EAI/Springer Innovations in Communication and Computing

Baoliu Ye
Weihua Zhuang
Song Guo *Editors*

2nd International Conference on 5G for Ubiquitous Connectivity

5GU 2018

EAI
RESEARCH MEETS INNOVATION

Springer

EAI/Springer Innovations in Communication and Computing

Series editor
Imrich Chlamtac, European Alliance for Innovation, Gent, Belgium

Editor's Note

The impact of information technologies is creating a new world yet not fully understood. The extent and speed of economic, life style and social changes already perceived in everyday life is hard to estimate without understanding the technological driving forces behind it. This series presents contributed volumes featuring the latest research and development in the various information engineering technologies that play a key role in this process.

The range of topics, focusing primarily on communications and computing engineering include, but are not limited to, wireless networks; mobile communication; design and learning; gaming; interaction; e-health and pervasive healthcare; energy management; smart grids; internet of things; cognitive radio networks; computation; cloud computing; ubiquitous connectivity, and in mode general smart living, smart cities, Internet of Things and more. The series publishes a combination of expanded papers selected from hosted and sponsored European Alliance for Innovation (EAI) conferences that present cutting edge, global research as well as provide new perspectives on traditional related engineering fields. This content, complemented with open calls for contribution of book titles and individual chapters, together maintain Springer's and EAI's high standards of academic excellence. The audience for the books consists of researchers, industry professionals, advanced level students as well as practitioners in related fields of activity include information and communication specialists, security experts, economists, urban planners, doctors, and in general representatives in all those walks of life affected ad contributing to the information revolution.

About EAI

EAI is a grassroots member organization initiated through cooperation between businesses, public, private and government organizations to address the global challenges of Europe's future competitiveness and link the European Research community with its counterparts around the globe. EAI reaches out to hundreds of thousands of individual subscribers on all continents and collaborates with an institutional member base including Fortune 500 companies, government organizations, and educational institutions, provide a free research and innovation platform.

Through its open free membership model EAI promotes a new research and innovation culture based on collaboration, connectivity and recognition of excellence by community.

More information about this series at http://www.springer.com/series/15427

Baoliu Ye • Weihua Zhuang • Song Guo
Editors

2nd International Conference on 5G for Ubiquitous Connectivity

5GU 2018

Springer

EAI
RESEARCH MEETS INNOVATION

Editors

Baoliu Ye
National Key Laboratory for Novel
Software Technology
Nanjing University
Nanjing, China

Weihua Zhuang
Department of Electrical and Computer
Engineering
University of Waterloo
Waterloo, ON, Canada

Song Guo
Department of Computing
The University of Polytechnic University
Kowloon
Hong Kong, Kowloon, Hong Kong

ISSN 2522-8595 ISSN 2522-8609 (electronic)
EAI/Springer Innovations in Communication and Computing
ISBN 978-3-030-22315-1 ISBN 978-3-030-22316-8 (eBook)
https://doi.org/10.1007/978-3-030-22316-8

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

We are delighted to introduce the proceedings of the 2nd International Conference on 5G for Ubiquitous Connectivity (5GU 2018). The aim of this conference is to bring together researchers and developers as well as regulators and policy makers to present their latest views on 5G: New networking, new wireless communications, resource control and management, future access techniques, new emerging applications, and of course, latest findings in key research activities on 5G.

The technical program of 5GU 2018 consisted of 15 full papers at the main conference tracks. The conference tracks were Track 1—New networking for 5G and beyond; Track 2—New wireless communications for 5G; Track 3—Resource control and management for 5G; Track 4—Future access techniques, and Track 5—New emerging applications. Aside from the high-quality technical paper presentations, the technical program also featured two keynote speeches. The two keynote speeches were by Dr. Ing. Thorsten Herfet from Saarland Informatics Campus, Germany, and Dr. Shi Jin from Southeast University, China.

Coordination with the steering chair, Prof. Imrich Chlamtac, was essential for the success of the conference. We sincerely appreciate the contribution of two general chairs, Prof. Baoliu Ye and Prof. Weihua Zhuang. It was also a great pleasure to work with such an excellent organizing committee team for their hard work in organizing and supporting the conference. In particular, the Technical Program Committee, led by our TPC Chair, Prof. Song Guo, who have completed the peer review process of technical papers and made a high-quality technical program. We are also grateful to all the authors who submitted their papers to the 5GU conference.

