Bernd Simon, Andrea Ortiz, Walid Saad, and Anja Klein
”Decentralized Online Learning in Task Assignment Games for Mobile Crowdsensing”, in *IEEE Transactions on Communications*, Vol. 72, Issue 8, August 2024.

Decentralized Online Learning in Task Assignment Games for Mobile Crowdsensing

Bernd Simon*, Andrea Ortiz*, Walid Saad† and Anja Klein*
*Communication Engineering Lab, Technische Universität Darmstadt, Darmstadt, Germany.
†Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Arlington, VA, USA.
Emails: {b.simon, a.ortiz, a.klein}@nt.tu-darmstadt.de, walids@vt.edu.

***Abstract*—The problem of coordinated data collection is studied for a mobile crowdsensing (MCS) system. A mobile crowdsensing platform (MCSP) sequentially publishes sensing tasks to the available mobile units (MUs) that signal their willingness to participate in a task by sending sensing offers back to the MCSP. From the received offers, the MCSP decides the task assignment. A stable task assignment must address two challenges: the MCSP's and MUs' conflicting goals, and the uncertainty about the MUs' required efforts and preferences. To overcome these challenges a novel decentralized approach combining matching theory and online learning, called collision-avoidance multi-armed bandit with strategic free sensing (CA-MAB-SFS), is proposed. The task assignment problem is modeled as a matching game considering the MCSP's and MUs' individual goals while the MUs learn their efforts online. Our innovative "free-sensing" mechanism significantly improves the MU's learning process while reducing collisions during task allocation. The stable regret of CA-MAB-SFS, i.e., the loss of learning, is analytically shown to be bounded by a sublinear function, ensuring the convergence to a stable optimal solution. Simulation results show that CA-MAB-SFS increases the MUs' and the MCSP's satisfaction compared to state-of-the-art methods while reducing the average task completion time by at least $16\,\%$.**

I. Introduction

Mobile devices such as smartphones and wearables are ubiquitous. In fact, by 2025 the number of mobile devices in the world is expected to reach 18.2 billion [1]. As these mobile devices are usually equipped with different sensors, they can be leveraged to collectively perform sensing tasks via mobile crowdsensing (MCS) techniques, e.g., see [2] and [3]. In MCS, a group or "crowd" of mobile units (MUs) performs sensing tasks. Compared with conventional wireless sensor networks, e.g., distributed data collection in Internet of Things applications [4], where sensor devices sequentially sample a physical process, MCS has much lower infrastructure costs, higher coverage, and a wider range of applications due to the mobility of the MUs [5]–[7]. It is, therefore, no surprise that the interest in MCS has steadily increased across academia and industry.

A typical MCS system is composed of one or multiple data requesters, an MCS platform (MCSP), and multiple MUs [7] and [8]. The data requesters submit their sensing requests to the MCSP who acts as the intermediary between the data requesters and the MUs. Particularly, the MCSP converts the sensing requests into sensing tasks, and publishes the tasks to the MUs including information about their type. The MUs independently decide whether to participate or not in each published task. This decision is selfishly and individually made by each MU depending on the effort needed to perform the task and the expected payment from the MCSP [9]. The MUs signal their willingness to participate in a task by sending a sensing offer to the MCSP containing a payment proposal, i.e., the number of monetary units the MU is charging the MCSP for performing the task. Based on the offers of the MUs, the MCSP then decides which task is assigned to each MU by sending them an acknowledgment to their sensing proposal. The revenue of the MCSP depends on its own earnings, i.e, the net payments received from the data requesters for their service after paying the MUs for performing the sensing tasks. The MUs' satisfaction depends on the number of sensing offers that were accepted by the MSCP.

A. Research Challenges

The assignment of the sensing tasks to the requesting MUs is a fundamental problem that will be a key determinant of the success of MCS. This assignment must be able to maximize both the satisfaction of the MUs, and the MCSP revenues [10], such that the MCSP and MUs do not have any incentive to deviate from the chosen task assignment. To achieve this the MCS must overcome two major challenges, as discussed next.

1) Considering multiple utility functions: The first key challenge is that the interests of the MCSP and the MUs are not aligned. Each participant in MCS, including MUs and the MCSP, have their own utility functions with technical and economic components. The MUs want to maximize the payment obtained from the MCSP while minimizing the expounded effort, in terms of energy consumption and completion time. The MCSP maximizes its revenue by assigning tasks to MUs which require a lower payment. Consequently, the MCSP and the MUs may act selfishly to maximize their own revenues.

2) Incomplete information: The second key challenge is that the MUs and the MCSP do not have complete information about the MCS system. This incomplete information spans two components: 1) incomplete information about the tasks and 2) incomplete information about the other participants. Firstly, the effort that an MU must spend to execute a given task is often not known beforehand. For instance, the MUs know the task types from the list of published tasks, but they have to explore how much effort is required to complete the tasks. Moreover, the characteristics of the published tasks and

This work has been funded by the German Research Foundation (DFG) within the Collaborative Research Center (CRC) 1053 MAKI and has been supported by the BMBF project Open6GHub (Nr. 16KISK014). This research was also, in part, supported by the U.S. National Science Foundation under Grant ECR-EDU-2201641.

the MU's conditions, such as the communication rate, change over time depending on factors like the sensing preferences of the data requesters and the mobility of the MUs. Both the task characteristics and the MU's conditions, are therefore appropriately modeled as random processes whose probability distributions are not known a priori. Furthermore, the MCSP does not know the effort that the MUs need to complete the sensing tasks, and the MUs can only measure this effort by executing that particular task.

Secondly, the MUs do not know what task types the other MUs prefer. This may result in colliding sensing offers and unstable assignments. A collision occurs when more than the allowable number of MUs send sensing offers for the same task type. Such concurrent sensing offers occur because the MUs cannot observe each other's sensing offers. Therefore, they are unaware of the effort required by other MUs to perform a task. Collisions should be avoided because they lead to performance degradation as the sensing capabilities of the MUs involved in the collision cannot be used until the next task arrives. In practical MCS systems, these two key research challenges have to be jointly solved because they incorporate the main characteristics of the MCSP and the MUs.

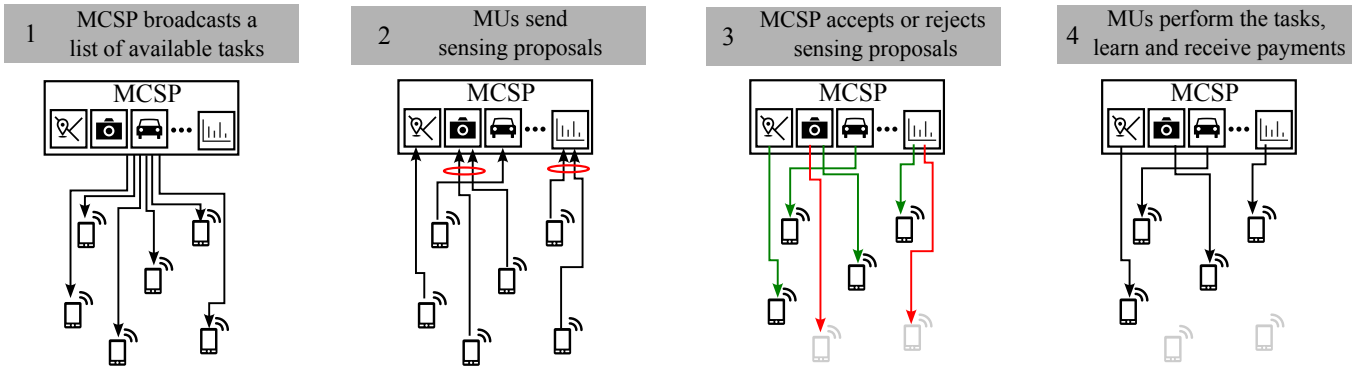
B. Related Works

Prior works [3], [7], [8] and [11]–[22] that attempted to address the aforementioned challenges related to MCS task assignment can be categorized into three directions: i) Optimization approaches, such as in [11] and [12], ii) Game theory approaches, such as in [3], [8] and [13]–[16], and iii) Online learning approaches, such as in [7] and [17]–[23]. Although the authors in [11] and [12] find optimal allocation policies that maximize the MCSP's utility, the MU's utility functions are not considered. We argue that this limitation to a single utility function is not realistic. Moreover, it requires complete non-causal information about the MCS system.

Following a game theory approach, the authors in [3] investigate an optimal incentive mechanism for the MCS using a two-stage Stackelberg game. Their goal is to efficiently recruit MUs to perform the available sensing tasks while assuming payments to be fixed in advance. In [8], the MU's effort is assumed to depend on its location. The authors propose a privacy-preserving approach to obtain information about the MU's location and thus, estimate the MU's efforts. The authors in [13] use matching theory to balance the preferences of the MCSP and the MUs while assuming the payments by the MCSP are fixed in advance. Similarly, assuming known preferences for the MCSP and the MUs, the authors in [14] formulate a two-stage matching problem to maximize the coverage in a MCS system. Following a social welfare maximization approach, the authors in [15] propose an auction-based method to balance the MCSP and MU's interests when assigning the sensing tasks. In [16], the authors propose a stable matching approach for task assignment to incorporate the MUs' and MCSP's preferences. The use of these game-theory-based approaches allows the consideration of the conflicting goals of the MCSP and MUs. However, similar to the optimization approaches [11] and [12], the game theory approaches [3], [8] and [13]–[16] are subjected to the strict requirement that information about the MUs' costs and/or payment requests is known in advance. This requirement makes these approaches infeasible in practical systems, as the tasks' characteristics and the effort to complete tasks are not known a priori and may change over time. Additionally, these related works on game theory rely on a deterministic task model. However, in real-world applications, the effort related to a task is best described by a probability distribution. The effort required for a task might vary because of different influences such as, for example, the weather or the time of the day.

The problem of task assignment under unknown MU efforts is investigated in [17]–[22]. In [17], the authors propose a location-prediction-based online task assignment strategy in which the MU's effort depends on its location in a mobile social network. In [18], Lyapunov optimization is used to derive a task assignment policy that maximizes the gain of the MCSP. The authors in [19] propose prediction methods to estimate the MU's effort at the MCSP. The task pricing problem in a point-to-point MCS system is considered in [20], where a two-stage mean field approximation Stackelberg differential game is used to model the MCSP-MU interaction. Combinatorial multi-armed bandits are considered in [21] to maximize the expected quality of the data received at the MSCP. The authors of [22] propose a federated reinforcement learning approach for the task assignment for sequentially arriving tasks in MCS. This approach aims to maximize a completion ratio of tasks given limited energy of mobile devices. In [23], the authors propose a decentralized multi-agent reinforcement learning approach to learn the task assignment with minimal communication overhead. Even though the solutions in [17]–[23] overcome the requirement of complete non-causal information about the MCS, they are limited to a single utility function, i.e., they only consider either the MCSP's or the MUs' perspective when the MU's efforts are unknown.

Clearly, as discussed, the prior art is limited in several ways. The conflicting interests of the MCSP and MUs under realistic conditions, i.e., when the MU's efforts are not known in advance, have not been considered yet. Furthermore, the prior art does not consider the problem of collision in the online learning scenario. Collisions may significantly reduce the overall performance and therefore need to be avoided. This open problem of online learning for the task assignment can be cast as a multi-player multi-armed bandit problem [24]. In the learning literature, multi-player multi-armed bandits have been investigated under some simplifying assumptions. For example, assuming that there are no individual preferences, the authors in [25] propose to divide the reward among colliding agents to improve the learning speed. In [26], a multi-armed bandit with a collision-avoidance mechanism is proposed. The authors assume that there are no individual costs or payments associated to the decisions in order to allow each player to learn its own preferences while avoiding collisions with competitors. Centralized and decentralized learning strategies are compared in [27], where the effect of sharing the learned preferences is analyzed. This work assumes a cooperative setting, in which all agents communicate their decisions with all other agents. Despite considering multi-agent multi-armed



(a) The MCSP broadcasts the list of available tasks to all MUs.

(b) The MUs send a sensing proposal to the MCSP. The red circle represents a collision.

(c) The MCSP accepts or rejects the sensing proposals of the MUs.

(d) The MUs perform the task, transmit the result, and receive their payment.

Fig. 1. Overview of the system model.

bandits, the solutions in [24]–[27] cannot be applied to the task allocation problem in MCS. Their simplifying assumptions clash with the requirements of MCS. Specifically, the MCSP and the MUs have individual preferences according to their capabilities and conditions. Moreover, the allocation of task implies an effort for the MUs and a payment for the MCSP, and the strict privacy constraints and communication overhead requirements limit the communication between the agents.

