A Price-based D2D Resource Allocation Scheme in Smart Manufacturing Scenarios

Shiyong Li
Yanshan University
Xiaofeng Zhang (zhangxf7@stumail.ysu.edu.cn)
Yanshan University
Huan Liu
Yanshan University
Wei Sun
Yanshan University

Research Article

Keywords: Smart Manufacturing, Device to Device, Resource allocation, Utility theory, Lyapunov stability

Posted Date: October 31st, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3499066/v1

Additional Declarations: No competing interests reported.

A Price-based D2D Resource Allocation Scheme in Smart Manufacturing Scenarios

Shiyong Li[1], Xiaofeng Zhang[1*], Huan Liu[1], Wei Sun[1]

[1]School of Economics and Management, Yanshan University, Street, Qinhuangdao, 066004, Hebei, China.

*Corresponding author(s). E-mail(s): zhangxf7@stumail.ysu.edu.cn; Contributing authors: shiyongli@ysu.edu.cn; liuhuan5011@foxmail.com; wsun@ysu.edu.cn;

Abstract

Smart Manufacturing has become a popular topic due to the growth of Internet of Things (IoT) and Cloud Manufacturing, which is driving traditional manufacturing companies to migrate towards digitization and intelligence. Device to Device (D2D) facilitates communication between terminal devices and improves output and operational efficiency. Firstly, by using the utility theory of microeconomics, this paper constructs a novel utility-oriented resource allocation model and addresses the resource allocation problem for D2D communication. This paper also proposes an interesting price-based resource allocation scheme along with an improved gradient descent method and proves the stability of the price-based resource allocation technique based on the Lyapunov stability theory. The simulation results demonstrate that the two proposed algorithms can achieve the equilibrium which is also the optimum of the resource allocation model, and that the gradient descent method is outperformed by the price-based mechanism method in terms of stability and convergence speed.

Keywords: Smart Manufacturing, Device to Device, Resource allocation, Utility theory, Lyapunov stability.

1 Introduction

IoT is a dynamic global network architecture that enables all objects to communicate and exchange information with each other [1]. The Industrial Internet of Things

(IIoT), a concept within the broader scope of IoT that focuses on the industrial sector, facilitates seamless interoperation and connectivity of industrial resources within a network [2]. The ongoing development of IIoT makes it easier for manufacturing enterprises to obtain a rich variety and a large volume of data to promote perceiving the risk [3, 4], making scientific decisions, reducing costs, and improving operational efficiency [5].

Resource allocation in smart manufacturing has emerged as a compelling topic for researchers around the world [6, 7]. Efficient communication is pivotal for the frequent exchange and sharing of industrial big data between devices, deeply influencing the operation and performance of intelligent manufacturing systems [8]. The level of communication efficiency directly affects the interoperation and linkage ability between devices. Devices with low communication rates hinder fast and accurate interoperation performance, resulting in decreased interoperation performance and delayed information feedback during production. This negatively impacts overall production efficiency, quality, real-time control capability, and decision making effectiveness [9, 10].

Regarding the issue of communication resource allocation in smart manufacturing, numerous scholars tend to examine it within the contexts of cloud computing or edge computing [11], while D2D mode is more used to solve the problem of offloading and cost optimization [12, 13]. Bello et al. believe that the D2D has the potential to become an integral part of the IoT [14]. Although they have provided categories of algorithms to facilitate device interoperation, no specific concrete algorithm has been put forward. Moreover, many scholars often employ fuzzy algorithms or traditional precise algorithms to realize optimal resource allocation, lacking the proposal of novel algorithms.

The primary contribution of this paper is the proposal for employing a price-based mechanism to allocate communication resources in the D2D mode. We prove the stability of this algorithm based on the Lyapunov stability theory, and compare its performance with the results obtained from the traditional gradient descent method to demonstrate the superiority and reliability of the proposed algorithm. This in-depth investigation provides new perspectives and methods for optimizing communication in the field of smart manufacturing.

The rest of this paper is structured as follows: Sect. 2 introduces the related work, while Sect. 3 presents an optimal D2D-based resource allocation model in smart manufacturing scenarios and theoretically analyzes it to obtain the optimal solution. Sect. 4, the main work of this paper, introduces the traditional gradient descent method and also a novel price-based scheme to realize the optimal resource allocation, and applies the low-pass filtering principles to enhance the algorithm. In Sect. 5, we combine a typical scenario in smart manufacturing and conduct some numerical simulations in the context of high-definition camera systems to discuss the convergence of both algorithms. Finally, this paper concludes the main results in Sect. 6.

2 Related research

At present, the three-tier architecture of edge computing is widely accepted by scholars, consisting of the cloud layer, edge layer, and device layer [15]. In the smart manufacturing scenario, the device layer is comprised of sensors in vehicles, controllers, actuators, and robots et al. [16]. With the popularity of wireless networks and the increasing number of new network terminals, a new inflection point has occurred [17]. The inefficiency of cloud computing in handling the performance and latency of long-haul networks directly contributes to prolonged transmission delays [18]. Therefore, the mode of Cloud-to-Device is not an appropriate communication way for guaranteeing interactive real-time applications [19] in smart manufacturing.

The edge layer, which is closer to the device, has the advantage of providing powerful storage, communication, computing, and network capabilities in real-time, thus can reduce the response latency substantially [20]. However, it is always be applied to improve system performance [21], such as optimizing business processes [22], reducing operation costs [23], and offering a lightweight transformation scheme for factory transformation [24, 25]. The rise of the IoT has promoted the development of wireless communication technology and sensor technology, providing technical support for D2D [1, 26]. Since edge computing technology solves the problem of real-time calculation and storage, it makes the communication advantage of D2D more obvious. It is very different from the traditional architecture in technology. Devices can directly communicate, transmit, and frequently exchange data between two adjacent devices without going through the base stations or cloud intermediaries [27]. For example, robots can use machine vision equipment to detect the parts on the conveyor belt and communicate with other robots on the production line. The flexible and interactive mode helps alleviate network load issues, enables high-speed data transmission with low latency between devices, and also avoids the issues of signal attenuation.

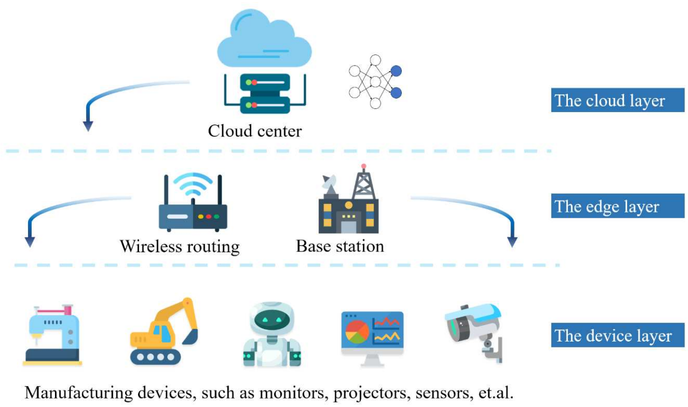

Fig. 1 Overview of collaborative device-edge-cloud allocation architecture

In terms of modelling, common resource allocation models fall into three categories: latency and energy consumption, communication and computing load balancing, and utility maximization [28]. In addition to these traditional metrics, fairness is also widely discussed by scholars. They argued that considering fairness can ensure that resource allocation does not favor certain devices, thereby ensuring the balance and sustainable development of the system [29]. Therefore, they have initiated research on fairness in IP networks [30]. Kelly has shown that proportional fairness can be satisfied when using the logarithmic form [31]. In the smart manufacturing scenarios, more scholars will reduce energy consumption and latency as a goal. Kamoun et al. aimed to reduce energy consumption and optimize the strategy of resource allocation [32]. Zhang et al. [33] deployed a DRL-based algorithm to maximize energy efficiency in the long term in a D2D-enabled heterogeneous cellular network. Latency is always considered a key indicator in smart manufacturing scenarios. Some scholars considered that the IIoT has the characteristics of fixed devices, a large amount of generated sensing data, and a low tolerance for time latency compared with the traditional IoT [34–36]. And Waheed et al. [37] believed self-driving cars, machine vision systems, autonomous storage equipment in logistics systems, industrial robots, et al. need to sense the environment in real-time in a changing environment to perform accurate actions. Fei et al. [38] also considered categorizing task requests by the shortest system delay.

