Optimizing Sensor Data Compression in Digital Twin Synchronization via a Stackelberg Game

Markus Krantzik, Anja Klein, and Lin Xiang
Communications Engineering Lab, Technische Universität Darmstadt, Germany
Emails: {m.krantzik, a.klein, l.xiang}@nt.tu-darmstadt.de

***Abstract*—Digital twin (DT) synchronization over resource-limited wireless networks has presented a significant challenge in ensuring the timely transmission and processing of high-volumes sensor data from physical systems (PSs). In this paper, we tackle this challenge by investigating data compression in DT synchronization, where sensors in the PSs compress their sensed data before transmission to a base station (BS) for processing and DT updating. Given the distributed nature of sensors, each sensor independently selects its compression ratio to minimize its *own* cost function. To align these *self-interested* sensors for efficient DT updating, we formulate a Stackelberg game to optimize data compression at the sensors and communication/computing resource allocation at the BS. By deriving the best responses (BRs) of the sensors, we further develop a low-complexity iterative algorithm to compute the Stackelberg equilibrium. Simulation results show that employing data compression can significantly reduce the DT synchronization time. Moreover, the proposed algorithm achieves near-optimal performance, closely matching the centralized joint optimization scheme with almost no price of anarchy.**

I. Introduction

Digital Twins (DTs) have emerged as a novel paradigm to revolutionize industrial production and enhance quality of life in the future society. By building digital, real-time, and precise replicas of physical systems (PSs), DTs can provide extended-reality services and enable seamless interactions with both the PSs and their digital counterparts [1]. For a successful deployment of DTs, it is crucial to facilitate timely, reliable, and accurate updating or synchronization of digital models for their PSs in an entangled manner. However, satisfying the stringent latency, reliability, and data rate requirements of DT synchronization presents a major challenge for the design of upcoming sixth-generation (6G) communication systems.

Thus far, existing studies [2]–[4] have explored the optimization of communication and/or computing to enhance DT synchronization. In [2], a hierarchical decision-making framework was developed to strategically select virtual service providers at the Internet of Things (IoT) devices while optimizing the DT synchronization intensity. In [3], a game-theoretic network selection approach was proposed to find the best relay node that minimizes the DT synchronization time in heterogeneous vehicular networks. The authors in [4] proposed a buffer-aided mobile relaying scheme and integrated it with stream computing to simultaneously overcome the communication and computing bottlenecks in DT synchronization, where the trajectory of the mobile relay was further optimized to reduce synchronization latency. However, the approaches in [2]–[4] solely consider communication and computing but ignore the impact of data compression at sensors on DT synchronization. As a result, they suffer from scalability issues when handling large volumes of sensed data.

Data compression is a well-established technique for mitigating transmission overhead and accelerating data delivery in resource-limited wireless communication systems [5]–[7]. For example, sensing and compression were jointly optimized in [5] to minimize energy consumption in wirelessly powered crowd sensing and maximize the utility of sensed data. In [6], communication and compression were jointly optimized to reduce energy consumption and latency in multi-user mobile edge computing. Moreover, the authors in [7] investigated data compression for offloading video detection tasks while balancing energy consumption and detection accuracy.

Recently, data compression was explored in DT-assisted electric vehicle charging [8], where communication and computing resources were jointly optimized to minimize the age of information of DTs. However, compression was only applied to the energy data, rather than the data used for DT synchronization. To our knowledge, no prior work has investigated the integration of data compression as well as the joint optimization of compression, communication, and computing, for DT synchronization. Note that directly applying the synchronization approaches in [2]–[4] to compressed sensor data can be strictly suboptimal, as they ignore the fundamental trade-off between data reduction and compression overhead.

To fill in this gap, this paper introduces for the first time *DT synchronization with data compression*, where sensors distributed in a PS can compress their data before transmission to a base station (BS) for processing and updating the DT. Unlike prior works [5]–[8], each sensor can independently select its compression ratio to minimize its *own* cost. To align these *self-interested* sensors toward efficient DT updating, we propose a Stackelberg game-based resource allocation framework to optimize communication/computing and data compression in DT synchronization. Our contributions are as follows:

- We consider DT synchronization with data compression and formulate a Stackelberg game to optimize data compression at the self-interested sensors and communication/computation resource allocation at the BS for minimizing the time required for DT synchronization.
- We derive the sensors' best response (BR) strategies in closed form and propose a low-complexity iterative algorithm to efficiently calculate the equilibrium strategy.
- Simulation results show that data compression can significantly lower the DT synchronization time and almost no price of anarchy is paid in the equilibrium solution.

In the remainder of this paper, Section II introduces the system model. Sections III and IV present the game formulation and the proposed solution with strategy analyses, respectively. Section V evaluates the proposed algorithm and finally, Section VI concludes this paper.

II. System Model

This section presents the system model for DT synchronization, including data compression and communication at sensors and data processing at a BS.

A. DT Synchronization with Sensor Data Compression

As illustrated in Fig. 1, a cloud server is connected via a high-speed backhaul to a BS located at $\mathbf{u}^{\mathrm{BS}}$. The cloud server is responsible for hosting, including constructing and synchronizing/updating, a real-time DT of a dynamically changing PS. The PS consist of K sensors that are distributed at locations $\mathbf{u}_k^{\mathrm{s}}$, $k \in \mathcal{K} \triangleq \{1, \ldots, K\}$. Due to the inherent inertia of the

This work has been supported by the LOEWE initiative (Hesse, Germany) within the emergenCITY center under grant LOEWE/1/12/519/03/05.001(0016)/72, by the BMBF project Open6GHub under grant 16KISKO14 and by DAAD with funds from the German Federal Ministry of Education and Research (BMBF).

