

Markus Krantzik, Anja Klein, and Lin Xiang,

"Optimizing Sensor Data Compression and Digital Twin Synchronization via a Stackelberg Game", in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Istanbul, Turkey, September 2025.

©2025 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this works must be obtained from the IEEE.

Optimizing Sensor Data Compression and 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—In this paper, we investigate the efficient compression, transmission, and processing of high-volume sensor data collected from physical systems (PSs) to enable timely and accurate digital twin (DT) synchronization over resource-limited wireless networks. The sensors distributed in the PSs compress their sensed data prior to transmission to a base station (BS) for DT updating. However, due to the lack of a centralized decision-making unit, each sensor independently selects its own compression ratio to balance between its transmission time and compression overhead. To coordinate the compression across the sensors and ensure efficient DT updating globally, we formulate the problem as a Stackelberg game, where the BS acts as the leader for allocating communication/computing resources, while the sensors act as followers to optimize their data compression. By deriving each sensor's best response (BR), we further propose a low-complexity iterative algorithm to compute the Stackelberg equilibrium. Simulation results show that incorporating data compression significantly reduces DT synchronization time. Furthermore, 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) functioning as digital replicas of physical systems (PSs) present a transforming technology for future society. By collecting real-time data from a PS, a DT can be created and hosted on a cloud server to provide a comprehensive digital representation of the PS. Moreover, DTs can support extended reality services and enable seamless interaction with their physical counterparts. To ensure a successful integration of DTs into the sixth-generation (6G) wireless communication system [1], it is essential to achieve accurate and timely synchronization between the DT and its associated PS. However, meeting the strict demands for low latency, high reliability, and high data rates during this synchronization process remains a substantial challenge.

Recently, a wide range of research [2]–[4] has addressed the topic of DT synchronization in different aspects. In [2] and [3], game-theoretic approaches were employed to optimize communication and computing for effective DT synchronization. Specifically, [2] optimized synchronization intensity by strategically selecting virtual service providers for Internet of Things (IoT) devices, while [3] minimized synchronization delay by intelligently selecting relay nodes in vehicular networks. In [4], the authors proposed a relaying scheme using buffer-aided mobile relays and determined the optimal trajectory to fully exploit both communication and computing resources during DT synchronization.

However, the aforementioned works [2]–[4] primarily focus on the efficient allocation of communication and computing resources. In this paper, we aim to extend and enhance these approaches by integrating *sensor data compression* into DT synchronization. Data compression is a well-established technique for eliminating redundancy and reducing transmission overhead, thereby accelerating data delivery in communication, sensing, and task offloading applications [5]–[7].

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 16KISK014 and by DAAD with funds from the German Federal Ministry of Research, Technology and Space (BMFTR).

Nevertheless, compression also induces extra overhead in data processing. Consequently, directly applying the synchronization approaches in [2]–[4] to compressed sensor data, as done in [8], can be strictly suboptimal, as they neglect the fundamental trade-off between data reduction and processing overhead. Instead, joint optimization of data compression, communication, and computing for DT synchronization becomes crucial, but has not yet been explored in the existing literature. Furthermore, joint optimization typically requires a centralized decision-making unit, which is, however, overly complicated for DT synchronization with distributed sensors.

This paper aims to address both challenges identified above. To demonstrate the advantages of incorporating data compression into DT synchronization without complicating the mathematical notations, we focus on investigating the lossless compression method in this paper. Lossless compression ensures exact recovery of the original data at the receiver, making it suitable for various downstream applications. It is also a key component in many practical compression solutions. A recent lossless compression scheme has been shown to significantly reduce the size of large language models (LLMs) without compromising their accuracy [9].

Under this setting, we further develop a Stackelberg game-based resource allocation framework that jointly optimizes data compression, communication, and DT computing. Our proposed framework allows each sensor in the PS to independently compress its sensed data before transmitting the compressed data to the base station (BS) for updating the DT, without a centralized unit. Meanwhile, the decision-making across the sensors are coordinated for globally efficient DT synchronization by enabling the BS to lead the communication/computing resource allocation. Our contributions include:

- We formulate a Stackelberg game to model DT synchronization with lossless data compression at the distributed sensors in the PS. This game-theoretic framework jointly optimizes sensor-side data compression along with communication/computing resource allocation at the BS for efficient DT updating without a centralized unit.
- For each sensor, we analytically derive its best strategy for compression to balance between its transmit time and compression overhead. Based on this, we develop a low-complexity iterative algorithm to compute the Stackelberg equilibrium.
- Simulation results show that the derived equilibrium solution significantly reduces DT synchronization time, with almost no price of anarchy being paid.

II. SYSTEM MODEL

In this section, we present the system model for DT synchronization with sensor data compression.