In terms of research methods, many scholars present various models based on metaheuristic algorithms for resource allocation optimization [39]. Seyfollahi et al. [40] used moth-flame optimization (MOF) to optimize the allocation of resources of servers in the IoT framework. Hussein et al. [41] used ant colony optimization (ACO) and particle swarm optimization (PSO) to schedule computation offloading tasks in computing networks, and Alqarni et al. [42] used binary cuckoo search algorithm (BCSA). Akbari et al. [43] investigated the scheduling problems in a distributed and heterogeneous cloud environment by a genetic algorithm. Tanha et al. [44] also used this algorithm and combined it with a thermodynamic simulated annealing algorithm to make improvements. However, optimization algorithms like gradient algorithm and quasi-Newton method can achieve good convergence effect in a short time, fewer researchers have come up with new precise algorithms. Li et al. [45] proposed a pricing scheme to realize reasonable resource allocation in a P2P file-sharing network, and verified its stability with Lyapunov theory. Zhan et al. [46] applied a policy gradient DRL-based approach to solving resource scheduling problems regarded as the Markov decision process. Gao et al. [47] used the gradient descent method to determine the frequency allocation and proved it can minimize the delay and energy. In this paper, we address the resource scheduling problem between edge and end devices as a combinatorial optimization. Traditional heuristic algorithms are found to be inefficient in solving this problem. Therefore, we propose a new allocation algorithm called price-based scheme and compare its performance with the commonly used precise algorithm, namely gradient descent algorithm. Numerical examples are provided to evaluate their performance.

3 Resource sharing model in smart manufacturing

3.1 Problem description

Under the D2D communication mode, devices act not only as resource receivers but also as demanders. Common interoperative application scenarios of D2D communication in smart manufacturing include robot control systems, smart sensor networks, and high-definition camera systems. The specific devices and requirements may vary depending on different application domains and scenarios.

Figure 2 illustrates the device communication in a high-definition camera system. Cameras are installed on monitors and unmanned aerial vehicles (UAVs), they have features such as autofocus and wide-angle views, providing clearer and more accurate visual data. Display devices and control devices interact with the camera devices, the former includes monitors, televisions, projectors, et al., while the latter includes other UAVs, robots, or sensors. By transmitting and receiving the captured video signals, these devices are capable of real-time monitoring, providing security, and enabling remote operation tasks.

High-definition camera systems necessitate high communication speeds and bandwidth due to the large amount of data involved. The actual rate requirements vary depending on specific application needs, device performance and technological limitations, communication environment, and image quality. For instance, common high-definition images such as 1080p full HD-resolution images can be adequately transmitted with a transfer rate of several tens of megabits per second.

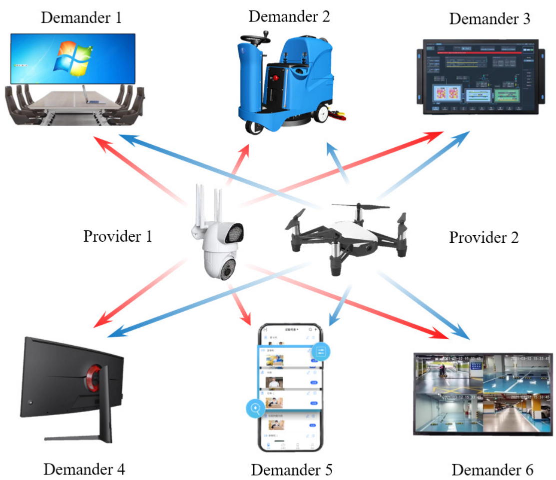

Fig. 2 A D2D-based high-definition camera system

We introduce the set P of resource provider nodes and the set S of resource demander nodes, and investigate the optimal resource allocation between providers and demanders. The goal of resource allocation optimization problem is to maximize the aggregated utility of all device demanders in the network while also considering the fairness of resource allocation. First of all, the objective function is logarithmic form to satisfy the proportional fairness in the optimal resource allocation. Secondly, considering the high latency requirements of devices, the value of ω_s in the objective function is positively correlated with the degree of latency required by the device. If the device has a higher demand for latency, the ω_s value will be larger.

In terms of constraints, the total amount of resources provided by all providers is equal to the total amount obtained by demanders. For each provider $p \in P$, it has its upper limit of upstream link capacity C_p^u for uploading resources, and for each demander $s \in S$, it also has its limit of downstream link capaciy C_s^d for downloading resources. Therefore, we model the resource allocation between devices to devices in edge manufacturing networks as the following optimization problem:

Objective function:

$$\begin{aligned} & Max \sum_{s:s\in S} U_s\left(y_s\right) \\ & U_s\left(y_s\right) = \omega_s \ln y_s \end{aligned} \tag{1}$$

Subject to:

$$\begin{aligned} & \sum_{p:p\in P(s)} x_{ps} = y_s \\ & \sum_{s:s\in S(p)} x_{ps} \le C_p^u \\ & \sum_{p:p\in P(s)} x_{ps} \le C_s^d \\ & x_{ps} \ge 0, p \in P, s \in S \end{aligned} \tag{2}$$

In the resource allocation problem, the notations are as follows:

Table 1 Notation list

Notations	Meanings
S	The set of resource demanders, each element is demander s
P	The set of resource providers, each element is demander p
P_s	The set of providers offering resources for demander $s \in S$
S_p	The set of demanders that request resource from provider $p \in P$
x_{ps}	The resource granted by provider p to demander s
y_s	The aggregated resource granted by providers to demander s
$U_s\left(y_s\right)$	The utility of demander s
C_s^d	The capacity of bandwidth resource of the downstream link of demander s
C_p^u	The capacity of reserved bandwidth resource of the upstream link of provider p for its demanders
ω_s	The parameter correlated with the latency requirement of the device of demander s

3.2 Modeling

In order to solve the above optimization problem, we apply the convex theory to investigate it and find that the utility function Eq. 1 is strictly concave in y_s, but not strictly concave in x_{ps}. At the same time, since the constraints are linear, based on the convex optimization theory [48], we can obtain the following result.

