Latency Minimization with Optimum Workload Distribution and Power Control for Fog Computing

Saman Atapattu, Chathuranga Weeraddana, Minhua Ding, Hazer Inaltekin and Jamie Evans

Abstract—This paper investigates a three-layer IoT-fog-cloud computing system to determine the optimum workload and power allocation at each layer. The objective is to minimize maximum per-layer latency (including both data processing and transmission delays) with individual power constraints. The resulting optimum resource allocation problem is a mixed-integer optimization problem with exponential complexity. Hence, the problem is first relaxed under appropriate modeling assumptions, and then an efficient iterative method is proposed to solve the relaxed but still non-convex problem. The proposed algorithm is based on an alternating optimization approach, which yields close-to-optimum results with significantly reduced complexity. Numerical results are provided to illustrate the performance of the proposed algorithm compared to the exhaustive search method. The latency gain of three-layer distributed IoT-fog-cloud computing is quantified with respect to fog-only and cloud-only computing systems.

Index Terms—Cloud computing, Fog computing, Internet of Things (IoT), Latency, Power allocation.

I. INTRODUCTION

The fifth generation (5G) of wireless networks and beyond are expected to support billions of connected devices, known as Internet-of-Things (IoT), by using brand-new technologies such as millimeter waves, small cells, multiple antennas, full-duplex and cooperative communications [1]–[4]. To achieve this goal, one key challenge is the efficient processing of data generated at the network edge to meet stringent delay and reliability requirements demanded by the wide range of applications. The cloud computing alone is often not enough to meet all the key performance indicators in these emerging use-cases [5]–[7]. The *fog computing* presents a potential solution for this problem by processing data locally at the fog devices.

A typical fog computing architecture can be modeled with three layers of devices: IoT layer, fog layer, and cloud layer [6]. To investigate different aspects of this set-up, various analytic models for control reliability [7], computing delay [8], transmission delay [8], and energy consumption [7]–[9] are proposed in the existing literature and references therein. For device-to-device fogging [10], an online task offloading problem is considered to minimize the average energy consumption. In [11], an online joint radio and computational resource

S. Atapattu and J. Evans are with the Department of Electrical and Electronic Engineering, the University of Melbourne, Australia (e-mail: {saman.atapattu, jse}@unimelb.edu.au).

C. Weeraddana is with Department of Electronic and Telecommunication Engineering, University of Moratuwa, Moratuwa, Sri Lanka.

M. Ding is with Department of Electronic and Telecommunication Engineering, University of Moratuwa, Moratuwa, Sri Lanka.

H. Inaltekin is with School of Engineering, Macquarie University, North Ryde, NSW 2109, Australia.

management algorithm is developed for multi-user mobile-edge computing systems. In [12], the resource allocation is performed between the end users and their associated small-cell base-stations when they provide cloud-like computing.

For a mobile-edge computing system, the user and data-set size selection is considered to minimize total energy consumption subject to computing latency in [13]. A similar system consisting of single user connected to a computationally capable helper is considered in [14]. In [15], a computing system where the end-users and the cloud are connected via a BS is considered to minimize the energy consumption. A delay-aware and energy efficient computation offloading scheme is proposed in [16] to minimize the consumption of the non-renewable grid energy. In [17], a collaborative computation offloading is studied for cloud and mobile-edge computing to minimize the total energy consumption. The authors in [18] considered a cloud-fog architecture to minimize the maximum computational latency. A trade-off between power consumption and transmission delay in a fog-cloud computing environment is investigated in [8]. Workload allocation strategies within the fog layer subject to a power efficiency constraint are explored in [19].

We note that the fog computing is essentially to complement the cloud computing but *not* to replace the cloud. In general, even the IoT layer can be equipped with certain computational capabilities. Thus, unlike in existing literature, in this paper, we consider a three-layer IoT-fog-cloud distributed computing architecture, where each layer has its *own computational capacity*. We investigate the problem of splitting the workload generated by the IoT layer among the IoT, fog, and cloud. The splitting is performed so that the maximum *latency* at each layer is minimized subject to individual per-layer power constraints. The resulting optimization problem is a mixed-integer program, which is intractable in general. The problem is relaxed with reasonable assumptions. An alternating optimization method is proposed with guaranteed convergence for computing a *good* feasible point of the relaxed non-convex problem. For all considered empirical scenarios, negligible loss of optimality is recorded. The latency gain due to the IoT-fog-cloud computing is quantified with respect to fog-only and cloud-only systems by using the proposed method.

II. SYSTEM MODEL

We consider a network consisting of an IoT device (I), a fog node (F), and a cloud server (C), as shown in Fig. 1. While the IoT device has computing resources, it may also offload some of its processing to the fog and cloud layers. We assume

Fig. 1. Configuration for three-layer IoT-fog-cloud computing system.

that the IoT device has total B [bits] to be processed. It may decide to offload m [bits], $m \leq B$, to the fog node and the fog node in turn may decide to offload k [bits], $k \leq m$, to the cloud node. The workload distribution over the IoT, fog, and cloud layers is then $B - m$, $m - k$ and k [bits], respectively.

