# Big Data Processing with Minimal Delay and Guaranteed Data Resolution in Disaster Areas

Junbo Wang, *Member, IEEE* Koichi Sato, *Member, IEEE* Song Guo, *Senior Member, IEEE*
Wuhui Chen, *Member, IEEE* Jie Wu, *Fellow, IEEE*

***Abstract*—Big data analysis is very important to support rescue activities when natural disaster happens, through understanding various situations such as power/water outage regions. The traditional way to process big data is based on high performance computation/storage resources in a cloud center. However this is hard to be guaranteed in a disaster scenario due to destruction of communication infrastructure. Meanwhile, high latency between local sensing devices and cloud center sets a big obstacle to enabling a near real-time big data analysis. On the other hand, movable BS such as vehicle-based MDRU developed by NTT, is a possible solution to reconstruct an emergency communication network and process data at the edge sites with reduced data transmission time. In this paper, we study the optimal overall delay in a fog/edge computing platform constructed by vehicle-based MDRUs with guaranteed data resolution. We formalize the problem as a Mixed Integer Nonlinear Program (MINLP) which is a well-known NP-hard problem, and then relax the original problem to a MILP. Finally, we propose a two-stage heuristic algorithm to solve it in a time-efficient manner. Through evaluation, the effectiveness of the proposed heuristic approach has been validated in terms of minimizing overall delay with sufficient given data resolutions.**

***Index Terms*—Big Data Processing, Fog/Edge Computing, Data Resolution, Disaster Scenarios.**

## I. Introduction

Big data analytics is to find hidden patterns behind the data, which can play an important role in supporting rescue activities after disaster. For example, Kwan et al. [1] explore an emergency response service by detecting obstructions caused by a major disaster, such as a hurricane using LiDAR data.

Unfortunately, communication systems can be destroyed after a disaster. For example, communication networks in the disaster areas lost their functions in the Great East Japan earthquake [2]. To deal with this hash scenario, movable base stations (MBSs), such as vehicle-based MDRUs from NTT [3], [4], [5], can be deployed in disaster areas to reconstruct a communication network [6], [7]. This emergency communication network brings new challenges to big data analytics, since it is generally conducted with high-performance PCs and servers. The transmission delay will be extremely long when raw data are collected from mobile phones to cloud center, via an MBS-based network.

Junbo Wang is with School of Computer Science and Engineering, The University of Aizu, Aizu-wakamatsu, Japan.
E-mail: j-wang@u-aizu.ac.jp

Manuscript received April 19, 2005; revised September 17, 2014.

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

To reduce this delay, fog/edge computing [8] gives a possible solution by offloading computation tasks from cloud to edge devices, e.g., routers, closer to the users. Fog computing has been investigated with different architectures for different applications [9], and generally it has the following types: Virtualized Fog Data Centers (V-FDCs) [10], Fog Radio Access Networks (F-RANs) [11], and Cloudlet [12]. In [10], multiple V-FDCs are considered to offload the service from traditional massive data centers to local fog nodes. Cloud Radio Access Networks (C-RANs) [13] are proposed to improve Spectral Efficiency by more efficient interference management due to the virtualization of baseband processing of Remote Radio Heads (RRHs). F-RANs [11] can further reduce the traffic overhead and latency by adding caching and signal processing capabilities in fog layer. Cloudlets [14], [12] can be deployed in an urban city to provide closer computing and storage to users. However when considering disaster scenarios, fog-supported big data processing still has the following challenges:

- MBSs-based emergency communication network is generally with a mesh structure, which is not matched with the above studies. It becomes a new challenge to study fog-supported big data processing in a mesh-based network considering limited computation and communication resources in disaster areas.
- Generally speaking, information loses through the data processing in fog layer. For the areas suffered serious damages (we call them prioritized areas), the information detail should be kept as much as possible for comprehensive situation understanding through data analysis. However, how to quantify information detail, and guarantee sufficient information detail in priority areas is a new challenge in fog-supported big data processing in disaster areas.
- Time-efficient data processing becomes critical in MBS-based emergency communication network that guarantees the information quality in prioritized areas.

To represent information detail in data, we propose a new concept called data resolution. Image resolution is the concept of representing how many details an image holds. Data resolution has a similar meaning, but shows the detail that the data keeps. Higher resolution means more information details in the data. It is a very important concept for studying big data analytics in fog computing, since losing too many details in the fog layer will impact the data mining result at the cloud site.

To this end, in this paper, we study big data processing with minimal delay and guaranteed data resolution in disaster areas. We consider that some MDRUs connect with the outside of disaster area via satellite, and other MDRUs can decide to process data locally or in cloud, and forward data to the MDRUs connecting with satellite hop by hop. We formalize the problem as a Mixed Integer Nonlinear Program (MINLP) which is a well-known NP-hard problem, and then relax the original problem to a MILP. Finally, we propose a two-stage heuristic algorithm to solve it in real-time.

To the best of our knowledge, this paper provides the first study on data offloading problem by involving data resolution in fog computing platform for disaster scenarios. The main contributions are summarized as follows:

- We formalize a fog-supported big data processing model in wirelessly connected disaster areas, including the overall delay and data resolution variation.
- We formalize a MINLP to minimize the overall delay with guaranteed data resolution. It is further linearized to a MILP by introducing new variables. Finally, we propose a two-stage heuristic algorithm to round the integer variables in MILP and solve it in a real-time manner.
- We perform a comprehensive evaluation, by considering variation of different variables and size of network, to show its performance.

This paper is organized as follows: section II gives a review of related works; section III discusses system model; section IV presents problem formulation to minimize the overall delay with guaranteed data resolution; section V describes linearization of the problem and introduces a two-stage heuristic algorithm to solve it in a real-time manner; section VI shows the evaluation methodology and results; finally, section VII concludes the paper.

## II. Related Works

### A. Big Data Analytics in Disaster Scenarios

Big data analytics in cloud has been well studied [15] and a comprehensive survey [16] has summarized big data techniques applied to the disaster management. Big data also presents its power in disaster scenario through several typical applications e.g. network planning after disaster occurrence [17], crisis response [18], user mobility [19], sentiment analysis [20] and so on.

For example, data synchronization problem among several isolated services after disaster occurrence has been studied in [21]. They consider the size and the priority of the data at each isolated data server, and then further formalize the problem as a stochastic program. However, it causes heavy traffic burden to synchronize raw data among the isolated servers after disaster occurrence. Thus it is more suitable to transmit the pre-processed data to the cloud, e.g., the data after removing redundant information. Meanwhile, geo-located data becomes a popular data resource for situation understanding after disaster occurrence, e.g., geo-located message or image data. In [22], geo-located twitter data are used to understand power shortage situations at New York city after Hurricane Sandy in 2012. They have proposed a dynamic spatial clustering algorithm to process big data in an efficient way. Rathore et. al. propose a system to process big data from geo-social networks in [23]. The data processing goes through preprocessing (i.e., filtering and classification), text analysis and so on, to provide services for decision-making. However, the data transmission delay from the sensing devices/local servers to the cloud can be extremely big, considering unstable internet access situations after disaster occurrence. The preprocessing tasks, e.g., filtering and text analysis, are possible to be performed at local computation units, to reduce transmission data size from local server to the cloud. In this paper, we investigate a new computation and communication structure by offloading preprocessing tasks from cloud to local computation units. The related literature in fog-computing supported big data processing is discussed as follows.