Theorem 1. *The optimal resource allocation y_s^* of each demander s is unique, but the resource value x_{ps}^* provided by each provider to each demander is not unique.*

We will come to Theorem 1 from the following analysis. First, we can obtain the Lagrangian function of this nonlinear optimization problem:

$$
\begin{aligned}
L(x, y; \lambda, \nu, \mu) = & \sum_{s:s \in S} U_s(y_s) + \sum_{s:s \in S} \lambda_s \left(\sum_{p:p \in P(s)} x_{ps} - y_s \right) \\
& + \sum_{s:s \in S} \nu_s \left(C_s^d - \sum_{p:p \in P(s)} x_{ps} \right) + \sum_{p:p \in P(S)} \mu_p \left(C_p^u - \sum_{s:s \in S(p)} x_{ps} \right)
\end{aligned}
\tag{3}
$$

where λ_s, ν_s, μ_{ps} are Lagrange multipliers, λ_s can be considered as the price per unit bandwidth paid by demander s, ν_s is the price per unit bandwidth charged by demander s for downloading communication along with the downstream link and μ_p is the price per unit bandwidth charged by provider p for uploading communication along with the upstream link. Then we can rewrite the Lagrangian function as follows.

$$
\begin{aligned}
L(x, y; \lambda, \nu, \mu) = & \sum_{s:s \in S} (U_s(y_s) - \lambda_s y_s) + \sum_{s:s \in S} \sum_{p:p \in P(s)} x_{ps}(\lambda_s - \nu_s - \mu_p) \\
& + \sum_{p:p \in P(s)} \mu_p C_p^u + \sum_{s:s \in S} v_s C_s^d
\end{aligned}
\tag{4}
$$

The first term in the above equation takes y_s as independent variable, the second term takes x_{ps} as independent variable. The objective function of the dual problem can be written as:

$$
\begin{aligned}
D(\lambda, \nu, \mu) = \max_{x,y} L(x, y; \lambda, \nu, \mu) = & \sum_{s:s \in S} \mathrm{A}_s(\lambda_s) \\
& + \sum_{s:s \in S} \sum_{p:p \in P(s)} \mathrm{B}_{ps}(\lambda_s, \nu_s, \mu_p) + \sum_{p:p \in P(s)} \mu_p C_p^u + \sum_{s:s \in S} v_s C_s^d
\end{aligned}
\tag{5}
$$

Among them,

$$
\mathrm{A}_s(\lambda_s) = \underset{y_s}{max}\, U_s(y_s) - \lambda_s y_s \tag{6}
$$

$$
\mathrm{B}_{ps}(\lambda_s, \nu_s, \mu_p) = \underset{x_{ps}}{max}\, x_{ps}(\lambda_s - \nu_s - \mu_p) \tag{7}
$$

From the above equations we can derive:

$$y_s^*(\lambda_s) = argmax\ U_s(y_s) - \lambda_s y_s \tag{8}$$
$$x_{ps}^*(\nu_s, \mu_p) = argmax\ x_{ps}(\lambda_s - \nu_s - \mu_p) \tag{9}$$

We can interpret the sub-problems Eq. 6 and Eq. 7 from an economic point of view. The subproblem represented by Eq. 6 is a demander problem. All demanders try to maximize their own utility, which depends on the total offered resource y_s for demander s. $\lambda_s y_s$ is the total cost that demander s is willing to pay. Thus, the subproblem Eq. 6 means that each demander wants to maximize its own profit.

And the subproblem represented by Eq. 7 is a provider problem. The goal of this problem is to maximize its total revenue of each provider. $x_{ps}\lambda_s$ is the fee paid by the demander s for using the bandwidth resource x_{ps}, $x_{ps}\nu_s$ is the cost charged by the demander s for down-loading and $x_{ps}\mu_p$ is the cost charged by the provider p for uploading.

Thus, the dual problem of resource allocation model is:

$$min\ D(\lambda, \nu, \mu) \quad over\ \lambda_s \geq 0, \nu_s \geq 0, \mu_p \geq 0, s \in S, p \in P \tag{10}$$

the goal of the dual problem Eq. 10 is to minimize the total price charged by all nodes' transmission links while ensuring that the providers efficiently offer resources to the demanders according to the degree of delay requirements.

3.3 Optimal resource allocation

Let $(x^*, y^*, \lambda^*, \nu^*, \mu^*)$ be the optimal primal and dual variables. And let $\partial L(x, y; \lambda, \nu, \mu)/\partial y_s = 0$. We can obtain the optimal total resource for service demander s:

$$y_s^* = U_s^{-1}(\lambda_s) = \frac{\omega_s}{\lambda_s}\ , {y_s}^* > 0, \forall s : s \in S \tag{11}$$

Substituting Eq. 11 into Eq. 4, and obtain the following Lagrangian function:

$$\begin{aligned} \widehat{L}(x; \lambda, \mu) = &\sum_{s:s\in S}\left(\omega_s \ln\left(\frac{\omega_s}{\lambda_s}\right) - \omega_s\right) + \sum_{s:s\in S}\sum_{p:p\in P} x_{ps}(\lambda_s - \mu_p) \\ &+ \sum_{p:p\in P}\mu_p C_p^u + \sum_{s:s\in S} v_s C_s^d \end{aligned} \tag{12}$$

Let $\partial\widehat{L}(x; \lambda, \mu)/\partial\lambda_s = 0$, and obtain the optimal price paid by demander s:

$$\lambda_s^* = \frac{\omega_s}{\sum\limits_{p:p\in P(s)} x_{ps}} \tag{13}$$

Substitute Eq. 13 into Eq. 12, and then take the partial derivative of x_{ps}, we can get

$$\overline{L}\left(x;\mu\right)=\sum_{s:s\in S}\omega_s\ln\left(\sum_{p:p\in P(s)}x_{ps}\right)+\sum_{p:p\in P}\mu_p\left(C_p^u-\sum_{s:s\in S(p)}x_{ps}\right)$$
$$+\sum_{s:s\in S}\nu_s\left(C_s^d-\sum_{p:p\in P(s)}x_{ps}\right) \tag{14}$$

When $x_{ps}>0$, let $\partial L\left(x;\mu\right)/\partial x_{ps}=0$, then the optimal total resource offered to demander s is

$$y_s^*=\sum_{s:s\in S}x_{ps}=\frac{\omega_s}{\mu_p-\nu_s} \tag{15}$$

In the following analysis, we firstly assume that the constraint on the downlink of de-mander s is nonactive, i.e., $\nu_s=0$, and can obtain the following result.

Theorem 2. *If two providers offer resources to the same demander at the same time, the prices of resources charged by the two providers are equal.*

If demander s obtains resources from two providers $p_1,\ p_2\ \in\ P(s)$, it will result in:

$$\lambda_s^*=\mu_{p1}^*=\mu_{p2}^*=\frac{\omega_s}{\sum\limits_{s:s\in S}x_{ps}}$$
$$x_{ps}^*>0,\forall s:s\in S,\forall p:p\in P \tag{16}$$

Assume that the whole network can be divided into κ regions. Each region has a subset P_κ of providers and a subset S_κ of demanders. According to Theorem 2, the optimal prices charged by resource providers in each region are all $\mu_p=\mu_\kappa,\forall p\in P_\kappa$. The new Lagrangian equation can be expressed as:

$$\begin{aligned}\overline{L}\left(x;\mu\right)&=\sum_{\kappa}\left(\sum_{s:s\in S\kappa}\omega_s(\ln\sum_{p:p\in P\kappa}x_{ps})+\sum_{p:p\in P\kappa}\mu_\kappa\left(C_p-\sum_{s:s\in S\kappa}x_{ps}\right)\right)\\&=\sum_{\kappa}\left(\sum_{s:s\in S\kappa}\sum_{p:p\in P\kappa}\omega_s\left(\ln\frac{\omega_s}{\mu_p}-\frac{\mu_\kappa}{\mu_p}\right)+\mu_\kappa\sum_{p:p\in P\kappa}C_p\right)\\&=\sum_{\kappa}\left(\sum_{s:s\in S\kappa}\omega_s\left(\ln\frac{\omega_s}{\mu_\kappa}-1\right)+\mu_\kappa\sum_{p:p\in P\kappa}C_p\right)\end{aligned} \tag{17}$$

