

# マルチユーザ MEC システムにおける 無線環境を考慮した移動ユーザの経路設計及び資源割当

棚木 拓海<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 東 10 号館  
E-mail: †{tanagi,adachi}@awcc.uec.ac.jp

**あらまし** インダストリー 4.0 では、人の代わりとして機能するロボットのような移動端末の活用が想定され、その適用先として災害現場やインフラ保守現場等が挙げられている。このような環境では移動端末に対して高度な情報処理能力が要求されるが、移動端末に搭載可能な情報処理能力には限りがある。この課題を解決する技術の一つとして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 では、強力な計算能力を持つ MEC サーバが端末から送信されたタスクを高速に代替処理し結果を端末へ返送することで、端末の不足した情報処理能力を補うことが可能となる。複数の MEC サーバを複数の移動端末で共有する環境では、適切な接続先 MEC サーバ選択、移動経路設計、及び通信・計算資源割当が処理能力向上のために重要となる。本研究では、動的計画法を用いた接続先 MEC サーバの選択を考慮する端末の移動経路決定法及びジョンソンアルゴリズムを用いた通信・計算資源割当法を提案し、タスク処理成功率の向上及びタスク処理遅延の短縮を図る。計算機シミュレーションにより、提案手法がベンチマークと比較してタスク処理成功率を向上させ、タスク処理遅延を短縮できることを示す。  
**キーワード** モバイルエッジコンピューティング、経路設計、資源割当、動的計画法、ジョンソンアルゴリズム

## Trajectory Optimization and Resource Allocation in Mobility-Aware MEC Systems

Takumi TANAGI<sup>†</sup> and Koichi ADACHI<sup>†</sup>

<sup>†</sup> Advanced Wireless & Communication Research Center, The University of Electro-Communications  
1-5-1 E-10 Bldg. Chofugaoka, Chofu-shi, Tokyo, 182-8585 Japan  
E-mail: †{tanagi,adachi}@awcc.uec.ac.jp

**Abstract** Mobile Edge Computing (MEC) is a technology that provides users with vast computing resources at low latency by allowing MEC servers to perform user-submitted tasks on their behalf. In MEC systems where multiple mobile users share multiple servers, server selection (i.e. user association), users' path planning, and communication and computing resources allocation are essential to take advantage of the MEC systems. In this paper, we propose a path planning and resource allocation method for multiple users sharing multiple MEC servers to improve the task processing success rate and shorten the task processing delay. Specifically, we adopt dynamic programming to derive the path that maximizes the sum of weighted communication rates considering user association and then use Johnson's algorithm to allocate the resources that minimize the maximum task processing delay. The numerical evaluation showed that the proposed method improved the MEC performance compared to the benchmark method.  
**Key words** MEC, Path planning, Resource allocation, Dynamic programming, Johnson's algorithm

### 1. はじめに

インダストリー 4.0 では、自動搬送車 (AGV: Automated Guided Vehicle) や無人航空機 (UAV: Unmanned Aerial Ve-

hicle)、人の代わりとして機能するロボットのような移動端末の活用が想定されている [1]。これらの技術の適用先として、災害現場やインフラ保守現場等が挙げられる。このような端末同士での協調動作が必要な環境において、端末は常に周囲の環境を把

握し、近傍の端末やネットワークを介して操縦者と通信を行う必要がある。そのため端末には高度な情報処理能力が求められるが、端末に搭載可能な情報処理能力には限りがある。この課題を解決する技術の一つとして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 [2].

MEC では、基地局 (BS: Base Station) などのネットワークエッジに強力な計算能力を持つ MEC サーバを配置する。MEC サーバは端末から BS へ送信 (オフロード) された計算タスクを代替処理し結果を端末へ返送することで、端末での処理を必要とせず高負荷な計算タスクの処理を可能とする。また、遠隔地に配置した強力な計算能力を持つクラウドサーバで端末からの計算タスクを代替処理するモバイルクラウドコンピューティング (MCC: Mobile Cloud Computing) と比較して、MEC サーバは端末との物理的距離が近いことによる低遅延なタスク処理を可能とする [3]. このような特徴から、MEC はインダストリー 4.0 の実現に必要な不可欠な技術であると考えられている [4].

