

異種セルラーネットワークにおける無線および計算資源割り当て法

齋藤 吏玖[†] 安達 宏一[†]

[†] 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: †{saito,adachi}@awcc.uec.ac.jp

あらまし 近年、スマートフォンやタブレット端末などの急速な普及拡大および性能の向上に伴い、高度な計算処理を要求するモバイルアプリケーションが増えている。モバイルエッジコンピューティング (MEC: Mobile Edge Computing) ではユーザに代わりこれらのアプリケーションを処理することで処理時間短縮やユーザ端末 (UE: User Equipment) のバッテリー消費を低減できる。一方で、複数端末が MEC サーバを利用する場合、無線資源や計算資源を適切にユーザへ割り当てる必要がある。また、分割可能なアプリケーションタスクは MEC サーバでの処理と端末での処理を並列に行うことで、どちらか一方のみで処理する場合と比較し短時間での処理が実現できる。本稿では、MEC サーバを具備した異種セルラーネットワークにおけるタスクスケジューリング法およびオフロード割合決定法を提案する。提案法では、タスクの返送時間が予測できない場合でもスケジューリングが可能である。スケジューリングされたタスクに対しオフロード割合を一定間隔で増加させることで効率的なオフロード割合を決定できる。計算機シミュレーションにより、提案スケジューリング法は従来法と比較し、タスクの平均処理時間を最大で 16% 向上させることができることを示す。加えて、提案オフロード割合決定法を適用することで、全てのタスクをオフロード処理する場合と比較し、最大で 79% 処理時間を短縮できることを示す。

キーワード 異種セルラーネットワーク, MEC, スケジューリング

Radio and Computational Resource Allocation in Heterogeneous Networks

Riku SAITO[†] and Koichi ADACHI[†]

[†] The University of Electro-Communications, Advanced Wireless & Communication Research Center,

1-5-1 Chofugaoka, Chofu, Tokyo, 182-8585 Japan

E-mail: †{saito,adachi}@awcc.uec.ac.jp

Abstract In recent years, with the rapid spread and improvement of smartphones and tablet terminals, the number of mobile applications that require advanced calculation processing is increasing. Mobile Edge Computing (MEC) can reduce processing time and battery consumption of user equipment (UE) by processing these applications on behalf of the user. On the other hand, when multiple UEs use the MEC server, it is necessary to appropriately allocate wireless resources and computational resources. Also, the application task that can be divided can be processed in a shorter time than when it is processed by only one of them by performing the processing on the MEC server and the terminal in parallel. This paper proposes a scheduling method and an offload ratio decision method for heterogeneous cellular networks. With the proposed method, scheduling is possible even if the task return time is unpredictable. An efficient offload rate can be determined by increasing the offload rate for scheduled tasks at regular intervals. Computer simulation shows that the proposed scheduling method can improve the average processing time of tasks by up to 16% compared to the conventional method. It is also shown that by applying the proposed offload ratio decision method, the processing time can be shortened by up to 79% compared to the case where all tasks are offloaded.

Key words HetNet, MEC, Scheduling

1. はじめに

近年、スマートフォンやタブレット端末などのユーザ機器 (UE: User Equipment) が急速に普及し、日常生活に欠かせないものとなっている。それに伴い、拡張現実や仮想現実、画像処理などの高度な計算処理を必要とするアプリケーションも増加している [1]。これらのアプリケーションの要求は UE のみで処理することが困難な場合がある。UE がこれらの重いアプリケーションタスクを処理できる場合でも、処理に長い時間がかかる可能性がある。さらに、タスク処理時間の増加は UE のバッテリー消費を加速させる。一般に高い計算処理能力を必要とするこれらのアプリケーションをサポートするには UE のソフトウェアとハードウェアのレベルを大幅にあげる必要がある [2]。しかし、UE にはサイズの制限が存在するためハードウェアレベルの変更による計算資源の獲得は困難である。そのため、ソフトウェアレベルを変更し、リモートの計算資源を使用してアプリケーションの処理をする方が効果的であり、それを実現するクラウドコンピューティングが注目されている。クラウドコンピューティングでは、膨大な計算資源をもつサーバがユーザに代わりタスクの処理を行う。サービスを利用するユーザはタスクをサーバへ送信 (オフロード) し、サーバにより処理された計算結果を受信する。これにより高度なアプリケーションタスクを処理できるとともに、UE のバッテリー消費を抑えることができる。特に基地局 (BS: Base Station) などのネットワークエッジにサーバを配置するモバイルエッジコンピューティング (MEC: Mobile Edge Computing) [3] では、ユーザとサーバの距離が近いこと、低遅延でのタスクの送受信が可能であり、オフロード処理の遅延も小さくなる [4]。しかしながら、複数ユーザが MEC サーバを効率的に利用するためには、タスクの送受信に必要となる周波数資源やタスクの処理に必要な計算資源の適切な配分が必要不可欠である。文献 [5]-[10] ではセルラーネットワークにおける MEC システムについて研究が行われている。これらの研究では計算処理後のタスクサイズが小さくなることを仮定しているため、タスクの返送にかかる時間を無視している。一方で、実際には返送サイズが無視できないほど大きくなる可能性があることが示されている [11]。また、多くの研究ではバイナリーオフローディングが採用されている。バイナリーオフローディングでは、ユーザはタスクを UE の持つ計算能力またはサーバでどちらかを選択し処理する。そのため、バイナリーオフローディングを採用しタスクをオフロード処理する場合、ユーザの計算資源は使われず無駄になってしまう。

