Toward Zero-Determinant Strategies for Optimal Decision Making in Crowdsourcing Systems

Wang, Jiali; Tang, Changbing; Lu, Jianquan; Chen, Guanrong

Published in:
Mathematics

Published: 01/03/2023

Document Version:
Final Published version, also known as Publisher’s PDF, Publisher’s Final version or Version of Record

License:
CC BY

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.3390/math11051153

Publication details:
Toward Zero-Determinant Strategies for Optimal Decision Making in Crowdsourcing Systems

Jiali Wang 1,2, Changbing Tang 3,*, Jianquan Lu 4 and Guanrong Chen 5

1 College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321004, China
2 The Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China
3 College of Physics and Electronics Information Engineering, Zhejiang Normal University, Jinhua 321004, China
4 School of Mathematics, Southeast University, Nanjing 210096, China
5 Department of Electrical Engineering, City University of Hong Kong, Hong Kong SAR, China

* Correspondence: tangcb@zjnu.edu.cn

Abstract: The crowdsourcing system is an internet-based distributed problem-solving and production organization model, which has been applied in human–computer interaction, databases, natural language processing, machine learning and other fields. It guides the public to complete some tasks through specific strategies and methods. However, rational and selfish workers in crowdsourcing systems will submit solutions of different qualities in order to maximize their own benefits. Therefore, how to choose optimal strategies for selfish workers to maximize their benefits is important and crucial in such a scenario. In this paper, we propose a decision optimization method with incomplete information in a crowdsourcing system based on zero-determinant (ZD) strategies to help workers make optimal decisions. We first formulate the crowdsourcing problem, where workers have “winner-takes-all” rules as an iterated game with incomplete information. Subsequently, we analyze the optimal decision of workers in crowdsourcing systems in terms of ZD strategies, for which we find conditions to reach the maximum payoff of a focused worker. In addition, the analysis helps understand what solutions selfish workers will submit under the condition of having incomplete information. Finally, numerical simulations illustrate the performances of different strategies and the effects of the parameters on the payoffs of the focused worker.

Keywords: optimal strategies; iterated games; ZD strategies; winner-takes-all; incomplete information

MSC: 91A06; 91A27; 91A35; 91A80

1. Introduction

1.1. Background and Motivation

A crowdsourcing system refers to the practical framework of a company or organization outsourcing work tasks that used to be performed by workers in a voluntary manner to non-specific mass networks. Over the past decade, crowdsourcing has become a low-cost effective way to obtain simple task solutions that are difficult for humans but easy for computers [1–3]. In addition, the crowdsourcing system can be regarded as a network containing many workers as nodes. Furthermore, crowdsourcing has developed into an effective model for many challenging issues, such as algorithm theory, artificial intelligence and algorithm mechanism design [4,5].

Specifically, the core of crowdsourcing lies in the wisdom of the crowd and outsourcing [6]. Firstly, the requesters post tasks on the crowdsourcing platform and provide relevant rewards. Then, the workers accept the tasks according to their own considerations. After the tasks are solved, workers submit their solutions to the crowdsourcing platform,
which delivers them to the requesters. Finally, the system rewards the workers based on
the quality of the solutions evaluated [7, 8].

Workers who accept the tasks are clear about the task objectives. Therefore, they
analyze various possible action plans and select the optimal one to form a complete
decision-making process. Specifically, each worker has some incomplete information; i.e.,
he or she only knows his or her own information but not that of the others [9]. In this case,
each worker only knows the quality of the solution he or she will provide but does not
know the quality of the solutions provided by other workers [10].

Moreover, workers are always strategic by submitting different quality solutions to
maximize their own benefits. They will choose the right strategy from their strategy sets to
accomplish the tasks. When a worker receives a task posted by a requester, he or she may be
selfish, while all workers want to maximize their own benefits. The solution submitted by
each worker is the best response to the solutions chosen by other workers [11]. However, in
cases where the requester posts more than one task in a crowdsourcing system, each worker
chooses to participate in the tasks and submit more than one solution. As a consequence,
in the context of incomplete information [12], how to make decisions for selfish and greedy
workers to obtain the maximum benefit becomes a crucial issue to explore in crowdsourcing
systems. The analysis provides a valuable reference for future research.

1.2. Problem Formulation

In general, crowdsourcing systems classify tasks into different types. For example,
some question answering software includes dozens of task types, from study to travel.
Therefore, a crowdsourcing process will last for a long time. In this case, there are
a number of tasks in the crowdsourcing systems, and the tasks are accomplished through
a bundled scheme, i.e., several tasks operate as a group with reward splitting [13]. When
a task is published, the requester submits its reward to the crowdsourcing platform. The
crowdsourcing platform bundles several tasks, which need to be completed one by one in
sequence. As for workers, they can invest different levels of effort to solve a task, which
results in different levels of contributions. After the task is resolved and the requester’s
comments are received, the crowdsourcing platform distributes the rewards to all workers
participating in the task. Specifically, if the quality of the solution is high, the requester will
give a large reward to the worker, while if the quality of the solution is low, the requester
will give a small reward or even no reward. This is called the “winner-takes-all” rule [13, 14].

We consider the problem of optimal decision making under the rule of “winner-takes-
all” in crowdsourcing systems [15], that is, how to provide optimal quality of solutions for
selfish workers to maximize their own benefits. Here, the problem is challenging in the
following aspects. First, the strategies of workers influence each other, as each worker’s
choices affect the benefits of other workers as well as the quality of the solutions they submit.
Second, the process is iterative toward a number of tasks, in which workers will make
certain adaptive decisions according to other workers’ strategies, and each worker’s current
choice will influence other workers’ future choices. Third, each worker has only incomplete
information. Therefore, under these conditions, it is difficult for a worker to guess the
behaviors of other workers so as to make an optimal decision in the iterating process.

1.3. Solution and Contributions

Our focus is to help a worker choose the best strategies in a crowdsourcing system
to maximize his or her benefits. Although a worker will not know the strategies of other
workers, he or she can try to find the sum benefits of other workers. In 2012, Press and
Dyson [16] showed the existence of ZD strategies, which allowed a player to unilaterally
enforce a linear relationship between his or her payoff and the co-player’s payoff, regardless
of the strategy of the co-player. Inspired by this, we apply ZD strategies to deal with the
optimal decision making for workers with incomplete information in a crowdsourcing
system. The contributions of this paper are as follows.
• A crowdsourcing scenario is modeled, where workers have incomplete information, as an iterative game. In this model, the requester allocates the reward according to the “winner-takes-all” rule, for which solutions provided by different workers are independent, and selfish workers compete for the reward also with incomplete information.

• A theoretic method with ZD strategies is proposed to analyze the optimal decision-making problem in crowdsourcing systems. Moreover, the conditions to reach the maximum payoff of the focused worker who uses ZD strategies are obtained.

• Our analysis helps understand what solutions selfish workers will submit under the condition of having incomplete information. Furthermore, we provide a new optimization method, by which the optimal decision is reached in a bottom–up manner subject to incomplete information.

This paper is organized as follows. In Section 2, the relevant literature is compared. In Section 3, the crowdsourcing system is modeled as an iterative game. In addition, the ZD strategies are applied to analyze the optimal solutions of crowdsourcing systems. In Section 4, the optimization strategies are simulated numerically. Finally, conclusions are drawn in Section 5.

2. Literature Review

Lately, crowdsourcing has attracted a lot of attention [17–19], which contains applications to the design [20,21] and algorithms [22]. Performance issues in user behavior have also been investigated, with applications to quality management, incentive design, and equity [23]. Crowdsourcing is designed primarily for solving challenging tasks that require specific skills. Elance [7] and Fiverr [8] are two real crowdsourcing systems. The least worker selection was studied in [24] to enable large-scale crowdsourcing systems to improve the effectiveness of perception tasks. In [25], an optimal model and blockchain-based architecture are developed to manage the operation of crowdsourced energy systems. Moreover, there are also some novel optimization algorithms. For example, the combination of data envelopment analysis and the Malmquist index method can be used to assess the efficiency of the cybersecurity industry [26].

