

Online Optimization for UAV-assisted Distributed Fog Computing in Smart Factories of Industry 4.0

Gilsoo Lee*, Walid Saad*, and Mehdi Bennis[†]

* Wireless@VT, Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA,

Emails: {gilsoolee, walids}@vt.edu.

[†] Centre for Wireless Communications, University of Oulu, Finland, Email: bennis@ee.oulu.fi.

Abstract—In this paper, the problem of unmanned aerial vehicle (UAV)-assisted fog computing in Industry 4.0 smart factories is studied. In particular, a novel online framework is proposed to enable a source UAV to offload computing tasks from ground sensors within a smart factory and allocate them to neighboring fog UAVs for distributed task computing, before the source UAV arrives at its destination. The online nature of the framework allows the UAVs to optimize their task allocation and decide on which neighbors to use for fog computing, even when the tasks are revealed to the source UAV in an online manner, and the information on future task arrivals is unknown. The proposed framework essentially maximizes the number of computed tasks by jointly considering the communication and computation latency. To solve the problem, an online greedy algorithm is designed and solved by using the primal-dual approach. Since the primal problem provides an upper bound of the original dual problem, the competitive ratio can be analytically derived as a function of the task sizes and the data rates of the UAVs. Simulation results show that the proposed online algorithm can achieve a near-optimal task allocation with an optimality gap that is no higher than 7.5% compared to the offline, optimal solution with complete knowledge of all tasks.

I. INTRODUCTION

In emerging smart factory scenarios (also known as Industry 4.0) [1], sensors will be able to detect malfunctions and send diagnostics signals to actuators in the factory. Therefore, factory systems need to be optimized to manage the process of sensory data transmission, low-latency computation, and proactive decision making in order to quickly react to new situations [2]. Therefore, some of the key challenges for enabling the smart factory vision include effective in-network computing and improvement of wireless connectivity to integrate physical and digital systems, i.e., networking and computation.

Computing sensory data in a timely manner is essential to operate a physical factory system. To this end, the concept of fog computing can be applied in cyber-physical smart factory systems where unmanned aerial vehicles (UAVs) are deployed and perform key functions such as data storage, computing, control, and transmission [3], [4]. Recently, the use of UAVs for wireless and computing scenarios has been studied in [5]–[11]. In particular, using UAVs as a relay is studied in [5]–[8]. The authors in [5] propose a framework that jointly optimizes UAV placement, user association, and uplink power control

so that UAVs can collect data from ground sensors. In [6], a UAV relaying scheme is studied to support a wireless cellular network by temporarily offloading traffic of overloaded cells into neighboring cells. In [7], UAVs are used as message ferries that collect information in wireless sensor networks and carry the data to the destination. The authors in [8] propose a framework using UAVs for data collection from ground sensor networks where relay nodes act as cluster heads for adjacent sensors, and the data collected by relays is transmitted to the UAVs. Also, airborne fog computing using a UAV is studied in [9]–[11]. In [9], the authors investigate a UAV-mounted cloudlet in which UAVs equipped with a computing processor offload and compute the tasks offloaded from ground devices. The authors in [10] study UAV-enabled wireless powered mobile edge computing system. The authors in [11] propose a relaying system that uses a UAV to store the processed data in a buffer and optimize the receiving and transmitting data size to minimize the energy consumption.

This prior art on UAV-based communications and computing [9]–[11], generally assumes that information on prospective computing tasks such as data size and arriving order is completely known. However, in practice, the information on tasks can be revealed gradually over time since sensory data is randomly generated. Hence, when a series of tasks are offloaded to a UAV, predicting prospective future tasks is often not possible. Moreover, instead of relying on a single UAV for computing, as done in [9]–[11], it is beneficial to leverage multiple, neighboring UAVs for distributed, fog-like computing of tasks. Consequently, unlike the existing literature [9]–[11] which assumes full information knowledge on tasks and single UAV computing models, our goal is to design an *online approach* to maximize the number of computed tasks on a network of multiple UAVs within a smart factory, when the information on tasks is revealed in an online manner.

The main contribution of this paper is a novel framework for distributed UAV-assisted fog computing. This framework allows tasks from ground sensors to be offloaded to a source UAV, which can subsequently allocate tasks to neighboring fog UAVs for computation before the source UAV reaches its destination. When the exact information on the offloaded tasks is unknown to the source UAV, it is challenging to decide which neighboring UAV has to compute which task. Therefore, an online optimization problem is formulated, and an online greedy algorithm is proposed to solve the problem. Then, the dual problem is derived to analyze the worst-case

This research was supported, in part, by the U.S. Office of Naval Research (ONR) under Grant N00014-15-1-2709, by the National Science Foundation under Grant CNS-1739642, by the Academy of Finland project CARMA, by 6Genesis Flagship (grant no. 318927), by the INFOTECH project NOOR, and by the Kvantum Institute strategic project SAFARI.

