Context-Aware Multi-Agent Coordination with Loose Couplings and Repeated Interaction*

Feifei Lin, Xu He, and Bo An

Nanyang Technological University
{linf0012,hexu0003,boan}@ntu.edu.sg

Abstract. Coordination between multiple agents can be found in many areas of industry or society. Despite a few recent advances, this problem remains challenging due to its combinatorial nature. First, with an exponentially scaling action set, it is challenging to search effectively and find the right balance between exploration and exploitation. Second, performing maximization over all agents' actions jointly is computationally intractable. To tackle these challenges, we exploit the side information and loose couplings, i.e., conditional independence between agents, which is often available in coordination tasks. We make several key contributions in this paper. First, the repeated multi-agent coordination problem is formulated as a multi-agent contextual bandit problem to balance the exploration-exploitation trade-off. Second, a novel algorithm called MACUCB is proposed, which uses a modified zooming technique to improve the context exploitation process and a variable elimination technique to efficiently perform the maximization through exploiting the loose couplings. Third, two enhancements to MACUCB are proposed with improved theoretical guarantees. Fourth, we derive theoretical bounds on the regrets of each of the algorithms. Finally, to demonstrate the effectiveness of our methods, we apply MACUCB and its variants to a realistic cloudlet resource rental problem. In this problem, cloudlets must coordinate their computation resources in order to optimize the quality of service at a low cost. We evaluate our approaches on a real-world dataset and the results show that MACUCB and its variants significantly outperform other benchmarks.

Keywords: Multi-agent contextual bandit · Multi-agent coordination · Loose couplings · Cloudlet computing.

1 Introduction

Many real-life problems could be considered as multi-agent coordination problems, which require agents to coordinate their actions repeatedly to optimize a global utility. It is an important issue in multi-agent systems, with a wide range of application domains. Examples include robotic systems [12], traffic light control [22] and maintenance planning [19]. However, the size of the joint action

* Supported by the Alibaba-NTU Singapore Joint Research Institute, Nanyang Technological University.

set scales exponentially with the number of agents. Thus, how to optimally coordinate in repeated settings becomes an extremely challenging task for several reasons: First, as agents are unaware of the expected payoffs associated with different joint actions, they must explore to learn these values. With a large joint action set, inefficient algorithms will spend a large proportion of time in exploring sub-optimal actions, resulting in high regret. Second, it is not trivial to perform the optimization over an exponentially growing action set since computation and storage grow exponentially in the number of agents.

Fortunately, loose couplings exist in many coordination problems, meaning that each agent's action only has a direct impact on a subset of adjacent agents. Therefore, the global utility can break down into local utilities that only depend upon a small subset of agents. In addition, most real-life applications have side information, which can be highly informative of which type of actions should be taken in the future, especially when the action set is very large. Thus, we are interested in multi-agent coordination problems where a set of loosely coupled agents repeatedly observe state (side information) and have to perform actions jointly such that the expected global utility is maximized.

Multi-agent coordination problems have long been of great interest given their importance. Some reinforcement learning studies also consider coordination problems with loose couplings [8, 10, 13, 18]. However, reinforcement learning focuses on sequential decision-making problems while ours is a single-stage setting. Moreover, most model-free learning works only concentrate on empirical results with no theoretical guarantee [13]. Our work is most relevant to [3, 21], which also exploit the loose couplings in multi-agent multi-armed settings. However, their works fail to link side information with rewards of actions, neither do they exploit the similarities across agents. Our learning problem is of a combinatorial nature. In this sense, it is related to combinatorial bandits, which extend the classical multi-armed bandit (MAB) framework by allowing multiple plays at each round [4, 5, 7, 9]. Similar to our settings, Qin et al. [15] study a contextual combinatorial bandits, with semi-bandit feedback [1], where action space grows exponentially and side information as well as the outcomes of all actions played are observable [15]. However, their work does not restrict the set of actions played, makes it not applicable to our problem, where each agent can only play a single action from its own action set. Different from previous works, we exploit the side information and loose couplings to address these issues and provide several key contributions. First, we formulate the repeated multi-agent coordination problem as a multi-agent contextual bandit problem to balance the exploration-exploitation tradeoff in the joint actions of multiple agents. Second, we present a novel algorithm called MACUCB, which combines a modified zooming technique [20] and a variable elimination algorithm [3, 16, 17] to adaptively exploit the context information and address the unavoidable scalability issues in multi-agent settings. Third, we propose two enhancements to our base algorithm with improved bound guarantees. One is to share context space among agents and the other algorithm takes advantage of the full feedback information. Fourth, we show that the regret of MACUCB and its two variants are bounded. Finally,

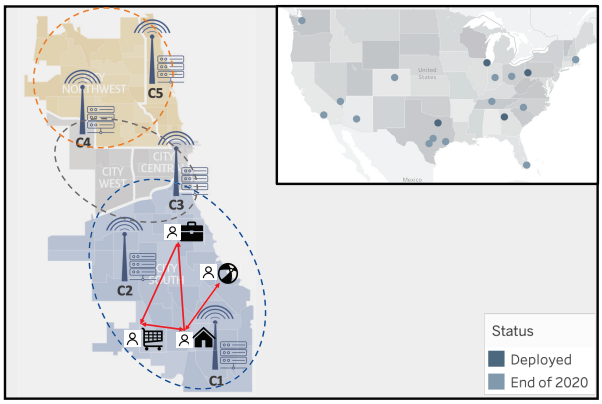


Fig. 1. A map of the United States showing cities with cloudlet infrastructures (upper-right corner). A map of Chicago with cloudlet Deployment locations (left)

we empirically compare our algorithms with other state-of-the-art methods in a cloudlet resource rental problem and show that MACUCB and its variants achieve much lower empirical regret.

2 Motivation Scenario

In this section, we use the cloudlet resource rental problem as a motivating example, while our model can be applied to a variety of multi-agent coordination scenarios.

Although mobile devices are getting more powerful recently, they still fall short to execute complex rich media applications like Pokémon Go. Computing offloading through the cloud is an effective way to solve this problem. However, cloud servers are usually located in the far distance, resulting in high latencies. In such context, cloudlets, deployed geographically near mobile users, have been proposed to provide services with low-latency responses. Foreseeing tremendous opportunities, many companies are expanding their investments in this field. For example, as shown in the upper-right corner of Figure 1, Vapor IO will have its Kinetic Edge live in 20 US metropolitan markets. Now assume Niantic, the application service provider (ASP) of Pokémon Go has decided to rent computation resources to deploy its application in Chicago. Let us see how the user experience of Pokémon Go players in Chicago will be affected by the rental decisions. As depicted in Figure 2, many players require for cloudlet services at the same time. Then, each cloudlet needs to decide how much resources to rent considering the side information e.g. past user demand pattern and the current time.

For example, assume that ASP makes rental decisions as shown in Figure 2. Since the ASP rents sufficient computation resources, the computation tasks of users 1 to 3 are offloaded to the Cloudlet 1, leading to low latency. Therefore, the user experience at Cloudlet 1 is high. However, due to limited or no computing resources rented at Cloudlet 2 and Cloudlet n, the tasks of mobile users 5, $k-2$

to k are rejected. Then these tasks have to be offloaded to Cloud via a macro base station (MBS) through congested backbone Internet (dash line), resulting in high service delay and bad user experience at these cloudlets. Therefore, joint rental decisions at multiple cloudlets have to be carefully decided in order to get excellent overall user experience with a low cost, which is measured in terms of time-saving in task processing.

Note that cloudlets are in fact loosely coupled, thus the global utility can be decomposed into local ones. For example, Figure 1 shows the daily activities of mobile user 1. As can be seen, this user spends most of the time in the blue region. Thus, given that mobile users can only access the nearest cloudlet, he/she will mostly be served by cloudlets (C1, C2, C3) located in that particular region. Therefore, user experience in each region only depends on a small subset of cloudlets.

However, there are lots of challenges involved in this scenario: First, the user demand, which is the main factor determining the benefit of rental decisions, is unknown ahead of time. Second, service demand and resource rental options might be different at different cloudlets as the market size and demand elasticity often vary across geographic locations. Thus, simply treating these cloudlets as a single agent might not work well. Third, the number of joint rental options increase exponentially with the number of cloudlets, making it not trivial to compute the optimal solution.

Fortunately, side information that combines contextual knowledge with historical data could be highly informative in the prediction of the future. Thus, in this paper, we model the scenario as a multi-agent contextual bandit problem and take advantage of the side information and loose couplings to address these challenges.