Processing time calculations in the proposed fog computing set-up is modelled by introducing computing powers as decision variables. To this end, we start by considering a computing device with processing frequency f [cycles/sec]. Assuming E_c [joules/cycle] is required to run each computing cycle, the power consumption for processing data becomes $P_{\mathrm{p}} = E_c f$ [Watts]. Here E_c is considered as an intrinsic device constant depending on the underlying silicon chip technology. Note that P_{p} is the total energy the computing device consumes per second. Extending this naive approach, a more refined model relating P_{p} and f is $P_{\mathrm{p}} = af^{\beta} + b$ [Watts], where β ranges from 2.5 to 3, and a and b are positive constants obtained by curve fitting against empirical measurements [8], [20]. The constant a embodies the effect of E_c and other device parameters. Equivalently, $f = ((P_{\mathrm{p}} - b)/a)^{\frac{1}{\beta}}$, which is the maximum data processing speed at a given computing power budget P_{p}. Now consider executing an algorithm $\mathcal{A}$ with some given complexity $\mathcal{C}(n)$, as a function of the number of input bits n and in *units of processor cycles* required for algorithm completion. For simplicity, assume that the complexity is linear and given by $\mathcal{C}(n) = cn$ for some positive c [cycles/bit]. Thus, allocating P_{p} to execute $\mathcal{A}$ with n bits requires a time

$$t_{\mathrm{p}} = c\,\frac{n}{f} = c\,n \left(\frac{P_{\mathrm{p}} - b}{a} \right)^{-\frac{1}{\beta}} \quad \text{[sec].} \tag{1}$$

An implicit constraint here is $P_{\mathrm{p}} > b$. Let P_{tI}, P_{tF} and P_{tC} denote the total power budgets for the IoT, fog, and cloud layers, respectively. Typically, $P_{\mathrm{tI}} \ll P_{\mathrm{tF}} \ll P_{\mathrm{tC}}$. The IoT node needs to allocate P_{tI} for its own data processing and communication with the fog layer. Let the local IoT power allocation for data processing and communication be denoted as P_{pI} and P_{cI}, respectively, where $P_{\mathrm{pI}} + P_{\mathrm{cI}} \leq P_{\mathrm{tI}}$. Similarly, power levels P_{pF} and P_{cF} are allocated for data processing and communication at the fog node, respectively, where $P_{\mathrm{pF}} + P_{\mathrm{cF}} \leq P_{\mathrm{tF}}$. Since the cloud only performs data processing at power P_{pC}, we require $P_{\mathrm{pC}} \leq P_{\mathrm{tC}}$. Indexing the a, b and c parameters in (1) with I, F and C, the data processing time at each layer is

$$\text{At I}: t_{pI} = c_I(B - m) \left(\frac{P_{\mathrm{pI}} - b_I}{a_I} \right)^{-\frac{1}{\beta}} \quad \text{[sec],}$$

$$\text{At F}: t_{pF} = c_F(m - k) \left(\frac{P_{\mathrm{pF}} - b_F}{a_F} \right)^{-\frac{1}{\beta}} \quad \text{[sec],}$$

$$\text{At C}: t_{pC} = c_C\,k \left(\frac{P_{\mathrm{pC}} - b_C}{a_C} \right)^{-\frac{1}{\beta}} \quad \text{[sec].}$$

The IoT and fog layers is connected via a wireless link with channel gain h_{IF} and bandwidth W_{IF}. The fog communicates with the cloud via a wireless link or an optical link having channel gain h_{FC} and bandwidth W_{FC}. The throughputs for the IoT-fog and fog-cloud links are given by

$$\text{I to F}: R_{\mathrm{IF}} = W_{\mathrm{IF}} \log_2 \left(1 + \frac{g_{\mathrm{IF}} P_{\mathrm{cI}}}{N_0 W_{\mathrm{IF}}} \right) \quad \text{[bits/sec],}$$

$$\text{F to C}: R_{\mathrm{FC}} = W_{\mathrm{FC}} \log_2 \left(1 + \frac{g_{\mathrm{FC}} P_{\mathrm{cF}}}{N_0 W_{\mathrm{FC}}} \right) \quad \text{[bits/sec],}$$

where $g_{\mathrm{IF}} = |h_{\mathrm{IF}}|^2$, $g_{\mathrm{FC}} = |h_{\mathrm{FC}}|^2$, and N_0 is the noise spectral density. The communication time over each link is given by

$$t_{\mathrm{c,IF}} = \frac{m}{R_{\mathrm{IF}}} \quad \text{[sec] and} \quad t_{\mathrm{c,FC}} = \frac{k}{R_{\mathrm{FC}}} \quad \text{[sec].} \tag{2}$$