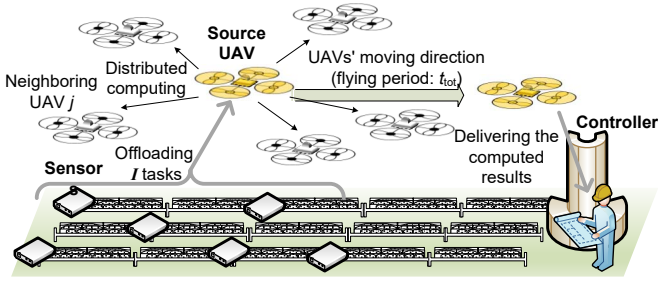


Fig. 1: Distributed fog computing framework to offload computational tasks from sensors and allocate the offloaded tasks to neighboring fog UAVs when the source UAV is moving to the destination. performance of the proposed algorithm. By doing so, the worst-case competitive ratio can be derived as a function of the task sizes and the communication performance of the UAVs. Simulation results show that the proposed online algorithm can maximize the number of computed tasks and achieve a performance that is near-optimal compared to an offline solution that has full information on tasks.

The rest of this paper is organized as follows. In Section II, we present the system model and formulate the proposed online problem. Section III presents our proposed solution. Simulation results are analyzed in Section IV while conclusions are drawn in Section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

Consider a smart factory in which a number of sensors monitor the status of the manufacturing process, thus generating a large data volume. For example, the generated sensory data can be used as input to machine learning algorithms, e.g., for classification, to predict any abnormality in the manufacturing process. Therefore, a number of computational tasks must be computed in order to make a decision on how to control the physical systems of the factory based on the information extracted from the data. However, due to the low computational capability of the sensors, it is not possible to compute the tasks locally at the sensors. Also, sensors are not able to transmit data over a long distance, and, hence, a flexible relay is necessary.

By using fog computing, computational tasks can be allocated to adjacent fog nodes that have higher computational capability than the sensors. In particular, fog-enabled UAVs can be used in a smart factory to gather the tasks from the sensors, compute the tasks, and deliver the computed results to the destination, e.g., a central factory controller that can control the actuators. Therefore, UAVs in smart factories can be seen as fog nodes in the fog layer of a generic fog computing architecture. Using fog UAVs for smart factories has several key advantages. First, UAVs can deliver better connectivity for the sensors. For example, when the sensors cannot access the uplink of terrestrial base stations due to high congestion, nearby UAV can be used to offload the data of sensors. Second, the transmit power of the sensors can be reduced if the fog UAVs move closely to the sensors, thus conserving sensor energy. Third, UAV networks can be readily deployed and

operated in a dynamic environment. Indeed, fixed network access points may not be able to support dynamic network changes when the production processes in smart factories are modified. However, fog UAVs can flexibly change their flying path and deliver data to moving destinations (e.g., equipment carried by factory engineers).

As shown in Fig. 1, the ground sensors generate a set \mathcal{I} of I tasks that are offloaded to a given UAV that we refer to hereinafter as *source UAV*. The role of the source UAV is to offload the computational task data from sensors, allocate the computational tasks to neighboring UAVs, and finally deliver the computed results to the destination. When tasks reach the source UAV, they are labeled by their arriving order i . For instance, a task that arrives a time instant i is denoted as task i . Since the source UAV processes the tasks using a first-input-first-output policy, it will sequentially compute the tasks. We also consider a set \mathcal{J} of UAVs that are neighbors to the source UAV. Each UAV $j \in \mathcal{J}$ is used to compute the allocated task i from the source UAV. We also consider that the set of neighboring UAVs \mathcal{J} is initially selected by the source UAV. In this regard, the source UAV selects the neighboring UAVs that are moving towards its same destination.

The source UAV allocates the computational tasks to other neighboring UAVs. Such distributed computing can reduce the overall computational latency when multiple tasks are computed. Also, to prevent excessive energy consumption of neighboring UAVs, it is assumed that only one task is allocated to one UAV. Therefore, if neighboring UAV j computes task i , it is denoted by $y_{ij} = 1$ and $y_{ij'} = 0, \forall j' \in \mathcal{J} \setminus \{j\}, \forall i \in \mathcal{I}$. Then, task allocation to neighboring UAV will incur a *transmission latency*. The data rate pertaining to the transmission of the data of task i to neighboring UAV j will be:

$$r_j = B \log_2 \left(1 + \frac{g_j P_t}{\sigma^2} \right), \quad (1)$$

where B is the bandwidth, σ^2 is the noise power, and g_j is the channel gain between the source UAV and neighboring UAV j . Therefore, when the data size of task i is d_i bits, the transmission latency becomes d_i/r_j . Once task i is received by neighboring UAV j , the *computational latency* is modeled as d_i/f_j where f_j is the computational speed of UAV j .