We strongly believe that 5GU 2018 conference provides a good forum for all researcher, developers, and practitioners to discuss all science and technology aspects that are relevant to 5G. We also expect that the future 5GU conference will be as successful and stimulating as indicated by the contributions presented in this volume.

Nanjing, China — Baoliu Ye
Waterloo, ON, Canada — Weihua Zhuang
Hong Kong, Kowloon, Hong Kong — Song Guo
Aizuwakamatsu, Japan — Peng Li

Conference Organization

Steering Committee	
Imrich Chlamtac	University of Trento, Italy
Organizing Committee	
General Chairs	
Baoliu Ye	Nanjing University, China
Weihua Zhuang	University of Waterloo, Canada
TPC Chair	
Song Guo	Hong Kong Polytechnic University
Local Chair	
Xin Wang	Hohai University, China
Workshops Chair	
Hongzi Zhu	Shanghai Jiaotong University, China
Publicity & Social Media Chair	
Guoping Tan	Hohai University, China
Publications Chair	
Peng Li	The University of Aizu, Japan
Web Chair	
Xujie Li	Hohai University, China
Conference Manager	
Kristina Lappyova	EAI
Technical Program Committee	
Shravan Garlapati	Virginia Tech University
Xiaojun Hei	Huazhong University of Science and Technology, China
Peng Liu	Hangzhou Dianzi University, China
Shengli Pan	China University of Geosciences (Wuhan), China
Tian Wang	Huaqiao University, China
Xiaoyan Wang	Ibaraki University, Japan
Xiaobo Zhou	Tianjin University, China
Shigeng Zhang	Central South University, China

Contents

Collaborative Inference for Mobile Deep Learning Applications

Qinglin Yang, Xiaofei Luo, Peng Li, and Toshiaki Miyazaki

1 Introduction

Algorithmic breakthroughs of deep learning in the past decades has attracted wide interest of developing artificial intelligence (AI) empowered mobile applications, such as Tencent QQ, Google Map, Apple Health, and Avast Mobile Security, etc., to conduct language translation, object recognition, health monitoring, and malware detection. The intelligent services provided by these mobile applications generally enable people to enjoy a more convenient as well as smarter mobile life. Although today's mobile devices become much more powerful than ever with greater computing capability and longer battery life, it might notice that not every people is able to be equipped with the newest and most powerful mobile devices. This indicates that significant heterogeneity (of available storage, CPUs, and batteries) exist between peoples' mobile devices. Furthermore, such heterogeneity will also emerge due to the different preferences of how people to use mobile devices, and sometimes leads related services to interrupt. It is an interesting yet much challenging topic to keep the accessibility of mobile services.

A nature way to tackle this challenge is to employ cloud computing by offloading the computation tasks to remote servers (aka on the cloud). For example, when the local mobile device needs to recognize the man in a picture, it only needs to upload this picture to the cloud and waits for a remote response of the final recognition result. However, there are two major concerns about this kind of could-computing based method: The first is the data transmission will consume a great amount of bandwidth for the cloud side. The traffic loads will get heavier as the users

Q. Yang · X. Luo · P. Li (✉) · T. Miyazaki
School of Computer Science and Engineering, University of Aizu, Aizuwakamatsu, Japan
e-mail: d8192105@u-aizu.ac.jp; d8202105@u-aizu.ac.jp; pengli@u-aizu.ac.jp;
miyazaki@u-aizu.ac.jp

B. Ye et al. (eds.), *2nd International Conference on 5G for Ubiquitous Connectivity*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-22316-8_1

accumulate, and eventually make adverse impact on the cloud's QoS; The second concern is the transmission latency between the local mobile device and the remote cloud. In some emergency scenarios, users might expect near real-time response from the remote, while the transmission latency will be a big problem. To address these challenges, local offloading like fog computing and mobile edge computing is developed. So that many mobile devices now are able to contribute great GPU or even NPU computation capability. These mobile devices can partially serve the role of the cloud computation, and is fast to connect through Wi-Fi, Bluetooth, or near field communication(NFC).