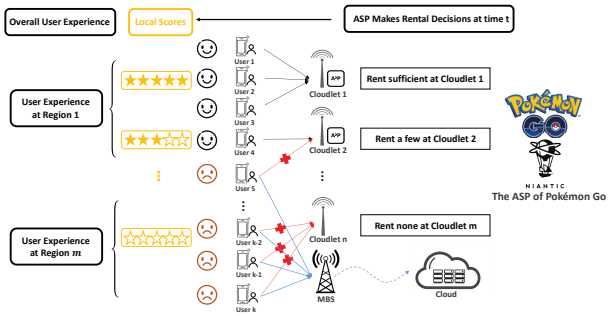


Fig. 2. An example of the cloudlet resource rental problem

3 Problem Description

We consider repeated interactions for a horizon of T rounds and the computation resource rental problem can be modelled as a tuple $\mathcal{N} = \langle \mathcal{C}, \mathcal{X}, \mathcal{A}, \mathcal{S}, \mathcal{F} \rangle$

- $\mathcal{C} = \{i\}$ ($|\mathcal{C}| = n$) is the set of n agents (e.g. cloudlets).
- $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ is the joint context space, which is the cross-product of the context space of individual context space $\mathcal{X}_i = \{x_i\}$. The joint context at time t is denoted by $\mathbf{x}_t = (x_{1,t}, \ldots, x_{n,t})$. In the cloudlet resource rental problem, side information $x_{i,t}$ can be user factor (e.g. past demand patterns), temporal factor (e.g. current time) or other relevant factors related to cloudlet i in round t.
- $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ is the joint action set, defined as the cross-product of the individual action sets $\mathcal{A}_i = \{a_i\}$. The joint action at time t is denoted by $\mathbf{a}_t = (a_{1,t}, \ldots, a_{n,t})$. In the resource rental problem, an action $a_i \in \mathcal{A}_i$ denotes the number of virtual machines rented at cloudlet i.
- $\mathcal{S}$ is the individual score function. For each agent i, $\mathcal{S}_i$ is defined on context and action set, i.e., $\mathcal{S}_i : \mathcal{A}_i \times \mathcal{X}_i \to [0, 1]$. The observed value of the score is denoted by $s(a_{i,t}, x_{i,t})$ and its expected value by $\mu(a_{i,t}, x_{i,t}) = \mathbb{E}\left(s(a_{i,t}, x_{i,t})\right)$. In the motivation scenario, it evaluates the service quality of cloudlet i (in terms of achieved delay reduction) minus rental cost.
- $\mathcal{F}$ measures the expected global utility. In particular, in the motivation scenario, it measures the improvement in service quality minus the cost associated with the rental decisions.

In this paper, we consider multi-agent coordination problems in which the expected global utility function satisfies two properties. Firstly, the expected global utility can be represented as a function of the joint action and the score expectation vector, i.e., $\mathcal{F}(\mathbf{a}, \boldsymbol{\mu})$, where $\boldsymbol{\mu} = \left\{\{\mu(a_i, x_i)\}_{a_i \in \mathcal{A}_i}\right\}_{i \in \mathcal{C}}$ denotes the score expectation vector of all actions of all agents in set $\mathcal{C}$. More specifically, the expected utility at time t is $\mathcal{F}(\mathbf{a}_t, \boldsymbol{\mu}_t) = \mathcal{F}\big(\left\{\mu(a_{i,t}, x_{i,t})\right\}_{i \in \mathcal{C}}\big)$, where $\left\{\mu(a_{i,t}, x_{i,t})\right\}_{i \in \mathcal{C}}$ is an n-dimensional vector restricted on the action $a_{i,t}$ taken by each agent i. For example, as shown in Figure 2, the expected overall user experience is determined by the rental decisions and the corresponding expected local scores obtained at each cloudlet.

Secondly, since agents are loosely coupled, they could be decomposed into m possible overlapping subsets $\mathcal{C}^j$. Correspondingly, the expected global utility could break down into m local expected utility functions f^j, i.e., $\mathcal{F}(\mathbf{a}, \boldsymbol{\mu}) = \sum_{j=1}^{m} f^j(\mathbf{a}^j, \boldsymbol{\mu}^j)$, where $\mathbf{a}^j$ and $\boldsymbol{\mu}^j$ are the local joint action and local score expectation vector respectively. For instance, as mentioned in the motivation scenario, the overall experience can be decomposed into regional ones. Often, the loose couplings structure can be illustrated using a coordination graph (CoG) [10, 13].

In each round t, agents observe the joint context $\mathbf{x}_t$ and are asked to choose a joint action $\mathbf{a}_t$. Once the decision is made, agents observe the local scores $\{s(a_{i,t}, x_{i,t})\}_{i \in \mathcal{C}}$ and receive a global utility. The objective is to maximize the

expected cumulative global utility $\sum_{t=1}^{T} \mathcal{F}(\mathbf{a}_t, \boldsymbol{\mu}_t)$ over T rounds. It is equivalent to minimize the expected cumulative regret Reg_T, defined as the difference in cumulative global utility between the joint actions we selected and the best actions $\mathbf{a}_t^*$, where $\mathbf{a}_t^* \triangleq \text{argmax}_{\mathbf{a}\in\mathcal{A}} \mathcal{F}(\mathbf{a}, \boldsymbol{\mu}_t)$. Let $\mathbf{a}_t^{j^*} = \{a_{i,t}^*\}_{i\in\mathcal{C}^j}$ be the best action restricted on agents in set C^j. Then the objective is to minimize the expected cumulative regret Reg_T

$$Reg_T = \sum_{t=1}^{T} \sum_{j=1}^{m} f^j(\mathbf{a}_t^{j^*}, \boldsymbol{\mu}_t^j) - f^j(\mathbf{a}_t^j, \boldsymbol{\mu}_t^j)$$

Before carrying out our analysis, let us make some natural assumptions about score functions and utility functions.

Lipschitz Score Functions Consider a metric space of context of any agent i $(\mathcal{X}_i, \mathcal{D})$, where $\mathcal{D}$ defines the distance on $\mathcal{X}_i$. The local score function is Lipschitz with respect to metric $\mathcal{D}$. More specifically, $\forall i, \forall a_{i,t} \in \mathcal{A}_i$, and $\forall \mathbf{v}, \mathbf{w} \in \mathcal{X}_i$, score function $\mathcal{S}_i$ satisfies

$$\left|\mathcal{S}_i\left(a_{i,t}, \mathbf{v}\right) - \mathcal{S}_i\left(a_{i,t}, \mathbf{w}\right)\right| \leq \mathcal{D}\left(\mathbf{v}, \mathbf{w}\right) \tag{1}$$

Without loss of generality, we assume that the diameter of $\mathcal{X}_i$ is not more than 1, i.e., $\forall i, \sup_{\mathbf{u},\mathbf{v}\in\mathcal{X}_i} \mathcal{D}(\mathbf{u}, \mathbf{v}) \leq 1$. Consider the motivation scenario. It is natural to assume that, when renting the same number of virtual machines, cloudlets with similar contexts (e.g. demand patterns) have similar score ratings.

On the other hand, the expected utility functions depend on the actual problem instance. It might be simply the sum of the expected scores of actions $\mathbf{a}^j$ taken by agents in $\mathcal{C}^j$, i.e., $f^j(\mathbf{a}_t^j, \boldsymbol{\mu}_t^j) = \sum_{i\in\mathcal{C}^j} \mu(a_{i,t}, x_{i,t})$. It might also be complicated non-linear utilities. We simply assume that the expected reward f^j satisfies the following two assumptions.

Lipschitz Utility Functions The expected local utility functions f^j is Lipschitz continuous with respect to the score expectation vector $\boldsymbol{\mu}^j$. In particular, there exists a universal constant $\alpha > 0$ such that, for $\forall j$ and any two score expectation vector $\hat{\boldsymbol{\mu}}^j$ and $\tilde{\boldsymbol{\mu}}^j$, we have

$$\left|f^j(\mathbf{a}_t^j, \hat{\boldsymbol{\mu}}_t^j) - f^j(\mathbf{a}_t^j, \tilde{\boldsymbol{\mu}}_t^j)\right| \leq \alpha \sum_{i\in\mathcal{C}^j} \left|\hat{\mu}(a_{i,t}, x_{i,t}) - \tilde{\mu}(a_{i,t}, x_{i,t})\right| \tag{2}$$

Take the cloudlet rental problem as an example. It is intuitive that similar score ratings at cloudlets lead to similar regional user experience and vice versa.

Monotonic Utility Functions The expected utility functions f^j is monotone non-decreasing with respect to the score expectation vector $\boldsymbol{\mu}^j$. Formally, for $\forall \mathbf{a}_t^j \in \mathcal{A}^j$, if $\hat{\mu}(a_{i,t}, x_{i,t}) \leq \tilde{\mu}(a_{i,t}, x_{i,t})$ for $\forall i \in \mathcal{C}^j$, we have

$$f^j(\mathbf{a}_t^j, \hat{\boldsymbol{\mu}}_t^j) \leq f^j(\mathbf{a}_t^j, \tilde{\boldsymbol{\mu}}_t^j) \tag{3}$$

The intuition behind the assumption is that the user experience definitely improves when score ratings at all cloudlets become higher. Additionally, it is not necessary for cloudlets to possess a direct knowledge of how the expected local utility functions $f^j\left(\mathbf{a}^j, \boldsymbol{\mu}^j\right)$ are defined. Instead, we assume there is an oracle, which takes joint action $\mathbf{a}$ and expected score $\boldsymbol{\mu}$ as input, and outputs the value of expected utilities $f^j\left(\mathbf{a}^j, \boldsymbol{\mu}^j\right)$.