The total latency at each stage is determined as follows. For local data processing at the IoT layer, we only have latency T_{I} for processing $B - m$ bits. The latency T_{F} for processing $m - k$ bits at the fog layer is the sum of communication latency of m bits from the IoT layer to the fog layer and the processing latency of the $m - k$ bits. For the cloud, the latency T_{C} for processing k bits is the sum of processing time at the cloud and communication latencies from the IoT layer to the fog layer and from the fog layer to the cloud layer. Assuming that data transmission and processing can be carried out simultaneously, the latencies are given by

$$T_{\mathrm{I}}(m, P_{\mathrm{pI}}) = t_{\mathrm{pI}}; \ T_{\mathrm{F}}(m, k, P_{\mathrm{pF}}, P_{\mathrm{cI}}) = t_{\mathrm{c,IF}} + t_{\mathrm{pF}}$$
$$T_{\mathrm{C}}(m, k, P_{\mathrm{pC}}, P_{\mathrm{cF}}) = t_{\mathrm{c,IF}} + t_{\mathrm{c,FC}} + t_{\mathrm{pC}}. \tag{3}$$

Based on (3), the effective system latency to complete the whole task is given by

$$T = \max\left(T_{\mathrm{I}}, T_{\mathrm{F}}, T_{\mathrm{C}}\right), \tag{4}$$

where T is a function of workload distribution and power allocations at IoT, fog, and cloud layers.

III. OPTIMUM RESOURCE ALLOCATION

A. The Latency Minimization Problem

Our goal is to discover the optimum workload distribution and power allocations at IoT, fog, and cloud layers to minimize T. This optimization problem can be formulated as

$$\text{minimize} \quad T\left(m, k, P_{\mathrm{pI}}, P_{\mathrm{cI}}, P_{\mathrm{pF}}, P_{\mathrm{cF}}, P_{\mathrm{pC}}\right) \tag{5a}$$
$$\text{subject to} \quad 0 \leq m \leq B, \ 0 \leq k \leq m \tag{5b}$$
$$P_{\mathrm{pI}} + P_{\mathrm{cI}} \leq P_{\mathrm{tI}}, \ P_{\mathrm{pF}} + P_{\mathrm{cF}} \leq P_{\mathrm{tF}} \tag{5c}$$
$$P_{\mathrm{pI}} > b_I, \ P_{\mathrm{pF}} > b_F, \ b_C < P_{\mathrm{pC}} \leq P_{\mathrm{tC}} \tag{5d}$$
$$k, m \in \mathbb{Z}, \tag{5e}$$

where $P_{\mathrm{pI}}, P_{\mathrm{cI}}, P_{\mathrm{pF}}, P_{\mathrm{cF}}, P_{\mathrm{pC}}, m$, and k are the decision variables. Note that the optimization problem in (5) is a *mixed-integer nonlinear* problem and is intractable in general [1].

---

[1]Even in the case of mixed integer linear problems, no efficient solution methods exists, except in certain special cases, e.g., total unimodularity conditions hold, see [21, § 13.2].

However, a plausible strategy, especially when the solution for m and k are expected to be *large* integers, is to relax the integer constraints [21, p. 307]. More specifically, we consider the related problem by replacing the integer constraint $k, m \in \mathbb{Z}$ of (5) by $k, m \in \mathbb{R}$, whose epigraph problem is

$$
\begin{aligned}
\text{minimize} \quad & t \\
\text{subject to} \quad & T_I\left(m, P_{\text{pI}}\right) \le t, \ T_F\left(m, k, P_{\text{pF}}, P_{\text{cI}}\right) \le t, \quad (6) \\
& T_C\left(m, k, P_{\text{pC}}, P_{\text{cF}}\right) \le t \\
& \text{Constraints (5b)-(5d)} ,
\end{aligned}
$$

with decision variables $t, P_{\text{pI}}, P_{\text{cI}}, P_{\text{pF}}, P_{\text{cF}}, P_{\text{pC}}, m$, and k [compare with (3) and (4)].

Lemma 1 (Total power usage). *At any optimal point, the power constraints of the problem* (6) *hold with equality, i.e.,* $P_{\text{pI}} + P_{\text{cI}} = P_{\text{tI}}$, $P_{\text{pF}} + P_{\text{cF}} = P_{\text{tF}}$ *and* $P_{\text{pC}} = P_{\text{tC}}$.

Proof: The proof is omitted due to space limitations. ∎

Using Lemma 1, the optimization problem in (6) can equivalently be reformulated as in (7), which is at the top of the next page, where the decision variables are t, α, γ, m, and k. The parameters $g = g_{\text{IF}}/(N_0 W_{\text{IF}})$ and $h = g_{\text{FC}}/(N_0 W_{\text{FC}})$ are introduced for clarity. Although the problem (7) does not exhibit any convexity with respect to the decision variables t, α, γ, m, and k, the problem possesses interesting structural properties that facilitate the application of alternating optimization techniques, as we will discuss next.