C. Contributions

The main contribution of this paper is a novel decentralized task assignment scheme for MCS that can improve the satisfaction of the MUs and the MCSP, which are considered to be individual rational decision makers with *incomplete information*. In the studied MCS system, the effort required for each task in terms of completion time and energy consumption is not known initially, which leads to a difficult learning problem. Using existing online learning solutions leads to many collisions between the MUs, which results in a high overhead and degraded overall system performance. In particular, we propose a novel decentralized algorithm termed collision-avoidance multi-armed bandit with strategic free sensing (*CA-MAB-SFS*), whose goal is to find a stable task assignment, i.e., a task assignment where neither the MUs nor the MCSP have an incentive to change the task assignment. Our contributions can therefore be summarized as follows:

- To balance the conflicting interests of the MCSP and the MUs, we propose the use of a novel decentralized online learning strategy which leverages elements from multi-armed bandits and game theory. Our approach has the advantage that it does not require a priori knowledge of the MU's effort for each task and it incorporates the individual utility functions of the MUs and the MCSP. In contrast to existing works in this space [17]–[21], our approach considers the MUs and the MCSP to be individual rational decision makers.
- We propose an new "free-sensing" mechanism to ensure that all MUs learn their expected effort for all task types thereby reducing future collisions. The idea behind the free-sensing strategy is that, occasionally, the MUs offer to perform tasks for free to ensure the tasks are assigned to them. Performing a task for free is seen as an investment from the MU's perspective, as the MU can improve its estimate of the required effort when performing said task.
- We show that the proposed decentralized *CA-MAB-SFS* converges to a stable task assignment, where neither the MUs nor the MCSP have an incentive to change the task assignment. Moreover, we prove that the stable regret, which is the expected loss incurred by not adopting the optimal assignment, is bounded by a sublinear function. Additionally, we show that the computational complexity of the proposed decentralized online learning is only linearly dependent on the number of task types.
- We evaluate the performance of the proposed algorithm by comparing it with state-of-the-art baseline algorithms. The results verify that, under various settings, the proposed mechanism is effective in terms of worker satisfaction and MCSP's utility. Simulation results show that we achieve the optimum of the social welfare, which is the sum of the utility functions of MUs and the MCSP. Moreover, the proposed algorithm achieves an improvement of $16\,\%$ in terms of average task completion time compared to a state-of-the-art online learning algorithm. The performance is scalable and remains near-optimal even for large network sizes.

The rest of this paper is organized as follows. In Section II, we introduce the MCS system model. The proposed *CA-MAB-SFS* is explained in Section III. In Section IV, we analyze the offline optimal solution and prove that the proposed algorithm converges to a stable solution. The numerical evaluation of *CA-MAB-SFS* is presented in Section V and finally, Section VI concludes the paper.

II. System Model

We first describe our MCS system model. A summary of the used notation is provided in Table I. We consider a set $\mathcal{K}$ of K MUs who seek to perform tasks for the MCSP. As shown in Figure 1, a single MCSP publishing N tasks is considered. We consider a set $\mathcal{Z}$ of Z different task types that represent several examples such as sensing temperature, taking a picture, or classifying an event. Each one of the N tasks is classified

TABLE I
TABLE OF NOTATIONS

SYMBOL	DESCRIPTION	SYMBOL	DESCRIPTION
z, Z, $\mathcal{Z}$	Task type, Number of task types, Set of task types	p_k^{comm}	Transmission power of MU k
p_k^{comp}	Power required for computation at MU k	s_z, c_z	Size of task type z, Complexity of task type z
$a_{n,t}$	Task published at time t	$P_{k,n,t}$	MU k earnings from task $a_{n,t}$
$\mathcal{A}_t$	Set of published tasks at time t	s_z	Average result size for task type a_n
$g_t : \mathcal{A}_t \to \mathcal{Z}$	Function mapping tasks to the type	$E_{k,n,t}$	Energy used by MU k to complete $a_{n,t}$
$\mathcal{A}_{z,t}$	Set of all tasks with type z	$\tau_{k,n,t}$	MU k completion time for task $a_{n,t}$
$\mathcal{I}$	Complete information	$\tau_z^{\max}$	Average deadline of task type a_n
$w_{z,t}$	MCSP earnings from completion of task $a_{n,t}$	$\tau_{k,n,t}^{\text{sense}}$	MU k sensing time for task $a_{n,t}$
$\mathcal{I}_n^{\text{Task}}$, $\mathcal{I}_k^{\text{MU}}$	MSCP-side, MU-side information	$\tau_{k,n,t}^{\text{comm}}$	MU k transmission time for task $a_{n,t}$
K	Number of MUs	$U_{k,n,t}^{\text{MU}}$	Utility of MU k for performing task $a_{n,t}$
$\mathcal{K}$	Set of MUs	$U_{k,n,t}^{\text{MCSP}}$	Utility of MSCP after task $a_{n,t}$ is performed
$\mathbb{E}\{X\}$	Expected value of random variable X	$\mathbb{P}(E)$	Probability of event E

according to their type $z \in \mathcal{Z}$. Time is divided into discrete time slots with index $t = 1, \ldots, T$. In each time slot t, the MCSP publishes a set of available tasks $\mathcal{A}_t = \{a_{n,t}\}$, which can be seen in Fig. 1a. The mapping between task $a_{n,t}$ and its type z is given by a function $g_t : \mathcal{A}_t \to \mathcal{Z}$, i.e., $g_t(a_{n,t}) = z$ means that $a_{n,t}$ is of type z. Furthermore, we collect all tasks of the same type z in the set $\mathcal{A}_{z,t} \subseteq \mathcal{A}_t$. We assume that the MCSP may publish multiple tasks of the same type and each published task requires only one MU to complete. If the MCSP requires multiple MUs to perform a task, the task can be included multiple times in the set $\mathcal{A}_t$ of tasks [23].

The tasks are assumed to be time-sensitive by nature, i.e., the task's result must arrive in time at the MCSP [28], [29]. Therefore, each task type z is characterized by the average size s_z of its result, measured in bits, and an average deadline $\tau_z^{\max}$. The duration of the time slots is chosen according to the maximum completion time of a task. We assume that the deadline $\tau_z^{\max}$ is shorter than the duration of a time slot, i.e., tasks always have to be completed within one time slot. Individual tasks $a_{n,t}$ of the same type z have different characteristics drawn from a type-specific, stationary probability distribution. This probability distribution is unknown to the MCSP and the MUs.

The MCSP earns $w_{z,t}$ monetary units for the timely completion of a task $a_{n,t} \in \mathcal{A}_{z,t}$. The earning $w_{z,t}$ is paid by the data requester. To incentivize the MUs to participate, the MCSP pays the executing MU k when the task is finished before the deadline. MUs are paid for the successful completion of the task according to the effort (time and energy) that MU k spent for the task completion [9].

A. Mobile Units

In every time slot t, each MU $k \in \mathcal{K}$ can perform at most one task $a_{n,t}$. Without loss of generality, we assume that every MU k is equipped with sensors that are capable of performing tasks from all Z task types. To complete the assigned task, MU k has to spend effort in terms of time and energy. The completion time $\tau_{k,n,t}$ of task $a_{n,t}$ contains three parts [29]: the sensing time $\tau_{k,n,t}^{\text{sense}}$, the computation time $\tau_{k,n,t}^{\text{comp}}$, and the communication time $\tau_{k,n,t}^{\text{comm}}$ for the transmission of the task's result. The sensing time $\tau_{k,n,t}^{\text{sense}}$ is the time required by the MU to obtain valid sensing data. For example, in a traffic monitoring scenario, the platform requires MU k to record a specific-duration traffic video in a certain position of a road. The sensing time $\tau_{k,n,t}^{\text{sense}}$ of MU k for task $a_{n,t} \in \mathcal{A}_{z,t}$ is drawn from a stationary random distribution with the probability density function (PDF) $f^z_{\tau_{k,n,t}^{\text{sense}}}(\tau_{k,n,t}^{\text{sense}})$. The expected value $\bar{\tau}_{k,z}^{\text{sense}} = \mathbb{E}(\tau_{k,n,t}^{\text{sense}})$ of the sensing time depends on the task's type z and the MU k performing the task [29].

The computation time $\tau_{k,n,t}^{\text{comp}}$ is the time required by MU k to preprocess the sensing data of a task of type z. Each MU is equipped with a central processing unit (CPU) with frequency f_k^{local}. The computation time is given by

$$\tau_{k,n,t}^{\text{comp}} = \frac{c_z s_z}{f_k^{\text{local}}}, \tag{1}$$

whereas c_z is the preprocessing complexity of the task type z.

The communication time $\tau_{k,n,t}^{\text{comm}}$ is the time required to transmit the preprocessed result of the task from MU k to the MCSP. This time depends on the communication rate between MU k and the MCSP and it is drawn from a stationary random distribution with the PDF $f^z_{\tau_{k,n,t}^{\text{comm}}}(\tau_{k,n,t}^{\text{comm}})$. The expected value $\bar{\tau}_{k,z}^{\text{comm}} = \mathbb{E}(\tau_{k,n,t}^{\text{comm}})$ of the communication time depends on the size s_z of the task result and the MU k's channel quality. The communication bandwidth is shared between the MUs using Orthogal Frequency Division Multiple Access (OFDMA) [30]. Using OFDMA, each MU is assigned an interference free part of the communication bandwidth. Furthermore, it is assumed that the system provides sensing resources orthogonal to the communication resources. Therefore sensing and communication do not interfere. The total time MU k spends for task completion is $\tau_{k,n,t} = \tau_{k,n,t}^{\text{sense}} + \tau_{k,n,t}^{\text{comm}} + \tau_{k,n,t}^{\text{comp}}$. The time $\tau_{k,n,t}$ for task completion needs to be smaller than the deadline $\tau_z^{\max}$.

Additionally, MU k must spend energy from its limited battery. We assume that the energy $E_{k,n,t}$ used by MU k for the task completion is given by

$$E_{k,n,t} = p_k^{\text{comm}} \cdot \tau_{k,n,t}^{\text{comm}} + p_k^{\text{comp}} \cdot \tau_{k,n,t}^{\text{comp}}, \tag{2}$$

where p_k^{comm} is the transmit power of MU k required to transmit the results of task $a_{n,t}$ and p_k^{comp} is the power required for the computation. We neglect the energy required for the sensors, as this energy consumption is small compared to the communication and computation energy [31].

In our model, all MUs have an MU-specific cost function $C_k^{\text{effort}}(\tau_{k,n,t}, E_{k,n,t})$ when performing a task. This cost function depends on the effort required to complete the task. For example, some MUs may have a low battery level that results in a high cost to use energy $E_{k,n,t}$. Other MUs might be concerned about the availability of their own communication, computation, or sensing resources, thus placing a high cost for

the time $\tau_{k,n,t}$ during which the MU's resources are used. We define the cost function as follows:

$$C_k^{\text{effort}}(\tau_{k,n,t}, E_{k,n,t}) = \alpha_k \tau_{k,n,t} + \beta_k E_{k,n,t}. \quad (3)$$

The cost function in (3) captures the tradeoff between the completion time $\tau_{k,n,t}$ and the consumed energy $E_{k,n,t}$, with α_k being an MU-specific time cost parameter and β_k an MU-specific energy cost parameter.

The MCSP pays $P_{k,n,t}$ monetary units to compensate MU k for the effort it spends to complete the task. This payment is defined as

$$P_{k,n,t} = P^{\text{effort}}(\tau_{k,n,t}, E_{k,n,t}), \quad (4)$$

where the payment function P^{effort} depends on the time and energy spent for the completion of the task. The utility of MU k in time slot t when performing task $a_{n,t}$ is

$$U_{k,n,t}^{\text{MU}} = P_{k,n,t} \mathbb{1}_{\tau_{k,n,t} \leq \tau_z^{\max}} - C_k^{\text{effort}}(\tau_{k,n,t}, E_{k,n,t}), \quad (5)$$

where $\mathbb{1}_{\tau_{k,n,t} \leq \tau_z^{\max}}$ is the indicator function for the case in which MU k completed the task before its deadline $\tau_z^{\max}$. The expected utility $\bar{U}_{k,z}^{\text{MU}}$ for performing a task of type z is:

$$\begin{aligned} \bar{U}_{k,z}^{\text{MU}} &= \mathbb{E}\{U_{k,n,t}^{\text{MU}} | a_{n,t} \in \mathcal{A}_{z,t}\} \quad (6) \\ &= \mathbb{E}\{P_{k,n,t}\} \cdot \mathbb{P}\{\tau_{k,n,t} \leq \tau_z^{\max}\} - \mathbb{E}\{C_k^{\text{effort}}(\tau_{k,n,t}, E_{k,n,t})\}. \end{aligned}$$

Note that MU k is able to observe the completion time $\tau_{k,n,t}$ and consumed energy $E_{k,n,t}$ independently, but only after task $a_{n,t}$ has been performed.