A. DT Synchronization with Sensor Data Compression

As depicted in Figure 1, a cloud server is responsible for creating and synchronizing a DT of a dynamic PS, using data collected from K sensors in the PS, where sensor k is located at $\mathbf{u}_k^s$, $k \in \mathcal{K} \triangleq \{1, \dots, K\}$. Thereby, each sensor transmits D_k bits to the BS every T_{PS} seconds, while the latter

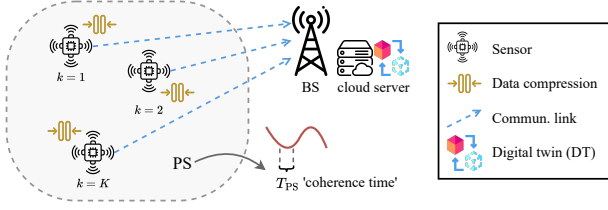


Fig. 1. Illustration of DT synchronization with data compression.

then forwards the data to the cloud server over a high-speed communication link. Based on Shannon's sampling theorem [10], this allows to accurately track any dynamics with a bandwidth of $1/T_{PS}$ Hz in the PS.

Due to limited radio resources for communication, each sensor first utilizes data compression to reduce the data volume before transmission to the BS. The cloud server then decompresses and further processes the received data. Therefore, *DT synchronization with data compression* encompasses the whole process of data compression, communication, decompression, and DT updating, as elaborated below.

1) *Data Compression and Decompression*: We employ lossless compression at the sensors such that the cloud server can exactly recover the sensed data without degrading the accuracy of DT models. Following [5]–[7], the data volume after compression is given by D_k/β_k , with $\beta_k \in [1, \beta_k^{\max}]$ denoting the compression ratio. Here, $\beta_k = 1$ and $\beta_k = \beta_k^{\max}$ indicate no data compression and the highest possible compression ratio for lossless data compression, respectively.

While the amount of compressed data to be transmitted decreases with β_k , reducing the time needed for transmission to the BS, this unfavorably increases the time needed to compress the sensor data. Therefore, a fundamental trade-off between transmission time and compression overhead exists. To capture this trade-off, the computational complexity of compression is modeled as [5]

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

where ϵ is a constant defined by the compression algorithm. The compression complexity $\eta(\beta_k, \epsilon)$ is zero when $\beta_k = 1$, and grows exponentially as a higher compression ratio is used. The time required to compress the sensed data at the sensors is given by

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

where f_k^S is the compression speed of a sensor.

Note that, due to large processing capabilities of the cloud server, the time required to decompress the transmitted data is negligible compared with T_k^{comp} . For this reason, it is ignored in this paper, similar to [7].

2) *Data Communication*: The sensors employ orthogonal frequency-division multiple access (OFDMA) to transmit the compressed data to the BS while effectively avoiding multi-user interference. To this end, the total bandwidth of B Hz is split into several nonoverlapping subchannels, which are allocated to the sensors for communication with the BS. Specifically, b_k denotes the bandwidth assigned to sensor $k \in \mathcal{K}$ and the available data rate for communication is defined using Shannon's capacity formula as

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

with the transmit power p_k , noise power spectral density N_0 , and complex channel gain $H_k \in \mathbb{C}$. The channel gain from

the k -th sensor to the BS is further modeled as

$$H_k = \sqrt{A_0} \cdot \|\mathbf{u}^{\text{BS}} - \mathbf{u}_k^s\|^{-\alpha/2} \cdot h_k, \quad (4)$$

where A_0 and $\alpha \geq 2$ denote the path loss at a reference distance of 1m and the path loss exponent, respectively. Here, $A_0/(d_k)^\alpha$ defines the path loss of the channel and $h_k \in \mathbb{C}$ denotes the complex channel fading coefficient [4]. Finally, the time needed by sensor $k \in \mathcal{K}$ to transmit the compressed data to the BS is given by

$$T_k^{\text{tr}}(\beta_k, b_k) = D_k / (\beta_k \cdot r_k(b_k)). \quad (5)$$

3) *DT Computing*: For processing the sensor data and updating the DT, the cloud server has a processing capacity of C_{DT} Hz for all K sensors. Based on the allocated processing speed f_k^{DT} , the time required to process the data from sensor $k \in \mathcal{K}$ and synchronize the DT is given by

$$T_k^{\text{DT}}(f_k^{\text{DT}}) = D_k \cdot c_k / f_k^{\text{DT}}, \quad (6)$$

where c_k is the computational complexity of DT processing.

B. DT Synchronization Time

Lastly, we define the total time occupied by sensor $k \in \mathcal{K}$ for compression, transmission, and processing of its data as

$$T_k^{\text{total}}(\beta_k, b_k, f_k^{\text{DT}}) = T_k^{\text{comp}} + T_k^{\text{tr}} + T_k^{\text{DT}}. \quad (7)$$

It is assumed that each sensor is able to start compressing, transmitting, or processing its data independently, without waiting for other sensors. The time required to complete the DT update of the whole PS is defined by the maximum of T_k^{total} and is referred to as *DT synchronization time*.

