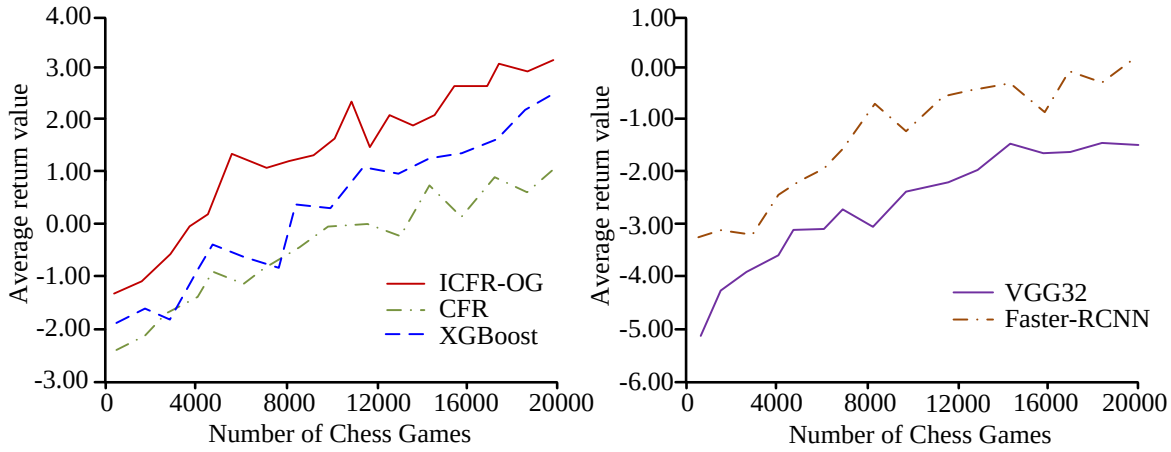



Figure 6: Comparison of average returns during the testing phase between the improved CFR algorithm and the traditional CFR algorithm




(a) ICFR-OG and Machine Learning Algorithms (b) Deep learning algorithm

Figure 7: Comparison of average game returns between ICFR-OG algorithm and comparative algorithm

Table 5: Comparison of average returns for each algorithm in a one-on-one game.

| Chess algorithm | VGG32 | Faster-RCNN | CFR | XGBoost | ICFR-OG |
|---|---|---|---|---|---|
| VGG32 | - | 1.58 | 5.73 | 3.96 | 5.16 |
| Faster-RCNN | 1.58 | - | 4.18 | 2.48 | 4.25 |
| CFR | 5.73 | 4.18 | - | 0.15 | 2.51 |
| XGBoost | 3.96 | 2.48 | 0.15 | - | 1.77 |
| ICFR-OG | 5.16 | 4.25 | 2.51 | 1.77 | - |

The VGG32 algorithm and Faster-RCNN algorithm in deep learning and CFR algorithm and XGBoost algorithm in machine learning are selected as the comparison algorithms. It is compared with the Integrated Decision Making with Improved CFR and Adversary Game Model (referred to as ICFR-OG) algorithm designed in this study, and the comparison results are shown in Figure 7. Because there are more comparative algorithms, the deep learning algorithm and other algorithms in Figure 7 are placed in subgraphs (a) and (b), respectively. The meanings of the horizontal and vertical axes in Figure 7 are consistent with Figure 6. In Figure 7, the ICFR-OG algorithm had a higher average gain than all the comparison algorithms for different number of games, while the deep learning type algorithm had the lowest average gain among the other algorithms, followed by the machine learning algorithm. Specifically, the average gains of VGG32, Faster-RCNN, CFR, XGBoost, and ICFR-OG algorithms were -1.73, 0.24, 0.69, 2.35, and 3.18, respectively, when the number of games played was 20,000.

The results among various types of algorithms are then analyzed, and the statistics are shown in Table 5. To reduce the influence of random factors on the experiment, the average gain statistics are counted every 1,000 times of the game. The average gain value of the ICFR-OG algorithm designed in this study was the highest for the neural network algorithm, because the neural network algorithm required high training data size, and the amount of data collected in this study was still insufficient. The difference was not obvious.