4 Algorithms

Algorithm 1: MACUCB

for *each agent* i **do**
  for *each action* $a \in \mathcal{A}_i$ **do**
    $B_{i,a} \leftarrow B(o, 1)$ where o is an arbitrary centre
    $n(B_{i,a}) = 0$; $\mathcal{B}_{i,a}^1 \leftarrow \{B_{i,a}\}$
for $t = 1, \ldots, T$ **do**
  for *each agent* i **do**
    Input context $x_{i,t}$
    $Relevant \leftarrow \{B \in \mathcal{B}_i^t : x_{i,t} \in dom_t(B)\}$
    for *each* $B_{i,a} \in Relevant$ **do**
      Calculate B_{rep} by Eq. (4) and update $s_t(B_{i,a})$ and $U_t(B_{i,a})$ by Eq. (5)
    for *each action* $a \in \mathcal{A}_i$ **do**
      Calculate $\hat{B}_{i,a}$ by Eq. (6) and update $\hat{\mu}_t(a, x_{i,t})$ and $U_t(a)$ by Eq. (7)
  Calculate $\boldsymbol{a_t}$ by Eq. (8)
  Execute $\boldsymbol{a_t}$ and observe local scores $\{s(x_{i,t}, a_{i,t})\}_{i \in \mathcal{C}}$
  for *each agent* i **do**
    $n_{t+1}(\hat{B}_{i,a_{i,t}}) = n_t(\hat{B}_{i,a_{i,t}}) + 1$
    $sum(\hat{B}_{i,a_{i,t}}) = sum(\hat{B}_{i,a_{i,t}}) + s(x_{i,t}, a_{i,t})$
    if $conf_{t+1}(\hat{B}_{i,a_{i,t}}) \leq R(\hat{B}_{i,a_{i,t}})$ **then**
      $B^{new} = B_{i,a_{i,t}}\left(x_{i,t}, \frac{1}{2}R(\hat{B}_{i,a_{i,t}})\right)$
      $\mathcal{B}_{i,a_{i,t}}^{t+1} = \mathcal{B}_{i,a_{i,t}}^t \cup B^{new}$; $n_t\left(B^{new}\right) = 0$

4.1 Description of MACUCB

The basic idea of MACUCB is as follows: for each round t, the algorithm maintains a collection of balls $\mathcal{B}_i^t$ for each agent i, which forms a partition of the context space $\mathcal{X}_i$. Basically, each ball is a score estimator and the shape of balls guarantees that all context falling into the partition are within distance r from the centre. Thus, by Eq. (1), we can control the estimation errors by controlling the radius of the balls. Therefore, by generating more balls with smaller radius over time, our estimation becomes more accurate. In detail, when context $x_{i,t}$ arrives, among all the balls whose domain contains $x_{i,t}$, the algorithm selects one ball $\hat{B}_{i,a}$ to estimate the score of each action $a_i \in \mathcal{A}_i$ according to the **estimation rule**. Specifically, the estimation rule selects the ball (estimator) with the highest upper confidence bound. Then, based on the estimation, the algorithm plays the joint action $\mathbf{a}_t$ returned by the **selection rule** which also follows UCB criterion. Then the observed scores are used to update the estimation. In the end, a new ball with a smaller radius may be generated for each agent i according to the **generation rule** to give a refined partition when we are more confident about the estimation.

Now let us introduce some notations and definitions before stating the three rules. $\mathcal{B}_{i,a}^t$ denotes the collection of all balls associated with action a of agent i in round t. Moreover, define $\mathcal{B}_i^t$ as the set containing all balls of agent i in round t and $\mathcal{B}^t$ as the set containing balls of all agents in round t, i.e., $\mathcal{B}^t \triangleq \bigcup_{i\in\mathcal{C}} \mathcal{B}_i^t \triangleq \bigcup_{i\in\mathcal{C}} \left(\bigcup_{a\in\mathcal{A}_i} \mathcal{B}_{i,a}^t\right)$.

For action a of agent i, a ball with center o and radius r is defined by $B_{i,a}(o,r) = \{x \in \mathcal{X}_i : \mathcal{D}(x,o) \leq r\}$. For simplicity, it is abbreviated as $B_{i,a}$ in the subsequent sections. In addition, let $R(B_{i,a})$ denote the radius of ball $B_{i,a}$. Then the domain $\mathrm{dom}_t(B_{i,a})$ of the ball $B_{i,a}$ in round t is defined as a subset of $B_{i,a}$ that excludes all balls $B' \in \mathcal{B}_{i,a}^t$ with a smaller radius, i.e.,$\mathrm{dom}_t(B_{i,a}) \triangleq B_{i,a} \backslash \Big(\bigcup_{B'_{i,a}\in\mathcal{B}_{i,a}^t : R(B'_{i,a}) < R(B_{i,a})} B'_{i,a} \Big)$.

Now we are ready to state the three rules.

Estimation rule: The estimation rule has three steps.

(1) Basic estimation We say that $B_{i,a}$ is a **relevant** ball of agent i in round t if $x_{i,t} \in \mathrm{dom}_t(B_{i,a})$. For each ball $B_{i,a}$, it keeps two estimation statistics: the average score $\bar{s}_t(B_{i,a})$ and the confidence level $conf_t(B_{i,a})$. Let $n_t(B_{i,a})$ denote the number of rounds that $B_{i,a}$ has been selected before t and $sum(B_{i,a})$ be the sum of payoffs from these rounds. Then the average score $\bar{s}_t(B_{i,a})$ and the confidence level $conf_t(B_{i,a})$ are defined as

$$\bar{s}_t(B_{i,a}) \triangleq \frac{sum(B_{i,a})}{n_t(B_{i,a})}$$

$$conf_t(B_{i,a}) \triangleq \sqrt{\frac{4\log T}{1+n_t(B_{i,a})}}$$

(2) Refinement To get a more accurate estimation, we perform a refinement for each relevant ball $B_{i,a}$, using statistics from the **representative ball** B_{rep} which gives an estimation with minimum uncertainty. It is defined as

$$B_{rep} \triangleq \mathrm{argmin}_{B\in\mathcal{B}_{i,a}^t} D(B_{i,a}, B) + conf_t(B) + R(B) \tag{4}$$

Then the refinement is conducted as follows.

$$\begin{aligned} s_t(B_{i,a}) &= \bar{s}_t(B_{rep}) \\ U_t(B_{i,a}) &= D(B_{rep}, B_{i,a}) + conf_t(B_{rep}) \\ &\quad + R(B_{rep}) + R(B_{i,a}) \end{aligned} \tag{5}$$

where $s_t(B_{i,a})$ and $U_t(B_{i,a})$ represent the refined mean and confidence term respectively.

(3) UCB Estimation. After refinement, the ball $\hat{B}_{i,a}$ with the best upper confidence bound is selected to give the final estimation of the score. Specifically, for each action $a \in \mathcal{A}_i$, $\hat{B}_{i,a}$ is defined as

$$\hat{B}_{i,a} \triangleq \mathrm{argmax}_{B_{i,a}\in Relevant} s_t(B_{i,a}) + U_t(B_{i,a}) \tag{6}$$

Then the final estimated mean score $\hat{\mu}_t(a, x_{i,t})$ and confidence term $U_t(a)$ are

$$\hat{\mu}_t(a, x_{i,t}) = s_t(\hat{B}_{i,a}) \text{ and } U_t(a) = U_t(\hat{B}_{i,a}) \tag{7}$$

The corresponding score expectation vector $\hat{\boldsymbol{\mu}}_t$ of all actions of all agents is $\hat{\boldsymbol{\mu}}_t = \left\{\{\hat{\mu}_t(a, x_{i,t})\}_{a\in\mathcal{A}_i}\right\}_{i\in\mathcal{C}}$.
Selection rule: The algorithm selects joint action and balls such that

$$\boldsymbol{a_t} = \text{argmax}_{\boldsymbol{a}\in\mathcal{A}}\, \mathcal{F}\big(\boldsymbol{a}, \hat{\boldsymbol{\mu}}_t\big) + C_t(\boldsymbol{a}) \tag{8}$$

where $\mathcal{F}\big(\boldsymbol{a}, \hat{\boldsymbol{\mu}}_t\big) = \sum_{j=1}^{m} f^j\big(\mathbf{a}^j, \hat{\boldsymbol{\mu}}_t^j\big)$ and $C_t(\boldsymbol{a}) = \alpha \sum_{j=1}^{m} \sum_{i\in\mathcal{C}^j} U_t(a_i)$. Then $\left\{\hat{B}_{i,a_{i,t}}\right\}_{i\in\mathcal{C}}$ are the corresponding balls selected.
Generation rule: For each agent i, if the ball selected $\hat{B}_{i,a_{i,t}}$ satisfies the inequality

$$conf_{t+1}(\hat{B}_{i,a_{i,t}}) \leq R(\hat{B}_{i,a_{i,t}}).$$

Then a new ball B^{new} is generated with center $x_{i,t}$ and radius $\frac{1}{2}R(\hat{B}_{i,a_{i,t}})$. We call $\hat{B}_{i,a_{i,t}}$ the **parent ball** of B^{new}.

