Appendix A Proof of Lemma 1

Each ED n has a strategy set $\pi_n = \{q_n, \tau_n\}$. Collectively, all EDs constitute the strategy set $\Sigma = \{\pi_1, \pi_2, \ldots, \pi_N\}$. Each MO adopts a strategy γ_m, forming the strategy set $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_M\}$. For all EDs that choose MO m, they form the set W_m, where $|W_m| = Z$, and they jointly compete for the payoff γ_m. Consequently, we can express the following equations:

$$\begin{cases} u_1 = \frac{b_1 q_1}{\sum_1^Z b_x q_x}\gamma_m - c_1 q_1 \\ u_2 = \frac{b_2 q_2}{\sum_1^Z b_x q_x}\gamma_m - c_2 q_2 \\ \vdots \\ u_Z = \frac{b_Z q_Z}{\sum_1^Z b_x q_x}\gamma_m - c_Z q_Z \end{cases}$$

For a given n, we find the first-order derivative with respect to q_n :

$$\frac{\partial u_n}{\partial q_n} = -\frac{b_n^2 q_n \gamma_m}{\left(\sum_1^Z b_x q_x\right)^2} + \frac{b_n \gamma_m}{\sum_1^Z b_x q_x} - c_n$$

Solving this equation, we can determine the value of q_n as follows:

$$q_n = -\frac{\sum_{x\neq n}^Z b_x q_x}{b_n} + \sqrt{\frac{\gamma_m \left(\sum_{x\neq n} b_x q_x\right)}{b_n c_n}}$$

Multiplying q_n by b_n, we get:

$$b_n q_n = -\sum_{x\neq n}^Z b_x q_x + \sqrt{\frac{\gamma_m b_n \left(\sum_{x\neq n} b_x q_x\right)}{c_n}}$$

Thus, we obtain:

$$\sum b_x q_x = \frac{(Z-1)\gamma_m}{\sum \frac{c_x}{b_x}}$$

Finally, we can deduce q_n :

$$q_n = \frac{(Z-1)\gamma_m \left[b_n \sum \frac{c_x}{b_x} - (Z-1)c_n\right]}{\left(b_n \sum \frac{c_x}{b_x}\right)^2}$$

Appendix B Proof of Lemma 2

Here, we present and prove a lemma: For the ordering $\frac{c_1}{b_1} < \frac{c_2}{b_2} < \ldots < \frac{c_n}{b_n}$, the further ahead an ED is, i.e., the lower the ratio of cost to contribution, the more competitive the ED is. When jobs are arranged in order of this ratio from lowest to highest, the set of workers currently constituted is $\mathcal{W} = \{W_1, \ldots, W_M\}$. When a new ED$e$ (where $\frac{c_e}{b_e}$ is greater than or equal to all the previously scheduled EDs) joins, the optimal choice for all the already scheduled EDs is to not change their current strategy, thus maintaining a Nash stable structure.

Assume ED $e(e > n)$ wants to join the task of some MO. It has two choices. Suppose e gets a higher payoff from set m than from set $m' : u_{em} \geq u_{em'}$. This can be expressed as:

$$u_{em} = \gamma_m \left[1 - \frac{c_e Z_m}{b_e \left(S_m + \frac{c_e}{b_e}\right)}\right]^2 \geq$$

$$u_{em'} = \gamma_{m'} \left[1 - \frac{c_e Z_{m'}}{b_e \left(S_{m'} + \frac{c_e}{b_e}\right)}\right]^2 .$$

We must then prove that an ED n in coalition m has no motivation to join another coalition m'.

To prove the inequality, let's first rewrite it in a more readable form:

$$\gamma_{m'} \left[1 - \frac{c_n Z_{m'}}{b_n \left(\sum_{x \in W_{m'}} \frac{c_x}{b_x} + \frac{c_n}{b_n}\right)}\right]^2 \leq$$

$$\gamma_m \left[1 - \frac{c_n Z_m}{b_n \left(\sum_{x \in W_m} \frac{c_x}{b_x} + \frac{c_e}{b_e}\right)}\right]^2 .$$

We are given the following conditions:

- Monotonicity: $\frac{c_1}{b_1} < \frac{c_2}{b_2} < \cdots < \frac{c_n}{b_n} < \frac{c_e}{b_e}$.
- e gets more payoff from set m than set m': $u_{em} \geq u_{em'}$.

We will use these conditions to prove the inequality step by step.

- Step 1: Divide both sides of the inequality by $\gamma_m \gamma_{m'}$.

$$\frac{1}{\gamma_m} \left[1 - \frac{c_n Z_{m'}}{b_n \left(\sum_{x \in W_{m'}} \frac{c_x}{b_x} + \frac{c_n}{b_n}\right)}\right]^2$$

$$\leq \frac{1}{\gamma_{m'}} \left[1 - \frac{c_n Z_m}{b_n \left(\sum_{x \in W_m} \frac{c_x}{b_x} + \frac{c_e}{b_e}\right)}\right]^2 .$$

- Step 2: Take the square root of both sides.

$$\sqrt{\frac{1}{\gamma_m}} \left[1 - \frac{c_n Z_{m'}}{b_n \left(\sum_{x \in W_{m'}} \frac{c_x}{b_x} + \frac{c_n}{b_n}\right)}\right]$$

$$\leq \sqrt{\frac{1}{\gamma_{m'}}} \left[1 - \frac{c_n Z_m}{b_n \left(\sum_{x \in W_m} \frac{c_x}{b_x} + \frac{c_e}{b_e}\right)}\right] .$$

- Step 3: Subtract from both sides, and multiply both sides by -1. This will reverse the inequality sign.

$$\sqrt{\frac{1}{\gamma_m}}\left[\frac{c_n Z_{m'}}{b_n\left(\sum_{x\in W_{m'}}\frac{c_x}{b_x}+\frac{c_n}{b_n}\right)}\right] \geq \sqrt{\frac{1}{\gamma_{m'}}}\left[\frac{c_n Z_m}{b_n\left(\sum_{x\in W_m}\frac{c_x}{b_x}+\frac{c_e}{b_e}\right)}\right].$$

From the equation above, we can derive:

$$\sqrt{\frac{1}{\gamma'_m}}\left[\frac{Z_m}{\left(\sum_{x\in W_m}\frac{c_x}{b_x}+\frac{c_e}{b_e}\right)}\right] \leq \sqrt{\frac{1}{\gamma_m}}\left[\frac{Z'_m}{\left(\sum_{x\in W'_m}\frac{c_x}{b_x}+\frac{c_e}{b_e}\right)}\right].$$

Because $\frac{c_n}{d_n} < \frac{c_e}{d_e}$, we can use the scaling method. Then, we can get:

$$\sqrt{\frac{1}{\gamma'_m}}\left[\frac{Z_m}{\left(\sum_{x\in W_m}\frac{c_x}{b_x}+\frac{c_e}{b_e}\right)}\right] \leq \sqrt{\frac{1}{\gamma_m}}\left[\frac{Z'_m}{\left(\sum_{x\in W'_m}\frac{c_x}{b_x}+\frac{c_e}{b_e}\right)}\right] \leq \sqrt{\frac{1}{\gamma_m}}\left[\frac{c_n Z_{m'}}{b_n\left(\sum_{x\in W_{m'}}\frac{c_x}{b_x}+\frac{c_n}{b_n}\right)}\right]. \tag{B1}$$

As as result, we have proven that n has no motivation to leave m to join into another m'.

Appendix C Proof of Theorem 1

The uniqueness of Nash equilibrium must meet the following three conditions.

- The strategy sets are convex, bounded, and closed.
- In the strategy space, the utility functions are quasi-concave and continuous.

We can obtain the second derivative of q_n:

$$\frac{\partial^2 u_{nm}}{\partial q_n^2} = -\frac{2b_n^2\gamma_m\sum_{x\neq n} b_x q_x}{\left(\sum_{x\in W_m}\frac{c_x}{b_x}\right)^3} < 0. \tag{C2}$$

This shows that when Γ and Π are both determined, for ED n, its optimal strategy is a concave problem, and there is a unique optimal solution q_n^*.

Meanwhile, we know in Lemma 2 that when selecting tasks in the order of $\frac{c_x}{b_x}$ from small to large, each ED's choice is unique, and other users have no motivation to deviate from their current strategy.

To sum up, for each ED, there is a unique Nash equilibrium in its data provision strategy q and MO selection strategy. Specifically, we use **algorithm 1** to find the unique Nash equilibrium solution.

Appendix D Proof of Lemma 3

We have the utility function:

$$\mathcal{U}_m = \theta_m \log\left(1 + \alpha\left(H + \frac{Z'_m\gamma_m}{\Delta_m}\right)\right) - \gamma_m - D_m\frac{Z'_m\gamma_m}{\Delta_m} - \mathcal{G}$$

Taking the partial derivative with respect to γ_m:

$$\frac{\partial\mathcal{U}_m}{\partial\gamma_m} = \frac{\theta_m\alpha Z'_m}{\Delta_m(1+\alpha(H+\frac{Z'_m\gamma_m}{\Delta_m}))} - 1 - D_m\frac{Z'_m}{\Delta_m}$$

Setting the partial derivative to zero:

$$0 = \frac{\theta_m\alpha Z'_m}{\Delta_m(1+\alpha(H+\frac{Z'_m\gamma_m}{\Delta_m}))} - 1 - D_m\frac{Z'_m}{\Delta_m}$$

Now, let's isolate γ_m:

$$1 + D_m\frac{Z'_m}{\Delta_m} = \frac{\theta_m\alpha Z'_m}{\Delta_m(1+\alpha(C+\frac{Z'_m\gamma_m}{\Delta_m}))}$$

We can cross-multiply to get rid of the fraction:

$$\Delta_m(1+\alpha(H+\frac{Z'_m\gamma_m}{\Delta_m}))(1+D_m\frac{Z'_m}{\Delta_m}) = \theta_m\alpha Z'_m$$

Expanding and simplifying the expression:

$$\Delta_m + \alpha\Delta_m H + \alpha Z'_m\gamma_m + \alpha D_m\frac{Z'^2_m\gamma_m}{\Delta_m} = \theta_m\alpha Z'_m$$

Now, let's isolate γ_m:

$$\gamma_m(Z'_m\alpha + \alpha D_m\frac{Z'^2_m}{\Delta_m}) = \theta_m\alpha Z'_m - \alpha\Delta_m H - \Delta_m$$

Finally, we can solve for γ_m:

$$\gamma_m = \frac{\theta_m\alpha Z'_m - \alpha\Delta_m H - \Delta_m}{Z'_m\alpha + \alpha D_m\frac{Z'^2_m}{\Delta_m}}$$

Appendix E Proof of theorem 2

To prove that every m has a unique solution γ_m to maximize their utility, we need to find the first-order condition (FOC) and second-order condition (SOC) for the optimization problem, assuming we are optimizing over the variable γ_m.

We already have the first-order derivative of the utility function with respect to γ_m from the previous discussion:

$$\frac{d\mathcal{U}_m}{d\gamma_m} = \frac{\theta_m \cdot \alpha(Z_m - 1)}{\Delta_m \left[1 + \alpha \left(\frac{K+(Z_m-1)\gamma_m}{sg}\right)\right]} - 1 - \frac{D_m(Z_m - 1)}{\Delta_m}$$

Now, let's compute the second-order derivative of the utility function with respect to γ_m:

$$\frac{d^2\mathcal{U}_m}{d\gamma_m^2}$$

This can be achieved by taking the derivative of the first-order derivative we derived earlier:

$$\frac{d^2\mathcal{U}_m}{d\gamma_m^2} = \frac{d}{d\gamma_m}\left[\frac{\theta_m \cdot \alpha(Z_m - 1)}{\Delta_m \left[1 + \alpha \left(\frac{K+(Z_m-1)\gamma_m}{sg}\right)\right]} - 1 - \frac{D_m(Z_m - 1)}{\Delta_m}\right]$$

To simplify the expression, let's define:

$$x(\gamma_m) = \frac{K + (Z_m - 1)\gamma_m}{\Delta}$$

So, we now have:

$$\frac{d\mathcal{U}_m}{d\gamma_m} = \frac{\theta_m \cdot \alpha(Z_m - 1)}{\Delta_m[1 + \alpha x(\gamma_m)]} - 1 - \frac{D_m(Z_m - 1)}{\Delta_m}$$

Now, we proceed to compute the second-order derivative:

$$\frac{d^2\mathcal{U}_m}{d\gamma_m^2} = \frac{d}{d\gamma_m}\left[\frac{\theta_m \cdot \alpha(Z_m - 1)}{\Delta_m[1 + \alpha x(\gamma_m)]} - 1 - \frac{D_m(Z_m - 1)}{\Delta_m}\right]$$

Calculating the derivative, we get:

$$\frac{d^2\mathcal{U}_m}{d\gamma_m^2} = -\frac{\theta_m \cdot \alpha^2(Z_m - 1)^2}{\Delta_m^2[1 + \alpha x(\gamma_m)]^2}$$

We can see that the second-order derivative is strictly negative for all values of γ_m. This implies that the utility function is strictly concave with respect to γ_m. So, we can conclude that there is a unique solution γ_m that maximizes the utility for every m.

Appendix F Proof of theorem 3

According to Theorems 1 and 2, this unique SE is evidently valid.

Appendix G Proof of theorme 4

The formation of the coalition is comparable to a series of transfer operations. Following the rules of the coalitional game, each current state Π_c can be transferred to the next state Π_{c+1}, and the Pareto improvement in social utility must be satisfied each time for the transfer to take place. From the initial state Π_0, our algorithm will produce the next transition

$$\Pi_0 \rightarrow \Pi_1 \rightarrow \ldots \rightarrow \Pi_c \rightarrow \Pi_{c+1}, \tag{G3}$$

where the implementation of a shift procedure is indicated by the $\rightarrow$ symbol. Every application of the shift rule generates two possible cases.

As we know, the number of coalitions a client can join is limited and cannot exceed the Bell number limit. Hence, it is inevitable that the transformation sequence will reach termination, culminating in convergence to a specific partition, denoted by Π_f.

Algorithm 1: Computing NE for CDP game.

Input: $\Gamma = \{\gamma_1, \gamma_2, ..., \gamma_M\}, \Pi = \{S_1, S_2, ..., S_k\}, \mathcal{D}_N = \{\mathscr{D}_1, \mathscr{D}_2, ..., \mathscr{D}_N\}, \mathcal{D}_M = \{\mathscr{D}_1, \mathscr{D}_2, ..., \mathscr{D}_M\}, \mathcal{B} = \{b_1, b_2, ..., b_N\}, \mathcal{C} = \{c_1, c_2, ..., c_n\}$

Output: $\Sigma^* = \{\pi_1^*, \pi_2^*, ..., \pi_N^*\}$

sort all of the clients by the $\frac{c_1}{b_1} \leq \frac{c_2}{b_2} \leq ... \leq \frac{c_N}{b_N}$;
for $i \leftarrow 1$ **to** N **do**
 for $j \leftarrow 1$ **to** M **do**
  if $D_{KL}(\mathscr{D}_i, \mathscr{D}_j) \leq \epsilon_1$ **then**
   $W_j = W_j \cup \{i\}$;
   calculate the utility u_{ij} of client i according to equation (11);
   $W_j = W_j \setminus \{i\}$;
  end
 end
 $J = arg \max_j u_{ij}$;
 if $u_{iJ} > 0$ **then**
  $s_i^* = J$;
  $U_i = u_{iJ}$;
 else
  $s_i^* = 0$;
 end
 for $i \leftarrow 1$ **to** N **do**
  if $s_i^* > 0$ **then**
   calculate q_i^* according to equation (11);
  end
 end
end

Algorithm 2: Computing NE for RDP game.

Input: $\Pi = \{S_1, S_2, ..., S_k\}, \mathcal{B} = \{b_1, b_2, ..., b_N\}, \mathcal{C} = \{c_1, c_2, ..., c_n\}$
Output: $\Gamma = \{\gamma_1, \gamma_2, ..., \gamma_M\}$

Initialize $\Gamma \leftarrow \{0, 0, ..., 0\}$;
Flag $\leftarrow$ True;
while *Flag=true* **do**
  Flag $\leftarrow$ False;
  foreach $S_k \in \Pi$ **do**
    foreach $m \in S_k$ **do**
      calculate the optimal reward $\gamma_{m'}$ according to equation (13);
      if $|\frac{\gamma_m - \gamma_{m'}}{\gamma_m}| > \epsilon$ **then**
        Flag $\leftarrow$ True;
      end
    end
  end
end

Algorithm 3: Computing new γ and ξ.

Input: $\Pi = \{S_1, S_2, ..., S_k\}, \mathcal{B} = \{b_1, b_2, ..., b_N\}, \mathcal{C} = \{c_1, c_2, ..., c_n\}$
Output: $\Gamma = \{\gamma_1, \gamma_2, ..., \gamma_M, \xi_1, \xi_2, ..., \xi_M\}$

Initialize $\Gamma \leftarrow \{0, 0, ..., 0\}$;
Set value of err_1, err_2 ;
repeat
  $\xi^{k+1} = \arg\min_\xi L\left(\xi, \gamma^k, z^k, y^k, \lambda^k, \mu^k\right)$;
  $\gamma^{k+1} = \arg\min_\gamma L\left(\xi^{k+1}, \gamma, z^k, y^k, \lambda^k, \mu^k\right)$;
  $z^{k+1} = \arg\min_z L\left(\xi^{k+1}, \gamma^{k+1}, z, y^k, \lambda^k, \mu^k\right)$;
  $y^{k+1} = \arg\min_y L\left(\xi^{k+1}, \gamma^{k+1}, z^{k+1}, y, \lambda^k, \mu^k\right)$;
  $\lambda^{k+1} = \lambda^k + \rho\left(\sum_{\mathrm{m \in S_k}} z_m^{k+1}\right)$;
  $\mu^{k+1} = \mu^k + \rho\left(y^{k+1} - \left(\xi^{k+1} - \gamma^{k+1}\right)\right)$;
  $\sigma_1 = \left\|\sum_{m \in S_k} z_m\right\|_2 + \left\|\sum_{m \in S_k} z_m\right\|_2$;
  $\sigma_2 = \left\|\rho\left(\sum_{m \in S_k} z_m^{k+1} - \sum_{m \in S_k} z_m^k\right)\right\|_2 + \left\|\rho\left(y^{k+1} - y^k\right)\right\|_2$;
until $\sigma_1 \leq err_1$ *and* $\sigma_2 \leq err_2$;

Algorithm 4: The Coalition Formation Algorithm

Input: The set of MOs $\mathcal{M}$

Output: the final stable coalition partition Π_f

$\Pi_0 = \{\{1\}, \{2\}, ..., \{M\}\}$;
repeat
    $Flag \leftarrow false$; $p \leftarrow random(1,3)$;
    if *p=1* **then**
        two random coalition S_k and S_j from the current coalition structure Π_c;
        $m \leftarrow$ random select from S_k;
        if $D_{KL}(\mathscr{D}_m, \mathscr{D}_i) \leq \epsilon_2, \forall i \in S_j$ *and* $\Pi_c' \triangleright \Pi_c$ **then**
            $\Pi_c \leftarrow \Pi_c'$; $Flag \leftarrow true$;
        end
    end
    if *p=2* **then**
        two random coalition S_k and S_j from the current coalition structure Π_c;
        $m, m' \leftarrow$ random select from S_k, S_j;
        if $D_{KL}(\mathscr{D}_m, \mathscr{D}_i) \leq \epsilon_2, \forall i \in S_j$ *and* $D_{KL}(\mathscr{D}_m', \mathscr{D}_i) \leq \epsilon_2, \forall i \in S_k$ *and* $\Pi_c' \triangleright \Pi_c$ **then**
            $\Pi_c \leftarrow \Pi_c'$; $Flag \leftarrow true$;
        end
    end
    if *p=3* **then**
        random coalition S_k from the current coalition structure Π_c;
        $m \leftarrow$ random select from S_k;
        if $\Pi_c' \triangleright \Pi_c$ **then**
            $\Pi_c \leftarrow \Pi_c'$; $Flag \leftarrow true$;
        end
    end
until *Flag=false*;