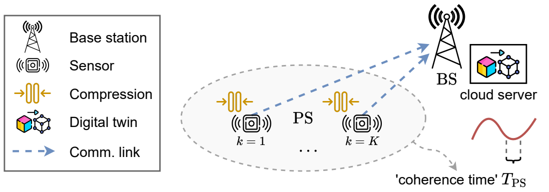

Fig. 1. Illustration of DT synchronization with data compression.

PS, the dynamics of the PS are assumed to be represented by a low-pass system with bandwidth B_{PS}. According to the Shannon's sampling theorem [9], the DT needs to be updated at the cloud server every $T_{\text{PS}} = 1/B_{\text{PS}}$ seconds to produce a timely representation of the PS. For a single update, each sensor needs to transmit D_k bits of data to the BS. Due to limited communication resources, each sensor first compresses the sensed data before transmitting it in reduced amounts to the BS. Upon receiving the compressed data from one sensor, the cloud server decompresses it to retrieve and further process the sensor data for DT updating. Thus, *DT synchronization with data compression* spans the whole process of data compression, transmission, decompression, and processing.

1) Data Compression and Decompression: To facilitate an analytical modeling of data compression at the sensors, the compression model from [5]–[7] is adopted in this work. Lossless compression is utilized such that the sensor data can be recovered by the cloud server without loss of critical information about the PS to guarantee accurate DT synchronization. The compression ratio is denoted by $\beta_k \in [1, \beta^{\max}]$. Here, $\beta_k = 1$ if the sensed data is sent without compression and $\beta^{\max}$ is the maximum compression ratio achievable with lossless compression. While using a large β_k can reduce the amount of sensed data that is sent to the BS and save communication resources, it increases the overhead for sensor data compression. To capture this fundamental trade-off, the computational complexity $\eta(\beta_k, \epsilon)$ for data compression is defined as [5]

$$\eta(\beta_k, \epsilon) = e^{\beta_k \cdot \epsilon} - e^{\epsilon}, \tag{1}$$

where ϵ is a constant specified by the applied compression algorithm. Based on (1), we have $\eta(\beta_k, \epsilon) = 0$ for $\beta_k = 1$, and the compression complexity $\eta(\beta_k, \epsilon)$ increases exponentially with the compression ratio when $\beta_k > 1$. Let f_k^{S} be the compression speed of sensor k. The time required for data compression, defined by $T_k^{\text{comp}}(\beta_k)$, is given as

$$T_k^{\text{comp}}(\beta_k) = \frac{D_k \cdot \eta(\beta_k, \epsilon)}{f_k^{\text{S}}}. \tag{2}$$

Since the cloud server usually possesses much greater processing power than the sensors, the time required for decompression is negligible and not considered in the existing research [7]. Following the approach in [7], the delay incurred in data decompression is also ignored in this work.

2) Channel Model for Communication: The sensors transmit the compressed sensing data to the BS using orthogonal frequency-division multiple access (OFDMA). Thereby, the available bandwidth B is divided into nonoverlapping subchannels and the sensors occupy orthogonal subchannels during data transmission. Let b_k be the bandwidth allocated for sensor $k \in \mathcal{K}$. The achievable data rate r_k of sensor k is defined using the Shannon's capacity formula [10] as

$$r_k(b_k) = b_k \cdot \log_2\left(1 + \frac{|H_k|^2 \cdot p_k}{N_0 \cdot b_k}\right), \tag{3}$$

where p_k and N_0 denote the transmit power and the noise power spectral density, respectively. $H_k \in \mathbb{C}$ is the channel gain of sensor $k \in \mathcal{K}$.

In this paper, we consider Rician fading subchannels between the sensors and the BS. Let $d_k = \left\|\mathbf{u}^{\text{BS}} - \mathbf{u}_k^{\text{s}}\right\|$ be the distance from sensor $k \in \mathcal{K}$ to the BS. The channel gain H_k is modeled as

$$H_k = \sqrt{A_0} \cdot (d_k)^{-\alpha/2} \cdot h_k, \tag{4}$$

where $A_0/(d_k)^{\alpha}$ defines the path loss of the channel. A_0 and $\alpha \geq 2$ denote the path loss at reference distance of 1 m and the path loss exponent, respectively. Additionally, $h_k \in \mathbb{C}$ is the channel fading gain, which is further modeled by a superposition of a line-of-sight (LoS) and a non-LoS (NLoS) component as follows [10]

$$h_k = \underbrace{\sqrt{\frac{\kappa}{\kappa+1}}\sigma_k e^{j\theta_k}}_{\text{LoS}} + \underbrace{\sqrt{\frac{1}{\kappa+1}}\mathcal{CN}\left(0, \sigma_k^2\right)}_{\text{NLoS}}. \tag{5}$$

In (5), θ_k is the phase of the LoS component, which is uniformly distributed within $[0, 2\pi]$, and σ_k is the standard deviation of the channel fading gain. Moreover, κ is the energy ratio between the LoS and NLoS component. The LoS component dominates in h_k if κ becomes large.

Based on (3), the time required for transmitting the compressed sensor data to the BS is given by

$$T_k^{\text{tr}}(\beta_k, b_k) = \frac{D_k}{\beta_k \cdot r_k(b_k)}, \tag{6}$$

where the volume of sensor data after compression is D_k/β_k.

3) DT Computing: The cloud server has a processing capacity of C_{DT} Hz, which is allocated to the K sensors for DT updating. Let f_k^{DT} be the DT processing speed allocated to sensor $k \in \mathcal{K}$. The time required for processing the received sensor data and synchronizing the DT can be modeled as

$$T_k^{\text{DT}}\left(f_k^{\text{DT}}\right) = \frac{D_k \cdot c_k}{f_k^{\text{DT}}}, \tag{7}$$

where c_k is the computational complexity of DT processing.