B. Mobile Crowdsensing Platform

In each time slot t, the MCSP publishes a list of available tasks $\mathcal{A}_t$ as shown in Fig. 1a. Each task from this list belongs to one of the Z task types. The MCSP is paid by a data requester to provide results for each task $a_{n,t} \in \mathcal{A}_t$. The earning $w_{z,t}$ depends on the task type z. Moreover, we assume $w_{z,t}$ to be deterministic and known beforehand to the MCSP, i.e., the MCSP and the data requester have made a contractual agreement. The utility $U_{k,n,t}^{\text{MCSP}}$ of the MCSP when assigning MU k to task $a_{n,t} \in \mathcal{A}_{z,t}$ is defined as

$$U_{k,n,t}^{\text{MCSP}} = (w_{z,t} - P_{k,n,t}) \mathbb{1}_{\tau_{k,n,t} \leq \tau_z^{\max}}. \quad (7)$$

The expected utility $\bar{U}_{k,z}^{\text{MCSP}}$ when assigning MU k to a task from task type z is given by

$$\begin{aligned} \bar{U}_{k,z}^{\text{MCSP}} &= \mathbb{E}\{U_{k,n,t}^{\text{MCSP}} | a_{n,t} \in \mathcal{A}_{z,t}\} \quad (8) \\ &= (w_{z,t} - \mathbb{E}\{P_{k,n,t}\}) \cdot \mathbb{P}\{\tau_{k,n,t} \leq \tau_z^{\max}\}. \end{aligned}$$

C. Available information

As the probability distributions $f^z_{\tau^{\text{comm}}_{k,n,t}}(\tau^{\text{comm}}_{k,n,t})$ and $f^z_{\tau^{\text{sense}}_{k,n,t}}(\tau^{\text{sense}}_{k,n,t})$ of the task characteristics are not known in advance, the MUs must estimate the average effort required for each task type. We define $\mathcal{I}_k^{\text{MU}} = \{\bar{U}_{k,z}^{\text{MU}}, \forall z\}$ as the *MU-side* information about the stochastic characteristics of the task types, i.e., the average achievable utility $\bar{U}_{k,z}^{\text{MU}}$ for each task type z. $\mathcal{I}_k^{\text{MU}}$ contains information about the expected energy consumption and the expected execution time for all task types $z \in \mathcal{Z}$. Note that $\mathcal{I}_k^{\text{MU}}$ is not available at the MUs and has to be learned over time from experience. Similarly, we define $\mathcal{I}_z^{\text{Task}} = \{\bar{U}_{k,z}^{\text{MCSP}}, \forall k\}$ as the *MCSP-side* information about the MUs. $\mathcal{I}_z^{\text{Task}}$ contains information about the earnings and the required payment for all MUs. As in the MU's case, $\mathcal{I}_z^{\text{Task}}$ is not available at the MCSP in advance. The combination of MU-side and MCSP-side information, $\mathcal{I} = \{\mathcal{I}_k^{\text{MU}} \cup \mathcal{I}_z^{\text{Task}}, \forall k, z\}$, is called the *complete* information and is unknown to the MUs and the MCSP.

Our goal is to optimize the assignment of tasks in a completely decentralized fashion without requiring prior knowledge of $\mathcal{I}$. For this purpose, the MUs learn the characteristics of each task type and find their most preferred task in each time slot t. In turn, the MCSP has to identify the best MU k to select for each task type. We assume strict privacy constraints, meaning that the MUs do not share information about $\mathcal{I}_k^{\text{MU}}$, neither with the MCSP nor with other MUs. Additionally, the MCSP does not share $\mathcal{I}_z^{\text{Task}}$ with the MUs. As shown in Fig. 1, there is no information sharing between the MUs. Moreover, the information exchange between the MCSP and each of the MUs is limited to only broadcasting the available tasks, sending sensing proposals, accepting or rejecting said sensing proposals, and sending the sensing results.

We argue that a decentralized online learning strategy is an efficient solution to the task assignment problem. Through online learning we can effectively address the key challenge of incomplete information. In contrast to centralized schemes, adopting a decentralized learning strategy ensures privacy for the MUs since they do not need to share their local information $\mathcal{I}_k^{\text{MU}}$. Furthermore, both the MUs and the MCSP are modeled as autonomous decision-makers aiming to maximize their own utility. Moreover, a decentralized approach reduces the complexity of the problem, compared to centralized case. This is because we can leverage the individual learning capabilities of each MU, thus eliminating the need to deal with a combinatorial problem at a centralized controller. To analyze the task assignment problem from the perspective of the MUs and the MCSP, we first present the task assignment game between MUs and MCSP.

D. Problem Formulation: Task Assignment Game

In contrast to either MU-centric MCS [28], or MCSP-centric MCS [32], we consider the perspective of both, the selfish MUs and the selfish MCSP. Contrary to [28] and [32], we do not formulate a global objective function for the performance of the task assignment. Instead, we consider all MUs and the MCSP to be rational decision makers with their individual preferences and decision making capabilities. Therefore, we use game theory, specifically matching theory [33], to analyze the task assignment problem. The main goal of matching theory is to obtain a stable matching, i.e., reaching a situation in which MUs and MCSP cannot simultaneously improve by changing the task assignment. This corresponds to selfishly-deciding MUs and an MCSP that individually try to obtain their best task assignment. A *stable matching* outcome is apropos for the presented MCS problem because it allows the maximization of satisfaction for both the MUs and the MCSPs, with regard to their individual preferences.

The matching game is a model for a two-sided market in which the MUs provide their sensing resources and the

MCSP requires sensing resources. These demands come in the form of indivisible sensing tasks, which the MUs execute in exchange of a payment [34]. The payment function P^{effort} and the MUs' cost function C_k^{effort} are given functions which depend on the task assignment [35]. The proposed, matching-based task assignment game $\mathcal{G}_t$ in time slot t is formally described by a tuple $\mathcal{G}_t = (\mathcal{K}, \mathcal{A}_t, \succeq_k^{\text{MU}}, \succeq_z^{\text{MCSP}})$ containing the set $\mathcal{K}$ of MUs, the set $\mathcal{A}_t$ of available tasks, the MUs' preference ordering $\succeq_k^{\text{MU}}$, and the MCSP's preference ordering $\succeq_z^{\text{MCSP}}$.

The MUs' preference ordering $\succeq_k^{\text{MU}}$ ranks task types according to the expected utility of the task type z, i.e.,

$$z \succeq_k^{\text{MU}} z' \iff \bar{U}_{k,z}^{\text{MU}} \geq \bar{U}_{k,z'}^{\text{MU}}. \tag{9}$$

In other words, MU k prefers task type z over z' if the MU's expected utility (6) of performing tasks of type z is higher than of tasks of type z'. The preference orderings $\succeq_k^{\text{MU}}$ can only be correctly determined with the MU-side information $\mathcal{I}_k^{\text{MU}}$.

The MCSP prefers MUs which yield the highest expected utility for each task type z, i.e.,

$$\text{MU } k \succeq_z^{\text{MCSP}} \text{MU } l \iff \bar{U}_{k,z}^{\text{MCSP}} \geq \bar{U}_{l,z}^{\text{MCSP}}. \tag{10}$$

The expression in (10) implies that when performing task type z, the MSCP prefers MU k because it provides a higher utility compared to MU l. This preference ranking can only be correctly determined with the MCSP-side information $\mathcal{I}^{\text{MCSP}}$.

MU k signals its willingness to participate in any task of the type z by sending a sensing offer $O_{k,t}$ as shown in Fig. 1b. Based on the received offers, the MSCP performs the assignment according to its preference ordering $\succeq_z^{\text{MCSP}}$ as depicted in Fig.1c. We denote the task assignment by the binary variable $x_{k,n,t}$. When $x_{k,n,t} = 1$, MU k is assigned to task $a_{n,t}$. Otherwise, $x_{k,n,t} = 0$. The variables $x_{k,n,t}$ associated to all MUs and tasks in time slot t are collected in the matrix $\boldsymbol{X}_t$.

Definition 1. *A task assignment $\boldsymbol{X}_t$ is* unstable *if there are two MUs, MU k and MU l, and two tasks, $a_{n,t}$ and $a_{m,t}$, such that: (i) $x_{k,n,t} = 1$, i.e. MU k is assigned to task $a_{n,t} \in \mathcal{A}_{z,t}$. (ii) $x_{l,m,t} = 1$, i.e. MU l is assigned to task $a_{m,t} \in \mathcal{A}_{z',t}$. (iii) $z' \succ_k^{\text{MU}} z$ and MU $k \succeq_{z'}^{\text{MCSP}}$ MU l, i.e., MU k strictly prefers the task with type z' over its current matched task of type z, and the MCSP would profit more if the task of type z' is performed by MU k instead of its current matched MU l.*

The pair (MU k, z') is called a blocking pair [36], because both the MU k and the MCSP are unsatisfied with the current assignment. The existence of the blocking pair (MU k, z') causes the matching $\boldsymbol{X}_t$ to be unstable because MU k could switch to $a_{m,t} \in \mathcal{A}_{z',t}$ and both, the MU k and the task $a_{m,t}$ would obtain a more efficient matching and therefore a higher expected utility.

The assignment $\boldsymbol{X}_t$ is said to be stable if no blocking pairs exist [36]. In such cases, no MU or task could change the assignment and improve their expected utilities. In MCS, this means that each MU is assigned to its most preferred task while the MCSP selects its most preferred MU for each task. Note that the stable matching may not be unique. There are, in fact, potentially multiple solutions. We denote the set of stable solutions as $\mathcal{X}^{\text{stable}}$ and define a_k^{stable} as a stable task for MU k. The expected utility of this task is $\bar{U}_k^{\text{MU,stable}} = \bar{U}_{k,a_k^{\text{stable}}}^{\text{MU}}$.

III. Collision-Avoidance Multi-Armed Bandit with Strategic Free Sensing

For most existing works on matching and assignment games, it is customary to use the so-called deferred acceptance algorithm (see Section IV-A) that guarantees convergence to a stable matching [37]. However, for our MCS problem, this approach would not be adequate because of several reasons. First, the MUs do not know how much effort is required for each task type z. Consequently, each MU has to learn its MU-side information $\mathcal{I}_k^{\text{MU}}$ and its preferences by exploration. Second, collisions with competing MUs occur while exploring different task types. To avoid collisions and to ensure a good learning performance, a collision-avoidance mechanism is required. As such, we propose a novel approach that combines online learning with matching theory including a collision-avoidance mechanism. This is more appropriate here because we can overcome the challenge of incomplete information and collisions due to the competition of the MUs.

In each time slot t, MU k may send one sensing offer $O_{k,t}$ for a task type z together with its payment proposal $\hat{P}_{k,z}$. The payment proposal $\hat{P}_{k,z}$ is calculated by the MUs based on their observed efforts for task type z. To lower the communication overhead between the MCSP and the MUs, we assume that the MUs can only send sensing offers for one task type at a time. The MUs' challenge in sending a good sensing offer lies in the fact that the MUs do not know their expected utility and effort, i.e. time and energy, required to complete tasks of type z in advance. When more MUs attempt to execute the same task type than tasks are available, i.e., sensing offers are colliding, the MCSP decides which MUs are assigned to the tasks according to the MCSP's utility (7) and the number $|\mathcal{A}_{z,t}|$ of tasks with type z. As shown in Fig. 1c, the MCSP then sends a response $\bar{O}_{k,t}$ which contains whether the sensing offer was accepted, and which task was assigned to the MU.

Only the MU accepted by the MCSP and therefore, assigned to $a_{n,t}$, i.e., $\bar{O}_{k,t} = a_{n,t}$, can perform the task. Therefore, it is the only MU able to measure its utility $U_{k,n,t}^{\text{MU}}$ and effort in terms of time $\tau_{k,n,t}$ and energy $E_{k,n,t}$. The MUs which were declined only learn that there are other MUs competing for task type z which were preferred by the MCSP. The competition between the MUs for the sensing tasks is especially challenging in the exploration phase, i.e., when the utility and effort for each task type are not well estimated. As a result, the payment proposals are either too low, which leads to a low utility, or too high, which increases the probability of a sensing offer being declined.