B. Solution Approach: Sequential Latency Minimization

For clarity, let $\alpha_{\max} = 1 - b_I/P_{\text{tI}}$ and $\gamma_{\max} = 1 - b_F/P_{\text{tF}}$. The key step in our method to solve (7) is to decompose the latency minimization (7) into two manageable sub-problems that can be solved sequentially in two stages until a convergence criterion is satisfied. The main idea is illustrated in Fig. 2. Let us consider the first iteration. At stage 1, the IoT layer solves for the optimum number of bits $m^{(1)}$ (out of B) that it can assign to the fog layer, so that the overall time to process B bits at the IoT and fog layers is jointly minimized. In particular, the following problem is solved at stage 1 of iteration 1:

$$
\begin{aligned}
\text{minimize} \quad & t \\
\text{subject to} \quad & c_I a_I^{(1/\beta)} \left[\frac{B - m}{\left([1 - \alpha]P_{\text{tI}} - b_I\right)^{\frac{1}{\beta}}} \right] \le t \\
& \left[\frac{m}{W_{\text{IF}} \log_2\left(1 + \alpha g P_{\text{tI}}\right)} + \frac{c_F a_F^{(1/\beta)} m}{\left(P_{\text{tF}} - b_F\right)^{\frac{1}{\beta}}} \right] \le t \\
& 0 \le m \le B, \quad 0 \le \alpha \le \alpha_{\max} , \quad (8)
\end{aligned}
$$

where the decision variables are t, m, α only and the solution is $(t^{(1)}, m^{(1)}, \alpha^{(1)})$. The optimization problem (8) is simply (7) with $k = 0$, $\gamma = 0$, and without the 3rd constraint. The idea is depicted in Figs. 2 and 3. In this example, the IoT layer requires 7 [secs] to process B bits alone. After solving stage 1 optimization problem (8), the IoT and fog layers together require only 5 [secs] to process B bits, which is around 30% latency improvement. The latency at stage 1

includes the processing time of $(B - m^{(1)})$ bits at the IoT *or* communication and processing times of $m^{(1)}$ bits at the fog.

In stage 2 of iteration 1, the fog solves for the optimum number of bits $k^{(1)}$ that can be assigned to the cloud, so that the overall time to process the already assigned $m^{(1)}$ bits at the fog and cloud are jointly minimized, as illustrated in Fig. 2. In other words, the following is solved at stage 2 of iteration 1:

$$
\begin{aligned}
\text{minimize} \quad & s \\
\text{subject to} \quad & \frac{m^{(1)}}{W_{\text{IF}} \log_2\left(1 + g P_{\text{t,I}} \alpha^{(1)}\right)} \\
& \qquad + \frac{c_F a_F^{(1/\beta)}\left(m^{(1)} - k\right)}{\left((1 - \gamma)P_{\text{t,F}} - b_F\right)^{\frac{1}{\beta}}} \le s \\
& \frac{m^{(1)}}{W_{\text{IF}} \log_2\left(1 + g P_{\text{t,I}} \alpha^{(1)}\right)} + \frac{c_C a_C^{(1/\beta)} k}{\left(P_{\text{t,C}} - b_C\right)^{\frac{1}{\beta}}} \\
& \qquad + \frac{k}{W_{\text{FC}} \log_2\left(1 + h P_{\text{t,F}} \gamma\right)} \le s \\
& 0 \le k \le m^{(1)}, \ 0 \le \gamma \le \gamma_{\max} , \quad (9)
\end{aligned}
$$

where the decision variables are s, k, γ and the solution is $\left(s^{(1)}, k^{(1)}, \gamma^{(1)}\right)$. The problem (9) is simply the problem (7), while leaving its first constraint out and considering $m = m^{(1)}$ and $\alpha = \alpha^{(1)}$. Fig. 3 illustrates a situation, where the the aggregate time for communication of $m^{(1)}$ bits from the IoT layer to the fog layer and the processing of $(m^{(1)} - k^{(1)})$ bits at the fog layer is 1 [sec]. So is the aggregate time to communicate and process $k^{(1)}$ bits at the cloud layer. We observe, however, that the total latency is still 5 [secs] since the latency at the IoT layer does not improve after solving the second stage optimization problem. This is the status at the end of the first iteration.

To further improve the latency bottleneck at the IoT layer, in the next iteration, we revert to the stage 1 optimization problem again, which results in 3 [secs] to process $(B - m^{(1)} - m^{(2)})$ bits at the IoT layer after solving (8) with the updated workload distribution. See Fig. 3. The aggregate communication and processing time of $(m^{(1)} + m^{(2)})$ bits at the fog layer is now equal to 3 [secs], without any change in the cloud latency. Then, the solution method again proceeds to stage 2 of the second iteration. The process is thus repeated. Fig. 3 shows the evolution of the aggregate time to process the data at different layers for a case in which the two-stage optimization procedure iterates twice.

Next, we will discuss the properties of the two-stage optimization procedure.