B. DT Synchronization Time

The total time required to compress, transmit, and process the data of each sensor $k \in \mathcal{K}$ can be calculated as

$$T_k^{\text{total}}\left(\beta_k, b_k, f_k^{\text{DT}}\right) = T_k^{\text{comp}} + T_k^{\text{tr}} + T_k^{\text{DT}}. \tag{8}$$

Here, we have assumed that no sensor has to wait for other sensors before starting its data compression or transmission. The time required for completing a round of DT synchronization with data compression, or the DT synchronization time for short, is defined by $T_k^{\text{total}}\left(\beta_k, b_k, f_k^{\text{DT}}\right)$ of the slowest sensor.

III. Problem Formulation

Due to the underlying trade-off between compression and communication, which is further coupled with DT computing, intelligent optimization of the compression ratio β_k, bandwidth allocation b_k, and DT processing speed f_k^{DT} for minimizing the DT synchronization time is crucial in order to facilitate timely DT synchronization at the cloud server. However, the communication/computing resources and the compression ratio are controlled by the BS and the sensors, respectively, which aim to achieve distinct goals via individual strategic decision-making for their controllable actions based on information observed by each. To capture the interaction between the BS and the sensors, in this section, we formulate the resource allocation problem for DT synchronization based on a Stackelberg game.

A. Stackelberg Game Formulation

We consider a hierarchical decision process between the BS and the sensors as follows. The BS acts as the leader in deciding the bandwidth allocation b_k and DT processing speed f_k^{DT} for the sensors. After the BS announces its decisions, each sensor acts as a follower to choose its own compression ratio β_k. Since each sensor only perceives its own compression time T_k^{comp} and transmission time T_k^{tr}, it is interested in minimizing the cost function below

$$U_k^{\mathrm{f}}\left(\beta_k; b_k\right) = T_k^{\mathrm{comp}}\left(\beta_k\right) + T_k^{\mathrm{tr}}\left(\beta_k, b_k\right), \tag{9}$$

for given bandwidth b_k. The sub-game of each follower is described by the following optimization problem,

$$\left(P_k^{\mathrm{f}}\right): \quad \min_{\beta_k} U_k^{\mathrm{f}}\left(\beta_k; b_k\right) \tag{10}$$

$$\text{s.t. C1.f: } \; 1 \leq \beta_k \leq \beta^{\max}, \tag{10a}$$

where constraint C1.f guarantees a valid compression ratio β_k for the sensor. In contrast, with knowledge about all sensors $k \in \mathcal{K}$, the BS aims to align the self-interested sensors for minimizing the DT synchronization time. The cost function of the BS is thus defined as

$$U^{\mathrm{l}}\left(\mathbf{b}, \mathbf{f}; \boldsymbol{\beta}\right) = \max_{k \in \mathcal{K}} T_k^{\mathrm{total}}\left(\beta_k, b_k, f_k^{\mathrm{DT}}\right), \tag{11}$$

for $\mathbf{b} \triangleq \{b_k | k \in \mathcal{K}\}$, $\mathbf{f} \triangleq \left\{f_k^{\mathrm{DT}} | k \in \mathcal{K}\right\}$, and $\boldsymbol{\beta} \triangleq \{\beta_k | k \in \mathcal{K}\}$. The sub-game of the BS is formulated as allocating bandwidth b_k and DT processing speed f_k^{DT} for given compression ratios β_k via the following optimization problem,

$$\left(P_{\mathrm{DT}}^{\mathrm{l}}\right): \quad \min_{\mathbf{b}, \mathbf{f}} U^{\mathrm{l}}\left(\mathbf{b}, \mathbf{f}; \boldsymbol{\beta}\right) \tag{12}$$

$$\text{s.t. C1.l: } \; \sum\nolimits_{k=1}^{K} b_k \leq B \tag{12a}$$

$$\text{C2.l: } \; \sum\nolimits_{k=1}^{K} f_k^{\mathrm{DT}} \leq C_{\mathrm{DT}}, \tag{12b}$$

where constraints C1.l and C2.l limit the total allocated bandwidth and DT processing speed by B and C_{DT}, respectively.

B. Stackelberg Equilibrium

By strategically interacting through their sub-games, the leader and the followers aim to reach an equilibrium solution in the Stackelberg game. For a given b_k the BR of the sensors is defined as $\mathrm{BR}\left(b_k\right) \triangleq \left\{\beta_k\colon \min_{1 \leq \beta_k \leq \beta^{\max}} U_k^{\mathrm{f}}\left(\beta_k; b_k\right)\right\}$, which determines the optimal β_k to minimize the utility cost $U_k^{\mathrm{f}}\left(\beta_k; b_k\right)$ of sensor k for any given b_k. According to [11], the solution $\mathbf{b}^*, \mathbf{f}^*$ and $\boldsymbol{\beta}^*$ is referred to as the Stackelberg Equilibrium (SE) if $\beta_k^* \in \mathrm{BR}\left(b_k\right)$ and, for any $\beta_k \in \mathrm{BR}\left(b_k\right)$,

$$U^{\mathrm{l}}\left(\mathbf{b}^*, \mathbf{f}^*; \boldsymbol{\beta}^*\right) \leq U^{\mathrm{l}}\left(\mathbf{b}, \mathbf{f}; \boldsymbol{\beta}\right) \tag{13}$$

holds. Here, $(\cdot)^*$ denotes the best strategy for the sensors and the BS. At the SE, neither the BS (leader) or the sensors (followers) have an incentive to change their strategy [12].

IV. Problem Solution

In this section, we find the SE for the defined Stackelberg game using the *backward-induction* method [12]. To this end, we first derive the BR of the followers. By substituting the BRs of the followers into the leader's optimization problem, we further propose an iterative algorithm for finding the SE.

A. Strategy Analysis for the Followers

In the proposed multiple-follower game, neither the objective functions nor decisions of the followers are coupled with each other. Hence, given any b_k, the BR of each follower reduces to the optimal solution of problem (10). Below, we will derive the BRs in closed form by revealing the strong duality in problem (10) and solving its dual problem.