In this section, our goal is to provide a fully decentralized online learning algorithm, which overcomes the challenges of the unknown information $\mathcal{I}$ and the competition between MUs. In particular, we propose a novel decentralized online learning method termed CA-MAB-SFS. The algorithm is fully decentralized and it consists of two strategies: The strategy of the MUs and the strategy of the MCSP. The strategy of the MU is to select the best task type z for which to send a sensing

Algorithm 1 CA-MAB-SFS (MUs' online learning)

Require: $\epsilon_t, \lambda \in [0,1), \alpha \in [0,1)$
1: Initialize $\hat{U}_{k,0}(z)$ and $\hat{J}_{k,0}(z)$, set $\gamma_{k,z} = 0\ \forall k \in \mathcal{K}, z \in \mathcal{Z}$
2: **for** $t = 1, \ldots, T$ **do**
3: MCSP publishes sensing tasks $\mathcal{A}_t$ and $P_{z,t-1} = \max\{\hat{P}_{k,z} | x_{k,n,t-1} = 1, a_{n,t} \in \mathcal{A}_{z,t}\}$.
4: Determine available task types $\mathcal{Z}$ from the set $\mathcal{A}_t$ of published tasks and the sets $\mathcal{A}_{z,t}$.
5: **if** $t = 1$ **then**
6: MU k sends sensing offer $O_{k,t} \leftarrow z$, to a uniformly random chosen task type $z \in \mathcal{Z}$.
7: **else**
8: Draw i.i.d. random variable $D_{k,t}$ with $\mathbb{P}(D_{k,t} = 1) = \lambda$, $\mathbb{P}(D_{k,t} = 0) = 1 - \lambda$.
9: **if** $D_{k,t} = 0$ **then**
10: **for each** $z \in 1, \ldots, Z$ **do**
11: **if** $\gamma_{k,z} > \epsilon^{\mathrm{a}}$ **then** $\hat{P}_{k,z} \leftarrow 0$ ▷ free sensing offer
12: **else** $\hat{P}_{k,z} \leftarrow P^{\mathrm{effort}}(\hat{J}_{k,t-1}(z))$ ▷ paid sensing offer
13: **end for**
14: Update plausible set, i.e., $S_k = \{z : P_{z,t-1} \geq \hat{P}_{k,z}, \forall z = 1, \ldots, Z\}$
15: Select $z \in S_k$ using ϵ - greedy and send sensing offer $O_{k,t} \leftarrow z$.
16: **else**
17: Send same sensing offer $O_{k,t} \leftarrow O_{k,t-1}$ as in the previous timestep.
18: **end if**
19: **end if**
20: Wait for the MCSP's decision $\bar{O}_{k,t}$ from Algorithm 2.
21: **if** MU k is accepted, i.e., $\bar{O}_{k,t} = a_{n,t}$ **then**
22: Assign the task to MU k, i.e., $x_{k,n,t} \leftarrow 1$, where $\bar{O}_{k,t} = a_{n,t}$.
23: Perform the task $a_{n,t}$ and observe $U^{\mathrm{MU}}_{k,n,t}$, $\tau_{k,n,t}$ and $E_{k,n,t}$.
24: Update estimates $\hat{U}_{k,t}(z)$ and $\hat{J}_{k,t}(z)$.
25: Reset rejection counter, i.e. $\gamma_{k,z} \leftarrow 0$.
26: **else**
27: $\hat{U}_{k,t}(z) \leftarrow \hat{U}_{k,t-1}(z)$, $\hat{J}_{k,t}(z) \leftarrow \hat{J}_{k,t-1}(z)$
28: **if** $t < \epsilon^{\mathrm{e}}$ **then** increase rejection counter of task type z, i.e., $\gamma_{k,z} \leftarrow \gamma_{k,z} + \frac{1}{t}$
29: **end if**
30: **end for**

offer and the payment proposal. The strategy of the MCSP is to select the best sensing offers out of the received MUs' sensing offers for each task type. Our algorithm only requires information exchange between the MUs and the MCSP. No information is exchanged between different MUs.

As mentioned before, a major challenge for the MUs is the exploration of task types, particularly at the beginning. Exploration is needed to estimate the effort associated with each task type. However, at the beginning, all MUs compete with each other because they all have only poor estimates of the required effort for each task type. Intuitively, MUs may get rejected by the MCSP because they overestimated the effort associated with a task type. This will cause high payment proposals for this task type in the future, leading to further rejections and, thus, to an inability to correctly learn the estimate of the effort. To overcome this, we propose the concept of *strategic free sensing*. MUs can decide to sense a task from a certain task type for free and in exchange learn about the task type characteristics. This is done in the following way: The MU proposes to the MCSP to perform the task for free, i.e., the payment proposal $P_{k,n,t}$ is 0. Each MU k updates a rejection counter $\gamma_{k,z}$ for each task type z if it has been rejected by the MCSP. After a threshold value is reached, the MU sends a free sensing offer to get accepted with a high probability.

Algorithm 1 describes the online learning process of each MU. In the beginning, each MU k initializes its estimates $\hat{J}_{k,0}(z)$ and $\hat{U}_{k,0}(z)$ (see line 1). If prior knowledge is available at MU k, it may initialize $\hat{J}_{k,0}(z)$ and $\hat{U}_{k,0}(z)$ according to its prior knowledge. Otherwise, these values are initialized to zero. In each time slot t, MU k receives a list of available sensing tasks $\mathcal{A}_t$ together with information about the payment proposal

$$P_{z,t-1} = \max\{\hat{P}_{k,z} | x_{k,n,t-1} = 1, a_{n,t} \in \mathcal{A}_{z,t}\} \tag{11}$$

of the MU which was most expensive in the previous task assignment in $t-1$ for each task type (line 3). In the first time slot $t = 1$, MU k sends a sensing offer for a random task type (line 5-7), as no information about the utility and the effort for each task type is available. For $t > 1$, MU k draws a random number $D_{k,t}$ which is equal to one with probability λ and zero with probability $1 - \lambda$ (line 8). If $D_{k,t} = 1$, MU k sends a sensing offer to the same type as in the offer sent in the last time slot $t-1$ (lines 16-18). The idea behind this mechanism is that not all MUs change their sensing offers simultaneously, which is required for the convergence of the online learning [26]. The parameter λ controls the trade-off between initial learning speed and convergence, which is discussed in Section V. If $D_{k,t} = 0$, MU k determines the payment proposal for each task type z based on its effort estimate $\hat{J}_{k,t}(z)$. If MU k's rejection counter $\gamma_{k,z}$ is larger than a predefined threshold ϵ^{a} (line 12), MU k offers to sense the task for free. Furthermore, MU k determines the plausible set S_k containing all task types z where its payment $\hat{P}_{k,z}$ is lower than $P_{z,t-1}$ from (11), i.e., all the task types which MU k can perform for a lower or equal payment than the most expensive MU who performed a task of the same type in the last assignment (line 14). A task type from the plausible set S_k is chosen according to the ϵ-greedy strategy [38], i.e. with probability ϵ_t a random task type is chosen, and with probability $1 - \epsilon_t$ the task with the highest expected utility is selected (line 15). The sensing offer $O_{k,t}$ with the payment proposal $P_{k,n,t}$ is sent to the MCSP. Afterwards, MU k waits for the response of the MCSP, described in Algorithm 2.

After MU k receives the MCSP's response, its next action depends on whether it was accepted or not. If MU k was accepted, the task $a_{n,t}$ is performed and the utility $U^{\mathrm{MU}}_{k,n,t}$ and the effort regarding time $\tau_{k,n,t}$ and $E_{k,n,t}$ is observed and used to update the estimate $\hat{U}_{k,t}(z)$ of the utility and the estimate $\hat{J}_{k,t}(z)$ of the task types's effort. The update of $\hat{U}_{k,t}(z)$ is then given as

$$\hat{U}_{k,t}(z) = \hat{U}_{k,t-1}(z) + \frac{1}{N_k(z)} \cdot (U^{\mathrm{MU}}_{k,n,t} - \hat{U}_{k,t-1}(z)), \tag{12}$$

which is the iterative estimate of the mean value of $U^{\mathrm{MU}}_{k,n,t}$, where $N_k(z)$ represents the number of times that MU k has been assigned to task type z. The estimate of the effort $\hat{J}_{k,t}(z)$ for task type z is updated analogously. If MU k was rejected by the MCSP, it receives no information about the utility of the task type and the required effort (line 26). Only the rejection counter $\gamma_{k,z}$ of task type z is increased by the value t/ϵ^{s} (line 27). The analysis of the convergence of the proposed CA-MAB-SFS is presented in the following Section IV.

Algorithm 2 describes the decision-making process of the

Algorithm 2 CA-MAB-SFS (MCSP's decision)

Require: $\mathcal{K}, \mathcal{A}, w_{z,t}$
1: **for** $t = 1, \ldots, T$ **do**
2: Publish available sensing tasks $\mathcal{A}_t$ and $P_{z,t-1} = \max\{\hat{P}_{k,z} | x_{k,n,t-1} = 1, a_{n,t} \in \mathcal{A}_{z,t}\}$.
3: Wait for all sensing offers $O_{k,t}$ and payment proposals $\hat{P}_{k,z}$.
4: **for** $z = 1, \ldots, Z$ **do**
5: Select the $|\mathcal{A}_{z,t}|$ MUs with the lowest payment proposals.
6: Send acceptance response to the selected MUs, i.e., $\bar{O}_{k,t} = a_{n,t}\ \forall a_{n,t} \in \mathcal{A}_{z,t}$
7: Send rejection response to all other MUs, i.e., $\bar{O}_{l,t} = \varnothing$.
8: **end for**
9: **end for**

MCSP for each task. After the list of available tasks is published by the MCSP, it waits for the MUs' sensing offers. Then, for each task type, the MCSP selects the MUs with the lowest payment proposal to complete all $|\mathcal{A}_{z,t}|$ tasks of type z (line 5). If the lowest payment proposal is larger than $w_{z,t}$, the MCSP rejects all MUs. For each MU k, the MCSP sends a response $\bar{O}_{k,t}$ indicating whether the MU is accepted or rejected.

IV. Convergence and Regret Bound Analysis for the Proposed CA-MAB-SFS Algorithm

In this section, we show that the proposed algorithm is guaranteed to converge to a stable solution and its regret bound is fixed. For the proof, we assume that in each round the number $|\mathcal{A}_{z,t}|$ of tasks of each type is fixed. Furthermore, we assume that the mapping function $g_t : \mathcal{A}_t \rightarrow \mathcal{Z}$ is constant over time, i.e., the type of the task $a_{n,t}$ is the same in every round. This applies to MCS scenarios in which each task has to be repeated regularly to update the measurements, e.g., traffic or temperature measurements in a smart city.