C. Basis for the Two-Stage Optimization Procedure

Implementation of the proposed solution technique requires solving the stage 1 and 2 optimization problems sequentially in an iterative manner. In other words, in any iteration, first the stage 1 optimization is performed followed by the stage 2 optimization. The stage 1 optimization in the ith iteration is

minimize $\quad t$

subject to $\quad c_I a_I^{(1/\beta)} \dfrac{B-m}{((1-\alpha)P_{tI}-b_I)^{\frac{1}{\beta}}} \le t, \quad \dfrac{m}{W_{IF}\log_2(1+\alpha g P_{tI})} + c_F a_F^{(1/\beta)} \dfrac{m-k}{((1-\gamma)P_{tF}-b_F)^{\frac{1}{\beta}}} \le t \qquad (7)$

$$\dfrac{m}{W_{IF}\log_2(1+\alpha g P_{tI})} + \dfrac{k}{W_{FC}\log_2(1+\gamma h P_{tF})} + \dfrac{c_C a_C^{(1/\beta)} k}{(P_{tC}-b_C)^{\frac{1}{\beta}}} \le t, \ 0 \le m \le B, \ 0 \le k \le m, \ \alpha \in [0,\alpha_{max}), \ \gamma \in [0,\gamma_{max}).$$

Fig. 2. Proposed Sequential Optimization Method: Alternating Stage 1 and Stage 2

Fig. 3. Proposed Sequential Optimization Method: (Total Processing + Communication) Time Evolution.

generally expressed as

minimize $\quad t$

subject to $\quad a_{1i}(\alpha) - b_{1i}(\alpha)m \le t, \ d_{1i}(\alpha) + c_{1i}(\alpha)m \le t$

$$0 \le m \le B - \textstyle\sum_{j=0}^{i-1} m^{(j)} \qquad (10)$$
$$0 \le \alpha \le \alpha_{max} - \textstyle\sum_{j=0}^{i-1} \alpha^{(j)} \ ,$$

where the decision variables are t, m, α and the solution is $(t^{(i)}, m^{(i)}, \alpha^{(i)})$. [2] The problem parameters a_{1i}, b_{1i}, c_{1i} and d_{1i} for $i = 1, 2, \ldots$ are defined in (11) on the next page.

For a *fixed* α, the solution of (10) can easily be computed by considering the *intersection* of the lines $a_{1i}(\alpha) - b_{1i}(\alpha)m$ and $d_{1i}(\alpha) + c_{1i}(\alpha)m$. Specifically, $m^{(i)}(\alpha) = \frac{a_{1i}(\alpha)-d_{1i}(\alpha)}{b_{1i}(\alpha)+c_{1i}(\alpha)}$ and $t^{(i)}(\alpha) = \frac{c_{1i}(\alpha)a_{1i}(\alpha)+d_{1i}(\alpha)b_{1i}(\alpha)}{b_{1i}(\alpha)+c_{1i}(\alpha)}$. Based on these, $\alpha^{(i)}$ which solves (10) is given by

$$\alpha^{(i)} = \underset{0 \le \alpha \le \alpha_{max}-\sum_{j=0}^{i-1}\alpha^{(j)}}{\operatorname{argmin}} t^{(i)}(\alpha), \qquad (12)$$

which can be computed by using a scalar grid search over the range of α. Substituting $\alpha^{(i)}$ yields the solutions $m^{(i)}$ and $t^{(i)}$ for (10), respectively.

Similarly, the stage 2 optimization in the ith iteration is

minimize $\quad s$

subject to $\quad L_{2i} + a_{2i}(\gamma) - b_{2i}(\gamma)k \le s$
$\qquad\qquad L_{2i} + d_{2i}(\gamma) + c_{2i}(\gamma)k \le s$

$$0 \le k \le \textstyle\sum_{j=0}^{i} m^{(j)} - \sum_{j=0}^{i-1} k^{(j)} \qquad (13)$$
$$0 \le \gamma \le \gamma_{max} - \textstyle\sum_{j=0}^{i-1} \gamma^{(j)} \ ,$$

where the decision variables are s, k, γ and the solution is $(s^{(i)}, k^{(i)}, \gamma^{(i)})$. The problem parameters $a_{2i}, b_{2i}, c_{2i}, d_{2i}$ and L_{2i} for $i = 1, 2, \ldots$ are given in (14) shown on the next page. Steps for computing the solution $(s^{(i)}, k^{(i)}, \gamma^{(i)})$ for (13) are similar to those for computing $(t^{(i)}, m^{(i)}, \alpha^{(i)})$ in (10). In particular, $k^{(i)}(\gamma) = \frac{a_{2i}(\gamma)-d_{2i}(\gamma)}{b_{2i}(\gamma)+c_{2i}(\gamma)}$ and $s^{(i)}(\gamma) = \frac{c_{2i}(\gamma)a_{2i}(\gamma)+d_{2i}(\gamma)b_{2i}(\gamma)}{b_{2i}(\gamma)+c_{2i}(\gamma)} + L_{2i}$, which are used to determine $(s^{(i)}, k^{(i)}, \gamma^{(i)})$. The optimal point $\gamma^{(i)}$ of (13) is given by

$$\gamma^{(i)} = \underset{0 \le \gamma \le \gamma_{max}-\sum_{j=0}^{i-1}\gamma^{(j)}}{\operatorname{argmin}} s^{(i)}(\gamma), \qquad (15)$$

and $k^{(i)}, s^{(i)}$ are computed by evaluating their expressions at $\gamma^{(i)}$.

D. Sequential Latency Minimization (SLM) Algorithm

In this section, based on the results in § III-C, we outline the sequential latency minimization (SLM) algorithm followed by its convergence properties.