First, to simplify the derivation, we replace the term $1/\beta_k$ in the utility cost of the sensor (9) by an auxiliary variable $\mu_k > 0$. To ensure a valid solution, we require $\frac{1}{\mu_k} \leq \beta_k$, as $T_k^{\mathrm{tr}}\left(\beta_k, b_k\right)$ decreases monotonically in β_k. By defining vector $\mathbf{x}_k \triangleq \left[\beta_k, \mu_k\right]^{\intercal}$ and function

$$f_k\left(\mathbf{x}_k; b_k\right) \triangleq \frac{D_k}{f_k^{\mathrm{S}}} \cdot e^{\beta_k \cdot \epsilon} + \frac{D_k}{r_k\left(b_k\right)} \cdot \mu_k, \tag{14}$$

we can rewrite the follower's problem (10) as

$$\left(P_k^{\mathrm{f},2}\right): \quad \min_{\{\beta_k \leq \beta^{\max}, \mu_k > 0\}} f_k\left(\mathbf{x}_k; b_k\right) \tag{15}$$

$$\text{s.t. C1.f: } \; 1 \leq \beta_k \tag{15a}$$

$$\text{C2.f: } \; 1/\mu_k \leq \beta_k. \tag{15b}$$

Problem $P_k^{\mathrm{f},2}$ is a convex optimization problem and fulfills the Slater's condition [13], for which strong duality holds. Let $\boldsymbol{\lambda}_k = \left[\lambda_{1,k}, \lambda_{2,k}\right]^{\intercal} \geq \mathbf{0}$ be the vector of dual variables associated with constraints C1.f and C2.f. The Lagrangian function is defined as

$$L_k\left(\boldsymbol{\lambda}_k, \mathbf{x}_k; b_k\right) = f_k\left(\mathbf{x}_k; b_k\right) + \boldsymbol{\lambda}_k^{\intercal} \mathbf{g}\left(\mathbf{x}_k\right), \text{ with} \tag{16a}$$

$$\mathbf{g}\left(\mathbf{x}_k\right) = \begin{pmatrix} 1 - \beta_k \\ \frac{1}{\mu_k} - \beta_k \end{pmatrix}. \tag{16b}$$

Exploiting its strong duality, problem $P_k^{\mathrm{f},2}$ can be solved via its dual problem defined as

$$\left(P_k^{\mathrm{f},3}\right): \quad \max_{\boldsymbol{\lambda}_k \geq 0, k \in \mathcal{K}} D\left(\boldsymbol{\lambda}_k; b_k\right), \tag{17a}$$

where $D\left(\boldsymbol{\lambda}_k; b_k\right) \triangleq \min_{\mathbf{x}_k} L_k\left(\boldsymbol{\lambda}_k, \mathbf{x}_k; b_k\right)$ is the dual function which must satisfy $\beta_k \leq \beta^{\max}$ and $\mu_k > 0$ as defined in (15). The following lemma defines the optimal β_k and μ_k that minimize $L_k\left(\boldsymbol{\lambda}_k, \mathbf{x}_k; b_k\right)$ for a given $\boldsymbol{\lambda}_k$.

Lemma 1. *For given dual variables $\boldsymbol{\lambda}_k \geq \mathbf{0}$, the optimal compression ratio β_k^* and the optimal auxiliary variable μ_k^* are uniquely determined as*

$$\beta_k^* = \min\left\{\frac{1}{\epsilon} \ln\left(\frac{f_k^{\mathrm{S}}}{\epsilon \cdot D_k}\left(\lambda_{1,k} + \lambda_{2,k}\right)\right), \beta^{\max}\right\}, \tag{18}$$

$$\mu_k^* = \sqrt{\lambda_{2,k} \cdot r_k\left(b_k\right) / D_k}. \tag{19}$$

Proof. Set the first-order derivative of $L_k\left(\boldsymbol{\lambda}_k, \mathbf{x}_k; b_k\right)$ with respect to β_k to zero, i.e.,

$$\frac{\partial}{\partial \beta_k} L_k\left(\boldsymbol{\lambda}_k, \mathbf{x}_k; b_k\right) = \frac{\epsilon \cdot D_k}{f_k^{\mathrm{S}}} \cdot e^{\beta_k \cdot \epsilon} - \lambda_{1,k} - \lambda_{2,k} = 0, \tag{20}$$

and solve β_k. (18) is then obtained by projecting the solution of β_k onto the interval $\beta_k \leq \beta^{\max}$. Similarly, μ_k^* in (19) is obtained by setting the first-order derivative of $L_k\left(\boldsymbol{\lambda}_k, \mathbf{x}_k; b_k\right)$ with respect to μ_k as zero,

$$\frac{\partial}{\partial \mu_k} L_k\left(\boldsymbol{\lambda}_k, \mathbf{x}_k; b_k\right) = \frac{D_k}{r_k\left(b_k\right)} - \lambda_{2,k} \cdot \frac{1}{\left(\mu_k\right)^2} = 0, \tag{21}$$

and solve μ_k. Note that $\mu_k^* \geq 0$ already holds. □

With the optimal solution $\mathbf{x}_k^*(\boldsymbol{\lambda}_k) = [\beta_k^*, \mu_k^*]^\intercal$ obtained in Lemma 1, the sub-gradient of $D(\boldsymbol{\lambda}_k)$ is further given by

$$\mathbf{g}\left(\mathbf{x}_k^*(\boldsymbol{\lambda}_k)\right) = \begin{pmatrix} 1 - \beta_k^* \\ 1/\mu_k^* - \beta_k^* \end{pmatrix}. \tag{22}$$

