Gradient-Guided Credit Assignment and Joint Optimization for Dependency-Aware Spatial Crowdsourcing

Yafei Li, Wei Chen, Jinxing Yan, Huiling Li, Lei Gao, Mingliang Xu

Zhengzhou University, Hong Kong Baptist University

AAAI-25 / IAAI-25 / EAAI-25
FEBRUARY 25 – MARCH 4, 2025 | PHILADELPHIA, USA

Abstract: Dependency-aware spatial crowdsourcing (DASC) addresses the unique challenges posed by subtask dependencies in spatial task assignments. This paper investigates the task assignment problem in DASC and proposes a two-stage Recommend and Match Optimization (RMO) framework, leveraging multi-agent reinforcement learning for subtask recommendation and a multi-dimensional utility function for subtask matching. The RMO framework primarily addresses two key challenges: credit assignment for subtasks with interdependencies and maintaining overall coherence between subtask recommendation and matching. Specifically, we employ meta-gradients to construct auxiliary policies and establish a gradient connection between two stages, which can effectively address credit assignment and joint optimization of subtask recommendation and matching, while concurrently accelerating network training. We further establish a unified gradient descent process through gradient synchronization across recommendation networks, auxiliary policies, and the matching utility evaluation function. Experiments on two real-world datasets validate the effectiveness and feasibility of our proposed approach.

1. Introduction

Recently, the increasing complexity and requirements of spatial tasks have induced the emergence of a sophisticated paradigm in spatial crowdsourcing , namely Dependency-Aware Spatial Crowdsourcing (DASC). DASC is characterized by its intricate task structure and complex execution requirements, further enhancing the challenges and potential applications of traditional Spatial Crowdsourcing. In DASC, spatial tasks consist of multiple interdependent subtasks that require a specific execution order, and each subtask may require different worker skills. Task completion is achieved only when all constituent subtasks are finished, and the platform receives remuneration at this time.

2. Problem Definition

Given a set of workers W and a stream of tasks Q, each task is denoted as a tuple $q = (\mathcal{T}, \mathcal{C})$, while $\mathcal{T}$ represents the set of subtasks, and $\mathcal{C}$ denotes the dependency constraints among these subtasks. The goal of the DASC problem is to find the best matching plan $M \subseteq W \times \mathcal{T}$ such that the total revenue of platform $\mathbb{E}(\mathcal{M})$ is maximized.

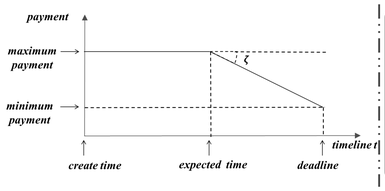


Fig. 1. Payment setting of task

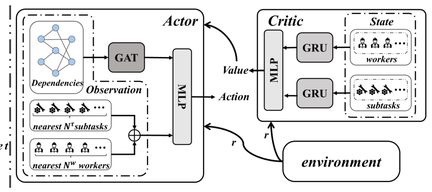


Fig. 2. Task recommendation network structure

3. Method

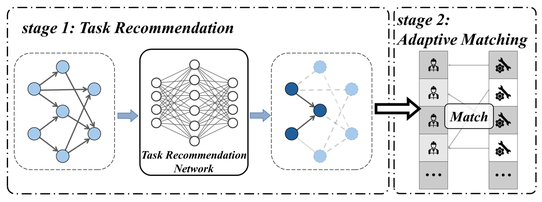


Fig. 3. RMO Framework. Stage 1 employs reinforcementlearning to build recommendation networks for task analysisand subtask recommendation; Stage 2 utilizes adaptive algo-rithms to match workers with the recommended subtasks. Each spatial subtask is treated as an agent.

The task recommendation process is modeled using two distinct Markov DecisionProcesses. **MDP with Sparse Reward:** This MDP is formulated to capture the DASC's base attribute, where payment is received only upon completion of all subtasks. **MDP with Reward Shaping:** This MDP converts revenue into rewards for individual subtasks, differing from the MDP with sparse reward only in its reward structure.

