INVITED PAPER

Learning and Management for Internet of Things: Accounting for Adaptivity and Scalability

In the context of online learning and management policies, this article illustrates ways of dealing with extreme heterogeneity, massive number of devices, and unpredictable dynamics in the context of the Internet of Things.

By Tianyi Chen, *Student Member IEEE*, Sergio Barbarossa, *Fellow IEEE*, Xin Wang, *Senior Member IEEE*, Georgios B. Giannakis, *Fellow IEEE*, and Zhi-Li Zhang, *Fellow IEEE*

ABSTRACT | Internet of Things (IoT) envisions an intelligent infrastructure of networked smart devices offering task-specific monitoring and control services. The unique features of IoT include extreme heterogeneity, massive number of devices, and unpredictable dynamics partially due to human interaction. These call for foundational innovations in network design and management. Ideally, it should allow efficient adaptation to changing environments, and low-cost implementation scalable to a massive number of devices, subject to stringent latency constraints. To this end, the overarching goal of this paper is to outline a unified framework for online learning and management policies in IoT through joint advances in communication, networking, learning, and optimization. From the network architecture vantage point, the unified framework leverages a promising fog architecture that enables smart devices to have proximity access to cloud functionalities at the network edge, along the cloud-to-things continuum. From the algorithmic perspective, key innovations target online approaches adaptive to different degrees of nonstationarity in IoT dynamics, and their scalable model-free implementation under limited feedback that motivates blind or bandit approaches. The proposed framework aspires to offer a stepping stone that leads to systematic designs and analysis of task-specific learning and management schemes for IoT, along with a host of new research directions to build on.

KEYWORDS | Internet of Things (IoT); mobile edge computing (MEC); network resource allocation; online learning; stochastic optimization

Manuscript received June 23, 2018; revised October 25, 2018; accepted January 25, 2019. Date of publication February 21, 2019; date of current version March 25, 2019. The work of T. Chen and G. B. Giannakis was supported by the National Science Foundation (NSF) under Grant 1509040, Grant 1508993, and Grant 1711471. The work of S. Barbarossa was supported by the H2020 Europe-Japan Project 5G-MiEdge under Grant 723171. The work of X. Wang was supported in part by the National Natural Science Foundation of China under Grant 61671154, in part by the National Key Research and Development Program of China under Grant 2017YFB0403402, and in part by the Innovation Program of Shanghai Municipal Science and Technology Commission under Grant 17510710400. The work of Z.-L. Zhang was supported in part by the U.S. Department of Defense (DoD) under Contract HDTRA1-14-1-0040 and in part by NSF under Grant 1411636 and Grant 1617729.
(Corresponding author: Tianyi Chen.)

T. Chen and **G. B. Giannakis** are with the Digital Technology Center, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: chen3827@umn.edu; georgios@umn.edu).
S. Barbarossa is with the Department of Information Engineering, Electronics, and Telecommunications, Sapienza University of Rome, 00184 Rome, Italy (e-mail: sergio.barbarossa@uniroma1.it).
X. Wang is with the Shanghai Institute for Advanced Communication and Data Science, Shanghai 200444, China, also with the Key Laboratory for Information Science of Electromagnetic Waves, Fudan University, Shanghai 200433, China, and also with the Department of Communication Science and Engineering, Fudan University, Shanghai 200433, China (e-mail: xwang11@fudan.edu.cn).
Z.-L. Zhang is with the Department of Computer Science, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: zhang@cs.umn.edu).

Digital Object Identifier 10.1109/JPROC.2019.2896243

I. INTRODUCTION

The past decade has witnessed a proliferation of connected devices and objects, where the notion of Internet of Things (IoT) plays a central role in the envisioned technological advances. Conceptually speaking, IoT foresees an intelligent network infrastructure with ubiquitous smart devices—home automation, interactive healthcare, and self-driving connected vehicles—that are typical in

IoT [2], [3] (see Fig. 1). Today, a number of IoT applications have already brought major benefits to many aspects of our daily life. The current generation of IoT can already afford an increasing amount of real-time automation, and thus intelligence toward the vision of real-time IoT. However, despite the popularity of IoT, several critical challenges must be addressed before embracing its full potential [4], [5]. To this end, we highlight three key challenges that are arguably expected to be at the epicenter of emerging IoT research fields.

A. Extreme Heterogeneity

The computational and communication capacities of devices differ due to differences in hardware (e.g., CPU frequency), communication protocol (e.g., ZigBee), and energy availability (e.g., battery level) [6]. The tasks carried out on various devices are often considerably diverse, e.g., motion sensors monitor human behavior in a smart home [7], while cameras are responsible for recognizing a suspicious behavior in a crowded environment, or vehicle plates in a parking garage.

B. Unpredictable Dynamics

Unlike many existing communication, computing, and networking platforms, the IoT dynamics can stem from multiple sources, where adaptivity is not only critical but also essential in designing hardware and management protocols. Such sources entail human-in-the-loop dynamics in addition to physical objects [7], demand response in energy systems [8], and intelligent automotive applications [9]. In these applications, IoT dynamics are intertwined with or even partially determined by human behavior [10]–[12] as such a high degree of adaptivity in the algorithm and hardware design is needed.

C. Scalability at the Core

IoT entails an intelligent network infrastructure with a massive number of devices. It is estimated that by 2020, there will be more than 50 billion devices connected through the Internet [13], which highlights scalability as a key challenge for IoT [2], [4]. Scalability is not only about computational efficiency but also about lower communication overhead (e.g., how often a device needs to communicate with the remote cloud center), as well as reduced information needed (e.g., what type of information a device needs before making sensible decisions).

Faced with these major IoT challenges, innovations in network design and management are desired to enable efficient online operations and seamless coexistence of humans with things [14]. Consequently, it is imperative to develop new tools for IoT management that tap into diverse inference, signal processing, communications, and networking techniques, by drawing from fields such as machine learning, optimization, and applied statistics. The novel expertise gleaned from these research areas, coupled with the solid analytical approach, are the best credentials for succeeding in IoT research [4].

From a network architecture perspective, to ensure the desired user experience and meet heterogeneous service requirements, IoT tasks are no longer only supported by the cloud data centers nowadays but also through a promising new architecture termed edge computing or in a broader sense fog computing. This architecture distributes computation, communication, and storage closer to the end IoT devices and users, along the cloud-to-things continuum [14]–[19]. This shift of computing paradigms is further promoted by the advanced communication techniques emerging with standards such as narrowband IoT (NB-IoT) [20], [21].

Given the huge volume of data in various IoT setups and the proliferation of learning and large-scale optimization advances, a pertinent direction is prompted by asking the following question: can we learn from historical data to improve the quality of network management policies in IoT? The rationale is that historical data contain statistics of the IoT environments [22], and learning from them can mitigate the uncertainty of future management tasks. Further armed with online adaptation capability to reinforce the current policies, it is envisioned that learn-and-adapt network management schemes can markedly improve IoT user experience in terms of low service delay, high system resilience, and adaptivity [23], [24]. Toward this goal, this paper will outline an offline-aided online approach with markedly improved performance by leveraging statistical learning from historical samples.

Taking a step further, online learning, with online convex optimization (OCO) as a special case, is an emerging methodology for sequential decision making with lightweight implementation and well-documented merits, especially when the environment (e.g., a sequence of convex costs) varies in an unknown and possibly adversarial manner [25], [26]. Targeting a scalable solution in a prohibitively complex IoT environment, this paper will also overview a new OCO framework designed for IoT, which further incorporates various forms of feedback, physical constraints, and performance metrics driven by IoT applications, relative to the standard settings [25]–[27]. Novel schemes tailored for this setting can lay a

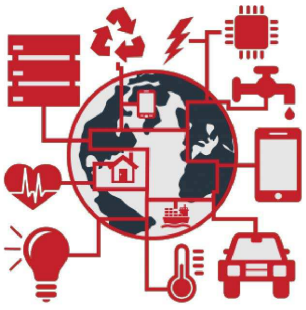


Fig. 1. ***Internet of Things [1].***

solid analytical foundation to delineate the trade-offs among algorithm scalability, performance guarantees, and degree of (non)stationarity present in the IoT environment [28], [29].

The rest of this paper is organized as follows. Section II deals with the heterogeneity in IoT demand and quality of service (QoS), along with a unified formulation for dynamic IoT tasks. Section III introduces methods for optimizing IoT performance under different levels of nonstationarity in IoT dynamics. Section IV summarizes scalable OCO-based schemes with different feedback options. Finally, concluding remarks and possible future research directions are highlighted in Section V.

Notation: Bold uppercase (lowercase) letters denote matrices (column vectors), while $(\cdot)^\top$ stands for transposition, and $\|\mathbf{x}\|$ denotes the ℓ_2-norm of a vector $\mathbf{x}$. The projection $[\mathbf{a}]^+ := \max\{\mathbf{a}, \mathbf{0}\}$ are defined entrywise. The indicator function $\mathbb{1}(A)$ takes value 1 when the event A happens, and 0 otherwise. $\mathcal{O}(x)$ denotes big order of x, i.e., $\mathcal{O}(x)/x \to 1$ as $x \to 0$; $\tilde{\mathcal{O}}$ neglects the lower order terms with a polynomial $\log x$ rate; and $\mathbf{o}(x)$ denotes small order of x, i.e., $\mathbf{o}(x)/x \to 0$ as $x \to 0$.

II. HETEROGENEITY IN IoT DEMAND AND QoS

Heterogeneity is inherent in IoT, and it manifests itself across different aspects, from application requirements and constraints to sensing and communication technologies.

A. Heterogeneous Applications

The range of IoT applications already spans several fields, and it is rapidly increasing. A few examples of applications are as follows [3], [5], [21]: 1) lifestyles (wearable gadgets, gaming, augmented/virtual reality, and wellness); 2) smart environments (homes, offices, and cities); 3) automotive (self-driving, traffic monitoring, intelligent transportation systems, and vehicle-to-vehicle communications); 4) industrial (full automation and control, structure monitoring, and logistic); 5) environmental monitoring (pollution, global warming, and waste management); and 6) healthcare (patient monitoring, body area networks, smart health, and elderly care); and 7) security and surveillance. These applications are characterized by highly diverse requirements in terms of data rate, latency, reliability, security, connectivity, mobility, and so on. To illustrate the extreme variability of requirements, we note that virtual reality require latencies in the order of a few milliseconds and data rates in the order of 25 Mb/s, while automated driving or certain industrial control applications require latencies in the order of milliseconds and high packet transmission reliability (in the order of 99.999%). Conversely, for environmental monitoring applications such as waste management, an update frequency of one packet/hour is sufficient, with a tolerable delay of 30 min.

A few paradigms are useful to outline the challenges facing IoT and the potential of our approaches to address them.

1) Automated Driving: The goal of this application is to enhance perception of an individual vehicle and, thus, improve safety. A common approach is to set up a cooperative perception system building on the information sharing between vehicles and roadside units (RSUs) [30]. The scope is to widen the visibility of the individual vehicle to prevent that an object unseen by a single vehicle might cause an accident [31], [32]. The signals to be exchanged go from (low data rate) range measurements to (high data rate) high-definition maps generated by sensors mounted on each vehicle. The communication channels between vehicles are highly dynamic and hard to predict, while the information available at each time slot can be outdated. Nevertheless, communication among vehicles and RSUs should be performed in a reliable and timely manner to ensure that emergency operations can take place in the due time.

2) Intelligent Transportation: A typical task in this application is to navigate a set of electric vehicles to their destination by collecting along the way data about traffic, state of the battery, and availability of parking slots. The objective here is to minimize fuel consumption and the time needed to reach a certain destination while ensuring that all vehicles find a proper refueling station for their batteries along the way. The remaining time to destination is updated online, depending on the time-varying traffic state, which is generally unpredictable.

B. Heterogeneous Technologies

The IoT ecosystem is composed of various components, whose functionality falls within the following categories [5]: identification, sensing, communication, computation, and services. Identification is crucial to assign a clear identity to each object in the network. The role of sensing elements is to gather data from the real world. Typically, sensors are integrated with single-board computers and Transmission Control Protocol/IP functionalities to create IoT devices such as Arduino or Raspberry PI, which are able to sense and send data to a decision entity. The role of communication is to propagate information from the sensing elements to a decision entity, possibly distributed, and back to actuators. There is a plethora of very heterogeneous communication technologies that are in use in IoT. As a broad classification, we can list: 1) short-range technologies to support machine-to-machine communications, like Bluetooth, IEEE 802.15.4, and ZigBee; 2) long-range networks, like LoRa supporting data rates of around 50 kb/s over ranges up to 15 km, or SigFox using ultranarrowband technologies to support ultralow power consumption and long ranges (up to 30–50 km in rural areas and 3–10 km in urban areas), at the expenses of limited data rates; 3) low-power Wi-Fi, also called IEEE 802.11ah supporting data rates up to 347 Mb/s; and

4) cellular networks. Current 4G cellular technologies, more specifically the third-generation partnership project long-term evolution (3GPP LTE), represent the state of the art in mobile communications. However, LTE has been primarily designed for broadband communications and, thus, not optimized for the machine-type communications (MTC) envisioned in IoT.

To partially overcome this discrepancy, 3GPP has introduced some modifications to the standards to enable the deployment of massive smart connected devices and services such as those in smart cities or smart grids: eMTC [33] and NB-IoT [20]. However, a truly disruptive framework enabling an effective deployment of IoT is 5G communication networks, thanks to increased data rate, reduced end-to-end (E2E) latency, and improved coverage relative to 4G [34]. The key features of 5G that are particularly suitable for IoT are: 1) the integration of heterogeneous access technologies; 2) virtualization of network functionalities; and 3) bringing cloud functionalities close to the end user by introducing mobile edge computing (MEC). While earlier network generations have been designed as general-purpose connectivity platforms, the vision underlying 5G is to create an ecosystem for technical and business innovations involving vertical markets such as automotive, energy, agriculture, city management, healthcare, manufacturing, and transportation. Since these services have very different requirements and constraints, the key challenge of 5G is to design a single platform being able to serve different purposes in an efficient way. The solution to tackle such a challenging question is network slicing. At the basis of network slicing, there is network function virtualization, which makes it possible to partition a single physical network into multiple virtual networks, each matched to its specific requirements and constraints. This enables operators to provide networks on an as-a-service basis while meeting a wide range of use cases in parallel. Virtualization is going to play a key role also in IoT to cope with high heterogeneity of requirements as well as the capabilities of devices. However, being able to meet the stringent latency requirements of IoT applications, virtualization needs to be coupled with a new architectural vision enabled by MEC.

