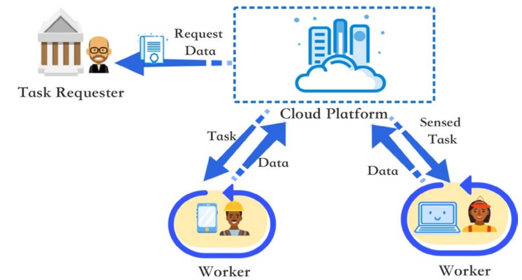
Optimizing Mobile Crowdsourcing Quality with Four-Party Evolutionary Game in Edge Cloud Environment

Haijing Zhang

Department of Computer Science

Content

Background

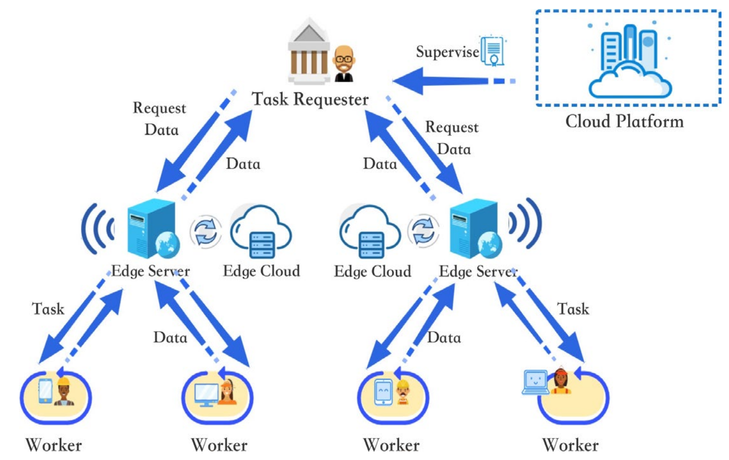


Mobile Crowdsourcing(MCS)	Publishing tasks through mobile devices
Cloud Computing Service	Low cost, high performance, and flexibility Challenges: Service quality, higher demand
The limitations in the current crowdsensing payoff models	- Devoid collusion - Complete rationality - Demand for high network bandwidth, low latency, real-time computing

Motivation

Optimizing MCS quality:

- Four-party evolutionary game model
  - ❖ Mobile Edge Computing (MEC)
- Replicator dynamics approach
  - ❖ Analysis of the strategic equilibrium points
- Incentive mechanism
- Potential collusion scenarios



Approaches

Incentive Mechanism	Material incentive Immaterial incentive
Evolutionary Game Theory	Game theory + dynamic evolutionary processes = dynamic equilibrium
Edge Computing	Offload computing tasks onto the edge server

System Model

- Problem description: self-interested four parties
- Game Model Parameters

	Trustworthy	Untrustworthy
Worker: Data quality	r	1 - r
Cloud platform	m	1 - m
Task requesters: compensation	p	1 - p
Edge Server: control data	g	1 - g

Description of Symbols in the Model

Notation	Description
P_i	The payment that requester pays platform and server
R	Reputation rewards for workers, edge servers, and platforms
S	Reputation loss for workers, edge servers, and platforms
R_h	Workers are compensated for providing high-quality data
R_l	Workers are compensated for providing low-quality data
C_{hi}	The cost incurred when workers provide high-quality data
C_{li}	The cost incurred when workers provide low-quality data
B_{tw}	Cost of collusion between workers and the platform
N_p	Platform regulation cost
B_{tq}	The cost associated with platform collusion with the requester
O_{ij}	Revenue generated for the requester through high-quality data
S_q	Reputation loss for the requesters
R_q	Reputation rewards for the requesters
A_g	Loss incurred due to low-quality data
C_{he}	Costs associated with strict quality control by Edge Servers
C_{le}	Costs associated with poor quality control by Edge Servers

Strategy Analysis

- Expected Revenue Function

➢ Worker expectations:

E_{11} =	$mpg(R_h + R - C_{hi}) + m(1 - p)g(R_h + R - C_{hi}) + mp(1 - g)(R_l + R - C_{hi}) + m(1 - p)(1 - g)(R_l + R - C_{hi}) + (1 - m)pg(R_h + R - C_{hi}) + (1 - m)(1 - p)g(R_l + R - C_{hi}) + (1 - m)p(1 - g)(R_h + R - C_{hi}) + (1 - m)(1 - p)(1 - g)(R_l + R - C_{hi})$
E_{12} =	$mpg(-C_{li} - S) + m(1 - p)g(-C_{li} - S) + mp(1 - g)(-C_{li} - S) + m(1 - p)(1 - g)(-C_{li} - S) + (1 - m)pg(R_h - C_{li} - S - B_{tw}) + (1 - m)(1 - p) g(R_l - C_{li} - S) + (1 - m)p(1 - g)(R_l - C_{li} - S - B_{tw}) + (1 - m)(1 - p)(1 - g)(R_l - C_{li} - S)$
$\bar{E}_1$ =	$rE_{11} + (1 - r)E_{12}$

➢ Worker strategy selection: Replicator dynamic equation

F(r) =	$dr/dt = r(E_{11} - \bar{E}_1) = -r(r - 1)(C_{li} - C_{hi} + R + S + pB_{tw} + mR_l + pR_h - pR_l - mpB_{tw} + mgR_h - mgR_l - gpR_h + pgR_l - mpR_h + mpR_l + mpgR_h - mpgR_l)$

Strategy Analysis

Expected Revenue Function

	Average: E_{i1}- *trust strategy* $E_{i2:}$ *distrust strategy*	Replicator dynamic equation
Platform	$\bar{E}_2 = mE_{21} + (1 - m)E_{22}$	$F(m) = dm/dt = m(E_{21} - \bar{E}_2)$
Task requester	$\bar{E}_3 = pE_{31} + (1 - p)E_{32}$	$F(p) = dp/dt = p(E_{31} - \bar{E}_3)$
Edge server	$\bar{E}_4 = gE_{41} + (1 - g)E_{42}$	$F(g) = dg/dt = g(E_{41} - \bar{E}_4)$

Stability analysis

➢Lyapunov first method

$$J = \begin{bmatrix} J1 & J2 & J3 & J4 \\ J5 & J6 & J7 & J8 \\ J9 & J10 & J11 & J12 \\ J13 & J14 & J15 & J16 \end{bmatrix}$$

$$= \begin{bmatrix} \partial F(r)/\partial r & \partial F(r)/\partial m & \partial F(r)/\partial p & \partial F(r)/\partial g \\ \partial F(m)/\partial r & \partial F(m)/\partial m & \partial F(m)/\partial p & \partial F(m)/\partial g \\ \partial F(g)/\partial r & \partial F(g)/\partial m & \partial F(g)/\partial p & \partial F(g)/\partial g \\ \partial F(p)/\partial r & \partial F(p)/\partial m & \partial F(p)/\partial p & \partial F(p)/\partial g \end{bmatrix}$$

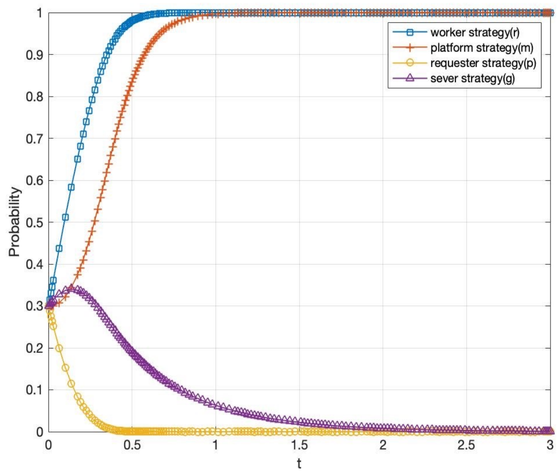
The Eigenvalues of the Jacobian Matrix

Assumption: C_{hi}-C_{li} > B_{tw} +S, R_h-R_l > B_{tq}+S_q, R+S > N_p, and S > R