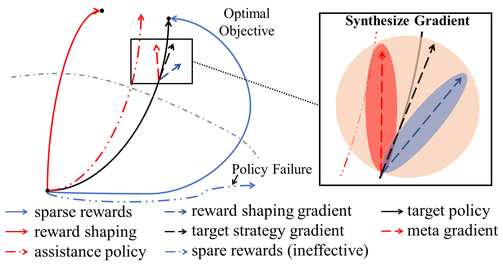


Fig. 4. Collaborative policy optimization implements a two-stage approach: it first leverages the target policy to constrain updates to the reward-shaped policy, then utilizes meta-gradients derived from the reward shaping process to enhance the target policy itself. This bidirectional mechanism not only alleviates training challenges but also expedites learning by establishing an effective feedback loop between the two policy networks.

Collaborative Policy Optimization

The objective function of the policy in PPO:

$$J(\theta) = \mathbb{E}_{a\sim\pi(\cdot|o)} \min(\psi, 1-\epsilon, 1+\epsilon)\, A^n, \qquad \text{where } \psi = \frac{\pi_\theta(a_t|o_t)}{\pi_{\theta_{old}}(a_t|o_t)}$$

- The policy network $\hat{\theta}$ trained by the MDP with reward shaping
- The policy network θ trained by the MDP Sparse Reward

$$\psi = \frac{\pi_{\hat{\theta}}(a_t|o_t)}{\pi_\theta(a_t|o_t)}$$

The meta-gradient can be expressed as

$$\nabla_\theta J(\hat{\theta}) = \nabla_{\hat{\theta}} J(\hat{\theta}) \cdot \nabla_\theta f(\omega, \hat{\theta}, \theta)$$

while $f(\cdot)$ denotes the update step size for θ with ω representing a sequence of experiences, such as $\omega = (S_t, O_t, A_t, \dots)$. For clarity, we use the Stochastic Gradient Descent (SGD) optimizer as an example to illustrate the construction of $f(\cdot)$ hence $\nabla_{\hat{\theta}} J(\hat{\theta})$ is defined as follows:

$$\nabla_\theta f(\omega, \hat{\theta}, \theta) = -\frac{\mu_f}{2} \frac{\partial^2 J(\hat{\theta})}{\partial \hat{\theta} \cdot \partial \theta}$$

The meta-gradient can be expressed as

$$\nabla_\theta J(\hat{\theta}) = \nabla_{\hat{\theta}} J(\hat{\theta}) \cdot \nabla_\theta f(\omega, \hat{\theta}, \theta)$$

while $f(\cdot)$ denotes the update step size for θ with ω representing a sequence of experiences, such as $\omega = (S_t, O_t, A_t, \dots)$. For clarity, we use the Stochastic Gradient Descent (SGD) optimizer as an example to illustrate the construction of $f(\cdot)$ hence $\nabla_{\hat{\theta}} J(\hat{\theta})$ is defined as follows:

$$\nabla_\theta f(\omega, \hat{\theta}, \theta) = -\frac{\mu_f}{2} \frac{\partial^2 J(\hat{\theta})}{\partial \hat{\theta} \cdot \partial \theta}$$

Target policy can be combined by two gradient:

$$\nabla J = \nabla_\theta J(\theta) + \xi \cdot \nabla_\theta J(\hat{\theta})$$

Utility Combination Parameter Optimization

- Skill Compatibility Utility: $u_{skill} = \frac{|K^\tau|}{|K^w|}$
- Distance Utility: $u_{dist} = \frac{L^w - \Delta_{dist}(l^w, l^\tau)}{L^w}$
- Time Remaining Utility: $u_{time} = \frac{1}{e} \cdot e^{\frac{t-t^e}{t^d - t^e}}$

$$u = v_1 \cdot u_{skill} + v_2 \cdot u_{dist} + v_3 \cdot u_{time}$$

$$\eta = \{\alpha, \beta\}, \qquad \begin{cases} v_1 = \alpha \cdot \beta \\ v_2 = \alpha \cdot (1-\beta) \\ v_3 = (1-\alpha) \end{cases}$$

The meta-gradient of parameter η is constructed similarly to collaborative policy optimization:

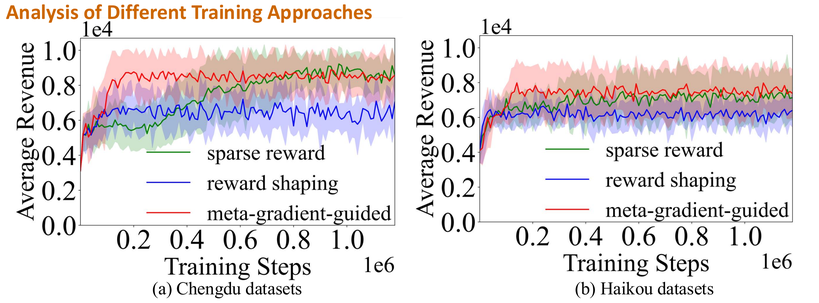
$$\Delta\eta = -\mu_\eta \cdot \nabla_{\hat{\theta}} J(\hat{\theta}) \cdot \nabla_\eta f(\omega, \hat{\theta}, \eta)$$