MEC 環境においては、接続先 BS の選択、オフロード時の通信資源割り当て及び MEC サーバでの代替処理時の計算資源割り当てが MEC で提供可能な処理性能に影響を及ぼすことが知られており、既存研究ではそれらの適切な決定手法が提案されている [6]~[9]. 加えて、端末及び BS の移動性を想定した場合、その移動経路も MEC システムが提供可能な性能に影響を及ぼす。文献 [6] では、複数の静止した端末が 1 台の BS へ計算タスクをオフロードする環境を想定し、深層強化学習を用いた通信資源割り当てによりタスク処理遅延の最小化を行なっている。文献 [7] では、1 台の静止した端末と複数の BS が存在する MEC 環境を想定し、オンライン学習を用いた接続先 BS の選択及び送信電力の決定によりタスク処理遅延の最小化を行なっている。文献 [8] では、複数の静止した端末が 1 台の移動基地局へ計算タスクをオフロードする環境を想定し、ブロック座標降下法に基づく反復アルゴリズムを用いた計算資源割り当て及び移動基地局の経路設計により端末と移動基地局のエネルギー消費量の加重和を最小化している。文献 [9] では、複数の移動端末と複数の BS がある MEC 環境を想定し、リアプノフ最適化に基づくアルゴリズムを用いた計算資源割り当て及び接続先 BS の選択及び送信電力の決定により端末のエネルギー消費量を低減しつつタスク処理遅延も同時に低減している。端末の移動経路にはランダムウォークモデルが用いられている。

既存研究では、適切な接続先 BS の選択、移動 BS の経路設計、及び通信・計算資源の割り当て等による MEC システムの性能向上を目的としている一方で、端末の移動を考慮していないものが多い。例えば、既存研究 [9] ではスマートフォンやウェアラブル端末等を移動端末として想定しており、その移動経路は最適化の対象となっていない。そこで本研究では、複数の移動端末をサポートする MEC 環境におけるタスク処理遅延の短縮及び実行可能なタスク量の増大を目的とし、接続先 BS の選択を考慮した端末の移動経路及び通信・計算資源の割り当てを決定する手法を提案する。具体的には、接続先選択考慮した通信レートの総和を最大化させる移動経路の近似解を動的計画法 (DP: Dynamic Programming) で、最大タスク処理遅延を最

小化させる資源割り当てをジョソソナルアルゴリズムで導出する。

本稿は以下の構成になっている。第 2 章は MEC におけるタスク処理について記す。第 3 章にシステムモデル、第 4 章に提案手法について記す。第 5 章に計算機シミュレーションに用いる指標、諸元、及びその結果を記し、第 6 章にまとめを記す。

## 2. モバイルエッジコンピューティング (MEC)

### 2.1 タスク処理

MEC でのタスク計算処理に関する主な技術としてバイナリオフロード方式及びパーシャルオフロード方式がある [5]. 前者は計算タスクをローカル処理又はオフロード処理のどちらかのみで処理し、後者はタスクを分割し、それぞれローカル処理及びオフロード処理で計算処理する。本研究ではタスクが常にオフロード処理される方式について検討する。端末は接続先 BS の選択規範および通信資源割り当てに従い保持する計算タスクをオフロードする。BS はオフロードされた計算タスクを受信し、具備する MEC サーバで計算資源割り当て規範に従い処理し、端末へ計算結果を返送する。

### 2.2 タスク処理遅延

タスク処理遅延  $\tau^{\text{delay}}$  [sec] は以下のように表せる。

$$\tau^{\text{delay}} = t^{\text{exe}} + \tau^{\text{exe}} + \tau^{\text{back}} - t^{\text{gen}} \quad (1)$$

ここで、 $t^{\text{exe}}$  は計算処理開始時刻、 $\tau^{\text{exe}}$  [sec] は計算時間、 $\tau^{\text{back}}$  [sec] は返送時間、 $t^{\text{gen}}$  はタスク生起時刻を表す。

タスクの計算処理はオフロード完了後に実行される必要があるため、計算処理開始時刻はオフロード完了時刻より遅く次式が成り立つ、 $t^{\text{exe}} \geq t^{\text{off}} + \tau^{\text{off}}$ . ここで、 $t^{\text{off}}$  はオフロード開始時刻、 $t^{\text{off}}$  [sec] はオフロード時間を表す。また、タスクのオフロードはタスク生起後に実行されるため、オフロード開始時刻はタスク生起時刻より遅く次式が成り立つ、 $t^{\text{off}} \geq t^{\text{gen}}$ . オフロード時間はオフロードするタスクサイズ  $A^{\text{off}}$  [bit] 及びオフロード時の通信レート  $R^{\text{off}}$  [bps] を用いて以下のように表される、 $\tau^{\text{off}} = A^{\text{off}}/R^{\text{off}}$ . 計算時間は計算するタスクサイズ  $A^{\text{exe}}$  [bit]、1 [bit] のタスクの計算に必要な CPU サイクル数 (以降、タスクタイプと呼ぶ)  $D^{\text{exe}}$  [CPU cycle/bit]、及び MEC サーバの CPU サイクル周波数  $F^{\text{exe}}$  [CPU cycle/sec] を用いて以下のように表される [7]

$$\tau^{\text{exe}} = A^{\text{exe}} D^{\text{exe}} / F^{\text{exe}} \quad (2)$$

返送時間はタスク計算結果のサイズ  $A^{\text{back}}$  [bit] 及び返送時の通信レート  $R^{\text{back}}$  [bps] を用いて以下のように表される、 $\tau^{\text{back}} = A^{\text{back}}/R^{\text{back}}$ .