C. Embedding IoT in the Edge Cloud

Even within the sophisticated architecture of 5G networks, meeting the stringent latency constraints required in some IoT applications over a wide area network can be still challenging, if not impossible. To guarantee low latencies, a popular solution is to bring cloud functionalities close to the end users through mobile (or multiaccess) edge computing [35], [36].

With MEC, computation and storage resources are brought at the edge of the network, represented by the network access points (APs). In this way, delay-sensitive applications launched by a mobile device can be offloaded to the nearest mobile edge host (MEH), and the most popular contents can also be cached in MEHs to minimize downloading time [37], [38]. Bringing computation and storage resources at the edge of the network makes it possible to guarantee low and stable delays. In practice, the applications launched by the user are executed by virtual machines running on nearby edge nodes, either cloudlets, exploiting a Wi-Fi connection [39], [40], or MEHs, using cellular communication technologies [16]. The further extension of MEC is fog computing, where the edge of the network can include devices as well, thus creating a continuum of devices able to sense, communicate, and compute [14], [41]. A critical aspect in this scenario is mobility management [42]. To handle mobility while offering seamless service continuity, it is necessary to migrate virtual machines quickly across MEH. This is a critical step because instantiating a conventional virtual machine can take times well beyond the latencies required in some IoT applications.

In MEC or fog computing settings, communication, computation, and storage resources can be seen as three aspects of a single system. From a user-centric perspective, what actually matters is the time needed to launch an application and receive the result back. The overall delay depends on communication time, computation time, and the distribution of contents across the network. This holistic vision calls for a joint dynamic optimization of communication, computation, and caching resources [15]. An application where communication and computation resources are closely mingled is computation offloading. This is a fundamental mechanism to enable simple devices to run sophisticated applications or to allow battery-powered devices to run their applications remotely to save energy and, thus, prolong battery lifetime. Computation offloading has gained growing popularity recently. For single-user MEC, it has been studied in [19] and [43]–[45]. The multiuser case was addressed in [46] and [47] and later extended to the dynamic case, using stochastic optimization in [48] and [49]. See recent surveys [15], [16], [50] and references therein.

D. Taming Heterogeneity via a Unified Formulation

With various applications and technologies in mind, the goal of this section is to put forth a unified model for IoT tasks that will guide subsequent algorithmic development.

1) Unifying Models: Consider discrete time $t \in \mathbb{N}$, and a time horizon of T slots. Per slot t, an IoT state variable $\mathbf{s}_t \in \mathbb{R}^p$ is defined, which characterizes all the critical parameters of the IoT environment. Assuming certain amount of knowledge about the environment, the IoT operator will make a decision $\mathbf{x}_t \in \mathbb{R}^d$, aiming to optimize task-specific performance, subject to different types of constraints. The decision $\mathbf{x}_t$ can, in turn, drive the next state $\mathbf{s}_{t+1}$. To model such decision-making processes, we consider a generic problem (1), shown at the top of the next page. The model

$$\begin{aligned}
&\underset{\{\mathbf{x}_t,\forall t\}}{\text{optimize}} && \sum_{t=1}^{T} f(\mathbf{x}_t;\mathbf{s}_t) && \texttt{/IoT performance metrics} && \text{(1a)}\\
&\text{s.t.} && \sum_{t=1}^{T} \mathbf{g}(\mathbf{x}_t;\mathbf{s}_t) \le \mathbf{0} && \texttt{/IoT long-term requirements} && \text{(1b)}\\
& && \mathbf{x}_t \in \mathcal{X}(\mathbf{s}_t),\ \forall t && \texttt{/IoT short-term requirements} && \text{(1c)}\\
& && \texttt{Per slot } t\texttt{, IoT state dynamics } \mathbf{s}_{t+1} = \mathbf{d}(\mathbf{s}_t,\mathbf{x}_t,\boldsymbol{\xi}_t) && && \text{(1d)}\\
& && \texttt{Per slot } t\texttt{, find } \mathbf{x}_t \texttt{ given information oracle } \mathcal{O}_t && && \text{(1e)}
\end{aligned}$$

here is general. The slot duration can vary from tens of microseconds in wireless networks, a few milliseconds in automated driving, and tens of seconds in intelligent transportation to minutes or even hours in smart power networks; the state $\mathbf{s}_t$ can represent the channel gain in wireless networks, the congestion level in data networks as well as transportation networks, and the renewable generation, and energy prices in power networks; and the decision $\mathbf{x}_t$ can include the transmitted power in communication, the size of data workloads, the number of vehicles, or the amount of energy. Regarding the objectives, constraints, and dynamics in (1), we will highlight their IoT relevance, especially of interest to communication and networking communities.

2) Performance Metrics: Given the state $\mathbf{s}_t$ and the decision $\mathbf{x}_t$, we consider the IoT performance as a generic time-invariant function $f(\mathbf{x}_t;\mathbf{s}_t)$ (use $f_t(\mathbf{x}_t)$ interchangeably) depending on the time-varying quantities $\mathbf{s}_t$ and $\mathbf{x}_t$. For MEC problem in Section II-C, $f(\mathbf{x}_t;\mathbf{s}_t)$ often represents the power consumption aggregating over all devices, the aggregated delay, or the system throughput [16], [18], [29]. Another line of recent research studies a new performance metric in MEC—age of information or age—which measures the timeliness of system status using the elapsed time since the most recently received packet was generated at its source [51]. Age of information is pertinent to mission-critical IoT applications [52], [53]. Furthermore, for traffic assignment tasks in intelligent transportation, $f_t(\mathbf{x}_t)$ can capture the overall fuel consumption and the travel time of vehicles on the road [54]; for demand response in smart grids, it is related to user utility and power balancing cost depending on the real-time energy prices [55]–[58]; and for applications related to wireless communications, throughput or achievable rate also plays a critical role in the objective.

3) Short-Term Constraints: The heterogeneous requirements in IoT are modeled via short-term and long-term constraints in (1). The short-term constraints are imposed to regulate $\mathbf{x}_t$ in accordance to short-term requirements, which can be collected in a compact set $\mathcal{X}(\mathbf{s}_t)$—that is either continuous or discrete and possibly depends on the IoT state $\mathbf{s}_t$. As an example, consider an MEC system composed of APs, MEC servers, and mobile user equipment (UE). To meet the stringent latency requirement, the E2E latency of each UE should be less than $\bar{l}$, that is

$$l_t^{\mathrm{tx}} + l_t^{\mathrm{bk}} + l_t^{\mathrm{exe}} + l_t^{\mathrm{rx}} \le \bar{l} \tag{2}$$

where 1) l_t^{tx} is the time spent to send the program state and input (encoded with b_t bits) from UE to AP, e.g., $l_t^{\mathrm{tx}} = b_t/r_t$, with r_t being the data rate (in bits/s); 2) l_t^{bk} is the backhaul latency between AP and MEC server, which appears when the computations are performed in a server that is not colocated with the AP; 3) l_t^{exe} is the server execution time defined as $l_t^{\mathrm{exe}} = c_t/u_t$, where c_t is the number of CPU cycles to be executed, and u_t is the number of CPU cycles/second allocated by the MEC server to UE; and 4) l_t^{rx} is the time for the MEC server to send back the result to UE. With $\mathbf{x}_t := \{u_t, r_t\}$ and $\mathbf{s}_t := \{l_t^{\mathrm{bk}}, l_t^{\mathrm{rx}}, c_t, b_t\}$ thus (2) included in $\mathcal{X}(\mathbf{s}_t)$, selecting $\mathbf{x}_t \in \mathcal{X}(\mathbf{s}_t)$ guarantees the E2E latency requirement in MEC. Short-term constraints also arise due to the physical limits of transmission lines and generators in power networks [8], transceivers in wireless communication [59], as well as vehicles in transportation networks [54].

4) Long-Term Constraints: In some IoT applications, the short-term constraints cannot accurately characterize the demand and requirements. For the latency requirement in MEC, the short-term constraint (2) makes implicit assumptions as follows: 1) no new task is generated before the old tasks are completed and 2) every single task is carried out within an established time frame. These assumptions may be restrictive in some cases. Also, consider a vehicle in the intelligent transportation application that must arrive at its destination within a certain interval. To guarantee on-time arrival, its long-term average speed instead of the instantaneous speed needs to be lower bounded. The long-term constraints are, thus, well-motivated to allow flexible adaptation of $\mathbf{x}_t$ to temporal variations of service requirements. Given the state $\mathbf{s}_t$ and the decision $\mathbf{x}_t$, they are modeled as a set of penalty functions $\mathbf{g}(\mathbf{x}_t;\mathbf{s}_t) := [g^1(\mathbf{x}_t;\mathbf{s}_t),\ldots,g^N(\mathbf{x}_t;\mathbf{s}_t)]^\top$ in (1b). Ideally, we want the accumulated penalty over the entire horizon below a certain threshold. For convenience, we let the threshold to be $\mathbf{0}$ in (1b), which is without loss of generality subject to a constant shift. Long-term constraints

Table 1 Overview of Heterogeneous IoT Settings Considered

Section	State dynamics $\mathbf{s}_t \to \mathbf{s}_{t+1}$	Information oracle $\mathcal{O}_t$
Section III-A	i.i.d. or Markovian	$f, \mathbf{g}, \mathcal{X}$ and $\{\mathbf{s}_1 \ldots, \mathbf{s}_t\}$
Section III-B	Partially controlled Markovian	$f, \mathbf{g}, \mathcal{X}$ and $\{\mathbf{s}_1 \ldots, \mathbf{s}_t\}$
Section III-C	Controlled Markovian	$f_1(\mathbf{x}_1), \mathbf{g}_1(\mathbf{x}_1), \ldots, f_{t-1}(\mathbf{x}_{t-1}), \mathbf{g}_{t-1}(\mathbf{x}_{t-1})$ and $\{\mathbf{s}_1 \ldots, \mathbf{s}_t\}$
Section IV-A	Generally nonstationary	$f_1, \mathbf{g}_1, \ldots, f_{t-1}, \mathbf{g}_{t-1}, \mathcal{X}$
Section IV-B	Generally nonstationary	$f_1(\mathbf{x}_1), \mathbf{g}_1, \ldots, f_{t-1}(\mathbf{x}_{t-1}), \mathbf{g}_{t-1}, \mathcal{X}$
Section IV-C	Generally nonstationary	$f_1(\mathbf{x}_1), \mathbf{g}_1(\mathbf{x}_1), \ldots, f_{t-1}(\mathbf{x}_{t-1}), \mathbf{g}_{t-1}(\mathbf{x}_{t-1}), \mathcal{X}$

also appear in wireless networks where often the average transmit power and link capacity are confined [59]. The challenge in dealing with long-term constraints is that the future states $\mathbf{s}_{t+1}, \cdots, \mathbf{s}_T$ are not known at slot t, which calls for adaptive optimization.

5) State Dynamics: One of the key challenges in IoT is its unpredictable dynamics. In (1), IoT dynamics are encoded by a state transition function $\mathbf{d}$ which generates the next state $\mathbf{s}_{t+1} = \mathbf{d}(\mathbf{s}_t, \mathbf{x}_t, \boldsymbol{\xi}_t)$ given $\mathbf{s}_t$ and $\mathbf{x}_t$ as well as an exogenous variable $\boldsymbol{\xi}_t$. In most of the cases, the exogenous variable $\boldsymbol{\xi}_t$ can be a random disturbance. For wireless communication applications where the state $\mathbf{s}_t$ represents the fading channel state, then $\mathbf{s}_{t+1}$ often does not depend on $\mathbf{s}_t$ and $\mathbf{x}_t$; that is, $\mathbf{s}_{t+1} = \mathbf{d}(\boldsymbol{\xi}_t) := \bar{\mathbf{s}} + \boldsymbol{\xi}_t$, where $\bar{\mathbf{s}}$ is the mean channel state, and $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_T$ are independent, identically distributed (i.i.d.) zero-mean random variables (see [59] and [60]). Markovian dynamics are also common in modeling energy prices, renewable generation processes [61], in which case $\mathbf{s}_{t+1} = \mathbf{d}(\mathbf{s}_t, \boldsymbol{\xi}_t)$ depends on the current state $\mathbf{s}_t$ and an i.i.d. noise $\boldsymbol{\xi}_t$ but not $\mathbf{x}_t$. We refer to both $\mathbf{s}_{t+1} = \mathbf{d}(\boldsymbol{\xi}_t)$ and $\mathbf{s}_{t+1} = \mathbf{d}(\mathbf{s}_t, \boldsymbol{\xi}_t)$, as noninteractive dynamics. The decision $\mathbf{x}_t$ can also play an important role in state transitions. Taking MEC as an example, a queueing model is usually incorporated to keep trace of the relevant quantities such as the amount of remaining tasks that need to be offloaded or processed. With $\{b_t, c_t, r_t, u_t\}$ defined in the following (2), we consider a transmission queue q_t^{tx} that quantifies the number of bits to be transmitted at slot t from UE, and a computation queue q_t^{exe} that quantifies the amount of computation that needs to be completed for UE. If Δt denotes the slot duration, the transmission queue evolves as

$$q_{t+1}^{\text{tx}} = \max\left[q_t^{\text{tx}} - r_t \Delta t, 0\right] + b_t \tag{3}$$

and the computation queue evolves as follows $q_{t+1}^{\text{exe}} = \max\left[q_t^{\text{exe}} - u_t \Delta t, 0\right] + c_t$. In this case, the IoT state is $\mathbf{s}_t := \{q_t^{\text{tx}}, q_t^{\text{exe}}\}$, the decision is $\mathbf{x}_t := \{u_t, r_t\}$, and the exogenous variable is $\boldsymbol{\xi}_t := \{b_t, c_t\}$. It then follows that $\mathbf{s}_{t+1} = \mathbf{d}(\mathbf{s}_t, \mathbf{x}_t, \boldsymbol{\xi}_t)$—what we term interactive dynamics, or more precisely, controlled Markovian dynamics if $\boldsymbol{\xi}_t$ is i.i.d. If the communication and computation resources are sufficient, an ideal policy should guarantee the queue stability [62], [63]. According to Little's law [64], the average execution delay experienced by each UE is proportional to the average queue lengths. Hence, a meaningful problem can be minimizing the average power, subject to the average delay constraints, which will be discussed in Section III-B. State variables of this type also include the location of a vehicle in the intelligent transportation or an unmanned aerial vehicle (UAV) that depends on their previous location and the current movement [65]–[67], and the energy level of a battery that depends on their instantaneous (dis)charging amounts. More complex dynamics are also possible in IoT due to, e.g., strategic human interactions and malicious attack [68]. In those cases, $\boldsymbol{\xi}_t$ can be a function of all the states $\mathbf{s}_1, \ldots, \mathbf{s}_t$ or even completely arbitrary.