III. PROBLEM FORMULATION

Based on (7), the fundamental trade-off in DT synchronization between data compression and data transmission is also dependent on DT computing. In order to address such complicated trade-offs for ultimately lowering the DT synchronization time, it requires intelligently optimizing the selection of the compression ratio β_k , the allocated bandwidth b_k , and the DT processing speed f_k^{DT} . However, there exists no central unit that has control over all variables. On the contrary, each sensor is responsible for independently choosing its own compression ratio while the BS allocates the bandwidth and DT processing speed, both following distinct objectives. Motivated by these strategic interactions between the sensors and the BS, in this section, we further propose a multi-follower Stackelberg game-based resource allocation framework for DT synchronization with data compression.

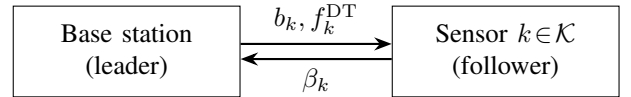


Fig. 2. Illustration of the introduced multi-follower Stackelberg game for DT synchronization with data compression.

A. Stackelberg Game Formulation

Figure 2 illustrates the proposed Stackelberg game, where the BS acts as the leader in allocating the available bandwidth b_k and DT processing speed f_k^{DT} . In reaction to the resource allocation from the BS, each sensor acts as a follower that optimizes its compression. Specifically, for a given b_k , sensor

k aims to minimize the time required for data compression and transmission to the BS as defined in their own cost function

$$U_k^f(\beta_k; b_k) = T_k^{\text{comp}}(\beta_k) + T_k^{\text{tr}}(\beta_k, b_k). \quad (8)$$

Problem P_k^f describes the sub-game of the k -th follower.

$$(P_k^f): \min_{\beta_k} U_k^f(\beta_k; b_k) \quad (9)$$

$$\text{s.t. C1.f: } 1 \leq \beta_k \leq \beta^{\max}. \quad (9a)$$

Constraint C1.f ensures a valid $\beta_k \in [1, \beta^{\max}]$. On the other hand, the BS aims to coordinate the self-interested sensors. Based on the $\beta_k, k \in \mathcal{K}$, the BS aims to coordinate the sensors by minimizing the maximum DT synchronization time $T_k^{\text{total}}(\beta_k, b_k, f_k^{\text{DT}})$. Therefore, the cost function is defined as

$$U^l(\mathbf{b}, \mathbf{f}; \beta) = \max_{k \in \mathcal{K}} T_k^{\text{total}}(\beta_k, b_k, f_k^{\text{DT}}), \quad (10)$$

where $\mathbf{b} \triangleq \{b_k | k \in \mathcal{K}\}$, $\mathbf{f} \triangleq \{f_k^{\text{DT}} | k \in \mathcal{K}\}$, and $\beta \triangleq \{\beta_k | k \in \mathcal{K}\}$. Problem P_{DT}^l defines the sub-game of the leader.

$$(P_{\text{DT}}^l): \min_{\mathbf{b}, \mathbf{f}} U^l(\mathbf{b}, \mathbf{f}; \beta) \quad (11)$$

$$\text{s.t. C1.l: } \sum_{k=1}^K b_k \leq B, \quad (11a)$$

$$\text{C2.l: } \sum_{k=1}^K f_k^{\text{DT}} \leq C_{\text{DT}}. \quad (11b)$$

Constraints C1.l and C2.l restrict the bandwidth and DT processing speed allocation, respectively.

B. Stackelberg Equilibrium

Given each b_k determined by the leader, the solution of a follower's sub-game P_k^f is defined by the best response (BR) below

$$\text{BR}(b_k) \triangleq \left\{ \beta_k : \min_{1 \leq \beta_k \leq \beta^{\max}} U_k^f(\beta_k; b_k) \right\}, \quad (12)$$

which specifies the optimal β_k that minimizes the sensor's cost function. By playing their sub-games, the sensors and the BS, namely the followers and the leader, aim to reach an equilibrium strategy, referred to as the Stackelberg Equilibrium (SE). According to [11], the SE consists of the optimal resource allocation $\mathbf{b}^*, \mathbf{f}^*$ and the optimal compression ratios β^* that satisfy $\beta_k^* \in \text{BR}(b_k)$ and

$$U^l(\mathbf{b}^*, \mathbf{f}^*; \beta^*) \leq U^l(\mathbf{b}, \mathbf{f}; \beta), \forall \beta_k \in \text{BR}(b_k), \quad (13)$$

where $(\cdot)^*$ is used to indicate the best decisions of the sensors and the BS. Upon reaching the SE, neither sensors nor the BS would like to change their decision [12].