## 3. システムモデル

本研究では、図 1 に示すように正方形の通信エリア内に  $K$  台の BS をランダムに設置し、 $I$  台の端末が割り当てられた経路を移動する環境を想定する。ただし、各 BS 間距離は  $D^{\text{BS}}$  [m] 以上となるようにする。端末  $i \in \{0, 1, \dots, I-1\}$  は移動中にタスクを生起し、BS にオフロードする。オフロードされたタスクは BS に設置された MEC サーバで計算処理され、端末  $i$  へ

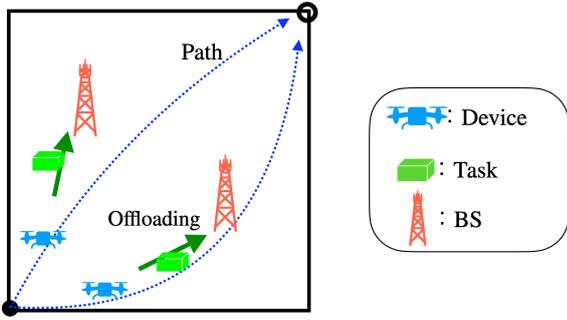


図1 システムモデルの概略図

返送される。

エリア内での位置を座標  $\mathbf{u} = [x, y, z]$  で表し、BS  $k \in \{0, 1, \dots, K-1\}$  の位置を  $\mathbf{u}_k^{\text{BS}}$  とする。全端末は、初期位置  $\mathbf{u}^{\text{S}}$  から目的位置  $\mathbf{u}^{\text{F}}$  までを移動時間  $T$  [sec] で移動し、時刻  $t \in [0, T]$  における端末  $i$  の位置を  $\mathbf{u}_i(t)$  とする。この時、全端末は同じ時刻で移動を開始するものとする。

### 3.1 接続先 BS の選択

接続先選択の変数として、時刻  $t$ 、端末  $i$ 、及び BS  $k$  を用いて表される 2 値変数  $\alpha_{i,k}(t) \in \{0, 1\}$  を定義する。この時、 $\alpha_{i,k}(t) = 1$  とは時刻  $t$  で端末  $i$  がタスクをオフロードする場合に接続先 BS として BS  $k$  が選択されることを、 $\alpha_{i,k}(t) = 0$  とは選択されないことを表す。ここで、各時刻において、各端末は 1 つの BS を接続先 BS として選択するものとする。接続先 BS の選択は受信信号電力規範で実施され、オフロード開始時に最もチャネル利得の高い BS を接続先 BS として端末は選択するものとする。

### 3.2 通信資源割り当て

同一 BS に複数の端末が接続している場合、端末間の通信資源割り当てには時分割多元接続 (TDMA: Time Division Multiple Access) を用いるものとする。通信資源割り当て変数として、時刻  $t$ 、端末  $i$  を用いて表される 2 値変数  $\beta_i(t) \in \{0, 1\}$  を定義する。この時、 $\beta_i(t) = 1$  とは時刻  $t$  で端末  $i$  はオフロードを実行可能、 $\beta_i(t) = 0$  とは不可能であることを表す。

### 3.3 計算資源割り当て

各 BS はオフロードされたタスクの計算処理開始時刻を計算資源割り当てに従い決定する。

### 3.4 タスク生起モデル

全端末は移動開始時刻でタスクを同時に生起し、以降、周期  $\tau^{\text{gen}}$  [sec] 毎にタスクを一つ生起し自身の端末バッファに保持する。この時、端末毎のタスクのサイズ [bit] とタイプ [CPU cycle/bit] は同じとする。各タスクは生起時刻からタスク処理遅延の許容時間経過後までに処理が未完了の場合破棄されるものとする。

### 3.5 オフロード

#### 3.5.1 チャンネルモデル

時刻  $t$  において、端末  $i$  と BS  $k$  間のチャネル利得  $\gamma_{i,k}(t)$  は次式で表される [10]。

$$\gamma_{i,k}(t) = 10a \log_{10} d_{i,k}(t) + b + 10c \log_{10} f^c \quad (3)$$

ここで、 $a$ ,  $b$ ,  $c$  はそれぞれパスロス指数、パスロスオフセット、搬送波周波数に関する係数を表し、 $d_{i,k}(t)$  [m] は時刻  $t$  における端末  $i$  - BS  $k$  間のユークリッド距離、 $f^c$  [GHz] は搬送波周波数を表す。