In addition, there are some studies focusing on strategic crowdsourcing. In social networks, in order to optimize social welfare and reduce time complexity, two efficient mechanisms are constructed for different-scale applications. However, there are few works focusing on the decision making in crowdsourcing systems. In particular, few works studied the behavior of “winner-takes-all” in crowdsourcing systems [27–29]. As an application example, a fully unsupervised pipeline was proposed to train a convolutional neural network that effectively eliminates misclassified pixels. As the model was trained, the quality of the generated labels was improved [30,31].

Meanwhile, game theory is used to explore crowdsourcing. For instance, a game theory framework is built to study user behavior and motivational patterns in social media networks in [32,33]. Especially, ZD strategies are a class of conditional strategies in game theory. Moreover, ZD strategies in finitely repeated (two-person) prisoners dilemma games with a general payoff matrix are discussed in [34,35]. For example, ZD strategies are applied to multi-player social dilemmas to obtain a ZD Nash equilibrium [36,37]. In addition, ZD strategies can also be applied to crowdsourcing systems [38] and blockchain [39].

In this paper, we focus on the decision making of selfish workers with incomplete information under the rule of “winner-takes-all” from the viewpoint of game theory. First, compared with the results of [13], we consider that in the complex interaction process among workers, the solutions submitted by each worker are not only determined by themselves but also influenced by the results of other workers in the last interaction process. In particular, each worker has incomplete information. Second, we consider the effect of “winner-takes-all” on the decision making in crowdsourcing systems. In contrast, in [28], the main concern is the “winner-takes-all” dynamics, and in [13], the incentive mechanism of the crowdsourcing system is studied under the rule of “winner-takes-all”. Third, compared with the results of [14,34,35] which only considered the dynamic problem of the strategy,
we extend the multi-player ZD strategy theory and realize the optimization of worker’s payoff in the case of incomplete information. Last, compared with [4,21] which mainly used the ZD strategy to improve the social welfare of the crowdsourcing system, this paper mainly focuses on maximizing the workers’ own payoffs.

3. Method and System Model

3.1. Crowdsourcing System

Generally, there are a number of tasks in a crowdsourcing system. A typical crowdsourcing system categorizes tasks into different types. In this paper, we mainly analyze one type of task, which can be extended to multiple tasks. In so doing, we can regard multiple-task process as a long-term iterative process.

We first consider the crowdsourcing process with one task. The requester publishes the task in the crowdsourcing system and provides reward for any worker to complete the task. A worker can choose to receive the task according to his or her convenience and provide a solution to the requester within the specified time. All workers’ solutions are mutually independent, and they can strategically select the level of the contribution they wish to submit. For simplicity, we focus on two levels of contribution in this paper, which corresponds to a high-quality solution $H$ and a low-quality solution $L$, respectively. At the same time, each worker only knows the solution he or she proposes but does not know that of other workers. That is, in the case of incomplete information, selfish workers would try to maximize their interests. Once the workers submitted their solutions, the requester evaluates their solutions and distributes the rewards under the “winner-takes-all” rule. The prepared rewards will be distributed to those who provide high-quality solutions, while the workers who provide low-quality solutions will receive no rewards. If no one provides a high-quality solution, then the rewards will be divided equally among all workers.

In crowdsourcing systems, it is common for requesters to distribute rewards based on the quality of workers’ solutions, e.g., Elance [7] and Fiverr [8]. Each worker chooses to take on tasks and demonstrate their skills. After completing the task, the corresponding payoffs will be obtained according to the “winner-takes-all” rule developed by requesters. For clarity, the notations are summarized in Table 1.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Meaning of Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>Total number of players</td>
</tr>
<tr>
<td>$r$</td>
<td>The reward provided by requester</td>
</tr>
<tr>
<td>$c_s$</td>
<td>The cost of worker with the $s$-th level of the solution</td>
</tr>
<tr>
<td>$R_k$</td>
<td>The reward of worker $k$</td>
</tr>
<tr>
<td>$k$</td>
<td>The index of worker</td>
</tr>
<tr>
<td>$j$</td>
<td>The index of focused worker</td>
</tr>
<tr>
<td>$x_k$</td>
<td>The strategy of worker $k$</td>
</tr>
<tr>
<td>$N_l$</td>
<td>The threshold value of strategy $t_s$</td>
</tr>
<tr>
<td>$i$</td>
<td>The $i$-th result of each round</td>
</tr>
<tr>
<td>$n$</td>
<td>The number of workers with high-quality solutions except the focused worker</td>
</tr>
<tr>
<td>$p_k$</td>
<td>Worker $k$’s mixed strategy vector</td>
</tr>
<tr>
<td>$p^j_i$</td>
<td>The focused worker $j$ takes a ZD strategy</td>
</tr>
<tr>
<td>$p^k_i$</td>
<td>The conditional probability of worker $k$ with outcome $i$</td>
</tr>
<tr>
<td>$U^k$</td>
<td>The payoff vector of worker $k$</td>
</tr>
<tr>
<td>$U^k_i$</td>
<td>The payoff of worker $k$ of $i$-th outcome</td>
</tr>
</tbody>
</table>
Workers will select their strategies from the strategy set because the payoff of workers will decrease as the number of high-level solutions. Thus, there is a threshold value where \( k \) worker share the reward equally.

Clearly, for a multiple-task process with repeated interactions of workers in crowdsourcing, the impacts of their actions during any rounds of future feedback from the other players. In the process, it is necessary for workers to consider their strategies under the situation of incomplete information. Under this condition, providing high-level solutions is not always a best choice for some workers, where \( k \) represents worker's strategy. Let \( u_k(X) \) be the payoff of worker taking strategy \( x_k \), where \( x_k \in X \). Now, define worker \( k \)'s payoffs as follows:

\[
u_k(X) = R_k(X) - c_s, \tag{1}
\]

where \( R_k(X) \) is worker \( k \)'s reward, and \( c_s \) is the cost corresponding to contribution \( H \) or \( L \), i.e., \( c_s = c_H \) (or \( c_s = c_L \)), if the worker chooses the strategy \( H \) (or \( L \)). We apply the “winner-takes-all” reward allocation scheme to express \( R_k(X) \). Under this allocation scheme, \( R_k(X) \) is determined by the number of workers submitting high-quality solutions. Only workers who submit high-quality solutions \( H \) will receive rewards; otherwise, there will be no rewards. However, if all workers provide a low-quality solution \( L \), then all workers share the reward equally.

Assume that \( r > c_H \) and \( \frac{r}{N} - c_L < 0 \), which result from the incentive for each worker for participation and the utility of each worker being smaller than 0, when many workers share the rewards. Thus, there is a threshold value \( N_l \) that satisfies the inequality \( r - c_L > r - c_H > \frac{r}{N} - c_H > 0 > \frac{r}{N_l+1} - c_L > \frac{r}{N_l+1} - c_H > \frac{r}{N} - c_L > \frac{r}{N} - c_H \) (see Table 2). Under this condition, providing high-level solutions is not always a best choice for some workers, because the payoff of workers will decrease as the number of high-level solutions increases. The low-level solutions become favorable when too many workers provide high-level solutions \( (\frac{r}{N_l+1} - c_H < -c_L) \). In this case, doing nothing is better than performing the task except for every worker providing low-level solutions. Thus, it is difficult for selfish workers to choose a best strategy under the situation of incomplete information.