This enables to solve the dual problem $P_k^{\mathrm{f},3}$ using the sub-gradient method [14]. Specifically, let l be an iteration index. We iteratively update the dual vector $\boldsymbol{\lambda}_k^{(l)}$ according to

$$\boldsymbol{\lambda}_k^{(l+1)} = \left[\boldsymbol{\lambda}_k^{(l)} + \mathbf{h}_\lambda^{(l)} \circ \mathbf{g}\left(\mathbf{x}_k^*\left(\boldsymbol{\lambda}_k^{(l)}\right)\right)\right]^+, \tag{23}$$

where $\boldsymbol{\lambda}_k^{(0)}$ is a given initial dual vector and $\mathbf{h}_\lambda^{(l)}$ is an adaptive step size obtained via line search to find an ascent direction of the dual function. Moreover, $\circ$ denotes the element-wise product of two vectors and $[\cdot]^+ \triangleq \max\{\mathbf{0}, \cdot\}$ pads negative elements to zeros.

B. Strategy Analysis for the Leader

To apply the backward-induction method, we substitute β_k^* and μ_k^* to replace the corresponding variables in the completion time T_k^{total} as

$$\begin{aligned} &T_k^{\mathrm{total}}\left(b_k, f_k^{\mathrm{DT}}; \boldsymbol{\lambda}_k\right) \\ &\quad = \frac{D_k}{f_k^{\mathrm{S}}} \eta\left(\beta_k^*, \epsilon\right) + \frac{D_k}{r_k(b_k)} \mu_k^* + \frac{c_k D_k}{f_k^{\mathrm{DT}}}. \end{aligned} \tag{24}$$

It is not difficult to verify that, for given dual variables $\boldsymbol{\lambda}_k$, $T_k^{\mathrm{total}}\left(b_k, f_k^{\mathrm{DT}}; \boldsymbol{\lambda}_k\right)$ is a strictly convex function. Consequently, the leader's problem can be reformulated into the convex problem given below

$$\begin{aligned} \left(P_{\mathrm{DT}}^{\mathrm{l},2}\right): \quad &\min_{\mathbf{b},\mathbf{f}} \max_{k \in \mathcal{K}} T_k^{\mathrm{total}}\left(b_k, f_k^{\mathrm{DT}}; \boldsymbol{\lambda}_k\right) \\ &\text{s.t. C1.l and C2.l,} \end{aligned} \tag{25}$$

and solved using off-the-shelf solvers such as CVX [15].

C. Proposed Iterative Algorithm

Algorithm 1 summarizes our proposed iterative procedure to compute the SE for the defined Stackelberg game. We use subscript $(\cdot)^{(l)}$ as the iteration index. In each iteration, the leader's resource allocation problem $P_{\mathrm{DT}}^{\mathrm{l},2}$ in (25) is first solved for given dual variables $\boldsymbol{\lambda}_k$. Then, for each sensor $k \in \mathcal{K}$, the dual variables $\boldsymbol{\lambda}_k$ are updated using the gradient-ascent search defined in (23). The iterations are repeated until the objective value converges or a maximum number of $L_{\max}$ iterations are reached. We now show that Algorithm 1 converges to the SE of the formulated game under mild conditions.

Proposition 2. *The Stackelberg game defined in* (10) *and* (12) *has a unique SE, which is given by* $(\mathbf{b}^*, \mathbf{f}^*)$ *and* $\boldsymbol{\beta}^*$. *Algorithm 1 converges to the SE, provided* $L_{\max} < +\infty$ *is large enough.*

Proof. Due to limited page space, we provide only a sketch of the proof. Let us define the domain $\mathcal{R} \triangleq \{(\mathbf{b}, \mathbf{f}, \boldsymbol{\beta}) \mid (\mathbf{b}, \mathbf{f})$ feasible to (12) and $\beta_k \in \mathrm{BR}(b_k), \forall k \in \mathcal{K}\}$. Note that the sensors' or followers' problems are always feasible, such that $\mathrm{BR}(b_k)$ always exists. Moreover, $\mathcal{R}$ is a compact set. As $U^{\mathrm{l}}(\mathbf{b}, \mathbf{f}; \boldsymbol{\beta})$ is a continuous function of $(\mathbf{b}, \mathbf{f}, \boldsymbol{\beta})$, existence of the SE is guaranteed.

Meanwhile, by applying suitable problem transformations, the sub-games of the sensors (followers) and the BS (leader) can be equivalently reformulated into convex optimization problems, whose objective functions are convex and strictly convex with respect to $(\mathbf{b}, \mathbf{f}, \boldsymbol{\beta})$, respectively. This implies that $\mathcal{R}$ is a convex set. Therefore, the SE solution to (13), which minimizes a strictly convex function over a convex set, is unique.

Algorithm 1: Proposed Iterative Algorithm for Finding the Stackelberg Equilibrium

```
Input: $\boldsymbol{\lambda}_k^{(0)}$, $\forall k \in \mathcal{K}$.
Output: $\boldsymbol{\beta}^*, \mathbf{b}^*, \mathbf{f}^*$.
Set $l = 1$.
repeat
    Solve the leader's problem $P_{\mathrm{DT}}^{\mathrm{l},2}$ in (25) for $\boldsymbol{\lambda}_k^{(l-1)}$ to obtain its optimal solution $\mathbf{b}^{(l)}$ and $\mathbf{f}^{(l)}$.
    foreach $k \in \mathcal{K}$ do
        Obtain $\boldsymbol{\lambda}_k^{(l)}$ based on $\mathbf{b}^{(l)}$ and $\mathbf{f}^{(l)}$ using the gradient-ascent method defined in (23).
        Update $\beta_k^{(l)}$ in $\boldsymbol{\beta}^{(l)}$ based on $\boldsymbol{\lambda}_k^{(l)}$ using the solution of the sensor in (18).
    end
    Update the maximum completion time $U^{\mathrm{l}}\left(\mathbf{b}^{(l)}, \mathbf{f}^{(l)}; \boldsymbol{\beta}^{(l)}\right)$ using (11) and calculate its difference to the last iteration $\Delta T_{\mathrm{total}}^{(l)}$.
    Set $l = l + 1$.
until $\Delta T_{\mathrm{total}}^{(l)} \leq \xi$ or $l \geq L_{\max}$;
Return $\mathbf{b}^{(l)}$, $\mathbf{f}^{(l)}$, and $\boldsymbol{\beta}^{(l)}$ from the last iteration $l$.
```

