

# データ処理時間最小化を目的とした MEC & UE の計算能力および無線 チャンネル状態に基づくリソース割り当て法の検討

小林 龍司<sup>†</sup> 安達 宏一<sup>†</sup>

<sup>†</sup> 電気通信大学 先端ワイヤレス・コミュニケーション研究センター  
〒182-8585 東京都 調布市 調布ヶ丘 1-5-1  
E-mail: †{r.kobayashi,adachi}@awcc.uec.ac.jp

あらまし モバイルデバイスの高性能化に伴い、モバイルアプリケーションやサービスに関心が集まっているが、アプリケーションを処理するためには高い計算能力が必要となる。モバイルデバイスの計算能力は依然として限られているため、複雑な処理を短時間で実現するモバイルエッジコンピューティング (MEC: Mobile Edge Computing) が注目されている。MEC 環境において、複数の無線端末が処理を必要とするデータを保持している場合、無線端末と MEC を具備した基地局間の無線チャンネル状態及び、MEC と無線端末のデータ処理能力を考慮した無線並びに計算リソースの割り当てが必要となる。本稿では、複数の無線端末が分割可能なタスクを有する MEC 環境において、計算タスクの処理完了時間を最小化するための無線リソースと計算リソースの割り当て法を検討する。この時、計算タスクの処理完了時間は、タスクのアップロード、MEC サーバでの計算タスクの実行、及び MEC サーバで処理終了後の計算結果の返送の 3 つのフェーズから構成され、モバイルデバイスは自身でもローカルで処理可能であるものとする。この最適化問題に対して二分探索法とジョンソンアルゴリズムを用いた無線リソース並びに計算リソース割り当て法を提案する。計算機シミュレーションにより、提案アルゴリズムは既存方式と比較して、最大で 37% 程度タスクの処理完了時間を低減できることを示す。

キーワード MEC, エッジコンピューティング, セルラーネットワーク, リソース割り当て

## Computational Capability of MEC & UE and Wireless Channel Condition Based Resource Allocation for Data Processing Time Minimization

Ryuji KOBAYASHI<sup>†</sup> and Koichi ADACHI<sup>†</sup>

<sup>†</sup> Advanced Wireless and Communication Research Center, The University of Electro-Communications  
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585  
E-mail: †{r.kobayashi,adachi}@awcc.uec.ac.jp

**Abstract** Due to the rapid advancement of the computing power of mobile devices, the interest in mobile applications and services have been rapidly growing. Since the computing power of mobile devices are not yet to be sufficient to handle heavy applications such as augmented reality and image recognition, the concept of mobile edge computing (MEC) has attracted huge attention. A MEC server which can handle such heavy applications are located at the edge of the radio access network such as base stations. Thus, user equipments (UEs) can request the MEC server to process the task and get the response within short time. If multiple UEs possess the data that need to be processed by MEC server, the radio and computational resources need to be allocated to each UE by taking into account the wireless channel condition and the computational power of UEs and MEC. In this paper, we consider the allocation method of the radio and computation resources to minimize the processing completion time of computation task when the tasks can be divided. The processing completion time of the tasks are composed of three phases, uploading the tasks, executing the computation tasks at the MEC server, and returning the computation result after the processing at the MEC server. In addition, the UE itself can compute locally. For the formulated optimization problem, we propose a radio and computational resources allocation method using binary search method and Johnson's algorithm. Our simulation result show that the proposed algorithm can reduce the task completion time up to 37% compared to the existing methods.

**Key words** Mobile edge computing, cellular network, resource allocation

## 1. はじめに

モバイルデバイスが日々の生活に浸透するにつれ、インターネットやビデオストリーム分析、拡張現実 (AR: Augmented Reality) やオブジェクト認証 (顔認証) など、より多くの新しいモバイルアプリケーションが出現している。しかしながら、モバイルデバイスは限られた計算能力や記憶容量など、リソースの制約がある。そこで高度な計算能力等を必要とするモバイルアプリケーションの要求に対応するために、モバイルクラウドコンピューティング (MCC: Mobile Cloud Computing) が注目されている。MCC はリソース活用と、計算能力を大幅に向上させる一方で、バックボーンネットワーク上での伝送容量制限や遅延変動を考慮すると、クラウドサーバとモバイルデバイスの遠隔性に起因して、オフロード効率が著しく劣化する可能性がある [1]。

そこで、無線ネットワークのエッジにクラウドコンピューティングとストレージ能力を提供し、高度なアプリケーション要求をモバイルデバイスの近くで処理できるようにするモバイルエッジコンピューティング (MEC: Mobile Edge Computing) と呼ばれる技術が注目を集めている。MEC はモバイルデバイスとの近接性に起因し、計算オフロードサービスにおいて高速な対話型応答や、アプリケーションの待ち時間、またデバイスのエネルギー効率などを大きく改善することができる。