The detail of the algorithm is presented in Algorithm 1. However, note that it is not trivial to calculate Eq. (8), since the joint action set $\mathcal{A}$ is exponential in the number of agents. Therefore, MACUCB calls a variable elimination algorithm (VE) [3, 16, 17] to perform this maximization.

4.2 Description of VE

The basic idea of VE is to exploit loose couplings to break down the problem into sub-problems, avoiding searching the whole joint action set. For each agent, we consider the sub-problem containing only local utility functions that have the agent in scope. Then agents are eliminated in sequence by calculating the value of agents' best responses to the neighbors in the sub-problems.

In details, let us rewrite the local utility functions (LUFs) f^j to include both the estimated utility and the confidence terms, i.e.,

$$f^j(\mathbf{a}^j) = f^j(\mathbf{a}^j, \hat{\boldsymbol{\mu}}^j) + \alpha \sum_{i\in\mathcal{C}^j} U(a_i)$$

By loose couplings, we can break down the problem into sub-problems. Let $\mathcal{F}$ be the set containing all LUFs, i.e., $\mathcal{F} \triangleq \{f^j\}_{j=1}^m$ and $\mathcal{F}_i$ be the set of LUFs that have agent i in scope. Then consider a subproblem that only has $\mathcal{F}_i$. Fixing a specific local joint action $\mathbf{a}^{\mathcal{C}^j-\mathbf{i}}$, the possible value $\mathcal{V}_i$ of $\mathcal{F}_i$ for all actions of agent i is

$$\mathcal{V}_i(\mathcal{F}_i, \mathbf{a}^{\mathcal{C}^j-\mathbf{i}}) = \bigcup_{a_i\in\mathcal{A}_i} \sum_{f^j\in\mathcal{F}_i} f^j(\mathbf{a}^{\mathcal{C}^j-\mathbf{i}} \times \{a_i\})$$

Since the action taken by agent i will only affect the global utility through $\mathcal{V}_i$, we can eliminate agent i by calculating the value of agent i 's best response to all joint action its neighbors can take $\mathbf{a}^{\mathcal{C}^j-\mathbf{i}} \in \mathcal{A}^{\mathcal{C}^j-i}$. Then a new LUF $\mathcal{F}_i^{new}$ can be constructed using these values, which depends only on $\mathbf{a}^{\mathcal{C}^j-\mathbf{i}}$, i.e.,

$$\mathcal{F}_i^{new}(\mathbf{a}^{\mathcal{C}^j-\mathbf{i}}) = \max\left(\mathcal{V}_i(\mathcal{F}_i, \mathbf{a}^{\mathcal{C}^j-\mathbf{i}})\right)$$

Then we replace $\mathcal{F}_i$ in $\mathcal{F}$ by the new factor. In addition, since we want to find the optimal joint action, we tag each value in $\mathcal{F}_i^{new}$ with the best response of agent i. We do these series of operations to eliminate all agents $i \in \mathcal{C}$ in a predetermined order q. Details of the algorithm are shown in Algorithm 2.

Algorithm 2: VE

Input A set of local utility functions f^j and an elimination order q containing all agents
$\mathcal{F} = \{f^j\}_{j=1}^m$
while *q is not empty* **do**
  $i \leftarrow q.dequeue()$
  for *each action* $\mathbf{a}^{\mathcal{C}^j - i} \in \mathcal{A}^{\mathcal{C}^j - i}$ **do**
    $\mathcal{F}_i^{new}(\mathbf{a}^{\mathcal{C}^j - i}) = \max\left(\mathcal{V}_i(\mathcal{F}_i, \mathbf{a}^{\mathcal{C}^j - i})\right)$
  end
  $\mathcal{F} \leftarrow (\mathcal{F} \backslash \mathcal{F}_i) \cup \mathcal{F}_i^{new}$
end
$\mathbf{v} \leftarrow \mathcal{V}(\mathcal{F})$
return the tag a^* attached to $\mathbf{v}$

4.3 Extensions

The algorithm proposed for multi-agent contextual bandits can be naturally extended to the special case where agents have similar action sets, contextual spaces and score functions. In such cases, we treat $\mathcal{X} = \bigcup_{i \in \mathcal{C}} \mathcal{X}_i$ and $\mathcal{A} = \bigcup_{i \in \mathcal{C}} \mathcal{A}_i$ and modify MACUCB to allow information sharing between agents by sharing the same adaptive space partition $\mathcal{B}$. The new algorithm is named Multi-agent Similar Contextual Upper Confidence Bound (MASCUCB). For instance, in the cloudlet rental problem, cloudlets usually have similar rental options and score functions (since time delay is determined by demand and computation power regardless of locations). MASCUCB follows the general framework of MACUCB but instead of having an individual partition for each agent, it maintains a public collection of balls for all agents, i.e., $\mathcal{B}^t = \mathcal{B}_1^t = \cdots = \mathcal{B}_n^t$. Then, both the estimation and the update are done with the public collection, which greatly expedites the exploration process and leads to a decrease in regret. The update procedure is shown in Algorithm 3.

Algorithm 3: Update in MASCUCB

for *each agent i* **do**
  Execute MACUCB Lines 16 - 17
  if $conf(\hat{B}_{i,a_{i,t}}) \leq R(\hat{B}_{i,a_{i,t}})$ **then**
    $B^{new} = B_{a_{i,t}}\left(x_{i,t}, \frac{1}{2}R(\hat{B}_{a_{i,t}})\right)$
    $\mathcal{B}_{a_{i,t}}^{t+1} \leftarrow \mathcal{B}_{a_{i,t}}^t \cup B^{new}$; $n_t\left(B^{new}\right) = 0$

For some other applications, full feedback is available, i.e., scores of all actions are available at the end of each round. For example, in the cloudlet resource rental problem, the service demand in each region is revealed at the end of each round.

Then the score, which is the delay reduction minus the rental cost, can be derived accordingly. Therefore, the scores of all rental decisions are observable. We call MASCUCB with full feedback as MASCUCBwF. As outlined in Algorithm 4, MASCUCBwF follows the general framework of MASCUCB, but updates balls for all actions $\{\hat{B}_{i,a}\}_{a_i \in \mathcal{A}_i}$ instead of only updating ball $\hat{B}_{i,a_{i,t}}$.

Algorithm 4: Update in MASCUCBwF

Execute $\boldsymbol{a_t}$ and observe local scores for all actions $\{s(a_i, x_{i,t})\}_{a_i \in \mathcal{A}_i, i \in \mathcal{C}}$
for *each agent* i **do**
  for *each action* a_i *in* $\mathcal{A}_i$ **do**
    $n_{t+1}(\hat{B}_{i,a_i}) \leftarrow n_t(\hat{B}_{i,a_i}) + 1$
    $sum(\hat{B}_{i,a_i}) = sum(\hat{B}_{i,a_i}) + s_t(x_{i,t}, a_i)$
    if $conf(\hat{B}_{i,a_i}) \leq R(\hat{B}_{i,a_i})$ **then**
      $B^{new} \leftarrow B_{i,a_i}\left(x_{i,t}, \frac{1}{2}R(\hat{B}_{i,a_i})\right)$
      $\mathcal{B}_{a_i}^{t+1} \leftarrow \mathcal{B}_{a_i}^{t} \cup B^{new}$
      $n_t(B^{new}) = 0$

5 Regret Analysis

In this section, we provide an upper bound on the cumulative regret for MACUCB.

Define the r-covering number of $(\mathcal{X}_i, D)$ as the minimal number of balls, whose diameters are not greater than r, needed to cover $\mathcal{X}_i$:

$$N_r(\mathcal{X}_i) \triangleq \min\left\{H : \exists V = v_1, \ldots, v_H, \mathcal{X}_i \subset \bigcup_{h=1}^{H} B(v_h, \frac{r}{2})\right\}$$

Then we have the following theorem regarding the expected regret achieved by MACUCB.

Theorem 1. *Assume Eqs.* (1) *and* (2) *hold. With probability at least* $1-2nT^{-2}$*, the expected global regret is bounded by*

$$Reg_T \leq 2\alpha \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} \inf_{r_i' \in (0,1)} \left(7r_i'T + |\mathcal{A}_i| \sum_{r \in \mathbf{R}(r_i', 1)} \frac{28N_r(\mathcal{X}_i)\log T}{r}\right)$$

where α *is the Lipschitz continuity coefficient in Eq.* (2) *and* $\mathbf{R}(a,b) = \{2^{-k} | k \in \mathbb{N} \wedge 2^{-k} \in (a,b]\}$.