そこで本稿では、タスク返送の考慮およびタスクを部分的にオフロード処理するパーシャルオフローディングを採用した場合におけるタスク処理時間の短縮を目的とし、効率的なタスクのスケジューリング法およびオフロードスキームを提案する。提案タスクスケジューリング法では、少ない情報でスケジューリングが可能であり、提案オフロードスキームでは UE 数の変化に伴いオフロード割合を変化することで柔軟に対応することができる。提案スケジューリング法では、既存手法と比較して

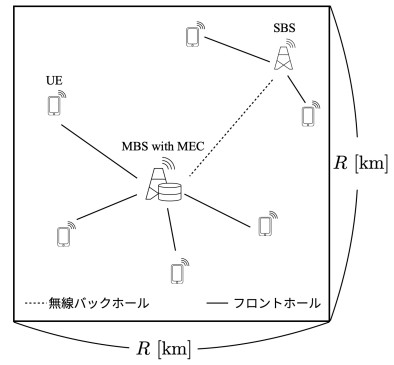


図1 システムモデル

タスクの平均処理時間を最大で 16% 短縮できることを示す。また、提案オフロードスキームでは全てのタスクをオフロード処理した場合と比較し全タスクの最大処理時間を最大で 79% 短縮できることを示す。

本稿は以下の構成になっている。第2節では本稿で想定したシミュレーションのモデルについて説明し、第3節では提案するスケジューリング法およびオフロードスキームを説明する。第4節では計算機シミュレーションの結果を説明し、第5節では本稿のまとめを行う。

2. システムモデル

2.1 シミュレーションモデル

本稿では、図1に示すように $R \times R$ [km²] のエリア内に U 台の UE がランダムに配置された環境を想定する。エリア中央に MEC サーバを具備したマクロ基地局 (MBS: Macro BS) を設置し、エリア内にランダムにスモール基地局 (SBS: Small BS) を配置する。本稿では UE の集合を $U = \{1, 2, \dots, U\}$ とし、 u 番目の UE を UE_u と定義する。各 UE は BS とのチャネル状態に基づき接続先 BS を選択する。ここで、SBS に接続する UE を SUE, MBS に接続する UE を MUE と定義する。SUE の集合は $S = \{1, 2, \dots, S\}$, MUE の集合は $M = \{1, 2, \dots, M\}$ で表記される。これらは $S \cup M = U$ および $S \cap M = \emptyset$ を満たす。 UE_u は分割可能な D_u [bits] のタスクを生成し、オフロード割合 $\alpha_u \in [0, 1]$ により一部をオフロードタスク、残りをローカルタスクに分割する。ローカルタスクは UE のローカル中央処理装置 (CPU: Central Processing Unit) で処理され、オフロードタスクは BS へオフロードされ MEC サーバで処理された後に UE へ返送される。

2.2 タスク処理モデル

図2に示すように、 UE_u はタスクサイズ D_u [bits], タスクタイプ A_u [cycle/bit] のタスクを保持しているものとする。タスクタイプは 1 ビットを処理するのに必要な CPU サイクル数である。各タスクはオフロード割合 α_u によりローカルタスクとオフロードタスクに分割される。そのため、 UE_u のオフロードタスクサイズは $\alpha_u D_u$ [bits], ローカルタスクサイズは $(1 - \alpha_u) D_u$ [bits] となる。各処理にかかる時間は以下のように定式化される [12]。