To transmit a task data from the source UAV to a neighboring UAV, device-to-device (D2D) communication can be used, in which the source UAV initiates a D2D link with one neighboring UAV at a time. The time period that the source UAV moves from the source to the destination is given by t_{tot} . When computing a certain number of tasks, the transmission and computation at all neighboring UAVs should be finished within t_{tot} . To determine t_{tot} , key features of UAV networks can be considered. For example, t_{tot} is determined depending on the mobility that is characterized by the speed and flying distance of the source UAV. Also, t_{tot} depends the different trajectories of the source UAV. While finding t_{tot} is out of the scope of this work, we focus on the efficient use of the given time budget t_{tot} by using distributed fog computing. Therefore, when the first task, $i = 1$, is being allocated, the latency

including transmission and computation of task 1 will be:

$$\sum_{j=1}^J d_1 \left(\frac{1}{r_j} + \frac{1}{f_j} \right) y_{1j} \leq t_{\text{tot}}. \quad (2)$$

Then, since the tasks are sequentially transmitted to the neighboring UAVs in the order of index i , there will be $i - 1$ transmissions before task i is transmitted. Therefore, the latency needed to complete task i , $\forall i \in [2, I]$, is

$$\sum_{i'=1}^{i-1} \sum_{j=1}^J d_{i'} \left(\frac{1}{r_j} \right) y_{i'j} + \sum_{j=1}^J d_i \left(\frac{1}{r_j} + \frac{1}{f_j} \right) y_{ij} \leq t_{\text{tot}}, \quad (3)$$

where the first term is the sum of the transmission latency of $i - 1$ tasks, and the second term is the transmission and computation of task i . Since tasks are allocated to and computed by neighboring UAVs in the order of index i , if task i is computed within period t_{tot} , then $i - 1$ tasks will also be computed in the given period. Given the defined system model, next, we formulate an online task allocation problem to study how tasks are distributed in the UAV-fog network.

B. Problem Formulation

Our goal is to allocate tasks to neighboring UAVs in order to complete the maximum number of tasks during the period t_{tot} needed for the source UAV to reach its destination. To compute the tasks, the source UAV must allocate each task to a neighboring UAV that achieves low transmission and computational latency. In practice, such computational tasks, having different data sizes, will dynamically arrive to the source UAV. As a result, the source UAV will be unable to know a priori the information on future tasks, and, therefore, optimizing the task distribution process becomes challenging. Under such uncertainty, selecting a neighboring UAV that computes a current task must also account for potential arrival of future tasks. To cope with the uncertainty of the future task arrivals while considering the data rate and computing capabilities of given neighboring UAVs, we introduce an *online optimization scheme* that can handle the problem of task allocation under uncertainty.

First, we formulate the following online task allocation problem whose goal is to maximize the number of computed tasks when the total latency is limited by t_{tot} :

$$(D) : \max_{\mathbf{y}} \quad \sum_{i=1}^I \sum_{j=1}^J y_{ij} \quad (4)$$

s.t. (2), (3),

$$\sum_{i=1}^I y_{ij} \leq 1, \forall j \in [1, J], \quad (5)$$

$$\sum_{j=1}^J y_{1j} \leq 1, \quad (6)$$

$$\sum_{j=1}^J (-y_{i-1j} + y_{ij}) \leq 0, \forall i \in [2, I], \quad (7)$$

where \mathbf{y} is the vector of y_{ij} , $\forall i \in \mathcal{I}, \forall j \in \mathcal{J}$. Hereinafter, this problem is called the dual problem. The number of the tasks and neighboring UAVs, i.e., I and J , respectively, are assumed

to be given parameters with $J \geq I$. Constraints (2) and (3) show that task i 's transmission and computation latency needs to be completed within time t_{tot} . Also, to satisfy (5), each neighboring UAV can compute at most one task. If task 1 is allocated to one of the neighboring UAVs, constraint (6) is satisfied. Constraint (7) implies that task i can be allocated to a neighboring UAV if the task allocation of task $i - 1$ is successful, i.e., $\sum_{j=1}^J y_{i-1j} = 1$. Otherwise, if $\sum_{j=1}^J y_{i-1j} = 0$, then, task i cannot be allocated to any UAV, and $\sum_{j=1}^J y_{ij} = 0$. Due to (6) and (7), it is shown that $\sum_{j=1}^J y_{ij} \leq 1, \forall i \in \mathcal{I}$, and, thus, one task is allocated to one of neighboring UAV.