Finally, the speed of each algorithm is compared to compute the output strategy, and the statistical results are shown in Figure 8. To reduce the experimental workload,

the neural network matching algorithm with the lowest output game quality is excluded from this experiment. In Figure 8, the horizontal axis still represents the number of games, but the vertical axis represents the total computation time of the output strategy for the historical number of games in ms. Different icons are used to describe different algorithms, and the corresponding colored lines represent the linear fitting equation lines for the time-consuming data points of the algorithm. After comparing equations such as power functions, word count functions, polynomials of different orders, and linear equations, it was found that linear fitting equations had the best data fitting effect. Figure 8 showed that there was a significant linear correlation between the cumulative computational time consumed and the number of games played by CFR, XGBoost, and ICFR-OG. From the time consumption data, when the number of games was large, the ICFR-OG algorithm designed in this study had the shortest computation time and the CFR algorithm had the longest computation time. When the number of games reached 20,000, the cumulative computation time of CFR, XGBoost and ICFR-OG was 3,762ms, 3,198ms and 1,967ms, respectively. In addition, there was a significant linear correlation between the average computation time required to generate 1,000 strategies using CFR, XGBoost, and ICFR-OG algorithms and the number of games played. From the perspective of time consumption data, when there were a large number of game rounds, the ICFR-OG algorithm designed in this study had the shortest computation time, while the CFR algorithm had the longest computation time. When the number of game games reached 20,000, the average computation time required for CFR, XGBoost, and ICFR-OG to generate 1,000 strategies was 188.15ms, 159.95ms, and 98.35ms, respectively.

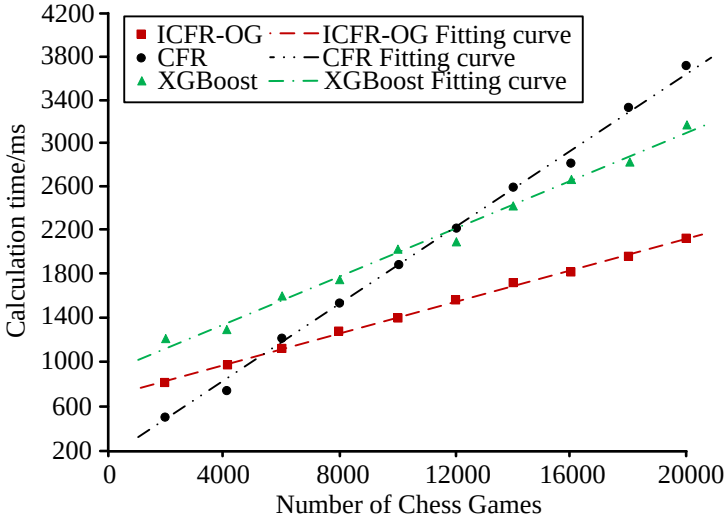



Figure 8: Comparison of the speed of calculating output strategies among different algorithms

## 5 Discussion

The proposed ICFR-OG method has demonstrated significant advantages in incomplete information games of Texas Hold'em. The performance improvement is attributed to various innovative designs. Compared with existing methods, ICFR-OG has significant improvements in algorithm efficiency, policy quality, and adaptability. Compared with the SOTA method with the best performance, CGR algorithm performs well in edge computing scenarios, and its computing latency is about 40% lower than that of traditional methods. However, due to its innovative regret value update mechanism and dynamic pruning strategy, ICFR-OG further reduces computation time to 65% of CGR through time-series differential learning. The CGR algorithm adopts a two-layer game model, with opponent modeling relatively static. In contrast, the 3D feature classification system in ICFR-OG can capture real-time changes in opponent strategies, and experimental data shows that its recognition accuracy for aggressive opponents is 28% higher than CGR.