Equilibrium Point	λ_1	λ_2	λ_3	λ_4	Stability Conclusion	Scenario
E6(1,1,0,0)	$C_{hi} - C_{li} - R - R_l - S$	$N_p - B_{tq} - R - S$	$R_l - R_h - P_i + R_q + S_q$	$C_{le} - C_{he} + R + S$	ESS	1
E11(0,0,1,1)	$C_{li} - C_{hi} + R + S + B_{tw}$	$- N_p + S + B_{tw} - P_i*v$	$R_h - R_l - R_q - S_q$	$P_i - C_{le} - S - P_i*v$	ESS	2
E16(1,1,1,1)	$C_{hi} - C_{li} - R - R_h - S$	$N_p - R - S$	$R_h - R_l - R_q - S_q$	$P_i - C_{le} - S - P_i*v$	ESS	3

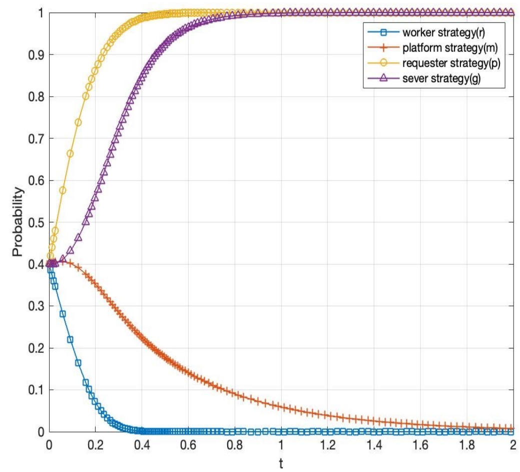
Simulation Experiments

Stability Analysis



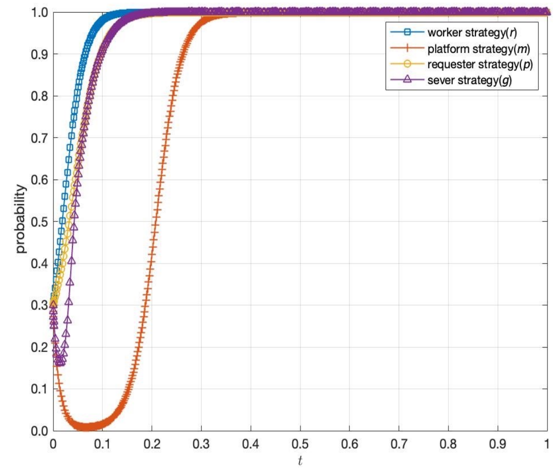
The Evolution of Four Parties in Scenario 1

$r=m=p=g=0.3$



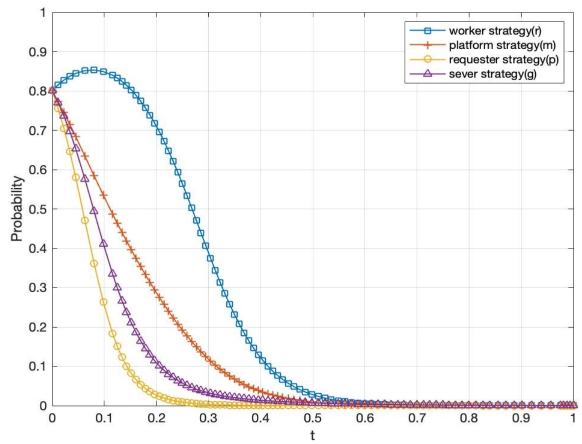
The Evolution of Four Parties in Scenario 2