MEC サーバは MCC と比較して高速な計算オフロードサービスを提供できる一方、リアルタイム性の高いアプリケーションの増加に伴い、さらなる低遅延性が求められる。さらに、MEC を備えた基地局 (BS: Base Station) と MEC でのアプリケーション処理を要求するモバイルユーザ (MU: Mobile User) 間の通信に必要な周波数や時間などの無線リンクでの無線リソースと、MEC サーバでの CPU などの計算リソースが MEC の主な制限要因としてあげられ、無線・計算リソースの有効利用が要求される。MU の持つ計算タスクの処理完了時間を短縮するためには、計算タスクの伝送時間と MEC サーバでの処理時間の両方、またはいずれかを削減する必要がある。また、複数のユーザが同じ MEC サーバに計算タスクをオフロードする場合、複数の MU 間で無線及び計算リソースを共有する必要がある。したがって、効率的なリソース割り当て法及び、スケジューリングアルゴリズムが重要となる。

従来研究では、車両ネットワーク [1] や仮想化ネットワーク [2]、セルラーネットワーク [3]–[8] において MEC に関わるリソース割り当ての問題が検討されている。しかしこれら多くの研究では、無線リソースまたは計算リソースのいずれかのタイプのリソース割り当てのみに着目している。例えば [7] ではユーザのタスクの完了時間を最小限に抑えることを目的として、無線チャンネルに対する同時最適化問題を定式化し、それを解決するための効果的なアルゴリズムが提案されている。[8] では総エネルギー消費を最小化することを目的として、エネルギー効率の良いリソース割当方式が検討されている。これらの研究は主に実行時間の最小化またはエネルギー消費の最小化を目的としているが、各タスクの MEC サーバで処理後の計

算結果のサイズは無視できると仮定し、計算結果をモバイルユーザに返送する下りリンク (DL: Downlink) の返送時間または BS のリソース消費を考慮していない [1]–[5], [7]。しかしながら、AR などの計算結果のサイズは無視できないほど大きくなる可能性がある。したがってこれら従来研究で得られた解は、大規模な計算結果を有するアプリケーションには適さない可能性がある。また、多くの研究ではローカル処理が考慮されておらず、各 MU が持つ計算タスクは完全にオフロードされるものと仮定していた [3]–[5], [8]。全ての MU が計算タスクを全てオフロードすることによって無線リソース及び MEC サーバにおける計算リソース割り当て条件が厳しくなる。MU 及び MEC サーバの処理能力を考慮して考えることによって、リソースの割り当てを UE にも割り当てることが可能になる。

本稿では、複数の MU と MEC サーバを具備した 1 つの BS から構成されるマルチユーザ MEC システムを検討し、計算タスクの処理完了時間を最小化する問題を考える。従来研究では、スケジューリングを考えない、タスク処理のみにかかる処理時間の最小化 [7] や、ローカル計算を考慮しない処理時間の最小化が取り組まれてきたが、本研究では従来研究とは異なり、すべてのユーザのタスク完了時間を最小化することを目的としている。MEC サーバで処理後の計算タスクの計算結果は無視できないものとし、処理完了時間は計算結果の DL 伝送時間も含めるものとする。[7] と同様に各 MU は 1 つの計算タスクを有しており、各計算タスクはローカルで計算されるタスクとオフロードされるタスクに分割することができるものとする。これにより 1 つのタスクを UE、MEC 両方での処理が可能となる。無線チャンネルでの無線リソース割り当て、MEC サーバでの計算リソース割り当てには、それぞれ時分割多元接続 (TDMA: Time Division Multiple Access) 方式を採用することにより、より効率的な無線リソース、計算リソースの割り当てアルゴリズムを提案する。具体的には、二分探索法を用いて各ユーザのローカルでの処理時間、アップロードの時間、MEC での処理時間、返送の時間を決定する。取得したアップロード時間、MEC での処理時間、返送の時間に対して、フローショップスケジューリング問題を適用し、ジョンソンアルゴリズムを用いることで各タスクの順序及び得られた時間での最小処理完了時刻を得る。得られた処理完了時刻を二分探索で用いる実現可能性条件に追加し最適な時間を各ユーザに割り当てる。

本稿の構成は以下のようになっている。まず第 2 章で、タスクモデル及び伝送・計算にかかる実行時間に関して記述し、本研究で用いるシステムモデルを示す。第 3 章では計算タスクの処理完了時間を最小化するための最適化問題を定式化する。第 4 章では、定式化された最適化問題に対し、二分探索法とジョンソンアルゴリズムを用いたアルゴリズムを提案する。第 5 章では、提案したアルゴリズムの性能を計算機シミュレーションにより調べる。最後に第 6 章で本論文のまとめを行う。