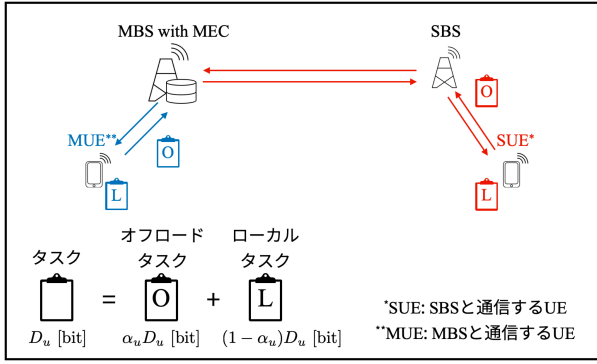


図2 タスク処理モデル

2.2.1 ローカル処理時間

UE_u はローカルタスクを自身の CPU で処理するため、UE の計算処理能力を f_{local} [cycle/sec] とすると、ローカル処理時間 $t_{\text{local},u}$ [sec] は次式で与えられる。

$$t_{\text{local},u}(\alpha_u) = \frac{(1 - \alpha_u) D_u A_u}{f_{\text{local},u}}, \forall u \in \mathcal{U}. \quad (1)$$

2.2.2 オフロード処理時間

MUE と SUE ではオフロード処理の行程が異なる。MUE のオフロードタスクはアップリンク (UL: UpLink) 通信により MBS へ送信され、MBS に配置された MEC サーバにより処理される。処理されたタスクの計算結果はダウンリンク (DL: DownLink) 通信で MUE へ返送される。そのため、MUE のオフロード処理における処理時間 $t_{\text{offload},m}$ [sec] は UL 通信時間、MEC サーバによる計算処理時間、DL 通信時間の 3 つから成り、次式で与えられる。

$$t_{\text{offload},m}(\alpha_m) = \frac{\alpha_m D_m}{r_{\text{UtM},m}} + \frac{\alpha_m D_m A_m}{f_{\text{MEC}}} + \frac{\alpha_m \beta_m D_m}{r_{\text{MtU},m}}, \forall m \in \mathcal{M} \quad (2)$$

ここで、 $r_{\text{UtM},m}$ [bit/sec] および $r_{\text{MtU},m}$ [bit/sec] は MUE_m と MBS 間の UL および DL チャネル容量である。また、 f_{MEC} [cycle/sec] は MEC サーバの CPU サイクル数であり、 β_m は返送後と返送前のタスクサイズの比である。

SUE のオフロードタスクは UL 通信により SBS を経由して MBS へ送信され、MBS に配置された MEC サーバにより処理される。処理されたタスクの計算結果は SBS を経由して SUE へ返送される。そのため、SUE のオフロード処理における処理時間 $t_{\text{offload},s}$ [sec] は次式で与えられる。

$$t_{\text{offload},s}(\alpha_s) = \frac{\alpha_s D_s}{r_{\text{UtS},s}} + \frac{\alpha_s D_s}{r_{\text{StM},s}} + \frac{\alpha_s D_s A_s}{f_{\text{MEC}}} + \frac{\alpha_s \beta_s D_s}{r_{\text{MtS},s}} + \frac{\alpha_s \beta_s D_s}{r_{\text{StU},s}}, \forall s \in \mathcal{S} \quad (3)$$

ここで、 $r_{\text{UtS},s}$ [bit/sec] および $r_{\text{StU},s}$ [bit/sec] は SUE と SBS 間の UL および DL チャネル容量である。また、 $r_{\text{StM},s}$ [bit/sec] と $r_{\text{MtS},s}$ [bit/sec] は SBS と MBS 間のチャネル容量である。式 (3) の第 1 項から第 5 項はそれぞれ SUE から SBS への UL 通信時間、SBS から MBS への UL 通信時間、MEC サーバの計算処理時間、MBS から SBS への DL 通信時間、SBS から SUE への DL 通信時間を表す。

2.2.3 タスクの処理完了時間

本稿では、ローカル処理およびオフロード処理の両方が完了

		UE ₁		UE ₂			
		ローカル	オフロード	ローカル	オフロード		
UE ₁	ローカル	○	○	○	○		
	オフロード	○	○	○	○		
UE ₂	ローカル	○	○	○	○		
	オフロード	○	○	○	○		