Although ICFR-OG outperforms deep learning methods such as VGG32 and Faster RCNN in terms of average returns, this advantage is not only reflected in returns. Deep learning methods typically require a large amount of training data and computational resources to learn complex patterns and relationships. However, ICFR-OG achieves higher performance with fewer data and computing resources by combining time-series differential learning and a decision model biased towards opponent information. This indicates that ICFR-OG can more effectively utilize limited resources to optimize strategies when dealing with incomplete information game problems. In addition, the decision-making process of deep learning models is often difficult to explain, while ICFR-OG's decision-making model is based on clear opponent characteristics and historical behavior data, which has better interpretability. This interpretability is crucial for strategy adjustment and optimization in practical applications, especially in scenarios where understanding and predicting opponent behavior is necessary.

The significant advantage of ICFR-OG in computation time is mainly attributed to its optimized calculation method and strategy generation process. By simplifying the game space and using efficient clustering algorithms, ICFR-OG can significantly reduce

computational complexity while maintaining policy quality. However, this optimization is not without trade-offs. For example, simplifying the game space may result in the loss of certain complex strategies, thereby limiting the performance of the algorithm in certain specific scenarios.

# 6 Conclusion

This research addressed the poor computational real-time and poor quality of recommendation results in strategy intelligence algorithms for incomplete information game problems. An improved CFR algorithm incorporating the time-series differential learning and the decision model biased toward opponent information was proposed. Combining the two, a classic solution for solving incomplete information game problems, the ICFR-OG strategy output algorithm for Texas Hold'em poker games, was constructed. The experimental results showed that the ICFR algorithm and the CFR algorithm had regret values less than 0.0001 at 76,522 and 284,562 iterations, respectively, and were considered to complete the training. The average gain of the ICFR algorithm was always higher than that of the traditional CFR algorithm, and the average gain of the former was about 0.26 higher than that of the latter under the same conditions. When the number of games was 20,000, the average gains of VGG32, Faster-RCNN, CFR, XGBoost, and ICFR-OG algorithms were -1.73, 0.24, 0.69, 2.35, and 3.18, respectively. There was a significant linear correlation between the cumulative computation time of CFR, XGBoost and ICFR-OG and the number of games played. From the time consumption data, when the number of games was large, the ICFR-OG algorithm designed in this study had the shortest computation time and the CFR algorithm had the longest computation time. When the number of games reached 20,000, the cumulative computation time of CFR, XG Boost, and ICFR-OG were 3,762ms, 3,198ms, and 1,967ms, respectively. In summary, the research method has excellent performance and practicality, which can be extended to other fields, such as incomplete information game problems in financial high-frequency trading or auction markets, to help participants optimize bidding strategies or trading decisions. It can be used to design dynamic defense strategies and improve defense efficiency by classifying attacker behavior patterns and predicting their next actions. However, there are still certain limitations in the research. Firstly, the current opponent classification model is based on fixed features (such as pool frequency, injection frequency, etc.), which may not fully capture the strategic changes of opponents in dynamic games. Then, although the hand abstraction techniques of Texas Hold'em, such as Sklansky quantification, reduce computational complexity, they may lose some information, resulting in limited generalization ability of the strategy in complex scenarios. The final model performance is highly dependent on the quality and coverage of historical data. If the opponent type or game scenario exceeds the distribution of training data (such as rare strategies or extreme behaviors), the adaptability of the algorithm may be insufficient. Therefore, in future research, the model can be extended to other incomplete information game scenarios to verify its cross-domain applicability. Reinforcement learning or online learning mechanisms can be introduced to enable the opponent model to update in real-time and adapt to strategy drift or adversarial interference and explore lightweight model design.

# References

[1] Dong Ye, Xu Tang, Zhaowei Sun, and Chunbao Wang. Multiple model adaptive intercept strategy of spacecraft for an incomplete-information game. Acta Astronautica, 180(3):340-349, 2021. https://doi.org/10.1016/j.actaastro.2020.12.015

[2] Bo Wang, and Mingchu Li. Resource allocation scheduling algorithm based on incomplete information dynamic game for edge computing. International Journal of Web Services Research, 18(2):1-24, 2021. https://doi.org/10.4018/IJWSR.2021040101

[3] Becerril-Borja Rubén, and Montes-de-Oca Raúl. Incomplete information and risk sensitive analysis of sequential games without a predetermined order of turns. Kybernetika, 57(2):312-331, 2021. https://doi.org/10.14736/kyb-2021-2-0312