6) Accessible Information: While various objectives, constraints, and state dynamics have been adopted to model heterogeneous problems in IoT, the level of accessible information directly affects how to solve the resultant problem given limited communication and computation resources—the epicenter of scalability barriers in IoT. Let the information oracle $\mathcal{O}_t$ collect all the information available to the IoT operator before making decision $\mathbf{x}_t$. For cases where the objectives and the constraints are easy-to-measure formulas (e.g., aggregated power, throughput, and distance), we consider $\mathcal{O}_t := \{f, \mathbf{g}, \mathcal{X}, \mathbf{s}_1 \ldots, \mathbf{s}_t\}$ that includes the explicit form of functions $\{f, \mathbf{g}\}$, set $\mathcal{X}$ and one-slot-ahead prediction $\mathbf{s}_t$. In some IoT settings, however, 1) the objective capturing user-centric quantities, e.g., service latency or reliability, security risk, and customer ratings, is hard to model; 2) the objective involving fast-varying quantities is hard to predict, e.g., the millimeter-wave links in 5G are prone to blocking events, thus hard to predict; and 3) even if modeling and predicting are possible in theory, the low-power smart devices may not afford the complexity of running statistical learning tools "on the fly." In such cases, we consider a fully causal information oracle $\mathcal{O}_t := \{f_1(\mathbf{x}_1), \mathbf{g}_1(\mathbf{x}_1), \ldots, f_{t-1}(\mathbf{x}_{t-1}), \mathbf{g}_{t-1}(\mathbf{x}_{t-1})\}$ that includes only the observed objective function "values" and constraint "penalties" at previous slots. IoT scenarios between these two extreme cases will also be discussed.

In Table 1, we summarize the heterogeneous settings that one may encounter in IoT. Targeting these settings, a set of suitable solvers will be discussed in the subsequent sections. While the methodologies presented in this paper mainly focus on stochastic optimization and online learning, approaches based on other methodologies such as game theory and robust optimization can be also applied to solve similar problems.

III. ADAPTIVITY TO DYNAMIC IoT ENVIRONMENTS

This section introduces methods for optimizing IoT performance under the (asymptotically) stationary assumption on IoT dynamics relative to control decisions in the fast timescale. Corresponding to different types of state dynamics in Table 1, we outline three classes of management schemes (see Fig. 2).

A. Leveraging Statistical Learning for IoT Management

As the generic problem (1), consider the IoT operator makes a per-slot decision $\mathbf{x}_t$, subject to the short-term constraints that are collected in a compact set $\mathcal{X}(\mathbf{s}_t)$ parameterized by the IoT state $\mathbf{s}_t \in \mathcal{S}$, as well as the long-term constraints that are expressed as a time-varying penalty function $\mathbf{g}(\mathbf{x}_t;\mathbf{s}_t) \in \mathbb{R}^N$. With the IoT cost $f(\mathbf{x}_t;\mathbf{s}_t)$, we wish to find a sequence of decisions $\{\mathbf{x}_t\}$ that minimize the expected limiting-average cost subject to the long-term and short-term constraints, that is

$$f^* := \min_{\{\mathbf{x}_t \in \mathcal{X}(\mathbf{s}_t),\, \forall t\}} \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[f(\mathbf{x}_t;\mathbf{s}_t)\right] \tag{4a}$$

$$\text{s.t.} \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\mathbf{g}(\mathbf{x}_t;\mathbf{s}_t)\right] \le \mathbf{0} \tag{4b}$$

where $\mathbb{E}$ is taken over the random state $\mathbf{s}_t$, and possible randomness we may opt to introduce in the decision $\mathbf{x}_t$. Comparing with (1), the infinite time horizon and the limiting average cost are used in (4) and throughout this section for mathematical simplicity. Indeed, assuming $\mathbf{s}_t$ is i.i.d. or generally stationary, the dynamic problem (4) shares the same optimal objective value as the following static problem [63]:

$$\boxed{\begin{aligned} f^* := \min_{\{\boldsymbol{\pi}(\mathbf{s}_t)\in\mathcal{X}(\mathbf{s}_t),\, \mathbf{s}_t\}} \; & \mathbb{E}\left[f(\boldsymbol{\pi}(\mathbf{s}_t);\mathbf{s}_t)\right] && \text{(5a)} \\ \text{s.t.} \; & \mathbb{E}\left[\mathbf{g}(\boldsymbol{\pi}(\mathbf{s}_t);\mathbf{s}_t)\right] \le \mathbf{0}. && \text{(5b)} \end{aligned}}$$

To this end, our goal is to determine a possibly randomized policy $\boldsymbol{\pi}$ that, given an IoT state $\mathbf{s}_t$, generates $\mathbf{x}_t = \boldsymbol{\pi}(\mathbf{s}_t)$ so as to minimize the average cost subject to both long- and short-term constraints in (5). The infinite-dimension functional optimization problem (5) is more tractable in its dual form, which entails a finite number of variables [24], [69]. With $\boldsymbol{\lambda} \in \mathbb{R}_+^N$ denoting the multipliers, the Lagrangian of (5) is $\mathcal{L}(\boldsymbol{\pi},\boldsymbol{\lambda}) := \mathbb{E}\big[\mathcal{L}(\boldsymbol{\pi}(\mathbf{s}_t),\boldsymbol{\lambda};\mathbf{s}_t)\big]$ where the instantaneous (per state) Lagrangian is $\mathcal{L}(\boldsymbol{\pi}(\mathbf{s}_t),\boldsymbol{\lambda};\mathbf{s}_t) := f(\boldsymbol{\pi}(\mathbf{s}_t);\mathbf{s}_t) + \boldsymbol{\lambda}^\top \mathbf{g}(\boldsymbol{\pi}(\mathbf{s}_t);\mathbf{s}_t)$. Correspondingly, the dual problem of (5) is

$$\max_{\boldsymbol{\lambda}\ge\mathbf{0}} \; \mathcal{D}(\boldsymbol{\lambda}) := \mathbb{E}\left[\mathcal{D}(\boldsymbol{\lambda};\mathbf{s}_t)\right] \tag{6}$$

where $\mathcal{D}(\boldsymbol{\lambda};\mathbf{s}_t) := \min_{\mathbf{x}\in\mathcal{X}(\mathbf{s}_t)} \mathcal{L}(\mathbf{x},\boldsymbol{\lambda};\mathbf{s}_t)$. With the optimal $\boldsymbol{\lambda}^*$ obtained for the dual problem (6), the optimal policy for the problem (5) could be retrieved as

$$\boldsymbol{\pi}^*(\mathbf{s}_t) := \arg\min_{\mathbf{x}\in\mathcal{X}(\mathbf{s}_t)} \mathcal{L}(\mathbf{x},\boldsymbol{\lambda}^*;\mathbf{s}_t). \tag{7}$$

The ensemble problem (6) is difficult to solve since the probability density distribution of $\mathbf{s}_t$ is usually unknown. To find the optimal multipliers $\boldsymbol{\lambda}^*$ in an efficient manner, existing methods mainly rely on the stochastic subgradient-based [stochastic gradient descent (SGD)] methods [59], [60], [63]. However, SGD is known to suffer from slow convergence, which implies that the IoT network needs to implement sufficient many suboptimal decisions generated during the transient stage of SGD.

From a different viewpoint, given the huge volume of historical data generated by IoT networks, (6) was first formulated in [24] as a statistical learning task involving both offline training and online operational phases. The rationale is that historical data contain statistics of the IoT states, and learning from them can aid coping with the uncertainty of future management tasks, leading to reduced transient time of adaptive algorithms.

Specifically, with a training set of N_0 historical IoT state samples $\hat{\mathcal{S}}_0 := \{\mathbf{s}_n, 1 \le n \le N_0\}$ available offline, (6) can be recast in an empirical form via sample averaging as

$$\max_{\boldsymbol{\lambda}\ge\mathbf{0}} \hat{\mathcal{D}}_{\hat{\mathcal{S}}_0}(\boldsymbol{\lambda}), \quad \text{with } \hat{\mathcal{D}}_{\hat{\mathcal{S}}_0}(\boldsymbol{\lambda}) := \frac{1}{N_0}\sum_{n=1}^{N_0} \hat{\mathcal{D}}_n(\boldsymbol{\lambda}) - \frac{\epsilon}{2}\|\boldsymbol{\lambda}\|^2 \tag{8}$$

where $\hat{\mathcal{D}}_n(\boldsymbol{\lambda}) := \mathcal{D}(\boldsymbol{\lambda};\mathbf{s}_n)$, and $\epsilon > 0$ is a regularization constant typically used in statistical learning to boost

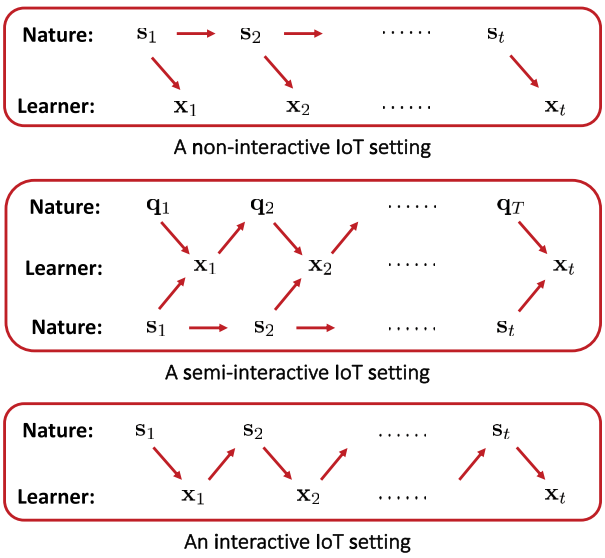


Fig. 2. *Three levels of interaction between IoT operator (learner) and nature corresponding to the three assumptions in Table 1.*

generalization capability [22]. Note that while an ℓ_2-norm regularizer is adopted in (8), other forms of regularization (e.g., ℓ_1 and total-variation norm) are also possible depending on *a priori* knowledge. Note that here t has been replaced by n to differentiate historical data from data in online phases.

Viewing (8) as a (negated) empirical risk minimization (ERM) task, we can resort to the state-of-the-art optimization methods for ERM, e.g., SAGA [70], which enjoys fast convergence and low complexity. Using SAGA, per iteration k, we evaluate a single summand of the empirical gradient, i.e., $\nabla\hat{\mathcal{D}}_{\nu(k)}(\boldsymbol{\lambda}_k)$ at the iterate $\boldsymbol{\lambda}_k$, with sample index $\nu(k) \in \{1,\ldots,N_0\}$ selected uniformly at random. Thus, the computational complexity of SAGA is that of an SGD iteration for (8). Furthermore, SAGA stores a collection of the outdated gradients $\{\nabla_{\text{old}}\hat{\mathcal{D}}_n\}$ for all samples, where $\nabla_{\text{old}}\hat{\mathcal{D}}_n$ was evaluated by $\boldsymbol{\lambda}_{k[n]}$—the most recent iteration $k[n]$ that $\mathbf{s}_n$ was drawn, i.e., $k[n] := \sup\{k' : \nu(k') = n, k' < k\}$. SAGA combines the fresh gradient with the stored ones as

$$\boldsymbol{\lambda}_{k+1} = \left[\boldsymbol{\lambda}_k + \alpha\left(\nabla\hat{\mathcal{D}}_{\nu(k)}(\boldsymbol{\lambda}_k) - \nabla_{\text{old}}\hat{\mathcal{D}}_{\nu(k)} + \nabla_{\text{old}}\hat{\mathcal{D}}_{\hat{\mathcal{S}}_0}\right)\right]^+ \tag{9}$$

where α is the predefined stepsize, and the stored gradients are $\nabla_{\text{old}}\hat{\mathcal{D}}_{\hat{\mathcal{S}}_0} := (1/N)\sum_{n=1}^{N}\nabla_{\text{old}}\hat{\mathcal{D}}_n - \epsilon\boldsymbol{\lambda}_k$.

The merits of SAGA lie in the fact that its gradient estimator in (9) is still unbiased as that with SGD. In addition to the unbiasedness, however, SAGA's gradient estimator attains considerably lower variance than SGD, thanks to the contribution of the stored previous gradients, which is now termed the variance reduction technique prevalent in large-scale machine learning tasks. Needless to mention the encouraging empirical results, the SAGA in (9) is provably convergent to the optimum of (8) with the linear convergence rate [24], [70]

$$\mathbb{E}_\nu\left[\hat{\mathcal{D}}^*_{\hat{\mathcal{S}}_0} - \hat{\mathcal{D}}_{\hat{\mathcal{S}}_0}(\boldsymbol{\lambda}_k)\right] = \mathcal{O}\left(\rho^k\right) \tag{10}$$

where $\hat{\mathcal{D}}^*_{\hat{\mathcal{S}}_0}$ is the optimal objective of (8), and $\rho \in (0,1)$ is the linear rate depending on the objective function of (8).

Hence, in the offline phase, we run KN_0 SAGA iterations (9) on set $\hat{\mathcal{S}}_0$—on average K iterations per sample. In the online phase, initialized with the offline output, SAGA (we term online SAGA) keeps acquiring data $\mathbf{s}_t$ with a growing training set $\hat{\mathcal{S}}_t := \hat{\mathcal{S}}_{t-1} \cup \mathbf{s}_t$. At slot t, online SAGA is initialized with the last iterate of slot $t-1$ and updates $\boldsymbol{\lambda}_t$ by running K iterations (9) (see Fig. 3). The IoT decision is generated using the current $\boldsymbol{\lambda}_t$ by $\boldsymbol{\pi}_t(\mathbf{s}_t) = \arg\min_{\mathbf{x}\in\mathcal{X}(\mathbf{s}_t)}\mathcal{L}(\mathbf{x},\boldsymbol{\lambda}_t;\mathbf{s}_t)$. This is the key idea of offline-aided online IoT operations.