A. Solution with complete information

In this section, the solution of the matching-based, task assignment game is discussed when all players have complete information $\mathcal{I}$. This assumption is unrealistic and it is only used to derive a baseline for our CA-MAB-SFS algorithm. We will only briefly discuss this approach and refer the reader to the related works [16] on stable matching for MCS. We define the oracle as a decision maker with complete information $\mathcal{I}$ who is able to calculate an stable solution in one time slot t. When every MU and the MCSP know $\mathcal{I}$, a stable solution of the task assignment game can be calculated using the deferred acceptance algorithm [37]. The deferred acceptance algorithm to reach a stable task assignment is presented in Algorithm 3. The input is the task assignment game $\mathcal{G}_t$ in time slot t, where all players have access to the complete information $\mathcal{I}$. Each MU is initialized without any assigned task and an empty sensing offer history $\mathcal{Z}_k^{\text{history}}$. After receiving the set $\mathcal{A}_t$ from the MCSP, each MU determines the set $\mathcal{Z}$ of available task types (line 1). The sensing offer history $\mathcal{Z}_k^{\text{history}}$ contains all the task types z to which MU k has sent a sensing offer $O_{k,t}$ in the considered time slot t (line 2). The algorithm is an iterative approach that runs as long as at least one MU remains unmatched and there are task types to which it has not yet sent a sensing offer (line 3). Each unmatched MU k sends a sensing offer considering its most preferred task type z which is not in the sensing offer history (line 4). If all the tasks $a_{n,t}$ of type

Algorithm 3 Offline Deferred Acceptance

Require: $\mathcal{G}_t = (\mathcal{K}, \mathcal{A}_t, \succeq_k^{\text{MU}}, \succeq_z^{\text{MCSP}})$
1: Determine available task types $\mathcal{Z}$ from the set $\mathcal{A}_t$ of published tasks.
2: $O_{k,t} \leftarrow \varnothing,\ \mathcal{Z}_k^{\text{history}} \leftarrow \{\},\ \forall k \in \mathcal{K}$
3: **while** $\exists O_{k,t} = \varnothing \wedge \mathcal{Z}_k^{\text{history}} \neq \mathcal{Z}$ **do**
4: Send sensing offer $O_{k,t}$ for task type z, with $z : z \succeq_k^{\text{MU}} z',\ z \neq z',\ \forall z, z' \in \{\mathcal{Z} \setminus \mathcal{Z}_k^{\text{history}}\}$
5: **if** all $a_{n,t} \in \mathcal{A}_{z,t}$ are assigned **then**
6: **if** MU $k \succeq_z^{\text{MCSP}}$ MU l **then**
7: Assign task $a_{n,t}$ to MU k instead of MU l, i.e., $x_{k,n,t} = 1, x_{l,n,t} = 0$
8: **end if**
9: **else**
10: **if** MU $k \succeq_z^{\text{MCSP}} \varnothing$ **then** assign task $a_{n,t}$ to MU k, i.e., $x_{k,n,t} = 1$
11: **end if**
12: $\mathcal{Z}_k^{\text{history}} \leftarrow \mathcal{Z}_k^{\text{history}} \cup \{z\}$ ▷ Add task type z to the proposal history
13: **end while**
14: **return** $\boldsymbol{X}_t = \{x_{k,n,t}\}_{\forall k,n}$

z are already assigned, and the sensing offer from MU k has a higher expected utility than any of the assigned MU l, the current assigned MU l is exchanged with MU k (lines 5-9). If there are still unassigned tasks of type z, MU k is assigned to one of these tasks as long as MU k has a positive utility (line 11). MU k adds the task type z to which it sent its sensing offer to its sensing offer history (line 13). When all MUs are either assigned to a task or have sent sensing offers to all task types, the output is a stable task assignment $\boldsymbol{X}_t$. Note that Algorithm 3 is only used as a benchmark and cannot be implemented in real applications due to its strict requirement on $\mathcal{I}$, which as discussed before, cannot be fulfilled.

B. Convergence and regret bound for CA-MAB-SFS

In the decentralized task assignment setting, the *stable regret* concept [26] is used to evaluate the performance of learning algorithms. The stable regret describes the performance compared to the offline stable task assignment with complete information from Section IV-A. We define the instantaneous stable regret in t as

$$r_k(t) = \bar{U}_k^{\text{MU,stable}} - \sum_{n=1}^{N} x_{k,n,t} \bar{U}_{k,z}^{\text{MU}}. \tag{13}$$

$r_k(t)$ is computed as the difference between the expected utility $\bar{U}_k^{\text{MU,stable}}$ for the stable matching and the expected utilities of the task assignment $\boldsymbol{X}_t$. The stable regret of a sequence of task assignments $\{\boldsymbol{X}_t\}_{t=1,\ldots,T}$ for MU_k is defined as

$$R_k(T) = \sum_{t=1}^{T} r_k(t). \tag{14}$$

$R_k(T)$ is computed as the sum of all instantaneous regrets over the whole time horizon T.

Theorem 1. *The stable regret is bounded by a sublinear function which is given by*

$$R_k(T) \leq O\left(\Delta_k \frac{8Z^5 K^2 e^{\frac{\Delta^2}{Z\Delta \text{U}}}}{\rho^{Z^4+1}(1 - \frac{\Delta^2}{Z\Delta \text{U}})} \log(T) T^{1 - \frac{\Delta^2}{Z\Delta \text{U}}} \right), \tag{15}$$

where $\rho = (1-\lambda)\lambda^{Z-1}$, $\Delta_k = \max_{z=1,\ldots,Z}\{\bar{U}_k^{\text{MU,stable}} - \bar{U}_{k,z}^{\text{MU}}\}$ *and* $\Delta = min_{i,j \in, i \neq j}\{\bar{U}_{k,i}^{\text{MU}} - \bar{U}_{k,j}^{\text{MU}}\}$.

Proof. See Appendix A. □

The stable regret $R_k(T)$ is bounded by a sublinear function, which means that the average instantaneous stable regret $\overline{r}_k(t) = R_k(T)/T$ goes to zero for $T \to \infty$. The average instantaneous stable regret of the task assignment for each MU diminishes during the online learning procedure.

To prove the convergence of *CA-MAB-SFS*, we analyze the probability $\mathbb{P}(\boldsymbol{X}_T \notin \mathcal{X}^{\text{stable}})$ of not reaching a stable matching in time step T.

Theorem 2. *The probability of not reaching a stable matching in time step T is bounded by*

$$\mathbb{P}(\boldsymbol{X}_T \notin \mathcal{X}^{\text{stable}}) \leq O\left(\frac{8Z^5K^2e^{\frac{\Delta^2}{Z\Delta \text{U}}}}{\rho^{Z^4+1}(1-\frac{\Delta^2}{Z\Delta \text{U}})}\frac{\log(T)}{T^{\frac{\Delta^2}{Z\Delta \text{U}}}}\right). \quad (16)$$

Proof. See Appendix B. □

This probability $\mathbb{P}(\boldsymbol{X}_T \notin \mathcal{X}^{\text{stable}})$ goes to 0 for $T \to \infty$ as $\lim_{T\to\infty} \frac{\log(T)}{T^{\frac{\Delta^2}{Z\Delta \text{U}}}} = 0$. This implies that the probability $\mathbb{P}(\boldsymbol{X}_T \in \mathcal{X}^{\text{stable}})$ of achieving a stable matching approaches 1, therefore CA-MAB-SFS converges. When reaching a stable matching, all MUs and the MCSP would not profit from changing the assignment.

C. Computational complexity analysis

We now analyze the computational complexity of the proposed CA-MAB-SFS algorithm from the perspective of the MUs and the MCSP. For the MUs, we analyze the complexity of one iteration of their learning algorithm (Algorithm 1). Note that the MU's decision only depends on the number Z of available task types. Therefore, we evaluate the algorithm's complexity with regard to Z. From Algorithm 1, we can see that the complexity of lines 1-9 does not grow with the number Z of task types, therefore the computational complexity of each of this lines is constant and of the order $O(1)$. The complexity of line 10-15 is linearly dependent on the number of task types, as the loop iterates over each task type once, and therefore is of the order $O(Z)$. The lines 16-30 are not dependent on Z and are of constant complexity $O(1)$. From this analysis, we can determine that the complexity of the proposed CA-MAB-SFS algorithm grows only linearly with the number Z of available task types, i.e., $O(Z)$.

The MCSP has to choose among the set of proposing MUs $\mathcal{K}$, and therefore the algorithm complexity is analyzed with regard to the number of MUs, K. For the MCSP, the maximum computational complexity stems from the selection of the cheapest payment for each task (Algorithm 2, line 5). For this, the MCSP has to evaluate the cost of each MU once, leading to a linear complexity with regard to the number K of MUs. Therefore, the computational complexity of the MCSP's algorithm is characterized by $O(K)$.

For both, the MUs and the MCSP, the communication overhead is low. The MCSP broadcasts the list of available tasks, receives the sensing offers and transmits the accept and defer messages. Each MU only receives the list of available tasks, submits one sensing proposal, and receives an accept or defer message.

TABLE II
EVALUATION PARAMETERS

Parameter	Value
Number of MUs	$K = 100$
Size of the sensing task result [29]	$s_z \in [50, 100]$ Mbits
Number of task types	$Z = 10$
Tasks per task type	$\lvert\mathcal{A}_{z,t}\rvert \in [5, 10]$
Mean communication rate	$\bar{\tau}_{k,z}^{\text{comm}} \in [0.025, 0.1] \frac{\text{s}}{\text{Mbit}}$
CPU frequency [39]	$f_k^{\text{local}} \in [1, 2]$ GHz
Mean sensing time [29]	$\bar{\tau}_{k,z}^{\text{sense}} \in [60, 180]$ s
Transmission power [39]	$p_k^{\text{comm}} = 200\,\text{mW}$
Power required for computation [39]	$p_k^{\text{comp}} = 1\,\text{W}$
Computational complexity [29]	$[200, 300]$ $\frac{\text{CPU Cycles}}{\text{bit}}$
Earning of MCSP	$w_{z,t} = 1.4 + 3 \cdot s_z$
MUs' cost for energy consumption	$\alpha_k = 0.01$ $\frac{\text{Monetary units}}{\text{J}}$
MUs' cost for time spent sensing	$\beta_k = 0.004$ $\frac{\text{Monetary units}}{\text{s}}$
Payment to the MUs	$P^{\text{effort}}(\tau_{k,n,t}, E_{k,n,t})$ $= 1.1 \cdot C_k^{\text{effort}}(E_{k,n,t}, \tau_{k,n,t})$
Exploration rate	$\epsilon_t = \min\{1, 1/t\}$
Collision-avoidance parameter	$\lambda = 0.1$
Free-sensing parameters	$\epsilon^{\text{e}} = 30$, $\epsilon^{\text{a}} = 0.5$

V. Simulation Results and Analysis

In this section, we evaluate the performance of the proposed CA-MAB-SFS algorithm and compare it to baseline schemes.

A. Evaluation metrics

As the MUs and the MCSP have different goals, the assessment of the system's performance depends on the considered perspective. We argue that different evaluation metrics need to be considered to assess the system's performance.

1) Social Welfare: Social welfare is often used in game theory to evaluate the performance of a solution from the whole network's perspective [15], as it represents a joint utility of all players in the game. The social welfare $U_t^{\text{SW}}(\boldsymbol{X}_t)$ at time slot t can be calculated by

$$U_t^{\text{SW}}(\boldsymbol{X}_t) = \sum_{k=1}^{K}\sum_{n=1}^{N} x_{k,n,t}(U_{k,n,t}^{\text{MCSP}} + U_{k,n,t}^{\text{MU}}), \quad (17)$$

which is the sum of all MUs' utilities and the MCSP's utility.

2) Average completion time: We consider the time that is required to complete the tasks of the MCSP.

3) Energy efficiency: We consider the energy that is required to complete the tasks of the MCSP.

4) Stability and number of blocking pairs: Stability ensures that the MCSP and all MUs are satisfied, i.e., neither the MCSP nor the MUs have an incentive to deviate from the current task assignment. Intuitively, stability is important to ensure that all MUs and the MCSP will use this strategy, as their individual goals are achieved [35]. The number of blocking pairs indicates how many MU-task pairs would profit from changing the task assignment. We measure the number of MUs that are part of a blocking pair, which represents how many MUs could improve their utility by adopting another task assignment.

B. Baseline Algorithms

We use the following algorithms to benchmark our proposed CA-MAB-SFS. Assuming complete information $\mathcal{I}$ for

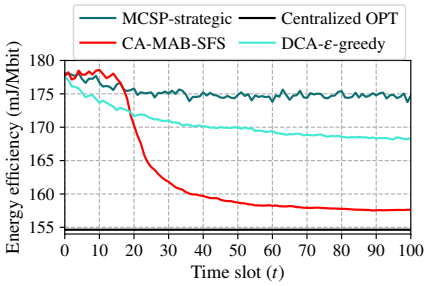


Fig. 2. Energy efficiency as a function of the time step t.

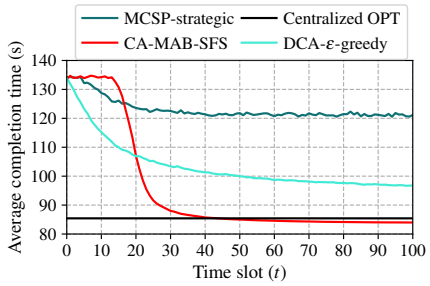


Fig. 3. Average task completion time as a function of the time step t.

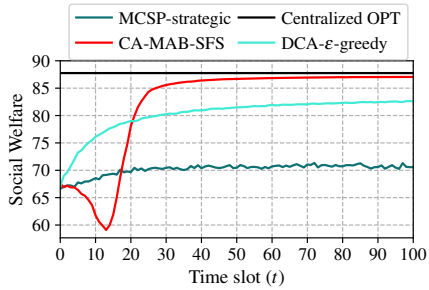


Fig. 4. Social welfare as a function of the time step t.

each MU and the MCSP, we consider the following offline approaches:

- *Offline Centralized Optimization*, which is abbreviated as *Centralized OPT*: For this approach, an optimization problem of the social welfare is formulated with complete information $\mathcal{I}$. This approach is based on the centralized optimization method proposed in [12], which is adapted to maximize the social welfare (17). The optimal solution is calculated using a solver from the OR-Tools [40].
- *Offline Game-Based Solution*, which is abbreviated as *Offline Game-Based*, as described in Section IV-A and Algorithm 3. We adapted the stable matching approach from [16] to our considered scenario. The complete information is available, therefore the payment of the MUs is calculated based on the actual effort required to perform the task, as specified in (4).