By taking the partial derivative of μ_κ, let $\partial\overline{L}\left(x;\mu\right)/\partial\mu_\kappa = 0$, then

$$\mu_\kappa^* = \frac{\sum\limits_{s:s\in S\kappa} \omega_s}{\sum\limits_{p:p\in P\kappa} C_p} \tag{18}$$

Substituting Eq. 18 into Eq. 15, we can get the optimal price charged by the providers in region κ, and the optimal resource for each demander as follows:

$$y_s^* = \sum_{s:s\in S} x_{ps} = \omega_s \frac{\sum\limits_{p:p\in P\kappa} C_p}{\sum\limits_{s:s\in S\kappa} \omega_s} \tag{19}$$

So, the optimal prices paid by demander are

$$\lambda_i^* = \lambda_j^* = \lambda_\kappa^* = \mu_\kappa^* \; i,j \in S_\kappa \tag{20}$$

Considering the maximum received resource constraint and also the minimum resource requirement of each demander, the optimal resource allocation for each demander can be obtained as follows:

$$y_s^* = \min\left\{C_s^d, \max\left\{A_s^m, \omega_s \frac{\sum\limits_{p:p\in P\kappa} C_p}{\sum\limits_{s:s\in S\kappa} \omega_s}\right\}\right\} \tag{21}$$

4 Resource allocation algorithms

4.1 Gradient descent method

The Gradient Descent method is a mathematical optimal algorithm that leverages local information to obtain optimal resource allocation. At each iteration, intelligent terminal devices calculate the price they should pay to the resource providers and the price charged for downloading resources based on the amounts of resources provided by the providers. Similarly, each resource provider calculates the price charged for uploading resources and updates the resource allocation to resource demanders accordingly. However, the resource allocation model Eq. 1 is not strictly concave. Non-uniqueness may cause oscillations in the optimal resource allocation. To overcome this issue, we introduce an augmented variable and apply the low-pass filtering theory. This approach eliminates potential fluctuations brought by the algorithm without affecting the optimal results, meanwhile accelerating convergence speed.

The concrete steps to realize the algorithm are as follows:

Step 1: Initialize variables and parameters, including the resource amount $x_{ps}(t)$ provided by p for s at time t.

Step 2: Judge whether the termination condition is achieved at this time. Here, the termination condition is set to be $x_{ps}(t+1) = x_{ps}(t)$, which indicates that the algorithm reaches a balanced state and obtains the optimal resource allocation.

Step 3: If the above termination condition is not satisfied, it is necessary to update each price and resource allocation to get $\lambda_s(t+1), \mu_p(t+1), \nu_s(t+1), x_{ps}(t+1)$.

Among them, the updating methods of payment price, upload and download resource prices and resource allocation are as follows:

i. The demander s needs to update the paid price $\lambda_s(t)$ to obtain resources $y_s(t)$ as follows

$$\begin{aligned} \lambda_s(t) &= \frac{\omega_s}{\max\{\eta, y_s(t)\}} \\ y_s(t) &= \sum_{p:p \in P(s)} x_{ps}(t) \end{aligned} \tag{22}$$

ii. The provider p updates the resource price $\mu_p(t)$ and the demander s updates the resource price $\nu_s(t)$

$$\begin{aligned} \nu_s(t+1) &= \left(\nu_s(t) + \alpha_s \frac{y_s(t) - C_s^d}{C_s^d}\right)^+_{v_s(t)} \\ \mu_p(t+1) &= \left(\mu_p(t) + \beta_s \frac{z_p(t) - C_p^u}{C_p^u}\right)^+_{\mu_{ps}(t)} \\ z_p(t) &= \sum_{p:p \in P(s)} x_{ps}(t) \end{aligned} \tag{23}$$

where α_s and β_s are iteration step-sizes, $z_p(t)$ is the total resource provided by provider p to all its customers.

iii. Provider p updates resource allocation $x_{ps}(t)$ provided for demander s

$$\begin{aligned} x_{ps}(t+1) &= (1-\theta_p)\, x_{ps}(t) + \theta_p \widetilde{x_{ps}}(t) + \theta_p \kappa_p x_{ps}(t) \left(\lambda_s(t) - \nu_s(t) - \mu_p(t)\right)^+_{x_{ps}(t)-\varepsilon} \\ \widetilde{x_{ps}}(t+1) &= (1-\theta_p)\, \widetilde{x_{ps}}(t) + \theta_p x_{ps}(t) \end{aligned} \tag{24}$$

where c=$(d)_a^+$ means $c = \max\{0, d\}$, if $a = 0$; and $c = d$ if $a > 0$.

In the above update formula, κ_p represents the iteration step-size and θ_p represents the low-pass filtering parameter, which can eliminate the oscillation phenomenon caused by the non-unique optimal allocation in the iterative optimization process of the algorithm, η and ε are both small positive values, so as to prevent $\lambda_s(t)$ and $x_{ps}(t)$ from being negative.

The flowchart of the basic steps of the proposed algorithm is shown in the following Figure 3:

It should be noted that the iterative step size has a great influence on the convergence speed. Only the selected step size is small enough to ensure its convergence, and if it is too small, the convergence speed will be slow. Choosing a suitable step size can

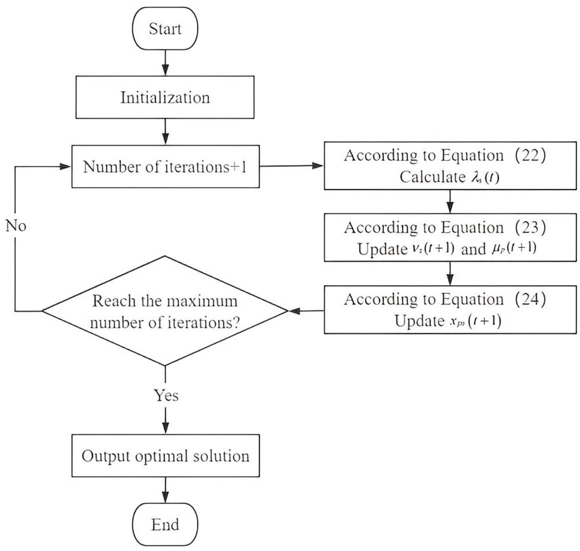

Fig. 3 Flow chart of gradient descent method

facilitate the algorithm to achieve the optimal resource allocation within a reasonable convergence time.