The SLM algorithm is summarized in Algorithm 1. Step 1 initializes the SLM. Steps 2 and 3 are the stage 1 and stage 2 optimization problems, respectively. The stopping criterion is checked at step 4. Finally, step 5 computes the aggregate workload at the fog and cloud layers $m^\star$ and $k^\star$, together with the power split values at the IoT and fog layers $\alpha^\star$ and $\gamma^\star$, respectively. The associated latency is given by $t^\star$. The SLM algorithm always terminates after finitely many iterations, as

---

[2] The formulation ensures that the cumulative number of bits transmitted from IoT by the end of stage 1 of ith iteration is no smaller than that of $(i-1)$th iteration and so is for the communication power.

$$a_{1i}(\alpha) = \left[\frac{c_I a_I^{(1/\beta)} \left(B - \sum_{j=0}^{i-1} m^{(j)} \right)}{\left(\left[1 - \sum_{j=0}^{i-1} \alpha^{(j)} - \alpha \right] P_{t,I} - b_I \right)^{\frac{1}{\beta}}} \right] ; \ c_{1i}(\alpha) = \frac{1}{W_{IF} \log_2 \left(1 + g P_{t,I} \left[\sum_{j=0}^{i-1} \alpha^{(j)} + \alpha \right] \right)} + \frac{c_F a_F^{(1/\beta)}}{\left(\left[1 - \sum_{j=0}^{i-1} \gamma^{(j)} \right] P_{t,F} - b_F \right)^{\frac{1}{\beta}}} ;$$

$$b_{1i}(\alpha) = \frac{a_{1i}(\alpha)}{B - \sum_{j=0}^{i-1} m^{(j)}} ; \ d_{1i}(\alpha) = \frac{\sum_{j=0}^{i-1} m^{(j)}}{W_{IF} \log_2 \left(1 + g P_{t,I} \left[\sum_{j=0}^{i-1} \alpha^{(j)} + \alpha \right] \right)} + \frac{c_F a_F^{(1/\beta)} \sum_{j=0}^{i-1} \left(m^{(j)} - k^{(j)} \right)}{\left(\left[1 - \sum_{j=0}^{i-1} \gamma^{(j)} \right] P_{t,F} - b_F \right)^{\frac{1}{\beta}}}. \quad (11)$$

$$a_{2i}(\gamma) = c_F a_F^{(1/\beta)} \left[\frac{\left(\sum_{j=0}^{i} m^{(j)} - \sum_{j=0}^{i-1} k^{(j)} \right)}{\left(\left[1 - \sum_{j=0}^{i-1} \gamma^{(j)} - \gamma \right] P_{t,F} - b_F \right)^{\frac{1}{\beta}}} \right] ; \ c_{2i}(\gamma) = \frac{1}{W_{FC} \log_2 \left(1 + h P_{t,F} \left[\sum_{j=0}^{i-1} \gamma^{(j)} + \gamma \right] \right)} + \frac{c_C a_C^{(1/\beta)}}{(P_{t,C} - b_C)^{\frac{1}{\beta}}} ; ;$$

$$b_{2i}(\gamma) = \frac{a_{2i}(\gamma)}{\sum_{j=0}^{i} m^{(j)} - \sum_{j=0}^{i-1} k^{(j)}} ; \ d_{2i}(\gamma) = c_{2i}(\gamma) \sum_{j=0}^{i-1} k^{(j)} ; \ L_{2i} = \frac{\sum_{j=0}^{i} m^{(j)}}{W_{IF} \log_2 \left(1 + g P_{t,I} \left[\sum_{j=0}^{i} \alpha^{(j)} \right] \right)}. \quad (14)$$

Algorithm 1 SLM Algorithm

1: **Initialization:** Set $i = 1$, $(m^{(i-1)}, \alpha^{(i-1)}) = (0,0)$ and $(k^{(i-1)}, \gamma^{(i-1)}) = (0,0)$. Let $\epsilon > 0$ be an accuracy level.
2: Solve problem (10) to yield $t^{(i)}, m^{(i)}$ and $\alpha^{(i)}$.
3: Solve problem (13) to yield $s^{(i)}, k^{(i)}$ and $\gamma^{(i)}$.
4: If $|t^{(i)} - s^{(i)}| \leq \epsilon$, go to step 5. Otherwise, set $i = i + 1$ and go to step 2.
5: **Output:** Let $t^{\star} = t^{(i)}$, $m^{\star} = \sum_{j=0}^{i} m^{(j)}$, $k^{\star} = \sum_{j=0}^{i} k^{(j)}$, $\alpha^{\star} = \sum_{j=0}^{i} \alpha^{(j)}$, and $\gamma^{\star} = \sum_{j=0}^{i} \gamma^{(j)}$ and STOP.

---

shown in Theorem 1, which is a consequence of following lemmas.