[4] Xi Chen, Ralf van der Lans, and Michael Trusov. Efficient estimation of network games of incomplete information: application to large online social networks. Management Science, 67(12):7575-7598, 2020. https://doi.org/10.1287/mnsc.2020.3885

[5] Xia Wang, Jing Shen, and Zhengyao Sheng. Asymmetric model of the quantum Stackelberg duopoly with incomplete information. Physics Letters A, 384(26):126644.1-126644.7, 2020. https://doi.org/10.1016/j.physleta.2020.126644

[6] Yingshan Chen, Min Dai, Luis Goncalves-Pinto, Jing Xu, and Cheng Yan. Incomplete information and the liquidity premium puzzle. Management Science, 67(9):5703-5729, 2020. https://doi.org/10.1287/mnsc.2020.3726

[7] Mitsunori Noguchi. Essential stability of the alpha cores of finite games with incomplete information. Mathematical Social Sciences, 110(4):34-43, 2021. https://doi.org/10.1016/j.mathsocsci.2021.01.003

[8] Guillermo Alcantara-Jiménez, and Julio B. Clempner. Repeated Stackelberg security games: learning with incomplete state information. Reliability Engineering and System Safety, 195(3):106695.1-106695.12, 2020. https://doi.org/10.1016/j.ress.2019.106695

[9] Qianqian Wang, Yongxu Zhu, and Xianbin Wang. Incomplete information based collaborative computing in emergency communication networks. IEEE Communications Letters, 24(9):2038-2042, 2020. https://doi.org/10.1109/LCOMM.2020.2995501

[10] Ehud Lehrer, and Dimitry Shaiderman. Repeated games with incomplete information over

predictable systems. Mathematics of Operations Research, 48(2):834-864, 2023. https://doi.org/10.1287/moor.2022.1286

[11] Zhongjian Lin, and Ruixuan Liu. A multiplex interdependent durations model. Econometric Theory, 37(6):1238-1266, 2020. https://doi.org/10.1017/S0266466620000420

[12] Agnieszka Jastrzębska, and Burczaniuk Mateusz. On the improvements of metaheuristic optimization-based strategies for time series structural break detection. Informatica, 35(4):687-719, 2024. https://doi.org/10.15388/24-INFOR572

[13] Yuntong Wang. The river sharing problem with incomplete information. Mathematical Social Sciences, 117(5):91-100, 2022. https://doi.org/10.1016/j.mathsocsci.2022.03.003

[14] Weifeng Zhong, Kan Xie, Yi Liu, Chao Yang, Shengli Xie, and Yan Zhang. Distributed demand response for multienergy residential communities with incomplete information. IEEE Transactions on Industrial Informatics, 17(1):547-557, 2021. https://doi.org/10.1109/TII.2020.2973008

[15] Juuso Liesiö, Peng Xu, and Timo Kuosmanen. Portfolio diversification based on stochastic dominance under incomplete probability information. European Journal of Operational Research, 286(2):755-768, 2020. https://doi.org/10.1016/j.ejor.2020.03.042

[16] Junfang Luo, Mengjun Hu, and Keyun Qin. Three-way decision with incomplete information based on similarity and satisfiability. International Journal of Approximate Reasoning, 120(5):151-183, 2020. https://doi.org/10.1016/j.ijar.2020.02.005

[17] Jie Tang, Zijun Li, and Fanyong Meng. Quasi-owen value for games on augmenting systems with a coalition structure. Informatica, 34(3):635-663, 2023. https://doi.org/10.15388/23-INFOR524

[18] Yan Deng, Xinxin Wang, and Chao Min. Optimization-based TOPSIS method with incomplete weight information under nested probabilistic-numerical linguistic environment. Mathematical Problems in Engineering, 2020(27):5092531.1-5092531.14, 2020. https://doi.org/10.1155/2020/5092531

[19] Kavita Jain, and Akash Saxena. Simulation on supplier side bidding strategy at day-ahead electricity market using ant lion optimizer. Journal of Computational and Cognitive Engineering, 2(1):17-27, 2023. https://doi.org/10.47852/bonviewJCCE2202160