In this paper, we propose employ local offloading to enable collaborative inference among local mobile devices. We first use a random structure to model the connections among mobile devices regarding their mobility. After the local link connections between mobile devices are established, the transmission latency on each link is assumed to remain constant yet various from each other. In what we show later the practical inference procedure is near real-time, mobile devices therefore are reasonably regarded as staying static until they receive the computation results. Then to accurately select the best local mobile device as the computation node illustrated in Fig. 1, our main concern naturally focuses on minimizing the whole time costs, which are induced by the data/result transmission between the computation nodes and the user nodes (that offload computation tasks), and task computation in the computation nodes. Unfortunately, the local connections will be updated from time to time due to the mobility of local mobile devices, making the optimal selection of computation nodes at one time not always suitable to the next time. This requires our collaborative inference scheme to not only find the optimal set of computation nodes in a short time, but also to be able to track the optimal

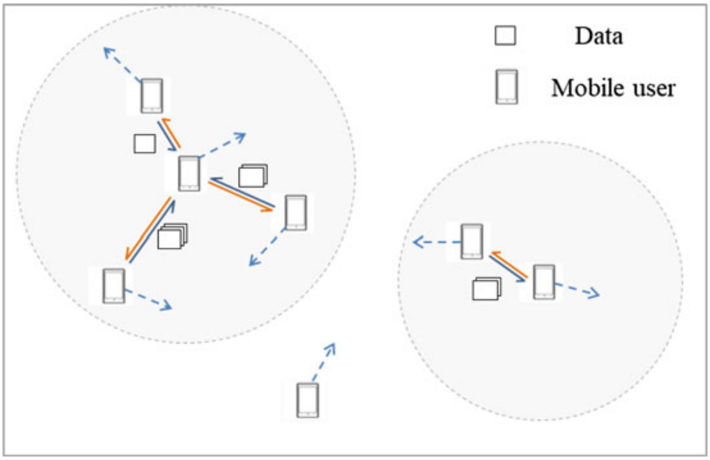

Fig. 1 System overview

selections in a dynamic environment. Obviously, optimization methods that are capable of constantly adapting the solution to a changing environment are expected. To meet this demand, we propose to employ partial swarm optimization (PSO) that is a versatile population-based stochastic optimization technique, to help design our collaborative inference scheme. The contributions of our work are summarized in the followings:

- We propose a system model with random structures to describe the (locally) collaborative inference among mobile services;
- We design an algorithm based on PSO to efficiently minimize the total time costs for collaborative inference, with a dynamic procedure of selecting the optimal computing nodes from local mobile devices;
- We conduct extensive simulations to evaluate the performances of our proposed algorithm, and demonstrate its comprehensive advantages to the optimal results obtained from Gurobi.

The rest of this paper are organized as below: in Sect. 2, we introduce the motivation of our work; in Sect. 3, we firstly formulate our system model and then we detail the PSO algorithm which is used to solve our problem; and the corresponding evaluation results are presented in Sect. 4. We demonstrate the related works in Sect. 5, and finally conclude this paper and in Sect. 6.

2 Motivation

We conduct experiments using three typical CNN models (AlexNet [10], Goog LeNet [13], and Lenet [11]) using Nvidia GTX1080. We collect GPU runtime information of the inference time by using Linux shell command. The CNN models are trained by Caffe [8], a popular open-source conventional neural networks framework which is widely used in both academia and industry. The training process and inference data of Lenet come from caffe models. As for GoogLeNet and AlexNet, we construct two figure recognition models with 209 classes in order to keep same model size with Lenet. The three models have the same amount of inference data(10000) downloaded from Imagenet [4]. And then, inference time under different batch sizes as shown in Fig. 2. The lines in Fig. 2 represents inference time per image on each architecture, with a function of image batch size(from 1 to 512). We notice that inference time across different batch sizes with a logarithmic ordinate. Missing data points are due to lack of enough graphics memory required to process larger batches. The inference time costing gradually decrease as the increase of batch size. Motivated by the trend, We think that to collect multi-users' data to handling by paralleling will be better than individually, though the transmission delay should be considered. We also find the view is novel and meaningful to research. We will introduce the model about the view detailed in next section.

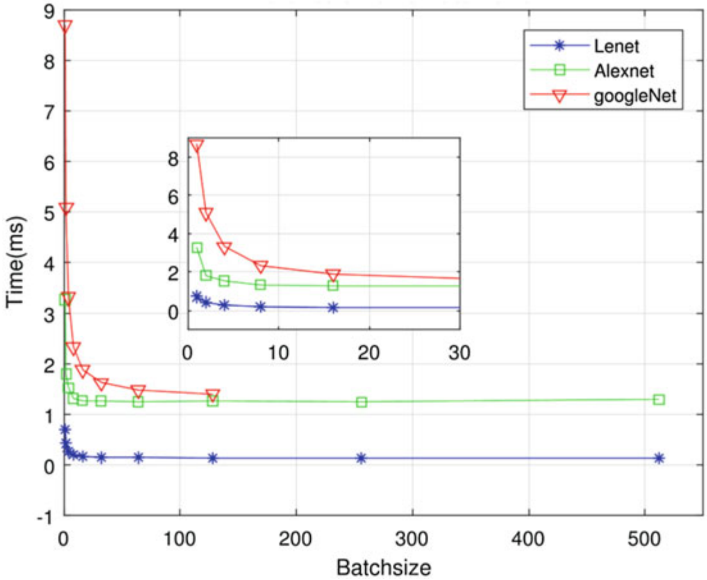

Fig. 2 Inference time *vs.* Batch size. This chart shows inference time across different batch size. Missing data points are due to lack of enough graphic memory required to process larger batches

3 System Model

We consider a set N of mobile devices running an application powered by deep neural networks (DNNs). The DNN model has been well trained and installed on mobile devices. Each device $i \in N$ has an amount of n_j data to process using equipped mobile GPU. These devices can connect with each other using direct links, e.g., Bluetooth and WiFi Direct, or cellular links. The neighbors of node i is included in set N_i. The communication delay between two devices i and j is denoted by d_{ij}. As we have shown that ML workload batching on GPU can effectively reduce the processing time, multiple mobile devices can aggregate their workloads on a single one.

We define a binary variable x_i to indicates whether node $i \in N$ is an aggregation node.

$$x_i = \begin{cases} 1, & \text{aggregation node} \\ 0, & \text{otherwise.} \end{cases}$$

Note that some nodes process only their own data, without receiving workloads from other nodes. We also treat them as aggregation nodes with $x_i = 1$. Each non-aggregation node may connect to multiple aggregation nodes. We define a variable y_{ij} to indicate the portion of workloads offloaded from node i to j. Since aggregation nodes do not offload their workload to others, we have $\sum_{j \in N_i} y_{ij} = 0$. For each non-aggregation node, we have $\sum_{j \in N_i} y_{ij} = 1$. In summary,

$$y_{ij} \le x_i, \forall i \in N, j \in N; \tag{1}$$

$$x_i + \sum_{j \in N} y_{ij} = 1, \forall i \in N. \tag{2}$$

We define a total cost T_i of node i as the sum of its computation and communication cost, i.e.,

$$T_i = 2 * \sum_j d_{ij} * y_{ij} + f\Big(\sum_j y_{ji} * n_j\Big), \tag{3}$$

where $f(\cdot)$ is a non-decreasing function that describes the relationship between GPU processing time and workloads. With an objective of minimizing the total of cost among all mobile devices, our studied problem can be formulated as:

$$\min \sum_{i \in N} T_i$$
$$\text{subject to: (1), (2) and (3).}$$

In this section, we propose a heuristic algorithm based on particle swarm optimization (PSO), which is motivated by the phenomenon of bird predating. The key of PSO is to iterative improve a candidate solution with regard to a given measure of quality [12]. The solution obtained by the PSO may not be a theoretically optimal, but it can quickly generate a solution with satisfied performance in practice.

3.1 Procedure for PSO

The Particle Swarm Optimization procedure consists of $\boldsymbol{v_i}$ and $\boldsymbol{X_i}$ two parts' update, as shown in line 6 of algorithm and line 7 of algorithm. The $\boldsymbol{v_i}$ is a group of randomly generated feasible break-reconnect information which consists of n_i, m_i and r_i, represented by (4).

$$[n_i, m_i, r_i] \in \boldsymbol{v_i} \tag{4}$$

The n_i denotes the non-aggregation node which should removed from m_i. The m_i denotes the aggregation node which connected to n_i. The r_i denotes the other aggregation node. A simple instance shown in Fig. 3 is used to explain the update for $\boldsymbol{X}$ in line 6 of algorithm, in which the structure of $\boldsymbol{v_i}$ and $\boldsymbol{X_i}$ are provided. In

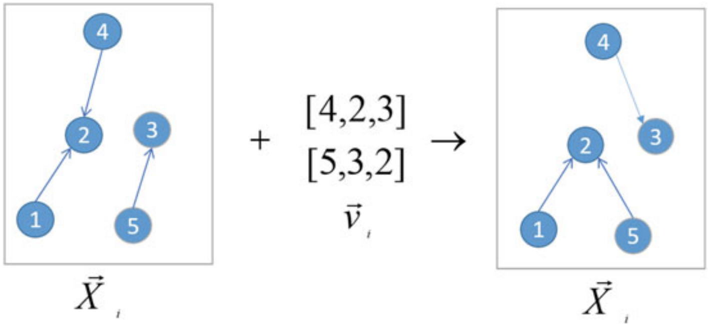

Fig. 3 Simple instance for line 7 of algorithm

Table 1 Variables and symbols

Notations	Description
$\boldsymbol{X_i}$	The present solution
$\boldsymbol{v_i}$	A group of randomly generated feasible break-reconnect information
$\boldsymbol{p_d}$	The local optimum in dth loop
p_g	The global optimum in all previous $\boldsymbol{p_d}$
w	The inertia weight
c_1, c_2	Acceleration constants, which are used to adjust step
h_1, h_2	Two random functions, whose field is [0,1], which are used to increase search randomness
$\boldsymbol{V}$	The set of $\boldsymbol{v_i}$
n_i	The non-aggregation node which should removed from m_i
m_i	Denotes the aggregation node which connected to n_i
r_i	Denotes the other aggregation node
F	The fitness for swarm
f	Describes the relationship between GPU processing time and workloads
P	Particle swarm, it is equal to the dimensions of $\boldsymbol{V}$

addition, R in line 6 of algorithm represents a set operations generate randomly in each iteration. The other major notations used in this algorithm are summarized in Table 1.

For instance, We assume there are five mobile nodes in the model, which may construct many different typologies as solutions for information propagation. We use $\boldsymbol{X_i}$ to represent the ith solution, which shown in the left of Fig. 3. The ith solution has two aggregation nodes whose number is 2 and 3 with other non-aggregation nodes connected. In order to get the optimal solution, the algorithm gives a operation set to make the ith solution converge to optimal solution, that is giving it a 'velocity' v_i towards the new solution.

The diagram shows the instance v_i as $[4, 2, 3]$ and $[5, 3, 2]$, in which the first set means the operation for non-aggregation node 4 would break the connectivity with aggregation node 2, and create a new connectivity to aggregation node 3; the similar operation towards the latter set. We named the aforementioned operation as *break-reconnect*. After the operation, the ith solution gets update, as shown in the right of Fig. 3.

As for a large model, there are much more complicated conditions that includes isolated aggregation node without nodes connected. There is also the condition for solution that makes aggregation node convert into non-aggregation,and then reconnect to other aggregation node. Hence, the previous non-aggregation nodes need to reallocate.

Line 6 of algorithm denotes how to get new break-reconnect information, and line 7 of algorithm represents how to get new solution. The right side in line 6 of algorithm is comprised of break-reconnect information, local optimum and global optimum. Then, according to update $\boldsymbol{v_i}$, the $\boldsymbol{X_i}$ gets its update. Parameters c_1 and c_2 are used to adjust the maximum step of iteration. In addition, h_1 and h_2 are two random numbers which contributed to search randomness.

3.2 Description of the Algorithm

First, in the initialization step, input $\boldsymbol{V} = \{\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_p}\}$ is a set of $\boldsymbol{v_i}$. The operation for particle made by the break-reconnect information $\boldsymbol{v_i}$ is to remove the connectivity between the n_i and m_i, and then create a new connectivity for n_i with the aggregation node r_i. P is particle swarm, which is equal to the dimensions of $\boldsymbol{V}$. After input $\boldsymbol{V}$ and P, we initialize $\boldsymbol{p_d}$ as p_0 which denotes the local optimum related to best solution in P before the loop start.

Then, we should consider all kinds of structures of $\boldsymbol{v_i}$. Firstly, we should consider when the n_i transformed from general node to a aggregation node, the n_i need to disconnect the connectivity without establishing any connection with others aggregation nodes at the time. Under the situation, we set $c_i = 0$. Similarly, there is also a case where r_i in $\boldsymbol{v_i}$, which means that the n_i who is the aggregation node in the old particle should be converted into the non-aggregation node in new particle and establish connectivity with aggregation node m_i. At this time, if the n_i acted as aggregation node in old particle has no connectivity with other non-aggregation nodes, then we just create a connection to aggregation node m_i in our algorithm.

When go in the loop, m_i, r_i, $\boldsymbol{v_i}$ and $\boldsymbol{X_i}$ are updated in each iteration. In addition, the fitness(k) can be calculated according to (3) and $f(\cdot)$. Once get the fitness value F, the local optimum and global optimum can be determined by line 1 of algorithm and line 15 of algorithm.

If the n_i is selected as the aggregation node by other general nodes in the old particle, we need to look for the computation which meet the condition that can be connected with other nodes in new particles. In addition, through analysis, it can be found that m_i and r_i are not equal to 0 at the same time. Finally, the output p_g is the global optimum we seek to.

Algorithm 1: Implementation of PSO algorithm

```
Input: $V$, $P$;
$d = 1$;
while $d \leq Loop$ do
    for $k$ in $P$ do
        for $i$ in $N$ do
            $\boldsymbol{v_i} \leftarrow w * \boldsymbol{R} + c_1 h_1 \otimes (\boldsymbol{p_d} - \boldsymbol{X_i}) + c_2 h_2 \otimes (p_g - \boldsymbol{X_i})$;
            $\boldsymbol{X_i} \leftarrow \boldsymbol{X_i} + \boldsymbol{v_i}$;
        end for
        $F \leftarrow Fitness(k)$;
        if $\boldsymbol{p_d} > F$ then
            $\boldsymbol{p_d} \leftarrow F$;
        end if
    end for
    if $p_g > \boldsymbol{p_d}$ then
        $p_g \leftarrow \boldsymbol{p_d}$;
    end if
    $d = d + 1$;
end while
Output: $p_g$;
```

4 Performance Evaluation

4.1 Settings

In this subsection, we implement an extensive simulation to evaluate the performance of PSO algorithm. We compare our approach with the optimal solution implemented by Gurobi that is state-of-the-art (http://www.gurobi.com/products/features-benefits). In the evaluation, assuming there are no more than 30 users in our experiment environment, because of limitation of the license of Gurobi. The algorithms write in Python, and the program runs on DELL, whose CPU Core is i5@2.30GHZ and memory 16GB. At the beginning, we set the popsize as 30 for our PSO. In other words, there are 30 groups random solution initially. The weight w and $Loop$ are normally recommended as 0.5 and 50.

4.2 Results Prediction

In this subsection, we show the experimental results for overall prediction performance of the proposed model. In Fig. 4, we mainly compare three conditions Random, PSO, and Optimal. Considering the Random, whose fitness calculated with

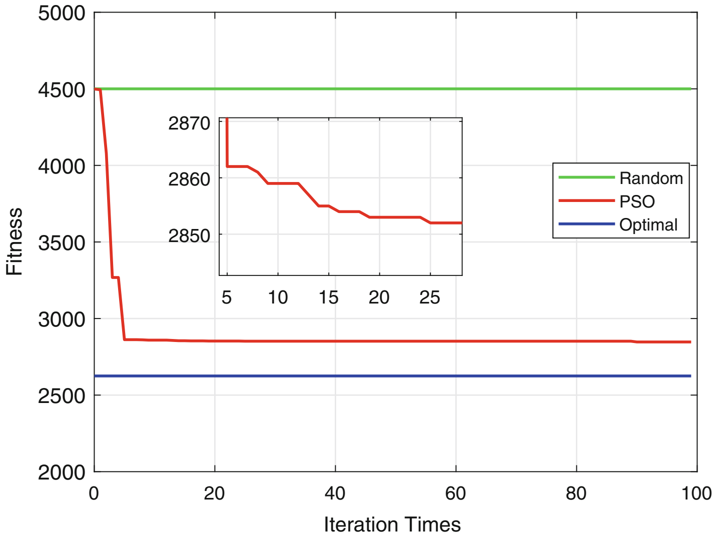

Fig. 4 Performance analysis comparison of the proposed algorithm with Gurobi and all users act as aggregation node

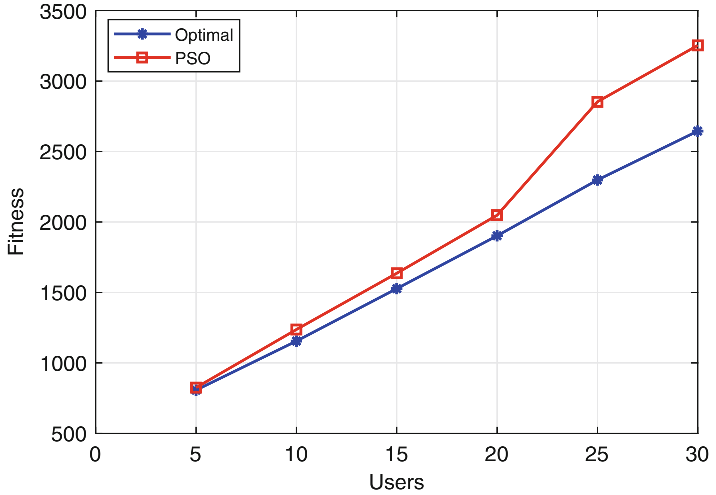

Fig. 5 With the increment of the magnitude of users *N*, which ranging from 1 to 30, the fitness of Optimal and PSO maintain the same growth trend

the condition that each user is treated as aggregation node. The red line denotes the optimal fitness generated by Gurobi, with 30 users in the environment. The green curve represents the process of iteration of fitness for our algorithm. As shown, when the iteration times exceeds 30, the green line tends to converge, whose value is far smaller than Random and close to the optimal. In Fig. 5, with the increase of

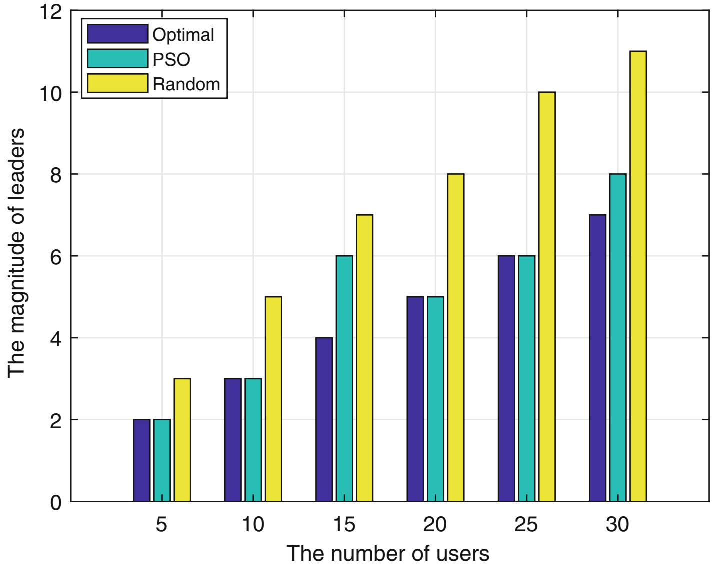

Fig. 6 The magnitude of aggregation nodes selected by different methods

the magnitude of users N, which range from 1 to 30, the fitness of Optimal and PSO maintain the same growth trend. And the value of fitness gained by PSO is very close to Optimal. In Fig. 6, the number of aggregation nodes of Random generated randomly at the beginning of initialization. Compared with Random, the number of aggregation nodes of Optimal and PSO is always smaller.

5 Related Work

5.1 Inference Process in Machine Learning

Much research work about efficiency in machine learning and cooperation in the mobile cloud has been done. In [3], Sharan Chetlur et al. improve the performance by 36% for convolution neural networks on caffe framework and reduce the memory consumption. In [1], Alfredo et al. do any analysis on accuracy, power consumption, inference time, memory footprint by experiments on framework named caffe. In [6], Han et al. benchmark the layer-wise speed up on CPU, GPU, and mobile GPU by deep compression for networks. In [14], Tang et al. propose a client-architecture where training process is implemented in a server, and then mobile device download the trained predictor from the server to make transmission decisions.

5.2 Job Scheduling and Cooperation

In [2], the lab established a SmartLab with 40 Android devices which cloud provide an open testbed to facilitate research and smart phone applications can be deployed massively. In [7], Heyi et al. propose a back-end general architecture which is able to require crowd-sourcing for mobile applications. And they adopt Microsoft Azure cloud computing platform to deploy their back-end. In [15], Yao et al. introduce a mobile cloud service framework based on crowdsourcing which meets mobile users requirement by sensing their context information and provide corresponding services to each of the users. In [9], Considering a dynamic network in which mobile devices may join and leave the network at any time, Ke et al. merge crowdsourcing into existing mobile cloud framework where data acquisition and processing can be conducted. In [5], Fan et al. propose a novel privacy-aware and trustworthy data aggregation protocol based on the malicious behavior like submitting data to damage the fog system for mobile sensing.

6 Conclusions

In this paper, we construct a model for transmission to implement local cooperation among mobile devices motivated by the inference process of machine learning, which can be used to guide the mobile users to choose which approach to handle their data for the goal of improving efficiency. By performance evaluation, we find that the collaborative inference scheme can reduce global dealing time in given field compared with handling the data which is affected by the high transmission latency between mobile device and cloud. As a global optimization random search algorithm, the particle swarm optimization algorithm has the characteristics of fast convergence and high precision.

References

1. Canziani, A., Paszke, A., Culurciello, E.: An analysis of deep neural network models for practical applications (2016). Preprint. arXiv:1605.07678
2. Chatzimilioudis, G., et al.: Crowdsourcing with smartphones. IEEE Internet Comput. **16**(5), 36–44 (2012)
3. Chetlur, S., et al.: cuDNN: efficient primitives for deep learning (2014). Preprint. arXiv:1410.0759
4. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database.In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
5. Fan, J., Li, Q., Cao, G.: Privacy-aware and trustworthy data aggregation in mobile sensing. 2015 IEEE Conference on Communications and Network Security (CNS). IEEE, Piscataway (2015)

6. Han, S., Mao, H., Dally, W.J.: Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding (2015). Preprint. arXiv:1510.00149
7. Heyi, M.H., Rossi, C.: On the evaluation of cloud web services for crowdsourcing mobile applications. In: 2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech). IEEE, Piscataway (2016)
8. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: UC Berkeley Eecs, Caffe: convolutional architecture for fast feature embedding. In: ACM Multimedia (2014)
9. Ke, H., Li, P., Guo, S.: Crowdsourcing on mobile cloud: cost minimization of joint data acquisition and processing. In: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, Piscataway (2014)
10. Krizhevsky, A., Sutskever, I., Hinton Geoffrey, E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NIPS) (2012)
11. LeCun, Y., Jackel, L.D., Bottou, L., Cortes, C., Denker, J.S., Drucker, H., Guyon, I., et al.: Learning algorithms for classification: a comparison on handwritten digit recognition. Neural Netw. Stat. Mech. Perspect. **261**, 276 (1995)
12. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization. Swarm Intell. **1**(1), 33–57 (2007)
13. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (2015)
14. Tang, Z., et al.: Energy-efficient transmission scheduling in mobile phones using machine learning and participatory sensing. IEEE Trans. Veh. Technol. **64**(7), 3167–3176 (2015)
15. Yao, D., et al.: Using crowdsourcing to provide QoS for mobile cloud computing. IEEE Trans. Cloud Comput. **7**(2), 344–356 (2015)

Compressed Sensing Channel Estimation for LTE-V

Kelvin Chelli, Ramzi Theodory, and Thorsten Herfet

1 Preliminaries

With the advent of 5G and the convergence of broadcast and broadband technologies, the consequences of high mobility at both the transmitter and receiver along with the methods to compensate the same has become an important consideration in the design and development of modern telecommunication systems. The Release-14 of LTE introduces various improvements to the standard to enable vehicular communication [2].

Vehicular environments are characterized by varying degrees of mobility resulting in a heterogeneous channel. In cases of high mobility, a temporally-varying multipath channel that displays selectivity in both the frequency and time domains whereas, under low mobility a pure frequency selective channel is present. Thus, channel estimation schemes have to work robustly in these heterogeneous channel conditions. Moreover, the computational complexity of these schemes must be relevant for consumer hardware implementation.

In our paper, we develop a scheme for channel estimation that takes into consideration the temporal variations in the channel and that is able to provide good results with normalized Doppler shifts of up to 10%. The *Rake-Matching Pursuit* (RMP) algorithm is a *Compressed Sensing* (CS) scheme that is able to exploit the inherent sparsity of wireless channels and supply a precise estimate of a channel that is doubly selective [4]. On the other hand, a simple scheme like the *Least Squares* (LS) estimator is used in low mobility conditions. Switching between the two schemes is enabled by a simple cognitive framework based on the *Index of Dispersion* that determines the time variation of the channel.

K. Chelli (✉) · R. Theodory · T. Herfet
University in Saarbrücken, Saarbrücken, Germany
e-mail: chelli@nt.uni-saarland.de; herfet@nt.uni-saarland.de

B. Ye et al. (eds.), *2nd International Conference on 5G for Ubiquitous Connectivity*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-22316-8_2