#### 3.5.2 ビームフォーミング

各 BS はアンテナ素子数  $L$ 、各素子の間隔  $\lambda \Delta^w$  [m]、及びアンテナの向きを通信エリアの中央に向けた等間隔線形アレーアンテナ (ULA: Uniform Linear Array) を具備しているものとする。従って、時刻  $t$  において、端末  $i^*$  へ指向性を向ける BS  $k$  での端末  $i$  からの送信信号に対するアンテナ利得  $G_{i,k,i^*}(t)$  は次式で与えられる。

$$G_{i,k,i^*}(t) = G(\theta_{i^*,k}(t), \phi_{i^*,k}(t), \theta_{i,k}(t), \phi_{i,k}(t)) \quad (4)$$

where

$$G(\theta^{\text{dsr}}, \phi^{\text{dsr}}, \theta^{\text{DoA}}, \phi^{\text{DoA}}) = \left| \sum_{\ell=0}^{L-1} \exp \{ j 2\pi \ell \Delta^w (\sin \theta^{\text{dsr}} \cos \phi^{\text{dsr}} - \sin \theta^{\text{DoA}} \cos \phi^{\text{DoA}}) \} \right|$$

ここで、 $\theta_{i,k}(t)$  [rad]、 $\phi_{i,k}(t)$  [rad] はそれぞれベクトル  $(\mathbf{u}_k^{\text{BS}} - \mathbf{u}_i(t))$  と  $z$  軸、ベクトル  $(\mathbf{u}_k^{\text{BS}} - \mathbf{u}_i(t))$  と BS  $k$  のアンテナの向きとのなす角を表す。また、各 BS は常に接続元端末へ指向性を向けるものとする。

#### 3.5.3 通信レート

時刻  $t$  において接続元端末  $i$  からの BS  $k$  における信号対干渉+雑音比 (SINR: Signal to Interference plus Noise power Ratio)  $\Gamma_{i,k}(t)$  は次式で与えられる。

$$\Gamma_{i,k}(t) = \alpha_{i,k}(t) \frac{\beta_i(t) P G_{i,k,i}(t) \gamma_{i,k}(t)}{\sum_{j \neq i} \beta_j(t) P G_{j,k,i}(t) \gamma_{j,k}(t) + N_0 B} \quad (5)$$

ここで、 $P$  [W] は端末の送信電力、 $N_0$  [W/Hz] は雑音電力スペクトル密度、 $B$  [Hz] は帯域幅を表す。

シャノンの通信容量定理より、時刻  $t$  における端末  $i$  - BS  $k$  間の通信容量は  $\log_2(1 + \Gamma_{i,k}(t))$  [bps/Hz] となる。従って、時刻  $t$  における端末  $i$  と BS  $k$  間の通信レート  $R_{i,k}(t)$  [bps] は次式で与えられる。

$$R_{i,k}(t) = B \log_2(1 + \Gamma_{i,k}(t)) \quad (6)$$

### 3.6 計算処理、返送処理

各 BS は受信完了したタスクを計算資源割り当てに従い計算処理する。各 BS はタスクを計算処理後、直ちにその計算結果を端末へ返送する。返送時の BS の送信電力は十分に大きい値であり、タスク計算結果のサイズは小さいものとする。また、端末は BS からの計算結果を理想的に受信可能であるとする。

## 4. 提案手法

本研究では、動的計画法 (DP) を用いて接続先 BS の選択を考慮した各端末の移動経路を決定する手法及びジョンソンアルゴリズムを用いて通信・計算資源の割り当てを決定する手法を提案する。

### 4.1 経路設計

経路設計では、重み付けされた通信レートで表現される目的

関数の総和を最大化することで移動経路を最適化する。

#### 4.1.1 目的関数

端末  $i$  に対して、目的関数  $f_i(t)$  を次式のように重み付けした通信レートで与える。

$$f_i(t) = B \log_2 \left( 1 + \frac{\sum_k \rho_{i,k}(t) P G_{i,k,i}(t) h_{i,k}(t)}{N_0 B} \right) \quad (7)$$

ここで、 $\rho_{i,k}(t)$  は重みを表し、以下で与える。

$$\rho_{i,k}(t) = \frac{\exp(\xi_{i,k}(t))}{\sum_m \exp(\xi_{i,m}(t))} \quad (8)$$

$$\text{where } \xi_{i,k}(t) = - \sum_{j < i} \alpha_{j,k}(t) + \max_{k'} \sum_{j < i} \alpha_{j,k'}(t)$$

式 (7) は、干渉信号を無視した各 BS との重み付けされた通信レートの総和を表す。各端末  $i$  の経路は添字  $i$  の昇順で greedy に決定される。従って、端末  $i$  ( $> 0$ ) の経路は端末  $j = \{0, 1, \dots, i-1\}$  の接続先 BS を考慮した重み  $\rho_{i,k}$  を用いて決定される。

#### 4.1.2 最適化問題

端末  $i$  について、重み付けされた通信レートの総和を最大化する最適化問題は以下のように与えられる。

$$(P1): \quad \text{maximize}_{\mathbf{u}_i(t)} \int_{t=0}^T f_i(t) \quad (9)$$

$$\text{subject to } \quad \dot{\mathbf{u}}_i(t) \leq V^{\max} \quad (10)$$

$$\mathbf{u}_i(0) = \mathbf{u}^S \quad (11)$$

$$\mathbf{u}_i(T) = \mathbf{u}^F \quad (12)$$

式 (9) は、端末  $i$  の最適な移動経路による目的関数の最大値を表す。式 (10) は、時間微分  $\dot{\mathbf{u}}_i(t)$  で表される端末  $i$  の移動速度が最大移動速度以下である制約を表す。式 (11) と式 (12) は、時刻 0 及び  $T$  における端末  $i$  の位置がそれぞれ初期位置、目的位置である制約を表す。