Additionally, we consider the following baseline algorithms which do not require complete information:

- *Decentralized ϵ-greedy multi-armed bandit (D-ϵ-greedy)*: Each MU uses the decaying ϵ-greedy online-learning algorithm [38] to learn the effort and utility for each task in a decentralized way. In case the sensing offer of the MU is rejected, the MU's utility is assumed to be zero. The exploration of tasks is performed according to a probability ϵ which is decreasing over time.
- *Decentralized collision-avoidance ϵ-greedy online-learning (DCA-ϵ-greedy)*: Similar to the decaying ϵ-greedy multi-armed bandit, each MU learns the effort and utility for each task using the ϵ-greedy online learning strategy. Additionally, each MU learns an acceptance probability representing the MCSP's preferences for each task type to avoid collisions with other MUs. The expected utility (12) and effort are weighted with the acceptance probability for the decision making.
- *Only MCSP-strategic*: Each MU randomly selects a task type z and sends a sensing offer to this task type. The payment proposal is calculated at the MUs using the average of the past efforts. The MCSP selects the MU with the lowest payment proposal.

C. Evaluation Setup

For the simulations, the parameters listed in Table II are considered, unless otherwise specified. The number K of MUs is chosen to be $K = 100$. The number N of tasks is chosen from the interval $[50, 100]$, whereas $Z = 10$ different task types are available. The sensing time varies every time slot for each MU and is drawn from a normal distribution with mean $\bar{\tau}^{\text{sense}}_{k,z}$ and standard deviation $10\,\text{s}$. The mean communication rate is randomly drawn from the interval $[10, 40]\,\text{Mbit}\,\text{s}^{-1}$, which corresponds to the mean communication time $\bar{\tau}^{\text{comm}}_{k,z} = [0.025, 0.1]\,\text{s}\,\text{Mbit}^{-1} \cdot s_z$. The communication time varies in every time slot for each MU, and it is drawn from a normal distribution with mean $\bar{\tau}^{\text{comm}}_{k,z}$ and standard deviation $0.01\,\text{s}\,\text{Mbit}^{-1}$. The mean CPU frequency available at each MU is $f^{\text{local}}_k \in [1, 2]$ GHz. Each time slot, it is drawn from a Gaussian distribution with the mean f^{local}_k and standard deviation 100 MHz. For each figure, 100 Monte-Carlo iterations were performed and the results are averaged.

D. Results and Discussion

We assess the energy efficiency of the proposed CA-MAB-SFS algorithm and the baseline algorithms in Fig. 2. The energy consumption is normalized to the size of the task result s_z, i.e., the energy efficiency is given by the energy consumed for each bit of the task result. The energy efficiency of the proposed CA-MAB-SFS is slightly lower than the baseline algorithms for $t < 20$. This is due to the strategic free sensing mechanism in the CA-MAB-SFS algorithm, where MUs explore task types for free. The MCSP prefers the MUs which perform the task for free over the most energy-efficient MUs, and therefore does not select the most efficient MU in this case. The exploration of task types is challenging due the competition between MUs, which initially causes poorer performance of the CA-MAB-SFS in the learning phase for $t < 20$. This degraded performance is due to the fact that the strategic free sensing activates simultaneously for multiple MUs. After the initial exploration of all Z task types, multiple MUs activate the free sensing mechanism (see Algorithm 1 line 11) simultaneously. When the exploration rate and the strategic free sensing reduces for $t > 20$, the CA-MAB-SFS shows a fast improvement in terms of energy efficiency. Fig. 2 demonstrates that for $t > 50$, the proposed CA-MAB-SFS algorithm achieves a $7.5\,\%$ increase in energy efficiency compared to the DCA-ϵ-greedy algorithm and an $11.5\,\%$ increase compared to the MCSP-strategic algorithm. Furthermore, the performance of the CA-MAB-SFS algorithm is within $1.2\,\%$ of the Centralized OPT algorithm which requires complete information. The D-ϵ-greedy algorithm is not shown, as its performance is significantly worse than its improved version given by DCA-ϵ-greedy.

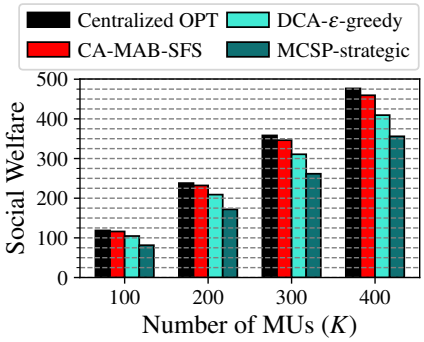


Fig. 5. Social welfare for an increasing network size for $K = N$, $Z = 10$.

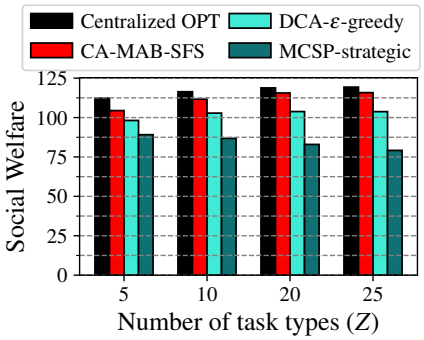


Fig. 6. Social welfare as a function of the number Z of task types, $K = N = 100$.

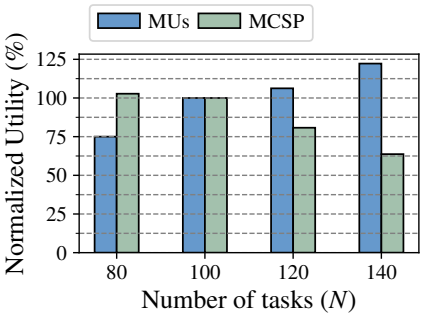


Fig. 7. Normalized utility of the MUs and the MCSP using CA-MAB-SFS as a function of N, $K = 100$, $Z = 20$.

The average time required to complete the tasks is shown in Fig. 3. For $t > 50$, the proposed CA-MAB-SFS algorithm outperforms the DCA-ϵ-greedy by $16\,\%$ and the MCSP-strategic by $41\,\%$. It achieves a slightly lower average task completion time than the Centralized OPT algorithm. This is due to the fact that the cost factor α_k of the MUs for the time is higher than the cost factor β_k for the energy, therefore the MUs prefer to execute tasks which require a lower completion time. The Centralized OPT algorithm maximizes the social welfare and therefore assigns tasks to MUs without considering their individual preferences, which will not yield the time-optimal result. Initially, the CA-MAB-SFS algorithm is slightly worse than the baseline algorithms due to the strategic free sensing procedure, but then outperforms the baseline algorithms significantly.

Figure 4 depicts the achieved social welfare of the different algorithms. The achievable maximum of the social welfare is given by the task assignment of the Centralized OPT algorithm. The proposed CA-MAB-SFS shows a good convergence to the social welfare maximum, whereas the DCA-ϵ-greedy and the MCSP-strategic algorithm are not able to converge to the optimum. The DCA-ϵ-greedy online learning achieves a $7.2\,\%$ lower social welfare than the optimum and the MCSP-strategic a $22\,\%$ lower social welfare. As in the Fig. 3, the decrease in social welfare of the proposed CA-MAB-SFS for $t < 20$ is due to the strategic free sensing mechanism, where some MUs execute tasks for free to learn more about the different task types.

The impact of the number K of MUs and the number N of tasks on the social welfare is shown in Fig. 5 for $t = 1000$. It can be seen that even for large MCS network sizes $K = N = 400$, the proposed CA-MAB-SFS is within $2\,\%$ of the optimal social welfare given by the Centralized OPT algorithm, whereas the DCA-ϵ-greedy achieves $14\,\%$ less social welfare. For larger networks, CA-MAB-SFS achieves near-optimal social welfare, while DCA-ϵ-greedy is $18\,\%$ below optimum.

The impact of the heterogeneity of tasks is shown in Fig. 6. The number Z of task types is varied while keeping the number K of MUs and the number N of tasks constant. We observe for all values of Z that the proposed algorithm achieves a near optimal performance within $6\,\%$ of the social welfare optimum.

Next, we analyze the effect of the competition between the MUs by varying the ratio K/N between the number of tasks and the number of MUs. The utility of the MUs and the MCSP using CA-MAB-SFS is shown in Fig. 7 for a varying ratio between MUs and tasks. For an increased competition between the MUs, i.e., less MUs than tasks ($K/N < 1$), we can see that the utility of the MUs decrease whereas the MCSP's utility increases. This is due to the fact that MUs with a lower payment proposal are selected by the MCSP and therefore the average payment between for each task decreases. When fewer MUs compete ($K/N > 1$), the utility of the MUs increases as they more frequently select tasks with higher payments.

To assess the stability of the solution, we depict the number of blocking pairs in Figure 8. Note that fewer blocking pairs result in more MUs satisfied with the task assignment. The proposed algorithm converges to zero blocking pairs, which demonstrates that CA-MAB-SFS converges to the stable solution, as shown in Theorem 2. The regular DCA-ϵ-greedy algorithm, which does not consider the competition between the MUs, leaves $60\,\%$ of MUs that could improve by changing the task assignment. As the MCSP-strategic algorithm does not consider the utility of the MUs, more than $80\,\%$ of the MUs are not satisfied with the task assignment.

To understand the impact of the collision-avoidance parameter λ on the performance, we varied λ in Fig. 9. For $\lambda = 0$, we observe a faster initial learning for $t < 12$. This is due to the fact that the MUs' suboptimal decisions are not repeated. However, this configuration does not converge to the maximum social welfare. For $\lambda = 0.4$, we observe a significantly lower learning speed, as $40\,\%$ of the MUs in average repeat the same decision as in the last time slot, which is ineffective. The collision-avoidance parameter therefore controls the trade-off between initial learning speed and convergence. Lower values of λ exhibit a higher initial learning speed, but may converge much slower. Higher values of λ have a lower initial learning speed, but converge faster. For our simulations, we chose $\lambda = 0.1$ as it empirically yields the best results.

Fig. 10 shows the cumulative number of free sensing offers, i.e., how many sensing offers without payment proposal have been sent. To analyze the impact of the free sensing parameter ϵ^a, we analyze the cumulative number of free sensing proposals. In our analysis of CA-MAB-SFS, we clearly see two phases: The phase with free sensing offers $t \leq \epsilon^{\mathrm{e}}$, and the phase without free sensing offers $t > \epsilon^{\mathrm{e}}$. In the phase with free sensing offers, MUs submit a free sensing offer after their

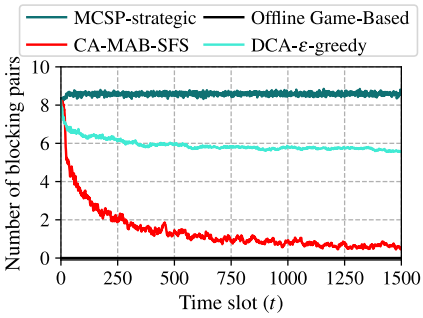


Fig. 8. Number of blocking pairs as a function of the time step t, $K = 100, N = 10, Z = 10$.

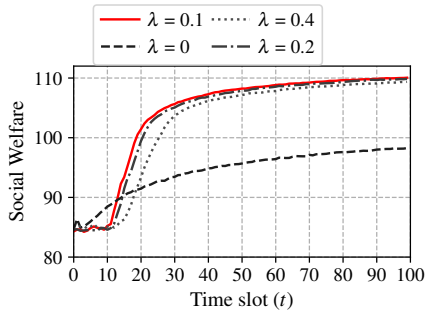


Fig. 9. Social welfare as a function of the time step t for varying collision-avoidance parameters λ.

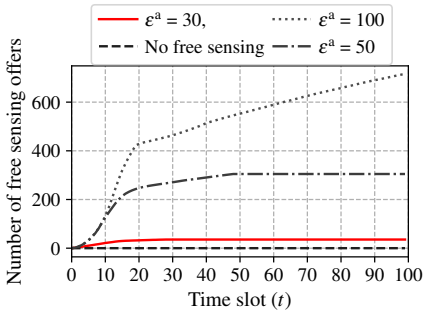


Fig. 10. Number of free sensing offers as a function of time step t for varying SFS parameters.