The offline-aided online scheme is not simply heuristic. In fact, the learning performance can be rigorously quantified via several concentration results in the learning theory [22], which uniformly bound the discrepancy between

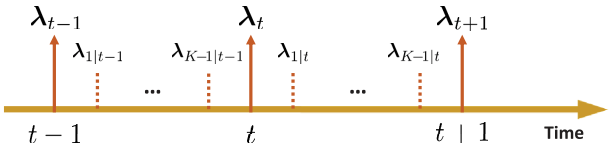


Fig. 3. *Timescale splitting for offline-aided online SAGA operations. Iterates $\{\boldsymbol{\lambda}_t\}$ generate actual IoT decision, while $\boldsymbol{\lambda}_{1|t},\ldots,\boldsymbol{\lambda}_{K|t}$ are K virtual iterates updated via (9) at slot t; and $\boldsymbol{\lambda}_{t+1} := \boldsymbol{\lambda}_{K|t}$.*

the empirical loss (8) and the loss (6) with high probability (whp), that is

$$\sup_{\boldsymbol{\lambda}\geq\mathbf{0}}|\mathcal{D}(\boldsymbol{\lambda}) - \hat{\mathcal{D}}_{\hat{\mathcal{S}}_t}(\boldsymbol{\lambda})| \leq \mathcal{H}_s(N_t),\ \text{whp} \tag{11}$$

where $\mathcal{H}_s(N_t)$ bounds the statistical error induced by the finite size N_t of the training set $\hat{\mathcal{S}}_t$. Under proper (so-termed mixing) conditions, the law of large numbers guarantees that $\mathcal{H}_s(N_t)$ is generally in the order of $\mathcal{O}(\sqrt{1/N_t})$ [22, Sec. 3.4]. On the other hand, let $\mathcal{H}_o(KN_t)$ upper bound the optimization error of solving (8) with $\hat{\mathcal{S}}_t$ due to running on average only finite (K) iterations per sample; i.e., $\hat{\mathcal{D}}^*_{\hat{\mathcal{S}}_t} - \hat{\mathcal{D}}_{\hat{\mathcal{S}}_t}(\boldsymbol{\lambda}_t) \leq \mathcal{H}_o(KN_t)$.

Online SAGA aims at a "sweet spot" between affordable complexity (controlled by K) and desirable overall learning error, which accounts for both the optimization and statistical errors $\mathcal{H}_s(N_t)+\mathcal{H}_o(KN_t)$. Specifically, if we select $N_0 \geq 3\kappa/4$ with κ denoting the condition number of (8), and $K \geq 6$, the optimization error is bounded by $\mathcal{H}_o(KN_t) \leq \mathcal{H}_s(N_t)$ [24]. In fact, even with $K = 1$, online SAGA can still guarantee that $\mathcal{H}_o(KN_t) = \mathcal{O}(\mathcal{H}_s(N_t))$. With the link between the optimal policy and the optimal multiplier (7) in mind, the key message here is that with sufficient historical samples, online SAGA only requires running a small number of iterations per slot to bring the optimization error close to the statistical accuracy provided by the current training set. Recent works along this line also include [71] and [72] that focused on algorithms for piecewise stationary environments. Learning more complex policies for noninteractive settings has been also studied by leveraging deep neural networks [73]. Possible future research along this line also includes developing algorithms under the assumption of stationarity in high-order moments, which is also pertinent in practice. Algorithms tailored for fully nonstationary settings will be presented in Section IV.

B. Learn-and-Adapt Approaches in Semi-Interactive Settings

The IoT environment in Section III-A is noninteractive, meaning that the dynamic of $\mathbf{s}_{t+1}$ in (5) does not change according to $\mathbf{x}_t$. The IoT states can be also driven by decisions, which include the job queue length in a data center [74], the lane length in a transportation network [65],

as well as the battery level in a smart grid [56], [75]. This section considers the case where such IoT states appear in the constraints but not in the objectives, which we call the semi-interactive settings.

Consider an IoT network represented as a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with nodes $\mathcal{N} := \{1, \ldots, N\}$ and edges $\mathcal{E} := \{1, \ldots, E\}$. The node-incidence matrix is formed with (n, e) entry $\mathbf{A}_{(n,e)} = 1(-1)$ if link e enters (leaves) node n, and $\mathbf{A}_{(n,e)} = 0$, otherwise. With $\mathbf{c}_t \in \mathbb{R}_+^N$ collecting the exogenous resources of all nodes per slot t, $\mathbf{x}_t \in \mathbb{R}^E$ for the endogenous resources across edges, the aggregate resource is $\mathbf{A}\mathbf{x}_t + \mathbf{c}_t$. Connecting with (5), $\mathbf{c}_t$ is included in the IoT state $\mathbf{s}_t$, and the constraint becomes $\mathbf{g}(\mathbf{x}_t; \mathbf{s}_t) = \mathbf{A}\mathbf{x}_t + \mathbf{c}_t$. With $\mathbf{q}_t$ collecting all buffered resources at slot t, we wish to solve (4) with the additional state dynamics and the long-term constraints as

$$\mathbf{q}_{t+1} = [\mathbf{q}_t + \mathbf{A}\mathbf{x}_t + \mathbf{c}_t]^+, \quad \forall t \tag{12a}$$

$$\lim_{T\to\infty} (1/T)\textstyle\sum_{t=1}^{T} \mathbb{E}\left[\|\mathbf{q}_t\|\right] < \infty. \tag{12b}$$

Due to the extra constraints in (12), the optimal objective of this new problem is at least f^* in (4). Furthermore, the dynamic of the interactive state $\mathbf{q}_t$ (a.k.a. queues) in (12a) also accounts for the transient performance of an adaptive algorithm. To see this, suppose that under $\boldsymbol{\pi}^*$, it holds that $\mathbf{A}\boldsymbol{\pi}^*(\mathbf{s}_t) + \mathbf{c}_t = \mathbf{0}, \forall \mathbf{s}_t$; and consider the convergence path of policy $\boldsymbol{\pi}_t$ induced by $\boldsymbol{\lambda}_t$ as $\boldsymbol{\pi}_1 \to \boldsymbol{\pi}_2 \to \boldsymbol{\pi}_3 = \ldots = \boldsymbol{\pi}_T = \boldsymbol{\pi}^*$, along with $\mathbf{A}\boldsymbol{\pi}_1(\mathbf{s}_1) + \mathbf{c}_1 = \mathbf{10}$ and $\mathbf{A}\boldsymbol{\pi}_2(\mathbf{s}_2) + \mathbf{c}_2 = \mathbf{5}$. In this case, if $\mathbf{q}_1 = \mathbf{0}$, then we have $\mathbf{q}_2 = \mathbf{10}$ and $\mathbf{q}_3 = \ldots = \mathbf{q}_T = \mathbf{15}$.

The simple example entails two variable insights: 1) constraint violations incurred by the suboptimal decisions during the transient stage (e.g., $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2$) accumulate via $\mathbf{q}_t$ and 2) once accumulated in the transient stage, $\mathbf{q}_t$ will not decrease in the steady state (e.g., $\boldsymbol{\pi}_t$, $t \geq 3$). This explains the suboptimal performance tradeoff of SGD for (4) with (12) (see also [23]).

To better control the interactive state $\mathbf{q}_t$, it suffices to:

S1) reduce the transient time of the adaptive algorithm;
S2) diminish $\mathbf{q}_t$ accumulated during the transient stage.

Following these two guidelines, we adopt a procedure in the online phase that we term online learning and adaptation (named LA-SAGA henceforth) (see Fig. 4). Regarding S1), LA-SAGA performs the offline-aided online learning as that in Fig. 3 by growing the training set $\hat{\mathcal{S}}_t$ based on which it better learns $\boldsymbol{\lambda}_t$; and for S2), LA-SAGA superimposes $\boldsymbol{\lambda}_t$ to the instantaneous state (buffered resources) $\mathbf{q}_t$ and comes up with an effective multiplier as

$$\underbrace{\boldsymbol{\theta}_t}_{\text{effective multiplier}} = \underbrace{\boldsymbol{\lambda}_t}_{\text{statistical learning}} + \underbrace{\mu\mathbf{q}_t - \mathbf{b}}_{\text{system interaction}} \tag{13}$$

where μ tunes emphasis to statistical versus interactive state information, and $\mathbf{b}$ is a constant that corrects the possible bias in the steady state—the intuition will become transparent soon. Based on $\boldsymbol{\theta}_t$, the real-time IoT decision $\mathbf{x}_t$ is obtained by minimizing the Lagrangian over $\mathcal{X}(\mathbf{s}_t)$, that is

$$\boldsymbol{\pi}_t(\mathbf{s}_t) = \mathbf{x}_t := \arg\min_{\mathbf{x}\in\mathcal{X}(\mathbf{s}_t)} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}_t; \mathbf{s}_t). \tag{14}$$

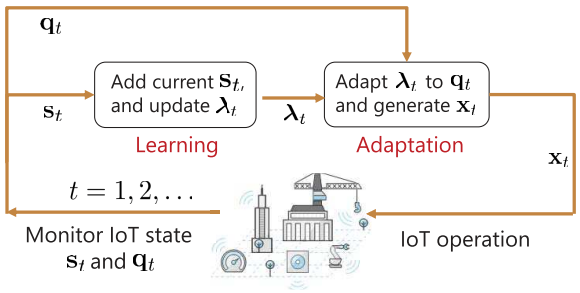


Fig. 4. *Learn-and-adapt diagram for IoT management.*

Note that different from stochastic allocation that is solely based on the system feedback $\boldsymbol{\theta}_t = \mu\mathbf{q}_t$ [63], and statistical learning that only relies on $\boldsymbol{\theta}_t = \boldsymbol{\lambda}_t$, LA-SAGA can take advantage of both through the use of effective multiplier $\boldsymbol{\theta}_t$.

To grasp how the effective multiplier accounts for S2), suppose that after sufficient learning processes, $\boldsymbol{\lambda}_t \approx \boldsymbol{\lambda}^*$, and $\mathbf{q}_t$ is large so that $\mu\mathbf{q}_t - \mathbf{b} \gg \mathbf{0}$. In this case, we have the "shadow price" $\boldsymbol{\theta}_t \gg \boldsymbol{\lambda}^*$, and thus, $\mathbf{x}_t$ obtained through (14) would ensure that $\mathbf{A}\mathbf{x}_t + \mathbf{c}_t < \mathbf{0}$ so that $\mathbf{q}_{t+1} < \mathbf{q}_t$ via (12a). Intuitively speaking, $\boldsymbol{\theta}_t$ will eventually oscillate around $\boldsymbol{\lambda}^*$, and thus, $\mathbf{q}_t$ will oscillate around $\mathbf{b}/\mu$—this also suggests a positive $\mathbf{b}$, otherwise it leads to a biased $\boldsymbol{\theta}_t \geq \boldsymbol{\lambda}^*$ since $\mathbf{q}_t \geq \mathbf{0}$.

Rigorous analysis demonstrates that through a proper selection of the bias $\boldsymbol{b} = \tilde{\mathcal{O}}(\sqrt{\mu})$, $\boldsymbol{\theta}_t$ will converge to the $\mathcal{O}(\sqrt{\mu})$-neighborhood of $\boldsymbol{\lambda}^*$ for (6); formally, we have [24]

$$\lim_{T\to\infty} (1/T)\textstyle\sum_{t=1}^{T} \mathbb{E}\left[f\left(\mathbf{x}_t; \mathbf{s}_t\right)\right] - f^* = \mathcal{O}(\mu) \tag{15a}$$

$$\lim_{T\to\infty} (1/T)\textstyle\sum_{t=1}^{T} \mathbb{E}\left[\|\mathbf{q}_t\|\right] = \tilde{\mathcal{O}}\left(1/\sqrt{\mu}\right) \tag{15b}$$

which asserts that LA-SAGA is $\mathcal{O}(\mu)$-optimal with an average queue length $\tilde{\mathcal{O}}(1/\sqrt{\mu})$—an elegant $[\mathcal{O}(\mu), \tilde{\mathcal{O}}(1/\sqrt{\mu})]$ tradeoff. Comparing with the tradeoff $[\mathcal{O}(\mu), \mathcal{O}(1/\mu)]$ under Lyapunov optimization in [63], LA-SAGA [23], [24] improves the performance in terms of constraint violations (queue lengths).

The idea of incorporating learning into network optimization is pioneered in [76]. However, the developed learning mechanism therein suffers from the curse of dimensionality. Targeting large-scale IoT networks, LA-SAGA can tackle settings with continuous $\mathbf{S}$ and $\mathbf{X}$ with possibly infinite elements and still be amenable

to efficient and scalable online operations. The important implication of the learn-and-adapt scheme is that it can perform the optimal IoT management, with reduced resources and improved QoS, namely, reduced queueing delay in data centers [23], faster virtual network function placement [77], and lower congestion in transportation networks, or smaller battery capacity in power grids [57].

As a closing remark of this section, note that while the problem considered in the semiinteractive setting here explicitly contains queueing-type constraints, the semiinteractive settings, in fact, cover a broader class of problems in IoT. For instance, throughput maximization in UAV-enabled wireless networks under trajectory constraints also belongs to the class of semiinteractive IoT settings [66], [78].

C. Reinforcement Learning for Interactive IoT Environments

The IoT environment considered in Section III-B is semiinteractive in the sense that only the dynamic of $\mathbf{q}_{t+1}$ (but not $\mathbf{s}_{t+1}$) changes according to $\mathbf{x}_t$ through (12a). To broaden the scope of the unified framework, this section introduces methods tailored for the fully interactive setups, where the dynamic of IoT state $\mathbf{s}_t$ that can appear both in the objectives and the constraints is driven by the decision $\mathbf{x}_t$. This setting captures the trajectory optimization in UAV-aided mobile communications (see [66], [78]), the dynamic caching with limited storage units (see [79]), and the route planning in intelligent transportation (see [54]).

For simplicity, consider an IoT environment with a finite state space $\mathcal{S}$ and a finite action space $\mathcal{X}$. The interaction between the operator and the IoT environment is uniquely captured by the transition probability of going from the current state $\mathbf{s}$ to the subsequent state $\mathbf{s}'$ under action $\mathbf{x} \in \mathcal{X}(\mathbf{s}) \subseteq \mathcal{X}$, given by $[\mathbf{P}^{\mathbf{x}}]_{\mathbf{s}\mathbf{s}'} := \mathbb{P}(\mathbf{s}_{t+1} = \mathbf{s} | \mathbf{s}_t = \mathbf{s}', \mathbf{x}_t = \mathbf{x})$. Similar to (5), the goal is to determine a possibly randomized policy $\boldsymbol{\pi}$, that given a state $\mathbf{s}_t$, generates $\mathbf{x}_t = \boldsymbol{\pi}(\mathbf{s}_t)$ so as to minimize the total discounted cost,[1] that is

$$\min_{\{\boldsymbol{\pi}(\mathbf{s}_t) \in \mathcal{X}(\mathbf{s}_t),\, \mathbf{s}_t\}} \lim_{T \to \infty} \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} f\left(\boldsymbol{\pi}(\mathbf{s}_t); \mathbf{s}_t\right)\right] \tag{16}$$

where $\gamma \in (0,1)$ is a discounting factor, and $\mathbb{E}$ is taken over the sample path of $\{\mathbf{s}_t\}$, as well as the random policy $\boldsymbol{\pi}$.