Before we prove Theorem 1, let us first propose to bound the difference between the mean score $\mu(a, O_{B_{i,a}})$ of agent i's action a with context $O_{B_{i,a}}$ and the average score $\bar{s}_t(B_{i,a})$ for all balls. Define the **good event** be that

$$\forall t \in [T], \forall i \in \mathcal{C}, \forall a \in \mathcal{A}_i, \forall B_{i,a} \in \mathcal{B}_{i,a}^t,$$
$$\left|\bar{s}_t(B_{i,a}) - \mu(a, O_{B_{i,a}})\right| \leq conf_t(B_{i,a}) + R(B_{i,a})$$

where $O_{B_{i,a}}$ denotes the centre of ball $B_{i,a}$ and $\mu(a, O_{B_{i,a}})$ is the mean score of action a with context $O_{B_{i,a}}$. The following lemma states that the good event happens with high probability.

Lemma 1. *Assume Eqs.* (1) *and* (2) *hold. For* $\forall t \in [T], \forall i \in \mathcal{C}, \forall a \in \mathcal{A}_i, \forall B_{i,a} \in \mathcal{B}_{i,a}^t$*, with probability at least* $1 - 2nT^{-2}$*, we have*

$$\left|\bar{s}_t(B_{i,a}) - \mu(a, O_{B_{i,a}})\right| \leq conf_t(B_{i,a}) + R(B_{i,a})$$

Proof of Lemma 1. Fix a ball $B_{i,a}$. If $n_t(B_{i,a}) = 0$ or $R(B_{i,a}) = 1$, we have

$$\left|\bar{s}_t(B_{i,a}) - \mu(a, O_{B_{i,a}})\right| \leq 1 \leq conf_t(B_{i,a}) + R(B_{i,a})$$

Thus, the inequality always holds if $n_t(B_{i,a}) = 0$ or $R(B_{i,a}) = 1$.

If $n_t(B_{i,a}) \geq 1$ and $R(B_{i,a}) < 1$, by Eq. (1) we have

$$\left|\mathbf{E}\left[\bar{s}_t(B_{i,a})\right] - \mu\left(a, O_{B_{i,a}}\right)\right| \leq \max_{x_{i,t} \in dom_t(B_{i,a})} \mathcal{D}(x_{i,t}, O_{B_{i,a}}) \leq R(B_{i,a})$$

Therefore, according to Hoeffding's inequality,

$$\begin{aligned}
&\Pr\left(\left|\bar{s}_t(B_{i,a}) - \mu(a, O_{B_{i,a}})\right| > conf_t(B_{i,a}) + R(B_{i,a})\right) \\
&\leq \Pr\left(\left|\bar{s}_t(B_{i,a}) - \mu(a, O_{B_{i,a}})\right| - \left|\mathbf{E}\left[\bar{s}_t(B_{i,a})\right] - \mu(a, O_{B_{i,a}})\right| > conf_t(B_{i,a})\right) \\
&\leq \Pr\left(\left|\bar{s}_t(B_{i,a}) - \mathbf{E}\left[\bar{s}_t(B_{i,a})\right]\right| > conf_t(B_{i,a})\right) \\
&\leq 2\exp\left(-2n_t(B_{i,a}) conf_t(B_{i,a})^2\right) \\
&\leq 2\exp\left(-\frac{2n_t(B_{i,a})}{n_t(B_{i,a}) + 1} \cdot 4\log T\right) \\
&\leq 2T^{-4}
\end{aligned}$$

Since each agent will generate at most one new ball in each round, the total number of balls with $n_t(B_{i,a}) \geq 1$ and $R(B_{i,a}) < 1$ is at most nt for any t. To complete the proof, we apply the Union bound over all rounds t and all such balls B,

$$\Pr\left[\text{bad event}\right] \leq T \cdot nT \cdot 2T^{-4} = 2nT^{-2}$$

□

Then, we show that the global regret can be upper bounded by the sum of the confidence of the action taken up to a constant factor.

Lemma 2. *Assume Eqs.* (1) *to* (3) *hold. With probability at least* $1 - 2nT^{-2}$*, the global regret over* T *rounds is bounded by*

$$Reg_T \leq 2\alpha \sum_{t=1}^{T} \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} U_t(a_{i,t})$$

Proof of Lemma 2. Consider the global regret incurred in any round t, by Lipschitz condition, we have

$$\begin{aligned}
Reg_t &= \mathcal{F}\left(\mathbf{a}_t^*, \boldsymbol{\mu}_t\right) - \mathcal{F}\left(\mathbf{a}_t, \boldsymbol{\mu}_t\right) \\
&\leq \mathcal{F}\left(\mathbf{a}_t^*, \hat{\boldsymbol{\mu}}_t\right) - \mathcal{F}\left(\mathbf{a}_t, \boldsymbol{\mu}_t\right) + \alpha \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} \left|\hat{\mu}(a_{i,t}^*, x_{i,t}) - \mu(a_{i,t}^*, x_{i,t})\right|
\end{aligned}$$

Then according to the optimality of $\boldsymbol{a_t}$, we obtain

$$\mathcal{F}\left(\mathbf{a}_t^*, \hat{\boldsymbol{\mu}}_t\right) \leq \mathcal{F}\left(\mathbf{a}_t, \hat{\boldsymbol{\mu}}_t\right) + \alpha \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} U_t(a_{i,t}) - \alpha \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} U_t(a_{i,t}^*)$$

Combining these two inequalities and by Lipschitz condition, we get

$$\begin{aligned} Reg_t &\leq \mathcal{F}\left(\boldsymbol{a_t}, \hat{\boldsymbol{\mu}}_{\boldsymbol{t}}\right) - \mathcal{F}\left(\boldsymbol{a_t}, \boldsymbol{\mu_t}\right) + \alpha \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} U_t(a_{i,t}) - \alpha \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} U_t(a_{i,t}^*) \\ &+ \alpha \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} \left| \hat{\mu}(a_{i,t}^*, x_{i,t}) - \mu(a_{i,t}^*, x_{i,t}) \right| \\ &\leq \alpha \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} \left| \hat{\mu}(a_{i,t}, x_{i,t}) - \mu(a_{i,t}, x_{i,t}) \right| + \alpha \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} U_t(a_{i,t}) - \alpha \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} U_t(a_{i,t}^*) \\ &+ \alpha \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} \left| \hat{\mu}(a_{i,t}^*, x_{i,t}) - \mu(a_{i,t}^*, x_{i,t}) \right| \end{aligned}$$

Now fix a single agent i. For simplicity, we might drop some subscript i in the subsequent equations, but implicitly all terms correspond to agent i. Consider the action $a_{i,t}$ taken and the corresponding ball $\hat{B}_{i,a_{i,t}}$ selected by agent i in round t. Denote the representative ball of $\hat{B}_{i,a_{i,t}}$ as $\hat{B}_{rep}$. By Lemma 1, under good event, we have

$$\left| \bar{s}_t(\hat{B}_{rep}) - \mu(a_{i,t}, O_{\hat{B}_{rep}}) \right| \leq conf_t(\hat{B}_{rep}) + R(\hat{B}_{rep})$$

By Lipschitz condition,

$$\begin{aligned} \left| \mu(a_{i,t}, O_{\hat{B}_{rep}}) - \mu(a_{i,t}, O_{\hat{B}_{i,a_{i,t}}}) \right| &\leq D(\hat{B}_{i,a_{i,t}}, \hat{B}_{rep}) \\ \left| \mu(a_{i,t}, O_{\hat{B}_{i,a_{i,t}}}) - \mu(a_{i,t}, x_{i,t}) \right| &\leq R(\hat{B}_{i,a_{i,t}}) \end{aligned}$$

Combining above inequalities, we obtain

$$\begin{aligned} \left| \bar{s}_t(\hat{B}_{rep}) - \mu(a_{i,t}, x_{i,t}) \right| &\leq conf_t(\hat{B}_{rep}) + D(\hat{B}_{i,a_{i,t}}, \hat{B}_{rep}) + R(\hat{B}_{rep}) + R(\hat{B}_{i,a_{i,t}}) \\ &= U_t(\hat{B}_{i,a_{i,t}}) = U_t(a_{i,t}) \end{aligned}$$

According to Eqs. (5) and (7), $\hat{\mu}_t(a_{i,t}, x_{i,t}) = s_t(\hat{B}_{i,a_{i,t}}) = \bar{s}_t(\hat{B}_{rep})$. Thus, we have

$$\left| \hat{\mu}_t(a_{i,t}, x_{i,t}) - \mu(a_{i,t}, x_{i,t}) \right| \leq U_t(a_{i,t})$$

Similarly, $|\hat{\mu}_t(a_{i,t}^*, x_{i,t}) - \mu(a_{i,t}^*, x_{i,t})| \leq U_t(a_{i,t}^*)$. Therefore, under good event,

$$Reg_t \leq 2\alpha \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} U_t(a_{i,t})$$

The global regret is simply the sum of Reg_t in all T rounds. □

Now let us consider a single agent i and establish an upper bound for $U_t(a_{i,t})$. There are two possibilities. (1) When the ball selected $\hat{B}_{i,a_{i,t}}$ is a parent ball; (2) When $\hat{B}_{i,a_{i,t}}$ is a non-parent ball;

