University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

Faculty Publications from the Department of Electrical & Computer Engineering, Department of of

5-16-2018

DECCO: Deep-Learning Enabled Coverage and Capacity Optimization for Massive MIMO Systems

Yang Yang

Yang Li

Kai Li

Shuang Zhao

Rui Chen

See next page for additional authors

Follow this and additional works at: https://digitalcommons.unl.edu/electricalengineeringfacpub

Part of the Computer Engineering Commons, and the Electrical and Computer Engineering Commons

This Article is brought to you for free and open access by the Electrical & Computer Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Faculty Publications from the Department of Electrical and Computer Engineering by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Authors

Yang Yang, Yang Li, Kai Li, Shuang Zhao, Rui Chen, Jun Wang, and Song Ci



Received March 4, 2018, accepted April 8, 2018, date of publication April 20, 2018, date of current version May 16, 2018. *Digital Object Identifier* 10.1109/ACCESS.2018.2828859

DECCO: Deep-Learning Enabled Coverage and Capacity Optimization for Massive MIMO Systems



⁴State Key Laboratory of ISN, Xidian University, Xi'an 710071, China

⁵Department of Computer Science, University College London, London WC1E 6BT, U.K.

⁶Department of Electrical and Computer Engineering, University of Nebraska–Lincoln, Lincoln, NE 68588, USA

Corresponding author: Yang Yang (yang.yang@wico.sh)

This work was supported in part by the National Natural Science Foundation of China under Grant 61571004, in part by the National Science and Technology Major Project under Grant 2016ZX03001024, and in part by the Science and Technology Commission of Shanghai Municipality under Grant 16510711600.

ABSTRACT System capacity and service coverage are the most critical performance metrics in cellular wireless communication networks. Usually, system capacity enhancements are at the expense of service coverage degradations, and vice versa. This capacity-coverage tradeoff and the associated joint optimization problem becomes very challenging in massive multiple-input multiple-output (MIMO) wireless systems, due to a large amount of antenna tilt values to be configured and very sophisticated inter-cell interference conditions, under massive antenna scenarios. This paper proposes a novel approach, namely group alignment of user signal strength (GAUSS), to efficiently support the user scheduling for the massive MIMO system, and thus serve as an effective parameter for the coverage and capacity optimization (CCO) problem. Together with a unified threshold of Quality of Service, i.e. the minimum signal-to-interference-plus-noise ratio $(SINR_{min})$ for user satisfaction, GAUSS can effectively control the variance of signal strengths of multiple users in the neighborhood. Moreover, an intelligent and efficient deep-learning enabled coverage and capacity optimization (DECCO) algorithm is proposed and evaluated, which adopts a pre-trained deep policy gradient-based neural network to dynamically derive GAUSS and SINRmin during CCO. Furthermore, an inter-cell interference coordination (ICIC) is proposed to enhance the CCO performance. Analytical and simulation results show that the proposed DECCO algorithm can effectively achieve a much better performance balance between system capacity and service coverage than traditional fixed optimization (FO) and proportional fair optimization (PFO) algorithms. Specifically, DECCO significantly increases the overall spectrum efficiency by 24% and 40%, respectively, than FO and PFO in a typical massive MIMO system.

INDEX TERMS Coverage and capacity optimization, deep reinforcement learning, massive MIMO, user scheduling, inter-cell interference coordination.

I. INTRODUCTION

The multiuser MIMO (MU-MIMO) technology plays an important role in modern wireless communication systems due to its capability of providing significant performance gains over the single-user MIMO (SU-MIMO). With MU-MIMO, a multi-antenna base station (BS) can simultaneously serve multiple user equipments (UEs) within a cell using the same spectrum resource, and thus greatly improving the

overall spectrum efficiency [2]. As a promising technology for the Fifth Generation (5G) wireless communication standard [3]–[5], massive MIMO with large scale antennas further enhances the system performance in terms of spectrum efficiency and energy efficiency. In real operations, we must concurrently balance system capacity and service coverage by appropriately tuning optimization parameters at BSs, which is referred as the Coverage and Capacity Optimization (CCO) problem. Usually, CCO related system parameters include reference signal power, antenna tilt, scheduling parameters, and etc. However, it is very difficult and expensive to configure a lagre number of antenna tilt values for adaptive CCO. We have to study and define more tractable optimization parameters for addressing the CCO in massive MIMO systems.

Specifically, user scheduling mechanisms are responsible for choosing how to allocate spectrum resources for the BSs with fine time and frequency resolutions, taking into account channel condition and QoS requirements. Thus we can seek for tuning the scheduling parameters for the CCO problem instead of antenna tilt. Combes et al. [6] proposed an α -fair scheduler technique to improve coverage performance at the expense of very small capacity losses. In [6], an optimization parameter α was proposed to adaptively adjust the scheduling rules based on whether the number of users is small or not. Comsa et al. [7] proposed a dynamic neural Q-learningbased scheduling technique that achieves a flexible system capacity and user fairness tradeoff, including the O-learning algorithm is used to adopt different policies of scheduling rules at each Transmission Time Interval (TTI). Nevertheless, both [6] and [7] focuse on SU-MIMO that schedules only one user for transmission. In the context of MU-MIMO and massive MIMO, Sun et al. [8] proposes a joint user scheduling and power allocation algorithm for Joint Spatial Division and Multiplexing (JSDM) [9] scheme in massive MIMO downlink systems, which schedules users and allocates power iteratively with the MAX user scheduling and the Lagrange power optimization method. Based on a two-stage precoding framework for the massive MIMO systems, Xu et al. [10] proposed an improved K-means user grouping scheme which allocates the users to different pre-beamforming groups using the second-order channel statistics, and then a user grouping scheme that considers both load balancing and precoding design. After user groups are determined, they presented a dynamic user scheduling scheme where second-stage precoding is designed based on instantaneous channel conditions. However, the researches of [8] and [10] are only to maximize the system sum rate and thus without considering to concurrently optimize the system capacity and coverage.

In this paper, we propose a novel parameter GAUSS to efficiently support the user scheduling for the massive MIMO system, and thus serve as an effective parameter for the CCO problem. Together with *SINR_{min}*, GAUSS can effectively control the variance of signal strengths of scheduled user group. Moreover, the DECCO is proposed to dynamically derive GAUSS and *SINR_{min}* during the process of coverage and capacity optimization. We also propose an inter-cell interference coordination scheme to enhance the CCO performance. The key contributions of this work are summarized as follows.

• A novel scheduling parameter GAUSS, together with a unified threshold of quality of service *SINR_{min}*, is proposed to address the challenging problem of CCO in massive MIMO systems.

- A novel CCO algorithm DECCO is proposed to dynamically derive the optimal combination of GAUSS and *SINR_{min}* with a pre-trained policy gradient neural network in the user scheduling scheme, together with a novel ICIC scheme.
- Analytical and simulation results show that the proposed DECCO algorithm can effectively achieve a much better performance balance between system capacity and service coverage. In particular, compared with traditional Fixed Optimization (FO) and Proportional Fair Optimization (PFO) algorithms, DECCO significantly increases both cell average and cell-edge spectrum efficiency in a typical massive MIMO system.

The rest of this paper is organized as follows. In section II, we briefly introduce the massive MIMO system model and the coverage and capacity optimization problem formulation. In section III, we present the novel optimization parameter GAUSS for the user scheduling scheme. In section IV, we present the DECCO algorithm, including the deep reinforcement learning-based user scheduling algorithm and the inter-cell interference coordination scheme. In section V, we give the simulation results to verify the effectiveness of the proposed parameters, and compare the coverage and capacity performance of our proposed DECCO algorithm with a algorithm using the best fixed configuration of the proposed parameters in the user scheduling scheme. We finally conclude this work in section VI.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we first briefly introduce the system model for multiuser massive MIMO and the user *SINR* estimation model. Then we provide the formulation of the coverage and capacity optimization problem.

A. MASSIVE MIMO SYSTEM MODEL

We consider the downlink transmission of a massive MIMO network as depicted in Figure 1. The shadow area of the cells is regarded as the cell center and the area between the dashed line and the solid line is defined as the cell edge. For the CCO problem of this system, the Key Performance



FIGURE 1. A typical massive MIMO network scenario with inter-cell interference.

Indicators (KPIs) of system capacity and service coverage are thus defined as the Cell Average Spectrum Efficiency (CASE) and the Cell-Edge Spectrum Efficiency (CESE), which are calculated as the average spectrum efficiency of all the users in the cell and the cell-edge users, respectively.

Each BS is equipped with M_t antennas, and can simultaneously serve K user terminals with N_r antennas. We assume that the $M_t \times (K \times N_r)$ dimensional channel matrix **H** is fixed for a certain block length, which is the coherence time of the channel, and changes from block to block. In order to reduce the Channel State Information (CSI) feedback overhead, two-stage precoding scheme is usually adopted for Frequency Division Duplexing (FDD) massive MIMO systems [9], [11], [12]. The inner precoder adopts zeroforcing scheme based on local CSI, and the outer precoder groups the UEs based on the similarity of the eigenspace of auto-correlation matrix of the UEs' downlink channel. The received signals at the user side using the two-stage precoding scheme is then given by

$$\mathbf{y} = \mathbf{H}^H \mathbf{B} \mathbf{P} \mathbf{d} + \mathbf{z},\tag{1}$$

where **y** is the received signal, **d** is transmitted data symbol vector, **z** is the additive Gaussian noise, **B** is outer precoder, **P** is the inner precoder, and **H** is the channel matrix. Assuming each UE's signal is allocated with equal power, the normalized received signal \tilde{y} can be further obtained as

$$\tilde{\mathbf{y}} = \sqrt{\frac{P_t/N}{Tr(\mathbf{B}\mathbf{P}\mathbf{P}^H\mathbf{B}^H)}}\mathbf{H}^H\mathbf{B}\mathbf{P}\mathbf{x} + \mathbf{n},$$
(2)

where P_t is the total transmit power at BS, N is the noise power, **x** and **n** are the normalized signal and Gaussian noise respectively, and $Tr(\cdot)$ denotes the matrix trace.

B. PROBLEM FORMULATION

Considering the interference, the user *SINR* could be estimated as

$$SINR = \frac{P_t/(N+I)}{Tr(\mathbf{BPP}^H\mathbf{B}^H)}.$$
(3)

In downlink of massive MIMO network with MU-MIMO transmission mode, the number of transmit antennas at the BS is much larger than the total number of receive antennas of scheduled UEs in a cell, thus the two-stage precoding scheme using distributed zero-forcing can apparently reduce intercell interference and intra-cell interference.

In each scheduling duration, users' *SINR* can be calculated according to the channel matrix composed by the scheduled users, and then the *SINR* of each user can be estimated accurately assuming we can get ideal CSI. When scheduling, *the minimum user* SINR *threshold*, denoted as *SINR_{min}*, can be used to control the minimum *SINR* value of the user and adjust coverage and capacity performance. The bandwidth of spectrum is denoted as *B*, which is shared by the users when scheduling. Multiple users can be scheduled simultaneously to improve the spatial gain. For the sake of simplicity, we assume the interference is perfectly cancelled through the

two-stage precoding scheme and the inter-cell interference coordination scheme. Thus the users' *SINR* within each cell in the network can be estimated as follows.

$$SINR = \frac{P_t/N}{Tr(\mathbf{BPP}^H \mathbf{B}^H)}.$$
(4)

The instantaneous spectrum efficiency of a cell at certain time step t can be calculated as

$$E(t) = \frac{\sum_{k=1}^{K^*} B \log_2(1+\rho_k)}{B} = \sum_{k=1}^{K^*} \log_2(1+\rho_k), \quad (5)$$

where *K* is the number of scheduled users, ρ_k is the *k*-th user's *SINR*, defined by Equation 4. Thus CASE can be computed as Cumulative Distribution Function (CDF) of 50% tile of the instantaneous spectrum efficiency, and CESE can be calculated as CDF of 5% tile of the instantaneous spectrum efficiency. A unified KPI of an optimization area, e.g., sector *i*, is defined as

$$KPI_i = wE_{5\%} + (1 - w)E_{50\%},$$
(6)

where E denotes E(t), and w is used as a weight factor to balance coverage and capacity performance. As mentioned in previous section, we utilize user scheduling scheme and interference coordination scheme for the optimization of coverage and capacity. The scheduling results of a sector has little impact on neighboring regions since the interference is perfectly cancelled through the two-stage precoding scheme and the inter-cell interference coordination scheme. In this sense, the scheduling result will result in little effect on the neighboring sectors' KPIs. Thus the coverage and capacity optimization problem can be formulated as a KPI maximization problem for a sector i:

$$\max_{G} KPI_{i} = wE_{5\%} + (1 - w)E_{50\%},$$

s.t. 0 < |G| ≤ K*,
0 ≤ w ≤ 1, (7)

where G is defined as the group of scheduled users, and the number of users in the group in constrained by K^* . Note that this optimization problem is constrained by how to schedule the users. We will offer the discussion in the following section.

III. GROUP ALIGNMENT OF USER SIGNAL STRENGTH

In this section, we introduce our proposed optimization parameter GAUSS, which is used to identify the qualified users to be scheduled.

According to the property of singular value decomposition (SVD), we can get

$$Tr(\mathbf{BPP^{H}B^{H}}) = Tr(\mathbf{u}\lambda\mathbf{v}^{H}\mathbf{vu}^{H})$$

$$= Tr(\lambda^{2}\mathbf{u}^{H}\mathbf{u})$$

$$= Tr(\lambda^{2})$$

$$= \sum_{k=1}^{K^{*}}\lambda_{k}^{2}, \qquad (8)$$

where **BP** = $\mathbf{u}\lambda\mathbf{v}^{H}$ with singular values $\lambda = diag(\lambda_1, \dots, \lambda_k, \dots, \lambda_K^*)$ in descending order. With the scheduling constraints, K^* should be no larger than the rank of **BPP**^H**B**^H. Substituting (8) into (5), with $\mu = \frac{P_i}{N}$, there is

$$E(t) = \sum_{k=1}^{K^*} \log_2(1 + \frac{\mu}{\sum_{i=1}^{K^*} \lambda_i^2}),$$
(9)

where λ_i^2 can be interpreted as the channel gain factor for each user in the MIMO scenario, this value is larger for the cell-edge user, and smaller for the cell-center user. Equation 9 can be rewritten as

$$E(t) = \sum_{k=1}^{K^*} \log_2(1 + \frac{\mu}{\lambda_1^2 \sum_{i=1}^{K^*} \gamma_i^2}),$$
(10)

where $\gamma_i = \frac{\lambda_i}{\lambda_1}$ and $\gamma_1 = 1$, $\gamma_i < 1(i \neq 1)$. Thus we can infer that spectrum efficiecy is mainly determined by a small number of users with larger channel gain factor. If the cell-center users and the cell-edge users are scheduled simultaneously, the throughput of the center users will be lowered by the celledge users. In order to ensure the system capacity will not decline while improving the spatial multiplexing rate, it needs to guarantee the gap between max(λ_i) and min(λ_i) within a certain range. In this sense, we introduce the user scheduling optimization parameter GAUSS to align the users' *SINR* in the scheduled user group, where GAUSS is defined as follows

$$R = \frac{\max(\lambda_i)}{\min(\lambda_i)}.$$
 (11)

If we increase the value of GAUSS, i.e. *R*, the number of scheduled user *K* increases while the *k*-th user's instantaneous *SINR* ρ_k decreases since more UE scheduled. We denote ξ_i as the average channel gain factor of user *i*. Then Equation 10 can be rewritten as follows

$$E(t) = \sum_{i=1}^{K^*} \log_2(1 + \frac{\mu}{\xi_i})$$
(12)

The user's average channel gain factor is sorted in ascending order, combined with group alignment of users' signal strength R, Figure 2 can be obtained. After selecting a user i, with the average channel gain factor of the user ξ_i as the center and the group alignment of users' signal strength R as the radius, the qualified users that can participate the scheduling process can be determined. The user channel condition at the left of ξ_i is superior to user i, the user channel condition at the right of ξ_i is inferior to user i. According to the previous analysis, in a scheduling process we should obtain the user set based on their channel condition, and the target user set is controlled by how we choose R, i.e. the value of GAUSS. Further considering *SINR*_{min}, we can obtain the following inequality

$$\xi_i \le \frac{\mu}{SINR_{\min}} = \beta. \tag{13}$$



FIGURE 2. Group alignment of users' signal strength.

If ξ_i is smaller than β , the scheduling user *i* can satisfy the constraint in (11) and can be scheduled with other users; otherwise, user *i* cannot satisfy the constraint in (11) and cannot be scheduled with other users to avoid lower *SINR*. In this sense, the users on the left side of β can be reused when scheduling, while the users on the right side of β cannot be reused when scheduling. Another observation is that *SINR*_{min} determines the location of β , as *SINR*_{min} decreasing, the point of β moves to the right. Hence the system capacity decreases due to more poor channel condition users are scheduled. However, since the *SINR* of the users located to the right of β is higher, the coverage performance can be improved. When we increase *SINR*_{min}, the system capacity and network coverage change vice versa.

IV. DEEP-LEARNING ENABLED COVERAGE AND CAPACITY OPTIMIZATION

In this section, we present the DECCO algorithm to perform the capacity-coverage optimization by the deep reinforcement learning-based user scheduling scheme and a subsequent inter-cell interference coordination scheme. The overall framework of DECCO is depicted in Figure 3, and the detailed implementation is presented in the following.



FIGURE 3. The framework of DECCO algorithm with a joint ICIC scheme and user scheduling scheme.

A. PRELIMINARIES

We will only introduce the basic concepts of deep reinforcement learning techniques that we build on in this paper. Detailed survey and rigorous derivations can be found in [13].

1) REINFORCEMENT LEARNING

Reinforcement learning is a model-free method to solve a *Markov decision process* (MDP). Typically, an MDP consists of a state space S, an action space A, a stationary transition distribution describing the environment dynamics $p(s_{t+1}|s_t, a_t)$ which satisfies the Markov property, and a reward function r. Figure 4 shows a general setting of reinforcement learning where an *agent* interacts with an *environment* [14]. The agent observes some state s_t and chooses an action a_t at each time step t. Once the action is done, the state of the environment transitions to s_{t+1} and the agent receives reward r_t . Specially, the state transitions and rewards of the environment are stochastic and are assumed to have the Markov property, thus they are only depending on the value of the previous timestep.



FIGURE 4. Reinforcement Learning with DNN as function approximator.

We should notice here that the agent can only control its actions and have no apriori knowledge of the environment. However, the agent can learn to act properly by randomly choosing actions and observing the transitions of the environment. The goal of learning is to maximize the expected cumulative discounted reward: $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$, where γ is a factor discounting future rewards within (0,1]. If we set a very small γ , learning will not depend on future rewards much and immediate rewards are dominating. On the other hand, if it is too large, learning will count on future rewards heavily.

2) POLICY

At each timestep *t*, the agent's decision making procedure is characterized by a *Policy*, $\pi(s, a) = Pr\{a_t = a | s_t = s\}, \forall s \in S, a \in A$, which is the probability distribution that action *a* is taken in state *s*. In practical problems, the state space and action space are normally large, which is intractable to store the policy in tabular form. In this sense, *function approximators* are utilized to parametrize the policy as $\pi_{\theta}(s, a)$ with a parameter vector θ , where $\theta \in \Re^l$, for l << |S|. Another advantage of function approximators is that the agent could take similar actions for "close-by" states.

There are many forms of function approximators that can be used to represent the policy. It is popular to use linear combinations of features of the state/action space as the function approximator. Recently Deep Neural Networks (DNNs) have been successfully used to account for large-scale reinforcement learning tasks. Generally speaking, algorithms that use DNNs are belong to Deep Learning techniques. The insight of using DNNs as the function approximators is that they do not need human-crafted features. In this paper, we use a DNN to represent the policy and thus our algorithm is Deep-learning enabled.

3) POLICY GRADIENT METHODS

Policy gradient methods are heavily used in the recent stateof-the-art reinforcement learning algorithms. In these methods, training of the policy is performed by following the gradient of cumulative discounted reward with respect to the parameter vector, which is given by [15]:

$$\nabla_{\theta} \mathbb{E}[\sum_{t=0}^{\infty} \gamma^{t} r_{t}] = \mathbb{E}_{\pi_{\theta}}[\nabla_{\theta} \log \pi_{\theta}(s, a) Q^{\pi_{\theta}}(s, a)], \quad (14)$$

where $Q^{\pi_{\theta}}(s, a)$ is the value of expected cumulative discounted reward by deterministically choosing action *a* in state *s*. The insight of policy gradient methods is to estimate the gradient by observing the trajectories of executions that are obtained by executing the policy. The agent emprically calculates the cumulative discounted reward v_t with the simple *Monte Carlo Method* [16] by sampling multiple trajectories, and uses it as an unbiased estimate of $Q^{\pi_{\theta}}(s_t, a_t)$. Thus the policy parameters are updated via gradient descent as follows

$$\theta \leftarrow \theta + \alpha \sum_{t} \nabla_{\theta} \log \pi_{\theta}(s_t, a_t) v_t,$$
(15)

where α is the step size. This equation leads to the episodic REINFORCE algorithm [15] which can be intuitively understood as follows. The agent should update the policy parameters following the direction of $\nabla_{\theta} \log \pi_{\theta}(s_t, a_t)$ thus to increase $\pi_{\theta}(s_t, a_t)$ (the probability of action a_t at state s_t). This results in the effect that reinforces actions that empirically lead to better performance.

B. USER SCHEDULING SCHEME

The user scheduling scheme takes into account both spectrum efficiency and user fairness. We choose the first user with classical proportional fair (PF) scheduling factor. While scheduling the other users, we introduce the group alignment of users' signal strength *R* to ensure spectrum efficiency, and meanwhile exploit *SINR*_{min} to ensure user estimated *SINR* no less than it thus ensuring system capacity. We assume that *L* is the user set to be scheduled, *g* is the scheduled user set, r_k is the instantaneous data rate of user *k*, D_k is the average data rate of user *k*, *M* is the number of scheduled users, K^* is the maximum number of scheduled users, λ_{max} and λ_{min} are the maximum and the minimum singular value respectively, *R* is the maximum group alignment of users' signal strength.

In this paper, we seek to dynamically choose $SINR_{min}$ and R for each TTI. The resulting user scheduling scheme consists of two phases: 1) In each TTI, $SINR_{min}$ and R are identified via the deep reinforcement learning algorithm. 2) Then $SINR_{min}$ and R are utilized by the subsequent user scheduling scheme.

1) THE DEEP REINFORCEMENT LEARNING FORMULATION

State space. Each sector of the cell is defined as a agent that aims to maximize both CASE and CESE of the cell. We define the continuous state as: $s_t = \{CASE_t, CESE_t\}$, where CASE and CESE of the cell at timestep *t* can be calculated by the previous definition.

Action space. The action space is constructed by the combinations of $SINR_{min}$ and R parameter sets. Suppose there are m discrete levels of $SINR_{min}$ and n discrete levels of R respectively, thus the action space consists of $m \cdot n$ combinations. We use DNN as function approximator to compute the policy that the agent should follow in a given state, where the policy is the combination of $SINR_{min}$ and R with the largest probability that can result in the largest reward.

$$a_t = \arg\max_a \pi_\theta(s_t, a) \tag{16}$$

DNN is used to approximate the policy in the sense that the state space is continuous in our scenario, where we can not use a tabular to store the learned policies.

Rewards. The reward function is calculated with Equation 17 by taking into account the sub-rewards for CASE and CESE.

$$r_t = \eta \cdot r_{CASE_t} + (1 - \eta) \cdot r_{CESE_t}, \tag{17}$$

where η ($0 \le \eta \le 1$) is the weight that enables the setting of a desired tradeoff between CASE and CESE. The sub-rewards are defined as follows

$$r_{CASE_t} = \begin{cases} 1, & CASE_{t+1} \ge CASE_t, \\ -1, & otherwise. \end{cases}$$
(18)

The same expression is applied for r_{CESE_t} .

2) TRAINING

Based on the above formulation, we represent the policy as a neural network (called policy network) which takes as input the state s_t , and outputs a probability distribution over all possible actions. The policy network is trained with a variant of the REINFORCE algorithm in an episodic setting. In each training iteration, we run N episodes for a fixed duration of T TTIs to explore the probabilistic space of possible actions using the current policy, and use the resulting data to improve the policy. We should notice here that an episode is the whole simulation procedure of a TTI consisting of computing $SINR_{min}$ and R via policy network, utilizing the parameters in user scheduling, and inter-cell interference coordination. The trajectories consisted of the state, action, and reward are recorded for all timesteps of each episode, and use these values to calculate the discounted cumulative reward, v_t , at each timestep t of each episode. The estimation of policy gradient with Equation 15 introduces high variance, thus we need to subtract a *baseline* from v_t . A simple approach that we calculate the baseline is to use the average of the discounted cumulative rewards. The implementation of the variant REINFORCE algorithm is described as Algorithm 1.

Algorithm 1 Policy gradient algorithm				
1:	Initialize policy with parameter θ_1			
2:	for iteration $k = 1, 2,$ do			
3:	$\Delta \theta \leftarrow 0$			
4:	run episode $i = 1, \ldots, N$:			
5:	Obtain $\{s_1^i, a_1^i, r_1^i, \ldots, s_T^i, a_T^i, r_T^i\}$ under policy π_{θ_k}			
6:	for $t = 1, \dots, T$ do			
7:	for $i = 1, \dots, N$ do			
8:	compute rewards: $v_t^i = \sum_{s=t}^T \gamma^{s-t} r_s^i$			
9:	compute baseline: $b_t^i = \frac{1}{N} \sum_{i=1}^N v_t^i$			
10:	$\Delta\theta \leftarrow \Delta\theta + \alpha \nabla_{\theta} \log \pi_{\theta}(s_t^i, a_t^i)(v_t^i - b_t^i)$			
11:	end for			
12:	end for			
13:	$\theta_k \leftarrow \theta_k + \Delta \theta$			
14:	14: end for			

C. INTER-CELL INTERFERENCE COORDINATION SCHEME

Interference coordination mechanism is completed by the measurement module and the precoding module [17]–[19]. The basic scheme for mitigating inter-cell interference is distributed zero forcing, which needs channel matrix information of neighbor cells. In order to control the spatial freedom of transmission antennas used for inter-cell interference coordination, we define a new parameter *percentage of cell-edge users needing inter-cell interference suppression* denoted by δ for setting the percentage of users that need interference suppression so as to adjust the spatial freedom resources between inter-cell and intra-cell interference cancellation. Denote e_s as the number of cell-edge users in cell *s*. δ with the value of 100% means that all the cell-edge users need to perform interference coordination scheme are as follows.

Step 1: Each service cell measures the downlink average *SINR* of each UE, and counts all users whose downlink average *SINR* is lower than $SINR_{min}$, and defines these users as cell-edge users. These cell-edge users are sorted by *SINR* in ascending order.

Step 2: Each service cell sends a command to the edge users to measure the strong interference cells and estimate the channel matrix of the strong interference cells.

Step 3: Each serving cell forms the interference matrix table as shown in Table 1.

TABLE 1. Interference matrix table.

Service cell-edge	Average SINR	Neighbor cell	Neighbor cell
user ID	(dB)	ID	channel matrix
CELL1_UE1	-3dB	CELL2	H_C1U1_C2
CELL1_UE2	-1dB	CELL3	H_C1U2_C3
•••	•••	•••	

Step 4: Each cell in the network interacts with the respective interference matrix table on X2 interface, and obtains interfering edge user information of neighbor cells and the channel matrix to these users.

Step 5: When precoding, according to ascending order of average *SINR*, each service cell selects the interfered users whose channel vectors constitute null space, and interference

Algorithm 2 DECCO Algorithm

Phase 1 – User Scheduling

- 1: Initialization: $L = \{1, 2, \dots, K\}, g = \emptyset, k^*$ $\operatorname{argmax}(\frac{r_k}{D_k}), g = g \cup \{k^*\}, M = 1, L = L \setminus \{k^*\}$
- 2: Compute $SINR_{min}$ and R_{max} with policy gradient network
- 3: *schedule_allow_flag* = True
- 4: while $M \leq K^*$ and schedule_allow_flag = True do
- 5: $L_{tmp} = L$
- while $L_{tmp} \neq \emptyset$ do 6:
- Select k^* from L_{tmp} , $g' = g \cup \{k^*\}$, calculate 7: **B**, **P** according to **H** formed by gg', calculate SINR, do singular value decomposition for **BP** and obtain λ_{max} and λ_{min}

8:	if $\frac{\lambda_{\max}}{\lambda_{\min}} \leq R_{\max}$ and SINR $\geq SINR_{\min}$ then
9:	$g = g \cup \{k^*\}, M = M + 1, L = L \setminus \{k^*\}$
	$schedule_allow_flag = True$
10:	break

- 10:
- else 11:
- $L_{tmp} = L_{tmp} \setminus \{k^*\}, schedule_allow_flag =$ 12:
- False
- 13: end if
- end while 14:
- 15: end while

Phase 2 – Inter-Cell Interference Coordination

- 16: Initialization: calculate SINR and obtain edge-user set e_s , obtain the null space matrix H of the edge-users'
- for each cell do 17:
- Obtain the outer precoding matrix **B** and achieve 18: interference suppression for neighboring cells by $\mathbf{H}^{H}\mathbf{B}$
- Obtain the inner precoding matrix **P** and form the 19: final precoding matrix $\mathbf{H}^{H}\mathbf{B}\mathbf{P}$

20: end for

suppression is performed by selecting channels of the first $\delta \times e_s$ users of each neighbor cell.

Step 6: The service cell generates the null space matrix of the interfered user channel matrix, and multiplies the outer precoder by the null space matrix to achieve interference suppression for neighbor cells.

Step 7: The service cell constructs inner precoding to form the final precoding matrix.

In conclusion, the overall DECCO algorithm is formed by user scheduling scheme with a pre-trained policy network and the subsequent inter-cell interference coordination scheme, and it is concluded in Algorithm 2.

V. SIMULATION RESULTS

In this section, we demonstrate performance and behavior of the proposed concepts and models for optimization of coverage and capacity by simulative evaluation [20] of representative case studies.

A. SIMULATION SETUP

We take the International Telecommunications Union's (ITU) three-dimensional urban macro cell (3D-UMa) model as our channel model. The number of antennas at BS is 64, and the number of antennas at UE is 2. We use the JSDM scheme as the downlink transmission method. SINR_{min} is quantified for 15 levels from 1dB to 15dB, and R is quantified for 20 levels from 25 to 500. Other parameters needed for the simulation are listed in Table 2.

Parameter	Value
Channel model	3D-Uma
Antenna configuration at BS	Number:64,
	plane array:8x8, cross-polarized
Antenna configuration at UE	Number:2, cross-polarized
Terminal moving speed	3km/h
Multi-user transmitting	JSDM
precoding algorithm	
Terminal receiving algorithm	ZF
Channel Estimation,	Ideal
Channel Measurement	
Total BS transmit power	46dBm
Simulation bandwidth	10MHz
Carrier frequency	2GHz
Number of UEs per cell	20
Traffic model	Full buffer
UE dropping	Outdoor:20%, Indoor:80%
Inter-cell interference	Enabled
cancellation	
Number of scheduling layers	8
The user minimum SINR	[1, 2, 3, 4, 5, 6, 7, 8, 9,
threshold(SINR _{min})	10, 11, 12, 13, 14, 15]dB
Group alignment of users'	[25,50,75,100,125,150,175,
signal strength (R)	200,225,250,275,300,325,350,
	375,400,425,450,475,500]
Discount factor(γ)	0.9
Episodes (Batch) for a update	20
Number of hidden layers	2

TABLE 2. System simulation parameter settings.

Once the simulation parameters are set, the architecture of the policy network for computing the proposed parameters SINR_{min} and R in the DECCO algorithm can be indentified. The input layer of the resulting neural network has 2 neurons, which accept CASE and CESE of the cell respectively. The output layer consists of 300 neurons, which denote the full combinations of SINR_{min} and R parameter sets and output the probability distribution of policies. We use two hidden layers to learn and approximate the optimal policy, where each layer consists of 100 neurons. Thus the policy network has 4 layers, and a total of 6,000,000 parameters.

B. POLICY NETWORK TRAINING

As disussed earlier, the policy network is used to derive the combination of SINR_{min} and R for the downlink transmission. Concretely, it maps an environment state into an policy (or action) by learning from tremendous trajectories that the agent experienced. In this sense, training the policy network is performed interactively between the agent and the environment. We leave the learning rate α as a hyperparameter of the policy network in the DECCO algorithm to tune during training. We run 1000 iterations for training, and in each training iteration, we run N = 20 Monte Carlo simulations

(also named episodes) in parallel. We update the policy network parameters using the *stochastic gradient ascent* with the configurable learning rate.

We tune the learning rate and investigate the average reward performance with different learning rate used by updating the policy network parameters. Figure 5 shows the average reward trend with three different learning rate 0.01, 0.05, 0.10. To make the figure more readable, we use the common logarithm of iteration. It is obvious that the average reward converges to a better stage with learning rate $\alpha = 0.01$.



FIGURE 5. Average reward per BS of the policy network training with different learning rate.

C. INFLUENCE OF THE WEIGHT FACTOR η

Recall that the weight factor η in the reward function controls the balance between maximizing CASE and CESE of the cell. We investigate the influence of weight factor η adopted in the DECCO algorithm on capacity-coverage performance as the number of BS N increases from 1 to 7. Specifically, we set δ , the percentage of cell-edge users needing inter-cell interference suppression, with 100% for this scenario. Unless otherwise specified, the results below are from the CCO algorithms with $\delta = 100\%$. Figure 6 and Figure 7 together shows the average CASE and CESE performances with different weight factors as the number of BSs grows. The weight factor with $\eta = 0.0$ and $\eta = 1.0$ are two special cases, where the two optimization objectives that maximize CASE and CESE are reduced to a single objective problem. When $\eta = 0.0$, the DECCO algorithm only optimizes CESE since the term of CASE becomes zero. This case may not happen in practical network operation. When $\eta = 1.0$, CASE is the only objective to be optimized by the DECCO algorithm, and this case may happen in practical network operation since only providing good service for the majorities is a possible option for the network operators. When $\eta = 0.8$, CASE is the dominant factor that contributes to the average reward thus CESE grows slower than CASE. While the vice versa trend of CASE versus CESE occurs when $\eta = 0.3$. An equal performance gains of CASE and CESE are achieved when we choose the weight factor η to be 0.5, which is



FIGURE 6. Average CASE per BS obtained with DECCO adopted different weight factors as the number of BSs increases. For all the three cases, the percentage of cell-edge users needing inter-cell interference suppression is 100%.



FIGURE 7. Average CESE per BS obtained with DECCO adopted different weight factors as the number of BSs increases. For all the three cases, the percentage of cell-edge users needing inter-cell interference suppression is 100%.

safe for whatever the number of BSs is. However, the intercell interference increases with the number of BSs. Thus CESE decreases more quickly than CASE as the number of BSs increases. In this sense, the weight factor of CESE that controls the contribution of CESE to the reward, $1 - \eta$, should be increased to effectively optimize CESE by learning as the number of BSs grows.

D. PERFORMANCE COMPARISON OF DIFFERENT CCO ALGORITHMS

We would like to point out that the evaluation of DECCO and other CCO algorithms is made within the user scheduling scheme, and all CCO algorithms use the same ICIC scheme as proposed. We compare the capacity-coverage performance between DECCO with the dynamic configuration of $SINR_{min}$ and *R* computed by the policy network in the user scheduling algorithm at runtime and a CCO algorithm with best fixed configuration of $SINR_{min}$ and *R* obtained through trial-anderror as the number of BS increases, where we denote the later algorithm as Fixed coverage and capacity Optimization (FO). Moreover, we set the CCO algorithm with proportional fair (PF) scheduling scheme and the proposed ICIC scheme as the baseline for the comparison, where we denote this algorithm as PFO. The weight factor η for the DECCO algorithm, and *SINR*_{min} and *R* for the FO algorithm with the specified number of BSs are listed in Table 3.

TABLE 3. Parameter settings.

Num of BSs	η	$SINR_{\min}$	R
1	0.8	12	75
2	0.8	10	100
3	0.7	10	150
4	0.6	8	175
5	0.5	7	150
6	0.4	8	175
7	0.3	5	200

Figure 8 and Figure 9 depict that the proposed DECCO algorithm outperforms the FO algorithm and the PFO algorithm on both CASE and CESE. In this sense, $SINR_{min}$ and R are two effective capacity-coverage optimization parameters. The FO algorithm outperforms the PFO algorithm on CASE, while it does not perform better than the PFO algorithm on CESE since it schedules the users with fixed thresholds which fails to track the changing inter-cell interference. As the number of BSs increases, the performance gains of the DECCO algorithm against the FO algorithm decrease. In this In practical systems, we can partition learning groups to alleviate learning performance degradation with large scale learning. For example, we can set every learning group with 4 base stations. Figure 10 depicts the empirical cumulative density



FIGURE 8. Average CASE per BS with different CCO algorithms as the number of BSs increases.



FIGURE 9. Average CESE per BS with different CCO algorithms as the number of BSs increases.



FIGURE 10. Empirical CDF of different CCO algorithms with the number of BSs N = 7.

TABLE 4. Capacity-coverage performance at runtime.

Performance		DECCO ($\eta = 0.3$)	DECCO ($\eta = 0.8$)	FO	PFO
Spectrum	5%	0.57	0.55	0.35	0.53
officiency	50%	5.47	6.32	5.17	4.63
[bpc/Hz]	95%	11.34	13.84	10.82	9.20
[0ps/fiz]	mean	5.44	6.41	5.15	4.58
	SD	1.54	1.23	2.32	1.67

function (Empirical CDF) of spectrum efficiency of different CCO algorithms at runtime, where the number of BSs N = 7. We also evaluate the influence of the different weight factors on our proposed DECCO algorithm working at runtime when $\eta = 0.3$ and $\eta = 0.8$. The results are concluded in Table 4, where SD is the abbreviation of Standard Deviation. The DECCO algorithm with $\eta = 0.3$ outperforms the FO algorithm and the PFO algorithm by 5.6% and 18.1% on CASE, respectively. When it comes to CESE, the DECCO algorithm with $\eta = 0.3$ outperforms the FO algorithm and the PFO algorithm by 62.9% and 7.5%, respectively. Thus it is explicit that the performance gain of CESE is larger than that of CASE since CESE gains more weight than CASE. Moreover, the PFO algorithm outperforms the FO algorithm on CESE just as we discussed earlier. When $\eta = 0.8$, the DECCO algorithm outperforms the FO algorithm and the PFO algorithm by 22.2% and 36.5% on CASE, respectively. In terms of CESE, the DECCO algorithm outperforms the FO algorithm and the PFO algorithm by 57.1% and 3.8%, respectively. In contrast to $\eta = 0.3$, the performance gain of CASE is larger than that of CESE due to the fact CASE plays a more important role in reward calculation with $\eta = 0.8$. This is in consistency with the investigations of influence of the weight factor we have conducted. An important observation is that the DECCO algorithm has more potential on maximizing CESE than CASE in comparison with the FO algorithm. Beyond that, the DECCO algorithm improves CASE larger than the PFO algorithm. In addition, the DECCO algorithm has a smaller SD value, which means the coverage and capacity optimization performance is more fair and stable. As a result, our proposed DECCO algorithm is a generally optimal method for capacity-coverage optimization.

VI. CONCLUSIONS

In this paper, we have proposed a novel Deep reinforcement learning Enabled Coverage and Capacity Optimization (DECCO) algorithm, in which a deep reinforcement learning-based user scheduling scheme and a novel intercell interference coordination (ICIC) scheme are contained, to address the coverage and capacity in massive MIMO networks. In addition, we proposed a novel optimization parameter GAUSS, i.e. Group Alignment of Users' Signal Strength, is proposed together with Together with a unified threshold of QoS, i.e. SINR_{min} to be dynamically configured with a pre-trained deep policy gradient-based neural network in each transmission time interval in the user scheduling scheme. Furthermore, a novel ICIC scheme has been proposed to further enhance the performance of the deep reinforcement learning-based user scheduling scheme. We conducted extensive simulations to compare the capacity-coverage performance between our proposed DECCO algorithm with a CCO algorithm utilizing the best fixed configuration of the proposed optimization parameters in the user scheduling scheme, and used the CCO algorithm with proportional fair scheduling as the baseline. Simulation results show that 1) our proposed optimization parameters are effective in optimizing the coverage and capacity, 2) our proposed DECCO algorithm outperforms the other two CCO algorithms with the same ICIC scheme on both coverage and capacity. This means our methods successfully track the dynamics of the considered systems. Moreover, we can set learning clusters to account for learning gains decreasing as opposed to the scale of networks. Thus future work will be done to enable coverage and capacity optimization in large scale learning.

ACKNOWLEDGMENT

A portion of this paper was presented at the IEEE Vehicular Technology Conference, Sydney, NSW, Australia, June 2017 [1].

REFERENCES

- K. Li, Y. Yang, Y. Chen, X. Yang, and H. Yi, "A novel network optimization method for cooperative massive MIMO systems," in *Proc. IEEE VTC Spring*, Jun. 2017, pp. 1–5.
- [2] H. Liu, H. Gao, S. Yang, and T. Lv, "Low-complexity downlink user selection for massive MIMO systems," *IEEE Syst. J.*, vol. 11, no. 2, pp. 1072–1083, Jun. 2017.
- [3] X. H. Ge et al., "Wireless single cellular coverage boundary models," IEEE Access, vol. 4, pp. 3569–3577, 2016.
- [4] X. H. Ge et al., "Wireless fractal cellular networks," IEEE Wireless Commun., vol. 23, no. 5, pp. 110–119, Oct. 2016.
- [5] X. Ge, R. Zi, X. Xiong, Q. Li, and L. Wang, "Millimeter wave communications with OAM-SM scheme for future mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2163–2177, Sep. 2017.
- [6] R. Combes, Z. Altman, and E. Altman, "A self-optimization method for coverage-capacity optimization in OFDMA networks with MIMO," in *Proc. 5th Int. ICST Conf. Perform. Eval. Methodol. Tools*, May 2011, pp. 37–46.
- [7] I. S. Comsa, S. Zhang, M. Aydin, P. Kuonen, and J. F. Wagen, "A novel dynamic Q-learning-based scheduler technique for LTE-advanced technologies using neural networks," in *Proc. IEEE Conf. Local Comput. Netw. (LCN)*, Oct. 2012, pp. 332–335.

- [8] Y. Sun, S.-S. Lv, S. Liu, and Y.-H. Zhang, "Joint user scheduling and power allocation for massive MIMO downlink with two-stage precoding," in *Proc. IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2016, pp. 2917–2921.
- [9] J. Nam, J.-Y. Ahn, A. Adhikary, and G. Caire, "Joint spatial division and multiplexing: Realizing massive MIMO gains with limited channel state information," in *Proc. 46th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2012, pp. 1–6.
- [10] Y. Xu, G. Yue, N. Prasad, S. Rangarajan, and S. Mao, "User grouping and scheduling for large scale MIMO systems with two-stage precoding," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 5197–5202.
- [11] Q. Li, S. Feng, A. Pandharipande, X. Ge, Q. Ni, and J. Zhang, "Wirelesspowered cooperative multi-relay systems with relay selection," *IEEE Access*, vol. 5, pp. 19058–19071, 2017.
- [12] J. Nam and J.-Y. Ahn, "Joint spatial division and multiplexing—Benefits of antenna correlation in multi-user MIMO," in *Proc. IEEE Inf. Symp. Inf. Theory*, Jul. 2013, pp. 619–623.
- [13] R. S. Sutton and A. G. Barto, *Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1998.
- [14] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proc. 15th ACM Workshop Hot Topics Netw. (HotNets)*, Nov. 2016, pp. 50–56.
- [15] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. NIPS*, 1999, pp. 1057–1063.
- [16] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.
- [17] D. Lee *et al.*, "Coordinated multipoint transmission and reception in LTE-advanced: Deployment scenarios and operational challenges," *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 148–155, Feb. 2012.
- [18] H. Dahrouj and W. Yu, "Coordinated beamforming for the multicell multiantenna wireless system," *IEEE Trans. Wireless Commun.*, vol. 9, no. 5, pp. 1748–1759, May 2010.
- [19] W. Yu, T. Kwon, and C. Shin, "Multicell coordination via joint scheduling, beamforming, and power spectrum adaptation," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 1–14, Jul. 2013.
- [20] Y. Yang, J. Xu, G. Shi, and C.-X. Wang, 5G Wireless Systems: Simulation and Evaluation Techniques. Berlin, Germany: Springer, 2017.



YANG YANG (S'99–M'02–SM'10–F'18) received the B.Eng. and M.Eng. degrees in radio engineering from Southeast University, Nanjing, China, in 1996 and 1999, respectively, and the Ph.D. degree in information engineering from The Chinese University of Hong Kong in 2002.

He has held faculty positions with The Chinese University of Hong Kong, Brunel University, and University College London, U.K. He is currently a Professor with the Shanghai Institute of Microsys-

tem and Information Technology, Chinese Academy of Sciences, serving as the Director of the CAS Key Laboratory of Wireless Sensor Network and Communication, the Director of the Shanghai Research Center for Wireless Communications, and the Co-Director of the Shanghai Institute of Fog Computing Technology. He is also a Distinguished Adjunct Professor with the School of Information Science and Technology, ShanghaiTech University.

Dr. Yang is a member of the Chief Technical Committee of the National Science and Technology Major Project New Generation Mobile Wireless Broadband Communication Networks from 2008 to 2020, which is funded by the Ministry of Industry and Information Technology, China. In addition, he is on the Chief Technical Committee for the National 863 Hi-Tech R&D Program 5G System R&D Major Projects, which is funded by the Ministry of Science and Technology, China. Since 2017, he has been serving the OpenFog Consortium as a Board Member and the Director of Greater China Region.

He has published over 160 papers and filed over 80 technical patents in wireless communications. His current research interests include wireless sensor networks, Internet of Things, fog computing, open 5G, and advanced wireless testbeds.

IEEEAccess



YANG LI received the B.E. degree in communication engineering from Hubei University, Wuhan, China, in 2012. He is currently pursuing the Ph.D. degree in co-supervision with the Key Laboratory of Wireless Sensor Network and Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, ShanghaiTech University, and also with the University of Chinese Academy of Sciences. His research interests include machine learning in

radio domain, wireless big data, WiFi based indoor localization, WiFi imaging, and practical experiments.



KAI LI is currently a System Engineer of the Key Laboratory of Wireless Sensor Network and Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences. He is in charge of designing and developing a new generation of wireless simulation platform for 5G and key technology verification. In recent two years his team has published the first version of 5G simulation platform and submitted several papers and patent applications and stan-

dard proposals. He has over 10 years of communication industry design and development experience. In HUAWEI Technologies Co. Ltd., he was responsible for the planning and designing of wireless product line operation and maintenance solutions, wireless network optimization system design and development work and the base station software development work. He is also familiar with CMM process and Agile development process and has many years of research and development project management practice.



SHUANG ZHAO received the B.S. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2016, and the M.E. degree from the Wuhan University of Technology, Wuhan, China, in 2013. She is currently pursuing the Ph.D. degree with the Key Laboratory of Wireless Sensor Network and Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences.

Her research interests focus on applications of artificial intelligence in wireless communication, wireless big data, and the resource management and scheduling in wireless networks.





RUI CHEN received the B.S., M.S., and Ph.D. degrees in communications and electronic system from Xidian University, Xi'an, China, in 2005, 2007, and 2011, respectively. From 2014 to 2015, he was a Visiting Scholar with Columbia University. He is currently an Associate Professor with the School of Telecommunications Engineering, Xidian University. His research interests include MIMO array signal processing of wireless communication systems and intelligent transportation systems.

JUN WANG is currently a Chair Professor of data science, computer science with University College London, and the Founding Director of M.Sc. Web Science and Big Data Analytics. He is also a Co-founder and a Chief Scientist with MediaGamma Ltd., a UCL spin-out focusing on AI for intelligent audience decision making.

He has published over 100 research papers and was a recipient of multiple best paper awards. His main research interests are in the areas of intelli-

gent information systems, covering information retrieval, personalization, data mining, online advertising, reinforcement learning, and deep learning. His team won the first global real-time bidding algorithm contest with 80+ participants worldwide. He was also a recipient of the Beyond Search Semantic Computing and Internet Economics Award by Microsoft Research and also received Yahoo! FREP Faculty Award. He has served as an Area Chair in ACM CIKM and ACM SIGIR. MediaGamma has received the UCLB One-to-Watch Award 2016.

He has more than 15 years track records of advising the industry and knowledge transfer. He was a technical advisor for startups such as Last.Fm, Passiv Systems, Massive Analytic, Context Scout, and Polecat, and had various projects with BT, Microsoft, Yahoo!, and Alibaba.

He received the bachelor's degree from Southeast University, Nanjing, China, the M.Sc. degree from the National University of Singapore, Singapore, and the Ph.D. degree from the Delft University of Technology, The Netherlands.



SONG CI (S'98–M'02–SM'06) is currently an Associate Professor with the ECE Department, University of Nebraska–Lincoln, USA. He has authored over 200 peer-reviewed articles in those areas, and his research has been support by NSF and other funding sources. His current research interests include large-scale dynamic complex system modeling and optimization, energy Internet and distributed energy management, green computing and communications, and mobile Internet.

He is a member of ACM. He has served as an Editor or a Guest Editor in many journals, such as the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE ACCESS, and the IEEE WIRELESS NETWORK. He served on TPC of numerous international conferences.

. . .