		MUE			SUE						
		UL	MEC	DL	UL1	UL2	MEC	DL1	DL2		
MUEの オフロード 処理	UL	×	○	○	○	×	○	○	○		
	MEC	○	×	○	○	○	×	○	○		
	DL	○	○	×	○	○	○	×	○		
SUEの オフロード 処理	UL1	○	○	○	×	○	○	○	○		
	UL2	×	○	○	○	×	○	○	○		
	MEC	○	×	○	○	○	×	○	○		
	DL1	○	○	×	○	○	○	×	○		
	DL2	○	○	○	○	○	○	○	×		

○	並列に処理可能
△	一部並列に処理可能
×	並列に処理不可能

図3 UE 間のタスク処理制約

した時間をタスクの処理完了時間 $t_{\text{comp},u}$ [sec] とし、次式で定義する。

$$t_{\text{comp},u}(\alpha_u) \triangleq \max(t_{\text{local},u}(\alpha_u), t_{\text{offload},u}(\alpha_u)) \quad (4)$$

ここで $\max(X, Y)$ は X と Y の最大値を返す関数である。

2.3 スケジューリング

本シナリオにおけるオフロード処理はフローショップのスケジューリング問題に置き換えることができる。フローショップ問題は複数のジョブに対して複数のプロセスが実行される加工順序問題の一つであり、すべてのジョブを同じ順序で処理する必要がある場合のスケジューリング問題である。タスクのスケジューリングは、MUE のみが存在する環境では 3 段階、SUE のみが存在する環境では 5 段階のフローショップ問題に置き換えることができる。これら 2 種類の UE が存在する場合、スケジューリング問題は 3 行程および 5 行程の複合フローショップ問題となる。本稿では、UE から MBS への UL 通信および SBS から MBS への UL 通信、MBS から UE への DL 通信および MBS から SBS への DL 通信は同時にできないものと仮定する。また、タスクの各処理行程は時分割方式で行われるものとする。図 3 に時分割方式を採用する場合の各 UE におけるタスクの処理制約を示す。UE 間のオフロード処理以外は全て並列に処理することができ、UE の同じ行程および前述した仮定のみ排他的に処理される必要がある。また、本稿における各スケジューリング手法では、全てのタスクをオフロード処理する場合における処理時間でスケジューリングが行われる。

2.3.1 MUE のスケジューリング

UL 通信、MEC での処理、および DL 通信の開始時刻をそれぞれ、 $T_{b,\text{UtM},m}$ 、 $T_{b,\text{MEC},m}$ 、 $T_{b,\text{MtU},m}$ とし、それぞれの終了時刻を $T_{c,\text{UtM},m}$ 、 $T_{c,\text{MEC},m}$ 、 $T_{c,\text{MtU},m}$ とする。このとき、各行程の開始時刻と終了時刻の関係は次式で与えられる。

$$\begin{cases} T_{c,\text{UtM},m} &= T_{b,\text{UtM},m} + \frac{D_m}{r_{\text{UtM},m}} \\ T_{c,\text{MEC},m} &= T_{b,\text{MEC},m} + \frac{D_m A_m}{f_{\text{MEC}}} \\ T_{c,\text{MtU},m} &= T_{b,\text{MtU},m} + \frac{\beta_m D_m}{r_{\text{MtU},m}} \end{cases}, \forall m \in \mathcal{M}. \quad (5)$$

また、MUE のオフロードタスク処理は UL 通信、MEC サーバでの処理、DL 通信の順に行われる必要がある。そのため、各行程の開始時刻および終了時刻の制約は次式で与えられる。

$$\begin{cases} T_{b,UtM,m} \geq 0 \\ T_{b,MEC,m} \geq T_{c,UtM,m}, \forall m \in \mathcal{M}. \\ T_{b,MtU,m} \geq T_{c,MEC,m} \end{cases} \quad (6)$$

2.3.2 SUE のスケジューリング

MUE と同様に、SUE から SBS への UL 通信、SBS から MBS への UL 通信、MEC サーバでの処理、MBS から SBS への DL 通信および SBS から SUE への DL 通信の開始時刻をそれぞれ $T_{b,UtS,s}$, $T_{b,StM,s}$, $T_{b,MEC,s}$, $T_{b,MtS,s}$, $T_{b,StU,s}$ とし、それぞれの終了時刻を $T_{c,UtS,s}$, $T_{c,StM,s}$, $T_{c,MEC,s}$, $T_{c,MtS,s}$, $T_{c,StU,s}$ とする。このとき、各行程の開始時刻および終了時刻に関する制約は次式で与えられる。

$$\begin{cases} T_{c,UtS,s} = T_{b,UtS,s} + \frac{D_s}{r_{UtS,s}} \\ T_{c,StM,s} = T_{b,StM,s} + \frac{D_s}{r_{StM,s}} \\ T_{c,MEC,s} = T_{b,MEC,s} + \frac{D_s A_s}{f_{MEC}} \\ T_{c,MtS,s} = T_{b,MtS,s} + \frac{\beta_s D_s}{r_{MtS,s}} \\ T_{c,StU,s} = T_{b,StU,s} + \frac{\beta_s D_s}{r_{StU,s}} \end{cases}, \forall s \in \mathcal{S}. \quad (7)$$

また、SUE のオフロードタスク処理は SUE から SBS への UL 通信、SBS から MBS への UL 通信、MEC サーバでの処理、MBS から SBS への DL 通信、SBS から SUE への DL 通信の順に行われる必要がある。そのため、各行程の開始時刻および終了時刻の制約は次式で与えられる。

$$\begin{cases} T_{b,UtS,s} \geq 0 \\ T_{b,StM,s} \geq T_{c,UtS,s} \\ T_{b,MEC,s} \geq T_{c,StM,s}, \forall s \in \mathcal{S}. \\ T_{b,MtS,s} \geq T_{c,MEC,s} \\ T_{b,StU,s} \geq T_{c,MtS,s} \end{cases} \quad (8)$$

3. 提案法

本提案法では、まずタスクスケジューリングを行い、その後スケジューリングされたタスクに対してオフロード割合を決定する。それにより、複雑なフローショップ問題に対して、繰り返し処理などを要さない効率的なスケジューリングが可能となる。提案オフロードスキームではスケジュールされたタスクを処理順序の昇順にオフロード割合を増加させ、効率的なオフロード割合を探索することでタスクスケジューリングのみを適用する場合と比較してタスク処理時間を短縮することができる。

3.1 タスクスケジューリングアルゴリズム

提案スケジューリングは 2 つのステップで構成される。図 4 は 2 台の MUE および 2 台の SUE に対して提案タスクスケジューリングを適用した例を示す。最初のステップでは、MUE および SUE をそれぞれ UL 通信時間の昇順にソートしスケ

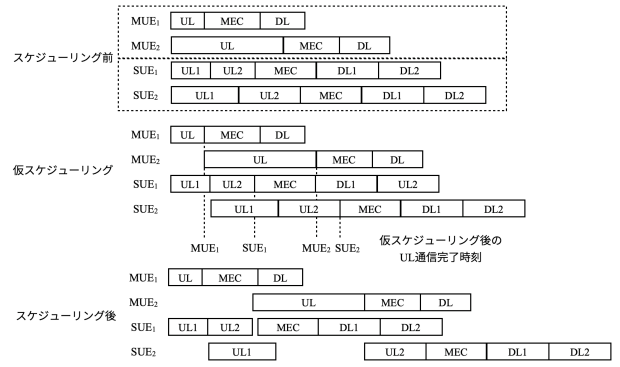


図 4 提案タスクスケジューリング例

ジューリングする。スケジューリング後の MUE の UL 通信が完了する時刻 $T_{UL,m}$ は次式で与えられる。

$$T_{UL,m} = \begin{cases} t_{UtM,m} & (m = 1) \\ T_{UL,m-1} + t_{UtM,m} & (\text{otherwise}) \end{cases} \quad (9)$$

ここで、 $t_{UtM,m}$ [sec] は MUE $_m$ の UL 通信時間である。同様に、SUE の UL 通信が完了する時刻 $T_{UL,s}$ は次式で与えられる。

$$T_{UL,s} = \begin{cases} t_{UtS,s} + t_{StM,s} & (s = 1) \\ \max((T_{UL,s-1} + t_{StM,s}), (T_{UL,s-1} - t_{StM,s} + t_{UtS,s} + t_{StM,s})) & (\text{otherwise}) \end{cases} \quad (10)$$

ここで、 $t_{UtS,s}$ [sec] および $t_{StM,s}$ [sec] は SUE から SBS への UL 通信時間および SBS から MBS への UL 通信時間である。次のステップでは、全 UE に対して最初のステップで得られた UL 通信完了時刻の昇順にソートしスケジューリングを行う。

3.2 オフロード割合決定アルゴリズム

図 5 は提案オフロードスキームのアルゴリズムを示す。提案アルゴリズムでは最初に全てのタスクのオフロード割合 α 、タスクインデックス i 、繰り返し回数 n の初期化が行われる。初期化後にタスクの最大処理時間の予測を行う。タスクの最大処理時間は全ての UE のタスク処理が完了した時間であり、次式で定義される。

$$t_{\text{totalmax},n} \triangleq \max \left(\max_{u \in \mathcal{U}} t_{\text{local},u}, T_{c,MtU,M}, T_{c,StU,S} \right) \quad (11)$$

その後、オフロード割合、タスクインデックス、繰り返し回数の更新を行い、再びタスクの最大処理時間の予測を行う。更新後のタスクの最大処理時間と更新前のタスクの最大処理時間を比較し、更新後の最大処理時間が更新前と比べて短縮される限りオフロード割合の更新を繰り返す。オフロード割合の更新によりタスクの最大処理時間が延長された場合、タスクインデックスの判定を行う。 $i \neq 1$ であれば i に 1 を代入し再び更新を繰り返し、 $i = 1$ であればオフロード割合の更新を終了しその時のオフロード割合を出力とする。

4. 数値結果

本稿では、伝搬モデルに文献 [13] のピコセルモデルを使用する。通信および計算処理パラメータは表 1 および表 2 に示す。

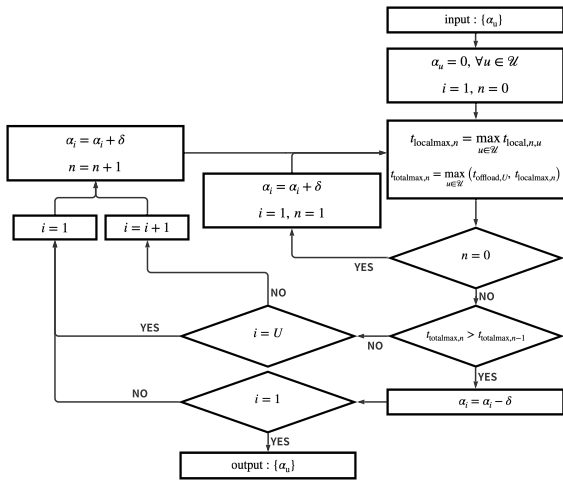


図5 提案オフロードスキームのアルゴリズム

表1 通信パラメータ

パラメータ	値
UE数 U	5 - 50
エリアサイズ $R \times R$	1×1 [km ²]
帯域幅 W	10 [MHz]
MBSの送信電力 P_{MBS}	20 [W]
SBSの送信電力 P_{SBS}	2 [W]
UEの送信電力 P_{UE}	200 [mW]
最大変調レート C_{max}	6 [bps/Hz]
雑音電力密度 N_0	3.98×10^{-18} [mW/Hz]
MBSの被干渉電力 I_{MBS}	3.98×10^{-10} [mW]
SBSの被干渉電力 I_{SBS}	3.98×10^{-11} [mW]
UEの被干渉電力 I_{UE}	3.16×10^{-11} [mW]

表2 計算処理パラメータ

パラメータ	値
ローカル CPU f_{local}	5×10^8 [CPUcycle/sec]
MEC CPU f_{MEC}	5×10^9 [CPUcycle/sec]
データサイズ D_u	[5, 15] [Mbits]
タスクタイプ A_u	100 [cycle/bit]
オフロード割合の増加幅 δ	0.01
返送サイズ比 β_u	1

4.1 比較手法

本稿では提案タスクスケジューリングの比較手法として全探索、ジョンソン法[14]、字引式順序法[15]、ランダム法の4つを考える。全探索は全通りの探索を実行するため、常に最適性が保証される一方で、計算量が $\mathcal{O}(U!)$ となるためUE数の増加に伴い膨大になる。また、スケジューリングの際に全行程にかかる時間の情報が必要である。ジョンソン法は2行程のフローショップ問題に対し最適性の保証がある。ジョンソン法を本シナリオに適用する場合、スケジューリングにはULおよびDL通信にかかる時間の情報が必要である。字引式順序法はフローショップ問題の近似解法の一つである。UL通信時間のみでスケジューリングが可能であり、最適性の保証はないが少ない計算量で比較的良い解を取得することができる。

提案オフロードスキームの比較手法は全タスクをローカル処

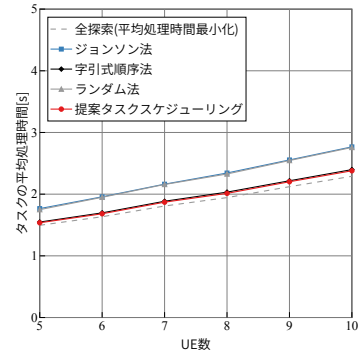


図6 各スケジューリング手法におけるタスクの平均処理時間

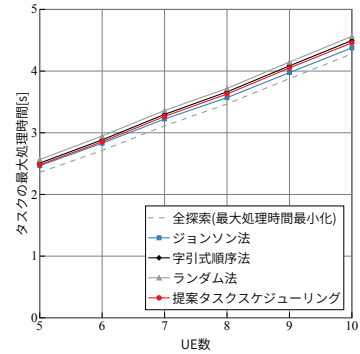


図7 各スケジューリング手法におけるタスクの最大処理時間

理する場合、全タスクをオフロード処理する場合、全タスクのオフロード割合を0.5に固定するする場合の3つを考える。

4.2 スケジューリング手法の比較

図6は各スケジューリング手法における平均処理時間の特性の比較である。提案法は全探索に次いでタスクの平均処理時間を短縮することができ、ジョンソン法およびランダム法と比較してタスクの平均処理時間を16%および15%短縮できることが分かる。また、全探索からの特性の劣化量を平均で3%まで抑えることができる。図7は各スケジューリング手法における最大処理時間の特性の比較である。提案法は全探索、ジョンソン法に次いでタスクの最大処理時間を短縮できている。全探索からの特性の劣化量は5%であり、ジョンソン法からの特性の劣化量は1%である。

4.3 オフロードスキームの比較

図8および9は各オフロードスキームにおけるタスクの平均処理時間および最大処理時間の特性の比較である。これらの図より、提案法は全ての比較手法の中で最も処理時間が短いことがわかる。また、各手法の平均処理時間および最大処理時間の振る舞いは似ている。表3に比較手法の提案法からの処理時間の最大増加量をまとめる。全タスクをローカル処理した場合はUE数の増加に伴う特性の変化はないが、提案法を含めた残りの3手法はUE数の増加に伴い処理時間が増加する。そのため、提案法と全タスクをローカル処理した場合の処理時間はUE数が少ない場合に特性差が大きくなる。また、オフロード割合を0.5に固定した場合および全タスクをオフロードした場合では、UE数の増加に伴いMECサーバの計算資源が枯渇する。その

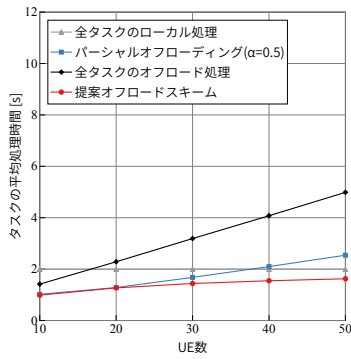


図 8 各オフロードスキームにおけるタスクの平均処理時間

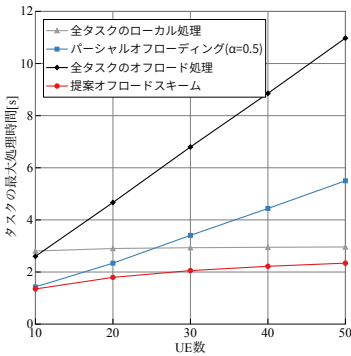


図 9 各オフロードスキームにおけるタスクの最大処理時間

表 3 処理時間の増加量 (提案法を 1 とした場合)

	提案法	全ローカル	$\alpha = 0.5$ 固定	全オフロード
平均処理時間	1	1.54	1.36	1.67
		($U = 10$)	($U = 50$)	($U = 50$)
最大処理時間	1	1.52	1.57	1.79
		($U = 10$)	($U = 50$)	($U = 50$)

ため、UE 数が増えるとローカル処理時間よりオフロード処理時間が大きくなり、タスクの処理時間が上限なく増加してしまう。一方で、提案法は UE 数の変化に伴いタスクのオフロード割合を柔軟に変更することができるため、処理時間が際限なく増加することはない。したがって、提案法とオフロード割合を 0.5 に固定した場合および全タスクをオフロードした場合の処理時間は UE 数が多い場合に特性差が大きくなる。

5. まとめ

本稿ではタスク処理時間短縮のためのタスクスケジューリングおよびオフロードスキームを提案した。提案タスクスケジューリングはタスクの UL 通信時間の情報のみでスケジューリングすることができ、提案オフロードスキームでは UE 数の変化に対しオフロード割合を変更することで柔軟に対応することができる。シミュレーション結果では、提案タスクスケジューリングがジョンソン法と比較しタスクの平均処理時間を 16% 短縮できることを示した。また、提案オフロードスキームは全てオフロード処理した場合と比較し、タスクの最大処理時間を最大で 79% 短縮できることを示した。

謝辞 This work was supported by the European Commission in the framework of the H2020-EUJ-02-2018 project 5GEnhance (Grant agreement no.815056) and the Ministry of Internal Affairs and Communications (MIC) of Japan (Grant no. JPJ000595).

文 献

- [1] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Commun. Surv. Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.
- [2] A. R. Khan, M. Othman, S. A. Madani and S. U. Khan, "A Survey of Mobile Cloud Computing Application Models," *IEEE Commun. Surv. Tuts.*, vol. 16, no. 1, pp. 393-413, First Quarter 2014.
- [3] "Mobile-Edge Computing-Introductory Technical White Paper", [online] Available: <https://www.etsi.org/component/rsfiles/preview?path=MEC+Presentations+at+Industry+Events%5C201601+-+RAN+USA+-+MEC+Intro.pdf>.
- [4] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Commun. Surv. Tuts.*, vol. 19, no. 4, pp. 2322-2358, Fourthquarter 2017.
- [5] 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. 2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, Beijing, 2016, pp. 166-171.
- [6] Y. Mao, J. Zhang, and K. B. Letaief, "Joint Task Offloading Scheduling and Transmit Power Allocation for Mobile-Edge Computing Systems," in *Proc. 2017 IEEE Wireless Commun. and Netw. Conf. (WCNC)*, San Francisco, CA, 2017, pp. 1-6.
- [7] K. Zhang et al., "Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks," *IEEE Access*, vol. 4, pp. 5896-5907, 2016.
- [8] Q. Pham, L. B. Le, S. Chung, and W. Hwang, "Mobile Edge Computing with Wireless Backhaul: Joint Task Offloading and Resource Allocation," *IEEE Access*, vol. 7, pp. 16444-16459, 2019.
- [9] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile Edge Computing Empowered Energy Efficient Task Offloading in 5G," *IEEE Trans. Veh. Tech.*, vol. 67, no. 7, pp. 6398-6409, Jul. 2018.
- [10] K. A. Noghani, H. Ghazzai, and A. Kessler, "A Generic Framework for Task Offloading in mmWave MEC Backhaul Networks," in *Proc. IEEE Global Commun. Conf.*, Abu Dhabi, United Arab Emirates, 2018, pp. 1-7.
- [11] J. Zhou, F. Wu, K. Zhang, Y. Mao, and S. Leng, "Joint optimization of Offloading and Resource Allocation in Vehicular Networks with Mobile Edge Computing," in *Proc. 10th Int. Conf. on Wireless Commun. and Signal Process. (WCSP 2018)*, Hangzhou, 2018, pp. 1-6.
- [12] J. Guo, Z. Song, Y. Cui, Z. Liu, and Y. Ji, "Energy-Efficient Resource Allocation for Multi-User Mobile Edge Computing," in *Proc. IEEE GLOBECOM 2017*, Singapore, 2017, pp. 1-7.
- [13] ETSI, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Frequency(RF) Requirements for LTE Pico Node B (3GPP) TR 36.931 version 9.0.0 Release 9", ETSI TR 136 931 V9.0.0, pp.10-11, May 2011.
- [14] S. M. Johnson, "Optimal Two-and Three Stage Production Schedules with Setup Times Included," *Naval research logistics quarterly*, vol. 1, no. 1, pp. 61-68, 1954.
- [15] "scheduling" [online] Available: <http://www.bunkyo.ac.jp/nemoto/lecture/seisan/2001/scheduling2001.pdf>.