For a fixed policy $\boldsymbol{\pi}$, the state value function is defined as

$$V_{\boldsymbol{\pi}}(\mathbf{s}) := \lim_{T \to \infty} \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} f\left(\boldsymbol{\pi}(\mathbf{s}_t); \mathbf{s}_t\right) \Big| \mathbf{s}_1 = \mathbf{s}\right] \tag{17}$$

and the state-action value function (so-termed Q-function) is $Q_{\boldsymbol{\pi}}(\mathbf{s}, \mathbf{x}) := f(\mathbf{x}; \mathbf{s}) + \gamma \mathbb{E}_{\mathbf{s}'|\mathbf{s},\mathbf{x}}\left[V_{\boldsymbol{\pi}}(\mathbf{s}')\right]$, where $\mathbb{E}$ is taken over the one-step transition from the current state $\mathbf{s}$ to $\mathbf{s}'$ under action $\mathbf{x}$. With the optimal policy $\boldsymbol{\pi}^*$, we have that[2]

$$\boldsymbol{\pi}^*(\mathbf{s}) := \arg \min_{\mathbf{x} \in \mathcal{X}(\mathbf{s})} Q^*(\mathbf{s}, \mathbf{x}) \tag{18}$$

and $V^*(\mathbf{s}) = Q^*(\mathbf{s}, \boldsymbol{\pi}^*(\mathbf{s}))$. Furthermore, the optimality condition of (16) that is termed the Bellman optimality equation can be written as (see [81])

$$Q^*(\mathbf{s}, \mathbf{x}) = f(\mathbf{x}; \mathbf{s}) + \gamma \mathbb{E}_{\mathbf{s}'|\mathbf{s},\mathbf{x}}\left[\min_{\mathbf{x}' \in \mathcal{X}(\mathbf{s})} Q^*(\mathbf{s}', \mathbf{x}')\right], \quad \forall \mathbf{x}, \mathbf{s} \tag{19}$$

which is a system of nonlinear equations of $\mathbf{Q}^* \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{X}|}$.

Switching the goal from (16) to the fixed point of the Bellman optimality equation (19), a classical yet popular approach is the so-termed Q-learning algorithm [82].

S1) At slot t, select the decision $\mathbf{x}_t$ by

$$\boldsymbol{\pi}_t(\mathbf{s}_t) = \mathbf{x}_t := \begin{cases} \arg \min_{\mathbf{x} \in \mathcal{X}(\mathbf{s}_t)} Q_t(\mathbf{s}_t, \mathbf{x}) & \text{w.p. } 1 - \epsilon_t \\ \text{random } \mathbf{x} \in \mathcal{X}(\mathbf{s}_t) & \text{w.p. } \epsilon_t \end{cases} \tag{20}$$

where $\epsilon_t > 0$ is a predefined exploration constant, and $\mathbf{s}_{t+1}$ is generated according to $\mathbb{P}(\mathbf{s}_{t+1} = \mathbf{s}) = [\mathbf{P}^{\mathbf{x}_t}]_{\mathbf{s}_t \mathbf{s}}$.

S2) Update the state-action value function as

$$Q_{t+1}(\mathbf{s}_t, \mathbf{x}_t) = Q_t(\mathbf{s}_t, \mathbf{x}_t) - \alpha_t \left(f(\mathbf{x}_t; \mathbf{s}_t) + \gamma \min_{\mathbf{x} \in \mathcal{X}(\mathbf{s}_{t+1})} Q_t(\mathbf{s}_{t+1}, \mathbf{x})\right) \tag{21}$$

where α_t is a predefined stepsize. Note that different from Sections III-A and III-B, the explicit form of the objective function $f(\cdot; \mathbf{s}_t)$ does not need to be known per slot t. Instead, only the functions values $\{f(\mathbf{x}_\tau; \mathbf{s}_\tau)\}_{\tau=1}^{t}$ along the trajectory $(\mathbf{s}_1, \mathbf{x}_1), \ldots, (\mathbf{s}_t, \mathbf{x}_t)$ are assumed to be known. With properly selected $\{\epsilon_t, \alpha_t\}$, the simple Q-learning algorithm is provably convergent under the finite state and action spaces (a.k.a. tabular case) [81]. To date, the convergence of Q-learning and its variants are mostly asserted for the tabular case.

To scale up Q-learning in the large-scale settings, recent efforts have been devoted to infer $\mathbf{Q}$ by minimizing the residual of the Bellman optimality equation (19), that is

$$\min_{\mathbf{Q}} \sum_{\mathbf{x},\mathbf{s}} \left(Q(\mathbf{s}, \mathbf{x}) - f(\mathbf{x}; \mathbf{s}) - \gamma \mathbb{E}_{\mathbf{s}'|\mathbf{s},\mathbf{x}}\left[\min_{\mathbf{x}' \in \mathcal{X}(\mathbf{s})} Q(\mathbf{s}', \mathbf{x}')\right]\right)^2. \tag{22}$$

[1] For simplicity, the infinite horizon discounted formulation is considered—a slight mismatch with the generic one (1). Other formulations with constraints or average costs can be also considered with additional assumptions [80].

[2] We interchangeably use $Q^*(\mathbf{s}, \mathbf{x}) = Q_{\boldsymbol{\pi}^*}(\mathbf{s}, \mathbf{x})$ and $V^*(\mathbf{s}) = V_{\boldsymbol{\pi}^*}(\mathbf{s})$.

Albeit its simple expression, several fundamental challenges arise when solving this fitting problem (22).

C1) The optimization scale can be prohibitively huge due to the possibly large state and action spaces.

C2) The unknown conditional expectation $\mathbb{E}_{\mathbf{s}'|\mathbf{s},\mathbf{x}}$ inside the square loss prevents an easy unbiased gradient estimator.

C3) The max operator inside the square loss introduces nonsmoothness and nonconvexity when performing optimization.

To tackle C1), function approximation methods have been studied using linear or nonlinear (random) basis functions [83], [84]. Roughly speaking, given a state-action pair $(\mathbf{x}, \mathbf{s})$ along with its predefined feature vector $\boldsymbol{\phi}_{\mathbf{x},\mathbf{s}} \in \mathbb{R}^d$, existing approaches will approximate the Q-function by $Q(\mathbf{s}, \mathbf{x}) := \mathbf{z}^\top(\boldsymbol{\phi}_{\mathbf{x},\mathbf{s}})\boldsymbol{\theta}$, where $\mathbf{z}(\boldsymbol{\phi}_{\mathbf{x},\mathbf{s}}) \in \mathbb{R}^{2D}$ is a lifted feature vector (e.g., random features or outputs of deep neural networks) generated from $\boldsymbol{\phi}_{\mathbf{x},\mathbf{s}}$ and $\boldsymbol{\theta} \in \mathbb{R}^{2D}$ is the wanted parameter vector. To this end, the task of finding the $|\mathcal{S}| \times |\mathcal{A}|$ function (matrix) $\mathbf{Q}$ reduces to find the 2-D vector $\boldsymbol{\theta}$. Along this line, several recent works based on primal-dual solvers have made significant progress on simultaneously resolving C1) and C2) [85], [86]. Regarding C3), while it is still an active research area, approaches leveraging smoothing techniques for nonsmooth functions in convex optimization have shed light on promising remedies [87], [88].

In addition to value iteration-based methods such as Q-learning, approaches based on direct policy search such as policy gradients and actor-critic methods are also prevalent nowadays (see [89]–[91]). This key idea behind policy gradient is to update the $\boldsymbol{\theta}$-parametrized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ using the gradient of the discounted objective (16) with respect to the policy parameters [89]. Convergence of the policy gradient with deep neural networks or kernel-based function approximators is now better understood than Q-learning, along with the limitations of policy gradient-based methods that arise from their high variance.

We conclude this section by remarking that approaches in light of the offline-aided-online learning have also been studied for (16) under the name of experience replay, which achieves tremendous success in various artificial intelligence tasks [92].

IV. SCALABILITY IN ONLINE LEARNING FOR IoT

The IoT settings considered in Section III involve slow-varying IoT dynamics that are (asymptotically) stationary relative to the timescale of making decisions. However, in large-scale IoT, real-time control and communications entail slow and fast time scales that prompt scalable online solvers for generally nonstationary settings—the topics of this section.

In addition to the general nonstationarity, special attention will be given to approaches designed under limited information about the environment, or equivalently, solvers requiring limited computation, and communication resources to sense the environment. Corresponding to different information that may be available in IoT, we outline three classes of scalable online learning approaches (see Fig. 5 for a comparison).

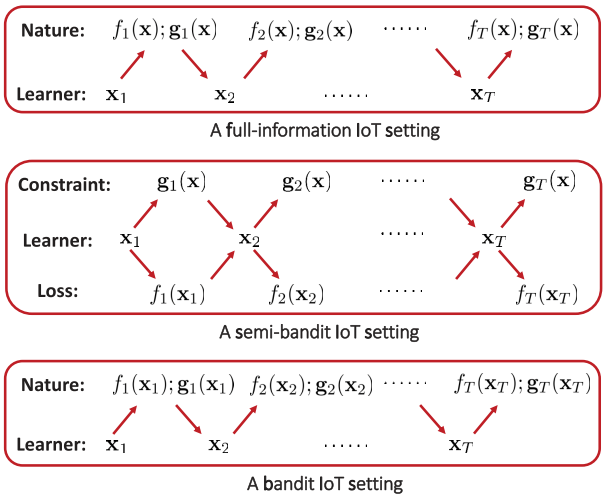


Fig. 5. *Three forms of feedback in IoT environments (termed nature) correspond to three different types of information oracle in Table 1.*

A. Constrained Online Learning for IoT Management

Consider a finite-time horizon T. Per slot t, the IoT operator selects an action $\mathbf{x}_t$ from a known and fixed convex set $\mathcal{X} \subseteq \mathbb{R}^d$, and the IoT environment (a.k.a. nature in OCO) then reveals a loss $f_t : \mathbb{R}^d \to \mathbb{R}$, along with a time-varying (possibly adversarial) penalty function $\mathbf{g}_t : \mathbb{R}^d \to \mathbb{R}^N$. The latter leads to a time-varying constraint $\mathbf{g}_t(\mathbf{x}) \leq \mathbf{0}$, which is driven by the unknown IoT dynamics. As in (1), the goal here is to generate a sequence of decisions that minimize the aggregate loss and ensure that the constraints are satisfied in the long term on average. Specifically, we wish to solve

$$\min_{\{\mathbf{x}_t \in \mathcal{X}, \forall t\}} \sum_{t=1}^{T} f_t(\mathbf{x}_t) \quad \text{s.t.} \sum_{t=1}^{T} \mathbf{g}_t(\mathbf{x}_t) \leq \mathbf{0}. \tag{23}$$

Comparing with the generic problem (1), we keep the time-varying IoT state $\mathbf{s}_t$ implicit in (23), e.g., $f_t(\mathbf{x}_t) := f(\mathbf{x}_t; \mathbf{s}_t)$ and $\mathbf{g}_t(\mathbf{x}_t) := \mathbf{g}(\mathbf{x}_t; \mathbf{s}_t)$, since the algorithms introduced in this section may not need to directly sense the state $\mathbf{s}_t$. For (23), if $\{f_t, \mathbf{g}_t\}$ are known and T is not prohibitively large, the optimal decisions can be found using any off-the-shelf batch solver. Along with the potentially high complexity of batch solvers, a key challenge is that loss and constraint functions in dynamic IoT setups are often unknown before allocating resources, due to unpredictable channel blocking, in millimeter-wave links, due to the

unpredictable routing, network congestion, device malfunctions, and malicious attacks nowadays.

Consider an edge layer with low-power sensors, a fog with N nodes in $\mathcal{N}$, and a cloud with multiple computing centers [14]. Per slot t, each node n collects data requests d_t^n from nearby sensors and has to decide among three options:

1) offloading an amount χ_t^n (from d_t^n) to the cloud;
2) offloading x_t^{nm} to node m for collaborative computing;
3) processing an amount x_t^{nn} using the *in situ* fog servers.

Variable $\mathbf{x}_t$ consists of all the decisions in 1)–3) (see Fig. 6).

Supposing that each fog node has a local queue to buffer unserved workloads, a long-term constraint is imposed to ensure that the cumulative amount of served workloads is no less than the arrived amount over T slots, that is

$$\sum_{t=1}^{T} g_t^n(\mathbf{x}_t) \le 0, \quad \forall n$$
$$g_t^n(\mathbf{x}_t) := d_t^n + \sum_{m\in\mathcal{N}_n^{\text{in}}} x_t^{mn} - \sum_{m\in\mathcal{N}_n^{\text{out}}} x_t^{nm} - \chi_t^n - x_t^{nn} \tag{24}$$

where $\mathcal{N}_n^{\text{in}}$ ($\mathcal{N}_n^{\text{out}}$) is the set of fog nodes with in-coming (out-going) links to (from) node n. Clearly, amounts χ_t^n, x_t^{nm}, and x_t^{nn} have caps depending on the communication protocols and computing cores in use. With $\bar{\mathbf{x}}$ collecting all these caps, the feasible set is $\mathcal{X} := \{\mathbf{0} \le \mathbf{x}_t \le \bar{\mathbf{x}}\}$.

Among candidate figures of merit in optimizing $\mathbf{x}_t$ is network delay of the online edge processing and offloading decisions [15], [93]. Specifically, the latency associated with χ_t^n is mainly due to the communication delay, which can be modeled as a time-varying convex function $l_t^n(\chi_t^n)$. Likewise, the communication delay related to x_t^{nm} is denoted by $l_t^{nm}(x_t^{nm})$. In addition, latency pertaining to x_t^{nn} comes from its limited computation capability, which can be modeled as a function $h_t^n(x_t^{nn})$ capturing dynamics during the computing processes.