$r=m=p=g=0.4$



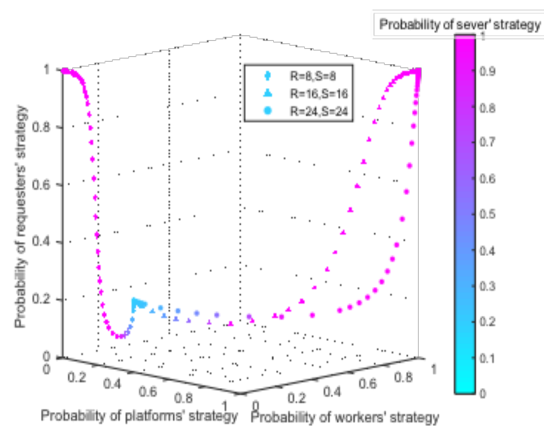
The Evolution of Four Parties in Scenario 3

Impacts of Reward and Punishment



The Evolution of Four Parties without Reward and Punishment

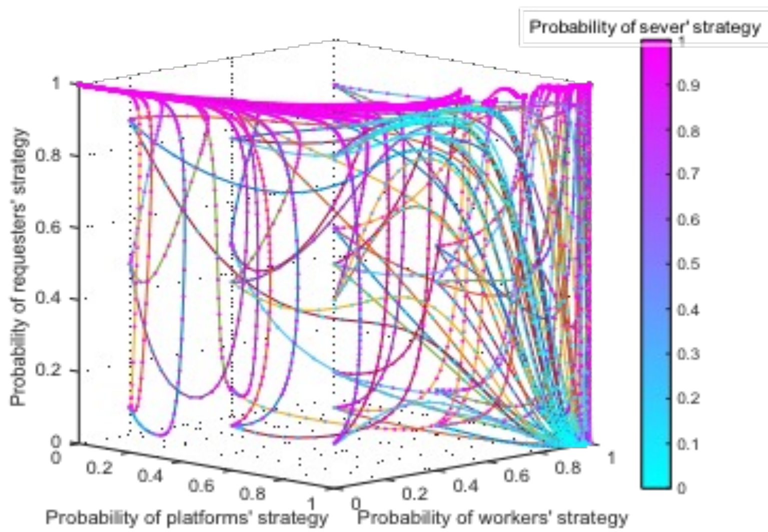
r=m=p=g=0.8



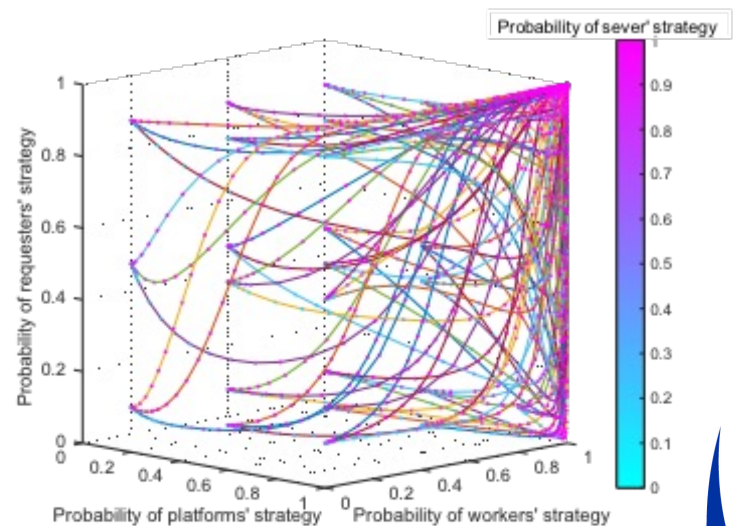
Comparison of Evolution Results without Reward and Punishment Strategies

r=m=p=g=0.2

Equilibrium States with Different Initial Conditions



The stable equilibria: (1,1,0,0) and (0,0,1,1)



The system stability points: (1, 1, 1, 1)

Conclusion

- A Four-Party evolutionary game model is developed
- Computational tasks on edge servers
- Incorporate the potential collusion
- Simulation experiments
- Addressing the issues including dishonesty and false reporting
- Proposed reward and punishment system

Future Work

- Refinement of the incentive mechanism
- Diverse strategic choice
- Enhance the model's adaptability and predictive capabilities

Questions?

Thank You!