Cost-efficient Deployment of Fog Computing Systems at Logistics Centers in Industry 4.0

Chun-Cheng Lin*, *Senior Member, IEEE*, and Jhih-Wun Yang

***Abstract*—In Industry 4.0, factories become increasingly smart and efficient through intelligent cyber-physical systems based on deployment of Internet of Things (IoT), mobile devices, and cloud computing systems. In practice, the cloud computing system in a factory is managed in a centralized way, and hence may not afford heavy computing loads from thousands of IoT devices in the factory. An approach to addressing this issue is to deploy fog/edge computing resources nearby IoT devices in a distributed way to provide real-time computing responses on sites. This work investigates deployment of an intelligent computing system consisting of a cloud center, gateways, fog devices, edge devices, and sensors attached to facilities in a logistics center. Except for locations of the cloud center and sensors that have been determined based on the factory layout, this work establishes an integer programming model for deploying gateways, fog devices, edge devices in their respective potential sites, so that the total installation cost is minimized, under constraints of maximal demand capacity, maximal latency time, coverage, and maximal capacity of devices. This work further solves this NP-hard facility location problem by a metaheuristic algorithm that incorporates discrete monkey algorithm to search for good-quality solutions and genetic algorithm to increase computational efficiency. Simulation verifies high performance of the proposed algorithm in deployment of intelligent computing systems in moderate-scale instances of intelligent logistics centers.**

***Index Terms* — Fog computing, monkey algorithm, genetic algorithm, deployment, Industry 4.0, logistics center**

I. Introduction

IN the industrial Internet of Things (IoT) [1], IoT devices are deployed along production/assembly lines to sense surrounding environments to further assist in making real-time response decisions on production processes or facilities, and executing the requested computing tasks according to analysis of the sensed data. Furthermore, integrated with advanced AI techniques, everything is acted as a lively object with autonomous intelligence and is connected with each other to execute more intelligent tasks. In Industry 4.0, smart factories are achieved by integrating technologies of the IoT, mobile devices, and cloud computing systems [2].

This work considers deployment of an intelligent computing system in a logistics center. With the arrival of the era of Industry 4.0, intelligent technologies drive the possibility of no manpower in the whole logistics process from storing to shipping products, so that the whole logistics procedure becomes increasingly efficient and smart. Advances in the IoT, robots, and drones also mitigate the constraints of manpower handling and conveyor.

In an intelligent logistics center (e.g., see Fig. 1, drawn according to a real-world unmanned logistics center [3]), suppliers' trucks (at the rightmost area in Fig. 1) deliver cargos to the receiving area. Then, cargos are uploaded and moved intelligently by mobile robots (blue disks in Fig. 1) (e.g., Kiva systems of Amazon) to the warehousing space, through RFID sensing doors (orange long bars in Fig. 1) to confirm the quantity of cargos from each supplier.

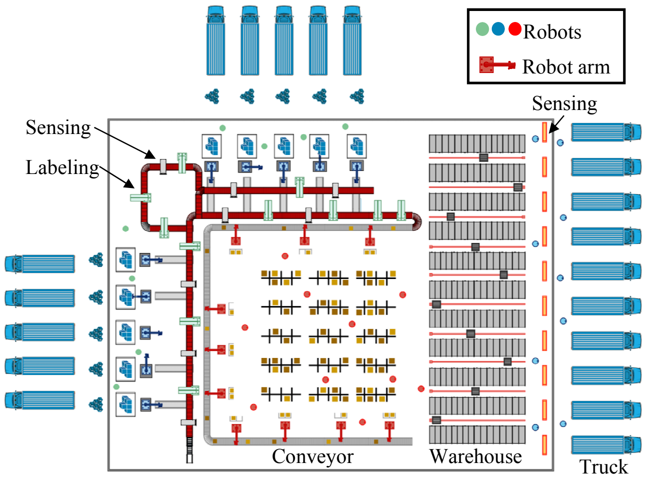

Fig. 1. Illustration of an intelligent logistics center.

Robots classify cargos and move them to a large-scale warehouse space (grey stacks in the right area of Fig. 1), and concurrently update data of inventories of the space in the system. When the system receives customer orders, robots (red disks) extract the required products from the warehouse space, and move them to the picking area (the shelf surrounded by the conveyor). Each robot intelligently plans the best routing path and dodges collision with other robots.

When receiving a task of picking a product, a robot will deliver the required product to a workstation with a robot arm (a red square attached to the conveyor in Fig. 1). Then, the robot arm picks the product up and puts it to the conveyor, along which the product moves to the packing area. When the product passes through some machines (green rectangles in Fig. 1) on the conveyor, it is posted by a tag and then moves to the sorting area. On the conveyor, the RFID sensing doors (grey rectangles in Fig. 1) are in charge of confirming the amount of products to be delivered. After sorting according to the

Manuscript received November 30, 2017; revised February 4, 2018; accepted April 13, 2018. This work was supported in part by Grant MOST 106-2221-E-009-101-MY3.

C.-C. Lin and J.-W. Yang are with Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu 300, Taiwan. E-mail: cclin321@nctu.edu.tw, bbblack50522@gmail.com.
* C.-C. Lin is the corresponding author of this paper.

customer orders, products are moved to the shipping area. Then, robots (green disks in Fig. 1) move them to trucks (shown in the top and left areas in Fig. 1), and then these trucks deliver products to customers.

Generally, clouds in an intelligent logistics center are located and managed in a centralized way. However, if the factory area is too large, it leads to a too long latency time for responding to the requests from enormous IoT sensors to the centralized cloud. Distributed fog computing has been an approach to addressing the heavy load and high latency owing to real-time response to thousands of IoT devices in centralized cloud computing systems [4]. A part of the computing tasks for logistics operations that can be finished according to only local information are taken by nearby fog/edge computing resources, rather than the centralized cloud. By doing so, these tasks can be finished more immediately and flexibly. Additionally, fog/edge computing is easily incorporated with the IoT environment, to increase efficiency and decrease power consumption over 40% more than conventional cloud computing systems [5].

A lot of previous works on deployment of networking systems in various industrial applications have existed, e.g., deployment of cloudlets in wireless metropolitan area networks (WMAN) [6], [7], deployment of wireless sensor networks in oil and gas refinery plants [8] and harsh industrial environments [9], and deployment of roadside units (RSUs) in vehicular networks [10]. To the best of our understanding, no previous works investigated the problem of deploying fog computing systems in logistics centers.