TABLE I
SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
N_0	-90 dBm	A_0	-30 dB
κ	10	α	2
σ_k	1	ϵ	3.5
c_k	10^4 cycles/bit	$L_{\max}$	30
ξ	10^{-6}		

Finally, when $L_{\max}$ becomes sufficiently large, the gradient-ascent search in (23) will converge to the BRs of the sensors, while the backward-induction procedure will converge to the SE of the defined game. □

V. Simulation Results

In this section, we evaluate the performance of the proposed algorithm via simulation. We consider $K = 4$ sensors located at the same distance $d_k = 378$ m away from the BS. The bandwidth and DT processing capacity available at the BS are set as $B = 100$ kHz and $C_{\mathrm{DT}} = 50$ MHz, respectively, which are shared among the sensors. We start with evaluating a *uniform* scenario in Section V-A, where each sensor transmits $D_k = 4$ kbit to the BS. To this end, each sensor can compress up to a ratio $\beta^{\max} = 3$ at a compression speed of $f_k^{\mathrm{S}} = 200$ kHz. The transmit power $p_k = 15$ dBm ≈ 31.6 mW and the computing complexity $c_k = 10^4$ cycles/bit are the same for all sensors, except that their channel fading gains are randomly generated according to (5). The settings of other parameters are provided in Table I. We will further extend the evaluation to *non-uniform* scenarios in Section V-B.

A. Performance Comparison in A Uniform Scenario

For performance comparison, we consider several baseline schemes which, unlike the proposed scheme, either equally distribute the bandwidth allocation b_k and/or DT processing speed f_k^{DT} among the K sensors, or abandon data compression at the sensors; that is, they optimize only the remaining variables. The detailed differences are listed below:

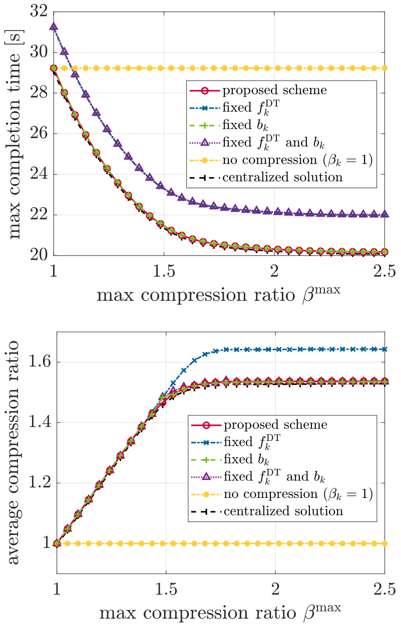

Fig. 2. (Top) Maximum completion time T_k^{total} and (below) average compression ratio $\bar{\beta}$ versus maximum compression ratio $\beta^{\max}$.

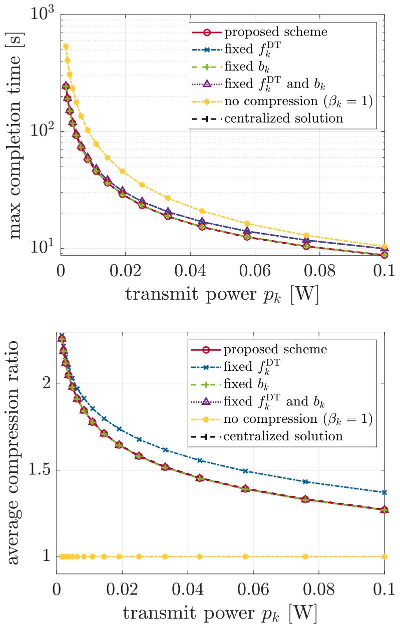

Fig. 3. (Top) Maximum completion time T_k^{total} in logarithmic scale and (below) average compression ratio $\bar{\beta}$ versus transmit power p_k.

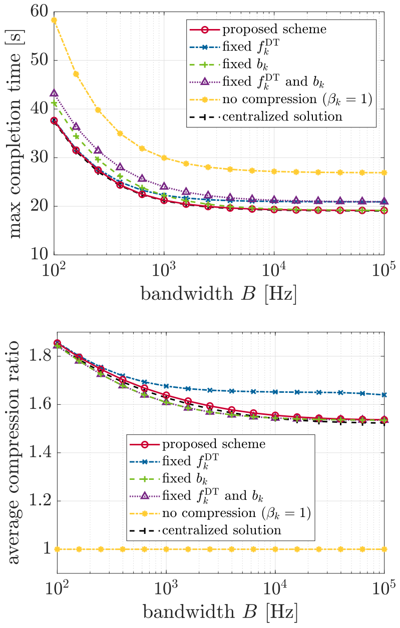

Fig. 4. (Top) Maximum completion time T_k^{total} and (below) average compression ratio $\bar{\beta}$ versus available bandwidth B in logarithmic scale.

- *Baseline scheme* 1 employs fixed DT processing speed $f_k^{\mathrm{DT}} = C_{\mathrm{DT}}/K$.
- *Baseline scheme* 2 uses fixed bandwidth $b_k = B/K$.
- *Baseline scheme* 3 fixes both the DT processing speed and bandwidth as $f_k^{\mathrm{DT}} = C_{\mathrm{DT}}/K$ and $b_k = B/K$.
- *Baseline scheme* 4 uses no data compression ($\beta_k = 1$).