## 2. システムモデル

図 1 に示すように、本稿ではシステム内に  $U$  個の単一アン

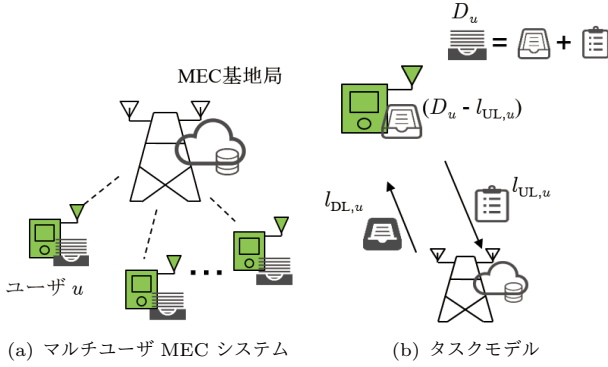


図1 システムモデル

テナ MUs と一つの単一アンテナ BS が存在するマルチユーザ MEC システムを考慮する。BS には一定の CPU サイクル周波数 [CPU cycle/s] を有する MEC サーバが配備されているものとする。集合  $\mathcal{U} = [1, 2, \dots, U]$  で与えられる MUs はそれぞれ計算能力を有しているものとする。各 MU は、厳しい遅延制約を持つ分割可能な計算タスクを持つものとし、リソース割り当ての結果に基づいて全てもしくは一部の計算タスクをローカルの CPU で処理する。リソース割り当てにより、計算タスクを MEC へとオフロードすることになった場合には、一部のタスクを上りリンク (UL: Uplink) で MEC サーバへ送信し、MEC サーバからの出力結果を DL で受信する。

### 2.1 タスクモデルと実行時間

MU  $u \in \mathcal{U}$  における計算タスクをタスク  $u$  とし、そのタスクサイズを  $D_u$  [bits], タスクの種類を  $A_u$  [CPU cycle/bit] とし記述する。ここで、タスクの種類とはタスク計算時 1 ビットあたりに必要な CPU サイクル数を示す。各タスクは MEC へオフロードされるオフロードタスクとローカル CPU によって処理されるローカルタスクに分割できるものとする。本稿では、ローカルタスクのサイズを  $(D_u - l_{UL,u})$  [bits], オフロードタスクのサイズを  $l_{UL,u}$  [bits] とする。更に、オフロードタスクの MEC における計算処理結果のサイズを  $l_{DL,u}$  [bits] とする。

(1) ローカルタスクの実行時間: ユーザ  $u$  のローカルタスク処理能力を  $F_{local,u}$  [CPU cycle/s] で表す。ユーザ  $u$  は  $(D_u - l_{UL,u})$  [bits] のタスクをローカルで処理する必要があるためローカルタスクの実行時間は次式で与えられる。

$$t_{local,u} = \frac{A_u(D_u - l_{UL,u})}{F_{local,u}}, \quad \forall u \in \mathcal{U} \quad (1)$$

(2) オフロードタスクの実行時間: ユーザ  $u$  の UL チャネル容量を  $C_{UL,u}$  [bit/s], DL チャネル容量を  $C_{DL,u}$  [bit/s] とする。また、MEC のタスク処理能力を  $F_{MEC}$  [CPU cycle/s], MEC サーバにて処理されることで得られる処理結果のサイズをオフロードサイズの関数  $f(l_{UL,u})$  [CPU cycle/s] として表記する。ユーザ  $u$  に割り当てられる UL の割合と DL の割合をそれぞれ、 $0 \leq x_u \leq 1$  と  $0 \leq y_u \leq 1$  とする。この時、オフロードしたタスクの出力結果をユーザが受信するまでにかかる時間 (タスク  $u$  の処理完了時間) は次式で与えられる。

$$t_{offload,u} = \frac{l_{UL,u}}{x_u C_{UL,u}} + \frac{A_u l_{UL,u}}{F_{MEC}} + \frac{f(l_{UL,u})}{y_u C_{DL,u}}, \quad \forall u \in \mathcal{U} \quad (2)$$

第 1 項はオフロードタスクを MEC へと送信するのに必要な時間、第 2 項はオフロードされたタスクを MEC サーバにて処理を実行するのに必要な時間、第 3 項は MEC サーバで処理されたタスクを UE に返送するために必要な時間である。

### 2.2 スケジューリング [8]

ユーザ  $u$  のオフロードタスクアップロード開始時刻と終了時刻を  $s_{UL,u}$  と  $c_{UL,u}$ , オフロードタスクの MEC サーバでの計算処理開始時刻と終了時刻を  $s_{PR,u}$  と  $c_{PR,u}$ , ユーザ  $u$  の計算結果返送開始時刻と終了時刻を  $s_{DL,u}$  と  $c_{DL,u}$  とする。