IV. PROBLEM SOLUTION

To derive the SE solution for the introduced Stackelberg Game, we first characterize the BR of the sensors in closed form. Subsequently, following the *backward-induction* [12] approach, the derived BR is utilized in the leader's problem to find the optimal resource allocation for the BS. Finally, based on these results, the SE solution for DT synchronization with data compression is calculated using an iterative algorithm.

A. Strategy Analysis for the Followers

Note that in problem (9), each follower has its own objective function and constraints. As such, its decision is

independent of the other followers and thus, the BR of sensor $k \in \mathcal{K}$ can be derived by solving problem (9).

We first show that strong duality holds for problem (9). To this end, an auxiliary variable $\mu_k > 0$ is introduced to replace $1/\beta_k$ in (8). Subsequently, the objective function is rewritten as

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

for $\mathbf{x}_k \triangleq [\beta_k, \mu_k]^\top$, and the optimization problem of the follower is reformulated into

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

$$\text{s.t. C1.f: } 1 \leq \beta_k, \quad (15a)$$

$$\text{C2.f: } 1/\mu_k \leq \beta_k. \quad (15b)$$

Since the completion time increases with $1/\beta_k$, the additional constraint C2.f ensures that problem $P_k^{f,2}$ is equivalent to (9). Note that $P_k^{f,2}$ is a convex optimization problem that satisfies Slater's condition, for which strong duality holds in $P_k^{f,2}$ [13].

Based on the strong duality result, we now derive the followers' BR in closed form by solving the corresponding dual problem of $P_k^{f,2}$. To define the dual problem, we first introduce the dual variables $\lambda_{1,k}, \lambda_{2,k} \geq 0$ for constraints C1.f and C2.f, respectively. Accordingly, the dual vector $\lambda_k = [\lambda_{1,k}, \lambda_{2,k}]^\top$ and the Lagrangian function

$$L_k(\lambda_k, \mathbf{x}_k; b_k) = f_k(\mathbf{x}_k; b_k) + \lambda_k^\top \mathbf{g}(\mathbf{x}_k) \quad \text{with} \quad (16a)$$

$$\mathbf{g}(\mathbf{x}_k) = \begin{pmatrix} 1 - \beta_k \\ 1/\mu_k - \beta_k \end{pmatrix} \quad (16b)$$

are defined. The dual problem of $P_k^{f,2}$ can be defined as

$$(P_k^{f,3}): \max_{\lambda_k \geq 0} D(\lambda_k; b_k) \quad (17)$$

with the dual function given by

$$D(\lambda_k; b_k) \triangleq \min_{\mathbf{x}_k} L_k(\lambda_k, \mathbf{x}_k; b_k) \quad (18)$$

$$\text{s.t. C1.f and C2.f.}$$

The optimal solution to the minimization problem in (18) can be analytically derived, despite its constrained form, as shown in the following lemma.

Lemma 1. *The optimal compression ratio β_k^* and the optimal value of the auxiliary μ_k^* are functions of the dual variables $\lambda_k \geq 0$ as given by*

$$\beta_k^* = \min \left\{ \frac{1}{\epsilon} \ln \left(\frac{f_k^S}{\epsilon \cdot D_k} (\lambda_{1,k} + \lambda_{2,k}) \right), \beta^{\max} \right\}, \quad (19)$$

$$\mu_k^* = \sqrt{\lambda_{2,k} \cdot r_k(b_k) / D_k}. \quad (20)$$

Proof. The first-order derivative of the Lagrangian function $L_k(\lambda_k, \mathbf{x}_k; b_k)$ with respect to the DT processing speed f_k^{DT} is given by

$$\frac{\partial}{\partial \beta_k} L_k(\lambda_k, \mathbf{x}_k; b_k) = \frac{\epsilon \cdot D_k}{f_k^S} \cdot e^{\beta_k \cdot \epsilon} - \lambda_{1,k} - \lambda_{2,k}. \quad (21)$$

Thus, the optimal β_k^* in (19) can be obtained by finding the root for $\frac{\partial}{\partial \beta_k} L_k(\lambda_k, \mathbf{x}_k; b_k)$ within the interval $\beta_k \leq \beta^{\max}$. Meanwhile, the optimal value of the auxiliary variable μ_k^* can be similarly derived based on the first-order derivative with

respect to μ_k ,

$$\frac{\partial}{\partial \mu_k} L_k(\lambda_k, \mathbf{x}_k; b_k) = \frac{D_k}{r_k(b_k)} - \lambda_{2,k} \cdot \frac{1}{(\mu_k)^2}, \quad (22)$$

for which a positive root can be easily identified. $\square$

Thanks to Lemma 1, we can easily obtain a sub-gradient [14] for the dual function in (18) as