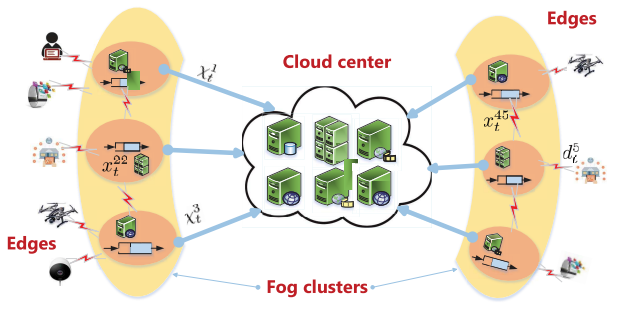


Fig. 6. ***Diagram for mobile computation offloading: IoT devices at the edge layer, fog clusters contain locally connected fog nodes, and the data center in the cloud layer.***

The overall performance in allocating $\mathbf{x}_t$ is quantified by aggregate latency metrics. Those include computational (l_t) and communication delays (h_t), namely

$$f_t(\mathbf{x}_t) := \sum_{n\in\mathcal{N}} \Big(l_t^n(\chi_t^n) + \textstyle\sum_{m\in\mathcal{N}_n^{\text{out}}} l_t^{nm}(x_t^{nm}) + h_t^n(x_t^{nn}) \Big). \tag{25}$$

While the aggregate delay in some cases cannot directly reflect user experience, a viable alternative is the maximum of computational and communication delays (see [93]). While the average-delay objective presumed in (23) may not be the optimal performance metric in some mission critical applications, our formulation can also cover the probabilistic delay requirements. The per-slot objective of the latter is an indicator function of the delay given by

$$f_t(\mathbf{x}_t) := \sum_{n\in\mathcal{N}} \mathbb{1}\left\{ \Big(l_t^n(\chi_t^n) + \sum_{m\in\mathcal{N}_n^{\text{out}}} l_t^{nm}(x_t^{nm}) + h_t^n(x_t^{nn}) \Big) \le \bar{l} \right\} \tag{26}$$

where $\bar{l}$ is a predefined upper bound of user delay. The price paid is that the resultant problem is non-convex, which can be tackled by, e.g., the approach in Section IV-C.

With $f_t(\mathbf{x}_t)$ as in (25) and constraints as in (24), the solution of (23) aims to minimize the aggregate delay while serving all IoT demands in the long term. Looking forward, more intriguing is to find such an optimal strategy in a fully causal setting, where $\{f_t(\mathbf{x}_t), d_t^n\}$ are unknown when deciding $\mathbf{x}_t$ but are revealed at the end of slot t after deciding $\mathbf{x}_t$.

To gauge the performance of online decisions, static regret is adopted by OCO to measure how far the aggregate loss of an OCO algorithm is from the best fixed solution in hindsight [25]. Since a static regret relies on a rather coarse benchmark, which is less useful in dynamic IoT [28], we are motivated to pursue the so-termed dynamic regret given by

$$\text{Reg}_T^{\text{d}} := \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}_t^*) \tag{27a}$$
$$\text{with } \mathbf{x}_t^* \in \arg\min_{\mathbf{x}\in\mathcal{X}} f_t(\mathbf{x}), \quad \text{s.t. } \mathbf{g}_t(\mathbf{x}) \le \mathbf{0} \tag{27b}$$

where the benchmark is now formed using the best sequence $\{\mathbf{x}_t^*\}$ for the instantaneous problem, subject to the instantaneous constraint. The metric in (5) is more suitable for assessing the performance of dynamic IoT networks than its static counterpart in [25] because a sublinear dynamic regret implies a sublinear static one, but the converse is not true.

Regarding feasibility of online decisions, the dynamic fit is also useful to quantify the accumulated

violations, that is

$$\text{Fit}_T^{\text{d}} := \left\| \left[\sum_{t=1}^{T} \mathbf{g}_t(\mathbf{x}_t) \right]^+ \right\|. \tag{28}$$

The long-term constraint implicitly assumes that the instantaneous constraint violations can be compensated by subsequent strictly feasible decisions, thus allowing adaptation of fog decisions to the unknown dynamics of IoT user demands.

Under the metrics in (27) and (28), an ideal algorithm will be one that achieves both sublinear dynamic regret and sublinear dynamic fit. A sublinear dynamic regret implies "no regret" relative to the clairvoyant dynamic solution on the long-term average, i.e., $\lim_{T\to\infty} \text{Reg}_T^{\text{d}}/T = 0$, while a sublinear dynamic fit indicates that the online strategy is also feasible on average, i.e., $\lim_{T\to\infty} \text{Fit}_T^{\text{d}}/T = 0$.

With $\boldsymbol{\lambda} \in \mathbb{R}_+^N$ denoting the Lagrange multiplier vector, the Lagrangian of (23) is

$$\mathcal{L}_t(\mathbf{x}, \boldsymbol{\lambda}) := f_t(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{g}_t(\mathbf{x}). \tag{29}$$

Building on (29), an online scheme termed modified saddle point (MOSP) approach has been developed first in [28] and later in [94]. We use the low-complexity variant in [94] for the subsequent illustration. Given $\mathbf{x}_t$ and $\boldsymbol{\lambda}_t$, the decision $\mathbf{x}_{t+1}$ is

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}} \left(\mathbf{x}_t - \alpha \nabla_{\mathbf{x}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) \right) \tag{30}$$

where $\mathcal{P}_{\mathcal{X}}(\mathbf{y}) := \arg\min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x} - \mathbf{y}\|^2$, α is a predefined constant, and $\nabla_{\mathbf{x}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) = \nabla f_t(\mathbf{x}_t) + \nabla^\top \mathbf{g}_t(\mathbf{x}_t) \boldsymbol{\lambda}_t$ is the gradient of $\mathcal{L}_t(\mathbf{x}, \boldsymbol{\lambda}_t)$ with respect to $\mathbf{x}$. In addition, the dual update takes the modified online gradient ascent form

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t + \mu (\mathbf{g}_t(\mathbf{x}_t) + \nabla^\top \mathbf{g}_t(\mathbf{x}_t)(\mathbf{x}_{t+1} - \mathbf{x}_t)) \right]^+ \tag{31}$$

where μ is the stepsize, and $\mathbf{g}_t(\mathbf{x}_t)$ is the gradient of $\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda})$ with respect to $\boldsymbol{\lambda}$. Note that (31) is a modified gradient update since the dual variable is updated along the first-order approximation of $\mathbf{g}_t(\mathbf{x}_{t+1})$ at $\mathbf{x}_t$ rather than the commonly used $\mathbf{g}_t(\mathbf{x}_t)$.

With properly chosen stepsizes, MOSP enjoys dynamic regret and fit bounded by [28]

$$\text{Reg}_T^{\text{d}} = \mathcal{O}\left(\mathbb{V}(\mathbf{x}_{1:T}^*) T^{\frac{1}{2}} \right) \quad \text{and} \quad \text{Fit}_T^{\text{d}} = \mathcal{O}\left(T^{\frac{1}{2}} \right) \tag{32}$$

where $\mathbb{V}(\mathbf{x}_{1:T}^*)$ is the accumulated variation of the per-slot minimizers $\mathbf{x}_t^*$ in (27) given by $\mathbb{V}(\mathbf{x}_{1:T}^*) := \sum_{t=1}^T \|\mathbf{x}_t^* - \mathbf{x}_{t-1}^*\|$. In words, MOSP's dynamic fit is sublinear, and its dynamic regret is also sublinear, so long as the variation of the minimizers is slow enough, i.e., $\mathbb{V}(\mathbf{x}_{1:T}^*) = \mathbf{o}(\sqrt{T})$.

Relevant approaches developed in similar settings also include those in [95]–[97]. Specifically, OCO with switching cost has been studied in [95], and feedback-based tracking algorithms have been developed in [96] and [97].

Remark 1 Learning via Task-Adaptive Stepsizes: The primal update (30) can be refined by adjusting each entry of the gradient using a per-entry stepsize in accordance with "each thing" in IoT applications [93]. Such an adaptive stepsize can be regarded as an inexpensive approximation of the Hessian used in the online Newton iteration [98]. Using edge computing as a paradigm, [93] showed that task-adaptive stepsizes can markedly reduce the network delay when the underlying IoT tasks are heterogeneous, where the resultant gradients could have distinct orders of magnitude over different coordinates.

B. Constrained Convex Bandit Learning for IoT Management

The online recursions (30) and (31) remain operational under the premise that the loss functions are known or their gradients are readily available. Clearly, none of these assumptions is always satisfied in IoT, because: 1) the loss function capturing user dissatisfaction, e.g., service latency or reliability, can be hard to model in dynamic settings and 2) even if modeling is possible, the low-power devices may not afford the complexity of running statistical learning tools such as deep neural networks online. These considerations motivate online bandit saddle point (BanSP) methods to broaden the scope of MOSP to IoT settings where the gradient is unavailable or computationally costly [29].

The key idea behind bandit learning is to construct (preferably stochastic) gradient estimates using limited function value information [99], [100]. Consider first a learner only observing the value of $f_t(\mathbf{x})$ at a single point $\mathbf{x}$ per slot t. The crux is to construct a (possibly unbiased) estimate of the gradient using this single piece of feedback—what is interestingly possible by one random function evaluation [99]. The intuition is easy to grasp in the 1-D case: for a binary variable u taking values $\{-1, 1\}$ equiprobably, and a small $\delta > 0$, the difference approximation of the derivative f_t' at x yields

$$f_t'(x) \approx \frac{f_t(x+\delta) - f_t(x-\delta)}{2\delta} = \mathbb{E}_u \left[\frac{u}{\delta} f_t(x + \delta u) \right] \tag{33}$$

where the equality follows from the definition of expectation. Dropping $\mathbb{E}_u$, the scaled single-value evaluation $f_t(x+\delta u)u/\delta$ is a nearly unbiased estimator of $f_t'(x)$. Generalizing this approximation to higher dimensions, with a random vector $\mathbf{u}$ drawn from the surface of a unit sphere, the scaled function evaluation at a perturbed point $\mathbf{x} + \delta\mathbf{u}$ yields an estimate of the gradient $\nabla f_t(\mathbf{x})$, given by [99]

$$\nabla f_t(\mathbf{x}) \approx \mathbb{E}_{\mathbf{u}} \left[\frac{d}{\delta} f_t(\mathbf{x} + \delta\mathbf{u}) \mathbf{u} \right] := \mathbb{E}_{\mathbf{u}} \left[\hat{\nabla} f_t(\mathbf{x}) \right] \tag{34}$$

where we define one-point gradient $\hat{\nabla} f_t(\mathbf{x}) := (d/\delta) f_t(\mathbf{x}+\delta\mathbf{u})\mathbf{u}$.

Building upon (34), consider the primal update [cf. (30)]

$$\hat{\mathbf{x}}_{t+1} = \mathcal{P}_{(1-\gamma)\mathcal{X}}\left(\hat{\mathbf{x}}_t - \alpha\left(\hat{\nabla} f_t(\hat{\mathbf{x}}_t) + \nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)\boldsymbol{\lambda}_t\right)\right) \quad (35)$$

where $(1-\gamma)\mathcal{X} := \{(1-\gamma)\mathbf{x} : \mathbf{x} \in \mathcal{X}\}$ is a subset of $\mathcal{X}$, and $\gamma \in [0,1)$ is a preselected constant dependent on δ. In the full-information case, $\mathbf{x}_t$ in (30) is the learner's action, whereas in the bandit case the learner's action is $\mathbf{x}_t := \hat{\mathbf{x}}_t + \delta\mathbf{u}_t$, which is the point for function evaluation instead of $\hat{\mathbf{x}}_t$ in (35). Projection in (35) is on a smaller convex set $(1-\gamma)\mathcal{X}$ in (35), which ensures feasibility of the perturbed $\mathbf{x}_t \in \mathcal{X}$. Similar to (31), the dual update of BanSP is given by

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t + \mu(\mathbf{g}_t(\hat{\mathbf{x}}_t) + \nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t))\right]^+ \quad (36)$$

where $\hat{\mathbf{x}}_t$ rather than $\mathbf{x}_t$ is used in this update. Compared with (30)–(31), the updates (35)–(36) with one-point bandit feedback do not increase computation or memory requirements; hence, they provide a lightweight surrogate for MOSP to enable gradient-free online bandit IoT network optimization.

If the mild conditions in [29] are satisfied, the online decisions generated by BanSP yield

$$\mathbb{E}\left[\mathrm{Reg}_T^{\mathrm{d}}\right] = \mathcal{O}\left(\mathbb{V}(\mathbf{x}_{1:T}^*)T^{\frac{3}{4}}\right) \text{ and } \mathbb{E}\left[\mathrm{Fit}_T^{\mathrm{d}}\right] = \mathcal{O}\left(T^{\frac{3}{4}}\right) \quad (37)$$

where $\mathbb{E}$ is taken over the sequence of the random actions $\mathbf{x}_t$ with randomness induced by $\{\mathbf{u}_t\}$ perturbations.

Depending on the underlying dynamics, BanSP can afford one or multiple loss function evaluations (bandit feedback) per slot. If BanSP is endowed with $M > 2$ function evaluations, the gradient estimate will be more accurate by querying the function values over M points in the neighborhood of $\hat{\mathbf{x}}_t$. Intuitively, the performance of BanSP will improve if multiple evaluations are available per slot. Indeed, the dynamic regret is provably $\mathcal{O}\left(\mathbb{V}(\mathbf{x}_{1:T}^*)T^{(1/2)}\right)$, and the dynamic fit $\mathrm{Fit}_T^{\mathrm{d}} = \mathcal{O}\left(T^{(1/2)}\right)$ [29], which markedly improve upon their single-point counterparts, and reduce to MOSP bounds in the full-information case [cf. (32)].

C. Constrained Multiarmed Bandit Learning

The salient assumption so far is that IoT decisions belong to a time-invariant convex set $\mathcal{X}$. However, IoT devices usually exhibit time-varying connectivity to the backbone due to mobility and cyber attacks, while network configurations are often selected from predetermined protocols. In this context, multiarmed bandit (MAB) methods can be employed to extend BanSP when $\mathcal{X}$ is time-varying and discrete [101], [102].

Consider the discrete feasible set $\mathcal{X} := \{\mathbf{x}^1, \ldots, \mathbf{x}^K\}$ with total K possible actions (a.k.a. arms in MAB). To account for dynamics, only the actions in $\mathcal{X}_t \subseteq \mathcal{X}$ are available per slot t, e.g., $\mathbf{x}_t \in \mathcal{X}_t$. The availability of actions could be stochastic, following a certain probability distribution or even adversarial, in which case nature can arbitrarily choose $\mathcal{X}_t$.