(1) オフロードモデルの完了時刻制約: オフロードタスクの各処理における完了時刻制約は次のように表される。

$$\begin{cases} s_{UL,u} + \frac{l_{UL,u}}{x_u C_{UL,u}} = c_{UL,u} \\ s_{PR,u} + \frac{A_u l_{UL,u}}{F_{MEC}} = c_{PR,u} \\ s_{DL,u} + \frac{f(l_{UL,u})}{y_u C_{DL,u}} = c_{DL,u} \end{cases} \quad \forall u \in \mathcal{U} \quad (3)$$

(2) ユーザ毎の順列に関する制約条件: ユーザ毎の計算タスクがアップロード、MEC サーバでの計算処理、計算結果の返送の順で行われるための制約条件は次のように表される。

$$\begin{cases} s_{UL,u} \geq 0 \\ s_{PR,u} \geq c_{UL,u} \\ s_{DL,u} \geq c_{PR,u} \end{cases} \quad \forall u \in \mathcal{U} \quad (4)$$

(3) 全ユーザの順列に関する制約条件: 全ユーザに対して、アップロード、MEC サーバでのタスクの計算処理、計算結果の返送が排他的に行われるための制約条件は次のように表される。

$$\begin{cases} s_{UL,[u]} \geq c_{UL,[u-1]} \\ s_{PR,[u]} \geq c_{PR,[u-1]} \\ s_{DL,[u]} \geq c_{DL,[u-1]} \end{cases} \quad \forall u \in \mathcal{U} \setminus \{1\} \quad (5)$$

## 3. 問題の定式化

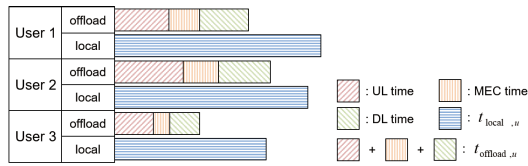
ユーザ  $u \in \mathcal{U}$  のタスク完了時刻  $T_{comp,u}$  はローカルタスク及びオフロードタスクの両方が完了した時刻として次のように定義される。

$$T_{comp,u} = \max\{t_{local,u}, c_{DL,u}\}, \quad \forall u \in \mathcal{U} \quad (6)$$

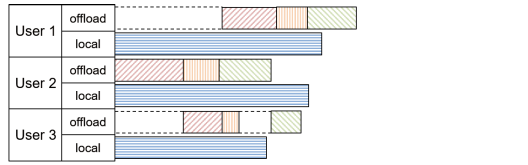
ここで、 $t_{local,u}$  はユーザ  $u$  のローカルタスクの完了時間、 $c_{DL,u}$  はオフロードタスクの完了時刻である。したがって、すべてのユーザの完了時間は以下のように与えられる。

$$T = \max_{1 \leq u \leq U} T_{comp,u} \quad (7)$$

本稿では、マルチユーザ MEC システムにおける全ユーザの処理完了時間最小化問題を考える。したがって、 $T$  の最小化を目的とする最適化問題は次式のように定式化できる。



(a) 時間割り当て



(b) 時刻割り当て

図 2 各ユーザのタスク処理時間及び時刻の割り当て

$$\begin{aligned}
 & \min_{\{l_{UL,u}\}, \{x_u\}, \{y_u\}, \mathbf{s} \in \mathcal{S}, \{s_{UL,u}\}, \{s_{PR,u}\}, \{s_{DL,u}\}} T & (P) \\
 & \text{subject to} & \\
 & t_{\text{local},u} \leq T, \quad \forall u \in \mathcal{U}, & (8) \\
 & t_{\text{offload},u} \leq T, \quad \forall u \in \mathcal{U}, & (9) \\
 & \frac{A_u l_{UL,u}}{F_{\text{MEC}}} \leq T, \quad \forall u \in \mathcal{U}, & (10) \\
 & \sum_{u=1}^U x_u = 1, \quad x_u \geq 0, & (11) \\
 & \sum_{u=1}^U y_u = 1, \quad y_u \geq 0, & (12) \\
 & 0 \leq l_{UL,u} \leq D_u, \quad \forall u \in \mathcal{U}, & (13) \\
 & (3), (4), (5)
 \end{aligned}$$

ここで、 $\mathcal{S}$  は全ユーザ  $U$  における実現可能なタスクスケジューリングの集合を示す。式 (8) はローカルタスクの実行時間に関する制約条件、式 (9) はオフロードタスクの実行時間に関する制約を表わす。さらに、式 (10) は MEC 処理時間が総処理完了時間を超えない制約条件である。式 (11)、式 (12) はそれぞれ UL、DL のチャンネル使用率に関する制約条件である。最後に、式 (13) はオフロードタスクのサイズに関する制約条件を表わしている。この最適化問題 (P) は、TDMA 方式による時間の割り当てと、各フェーズにおける処理の開始、終了の時刻の割り当て及びスケジューリングを行う。しかし、時間及び時刻の概念が混在しているため、このままでは最適化問題を解くことができない。したがって、この最適化問題を解くために問題を二つの問題 (P1)、(P2) に分割する。