This work investigates deployment of an intelligent computing system in a logistics center, which consists of a cloud center, gateways, fog devices, edge devices, and sensing devices, which are attached to box shuttles, automated guided vehicles (AGVs), robotic arms, machines, and so on. This system supposes that locations of the cloud center and sensing devices are known. Although some sensing devices are mobile, their movement region is generally fixed and small. Therefore, the decision of the problem concerned in this work is to determine whether to deploy the other three types of devices (i.e., gateways, fog devices, and edge devices) in their respective pre-known potential sites of the logistics center, so that the total installation cost is minimized, under constraints of maximal demand capacity, maximal latency time, coverage, and maximal capacity of devices.

The deployment in logistics centers differs from other deployment applications as follows: 1) Deployment in vehicular networks or smart cities is of large scope and open space, whereas deployment in logistics centers is of relatively small scope and closed space. 2) Deployment in different production factories is affected by the location layout of machines and facilities. Recent works (e.g., [9], [11]) have focused on deployment of factories with harsh environments (e.g., refineries and nuclear power plants), in which positions of sensors follow a probability distribution (i.e., it would not be easy to deploy precise positions of each sensor), and communications of all devices may be restricted to the harsh environment. Differently, logistics centers do not have these issues because only placements, package, and delivery of products are concerned.

This work establishes an integer programming (IP) model for this problem. However, IP is NP-hard generally [12], and hence this work further proposes a metaheuristic algorithm for this problem. Monkey algorithm (MA) [13] performs better than some conventional metaheuristics (e.g., genetic algorithm (GA), particle swarm optimization (PSO) [14], and ant colony optimization (ACO) [15]) in addressing optimization problems with high dimensions and nonlinear nondifferential functions, and its parameters are fewer [16]. Therefore, our metaheuristic approach is based on MA. To address the concerned discrete deployment problem, this approach is incorporated with the discrete MA (DMA) [14] to search for good quality solutions and GA [17] to increase computational efficiency.

The main contributions of this work are as follows:

- This work is the first to investigate deployment of a fog computing system in a logistics center, and to establish a IP model for this problem. Different from previous works (e.g., [6], [18], [19]), the proposed problem model additionally considers 1) demand capacities of gateways, fog devices, and edge devices; 2) maximal latency times of edge devices, fog devices, and gateways; and 3) costs of installing fog and edge devices.
- This work proposes a hybrid DMA and GA (DMGA) for the problem. The original DMA in [14] was designed for addressing discrete decision variables. The concerned problem includes only binary decision variables, but the original DMA cannot be directly degenerated to solve problems with binary decision variables. Therefore, one of our contributions is to devise the DMA operators to specifically address binary decision variables.

II. Preliminaries

A. System framework

Consider the framework of a computing system deployed in a logistics center (Fig. 2), consisting of a cloud computing center, gateways, fog devices, edge devices, and sensing devices in a top-down fashion, which are explained as follows.

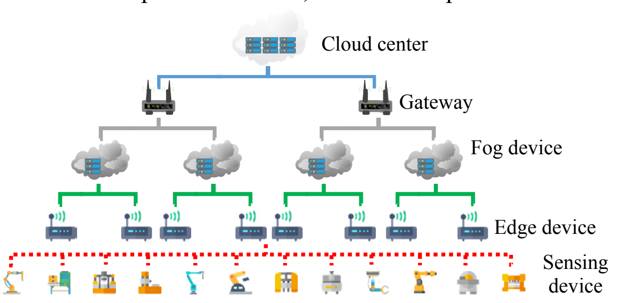

Fig. 2. The architecture of fog computing.

The cloud center provides a computing model for accessing the Internet anytime anywhere, and utilizing the sharing computing facilities, storage devices, and applications according to requests and convenience of use. It can cope with problems in the logistics center through infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Gateway connects incompatible network facilities or APs. Fog devices at the lower level transmit data to the cloud center through gateways.

Fog device collects data from extreme edges at the lower level, and generally must analyze and respond to the data within one second. Edge devices are deployed closely to physical machines, conveyor lines, and warehouses in the logistics center, so that the logistics decisions requiring only local information can be determined in a short distance without cloud computing. Note that edge computing is deployed on terminal facilities or sensors, whereas fog computing is deployed on the data center nodes within a certain region. Edge computing does not need to support IaaS, PaaS, and SaaS but focuses on responding to requests of terminal facilities; whereas, differently, fog computing extends cloud services to network edges.

Sensing devices in the logistics center aim to sense/monitor box shuttles, AGVs, and robotic arms. Box shuttles implement 3D warehouses through commuting at a high speed on narrow shelf tracks, precisely storing and picking cargo boxes, and delivering cargos to the conveyor. AGVs achieve production planning for the whole logistics center by adopting techniques of computer vision, image processing, and automatic navigation, incorporating systematic dispatching tasks and computing inventory to replenish the warehouse.

In the system deployed in a logistics center, edge devices are not only used for data transmission, but also provide AGVs real-time responses when encountering other AGVs on the movement trajectory. The data required to make the decision is identified and collected by edge devices, and is transmitted to fog devices that temporarily store and process the data, and send back the decision back to AGVs. At the same time, the data is also transmitted to the cloud center for historical analysis and long-run storage.

B. Related works

Fog computing has a lot of various applications [20], [21]. This section focuses on reviewing recent works on deployment of networking systems in applications of smart factories, smart cities, and vehicular networks.

In smart factories, the work in [22] transformed the original centralized communications into P2P communications through a fog computing system, to make production processes be operated more efficiently. The work in [9] effectively deployed wireless sensors to collect data in harsh industrial environments. The work in [8] surveyed the challenges to deploy wireless sensor networks in oil and gas refinery plants. The work in [11] proposed a hybrid harmony search and genetic algorithm for deploying group-based industrial wireless sensor networks along a production/assembly line.

The work in [23] investigated the application of deploying fog computing systems in smart cities to address real-time surveillance video stream processing in emergency situations. The works in [6] and [7] investigated deployment of cloudlets in WMAN, in which the former proposed an integer linear programming model, an approximation algorithm, and an online algorithm with theoretical bounds of the results; and the latter deployed cloudlets to optimize movements of users and to minimize the latency of cloud services.

The work in [24] proposed a vehicular fog computing system in which each vehicle is regarded as a facility with communications and computing resources. This system serves as an infrastructure for communications and computation for edge devices of multiple end-users, to efficiently utilize communications and computing resources of each vehicle. The work in [25] applied fog computing in lane-change assistance in intelligent transportation. The work in [10] efficiently deployed RSUs on roads to minimize costs of RSUs while the demand requirements of vehicles are satisfied. The works in [26], [27] deployed RSUs and sensors on two-lane loads to minimize the total distance and the total number of hops.

III. Problem Model

This work deploys an intelligent computing system whose architecture is shown in Fig. 2 in a logistics center. In the logistics center, there is a centralized cloud center whose location has been known and fixed; and the locations of sensing devices attached to warehouse space, conveyors, and robot arms, and AGVs have also been known, because it is supposed that the layout of these devices has been determined for operational efficiency. Although AGVs are mobile, the region of their activities is basically fixed.

In light of the above, the locations of gateways, fog devices, and edge devices are unknown, and the network of this system (i.e., how to connect all devices) is also unknown. In general, based on the floorplan of the logistics center and the locations of the cloud center and AGVs, the potential sites of all the concerned devices have been known. Therefore, the decisions of determining locations of devices in a logistics center are reduced to determining whether the potential site of each device is deployed with this device.

Without loss of generality and for simplicity of the model, suppose that all sensing devices are AGVs. The problem concerned in this work is to determine the locations of gateways, fog devices, and edge devices and to connect all devices according to the hierarchy in Fig. 2 (in which links between cloud centers and gateways, between gateways and fog devices, and between fog devices and edge devices are wired; whereas links between edge devices and AGVs are wireless), so that the total deployment cost is minimized, under constraints of maximal demand capacity, maximal latency time, coverage, and maximal capacity of devices. The notations used in the problem model is given in Table 1. The concerned problem is modelled as the following IP model:

Minimize
$$c_{\mathrm{f}} \sum_{(i,j)\in\{s\}\times\Omega_{\mathrm{G}}\cup\Omega_{\mathrm{G}}\times\Omega_{\mathrm{F}}\cup\Omega_{\mathrm{F}}\times\Omega_{\mathrm{E}}} x_{ij}\cdot d_{ij} + c_{\mathrm{G}}\sum_{m\in\Omega_{\mathrm{G}}} g_m + c_{\mathrm{F}}\sum_{n\in\Omega_{\mathrm{F}}} f_n + c_{\mathrm{E}}\sum_{t\in\Omega_{\mathrm{E}}} q_t$$
Subject to (1)–(23)
$$x_{ij}, g_m, f_n, q_t \in \{0,1\},\quad \forall (i,j)\in\{s\}\times\Omega_{\mathrm{G}}\cup\Omega_{\mathrm{G}}\times\Omega_{\mathrm{F}}\cup\Omega_{\mathrm{F}}\times\Omega_{\mathrm{E}},$$
$$m\in\Omega_{\mathrm{G}}, n\in\Omega_{\mathrm{F}}, t\in\Omega_{\mathrm{E}}.$$

In the above objective, the first term is the cost of the fiber links between the cloud center and gateways, between gateways and fog devices, and between fog devices and edge devices; the other three terms are costs of installing gateways, fog devices, and edge devices, respectively.

TABLE 1.
DEFINITIONS OF NOTATIONS

Parameter	Definition
s	Index of the cloud center.
Ω_G	Set of potential sites for gateways.
Ω_F	Set of potential sites for fog devices.
Ω_E	Set of potential sites for edge devices.
Ω_A	Set of AGVs.
c_f	The price of one unit length of fiber.
c_G	Cost of installing a gateway.
c_E	Cost of installing an edge.
c_F	Cost of installing a fog device.
H_m^G	The maximal demand that gateway m can fulfill.
H_n^F	The maximal demand that fog device n can fulfill.
H_t^E	The maximal demand that edge device t can fulfill.
N_G	The maximum number of fog devices that a gateway can accommodate.
N_F	The maximum number of edge devices that a fog device can accommodate.
N_E	The maximum number of AGVs which are covered by an edge device.
d_{ij}	Distance between nodes i and j.
L_A	The data length for an edge device from an AGV.
L_E	The data length for a fog device from an edge device.
L_F	The data length for a gateway from a fog device.
γ_A	The data rate from an AGV to an edge device.
γ_E	The data rate from an edge device to a fog device.
γ_F	The data rate from a fog device to a gateway.
R_E	Diameter of the coverage range of an edge device.
r_A	Demand of an AGV.
D_{tk}^E	Maximum latency time of linking edge device t to AGV k.
D_{nt}^F	Maximum latency time of linking fog device n to edge device t.
D_{mn}^G	Maximum latency time of linking gateway m to fog device n.
Variable	**Definition**
$\omega(P_i)$	Demand at potential site P_i.
Decision variable	**Definition**
x_{ij}	A binary variable deciding if a link between nodes i and j exists.
g_m	A binary variable deciding if the potential site for gateway m is selected to place a gateway.
f_n	A binary variable deciding if the potential site for fog device n is selected to place a fog device.
q_t	A binary variable deciding if the potential site for edge device t is selected to place an edge device.

Constraints for establishing the network topology of the whole system in a bottom-up fashion are explained as follows. Constraint (1) enforces that for each AGV k, only one link x_{tk} between edge device t and AGV k is connected. Constraints (2) and (3) considers that the binary decision variable q_t of whether to deploy an edge device t is determined by the links with AGVs. Constraint (4) enforces that there must be exact one fog device n linked with edge device t. Similarly, Constraints (5)–(7) determine binary decision variable f_n of whether to deploy a fog device n; and Constraints (8)–(10) determine binary decision variable g_m of whether to deploy a gateway m.

$$\sum\nolimits_{t\in\Omega_E} x_{tk} = 1, \quad \forall k\in\Omega_A \tag{1}$$
$$q_t \le \sum\nolimits_{k\in\Omega_A} x_{tk}, \quad \forall t\in\Omega_E \tag{2}$$
$$x_{tk} \le q_t, \quad \forall t\in\Omega_E, \forall k\in\Omega_A \tag{3}$$
$$\sum\nolimits_{n\in\Omega_F} x_{nt} = q_t, \quad \forall t\in\Omega_E \tag{4}$$
$$f_n \le \sum\nolimits_{t=\Omega_E} x_{nt}, \quad \forall n\in\Omega_F \tag{5}$$
$$x_{nt} \le f_n, \quad \forall n\in\Omega_F, \forall t\in\Omega_E \tag{6}$$
$$\sum\nolimits_{m\in\Omega_G} x_{mn} = f_n, \quad \forall n\in\Omega_F \tag{7}$$
$$g_m \le \sum\nolimits_{n\in\Omega_F} x_{mn}, \quad \forall m\in\Omega_G \tag{8}$$
$$x_{mn} \le g_m, \quad \forall m\in\Omega_G, \forall n\in\Omega_F \tag{9}$$
$$x_{sm} = g_m, \quad \forall m\in\Omega_G \tag{10}$$

Constraints for the demand capacity of all devices are explained as follows. Constraint (11) enforces the demand amount $\omega(q_t)$ served by edge device t to be the sum of demand of all AGVs determined by all links between edge device t and each AGV k; and Constraint (12) enforces it to be no greater than the maximum capacity H_t^E for an edge device t. Similarly, Constraints (13)–(14) consider the maximum capacity H_n^F for a fog device n; and Constraints (15)–(16) consider the maximum capacity H_m^G for a gateway m.

$$\sum\nolimits_{k\in\Omega_A} r_A \cdot x_{tk} = \omega(q_t), \quad \forall t\in\Omega_E \tag{11}$$
$$\omega(q_t) \le q_t \cdot H_t^E, \quad \forall t\in\Omega_E \tag{12}$$
$$\sum\nolimits_{t\in\Omega_E} \omega(g_m)\cdot x_{nt} = \omega(f_n), \quad \forall n\in\Omega_F \tag{13}$$
$$\omega(f_n) \le f_n \cdot H_n^F, \quad \forall n\in\Omega_F \tag{14}$$
$$\sum\nolimits_{n\in\Omega_F} \omega(f_n)\cdot x_{mn} = \omega(g_m), \quad \forall m\in\Omega_G \tag{15}$$
$$\omega(g_m) \le g_m \cdot H_m^G, \quad \forall m\in\Omega_G \tag{16}$$

Constraints (17), (18), and (19) calculate the latency times between edge device t and AGV k, between fog device n and edge device t, and between gateway m and fog device n, and enforces them not to be greater than the maximal latency times D_{tk}^E, D_{nt}^F, and D_{mn}^G, respectively.

$$L_A / (\sum\nolimits_{k\in\Omega_A} x_{tk}\cdot\gamma_A)\cdot q_t \le D_{tk}^E, \quad \forall t\in\Omega_E \tag{17}$$
$$L_E / (\sum\nolimits_{t\in\Omega_E} x_{nt}\cdot\gamma_E)\cdot f_n \le D_{nt}^F, \quad \forall n\in\Omega_F \tag{18}$$
$$L_F / (\sum\nolimits_{n\in\Omega_F} x_{mn}\cdot\gamma_F)\cdot g_m \le D_{mn}^G, \quad \forall m\in\Omega_G \tag{19}$$

For wireless links between edge devices and AGVs, Constraint (20) considers that the distance between edge devices and AGVs must not be longer than the coverage range R_E of an edge device.

$$x_{tk} \cdot d_{tk} \le R_{\mathrm{E}} / 2, \quad \forall t \in \Omega_{\mathrm{E}}, k \in \Omega_{\mathrm{A}} \tag{20}$$

Number of links for each edge device, fog device, and gateway has a maximal capacity in Constraints (21)–(23), respectively.

$$\sum\nolimits_{k \in \Omega_{\mathrm{A}}} x_{tk} \le N_{\mathrm{E}}, \quad \forall t \in \Omega_{\mathrm{E}} \tag{21}$$

$$\sum\nolimits_{t \in \Omega_{\mathrm{E}}} x_{nt} \le N_{\mathrm{F}}, \quad \forall n \in \Omega_{\mathrm{F}} \tag{22}$$

$$\sum\nolimits_{n \in \Omega_{\mathrm{F}}} x_{mn} \le N_{\mathrm{G}}, \quad \forall m \in \Omega_{\mathrm{G}} \tag{23}$$

This model differs from previous models as follows:

- Different from previous models (e.g., [18]), this model additionally considers Constraints (12)–(16) for demand capacities of gateways, fog devices, and edge devices, and the model is designed for deployment in a logistics center.
- Different from previous models (e.g., [18], [19], [6]), this model additionally considers Constraints (17)–(19) for maximal latency times of edge devices, fog devices, and gateways.
- Different from previous models (e.g., [6]), this model additionally includes edge devices, and considers costs of installing fog and edge devices.

IV. Proposed Algorithm

Because IP cannot be solved in deterministically polynomial time, this work proposes a metaheuristic approach for the problem. Monkey algorithm (MA) [13] searches for the optimal solution for a large-scale optimization problem through imitating the mountain-climbing process of a population of monkeys, including the climb, watch-jump, and somersault processes. MA has been shown to be perform more efficiently than conventional metaheuristic algorithms [16]. Because all the decision variables of the concerned problem model are binary, this work modifies the DMA [14] to propose a DMGA which considers binary solution encoding and includes GA operators [17], detailed in Algorithm 1.

Algorithm 1 DMGA

```
 1: Initialize a population of $\lambda$ monkeys, and evaluate cost of each monkey
 2: while each termination criterion is not achieved do
 3:     while a maximal number of iterations is not achieved do
 4:         Conduct a climb process on each monkey
 5:         Conduct a watch-jump process on each monkey
 6:         Update the optimal solution found so far
 7:         Conduct a cooperation process on each monkey
 8:         Conduct crossover and mutation on each monkey
 9:     end while
10:     Conduct a somersault process on each monkey
11: end while
12: Output the optimal solution found so far
```

A. Solution representation

Let M, N, and T denote the numbers of potential sites of gateways, fog devices, and edge devices, respectively. The problem includes four binary decision variables: x_{ij} for links between nodes i and j, g_m for installing gateway m, f_n for installing fog device n, and q_t for installing edge device t. It suffices to determine decision variables of installing devices, after which the decision variable x_{ij} for links can be determined. Therefore, the position of a monkey (solution) is encoded as $(\chi_1, \chi_2, \ldots, \chi_\mu) = (g_1, g_2, \ldots, g_M \mid f_1, f_2, \ldots, f_N \mid q_1, q_2, \ldots, q_T)$.

B. Cost function

Given a monkey position $(g_1, g_2, \ldots, g_M \mid f_1, f_2, \ldots, f_N \mid q_1, q_2, \ldots, q_T)$, the decision variable x_{ij} for each pair of links is set to 1 if the two end nodes of this link are installed; otherwise, it is set to 0. Then, the performance of this position is evaluated by a cost function as follows:

$$\begin{aligned} & c_{\mathrm{f}} \sum_{(i,j) \in \{s\} \times \Omega_{\mathrm{G}} \cup \Omega_{\mathrm{G}} \times \Omega_{\mathrm{F}} \cup \Omega_{\mathrm{F}} \times \Omega_{\mathrm{E}}} x_{ij} \cdot d_{ij} + c_{\mathrm{G}} \sum_{m \in \Omega_{\mathrm{G}}} g_m + c_{\mathrm{F}} \sum_{n \in \Omega_{\mathrm{F}}} f_n \\ & + c_{\mathrm{E}} \sum_{t \in \Omega_{\mathrm{E}}} q_t + \kappa \cdot (\eta_{\text{link}} + \eta_{\text{demand}} + \eta_{\text{latency}} + \eta_{\text{cover}} + \eta_{\text{capacity}}) \end{aligned} \tag{24}$$

where κ is the penalty cost, which a very large number; η_{link}, η_{demand}, η_{latency}, η_{cover}, and η_{capacity} are numbers of violating linkage between two layers of the architecture (i.e., (2), (5), and (8)), satisfaction of demand (i.e., (11)–(16)), maximal latency time (i.e., (17)–(19)), coverage (i.e., (20)), and maximal capacity (i.e., (21)–(23)). In (26), the first four items are from the objective function, and the last item is used to penalize the violation of constraints.

Given a monkey position $(g_1, g_2, \ldots, g_M \mid f_1, f_2, \ldots, f_N \mid q_1, q_2, \ldots, q_T)$, the algorithm of evaluating the cost of this position is given in Algorithm 2.

Algorithm 2 Cost_Evaluation($g_1, g_2, \ldots, g_M \mid f_1, f_2, \ldots, f_N \mid q_1, q_2, \ldots, q_T$)

```
1: Based on the solution encoding, construct the hierarchical architecture of the system (i.e., assigning all $x_{ij}$) in a bottom-up fashion as follows. First, let all link variables $x_{ij}$ between nodes $i$ and $j$ to be 0. Then, for each AGV $k$, find the closest edge device $t$ whose decision variable $q_t = 1$ in the monkey position, and link the two devices, i.e., $x_{tk} = 1$. Then, for each edge device $t$ whose $q_t = 1$, find the closet fog device $n$ whose $f_n = 1$, and link the two devices, i.e., $x_{nt} = 1$. And, set $x_{mn} = 1$ and $x_{sm} = 1$ similarly.
2: Compute $\eta_{link}$ by traversing the architecture created in Line 1. That is, for each active device (i.e., $g_m = 1$, $f_n = 1$, or $q_t = 1$) in the hierarchical architecture, if this device is not linked with an active device at a higher level, then $\eta_{link} = \eta_{link} + 1$.
3: Initialize penalty counters $\eta_{demand} = \eta_{latency} = \eta_{cover} = \eta_{capacity} = 0$.
4: For each $q_t$, if $\sum\nolimits_{k \in \Omega_{\mathrm{A}}} r_A \cdot x_{tk} > H_t^{\mathrm{E}}$, then $\eta_{demand} = \eta_{demand} + 1$. For each $f_n$, if $\sum\nolimits_{t \in \Omega_{\mathrm{E}}} \omega(g_m) \cdot x_{nt} > H_n^{\mathrm{F}}$, then $\eta_{demand} = \eta_{demand} + 1$. For each $g_m$, if $\sum\nolimits_{n \in \Omega_F} \omega(f_n) \cdot x_{mn} > H_m^{\mathrm{G}}$, then $\eta_{demand} = \eta_{demand} + 1$.
5: For each $q_t$, if $\sum\nolimits_{k \in \Omega_{\mathrm{A}}} x_{tk} > N_E$, then $\eta_{capacity} = \eta_{capacity} + 1$. For each $f_n$, if $\sum\nolimits_{t \in \Omega_{\mathrm{E}}} x_{nt} > N_F$, then $\eta_{capacity} = \eta_{capacity} + 1$. For each $g_m$, if $\sum\nolimits_{n \in \Omega_{\mathrm{F}}} x_{mn} > N_G$, then $\eta_{capacity} = \eta_{capacity} + 1$.
6: For each $q_t$, if $L_{\mathrm{A}} / (\sum\nolimits_{k \in \Omega_{\mathrm{A}}} x_{tk} \cdot \gamma_{\mathrm{A}}) \cdot q_t > D_{tk}^{\mathrm{E}}$, then $\eta_{latency} = \eta_{latency} + 1$. For each $f_n$, if $L_{\mathrm{E}} / (\sum\nolimits_{t \in \Omega_{\mathrm{E}}} x_{nt} \cdot \gamma_{\mathrm{E}}) \cdot f_n \le D_{nt}^{\mathrm{F}}$, then $\eta_{latency} = \eta_{latency} + 1$. For each $g_m$, if $L_{\mathrm{F}} / (\sum\nolimits_{n \in \Omega_{\mathrm{F}}} x_{mn} \cdot \gamma_{\mathrm{F}}) \cdot g_m \le D_{mn}^{\mathrm{G}}$, then $\eta_{latency} = \eta_{latency} + 1$.
7: For each $t$ and $k$, if $x_{tk} = 1$ and $x_{tk} \cdot d_{tk} > R_{\mathrm{E}}/2$, then $\eta_{cover} = \eta_{cover} + 1$.
8: With the information of $x_{ij}$ determined in Line 1 and all penalty counters determined in Lines 2–7, the cost of the input position is evaluated according to Equation (24).
```

Let K denote the number of AGVs. Generally, $K \geq T \geq N \geq M$. The time complexity of Algorithm 2 is analyzed as follows.

Theorem 1. *Algorithm 2 can be executed in time $O(K \cdot T)$.*

Proof. All possible connections between devices can be represented as a multi-layer graph $G = \{s\} \times \Omega_G \times \Omega_F \times \Omega_E \times \Omega_A$, in which each device is a node, and each possible link between devices is an edge. Line 1 of Algorithm 2 is to find a tree rooted at node s in graph G, and can be done in time $O(K \cdot T + T \cdot N + N \cdot M)$ by traversing all nodes in a bottom-up fashion (i.e., from Ω_A to Ω_E, then to Ω_F, then to Ω_G, then to $\{s\}$), in which each node at a lower level finds the closest node at a higher level, e.g., each node at Ω_A considers all $|\Omega_E|$ nodes to find the closest node at Ω_E. Line 2 of Algorithm 2 checks the linkage constraint for each active device (i.e., whether each active device is linked with an active device at a higher level), and there are at most μ devices. Hence, Line 2 is done in time $O(\mu)$. Line 3 is done in O(1). In Line 4, for at most μ active devices, each active device at a higher level computes its demand from the active devices at its lower level and then checks the demand constraint. Hence, Line 4 is done in $O(K \cdot T + T \cdot N + N \cdot M)$. Similarly, Lines 5 and 6 are also done in $O(K \cdot T + T \cdot N + N \cdot M)$. It is easy to check that Line 7 is done in $O(K \cdot T)$ and Line 8 is done in $O(K \cdot T + T \cdot N + N \cdot M)$. Since $K \geq T \geq N \geq M$, $O(K \cdot T + T \cdot N + N \cdot M) = O(K \cdot T)$. □

C. Climb process

Let monkey i's position denoted by $X_i = (\chi_{i1}, \chi_{i2}, \ldots, \chi_{i\mu})$. Then, this monkey adopts large-step and small-step climb processes sequentially to reach the mountaintop (locally optimal position). The large-step climb process is as follows:

1. Randomly generate a vector $\Delta X_i = (\Delta\chi_{i1}, \Delta\chi_{i2}, \ldots, \Delta\chi_{i\mu})$,
$$\Delta\chi_{ij} = \begin{cases} (\chi_{ij}+1) \bmod 2, & \text{with probability } 1/2; \\ 0, & \text{with probability } 1/2 \end{cases} \tag{25}$$
for $j = 1, 2, \ldots, \mu$.
2. Let $c(X_i)$ denote the cost of X_i. Calculate $c(X_i + \Delta X_i)$.
3. If $c(X_i + \Delta X_i) < c(X_i)$, then let $X_i = X_i + \Delta X_i$.
4. Repeat Steps 1 to 3 until the cost value remains unchanged for a given number of successive iterations or the maximal number of iterations is achieved.

The small-step climb process is the same with the large-step climb process except for the following step 1:

1. Randomly generate a vector $\Delta X_i = (0, 0, \ldots, \Delta\chi_{ij}, 0, \ldots, 0)$ in turn for $j = 1, 2, \ldots, \mu$, in which $\Delta\chi_{ij} = (\chi_{ij} + 1) \bmod 2$.

The maximal number of iterations for large-step and small-step climb processes are denoted by $N_{C,L}$ and $N_{C,S}$ respectively.

D. Watch-jump process

After the climb process, monkeys have reached their respective mountaintop. Then, they stand at the top, and watch around. If a higher position is found, the monkeys will jump there. Consider monkey i's position denoted by $X_i = (g_{i1}, g_{i2}, \ldots, g_{iM} \mid f_{i1}, f_{i2}, \ldots, f_{iN} \mid q_{i1}, q_{i2}, \ldots, q_{iT})$. The DMGA conducts three types of watch-jump processes for $(g_{i1}, g_{i2}, \ldots, g_{iM})$, $(f_{i1}, f_{i2}, \ldots, f_{iN})$, and $(q_{i1}, q_{i2}, \ldots, q_{iT})$, respectively. The watch-jump processes for the three parts are the same with the large-step climb process except ΔX_i at step 1 is differently defined for the three parts as follows: $\Delta X_i = (\Delta\chi_{i1}, \Delta\chi_{i2}, \ldots, \Delta\chi_{iM}, \mid 0, \ldots, 0 \mid 0, \ldots, 0)$ for the first part; $\Delta X_i = (0, \ldots, 0 \mid \Delta\chi_{i(M+1)}, \Delta\chi_{i(M+2)}\ldots, \Delta\chi_{i(M+N)}, \mid 0, \ldots, 0)$ for the second part; and $\Delta X_i = (0, \ldots, 0 \mid 0, \ldots, 0 \mid \Delta\chi_{i(M+N+1)}, \Delta\chi_{i(M+N+2)}, \ldots, \Delta\chi_{i\mu})$ for the third part, in which $\Delta\chi_{ij}$ is defined as (27). Note that the maximal numbers of iterations for the three watch-jump processes are denoted by $N_{W,1}$, $N_{W,2}$, and $N_{W,3}$, respectively.

E. Cooperation process

Monkeys do not coordinate in climb and watch-jump processes. Hence, the cooperation process allows monkeys to refer to the best position found by all monkeys so far to improve their respective positions. Consider that monkey i's position is $X_i = (\chi_{i1}, \chi_{i2}, \ldots, \chi_{i\mu})$, and the best position found so far is $X^* = (\chi_1^*, \chi_2^*, \ldots, \chi_\mu^*)$. The cooperation process is the same with the large-step climb process except for the following step 1:

1. Randomly generate a vector $\Delta X_i = (\Delta\chi_{i1}, \Delta\chi_{i2}, \ldots, \Delta\chi_{i\mu})$,
$$\Delta\chi_{ij} = \begin{cases} \chi_j^* - \chi_{ij}, & \text{with probability } 1/2; \\ 0, & \text{with probability } 1/2 \end{cases}$$
for $j = 1, 2, \ldots, \mu$.

The maximal number of iterations for the cooperation process is denoted by N_O.

F. Crossover and mutation

This process selects a number of parent monkeys under a crossover rate, and conducts one-point crossover operation on the positions of each pair of the parent monkeys to reproduce two offspring monkeys. Then, conduct mutation on the whole population under a mutation rate. New monkeys replaces the current monkeys that perform worse than new monkeys.

G. Somersault process

The somersault process allows monkeys to search for a new domain based on the median position of all monkeys. The somersault process is divided into two sub-processes. The first sub-process is the same with the large-step climb process except for the following step 1:

1. Calculate p_j for $j \in \{1, 2, \ldots, \mu\}$ as follows:
$$p_j = \begin{cases} 1, & \text{if } (\sum_{i=1}^{\lambda} \chi_{ij}) / \lambda > 0.5; \\ 0, & \text{otherwise.} \end{cases}$$
Randomly generate a vector $\Delta X_i = (\Delta\chi_{i1}, \Delta\chi_{i2}, \ldots, \Delta\chi_{i\mu})$,
$$\Delta\chi_{ij} = \begin{cases} p_j - \chi_{ij}, & \text{with probability } 1/2; \\ 0, & \text{with probability } 1/2 \end{cases}$$
for $j = 1, 2, \ldots, \mu$.

The maximal number of iterations for the somersault process is denoted by N_S. Then, the second sub-process check if χ_{ij} for each i is equal to p_j. If yes, then randomly select one monkey

k, and let $\chi_{kj} = (\chi_{kj} + 1) \bmod 2$.

H. Termination criteria

If the maximal number of iteration is achieved, or the optimal solution found so far have not been changed for K successive iterations, the algorithm is terminated.

V. Implementation and Simulation Results

This section implements the proposed DMGA, and conducts simulation on a moderate-size problem instance. The simulation runs on a PC with Intel Core i7-3770 CPU and 16.0 GB memory. The parameter setting used in the simulation is given as follows: the area of the logistics center is 200m × 180m; from the work in [18], numbers of gateways, fog devices, edge devices, and AGVs are set to 3, 15, 70, and 500, respectively. A generic cost unit (gcu) is defined to evaluate the system cost [28]. From [18], this work sets costs $c_f = 50$, $c_G = 200$, $c_E = 80$, and $c_F = 100$; data rates $\gamma_A = 1000$, $\gamma_E = 4000$, and $\gamma_F = 7000$; the coverage diameter of an edge device $R_E = 75$; maximal capacities $N_E = 20$, $N_F = 10$, and $N_G = 10$; data lengths $L_A = 5000$, $L_E = 10000$, and $L_F = 15000$. The maximal demand capacities are referred to [29], i.e., $H_t^E = 3 \times 10^3$, $H_n^F = 3 \times 10^4$, $H_t^G = 3 \times 10^5$. The maximal latency times are referred to [30], [31], i.e., $D_{tk}^E = 80$, $D_{nt}^F = 20$, and $D_{mn}^G = 15$. From [6], the demand of each AGV follows a uniform distribution on [50, 200]. The parameter setting of the proposed DMGA is given as follows: number of monkeys is 20, crossover rate is 0.05; mutation rate is 0.01; number of iterations is 500; penalty cost κ is 10^5.

Based on the above parameter setting, the simulation results of using the proposed DMGA to deploy edge devices and fog devices in the logistics center in Fig. 1 are shown in Fig. 3. In Fig. 3, black points represent locations of AGVs in the logistics center, and circles represent potential sites of edge devices, in which the DMGA installs edge devices on 23 solid circles, but does not on the other hollow circles. Hence, the installation cost of edge devices is 23 × 80 = 1840 gcu. Similarly, in Fig. 3, squares represent potential sites of fog devices, in which fog devices are installed on 13 solid squares, but are not on the other hollow squares. Hence, the installation cost of edge devices is 13 × 100 = 1300 gcu.

Table 2 shows the statistics of the experimental results of executing 10 times of three algorithms, in which 'Best', 'Average', 'Worst', and 'StdDev' record the best, average, worst, standard deviation of the 10 results; and 'Time' records the average CPU time. Note that the number of iterations of the GA is determined when more iterations do not have any improvement. The average CPU time of the GA (31.907s) is the shortest, followed by DMGA (89.159s), and then DMA (77.521s). That is, the CPU time of the proposed DMGA is improved by including GA operators in the DMA. As for solution performance measures, the DMGA performs best, followed by DMA and then GA. On average, all the results of DMGA, DMA, and GA are feasible (i.e., less than 10^5).

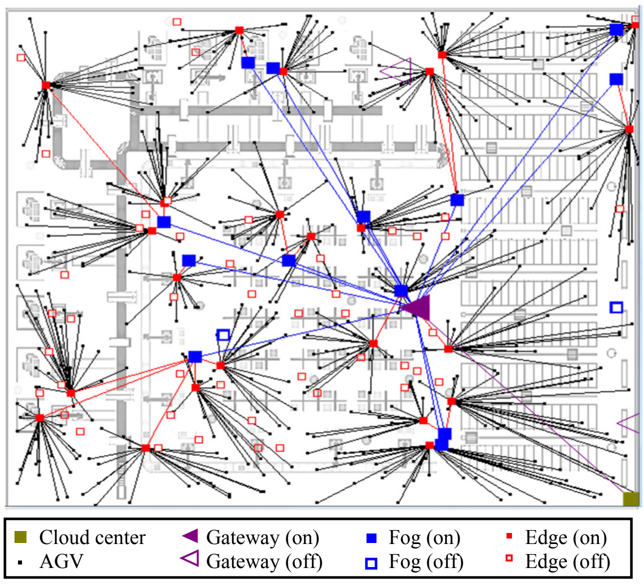

Fig. 3. Result of using the proposed DMGA in the logistics center in Fig. 1.

TABLE 2.
STATISTICS OF EXPERIMENTAL RESULTS OF THREE ALGORITHMS

Approach	Best	Average	Worst	StdDev	Time (s)
DMGA	68121.5	74015.11	80905.0	4852.34	89.159
DMA	69115.5	78929.71	87114.5	6857.06	77.521
GA	69574.0	80675.02	75931.9	3096.81	31.907

To further analyze the stability, Fig. 4 shows the results of 20 different runs of the three algorithms. From Fig. 4, all the three algorithms show high stability and feasibility. Note that the work in [32] showed that the performance of any two optimization algorithms is the same when averaged across all possible problems. Hence, this algorithm may perform well in this problem instance, but may not always perform best.

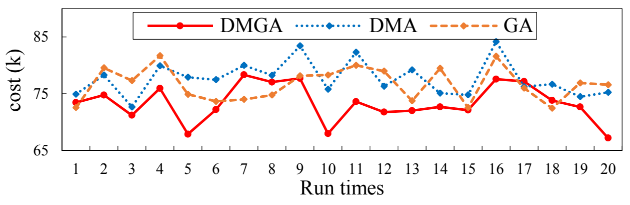

Fig. 4. Comparison of the results of 20 runs of three algorithms.

VI. Conclusion

This work has proposed a framework of the intelligent computing system with fog and edge devices deployed in a logistics center. The deployment of the system was modelled as a mathematical programming model, which determines locations of gateways, fog devices, and edge devices in the logistics center so that the total installation cost is minimized, under constraints of maximal demand capacity, maximal latency time, coverage, and maximal capacity of devices. This work further proposed a DMGA with merits of both MA and GA for solving this problem. Simulation results showed that the proposed DMGA performs well in the problem instances of moderate-size logistics centers.

In the future, it would be of interest to investigate the effect of the deployment on later product flows or scheduling. If the factory layout can be adjusted flexibly, dynamically optimal deployment of fog devices at different times would also be a line of future work. To address larger-scale problem instances, it would be of interest to propose more efficient algorithms using advanced techniques, e.g., coevolution, many-objective optimization, and decomposition. In addition, it would also be interesting to study hybrid methods of metaheuristics and machine learning techniques. Other objectives and uncertain factors of the deployment should also be discussed, e.g., data traffic [33], latency, energy consumption, load balance, heterogeneity, fairness, and QoS.

References

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

[2] D. Georgakopoulos, P. P. Jayaraman, M. Fazia, M. Villari, and R. Ranjan, "Internet of Things and edge cloud computing roadmap for manufacturing," *IEEE Cloud Comput.*, vol. 3, no. 4, pp. 66-73, 2016.

[3] JD.com, Inc., JD.com Fully Automated Warehouse in Shanghai. [Online]. Available: https://www.youtube.com/watch?v=RFV8IkY52iY

[4] J. Zhu, D. S. Chan, M. S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving web sites performance using edge servers in fog computing architecture," in *Proc. of SOSE 2013*, IEEE Press, 2013, pp. 320-323.

[5] S. Sarkar and S. Misra, "Theoretical modelling of fog computing: a green computing paradigm to support IoT applications," *IET Networks*, vol. 5, no. 2, pp. 23-29, 2016.

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

[7] M. Jia, W. Liang, Z. Xu, and M. Huang, "Cloudlet load balancing in wireless metropolitan area networks," in *Proc. of INFOCOM 2016*, IEEE Press, 2016, pp. 1-9.

[8] S. Savazzi, S. Guardiano, and U. Spagnolini, "Wireless sensor network modeling and deployment challenges in oil and gas refinery plants," *Int. J. Distrib. Sens. Netw.*, vol. 9, no. 3, article no. 383168, 2013.

[9] J. Lee, T. Kwon, and J. Song, "Group connectivity model for industrial wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1835-1844, 2010.

[10] T.-J. Wu, W. Liao, and C.-J. Chang, "A cost-effective strategy for road-side unit placement in vehicular networks," *IEEE Trans. Commun.*, vol. 60, no. 8, pp. 2295-2303, 2012.

[11] C.-C. Lin, D.-J. Deng, Z.-Y. Chen, and K.-C. Chen, "Key design of driving Industry 4.0: Joint energy-efficient deployment and scheduling in group-based industrial wireless sensor networks," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 46-52, 2016.

[12] G. L. Nemhauser and L. A. Wolsey, *Integer Programming and Combinatorial Optimization*, Wiley Press, 1988.

[13] R. Zhao and W. Tang, "Monkey algorithm for global numerical optimization," *Journal of Uncertain Systems*, vol. 2, no. 3, pp. 165-176, 2008.

[14] J. Wang, Y. Yu, Y. Zeng, and W. Luan, "Discrete monkey algorithm and its application in transmission network expansion planning," in *Proc. of 2010 IEEE PES General Meeting*, 2010, pp. 1-5.

[15] M. K. Marichelvam, Ö. Tosun, and M. Geetha, "Hybrid monkey search algorithm for flow shop scheduling problem under makespan and total flow time," *Appl. Soft Comput.*, vol. 55, pp. 82-92, 2017.

[16] L. Zheng, "An improved monkey algorithm with dynamic adaptation," *Appl. Math. Comput.*, vol. 222, pp. 645-657, 2013.

[17] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1998.

[18] P. Guo, B. Lin, X. Li, R. He, and S. Li, "Optimal deployment and dimensioning of fog computing supported vehicular network," in *Proc. of 2016 IEEE Trustcom/BigDataSE/ISPA*, 2016, pp. 2058-2062.

[19] L. Yang, J. Cao, G. Liang, and X. Han, "Cost aware service placement and load dispatching in mobile cloud systems," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1440-1452, 2016.

[20] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: A survey," *Future Gener. Comput. Syst.*, vol. 56, pp. 684-700, 2016.

[21] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for Internet of Things applications: Challenges and solutions," *IEEE Commun. Surveys Tut.*, vol. 20, no. 1, pp. 601-628, 2018.

[22] M. Santos De Brito, S. Hoque, R. Steinke, and A. Willner, "Towards programmable fog nodes in smart factories," in *Proc. of FAS*W 2016*, IEEE Press, 2016, pp. 236-241.

[23] N. Chen, Y. Chen, Y. You, H. Ling, P. Liang, and R. Zimmermann, "Dynamic urban surveillance video stream processing using fog computing," in *Proc. of BigMM 2016*, IEEE Press, 2016, pp. 105-112.

[24] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860-3873, 2016.

[25] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular adhoc network with fog computing," in *Proc. of 2015 IFIP/IEEE IM*, 2015, pp. 1202-1207.

[26] C.-C. Lin and D.-J. Deng, "optimal two-lane placement for hybrid VANET-sensor networks," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7883-7891, 2015.

[27] C.-C. Lin, P.-C. Chen, and L.-W. Chang, "On different-dimensional deployment problems of hybrid VANET-sensor networks with QoS considerations," *Mobile Netw. Appl.*, vol. 22, no. 1, pp. 125-138, 2017.

[28] D. A. Chaves, E. A. Barboza, C. J. Bastos-Filho, and J. F. Martins-Filho, "A multi-objective approach to design all-optical and translucent optical networks considering CAPEX and QoT," in *Proc. of ICTON 2012*, IEEE Press, 2012, pp. 1-4.

[29] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171-1181, 2016.

[30] H. Yao, C. Bai, M. Xiong, D. Zeng, and Z. Fu, "Heterogeneous cloudlet deployment and user-cloudlet association toward cost effective fog computing," *Concurr. Comput. Pract. E.*, vol. 29, no. 16, 2017.

[31] B. Confais, A. Lebre, and B. Parrein, "Performance analysis of object store systems in a fog and edge computing infrastructure," in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXIII*, vol. 10430 of LNCS, 2017, pp. 40-79.

[32] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67-82, 1997.

[33] M. Mukherjee, L. Shu, D. Wang, K. Li, Y. Chen, "A fog computing-based framework to reduce traffic overhead in large-scale industrial applications," in *Proc. of INFOCOM WKSHPS*, IEEE Press, 2017, pp. 1-4.

Chun-Cheng Lin (S'06–M'08-SM'17) received the B.S. degree in Mathematics, M.B.A. degree in Business Administration, and Ph.D. degree in Electrical Engineering from National Taiwan University, Taiwan, in 2000, 2007, and 2009, respectively. He has been a Professor of Industrial Engineering and Management (since 2016) and an Associate Dean of College of Management (since 2017) at National Chiao Tung University, which he joined as an Assistant Professor in 2011. He was an Assistant Professor of Computer Science at University of Taipei (2010–2011) and National Kaohsiung University of Applied Sciences (2009–2010). His main research interests include wireless networks, information visualization, algorithm, as well as computational management science. He is a senior member of the IEEE and treasurer of IEEE Taipei Section.

Jhih-Wun Yang received his B.S. degree in Industrial and Information Management from National Yunlin University of Science and Technology, Taiwan, in 2016. He has pursued his M.S. degree in Industrial Engineering and Management from National Chiao Tung University since 2017. His main research interests include facilities planning, logistics management, metaheuristic algorithms, supply chain management, and cloud computing.