上記の問題 (P1) を全端末について解くことで、各時刻における各端末の位置が決定され最適な移動経路が求められる。しかし、目的関数が非凸であるため、問題 (P1) を解くことは一般的に難しい。従って、問題 (P1) を問題 (P2) として再定式する。

$$(P2): \quad \text{maximize}_{\mathbf{c}_i(t)} \int_{t=0}^T f_i(\mathbf{s}_i(t)) \quad (13)$$

$$\text{subject to } \quad \dot{\mathbf{s}}_i(t) = \mathbf{g}(\mathbf{c}_i(t)) \quad (14)$$

$$\sqrt{(\delta x_i(t))^2 + (\delta y_i(t))^2} \leq V^{\max} \quad (15)$$

$$\mathbf{s}_i(0) = \mathbf{u}^S \quad (16)$$

$$\mathbf{s}_i(T) = \mathbf{u}^F \quad (17)$$

制御入力  $\mathbf{c}_i(t) = (\delta x_i(t), \delta y_i(t))$  は時刻  $t$  における端末  $i$  の座標  $x, y$  軸での移動距離  $\delta x_i(t)$  [m],  $\delta y_i(t)$  [m] を表す。関数  $\mathbf{g}(\mathbf{c}_i(t)) = [\delta x_i(t), \delta y_i(t), 0]$  は制御入力による移動量を表す。式 (14) は、端末  $i$  の状態遷移についての制約を表す。状態  $\mathbf{s}_i(t)$  は時刻  $t$  における端末  $i$  の位置を表し、その時間微分  $\dot{\mathbf{s}}_i(t)$  に従い状態遷移を行う。式 (15) は、端末  $i$  の移動速度が最大移動

速度以下である制約を表す。式 (16) と (17) は、時刻 0 及び  $T$  における端末  $i$  の状態すなわち位置がそれぞれ初期位置、目的位置である制約を表す。

上記の問題 (4.5) には動的計画法を使用できるため、全端末について解くことで最適な移動経路を求めることができる。しかし、問題 (P2) にある連続的な関数に対して動的計画法を使用することは一般的に困難である。そこで、問題 (P2) を離散的に扱うことで動的計画法の使用による最適経路の近似解を導出する。

#### 4.1.3 動的計画法 (DP) [11]

移動時間  $T$  [sec] を  $N+1$  個のタイムスロットで表現すると、時刻  $t \in [0, T]$  に対応するタイムスロット  $n$  は次式で表される、 $n = \lfloor Nt/T \rfloor$ 。ただし、 $\lfloor \cdot \rfloor$  は床関数を表す。

時刻  $t$  をタイムスロット  $n$  で表現することで状態  $\mathbf{s}_i$  の状態遷移は以下のように表せる。

$$\mathbf{s}_i[n+1] = \mathbf{s}_i[n] + \mathbf{g}(\mathbf{c}_i[n]), \quad n = 0, 1, \dots, N-1 \quad (18)$$

$$\text{where } \quad \mathbf{g}(\mathbf{c}_i[n]) = [\delta x_i[n], \delta y_i[n], 0]$$

ここで、状態  $\mathbf{s}_i[n]$  はタイムスロット  $n$  における端末  $i$  の位置を表す。制御入力  $\mathbf{c}_i[n]$  はタイムスロット  $n$  における端末  $i$  の座標  $x, y$  軸での移動距離  $\delta x_i[n]$  [m],  $\delta y_i[n]$  [m] を表す。関数  $\mathbf{g}(\mathbf{c}_i[n])$  は制御入力  $\mathbf{c}_i[n]$  による移動量を表す。ここで、初期状態として  $\mathbf{s}_i[0] = \mathbf{u}^S$  が成り立つものとする。

端末  $i$  について、制御入力の組  $\pi_i$  として式  $\pi_i = \{\mathbf{c}_i[0], \mathbf{c}_i[1], \dots, \mathbf{c}_i[N-1]\}$  が成り立つように与えられた時、評価関数  $J_{\pi_i}(\mathbf{s}_i[0])$  は以下のように表される。

$$J_{\pi_i}(\mathbf{s}_i[0]) = J(\mathbf{s}_i[N]) + \sum_{n=0}^{N-1} f_i[n] \quad (19)$$

$$\text{where } \quad J(\mathbf{s}_i[N]) = \begin{cases} -\infty, & \mathbf{s}_i[N] \neq \mathbf{u}^F \\ f_i[N], & \mathbf{s}_i[N] = \mathbf{u}^F \end{cases}$$

