2024年度　芝浦工業大学大学院

修　士　論　文

題目：ジョブの処理順を考慮した非協力ゲーム理論
によるクラウドレット間のオフロード最適化

専　攻　理工学研究科（修士課程）電気電子情報工学専攻

学籍番号　MA23203

ふりがな　よこた　ゆうき

氏　名　横田　侑紀

指導教員　上岡　英史

目次

図目次

第 1 章

序論

近年，モバイルクラウドコンピューティングは，携帯端末の限定的な電源，記憶容量，演算能力といった課題を補う技術として発展してきた．特に，仮想現実やオンラインゲーム，テレオペレーションなどの平均遅延時間が約 10-100 ms であることが求められるアプリケーションの増加に伴い，遅延の削減を目的としたエッジコンピューティングの技術を用いた解決策が注目されている [1]．その一つであるクラウドレットは，Mahadev ら [2] によって提唱され，ユーザと同一 LAN 内に存在し 1 ホップで接続可能なコンピュータクラスタ群である．クラウドレットは，従来のエッジコンピューティングが基地局などにサーバを配置するのに対し，LAN 内のルータ等に設置されるため，より低遅延での通信が可能となる．

クラウドレットの問題点として，従来のクラウドコンピューティングやエッジコンピューティングと比べると規模が小さく，処理能力が著しく低いことが挙げられる．そのため，クラウドレットの限られたリソースを効率よく利用することが求められる．無線アクセスネットワークにおける既存研究としてクラウドレットの配置問題に注目した手法 [3,4] があるが，分散型コンピューティングシステムでは到着した各ジョブがどのコンピュータで処理されるかを決めることで，システム全体の負荷分散を行うことが重要である [5]．そのため，携帯端末からクラウドレットへのジョブ要求の割り当てを平均遅延時間などに基づいて決める手法 [6,7] などが提案されている．ただ，携帯端末ユーザの複雑な移動パターン及び到着率のランダム性を考慮すると，クラウドレットの負荷の分散が大きくなることは免れられない [8]．高すぎる負荷は，遅延によるシステムの Quality of Service(以下 QoE) の低下を招き，反対に，低すぎる負荷はクラウドレット提供者の利益を下げ，サービスの維持が困難になる．

そこで，本研究の目的は，クラウドレット間でジョブのオフロードを行い，効率的なリソースの利用を可能とし，システムの QoE とクラウドレットの満足度を両立させた手法の検討である．

このような問題に対し，Sourav ら [9] は，システムの負荷を分散させるため，クラウドレット間でジョブをオフロードする手法を提案した．この手法では，クラウドレットが可能な限り多くのジョブを処理し自身の満足度を満たしつつ，ユーザの平均遅延が許容遅延以下に保たれるように，非協力ゲーム理論を用いてオフロードするジョブの割合を決定する．しかし，この手法ではオフロード割合の低いジョブの遅延が許容遅延を上回る可能性があるため，オフロードされたジョブの遅延制御が十分に行われていないという課題がある．

そこで本論文では，まずジョブの各割合の遅延を直接評価する手法 Latency Not-Weighted offloading（以下 LNW）の評価を行う [10]．これにより，従来の手法における重み付けされた遅延ではなく，各オフロード割合のジョブの遅延を計算することが可能となることを示す．また，遅延を評価する項のみに注目した利得関数を新たに提案し，LNW のみを用いた利得関数による非協力ゲームを解析し，LNW [10] の有効性を調査する．具体的には，この手法を用いて，Sourav ら [9] による Conventional Load Balancing（以下 CLB）手法と比較し，異なるトラヒック負荷において，優れた負荷分散を達成する事を示す．

一方で，これらの手法では，ジョブの処理をすべて到着順である first come, first served（以下 FCFS）でモデル化している．FCFS では，任意の処理の長いジョブをオフロードするかどうかによって全体の処理遅延が大きく変わるため，オフロード選択に限界があるという問題点が挙げられる．この問題を解決するため，新たな提案手法として，オ

フロード後の到着流に着目し，非割込み優先規律に基づいて導出した処理遅延を用いて，オフロードしたジョブを優先する処理法 Offload Prioritized（以下 OP），オフロードしていないジョブを優先する処理法 Offload Not-Prioritized（以下 ONP）の 2 つを提案する．これら OP，ONP と従来の FCFS による処理法の処理遅延と比較するため，異なる遅延による 3 種類の非協力ゲームにおける利得関数を新たに定義し，ナッシュ均衡の平均利得によって，提案モデルの有効性を示す．

以上のことから，本論文の貢献は以下の 4 つの要素にある:

- オフロードされたジョブの遅延を考慮するため，ジョブの各割合の遅延を直接評価する手法（LNW）のみを用いた，利得関数を提案する．
- クラウドレットの遅延のみに着目した利得関数を提案し，ナッシュ均衡における分散度合いによって，LNW の有効性を示す．
- クラウドレット間のオフロード処理を従来の FCFS モデルだけでなく，複数到着流を持つ非割り込み優先規律を用いた $M/M/1$ 待ち行列モデルでモデル化し，異なる処理法 (OP, NOP) を提案する．
- 異なる処理順のモデルを用いて，すべてのクラウドレットにとって最良な選択となるオフロード選択の組み合わせを導出し，利得の平均値によって異なる環境下における提案手法の有効性を示す．

本論文の構成は以下の通りである．まず，2 章でクラウドレットの資源の効率化について，またゲーム理論を用いた手法に関する関連研究について述べ，3 章で本論文における想定システムのモデル化を示す．4 章では，遅延を評価する項のみに注目した利得関数と，待ち行列理論を用いた提案手法の処理遅延を導出し，それぞれの非協力ゲームにおけるナッシュ均衡の導出過程を示す．5 章で数値解析，6 章でまとめと今後の課題を述べる．

第 2 章

関連研究

既存のクラウドレットの負荷を分散させる手法として，エリア内のユーザ密度に応じてクラウドレットを設置するアクセスポイントを決め，各ユーザの平均遅延時間が最も小さくなるような配置決定を行っている手法 [3, 4]，携帯端末からクラウドレットへのジョブ要求の割り当てを平均遅延時間などに基づいて決める手法 [6, 7]，そしてクラウドレット間でジョブをオフローディングを行う手法 [11, 12] などが提案されている．クラウドレットの配置決定の研究 [4] 及びジョブ割り当て [7] では，現実のユーザの複雑な移動パターンによるジョブの到着率のランダム性を考慮することが難しく，限定的な環境での解析のみを行っている．また，多くの研究 [4, 6, 7] ではユーザの平均遅延時間をできるだけ小さくすることに注目しているため，トレードオフによりクラウドレットの設置コストや台数への影響が懸念される．加えて，これらの手法で考えられているシステムは，中央集権型のロードバランサの役割を持っている．このようなシステムでは，システム全体としての利が優先されるため，個々のクラウドレットのサービスの維持が困難となる場合が考えられる．

システム全体の利ではなく，個々のクラウドレットの満足度に注目した研究として，ゲーム理論を用いた研究 [9, 10] がある．Sourav ら [9] はシステムの負荷を分散させるため，クラウドレット間でジョブをオフロードする手法を提案し，その中で遅延が一定の閾値以下であればユーザ満足度は変化しないことに着目した．この研究では，クラウドレットが可能な限り多くのジョブを処理しつつ，ユーザの平均遅延が許容遅延以下に保たれるように非協力ゲーム理論を用いてオフロードするジョブの割合が決定される．しかし，許容遅延を各クラウドレットの利用率によって重み付けしているため，オフロード割合の低いジョブの遅延が許容遅延を上回る結果になっている．そこで，遅延を重み付けせず，公平に評価したオフロードに着目した既存研究 [10] では，クラウドレット間の利用率の差を最小化する値がナッシュ均衡解となるため，従来の許容遅延内負荷分散法より高い平均利得を得ることができた．一方で，これらの手法では，ジョブの処理をすべて到着順である FCFS でモデル化している．FCFS では，任意の処理の長いジョブをオフロードするかどうかによって全体の処理遅延が大きく変わるため，オフロード選択に限界があるという問題点が挙げられる．

本論文ではこれらの手法を参考にし，新しく提案した処理法を用いて，クラウドレットが行う最良の選択における平均利得によって，提案手法を評価する．

第 3 章

システムモデル

3.1 想定システムと基本モデル

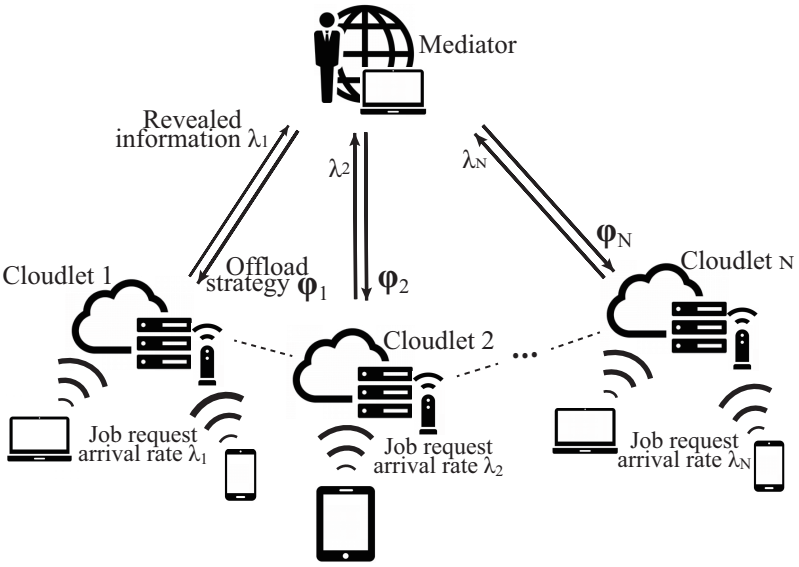

図 3.1: システムモデル

図 3.1 に，本手法におけるシステムのモデルを示す．本論文では，ユーザから無線環境下でクラウドレットへと送られる様々な計算処理をジョブと定義し，これらを処理する同一 LAN 内のクラウドレット $i \in \{1, 2, ..., N\} = C$, $N \geq 2$ について考える．すべてのクラウドレットは有線で互いに通信可能であり，管理者 (Mediator) を通して共有される互いの平均到着率の情報 (Revealed Information) をもとに，各クラウドレットが自身のジョブのオフロード割合を決定する．

クラウドレット i に到着するユーザから送信されるジョブの到着率が，平均 λ_i ポアソン分布に従うと仮定し，各ジョブの処理率は平均処理時間 b を用いて平均 $\mu = \frac{1}{b}$ の指数分布に従うと仮定する．このとき，クラウドレット i におけるジョブの到着過程と処理は $M/M/1$ 待ち行列モデルでモデル化できる．クラウドレット i に到着したジョブのオフロード先を決めるため，割合ベクトル $\varphi_i = (\varphi_{i1}, \varphi_{i2}, ..., \varphi_{iN}) \subset \mathbb{R}^N$, $\varphi_{ij} = [0, 1] \in \mathbb{R}$, $\sum_{j=1}^{N} \varphi_{ij} = 1, i, j \in C$ を定義する．この時，φ_{ij} は i 番目のクラウドレットが j 番目のクラウドレットにオフロードするジョブの割合を示す．したがって，$\varphi_{ii}\lambda_i$ のジョブがオフロードされずにクラウドレット i で処理され，$\varphi_{ij}\lambda_i$ のジョブがクラウドレット j へとオフロードされる．また，個々のクラウドレット間の平均往復遅延は，t_{ij}, $i \in C$, $j \in C \setminus \{i\}$ と定義する．