Lemma 3. *Assume Eqs.* (1) *and* (2) *hold. With probability at least* $1 - 2nT^{-2}$, *for* $\forall t \in [T], i \in \mathcal{C}$, $U_t(a_{i,t})$ *is bounded by*

$$U_t(a_{i,t}) \leq 7R(\hat{B}_{i,a_{i,t}})$$

Moreover, if $\hat{B}_{i,a_{i,t}}$ *is a parent ball in round* t, *the bound can be improved to*

$$U_t(a_{i,t}) \leq 3R(\hat{B}_{i,a_{i,t}})$$

Proof of Lemma 3. We use $\hat{B}_{par}$ and $\hat{B}_{rep}$ to denote the parent ball and the representative of $\hat{B}_{i,a_{i,t}}$ respectively. Since $\hat{B}_{rep} = \operatorname{argmin}_{B \in \mathcal{B}^t_{i,a_{i,t}}} D(\hat{B}_{i,a_{i,t}}, B) + conf_t(B) + R(B)$, we have

$$\begin{aligned} U_t(a_{i,t}) = U_t(\hat{B}_{i,a_{i,t}}) &= D(\hat{B}_{i,a_{i,t}}, \hat{B}_{rep}) + conf_t(\hat{B}_{rep}) + R(\hat{B}_{rep}) + R(\hat{B}_{i,a_{i,t}}) \\ &\leq D(\hat{B}_{i,a_{i,t}}, \hat{B}_{par}) + conf_t(\hat{B}_{par}) + R(\hat{B}_{par}) + R(\hat{B}_{i,a_{i,t}}) \end{aligned}$$

By the rule of parent ball, we have

$$\begin{aligned} conf_t(\hat{B}_{par}) &\leq R(\hat{B}_{par}) \\ D(\hat{B}_{i,a_{i,t}}, \hat{B}_{par}) &\leq R(\hat{B}_{par}) \\ R(\hat{B}_{par}) &= 2R(\hat{B}_{i,a_{i,t}}) \end{aligned}$$

Therefore,

$$U_t(a_{i,t}) \leq 3R(\hat{B}_{par}) + R(\hat{B}_{i,a_{i,t}}) = 7R(\hat{B}_{i,a_{i,t}})$$

In cases when $\hat{B}_{i,a_{i,t}}$ is a parent ball, similarly, by the rule of parent ball, we have

$$\begin{aligned} U_t(a_{i,t}) &= D(\hat{B}_{i,a_{i,t}}, \hat{B}_{rep}) + conf_t(\hat{B}_{rep}) + R(\hat{B}_{rep}) + R(\hat{B}_{i,a_{i,t}}) \\ &\leq D(\hat{B}_{i,a_{i,t}}, \hat{B}_{i,a}) + conf_t(\hat{B}_{i,a_{i,t}}) + R(\hat{B}_{i,a_{i,t}}) + R(\hat{B}_{i,a_{i,t}}) \\ &\leq conf_t(\hat{B}_{i,a_{i,t}}) + R(\hat{B}_{i,a_{i,t}}) + R(\hat{B}_{i,a_{i,t}}) \\ &\leq 3R(\hat{B}_{i,a_{i,t}}) \end{aligned}$$

□

To continue with the proof, define $\mathcal{B}^T_{i,a}(r)$ as the collection of balls of radius r in $\mathcal{B}^T_{i,a}$, i.e.,

$$\mathcal{B}^T_{i,a}(r) = \{B \in \mathcal{B}^T_{i,a} | R(B) = r\}$$

Then, let's derive an upper bound for the number of balls with radius r in $\mathcal{B}^T_{i,a}$.

Lemma 4. *For* $\forall i$, $\forall a \in \mathcal{A}_i$ *and* $\forall r = 2^{-k}, k \in \mathbb{N}$,

$$\left|\mathcal{B}_{i,a}^T(r)\right| \leq N_r(\mathcal{X}_i)$$

where $N_r(\mathcal{X}_i)$ *is the* r*-covering number of* $\mathcal{X}_i$

Proof of Lemma 4. First, for any $i \in \mathcal{C}$ and $a \in \mathcal{A}_i$, we show that in $\mathcal{B}_{i,a}^T$, the centers of balls whose radius is r are within distance at least r from one another. Consider $\forall B_{i,a}, B'_{i,a} \in \mathcal{B}_{i,a}^T$ where $R(B_{i,a}) = R(B'_{i,a}) = r$ and $B_{i,a}$ and $B'_{i,a}$ are generated at round t and t' respectively. Without loss of generality, assume that $t < t'$. Let B'_{par} denote the parent of $B'_{i,a}$. Recall that $\text{dom}_t(B'_{par})$ is a subset of B'_{par} that excludes all balls in $\mathcal{B}_{i,a}^t$ with a smaller radius. Thus, $\text{dom}_{t'}(B'_{par}) \cap B_{i,a} = \varnothing$. Moreover, according to the rule of parent, $O_{B'_{i,a}} \in \text{dom}_{t'}(B'_{par})$. As a result, $O_{B'_{i,a}} \notin B_{i,a}$. and therefore we have

$$D(B_{i,a}, B'_{i,a}) \geq r$$

Now we proceed with the proof. Suppose $\left|\mathcal{B}_{i,a}^T(r)\right| = N_r(\mathcal{X}_i)+1$, which means that there are $N_r(\mathcal{X}_i) + 1$ balls $B_1, \ldots, B_{N_r(\mathcal{X}_i)+1}$ of radius r in $\mathcal{B}_{i,a}^T$. Then by pigeonhole, we must have two balls B_m and B_n whose centers fall into the same $B(v_h, \frac{r}{2})$. This means that the distance between the centers of these two balls cannot be more than the diameter of the ball $B(v_h, \frac{r}{2})$, i.e., $D(B_m, B_n) \leq r$, which leads to a contradiction. Therefore, we have

$$\left|\mathcal{B}_{i,a}^T(r)\right| \leq N_r(\mathcal{X}_i)$$

□

Now we are ready to prove the theorem.

Proof of Theorem 1. Let $Reg_T^i = 2\alpha \sum_{t=1}^T U_t(a_{i,t})$. Then we can separate the global regret into individual ones, i.e., $Reg_T \leq \sum_{j=1}^m \sum_{i \in \mathcal{C}^j} Reg_T^i$. Now, consider a single agent i. For any ball $B \in \mathcal{B}_i$, let A_B be a singleton set containing the round when B is activated and S_B denotes the set of rounds when B is selected and is not a parent ball. Then by construction, $\bigcup_{B \in \mathcal{B}_i} \{A_B, S_B\}$ forms a partition of set $[T]$. Moreover, note that in the round t when B is activated, the ball selected $\hat{B}_{i,a}$ must be a parent ball. Thus, if we use $\mathbb{1}\{\cdot\}$ to denote the indicator function, Reg_T^i can be represented as

$$\begin{aligned}
Reg_T^i &= 2\alpha \sum_{t \in [T]} \sum_{r \in \mathbf{R}(0,1)} \sum_{B \in \mathcal{B}_i^T(r)} \mathbb{1}\{t \in A_B \cup S_B\} \cdot U_t(a_{i,t}) \\
&= 2\alpha \overbrace{\sum_{t \in [T]} \sum_{r \in \mathbf{R}(0,r'_i)} \sum_{B \in \mathcal{B}_i^T(r)} \mathbb{1}\{t \in A_B \cup S_B\} \cdot U_t(a_{i,t})}^{(1)} \\
&\quad + 2\alpha \overbrace{\sum_{t \in [T]} \sum_{r \in \mathbf{R}(r'_i,1)} \sum_{B \in \mathcal{B}_i^T(r)} \mathbb{1}\{t \in A_B \cup S_B\} \cdot U_t(a_{i,t})}^{(2)}
\end{aligned}$$

where r_i' can take any value in $(0,1)$. On one hand, by Lemma 3, we can obtain a bound for part (1):

$$(1) \leq \sum_{t\in[T]} \sum_{r\in\mathbf{R}(0,r_i')} \sum_{B\in\mathcal{B}_i^T(r)} \mathbb{1}\{t \in A_B \cup S_B\} \cdot 7r_i' \leq 7r_i'T$$

On the other hand, for part (2):

$$\begin{aligned}(2) &= \sum_{t\in[T]} \sum_{r\in\mathbf{R}(r_i',1)} \sum_{B\in\mathcal{B}_i^T(r)} \mathbb{1}\{t \in A_B\} \cdot U_t(a_{i,t}) + \mathbb{1}\{t \in S_B\} \cdot U_t(a_{i,t}) \\ &\leq \sum_{t\in[T]} \sum_{r\in\mathbf{R}(r_i',1)} \sum_{B\in\mathcal{B}_i^T(r)} \mathbb{1}\{t \in A_B\} \cdot 6R(B) + \mathbb{1}\{t \in S_B\} \cdot 7R(B) \\ &\leq \sum_{r\in\mathbf{R}(r_i',1)} \sum_{B\in\mathcal{B}_i^T(r)} 6r + 7r \cdot \big(\frac{4\log T}{r^2} - 2\big)\end{aligned}$$

For the first inequality, when $t \in |S_B|$, B is the ball selected. Then by Lemma 2 and Lemma 3, we have $U_t(a_{i,t}) \leq 7R(B)$. On the other hand, $t \in A_B$ means that B is activated in time t. Then the ball selected $\hat{B}_{i,a_{i,t}}$ is the parent ball of B. Thus, we have $U_t(a_{i,t}) \leq 3R(\hat{B}_{i,a_{i,t}}) = 6R(B)$. For the second inequality, as defined, B is not a parent ball when $t \in A_B \cup S_B$. Thus, by the rule of parent, we can get an upper bound for the cardinality of $|S_B|$,

$$conf_t(B) = \sqrt{\frac{4\log T}{1+n_t(B)}} > R(B)$$

It means that

$$n_t(B) = \sum_{t\in[T]} \mathbb{1}\{t \in A_B\} + \mathbb{1}\{t \in S_B\} < \frac{4\log T}{R(B)^2} - 1$$

Therefore, we have $|A_B| = 1$ and $|S_B| < \frac{4\log T}{R(B)^2} - 2$.