### Table 2. Payoff matrix of workers.

<table>
<thead>
<tr>
<th>Number of ( H )</th>
<th>( N - 1 )</th>
<th>( ..., )</th>
<th>( N_l - 1 )</th>
<th>( ..., )</th>
<th>( 1 )</th>
<th>( 0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payoff of ( H )</td>
<td>( \frac{r}{N} - c_H )</td>
<td>( ..., )</td>
<td>( \frac{r}{N} - c_H )</td>
<td>( ..., )</td>
<td>( \frac{r}{N} - c_H )</td>
<td>( r - c_H )</td>
</tr>
<tr>
<td>Payoff of ( L )</td>
<td>( -c_L )</td>
<td>( ..., )</td>
<td>( -c_L )</td>
<td>( ..., )</td>
<td>( -c_L )</td>
<td>( \frac{r}{N} - c_L )</td>
</tr>
</tbody>
</table>

As for a multiple-task process, the continuous interactions of workers are modeled as an iterated game in this paper. In the process, it is necessary for workers to consider the impacts of their actions during any rounds of future feedback from the other players. Clearly, for a multiple-task process with repeated interactions of workers in crowdsourcing, the situation becomes more complicated. Thus, how to make decisions for selfishness workers to obtain the maximum payoff over the course of a long iterative process becomes...
a challenging question. In the following subsection, we will consider how workers can make an optimal decision in an incomplete information scenario.

3.3. ZD Strategies for Multiple-Player Iterated Games

An iterative game consists of several consecutive games played by the same opponents. For the infinite iterative game, it was found that there was no advantage for the long-memory player over the short-memory player in [16]. Thus, we assume that every player remembers only the previous move, i.e., at the current iteration of the game, the actions of all players depend only on the outcome of the previous round. As a result, players will only adjust their strategies based on the outcome of the previous round. Then, the process can be a stochastic process, which can be represented by a Markov chain. As a result, there is a corresponding transition matrix \( M \), which is calculated based on the output probability of the previous round and the possible output probability of this round. In particular, if \( M_{ij} \neq 0 \) (for all \( i, j \)), then the chain is ergodic. In this case, \( \lim_{t \to \infty} M^t = (\pi, \pi, \ldots, \pi) \), where \( \pi \) is the stationary distribution of the chain and is captured by \( \pi^T \cdot M = \pi^T \). Therefore, the limit distribution is the stationary distribution. This implies that if the repeated round is sufficiently many in number, the finite iterative game can be taken as an approximation of the infinitely repeated game. Since there are many tasks in the crowdsourcing systems, there are many interactions among workers. Therefore, the crowdsourcing system is a long-term iterative process, which can also be taken as an approximation of the infinitely repeated game.

Consider each worker with two strategies in multi-worker games. Assume that there are \( N \) workers and two strategies, so that there are \( 2^N \) possible results in a round. For worker \( k \), the mixed strategy \( \mathbf{p}^k \) is used to represent the conditional transition probability vector for each possible result.

In addition, \( \mathbf{p}^k \) is a \( 2^N \)-dimensional vector, i.e.,

\[
\mathbf{p}^k = [p^k_1, \ldots, p^k_i, \ldots, p^k_{2^N}]^T,
\]

where \( p^k_i \) is the probability that worker \( k \) chooses to provide a high-quality solution in this round under the premise of the \( i \)-th output result of the previous round.

Suppose that worker \( k \) provides a high-quality solution \( H \) in the last round. In addition, there are \( n \in \{0, 1, 2, \ldots, N - 1\} \) opponents chosen to provide high-quality solutions for worker \( k \). Then, the probability that he or she provides a high-quality solution in this round is \( p_{H,n}^k \). In contrast, if he or she provides a low-quality solution \( L \) in the last round, and worker \( k \) has \( n \in \{0, 1, 2, \ldots, N - 1\} \) opponents chosen to provide high-quality solutions; then, the probability that he or she chooses to provide a high-quality solution in this round is \( p_{L,n}^k \). However, it is not important for worker \( k \) to know the strategy of a specific opponent. What is important is to know how many opponents chosen to provide high-quality solutions \( H \). Therefore, the \( 2^N \) components in (2) can be transformed into a \( 2^N \)-dimensional vector:

\[
\mathbf{p}^k = [p^k_{H,0}, \ldots, p^k_{H,n}, \ldots, p^k_{H,N-1}, p^k_{L,0}, \ldots, p^k_{L,n}, \ldots, p^k_{L,N-1}]^T,
\]

where \( p^k_{H,n} \) and \( p^k_{L,n} \) each contain \( \binom{N-1}{n} \) terms. For instance, the outcomes \( HHL \) and \( HLH \) when \( N = 3 \) have the same transition probabilities from the same state. Therefore, we can reduce the dimensions of vector \( \mathbf{p}^k \) from \( 2^N \) to \( 2N \). According to the payoff matrix in Table 2, for a worker \( k \) who submits a solution at the level of \( H \) in the outcome \( i \), the payoff is

\[
U^k_i = r/(n+1) - c_H.
\]

In outcome \( i \), if worker \( k \) submits an \( L \)-level solution, the payoff will be

\[
U^k_i = \begin{cases} 
  r/N - c_L, & \text{if } n = 0 \\
  - c_L, & \text{otherwise}.
\end{cases}
\]


Hence, the payoff vector of worker $k$ is
\[ \mathbf{U}^k = [U^k_1, \ldots, U^k_i, \ldots, U^k_{2N}]^T. \] (6)

If worker $k$ chooses $H$ (or $L$), and the number of opponents who choose to provide a high-quality solution is $n$, then
\[ \mathbf{U}^k = [U^k_{H,0}, \ldots, U^k_{H,n}, \ldots, U^k_{H,N-1}, U^k_{L,0}, \ldots, U^k_{L,n}, \ldots, U^k_{L,N-1}]^T. \] (7)

There is a Markov chain with a state transition matrix $\mathbf{M}$ to represent this process. In this paper, we assume that the transition matrix $\mathbf{M}$ is regular. There is a stationary vector. Let $\mathbf{M}' \equiv \mathbf{M} - \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Define $\mathbf{v}$ as a stationary vector, satisfying $\mathbf{v}^T \cdot \mathbf{M} = \mathbf{v}^T$ and $\mathbf{v}^T \cdot \mathbf{M}' = 0$. In addition, define $\mathbf{f}$ as the last column of $\mathbf{M}'$. In [40], the equation $\mathbf{v}^T \cdot \mathbf{f} = \text{det}(\mathbf{p}^1, \ldots, \mathbf{p}^k, \ldots, \mathbf{f})$ is derived, where $(\mathbf{p}^1, \ldots, \mathbf{p}^k, \ldots, \mathbf{f})$ is a $2^N \times 2^N$ matrix. By Laplace expansion, $\mathbf{f}$ can be replaced by $\alpha \mathbf{U}^1 + \sum_{k=2}^N \beta_k \mathbf{U}^k - \gamma \mathbf{1}$. Then, a linear combination of all the players’ expected payoffs is obtained, i.e.,
\[ \alpha \mathbf{E}^1 + \sum_{k=2}^N \beta_k \mathbf{E}^k - \gamma = \frac{\text{det}(\mathbf{p}^1, \ldots, \mathbf{p}^k, \ldots, \alpha \mathbf{U}^1 + \sum_{k=2}^N \beta_k \mathbf{U}^k - \gamma \mathbf{1})}{\text{det}(\mathbf{p}^1, \ldots, \mathbf{p}^k, \ldots, \mathbf{1})}, \] (8)
where $\gamma$ is a scalar, and $\alpha, \beta_k (k \in 2, 3, 4, \ldots, N)$ are weight factors of $\mathbf{U}^k$.