$$\mathbf{g}(\mathbf{x}_k^*(\lambda_k)) = \left(\frac{1 - \beta_k^*}{1/\mu_k^* - \beta_k^*} \right), \quad (23)$$

which can be further utilized to solve the dual problem in (17). This is achieved via an iterative gradient ascent search defined by

$$\lambda_k^{(l+1)} = \left[\lambda_k^{(l)} + \mathbf{h}_\lambda^{(l)} \circ \mathbf{g}(\mathbf{x}_k^*(\lambda_k^{(l)})) \right]^+, \quad (24)$$

where $(\cdot)^{(l)}$ denotes the iteration index l , $\circ$ is the element-wise multiplication operator, and $[\cdot]^+ \triangleq \max\{0, \cdot\}$. We set $\lambda_k^{(0)}$ as the initial value of the dual vector and determine the step size $\mathbf{h}_\lambda^{(l)}$ via line search.

B. Strategy Analysis for the Leader

Meanwhile, for given λ_k , we can derive the optimal strategy of the leader via backward induction. Specifically, by substituting the followers' solution $\mathbf{x}_k^*(\lambda_k)$ in (7), the completion time T_k^{total} is rewritten as

$$T_k^{\text{total}}(b_k, f_k^{\text{DT}}; \lambda_k) = \frac{D_k}{f_k^{\text{S}}} \eta(\beta_k^*, \epsilon) + \frac{D_k \mu_k^*}{r_k(b_k)} + \frac{c_k D_k}{f_k^{\text{DT}}}. \quad (25)$$

Moreover, the optimization problem $P_{\text{DT}}^{1,2}$ of the leader is redefined as

$$\left(P_{\text{DT}}^{1,2} \right) : \min_{\mathbf{b}, \mathbf{f}} \max_{k \in \mathcal{K}} T_k^{\text{total}}(b_k, f_k^{\text{DT}}; \lambda_k) \quad (26)$$

s.t. C1.1 and C2.1.

Note that for given λ_k , the completion time $T_k^{\text{total}}(b_k, f_k^{\text{DT}}; \lambda_k)$ is strictly convex in b_k and f_k^{DT} . Thus, problem $P_{\text{DT}}^{1,2}$ is a convex optimization problem and can be conveniently solved using e.g. CVX solver [15].

C. Proposed Iterative Algorithm

It remains to derive the SE of the introduced Stackelberg game, which is readily achieved via an iterative algorithm that integrates the gradient ascent search in (24). The overall procedure is presented in Algorithm 1, which requires inputs about the initial value dual variables $\lambda_k^{(0)}$ for all K sensors. In each iteration, indexed by l , the leader solves problem $P_{\text{DT}}^{1,2}$ to obtain the optimal resource allocation and exchange it with the followers (cf. lines 2 and 3). Subsequently, each sensor $k \in \mathcal{K}$ employs the gradient-ascent search in (24) to update $\lambda_k^{(l)}$ and compute the corresponding $\beta_k^{(l)}$ (cf. lines 5 and 6). The updated decisions of the leader and the followers are used to calculate the maximum completion time T^{total} and the difference to the last iteration $\Delta T_{\text{total}}^{(l)}$ (cf. lines 8 and 9). The procedure continues until the difference between two iterations $\Delta T_{\text{total}}^{(l)}$ is smaller than ξ or the maximum number of iterations is reached. As stated in Proposition 2, under mild conditions Algorithm 1 will converge to the SE of the introduced Stackelberg game. The complexity and scalability of Algorithm 1, utilizing the interior-point method, can be approximated using big- $\mathcal{O}$ notation as $\mathcal{O}(K^{3.5} L_{\text{max}})$.

Proposition 2. For sufficiently large L_{max} , Algorithm 1 converges to the solution $(\mathbf{b}^*, \mathbf{f}^*)$ and β^* , which defines a unique SE of the Stackelberg game introduced in (9) and (11).

Proof. First of all, since the sensors' problems are always feasible, the existence of $\text{BR}(b_k)$ is guaranteed. This implies that the domain $\mathcal{R} \triangleq \{(\mathbf{b}, \mathbf{f}, \beta) \mid (\mathbf{b}, \mathbf{f}) \text{ feasible to (12) and } \beta_k \in \text{BR}(b_k), \forall k \in \mathcal{K}\}$ is a nonempty and compact set. As the cost-function of the BS $U^1(\mathbf{b}, \mathbf{f}; \beta)$ is a continuous function in $(\mathbf{b}, \mathbf{f}, \beta)$, the SE should always exist. Furthermore, each subgame can be transformed into a convex optimization problem, whose objective function is either convex or strictly convex with respect to $(\mathbf{b}, \mathbf{f}, \beta)$. Thus, $\mathcal{R}$ is a convex set. As (13) minimizes a strictly convex function over a convex set, the SE solution is unique. Lastly, a sufficiently large L_{max} guarantees convergence of the gradient-ascent searches to the sensors' BRs, following which Algorithm 1 converges to the SE of the introduced Stackelberg game. $\square$

Algorithm 1 Iterative Algorithm for Finding the SE

Input: Initial $\lambda_k^{(0)}$ for all sensors $k \in \mathcal{K}$ and iteration index $l=0$.