従って、端末  $i$  の最適経路の近似解  $\pi_i^*$  は以下のように表すことができる。

$$\pi_i^* = \arg \max_{\pi_i \in \Pi_i} J_{\pi_i}(\mathbf{s}_i[0]) \quad (20)$$

$$\text{where } \quad \Pi_i = \{\mathbf{c}_i[n], n = 0, \dots, N-1 \mid \mathbf{c}_i[n] \in \mathcal{C}_i\}$$

ここで  $\mathcal{C}_i$  は端末  $i$  がタイムスロット  $n$  の時間内で移動可能な格子点へ移動するための制御入力  $\mathbf{c}_i$  の集合である。ただし、格子点はシミュレーションエリアを間隔  $\Delta^{\text{grid}}$  [m] の格子線で区切った際に現れる格子点とする。最大の評価関数値 (19) は、ベルマン方程式を用いることで以下のように時間を逆行させて再帰的に計算することができる。

$$J(\mathbf{s}_i[n]) = \max_{\mathbf{c}_i[n]} f_i[n] + J(\mathbf{s}_i[n+1]), \quad n = N-1, N-2, \dots, 0$$

## 4.2 通信・計算資源割り当て

資源割り当てでは、最大タスク処理遅延を最小化することで通信・計算資源割り当てを最適化する。

BS  $k$  について、各時刻  $t$  において接続元端末  $i$  ( $\alpha_{i,k}(t) = 1$ ) が保持するタスクに対して適切な通信・計算資源割り当てを決定即ち最適なオフロード開始時刻及び計算処理開始時刻により最大タスク処理遅延を最小化する。ここで、全端末が周期的にタスクを同時生起することを考慮すると、上記問題を時刻  $t = 0$  からタスク生起間隔毎に解く規模の小さい問題へ置き換えることができる。この問題は、TDMA による端末間のオフロードの排他性及び 2.2 節で述べた計算処理開始時刻がオフロード完了時刻で制約されることを考慮するとジョンソンアルゴリズムで解くことが可能である。ただし、オフロード完了時刻を構成するオフロード時間について、他端末からの干渉信号を考慮した式 (5) を用いると組合せ最適化問題となり解くことが困難なため他端末からの干渉信号を無視する。この制約緩和により、最適化されたオフロード開始時刻及び計算開始時刻が 2.2 節で述べた制約を満たさないため、最適化された各時刻の大小から決定される優先度を通信・計算資源割り当ての規範とする。

#### 4.2.1 ジョンソンアルゴリズム (Johnson's Algorithm) [12]

順序関係のある複数の作業で構成される複数のタスクを複数の機械で処理する時、全体の処理時間が最小になるようなタスクの処理順序を決定する問題をフローショップスケジューリング問題という。特に、タスクを構成する 2 つの作業がそれぞれ 2 つの機械で処理される時、上記問題は 2 ステージフローショップスケジューリング問題となり、ジョンソンアルゴリズムを用いて解くことが可能である。

タスクを構成する 2 つの作業を前作業及び後作業とし、その順序関係は前作業、後作業の順とする。また、前作業を処理する機械を機械 1、後作業を処理する機械を機械 2 とする。任意のタスクについて、そのタスクの前作業の処理が完了しない限り後作業の処理は開始できないことに注意する。ジョンソンアルゴリズムでは、始めに各作業の処理にかかる時間のリストを作成する。次に、作成されたリスト内で作業時間が最小の作業を探索する。探索で得られた作業が前作業の場合、その作業から構成されるタスクをリスト内のタスクで最も早く処理する。一方、探索で得られた作業が後作業の場合、その作業から構成されるタスクをリスト内のタスクで最も遅く処理する。次に、探索で得られた作業から構成されるタスクをリスト内から削除する。最後に、リスト内にタスクが存在する場合は作業の探索から再度実行し、存在しない場合はアルゴリズムを終了する。

オフロードを前作業及び計算処理を後作業と見なすと、通信・計算資源割り当ては 2 ステージフローショップスケジューリング問題となる。

## 5. 計算機シミュレーション

### 5.1 評価指標、諸元

本研究では、タスク処理成功率及び計算処理が完了したタスク (以降、完了タスクと呼ぶ) の処理遅延を評価する。タスク処理成功率は、全生起タスク数に対する完了タスク数の割合とする。タスク処理遅延は、各完了タスクの処理遅延の和を完了タスク数で平均化したものとする。ただし、タスク計算結果のサイズ  $A^{\text{back}}$  [bit] が小さく返送時間  $\tau^{\text{back}}$  [sec] は無視できるも