### B. Fog Computing Supported Big Data Processing

Fog computing is first introduced by Cisco [8] and leverages cloud resources for edge devices, bringing the computation and communication resources closer to the users. Fog computing can extend cloud functionality from remote cloud center to fog nodes nearby the users as studied in Virtualized Fog Data Centers (V-FDCs) [10], and can enhance wireless caching function extended from cloud in Fog Radio Access Networks F-RANs [11]. However, it is very difficult to adopt the above existing research results into disaster scenarios, considering the destroyed communication infrastructure, and the limited computation and communication resources in a MBS-based communication networks.

An edge-based computation infrastructure was proposed in [24] based on ad hoc connectivity in disaster scenarios. And to process IoT data in disaster scenarios, a fog-computing based structure was designed in [25]. However, even they presented possibility to apply a edge/fog-based structure in disaster scenarios, unfortunately they did not study detailed algorithms to enable an efficient computation and communication platform. E. M. Trono et. al. in [26] studied a edge-computing supported data processing architecture based on DTN (Delay Tolerance Network) in disaster scenarios. It infers a pedestrian map based on the trajectory information collected from smart phones. In the proposed system, local computation nodes process the trajectory data collected in their covered areas, and generate subgraphs in a parallel way. However, sharing of data processing tasks among local and cloud nodes is not considered in the study, so that it is hard to find an optimal solution to enable efficient data processing and transmission. Y. Wang et. al. in [27] have formalized a big data processing problem in disaster scenario based on a tree-based communication network constructed by ad hoc connectivity. Each node can decide the ratio of data to be processed at local or at cloud. However it is more general to apply a mesh-based communication network as studied in this paper, to achieve a more efficient usage of communication and computation resources.

Cloudlet has similar and possible network structures for disaster senarios, since it can construct an ad-hoc network as studied in Wireless Metropolitan Area Networks [12] and Mobile Computing Environment [28]. In [12], authors studied cloudlet placement problem and proposed a set of placement method to minimize response time, e.g., heaviest-AP first placement strategy. And further the solution is enhanced to deal with workload balancing problem when one user can connect with multi-cloudlets at the same time in [29]. However the sharing of processing tasks among different Cloudlets is not tackled in this sutdy. An auction-based incentive mechanism is proposed in [28] to share resources among cloudlets in an optimal way. Meanwhile, base stations in communication infrastructure can be considered as fog nodes to support medical cyber-physical system [30] or software-defined embedded system [31]. However, the above research results cannot be directly adopted into this study, since (1) this study tackles a special architecture in disaster scenarios, where some of MBS are connecting with satellite, and others forward data to them based on a mesh-structure; (2) the existing researches treat all fog nodes as same priority level, which is not match with disaster scenario that prioritized areas need more detailed information for situation understanding.

## III. Models

### A. System Model

The system model is represented in Fig. 1. After a disaster, movable base stations (MBSs, a general representation of MDRU in Section I) can be deployed in the disaster-stricken areas, to reconstruct a communication network. Users can upload data through the MBSs.

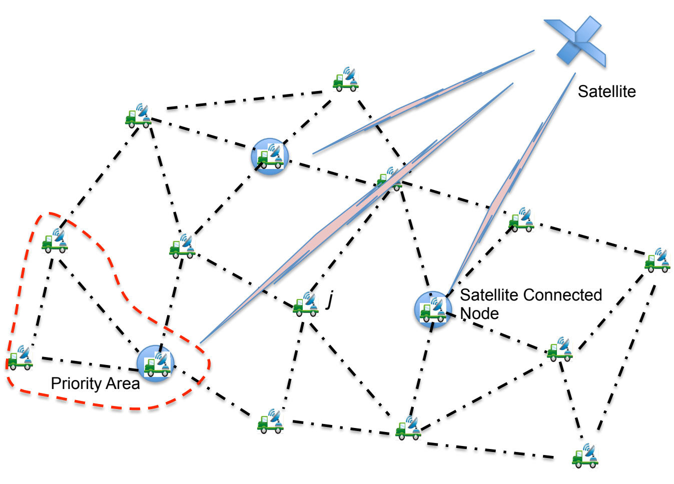



Fig. 1. System Model

Let $j$ denote an MBS, which integrates both wireless communication and computation functions. First, smartphone users connect with $j$ and upload data; the data size is represented as $d_j$. Then we assume a part of the big data processing algorithm has been deployed in the MBS beforehand, and that the data size compression ratio after processing in node $j$ is $\rho$, where $0 < \rho \leq 1$. The processing rate in each MBS $j$ is represented as $\mu$.

We also assume that the MBSs are connected via a wireless medium, and that the communication rate is represented by $R_1$, for each link between two MBSs. MBSs are connected in a mesh shape for data transmission. We assume there are several MBSs as marked in Fig 1, which have the ability to communicate with satellite, to transmit generated data outside of disaster area. The transmission speed is denoted as $R_2$. Each node $j$ can decide to process parts of data in $k$, its neighbor nodes, and cloud. The decision variable is shown in Table I for details.

Meanwhile, after data is processed locally, information in data will be lost, while data resolution represents the details that the data keeps similarly to image resolution. Raw data has the maximum data resolution i.e. 1 or lossless. The parameter $\xi$ is defined to be the data resolution variation after data processing. The notations can be found in Table I.

TABLE I
NOTATIONS

| | |
|---|---|
| **Constants** | |
| $J$ | A set of Movable BS (MBS), while $i$, $j$, $k$, $m$ are instances of MBS. |
| $d_j$ | Raw data collected in $j$. |
| $h_{jk}$ | Hops from MBS $j$ to $k$. |
| $\lambda_j$ | Average generation rate of data processing task in $j$. |
| $\mu$ | Data processing rate in local computing node. |
| $R_1$ | Communication speed between two MBSs. |
| $R_2$ | Communication speed with satellite. |
| $\rho$ | Data compression ratio after data is processed locally. |
| $\xi$ | Data resolution variation after data is processed locally. |
| **Variables** | |
| $z_k$ | A binary variable to indicate whether MBS $k$ is a connecting node with satellite. |
| $l_{jk}$ | A raw data transmission link from $j$ to cloud via $k$, when $k$ is a connecting node with satellite. |
| $y_{mk}$ | A binary variable to indicate whether MBS $m$ offloads data processing task to MBS $k$. |

### B. Service Model

In this study, we consider that there are limited number of MBSs connecting with satellite directly, since communication resources may not enough after disaster occurrence. For the other MBSs, the generated data are forwarded to the MBS connecting with satellite hop by hop. Meanwhile, for each MBS, we consider that there is at least one path to the cloud. We consider cloud server is located outside of disaster area, and the data generated inside of disaster area are transmitted to the cloud via satellite communications. The capacity of cloud server is considered as sufficient for data processing in disaster scenario.