If worker $j$ selects $\mathbf{p}^j$ properly to satisfy the following equation:
\[ \mathbf{p}^j = \lambda (\alpha \mathbf{U}^1 + \sum_{k=2}^N \beta_k \mathbf{U}^k - \gamma \mathbf{1}), \] (9)
where $\lambda$ is a scaling coefficient. For worker $j$, he/she can unilaterally form a linear relationship among the excepted payoffs of all workers:
\[ \alpha \mathbf{E}^1 + \sum_{k=2}^N \beta_k \mathbf{E}^k - \gamma = 0. \] (10)

The linear relationship of $\mathbf{p}^j$ can be called ZD strategies, in which $\text{det}(\mathbf{p}^1, \ldots, \mathbf{p}^j, \ldots, \mathbf{f}) = 0$. Note that $\mathbf{p}^j$ is the last column of $\mathbf{M}'$. We can know that it is determined by $\mathbf{p}^j$. We denote it as
\[ \mathbf{p}^j = [-1 + p_{H,0}^j, \ldots, -1 + p_{H,N-1}^j, p_{L,0}^j, \ldots, p_{L,N-1}^j]. \] (11)

For example, consider a 3-player repeated game that follows the “winner-takes-all” rule. Each player chooses his or her own strategy independently during each step of the game. Each player is set to have only one memory. The choices that each player makes in one round are related to the choices they had made in the last round and to the number of players who chose to cooperate in the previous round. As before, we express the strategy of high-level work as $H$ and that of low-level work as $L$. As a result, in the three-player crowdsourcing system, the outcome would be $\{HHH, HHL, HLH, HLL, LHH, LLH, LHL, LLL\}$. For an arbitrary player $x \in \{1, 2, 3\}$, a mixed strategy $\mathbf{p}^x$ is a vector that consists of conditional probabilities for high-quality solutions with respect to each of the following possible outcomes:
\[
\begin{align*}
\mathbf{p}^1 &= [p_{H,2}^1, p_{H,1}^1, p_{H,0}^1, p_{L,2}^1, p_{L,1}^1, p_{L,0}^1]^T \\
\mathbf{p}^2 &= [p_{H,2}^2, p_{H,1}^2, p_{H,0}^2, p_{L,2}^2, p_{L,1}^2, p_{L,0}^2]^T \\
\mathbf{p}^3 &= [p_{H,2}^3, p_{H,1}^3, p_{H,0}^3, p_{L,2}^3, p_{L,1}^3, p_{L,0}^3]^T
\end{align*}
\] (12)

We consider an update to the “winner-takes-all” rule. The modified rule would imply that only those who provide high-quality solutions would receive rewards. However, if no one chooses to deliver a high-quality solution, then all of them would share the reward equally.
Theorem 2. In the multi-worker crowdsourcing game, when worker \( j \) uses the ZD strategies \( \tilde{p}^j \), and the parameters satisfy \( \alpha \neq 0, \beta_k = \beta = 0 \), the expected payoff for worker \( j \) is \( (E^1)_{\text{max}} \).

Proof. See Appendix B. \( \square \)

Theorem 1. In the multi-worker crowdsourcing game, when worker \( j \) uses the ZD strategies \( \tilde{p}^j \), and the parameters satisfy \( \alpha \neq 0, \beta_k = \beta = 0 \), the expected payoff for worker \( j \) is \( (E^1)_{\text{max}} \).

Proof. See Appendix A. \( \square \)

We set the reward to \( r \). Then, we can obtain the payoff vectors \( u^x \) for the three-player games using the “winner-takes-all” rule as follows.

\[
\begin{align*}
   u^1 &= \left[ r - \epsilon_H, \frac{r}{3} - \epsilon_H, \frac{r}{2} - \epsilon_H, r - \epsilon_H, -\epsilon_H, -\epsilon_L, r - \epsilon_L, \frac{r}{3} - \epsilon_L \right] \\
   u^2 &= \left[ \frac{r}{3} - \epsilon_H, \frac{r}{3} - \epsilon_H, -\epsilon_L, r - \epsilon_L, -\epsilon_H, -\epsilon_L, r - \epsilon_H, \frac{r}{3} - \epsilon_L \right] \\
   u^3 &= \left[ \frac{r}{3} - \epsilon_H, r - \epsilon_L, \frac{r}{2} - \epsilon_H, r - \epsilon_L, -\epsilon_H, r - \epsilon_L, -\epsilon_H, \frac{r}{3} - \epsilon_L \right]
\end{align*}
\]  

(13)

Let \( u \) denote the last column of \( M' \). After some elementary column operations on matrix \( M' \), the dot product of an arbitrary vector \( u \) with the stationary vector \( v \) is obtained to be equal to the determinant \( \det(p^1, p^2, p^3, u) \), in which the fourth, sixth, and seventh columns \( p^1, p^2, \) and \( p^3 \) are controlled only by the worker 1, 2, and 3, respectively. Specifically,

\[
v^T \cdot u = \det(p^1, p^2, p^3, u) =
\]

\[
\begin{bmatrix}
-1 + p_{H_2}^1 p_{H_2}^2 p_{H_2}^3 & \cdots & -1 + p_{H_2}^1 & \cdots & -1 + p_{H_2}^1 & u_1 \\
p_{H_1}^1 p_{H_1}^2 p_{H_2}^3 & \cdots & -1 + p_{H_1}^1 & \cdots & p_{L_2}^1 & u_2 \\
p_{H_1}^1 p_{H_1}^2 p_{H_1}^3 & \cdots & -1 + p_{H_1}^1 & \cdots & p_{H_3}^1 & u_3 \\
p_{H_1}^1 p_{L_3}^1 p_{L_3}^1 & \cdots & -1 + p_{H_1}^1 & \cdots & p_{L_3}^1 & u_4 \\
p_{L_2}^1 p_{H_2}^1 p_{H_1}^3 & \cdots & p_{L_2}^1 & \cdots & -1 + p_{H_1}^1 & u_5 \\
p_{L_1}^1 p_{H_3}^1 p_{L_3}^1 & \cdots & p_{L_1}^1 & \cdots & -1 + p_{H_3}^1 & u_6 \\
p_{L_1}^1 p_{L_3}^1 p_{H_0}^1 & \cdots & p_{L_1}^1 & \cdots & -1 + p_{H_0}^1 & u_7 \\
p_{L_1}^1 p_{L_0}^1 p_{L_0}^1 & \cdots & p_{L_1}^1 & \cdots & -1 + p_{H_0}^1 & u_8 \\
\end{bmatrix}
\]

(14)

If player 1 can properly set \( p^1 \), then \( p^3 \) would satisfy \( \tilde{p}^3 = a u^1 + \beta_2 u^2 + \beta_3 u^3 + \mu \), where \( a, \beta_2, \beta_3 \) and \( \mu \) are all correlation coefficients. Thus, each player’s expected payoff can be made linear, i.e., \( a E^1 + \beta_2 E^2 + \beta_3 E^3 + \mu = 0 \). This is called the three-player zero-determinant strategy.

3.4. Game Analysis with ZD Strategies

In this subsection, we discuss optimization for workers with ZD strategies in Equation (9).

When \( \tilde{p}^j = \lambda (a U^1 + \sum_{k=2}^N \beta_k U^k - \gamma 1) \), according to Equation (10), we have \( \alpha E^1 = -\sum_{k=2}^N \beta_k E^k + \gamma \). Then

\[
\begin{align*}
\left\{ \begin{array}{l}
   p_{H,n}^{j} &= 1 + \lambda (a U_{H,n}^1 + \sum_{k=2}^N \beta_k U_{H,n}^k) - a E^1 - \sum_{k=2}^N \beta_k E^k \\
   p_{L,n}^{j} &= \lambda (a U_{L,n}^1 + \sum_{k=2}^N \beta_k U_{L,n}^k) - a E^1 - \sum_{k=2}^N \beta_k E^k \\
\end{array} \right.
\end{align*}
\]  

(15)

If we want to obtain an optimal decision in the multi-worker crowdsourcing game, the maximum payoff of \( E^1 \) can be transformed into the following optimization problem:

\[
\begin{align*}
\max \quad & E^1 \\
\text{s.t.} \quad & 0 \leq p_{H,n}^{j}, p_{L,n}^{j} \leq 1, \ \forall n \in \{ 0, \ldots, N-1 \} \\
 & \tilde{p}^j = \lambda (a U^1 + \sum_{k=2}^N \beta_k U^k - \gamma 1) \\
 & \lambda \neq 0.
\end{align*}
\]  

(16)
Theorem 3. In the multi-worker crowdsourcing game, when worker \( j \) takes the ZD strategies \( \mathbf{p}^j \), and the parameters satisfy \( \alpha \neq 0, \beta_k \neq \beta \neq 0 \), the expected payoff for worker \( j \) is \( (E^j)_{\text{max}} \).