In addition, based on Lemma 4 and $\mathcal{B}_i = \bigcup_{a\in\mathcal{A}_i} \mathcal{B}_{i,a}$, we have $\big|\mathcal{B}_i^T(r)\big| \leq |\mathcal{A}_i| N_r(\mathcal{X}_i)$. Now we can bound the regret of agent i as follows:

$$\begin{aligned}(2) &\leq \sum_{r\in\mathbf{R}(r_i',1)} \sum_{B\in\mathcal{B}_i^T(r)} 6r + 7r \cdot \big(\frac{4\log T}{r^2} - 2\big) \\ &\leq |\mathcal{A}_i| \sum_{r\in\mathbf{R}(r_i',1)} 6N_r(\mathcal{X}_i)r + 7N_r(\mathcal{X}_i)r \cdot \big(\frac{4\log T}{r^2} - 2\big) \\ &= |\mathcal{A}_i| \sum_{r\in\mathbf{R}(r_i',1)} \frac{28N_r(\mathcal{X}_i)\log T}{r} - 8N_r(\mathcal{X}_i)r\end{aligned}$$

Combining (1) and (2), we get

$$Reg_T \leq 2\alpha \sum_{j=1}^{m} \sum_{i\in\mathcal{C}^j} \Big(7r_i'T + |\mathcal{A}_i| \sum_{r\in\mathbf{R}(r_i',1)} \frac{28N_r(\mathcal{X}_i)\log T}{r} - 8N_r(\mathcal{X}_i)r\Big). \quad .$$

Therefore, we complete the proof.

$$Reg_T \leq 2\alpha \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} \inf_{r_i' \in (0,1)} \left(7r_i' T + |\mathcal{A}_i| \sum_{r \in \mathbf{R}(r_i', 1)} \frac{28 N_r(\mathcal{X}_i) \log T}{r} \right)$$

□

Moreover, define the **covering dimension** d_i for any agent i as

$$d_i \triangleq \inf \left\{ d > 0 : N_r(\mathcal{X}_i) \leq \beta r^{-d_i} \quad \forall r \in (0,1) \right\}.$$

Then substitute $N_r(\mathcal{X}_i) \leq \beta r^{-d_i}$ into the inequality in Theroem 1, we obtain the following corollary.

Corollary 1.

$$Reg_T \leq O(K T^{\frac{d+1}{d+2}} \log T^{\frac{1}{d+2}}) \leq \tilde{O}(T^{\frac{d+1}{d+2}})$$

where $d = \max(d_1, \ldots, d_n)$ *and* $K = \sum_{j=1}^{m} \sum_{i \in \mathcal{C}^j} |\mathcal{A}_i|$.

In addition, the sublinear regret bounds of MASCUCB and MASCUCBwF can be derived in a similar way to Theroem 1.

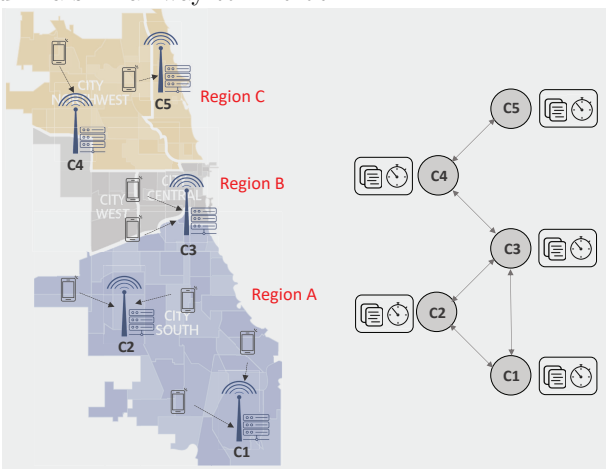


Fig. 3. Loose couplings and context information

6 Experiment

In this section, we evaluate the performance of the proposed algorithms in a real-life scenario.

Table 1. Hyperparameter

Parameter	Value
Time horizon T	5000
2D Context space $Context$	[Demand in last 24 hrs, Current time]
Input data size per task S	1MB
Required CPU cycles per task C	10^9
CPU frequency of each VM F	2×10^9Herts
Price per VM P	0.1 unit
Maximum service demand per VM D_{max}	80
Expected transmission rate of cloudlets R_{c}	5Mbps
Expected transmission rate of the cloud R_{remote}	2Mbps
Expected backbone transmission rate R_b	10Mbps
Processor capacity per task at Cloud center V	10×10^9Herts
Round-trip travel time to the Cloud h_t	1

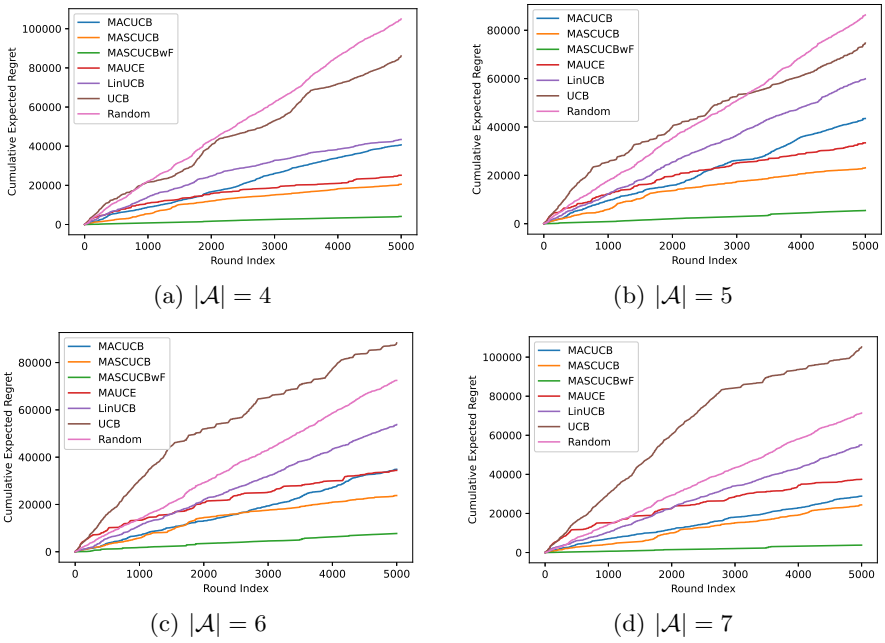
6.1 Experiment Setting

The dataset used is the AuverGrid dataset from the Grid Workloads Archive (GWA) [11]. This dataset records the real-word computational demand received by large-scale multi-site infrastructures to support e-Science. It contains 400k task requests of 5 grids.

The learning problem is formulated as follows. Consider each cloudlet as an agent. There are 5 cloudlets in total. As shown in Figure 3, cloudlets [C1, C2, C3] serve users in region A, [C3, C4] provide services in region B and region C has two cloudlets [C4, C5]. In each round, side-information is observed by each cloudlet. Then rental decisions need to be determined. There are some options available, i.e., $\mathcal{A} = [0, 2, 4, 6]$, corresponding to the number of virtual machines (VM) to rent by each cloudlet. The goal is to maximize ASP's global utility.

More specifically, the score function measures the quality of service (QoS) minus the cost incurred. Herein QoS is measured as the processing time saved by computing at cloudlets instead of the remote Cloud. For each task, the processing time per task at cloudlets equals to transmission delay plus processing delay. Take Cloudlet C1 as an example. If the number of VMs rented at C1 is a_1 with $a_1 > 0$, then the processing time at C1 is $T_1 = \frac{S}{R_{\mathrm{c}}} + \frac{C}{Fa_1}$. In comparison, the processing time at the remote Cloud is $T_{Remote} = \frac{S}{R_{\mathrm{remote}}} + \frac{S}{R_b} + \frac{C}{V} + h_t$. Thus, QoS per task at C1 is $QoS = \Delta t = T_{Remote} - T_1$. In cases when $a_1 = 0$, QoS is set to zero. While the rental cost of VMs is $Cost_1 = Pa_1$. Let d_t be the service demand (number of tasks) covered by C1. Then, given the maximum service demand processed per VM D_{max}, the score achieved by taking action a_1 at C1 is $\mathcal{S}_1 = min(d_t, a_1 D_{max})QoS - Cost_1$. Similar formulation has been considered in [6].