$\nabla_{\hat{\theta}} J(\hat{\theta})$ is the optimization gradient of the network $\hat{\theta}$ itself. By further derivation (omitted here), we can use the following formula to differentiate $f(\cdot)$ with respect to η:

$$\nabla_\eta f(\omega, \hat{\theta}, \eta) = -\frac{\mu_f}{2} \cdot \frac{\partial}{\partial \hat{\theta}} \left(\frac{\partial J}{\partial A}\right) \cdot \frac{\partial u}{\partial \eta}$$

Experiments

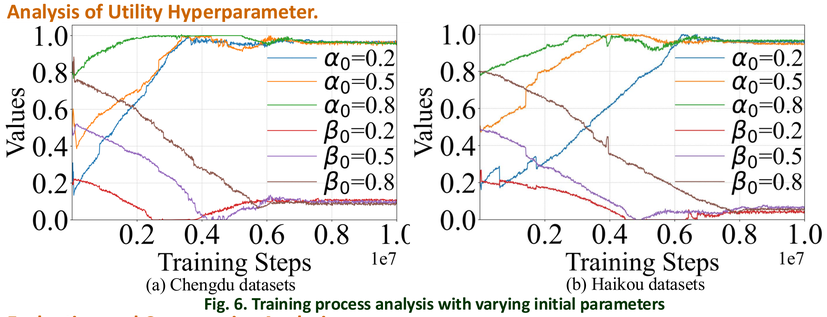
Analysis of Different Training Approaches



(a) Chengdu datasets (b) Haikou datasets

Fig. 5. Comparative analysis: impact of sparse rewards, reward reshaping, and meta-gradient assisted approach

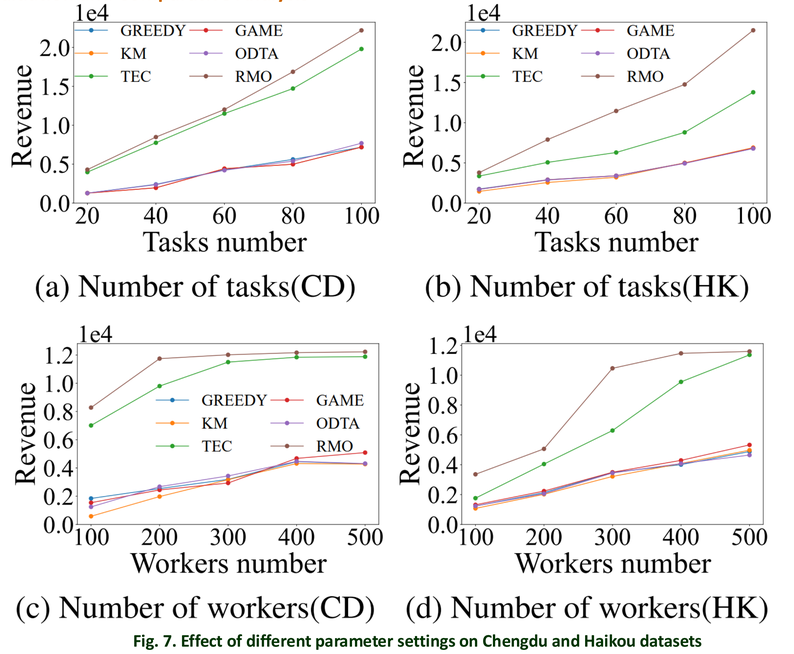
Analysis of Utility Hyperparameter.



(a) Chengdu datasets (b) Haikou datasets

Fig. 6. Training process analysis with varying initial parameters

Evaluation and Comparative Analysis



(a) Number of tasks(CD) (b) Number of tasks(HK)

(c) Number of workers(CD) (d) Number of workers(HK)

Fig. 7. Effect of different parameter settings on Chengdu and Haikou datasets

* Please contact *iexumingliang@zzu.edu.cn* for further information.