### 3.1 時間割り当て

図 2(a) に示すように、ユーザ  $u$  のタスクの完了時間は時間だけに注目すると、次式のように表すことができる。

$$T_{\text{comp},u}^{(P1)} = \max\{t_{\text{local},u}, t_{\text{offload},u}\}, \quad \forall u \in \mathcal{U} \quad (14)$$

ここで、 $t_{\text{local},u}$  はユーザ  $u$  のローカルで処理にかかる時間、 $t_{\text{offload},u}$  はユーザ  $u$  のオフロードしたタスクの実行時間である。したがって、時間だけに注目した場合、全てのユーザの処理が完了する時間は次式で表せる。

$$T^{(P1)} = \max_{1 \leq u \leq U} T_{\text{comp},u}^{(P1)} \quad (15)$$

全てのユーザの完了時間が最も短くなるようなタスクサイズ及び時間を割り当てることによって、問題 (P) の時間に対する最適リソースを得ることができる。したがって、時間に関する最適化問題 (P1) を得ることができ、この問題を解くことにより、時間に関する最適リソースを得る。

$$\begin{aligned}
 & \min_{\{l_{UL,u}\}, \{x_u\}, \{y_u\}} T^{(P1)} & (P1) \\
 & \text{subject to} (8), (9), (10), (11), (12), (13)
 \end{aligned}$$

### 3.2 時刻の割り当て (スケジューリング)

次に時刻に注目する。本研究では TDMA 方式を使用するので、各ユーザは同じ時刻でのタスクのアップロード、MEC での処理、タスクのダウンロードを行うことができない。したがって、スケジューリングを行うことによって時刻を決定し、問題 (P) の時刻に対する最適値を得る必要がある。割り当てる時間が決定し、図 2(b) に示すようにスケジューリングを行うと最後に完了するタスクの終了時刻が得られる。最後に完了するタスクの終了時刻が最小になるように開始時間の割り当て及び並び替えを行うことで、問題 (P) の最適値を得る。したがって、時刻に関する最適化問題は (P2) のようになる。

$$\begin{aligned}
 & \min_{\mathbf{s} \in \mathcal{S}, \{s_{UL,u}\}, \{s_{PR,u}\}, \{s_{DL,u}\}} c_{DL,[u]} & (P2) \\
 & \text{subject to} (3), (4), (5)
 \end{aligned}$$

すなわち、問題 (P1) は時間の割り当て、問題 (P2) では時刻の割り当てに関する問題である。この二つの問題をそれぞれ解くことによって最適解を得る。

## 4. 提案アルゴリズム

### 4.1 二分探索法 (Bisection Search) [7]

問題 (P1) を解くため、 $T^{(P1)}$  に関する最適値探索を二分探索により行う。この時、制約 (8)-(13) の実現可能性問題を解く必要がある。ここでは、文献 [7] で提案されている二分探索による方法を取り入れる。文献 [7] では、UL と MEC での処理時間のみが考慮されているが、本稿ではそれに加えて DL にかかる時間も考慮する。

(1) オフロードタスク  $l_u$  の決定法: 制約 (8) より、次式が成り立つ。

$$D_u - \frac{TF_{\text{local}}}{A_u} \leq l_{UL,u} \quad (16)$$

これにより、ローカルタスク ( $D_u - l_{UL,u}$ ) が完了時間  $T^{(P1)}$  以内に完了するためには、 $l_{UL,u}$  が  $(D_u - \frac{TF_{\text{local}}}{A_u})$  以上になる必要がある。したがって、その場合のオフロードタスクの最小値  $l_{UL,u}^{\min}$  は次式で表せる。

$$l_{UL,u}^{\min} = \max\left\{0, D_u - \frac{TF_{\text{local}}}{A_u}\right\} \quad (17)$$

式 (17) より、最適化問題 (P1) は 0 が最小となるような  $x_u, y_u$  を求める最適化問題として、次のように書き換えることができる。

$$\begin{aligned} \min_{\{x_u\}, \{y_u\}} \quad & 0 \quad (\text{P1A}) \\ \text{subject to} \quad & \frac{l_{UL,u}^{\min}}{x_u C_{UL,u}} + \frac{A_u l_{UL,u}^{\min}}{F_{MEC}} + \frac{f(l_{UL,u}^{\min})}{y_u C_{DL,u}} \leq T, \\ & \forall u \in \mathcal{U} \quad (18) \end{aligned}$$

$$\frac{A_u l_{UL,u}^{\min}}{F_{MEC}} \leq T, \quad \forall u \in \mathcal{U}, \quad (19)$$

$$\sum_{u=1}^U x_u = 1, \quad x_u \geq 0, \quad (20)$$

$$\sum_{u=1}^U y_u = 1, \quad y_u \geq 0, \quad (21)$$

(2) 実現可能性条件:  $a_u = \frac{l_{UL,u}^{\min}}{C_{UL,u}}$ ,  $b_u = \frac{A_u l_{UL,u}^{\min}}{F_{MEC}}$ ,  $c_u = \frac{f(l_{UL,u}^{\min})}{C_{DL,u}}$  とすると, 実現可能性問題 (P1A) に必要かつ十分な条件は次式で表せる.