Proof. See Appendix C. \( \square \)

Remark 1. From the above theorems, we obtain the maximum payoff of the focused worker under different values of \( \alpha \) and \( \beta_k \), respectively. That is, we can clearly see the range of payoffs for the focused worker with ZD strategies, where the focused worker can make optimal decisions under different values of \( \alpha \) and \( \beta_k \). Take Theorem 1 as an example. In Equation (A5),

\[
E^1_{\text{max}} = \frac{\lambda}{\lambda + 1} - c_H + \frac{\lambda}{\lambda + 1},
\]

where \( \lambda > 0 \) and \( \alpha > 0 \). We can see that with the increase of \( \alpha \), the payoff of the focused worker will decrease. Therefore, in this case, the focused worker should choose as small an \( \alpha \) as possible when choosing strategies. In contrast, in Equations (A8) and (A11), \( \lambda > 0 \) and \( \alpha < 0 \). We can see that as \( \alpha \) increases, so does the payoff to the focused worker; therefore, the focused worker should choose as large a \( \alpha \) as possible when choosing strategy.

Remark 2. The total revenue of other players, \( \sum_{k=2}^{N} E^k \), can be calculated using mathematical expectation. However, the calculation process uses scaling, so the result may not be accurate. Nevertheless, we can always obtain the simulation results.

4. Numerical Results and Discussion

In this section, we evaluate the performances of the ZD strategies for different situations discussed in Section 3 through several simulation experiments.

Let the initial probability be \( v_0 = [0.15, 0.15, 0.15, 0.15, 0.15, 0.15] \), and \( r = 3 \), \( c_L = 0.1 \), \( c_H = 0.2 \) in the three-worker iterated games. We consider this to be a simulation run when the payoffs of all workers converge from the random initial state to the stable state. We set the step size to 50 and average 20 independent runs.

To verify the effectiveness of our proposed method, ZD strategies are compared with other possible strategies. In Figures 1–3, the results of Theorem 1 are displayed, where \( \alpha = 0.1 \), \( \beta_2 = \beta_3 = 0 \). As shown in Figure 1, when worker 1 adopts ZD1 strategies to maximize his or her payoff and other workers follow the strategies of \( \mathbf{p}^1 = [1, 1, 0, 0, 0, 0, 0, 1]^T \) and \( \mathbf{p}^2 = [1, 0, 0, 1, 0, 1, 0, 0, 0, 1]^T \), the payoff of worker 1 is greater than that of worker 2 or 3. In addition, their payoffs basically reach the stable state, which are 1.5, 1.1 and 0.5, respectively. We can see that worker 1 will benefit most from adopting ZD strategies. In Figure 2, the effect of the weight factor on the payoff of worker 1 is shown. When worker 1 adopts ZD1 strategies with different \( \alpha \) values, the payoff of worker 1 is basically stable between 1.2 and 1.6. However, the payoff of worker 1 decreases as the parameter \( \alpha \) increases. In Figure 3, we compare ZD1 strategies with TFT and WSLS, respectively. We can see that if worker 1 takes ZD1 strategies, while other workers take TFT strategies, the payoff of worker 1 will reach the maximum value 1.4.

Remark 3. TFT strategies have two steps: (1) cooperate in the first round; (2) next round depends on the strategies of the others in the last round. If the other worker betrayed last time, worker 1 will also betray this round. If the other workers cooperated the last time, they will cooperate again in this round, where the cooperation is equivalent to high-quality solutions submitted by workers (conversely, defection is equivalent to low-quality solutions submitted by workers). As for WSLS strategies, when the yield meets the expected value, the previous behavior will continue, but if the yield value is too low, the behavior will change.

For ZD2 strategies of Theorem 2, we conduct simulation experiments with the weight factor \( \lambda = 0.001 \) (see Figures 4–7). As shown in Figure 4, when worker 1 adopts ZD2 strategies to maximize the payoff and other workers follow the strategies of \( \mathbf{p}^1 = [1, 1, 0, 0, 0, 0, 0, 1]^T \) and \( \mathbf{p}^2 = [1, 1, 0, 1, 0, 1, 0, 0, 1]^T \), the payoff of worker 1 can be the most, which can be basically stable around value 2. In addition, the payoff of workers 2 and 3 converge to about 1.45 and 0.65, respectively. The effect of \( \alpha \) on the payoff of worker 1 in Theorem 2
is shown by Figure 5. We set $\beta = -0.3$ with different values of $\alpha$. In this situation, when worker 1 adopts ZD2 strategies, the payoff of worker 1 is basically stable between 1.7 and 2.2. However, the payoff of worker 1 is inversely proportional to the parameter $\alpha$. Next, in Figure 6, the effect of $\beta$ on the payoff of worker 1 in Theorem 2 is shown. We note that under the premise of constant $\alpha$, the smaller the value of $\beta$, the greater the payoff of worker 1. It can be seen that with the change of $\beta$, the payoff of worker 1 is still stable between 1.95 and 2.1, which is in a convergent state. In Figure 7, we compare ZD2 strategies with WSLS and TFT, respectively. We can see that if worker 1 takes ZD2 strategies, while another worker takes TFT strategies, the payoff of worker 1 will reach the maximum value of 1.9.

![Figure 1](image1.png)

Figure 1. The payoff of 3 workers where worker 1 uses the ZD1 strategies.

![Figure 2](image2.png)

Figure 2. The payoff of worker 1 using ZD1 strategies with different $\alpha$. 
For ZD3 strategies in Theorem 3, we performed simulation experiments with the weight factor $\lambda = 0.001$ (see Figures 8 and 9). As shown in Figure 8, when worker 1 adopts ZD3 strategies and other workers follow the strategies of $p^1 = [1, 1, 0, 0, 0, 0, 0, 1]^T$ and $p^2 = [1, 1, 0.1, 0.1, 0.1, 0, 0, 1]^T$, the payoff of worker 1 can be stable around 2.1. Worker 1 obtains the highest payoff among three workers. After about 200 iterations, workers’ payoffs gradually leveled off. By Figure 9, the effect of $\lambda$ on the payoff of worker 1 in Theorem 3 is shown. We set $\beta_1 = -0.005$, $\beta_2 = -0.006$, $\alpha = 0.008$ and different values of $\lambda$. We find that the larger the value of $\lambda$, the greater the payoff of worker 1.

![Figure 3. ZD1 strategies vs. WSLS and TFT.](image)

![Figure 4. The payoff of 3 workers where worker 1 uses the ZD2 strategies.](image)
In conclusion, we find that when worker 1 adopts ZD strategies, his or her payoff is always the highest among three workers. It can be observed that the payoffs of three workers finally converge to stable states (see Figures 1, 4 and 8). When analyzing the effects of parameters on worker 1’s payoff, some figures fail to reach the convergent state due to the uncertainty of parameters. However, these figures show that worker 1’s payoff is basically stable within a range.

**Figure 5.** The payoff of worker 1 using ZD2 strategies with different $\alpha$.

**Figure 6.** The payoff of worker 1 using ZD2 strategies with different $\beta$. 
Figure 7. ZD2 strategies vs WSLS and TFT.

Figure 8. The payoff of 3 workers where worker 1 uses ZD3 strategies.

Remark 4. To further verify the accuracy of our theoretical results, we also conducted some simulation experiments with 4 workers (see Figures 10 and 11). We found that all the results are similar and consistent; therefore, they are not repeated here.
Figure 9. The payoff of worker 1 using ZD3 strategies with different $\lambda$.