Note that problem (D) is an *online optimization problem* and is challenging to solve using conventional, offline approaches. It is due the fact that the value of $d_i, \forall i$, is sequentially revealed. When the tasks that different sensors send to the source UAV have random size, the arrival sequence of d_i is assumed to be unpredictable and unknown.

At the moment when d_i is disclosed, the source UAV knows only the current and past tasks. However, the source UAV must make an *instant and irrevocable online decision* on which neighboring UAV will compute task i . Under such uncertainty on d_i , allocating tasks to existing neighboring UAVs must also account for potential arrival of new tasks. In fact, even if a given task allocation can compute an existing task successfully, it may have a detrimental effect on the allocation of incoming tasks. In particular, if a UAV having a high data rate and high computational speed is already assigned to compute a previous task, it may not be possible to compute a future task having a large size. Therefore, it is challenging to optimize the task allocation between incoming tasks and neighboring UAVs.

III. ONLINE TASK ALLOCATION STRATEGY

Our goal is to determine the decision \mathbf{y} so that the maximum number of sequentially arriving tasks is successfully computed by distributed UAV-enabled fog computing. When task size d_i is unpredictable, the decision is not trivial since the current decision may affect the task allocation of future tasks, and all tasks cannot be computed due to the limited time resource t_{tot} . Under such incomplete information, finding the optimal solution of (4) is challenging, and, as such, one has to seek an online, sub-optimal solution that is robust to uncertainty.

When solving an online optimization problem, it is difficult to achieve the optimal performance since the information is revealed to the algorithm over time. Therefore, the notion of a *competitive ratio* [12] from competitive analysis can be used to measure the performance of an online algorithm. It is an effective metric that compares the ratio between the the objective function's value achieved by an online algorithm and that of the offline optimal solution. In particular, a competitive ratio can be defined as α such that

$$1 \leq \frac{D_{\text{IP,OPT}}}{D_{\text{IP}}} \leq \alpha, \quad (8)$$

where $D_{\text{IP,OPT}}$ denotes the offline optimal solution (OPT) of problem (D) in the form of integer programming (IP), i.e., the maximum number of computed tasks with the integer solution

Algorithm 1 Task Allocation Algorithm

```
1 : Initialize  $y_{ij} = x_i = z_j = u_i = 0, \forall i, j$ , and  $\mathcal{J}_1 \leftarrow \mathcal{J}$ .
2 : for  $i \in \mathcal{I}$ 
3 :   Task  $i$  arrives.
4 :   Select UAV  $j_i^* = \operatorname{argmin}_j 1/r_j, j \in \mathcal{J}_i$ .
5 :   if (2) and (3) are satisfied, and  $\sum_j y_{i-1j} = 1$ ,
6 :      $y_{ij} \leftarrow 1$ .
7 :     Allocate task  $i$  to UAV  $j_i^*$ .
8 :      $x_I \leftarrow \max(x_I, \frac{1}{d_i/r_{j_i^*}})$ .
9 :   otherwise,
10 :     $y_{ij} \leftarrow 0$ .
11 :     $z_j \leftarrow \max(1 - x_I d_i/r_j, z_j, 0), \forall j \in \mathcal{J}$ .
12 :   end if
13 :    $\mathcal{J}_{i+1} \leftarrow \mathcal{J}_i \setminus \{j_i^*\}$ 
14 : end for
```

of y_{ij} . We will measure the performance of the proposed algorithm by observing the upper bound value defined by α .

To find the upper bound of problem (D), we use the structure of the primal and dual approach [12]. To this end, the optimization variable y_{ij} is relaxed to be linear, i.e., $y_{ij} \in [0, 1]$. By using the duality of linear programming, problem (D) can be rewritten as:

$$(P) : \min_{\mathbf{x}, \mathbf{z}, u_1} \sum_{i=1}^I t_{\text{tot}} x_i + \sum_{j=1}^J z_j + u_1, \quad (9)$$

$$\text{s.t.} \quad \left(\frac{1}{r_j} + \frac{1}{f_j} \right) d_i x_i + \left(\frac{d_i}{r_j} \right) \sum_{i'=i+1}^I x_{i'} + z_j + u_i - u_{i+1} \geq 1, \forall i \in [1, I-1], \forall j \in \mathcal{J}, \quad (10)$$

$$\left(\frac{1}{r_j} + \frac{1}{f_j} \right) d_I x_I + z_j + u_I \geq 1, \forall j \in \mathcal{J}, \quad (11)$$
$$x_i \geq 0, z_j \geq 0, u_i \geq 0, \quad (12)$$

where \mathbf{x} and \mathbf{z} are vectors with elements x_i and z_j , respectively. This problem is called the *primal problem*. In problem (9), $x_1, x_{i \geq 2}, z_j, u_1$ and $u_{i \geq 2}$ are the dual variables associated, respectively, with the constraints (2), (3), (5), (6), and (7) in problem (D). Next, we develop an online greedy algorithm to solve problems (D) and (P).