上記パラメータを用いて，FCFS で処理を行う従来の $M/M/1$ 待ち行列モデルと，提案する処理法である OP，ONP で処理を行う非割込み優先規律を用いた $M/M/1$ 待ち行列モデルを用いて，各クラウドレットを待ち行列理論でモデル化し，各到着流の平均遅延時間を導出する．

3.2　FCFS 待ち行列モデル

本節では，クラウドレットがジョブの到着順に処理を行う FCFS の場合の平均遅延時間を考える．

クラウドレット i におけるジョブの処理は $M/M/1$ でモデル化されるため，パラメータ λ_i, μ, φ_i を用いて，定常状態におけるクラウドレット i に到着する各オフロード割合の平均処理遅延 T_{ij}^{FCFS}, $i, j \in C$ は，リトルの公式で導出できる．オフロードされずに処理されるジョブ $\varphi_{ii}\lambda_i$ が，クラウドレット i に到着してからユーザへと返されるまでの平均遅延 T_{ii}^{FCFS} は，次式で導出できる．

$$T_{ii}^{\mathrm{FCFS}} = \frac{1}{\mu - (1 - \sum_{j=1, j\neq i}^{N} \varphi_{ij})\lambda_i - \sum_{j=1, j\neq i}^{N} \varphi_{ji}\lambda_j}. \tag{3.1}$$

また，クラウドレット j へとオフロードされたジョブ（$j \neq i$ の場合）の平均遅延 T_{ij}^{FCFS} は，クラウドレット間の往復遅延 t_{ij} を用いて次のように表される．

$$T_{ij}^{\mathrm{FCFS}} = t_{ij} + \frac{1}{\mu - \varphi_{ij}\lambda_i - \sum_{k=1, k\neq i}^{N} \varphi_{kj}\lambda_k}. \tag{3.2}$$

3.3　OP, NOP 待ち行列モデル

本節では，オフロードされたジョブを優先する処理法 OP と，オフロードされていないジョブを優先する処理法 ONP の平均遅延時間について考える．

クラウドレットは，自身に到着するジョブをその到着流のクラス $m \in \{1, 2, ..., N\}$ に応じて，$1 < 2 < ... < m < ... < N$ の順に処理を行うとする．このような処理は，複数到着流を持つ非割り込み優先規律を用いた $M/M/1$ 待ち行列モデルでモデル化できる [15].

クラウドレット i からクラウドレット j にオフロードされる任意のジョブの到着流をクラス m としたとき，そのクラスのジョブの利用率を ρ_m と表す．また，任意のクラス m から見たクラス m 以下のジョブを下位クラス，m より大きいクラスのジョブを上位クラスと呼ぶ．定常状態におけるクラス m のジョブの平均処理遅延は，そのジョブの処理に要する時間に加えて，到着時点での平均待ち時間 $W_{ij,m}$ で表される．この $W_{ij,m}$ は，ジョブ到着時点で進行中の処理の残り時間，既に到着している下位クラスのジョブの処理時間，そしてクラス m のジョブの処理が開始されるまでに，新たに到着する下位クラスのジョブの処理時間の和で導出できる．

最初に，クラウドレット i からクラウドレット j に到着するクラス 1 の到着流のジョブに注目する．この到着流の平均処理時間 $T_{ij,1}$ は，そのジョブの平均処理時間 b と，平均待ち時間 $W_{ij,1}$ で求められる．ここで，クラス 1 のジョブには下位クラスとなるジョブが他に存在しないため，$W_{ij,1}$ はジョブ到着時点で進行中の処理の平均残り時間と，既に到着しているクラス 1 のジョブの処理時間の和のみとなり，リトルの公式の式変形により導出可能である [**?**]．したがって，処理率が指数分布に従うことを考慮すると，平均処理時間の 2 次積率 $b^{(2)} = 2b^2$ を用いて $T_{ij,1}$ は以下のように求められる．

$$\begin{aligned} T_{ij,1} &= W_{ij,1} + b \\ &= \frac{\sum_{m=1}^{N} b^{(2)}\rho_m}{2b(1-\rho_1)} + b \\ &= \frac{\sum_{m=1}^{N} \rho_m b}{(1-\rho_1)} + b. \end{aligned} \tag{3.3}$$

クラス 2 以上の到着流のジョブの平均待ち時間 $W_{ij,m}, (m \geq 2)$ は前述したとおりとなるため，任意のクラス

M $(2 \leq M \leq N)$ の平均処理遅延 $T_{ij,M}$ は以下のように求められる．

$$\begin{aligned} T_{ij,M} &= \frac{\sum_{m=1}^{N} b^{(2)} \rho_m}{2b(1-\sum_{m=1}^{M-1}\rho_m)(1-\sum_{m=1}^{M}\rho_m)} + b \\ &= \frac{\sum_{m=1}^{N} \rho_m b}{(1-\sum_{m=1}^{M-1}\rho_m)(1-\sum_{m=1}^{M}\rho_m)} + b. \end{aligned} \tag{3.4}$$

3.4　クラウドレットの満足度

本論文におけるクラウドレットの満足度指標は，クラウドレットがオフロードを選択した際の利用率と，オフロード選択による利用率の変化量の和として定義される．

まず，クラウドレットの利用率は稼働率を反映しており，平均稼働率が高い場合には，計算資源の効率的な利用が達成されていることを示唆するため，クラウドレットの満足度は高いと評価される．また，他のクラウドレットからジョブをオフロードされた場合，ジョブ到着率が増加し利用率が上昇する．反対に，他のクラウドレットへジョブをオフロードした場合，ジョブ到着率が減少し利用率も低下する．このことから，オフロードによる利用率の変化量は，オフロード操作がクラウドレットにおけるジョブ処理量へ与える影響を反映しており，オフロードという手法によるクラウドレット満足度の変化を定量的に評価可能である．一方で，利用率の増加は処理遅延の増加に直結するため，上記の値のみでクラウドレットの満足度を考えると，システムの QoS は悪くなる．

そこで，遅延が許容時間以下であれば利用者としては十分であることを考慮し，クラウドレット i に到着するジョブの平均遅延時間が許容遅延 D_{rq} を満たせばその分だけクラウドレットの満足度が増加し，満たさなければその分に応じて満足度が減少するような値 E_i を，LNW [10] に基づいて利用率で重み付けせず導入する．これにより，クラウドレットはできる限り許容遅延を満たすようなオフロードを行うため，QoS の維持とクラウドレットの満足度を直結させることができる．

第 4 章

提案手法

本論文では，システム内のすべてのクラウドレット同士がプレイヤーとなり，お互いへのオフロード割合を戦略として決定する過程を，非協力ゲームでモデル化する．

ゲーム理論において，ナッシュ均衡とは，各プレイヤーが自分の戦略を変更しても利得を改善できないような戦略の組み合わせを示す．この均衡では，すべてのプレイヤーが他のプレイヤーの戦略を固定したとき，自分にとって最適な戦略を選んでいるため，誰も一方的に戦略を変える動機を持たず，ゲーム全体が安定した状態になる．したがって，本論文における最適なオフロード割合とは，各ゲームのナッシュ均衡の組み合わせとする．

4.1 LNW の調査

4.1.1 遅延の評価を用いた利得関数と非協力ゲーム

LNW [10] は，クラウドレットの遅延が許容遅延を満たしているかを公平に評価する手法である．具体的には，各クラウドレットで処理されるジョブの到着流ごとに，その遅延の値を重み付けせず，許容遅延との差を計算する．これにより，各到着流の遅延が許容遅延を超えていればそれぞれの項が負となり，反対に，遅延が許容遅延を満たせばその分を利得として獲得できる．よって LNW は，遅延の公平な評価がクラウドレットの満足度につながる手法となっている．

LNW の調査を行うため，遅延の評価のみを行う利得関数を提案する．本提案では，FCFS で処理されるクラウドレット間のオフロードに注目する．FCFS におけるクラウドレット i の到着流の平均遅延は，式 (3.1)，(3.2) を用いて導出できる．以下に，遅延の項のみに着目した利得関数 U_i^{LNW} を示す．

$$\begin{aligned} U_i^{\mathrm{LNW}}(\varphi_i,\ \varphi_{-i}) = -\Bigg\{ & \frac{1}{\mu - (1 - \sum_{j=1, j\neq i}^{N} \varphi_{ij})\lambda_i - \sum_{j=1, j\neq i}^{N} \varphi_{ji}\lambda_j} - D_{\mathrm{rq}}) \\ & +\Big(\sum_{j=1, j\neq i}^{N} \left[t_{ij} + \frac{1}{\mu - \varphi_{ij}\lambda_i - \sum_{k=1, k\neq i}^{N} \varphi_{kj}\lambda_k} - D_{\mathrm{rq}} \right] \Big) \Bigg\}. \end{aligned} \tag{4.1}$$

各クラウドレットは，この U_i^{LNW} がより大きくなるようなオフロード割合をそれぞれ決定する．そのため，自身がオフロードを行うことで得られる利得が，オフロードを全く行わなかった利得を下回ることは非合理的である．したがって，自身がオフロードを全く行わなかった場合の利得を $U_i^{0\ \mathrm{LNW}}, i \in C$ とすると，以下のような不等式条件式が定義できる．

$$\begin{aligned} U_i^{\mathrm{LNW}}(\varphi_i,\ \varphi_{-i}) \geq U_i^{0\ \mathrm{LNW}} = -\Bigg\{ & \frac{1}{\mu - \lambda_i - \sum_{j=1, j\neq i}^{N} \varphi_{ji}\lambda_j} - D_{\mathrm{rq}}) \\ & +\Big(\sum_{j=1, j\neq i}^{N} \left[t_{ij} + \frac{1}{\mu - \sum_{k=1, k\neq i}^{N} \varphi_{kj}\lambda_k} - D_{\mathrm{rq}} \right] \Big) \Bigg\}. \end{aligned} \tag{4.2}$$

ゲーム理論を用いると，LNW で扱うゲームは $\Gamma^{\mathrm{LNW}} = (C,\ \{\boldsymbol{\theta}_i\}_{i\in C},\ \{U_i^{\mathrm{LNW}}(\boldsymbol{\varphi}_i,\ \boldsymbol{\varphi_{-i}})\}_{i\in C})$ でモデル化できる．クラウドレットの戦略の集合は $\Theta = (\boldsymbol{\theta}_i = \{\boldsymbol{\varphi}_i : U_i^{\mathrm{LNW}}(\boldsymbol{\varphi}_i,\ \boldsymbol{\varphi}_{-i}) \geq U_i^{0\ \mathrm{LNW}}\},\ i \in C)$ で定義され，クラウドレット i は $U_i^{0\ \mathrm{LNW}}$ 以上の利得を得られるような集合の中から戦略を選ぶ．したがって，ゲーム Γ^{LNW} では各プレイヤーの戦略がお互いの利得だけでなくお互いの戦略の集合にも影響しているため，Generalized Nash Equilibrium(以下 GNE) 問題として扱われる．通常 GNE 問題は計算が非常に複雑になるため，解を見つけるのが困難であるが，特定の条件を満たす場合において，Variational Inequality(以下 VI) 問題に置き換えられることが証明でき，その VI 問題においてただ一つの解が求められれば，それが GNE 問題のナッシュ均衡と一致することが分かっている [9].