### C. Data Compression Ratio $\rho$

The data compression ratio depends on mining types and data reduction technologies in the data mining algorithms. Some types of data reduction technologies in spatial clustering algorithms are summarized as follows to explain the data compression ratio.

- Representatives based on *Core-points*: The most basic technique in data reduction is to represent clusters by core points inside of each cluster. For example, in DBSCAN

[32], core points are selected to represent a cluster if in a given radius (*EPs*), at least a minimum number of points (MinPts) exists with the cardinally of the neighborhood exceeds some threshold.

- Representatives based on *Specific Core-points*: Specific core-points are several special points selected from the group of core points, to represent the cluster. Thus data size can be further reduced by representing clusters by specific core-points. The spatial clustering algorithm Mr. Scan uses minimum set of the core points to represent a cluster in [33][34]. For a grid cell of arbitrary density, eight points are selected in [33][34] to represent all core points in a grid cell.
- Representatives based on *Boundary Points*: For example in [35], Boundary Points are used to be representative points. The number of boundary points depends on the covered area of the spatial clustering.

Different data reduction technologies can achieve different data compression ratios. Generally, (1) has smallest data compression ratio, while it increases when adopting data reduction technology (2) and (3).

## IV. Problem Formulation

Generally speaking, data analysis in cloud centers will not start until all the data is collected from the whole area. To enable quick decision-making in the disaster scenario, in this paper, we investigate the minimal overall delay between mobile phones and the cloud, through offloading data processing tasks locally.

### A. Data Processing Delay in $k$

Data processing delay for each node $k$ depends on where the processing task is allocated, computation rate and idle situation of the computation node. Suppose node $k$ acts as a computation node/server with computation rate $\mu$, and is shared by multiple clients for data processing. Consider data processing tasks coming from other nodes randomly following poisson process. The overall data processing task arrival rate can be calculated as follows,

$$\Lambda_k = \sum_{m \in J} \varpi / d_m * y_{mk} \lambda_m, \forall k \in J \tag{1}$$

where $\varpi$ is a constant value that represents a specific size of data task is offloaded from one node to the other node. In this case, data processing task is offloaded from $m$ to $k$. Due to the complexity of the whole model, we assume a static $\varpi$ in this paper. $\lambda_m$ represents generating rate of data processing rate in node $m$, which is directly proportionally to data size $d_m$.

Then task computation time is exponentially distributed on a local node, which can be represented as M/M/1 queue. And the average data processing time in node $k$ for each piece of task can be represented as

$$\tau_k^p = \frac{1}{\mu - \Lambda_k}, \forall j \in J \tag{2}$$

with the following constraints,

$$\mu > \Lambda_k \tag{3}$$

### B. Transmission Delay from a Node $j$ to Cloud

Besides data processing delay, the data transmission delay from a node $j$ to cloud is formalized as follows.

First we consider the delay for the data collected in $j$, processed in $k$, and finally be transmitted to cloud. $D_j^l$ denotes the maximum delay for the data processed locally and can be represented as follows in Eq. (4).

$$D_j^l \geq \varpi y_{jk} (h_{jk}/R_1 + \lambda_j \tau_k^p / d_j + \sum_{i \in J} z_i l_{ki} \rho h_{ki} / R_1 + \rho / R_2) \tag{4}$$

where the data is offloaded from $j$ to $k$, and after data is processed in $k$ with compression ratio $\rho$, it is finally uploaded to cloud via MBS $i$.

Then for the data to be processed in a cloud center, the transmission delay is denoted by $D_j^c$ and represented as follows in Eq. (5).

$$D_j^c = \sum_{i \in J} z_i l_{ji} (d_j - \sum_{k \in J} \varpi y_{jk}) h_{ji} / R_1 + (d_j - \sum_{k \in J} \varpi y_{jk}) / R_2 \tag{5}$$

Data analysis task works after all parts of data from $j$ are collected. The transmission delay is denoted by $D_j$ and represented as follows,

$$D_j = \max \left\{ D_j^l, D_j^c \right\} \tag{6}$$

For the whole network, the overall delay is represented in the worst case as follows,

$$D^o = \max\{D_j, j \in J\} \tag{7}$$

### C. System Constraints

To guarantee the whole system works correctly, the following constraints are considered.

First, we should guarantee all processed data does not exceed the whole generated data size as follows,

$$\sum_{k \in J} \varpi y_{jk} \leq d_j, \forall j \in J \tag{8}$$

Without this constraint, when the system is finding a set of optimal $\{y_{jk}\}$ to maximize the overall delay $D^o$, there is a possibility that the processing data exceeds the whole generated data size.

Then we assume the MBS connecting to satellite is bounded by a constant $Z_0$, due to limited communication resources in disaster scenarios.

$$\sum_{k \in J} z_k \leq Z_0 \tag{9}$$

Meanwhile, we consider a single path and at least one path that exits from each node $j$ to cloud, which is bounded as follows.

$$\sum_{k \in J} z_k l_{jk} = 1, \forall j \in J \tag{10}$$

$$\sum_{k\in J} l_{jk} = 1, \forall j \in J \tag{11}$$

### D. Data Resolution Variation

Image resolution is the concept of representing how many details an image holds. Data resolution has a similar meaning for the information embedded in data. Higher resolution means more information details in the data. It is a very important concept in studying big data analytics in the fog computing, since losing too many details in the fog layer would impact data mining results at the cloud site. Take spatial big data analytics as an example, after data processing in local server, only statistic information for each unit of area will be uploaded to the cloud center, where data resolution/details is correspondingly reduced.