表 1 シミュレーション諸元

試行回数	1000
通信エリア	200 × 200 [m <sup>2</sup> ]
BS 台数 $K$	3
BS 間の最低距離 $D^{\text{BS}}$	100 [m]
BS の高さ	15 [m]
BS のアンテナ素子数 $L$	5
MEC サーバの CPU 周波数 $F^{\text{exe}}$	5 [GCPU cycle/sec]
雑音電力スペクトル密度 $N_0$	-174 [dBm/Hz]
雑音指数	14 [dB]
端末台数 $I$	{1, 2, ..., 50}
最大移動速度 $V^{\text{max}}$	20 [km/h]
移動時間 $T$	100 [sec]
タイムスロット数 $N$	1000
端末の送信電力 $P$	100 [mW]
搬送波周波数 $f^c$	6 [GHz]
帯域幅 $B$	100 [MHz]
タスクのサイズ, タイプ	1 [Mbit], 200 [CPU cycle/bit]
タスクの生起間隔, 処理許容時間	1 [sec], 1 [sec]
格子線間隔 $\Delta^{\text{grid}}$	0.4 [m]

のとする [6].

シミュレーション諸元を表 1 に示す。各端末は通信エリアの 2 組の対頂点中のランダムな 1 組を初期位置、目的位置とする。

### 5.2 シミュレーション結果

提案手法と各比較手法による端末台数に対するタスク処理成功率及びタスク処理遅延をそれぞれ図 2 と図 3 に示す。図中の凡例 “Prop.” は、提案する経路設計および通信・計算資源割り当てを行った場合を示す。また、“Prop. w/o PP” は、全端末が初期位置及び目的位置を直線的に結ぶ経路を等速で移動し、提案する通信・計算資源割り当てを行った場合、“Prop. fixed  $\rho$ ” は、全端末は重み  $\rho$  が次式で与えられるような接続先 BS の選択が未考慮の最適経路を移動する、 $\rho_{i,k}(\forall t) = 1$  (if  $i \equiv k \pmod{K}$ ), 0 (otherwise). “Prop. w/o RA” は、DP による経路設計を行った後に、端末の添字  $i$  の昇順で通信・計算資源割り当てのオフロード及び計算処理の優先度を決定する場合を示す。

#### 5.2.1 経路設計法における比較

図から分かるとおり、DP による適応的な経路を行うことにより、固定経路と比較してタスク処理成功率を増大させ、タスク処理遅延を低減させることが可能である。これは、DP で通信レートの総和を最大化させる経路設計を行うため、通信時間が短縮されタスク処理遅延が低減されたためである。さらに、タスク処理遅延が低減されたことにより、タスク処理遅延の許容時間内に処理されるタスク数が増大しタスク処理成功率が向上した。また、接続先 BS の選択を考慮した DP の経路は未考慮の DP の経路と比較してタスク処理成功率を増大し、タスク処理遅延を低減させることもわかる。これは、接続先 BS の選択を考慮した DP では多くの端末から接続先として選択された BS に近づく経路の目的関数値を小さくして経路選択を行うため、特定の BS への接続先選択即ちタスク計算処理の負荷が