Additionally, *Baseline scheme* 5 minimizes the cost function of the BS (11) by jointly optimizing β_k, b_k, and f_k^{DT} constrained by (10a), (12a), and (12b) in a centralized manner.

Figure 2 compares the maximum completion time T_k^{total} and the average optimal compression ratio $\bar{\beta} = \frac{1}{K}\sum_{k=1}^{K} \beta_k^*$ of the considered schemes versus the maximum allowable compression ratio $\beta^{\max} \in [1, 2.5]$. Without data compression, baseline scheme 4 performs the worst among all considered schemes. The other schemes employ DT synchronization with data compression, whose performance improves as $\beta^{\max}$ increases, till saturates. This is because the sensors can choose higher optimal compression ratios on average to reduce the volume of transmit data and minimize their own utility functions in (9) whereas the increase in compression time is negligible, which helps the BS lower its utility function, namely the DT synchronization time. However, the benefits of compression saturates when the average compression ratio approaches $\bar{\beta} = 1.54$. In this case, the sensors have no incentive to choose a larger $\bar{\beta}$, which would otherwise penalize the compression time. By strategically adjusting the compression ratio β_k, DT processing speed f_k^{DT}, and bandwidth b_k, the proposed scheme achieves the lowest DT synchronization time, which outperforms baseline scheme 4 by up to 30.9% and gains by 8.3% than baseline schemes 1 and 3 that employ fixed f_k^{DT}. The performance gains of the proposed scheme also enlarge with $\beta^{\max}$, till saturate. Interestingly, even though their utility functions are independent of f_k^{DT}, the sensors tend to select higher compression ratios on average under baseline scheme 1 than under other baseline schemes. This is because the sensor with the smallest channel fading gain $|h_k|$ is allocated with a large bandwidth allocation by the BS under baseline scheme 1. With less bandwidth allocation, the other sensors will increase the compression ratios to minimize their utility functions. By contrast, such behavior is not observed in baseline scheme 3, due to fixed bandwidth allocation.

Figure 3 shows the maximum completion time T_k^{total} and the compression ratio β_k versus the transmit power $p_k \in [1.6\ \mathrm{mW}, 100\ \mathrm{mW}]$. For a low transmit power of $p_k = 1.6$ mW, the proposed scheme significantly outperforms baseline scheme 4 by 54.6%, owing to utilizing a high β_k. For the schemes employing DT synchronization with data compression, both T_k^{total} and $\bar{\beta}$ decrease as p_k increases, since transmission times can be lowered via allocating more power, which reduces the need for data compression. For small p_k, the schemes employing DT synchronization with data compression achieve similar performances since the transmission time T_k^{tr} dominates the DT synchronization time, where optimizing b_k or f_k^{DT} has only negligible impact. However, when p_k becomes large, optimizing the DT processing speed f_k^{DT} becomes crucial for accelerating DT synchronization. Compared to baseline schemes 1 and 3, the proposed schemes can reduce T_k^{total} by 12.0%.

In Figures 2 and 3, baseline scheme 2 performs very close to the proposed scheme. For insights into this, Figure 4 further evaluates the maximum completion time T_k^{total} and the compression ratio $\bar{\beta}$ versus the available bandwidth $B \in [100\ \mathrm{Hz}, 100\ \mathrm{kHz}]$. As expected, increasing B results in a lower T_k^{total} and a lower $\bar{\beta}$. However, in the lower bandwidth regime, the proposed scheme significantly outperforms baseline scheme 2, due to adaptive optimization of bandwidth allocation. By contrast, in the higher bandwidth regime, the proposed scheme outperforms baseline scheme 1, due to adaptive allocation of DT processing speed. By strategically adjusting both b_k and f_k^{DT}, the proposed scheme achieves the lowest T_k^{total}, which reduces the DT synchronization time by 8.9% compared to baseline schemes 2 and 3 at $B = 100$ Hz, and by 8.9% compared to baseline schemes 1 and 3 at $B = 100$ kHz. Figure 4 also implies that adopting fixed f_k^{DT} and fixed b_k, as in baseline schemes 1 and 2, are close-to-optimal in the low and high bandwidth regime, respectively.

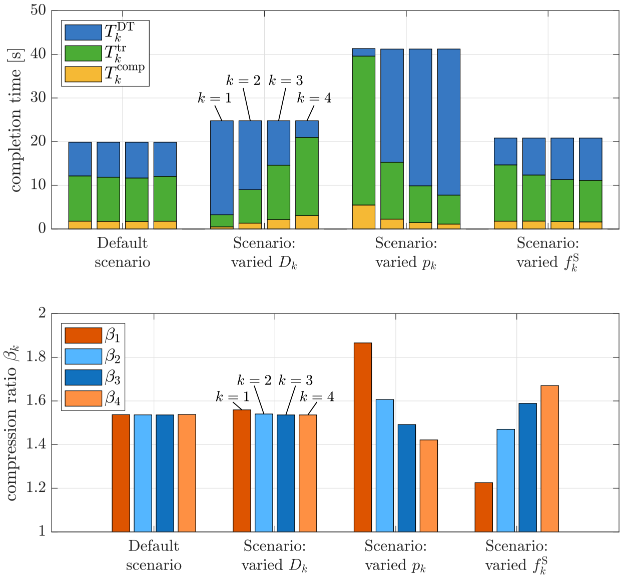

Fig. 5. (Top) Completion time $\{T_k^{\mathrm{comp}}, T_k^{\mathrm{tr}}, T_k^{\mathrm{DT}}\}$ and (below) compression ratio β_k for $K = 4$ sensors in different scenarios.