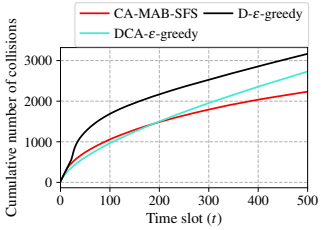


Fig. 11. Cumulative number of collisions as a function of time step t, $K = 200, N = 20, Z = 10$.

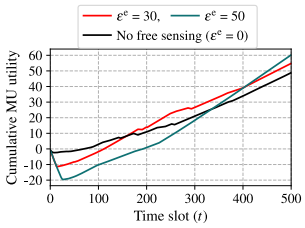


Fig. 12. Cumulative utility of the MUs as a function of the time step t for varying free-sensing parameters.

respective rejection threshold ϵ^{a} is exceeded. This is done to ensure that the MU will be accepted by the MCSP and the effort estimate will improve. From Fig. 10, we can see that increasing ϵ^a leads to a higher number of cumulative free sensing offers. The number of free sensing offers does not increase for $t > \epsilon^a$, so it does not increase indefinitely.

Fig. 11 shows the cumulative number of collisions, i.e., how many MUs are rejected by the MCSP because too many MUs propose to the same task type. For $t < 150$, the proposed CA-MAB-SFS algorithm suffers from a slightly higher number of collisions than the DCA-ϵ-greedy algorithm. This is due to the free sensing mechanism, which stimulates the MUs' exploration of the task types. During this exploration, more collisions might occur, as MUs try to estimate the expected effort for different task types. For $t = 500$, the proposed CA-MAB-SFS algorithm reduces the cumulative number of collisions by $33\,\%$ compared to the DCA-ϵ-greedy algorithm and by $47\,\%$ compared to the D-ϵ-greedy algorithm.

Fig. 12 shows the cumulative utility of an MU depending on different free-sensing parameters. The cumulative utility is averaged over all MUs. Clearly, the strategic free sensing mechanism requires an initial investment from the MU's perspective, as the MU performs tasks without payment from the MCSP. Performing tasks for free results in a negative utility of the MU at the beginning. For $\epsilon^{\mathrm{e}} = 0$, i.e., no free sensing, the utility is slightly negative for $t < 80$ as the estimates of the efforts are incorrect in the beginning. For $\epsilon^{\mathrm{e}} = 50$, the cumulative utility of the MU reaches a minimum of -20, but for $t = 500$, it achieves a $20\,\%$ higher cumulative utility for the MU than the version without free sensing. The free sensing parameter ϵ^{e} controls the tradeoff between the initial investment from the MUs and the long term reward achieved by the better exploration of task types. A higher value of ϵ^{e} leads to a lower utility of the MU in the beginning, but leads to a higher cumulative utility in the long run.

VI. Conclusion and Future Work

In this paper, we have studied the assignment of tasks in MCS. We have analyzed the conflicting interests of the MCSP and the MUs, the statistical nature of the tasks and MU's characteristics, as well as the competition between MUs. To consider the conflicting goals of the MCSP and MUs, we have formulated a matching-based task assignment game. We have proposed a novel decentralized online learning algorithm for the task assignment game, termed CA-MAB-SFS, which incorporates an innovative free sensing strategy. We have then proven its convergence to a stable task assignment, i.e., an assignment where neither the MUs nor the MCSP can improve. The stable regret, i.e., the loss of the online learning compared to having complete information, is bounded by a sublinear function and decreases to zero. Furthermore, we showed that the computational complexity for each MU and the MCSP is low. Simulation results show that, compared to the popular decentralized ϵ-greedy online learning approach, our proposed CA-MAB-SFS algorithm does not only reduce the average completion time of tasks by $16\,\%$, but also enhances the energy efficiency of the MCS system by up to $7.5\,\%$. We have also shown that the number of blocking pairs, i.e., the number of MUs that would improve by deviating from the task assignment, converges to zero. Furthermore, we have proven that our proposed CA-MAB-SFS converges to the maximum of the social welfare, whereas state-of-the-art online learning approaches are not able to reach it.

Future works could analyze how to assign multiple MUs to a task in parallel. When multiple MUs perform a task in parallel, the interactions between the assigned MUs in terms of willingness to cooperate or social relationships may affect the results. Another interesting direction for future research is the use of data inference methods, e.g., inspired by [41], to increase the coverage of sensing data provided by MCS and to overcome missing data when only a small number of MUs is participating. Additionally, an interesting area for future works is the use of non-orthogonal multiple access schemes which requires the research of novel resource allocation schemes to mitigate interference between sensing and communication. An exciting future research area is the application of multi-armed bandits to handle non-stationary probability distributions in MCS. Finally, a fully-fledged implementation of the proposed

approach over a real-world testbed, will be also of interest for future work.

REFERENCES

[1] Statista, "Forecast number of mobile devices worldwide from 2020 to 2025 (in billions)," https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/, accessed: 28.07.2022, 2021.

[2] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A Survey on Mobile Crowdsensing Systems: Challenges, Solutions, and Opportunities," *IEEE Commun. Surveys & Tutorials*, vol. 21, no. 3, pp. 2419–2465, Apr. 2019.

[3] J. Nie, J. Luo, Z. Xiong, D. Niyato, and P. Wang, "A Stackelberg Game Approach Toward Socially-Aware Incentive Mechanisms for Mobile Crowdsensing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 724–738, Dec. 2019.

[4] S. Wang, M. Chen, Z. Yang, C. Yin, W. Saad, S. Cui, and H. V. Poor, "Distributed Reinforcement Learning for Age of Information Minimization in Real-Time IoT Systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 501–515, Jan. 2022.

[5] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 29–35, Aug. 2014.

[6] W. Gong, B. Zhang, and C. Li, "Task Assignment in Mobile Crowdsensing: Present and Future Directions," *IEEE Network*, vol. 32, no. 4, pp. 100–107, Mar. 2018.

[7] S. Dongare, A. Ortiz, and A. Klein, "Deep reinforcement learning for task allocation in energy harvesting mobile crowdsensing," in *Proc. of the IEEE Global Commun. Conf. (GLOBECOM)*, Rio de Janeiro, Dec. 2022, pp. 269–274.

[8] Z. Wang, J. Hu, R. Lv, J. Wei, Q. Wang, D. Yang, and H. Qi, "Personalized Privacy-Preserving Task Allocation for Mobile Crowdsensing," *IEEE Trans. Mobile Computing*, vol. 18, no. 6, pp. 1330–1341, Jul. 2019.

[9] K. Ahuja and M. V. d. Schaar, "Dynamic matching and allocation of tasks," *ACM Trans. Economics and Computation*, vol. 7, no. 4, pp. 1–27, Oct. 2019.

[10] J. Wang, L. Wang, Y. Wang, D. Zhang, and L. Kong, "Task Allocation in Mobile Crowd Sensing: State-of-the-Art and Future Opportunities," *IEEE Internet of Things J.*, vol. 5, no. 5, pp. 3747–3757, Aug. 2018.

[11] X. Gong, X. Chen, J. Zhang, and H. V. Poor, "Exploiting Social Trust Assisted Reciprocity (STAR) Toward Utility-Optimal Socially-Aware Crowdsensing," *IEEE Trans. on Signal and Information Processing over Networks*, vol. 1, no. 3, pp. 195–208, Aug. 2015.

[12] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User Recruitment for Mobile Crowdsensing over Opportunistic Networks," in *Proc. of the IEEE Conf. on Computer Commun. (INFOCOM)*, Hong Kong, China, Apr. 2015, pp. 2254–2262.

[13] F. Yucel and E. Bulut, "Online Stable Task Assignment in Opportunistic Mobile Crowdsensing With Uncertain Trajectories," *IEEE Internet of Things J.*, vol. 9, no. 11, pp. 9086–9101, Oct. 2022.

[14] B. Simon, S. Dongare, T. Mahn, A. Ortiz, and A. Klein, "Delay- and Incentive-Aware Crowdsensing: A Stable Matching Approach for Coverage Maximization," in *Proc. of the IEEE Int. Conf. Commun. (ICC)*, Seoul, May 2022.

[15] Y. Wang, Z. Cai, Z.-H. Zhan, Y.-J. Gong, and X. Tong, "An Optimization and Auction-Based Incentive Mechanism to Maximize Social Welfare for Mobile Crowdsourcing," *IEEE Trans. Computational Social Syst.*, vol. 6, no. 3, pp. 414–429, Apr. 2019.

[16] G. Yang, B. Wang, X. He, J. Wang, and H. Pervaiz, "Competition-Congestion-Aware Stable Worker-Task Matching in Mobile Crowd Sensing," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3719–3732, 2021.

[17] M. Xiao, J. Wu, L. Huang, R. Cheng, and Y. Wang, "Online Task Assignment for Crowdsensing in Predictable Mobile Social Networks," *IEEE Trans. Mobile Computing*, vol. 16, no. 8, pp. 2306–2320, Oct. 2017.

[18] X. Wang, R. Jia, X. Tian, and X. Gan, "Dynamic Task Assignment in Crowdsensing with Location Awareness and Location Diversity," in *Proc. of the IEEE Conf. on Computer Commun. (INFOCOM)*, Honolulu, USA, Apr. 2018, pp. 2420–2428.

[19] J. Zhang and X. Zhang, "Multi-Task Allocation in Mobile Crowd Sensing with Mobility Prediction," *IEEE Trans. on Mobile Computing*, pp. 1081–1094, Jun. 2021.

[20] H. Gao, H. Xu, L. Li, C. Zhou, H. Zhai, Y. Chen, and Z. Han, "Mean Field Game based Dynamic Task Pricing in Mobile Crowd Sensing," *IEEE Internet of Things J.*, pp. 18 098–18 112, Sep. 2022.

[21] M. Xiao, B. An, J. Wang, G. Gao, S. Zhang, and J. Wu, "CMAB-based Reverse Auction for Unknown Worker Recruitment in Mobile Crowdsensing," *IEEE Trans. Mobile Computing*, pp. 3502–3518, Feb. 2021.

[22] S. Dongare, A. Ortiz, and A. Klein, "Federated Deep Reinforcement Learning for Task Participation in Mobile Crowdsensing," in *Proc. of the IEEE Global Commun. Conf. (GLOBECOM)*, Kuala Lumpur, Malaysia, Dec. 2023.

[23] C. Xu and W. Song, "Decentralized Task Assignment for Mobile Crowdsensing With Multi-Agent Deep Reinforcement Learning," *IEEE Internet of Things Journal*, vol. 10, no. 18, pp. 16 564–16 578, Sep. 2023.

[24] A. Magesh and V. V. Veeravalli, "Decentralized Heterogeneous Multi-Player Multi-Armed Bandits With Non-Zero Rewards on Collisions," *IEEE Trans. on Information Theory*, vol. 68, no. 4, pp. 2622–2634, Dec. 2022.

[25] C. Shi and C. Shen, "Multi-player multi-armed bandits with collision-dependent reward distributions," *IEEE Trans. on Signal Processing*, vol. 69, pp. 4385–4402, Jul. 2021.

[26] L. T. Liu, F. Ruan, H. Mania, and M. I. Jordan, "Bandit Learning in Decentralized Matching Markets." *J. Mach. Learn. Res.*, vol. 22, pp. 1–50, Sep. 2021.

[27] L. T. Liu, H. Mania, and M. Jordan, "Competing Bandits in Matching Markets," in *Proc. of the Twenty Third Int. Conf. on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 26–28 Aug 2020, pp. 1618–1628. [Online]. Available: https://proceedings.mlr.press/v108/liu20c.html

[28] M. H. Cheung, F. Hou, J. Huang, and R. Southwell, "Distributed Time-Sensitive Task Selection in Mobile Crowdsensing," *IEEE Trans. on Mobile Computing*, vol. 20, no. 6, pp. 2172–2185, Feb. 2021.

[29] Y. Huang, H. Chen, G. Ma, K. Lin, Z. Ni, N. Yan, and Z. Wang, "OPAT: Optimized Allocation of Time-Dependent Tasks for Mobile Crowdsensing," *IEEE Trans. on Industrial Informatics*, vol. 18, no. 4, pp. 2476–2485, Jul. 2022.

[30] X. Li, G. Feng, Y. Liu, S. Qin, and Z. Zhang, "Joint Sensing, Communication, and Computation in Mobile Crowdsensing Enabled Edge Networks," *IEEE Transactions on Wireless Communications*, vol. 22, no. 4, pp. 2818–2832, 2023.