4.2 Price-based mechanism

In this paper we also present a novel price-based resource allocation scheme. The initialization rule and price updating rule of the price-based resource allocation mechanism are the same as those of the gradient descent method above, but the updating scheme of the resource allocation $x_{ps}(t)$ provided by provider p for demander s is different. At time $t = 1, 2, ...$, each resource provider updates its resource allocation according to the following rule:

$$\begin{aligned} x_{ps}(t+1) &= (1-\theta_p)\, x_{ps}(t) + \theta_p \widetilde{x_{ps}}(t) \\ &\quad + \theta_p \tilde{\kappa}_p x_{ps}(t) \left(C_p \lambda_{\mathrm{s}}(t) - \sum_{r:r\in S(p)} x_{pr}(t)\, \lambda_r(t) \right)^{+}_{x_{ps}(t)-\varepsilon} \end{aligned} \tag{25}$$

Similar to the gradient descent method above, the step size selected in the algorithm has a great influence on the convergence speed, so the step size should be small enough to ensure convergence. We can prove the equilibrium and stability of this algorithm by the following derivation:

1. Equilibrium

Among them, $C_p\lambda_s(t)$ represents the total cost charged by provider p when offering resource to demander s. When the total cost $\sum_{r:r\in S(p)} x_{pr}(t)\, \lambda_r(t)$ paid by all demanders is equal to the actual cost charged by the provider, i.e.,

$$C_p \lambda_s^*(t) = \sum_{r:r\in S(p)} x_{pr}^*(t)\lambda_r^*(t) \tag{26}$$

Meanwhile, at the equilibrium

$$\lambda_s^* = \frac{\omega_s}{y_s^*} = \frac{\omega_s}{\sum\limits_{p:p\in P(s)} x_{ps}^*} \tag{27}$$

Based on our earlier analysis in Eq. 17 into Eq. 19, we obtained the optimal resource allocation for each demander and corresponding price charged by providers in region κ. In order to validate the robustness of the algorithm, we are currently conducting an analysis to determine whether this solution can be consistently achieved.

$$\begin{aligned}\sum_{p:p\in P_\kappa} C_p &= \sum_{p:p\in P_\kappa} \frac{\sum\limits_{r:r\in S(p)} x_{pr}^*\lambda_r^*}{\lambda_s^*} = \frac{1}{\lambda_s^*}\sum_{r:r\in S_\kappa}\sum_{p:p\in P(r)} x_{pr}^*\lambda_r \\ &= \frac{y_s^*}{\omega_s}\sum_{r:r\in S_\kappa}\lambda_r\sum_{p:p\in P(r)} x_{pr}^* = \frac{y_s^*}{\omega_s}\sum_{r:r\in S_\kappa}\omega_r\end{aligned} \tag{28}$$

Therefore,

$$\begin{aligned} y_s^* &= \omega_s\frac{\sum\limits_{p:p\in P_\kappa} C_p}{\sum\limits_{r:r\in S_\kappa}\omega_r} \\ \lambda_s^* &= \frac{\sum\limits_{r:r\in S_\kappa}\omega_r}{\sum\limits_{p:p\in P_\kappa} C_p}\end{aligned} \tag{29}$$

This is the same as the result of the model analysis, which shows that when the system is in an equilibrium state, the equilibrium point is the global optimum of the resource allocation model. At the same time, as long as the provider has at least one demander to supply resources, the resources offered by the provider are always the same as its capacity, since the demander is to maximize its utility when the resources of providers are best used.

2. Stability

Theorem 3. *Based on Lyapunov stability theory, the dynamic system can finally converge to the system equilibrium point along all trajectories.*

Proof. The proof process will be mentioned in the appendix. □

5 Simulation and numerical examples

This part gives some numerical examples to evaluate the performance of the resource sharing schemes in smart manufacturing. First, we assume a simple bandwidth resource allocation model in D2D scenario. We consider the high-definition camera system discussed in Sect. 3 as an example. The system is composed of eight devices. Among them, six devices act as demanders while the remaining two devices serve as resource providers. The overall structure of the system is illustrated in Figure 2. The utility functions of the six demanders are respectively:

$$U_1 = 15\ln(y_1)\ ,\ U_2 = 18\ln(y_2,\ U_3 = 14\ln(y_3))$$
$$U_4 = 12\ln(y_4)\ ,\ U_5 = 7\ln(y_5)\ ,\ U_6 = 16\ln(y_6)$$

We consider the bandwidth allocation since it is very important for guaranteeing better communication among devices in edge manufacturing. The provider's upload bandwidth resources are limited to $\boldsymbol{C}$, $\boldsymbol{C} = (C_1^u, C_2^u) = (200, 300)\,\mathrm{Mb/s}$. The bandwidth of download resources of demanders is limited to $\boldsymbol{C'}$,

$$\boldsymbol{C'} = (C_1^d, C_2^d, C_3^d, C_4^d, C_5^d, C_6^d) = (150, 150, 130, 150, 80, 100)\,\mathrm{Mb/s}$$

Firstly, choose step sizes $\eta = \varepsilon = 0.001$, $\kappa_p = 0.5, \tilde{\kappa}_p = 0.1, \alpha_s = 0.2, \beta_s = 0.2$ and the filter factor $\theta_p = 0.2$, and initialize bandwidth resources $x_{ps} = 1Mb/s$. After executing the MATLAB software, a comprehensive analysis reveals that a visually illustrative depiction is acquired after numerous iterative processes. From Figure 4, it becomes apparent that after a certain number of iterations, the charging price of the provider is equal to the payment price of the demander.

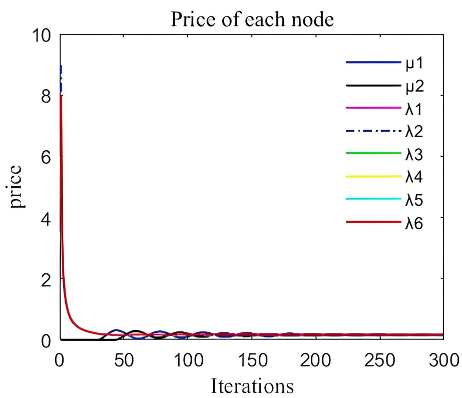

Fig. 4 Price charged by providers and price paid by demanders

The line plots, as depicted in Figure 5 and Figure 6, showcase the resources allocated to individual demanders employing the two algorithms. Remarkably, both algorithms exhibit convergence towards the optimal solution, exhibiting consistent stability across the observed timeframe. We compare the performance of the two kinds of resource allocation algorithms in Figures 5 and 6, in which the solid line represents the convergence process of gradient descent method and the dotted line represents the convergence process of price-based mechanism.

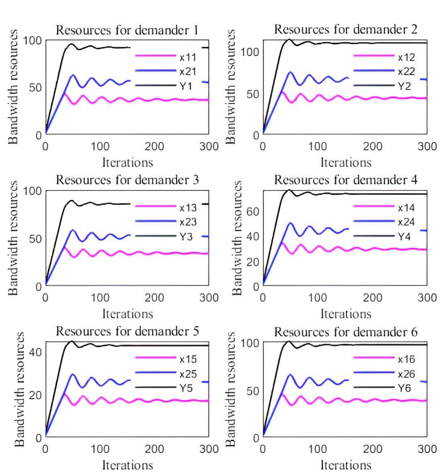

Fig. 5 Optimal resource and utility of demanders obtained by gradient descent method

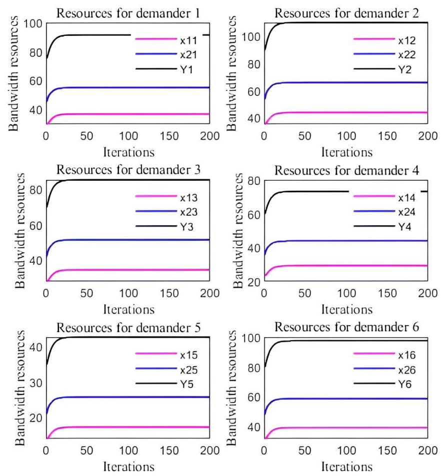

Fig. 6 Optimal resource and utility of demanders obtained by the price-based mechanism

Figure 7 shows the resource allocation of demander 1 obtained by the two algorithms, and Figure 8 shows the convergence process of utility value and total utility of each demander.