We propose an online greedy algorithm shown as Algorithm 1. Based on a general online optimization framework using the primal and dual approach of [12], we propose a novel algorithm specifically designed to solve our problems (D) and (P). When task i arrives to the source UAV, the original dual problem is solved by determining the value of y_{ij} . Also, other dual variables x_i, z_j , and u_i are updated in order to find the performance bound of the proposed online algorithm. At the initial step of Algorithm 1, all variables are set to 0, and the algorithm selects UAV j^* that has the highest data rate. Then, the algorithm makes a decision and updates the variables when task i arrives. The algorithm selects which UAV should compute task i . To this end, the algorithm selects UAV j_i^* having the smallest $1/r_j$ among the current set \mathcal{J}_i . \mathcal{J}_i is defined as the set of available UAVs that do not have any allocated task since one UAV computes at most one task. Also, to make a decision on whether to compute task i , constraints (2) and (3) are checked. If UAV j_i^* satisfies (2) and (3), then, task i is allocated to UAV j_i^* , and, thus, the algorithm sets

$y_{ij} = 1$. Meanwhile, x_I is updated to the value given by $\max(x_I, 1/(d_i/r_{j_i^*}))$ where $j_i^* = \operatorname{argmin}_j 1/r_j, j \in \mathcal{J}$. The update of x_I satisfies constraints (10) and (11). Otherwise, if the UAVs in \mathcal{J}_i do not satisfy (2) and (3), then, the tasks arriving after task i cannot be computed, i.e., $y_{ij} = 0$. In this case, $z_j, \forall j \in \mathcal{J}$, is updated to satisfy constraint (10) and (11). After the algorithm selects j_i^* for task i , j_i^* is removed from set \mathcal{J}_i regardless of the decision on the task allocation of y_{ij} . By doing so, all tasks in \mathcal{I} are matched to at most one of neighboring UAVs in \mathcal{J} .

After running Algorithm 1, the values of (4) and (9) are denoted by P_{LP} and D_{IP} , respectively. With P_{LP} and D_{IP} , a *competitive ratio* in (8) is derived. From the dual and primal problem formulation, it can be shown that $D_{IP} \leq D_{LP} \leq D_{LP,OPT} \leq P_{LP,OPT} \leq P_{LP}$. The first inequality is due to the fact that linear relaxation allows problem (D) in the form of linear programming (LP) to have a higher value. The second inequality indicates that the offline optimal solution always achieves a value higher than or equal to the online solution of problem (D). The third inequality captures the slackness of the primal and dual problems. In the fourth inequality, the offline optimal solution of problem (P) is smaller than or equal to any online solution. Also, we have $D_{IP} \leq D_{IP,OPT} \leq D_{LP,OPT}$. The first inequality follows from the optimality gap between the online and offline solutions when y_{ij} is an integer. The second inequality shows that linear relaxation of y_{ij} allows to have a higher value in problem (D). Thus, the ratio in (8) becomes:

$$\frac{D_{IP,OPT}}{D_{IP}} \leq \frac{P_{LP}}{D_{IP}}, \quad (13)$$

where P_{LP}/D_{IP} corresponds to α in (8). Therefore, P_{LP}/D_{IP} becomes a competitive ratio. To check whether P_{LP}/D_{IP} can be a valid competitive ratio, we check the feasibility of P_{LP} and D_{IP} as follows.

Feasibility of the solutions: We first check whether P_{LP} and D_{IP} from Algorithm 1 are feasible solutions. First, we update $y_{ij} = 1$ if (2) and (3) are satisfied. Also, since there is only one j_i^* corresponding to i , then constraint (5) is satisfied. Otherwise, if (2) and (3) are not satisfied, we set $y_{ij} = 0$. If $y_{i'j} = 0$ for task i' , the algorithm sets $y_{ij} = 0, \forall i \geq i'$, and, therefore, constraints (6) and (7) are satisfied. In problem (P), if $y_{ij} = 1, \forall i \leq I'$, x_I is given by $1/\min_{i \leq I'}(d_i/r_{j_i^*})$ when $i = I'$. In (10), the coefficient of x_I is d_i/r_j , and we observe $(d_i/r_j)/\min_{i \leq I'}(d_i/r_{j_i^*}) \geq 1$. In (11), since the coefficient of x_I is $d_I(1/r_j + 1/f_j)$, (11) becomes $(d_I(1/r_j + 1/f_j))/\min_{i \leq I'}(d_i/r_{j_i^*}) \geq 1$. Then, if $y_{ij} = 0, \forall i > I', z_j = \max_{i > I'}(1 - x_I d_i/r_j, 0), \forall j \in \mathcal{J}$.