Remark 5. In simulations, only three (or four) workers participated in the crowdsourcing process, but the previous theoretical part of this article is also applicable to more workers. All the results verify that when the focused worker adopts the ZD strategy, its benefits can be maximized.
5. Conclusions

In this paper, we consider the optimal decision-making problem for selfish workers with incomplete information in a crowdsourcing system according to the “winner-takes-all” rule. We reformulate crowdsourcing as an iterative game, in which a group of workers complete a task and each worker with a strategy that is relevant not only to itself but also to other workers. We applied ZD strategies to analyze the behavior of workers in crowdsourcing, in which the crowdsourcing system is modeled as a multi-player iterative game. Taking the advantage of ZD strategies that workers can form a linear relationship in the sum of their opponents’ payoffs and their payoffs, we analyze the theoretic conditions to reach the maximum payoff of the focused worker. Finally, we perform numerical simulations to analyze the effect of the parameters on focused players’ payoff and compare the performances of different strategies. The optimization objective of the existing spatial crowdsourcing research is mainly a single objective. However, practical applications often require the joint optimization of several factors. Therefore, we plan to consider joint optimization objectives as well as the consideration of incentives and budgets in future works.

Author Contributions: Conceptualization, J.W.; methodology, C.T.; software, J.W.; validation, J.L. and G.C.; formal analysis, J.W.; investigation, C.T.; resources, C.T.; data curation, J.L.; writing—original draft preparation, J.W.; writing—review and editing, J.W. and C.T.; visualization, J.W.; supervision, G.C.; project administration, C.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the National Natural Science Foundation of China (No. 62103375) and the Zhejiang Provincial Natural Science Foundation of China (No. LY22F030006).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Proof of Theorem 1. According to Equation (10), we have

\[ aE^1 - \gamma = 0. \]  \hspace{1cm} (A1)
i.e.,
\[ E^1 = \frac{\gamma}{\alpha}. \] (A2)

According to Equation (15), we have
\[
\begin{align*}
\begin{cases}
    p^1_{H,n} = 1 + \lambda a(U_{H,n} - aE^1) \\
    p^1_{L,n} = \lambda a(U_{n,n} - aE^1).
\end{cases}
\end{align*}
\] (A3)

When \( \lambda > 0 \), it is assumed that \( a > 0 \). The probabilities are \( 0 \leq p^1_{H,n} \leq 1 \) and \( 0 \leq p^1_{L,n} \leq 1 \). For different \( n \), we have
\[
\begin{align*}
\begin{cases}
    E^1 \leq \min\{U_{1,n},U_{1,n} + \frac{1}{\lambda a}\} \\
    E^1 \geq \max\{U_{1,n},U_{1,n} - \frac{1}{\lambda a}\}.
\end{cases}
\end{align*}
\] (A4)

If \( n = 0 \), then \( U_{1,0} = r - c_H \), \( U_{1,0} = \frac{r}{\lambda} - c_L \), and \( E^1 \leq \min\{\frac{r}{\lambda} - c_L, r - c_H + \frac{1}{\lambda a}\} \). Since \( \frac{r}{\lambda} - c_L < 0 < r - c_H < r - c_H + \frac{1}{\lambda a} \), we have \( E^1 \leq U_{1,0} \). In addition, we can obtain \( E^1 = \max\{U_{1,0},U_{1,0} - \frac{1}{\lambda a}\} = r - c_H \) in the same way. We denote it \( E^1 \geq U_{1,0} \). However, \( U_{1,0} > U_{1,0} \). Obviously, this is a contradiction. As a result, \( E^1 \) does not exist in this case.

If \( n \neq 0 \), then \( U_{1,n} = \frac{r}{\lambda} + c_H \), \( U_{1,n} = -c_L \). Now, we analyze the properties of \( U_{1,n} \). Because \( r > 0 \) and \( c_H > 0 \), \( U_{1,n} \) decreases monotonically with respect to \( n \). When \( n = N - 1 \), \( U_{1,n} \) can obtain the minimum \( \frac{r}{\lambda} - c_2 \). After a comparison with \( U_{1,n} \), we can draw the following conclusions. When \( r \geq N(c_H - c_L) \), \( \min\{U_{1,n},U_{1,n} + \frac{1}{\lambda a}\} = U_{1,n} = -c_L \). When \( r < N(c_H - c_L) \), \( \min\{U_{1,n},U_{1,n} + \frac{1}{\lambda a}\} = U_{1,n} = -c_L \). Because \( E^1 \geq \max\{U_{1,n},U_{1,n} - \frac{1}{\lambda a}\} \), \( U_{1,n} \) can obtain the maximum \( \frac{r}{\lambda} - c_H \). After comparing \( U_{1,n} \) with \( U_{1,n} - \frac{1}{\lambda a} \), we have \( E^1 = U_{1,n} \). Because \( \min\{U_{1,n},U_{1,n} + \frac{1}{\lambda a}\} \geq \max\{U_{1,n},U_{1,n} - \frac{1}{\lambda a}\} \), we reach the following conclusions: when \( r > N(c_H - c_L) \), \( E^1 \) does not exist; when \( 2c_H \leq r \leq \min\{N(c_H - c_L), N/\lambda a\} \), \( E^1 \) exists and
\[
E^1_{\text{max}} = U_{1,n-1} + \frac{1}{\lambda a} = \frac{r}{\lambda} - c_H + \frac{1}{\lambda a}. \] (A5)

Another situation is \( \lambda > 0 \) and \( a < 0 \). Since \( 0 \leq p^1_{H,n} \leq 1 \) and \( 0 \leq p^1_{L,n} \leq 1 \), we have
\[
\begin{align*}
\begin{cases}
    E^1 \leq \min\{U_{1,n},U_{1,n} - \frac{1}{\lambda a}\} \\
    E^1 \geq \max\{U_{1,n},U_{1,n} + \frac{1}{\lambda a}\}.
\end{cases}
\end{align*}
\] (A6)

This situation is similar to the first one. We can easily reach the conclusions after some calculations. At first, we discuss the situation of \( n = 0 \). When \( 0 < r \leq \frac{N}{N-1}(c_H - c_L - \frac{1}{\lambda a}) \),
\[
E^1_{\text{max}} = U_{1,0}; \] (A7)
when \( \frac{N}{N-1}(c_H - c_L - \frac{1}{\lambda a}) < r \leq \frac{N}{N-1}(c_H - c_L - \frac{2}{\lambda a}) \),
\[
E^1_{\text{max}} = U_{1,0} - \frac{1}{\lambda a}. \] (A8)

When \( N(c_H - c_L) < r \leq c_H - c_L - \frac{1}{\lambda a} \), and \( 2 < N < 1 - \frac{1}{(c_H - c_L)\lambda a} \), \( E^1 \) exists and
\[
E^1_{\text{max}} = U_{1,n-1}. \] (A9)

When \( c_H - c_L - \frac{1}{\lambda a} < r \leq \min\{N(c_H - c_L - \frac{1}{\lambda a}), \frac{N}{(1-N)\lambda a}\} \), \( E^1 \) exists and
\[
E^1_{\text{max}} = U_{1,n-1}. \] (A10)
When \( N(c_H - c_L - \frac{1}{\lambda}) \leq r < c_H - c_L - \frac{2}{\lambda} \) and \( N < 1 - \frac{1}{(c_H - c_L)\lambda - 1} \), \( E^3 \) exists and
\[
E^3_{max} = U^1_{1,n} - \frac{1}{\lambda}. \tag{A11}
\]
Thus, the proof is completed. \( \square \)