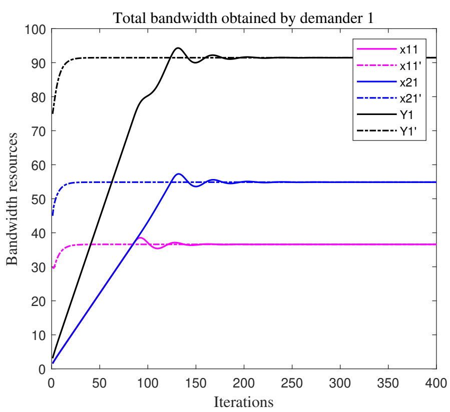

Fig. 7 Resource allocation of demander 1 under the two algorithms

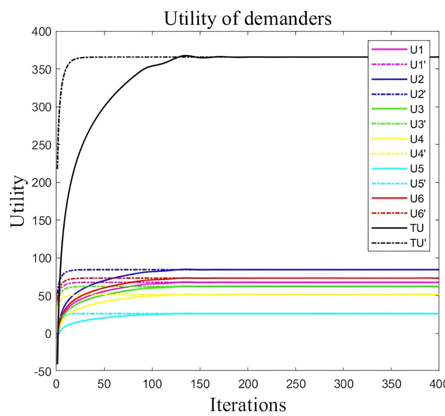

Fig. 8 The utility convergence process of the two algorithms

This result shows that the optimal utility of the two algorithms can be achieved in a limited number of iterations, and compared with the gradient descent algorithm, the price-based mechanism achieves faster convergence and has better stability, while the gradient descent algorithm converges after a period of fluctuation.

By using the nonlinear programming software LINGO, we obtain the optimal resource allocation in Table 2.

Table 2 Results from the two algorithms and LINGO

Variable	GD-based	Price-based	LINGO	Variable	GD-based	Price-based	LINGO
x_{11}^*	36.562	36.586	0	x_{16}^*	39.001	39.026	34.692
x_{21}^*	54.902	54.877	91.463	x_{26}^*	58.56	58.535	62.869
x_{12}^*	43.874	43.905	31.786	y_1^*	39.001	39.026	34.692
x_{22}^*	65.882	65.851	77.97	y_2^*	109.756	109.756	109.756
x_{13}^*	34.124	34.147	85.366	y_3^*	85.366	85.366	85.366
x_{23}^*	51.241	51.219	0	y_4^*	73.171	73.171	73.171
x_{14}^*	29.25	29.367	31.501	y_5^*	42.683	42.683	42.683
x_{24}^*	43.921	43.904	41.67	y_6^*	97.561	97.561	97.561
x_{15}^*	17.062	17.069	16.655	λ	0.164	0.164	0.164
x_{25}^*	25.614	25.614	26.028	TU	365.643	365.644	365.643

It is easy to observe that the utility and total amount of optimal bandwidth resources received by each demander is unique, but the optimal bandwidth resources obtained from different providers are not unique.

6 Conclusion

In this paper, we consider D2D resource allocation in edge manufacturing scenarios and propose a utility maximization model from two aspects of equipment importance and satisfaction. Then, we present the gradient descent algorithm and the price-based mechanism to realize the optimal resource allocation, and apply the low-pass filtering principle to eliminate the possible fluctuations of the algorithm. Through mathematical analysis and Lyapunov theory, it is proved that the price-based mechanism is asymptotically stable, and all trajectories along the price-based scheme can ultimately converge to the optimum of the resource allocation problem. Finally, some numerical examples illustrate that the price-based mechanism can converge in a short time compared with the gradient descent method, and the fluctuation is smaller, which reflects the superiority of the price-based algorithm.

Acknowledgments This work was supported by the National Natural Science Foundation of China (No. 71971188), the Humanities and Social Science Fund of Ministry of Education of China (No. 22YJCZH086), the Natural Science Foundation of Hebei Province (No. G2022203003), and the support Funded by Science and Technology Project of Hebei Education Department (No. ZD2022142).

Author Contributions All authors contributed to the study conception and design. S.L., H.L., X.Z. conceived of the presented idea and S.L., H.L. investigate the algorithms for proposed problem. X.Z. wrote the first draft of the manuscript. S.L. and W.S. revised the manuscript and all authors commented on previous versions of it.

Availability of data and materials No data and materials used in this manuscript.

Declarations

Conflict of interest The authors declare that there are no conflict of interest regarding the pubication of this paper.

Ethics approval Not applicable.

Appendix A Appendix

Define the following Lyapunov function:

$$Q\left(t\right)=Q_1\left(t\right)+Q_2\left(t\right)=\sum_{s:s\in S\kappa}\int\limits_{y_s(t)}^{y_s^*}\left(\frac{\omega_s}{\nu}-\lambda_\kappa^*\right)d\nu+\sum_{p:p\in P\ \kappa}\lambda_\kappa^*\left(C_p-\gamma_p\left(t\right)\right)$$

where, $\gamma_p\left(t\right)=\sum\limits_{s:s\in \mathrm{S}(p)}x_{ps}\left(t\right),p:p\in P_\kappa$, $\lambda_\kappa^*=\lambda_s^*,\forall s\in S_\kappa$. Since $y_s^*>0$ and $y_s\left(t\right)>0$, the first term of the Lyapunov function $Q_1\left(t\right)$ is:

$$\int\limits_{y_s(t)}^{y_s^*}\left(\frac{\omega_s}{\nu}-\lambda_\kappa^*\right)d\nu=\omega_s\left(\ln y_s^*-\ln y_s\left(t\right)\right)-\lambda_\kappa^*\left(y_s^*-y_s\left(t\right)\right)=\omega_s\left(\frac{y_s\left(t\right)}{y_s^*}-1-\ln\frac{y_s\left(t\right)}{y_s^*}\right)\geq 0.$$

From this, it can be seen that, $Q_1\left(t\right)=0$ if and only if $y_s^*=y_s\left(t\right)$. Meanwhile, since $\gamma_p\left(t\right)=\sum_{s:s\in S(p)}x_{ps}\left(t\right)\leq C_p$, the second part of the Lyapunov function $Q_2\left(t\right)\geq 0$, and $Q_2\left(t\right)=0$ if and only if $C_p=\gamma_p\left(t\right)=\sum_{s:s\in S(p)}x_{ps}\left(t\right)$. Thus, it is a positive-definite function, and it is 0 only if the above equation is satisfied $y_s^*=y_s\left(t\right)$ and $C_p=\gamma_p\left(t\right)=\sum_{s:s\in S(p)}x_{ps}\left(t\right)$. Next, verify its asymptotic stability by calculating the derivative of $Q\left(t\right)$.