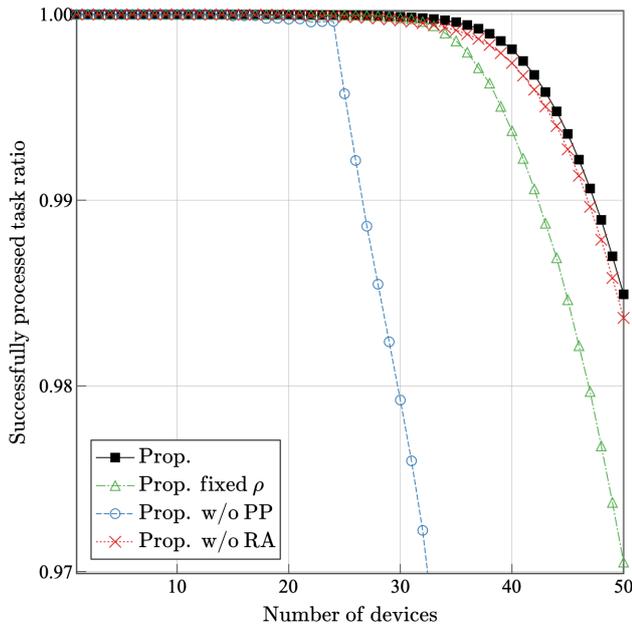


図2 端末台数に対するタスク処理成功率

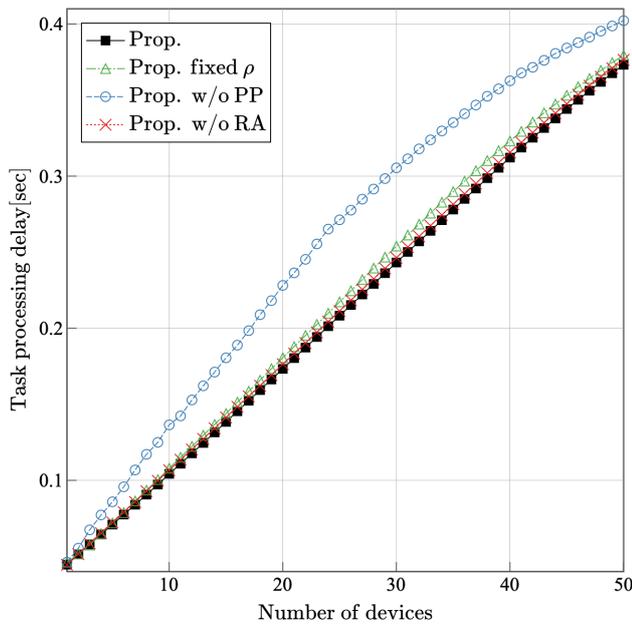


図3 端末台数に対するタスク処理遅延

偏るのを防いでいると考える。各BSでのタスク計算処理の負荷を均一化することで、タスク処理遅延の許容時間内に処理されるタスクが増えタスク処理成功率が向上、タスクの計算待ち遅延が短縮されタスク処理遅延が低減されたと考えられる。また、接続先BSの選択を均一化することで、タスクの通信待ち遅延も短縮されタスク処理遅延が低減されたと考えられる。

### 5.2.2 資源割り当て法における比較

結果より、ジョンソンアルゴリズムによる資源割り当て法は固定の割り当て法と比較してタスク処理成功率を増大させ、タスク処理遅延を低減させることが可能である。これは、ジョンソンアルゴリズムでは最大タスク処理遅延を最小化させる資源

割り当てを行うため、タスク処理遅延が最大のタスクをタスク処理遅延の許容時間内に処理させる可能性が向上しタスク処理成功率が増大したためである。また、適切なオフロード及び計算処理の優先度決定により通信・計算待ち遅延が低減され平均タスク処理遅延が低減したと考えられる。

## 6. まとめ

本稿では、接続先選択を考慮した端末の移動経路に動的計画法(DP)を、通信・計算資源割当にジョンソンアルゴリズムを用いて決定する手法を提案した。提案手法はBSの配置に応じた移動経路を柔軟に設計しタスクのオフロード時間及び計算処理時間に応じた適切な資源を割り当てることで、タスク処理成功率を向上させ、タスク処理遅延を短縮できることを示した。今後の課題として、経路設計に用いた重み関数についての検討、端末の物理的衝突を考慮した経路設計手法の提案が挙げられる。

## 文献

- [1] A. Fellan, C. Schellenberger, M. Zimmermann and H. D. Schotten, "Enabling Communication Technologies for Automated Unmanned Vehicles in Industry 4.0," in *Proc. Int. Conf. ICTC*, pp. 171-176, Oct. 2018.
- [2] H. Hu, W. Song, Q. Wang, F. Zhou and R. Q. Hu, "Mobility-Aware Offloading and Resource Allocation in MEC-Enabled IoT Networks," in *Proc. Int. Conf. MSN*, pp. 554-560, Dec. 2020.
- [3] H. Wu *et al.*, "EEDTO: An Energy-Efficient Dynamic Task Offloading Algorithm for Blockchain-Enabled IoT-Edge-Cloud Orchestrated Computing," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2163-2176, Feb. 2021.
- [4] S. K. Datta and C. Bonnet, "MEC and IoT Based Automatic Agent Reconfiguration in Industry 4.0," in *Proc. IEEE Int. Conf. ANTS*, pp. 1-5, Dec. 2018.
- [5] H. Wang, Z. Lin and T. Lv, "Energy and Delay Minimization of Partial Computing Offloading for D2D-Assisted MEC Systems," in *Proc. IEEE WCNC*, pp. 1-6, May 2021.
- [6] V. Dat Tuong *et al.*, "Delay-Sensitive Task Offloading for Internet of Things in Nonorthogonal Multiple Access MEC Networks," in *Proc. Int. Conf. ICTC*, pp. 597-599, Oct. 2020.
- [7] K. Guo, R. Gao, W. Xia and T. Q. S. Quek, "Online Learning Based Computation Offloading in MEC Systems With Communication and Computation Dynamics," *IEEE Trans. Commun.*, vol. 69, no. 2, pp. 1147-1162, Feb. 2021.
- [8] J. Ji, K. Zhu, C. Yi and D. Niyato, "Energy Consumption Minimization in UAV-Assisted Mobile-Edge Computing Systems: Joint Resource Allocation and Trajectory Design," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8570-8584, May 2021.
- [9] H. Hu, W. Song, Q. Wang, F. Zhou and R. Q. Hu, "Mobility-Aware Offloading and Resource Allocation in MEC-Enabled IoT Networks," in *Proc. Int. Conf. on Mobility, Sensing and Networking*, pp. 554-560, Dec. 2020.
- [10] ITU-R. P1411-10, *Propagation Data and Prediction Methods for the Planning of Short-Range Outdoor Radio Communication Systems and Radio Local Area Networks in the Frequency Range 300 MHz to 100 GHz*, Aug. 2019.
- [11] D. Bertsekas, *Dynamic Programming and Optimal Control.*, Athena Scientific, 2012.
- [12] 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.