In Figures 2 to 4, the proposed scheme always achieves almost the same results as the centralized optimization solution, where the maximum completion time only deviates by 0.5%. This suggests that, for the Stackelberg game defined between the self-interested sensors and the BS, no price of anarchy is paid in the equilibrium solution, i.e., the SE is efficient.

B. Further Results in Non-Uniform Scenarios

For insights into the coupling and trade-off among communication, computing, and compression in the proposed scheme, Figure 5 compares the completion time, including $T_k^{\mathrm{comp}}, T_k^{\mathrm{tr}}$, and T_k^{DT}, and the compression ratio β_k of each sensor in the *uniform* scenario considered so far and in three *non-uniform* scenarios to be newly defined below. In the latter scenarios, one of the parameters, including the volume of sensor data, the transmit power, and the compression speed, is varied across the $K = 4$ sensors according to $D_k = \upsilon_k * 4$ kbit, $p_k = \upsilon_k * 31.6$ mW, or $f_k^{\mathrm{S}} = \upsilon_k * 200$ kHz, where υ_k is a weight assigned to sensor k. We set the weights as $\upsilon_1 = 0.25, \upsilon_2 = 0.75, \upsilon_3 = 1.25$, or $\upsilon_4 = 1.75$, to redistribute the available resources among the sensors. From Figure 5 we observe that, compared with the default uniform scenario, redistributing the volume of data D_k across the sensors results in increased maximum T_k^{total}. This is because the time required for compression and transmission at sensor $k = 4$, which has the highest D_k, significantly increases, even after allocating more DT processing speed f_k^{DT} to it. However, although the BR β_k^*, cf. (18), of each sensor is a function of D_k in the sensors sub-game (15), the compression ratios β_k of all sensors change only slightly for different volumes of data D_k. Meanwhile, redistributing the transmit power p_k across the sensors significantly increases the maximum T_k^{total}. Because the sensor with the lowest p_k defines a bottleneck during data transmission, even after applying a high β_k and allocating a large DT computation speed. Further, by varying the compression speeds across the sensors, sensors $k = 3$ and $k = 4$ with higher f_k^{S} can increase their compression ratios β_k without increasing the times needed for compression and transmission. In contrast, sensors $k = 1$ and $k = 2$ with lower compression speeds need to reduce their compression ratios to keep the compression time low, which however increases the time for transmission and DT synchronization. In Figure 5, all sensors always complete DT synchronization within the same time period for all considered scenarios. This result suggests that, by jointly optimizing the compression and resource allocation, the proposed algorithm can flexibly adjust to different scenarios to minimize the maximum completion time across the sensors and accelerate DT synchronization. Note that the compression ratios of all sensors never become large, in order to best trade-off between compression and communication among all sensors.

VI. Conclusions

In this paper, we investigated DT synchronization with sensor *data compression* to reduce redundancy in the sensor data and find the best trade-off to speed up DT updating. A Stackelberg game was formulated to model the strategic interactions between self-interested sensors (followers) and the BS (leader) in optimizing the compression ratio and the communication/computing resources, respectively, during DT synchronization. The BR of each sensor was derived by exploiting strong duality in the follower's sub-game. Based on this, a low-complexity iterative algorithm was developed to search for the SE of the introduced Stackelberg game. Simulation results in different scenarios showed that integrating data compression into DT updating holds significant promises to ultimately accelerate DT synchronization. Moreover, the proposed algorithm can efficiently coordinate the self-interested sensors with the BS towards achieving as low DT synchronization time as the centralized joint optimization scheme.

References

[1] L. U. Khan *et al.*, "Digital twin of wireless systems: Overview, taxonomy, challenges, and opportunities," *IEEE Commun. Surveys & Tutorials*, vol. 24, no. 4, pp. 2230–2254, 2022.

[2] Y. Han *et al.*, "A dynamic hierarchical framework for iot-assisted digital twin synchronization in the metaverse," *IEEE Internet of Things Journal*, vol. 10, no. 1, pp. 268–284, 2023.

[3] J. Zheng *et al.*, "Data synchronization in vehicular digital twin network: A game theoretic approach," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 7635–7647, 2023.

[4] Y. Wang *et al.*, "Joint communication and computing optimization for digital twin synchronization with aerial relay," in *IEEE Int. Conf. Commun. (ICC)*, 2024, pp. 3907–3912.

[5] X. Li *et al.*, "Wirelessly powered crowd sensing: Joint power transfer, sensing, compression, and transmission," *IEEE Journal Sel. Areas Commun.*, vol. 37, no. 2, pp. 391–406, 2019.

[6] J.-B. Wang *et al.*, "Joint optimization of transmission bandwidth allocation and data compression for mobile-edge computing systems," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2245–2249, 2020.

[7] X. Zhang *et al.*, "Energy-efficient computation offloading and data compression for uav-mounted mec networks," in *IEEE Globecom*, 2023, pp. 6904–6909.

[8] H. Liao *et al.*, "Integration of 6g signal processing, communication, and computing based on information timeliness-aware digital twin," *IEEE Journ. of Sel. Topics in Signal Proc.*, vol. 18, no. 1, pp. 98–108, 2024.

[9] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing* . Pearson Deutschland, 2013.

[10] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[11] A. Bressan, "Noncooperative Differential Games," *Milan Journal of Mathematics*, vol. 79, no. 2, pp. 357–427, Dec. 2011.

[12] Y. Xu *et al.*, "A one-leader multi-follower bayesian-stackelberg game for anti-jamming transmission in uav communication networks," *IEEE Access*, vol. 6, pp. 21 697–21 709, 2018.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, England: Cambridge University Press, Mar. 2004.

[14] W. Yu and R. Lui, "Dual methods for nonconvex spectrum optimization of multicarrier systems," *IEEE Trans. Commun.*, vol. 54, no. 7, pp. 1310–1322, 2006.

[15] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," https://cvxr.com/cvx, Mar. 2014.