The joint utility in each region can take any form as long as it satisfies Eqs. (2) and (3). In this experiment, we use the Worst Performance Metric to measure the local utility in one region, i.e. $f_a = \min\big(s_1(a_1, x_1), s_2(a_2, x_2), s_3(a_3, x_3)\big)$. Meanwhile, the global utility is the sum of the utilities in all regions $\mathcal{F} = f_a + f_b + f_c$. Please see Table 1 for details about the parameter configurations.



(a) $|\mathcal{A}| = 4$ (b) $|\mathcal{A}| = 5$

(c) $|\mathcal{A}| = 6$ (d) $|\mathcal{A}| = 7$

Fig. 4. Comparison of cumulative regrets between different algorithms

6.2 Experimental Results

To test its performance, we compared MACUCB algorithm and its variants with some classical algorithms:

UCB1 [2]: UCB1 can also be applied to multi-agent cases. The key idea is to treat each possible combination of rental decisions as a different action.

LinUCB [14]: LinUCB makes use of the context information of agents by assuming that the expected utility is defined by the inner product between the context vector and an unknown coefficient vector.

MAUCE [3]: MAUCE is an algorithm for multi-agent multi-armed bandits, which also exploits loose couplings.

MASCUCB: In MASCUCB, agents share the collection of balls for estimation. By sharing historical observations, agents could take advantage of the similarities to make better decisions.

MASCUCBwF: For computing resource rental problem, the scores of other actions are also revealed once we observe the service demand. Thus, MASCUCBwF makes full use of this extra information to adjust the estimation.

Random: The algorithm randomly selects a possible combination of resource rental decisions in each round.

Figure 4 depicts the cumulative regrets incurred by these algorithms under different rental option sets in the first 5000 rounds. It can be seen that MACUCB and its variants significantly outperform other benchmarks across the time period considered when the action set is large. Specifically, the rental options selected to

conduct the experiments are $\mathcal{A} = [0, 2, 4, 6]$, $\mathcal{A} = [0, 2, 4, 6, 8]$, $\mathcal{A} = [0, 2, 4, 6, 8, 10]$ and $\mathcal{A} = [0, 2, 4, 6, 8, 10, 12]$ respectively.

Note that the performance of UCB1 degrades significantly with increasing action set. It is even worse than Random algorithm when $|\mathcal{A}| \geq 6$. This poor performance can be mainly explained by two reasons: Firstly, since UCB1 treats each possible combination of rental decisions as an action, the problem size grows exponentially with the number of agents. As a result, UCB1 needs to spend a large fraction of time in the exploration phase, making it inefficient. Secondly, UCB1 fails to establish a link between contexts and utilities. Although LinUCB considers the context information in the estimation, its performances are worse than MAUCE, MACUCB and its variants. Same as UCB1, due to a large joint action set, LinUCB spends too much time in the exploration phase, preventing it from taking the optimal action frequently. Similar to our algorithm, MAUCE also exploits loose couplings. Although it achieves smaller regrets than MACUCB when the size of the action set is small, the gap narrows with increasing action set. Eventually, MACUCB outperforms MAUCE when $|\mathcal{A}| = 7$. Moreover, since MAUCE fails to exploit the similarities across agents, it performs worse than MASCUCB and MASCUCBwF across all sessions. In addition, it is highly helpful to exploit the similarities between cloudlets, as evidenced by comparing the performance of MASCUCB and MACUCB. Figure 4 shows that MASCUCB outperforms MACUCB in all sessions. Furthermore, comparing MASCUCB and MASCUCBwF, we see that observing more information about scores of actions at cloudlets increases the accuracy of estimation and results in a lower cumulative regret. Since we are using historical data to estimate current scores, uncertainty is always present. The Lipschitz assumption between context and scores only roughly holds. Having more information about real scores of actions could help to correct the bias and reduce the uncertainty to some extent.

7 Conclusion

In this paper, we formulate the multi-agent coordination problem as a multi-agent contextual bandit problem and an online algorithm called MACUCB is proposed to address it. To efficiently perform the maximization in multi-agent settings, MACUCB applies a variable elimination technique to exploit loose couplings. Meanwhile, a modified zooming technique is used in MACUCB to adaptively exploit the context information. Besides, two enhancement methods are proposed which achieve better theoretical and practical performance. One shares the common context space among the agents and the other makes use of full feedback information available. Moreover, sublinear regret bounds were derived for each of the proposed algorithms. Finally, the experiment results on a real-world dataset show that the proposed algorithms outperform other benchmarks.

References

1. Audibert, J.Y., Bubeck, S., Lugosi, G.: Minimax policies for combinatorial prediction games. In: Proceedings of the 24th Annual Conference on Learning Theory. pp. 107–132 (2011)
2. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine learning **47**(2-3), 235–256 (2002)
3. Bargiacchi, E., Verstraeten, T., Roijers, D., Nowé, A., Hasselt, H.: Learning to coordinate with coordination graphs in repeated single-stage multi-agent decision problems. In: International Conference on Machine Learning. pp. 491–499 (2018)
4. Bubeck, S., Cesa-Bianchi, N., et al.: Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends® in Machine Learning **5**(1), 1–122 (2012)
5. Cesa-Bianchi, N., Lugosi, G.: Combinatorial bandits. Journal of Computer and System Sciences **78**(5), 1404–1422 (2012)
6. Chen, L., Xu, J.: Budget-constrained edge service provisioning with demand estimation via bandit learning. IEEE Journal on Selected Areas in Communications **37**(10), 2364–2376 (2019)
7. Chen, W., Wang, Y., Yuan, Y.: Combinatorial multi-armed bandit: General framework and applications. In: International Conference on Machine Learning. pp. 151–159 (2013)
8. De, Y.M., Vrancx, P., Nowé, A.: Learning multi-agent state space representations. In: Proceedings of 9th International Conference of Autonomous Agents and Multiagent Systems. pp. 715–722 (2010)
9. Gai, Y., Krishnamachari, B., Jain, R.: Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. IEEE/ACM Transactions on Networking **20**(5), 1466–1478 (2012)
10. Guestrin, C., Koller, D., Parr, R.: Multiagent planning with factored mdps. In: Advances in neural information processing systems. pp. 1523–1530 (2002)
11. Iosup, A., Li, H., Jan, M., Anoep, S., Dumitrescu, C., Wolters, L., Epema, D.H.: The grid workloads archive. Future Generation Computer Systems **24**(7), 672–686 (2008)
12. Kok, J.R., Spaan, M.T., Vlassis, N., et al.: Multi-robot decision making using coordination graphs. In: Proceedings of the 11th International Conference on Advanced Robotics, ICAR. vol. 3, pp. 1124–1129 (2003)
13. Kok, J.R., Vlassis, N.: Collaborative multiagent reinforcement learning by payoff propagation. Journal of Machine Learning Research **7**(Sep), 1789–1828 (2006)
14. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th international conference on World wide web. pp. 661–670 (2010)
15. Qin, L., Chen, S., Zhu, X.: Contextual combinatorial bandit and its application on diversified online recommendation. In: Proceedings of the 2014 SIAM International Conference on Data Mining. pp. 461–469. SIAM (2014)
16. Roijers, D.M., Whiteson, S., Oliehoek, F.A.: Computing convex coverage sets for faster multi-objective coordination. Journal of Artificial Intelligence Research **52**, 399–443 (2015)
17. Rollón, E., Larrosa, J.: Bucket elimination for multiobjective optimization problems. Journal of Heuristics **12**(4-5), 307–328 (2006)
18. Scharpff, J., Roijers, D.M., Oliehoek, F.A., Spaan, M.T., de Weerdt, M.M.: Solving transition-independent multi-agent mdps with sparse interactions. In: Thirtieth AAAI Conference on Artificial Intelligence. pp. 3174–3180 (2016)

19. Scharpff, J., Spaan, M.T., Volker, L., De Weerdt, M.M.: Planning under uncertainty for coordinating infrastructural maintenance. In: Twenty-Third International Conference on Automated Planning and Scheduling. pp. 169–170 (2013)
20. Slivkins, A.: Contextual bandits with similarity information. The Journal of Machine Learning Research **15**(1), 2533–2568 (2014)
21. Verstraeten, T., Bargiacchi, E., Libin, P.J., Helsen, J., Roijers, D.M., Nowé, A.: Thompson sampling for loosely-coupled multi-agent systems: An application to wind farm control. Adaptive and Learning Agents Workshop 2020, ALA 2020 (2020), https://ala2020.vub.ac.be
22. Wiering, M.: Multi-agent reinforcement learning for traffic light control. In: Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000). pp. 1151–1158 (2000)