以上より，まずはクラウドレットの戦略の集合 Θ がコンパクトかつ凸な集合であると同時に，利得関数 $U_i^{\mathrm{LNW}}(\boldsymbol{\varphi_i},\ \boldsymbol{\varphi_{-i}})$ が凸関数であることを示すことで，ゲーム Γ^{LNW} が VI 問題に置き換えられることを証明する．次に，クラウドレット i の最適反応関数である $\frac{\partial U_i}{\partial \varphi_{ij}} = 0$ の単調増加性を示すことで VI が唯一解を持つことを示し，ゲーム Γ^{LNW} のナッシュ均衡である最適オフロード割合 $\boldsymbol{\varphi}_i^{*\ \mathrm{LNW}}$ の存在を明らかにする．

4.1.2　VI 問題と唯一解の証明

本論文のゲーム Γ^{LNW} が VI 問題に置き換えられることを証明するため，Conventional [9] における証明を参考に，$F = \nabla_{\boldsymbol{\varphi}_i}(U_i^{\mathrm{LNW}}(\boldsymbol{\varphi}_i,\ \boldsymbol{\varphi}_{-i}))$ からなる $\mathbf{VI}(\Psi,\ F)$ となるような VI 問題を仮定する．この時，Ψ は $\mathbf{VI}(\Psi,\ F)$ の解集合であり，$\Psi = \Theta \subseteq [0,\ 1]^{N\times N}$ を指す．これより，Ψ は有界閉集合であるため，有次元空間 $\mathbb{R}^{N\times N}$ においてコンパクト集合であると同時に，凸集合となる [13], [14].

定常状態における利得関数 $U_i^{\mathrm{LNW}}\ (\boldsymbol{\varphi_i},\ \boldsymbol{\varphi_{-i}})$ の凸性を示すため，(4.1) 式の変数 $\varphi_{ij},\ \varphi_{ji},\ \varphi_{jk}, i \in C,\ j \in C \setminus \{i\},\ k \in C \setminus \{i, j\}$ を用いて一階偏微分を式 (4.3–4.5) で計算する．

$$\begin{aligned}\frac{\partial U_i^{\mathrm{LNW}}}{\partial \varphi_{ij}} = &-\frac{1}{N}\frac{\lambda_i}{\mu}\left\{\left(\frac{-\lambda_i}{(\mu-(1-\sum_{j=1,j\neq i}^{N}\varphi_{ij})\lambda_i-\sum_{j=1,j\neq i}^{N}\varphi_{ji}\lambda_j)^2}\right)\right.\\ &\left.+\left(\frac{\lambda_i}{(\mu-\varphi_{ij}\lambda_i-\sum_{k=1,k\neq i}^{N}\varphi_{kj}\lambda_k)^2}\right)\right\},\end{aligned} \tag{4.3}$$

$$\begin{aligned}\frac{\partial U_i^{\mathrm{LNW}}}{\partial \varphi_{ji}} = &-\frac{1}{N}\frac{\lambda_i}{\mu}\left\{\left(\frac{\lambda_j}{(\mu-(1-\sum_{j=1,j\neq i}^{N}\varphi_{ij})\lambda_i-\sum_{j=1,j\neq i}^{N}\varphi_{ji}\lambda_j)^2}\right)\right.\\ &\left.+\left(\frac{-\lambda_j}{(\mu-\varphi_{ij}\lambda_i-\sum_{k=1,k\neq i}^{N}\varphi_{kj}\lambda_k)^2}\right)\right\},\end{aligned} \tag{4.4}$$

$$\frac{\partial U_i^{\mathrm{LNW}}}{\partial \varphi_{jk}} = \frac{1}{N}\frac{\lambda_i}{\mu}\left\{\left(\frac{\lambda_j}{(\mu-\varphi_{ij}\lambda_i-\sum_{k=1,k\neq i}^{N}\varphi_{kj}\lambda_k)^2}\right)\right\}. \tag{4.5}$$

続いて，ヘッセ行列の各成分を式（4.6–4.8）に示す．

$$\begin{aligned}\frac{\partial^2 U_i^{\mathrm{LNW}}}{\partial \varphi_{ij}^2} = &-2\frac{1}{N}\frac{\lambda_i}{\mu}\left\{\left(\frac{\lambda_i^2}{(\mu-(1-\sum_{j=1,j\neq i}^{N}\varphi_{ij})\lambda_i-\sum_{j=1,j\neq i}^{N}\varphi_{ji}\lambda_j)^3}\right)\right.\\ &\left.+\left(\frac{\lambda_i^2}{(\mu-\varphi_{ij}\lambda_i-\sum_{k=1,k\neq i}^{N}\varphi_{kj}\lambda_k)^3}\right)\right\} < 0.\end{aligned} \tag{4.6}$$

$$\begin{aligned}\frac{\partial^2 U_i^{\mathrm{LNW}}}{\partial \varphi_{ij}\varphi_{ik}} = &-2\frac{1}{N}\frac{\lambda_i}{\mu}\left\{\left(\frac{\lambda_i^2}{(\mu-(1-\sum_{j=1,j\neq i}^{N}\varphi_{ij})\lambda_i-\sum_{j=1,j\neq i}^{N}\varphi_{ji}\lambda_j)^3}\right)\right\}\\ &< 0.\end{aligned} \tag{4.7}$$

$$\frac{\partial^2 U_i^{\mathrm{LNW}}}{\partial \varphi_{ji}^2} = -2\frac{1}{N}\frac{\lambda_i\lambda_j}{\mu}\left\{\left(\frac{\lambda_j}{(\mu-(1-\sum_{j=1,j\neq i}^{N}\varphi_{ij})\lambda_i-\sum_{j=1,j\neq i}^{N}\varphi_{ji}\lambda_j)^3}\right) + \left(\frac{\lambda_j}{(\mu-\varphi_{ij}\lambda_i-\sum_{k=1,k\neq i}^{N}\varphi_{kj}\lambda_k)^3}\right)\right\} < 0. \tag{4.8}$$

式 (4.6)-(4.8) より，定常状態であればヘッセ行列の各成分が常に負になることが確認できる．よって利得関数 $U_i^{\mathrm{LNW}}(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_{-i})$ は上に凸の関数であり，ゲーム Γ^{LNW} の解空間も凸集合である．以上のことから，ゲーム Γ^{LNW} は $\boldsymbol{VI}(\Psi, F)$ に置き換えて解くことができる．

クラウドレット i の最適反応関数である $\frac{\partial U_i^{\mathrm{LNW}}}{\partial \varphi_{ij}} = 0$ では，クラウドレット i のオフロード割合 $\boldsymbol{\varphi}_i$ の各変数が密接に関わっているため，最適反応関数の単調増加性を示すための計算は非常に複雑である．したがって，まずは $N = 2$ と置いたときの単調増加性を確認する．

クラウドレットが $N = 2$ のとき，利得関数 U_1^{LNW}, U_2^{LNW} の 2 階微分の各項を以下に示す．

$$\frac{\partial^2 U_1^{\mathrm{LNW}}}{\partial \varphi_{12}^2} = -2\frac{1}{N}\frac{\lambda_1}{\mu}\left\{\left(\frac{\lambda_1^2}{(\mu-(1-\varphi_{12})\lambda_1-\lambda_2)^3}\right) + \left(\frac{\lambda_1^2}{(\mu-\varphi_{12}\lambda_1-(1-\varphi_{21})\lambda_2)^3}\right)\right\} < 0. \tag{4.9}$$

$$\frac{\partial^2 U_2^{\mathrm{LNW}}}{\partial \varphi_{21}^2} = -2\frac{1}{N}\frac{\lambda_1\lambda_2}{\mu}\left\{\left(\frac{\lambda_2}{(\mu-(1-\varphi_{12})\lambda_1-\varphi_{21}\lambda_2)^3}\right) + \left(\frac{\lambda_2}{(\mu-\varphi_{12}\lambda_1-(1-\varphi_{21})\lambda_2)^3}\right)\right\} < 0. \tag{4.10}$$

$$\frac{\partial^2 U_1^{\mathrm{LNW}}}{\partial \varphi_{12}\varphi_{21}} = \frac{1}{N}\frac{\lambda_1\lambda_2}{\mu}\left\{\left(\frac{\lambda_1}{(\mu-(1-\varphi_{12})\lambda_1-\varphi_{21}\lambda_2)^3}\right) + \left(\frac{\lambda_1}{(\mu-\varphi_{12}\lambda_1-(1-\varphi_{21})\lambda_2)^3}\right)\right\} > 0. \tag{4.11}$$

$$\frac{\partial^2 U_2^{\mathrm{LNW}}}{\partial \varphi_{21}\varphi_{12}} = \frac{1}{N}\frac{\lambda_1\lambda_2}{\mu}\left\{\left(\frac{\lambda_2}{(\mu-(1-\varphi_{12})\lambda_1-\varphi_{21}\lambda_2)^3}\right) + \left(\frac{\lambda_2}{(\mu-\varphi_{12}\lambda_1-(1-\varphi_{21})\lambda_2)^3}\right)\right\} > 0. \tag{4.12}$$

単調増加性を示すため，陰函数定理を用いて $\frac{\varphi_{12}}{\varphi_{21}}$, $\frac{\varphi_{21}}{\varphi_{12}}$ を式 (4.9)-(4.12) により以下に示す．

$$\frac{\varphi_{12}}{\varphi_{21}} = -\frac{\left(\frac{\partial^2 U_2^{\mathrm{LNW}}}{\partial \varphi_{21}\varphi_{12}}\right)}{\left(\frac{\partial^2 U_1^{\mathrm{LNW}}}{\partial \varphi_{12}^2}\right)} > 0. \tag{4.13}$$

$$\frac{\varphi_{21}}{\varphi_{12}} = -\frac{\left(\frac{\partial^2 U_1^{\mathrm{LNW}}}{\partial \varphi_{12}\varphi_{21}}\right)}{\left(\frac{\partial^2 U_2^{\mathrm{LNW}}}{\partial \varphi_{21}^2}\right)} > 0. \tag{4.14}$$

したがって，2 つのクラウドレットの最適反応関数は単調増加性を持つ．クラウドレットの数が $N \geq 2$ のゲーム Γ^{LNW} の利得関数は，$N = 2$ のゲームの利得関数の線形拡張として考えられるため，ゲーム Γ^{LNW} の全てのクラウドレットの最適反応関数は単調増加性を持つ．よって，ゲーム Γ^{LNW} はただ一つのナッシュ均衡である最適オフロード割合 $\boldsymbol{\varphi}_i^{*\,\mathrm{LNW}}$ を持つといえる．これだけではゲーム Γ^{LNW} に代わる $\boldsymbol{VI}(\Psi, F)$ が唯一解を持つとは限らないが，$\boldsymbol{VI}(\Psi, F)$ の解集合 Ψ はコンパクトで凸な集合であるため，解の唯一性は保証できる [13], [14].

Algorithm 1 Projection Algorithm with Constant Step Size

1: **Initialization:** Choose any Lagrange multipliers $\boldsymbol{\alpha}^0,\ \boldsymbol{\beta}^0,\ \boldsymbol{\xi}^0 \geq 0$, step size $\omega > 0$, and tolerance limit $\epsilon > 0$. Set the index $t = 0$.

2: **Output:** NE of the computation offload game $\boldsymbol{\varphi}^*$.

3: If $\boldsymbol{\alpha}^t$, $\boldsymbol{\beta}^t$, and $\boldsymbol{\xi}^t$ satisfies a desirable tolerance limit: STOP

4: Given $\boldsymbol{\alpha}^t$, $\boldsymbol{\beta}^t$, and $\boldsymbol{\xi}^t$, compute $\boldsymbol{\varphi}^t(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^t, \boldsymbol{\xi}^t)$ as the NE of the GNE problem (4.15)-(4.18) with fixed Lagrange multipliers $\boldsymbol{\alpha} = \boldsymbol{\alpha}^t, \boldsymbol{\beta} = \boldsymbol{\beta}^t$, and $\boldsymbol{\xi} = \boldsymbol{\xi}^t$.

5: Update all the Lagrange multipliers: for all $i,\ j \in C$, compute

$$\begin{aligned}
\alpha_{ij}^{t+1} &= [\alpha_{ij}^t - \omega(\varphi_{ij})]^+,\ i \neq j \\
\beta_{ij}^{t+1} &= [\beta_{ij}^t - \omega(1 - (\varphi_{ij}))]^+,\ i \neq j \\
\xi_{ij}^{t+1} &= [\xi_{ij}^t - \omega(U_i - U_i^0)]^+,
\end{aligned}$$

where $[z]^+ = \max\{0,\ z\}$.

6: Set $t \leftarrow t + 1$; go to Step 3.

4.1.3　ナッシュ均衡 $\boldsymbol{\varphi}_i^{*\,\mathrm{LNW}}$ の導出

最適オフロード割合 $\boldsymbol{\varphi}_i^{*\,\mathrm{LNW}}$ を導出するため，ラグランジュ乗数 $\boldsymbol{\alpha}_i \in \mathbb{R}^{N-1}$, $\boldsymbol{\beta}_i \in \mathbb{R}^{N-1}$, $\boldsymbol{\xi}_i \in \mathbb{R}^N$, $i \in C$ を用いて，クラウドレット i の Karush-Kuhn-Tucker(KKT) 条件をいかに示す．

$$\nabla_{\boldsymbol{\varphi}_i} U_i^{\mathrm{LNW}} + \nabla_{\boldsymbol{\varphi}_i} \left(\boldsymbol{\alpha}_i^{\mathrm{T}} \boldsymbol{\varphi}_i + \boldsymbol{\beta}_i^{\mathrm{T}} (1 - \boldsymbol{\varphi}_i) + \boldsymbol{\xi}_i^{\mathrm{T}} (U_i^{\mathrm{LNW}} - U_i^{0\,\mathrm{LNW}})_{i \in C} \right), \tag{4.15}$$

$$\boldsymbol{\alpha}_i^{\mathrm{T}} \boldsymbol{\varphi}_i = 0, \tag{4.16}$$

$$\boldsymbol{\beta}_i^{\mathrm{T}} (1 - \boldsymbol{\varphi}_i) = 0, \tag{4.17}$$

$$\boldsymbol{\xi}_i^{\mathrm{T}} (U_i - U_i^0) = 0. \tag{4.18}$$

上記の式（4.15–4.18）に基づき，射影勾配法を用いたナッシュ均衡 $\boldsymbol{\varphi}_i^{*\,\mathrm{LNW}}$ の導出法をアルゴリズム 1 に示す [9].

4.2　提案手法 OP, ONP

4.2.1　異なる処理法を用いた利得関数と非協力ゲーム

本論文では，異なる処理法 OP，ONP を用いた提案手法と，これらを比較するための処理法 FCFS において，クラウドレット i の利得関数 $U_i^{\mathrm{FCFS}}, U_i^{\mathrm{OP}}, U_i^{\mathrm{ONP}}$ を提案する．これら利得関数は，それぞれクラウドレット i の満足度を示しており，許容遅延による満足度 E_i と，クラウドレット i に到着するジョブを処理する上で得られる $\sum_{j=1, j\neq i}^{N} \varphi_{ji}\lambda_j b + \sum_{j=1}^{N} \varphi_{ji}\lambda_j b$ から，$\sum_{j=1, j\neq i}^{N} \varphi_{ij}\lambda_i b$ を差し引いた和で示される．各ジョブの平均遅延時間は処理順によって異なるため，許容遅延に基づく満足度の値 E_i は，$E_i^{\mathrm{FCFS}}, E_i^{\mathrm{OP}}, E_i^{\mathrm{ONP}}$ の 3 通りで導出でき，式 (3.1)〜(3.4) を用いて次の 3 通りで表される．

$$E_i^{\mathrm{FCFS}} = D_{\mathrm{rq}} - T_{ii}^{\mathrm{FCFS}} + \sum_{j=1, j\neq i}^{N} (D_{\mathrm{rq}} - T_{ji}^{\mathrm{FCFS}}). \tag{4.19}$$

$$E_i^{\mathrm{OP}} = D_{\mathrm{rq}} - T_{ii,N} + \sum_{j=1, j\neq i}^{N} (D_{\mathrm{rq}} - T_{ij,m} - t_{ij}).\ (1 \leq m \leq N-1) \tag{4.20}$$

$$E_i^{\mathrm{ONP}} = D_{\mathrm{rq}} - T_{ii,1} + \sum_{j=1, j\neq i}^{N} (D_{\mathrm{rq}} - T_{ij,M} - t_{ij}). \tag{4.21}$$

以上を用いて，クラウドレット i の利得関数 $U_i^{\mathrm{FCFS}}, U_i^{\mathrm{OP}}, U_i^{\mathrm{ONP}}$ は以下のように与えられる．ただし，α, β, γ は各項の重みを表す．

$$\begin{aligned} U_i^{\mathrm{FCFS}}(\boldsymbol{\varphi}_i,\ \boldsymbol{\varphi}_{-i}) = &-\alpha \sum_{j=1, j\neq i}^{N} \varphi_{ij}\lambda_i b + \alpha \sum_{j=1, j\neq i}^{N} \varphi_{ji}\lambda_j b \\ &+ \beta \sum_{j=1}^{N} \varphi_{ji}\lambda_j b - \gamma E_i^{\mathrm{FCFS}}. \end{aligned} \tag{4.22}$$

$$\begin{aligned} U_i^{\mathrm{OP}}(\boldsymbol{\varphi}_i,\ \boldsymbol{\varphi}_{-i}) = &-\alpha \sum_{j=1, j\neq i}^{N} \varphi_{ij}\lambda_i b + \alpha \sum_{j=1, j\neq i}^{N} \varphi_{ji}\lambda_j b \\ &+ \beta \sum_{j=1}^{N} \varphi_{ji}\lambda_j b - \gamma E_i^{\mathrm{OP}}. \end{aligned} \tag{4.23}$$

$$\begin{aligned} U_i^{\mathrm{ONP}}(\boldsymbol{\varphi}_i,\ \boldsymbol{\varphi}_{-i}) = &-\alpha \sum_{j=1, j\neq i}^{N} \varphi_{ij}\lambda_i b + \alpha \sum_{j=1, j\neq i}^{N} \varphi_{ji}\lambda_j b \\ &+ \beta \sum_{j=1}^{N} \varphi_{ji}\lambda_j b - \gamma E_i^{\mathrm{ONP}}. \end{aligned} \tag{4.24}$$

また，定常状態のシステムのみを考えるため，オフロード後の利用率が 1 を超えてはならない．よって，利得関数が $[\mu - (1 - \sum_{j=1, j\neq i}^{N} \varphi_{ij})\lambda_i - \sum_{j=1, j\neq i}^{N} \varphi_{ji}\lambda_j] \geq 0,\ i \in C,\ j \in C \setminus \{i\}$ を満たす中でのみ利得関数を定義する．

上記で定義された利得関数 $U_i^{\mathrm{FCFS}}, U_i^{\mathrm{OP}}, U_i^{\mathrm{ONP}}$ を用いて，次節で 3 種類の非協力ゲームを提案し，ナッシュ均衡となるオフロード割合を導出する．

4.2.2　ナッシュ均衡の導出

本節では，処理法の異なる提案手法における非協力ゲーム $\Gamma^{\mathrm{FCFS}}, \Gamma^{\mathrm{OP}}, \Gamma^{\mathrm{ONP}}$ のそれぞれのクラウドレット i のナッシュ均衡 $\boldsymbol{\varphi}_i^{*\ \mathrm{FCFS}}, \boldsymbol{\varphi}_i^{*\ \mathrm{OP}}, \boldsymbol{\varphi}_i^{*\ \mathrm{ONP}}$ の導出方法を示す．Γ^{LNW} と同様に，これらのゲームは GNE 問題となるが，利得関数 $U_i^{\mathrm{FCFS}}, U_i^{\mathrm{OP}}, U_i^{\mathrm{ONP}}$ の凸性の証明は困難である．そこで，離散的な戦略空間 $\mathcal{S} = \{0, 0.01, 0.02, \ldots, 1.0\} = \{0.01 \cdot k \mid k \in \{0, 1, 2, \ldots, 100\}\}$ を定義し，これを用いて得られる全てのクラウドレットの利得行列から，全解探索によりナッシュ均衡を導出する．このとき，解が 2 以上現れる場合，全てのクラウドレットの平均利得 $\bar{U}^{\mathrm{FCFS}}, \bar{U}^{\mathrm{OP}}, \bar{U}^{\mathrm{ONP}}$ をそれぞれ計算することで，この値が最も大きいオフロード割合の組み合わせを最適解とする．

第 5 章

数値解析

第 5 章では，提案手法 LNW（Latency Not-Weighted offloading）と従来手法を比較し，クラウドレット間の負荷分散の有効性を評価する．本章の数値計算を通じて，クラウドレットの処理能力や遅延の影響を分析し，異なるオフロード戦略におけるナッシュ均衡を導出する．LNW がクラウドレット間の利用率の差を最小化し，より効率的な負荷分散を実現するかを検証するとともに，処理順の違いが遅延に与える影響や、オフロードコストの考慮といった課題についても議論する．

5.1 LNW の数値解析

5.1.1 数値パラメータ設定および解析手順

LNW の有効性を示すため，以下の数値解析を Matlab で実施した．クラウドレットの処理能力は $\mu = 10000$ [jobs/s]，伝播遅延 $t_{\text{prop}} = 2$ [ms]，許容遅延を既存研究 CLB [9] に基づき $D_{rq} = 5 \times 10^{-}3$ [s] と定義した．

クラウドレット 2 の到着率 $\lambda_2 = 9000$ [jobs/s]，およびクラウドレット 1 の到着率 λ を 0 [jobs/s] から 9000 [jobs/s] の範囲で設定し，最適なオフロード割合を評価する．クラウドレット 2 の到着率が $\lambda_2 = 9000$ [jobs/s] に固定されると，オフロード前の利用率は 0.9 に固定される．一方，クラウドレット 1 の到着率 λ_1 は 0 [jobs/s] から 9000 [jobs/s] の範囲で設定され，これによりオフロード前のクラウドレット 1 の利用率は 0 から 0.9 に変化する．処理能力 μ は一定であるため，初期の到着率によって初期の利用率が決定される．言い換えれば，初期の利用率の変化を分析することは，初期到着率の変化を分析することと同義である．初期到着率の変化を観察することで，様々なユーザ環境における LNW の有効性を評価することができる．このような条件下で，オフロード後の利用率の差 ρ_{diff} を評価し，以下のように計算される．

$$\rho_{\text{diff}} = \left| \frac{\lambda_2 - \lambda_1}{\mu} \right|. \tag{5.1}$$

初期のクラウドレット 1 の利用率の変化に対する ρ_{diff} の依存性を分析することにより，様々なシステム環境における異なる手法の負荷分散の度合いを評価できる．

5.1.2 解析結果

図 5.1 は，LWN と既存手法 CLB となる Conventional を，オフロード後の 2 つのクラウドレット間の利用率の差の観点から比較した結果を示している．この結果から，提案された遅延重視の利得関数が負荷分散を完全に実現しており，クラウドレット 1 の負荷にかかわらず利用率の差がほぼ 0 で維持されていることが明らかである．ゲーム理論による解は，すべてのプレイヤーにとって戦略変更による利得の差が最小となる妥協点を達成する．よって，提案された遅延にのみ注目した利得関数では，全体の遅延が許容範囲を超えないようにするための変化が最小であることを意味している．したがって，到着プロセスを $M/M/1$ 待ち行列でモデル化し，定常状態環境のみを仮定している場合，

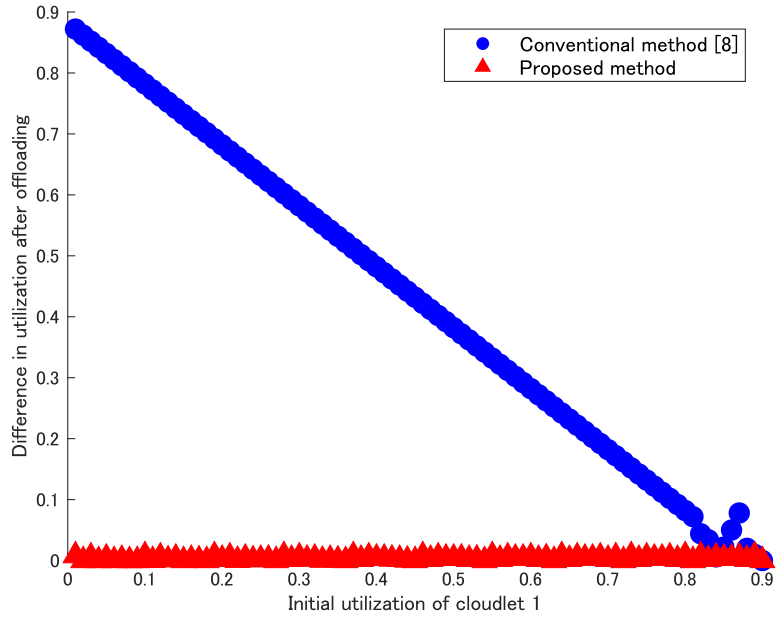

図 5.1: オフロード後の初期利用率 ρ_{diff}

遅延のみに焦点を当てると，最適戦略はすべてのクラウドレットに同じ量のジョブが到着する点となり，これが図 5.1 に示すような完全な負荷分散に繋がる．

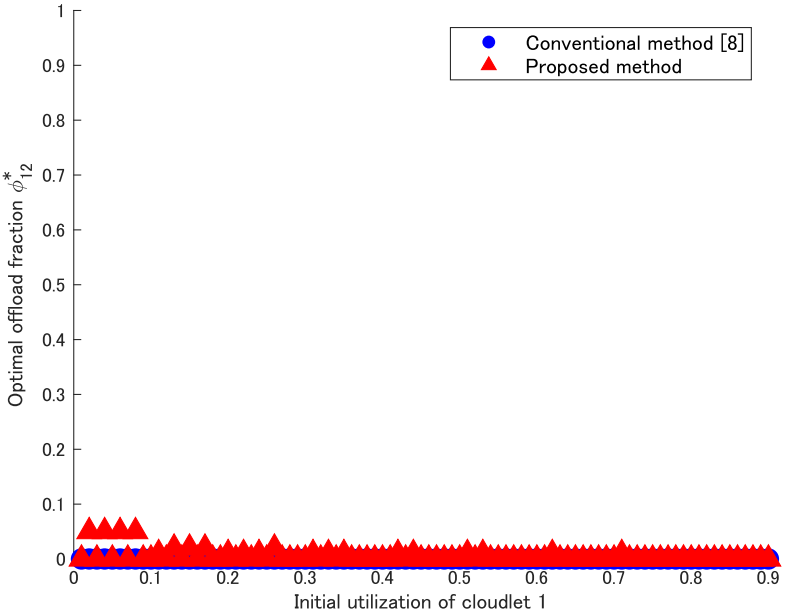

図 5.2: ナッシュ均衡 φ_{12}^*

図 5.2 は，クラウドレット 1 の初期利用率に対する最適なオフロード割合 φ_{12}^* を示している．グラフに示されるように，最適なオフロード割合は LWN と CLB の両方でほぼ 0 となっている．初期利用率が低い場合，提案手法はわずかなずれを示すが，これは Matlab を用いた計算誤差によるものである．この結果から，CLB と LNW のいずれも，初期利用率が低いクラウドレットへのオフロードを行わないことが結論として導かれる．

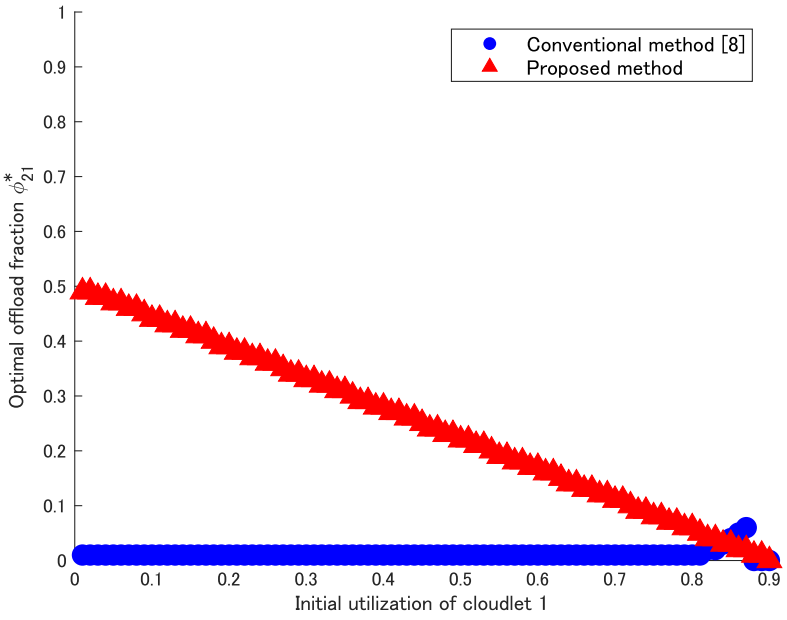

図 5.3: ナッシュ均衡 φ^*_{21}

図 5.3 は，クラウドレット 1 の初期利用率に対する最適なオフロード割合 $\varphi^{*\mathrm{LNW}}_{21}$ を示している．結果から，従来の手法 [9] はオフロードを制限的に行い，最終的には均等に高い初期利用率を持つ環境でオフロードを行うことが分かる．対照的に，提案手法はオフロード割合が 0.5 から線形的に減少し，初期利用率が異なる環境ではより多くのオフロードを行っている．

以上の結果から，LNW は，CLB よりも多くのジョブをオフロードすることができ，これによりクラウドレット間の利用率の差が小さくなり，かつ許容遅延を満たすことが確認できた．よって，負荷を分散するためにクラウドレット間で非協力的にオフロードを行う際の許容遅延の計算として，LNW は CLB よりも有効である．

一方で，LNW は到着するジョブを FCFS で処理しているため，特に処理時間の長いジョブをオフロードした場合に遅延という観点で課題が残っている．また，この解析で用いている利得関数は，許容遅延での評価しか行っていないため，オフロードによるコストや，クラウドレット単体で見たときのジョブの処理数の低下など，マイナスな要素が考慮できていない．

そこで，次節では，FCFS とは異なる処理順である OP と ONP における遅延を用いて LNW による評価を行い，クラウドレットの満足度を考慮した利得関数を用いた非協力ゲームの数値解析を行う．

5.2　OP, ONP の数値解析

5.2.1　数値パラメータ設定および解析手順

本解析では，$N = 2$ のクラウドレット間でのオフローディングについて評価を行った．

まず，オフロードが必要となるトラヒック負荷の調査を行うため，FCFS と OP，ONP の遅延 $T^{\mathrm{FCFS}}_{11}, T_{21,1}, T_{11,2}$ を比較する．ただし，$T^{\mathbf{FCFS}}_{11}$ は，他クラウドレットとオフロードせず FCFS で処理を行った場合における遅延，$T_{21,1}, T_{11,2}$ は，到着するジョブのうち 3 割がクラウドレット 2 からのジョブで，クラス 1 となる場合のクラウドレット 1 における各クラスの遅延である．図 5.4 に，本調査におけるネットワーク図を示す．処理が FCFS で行われる場合の遅延 T^{FCFS}_{11} は，オフロードの有無によらず到着率に依存するため，図 5.4 のような到着流として処理される．一方，OP で処理される場合，オフロードされた到着流が優先されるため，その遅延は $T_{21,1}$ で求められ，オフロードされない遅延

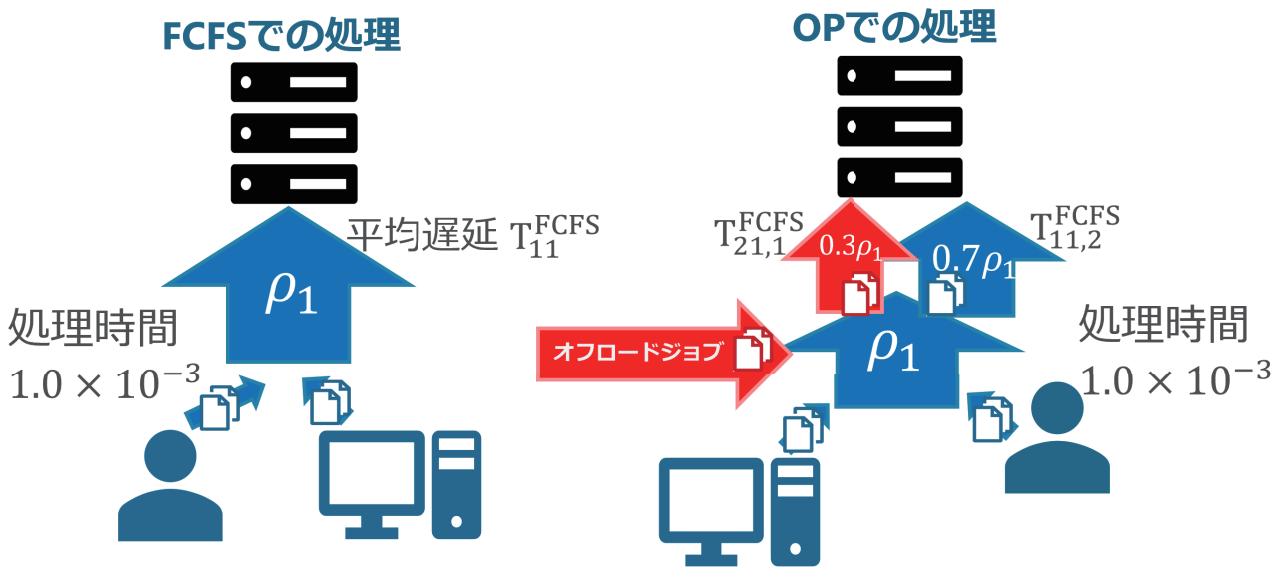

図 5.4: パラメータ調査のためのネットワーク図

は $T_{11,2}$ で示される．このとき，ONP での処理は，この優先度が逆になると考えられるため，到着流の扱いとしては図 5.4 における OP での処理と同様に考えられ，優先されるクラス 1 の遅延も同じ値になる．

平均処理時間 $b = 1.0 \times 10^{-3}$ のクラウドレット 1 の利用率 $\rho_1 = \lambda_1 b$ を 0 から 1 の間で変化させ，これら 3 つの遅延を図 5.5 に示す．

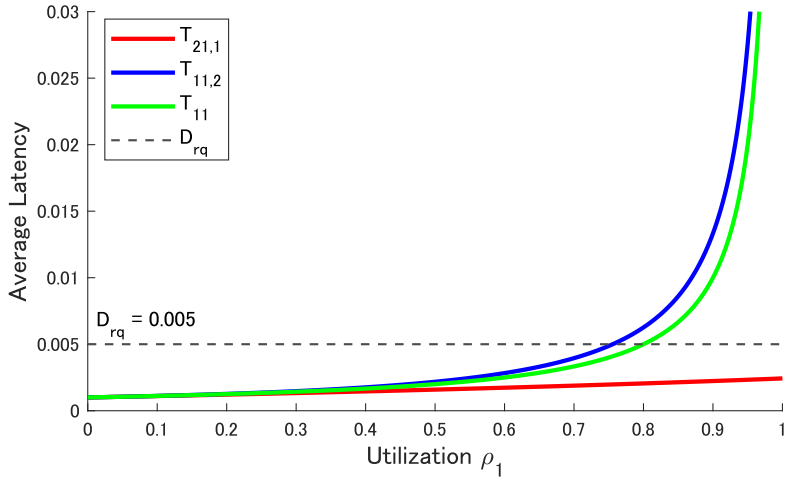

図 5.5: 理論的な遅延 $T_{11}^{FCFS}, T_{21,1}, T_{11,2}$ の比較

許容遅延を CLB [9] に基づき $D_{rq} = 5 \times 10^{-3}$[s] とすると，クラス 1 の遅延は利用率に関わらず許容遅延以下でほぼ一定である．また，利用率が 0.75 以上でクラス 2 のジョブと FCFS で処理されるジョブの両方が許容遅延を超えていることが確認できる．したがって，本環境では利用率 0.75 より高い領域でオフロードが必要になることがわかる．

上記内容を踏まえて，本解析では，初期利用率が 0.75 以上となるようなクラウドレットが存在する間でのオフロードについて，評価を行った．基本的なパラメータは前述のとおりとし，クラウドレット間の往復遅延 $t_{ij} = 2 \times 10^{-3}$ [s]，$\alpha = 1, \beta = 1, \gamma = 100$ と設定することで，利得関数の各項が同等の価値となるようにする．そのもとで，ジョブの到着環境の違いによる影響を調査するため，クラウドレット 1 のオフロード前の利用率を $\rho_1 = 0.25, 0.5.0.75$ の 3 パターンで固定したとき，それぞれにおいてクラウドレット 2 のオフロード前の利用率 ρ_2 を 0.75 から 1 の間で変化させ，

φ^* を導出した．このとき，処理順の違いにおけるクラウドレットの平均利得 $\bar{U} = \frac{1}{N}\sum_{i=1}^{N} U_i$ を，図 5.6–5.8 に示す．

5.2.2　解析結果

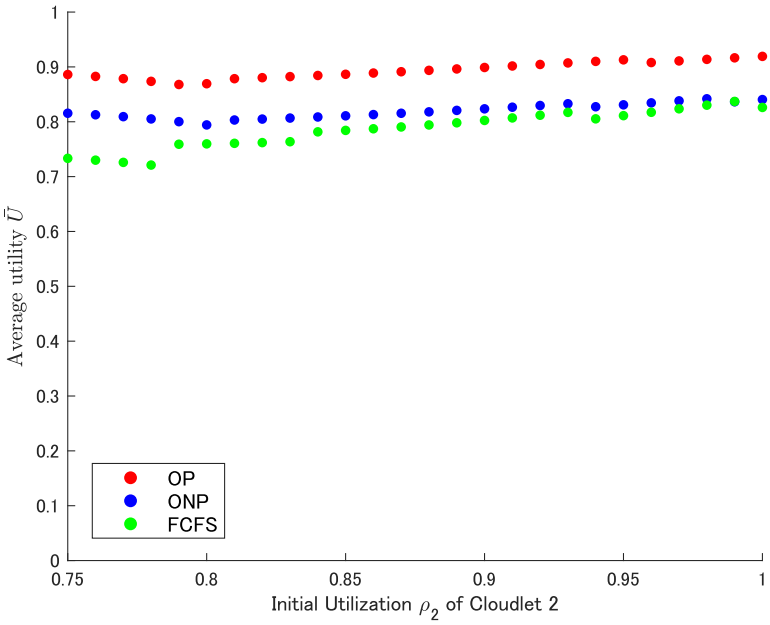

図 5.6: $\rho_1 = 0.25$ の時のクラウドレットの平均利得 $\bar{U}$

図 5.6 は，$\lambda_1 = 250$ [jobs/s] におけるナッシュ均衡の平均利得 $\bar{U}_i$ を示す．クラウドレット 1 の初期利用率は 0.25 であるため，低負荷であり，オフロードを受け入れられる容量が大きい．従って，クラウドレット 2 の初期利用率の増加に伴ってオフロードする量も同様に増え，処理法によらず $\bar{U}_i$ は増加傾向にある．提案手法である OP は，オフロードジョブが優先され遅延が小さくなるため，オフロードが起きやすい環境で最も利得の値が大きくなるとわかる．また，ONP が FCFS を若干上回っている理由として，ONP のナッシュ均衡のオフロード割合が FCFS より少しだけ小さいことが挙げられる．これは，ONP ではオフロードされていないジョブが優先されるため，あえてオフロードせずに自身で処理を行う方が利得にとってよい傾向にあるためである．

図 5.7 は，$\lambda_1 = 500$ [jobs/s] におけるナッシュ均衡の平均利得 $\bar{U}_i$ を示す．結果から，提案手法である OP は他処理順と比較するとつねに上回っていることが確認できる．反面，ONP は利用率の増加に伴い緩やかに減少しており，この環境下では，オフロードジョブを優先しない処理順は有効ではないことが示せた．

図 5.8 は，$\lambda_1 = 750$ [jobs/s] におけるナッシュ均衡の平均利得 $\bar{U}_i$ を示す．これは，クラウドレット 1 の初期利用率が 0.75 と非常に高負荷な環境での結果となる．図 5.5 より得られたオフロードが必要となる $\rho_2 = 0.82$ までは，OP の利得が最も高くなるが，その後 OP と ONP は FCFS より少し小さい値で同じ値を取る．これは，高負荷なクラウドレット間において OP と ONP でオフロードを行うと，どちらの場合でも遅延が大きくなるジョブが発生してしまうためである．よって，オフロードによって利得が大きくなることが無くなるため，オフロードをしないことがナッシュ均衡となる．この場合，本システムは $M/M/1$ 待ち行列モデルでモデル化されているため，ρ_2 が 1 に近づくと遅延は無限大に増加し，許容遅延の項によって利得は無限大に減少する．一方で，FCFS では OP，ONP と比べて優先されるジョブが発生しないため，$\rho_2 = 0.87$ までオフロードが許容される．$\rho_2 = 0.9$ 付近では，FCFS でもオフロードによる利得の改善が見込めなくなり，OP，ONP と同じオフロードをしない利得がナッシュ均衡となる．ρ_2 が 0.95 以上になると，少量のジョブをあえてオフロードすることで，オフロードされないジョブの数が多くなるため，ONP ではオフロードの選択肢が改めて発生し，利得が 0 で抑えられている．

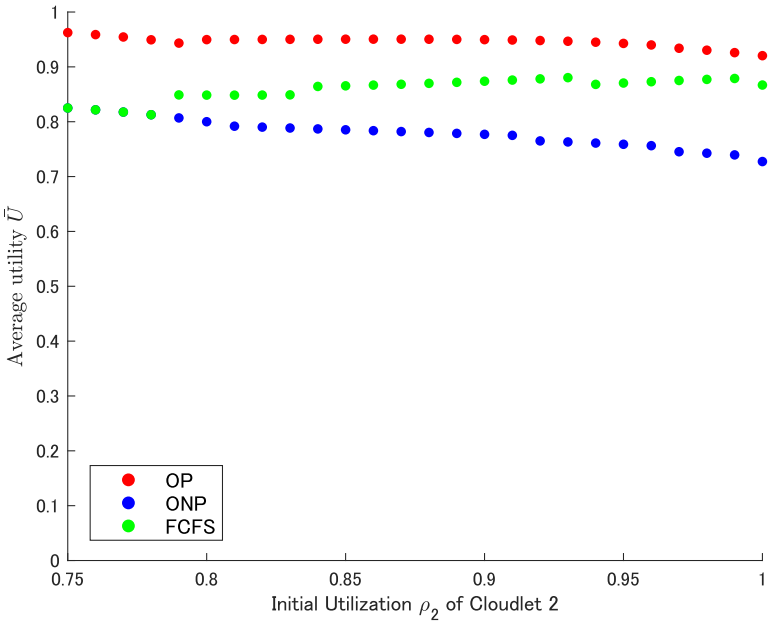

図 5.7: $\rho_1 = 0.5$ の時のクラウドレットの平均利得 $\bar{U}$

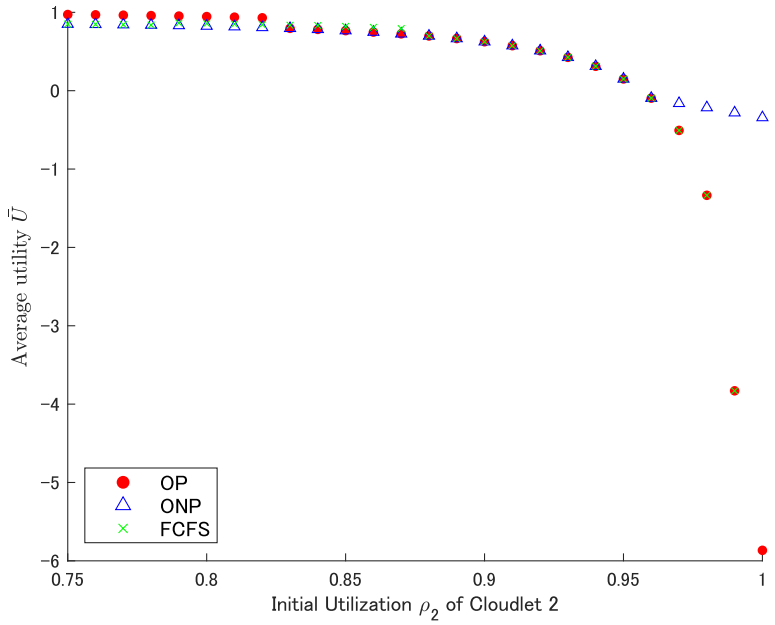

図 5.8: $\rho_1 = 0.75$ の時のクラウドレットの平均利得 $\bar{U}$

本ゲームの利得は，QoS を維持したクラウドレットの満足度を表しており，異なる到着環境においても平均的に高い利得を得られている OP は，オフロードが必然である本解析環境において他手法より優位であり，提案手法の有効性を示すことができた．また，システム内のクラウドレットが全体的に高負荷である場合では，ONP が優位となる場合もあり，到着環境に応じて処理法を変更することで，より拡張性の高いシステムになると考えられる．

図 5.9–5.11 にそれぞれの初期利用率におけるナッシュ均衡 $\varphi_i^{*\,\mathrm{FCFS}}, \varphi_i^{*\,\mathrm{OP}}, \varphi_i^{*\,\mathrm{ONP}}$ を示す．図の赤点が OP，青点が ONP，緑点が FCFS の結果となっている．パラメータ設定どおり，ρ_2 が 0.8 までは遅延が許容遅延以内に収まるため，ρ_1 がどんな値でもオフロードがほとんど起きない組み合わせが均衡点であることが分かる．また，図 5.9 と図

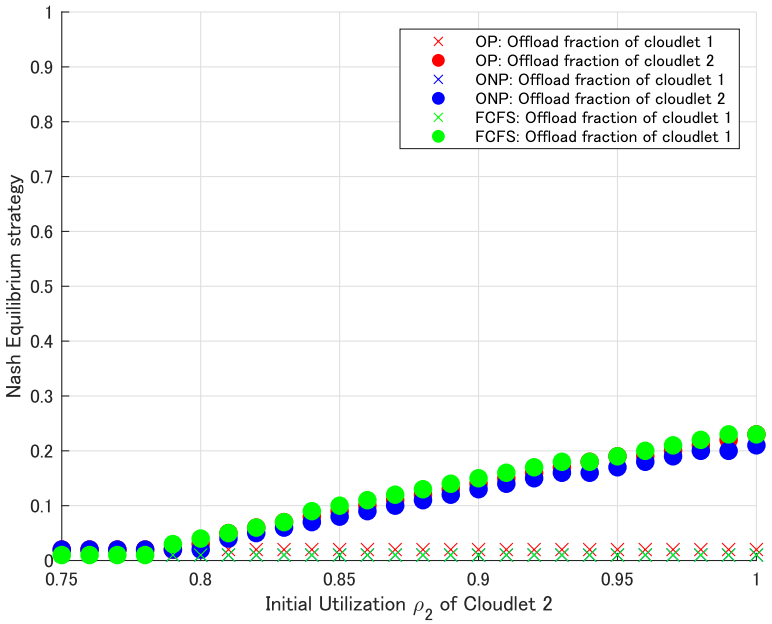

図 5.9: $\lambda_1 = 250$ の時のクラウドレットのナッシュ均衡

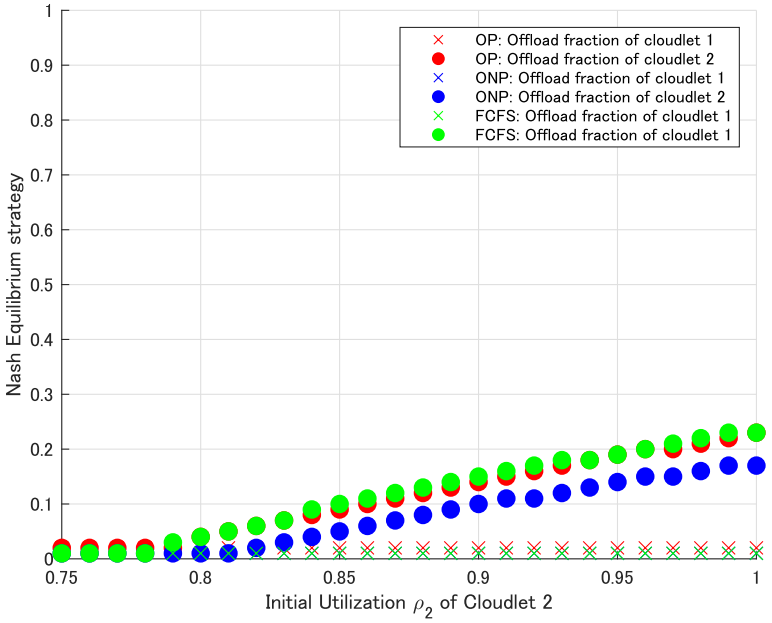

図 5.10: $\lambda_1 = 500$ の時のクラウドレットのナッシュ均衡

5.10 を見れば，初期利用率によって遅延が許容遅延を上回る領域において，オフロードが起き始める地点は処理法によらず同じであり，その増加度合いもほとんど差異はない．ただ，OP と ONP のどちらもクラウドレット 2 が必ず 0.01 程のジョブをオフロードしており，FCFS に比べて戦略の幅が広がっていることが確認できる．図 5.11 は，2 つのクラウドレットの負荷が非常に高い状態であるため，OP ではオフロードによって利得を改善することが難しく，ナッシュ均衡はほとんどの領域で 0 を示す．反対に，限界まで高負荷な状態となると，OP ではかえってほとんどのジョブをオフロードするという均衡状態になることが分かる．到着してくるジョブすべてをオフロードするコストは，現実のシステムでは非常に高いため，このように高負荷な環境では，オフロードした分の利用率にかかる重み

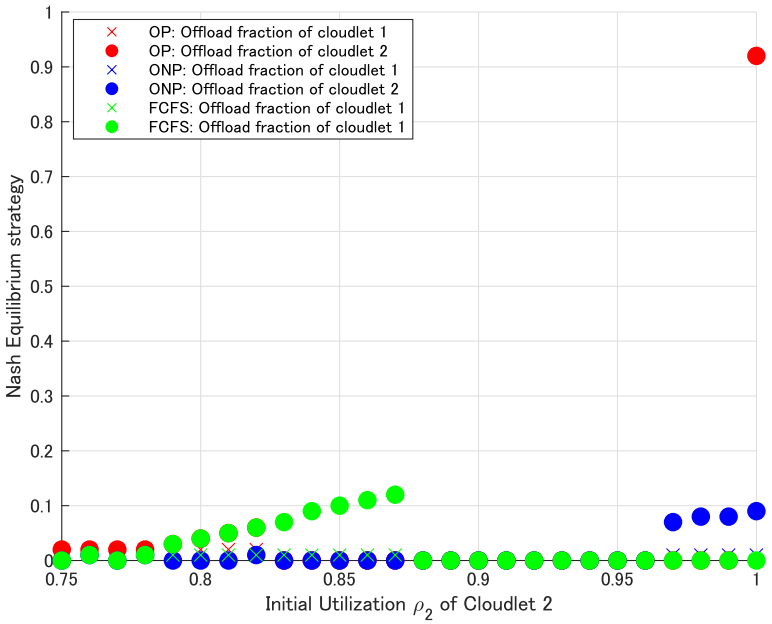

図 5.11: $\lambda_1 = 750$ の時のクラウドレットのナッシュ均衡

α の調整が必要だと考えられる．

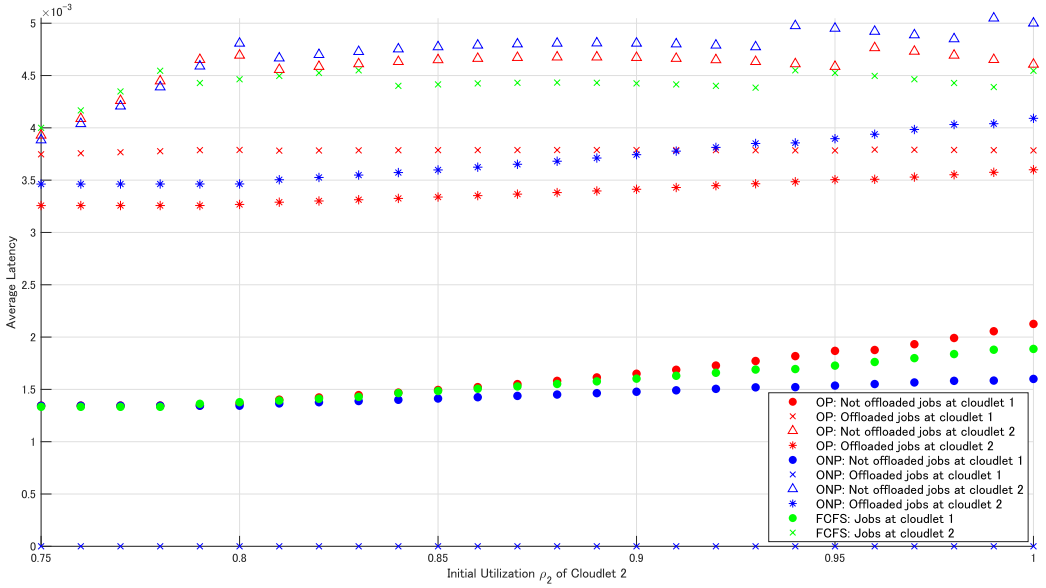

図 5.12: $\lambda_1 = 250$ の時のクラウドレットの平均遅延

図 15.12 と図 5.13 に，ナッシュ均衡における各到着流の平均遅延を示す．平均遅延が 0 となっている到着流は，そのときのオフロード割合が 0 であるため，存在していないことを示している．図 5.12 ではオフロード戦略が有効であるため，初期利用率が増加しても全体として許容遅延以下に平均遅延が収まっていることが確認できる．一方で，図 5.13 の結果では，ナッシュ均衡が存在していても，ONP の手法で許容遅延を上回ってしまっていることが分かる．これは，利得関数において許容遅延の項の重みが他の項と同等程度であるため，ある程度の超過を許してしまっていることが原因である．つまり，ある程度オフロードによって解決できる環境で，ONP を用いる場合，遅延の項の重み

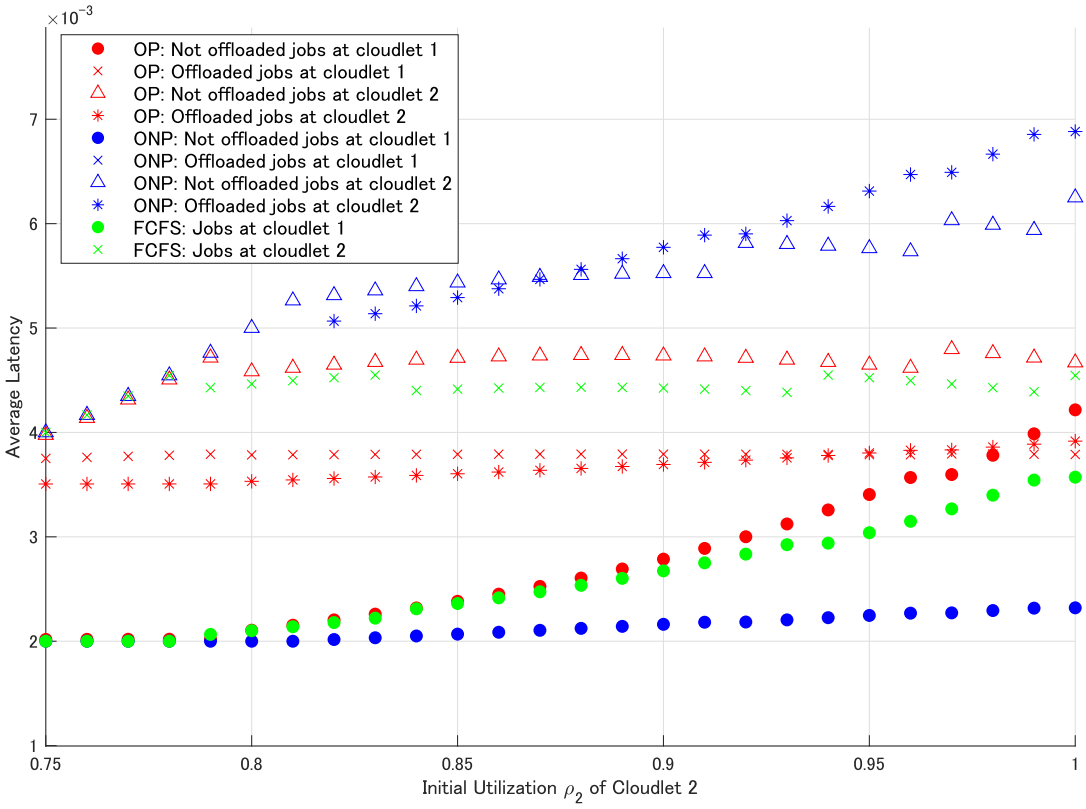

図 5.13: $\lambda_1 = 500$ の時のクラウドレットの平均遅延

γ の調整が必要だと考えられる．

5.3　まとめ

本章では，提案手法 LNW の数値解析を行い，従来手法と比較した際の負荷分散の有効性を検証した．数値計算の結果，LNW を用いることでクラウドレット間の利用率の偏りが大幅に削減され，特に高負荷時には最大遅延を抑えながら効率的な負荷分散が可能であることが示された．

さらに，オフロード後のジョブ処理順序が負荷分散の効率に与える影響についても分析を行った．FCFS を適用した場合，一部の条件下でオフロードされたジョブの遅延が許容範囲を超える可能性があり，オフロードされたジョブを優先処理する OP（Offload Prioritized）や，非オフロードジョブを優先する ONP（Offload Not-Prioritized）の手法を導入することで，処理遅延の改善が見込めることを示した．特に，低から中程度の負荷環境においては OP の適用により，クラウドレットの負荷分散がより安定し，平均利得が向上することが確認された．

一方で，クラウドレットの負荷の状況に応じて，利得関数の重みによるナッシュ均衡解への影響が確認された．これにより，遅延やコストの観点から，実環境では実現しがたいオフロード選択が選ばれるケースが発生するため，今後の課題として，負荷に応じた利得の重み考慮した負荷分散戦略のさらなる最適化が必要である．

これらの結果を踏まえ，第 6 章では，LNW を用いた OP，ONP の適用範囲や実運用における課題を考察し，処理順序の影響を軽減する手法や，利得の重みを考慮した最適な負荷分散戦略の可能性について議論する．

第 6 章

考察

まず，LNW を用いた OP，ONP の適用範囲について考察する．

図 5.5 におけるパラメータ調査の結果から，利用率が 0.8 を超えると平均遅延が許容遅延を超えるため，オフロードが不可欠となることが分かった．一方で，図 5.9-5.11 を見れば，FCFS ではオフロードが全く行われないのに対し，処理法 OP と ONP では 0.8 より低い段階でオフロードを行うことが解となっている．これは，ジョブの処理に優先度を設けることで，到着順で処理を行うよりも，遅延が小さくなるジョブが多くなるためである．したがって，OP，ONP はシステム全体が低負荷でも FCFS より有効であり，十分に適用可能だと考えられる．

システム全体が高負荷な環境では，処理法 OP は他手法よりも平均利得が高く，特にオフロードが起きやすい環境で優位性を示した．また，高負荷時にはオフロードによる利得が減少し，ONP の方が効果的となる場合も確認された．これは，異なる負荷によって，優先されるジョブの量が変わることが理由である．したがって，負荷の状況に応じてジョブの優先順位を変えて，処理法を変えることで，2 つの提案手法のメリットを最大限得られると考えられる．

続いて，利得の重みについて考察する．

システム全体が高負荷な環境では，図 5.11 より，ナッシュ均衡はほとんどの領域で 0 となり，オフロードは行われない．ただ，非常に高負荷な状況では，OP ではかえってほとんどのジョブをお互いにオフロードしてしまっている．これは，オフロードコストや帯域制限などの観点から非現実的な解となっている．このような組み合わせがナッシュ均衡となる原因として，自身がオフロードした分の利用率にかかる重み α の値が，適切に設定されていないことが挙げられる．通常，オフロードするジョブの量に応じてかかるコストは重くなり，オフロードされるジョブが多ければ利用可能な帯域も狭まる．よって，重み α は負荷に応じて変わるべきである．

また，図 5.13 の結果では，ナッシュ均衡が存在していても，ONP の手法でオフロードされたジョブの遅延が許容遅延を上回ってしまっていることが分かる．これは，利得関数において許容遅延の項の重み γ が他の項と同等程度であるため，ある程度の超過を許してしまっていることが原因である．ONP では，オフロードされないジョブが優先される．つまり，少量のオフロードしたジョブの遅延を犠牲に，多くのオフロードされないジョブの遅延を改善しているため，結果として平均利得が高くなり，一部の遅延が許容遅延を上回っていても，ナッシュ均衡解になっている．実環境では，許容遅延は守られるべき閾値であるため，許容遅延の項の重み γ を厳しく設定するべきである．

以上の考察から，本論文の提案手法である LNW を用いた異なる処理法 OP，ONP は，負荷の環境が異なる場合でも従来の FCFS による処理法より優れており，負荷に応じて適用させることで効果的に負荷分散され，平均利得が高くなる．しかし，それぞれの利得関数で用いられている重みは，負荷に合わせた最適化によって，より実環境に合わせたオフロード組み合わせの導出が可能となる．

第 7 章

結論

本論文では，システムの QoE とクラウドレットの満足度を両立させたオフロードを目的とし，許容遅延による遅延の評価に着目した評価法である LNW の評価と，オフロード後の到着流に着目した，非割込み優先規律に基づくオフロードしたジョブを優先する処理法 OP，そしてオフロードしていないジョブを優先する処理法 ONP の 2 つを提案した．また，これらの手法を用いて，クラウドレット間のオフロードを非協力ゲームでモデル化し，それぞれの最適オフロード戦略となるナッシュ均衡を導出することができた．

結果として，許容遅延による遅延の評価のみによって完全なオフロードが達成された．したがって，LNW は負荷分散という観点で，従来の評価手法よりも効果的であることが示せた．さらに，オフロードが起きやすい環境では，オフロードしたジョブが優先される OP によって，各クラウドレットの平均的な利得が他処理法と比較して最も高くなることが示せた．システム全体が高負荷となるような環境では，オフロードしないジョブを優先する NOP によって利得を保つことが可能となることが示せた．また，負荷に応じてこれらの手法を適用させることで，効果的に負荷分散され，平均的な利得がより高くなることが考察された．これらのゲームにおける平均利得は，システムの QoE を考慮したクラウドレットの満足度を示すため，提案手法 OP，ONP によって，これまで用いられていた FCFS による処理よりも，より優れたオフロード選択が達成されていると言える．

今後の課題として，負荷に応じた利得関数の適切な重みの設定によって，より現実的なクラウドレットの選択と動向の調査が可能になると考えられる．また，待ち行列理論を用いたモデル化では，確率統計学的平均値しか導出できないため，シミュレーションによるミクロな視点での数値解析や，動的な処理順変更を考慮したオフロード選択の解析が，今後の展望として挙げられる．

謝辞

本論文を執筆するにあたり，2 年間ご指導いただいた宮田先生と上岡先生に感謝いたします．また，これまでともに切磋琢磨しあった研究室の仲間たちにも感謝と，今後のさらなる活躍を祈っています．

参考文献

[1] E. Wong, M. Pubudini Imali Dias, and L. Ruan, “Predictive resource allocation for tactile internet capable passive optical lans,” in *Journal of Lightwave Technology*, vol. 35, no. 13, pp. 2629–2641, 2017.

[2] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” in *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[3] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, “Efficient algorithms for capaci- tated cloudlet placements,” in *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 10, pp. 2866–2880, 2016.

[4] Q. Fan and N. Ansari, “Cost aware cloudlet placement for big data processing at the edge,” in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2017.

[5] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, “All one needs to know about fog computing and related edge computing paradigms: A complete survey,” in *Journal of Systems Architecture*, vol. 98, pp. 289–330, 2019.

[6] H. Cao and J. Cai, “Distributed multiuser computation offloading for cloudlet- based mobile cloud computing: A game-theoretic machine learning approach,” in *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 752–764, 2018.

[7] Q. Fan and N. Ansari, “Application aware workload allocation for edge computing- based iot,” in *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2146–2153, 2018.

[8] Y. Jiang, “A survey of task allocation and load balancing in distributed systems,” IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 2, pp. 585–599, 2016.

[9] S. Mondal, G. Das, and E. Wong, “A game-theoretic approach for non-cooperative load balancing among competing cloudlets,” in *IEEE Open Journal of the Communications Society*, vol. 1, pp. 226–241, 2020.

[10] Y. Yokota, and S. Miyata, “Game Theoretic Approach for Non-Cooperative Load Balancing between Local Cloudlets,” *Nonlinear Theory and Its Applications, IEICE*, vol.15, no.2, pp.473-484, 2024.

[11] M. