**MULTIMEDIA COMMUNICATIONS TECHNICAL COMMITTEE**

***http://www.comsoc.org/~mmc***

***MMTC Communications - Frontiers***

**Vol. 15, No. 5, September 2020**

**CONTENTS**

[**SPECIAL ISSUE ON Emerging Technologies for Internet of Electric Vehicles** 2](#_Toc88340230)

[*Guest Editor: Ran Wang\* and Hehong Zhang\*\** 2](#_Toc88340231)

[*Nanjing University of Aeronautics and Astronautics, China\** 2](#_Toc88340232)

[*Fuzhou University, China\*\* Email: wangran@nuaa.edu.cn, hzhang030@e.ntu.edu.sg* 2](#_Toc88340233)

[**Multi-objective Mobile Charging Scheduling on the Internet of Electric Vehicles: a DRL Approach** 3](#_Toc88340234)

[*Hui Wang1, Ran Wang1, Hu Xu1,* *Zhu Kun1, Changyan Yi1, Dusit Niyato2* 3](#_Toc88340235)

[*1Nanjing University of Aeronautics and Astronautics, China ,*2*Nanyang Technological University, Singapore* 3](#_Toc88340236)

[**Predicting Crowd Flow: A survey from perspective of TODO** 12](#_Toc88340237)

[*Shuyu Wang1,3, Yuhang Xu2, Yan Lyu2, Weiwei Wu2* 12](#_Toc88340238)

[1 *School of Cyber Science and Engineering, Southeast University, Nanjing, China* 12](#_Toc88340239)

[2 *School of Computer Science and Engineering, Southeast University, Nanjing, China* 12](#_Toc88340240)

[3*School of Information Engineering, Xizang Minzu University, Xianyang, China* 12](#_Toc88340241)

[**Collaborative Video Analytics with Edge-Cloud Collaboration: Implementation and Evaluation** 18](#_Toc88340242)

[*Xuezhi Wang and Guanyu Gao* 18](#_Toc88340243)

[*Nanjing University of Science and Technology* 18](#_Toc88340244)

[**Discrete-Time Optimal Control based Tracking Differentiator Algorithms** 24](#_Toc88340245)

[*Hehong Zhang1, Yunde Xie2, Qiumeng Qian3,4\** 24](#_Toc88340246)

[*1College of Computer and Data Science, Fuzhou University, Fuzhou, China* 24](#_Toc88340247)

[*2Beijing Rail Transit Technology Equipment Group Co., Ltd, Beijing, China* 24](#_Toc88340248)

[*3School of Automation Science and Electrical Engineering, Beihang University, China* 24](#_Toc88340249)

[*4Sichuan Gas Turbine Establishment, Aero Engine Corporation of China , Mianyang, China* 24](#_Toc88340250)

[**MMTC OFFICERS (Term 2020 — 2022)** 30](#_Toc88340251)

**SPECIAL ISSUE ON** **Emerging Technologies for Internet of Electric Vehicles**

*Guest Editor: Ran Wang\* and Hehong Zhang\*\**

*Nanjing University of Aeronautics and Astronautics,* *China\**

*Fuzhou University, China\*\*  
Email:* [*wangran@nuaa.edu.cn*](mailto:wangran@nuaa.edu.cn)*,* [*hzhang030@e.ntu.edu.sg*](mailto:hzhang030@e.ntu.edu.sg)

This special issue of Frontiers focuses on the recent progresses of the emerging technologies for Internet of Electric Vehicles (EVs). The research topics of the papers in this special issue include mobile charging services for EVs, crow flow prediction and human trajectory prediction, edge-cloud collaboration framework for video analytics for autonomous vehicle, and time optimal control-based tracking differentiator algorithm.

The first paper studied the scheduling of mobile charging services (MCSs) to provide electric vehicles (EVs) with energy, and the aim is to promote the overall quality of the MCS. The authors modeled the charging scheduling problem as a multi-objective optimization problem. The focus is to minimize the total service time (EV waiting time and charging time) and maximize the charging benefit of all EVs simultaneously. The authors adopted a deep reinforcement learning-based approach to model and solve the MCV scheduling problem.

The second paper presented a comprehensive review for crowd flow prediction, including crowd counting and density estimation, human trajectory prediction, and crowd flow prediction. The authors identified three challenging problems that are critical for predicting indoor dense crowd flow and discussed the main challenges for indoor dense crowd flow prediction and the value and significance of the research in this field.

The third paper presented the design, implementation, and evaluation of an open-source framework for real-time video analytics by leveraging the edge-cloud collaboration. The framework can leverage the advantages of the edge computing and the cloud computing to improve the performances of video analytics. The users can easily customize the system control policies to conduct performance evaluations, which can facilitate the research in this field. The authors also presented some application scenarios of the framework, e.g., video surveillance, autonomous driving.

The fourth paper investigated the time optimal control-based tracking differentiator algorithms. The authors studied a generalized discrete time optimal control algorithm that can be used to construct the differentiators and obtained the general form of DTOC-TD algorithms by flexibly modifying characteristic points. The proposed method can achieve better performance and higher precision in signal-tracking filtering and differentiation acquisition.

**Ran Wang** (M'18) is an Associate professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, and Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, P.R. China. He received his B.E. in Electronic and Information Engineering from Honors School, Harbin Institute of Technology, P.R. China in July 2011 and Ph.D. in Computer Science and Engineering from Nanyang Technological University, Singapore in April 2016. He was a research fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore from October 2015 to August 2016. He has authored or coauthored over 50 papers in top-tier journals and conferences. He received the Nanyang Engineering Doctoral Scholarship (NEDS) Award, Singapore and innovative and entrepreneurial Ph.D. Award of Jiangsu Province, China in 2011 and 2017, respectively. His current research interests include network performance analysis and internet of electric vehicles, etc.

****

**Hehong Zhang** received his Ph.D. degree in smart control and data analysis from Nanyang Technological University, Singapore, in 2020. He is now an Associate Professor in College of Computer and Data Science, Fuzhou University, Fuzhou, China. His research interests include differentiator theory, state estimation, signal processing, theory and application of optimal control and nonlinear control theory in the altitude ground test facility and maglev train system.

**Multi-objective Mobile Charging Scheduling on the Internet of Electric Vehicles: a DRL Approach**

*Hui Wang1, Ran Wang1, Hu Xu1,* *Zhu Kun1, Changyan Yi1, Dusit Niyato2*

*1Nanjing University of Aeronautics and Astronautics, China ,*2*Nanyang Technological University, Singapore*

[*wanghui9727@nuaa.edu.cn*](mailto:wanghui9727@nuaa.edu.cn)*,* [*wang0686@e.ntu.edu.sg*](mailto:wang0686@e.ntu.edu.sg)*,* [*xuhu1998@nuaa.edu.cn*](mailto:xuhu1998@nuaa.edu.cn)*,* [*hiiamzk@gmail.com*](mailto:hiiamzk@gmail.com)*,* [*changyan.yi@nuaa.edu.cn*](mailto:changyan.yi@nuaa.edu.cn)*,* [*dniyato@ntu.edu.sg*](mailto:dniyato@ntu.edu.sg)

**1. Introduction**

Compared with internal combustion engine vehicles powered by gasoline or diesel oil, electric vehicles (EVs) exhibit higher energy efficiency and lower noxious gas emissions. Driven by such advantages, EVs are deemed as an eco-friendly and efficient alternative to traditional fossil fuel based vehicles. However, a key barrier to the wide adoption of EVs is the limited battery capacity. Furthermore, depleted batteries typically require several hours to be fully charged, while gasoline-driven vehicles can fill up the tanks in just a few minutes. Recent techniques, such as the adoption of quick chargers and battery swapping, may partially overcome these concerns; however, the corresponding huge capital investments may be unprofitable for areas with low EV adoption rates (For example, a battery swap station typically costs around $500,000 [1].). In addition, due to mobility issues, randomly distributed EVs may have diverse charging demands both in space and time, severely reducing the quality of current fixed charging stations (FCSs). In addition, FCSs with a limited number of charging piles are unable to cope with sudden high charging demands. During the peak time, Elevated charging requirements may consequently result in the charging requests from EVs being denied or the marked increase in EV waiting times at the FCS[2].

To fulfill the diverse charging demands of EVs both in space and time, mobile charging services (MCSs) provide a promising solution as an alternative implementation of the present battery recharge options (e.g., fixed charging station, quick charger outlets, and battery swap station) [1]. The service is provided by mobile charging vehicles (MCVs), which have a self-contained energy storage system employed to replenish the energy of a certain number of EVs. Its removable, inexpensive, and easy to deploy properties facilitate charging services by proactively and instantly responding to charging requirements, particularly at moments of potentially high grid pressure.

MCV scheduling constitutes one of the key problems in MCS, which requires determining the charge sequence and the amount of charging when serving multiple EVs by one MCV. In general, EV owners concern about two aspects: the waiting time and the charging benefit. The former relates to the traveling distance and the charging time, while the latter only references the amount of charging electricity. Obviously, these two aspects are conflicting with each other because a shorter waiting time requires charging less electricity that will reduce the charging benefit. Under such a case, the scheduling of MCV has to consider the aforementioned factors to ensure the quality of charging services. A set of trade-off solutions, termed Pareto optimal solutions, are expected to be found for MOPs. In addition, to ensure the real-time response of the charging services, MCV scheduling has high requirements for the computation time and generalization ability of the solution. Therefore, multi-objective evolutionary algorithms (MOEAs), which have long been recognized as suitable methods to handle MOPs, are not suitable for solving the multi-objective MCV scheduling problem. In this paper, a deep reinforcement learning (DRL) based method is proposed to efficiently solve the multi-objective MCV scheduling problem. The main contributions of our work are as follows:

**Contributions**

To promote the overall quality of the MCS, the charging scheduling problem is modeled as a multi-objective optimization problem. The focus is on minimizing the total service time (EV waiting time and charging time) and maximizing the charging benefit of all EVs simultaneously.

A framework for solving the multi-objective charging scheduling problem of MCV by DRL is developed, wherein a neighborhood-based parameter-transfer strategy is utilized to accelerate the training process. To the best of our knowledge, this is the first time that the DRL-based method is adopted to model and solve the MCV scheduling problem.

Simulations are performed to compare the proposed method with different evolutionary algorithms. The results clearly show that the trained model solves the MCV scheduling problem efficiently and effectively, and outperforms other algorithms in terms of solution convergence and solution diversity while requiring less running time.

**2. Related Work**

Much of the recent MCS-related literature concentrates on how to provide convenient services to EVs by dispatching MCVs. In Huang *et al*. [1], a mobile charging service was proposed for an urban scenario, where a queuing-based framework was applied to employ distinct service disciplines to cope with several charging requests. Discussions on the arrival patterns of charging requests were also provided. Zhang *et al*. [3] employed a deep neural network to predict the charging demand of the entire city. The predictions were then used to dispatch MCVs to high-demand areas in order to relieve the pressure of busy FCSs, and a teaching-learning based optimization approach was subsequently exploited to optimize MCV parking strategies. Further research applied the Lyapunov optimization theory to maximize MCV profits based on the randomness of EV users' arrival and the dynamic power supply [4]. Zhang *et al*. [5] aimed to present a novel mobile charging control mechanism by promoting charging reservations. Instead of focusing on service discipline on EV selection, the paper paid more attention to selecting appropriate MCV. The above papers describe the overall charging process and how to choose the next charging object. However, the weight ratios of waiting time and charging benefit are fixed, which may conflict with reality.

Recently, a DRL-based method has been proposed to deal with MOPs, which exhibits some encouragingly new characteristics compared to MOEAs (e.g., strong generalization ability, fast solving speed, and promising quality of the solutions) [6]. This paper adopted an end-to-end framework to solve MOPs by using DRL. Yang *et al*. [7] employed an actor-critic reinforcement learning (ACRL) algorithm to settle the charging scheme problem in dynamic wireless rechargeable sensor networks. [8] introduced a new scalable multi-objective deep reinforcement learning framework based on a deep Q-network. The proposed framework is highly modularized, which allows the integration of different DRL.

**3.** **System Model**

In this paper, we consider a city area comprising of a number of EVs, an MCV, and a depot. The MCV can be regarded as a kind of conventional van with plug-in batteries onboard and DC-DC converter, capable of providing level-3 charging service at any position. The depot is the public infrastructure that provides parking, charging, and battery swapping services for the MCVs. MCV starts from the depot and returns back to the depot when the remaining battery of MCV is not capable of satisfying the next charging service or when all the charging services are completed. Let denote the set of EVs distributed in the city. The MCV needs to find a tour of *n* EVs and the amount of charging electricity to each EV to minimize two cost functions simultaneously.

The charging sequence vector and the amount of charging vector are expressed as and , where represents the *j*-th visiting target which can be an EV or a depot, and means the amount of charging electricity that MCV charges to *k*-th EV. We use k = 0 to refer to the depot. refers to each EV.

For each visiting target, we have the following constraint:

which means that the MCV should not visit the same target two times except the depot. The constraints on minimum and maximum energy to be charged are expressed as:

where and are the state of charge (SOC) and the battery capacity of *k*-th EV, respectively.

represents the minimal amount of electricity that *k*-th EV demands. In this paper, we assume that EV submits a minimal amount of required electricity rather than the actual amount of required electricity. As a result, the actual amount of energy that the $k$-th EV will receive is decided by MCV, which should be greater than or equal to but lower than or equal to .

The MCV first aims at maximizing the average charging benefits of all EVs. For the convenience of subsequent presentation, this objective can also be written in the form of minimizing by expressing as:

When considering the charging benefit, we introduce the diminishing marginal effect that as the amount of charging increases, the rate of increase in benefit will decrease because the EV may not need too much energy and find the energy excessive. We can see that the benefit function is a non-negative, monotonically increasing, and concave function.

The MCV also considers minimizing the average waiting time of all EVs, which can be written as:

where and are the time of arrival and the charge completion time of *k*-th EV, respectively. In which, the calculation of is as follows:

The waiting time consists of two parts: the travel time and the charging time. represents the distance between and . In this paper, Manhattan distance is utilized to calculate the travel distance between two points. *v* and refer to the speed and the charging rate of MCV, respectively.

The aim of the multi-objective MCV scheduling problem is to determine the charge sequence and the actual amount of charging to each EV that maximize the average charge benefits and minimize the average waiting time simultaneously subject to the MCV operation constraints, which can be expressed as follows:

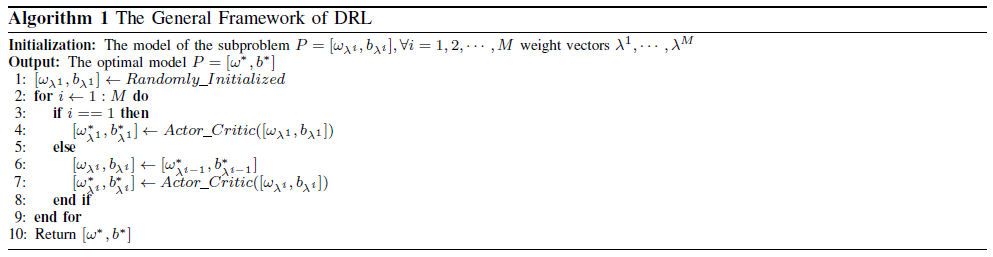
**4.** **The Optimization Algorithm**

In this section, we propose an effective DRL-based framework to solve the charging scheduling problem. First, the decomposition strategy is adopted to decompose the MOP into multiple subproblems modeled as an independent neural network. Then the model parameters of each subproblem are optimized in a collaborative manner according to the neighborhood-based parameter transfer strategy [6] and the Actor-Critic algorithm.

**The General Framework**

Decomposition is a simple but effective way to design the MOP algorithm. First, a set of uniformly spread weight vectors is required, where . Then the original problem is converted into *M* scalar optimization subproblems, like Eq. (6). After solving each scalar subproblem, the Pareto Front (PF) can be obtained.

The *M* optimization subproblems are solved collaboratively by the neighborhood-based parameter transfer strategy, which can accelerate the training process of neural networks. It is obvious that if the weight vectors of two subproblems are close enough, these two neighboring subproblems will have similar optimal solutions. Thus, the knowledge of one subproblem can help the training process of its neighboring subproblems. In this work, as the subproblem is modeled as a neural network, the parameters of the network model of the i-th subproblem can be presented as . Assume that represents the parameters of the (i-1)-th neural network model that have been optimized already. Then the network parameters of the (i-1)-th subproblem are set as the starting point of the i-th subproblem. Similarly, the network parameters of the i-th subproblem can also be utilized for the training of the next subproblem. As a result, the neighborhood-based parameter transfer strategy saves a tremendous amount of time for training the *M* subproblems. The general framework is presented in Algorithm 1.



**Actor-Critic Algorithm**

After decomposing the charging scheduling problem into a set of subproblems and solving each subproblem based on the aforementioned framework, each subproblem is solved by means of DRL. In this section, a modified Pointer Network is introduced to model the subproblem, and an Actor-Critic algorithm is utilized for the training.

First, we introduce the input structure of the neural network. Let the set of input be , where *N* refers to the number of EVs. Each is represented by a sequence of tuples , where and are the static and dynamic elements of the input, respectively. It is noteworthy that dynamic elements of each input are allowed to change between the decoding steps. For example, during the charging process, the demand for an EV becomes 0 after MCV coming and providing the charging service. However, the static elements, corresponding to EV *i*'s location, remaining battery, and battery capacity, do not change. can be viewed as a vector of features that represents the state of input *i* at time *t*. The set of all input states at time *t* is denoted as .

The output of the model is a permutation of the EVs and depot , where *T* refers to the length of decision steps. At every decoding time *t*, points to one of the EVs or depots, determining the next visiting target. The process continues until the termination condition is satisfied when the demand of all EVs has been satisfied. This process then generates a variable-length sequence . The reason that the length is variable is that MCV may have to go back to the depot to refill when necessary. The probability chain rule is utilized to decompose the probability of generating sequence *Y* as follows:

The inputs are updated every time when an EV is visited. A modified Pointer network is utilized to model as Eq. (7). The basic structure is s Sequence-to-Sequence model, which maps one sequence to another. This model consists of two RNN networks, termed encoder network and decoder network. The encoder network encodes the input into a code vector that contains knowledge of the input. Then decoder network decodes the code vector to the desired sequence. However, we argue that the RNN encoder is not necessary because the order of EV locations in the input does not affect the outcome. For example, if we swap the order of two EVs, the input information should be the same as the original inputs. Therefore, the embedded layer is utilized to encode the inputs to a vector instead of an RNN. The method also helps reduce the computational complications without decreasing the efficiency.

Different from the encoder, an RNN is still required to model the decoder network because the previous steps help to decide . The RNN decoder hidden state stores the knowledge of previously selected visiting targets . Then coping with the encoding of the input are utilized together to calculate the conditional probability . The index of the next visiting target can be calculated by the attention mechanism. The attention mechanism is a structure for addressing the inputs, which extracts the relevant information from the inputs. The most relevant one can be selected as the next visiting target. The calculation is as follows:

where *v*, , and are trainable parameters. For each EV and depot, its is computed according to and the encoder hidden state . In this paper, the greedy decoder is utilized to select the visiting target with the largest probability. However, the next visiting target is assigned according to the sampling from probability distribution during the training process. The proposed model is shown in Fig.1.

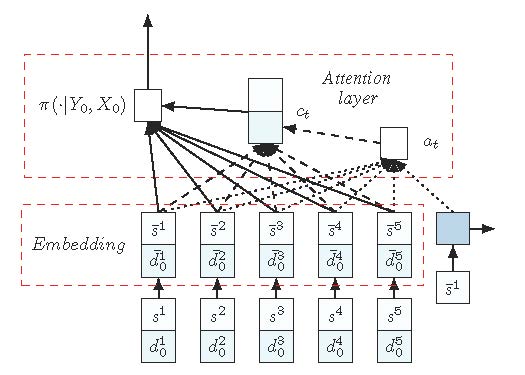


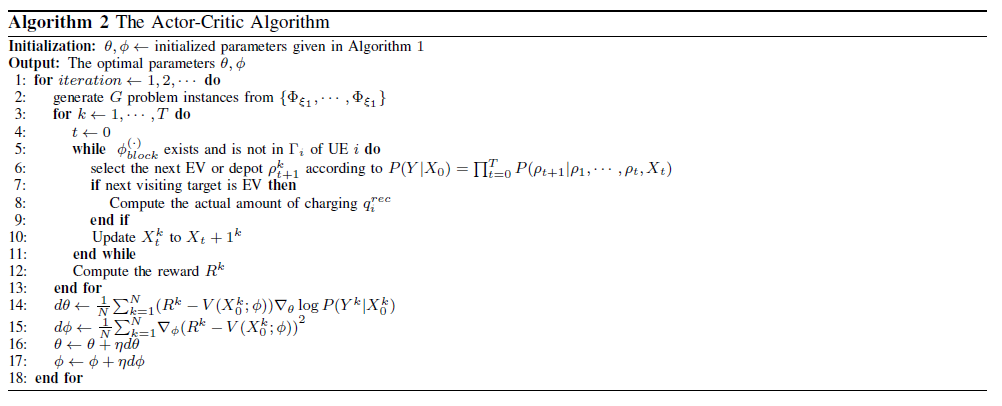
Figure 1. Neural network model. The embedding layer maps the inputs to a high-dimensional vector space. The, the RNN hidden state and the embedded input produce a probability distribution over the next input using attention mechanism.

In addition to the charging sequence, our model ought to decide the actual amount of charging, which lies between minimum amount and battery capacity minus SOC of EV. Assume that the next serving target is EV *i*. By solving the following optimization problem, the actual amount of charging that EV *i* will receive can be determined:

where and refer to the weights of the charging benefit and the waiting time, respectively. *l* represents the number of remaining EVs. denotes the charging benefit of EV *i* related to its receiving electricity, battery capacity, and SOC, and represents the overall waiting time of the remaining EVs. This is a function that increases first and then decreases. The stagnation point is . According to , the solution can be obtained as follows:

To train the network, we use the well-known Actor-Critic method. In principle, the algorithm contains two networks: (i) an actor network, which is the modified Pointer Network in this work, computes the probability distribution for choosing the next target, and (ii) a critic network that estimates the reward with a given problem state. Algorithm 2 presents the training procedure.

The training is conducted in an unsupervised way. The experience instances are generated from distribution . refers to different input features of the EVs, e.g., the EV locations and its demand. *G* instances are sampled from for training the actor and critic networks with parameters and , respectively. For each instance, the actor network produces the permutation of EVs and the reward with the current parameter . Then policy gradient approaches use an estimate of the gradient of the expected return to improve the policy in Line 16 iteratively. By reducing the difference between the observed rewards and the approximated rewards, the critic network is updated in line 17 of Algorithm 2.



**5.** **Performance Evaluation and Analysis**

In this section, we compare our framework with solutions obtained from classical multi-objective evolutionary algorithms (MOEAs), including NSGA-II and MOEA/D. The effects of different training sizes are also evaluated.

**Parameters and Settings**

The charging service area is set as a square with an area of 30 km2 in which EVs are randomly distributed. The depot is at the center of the map. We run our tests with 10 and 20 EVs. The SOC and the battery capacity of EVs are 5 kWh and 50 kWh, respectively. The minimal amount of electricity that EV demands are randomly distributed between 5 kWh and 15 kWh. The battery charge capacity and the average driving speed of the MCV are assumed to be 200 kWh and 30 km/h with a charging rate of 60 kW. The number of iterations for NSGA-II and MOEA/D is set to 500, 1000, 2000, and 4000, with a population size of 100. The number of subproblems for DRL is set to 40 when compared with MOEAs. In addition, the charging benefits are set as a constant number minus itself for better training effect.

**Results and Discussions**

The performance of the PFs obtained by all the compared algorithms is shown in Fig. 2. All of the compared algorithms show a great ability of convergence. It is observed that the solutions created by our method perform much better than NSGA-II and MOEA/D in terms of both convergence and diversity. By increasing the number of iterations, NSGA-II and MOEA/D show a better ability to find a shorter path which can be reflected in some subproblems of high weight on waiting time. However, the large number of iterations cannot lead to a better performance of charging benefits.

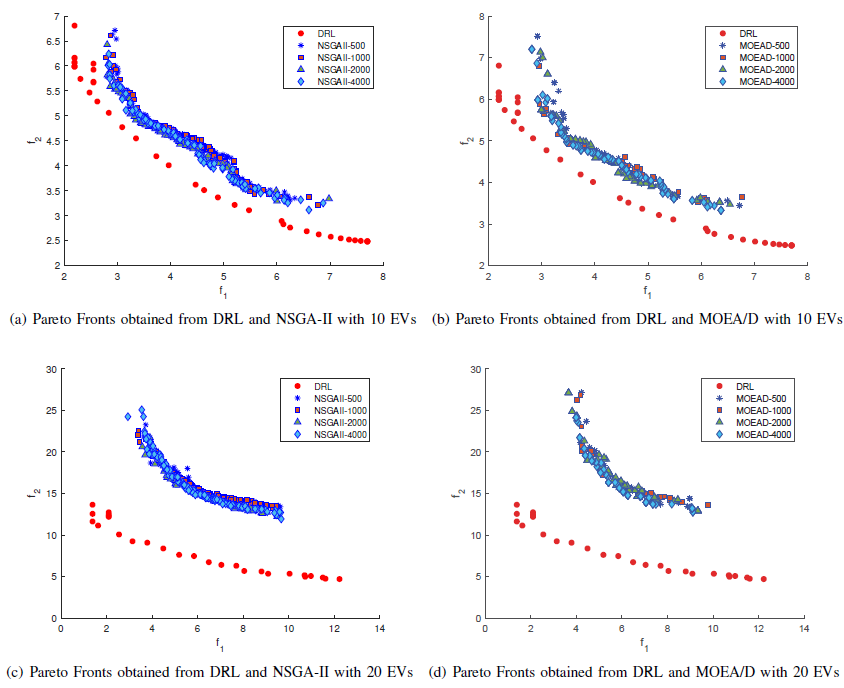


Figure 2. A randomly generated 10-EV bi-objective MCV scheduling problem instance: the PF obtained using our method (trained using 10-EV instances) in comparison with NSGA-II and MOEA/D of 500, 1000, 2000, and 4000 iterations.

As shown in Fig.2-a and Fig. 2-b, a larger number of iterations even can lead to less charging benefits in some subproblems. In addition, the large number of iterations will increase the amount of computing time. The computing times of NSGA-II of 500, 1000, 2000, and 4000 iterations are 3.53, 6.45, 12.75, and 25.3 seconds, and the computing times of MOEA/D of 500, 1000, 2000, and 4000 iterations are 15.48, 30.81, 61.41, and 127.91 seconds, respectively. In contrast, our method requires only 3.2 seconds from loading the model to achieve the results.

Overall, the experimental results show the effectiveness of our method on solving bi-objective MCV scheduling problems. The trained model has learned how to select the next EV with the given information of EVs and previous choice. In contrast, NSGA-II and MOEA/D fail to converge by increasing the number of iterations.

We further evaluate the performance of our method on 20-EV instances. The model is still trained on 10-EV instances, and it is utilized to approximate the PF of the 20-EV instance. The results in Fig.2-c and Fig.2-d show that DRL significantly outperforms other algorithms. There is an significant gap in performance between DRL and two MOEAs of different iterations.

In addition, the running time of DRL is still much lower than those of two MOEAs for 20-EV instances. It takes 5.88 seconds for DRL to achieve the PF. However, it requires 3.68, 7.19, 14.25, and 27.90 seconds for NSGA-II for running 500, 1000, 2000, and 4000 iterations, respectively. The computing time of MOEA/D is even higher, which takes 16.10, 34.11, 68.57, and 137.32 seconds to reach an acceptable level of convergence.

Observed from the experimental results, we can conclude that our method is able to handle the MCV scheduling problem both effectively and efficiently. A better balance between the waiting time and charging benefits is guaranteed when compared to MOEAs. Moreover, once the trained model is available, it can apply to newly encountered problems without retraining. In other words, its performance is less affected when confronted with different numbers of EVs.

**6.** **Conclusions**

This paper provided a new way of solving multi-objective MCV scheduling problems by means of DRL. By controlling the charge sequence and actual amount of charging electricity, the aim of the scheduling was to minimize the average waiting time and maximize the average charging benefit of EVs simultaneously. First, we decomposed the MOP into a set of scalar optimization subproblems, wherein a neighborhood-based parameter transfer strategy was adopted to accelerate the training process. Then an actor-critic algorithm and a modified pointer network were adopted to solve each subproblem. Experimental results demonstrate that the proposed method solves the MCV scheduling problem efficiently and effectively, as well as outperforms NSGA-II and MOEA/D in terms of solution convergence, solution diversity, and computing time. These results, as we believe, shall provide useful insights helping the deployment of MCVs and the wide adoption of EVs.

**References**

[1] S. Huang, L. He, Y. Gu, K. Wood, and S. Benjaafar, “Design of a mobile charging service for electric vehicles in an urban environment,” IEEE Trans. Intell. Transp. Syst., vol. 16, no. 2, pp. 787–798, 2014.

[2] K. Clement-Nyns, E. Haesen, and J. Driesen, “The impact of charging plug-in hybrid electric vehicles on a residential distribution grid,” IEEE Trans. Power Syst., vol. 25, no. 1, pp. 371–380, 2009.

[3] H. Zhang, B. Jin, J. Li, J. Gao, J. Zhao, M. Hou, G. Yu, and H. Liu, “Optimized scheduling for urban-scale mobile charging vehicle,” in World Symposium on Communication Engineering. IEEE, 2019, pp. 164–172.

[4] H. Chen, Z. Su, Y. Hui, and H. Hui, “Dynamic charging optimization for mobile charging stations in internet of things,” IEEE Access, vol. 6, pp. 53 509–53 520, 2018.

[5] X. Zhang, Y. Cao, L. Peng, J. Li, N. Ahmad, and S. Yu, “Mobile charging as a service: A reservation-based approach,” IEEE Trans. Autom. Sci. Eng., vol. 17, no. 4, pp. 1976–1988, 2020.

[6] K. Li, T. Zhang, and R. Wang, “Deep reinforcement learning for multiobjective optimization,” IEEE Trans. Cybern., pp. 1–12, 2020.

[7] M. Yang, N. Liu, L. Zuo, Y. Feng, M. Liu, H. Gong, and M. Liu, “Dynamic charging scheme problem with actor–critic reinforcement learning,” IEEE Internet Things J., vol. 8, no. 1, pp. 370–380, 2021.

[8] T. T. Nguyen, N. D. Nguyen, P. Vamplew, S. Nahavandi, R. Dazeley, and C. P. Lim, “A multi-objective deep reinforcement learning framework,” Eng. Appl. Artif. Intell., vol. 96, p. 103915, 2020.

**Hui Wang** received the B.E. degree from the College of Computer Science and Technology, Jinling Institute of Technology, Nanjing, China, in 2019. He is currently pursuing the Master degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include internet of electric vehicles, machine learning in smart grid, etc.

**Ran Wang** (Member, IEEE) received the B.E. degree in electronic and information engineering from Honors School, Harbin Institute of Technology (HIT), China, in July 2011, and the Ph.D. degree in computer science and engineering from Nanyang Technological University (NTU), Singapore, in April 2016. He is currently an Associate professor at the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), and at the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, China. He was a Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), from October 2015 to August 2016. He has authored or coauthored over 40 papers in top-tier journals and conferences. He received the Nanyang Engineering Doctoral Scholarship (NEDS) Award in 2011, and the Singapore and Innovative and Entrepreneurial Ph.D. Award of Jiangsu Province, China in 2017. His current research interests include intelligent management and control in smart grid, network performance analysis, and the Internet of electric vehicles.

**Hu Xu** received the B.E. degree from the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2019. He is currently pursuing the Master degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include reinforcement learning, its application, etc.

**Kun Zhu** (Member, IEEE) received the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2012.,He is currently a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He is also a Jiangsu Specially Appointed Professor. His research interests include intelligent optimization algorithms, 5G/B5G, resource management, applied game theory and optimization, and self-organization networks.,Dr. Zhu has been serving as a TPC Member for IEEE INFOCOM, GLOBECOM, IEEE International Conference on Communications (ICC), and so on. He won several research awards, including the IEEE WCNC 2019 “Best Paper Award” and the ACM Rising Star Chapter Award in 2018. He has been serving as the Program Chair of the 4th MLICOM Conference. He is also a PI of more than ten research projects. He has served as a guest editor for several journals. He is also an Editor of Computer Communications.

**Changyan Yi** (Member, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada, in 2018.,From September 2018 to August 2019, he worked as a Research Associate with the University of Manitoba. Since September 2019, he has been with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, and the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, where he is currently a Professor. His research interests include mechanism design, game theory, queueing theory and their applications in various wireless networks, including edge/fog computing, IoT, and 5G and beyond.,Prof. Yi was awarded the Chinese Government Award for Outstanding Students Abroad in 2017, the University of Manitoba Graduate Fellowship for 2015–2018, and the IEEE ComSoc Student Travel Grant for IEEE Globecom 2016.

**Dusit Niyato** [M' SM'15, F'17] (dniyato@ntu.edu.sg) is currently a professor in the School of Computer Science and Engineering, Nanyang Technological University. He received his B.Eng. from King Mongkuts Institute of Technology Ladkrabang, Thailand, in 1999, and his Ph.D. in electrical and computer engineering from the University of Manitoba, Canada, in 2008. His research interests are in the areas of energy harvesting for wireless communication, the Internet of Things, and sensor networks.

**Predicting Crowd Flow: A survey from perspective of TODO**

*Shuyu Wang1,3, Yuhang Xu2, Yan Lyu2, Weiwei Wu2*

1 *School of Cyber Science and Engineering, Southeast University, Nanjing, China*

2 *School of Computer Science and Engineering, Southeast University, Nanjing, China*

3*School of Information Engineering, Xizang Minzu University, Xianyang, China*

shywang@seu.edu.cn, yuhang\_xu@seu.edu.cn, lvyanly@seu.edu.cn, weiweiwu@ seu.edu.cn

**Abstract**: Recently, more and more attention has been given to public safety in crowded scenes, such as subway stations, railway stations, shopping malls, concerts, and stadiums. Under this background, crowd analysis has attracted more interests as a powerful tool to ensure public safety. Crowd analysis involves a range of specific tasks, including crowd counting and density estimation, human trajectory prediction, and crowd flow prediction. A comprehensive summary of these tasks is provided in this paper. The results of our survey suggest that the crowd flow prediction in dense indoor scenes is still under-researched. We present the challenges for crowd flow prediction in dense indoor scenes and the promising future research directions.

**Index Terms**: crowd analysis, human trajectory prediction, crowd counting and density estimation, crowd flow prediction, indoor dense scene.

**1. Introduction**

With the development of technology and economy, cities are becoming more and more populated, while the number of densely populated scenes is also dramatically increasing. Densely populated scenes are more prone to public safety incidents causing serious damage, such as the severe stampede caused by overcrowding on the Bund in Shanghai on New Year's Eve 2014. Crowd analysis is a powerful tool that empowers city managers to accurately analyze the movement of crowds and take early action to avoid overcrowding[1][2]. Crowd analysis is the practice of interpreting data on the natural movement of groups or objects[3]. Crowd analysis aims to capture patterns of movement and trends in crowds by tracking them. So far, it has already been successfully applied in some typical urban crowd scenes, such as subway stations[4][5], railway stations[6], airport terminal[7], shopping malls[8]. Meanwhile, crowd analysis is also essential for interior design. From a crowd safety and convenience perspective, crowd analysis can effectively reveal design flaws and layout problems in the design of interior space. The results of the analysis can be applied to improve the design of future public buildings to provide more convenient services to people in a safer manner[9][10].

Crowd flow prediction is a key step for achieving accurate crowd analysis in densely populate indoor scenes. However, there has been few studies on the crowd flow prediction in densely populated indoor scenes. Fortunately, there are a great deal of established crowd analysis techniques that can provide insight and technical support for solving this problem. In this paper, we categorize the crowd analysis techniques that are closely related to crowd flow prediction in densely populated indoor scenes into three categories, namely, crowd counting and density estimation, human trajectory prediction, and crowd flow prediction. Crowd counting assesses the number of specific objects in a target region based on density map estimation methods. Considering its ability to model and characterize crowds, the results of crowd counting can help us to obtain accurate flow data in dense indoor scenes. Human trajectory prediction aims to predict the future trajectories of target pedestrians based on their past movement trajectories which takes social interaction into account. At the same time, social interaction is also an important factor closely related to crowd flow prediction in densely populated indoor scenes. Therefore, the accuracy of crowd flow prediction in indoor crowded scenes may be effectively improved by drawing on the processing mechanism of social interaction in human trajectory prediction. Crowd flow prediction for city-scale networks, such as ride-hailing systems and bike-sharing systems, has been well-studied, but crowd flow prediction in indoor scenes is almost non-existent. We believe that the mechanisms used in existing methods for capturing spatial-temporal relationships contained in historical data can also be applied for indoor scenario prediction tasks. Therefore, it is necessary to provide a systematic summary of representative works on these three types of techniques.

There have been many comprehensive surveys for crowd counting and density estimation[11][12], human trajectory prediction[13][14], and crowd flow prediction[15]. In [11] CNN-based methods for crowd counting and density estimation are categorized based on the training process and network property. In [12], Gao et al. also give a survey of CNN-based methods, in which methods are classified according to a different set of criteria, i.e., network architecture, reference manner, supervision form, domain. In [13], the authors propose a taxonomy classifying human trajectory prediction techniques based on application domains, including service robotics, self-driving vehicles, and advanced surveillance systems. The latest work in [14] focuses on tackling the challenge of modeling social interactions between pedestrians in the same crowd, and the authors presented an in-depth analysis of the design of interaction modules proposed by others and introduced two domain-knowledge inspired interaction models. In [15], Xie, et al. conduct a comprehensive overview of the recent development methods for urban spatio-temporal flow prediction and existing works are classified as statistics-based, traditional machine learning-based, deep learning-based, reinforcement learning-based, and transfer learning-based methods. All these review papers above focus on a categorical summary of existing methods, neglecting to explore the implications of current methods for future research. To the best of our knowledge, few works address the problem of crowd flow prediction in dense indoor scenes. Therefore, in this paper, we provide a concise summary of the above three types of techniques in the context of the crowd flow prediction in dense indoor scenes.

The contributions in this paper are mainly in the following three folds:

* From the macroscopic perspective, we summarize the three research fields of crowd analysis and the challenges in each field. The mechanisms for modeling and representation of crowds or human in these three research fields have a relatively mature research basis, and they can help to predict crowd flows in indoor environment with Camera Videos.
* From a TODO perspective, we propose feasible and promising future research directions, especially the prediction task of indoor dense crowd flow. Indoor crowd flow prediction is of great significance to public safety and interior design problem discovery, and we also qualitatively analyze the challenges involved.
* We also identify three challenging problems that are critical for predicting indoor dense crowd flow, namely, (1) how to establish a representation method for large-scale crowds, (2) how to model social interactions efficiently so that crowd movement patterns can be accurately captured, (3) how to consider the impact of physical environment constraints on crowd flow.

The rest of the paper are organized as follows. Section 2 introduces the methods of crowd counting and density estimation based on CNN from patch-based methods and whole-image-based methods and summarizes the characteristics of research field. Section 3 describes the research on human trajectory prediction, in which different factors considered in social interaction. Section 4 conducts the current situation and methods of crowd flow prediction from city-scale and indoor scenes. Section 5 summarizes the challenges of indoor dense crowd flow prediction. Finally, the conclusion is concluded in Section 6.

**2. Crowd counting and density estimation**

Estimating crowd counting and density maps from images has a wide range of applications, such as video surveillance, public safety, and urban planning. However, the task remains challenging due to factors such as occlusion, uneven density, image perspective and scale variability. Extensive application prospects and challenging coexistence, as well as features extracted based on CNN are better, therefore, this field has attracted more and more researchers to focus on and solve the problem of dense crowd analysis combined with CNN methods. **Like some other surveys, we classify the CNN-based approaches based on the different training manners into the following two categories: patched-based inference and whole image-based inference.**

Patch-based methods is required to train using patches randomly cropped from the image. In the test phase, using a sliding window runs over the whole test image, and estimation are obtained for each window and finally aggregated to obtain total count in the image. Wang[16] is the first to propose that CNN can be used for crowd counting estimation in images, which can be used to count pedestrians from extremely dense crowd images. They adopted AlexNet[17] in their network, in which 4096 neurons of the last layer were replaced by a single neuron layer for crowd counting. In addition, to reduce the influence of false backgrounds such as buildings and trees in the images, negative samples were added to the training data, whose label was set to 0. Wang's[16] method relies on the training data. When the training data do not contain a new scene, the performance of crowd counting estimation in the new scene is greatly compromised. In addition, there are also some problems such as limited annotated dense crowd data set and large tasks in the annotation process. For this reason, Zhang[18] proposed a method of cross-scene crowd counting and density estimation. Because crowd counting and density estimation are correlated, better local optimization can be obtained by alternating training of the two tasks. To apply the training network to the new scene, this method solves the problem that the new scene does not have additional labeling information, and the pre-trained network can be fine-tuned to adapt to the task of counting and density estimation in the new scene.

Patch-based methods always neglect global information and burden much computation cost due to the sliding window operation. Thus, the whole image-based methods usually take the whole image as input, and output corresponding density map and crowd counting, which is more convergence but may lose local information sometimes. Ding[19] noted that the CNN model with complex structure could not deal with multi-scale problems well, and the existing methods[20] mostly focused on the accuracy of crowd counting, while ignoring the accuracy of density distribution. To this end, the authors propose a novel symmetric CNN structure to train the end-to-end network, which can combine features from different layers to predict the crowd density map. Because of the improved feature utilization, the method can obtain more accurate density map. Changeable environment, large-range number of people cause the current methods cannot work well. In addition, due to the scarce data, many methods suffer from over-fitting to a different extent. Wang[21] introduced spatial fully convolutional network (SFCN), to remedy the above two problems, firstly, they develop a data collector and labeler, which can generate the synthetic crowd scenes and simultaneously annotate them without any manpower. Secondly, they propose two schemes that exploit the synthetic data to boost the performance of crowd counting.

The above method solves the problem of crowd counting and density map estimation in dense crowd scene, and the results can be used as the basis for the task decision of crowd analysis. The characteristic of this research is that the input is the current video frame, and its output is the crowd counting and density map estimation at the current moment. **The outputs of these research fields are helpful for indoor dense crowd flow prediction.**

**3. Human trajectory prediction**

With growing numbers of intelligent autonomous systems in human environments, the ability of such systems to perceive, understand, and anticipate human trajectory becomes increasingly important. The challenge of accurately predicting human trajectory stems from the complexity of human behavior and the diversity of internal and external stimuli. Human trajectory prediction affects many applications, such as autonomous vehicles and service robots.**Social interaction is a key part of trajectory prediction, in which different methods have different considerations.**

In some earlier studies, the social interaction factors of future human trajectories only considered other pedestrians in the environment, excluding the physical environment. Social-LSTM[22] and social-GAN[23] proposed by Feifei Li's team are typical representatives of solving trajectory prediction tasks, which are different from Social Force Model[24] with artificially designed social interaction features. In social-LSTM and social-GAN, social interaction is the feature extracted through deep learning. Social-LSTM transforms the trajectory prediction task into a sequence generation problem. In the spatial dimension, it considers the future trajectory to be affected by nearby pedestrians, and in the time dimension, it predicts the trajectory path in the short-term future through historical trajectory data. Social-GAN is an improvement based on social-LSTM. Firstly, the future trajectory is not only affected by neighbors, but also the influence of all pedestrians in the scene should be considered. Secondly, adversarial loss is used in the training network, which makes the prediction result more reasonable rather than feasible.

Unlike these studies[22][23], future trajectories are influenced not only by other pedestrians in the environment, but also by the physical environment or the goals. Amir[25] proposed a trajectory prediction network named Sophie that considers the interaction of various agents and constraints of physical scenes at the same time. The network is GAN structure based on LSTM. Considering that there are many feasible paths for future trajectories, the prediction result is the trajectory distribution of each agent. BicycleGAN[26] introduces a noise encoder to learn the mapping relationship between noise and output. By learning BicycleGAN, Kosaraju[27] proposes a potential spatial encoder that can be used to explain the generation of multimodal agents trajectories. The social interaction among agents is constructed into a graph, in which nodes represent agents and edges represent social interaction between agents. Higher edge weights correspond to more important interactions. By keeping the graph completely connected, social interaction can be considered from a local or global perspective. Y-net[28] models the epistemic un-certainty through multimodality in long term goals and the aleatoric un-certainty through multimodality in waypoint & paths to predict trajectories with prediction horizons upto a minute.

The premise of the above research is to be able to redirect pedestrians (Re-identification,ReID) in different images but ReID itself is a very challenging task. Li[29] proposed ReID method that contains less pedestrian information in low-resolution images. Huang[30] solved ReID's difficulty that the pedestrian images captured by surveillance cameras usually contain various degradation due to the influence of weak light. Huang[31] and Hou[32] solved the ReID problem caused by partial occlusion.

Based on the above research, it is extremely difficult for dense crowds to track a single pedestrian or track every pedestrian is computationally expensive and redundant for some particular applications. For example, some applications only need to know where and when crowd congestion occurs, so that unsafe events caused by high crowd density can be avoided. In the interior design application, the simulation prediction model outputs the possible frequent congestion location, to improve the design scheme. **In these research fields, social interaction is an important factor in trajectory prediction. It may be possible to improve crowd flow prediction by considering social interactions in dense crowd scenarios.**

**4. Crowd flow prediction**

The characteristics of crowd dynamic change are very important, which can also be used as an important basis for crowd analysis and decision-making. In intelligent transportation systems, urban planning, crowd management, public safety, interior design and other applications, the research of crowd flow prediction has a wide application prospect. **The crowd flow prediction of city-scale is more, while indoor crowd flow prediction is less.**

There are a lot of studies on city-scale predictions. With the popularization of the Internet of Things, there are more and more taxi and shared bike track data, public transport check-in data and location-based social network check-in data. How to use the above data has important economic significance in intelligent transportation, urban planning, crowd management, public security and other application fields. The classical time series prediction model ARIMA[33] has a far-reaching impact on urban traffic prediction tasks. However, since this type of model only considers the law in time dimension, and fails to take into account the law at the same time and different space, its prediction results are not reliable. With the rapid development of deep learning, convolutional neural network (CNN), graph neural network (GNN), recurrent neural network (RNN) and adversarial neural network (GAN) have achieved great success in various machine learning tasks. Yao[34] proposed traffic prediction model named STDN, in which the method of rasterizing cities makes it more suitable for capturing spatial correlation through CNN. Du[35] proposed DTCNN by combining GNN and RNN to accurately predict traffic demand. Wang[36]proposed the generative adversarial network model named SEQST-GAN for multi-step spatial-temporal flow prediction by combining GAN and RNN, in which the adversarial loss adopted by GAN network made the prediction result more consistent with the reality. Inspired by the widespread existence of WLAN infrastructure, He[37] proposed crowd-flow identification (CFid), which realizes crowd flow identification through fine-grained temporal and spatial signal fusion. The mobile phone data that this method relies on may cause privacy leaks.

Different from city-scale crowd prediction tasks, crowd flow in complex indoor environments refers to the cumulative distribution of crowd dynamics in the environment at a specific time, a concept that has been fully studied because it is critical to analyze and manage crowd movement in confined spaces[38]. Sohn[10] proposed a long-term crowd flow prediction framework that considers environmental complexity and crowd density at the same time, the problem they solve is to efficiently and timely output all possible crowd flow distributions in the environment, in order to overcome the difficulty of obtaining real data, they developed a program to generate the required input data. The simulation results of efficient and real-time prediction of crowd flow can provide reliable decision-making basis for indoor layout design. Niu[39] proposed the problem of crowd distribution prediction in dense crowd scenes, whose goal is to predict the future distribution of crowds without any identification. This method does not distinguish between indoor and outdoor environments and focuses on forecasting future distribution of dense crowds.

There are many methods to predict city-scale crowd flow, but few research on prediction of indoor dense crowd flow. A lot of data can be used to urban crowd flow prediction, while the data of indoor dense scene are mainly surveillance videos. **The mechanisms of feature extraction of urban spatial-temporal data can be learned to deal with indoor crowded scene.**

**5. Challenges**

In summary, there are few studies on crowd flow prediction for indoor dense crowd scenes, some similar studies did not highlight the characteristics of indoor dense crowd scenes, however, accurately predicting the location of indoor crowds where congestion occurs and foreseeing the emergence of high-density crowds is of practical significance for applications such as public safety and the discovery of indoor passage problems. At the same time, through the method of machine learning, how to dig out the change rules of indoor dense crowd is very challenging, which makes this problem has theoretical research value. The main challenges in the study of indoor dense crowd flow prediction are as follows: (1) Large-scale crowds are different from the modeling of sparse crowds, because the continuous movement of a single pedestrian cannot be tracked due to occlusion and other reasons, how to establish a representation method for large-scale crowds is extremely challenging. (2) Various interactions between dense crowds occur from time to time, which are also important factors affecting crowd flow. How to model social interactions scientifically affects whether crowd movement patterns can be accurately captured. (3) Changes in crowd movement patterns will be restricted by the physical environment, and how to consider the impact of physical environment constraints on crowd flow is also very important. Based on the above considerations, the indoor dense crowd flow prediction has important research value and research significance.

**6. Concluding Remarks**

In this paper, we make a survey from three categories, including crowd counting and density estimation, human trajectory prediction, and crowd flow prediction. Although not all research methods were covered, the analysis selected representative papers with distinctive features. We combed through the relationship between these three categories and indoor dense crowd flow prediction. At the end of this paper, we discuss the challenges of indoor dense crowd flow prediction, and analysis its research value and significance. Based on the current research results, it provides feasible and meaningful research direction for crowd flow prediction in the future.

**References**

1. Abdelghany A, Abdelghany K, Mahmassani H, et al. Modeling framework for optimal evacuation of large-scale crowded pedestrian facilities[J]. European Journal of Operational Research, 2014, 237(3): 1105-1118.
2. Dong H, Zhou M, Wang Q, et al. State-of-the-art pedestrian and evacuation dynamics[J]. IEEE transactions on intelligent transportation systems, 2019, 21(5): 1849-1866.
3. Gamma. Crowd analysis. (2015).URL: <https://en.wikipedia.org/wiki/Crowdanalysis>.
4. Gu J, Jiang Z, Fan W D, et al. Real-time passenger flow anomaly detection considering typical time series clustered characteristics at metro stations[J]. Journal of Transportation Engineering, Part A: Systems, 2020, 146(4): 04020015.
5. King D, Srikukenthiran S, Shalaby A. Using simulation to analyze crowd congestion and mitigation at Canadian subway interchanges: case of Bloor-Yonge Station, Toronto, Ontario[J]. Transportation Research Record, 2014, 2417(1): 27-36.
6. Zhou Q, Zhang J, Che L, et al. Crowd counting with limited labeling through submodular frame selection[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 20(5): 1728-1738.
7. Liu M, Jiang J, Guo Z, et al. Crowd counting with fully convolutional neural network[C]//2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018: 953-957.
8. Liu J, Gao C, Meng D, et al. Decidenet: Counting varying density crowds through attention guided detection and density estimation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 5197-5206.
9. Lu L, Chan C Y, Wang J, et al. A study of pedestrian group behaviors in crowd evacuation based on an extended floor field cellular automaton model[J]. Transportation research part C: emerging technologies, 2017, 81: 317-329.
10. Sohn S S, Zhou H, Moon S, et al. Laying the foundations of deep long-term crowd flow prediction[C]//European Conference on Computer Vision. Springer, Cham, 2020: 711-728.
11. Sindagi V A, Patel V M. A survey of recent advances in cnn-based single image crowd counting and density estimation[J]. Pattern Recognition Letters, 2018, 107: 3-16.
12. Gao G, Gao J, Liu Q, et al. Cnn-based density estimation and crowd counting: A survey[J]. arXiv preprint arXiv:2003.12783, 2020.
13. Rudenko A, Palmieri L, Herman M, et al. Human motion trajectory prediction: A survey[J]. The International Journal of Robotics Research, 2020, 39(8): 895-935.
14. Kothari P, Kreiss S, Alahi A. Human trajectory forecasting in crowds: A deep learning perspective[J]. IEEE Transactions on Intelligent Transportation Systems, 2021.
15. Xie P, Li T, Liu J, et al. Urban flow prediction from spatiotemporal data using machine learning: A survey[J]. Information Fusion, 2020, 59: 1-12.
16. Wang C, Zhang H, Yang L, et al. Deep people counting in extremely dense crowds[C]//Proceedings of the 23rd ACM international conference on Multimedia. 2015: 1299-1302.
17. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25: 1097-1105.
18. Zhang C, Li H, Wang X, et al. Cross-scene crowd counting via deep convolutional neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 833-841.
19. Ding X, He F, Lin Z, et al. Crowd density estimation using fusion of multi-layer features[J]. IEEE Transactions on Intelligent Transportation Systems, 2020.
20. Sindagi V A, Patel V M. Generating high-quality crowd density maps using contextual pyramid cnns[C]//Proceedings of the IEEE international conference on computer vision. 2017: 1861-1870.
21. Wang Q, Gao J, Lin W, et al. Learning from synthetic data for crowd counting in the wild[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 8198-8207.
22. Alahi A, Goel K, Ramanathan V, et al. Social lstm: Human trajectory prediction in crowded spaces[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 961-971.
23. Gupta A, Johnson J, Fei-Fei L, et al. Social gan: Socially acceptable trajectories with generative adversarial networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 2255-2264.
24. Helbing D, Molnar P. Social force model for pedestrian dynamics[J]. Physical review E, 1995, 51(5): 4282.
25. Sadeghian A, Kosaraju V, Sadeghian A, et al. Sophie: An attentive gan for predicting paths compliant to social and physical constraints[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 1349-1358.
26. Zhu J Y, Zhang R, Pathak D, et al. Multimodal Image-to-Image Translation by Enforcing Bi-Cycle Consistency[C]//Advances in neural information processing systems. 2017: 465-476.
27. Kosaraju V, Sadeghian A, Martín-Martín R, et al. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks[J]. arXiv preprint arXiv:1907.03395, 2019.
28. Mangalam K, An Y, Girase H, et al. From goals, waypoints & paths to long term human trajectory forecasting[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 15233-15242.
29. Li X, Zheng W S, Wang X, et al. Multi-scale learning for low-resolution person re-identification[C]//Proceedings of the IEEE International Conference on Computer Vision. 2015: 3765-3773.
30. Huang Y, Zha Z J, Fu X, et al. Illumination-invariant person re-identification[C]//Proceedings of the 27th ACM International Conference on Multimedia. 2019: 365-373.
31. Huang H, Li D, Zhang Z, et al. Adversarially occluded samples for person re-identification[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 5098-5107.
32. Hou R, Ma B, Chang H, et al. Vrstc: Occlusion-free video person re-identification[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 7183-7192.
33. Lippi M, Bertini M, Frasconi P. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2013, 14(2): 871-882.
34. Yao H, Tang X, Wei H, et al. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction[C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 5668-5675.
35. Du B, Hu X, Sun L, et al. Traffic demand prediction based on dynamic transition convolutional neural network[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(2): 1237-1247.
36. Wang S, Cao J, Chen H, et al. SeqST-GAN: Seq2Seq generative adversarial nets for multi-step urban crowd flow prediction[J]. ACM Transactions on Spatial Algorithms and Systems (TSAS), 2020, 6(4): 1-24.
37. He S, Shin K G. Crowd-flow graph construction and identification with spatio-temporal signal feature fusion[C]//IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, 2019: 757-765.
38. Fang Z, Lo S M, Lu J A. On the relationship between crowd density and movement velocity[J]. Fire Safety Journal, 2003, 38(3): 271-283.
39. Niu Y, Shi W, Liu W, et al. Over-crowdedness Alert! Forecasting the Future Crowd Distribution[J]. arXiv preprint arXiv:2006.05127, 2020.

**Collaborative Video Analytics with Edge-Cloud Collaboration: Implementation and Evaluation**

*Xuezhi Wang and Guanyu Gao*

*Nanjing University of Science and Technology*

*120106222717@njust.edu.cn, gygao@njust.edu.cn*

**1. Introduction**

Video analytics is widely adopted in video surveillance, autonomous vehicle, self-service supermarket to analyze the captured video content to understand environments. The state-of-the-art video analytics algorithms are mostly implemented with Deep Neural Networks (DNNs), which consist of hundreds of layers and are compute intensive. The computational capacities of cameras and edge nodes are limited. Therefore, it may incur intolerable delays if all video analytics operations are performed locally on the cameras or the edge nodes. On the other hand, it will incur substantial bandwidth costs and delays if all video contents are transmitted to the cloud for video analytics. A practical approach to improve the performances of video analytics services is to leverage the collaboration of edge and cloud for video analytics.

Leveraging the edge-cloud collaboration for video analytics can combine the advantages of edge and cloud. The edge nodes are close to local cameras, and video contents can be transmitted to the edge with lower transmission delays and less bandwidth cost. On the other hand, the computing capacity of the edge node is limited, the video contents can only be processed with small DNN models on the edge node or be preprocessed on the edge node to reduce video sizes and then transmitted to the cloud for processing. The cloud can provision large DNN models which have high recognition accuracy. Therefore, the cloud can support video analytics tasks that require a large processing capacity and high accuracy.

Conducting video analytics with the edge-cloud collaboration is non-trivial. The video analytics pipelines involve various processing steps [1], such as frame-rate and resolution tuning for video preprocessing, model selection for video analytics, on-edge processing, or offloading. Each step significantly influences the service delays and cost. Therefore, the video analytics system must be carefully designed by considering the processing capacities of the edge node and the cloud, the bandwidth, and the volume of transmitted video data, and the delay requirement.

The previous works mainly focus on optimizing different components of video analytics pipelines to improve the performances. However, researchers and developers need to build each component of the video analytics pipelines from scratch to evaluate their methods. A flexible opensource framework is in need to facilitate the research and development in this filed. This paper presents an open source framework named SmartEye (https://github.com/MSNLAB/SmartEye) for real-time video analytics by leveraging the edge-cloud collaboration. The system consists of three layers. The on-edge processing layer is implemented on the edge node. It reads video frames from the networked cameras or video files, preprocesses video frames for offloading, or perform inferences on the edge. It also determines to adopt which server and DNN models for inference. The forwarding layer is implemented on a cloud frontend server. It accepts the offloading requests from the edge nodes and forwards the requests to the backend workers. The backend worker layer is implemented on the cloud backend servers. The workers receive the offloaded video analytics tasks from the forwarding server, perform inferences on GPU, and return the inference results.

Our system is modularized and flexible. The users can easily add new components or customize the existing modules. Our system contains the main components of video analytics pipelines, and the framework can relieve the burdens of the researchers from developing the system from scratch by building upon our provided modules and make the research and development in this field more convenient. The main features of our system are as follows.

**Edge-cloud collaborative or standalone processing.** The edge and the cloud can process inference requests collaboratively. The decision engine dynamically determines whether an inference is performed on the edge node or offloaded to the cloud. Our system also supports edge/cloud standalone processing. The edge node and the cloud can perform inferences without relying on each other, which can facilitate the video analytics under different application scenarios.

**Dynamic model selection and management.** The decision engine can dynamically select a model for inference based on the control policy, and dynamically load or unload modes for cache management. The model configuration file contains the offline profiling information of each model, e.g., accuracy, inference time on edge and cloud, which can facilitate decision-making.

**Easy configuration of system control knobs.** We provide the implementation of many commonly used video preprocessing methods, model selection policies, offloading policies, and request policies, and the users can easily configure the control knobs or implement their own control policies. We encapsulate the control policies into separate modules, and the users can quickly implement their control policies without changing the other parts of the codes.

The rest of this paper are as follows, Section II presents the related works, Section III introduces the system design and workflows for the video analytics system, Section IV presents the system implementation, Section V evaluates the performances in a real-world testbed, Section VI discusses some possible application scenarios, and Section VII concludes this paper.

**2. Related Work**

In this section, we introduce the related works for the video analytics with the edge-cloud collaboration.

With the wide deployment of networked cameras, video analytics has been one of the killer applications for edge computing [2]. Many previous works have studied how to optimize video analytics pipelines, and the existing works focus on designing different mechanisms for video analytics to improve the performances. For instances, Awstream [3] adopts frame-rate and resolution tuning, and Reducto [4] sends only video frames with new objects to save bandwidth cost. VideoStorm [5] and Chameleon [6] adopt video frame sampling and resolution downsizing to save the compute cost of inferences on cheap DNN models. VideoEdge [7] designs an optimization framework to manage the tradeoff between accuracy and resource consumption for video analytics. CEVAS [8] adopts a serverless design for workload partition between edge and cloud for video analytics. Runespoor [9] proposed an edge-cloud video analytics system for managing the tail accuracy for emerging robotics applications. Jain et al. [10] studied the optimization of video analytics system for large scale camera deployments. Different from the previous works, we design a general framework for the video analytics pipelines, which can incorporate different video preprocessing methods, offloading methods, and model selection methods. One can easily implement a customized video analytics system and evaluate the performances with our provided modules.

**3. System Design**

In this section, we first present the system architecture of SmartEye and then introduce the workflows for video analytics.

**3.1 Architecture**

We illustrate the system architecture of SmartEye in Fig. 1. The system mainly consists of three components: the edge node, the forwarding server, and the inference server.

图示

描述已自动生成

Fig. 1. The system architecture of SmartEye. SmartEye consists of three components, namely, the edge node, the forwarding server, and the inference server.

**Edge Node.** Theedge node can conduct video processing on the edge or offload the video frames to the cloud for processing. The main functionalities of the edge node include video frame reading, video preprocessing, DNN model selection, local inference, offloading, and decision-making. The edge node reads video frames from the camera or video files and preprocesses the video frames. After video preprocessing, the inference for a video frame can be performed on the edge node or offloaded to the cloud for execution. The edge node makes the video preprocessing, offloading, and model selection decisions based on the specified control policies.

**Forwarding Server.** The forwarding server serves as the cloud’s gateway to respond to the edge node’s offloading requests. When offloading an inference task, the edge node submits an HTTP request to the forwarding server by attaching the video frame. The forwarding server dispatches the inference requests to the backend servers based on the forwarding policy. The forwarding server also monitors the resource utilization, workload, and loaded models of each backend server and uses the backend servers’ status information to make dispatch decisions.

**Inference Server.** The inference servers are provisioned in the cloud to conduct video analytics inferences. Each inference server loads several DNN models for video analytics. The inference servers receive the offloaded tasks from the forwarding server and make inferences with the specified models.

**3.2 Workflow**

The system has many control knobs, which control the workflows of the video analytics pipelines. The main workflows are as follows. The edge node reads video frames from the camera, and the interval between reading two frames from the camera or the video file is configurable. The decision engine of the edge node makes decisions on how to preprocess a video frame, and whether the inference should be offloaded to the cloud or performed on the edge node, and which DNN models should be selected for video analytics. These control policies are coupled with each other because the preprocessing influences the data size, which further influences the transmission delays and inference time. The forwarding server can make the forwarding decision based on its forwarding policy or based on the edge node’s specification. The inference server performs inferences with the specified DNN model of the edge node.

**4. System Implementation**

In this section, we present the implementation of SmartEye. We illustrate the core modules and the relationships among different modules in Fig. 2.

图示

描述已自动生成

Fig. 2. The implementation of the core modules in SmartEye. The system is divided into three layers, namely, the on-edge processing layer, the forwarding layer, and the backend inference layer.

**4.1 On-Edge Processing Layer**

**Video Reader.** The Video Reader module encapsulates the interfaces for reading video frames, and we provide three types of interfaces. The edge node can read video frames from the networked camera using Real-Time Streaming Protocol (RTSP) and read video frames from the physically connected camera as a local device. We also provide the interface to read frames from video files. The users can select one interface based on the video sources.

**Decision Engine.** The Decision Engine module contains the policies for video frame preprocessing, model selection, local inference, and offloading. These control actions couple with each other, and the control policy can make joint decisions. The users can also implement their policies in this module. One can specify a policy for video frame preprocessing, local inference/offloading, model selection by setting the value of item control policy in the edge-setting section of config.ini. For instance, we implement a threshold-based control policy. Under this policy, if the local pending task number is less than a threshold (i.e, 2), the inference for the next video frame will be conducted on edge with the fastest model without resolution downsizing. Otherwise, the pre-processor downsizes the resolution of the next video frame to 240P. The edge node will offload the task to the cloud for inference with the most accurate model.

**Pre-Processor.** The Pre-Processor module encapsulates the functions for processing a video frame or a video file. The functions include resolution downsizing, quality tuning, frame-rate down sampling, encoding video frames into a video file. The Pre-Processor processes video frames under the control of Decision Engine.

**Inference Worker.** The Inference Worker is a thread that performs inferences on the edge node. It fetches the preprocessed video frames from the task queue and performs the inference for the video frames using the model selected by the Decision Engine. It will be blocked by the task queue when no task is available.

**Offloading Worker.** The Offloading Worker is implemented with a thread pool. An offloaded inference will be assigned to a thread in the thread pool. The thread will submit the offloading task to the forwarding server using the HTTP Post method. Whenever a new offloading task comes, it will be processed by an idle thread in the thread pool. The offloading tasks can be submitted to the cloud in parallelism to reduce queuing delays. However, the number of threads in the thread pool is limited, an offloading request will be blocked if there is no idle thread in the thread pool.

**Model Manager.** The Model Manager module provides the APIs to manage the DNN models for inference. Both the edge node and the backend workers use this module for DNN model management. The functionalities include loading a set of models into memory, performing inference with a specified model, load/unload a model. The module also provides the interfaces to obtain a model’s profile via reading the configuration file, for example, obtaining the precision of a model, average inference delays for a model on the edge node or on the backend server.

**Offloading.** The Offloading module provides the interfaces to send an offloading task to the forwarding server using the HTTP Post method. The video frame sent to the forwarding server with POST is stored in the request body of the HTTP request. The selected model for inference is specified in the request.

**System Information.** The System Information module collects the edge node’s resource utilization and the performances of inference and offloading, which includes CPU and memory usage, bandwidth, and processing delays. The Decision Engine can utilize the collected information to make control decisions.

**4.2 Forwarding Layer**

**Forwarding Server.** The Forwarding Server is an HTTP-based web server implemented with Flask. It receives the offloading requests from the edge node and forwards the inference requests and the video frames to the backend inference servers to obtain inference results. The communications between the forwarding server and the backend inference servers are implemented with Remote Procedure Call (RPC).

**Server Monitor.** The Server Monitor module is implemented as a background process of the Flask server. It quires the resource utilization (e.g., CPU usage, memory usage) of each inference server under a fixed time interval. The collected information is utilized for server health monitoring and making dispatch decisions.

**Dispatch Policy.** The Dispatch Policy module provides the policies of how to dispatch the offloaded tasks among the backend inference servers. The forwarding server makes the dispatching decision based on the dispatch policy and the status of the inference servers. We have implemented some dispatch policies, including the shortest waiting queue, random dispatching. The users can also implement their customized dispatch policies.

**4.3 Backend Layer**

**Inference Server.** The inference servers are the physical servers or the virtual machines in the cloud.The Inference Server is responsible for processing the inference requests offloaded from the edge nodes. The inference servers make inferences for video frames with the specified DNN models and return the inference results to the forwarding server.

**Model Manager.** The Model Manager module for Inference Sever is same with the edge node. However, it returns the performances on GPU when querying the performance of a model. Meanwhile, the inference servers and the edge nodes can load different models due to their different memory sizes and processing speeds. The Model Manage module provides different interfaces for model management in the edge or cloud.

**5. Performance Evaluation**

In this section, we present the experiment settings of our real-world testbed and evaluate the performances of our system under different settings.

**Testbed.** We implement a real-world testbed to evaluate the performances of our system. The edge node is an NVIDIA Jetson TX2, with an NVIDIA Pascal GPU, 8GB memory, and an ARMv8 CPU. We use two servers as the cloud backend inference servers. Each server has 128GB memory, a GeForce RTX 2080Ti GPU, and an Intel Xeon Gold 6242R CPU. One server hosts the forwarding server. The edge node is in Jiangyin, China, and the cloud servers are in Nanjing, China. We choose object detection for video analytics and use the pre-trained models from PyTorch Hub. The test video is a traffic monitoring video.

图表, 条形图, 直方图

描述已自动生成

Fig. 3. The average delays for inference on edge and offloading to cloud under different video resolutions.

**Performance influencing factors for inference on edge and cloud.** We illustrate the average processing delay using different DNN models and video resolutions for on-edge inference and offloading to the cloud for inference in Fig. [3.](#_bookmark7) The on-edge inference is very sensitive to the models’ complexities and less sensitive to resolution. The average local processing delay of RetinaNet is about seven times longer than Faster RCNN. However, the processing delay for offloading to the cloud is mainly occupied by transmission time. If the video frames are downsized (e.g., 240P, 360P), the average processing delays for offloading can be significantly reduced, even smaller than on-edge inference. However, downsizing resolution comes with the cost of lower accuracy. As with several previous works [[1],](#_bookmark17) [[4],](#_bookmark20) we adopt the recognition results from a model on the original resolution (1080P) as the ground truth, and evaluate the accuracy of the model on the downsized video frames by comparing with the results from the original resolution. As illustrated in Table [I,](#_bookmark14) the COCO mAP of Faster RCNN drops to 0.752, 0.807, 0.854, 0.900, and RetinaNet drops to 0.689, 0.766, 0.813, 0.915 respectively, when the resolution is downsized to 240P, 360P, 480P, and 720P. The results verify that the resolution has an influence on the recognition accuracy of a DNN model. Therefore, one needs to consider the tradeoff between the recognition accuracy and the transmission delay to the cloud when designing the offloading policy to the cloud.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 240P | 360P | 480P | 720P |
| Faster RCNN | 0.752 | 0.807 | 0.854 | 0.900 |
| RetinaNet | 0.689 | 0.766 | 0.813 | 0.915 |

Table I: The precision under different resolutions by comparing 1080P as the ground truth.

**Performance advantages of edge-cloud collaboration for video analytics.** We set the frame reading interval as different values and evaluate the average processing delays for 200 video frames. Smaller intervals lead to heavier workloads. We adopt the threshold offload policy introduced in the use case to verify the performance advantage of edge-cloud collaboration. We compared the performances with purely on-edge inference and purely offloading to the cloud with pre-downsizing to 240P on edge. The results are illustrated in Fig. [4.](#_bookmark13) More inferences will be offloaded to the cloud when the loads are heavier, and the edge-cloud threshold offloading can maintain lower processing delays under heavy workloads. This verifies that the video analytics performances can be improved under the edge-cloud paradigm with properly designed offloading policy and video pre-processing policy.

图表

描述已自动生成

Fig. 4. Performances of pure on-edge, pure offloading, and edge-cloud threshold-based offloading policies under different loads.

**6. Application**

In this section, we discuss how to deploy SmartEye in some applications which rely on video analytics.

**Video surveillance:** Cameras are deployed in every corner of the cities for video surveillance. These cameras generate a huge volume of video content for video analytics. The edge node can read video frames from the networked cameras and perform some video preprocessing tasks. The captured video content can also be analyzed on the edge with the light-weight models, and the compute-intensive tasks can be performed on the cloud. Transmitting the original video content or the pre-processed video frames to the cloud incur substantial bandwidth cost, therefore, one main research focus in this application scenario is to partition the video analytics tasks between the edge node and the cloud. Because the computational capacity of the edge node is limited, the allocated tasks to the edge node cannot be too large to avoid intolerable delays. On the other hand, the offloaded data to the cloud should not exceed the bandwidth capacity.

**Autonomous driving:** The autonomous vehicles leverage cameras to capture the environment data to make control decisions for self-driving. The captured video content by the cameras must be analyzed in real-time to make control decisions. In this application scenario, the edge nodes can be deployed at the base station, and the autonomous vehicles can connect with the edge node through mobile network. The pre-processed video frames will be offloaded to the edge node. On the other hand, the vehicles will continuously observe new scenes and objects. The edge nodes can collect the video data from different vehicles for collaborative learning to improve the prediction accuracy of the video analytics. The continuous training of the deep learning model consumes substantial computational resources, therefore, the training tasks can be offloaded to the cloud, and the models on the edge nodes can be updated with the trained models in the cloud.

**7. Conclusion**

This paper presented the implementation of an open source framework named SmartEye for real-time video analytics with the edge-cloud collaboration. The system is well implemented and easy to customize, and it can free users from the tedious works of building pipelines from scratch. We conducted some quantitative evaluations to verify the performances. We will continue to maintain SmartEye as an open source project and provide more implementations of the video processing algorithms and control algorithms. In the future, we will continue our research from the following directions. First, we will study the problem of dynamic model provisioning in the edge and the cloud to meet the performance constraints. Second, we will study the continual learning problem for updating the video analytics models using the newly collected data from the cameras to improve recognition accuracy. Finally, we will consider the collaborative learning problem for transfer knowledge among the edge nodes to improve the recognition accuracy by knowledge sharing.

**References**

1. Z. Xiao, Z. Xia, H. Zheng, B. Y. Zhao, and J. Jiang, “Towards performance clarity of edge video analytics,” arXiv preprint arXiv:2105.08694, 2021.
2. G. Ananthanarayanan, P. Bahl, P. Bod´ık, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, “Real-time video analytics: The killer app for edge computing,” computer, vol. 50, no. 10, pp. 58–67, 2017.
3. B. Zhang, X. Jin, S. Ratnasamy, J. Wawrzynek, and E. A. Lee, “Awstream: Adaptive wide-area streaming analytics,” in Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, 2018, pp. 236–252.
4. Y. Li, A. Padmanabhan, P. Zhao, Y. Wang, G. H. Xu, and R. Netravali, “Reducto: On-camera filtering for resource-efficient real-time video analytics,” in Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication, 2020, pp. 359–376.
5. H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, and M. J. Freedman, “Live video analytics at scale with approximation and delay- tolerance,” in 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17), 2017, pp. 377–392.
6. J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica, “Chameleon: scalable adaptation of video analytics,” in Proceedings of SIGCOMM 2018, 2018, pp. 253–266.
7. C.-C. Hung, G. Ananthanarayanan, P. Bodik, L. Golubchik, M. Yu, P. Bahl, and M. Philipose, “Videoedge: Processing camera streams using hierarchical clusters,” in 2018 IEEE/ACM Symposium on Edge Computing (SEC). IEEE, 2018, pp. 115–131.
8. M. Zhang, F. Wang, Y. Zhu, J. Liu, and Z. Wang, “Towards cloud-edge collaborative online video analytics with fine-grained serverless pipelines,” in
9. Proceedings of the 12th ACM Multimedia Systems Conference, 2021, pp. 80–93.
10. Y. Wang, W. Wang, D. Liu, X. Jin, J. Jiang, and K. Chen, “Enabling edge-cloud video analytics for robotics applications,” in Proceedings of the IEEE International Conference on Computer Communications, Virtual Conference, 2021, pp. 10–13.
11. S. Jain, G. Ananthanarayanan, J. Jiang, Y. Shu, and J. Gonzalez, “Scaling video analytics systems to large camera deployments,” in Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications, 2019, pp. 9–14.

**Discrete-Time Optimal Control based Tracking Differentiator Algorithms**

*Hehong Zhang1, Yunde Xie2, Qiumeng Qian3,4\**

*1College of Computer and Data Science, Fuzhou University, Fuzhou, China*

*2Beijing Rail Transit Technology Equipment Group Co., Ltd, Beijing, China*

*3School of Automation Science and Electrical Engineering, Beihang University, China*

*4Sichuan Gas Turbine Establishment, Aero Engine Corporation of China , Mianyang, China*

[*qqm7389190@126.com*](mailto:qqm7389190@126.com)

1. **Introduction**

The real-time filtering and differentiation acquisition of a given signal is a common and important problem in control engineering practice [1][2]. For example, in the altitude ground test facility, acquiring real-time and accurate pressure signal and its corresponding velocity is critically important for designing a feedback controller to provide total pressure conditions as that occur during flight operation for the tested aero-engines at chamber [3]. For magnetic levitation systems, knowing effective gap (between the electromagnet and the objective) and its corresponding velocity signals is vital to construct the suspension feedback controller to guarantee the vehicle to stably suspend on the guideway [4]. The same happens in other control and signal processing related cases.

For a given signal with the explicit mathematical expression, like basic functions, we can calculate its differentiation outputs by performing the derivative operation. For most cases in real-life engineering, however, a signal sequence that is usually collected from a sensor has no definite mathematical expression, and the difference method is usually applied to determine its differentiation outputs. Inevitable noises existing in signals from sensors make the derivative outputs inaccurate. The state observer or disturbance estimator is utilized to estimate the differential signals when the dynamic model of a system is known. However, the system model is not always accessible. The design of a differentiator then becomes necessary in some cases and has attracted much attention in recent years. For example, a finite-time-convergent differentiator, a high-gain observer [5], a linear time derivative tracker [6], a super-twisting second-order sliding mode algorithm [7], robust exact differentiation [8], a finite time convergent differentiator [9], and so on.

Initially proposed by Han, a noise-tolerant time optimal control (TOC)-based tracking differentiator (TD) allows one to avoid a setpoint jump in the emerging active disturbance rejection controller [10]. The advantage of this TD is that it sets a weak condition on the stability of the systems to be constructed for TD and requires a weak condition on the input. In addition, it also has the advantage of maintaining a greater level of smoothness compared to the chattering problem encountered by sliding mode-based differentiators. Among extensive studies on differentiators, to the best of our knowledge, the DTOC-TD is one of most commonly-used algorithms in real-life engineering. This paper focuses on the summary of the DTOC-TD algorithms that we have proposed.

1. **Problem Formulation**

Based on a given known second-order system, one can construct a corresponding differentiator by designing the control algorithm for that system to acquire filtering signals and the corresponding differential signals simultaneously (see Fig. 1). As shown in Fig. 1, a differentiator system only needs measurable and bounded signals collected from sensors as its inputs.

图示

描述已自动生成

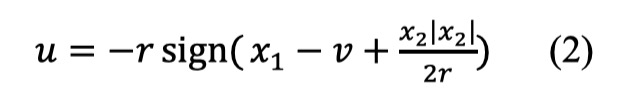
Fig. 1 The schematic diagram of a differentiator system

The following presents the outline for the construction of the TD. The double-integral system is defined as

文本

描述已自动生成

where 𝑟 is a constant. The resulting feedback control law that drives the states from any initial point to the origin in the shortest time is



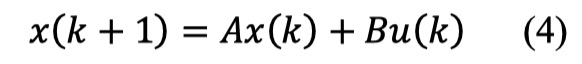
where is the desired value for . The switching curve function is =. Using this principle, we can obtain the desired trajectory and its derivative by solving the following differential equations:

文本

描述已自动生成

where is the desired trajectory and is its derivative. With the developments in computer control technology, most control algorithms are now implemented in the discrete time domain.

Thus, we consider the following discrete-time double-integral system



where |𝑢| ≤ 𝑟 gives the limitation of system acceleration, , . In practice, direct digitization of a continuous TOC solution of (1) is problematic because of the high-frequency chattering of the control signals. The discrete time optimal control (DTOC) law (Fhan) of the TD is determined by comparing the position of the initial state with the isochronic region (IR) obtained through nonlinear boundary transformation [11]. This makes the structure of a TD to be complex with nonlinear calculations, including square root calculations. The objective here is to derive a general TOC law directly in discrete time domain. The problem is defined as follows.

**DTOC Law:** Given the system (4) and its initial state 𝑋(0), we can determine the control signal sequence, 𝑢(0), 𝑢(1) . . . , 𝑢(𝑘), such that the state 𝑋(𝑘) is driven back to the origin in a finite and minimum number of steps, subject to the constraint of | 𝑢(𝑘)|. That is, finding , | 𝑢(𝑘)|≤ 𝑟, such that =min{𝑘|𝑥(𝑘 + 1) = 0}. For a discrete-time system, the state is measured only at the sampling instant, 𝑡 = 𝑘ℎ, where ℎ is the sampling period. If we treat the measurement, 𝑥(𝑘ℎ), as though it were an initial condition, 𝑋(0), then all we need to find is 𝑢(0), as defined by DTOC Law at each sampling instant. This is repeated until the state reaches the origin.

1. **A General Tracking Differentiator**

The proposed general DTOC law is based on IR, which is referred to as the 𝐺(𝑘) approach. 𝐺(𝑘) denotes the set of states that, for any initial state inside a specific IR, at least one admissible control sequence exists, i.e., 𝑢(0), 𝑢(1) . . . , 𝑢(𝑘), that makes the solution of (4) satisfy 𝑋(𝑘 + 1) = 0. Note that the IR grows in volume as 𝑘 increases, i.e., 𝐺(𝑘 − 1) ⊂ 𝐺(𝑘). The basic idea in deriving the DTOC law is to identify a control signal sequence for any 𝑋(0) ⊂𝐺(𝑘) and 𝑋(0) ∉ 𝐺(𝑘 − 1), such that the next state 𝑋(1), calculated from the discrete-time double-integral system, satisfies 𝑋(1) ⊂ 𝐺(𝑘 − 1). This process is divided into two tasks:

1. Determine the boundary curves of IR by connecting the points of 𝐺(𝑘), as well as the control characteristic curve ( ), from which the state can be driven back to the origin in finite steps;
2. For any given initial condition, find the corresponding control signal sequence as a function of

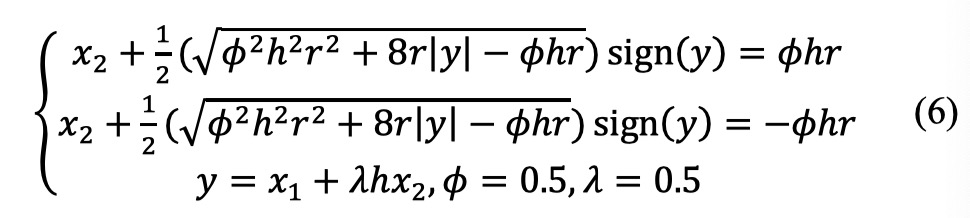
𝑋(0).

According to the above steps, we have obtained the mathematical derivation of a general closed- form DTOC law (DTOC Law) as a function of , and ℎ, denoted as . The traditional DTOC law (Fhan) of the TD is determined by comparing the position of the initial state with the IR through non-linear boundary transformation functions. In contrast, the proposed general DTOC law (Fhh) is created by the boundary curves, a control characteristic curve, and three corresponding boundary characteristic points (, , and ). This produces a one-to-one correspondence between boundary transformation functions and boundary characteristic points. Therefore, obtaining the general form of DTOC laws can be accomplished flexibly by modifying characteristic points. For the proposed Fhh law, the boundary characteristic points are

文本, 信件

描述已自动生成

The boundary functions of that law are presented as follows:



Based on the control law Fhh, we constructed the following TD:

文本, 信件

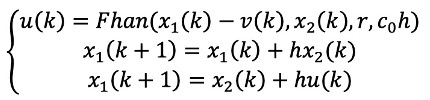
描述已自动生成

where 𝑟 is the quickness factor, is the filtering factor, ℎ is the sampling period, and 𝑣 is the given signal.

1. **Performance Analysis**

We ran numerical simulations to compare the performance of the proposed differentiator with that of the existing ones for signal-tracking filtering and differentiation acquisition.

DI. Tracking differentiator based on Fhan [10].



DII. Tracking differentiator based on Ftd [12].

文本, 信件

描述已自动生成

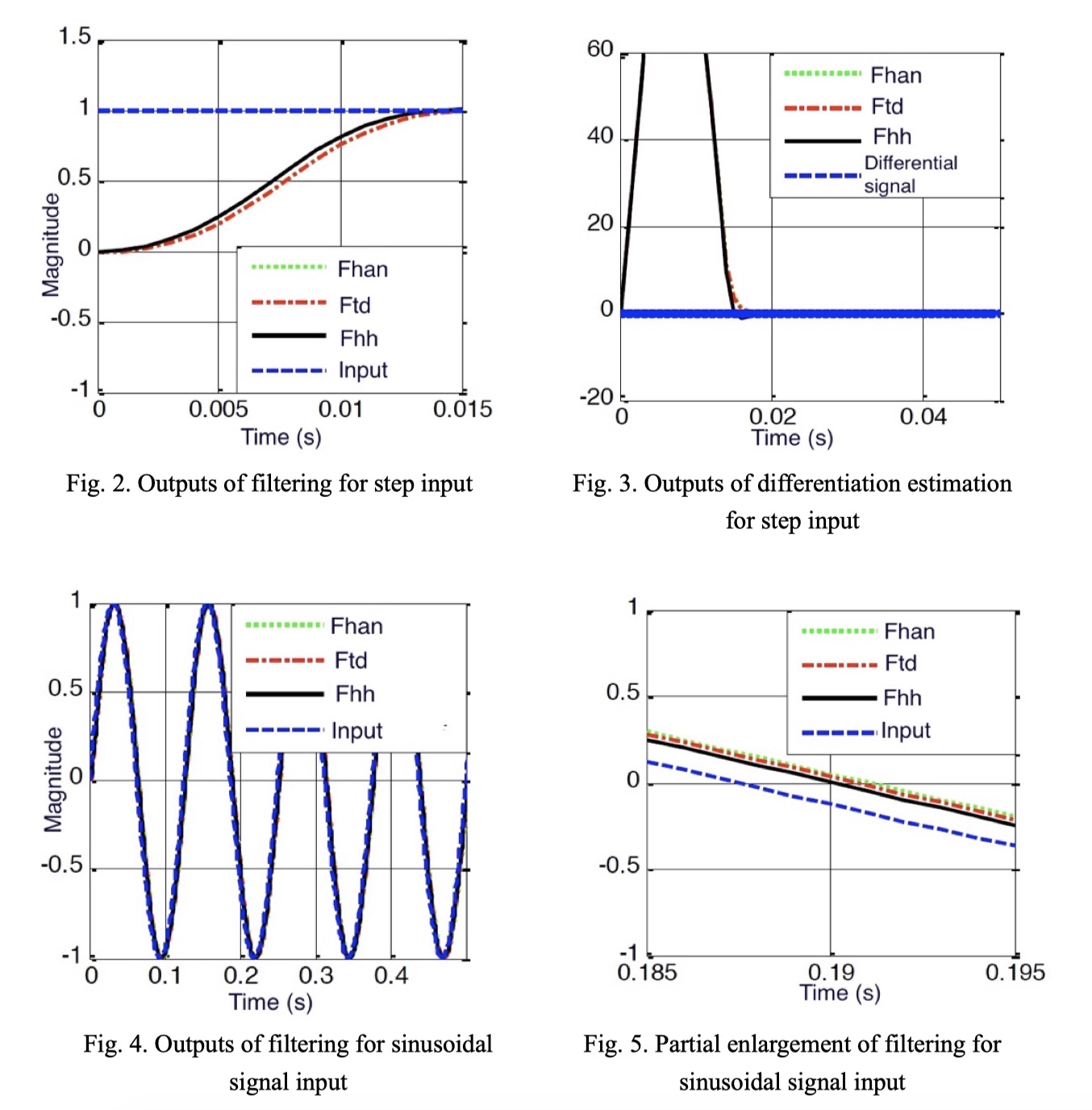
DIII. Tracking differentiator based on Fhh.

文本, 信件

描述已自动生成

For these simulations, we selected the step function and sinusoidal signal as our input signal sequences, setting the same initial value ((0) = 0, (0) = 0) for all of the simulations. For parameter setting, with the trial and error method, we first set the parameters for the traditional TD to get the best possible performance. Then we take on the same parameters for the proposed differentiator, that is, sampling period, ℎ=0.005𝑠; quickness factor, 𝑟 =500; and filtering factor =5. We plotted the results for comparisons among them in signal-tracking filtering and differentiation acquisition.

Figs. 2, 4 and 5 revealed that all the three different TDs could quickly track the input signals without overshooting and chattering. Our proposed Fhh-based TD proved to be the most rapid one in signal-tracking. As shown in Figs. 3, 6 and 7, although the Fhan-based TD was, to some extent, capable of producing good differential signals, some intermittent jumps occurred in differentiation acquisition. In contrast, the proposed Fhh-based TD avoided such jumps and obtained the highest precision of differential signals within the shortest time.



图表, 折线图

描述已自动生成

1. **Conclusion and Future Work**

In this paper, we investigated the time optimal control-based tracking differentiator algorithms. Time optimal control algorithms with boundary level have advantages at the aspects of chattering alleviation and quickness, which offer a great potential to construct the corresponding differentiators. In particular, we studied a generalized and simple discrete time optimal control algorithm that can be used to construct the differentiators. Further, we can obtain the general form of DTOC-TD algorithms by flexibly modifying characteristic points. Numerical simulation results indicated that, when compared with two existing differentiators, the proposed TD achieves better performance and higher precision in signal-tracking filtering and differentiation acquisition. Other future investigations on TD will include an analysis of the accuracy of tracking and differentiation for the proposed TDs, and the selection of the optimal TD’s parameters using a heuristic algorithm.

**Reference**

1. Y. Li, K. H. Ang, and G. C. Y. Chong, “Patents, software, and hardware for PID control: An overview and analysis of the current art,” IEEE Control Syst., vol. 26, no. 1, pp. 42–54, Feb. 2006.
2. K. H. Ang, G. Chong, and Y. Li, “PID control system analysis, design, and technology,” IEEE Trans. Control Syst. Technol., vol. 13, no. 4, pp. 559–576, Jul. 2005.
3. P. F. Ashwood: An altitude test facility for large turbofan engines. J Aircr., vol. 10, no. 8, pp. 468-474, 1973.
4. H. Zhang, Y. Xie, G. Xiao, and C. Zhai, and Z. Long, “A simple discrete-time tracking differentiator and its application to speed and position detection system for a maglev train,” IEEE Trans. Control Syst Technol., vol. 27, no. 4, pp. 1728–1734, 2018.
5. X. Wang, Z. Chen, and G. Yang, “Finite-time-convergent differentiator based on singular perturbation technique,” IEEE Trans. Autom. Control, vol. 52, no. 9, pp. 1731–1737, Sep. 2007.
6. A. M. Dabroom and H. K. Khalil, “Discrete-time implementation of high-gain observers for numerical differentiation,” Int. J. Control, vol. 72, no. 17, pp. 1523–1537, 1999.
7. S. Ibrir, “Linear time-derivative trackers,” Automatica, vol. 40, no. 3, pp. 397–405, 2004.
8. J. Davila, L. M. Fridman, and A. Levant, “Second-order sliding-mode observer for mechanical systems,” IEEE Trans. Autom. Control, vol. 50, no. 11, pp. 1785–1789, Nov. 2005.
9. A. Levant, “Robust exact differentiation via sliding mode technique,” Automatica, vol. 34, no. 3, pp. 379–384, Mar. 1998.
10. J. Han, “From PID to active disturbance rejection control,” IEEE Trans. Ind. Electron., vol. 56, no. 3, pp. 900–906, Mar. 2009.
11. Z. Gao, “On discrete time optimal control: A closed-form solution,” in Proc. Amer. Control Conf., vol. 1, pp. 52–58, 2004.
12. H. Zhang, Y. Xie, G. Xiao, and C. Zhai, “Closed-form solution of discrete-time optimal control and its convergence,” IET Control Theory Appl., vol. 12, no. 3, pp. 413–418, 2017.

穿着蓝色衣服的男孩

描述已自动生成**Hehong Zhang** received his Ph.D. degree in smart control and data analysis from Nanyang Technological University, Singapore, in 2020. He is now an Associate Professor in College of Computer and Data Science, Fuzhou University, Fuzhou, China. His research interests include differentiator theory, state estimation, signal processing, theory and application of optimal control and nonlinear control theory in the altitude ground test facility and maglev train system.

穿着西装的男人在微笑

描述已自动生成

**Yunde Xie** received his Ph. D. degree in control science and control engineering from National University of Defense Technology in 1998. He was an Associate Professor in National University of Defense Technology, China, in 2004. He is now a senior engineer at Beijing Maglev Transportation Development Co.,Ltd. His research interests include nonlinear control theory and application and maglev control techniques.

穿着西装笔挺的男子

描述已自动生成**Qiumeng Qian** received the M.S. degrees in Pattern Recognition and Intelligent System from Nanjing University of Aeronautics and Astronautics, in 2013. He is currently working as a Senior engineer in AECC Sichuan Gas Turbine Establishment, China. His research interests include differentiator theory, industrial control, control and simulation of aero-engine altitude test.

**MMTC OFFICERS (Term 2020 — 2022)**

**CHAIR STEERING COMMITTEE CHAIR**

**Jun Wu**  **Joel J. P. C. Rodrigues**

Fudan University Federal University of Piauí (UFPI)

China Brazil

**VICE CHAIRS**

**Shaoen Wu** (**North America)** **Liang Zhou** (Asia)

Illinois State University Nanjing University of Post and Telecommunications

USA China

**Abderrahim Benslimane** (Europe) **Qing Yang (Letters & Member Communications)**

University of Avignon University of North Texas

France USA

**SECRETARY STANDARDS LIAISON**

**Han Hu Weiyi Zhang**

Beijing Institute of Technology AT&T Research

China USA

**MMTC Communication-Frontier BOARD MEMBERS (Term 2016—2018)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Danda Rawat** | Director | Howard University | USA |
| **Sudip Misra** | Co-Director | IIT Kharagpur | India |
| **Guanyu Gao** | Co-Director | Nanjing University of Science and Technology | China |
| **Rui Wang** | Co-Director | Tongji University | China |
| **Lei Chen** | Editor | Georgia Southern University | USA |
| **Tasos Dagiuklas** | Editor | London South Bank University | UK |
| **ShuaiShuai Guo** | Editor | King Abdullah University of Science and Technology | Saudi Arabia |
| **Kejie Lu** | Editor | University of Puerto Rico at Mayagüez | Puerto Rico |
| **Nathalie Mitton** | Editor | Inria Lille-Nord Europe | France |
| **Zheng Chang** | Editor | University of Jyväskylä | Finland |
| **Dapeng Wu** | Editor | Chongqing University of Posts & Telecommunications | China |
| **Luca Foschini** | Editor | University of Bologna | Italy |
| **Mohamed Faten Zhani** | Editor | University of Quebec | Canada |
| **Armir Bujari** | Editor | University of Padua | Italy |
| **Kuan Zhang** | Editor | University of Nebraska-Lincoln | USA |
| **Bin Tan** | Editor | Jinggangshan University | China |