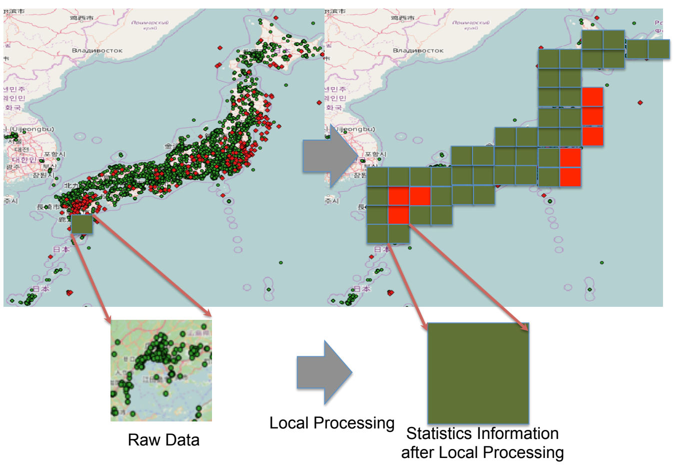



Fig. 2. An Example of Data Resolution Variation

Then the data resolution for data collected from priority areas, denoted by $\Psi^p$, can be represented as

$$\Psi^p = \frac{\sum_{j\in P}\left(d_j - \sum_{k\in J} y_{jk}\varpi(1-\rho)\right)}{\sum_{j\in P} d_j} \tag{12}$$

where $P$ is denoted as a priority area.

To guarantee the propriety area has enough data resolution for situation understanding, we have the following constraint.

$$\Psi^p \geq \Psi^0 \tag{13}$$

### E. An MINLP Problem Formulation

To guarantee a near real-time data analysis to support decision-making after disaster occurs, our goal is to minimize the overall delay by choosing the best setting of $z_k$, $l_{jk}$, and $y_{mk}$. By summarizing all definitions and constraints discussed above, we can formalize Optimal Fog Computing Platform Problem in Disaster Scenario (OFCP-DS) as a mixed-integer non-linear programing (MINLP) problem as follows:

**MINLP: OFCP-DS:**

$$\begin{aligned} \text{minimize} \quad & D^o \\ \text{subject to} \quad & (1)-(11),(13) \\ & z_k \in \{0,1\}, l_{jk} \in \{0,1\} \\ & y_{mk} \in \{0,1\} \end{aligned} \tag{14}$$

## V. A Heuristic Approach

### A. Linearization

MINLP is a well-known NP-hard problem, which may take a extremely long time for optimal solutions. We discuss how to linearize it by introducing new parameters as follows. To reduce the computation complexity, first we relax nonlinear equation in Eq. (6) and Eq. (7) by adding a new variable $D$ and new constraints in $C1$ as follows:

**MINLP: OFCP-DS:**

$$\begin{aligned} \text{minimize} \quad & D \\ \text{subject to} \quad & C1: D \geq D_j^l, D \geq D_j^c, j \in J \\ & C2: (3),(4),(5),(8),(9),(10),(11),(13) \\ & C3: z_k \in \{0,1\}, l_{jk} \in \{0,1\} \\ & y_{mk} \in \{0,1\} \end{aligned} \tag{15}$$

However, there are still non-linear equations in (4) and (5), but fortunately we can linearize them as follows.

First, we introduce a new parameter $\omega_{jk} = h_{jk}/R_1 + \rho/R_2$ to make the equation look simple and have a good shape. Then we substitute Eq. (1) and Eq. (2) into Eq. (4) and achieve

$$\begin{aligned} & D_j^l\mu - D_j^l \sum_{m\in J} \varpi/d_m * y_{mk}\lambda_m \geq \\ & \varpi\mu y_{jk}\omega_{jk} - \varpi y_{jk}\omega_{jk} * \sum_{m\in J} \varpi/d_m * y_{mk}\lambda_m + \varpi y_{jk}\lambda_j/d_j \\ & + \varpi y_{jk}\mu \sum_{i\in J} z_i l_{ki}\rho h_{ki}/R_1 \\ & - \varpi y_{jk} \sum_{m\in J} \varpi/d_m * y_{mk}\lambda_m \sum_{i\in J} z_i l_{ki}\rho h_{ki}/R_1 \end{aligned} \tag{16}$$

In Eq. (16) we still have equations with high degree as follows: $D_j^l y_{mk}$, $y_{jk}y_{mk}$, $y_{jk}z_i l_{ki}$, $y_{jk}y_{mk}z_i l_{ki}$. First we introduce a new parameter $\sigma_{mjk}$ and let $\sigma_{mjk} = D_j^l y_{mk}$. Since $y_{mk}$ is a binary variable, and we can replace quadratic equation by adding the following new constraints, based on McCormick envelopes [36].

$$\frac{1}{N} y_{mk} \leq \sigma_{mjk} \leq N y_{mk} \tag{17}$$

$$\sigma_{mjk} \leq D_j^l - \frac{1}{N}(1 - y_{mk}) \tag{18}$$

$$\sigma_{mjk} \geq D_j^l - N(1 - y_{mk}) \tag{19}$$

Similarly, we introduce several new variables, $\varpi_{ki} = z_i l_{ki}$, $\delta_{mjk} = y_{mk}y_{jk}$, $\phi_{ijk} = y_{jk}\varpi_{ki}$, $\varepsilon_{mijk} = \delta_{mjk}\varpi_{ki}$, with the following new constraints:

$$0 \leq \varpi_{ki} \leq z_i \tag{20}$$

$$z_i + l_{ki} - 1 \leq \varpi_{ki} \leq l_{ki} \tag{21}$$

$$0 \leq \delta_{mjk} \leq y_{mk} \tag{22}$$

$$y_{mk} + y_{jk} - 1 \leq \delta_{mjk} \leq y_{jk} \tag{23}$$

$$0 \leq \phi_{ijk} \leq \varpi_{ki} \tag{24}$$

$$\varpi_{ki} + y_{jk} - 1 \leq \phi_{ijk} \leq y_{jk} \tag{25}$$

$$0 \leq \varepsilon_{mijk} \leq \varpi_{ki} \tag{26}$$

$$\varpi_{ki} + \delta_{mjk} - 1 \leq \varepsilon_{mijk} \leq \delta_{mjk} \tag{27}$$

Then Eq. (16) is simplified as a linear function in Eq. (28) as follows:

$$\begin{aligned} &\mu D_j^l - \varpi \sum_{m \in J} \lambda_m \sigma_{mjk}/d_m - \varpi \mu \omega_{jk} y_{jk} + \\ &\varpi^2 \sum_{m \in J} \lambda_m \omega_{jk} \delta_{mjk}/d_m - \varpi \lambda_j y_{jk}/d_j - \\ &\varpi \mu \sum_{i \in J} \rho h_{ki} \phi_{ijk}/R_1 + \varpi^2 \sum_{m \in J} \lambda_m/d_m \sum_{i \in J} \rho h_{ki} \varepsilon_{mijk}/R_1 \geq 0 \end{aligned} \tag{28}$$

Similarly, we introduce a new parameter $\zeta_{jik} = \gamma_{ji} y_{jk}$, Eq. (5) is linearized as

$$\begin{aligned} &C5: D_j^c - \sum_{i \in J} \gamma_{ji} d_j h_{ji}/R_1 + \sum_{i \in J} \sum_{k \in J} \varpi \zeta_{jik} h_{ji}/R_1 - d_j/R_2 \\ &+ \sum_{k \in J} y_{jk} \varpi/R_2 \geq 0, \forall j \in J \end{aligned} \tag{29}$$

with the following new constraints.