Objective function value in (D) and (P): We derive the value of the dual and primal problems when Algorithm 1 is used. In problem (D), the objective function increases by one, and, if $y_{ij} = 1, \forall i \leq I'$, then $D_{IP} = I'$. In problem (P), we can use the final values of x_I and $z_j, \forall j$, to calculate P_{LP} . Also, note that there is a trivial value, $P_{LP} = I$. For example, this value can be achieved in the primal problem with a solution given by $u_1 = I$ and $u_i = I - (i - 1), \forall i \in \mathcal{I}$. Therefore, since we

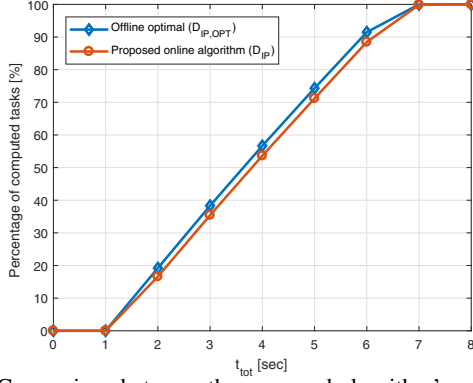


Fig. 2: Comparison between the proposed algorithm's result and the offline optimal solution in terms of percentage of computed tasks for different t_{tot} .

want to find the smallest possible value of P_{LP} , we have:

$$P_{LP} = \min \left(\frac{t_{\text{tot}}}{\min_{i \leq I'} \frac{d_i}{r_{j^*}}} + \sum_{j=1}^J \max \left(1 - \frac{\min_{i > I'} \frac{d_i}{r_j}}{\min_{i \leq I'} \frac{d_i}{r_{j^*}}}, 0 \right), I \right). \quad (14)$$

Therefore, according to (13), a competitive ratio is given by the ratio between P_{LP} in (14) and $D_{IP} = I'$. The derived ratio can be close to one as the proposed online algorithm finds a solution similar to the offline optimal solution that can be achieved by having complete information on all tasks in advance. Given the metrics in (8) and (13), next, we evaluate the performance of the proposed algorithm through simulations.

IV. SIMULATION RESULTS

For our simulations, we use a MATLAB simulator in which we consider that the source UAV initially forms a network with $J = 10$ neighboring fog UAVs uniformly distributed within a circular area of radius between 10 m and 70 m. The task size follows a uniform distribution between 50 and 70 Mbits, and the number of tasks is $I = 10$. The total bandwidth is 5 MHz. The power spectral density of the noise is -174 dBm/Hz and $P_t = 20$ dBm. The channel gain is calculated by using a free space model due to air-to-air communications [13]. The computational speed of each neighboring UAV is randomly determined from a uniform distribution between 5×10^7 and 8×10^7 bits/sec. The offline optimal solution is calculated by using a mixed-integer linear programming (MILP) solver with the assumption that the size d_i of task i , $\forall i \in \mathcal{I}$, is completely known. All statistical results are averaged over a large number of independent simulation runs.

Fig. 2 shows the percentage of computed tasks for different values of t_{tot} from 0 to 8 sec. For comparison, we calculate the offline optimal solution of the dual integer problem, i.e., $D_{IP,OPT}$, by assuming that all task sizes, d_i , $\forall i$, are known in advance. The offline optimal $D_{IP,OPT}$ shows that the percentage of computed tasks increases with t_{tot} that is a given parameter in problem (D). The design goal of our online algorithm is to achieve a performance that is similar to the offline optimal when task size d_i is revealed one by one. To this end, in Fig. 2, we can observe that the optimal solution and the solution

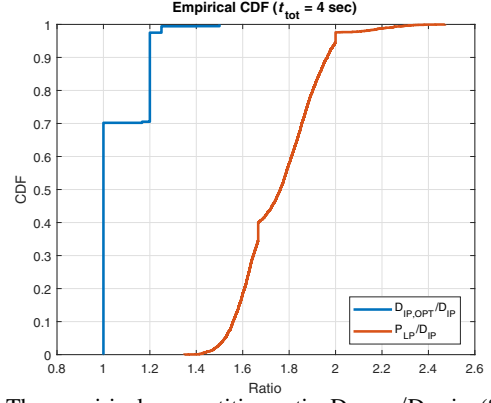


Fig. 3: The empirical competitive ratio $D_{IP,OPT}/D_{IP}$ in (8) and the analytical competitive ratio P_{LP}/D_{IP} in (13).

found by Algorithm 1 are very close for all values of t_{tot} . This demonstrates the effectiveness of the proposed algorithm that can select properly neighboring UAVs to offload tasks while maximizing the number of computed tasks. For instance, Fig. 2 shows that the maximum gap between the offline optimality and the online solution is only 7.5% when $t_{\text{tot}} = 3$.