$$\begin{aligned}\frac{dQ\left(t\right)}{dt}&=\sum_{s:s\in S_\kappa}\frac{\partial Q\left(t\right)}{\partial y_s\left(t\right)}\frac{dy_s\left(t\right)}{dt}+\sum_{p:p\in P\kappa}\frac{\partial Q\left(t\right)}{\partial\gamma_p\left(t\right)}\frac{d\gamma_p\left(t\right)}{dt}\\&=-\sum_{s:s\in S_\kappa}\left(\frac{\omega_s}{y_s\left(t\right)}-\lambda_\kappa^*\right)\sum_{p:p\in P(s)}\frac{dx_{ps}\left(t\right)}{dt}-\sum_{p:p\in P_\kappa}\lambda_\kappa^*\sum_{s:s\in S_{(p)}}\frac{dx_{ps}\left(t\right)}{dt}\end{aligned}$$

$$
\begin{aligned}
&= -\sum_{s:s\in S_\kappa}\left(\lambda_s(t)-\lambda_\kappa^*\right)\sum_{p:p\in P(s)}\frac{dx_{ps}(t)}{dt}-\sum_{p:p\in P_\kappa}\lambda_\kappa^*\sum_{s:s\in S_{(p)}}\frac{dx_{ps}(t)}{dt}\\
&= -\sum_{s:s\in S_\kappa}\sum_{p:p\in P(s)}\theta\lambda_s(t)\,x_{ps}(t)\times\left(C_p\lambda_s(t)-\sum_{r:r\in S(p)}x_{ps}(t)\,\lambda_r(t)\right)\\
&= -\sum_{s:s\in S_\kappa}\sum_{p:p\in P(s)}\theta C_p\lambda_s^2(t)\,x_{ps}(t)+\sum_{p:p\in P_\kappa}\sum_{r:r\in S(p)}\theta x_{ps}(t)\,\lambda_s(t)\sum_{r:r\in S(p)}x_{ps}(t)\,\lambda_r(t)\\
&= -\sum_{s:s\in S_\kappa}\sum_{p:p\in P(s)}\theta\lambda_s^2(t)\,x_{ps}(t)\left(C_p-x_{ps}(t)\right)\\
&+\sum_{s:s\in S(p)}\sum_{p:p\in P_\kappa}\sum_{r:r\in S(p)\backslash\{s\}}\theta x_{ps}(t)\,\lambda_s(t)\times x_{pr}(t)\,\lambda_r(t)
\end{aligned}
$$

Add $\sum_{s:s\in S(p)}\sum_{p:p\in P_\kappa}\sum_{r:r\in S(p)\backslash\{s\}}\theta x_{ps}(t)\,\lambda_s(t)\times x_{pr}(t)\,\lambda_r(t)$ to the first part of the derivative above, and subtract the same term from the second part.

$$
\begin{aligned}
\frac{dQ(t)}{dt} &= -\sum_{s:s\in S_\kappa}\sum_{p:p\in P(s)}\theta\lambda_s^2(t)\,x_{ps}(t)\left(C_p-x_{ps}(t)\right)\\
&+\sum_{s:s\in S(p)}\sum_{p:p\in P_\kappa}\sum_{r:r\in S(p)\backslash\{s\}}\theta x_{ps}(t)\,x_{pr}(t)\,\frac{\left(\lambda_s(t)-\lambda_r(t)\right)^2}{2}
\end{aligned}
$$

Therefore, the conditions for obtaining the positive and negative derivatives are as follows: $\frac{dQ(t)}{dt}=\begin{cases}\le 0 & C_p\ge\sum_{r:r\in S(p)}x_{pr}(t)\\ =0 & C_p=\sum_{r:r\in S(p)}x_{pr}(t),\lambda_\kappa^*=\lambda_s^*=\lambda_s=\lambda_r\end{cases}$. It can be seen that the equilibrium point Eq. 19 of the dynamic system Eq. 23 into Eq. 26 is asymptotically stable based on Lyapunov's stability theory.

References

[1] Al-Fuqaha, A., Guizani, M., Mohammadi, M.: Internet of things: A survey on enabling technologies protocols and applications. IEEE. Commun. Surv. Tutor. **17**(4), 2347–2376 (2015) https://doi.org/10.1109/comst.2015.2444095

[2] Sisinni, E., Saifullah, A., Han, S., *et al.*: Industrial internet of things: Challenges opportunities and directions. IEEE Trans. Ind. Inform. **14**(11), 4724–4734 (2018) https://doi.org/10.1109/tii.2018.2852491

[3] Gao, R.X., Wang, L.H., Helu, M., *et al.*: Big data analytics for smart factories of the future. CIRP Ann. **69**(2), 668–692 (2020) https://doi.org/10.1016/j.cirp.2020.05.002

[4] Wang, J.L., Xu, C.Q., Zhang, J., *et al.*: Big data analytics for intelligent manufacturing systems: A review. J. Manuf. Syst. **62**, 738–752 (2021) https://doi.org/10.1016/j.jmsy.2021.03.005

[5] Parisi, G.I., Kemker, R., Part, J.L., *et al.*: Continual lifelong learning with neural networks: a review. Neural Netw. **113**, 54–71 (2019) https://doi.org/10.1016/j.neunet.2019.01.012

[6] J., Z., Zhou, Y.H., Wang, B.C., *et al.*: Human-cyber-physical systems (hcpss) in the context of new-generation intelligent manufacturing. Eng. **5**(4), 624–636 (2019) https://doi.org/10.1016/j.eng.2019.07.015

[7] Wang, L.H.: From intelligence science to intelligent manufacturing. Eng. **5**(4), 615–618 (2019) https://doi.org/10.1016/j.eng.2019.04.011

[8] Zhong, R.Y., Xu, X., Klotz, E., *et al.*: Intelligent manufacturing in the context of industry 4.0: a review. Eng. **3**(5), 616–630 (2017) https://doi.org/10.1016/j.eng.2017.05.015

[9] Sundarskar, R., Shakeel, P., Baskar, S., *et al.*: Adaptive energy aware quality of service for reliable data transfer in under water acous-tic sensor networks. IEEE Access **7**, 80093–80103 (2019) https://doi.org/10.1109/access.2019.2921833

[10] Wan, J.F., Chen, B.T., Imran, M., *et al.*: Toward dynamic resources management for iot-based manufacturing. IEEE Commun. Mag. **56**(2), 52–59 (2018) https://doi.org/10.1109/mcom.2018.1700629

[11] Tang, C.G., Zhu, C.S., Zhang, N., *et al.*: Sdn-assisted mobile edge computing for collaborative computation offloading in industrial internet of things. IEEE IoT J. **9**(23), 24253–24263 (2022) https://doi.org/10.1109/jiot.2022.3190281

[12] Dai, X.X., Xiao, Z., Jiang, H.B., *et al.*: Task co-offloading for d2d-assisted mobile edge computing in industrial internet of things. IEEE Trans. Ind. Inform. **19**(1), 480–490 (2023) https://doi.org/10.1109/tii.2022.3158974

[13] Chen, Y., Liu, Z.Y., Zhang, Y.C., *et al.*: Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things. IEEE Trans. Ind. Inform. **17**(7), 4925–4934 (2021) https://doi.org/10.1109/tii.2020.3028963

[14] Bello, O., Zeadally, S.: Intelligent device-to-device communication in the internet of things. IEEE Syst. J. **10**(3), 1172–1182 (2016) https://doi.org/10.1109/jsyst.2014.2298837

[15] Rahimi, M., Songhorabadi, M., Kashani, M.H.: Fog-based smart homes: A systematic review. J. Netw. Comput. Appl. **153**, 102531 (2020) https://doi.org/10.1016/j.jnca.2020.102531

[16] Liao, H.J., Zhou, Z.Y., Zhao, X.W., *et al.*: Learning-based context-aware resource allocation for edge-computing-empowered in-dustrial iot. IEEE IoT J. **7**(5), 4260–4277 (2020) https://doi.org/10.1109/jiot.2019.2963371

[17] Chiang, M., Zhang, T.: Fog and iot: An overview of research opportunities. IEEE IoT J. **3**(6), 854–864 (2016) https://doi.org/10.1109/jiot.2016.2584538

[18] Wang, S., Zhang, X., Zhang, Y., *et al.*: A survey on mobile edge networks: Convergence of computing caching and communi-cations. IEEE Access **5**, 6757–6779 (2017) https://doi.org/10.1109/access.2017.2685434