**Appendix B**

**Proof of Theorem 2.** According to Equation (9), \( \tilde{p}^j \) satisfies
\[
\tilde{p}^j = \lambda \left( \alpha U^1 + \beta \sum_{k=2}^{N} U^k - \gamma 1 \right), \tag{A12}
\]
therefore, worker \( j \) can unilaterally form a linear relationship between all excepted payoffs:
\( \alpha E^1 + \beta \sum_{k=2}^{N} E^k - \gamma = 0 \). After a mathematical transformation, the payoff of worker \( j \) is
\( E^1 = -\frac{\beta}{\alpha} \sum_{k=2}^{N} E^k + \frac{\gamma}{\alpha} \). Without loss of generality, we suppose \( \lambda > 0 \), \( \alpha > 0 \) and \( \beta < 0 \). According to Equation (15),
\[
\begin{align*}
p^j_{H,n} &= 1 + \lambda (\alpha U^1_{H,n} + \beta \sum_{k=2}^{N} U^k_{H,n} - \alpha E^1 - \beta \sum_{k=2}^{N} E^k) \\
p^j_{L,n} &= \lambda (\alpha U^1_{L,n} + \beta \sum_{k=2}^{N} U^k_{L,n} - \alpha E^1 - \beta \sum_{k=2}^{N} E^k).
\end{align*} \tag{A13}
\]

Denote \( W_{H}(n) = \alpha U^1_{H,n} + \beta \sum_{k=2}^{N} U^k_{H,n} \) and \( W_{L}(n) = \alpha U^1_{L,n} + \beta \sum_{k=2}^{N} U^k_{L,n} \), respectively. When \( \lambda > 0 \), as \( 0 \leq p^j_{H,n} \leq 1 \) and \( 0 \leq p^j_{L,n} \leq 1 \), we obtain
\[
\begin{align*}
-1 &\leq \lambda (W_{H}(n) - \alpha E^1 - \beta \sum_{k=2}^{N} E^k) \leq 0 \\
0 &\leq \lambda (W_{L}(n) - \alpha E^1 - \beta \sum_{k=2}^{N} E^k) \leq 1. \tag{A14}
\end{align*}
\]

Using mathematical derivations, we have
\[
\begin{align*}
E^1 &\leq \min \left\{ \frac{1}{\lambda} \left[ W_{H}(n) - \beta \sum_{k=2}^{N} E^k \right], \frac{1}{\lambda} \left[ W_{L}(n) - \beta \sum_{k=2}^{N} E^k \right] \right\} \\
E^1 &\geq \max \left\{ \frac{1}{\lambda} \left[ W_{H}(n) - \beta \sum_{k=2}^{N} E^k \right], \frac{1}{\lambda} \left[ W_{L}(n) - \beta \sum_{k=2}^{N} E^k \right] \right\}.
\end{align*}
\]

First, we discuss the situation of \( n = 0 \). When \( n = 0 \), \( W_{H}(0) = \alpha (r - c_H) - \beta c_L (N - 1) \), and \( W_{L}(0) = \alpha (\frac{r}{\lambda} - c_L) + \beta (\frac{r}{\lambda} - c_L) (N - 1) \). Since \( E^1 \leq \min \left\{ \frac{1}{\lambda} \left[ W_{H}(0) - \beta \sum_{k=2}^{N} E^k + \frac{1}{\lambda} \right], \frac{1}{\lambda} \left[ W_{L}(0) - \beta \sum_{k=2}^{N} E^k \right] \right\} \), comparing \( W_{H}(0) - \beta \sum_{k=2}^{N} E^k + \frac{1}{\lambda} \) with \( W_{L}(0) - \beta \sum_{k=2}^{N} E^k \) is equivalent to comparing \( W_{H}(0) + \frac{1}{\lambda} \) with \( W_{L}(0) \). As a consequence, when \( r > \frac{a N (c_h - c_L) - \frac{N}{2}}{(N-1)(\alpha - \beta)} \), \( \min \left\{ \frac{1}{\lambda} \left[ W_{H}(0) - \beta \sum_{k=2}^{N} E^k + \frac{1}{\lambda} \right], \frac{1}{\lambda} \left[ W_{L}(0) - \beta \sum_{k=2}^{N} E^k \right] \right\} = \frac{1}{\lambda} \left[ W_{L}(0) - \beta \sum_{k=2}^{N} E^k \right] \). We have
\[
E^1 \leq \frac{1}{\lambda} \left[ W_{L}(0) - \beta \sum_{k=2}^{N} E^k \right]. \tag{A16}
\]

Furthermore, when \( 0 < r \leq \frac{a N (c_h - c_L) - \frac{N}{2}}{(N-1)(\alpha - \beta)} \), \( \min \left\{ \frac{1}{\lambda} \left[ W_{H}(0) - \beta \sum_{k=2}^{N} E^k + \frac{1}{\lambda} \right], \frac{1}{\lambda} \left[ W_{L}(0) - \beta \sum_{k=2}^{N} E^k \right] \right\} = \frac{1}{\lambda} \left[ W_{H}(0) - \beta \sum_{k=2}^{N} E^k \right] \). That is,
\[
E^1 \leq \frac{1}{\lambda} \left[ W_{H}(0) - \beta \sum_{k=2}^{N} E^k + \frac{1}{\lambda} \right]. \tag{A17}
\]

On the other hand, \( E^1 \geq \max \left\{ \frac{1}{\lambda} \left[ W_{H}(0) - \beta \sum_{k=2}^{N} E^k \right], \frac{1}{\lambda} \left[ W_{L}(0) - \beta \sum_{k=2}^{N} E^k \right] \right\} \). After some calculations, we have the following conclusion: when \( r > \frac{a N (c_h - c_L) - \frac{N}{2}}{(N-1)(\alpha - \beta)} \),
\[ \max \left\{ \frac{1}{a} [W_H(0) - \beta \sum_{k=2}^{N} E_k], \frac{1}{a} [W_L(0) - \beta \sum_{k=2}^{N} E_k - \frac{1}{\lambda}] \right\} = \frac{1}{a} [W_H(0) - \beta \sum_{k=2}^{N} E_k]. \]

Therefore,
\[ E_1^1 \geq \frac{1}{a} \left[ W_H(0) - \beta N \sum_{k=2}^{N} E_k \right]. \quad (A18) \]

Furthermore, when \( 0 < r \leq \frac{aN(c_H-c_L) - \frac{a}{2}}{(N-1)(\alpha-\beta)} \), \max \left\{ \frac{1}{a} [W_H(0) - \beta \sum_{k=2}^{N} E_k], \frac{1}{a} [W_L(0) - \beta \sum_{k=2}^{N} E_k - \frac{1}{\lambda}] \right\} = \frac{1}{a} [W_L(0) - \beta \sum_{k=2}^{N} E_k - \frac{1}{\lambda}]. \)

That is,
\[ E_1^1 \geq \frac{1}{a} \left[ W_L(0) - \beta N \sum_{k=2}^{N} E_k - \frac{1}{\lambda} \right]. \quad (A19) \]

Based on what has been discussed above, when \( 0 < r \leq \frac{aN(c_H-c_L) - \frac{a}{2}}{(N-1)(\alpha-\beta)} \), according to Equations (A15) and (A17), \( E_{min}^1 < E_{max}^1 \). Thus, when \( \max (0, \frac{aN(c_H-c_L) - \frac{a}{2}}{(N-1)(\alpha-\beta)}) < r < \frac{aN(c_H-c_L) - \frac{a}{2}}{(N-1)(\alpha-\beta)} \),
\[ E_{max}^1 = \frac{1}{a} \left[ W_H(0) - \beta \sum_{k=2}^{N} E_k + \frac{1}{\lambda} \right]. \quad (A20) \]

Furthermore, when \( \frac{aN(c_H-c_L) - \frac{a}{2}}{(N-1)(\alpha-\beta)} < r < \frac{aN(c_H-c_L) - \frac{a}{2}}{(N-1)(\alpha-\beta)} \), according to Equations (A14) and (A16),
\[ E_{max}^1 = \frac{1}{a} \left[ W_L(0) - \beta \sum_{k=2}^{N} E_k \right]. \quad (A21) \]