$$b_u \leq T, \quad \forall u \in \mathcal{U} \quad (22)$$

$$\left( \sum_{u=1}^U \frac{\sqrt{a_u c_u}}{T - b_u} \right)^2 \leq \left( 1 - \sum_{u=1}^U \frac{a_u}{T - b_u} \right) \left( 1 - \sum_{u=1}^U \frac{c_u}{T - b_u} \right) \quad (23)$$

$$\sum_{u=1}^U \frac{a_u}{T - b_u} \leq 1, \quad \sum_{u=1}^U \frac{c_u}{T - b_u} \leq 1 \quad (24)$$

#### 4.2 ジョソソナルゴリズム (Johnson's Algorithm)

二分探索法によって割り当てられた時間に対し, (P2) を解くことによって時刻を割り当てる. (P2) は複数のタスクを複数機械で加工するとき, どの順序で行えば全体の処理時間を最小にできるかというジョブスケジューリング問題とみなすことができる. ジョブスケジューリングには, フローショップ問題, ジョブショップ問題, オープンショップ問題とタスク処理の順序により3つに分類することが可能である. 本研究ではアップロード, MECサーバでの処理, 計算結果の返送を3つの操作とみなし, (P2) は3つの処理を行う3-ステージフローショップスケジューリング問題とみなすことができる [8]. 3-ステージフローショップスケジューリング問題はすべてのタスクを同じ処理順序 (アップロード, MECサーバでの処理, 計算結果の返送) で3つの機械で処理を行う. (P2) の最適解を  $T^{(P2)}$  とすると, 問題 (P2) は次式のように書き換えられる.

$$T^{(P3)} = \min_{s \in \mathcal{S}} T^{(P2)} \quad (P3)$$

ここで,  $T^{(P2)}$  は次式で与えられ, (3), (4), (5) の制約条件の下で  $s_{UL,u}$ ,  $s_{PR,u}$ ,  $s_{DL,u}$  を最適化することによって得られる [8].

$$\begin{aligned} T^{(P2)} = \max_{1 \leq i \leq j \leq U} \left\{ \left( \sum_{u=1}^j \frac{A_{[u]} l_{UL,[u]}}{F_{MEC}} - \sum_{u=1}^{j-1} \frac{f(l_{UL,[u]})}{y_{[u]} C_{DL,[u]}} \right) \right. \\ \left. + \left( \sum_{u=1}^i \frac{l_{UL,[u]}}{x_{[u]} C_{UL,[u]}} - \sum_{u=1}^{i-1} \frac{A_{[u]} l_{UL,[u]}}{F_{MEC}} \right) \right\} + \sum_{u=1}^U \frac{f(l_{UL,[u]})}{y_{[u]} C_{DL,[u]}} \quad (25) \end{aligned}$$

(P3) は3-ステージフローショップ問題であるので, ある条件を満たすときジョソソナルゴリズムにより最適解を求めることができる. (P3) に対する最適解は (P2) に対しても最適解

なので, (P3) をジョソソナルゴリズムで最適解を求める [8]. 本研究における3つの別々の機械 (アップロード, MEC処理, ダウンロード) を用いた3-ステージフローショップスケジューリングは以下の条件のいずれかが満たされた場合, ジョソソナルゴリズムを用いることで最適解を得ることができる.

$$\max_{u \in \mathcal{U}} \left( \frac{A_u l_{UL,u}}{F_{MEC}} \right) \leq \min_{u \in \mathcal{U}} \left( \frac{l_{UL,u}}{x_u C_{UL,u}} \right) \quad (26)$$

$$\max_{u \in \mathcal{U}} \left( \frac{A_u l_{UL,u}}{F_{MEC}} \right) \leq \min_{u \in \mathcal{U}} \left( \frac{f(l_{UL,u})}{y_u C_{DL,u}} \right) \quad (27)$$

上記の条件が満たされる時, 得られた解は最適解である. MECサーバは強力な計算能力を持つため, 各タスクの処理時間は通常小さくなり, 上記の条件 (26),(27) は通常満たされるので最適解となる. しかし, 上記の条件が満たされない場合は最適な性能に近い準最適解となる. ジョソソナルゴリズムではまず各機械でのタスク処理にかかる時間のリストを作成する. 3機械の場合は, 1機械目+2機械目, 2機械目+3機械目という仮想機械を想定し, 2機械でのジョソソナルゴリズムを適用する. 作成された順序づけされていないタスクリストの中から, 処理時間の最小の処理を見つけ, 見つけた最小処理時間の処理が仮想機械における前処理 (1機械+2機械) にかかる時間であれば, そのタスクをはじめに処理する. 一方で仮想機械における後処理 (2機械+3機械) にかかる時間であれば, そのタスクの処理を最後に行う. タスクの処理順序が決定したらそのタスクをタスクリストから削除する. この操作をすべてのタスクの順序が決定するまで繰り返す.