Output: Optimal bandwidth allocation $\mathbf{b}^*$, DT processing speeds $\mathbf{f}^*$, and compression ratios β^* .

```

1: repeat
2:   Set  $l = l + 1$ .
3:   Obtain the resource allocation  $\mathbf{b}^{(l)}$  and  $\mathbf{f}^{(l)}$  by solving the problem
     of leader  $P_{\text{DT}}^{1,2}$  ▷ Eq. (26)
4:   for all  $k \in \mathcal{K}$  do
5:     Use  $\mathbf{b}^{(l)}$  and  $\mathbf{f}^{(l)}$  to receive  $\lambda_k^{(l)}$  via one step of the gradient-
       ascent search. ▷ Eq. (24)
6:     Use  $\lambda_k^{(l)}$  to update  $\beta_k^{(l)}$  in  $\beta^{(l)}$  ▷ Eq. (19)
7:   end for
8:   Use  $\mathbf{b}^{(l)}$ ,  $\mathbf{f}^{(l)}$ , and  $\beta^{(l)}$  to update  $U^1(\mathbf{b}^{(l)}, \mathbf{f}^{(l)}; \beta^{(l)})$ . ▷ Eq. (10)
9:   Calculate the difference to the last iteration  $\Delta T_{\text{total}}^{(l)}$ .
10: until  $\Delta T_{\text{total}}^{(l)} \leq \xi$  or  $l \geq L_{\text{max}}$ 
11: Set the output  $\mathbf{b}^* = \mathbf{b}^{(l)}$ ,  $\mathbf{f}^* = \mathbf{f}^{(l)}$ , and  $\beta^* = \beta^{(l)}$ .
```

V. SIMULATION RESULTS

To evaluate the performance of the proposed algorithm, in this section we simulate a PS with $K=4$ sensors deployed at an equal distance of $d_k=378$ m to the BS. Each sensor is able to utilize a compression speed of $f_k^{\text{S}}=200$ kHz to compress its data up to a compression ratio of $\beta^{\text{max}}=3$ for lossless compression. The sensors share the total available bandwidth $B=100$ kHz for communication and the DT processing capacity of $C_{\text{DT}}=50$ MHz for DT updating.

TABLE I
SIMULATION PARAMETERS [4], [5]

Parameter	Value	Parameter	Value
N_0	-90 dBm	A_0	-30 dB
ϵ	3.5	α	2
ξ	10^{-6}	L_{max}	30

We start with evaluating a *uniform* scenario in Section V-A, where each sensor transmits $D_k=4$ kbits to the BS at the same transmit power $p_k=15$ dBm ≈ 31.6 mW. At the cloud server, each sensor also has the same computing complexity $c_k=10^4$ cycles/bit for DT updating. Only the channel fading gain h_k , assuming Rician channel fading as in [4], is randomly generated for each sensor. Similar to [4], the standard deviation of the channel fading gain is set to $\sigma_k=1$ and the energy ratio between the line-of-sight (LoS) and non-LoS (NLoS) components to $\kappa=10$. Such a uniform setting allows us to evaluate the impact of channel fading, before

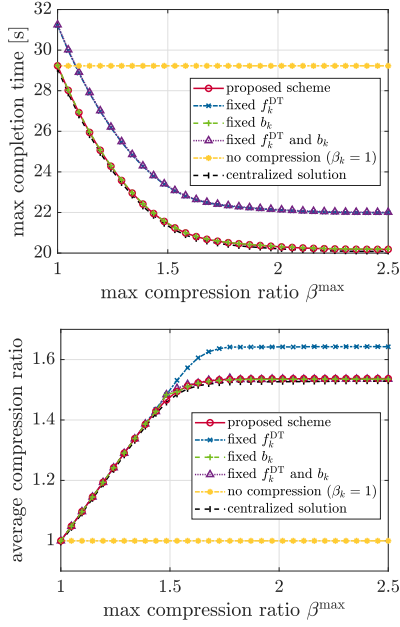


Fig. 3. (Top) Maximum completion time T_k^{total} and (below) average compression ratio $\bar{\beta}$ versus maximum compression ratio β^{max} .