Per slot t, collect the objective values of all actions into vector $\mathbf{f}_t := [f_t(\mathbf{x}^1), \ldots, f_t(\mathbf{x}^K)]^\top$, and likewise the constraints into matrix $\mathbf{G}_t := [\mathbf{g}_t(\mathbf{x}^1), \ldots, \mathbf{g}_t(\mathbf{x}^K)] \in \mathbb{R}^{N\times K}$. If the learner's strategy is to select an action $\mathbf{x}_t = \mathbf{x}^k$ with k from a distribution $k \sim \mathbf{p}_t \in \mathbb{R}^K$, then (23) can be reformulated as an optimization problem over distributions $\{\mathbf{p}_t\}$, namely

$$\min_{\{\mathbf{p}_t \in \Delta(\mathcal{X}_t), \forall t\}} \sum_{t=1}^T \mathbf{f}_t^\top \mathbf{p}_t \quad \text{s.t.} \sum_{t=1}^T \mathbf{G}_t \mathbf{p}_t \le \mathbf{0} \quad (38)$$

where the $\mathcal{X}_t$-supported "probability simplex" is defined as

$$\Delta(\mathcal{X}_t) := \left\{\sum_{\mathbf{x}^k \in \mathcal{X}_t} p(\mathbf{x}^k) = 1;\ p(\mathbf{x}^k) \ge 0;\ p(\mathbf{x}^k) = 0, \mathbf{x}^k \notin \mathcal{X}_t\right\}. \quad (39)$$

It is worth mentioning that $f_t(\mathbf{x}^k)$ and $\mathbf{g}_t(\mathbf{x}^k)$ are well defined even when the action $\mathbf{x}^k \notin \mathcal{X}_t$ is not available, and the values $f_t(\mathbf{x}^k)$ and $\mathbf{g}_t(\mathbf{x}^k)$ are not revealed.

In order to employ an MOSP solver for (38), the gradient of the associated Lagrangian is needed, meaning $\mathbf{f}_t$ and $\mathbf{G}_t$ must be known. The challenge is that such information is hardly available in large-scale IoT settings, where one only knows $f_t(\mathbf{x}_t)$ and $\mathbf{g}_t(\mathbf{x}_t)$ given that $\mathbf{x}_t$ is implemented. The time-varying action set also prevents a direct implementation of BanSP to solve (38). To tackle such a challenging setting, a novel EXP3SP algorithm was developed in the recent work [101] that builds on the elegant exponential-weight algorithm for exploration and exploitation (EXP3) [103].

Per slot t, the learner observes the action set $\mathcal{X}_t$ and selects $\mathbf{x}_t$ according to the current distribution $\mathbf{p}_t$ given by

$$p_t(\mathbf{x}^k) = \frac{\tilde{p}_t(\mathbf{x}^k)\mathbb{1}(\mathbf{x}^k \in \mathcal{X}_t)}{\sum_{\mathbf{x}^k \in \mathcal{X}} \tilde{p}_t(\mathbf{x}^k)\mathbb{1}(\mathbf{x}^k \in \mathcal{X}_t)}, \quad \forall \mathbf{x}^k \in \mathcal{X} \quad (40)$$

where $\tilde{p}_t(\mathbf{x}^k)$ is the unnormalized weight of $\mathbf{x}^k$ at slot t, the value of which will be specified later. Once $f_t(\mathbf{x}_t)$ and $\mathbf{g}_t(\mathbf{x}_t)$ become available, unbiased estimates of $\mathbf{f}_t$ and $\mathbf{G}_t$ are

$$\hat{f}_t(\mathbf{x}^k) = \frac{f_t(\mathbf{x}^k)\mathbb{1}(\mathbf{x}_t = \mathbf{x}^k)}{p_t(\mathbf{x}^k)}, \quad \forall \mathbf{x}^k \in \mathcal{X} \quad (41a)$$

$$\hat{\mathbf{g}}_t(\mathbf{x}^k) = \frac{\mathbf{g}_t(\mathbf{x}^k)\mathbb{1}(\mathbf{x}_t = \mathbf{x}^k)}{p_t(\mathbf{x}^k)}, \quad \forall \mathbf{x}^k \in \mathcal{X}. \quad (41b)$$

Adopting the gradient estimators in (41), the primal update uses the exponential gradient recursion,

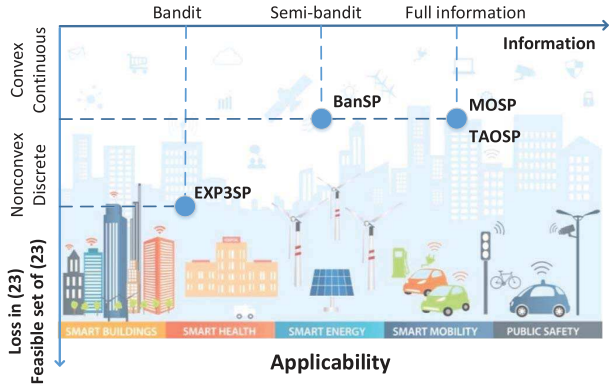


Fig. 7. ***Summary of learning approaches corresponding to Fig. 5.***

namely, $\forall \mathbf{x}^k \in \mathcal{X}$

$$\tilde{p}_{t+1}(\mathbf{x}^k) = \tilde{p}_t(\mathbf{x}^k) \exp\left[-\mu\left(\hat{f}_t(\mathbf{x}^k) + \boldsymbol{\lambda}_t^\top \hat{\mathbf{g}}_t(\mathbf{x}^k)\right)\right]. \quad (42)$$

The weight $\tilde{\mathbf{p}}_{t+1}$ is, in turn, used to generate the action distribution in the next slot [cf. (40)]. The dual update is

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t + \mu\left(\hat{\mathbf{G}}_t \mathbf{p}_t - \delta\mu\boldsymbol{\lambda}_t\right)\right]^+ \quad (43)$$

where δ is a tuned constant to ensure a bounded multiplier.

If $\mathcal{X}_t$ is stochastic, EXP3SP achieves both sublinear regret and fit [101]. A robust modification of EXP3SP has been also developed recently to cope with adversaries blocking access of IoT devices to their edge servers [102] while further securing edge computing and ensuring sublinear regret and fit.

Our scalable online learning schemes are recapped in Fig. 7.

V. LESSONS LEARNED AND THE ROAD AHEAD

We have presented a unified framework for deriving and analyzing adaptive and scalable network design and resource allocation schemes for IoT. Leveraging the contemporary communication, networking, and optimization advances, the resultant online learning and management policies not only facilitate low-complexity and scalable implementations with limited feedback but also enjoy efficient adaptation to changing environments with analytical performance guarantees.

The proposed framework lays a solid analytical foundation to delineate the tradeoffs among performance guarantees, degree of (non)stationarity in modeling IoT dynamics, algorithm scalability, and levels of accessible information (see Fig. 8).

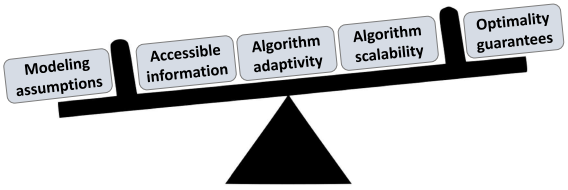


Fig. 8. ***Tradeoff among modeling assumptions, accessible information, algorithm adaptivity, scalability, and optimality guarantees.***

A. Modeling Assumptions Versus Optimality Guarantees

While both deal with IoT management with unknown dynamics, the modeling assumptions in Sections III and IV differ considerably. Specifically, those in Section III assume a generally stationary IoT environment that corresponds to either the simplest i.i.d. case or to the Markovian case eventually converging to a stationary distribution. In contrast, the approaches in Section IV can afford arbitrary dynamics even those manipulated by adversaries. However, such minimal assumption does not come for free. As a matter of fact, the performance guarantee in terms of the sublinear regret in Section IV is weaker than the optimality gap in Section III—see an analytical comparison in [28]. Nevertheless, as the effectiveness of the optimal solution in Section III also depends on the discrepancy between the real IoT settings and the modeled stationary ones, the actual online performance of these approaches requires further evaluation.

The vantage point of this overview opens up a number of exciting directions for future research.

B. Distributed Machine Learning

Considering the massive amount of mobile devices in IoT, centralized learning becomes computationally intractable and also rises serious privacy concerns. To date, the widespread consensus is that besides data centers at the cloud, future machine learning and artificial intelligence tasks have to be performed starting from the network edge, namely, mobile devices. This is the overarching goal of the emerging federated learning paradigm [104], [105]. Toward this goal, future challenges and opportunities include reducing the communication overhead during the distributed learning processes and enhancing the robustness of learning algorithms under adversarial attacks. Recent advances in the direction of communication-efficient learning include the adaptive communication mechanism in [106] that enjoys the first provably bound on the reduced number of communication rounds. Challenges of distributed learning also lie in asynchrony and delay introduced by, e.g., IoT mobility and heterogeneity. Asynchronous parallel learning schemes are, thus, worth investigating by leveraging advances in static optimization settings [107], [108]. From distributed

machine learning to distributed control, multiagent reinforcement learning will play a critical role in distributed control for IoT [109]. Decentralized actor-critic algorithms have been recently developed in [91] and [110] for multiagent reinforcement learning, while communication-efficient schemes can be found in [111].

C. Communication, Computation, and Control Codesign

The past decade has witnessed the convergence of the communication and computing processes [16]. The current brief is that next-generation communication networks should support emerging large-scale control applications in IoT with millions of diverse devices over a large geographical area. This calls for codesigning communication, computing, and control mechanisms. The challenges naturally arise in developing the desired network architecture, the role of different network entities, pertinent performance metrics, and the corresponding policies to simultaneously satisfy the timeliness, reliability, and efficiency of all three intertwined systems.

Over the decades, the focus of wireless communications has been anytime, anywhere, anyone connection of the humans, whereas the emerging IoT paradigm largely extends the scope of wireless networking to connecting everything, along the humans-to-things and things-to-things continuum. The IoT challenges such as extreme heterogeneity, unpredictable dynamics, and massive scale call for game-changing innovations in network design and management. We hope that the proposed unified framework can serve as a stepping stone that leads to systematic designs and rigorous analysis of adaptive and scalable learning and management schemes for IoT, and a host of new research venues to pursue. ■

REFERENCES

[1] (Jun. 2015). *White Paper: A New Approach to IoT Security*. [Online]. Available: www.pubnub.com/blog/2015-06-29-white-paper-a-new-approach-to-iot-security/

[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[4] J. A. Stankovic, "Research directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.

[5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[6] T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson, and P. Dutta, "The Internet of Things has a gateway problem," in *Proc. ACM HotMobile*, Santa Fe, NM, USA, Feb. 2015, pp. 27–32.

[7] D. Lymberopoulos, A. Bamis, and A. Savvides, "Extracting spatiotemporal human activity patterns in assisted living using a home sensor network," *Univ. Access Inf. Soc.*, vol. 10, no. 2, pp. 125–138, Jun. 2011.

[8] G. B. Giannakis, V. Kekatos, N. Gatsis, S.-J. Kim, H. Zhu, and B. F. Wollenberg, "Monitoring and optimization for power grids: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 107–128, Sep. 2013.

[9] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014.

[10] S. Munir, J. A. Stankovic, C.-J. M. Liang, and S. Lin, "Cyber physical system challenges for human-in-the-loop control," in *Proc. Feedback Comput.*, San Jose, CA, USA, Jun. 2013, pp. 1–4.

[11] D. S. Nunes, P. Zhang, and J. S. Silva, "A survey on human-in-the-loop applications towards an Internet of all," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 944–965, 2nd Quart., 2015.

[12] L. Duan, L. Huang, C. Langbort, A. Pozdnukhov, J. Walrand, and L. Zhang, "Human-in-the-loop mobile networks: A survey of recent advancements," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 4, pp. 813–831, Apr. 2017.

[13] B. Gerhardt, K. Griffin, and R. Klemann, "Unlocking value in the fragmented world of big data analytics," Cisco Internet Bus. Solutions Group, Tech. Rep., Jun. 2012.

[14] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[15] S. Barbarossa, S. Sardellitti, E. Ceci, and M. Merluzzi. (2018). "The edge cloud: A holistic view of communication, computation and caching." [Online]. Available: https://arxiv.org/abs/1802.00700

[16] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 45–55, Nov. 2014.

[17] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[18] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[19] X. Lyu *et al.*, "Optimal schedule of mobile edge computing for Internet of Things using partial information," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2606–2615, Nov. 2017.

[20] *Narrowband IoT (NB-IoT)*, document RP-151621, 3GPP TSG RAN Meeting 69, Qualcomm, 2015.

[21] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the Internet of Things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[22] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 2000.

[23] T. Chen, Q. Ling, and G. B. Giannakis, "Learn-and-adapt stochastic dual gradients for network resource allocation," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 4, pp. 1941–1951, Dec. 2018.

[24] T. Chen, A. Mokhtari, X. Wang, A. Ribeiro, and G. B. Giannakis, "Stochastic averaging for constrained optimization with application to online resource allocation," *IEEE Trans. Signal Process.*, vol. 65, no. 12, pp. 3078–3093, Jun. 2017.

[25] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. Int. Conf. Mach. Learn.*, Washington, DC, USA, Aug. 2003, pp. 928–936.

[26] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[27] E. Hazan, "Introduction to online convex optimization," *Found. Trends Optim.*, vol. 2, nos. 3–4, pp. 157–325, 2016.

[28] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to proactive network resource allocation," *IEEE Trans. Signal Process.*, vol. 65, no. 24, pp. 6350–6364, Dec. 2017.

[29] T. Chen and G. B. Giannakis, "Bandit convex optimization for scalable and dynamic IoT management," *IEEE Internet Things J.*, to be published.

[30] K. Sakaguchi and R. Fukatsu, "Cooperative perception realized by millimeter-wave V2V for safe automated driving," in *Proc. Asia–Pacific Microw. Conf.*, Kyoto, Japan, Nov. 2018, pp. 180–182.

[31] S.-W. Kim *et al.*, "Multivehicle cooperative driving using cooperative perception: Design and experimental validation," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 663–680, Apr. 2015.

[32] K. Sakaguchi *et al.*, "Where, when, and how mmWave is used in 5G and beyond," *IEICE Trans. Electron.*, vol. 100, no. 10, pp. 790–808, 2017.

[33] *Further LTE Physical Layer Enhancements for MTC*, document RP-141660, 3GPP TSG RAN Meeting 65, Ericsson, Nokia, 2014.

[34] M. R. Palattella *et al.*, "Internet of Things in the 5G era: Enablers, architecture, and business models," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 510–527, Mar. 2016.

[35] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," ETSI, Sophia Antipolis, France, White Paper 11, 2015, pp. 1–16.

[36] B. Liang, *Mobile Edge Computing*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[37] W. Chu, M. Dehghan, D. Towsley, and Z.-L. Zhang, "On allocating cache resources to content providers," in *Proc. ACM Conf. Inf.-Centric Netw.*, Kyoto, Japan, Sep. 2016, pp. 154–159.

[38] W. Chu, M. Dehghan, J. C. S. Lui, D. Towsley, and Z.-L. Zhang, "Joint cache resource allocation and request routing for in-network caching services," *Comput. Netw.*, vol. 131, pp. 1–14, Feb. 2018.