[31] A. Capponi, C. Fiandrino, D. Kliazovich, P. Bouvry, and S. Giordano, "A cost-effective distributed framework for data collection in cloud-based mobile crowd sensing architectures," *IEEE Trans. on Sustainable Computing*, vol. 2, no. 1, pp. 3–16, Feb. 2017.

[32] G. Gao, J. Wu, M. Xiao, and G. Chen, "Combinatorial Multi-Armed Bandit Based Unknown Worker Recruitment in Heterogeneous Crowdsensing," in *Proc. of the IEEE Conf. on Computer Commun. (INFOCOM)*, Toronto, Canada, Jul. 2020, pp. 179–188.

[33] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: fundamentals and applications," *IEEE Commun. Magazine*, vol. 53, no. 5, pp. 52–59, May 2015.

[34] L. S. Shapley and M. Shubik, "The Assignment Game I: The Core," *Int. J. Game Theory*, vol. 1, no. 1, p. 111–130, Dec. 1971. [Online]. Available: https://doi.org/10.1007/BF01753437

[35] S. H. Cen and D. Shah, "Regret, stability & fairness in matching markets with bandit learners," in *Proc. of the Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, Valencia, Spain, Mar. 2022, pp. 8938–8968.

[36] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press, 2007.

[37] A. E. Roth, "Deferred acceptance algorithms: History, theory, practice, and open questions," *Int. Journal of Game Theory*, vol. 36, no. 3, pp. 537–569, Jan. 2008.

[38] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2, pp. 235–256, May 2002.

[39] T. Mahn and A. Klein, "A Global Orchestration Matching Framework for Energy-Efficient Multi-Access Edge Computing," in *Proc. of the IEEE Int. Conf. on Cloud Networking (CloudNet)*, Cookeville, USA, Nov. 2021, pp. 11–18.

[40] L. Perron and V. Furnon, "OR-Tools," Google. [Online]. Available: https://developers.google.com/optimization/, accessed: 30.7.2022

[41] J. Huo, L. Wang, X. Wen, D. Gesbert, and Z. Lu, "Cost-Efficient Vehicular Crowdsensing Based on Implicit Relation Aware Graph Attention Networks," *IEEE Transactions on Industrial Informatics*, pp. 1–11, Sep. 2023.

APPENDIX A
PROOF OF THEOREM 1

The core idea is to bound the probability $\mathbb{P}(\mathbf{x}_t \notin X^{\text{stable}})$ of the event that no stable matching is achieved until time T. If a stable matching is reached, the stable regret of all MUs

will be zero, otherwise, we bound the regret by the maximum regret over all MUs which is given by

$$\Delta_k = \max_{z=1,\dots,Z}\{\bar{U}_k^{\mathrm{MU,stable}} - \bar{U}_{k,z}^{\mathrm{MU}}\}. \tag{18}$$

Therefore, we formulate the following stable regret bound:

$$R_k(T) \leq \Delta_k \sum_{t=0}^{T} \mathbb{P}(\mathbf{x}_t \notin X^{\mathrm{stable}}) \tag{19}$$

As Δ_k can be directly calculated from the MUs expected utilities $\bar{U}_{k,z}^{\mathrm{MU}}$, we only need to bound the probability of an unstable matching $\mathbb{P}(\mathbf{x}_t \notin X^{\mathrm{stable}})$. The following events $E_{1,t}$ and $E_{2,t}$ prevent a stable matching:

- $E_{1,t}$: At least one user is exploring according to ϵ-greedy and not selecting its stable task a_k^{stable} in t.
- $E_{2,t}$: Either one MU has statistical ranking mistakes, i.e. its estimates of the utility result in a sensing offer for a suboptimal task type, or there were no statistical ranking mistakes but the matching at time $t-1$ was unstable.

Considering $E_{1,t}$ and $E_{2,t}$, we use the following bound

$$\mathbb{P}(\mathbf{x}_t \notin X^{\mathrm{stable}}) \leq \mathbb{P}(E_{1,t}) + \mathbb{P}(\overline{E}_{1,t})\mathbb{P}(E_{2,t}), \tag{20}$$

with $\overline{E}_{1,t}$ as the complementary event of $E_{1,t}$. The stable regret bound is given by

$$R_k(T) \leq \Delta_k \left(\sum_{t=1}^{T} \mathbb{P}(E_{1,t}) + \sum_{t=1}^{T} \mathbb{P}(E_{2,t}) \right), \tag{21}$$

using $\mathbb{P}(\overline{E}_{1,t}) \leq 1$. In the following, we derive the probability of the events $E_{1,t}$ and $E_{2,t}$ separately.

Derivation of $\mathbb{P}(E_{1,t})$: The probability of $E_{1,t}$ is given by

$$\begin{aligned}\mathbb{P}(E_{1,t}) &\leq 1 - \left((1-\epsilon_t) + \epsilon_t \frac{1}{Z} \right)^K \\ &= 1 - \left(1 - \epsilon_t \frac{Z-1}{Z} \right)^K \end{aligned} \tag{22}$$

which is the complementary event of all K MUs exploiting in time t or randomly selecting the stable task out of the N tasks. The summation over all time slots t yields

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{P}(E_{1,t}) &\leq \sum_{t=1}^{T} 1 - \left(1 - \epsilon_t \frac{Z-1}{Z} \right)^K \\ &= T - \sum_{t=1}^{T} \left(1 - \epsilon_t \frac{Z-1}{Z} \right)^K. \end{aligned} \tag{23}$$

We bound the inner term of the sum by Bernoulli's inequality

$$\left(1 - \epsilon_t \frac{Z-1}{Z} \right)^K \geq 1 + K \cdot \epsilon_t \frac{Z-1}{Z}, \tag{24}$$

that holds for $\epsilon_t \frac{Z-1}{Z} \leq 1$, which can be easily checked. Therefore,

$$\sum_{t=1}^{T} \mathbb{P}(E_{1,t}) \leq \sum_{t=1}^{T} K \cdot \epsilon_t \frac{Z-1}{Z} \tag{25}$$

For a sufficiently fast decaying ϵ_t, e.g. $\epsilon_t = \min\{1, 1/t\}$, we can show that:

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{P}(E_{1,t}) &\leq K \frac{Z-1}{Z} \sum_{t=1}^{T} \min\left(1, \frac{1}{t}\right) \\ &\leq K \frac{Z-1}{Z} (\log(T) + 1). \end{aligned} \tag{26}$$

Derivation of $\mathbb{P}(E_{2,t})$: Note that $\mathbb{P}(E_{2,t})$ is the probability of MUs having statistical ranking mistakes or not achieving a stable matching. For this proof, we assume that, as in real applications, the utility of the MUs is limited to finite values in an interval $[U_{\min}, U_{\max}]$. Therefore, we can define $\Delta \mathrm{U} = U_{\max} - U_{\min}$. The proof is an adapted version of the proof in [26], using arguments for ϵ-greedy MABs from [38]. The authors of [26] show that

$$\sum_{t=1}^{T} \mathbb{P}(E_{2,t}) \leq 4 \frac{Z^5 K^2}{\rho^{Z^4+1}} \log(T)(x_0 + 12), \tag{27}$$

where $x_0 = \sum_{t=1}^{T} P\{\hat{Q}_{k,t}(i) > \hat{Q}_{k,t}(j) \cap x_{k,n,t} = 1\} \leq \sum_{t=1}^{T} P\{\hat{Q}_{k,t}(i) > \hat{Q}_{k,t}(j)\}$ denotes the expected number of statistical ranking mistakes up to T and $\rho = (1-\lambda)\lambda^{Z-1}$.

The expected number k_t of sensing offers to a suboptimal task type in time slot t during the exploration phase is defined as

$$k_t = \frac{1}{Z} \sum_{t'=1}^{t} \epsilon_{t'} \leq \frac{1}{Z} (\log(t) + 1). \tag{28}$$

Using Hoeffding's inequality and the definition in (28), we can show that

$$\begin{aligned}\mathbb{P}\{\hat{Q}_{k,t}(i) > \hat{Q}_{k,t}(j)\} &\leq 2e^{-\frac{k_t \Delta^2}{\Delta \mathrm{U}}} \\ &= 2e^{\frac{\Delta^2}{Z \Delta \mathrm{U}}} \frac{1}{t^{\frac{\Delta^2}{Z \Delta \mathrm{U}}}}, \end{aligned} \tag{29}$$

with $\Delta = \min_{i,j \in, i \neq j}\{\bar{U}_{k,i}^{\mathrm{MU}} - \bar{U}_{k,j}^{\mathrm{MU}}\}$. The expected number x_0 of statistical ranking mistakes up to T can be bounded by

$$\begin{aligned} x_0 &\leq 2e^{\frac{\Delta^2}{Z \Delta \mathrm{U}}} \sum_{t=1}^{T} \frac{1}{t^{\frac{\Delta^2}{Z \Delta \mathrm{U}}}} \leq 2e^{\frac{\Delta^2}{Z \Delta \mathrm{U}}} \int_{t=0}^{T} \frac{1}{t^{\frac{\Delta^2}{Z \Delta \mathrm{U}}}} \, \mathrm{d}t \\ &= \frac{2e^{\frac{\Delta^2}{Z \Delta \mathrm{U}}}}{1 - \frac{\Delta^2}{Z \Delta \mathrm{U}}} T^{1 - \frac{\Delta^2}{Z \Delta \mathrm{U}}}. \end{aligned} \tag{30}$$

Using (26), (27) and (30), the total regret bound can then be calculated as

$$\begin{aligned} R_k(T) \leq \Delta_k \Bigg(& K \frac{Z-1}{Z} (\log(T) + 1) \\ & + 8 \frac{Z^5 K^2}{\rho^{Z^4+1}} \log(T) \left(\frac{e^{\frac{\Delta^2}{Z \Delta \mathrm{U}}}}{1 - \frac{\Delta^2}{Z \Delta \mathrm{U}}} T^{1 - \frac{\Delta^2}{Z \Delta \mathrm{U}}} + 6 \right) \Bigg). \end{aligned} \tag{31}$$

One can see that the leading order of the stable regret bound is a sublinear function which is given by

$$O\left(\Delta_k \frac{8 Z^5 K^2 e^{\frac{\Delta^2}{Z \Delta \mathrm{U}}}}{\rho^{Z^4+1} (1 - \frac{\Delta^2}{Z \Delta \mathrm{U}})} \log(T) T^{1 - \frac{\Delta^2}{Z \Delta \mathrm{U}}} \right). \tag{32}$$

B Proof of Theorem 2

Starting from (20), we formulate

$$\sum_{t=1}^{T}\mathbb{P}(\boldsymbol{X}_t \notin X^{\text{stable}}) \leq \sum_{t=1}^{T}\mathbb{P}(E_{1,t}) + \sum_{t=1}^{T}\mathbb{P}(\overline{E}_{1,t})\mathbb{P}(E_{2,t}). \tag{33}$$

We use

$$T\cdot\mathbb{P}(\boldsymbol{X}_T \notin X^{\text{stable}}) \leq \sum_{t=1}^{T}\mathbb{P}(\boldsymbol{X}_t \notin X^{\text{stable}}) \tag{34}$$

as $\mathbb{P}(\boldsymbol{X}_t \notin X^{\text{stable}})$ is monotonically decreasing in t. Using (26), (27) we can show that

$$\begin{aligned}\mathbb{P}(\boldsymbol{X}_T \notin X^{\text{stable}}) \leq \frac{1}{T}\Bigg(K\frac{Z-1}{Z}(\log(T)+1) \\ + 8\frac{Z^5K^2}{\rho^{Z^4+1}}\log(T)\Bigg(\frac{e^{\frac{\Delta^2}{Z\Delta \mathrm{U}}}}{1-\frac{\Delta^2}{Z\Delta \mathrm{U}}}T^{1-\frac{\Delta^2}{Z\Delta \mathrm{U}}}+6\Bigg)\Bigg)\end{aligned} \tag{35}$$

One can see that the leading order of the probability $\mathbb{P}(\boldsymbol{X}_T \notin X^{\text{stable}})$ is given by a function

$$O\Bigg(\frac{8Z^5K^2e^{\frac{\Delta^2}{Z\Delta \mathrm{U}}}}{\rho^{Z^4+1}(1-\frac{\Delta^2}{Z\Delta \mathrm{U}})}\frac{\log(T)}{T^{\frac{\Delta^2}{Z\Delta \mathrm{U}}}}\Bigg). \tag{36}$$