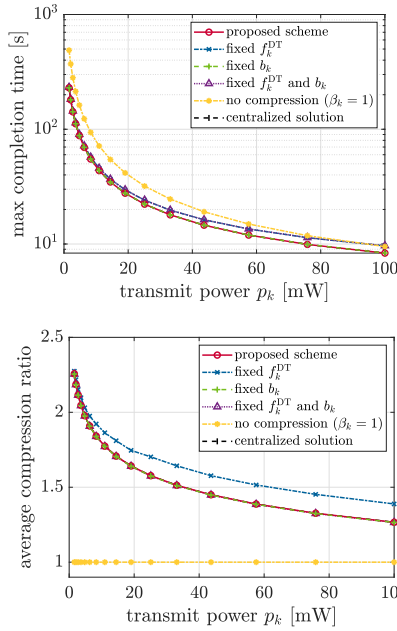


Fig. 4. (Top) Maximum completion time T_k^{total} in logarithmic scale and (below) average compression ratio $\bar{\beta}$ versus transmit power p_k .

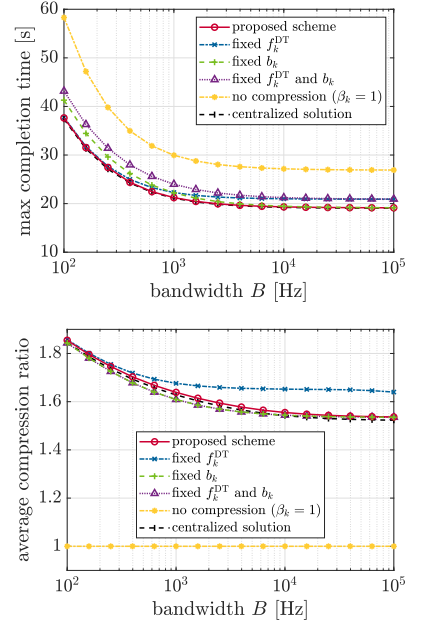


Fig. 5. (Top) Maximum completion time T_k^{total} and (below) average compression ratio $\bar{\beta}$ versus available bandwidth B in logarithmic scale.

extending the evaluation to the joint impact of channel fading and *non-uniform* scenarios in Section V-B. The remaining parameters are defined in Table I.

A. Performance Comparison in A Uniform Scenario

We compare the proposed scheme with several baseline schemes. Each baseline scheme fixes a subset of the optimization variables in the leader's or the followers' optimization problem and optimizes only the remaining variables using modified Algorithm 1, as stated in the following:

- *Baseline scheme 1* fixes the allocation of the DT processing speed $f_k^{\text{DT}} = C_{\text{DT}}/K$.
- *Baseline scheme 2* fixes bandwidth allocation $b_k = B/K$.
- *Baseline scheme 3* fixes $f_k^{\text{DT}} = C_{\text{DT}}/K$ 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 (10) by jointly optimizing β_k , b_k , and f_k^{DT} constrained by (9a), (11a), and (11b) in a centralized manner.

In Figure 3, we evaluate the maximum completion time T_k^{total} and the average of the optimal compression ratios $\bar{\beta} = \frac{1}{K} \sum_{k=1}^K \beta_k^*$ for the proposed scheme and all baseline schemes with varying maximum compression ratios β^{max} . Without compressing the data, baseline scheme 4 has the highest completion time among all schemes. The other schemes are able to utilize data compression to lower the maximum T_k^{total} by increasing β_k until reaching an optimal trade-off between the extra time required to compress the sensed data T_k^{comp} and the time reduction for transmission T_k^{tr} . Consequently, the completion time saturates when $\bar{\beta}$ surpasses 1.54. Since the proposed scheme strategically adapts the DT processing speed f_k^{DT} , bandwidth b_k , and compression ratios β_k , it reduces the maximum T_k^{total} by up to 30.9% in comparison to baseline scheme 4 and by 8.3% compared to baseline schemes 1 and 3. Interestingly, although baseline schemes 1 and 3 achieve approximately the same completion time, baseline scheme 1 results in a higher $\bar{\beta}$. This is because the sensor with the worst channel gain generally limits the performance, for which higher b_k and f_k^{DT} are assigned in order to keep the overall

completion time low. In baseline scheme 1, fixing f_k^{DT} causes an even larger b_k to be allocated to the worst sensor. This in turn increases β_k for all other sensors.

Figure 4 compares the maximum completion time T_k^{total} and the compression ratio β_k versus the transmit power $p_k \in [1.6 \text{ mW}, 100 \text{ mW}]$. For all DT synchronization schemes employing data compression, both T_k^{total} and $\bar{\beta}$ decrease as p_k increases since a higher transmit power leads to an increased transmit data rate, which in turn reduces the need for data compression and its overhead. At a low transmit power of $p_k = 1.6 \text{ mW}$, the proposed scheme significantly outperforms baseline scheme 4 by 54.6%, owing to the use of a high β_k . However, in the regime of small transmit powers, the DT synchronization schemes employing data compression achieve similar performance. This is because the transmission time T_k^{tr} dominates the DT synchronization time, while the gain from optimizing b_k or f_k^{DT} is only negligible. When p_k becomes large, optimizing the DT processing speed f_k^{DT} becomes crucial for accelerating DT synchronization. Here, compared to baseline schemes 1 and 3, the proposed scheme can reduce T_k^{total} by 12.0%.