Fig. 3 shows the empirical ratio between the offline optimal and online solutions, $D_{IP,OPT}/D_{IP}$, and the analytical competitive ratio, P_{LP}/D_{IP} in (13) for $t_{\text{tot}} = 4$. In Fig. 3, the cases in which the ratio is one correspond to scenarios in which the proposed algorithm finds the optimal solution. For example, Fig. 3 shows that the proposed algorithm achieves the optimal solution in about 70% of 3000 iterations. Also, by the definition of competitive ratio α in (8), the number of computed tasks with the proposed algorithm is at least $D_{IP,OPT}/\alpha$. For instance, in Fig. 3, the largest empirical competitive ratio is shown to be 1.5 which implies that the number of computed task is at least $D_{IP,OPT}/1.5$ when the proposed algorithm is executed with the given simulation parameters. Moreover, Fig. 3 shows that the analytical competitive ratio P_{LP}/D_{IP} provides an upper bound of the empirical competitive ratio $D_{IP,OPT}/D_{IP}$, and the resulting value of P_{LP}/D_{IP} is at most 2.5. In Fig. 3, the ratio $D_{IP,OPT}/D_{IP}$ is shown to have a step-like shape since both $D_{IP,OPT}$ and D_{IP} are integer, and there exists a limited number of possible values for $D_{IP,OPT}/D_{IP}$ for specific settings of the simulations.

Fig. 4 shows the percentage of computed tasks for two different ranges of computational speeds of the UAVs and different task sizes when the bandwidth is changed from 3 to 7 MHz with $t_{\text{tot}} = 7$. In Fig. 4, neighboring UAVs with low computational speeds are represented by $f_j \in [5 \times 10^7, 8 \times 10^7]$, whereas UAVs with high computational speeds are assumed to have $f_j \in [5 \times 10^8, 8 \times 10^8]$. Also, we consider two scenarios with small-size tasks $d_i \in [50 \times 10^6, 70 \times 10^6]$ and large-size tasks $d_i \in [70 \times 10^6, 90 \times 10^6]$, respectively. From Fig. 4, we can see that the number of computed tasks can increase if the bandwidth increases. This is due to the fact that a higher bandwidth can increase the data rate and reduces tasks' transmission latency. Therefore, more tasks can be allocated to neighboring UAVs. For instance, the number of computed tasks

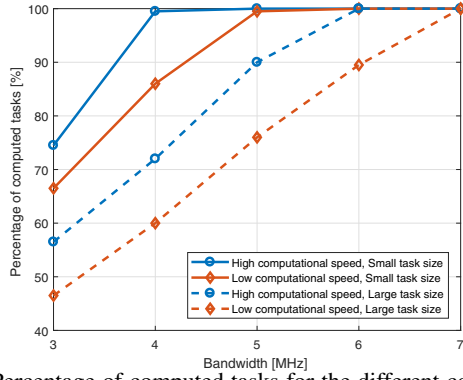


Fig. 4: Percentage of computed tasks for the different computational speeds of neighboring UAVs and different task sizes when bandwidth is varying between 3 and 7 MHz.

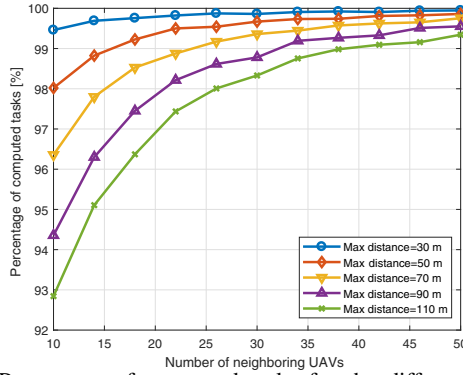


Fig. 5: Percentage of computed tasks for the different number of neighboring UAVs for different maximum air-to-air communication distances.

can increase about two-fold if the bandwidth changes from 3 MHz to 7 MHz in the case of UAVs with low computational speeds and large-size tasks. Also, Fig. 4 shows that using UAVs with high computational speeds is effective to increase the number of computed tasks. For example, the percentage of computed tasks increases from 86% to 99.5% by using UAVs having high computational speeds when bandwidth is 4 MHz and the task sizes are small. Moreover, in Fig. 4, we can observe that more tasks can be computed as task sizes become smaller; for example, small-sized tasks result in 43% more computed tasks compared to that of large-sized tasks in the case of 3 MHz.