#### 4.3 提案アルゴリズムと最適解

本研究では, 元の最適化問題 (P) をサブ問題 (P1) と (P2) に分割し, (P1) では時間割り当てを, (P2) ではスケジューリングによる時刻割り当てをそれぞれ行い, それぞれの最適解を得た. 具体的には, 二分探索法を用いて各ユーザのローカルでの処理時間, アップロードの時間, MECでの処理時間, 返送の時間を決定する. 取得したアップロード時間, MECでの処理時間, 返送の時間に対して, フローショップスケジューリング問題を適用し, ジョソソナルゴリズムを用いることで各タスクの順序及び得られた時間での最小処理完了時刻を得る. この得られた処理完了時刻を二分探索で用いる実現可能性条件に追加し用いることによって更なる時間を取得し, 最適な時間を各ユーザに割り当てる. このように二分探索法とジョソソナルゴリズムを交互に繰り返すことによって, 全体的な処理完了時刻を最小にする. このように, 本研究で提案する全体的な最小処理完了時間を達成するためのアルゴリズムは Algorithm1 によって与えられる.

#### 5. 数値結果

本研究では, トイモデルを用いてシミュレーションを行う. 今回用いたシミュレーション諸元を表1に示す.

図3はユーザ数10の場合の各手法における総処理時間を示す. ALL\_LOCAL及びALL\_MECはそれぞれローカルCPU, MECサーバのみでタスク処理を行う手法, ALL\_MEC\_jhonsomはジョソソナルゴリズムを用いてすべてMECサーバで処理する手法, ONLY\_Bisectionはローカル



### Algorithm 1 提案アルゴリズム

```

1: Initialize:  $T_{low} = 0, T_{high} = \min_{1 \leq u \leq U} \frac{A_u D_u}{F_{local}}$ , set  $\epsilon$ 
2: while  $T_{high} - T_{low} \leq \epsilon$ , アルゴリズム終了 do
3:    $T = \frac{T_{high} + T_{low}}{2}$  と設定し,  $l_{UL,u}^{\min}, a_u, b_u, c_u$  を計算
4:   if 実現可能性条件 (22) が実現可能 then
5:     ジョンソンアルゴリズムの実行
6:   else 実現不可能
7:      $T_{low} = T$  と設定し, step2 へ行く
8:   end if
9:   if ジョンソンアルゴリズム実行後,  $c_{DL,[U]} < T$  及び, 実現可能性条件 (24) が実現可能 then
10:     $T_{high} = T$  と設定し, step2 へ行く
11:   else 実現不可能
12:     $T_{low} = T$  と設定し, step2 へ行く
13:   end if
14: end while

```

表 1 シミュレーション諸元

UE 数 $U$	10, 20, 30, 40, 50, 100
基地局数	1
ローカル CPU 周波数	$100 \leq F_{local} \leq 1000$ 100 間隔 [CPU cycle/s]
MEC CPU 周波数	$F_{MEC} = 10 \times 10^3$ [CPU cycle/s]
データサイズ	$1 \leq D_u \leq 10$ 1 間隔 [bits]
タスクタイプ	$5 \leq D_u \leq 15$ 1 間隔 [CPU cycle/bit]
UL 容量	$C_{UL,u} \in [10, 50]$ [bit/s]
DL 容量	$C_{DL,u} \in [50, 100]$ [bit/s]

での計算, MEC サーバでの計算両方を考慮したとき二分探索法のみを用いてリソース割り当てを行う既存手法である。図より, ローカルですべてのタスクを処理するのに比べ, MEC にタスクをオフロードすることによって大幅に処理時間を削減できている。さらに, 従来の二分探索のみを行うリソース割り当て法に比べ, 提案手法は最大で 37% 程度の総処理時間の低減が実現している。これはジョンソンアルゴリズムによって, 時刻の割り当てを考慮したリソース割り当てが可能になっているからである。図 4 はユーザ数を変化させたときの各手法における総処理時間を示している。図より, ユーザ数を増やすことによって, 手法における総処理時間に大きな差が出ていくことがわかる。ユーザ数が増えるにつれ, 提案方式は既存方式よりも最適にリソース割り当てを行うことができている。