$$0 \leq \zeta_{jik} \leq \gamma_{ji} \tag{30}$$

$$\gamma_{ji} + y_{jk} - 1 \leq \zeta_{jik} \leq y_{jk} \tag{31}$$

Finally, original MINLP-OFCP-DS is reformed as a MILP problem as follows:

**MILP:OFCP-DS:**

$$\begin{aligned} &\text{minimize} && D \\ &\text{subject to} && C1: D \geq D_j^l, D \geq D_j^c, (28), (29), j \in J \\ & && C2: (3), (8), (9), (10), (11), (13) \\ & && C3: z_k \in \{0,1\}, l_{jk} \in \{0,1\}, y_{mk} \in \{0,1\} \\ & && C4: (18) - (28), (31), (32) \end{aligned} \tag{32}$$

### B. A Heuristic Algorithm

However, too many integer variables always make solving the problem time-consuming. We further introduce a two-stage heuristic algorithm as shown in Algorithm 1. The basic idea is that in stage 1, relax all integer variables and then round up $z_j$ and $l_{jk}$. In the stage 2, substitute $z_j$ and $l_{jk}$ into the original problem to reduce two integer variables, and then solve integer programming to achieve the final results.

**Algorithm 1** A Two-Stage Heuristic Algorithm

1: Stage 1: Relax integer variables in MILP-OFCP-DS, and solving linear programming to achieve $z_j^*$ and $l_{jk}^*$.
2: Sort $z_j^*$ in descending order into set $Z$
3: **for all** $z_j \in Z$ **do**
4:     **if** $\sum_{j \in J} z_j \leq Z_0$ **then**
5:         $z_j \leftarrow 1$
6:     **else**
7:         $z_j \leftarrow 0$
8:     **end if**
9: **end for**
10: **for all** $j \in J$ **do**
11:     Sort $l_{jk}^*$, $\forall k \in J$ in descending order into set $L_j$
12:     $flag \leftarrow 0$
13:     **for all** $l_{jk} \in L_j$ **do**
14:         **if** $z_k == 1$ and $flag == 0$ **then**
15:             $l_{jk} \leftarrow 1$
16:             $flag \leftarrow 1$
17:         **end if**
18:         $l_{jk} \leftarrow 0$
19:     **end for**
20: **end for**
21: Stage 2: substitute $\{z_j\}$ and $\{l_{jk}\}$ into original MILP-OFCP-DS, and then solve it based on integer programing to get $\{y_{jk}\}$.
22: **return** $\{z_k\}$, $\{l_{jk}\}$, $\{y_{jk}\}$

In Sec. V-A, we relax the original MINLP to a MILP, by adding new parameters. The two problems are totally the same by adding new parameters, so that there is no relaxation gap exist. In Sec V-B, we further propose a heuristic algorithm to solve MILP with quick response based on the following considerations to decrease the integrality gap.

- More MBSs connecting to satellite in the system, smaller transmission delay the system can achieve. Since the data transmission delay among MBSs can be decreased when the number of MBSs connecting to satellite increases. Therefore, in the Two-Stage Heuristic Algorithm, we fill up the number of MBSs connecting to satellite as many as possible in line 2-9, to approach the minimal delay, i.e., decreasing the integrality gap.
- Solution can be much closer to the ideal one when rounding rounding bigger value of $z_j^*$ and $l_{jk}^*$ to 1. Based on the definitions in Eq. (4) and (5), the gap to the ideal solution can be bigger when rounding smaller value of $z_j^*$ and $l_{jk}^*$ to 1.

Meanwhile, in the algorithm $\sum_{j \in J} z_j \leq Z_0$ is to guarantee that the total number of MBSs connecting to satellite not exceeds the maximal number, and $z_k == 1$ is to guarantee that $k$ should be a MBS connecting to satellite once setting $l_{jk}$ as 1. They ensure the feasibility of the results from the heuristic algorithm. Finally, we substitute $\{z_j\}$ and $\{l_{jk}\}$ into original MILP-OFCP-DS, and then solve it based on integer programing to get $\{y_{jk}\}$. The heuristic algorithm can achieve near-optimization with quick response.

## VI. Evaluation

We evaluate the proposed heuristic approach by comparing it with other solutions, through varying computation rate and communication speed in two different sizes of network.

### A. Evaluation Methodology

The first baseline case considered in the evaluation is when all $y_{mk}$ values are set to 0, i.e., a typical cloud computing system where all of the data is processed at the cloud. The second case is using the proposed heuristic solution where each node processes suitable data that it receives immediately. Meanwhile, we further split the second case into different specific scenarios based on the number of nodes located in the priority area. The scenario includes that (1) all the nodes, (2) half the nodes, and (3) no nodes located in the priority area.

The modeling and calculations are coded in ruby with Gurobi to solve LP problem. We evaluate the proposal on small and medium-sized networks. A small network includes 20 nodes, and a medium network consists of 50 nodes. The medium network is approximately equivalent to half the island of Okinawa, which according to the 2016 report by the Japanese Ministry of Internal Affairs and Communications stated to have 105 LTE base stations [37].

Meanwhile, through the survey on wireless communication technologies, WiMax can achieve 17 to 400 Mbps, and WiFi also can generally achieve 100 to 600 Mbps [38]. Therefore, in the evaluation, $R_1$ is varied from 50 to 500 Mbps to evaluate the effectiveness of the proposed solution in different communication speeds, as shown in Fig. 4 and Fig. 6. For data communication rate in satellite internet $R_2$, HughesNet's Jupiter achieved more than 15Mbps in 2012, and the communication speed can be further enhanced to 500Mbps with $K_a$ band in the future [39]. Therefore in the evaluation, we set the communication rate $R_2$ as 100Mbps.

### B. Results when Varying $\mu$ and $R_1$ in a Small Size Network

First we discuss the evaluation results when varying $\mu$ and $R_1$ in a small size network.

Fig. 3 and Fig. 4 represent the results of minimal delay with changing computation rate and communication speed, respectively. The evaluation is performed by considering the following four cases: (1) Case 1: all processing done at cloud, (2) Case 2: the proposed solution where all nodes are located in the priority area, (3) Case 3: the proposed solution where almost half of the nodes are located in the priority area, and (4) Case 4: the proposed solution where no node is located in the priority area.

First, from the results in Fig. 3 we can see that the proposed solution worked efficiently to reduce minimal delay in transmitting data from edge site to cloud, by using the local computation resource effectively. More specifically, (1) when the computation rate was set in the range from 4 to 6 Gbps, which can be considered as a normal case, the proposed solution achieved a 20% decrease in overall delay by offloading computation from cloud to local edge nodes in wirelessly connected disaster areas. (2) Given a high performance computation mechanism e.g. 12 Gbps, the proposed solution achieved better results by decreasing around 85% overall delay in Case 4, and around 78% overall delay in Case 3, where we set data resolution as 0.9 in the experiment. Even in the worst case that all the MBSs were located in the priority area in Case 2, the proposed solution achieved almost a 50% decreasing of overall delay through the experiment.

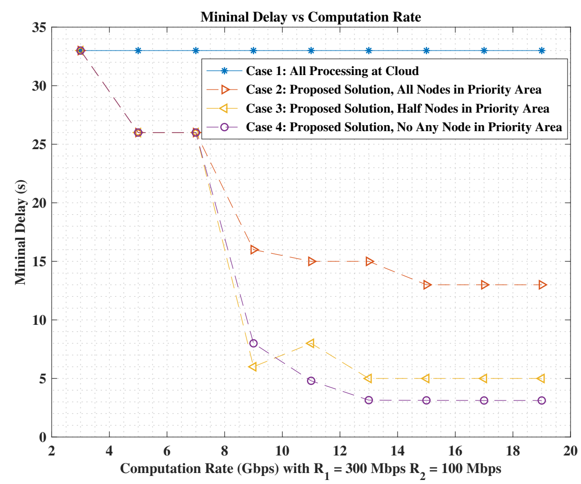



Fig. 3. Overall Delay Varying with Computation Rate $\mu$ in a Small Size Network

Meanwhile, we evaluate the proposed solution with different communication speeds $R_1$ as shown in Fig. 4. When the disaster network is constructed by low speed wireless mediums, e.g. WiFi, all the cases suffer a bigger delay. However the proposed solution in Case 2, 3, and 4 can achieve an almost 40% deceased in overall delay when communication speed was set as 50 Mbps, and 50% decreased when communication speed was set as 100 Mbps. In a better communication environment, the overall delay in all the four cases decreased, while the proposed solution in Case 2,3 and 4 achieved a much better performance by reducing around 50% overall delay.

### C. Results when Varying $\mu$ and $R_1$ in a Medium Size Network

The corresponding evaluation results when we varied $\mu$ and $R_1$ in a medium size network are shown in Fig 5 and 6, respectively. Also the number of MBSs which connects with satellite was increased correspondingly to support a bigger disaster area.

From Fig. 5 we can see that the proposed heuristic solution achieved effective decreasing in overall delay as the computation rate $\mu$ increased. For a normal case where $\mu$ was in the

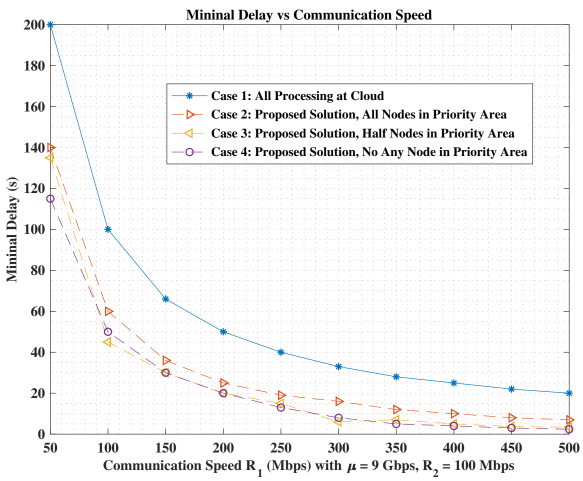



Fig. 4. Overall Delay Varying with Communication Speed $R_1$ in a Small Size Network

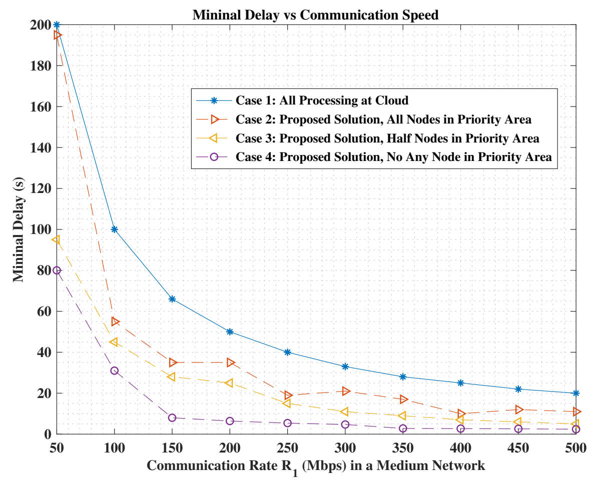



Fig. 6. Overall Delay Varying with Communication Speed $R_1$ in a Medium Size Network

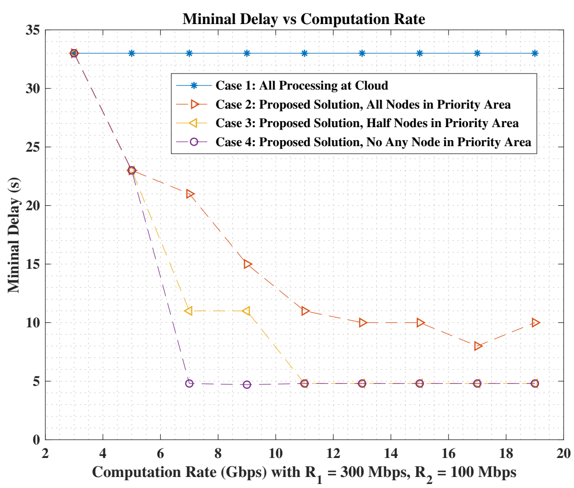



Fig. 5. Overall Delay Varying with Computation Rate $\mu$ in a Medium Size Network

range from 4 to 6 Gbps, the overall delays in Case 2, 3 and 4 were decreased around 30% to 60%. The proposed solution got much more effective with a much more power computation unit, e.g. around 60% to 80% decreases in overall delay when $\mu$ was larger than 8.

Fig. 6 presents the results when we varied the communication rate $R_1$, where we can see that the proposed solution decreased overall delay clearly with different communication rates, in all the Case 2, 3 and 4.

### *D. Data Resolution*

We further investigate the minimal overall delay by varying required data resolution for each priority area as shown in Fig. 7. The evaluation was performed in the small network and we considered two cases, i.e., Case 1: half nodes were located in the priority area, and Case 2: All the nodes were located in the priority area. From the evaluation results we can see that with a higher required data resolution, the system suffered a bigger delay, while system managers could accordingly get more details in data for data analysis. A balance point could be found to guarantee acceptable data resolution while guaranteeing minimal overall delay.

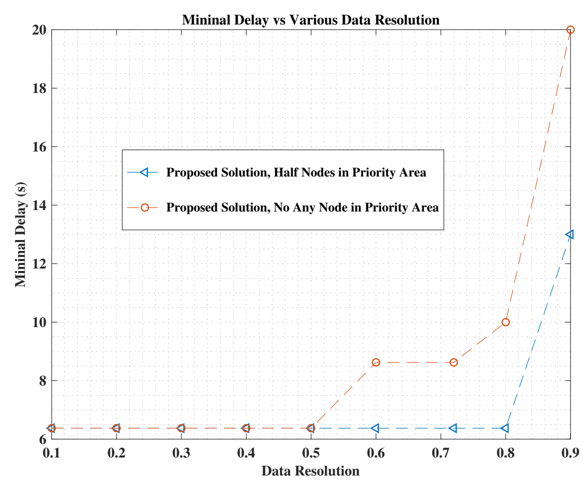



Fig. 7. Overall Delay Varying with Data Resolution

Finally, we evaluated the performance of the proposed heuristic approach as shown in Fig. 8. The evaluation is performed by calculating consuming time of the program. From Fig. 8 we can see that the proposed heuristic approach clearly reduced the consuming time needed to solve the problem. Quicker solver is definitely better in disaster scenarios.

## VII. Conclusion

Big data analysis is important in disaster scenarios, to better understand the situations and support decision-making. Traditional cloud-based big data analysis suffers big latency for data transmission, while the problem is possible to be

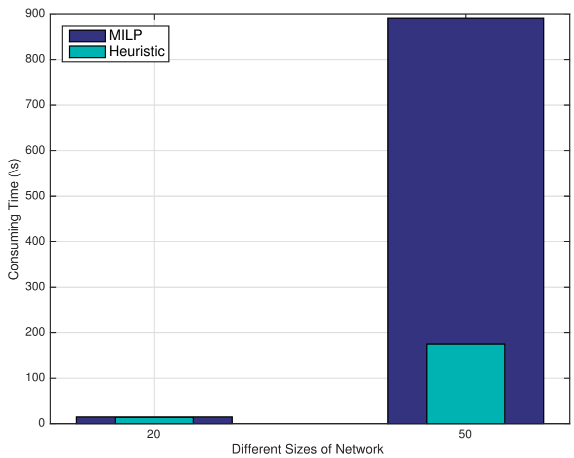



Fig. 8. System Performace

solved in a Fog/Edge supported computation platform. In this paper, we focus on disaster scenarios while network infrastructure is reconstructed by MBSs. MBS can collect data from smartphones, and decide to process data in local or remote cloud. We have formalized the problem as a Well-Known MINLP and propose a two-stage heuristic algorithm to solve it in a real time manner. Through comprehensive evaluation, we have presented the effectiveness of the proposed heuristic algorithm in reducing overall delay by varying system parameters and sizes of network.

## Acknowledgment

This research was supported by the Japan Science and Technology Agency (JST) Strategic International Collaborative Research Program (SICORP).

## References

[1] M.-P. Kwan and D. M. Ransberger, "Lidar assisted emergency response: Detection of transport network obstructions caused by major disasters," *Computers, Environment and Urban Systems*, vol. 34, no. 3, pp. 179–188, 2010.

[2] J. Wang, Z. Cheng, P. Li, J. Chen, and Y. Zhou, "Design of a best load balancing method for anti-disaster mobile mesh communication networks," in *Proceedings of the 2013 IEEE MS 2013*, Washington, DC, USA, pp. 63–69.

[3] T. Sakano, S. Kotabe, T. Komukai, T. Kumagai, Y. Shimizu, A. Takahara, T. Ngo, Z. M. Fadlullah, H. Nishiyama, and N. Kato, "Bringing movable and deployable networks to disaster areas: development and field test of mdru," *IEEE Network*, vol. 30, no. 1, pp. 86–91, January 2016.

[4] K. Suto, H. Nishiyama, and N. Kato, "Postdisaster user location maneuvering method for improving the qoe guaranteed service time in energy harvesting small cell networks," *IEEE Trans. Vehicular Technology*, vol. 66, no. 10, pp. 9410–9420, 2017. [Online]. Available: https://doi.org/10.1109/TVT.2017.2702750

[5] T. Ngo, H. Nishiyama, N. Kato, T. Sakano, and A. Takahara, "An efficient safety confirmation method using image database in multiple-mdru-based disaster recovery network," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2556–2565, 2017.

[6] W. Junbo, S. Guo, Z. Cheng, P. Li, and J. Wu, "Optimization of deployable-base-stations with guaranteed qoe in disaster scenarios," *IEEE Transactions on Vehicular Technology*, 2016.

[7] H. Nishiyama, K. Suto, and H. Kuribayashi, "Cyber physical systems for intelligent disaster response networks: Conceptual proposal and field experiment," *IEEE Network*, vol. 31, no. 4, pp. 120–128, 2017.

[8] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, New York, NY, USA, 2012, pp. 13–16.

[9] M. Mukherjee, L. Shu, and D. Wang, "Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges," *IEEE Communications Surveys & Tutorials*, no. c, pp. 1–1, 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8314121/

[10] H. Zhang, Y. Xiao, S. Bu, D. Niyato, R. Yu, and Z. Han, "Fog Computing in Multi-Tier Data Center Networks : A Hierarchical Game Approach," pp. 1–6, 2016.

[11] S.-H. Park, O. Simeone, and S. S. Shitz, "Joint optimization of cloud and edge processing for fog radio access networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7621–7632, 2016.

[12] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Efficient algorithms for capacitated cloudlet placements," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 10, pp. 2866–2880, 2016.

[13] O. Simeone, A. Maeder, M. Peng, O. Sahin, and W. Yu, "Cloud radio access network: Virtualizing wireless access for dense heterogeneous systems," *Journal of Communications and Networks*, vol. 18, no. 2, pp. 135–149, 2016.

[14] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Capacitated cloudlet placements in wireless metropolitan area networks," in *LCN*. IEEE Computer Society, 2015, pp. 570–578.

[15] L. Gu, D. Zeng, P. Li, and S. Guo, "Cost minimization for big data processing in geo-distributed data centers," *IEEE Trans. Emerging Topics Comput.*, vol. 2, no. 3, pp. 314–323, 2014. [Online]. Available: https://doi.org/10.1109/TETC.2014.2310456

[16] J. Wang, Y. Wu, N. Yen, S. Guo, and Z. Cheng, "Big data analytics for emergency communication networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1758–1778, 2016.

[17] L. Zhong, K. Takano, K. Yoda, Y. Ji, and S. Yamada, "Spatio-temporal estimation of mobile-phone call demand in the kumamoto earthquakes," in *Information and Communication Technologies for Disaster Management (ICT-DM), 2017 4th International Conference on*. IEEE, 2017, pp. 1–7.

[18] C. Yang, G. Su, and J. Chen, "Using big data to enhance crisis response and disaster resilience for a smart city," in *Big Data Analysis (ICBDA), 2017 IEEE 2nd International Conference on*. IEEE, 2017, pp. 504–507.

[19] Y. Qiao, Z. Xing, Z. M. Fadlullah, J. Yang, and N. Kato, "Characterizing flow, application, and user behavior in mobile networks: A framework for mobile big data," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 40–49, 2018. [Online]. Available: https://doi.org/10.1109/MWC.2018.1700186

[20] U. H. Zaki, R. Ibrahim, S. A. Halim, K. A. M. Khaidzir, and T. Yokoi, "Sentiflood: Process model for flood disaster sentiment analysis," in *Big Data and Analytics (ICBDA), 2017 IEEE Conference on*. IEEE, 2017, pp. 37–42.

[21] K. Anazawa, T. Miyazaki, P. Li, and X. Wang, "Big data synchronization among isolated data servers in disaster," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[22] Y. Wu, K. Kant, S. Zhang, A. Pal, and J. Wang, "Disaster network evolution using dynamic clustering of twitter data," in *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, June 2017, pp. 348–353.

[23] M. M. Rathore, A. Paul, A. Ahmad, M. Imran, and M. Guizani, "Big data analytics of geosocial media for planning and real-time decisions," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[24] D. Krishnaswamy, R. Krishnan, A. Qamar, and K. Rajagopal, "Sankat-edge: A distributed network edge infrastructure framework for disaster recovery," in *Global Humanitarian Technology Conference-South Asia Satellite (GHTC-SAS), 2014 IEEE*. IEEE, 2014, pp. 216–221.

[25] A. Rauniyar, P. Engelstad, B. Feng *et al.*, "Crowdsourcing-based disaster management using fog computing in internet of things paradigm," in *Collaboration and Internet Computing (CIC), 2016 IEEE 2nd International Conference on*. IEEE, 2016, pp. 490–494.

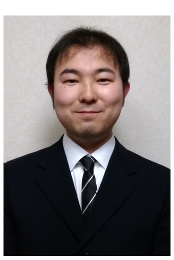
[26] E. M. Trono, M. Fujimoto, H. Suwa, Y. Arakawa, and K. Yasumoto, "Generating pedestrian maps of disaster areas through ad-hoc deployment of computing resources across a dtn," *Computer Communications*, vol. 100, pp. 129–142, 2017.

[27] Y. Wang, M. C. Meyer, J. Wang, and X. Jia, "Delay minimization for spatial data processing in wireless networked disaster areas," in *2017 IEEE Global Communications Conference (GLOBECOM)*, To be published in Dec 2017, pp. 1–6.

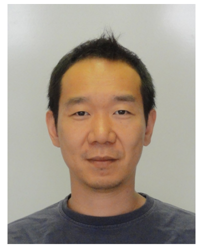
[28] A.-l. Jin, W. Song, S. Member, P. Wang, and S. Member, "Sharing for Cloudlets in Mobile Cloud Computing," vol. 9, no. 6, pp. 895–909, 2016.

[29] M. Jia, W. Liang, Z. Xu, and M. Huang, "Cloudlet load balancing in wireless metropolitan area networks," *Proceedings - IEEE INFOCOM*, vol. 2016-July, 2016.
[30] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 108–119, Jan 2017.
[31] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Trans. Computers*, vol. 65, no. 12, pp. 3702–3712, 2016. [Online]. Available: https://doi.org/10.1109/TC.2016.2536019
[32] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
[33] B. Welton and B. P. Miller, "Data Reduction and Partitioning in an Extreme Scale GPU-Based Clustering Algorithm," 2017.
[34] B. Welton, E. Samanas, and B. P. Miller, "Mr. scan: Extreme scale density-based clustering using a tree-based network of gpgpu nodes," in *2013 SC - International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, Nov 2013, pp. 1–11.
[35] M. Bendechache and M. T. Kechadi, "Distributed clustering algorithm for spatial data mining," in *2015 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM)*, July 2015, pp. 60–65.
[36] "Mccormick envelopes." [Online]. Available: https://optimization.mccormick.northwestern.edu
[37] M. of Internal Affairs and Communications, "Information Communications Statistics Database H28.1Q Report," http://www.soumu.go.jp/johotsusintokei/field/denpa01.html, 2016, [Online: accessed 29-March-2017].
[38] "https://en.wikipedia.org/wiki/comparison_of_wireless_data_standards."
[39] "https://en.wikipedia.org/wiki/satellite_internet_access."

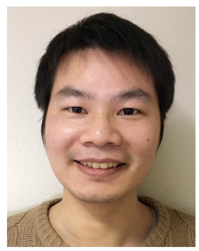
**Junbo Wang** received his Bachelor and Master degrees from Yanshan University, China, in electrical engineering. He received his PhD degree from the University of Aizu, Japan in computer science and engineering in 2011. Currently, he is an Associate Professor in the University of Aizu. He is a member of IEEE and IEICE. His research interests include emergency communication networks, big data analytics, and the Internet of Things.



**Koichi Sato** received the bachelor's and master's degree in computer science and engineering from the University of Aizu, Aizuwakamatsu, Japan. He is currently pursuing the Ph.D. degree in computer science and engineering at the University of Aizu. His research interests include big data analysis and Twitter-based event detection.



**Song Guo** (M'02-SM'11) is a Full Professor at Department of Computing, The Hong Kong Polytechnic University. His research interests are mainly in the areas of big data, cloud computing and networking, and distributed systems. His work was recognized by the 2016 Annual Best of Computing in ACM Computing Reviews. He is the recipient of the 2017 IEEE Systems Journal Annual Best Paper Award and other five Best Paper Awards from IEEE/ACM conferences. Prof. Guo was an Associate Editor of IEEE TPDS and an IEEE ComSoc Distinguished Lecturer. He is now on the editorial board of IEEE TCC, IEEE TETC, IEEE TSUSC, IEEE TGCN, IEEE Network, etc. Prof. Guo also served as General and TPC Chair for numerous IEEE conferences. He currently serves as a Director and Member of the Board of Governors of ComSoc.



**Wuhui Chen** received his bachelor?s degree in 2008 from Software College Northeast University, China. He received his master?s and doctoral degrees from School of Computer Science and Engineering, University of Aizu, Japan in 2011 and 2014, respectively. Now he is an associate professor at Sun Yat-sen university, China. He was a researcher at the Revitalization Center, University of Aizu, Japan. His research interests include services computing, edge computing, and cloud robotics.

**Jie Wu** is the chair and a Laura H. Carnell Professor in the Department of Computer and Information Sciences at Temple University. Prior to joining Temple University, he was a program director at the National Science Foundation and Distinguished Professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Computers, IEEE Transactions on Service Computing, and Journal of Parallel and Distributed Computing. Dr. Wu was general co-chair/chair for IEEE MASS 2006, IEEE IPDPS 2008 and IEEE ICDCS 2013, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013, and general chair for ACM MobiHoc 2014. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.