In Fig. 5, the percentage of computed tasks is shown for different numbers of neighboring UAVs with $I = 10$. Clearly, the number of computed tasks increases with the number of neighboring UAVs. As the set of neighboring UAVs becomes larger, the source UAV has a higher probability to allocate its tasks to the neighboring UAVs having a high data rate and computational speed. Fig. 5 also shows that the number of computed tasks increases if the maximum communication distance between UAVs is reduced.

V. CONCLUSION AND FUTURE WORKS

In this paper, we have proposed a novel framework to maximize the number of successful computations for UAV-assisted fog networks within an Industry 4.0 smart factory. This

framework allows a source UAV to offload tasks from ground sensors and distributed the tasks to neighboring fog UAVs in order to compute the tasks before the source UAV arrives at the destination in a factory. When the exact information on the offloaded tasks is unknown to the source UAV, it is challenging to optimize the decision of which neighboring UAV has to compute each task. Therefore, we have formulated online optimization problem that jointly optimizes the communication and computation latency is formulated and introduced an online greedy algorithm to solve the problem. Then, by using the structure of the primal-dual problem formulation, we have derived a feasible competitive ratio as a function of the task sizes and the data rates of the UAVs. Simulation results have shown that the proposed online algorithm can allocate tasks to neighboring UAVs that is no less than 7.5% below the optimal number of computed tasks found in the offline scenario. For future work, the trajectory of the UAV can be optimized jointly with the task allocation in the proposed framework. Also, the proposed framework can be extended to a scenario in which multiple fog UAVs are deployed to offload and compute the tasks of sensors. Moreover, the proposed approach can be extended to take into account medium access and transmission protocols when the tasks are offloaded from sensors to the UAV.

REFERENCES

- [1] S. Wang, J. Wan, D. Li, and C. Zhang, "Implementing smart factory of industrie 4.0: An outlook," *International Journal of Distributed Sensor Networks*, vol. 12, no. 1, p. 3159805, Jan. 2016.
- [2] D. Zuehlke, "Smartfactorytowards a factory-of-things," *Annual Reviews in Control*, vol. 34, no. 1, pp. 129–138, Mar. 2010.
- [3] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *arXiv preprint arXiv:1803.00680*, 2018.
- [4] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 3949–3963, Jun. 2016.
- [5] —, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov 2017.
- [6] S. Rohde and C. Wietfeld, "Interference aware positioning of aerial relays for cell overload and outage compensation," in *Proc. IEEE Veh. Technol. Conf.*, Quebec City, Canada, Sept 2012, pp. 1–5.
- [7] Y. Wang, W. Peng, Q. Dou, and Z.-h. Gong, "Energy-constrained ferry route design for sparse wireless sensor networks," *Journal of Central South University*, vol. 20, no. 11, pp. 3142–3149, Nov 2013.
- [8] I. Jawhar, N. Mohamed, J. Al-Jaroodi, and S. Zhang, "A framework for using unmanned aerial vehicles for data collection in linear wireless sensor networks," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1, pp. 437–453, Apr 2014.
- [9] S. Jeong, O. Simeone, and J. Kang, "Mobile cloud computing with a UAV-mounted cloudlet: optimal bit allocation for communication and computation," *IET Commun.*, vol. 11, no. 7, pp. 969–974, May 2017.
- [10] F. Zhou, Y. Wu, H. Sun, and Z. Chu, "UAV-Enabled Mobile Edge Computing: Offloading Optimization and Trajectory Design," *ArXiv e-prints*, Feb. 2018.
- [11] L. Wang and M. Hua, "Optimal bit allocation for UAV-enabled mobile communication," in *Prof. IEEE Int. Conf. on Computer and Commun.*, Chengdu, China, Dec. 2017, pp. 474–478.
- [12] N. Buchbinder and J. S. Naor, "The design of competitive online algorithms via a primaldual approach," *Foundations and Trends in Theoretical Computer Science*, vol. 3, no. 23, pp. 93–263, May 2009.
- [13] W. Khawaja, I. Guvenc, D. Matolak, U.-C. Fiebig, and N. Schneckenberger, "A Survey of Air-to-Ground Propagation Channel Modeling for Unmanned Aerial Vehicles," *ArXiv e-prints*, Jan. 2018.