### 6. おわりに

本研究では, 全ユーザの計算タスクの処理完了時間を最小化するためにマルチユーザ MEC システムにおける無線リソース及び計算リソース割り当て法を検討した。定式化した時間及び時刻の混在する最適化問題に対し, 二分探索法とジョンソンアルゴリズムを用いた無線・計算リソース割り当て法を提案した。シミュレーション結果より, 提案手法は既存方式と比較して最大で 37% 程度処理時間を低減可能なことを示した。

### 文 献

- [1] K. Zhang, Y. Mao, S. Leng, A. Vinel and Y. Zhang, "Delay constrained offloading for Mobile Edge Computing in cloud-enabled vehicular networks", in *Proc. 11th International Workshop on Communication Technologies for Vehicles*, Sweden, Sep. 2016
- [2] Y. Sun, S. Zhou, S. Zhou, and J. Xu, "EMM: Energy-Aware

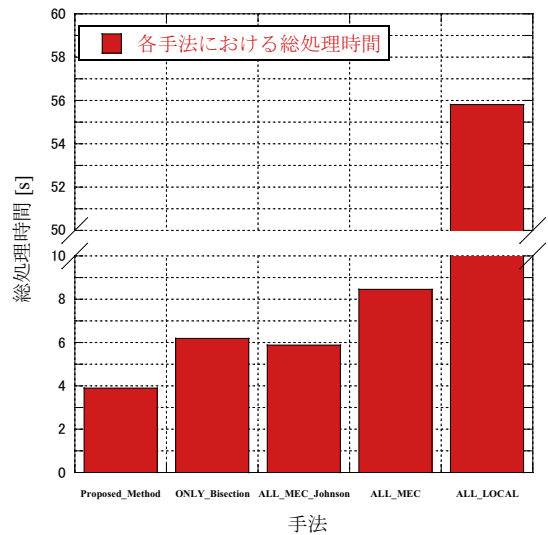


図 3 各手法における総処理時間

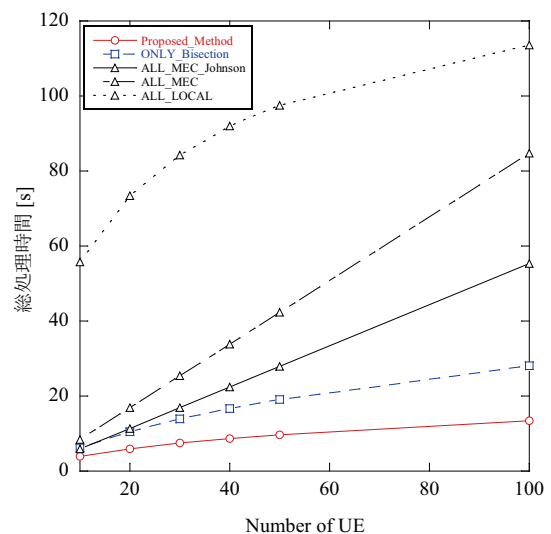


図 4 UE 数を変化させたときの総処理時間

- Mobility Management for Mobile Edge Computing in Ultra Dense Networks", *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2637–2646, 2017
- [3] M. Li, F. R. Yu, P. Si, H. Yao, E. Sun and Y. Zhang, "Energy-efficient M2M communications with mobile edge computing in virtualized cellular networks", in *Proc. IEEE ICC. Conf.*, France, May. 2017
  - [4] X. Lin, H. Zhang, H. Ji, and V. C. M. Leung, "Joint computation and communication resource allocation in mobile-edge cloud computing networks", in *Proc. IEEE IC-NIDC. Conf.*, China, Sep. 2016
  - [5] Y. Mao, J. Zhang, and K. B. Letaief, "Joint Task Offloading Scheduling and Transmit Power Allocation for Mobile-Edge Computing Systems", in *Proc. IEEE WCNC. Conf.*, pp. 1-6, San Francisco, CA, USA, Mar. 2017
  - [6] M. -H. Chen, M. Dong and B. Liang, "Joint offloading decision and resource allocation for mobile cloud with computing access point", in *Proc. IEEE ICASSP. Conf.*, Shanghai, China, Mar. 2016
  - [7] H. Q. Le, H. Al-Shatri, and A. Klein, "Efficient Resource Allocation in Mobile-edge Computation Offloading: Completion Time Minimization", in *Proc. IEEE ISIT. Conf.*, pp. 2513-2517, Aachen, Germany, Jun. 2017
  - [8] J. Guo, Z. Song, Y. Cui, Z. Liu, and Y. Ji, "Energy-Efficient Resource Allocation for Multi-User Mobile Edge Computing", in *Proc. IEEE Global Commun. Conf.*, pp. 1-7, Singapore, Singapore, Dec. 2017