Secondly, we discuss the situation of \( n \neq 0 \). After some calculations, we have
\[ \left\{ \begin{array}{l} W_H(n) = \frac{ar + \beta n}{\alpha + 1} + \beta (c_L - c_H)n - (ac_H + \beta c_L)(N - 1) \\ W_L(n) = \beta (c_L - c_H)n + \beta r - ac_L - \beta c_L(N - 1). \end{array} \right. \quad (A22) \]

We notice that \( W_H(n) \) and \( W_L(n) \) are functions of \( n \). Since \( \beta < 0 \), and \( c_L - c_H < 0 \), the function \( W_L(n) \) increases with \( n \). Furthermore, the function can obtain a minimum value when \( n = 1 \), i.e.,
\[ \min_{0 \leq n \leq N-1} W_L(n) = W_L(1) = \beta (c_L - c_H) + \beta r - ac_L - \beta c_L(N - 1). \quad (A23) \]

Now, consider another function. The stagnation point of the function can be obtained by deriving the function \( W_H(n) \). Meanwhile, the monotonicity of the function can be verified by taking a derivative. The function \( W_H(n) \) is decreasing first and then increasing. Therefore, \( n = \sqrt{\frac{br - ar}{\beta (c_H - c_L)}} - 1 \) is identified as the minimum point of function \( W_H(n) \). However, \( n \) must be an integer. Therefore, \( n \) is rounded off to the nearest integer \( n_1 \). Therefore, when \( r > \frac{\beta (c_H - c_L)}{\beta - a} \), \( n = n_1 \), the minimum value of \( W_H(n) \) can be obtained, i.e.,
\[ \min_{0 \leq n \leq N-1} W_H(n) = W_H(n_1). \quad (A24) \]

Next, when \( 0 < r \leq \frac{\beta (c_H - c_L)}{\beta - a} \), \( W_H(n) \) is monotonically increasing. Then, \( n = 1 \), \( W_H(n) \) obtains a minimum value, i.e.,
\[ \min_{0 \leq n \leq N-1} W_H(n) = W_H(1). \quad (A25) \]

Compare \( [W_H(n)]_{min} + \frac{1}{a} \) with \( [W_L(n)]_{min} \) first. When \( 0 < r \leq \frac{\beta (c_H - c_L)}{\beta - a} \), we have the following conclusions according to Equations (A21) and (A23).
If \( \frac{2\alpha(c_H-c_L)-\tau}{\alpha-\beta} > \frac{\beta(c_H-c_L)}{\beta-\alpha} \), a minimum does not exist. Therefore, it must satisfy
\[
\frac{2\alpha(c_H-c_L)-\tau}{\alpha-\beta} < \frac{\beta(c_H-c_L)}{\beta-\alpha}.
\]
When \( 0 < r \leq \frac{2\alpha(c_H-c_L)-\tau}{\alpha-\beta} \), \( \min\{\frac{1}{\alpha}[W_H(n) - \beta \sum_{k=2}^{N} E^k + \frac{1}{\lambda}]\} \),
\[
\frac{1}{\alpha}[W_L(n) - \beta \sum_{k=2}^{N} E^k]\]. Therefore,
\[
E^1 \leq \frac{1}{\alpha} \left[ W_L(1) - \beta \sum_{k=2}^{N} E^k \right]. \quad (A26)
\]
Furthermore, when \( \frac{2\alpha(c_H-c_L)-\tau}{\alpha-\beta} < r \leq \frac{\beta(c_H-c_L)}{\beta-\alpha} \), \( \min\{\frac{1}{\alpha}[W_H(n) - \beta \sum_{k=2}^{N} E^k + \frac{1}{\lambda}]\} \),
\[
\frac{1}{\alpha}[W_L(n) - \beta \sum_{k=2}^{N} E^k]\]. Therefore,
\[
E^1 \leq \frac{1}{\alpha} \left[ W_L(1) - \beta \sum_{k=2}^{N} E^k \right]. \quad (A27)
\]
When \( r > \frac{\beta(c_H-c_L)}{\beta-\alpha} \), we have the following conclusions according to Equations (A21) and (A23).
\[
\frac{\beta(c_H-c_L)}{\beta-\alpha} < \frac{n+1}{\alpha-\beta} \left[ \beta(c_H-c_L)(n_1-1) + \alpha(c_H-c_L) - \frac{1}{\lambda} \right] \text{ and } r > \frac{n+1}{\alpha-\beta} \left[ \beta(c_H-c_L)(n_1-1) + \alpha(c_H-c_L) - \frac{1}{\lambda} \right], \min\{\frac{1}{\alpha}[W_H(n) - \beta \sum_{k=2}^{N} E^k + \frac{1}{\lambda}]\}, \frac{1}{\alpha}[W_L(n) - \beta \sum_{k=2}^{N} E^k]\}. \]
We have
\[
E^1 \leq \frac{1}{\alpha} \left[ W_L(1) - \beta \sum_{k=2}^{N} E^k \right]. \quad (A28)
\]
When \( \frac{\beta(c_H-c_L)}{\beta-\alpha} < r \leq \frac{n+1}{\alpha-\beta} \left[ \beta(c_H-c_L)(n_1-1) + \alpha(c_H-c_L) - \frac{1}{\lambda} \right] \), \( \min\{\frac{1}{\alpha}[W_H(n) - \beta \sum_{k=2}^{N} E^k + \frac{1}{\lambda}]\}, \frac{1}{\alpha}[W_L(n) - \beta \sum_{k=2}^{N} E^k]\}. \]
We have
\[
E^1 \leq \frac{1}{\alpha} \left[ W_H(n_1) - \beta \sum_{k=2}^{N} E^k + \frac{1}{\lambda} \right]. \quad (A29)
\]
However, if \( \frac{\beta(c_H-c_L)}{\beta-\alpha} > \frac{n+1}{\alpha-\beta} \left[ \beta(c_H-c_L)(n_1-1) + \alpha(c_H-c_L) - \frac{1}{\lambda} \right] \), \( \min\{\frac{1}{\alpha}[W_H(n) - \beta \sum_{k=2}^{N} E^k + \frac{1}{\lambda}]\}, \frac{1}{\alpha}[W_L(n) - \beta \sum_{k=2}^{N} E^k]\}. \]
We have
\[
E^1 \leq \frac{1}{\alpha} \left[ W_L(1) - \beta \sum_{k=2}^{N} E^k \right]. \quad (A30)
\]
In addition, we need to know the value of \( \beta \sum_{k=2}^{N} E^k \). To this end, the proof is completed.

\[ \Box \]

Appendix C

Proof of Theorem 2. According to the payoff (2), \( W_H(n) = (\beta_2 + \cdots + \beta_{N-n}) \cdot u_{n+1}^k + (\alpha + \beta_{N-n+1} + \cdots + \beta_N) \cdot u_{n+1}^k \) and \( W_L(n) = (\alpha + \beta_2 + \cdots + \beta_{N-n}) \cdot u_{n+1}^k + (\beta_{N-n+1} + \cdots + \beta_N) \cdot u_{n+1}^k \) are obtained. Therefore, when \( \lambda > 0 \), appropriate parameters can be chosen to obtain the maximum value \( \Gamma_{\text{max}} = \min\{W_L(n), W_H(n) + \frac{1}{\lambda}\} \), which is independent of other workers’ strategies. The similarity also applies to the case of \( \lambda < 0 \), in which \( \gamma_{\text{max}} = \max\{W_L(n) - \frac{1}{\lambda}, W_H(n)\} \).

Furthermore, \( \alpha E^1 = \gamma - \sum_{k=2}^{N} \beta_k E^k \) is obtained. Since the value of parameter \( \alpha \) is uncertain, we need to know the minimum value and the maximum value of \( \sum_{k=2}^{N} \beta_k E^k \). Thus, the proof is completed.  \[ \Box \]
References


6. Dorchheimer, J. Collective Intelligence in Design Crowdsourcing. Mathematics 2022, 10, 539. [CrossRef]


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.