Lemma 2. *For any positive integer i, $t^{(i)} \geq s^{(i)}$.*

Proof: This follows simply by noting that $s = t^{(i)}$, $k = 0$, and $\gamma = 0$ is feasible for problem (13). Thus, the optimal value $s^{(i)}$ of problem (13) no greater than $t^{(i)}$. ∎

Lemma 3. *The sequence $t^{(i)}$ is strictly monotonically decreasing and bounded below. Moreover, the sequence $s^{(i)}$ is strictly monotonically increasing and bounded above.*

Proof: Only an outline of the proof is provided. At the end of the first iteration, $t^{(1)} \geq s^{(1)}$ according to Lemma 2. If $t^{(1)} = s^{(1)}$, the algorithm exits. Otherwise, $t^{(1)} > s^{(1)}$. Assuming this is the case, consider the second iteration. To solve the stage 1 problem, the left-hand sides of the first two inequalities in (10) must be set equal, which leads to $t^{(1)} > t^{(2)} > s^{(1)}$. Similarly, to solve the stage 2 problem in (13), the left-hand sides of the first two inequality constraints must be balanced, and thus $s^{(1)} < s^{(2)} < t^{(2)}$. Therefore, at the end of the second iteration, $t^{(2)} < t^{(1)}$, $s^{(2)} > s^{(1)}$. The iterations continue in this manner and the proof is concluded. ∎

Theorem 1. *The SLM algorithm terminates in finite time. In particular, $\lim_{i \to \infty} \left(t^{(i)} - s^{(i)} \right) = 0$.*

Proof: The proof is based on Lemma 3. ∎

IV. NUMERICAL RESULTS

In this section, numerical examples are provided to compare SLM algorithm and the optimum exhaustive search method. We consider a computing scenario in which the IoT, fog, and cloud layers are implemented with processors Quark X1000 400 MHz, Xeon E7450 Dunnington 2.4 GHz, and Xeon Platinum 8156-Intel 3.6 GHz, respectively, with maximum power dissipations of 2.2 W, 90 W, and 105 W, as given in various Intel CPU specifications. According to (1), we select a_I, a_F and a_C to satisfy these maximum powers for $\beta = 3$ and $b_I = b_F = b_C = 10^{-3}$. The signal-to-noise ratios (SNRs) of the links between the IoT and fog layers and the fog and cloud layers are defined as $\text{SNR}_{IF} = P_{tI}/(N_0 W_{IF})$ and $\text{SNR}_{FC} = P_{tF}/(N_0 W_{FC})$, respectively. The wireless channel gain between the IoT and fog layers is exponentially distributed with unit mean. We set other parameters as $c_I = 5$, $c_F = 2$, $c_C = 1$, $W_{IF} = W_{FC} = 500$ MHz, $\text{SNR}_{FC} = 32$ dB, and $N_0 = 10^{-10}$ Watts/Hz. We calculate the average latency over 4000 channel realizations.
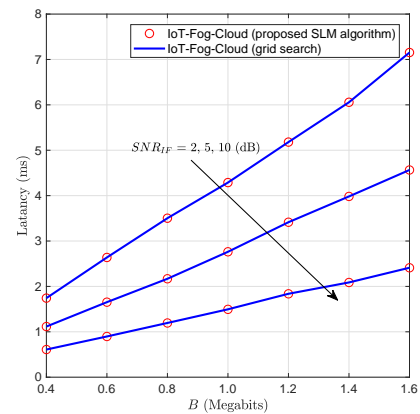

Fig. 4. The average latency vs workload B for different SNR_{IF}.

Figure 4 shows the average latency (in milli-seconds) vs workload B (in Megabits) for both the proposed SLM algorithm and the optimal grid search when $\text{SNR}_{IF} = 2, 5, 10$ dB.
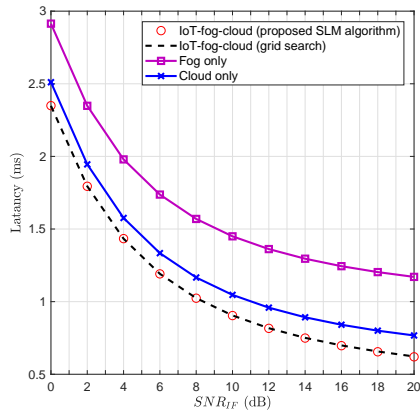
Fig. 5. The average latency vs SNR_{IF} for different computing systems.

The optimum value is obtained through the exhaustive two-dimensional grid search with a granularity of 10^{-2} in each dimension. Clearly, the results of both methods coincide, which suggests that the SLM algorithm performs very close to the optimum method. Figure 4 also indicates that the latency increases almost linearly with workload B. For example, for the simulated range at $\text{SNR}_{\text{IF}} = 5\,\text{dB}$, latency increases from $1.1\,\text{ms}$ to $4.5\,\text{ms}$, where we need $2.9\,\text{ms}$ to process one Megabits of data. Further, to achieve $2\,\text{ms}$ latency, we can process approximately 0.45, 0.75 and 1.35 Megabits when $\text{SNR}_{\text{IF}} = 2, 5, 10\,\text{dB}$, respectively.

Figure 5 depicts the average latency (in milli-seconds) vs SNR_{IF} when workload $B = 1\,\text{Megabits}$. It compares three different architectural choices: i) IoT-fog-cloud; ii) fog-only; and iii) cloud-only. The average latency decreases when SNR_{IF} increases, as expected. Results shows that the IoT-fog-cloud computing architecture always outperforms others. For example, to yield a $1\,\text{ms}$ latency, the IoT-fog-cloud computing system requires $\text{SNR}_{\text{IF}} = 8\,\text{dB}$, whereas the cloud-only computing system needs $\text{SNR}_{\text{IF}} = 11\,\text{dB}$. The fog-only computing system cannot yield a $1\,\text{ms}$ latency even when $\text{SNR}_{\text{IF}} = 20\,\text{dB}$. The IoT-fog-cloud computing architecture always yields a decrease in the latencies, irrespective of SNR_{IF}. For example, at $\text{SNR}_{\text{IF}} = 16\,\text{dB}$, the increase in latencies of the fog-only and cloud-only computing systems, compared to the IoT-fog-cloud computing system is 79% and 21%, respectively.

V. Conclusion

The power and workload allocation problem to minimize data processing latency for a three-layer IoT-fog-cloud computing systems was investigated. The resulting problem is non-convex. To devise an efficient solution method, a constraint relaxation was considered yielding, under reasonable grounds, *a very good* approximation to the original problem formulation. A sequential latency minimization (SLM) algorithm based on alternating optimization was proposed to handle the relaxed problem. Convergence of the SLM algorithm was established. Numerical results suggested that the performance of SLM algorithm was almost identical to that of the optimum exhaustive search method for the relaxed problem. Finally, we evaluated numerically the gains of the three-layer IoT-

fog-cloud computing over fog-only and cloud-only computing, in terms of data processing latencies. Results suggest that the three-layer computing is more potent, for yielding better latencies, than fog-only or cloud-only computing systems.

References

[1] L. Liu, R. Chen, S. Geirhofer, K. Sayana, Z. Shi, and Y. Zhou, "Downlink MIMO in LTE-advanced: SU-MIMO vs. MU-MIMO," *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 140–147, Feb. 2012.

[2] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proc. the IEEE*, vol. 102, no. 3, pp. 366–385, Mar. 2014.

[3] S. Atapattu, Y. Jing, H. Jiang, and C. Tellambura, "Relay selection and performance analysis in multiple-user networks," *IEEE J. Select. Areas Commun.*, vol. 31, no. 8, pp. 1517–1529, Aug. 2013.

[4] S. Atapattu, P. Dharmawansa, M. Di Renzo, C. Tellambura, and J. S. Evans, "Multi-user relay selection for full-duplex radio," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 955–972, Feb. 2019.

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

[6] M. Gorlatova, H. Inaltekin, and M. Chiang, "Characterizing task completion latencies in fog computing," *Technical Report*, Nov. 2018. [Online]. Available: https://arxiv.org/abs/1811.02638.

[7] H. Inaltekin, M. Gorlatova, and M. Chiang, "Virtualized control over fog: Interplay between reliability and latency," *IEEE Internet of Things J.*, vol. 5, no. 6, pp. 5030–5045, Dec. 2018.

[8] 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 of Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.

[9] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog computing may help to save energy in cloud computing," *IEEE J. Select. Areas Commun.*, vol. 34, no. 5, pp. 1728–1739, May 2016.

[10] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration," *IEEE J. Select. Areas Commun.*, vol. 34, no. 12, pp. 3887–3901, Dec. 2016.

[11] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[12] L. Chen, S. Zhou, and J. Xu, "Computation peer offloading for energy-constrained mobile edge computing in small-cell networks," *IEEE/ACM Trans. Networking*, vol. 26, no. 4, pp. 1619–1632, Aug. 2018.

[13] C. You, K. Huang, H. Chae, and B. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[14] Y. Tao, C. You, P. Zhang, and K. Huang, "Stochastic control of computation offloading to a dynamic helper," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2018.

[15] N. T. Ti and L. B. Le, "Computation offloading leveraging computing resources from edge cloud and mobile peers," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017.

[16] X. He, Y. Chen, and K. K. Chai, "Delay-aware energy efficient computation offloading for energy harvesting enabled fog radio access networks," in *Proc. IEEE Vehicular Technology Conf. (VTC)*, Jun. 2018.

[17] H. Guo and J. Liu, "Collaborative computation offloading for multi-access edge computing over fiberwireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4514–4526, May 2018.

[18] G. Lee, W. Saad, and M. Bennis, "An online secretary framework for fog network formation with minimal latency," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017.

[19] Y. Xiao and M. Krunz, "QoE and power efficiency tradeoff for fog computing networks with fog node cooperation," in *Proc. IEEE INFOCOM*, May 2017.

[20] L. Rao, X. Liu, M. D. Ilic, and J. Liu, "Distributed coordination of internet data centers under multiregional electricity markets," *IEEE Proc.*, vol. 100, no. 1, pp. 269–282, Jan. 2012.

[21] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity.* Englewood Cliffs New Jersey: Prentice-Hall, 1982.