In Figures 3 and 4, baseline scheme 2 performs very close to the proposed scheme. For insights into this, Figure 5 further evaluates the maximum completion time T_k^{total} and the compression ratio $\bar{\beta}$ versus the available bandwidth $B \in [100 \text{ Hz}, 100 \text{ kHz}]$. As expected, increasing B results in a lower T_k^{total} and a lower $\bar{\beta}$. In the lower bandwidth regime, the proposed scheme significantly outperforms baseline scheme 2 and 3, due to adaptive optimization of bandwidth allocation. However, 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 $B = 100 \text{ Hz}$ and by 8.9% compared to baseline schemes 1 and 3 at $B = 100 \text{ kHz}$. Figure 5 also implies that adopting fixed f_k^{DT}

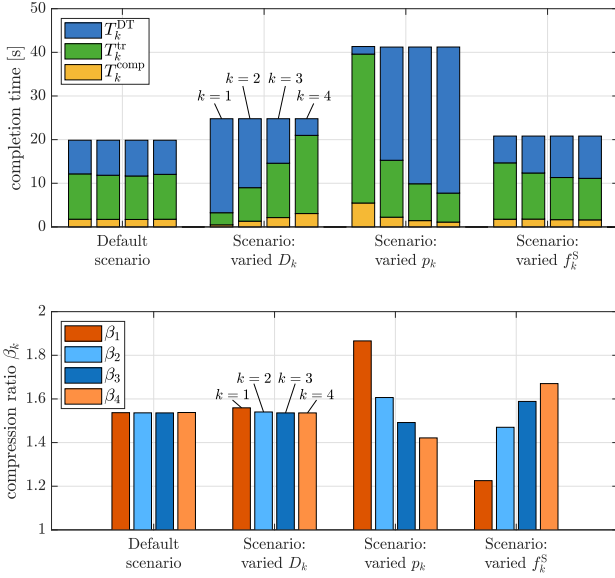


Fig. 6. (Top) Completion time $\{T_k^{\text{comp}}, T_k^{\text{tr}}, T_k^{\text{DT}}\}$ and (below) compression ratio β_k for $K=4$ sensors in different scenarios.

and fixed b_k , as in baseline schemes 1 and 2, is close-to-optimal in the low and high bandwidth regime, respectively.

In Figures 3 to 5, 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 6 compares the completion time including T_k^{comp} , T_k^{tr} , and T_k^{DT} , and the compression ratio β_k of each sensor. In addition to the *uniform* scenario considered so far, three *non-uniform* scenarios are newly defined and simulated. In these non-uniform scenarios, the volume of sensor data, the transmit power, or the compression speed is varied across the $K=4$ sensors according to $D_k = v_k \times 4$ kbit, $p_k = v_k \times 31.6$ mW, or $f_k^S = v_k \times 200$ kHz, respectively, where v_k is a weight assigned to sensor k . We set the weights as $v_1 = 0.25, v_2 = 0.75, v_3 = 1.25$, and $v_4 = 1.75$, to redistribute the available resources among the sensors. From Figure 6 we observe that, compared to the default uniform scenario, redistributing the volume of data D_k across the sensors results in an 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. (19)) 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} . 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 penalizing the time 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 increases the time for transmission and DT synchronization. In Figure 6, 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 adjust flexibly 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

We investigated the joint optimization of sensor-side lossless data compression and DT synchronization to balance the goals of timely DT updating and reducing redundancy in the sensor data, without a centralized decision-making unit. Strategic interactions between the sensors (followers) and the BS (leader) were modeled as a multi-follower Stackelberg game, where the sensors adjust their compression ratios while the BS allocates communication and computing resources. By exploiting the closed-form solution of the followers' BR, we defined a low-complexity iterative algorithm for efficiently calculating the SE strategy. Extensive simulations across a variety of scenarios demonstrated that the strategic data compression ultimately enables accelerated DT synchronization. Moreover, the proposed algorithm can efficiently coordinate the self-interested sensors with the BS to achieve DT synchronization times that are just as low as those achieved by 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.
- [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.
- [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] T. Zhang *et al.*, "70% size, 100% accuracy: Lossless LLM compression for efficient GPU inference via dynamic-length float," 2025. [Online]. Available: <https://arxiv.org/abs/2504.11651>
- [10] A. Oppenheim and R. Schaffer, *Discrete-Time Signal Processing*. Pearson Deutschland, 2013.
- [11] A. Bressan, "Noncooperative differential games," *Milan Journal of Mathematics*, vol. 79, no. 2, pp. 357–427, 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, 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>, 2014.