[19] Zhou, Y.Z., Zhang, D.: Near-end cloud computing: Opportunities and challenges in the post-cloud compu-ting era. Chin. J. Comput. **42**(04), 677–700 (2019)

[20] Sardellitti, S., Scutari, G., Barbarossa, S.: Joint optimization of radio and computational resources for multicell mobile-edge computing. IEEE Trans. Signal Inf. Process Netw. **1**(2), 89–103 (2015) https://doi.org/10.1109/tsipn.2015.2448520

[21] Sun, W., Liu, J.J., Yue, Y.L.: Ai-enhanced offloading in edge computing: When machine learning meets industrial iot. IEEE Netw. **33**(5), 68–74 (2019) https://doi.org/10.1109/mnet.001.1800510

[22] Qiu, T., Chi, J.C., Zhou, X.B., *et al.*: Edge computing in industrial internet of things: Architecture, advances and chal-lenges. IEEE Commun. Surv. Tutor. **22**(4), 2462–2488 (2020) https://doi.org/10.1109/comst.2020.3009103

[23] Jun, C., Lee, J.Y., Kim, B.H.: Cloud-based big data analytics platform using algorithm templates for the manufactur-ing industry. Int. J. Comput. Integr. Manuf. **32**(8), 723–738 (2019) https://doi.org/10.1080/0951192x.2019.1610578

[24] Lin, C.C., Deng, D.J., Chih, Y.L., *et al.*: Smart manufacturing scheduling with edge computing using multiclass deep q network. IEEE Trans. Ind. Inform. **15**(7), 4276–4284 (2019) https://doi.org/10.1109/tii.2019.2908210

[25] Rai, R., Sahu, C.K.: Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (cps) focus. IEEE Access **8**, 71050–71073 (2020) https://doi.org/10.1109/access.2020.2987324

[26] Luo, Z., Zheng, X., Wang, B., et al.: Hierarchical computing network collaboration architecture for industrial internet of things. Proc. of 28th Conference on Parallel and Distributed Systems, 57–64 (2023)

[27] Mansour, M., Gamal, A., Ahmed, A.I., *et al.*: Internet of things: A comprehensive overview on protocols, architectures, technolo-gies, simulation tools, and future directions. Energies **16**, 3465 (2023) https://doi.org/10.3390/en16083465

[28] Xu, Z.S., Cai, X.: On consensus of group decision making with interval utility values and interval preference orderings. Group Decis. Negot. **22**(6), 997–1019 (2013) https://doi.org/10.1007/s10726-012-9298-2

[29] Chiang, M., Low, S.H., Calderbank, A.R., *et al.*: Layering as optimization decomposition: A mathematical theory of network architec-tures. Proc. IEEE **95**(1), 255–312 (2017) https://doi.org/10.1109//jproc.2006.887322

[30] Shakkottai, S., Srikant, R.: Network optimization and control. Found. Trends® Netw. **2**(3), 271–379 (2008) https://doi.org/10.1561/1300000007

[31] F., K.: Charging and rate control for elastic traffic. ETT. **8**(1), 33–37 (1997) https://doi.org/10.1002/ett.4460080106

[32] Kamoun, M., Labidi, W., Sarkiss, M.: Joint resource allocation and offloading strategies in cloud enabled cellular networks. Proc. of International Conference on Communications, 5529–5534 (2015) https://doi.org/10.1109/ICC.2015.7249203

[33] Zhang, T., Zhu, K., Wang, J.H.: Energy-efficient mode selection and resource allocation for d2d-enabled heterogeneous networks: A deep reinforcement learning approach. IEEE Trans. Wireless Commun. **20**(2), 1175–1187 (2021) https://doi.org/10.1109/twc.2020.3031436

[34] Xu, H.S., Yu, W., Griffith, D., *et al.*: A survey on industrial internet of things: A cyber-physical systems perspective. IEEE Access **6**, 78238–78259 (2018) https://doi.org/10.1109/access.2018.2884906

[35] Sisinni, E., Saifullah, A., Han, S., *et al.*: Industrial internet of things: Challenges opportunities and directions. IEEE Trans. Ind. Inform. **14**(11), 4724–4734 (2018) https://doi.org/10.1109/tii.2018.2852491

[36] Zhu, L., Li, J.J., Liu, Z.J., *et al.*: A multi-resource scheduling scheme of kubernetes for iiot. J. Syst. Eng. Electron **33**(3), 683–692 (2022) https://doi.org/10.23919/jsee.2022.000063

[37] Waheed, A., Shah, M.A., Mohsin, S.M., *et al.*: A comprehensive review of computing paradigms enabling computation offloading and task execution in vehicular networks. IEEE Access **10**, 3580–3600 (2018) https://doi.org/10.1109/access.2021.3138219

[38] Fei, Z., Wang, Y., Sun, R., et al.: Delay-oriented task scheduling and bandwidth allocation in fog computing networks. Proc. of Global Communications Conference, 1–6 (2019)

[39] Lin, D., Hu, S., Gao, Y., *et al.*: Heuristic-learning-based network architecture for device-to-device user access control. IEEE Commun. Mag. **57**(11), 96–101 (2019) https://doi.org/10.1109/mcom.001.1800862

[40] Seyfollahi, A., Moodi, M., Ghaffari, A.: Mfo-rpl: a secure rpl-based routing protocol utilizing moth-flame optimizer for the iot applications. comp stand interfaces. J. Syst. Eng. Electron **82**, 103622 (2022) https://doi.org/10.1016/j.csi.2022.103622

[41] Hussein, M.K., Mousa, M.H.: Efficient task offloading for iot-based applications in fog computing using ant colony optimization. IEEE Access **8**, 37191–37201 (2022) https://doi.org/10.1109/access.2020.2975741

[42] Alqarni, M., Cherif, A., Alkayyal, E.: Odm-bcsa: an offloading decision-making framework based on binary cuckoo search algorithm for mobile edge computing. Comp. Netw. **226**, 109647 (2022) https://doi.org/10.1016/j.comnet.2023.109647

[43] Akbari, M., Rashidi, H., Alizadeh, S.H.: An enhanced genetic algorithm with new operators for task scheduling in heterogene-ous computing systems. Eng. Appl. Artif. Intell. **61**, 19–36 (2017) https://doi.org/10.1016/j.engappai.2017.02.013

[44] Tanha, M., Shirvani, M.H., Rahmani, A.M.: A hybrid meta-heuristic task scheduling algorithm based on genetic and thermodynam-ic simulated annealing algorithms in cloud computing environments. Neural Comput. Appl. **33**, 16951–16984 (2021) https://doi.org/10.1007/s00521-021-06289-9

[45] Li, S.Y., Sun, W., E, C.G., *et al.*: A scheme of resource allocation and stability for peer–to–peer file–sharing networks. Int. J. App. Math. **26**(3), 707–719 (2016) https://doi.org/10.1515/amcs-2016-0049

[46] Zhan, Y.F., Guo, S., Li, P., *et al.*: A deep reinforcement learning based offloading game in edge computing. IEEE Trans. Comput. **69**(6), 883–893 (2020) https://doi.org/10.1109/tc.2020.2969148

[47] Gao, J., Kuang, Z.F., Gao, J., *et al.*: Joint offloading scheduling and resource allocation in vehicular edge computing: A two layer solution. ieee trans veh technol. IEEE Trans. Veh. Technol. **72**(3), 3999–4009 (2023) https://doi.org/10.1109/tvt.2022.3220571

[48] Bertsekas, D.P. (ed.): Nonlinear Programming. Athena Scientific: Belmont, USA (2003)