[39] E. Cuervo *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. ACM MobiSys*, San Francisco, CA, USA, Mar. 2010, pp. 49–62.

[40] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 945–953.

[41] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.

[42] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2637–2646, Nov. 2017.

[43] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 1451–1455.

[44] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.

[45] Y. Mao, J. Zhang, Z. Chen, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[46] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[47] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[48] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[49] S. Sardellitti, M. Merluzzi, and S. Barbarossa, "Optimal association of mobile users to multi-access edge computing resources," in *Proc. IEEE Int. Conf. Commun.*, Kansas, MO, USA, May 2018, pp. 1–6.

[50] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. (Jan. 2017). "Mobile edge computing: Survey and research outlook." [Online]. Available: https://arxiv.org/abs/1701.01090

[51] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 2731–2735.

[52] M. Costa, M. Codreanu, and A. Ephremides, "On the age of information in status update systems with packet management," *IEEE Trans. Inf. Theory*, vol. 62, no. 4, pp. 1897–1910, Apr. 2016.

[53] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7492–7508, Nov. 2017.

[54] J. Zhang, S. Pourazarm, C. G. Cassandras, and I. C. Paschalidis, "The price of anarchy in transportation networks: Data-driven evaluation and reduction strategies," *Proc. IEEE*, vol. 106, no. 4, pp. 538–553, Apr. 2018.

[55] N. Li, L. Chen, and S. H. Low, "Optimal demand response based on utility maximization in power networks," in *Proc. IEEE Power Energy Soc. General Meeting*, Detroit, MI, USA, Jul. 2011, pp. 1–8.

[56] W. Shi, N. Li, C.-C. Chu, and R. Gadh, "Real-time energy management in microgrids," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 228–238, Jan. 2017.

[57] B. Li, T. Chen, X. Wang, and G. B. Giannakis, "Real-time energy management in microgrids with reduced battery capacity requirements," *IEEE Trans. Smart Grids*, to be published.

[58] X. Wang, Y. Zhang, T. Chen, and G. B. Giannakis, "Dynamic energy management for smart-grid-powered coordinated multipoint systems," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1348–1359, May 2016.

[59] X. Wang, G. B. Giannakis, and A. G. Marques, "A unified approach to QoS-guaranteed scheduling for channel-adaptive wireless networks," *Proc. IEEE*, vol. 95, no. 12, pp. 2410–2431, Dec. 2007.

[60] N. Gatsis, A. Ribeiro, and G. B. Giannakis, "A class of convergent algorithms for resource allocation in wireless fading networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 5, pp. 1808–1823, May 2010.

[61] A. M. González, A. M. S. Roque, and J. García-González, "Modeling and forecasting electricity prices with input/output hidden Markov models," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 13–24, Feb. 2005.

[62] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[63] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures Commun. Netw.*, vol. 3, no. 1, pp. 1–211, 2010.

[64] S. M. Ross, *Introduction to Probability Models*. New York, NY, USA: Academic, 2014.

[65] J. Gregoire, X. Qian, E. Frazzoli, A. D. L. Fortelle, and T. Wongpiromsarn, "Capacity-aware backpressure traffic signal control," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 2, pp. 164–173, Jun. 2015.

[66] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

[67] J. Xu, Y. Zeng, and R. Zhang, "UAV-enabled wireless power transfer: Trajectory design and energy optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5092–5106, Aug. 2018.

[68] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proc. IEEE*, vol. 104, no. 9, pp. 1727–1765, Sep. 2016.

[69] A. Ribeiro, "Ergodic stochastic optimization algorithms for wireless communication and networking," *IEEE Trans. Signal Process.*, vol. 58, no. 12, pp. 6369–6386, Dec. 2010.

[70] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 1646–1654.

[71] M. Eisen, K. Gatsis, G. J. Pappas, and A. Ribeiro. (Mar. 2018). "Learning in wireless control systems over non-stationary channels." [Online]. Available: https://arxiv.org/abs/1803.01078

[72] L. Huang, "Receding learning-aided control in stochastic networks," *Perform. Eval.*, vol. 91, pp. 150–169, Sep. 2015.

[73] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos. (May 2017). "Learning to optimize: Training deep neural networks for wireless resource management." [Online]. Available: https://arxiv.org/abs/1705.09412

[74] T. Chen, X. Wang, and G. B. Giannakis, "Cooling-aware energy and workload management in data centers via stochastic optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 2, pp. 402–415, Mar. 2016.

[75] X. Wang, X. Chen, T. Chen, L. Huang, and G. B. Giannakis, "Two-scale stochastic control for integrated multipoint communication systems with renewables," *IEEE Trans. Smart Grids*, vol. 9, no. 3, pp. 1822–1834, May 2018.

[76] L. Huang, X. Liu, and X. Hao, "The power of online learning in stochastic network optimization," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 1, pp. 153–165, Jun. 2014.

[77] X. Chen *et al.* (Apr. 2018). "Multi-timescale online optimization of network function virtualization for service chaining." [Online]. Available: https://arxiv.org/abs/1804.07051

[78] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.

[79] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, "Optimal and scalable caching for 5G using reinforcement learning of space-time popularities," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 180–190, Feb. 2018.

[80] E. Altman, *Constrained Markov Decision Processes*. Boca Raton, FL, USA: Chapman & Hall, 1999.

[81] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[82] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, May 1992.

[83] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. Boca Raton, FL, USA: Taylor & Francis, 2010.

[84] Y. Shen, T. Chen, and G. B. Giannakis, "Online ensemble multi-kernel learning adaptive to non-stationary and adversarial environments," in *Proc. Int. Conf. Artif. Intell. Stat.*, Lanzarote, Canary Islands, Apr. 2018, pp. 1–10.

[85] B. Dai, N. He, Y. Pan, B. Boots, and L. Song, "Learning from conditional distributions via dual embeddings," in *Proc. Int. Conf. Artif. Intell. Stat.*, Fort Lauderdale, FL, USA, Apr. 2017, pp. 1458–1467.

[86] X. Wei, H. Yu, and M. J. Neely, "Online learning in weakly coupled Markov decision processes: A convergence time study," in *Proc. SIGMETRICS*, Irvine, CA, USA, Jun. 2018, pp. 1–37.

[87] B. Dai *et al.*, "Smoothed dual embedding control," in *Proc. NIPS Symp. Deep Reinforcement Learn.*, Long Beach, CA, USA, Dec. 2017, pp. 1–28.

[88] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, "Bridging the gap between value and policy based reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 2772–2782.

[89] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, Dec. 2000, pp. 1057–1063.

[90] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.*, Lille, France, Jul. 2015, pp. 1889–1897.

[91] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proc. Int. Conf. Mach. Learn.*, Stockholm, Sweden, 2018, pp. 5872–5881.

[92] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, Feb. 2015.

[93] T. Chen, Y. Shen, Q. Ling, and G. B. Giannakis, "Heterogeneous online learning for 'thing-adaptive' fog computing in IoT," *IEEE Internet Things J.*, to be published.

[94] H. Yu, M. Neely, and X. Wei, "Online convex optimization with stochastic constraints," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 1427–1437.

[95] Y. Li, G. Qu, and N. Li. (Jan. 2018). "Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit." [Online]. Available: https://arxiv.org/abs/1801.07780

[96] A. Bernstein, E. Dall'Anese, and A. Simonetto. (Apr. 2018). "Online optimization with feedback." [Online]. Available: https://arxiv.org/abs/1804.05159

[97] E. Dall'Anese and A. Simonetto, "Optimal power flow pursuit," *IEEE Trans. Smart Grids*, vol. 9, no. 2, pp. 942–952, Mar. 2018.

[98] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.

[99] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proc. ACM SODA*, Vancouver, BC, Canada, Jan. 2005, pp. 385–394.

[100] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Found. Comput. Math.*, vol. 17, no. 2, pp. 527–566, Apr. 2017.

[101] B. Li, T. Chen, X. Wang, and G. B. Giannakis, "Sleeping multi-armed bandit with constraints," in *Proc. AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, Jan. 2019.

[102] B. Li, T. Chen, X. Wang, and G. B. Giannakis, "Secure edge computing in IoT via online learning," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Oct. 2018.

[103] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, 2002.

[104] B. McMahan and D. Ramage. (Apr. 2017). Federated learning: Collaborative machine learning without centralized training data. Google Res.Blog. [Online]. Available: https://research.googleblog.com/2017/04/federated-learning-collaborative.html

[105] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat.*, Fort Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282.

[106] T. Chen, G. B. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2018.

[107] Z. Peng, Y. Xu, M. Yan, and W. Yin, "ARock: An algorithmic framework for asynchronous parallel coordinate updates," *SIAM J. Sci. Comput.*, vol. 38, no. 5, pp. 2851–2879, Sep. 2016.

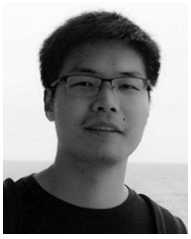
[108] L. Cannelli, F. Facchinei, V. Kungurtsev, and G. Scutari. (Jul. 2016). "Asynchronous parallel algorithms for nonconvex big-data optimization: Model and convergence." [Online]. Available: https://arxiv.org/abs/1607.04818

[109] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Info. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 6379–6390.

[110] K. Zhang, Z. Yang, and T. Basar, "Networked multi-agent reinforcement learning in continuous spaces," in *Proc. IEEE Conf. Decis. Control*, Miami, FL, USA, Dec. 2018, pp. 5872–5881.

[111] T. Chen, K. Zhang, G. B. Giannakis, and T. Başar. (Dec. 2018). "Communication-efficient distributed reinforcement learning." [Online]. Available: https://arxiv.org/abs/1812.03239

ABOUT THE AUTHORS



Tianyi Chen (Student Member, IEEE) received the B. Eng. degree (highest honors) in communication science and engineering from Fudan University, Shanghai, China, in 2014, and the M.Sc. degree in electrical and computer engineering (ECE) from the University of Minnesota (UMN), Minneapolis, MN, USA, in 2016, where he is currently working toward the Ph.D. degree.

His current research interests include large-scale optimization and machine learning with applications to Internet of Things (IoT).

Mr. Chen was a recipient of the Best Student Paper Award (finalist) of the Asilomar Conference on Signals, Systems, and Computers, the Best Demo Award in the International Conference on IoT, the National Scholarship from China in 2013, the UMN ECE Department Fellowship in 2014, and the UMN Doctoral Dissertation Fellowship in 2017.

Sergio Barbarossa (Fellow, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from the Sapienza University of Rome, Rome, Italy, 1984 and 1988, respectively.

He has held several visiting positions at the Environmental Research Institute of Michigan, Ann Arbor, MI, USA, in 1988, the University of Virginia, Charlottesville, VA, USA, in 1995 and 1997, and the University of Minnesota, Minneapolis, MN, USA, in 1999. He has been a Scientific Coordinator of EU projects on wireless sensor networks, small cell networks, and distributed mobile cloud computing. He is currently managing the EU/Japan project 5G-MiEdge merging millimeter wave and edge cloud technology for 5G networks. He is currently a Full Professor with the Sapienza University of Rome. His current research interests include 5G networks, mobile edge computing, machine learning, and signal processing over graphs.

Dr. Barbarossa is a Fellow of the European Association for Signal Processing (EURASIP). He was a recipient of the 2010 Technical Achievements Award from the EURASIP Society for his contributions on radar, communication, and networks and the IEEE Best Paper Awards from the IEEE Signal Processing Society in 2000 and 2014. He served as an IEEE Distinguished Lecturer.

Xin Wang (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Fudan University, Shanghai, China, in 1997 and 2000, respectively, and the Ph.D. degree in electrical engineering from Auburn University, Auburn, AL, USA, in 2004.

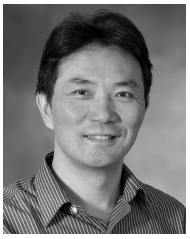
From 2004 to 2006, he was a Post-doctoral Research Associate with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA. In 2006, he joined the Department of Electrical Engineering, Florida Atlantic University, Boca Raton, FL, USA, as an Assistant Professor, and promoted as a Tenured Associate Professor in 2010. He is currently a Distinguished Professor and the Chair of the Department of Communication Science and Engineering, Fudan University. His current research interests include stochastic network optimization, energy-efficient communications, cross-layer design, and signal processing for communications.

Dr. Wang served as an Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and as an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS. He currently serves as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He is an IEEE Distinguished Lecturer for the Vehicular Technology Society.

Georgios B. Giannakis (Fellow, IEEE) received the Diploma degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1981, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1983 and 1986, respectively.

From 1987 to 1998, he was a Faculty Member with the University of Virginia, Charlottesville, VA, USA. Since 1999, he has been a Professor with the University of Minnesota, Minneapolis, MN, USA, where he holds an ADC Endowed Chair, a University of Minnesota McKnight Presidential Chair in electrical and computer engineering, and serves as the Director of the Digital Technology Center. His general research interests include the areas of statistical learning, communications, and networking subjects on which he has authored or co-authored more than 450 journal papers, 750 conference papers, 25 book chapters, two edited books, and two research monographs (h-index 142). He has (co-)invented 32 issued patents. His current research interests include data science and network science with applications to the Internet of Things, social, brain, and power networks with renewables.

Dr. Giannakis is a fellow of the European Association for Signal Processing (EURASIP). He was a (co-)recipient of nine best journal paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications, the Technical Achievement Awards from the SP Society in 2000 from EURASIP in 2005, a Young Faculty Teaching Award, the G. W. Taylor Award for Distinguished Research from the University of Minnesota, and the IEEE Fourier Technical Field Award (inaugural recipient in 2015). He has served a Distinguished Lecturer for the IEEE-SPS.



Zhi-Li Zhang (Fellow, IEEE) received the B.S. degree in computer science from Nanjing University, Nanjing, China, in 1986, and the M.S. and Ph.D. degrees in computer science from the University of Massachusetts, Boston, MA, USA, in 1992 and 1997.

In 1997, he joined the Computer Science and Engineering Faculty, University of Minnesota, Minneapolis, MN, USA, where he is currently a Qwest Chair Professor and a Distinguished McKnight University Professor. His current research interests include computer communication and networks, Internet technology, content distribution systems, cloud computing, and emerging Internet of Things applications.

Dr. Zhang is a member of the Association for Computing Machinery (ACM). He has co-chaired several conferences/workshops (as general or Transmission Control Protocol chairs), including IEEE INFOCOM, IEEE ICNP, ACM CONEXT, and IFIP Networking, and served on the TPC of numerous conferences/workshops. He was a co-recipient of several best paper awards and a number of other awards.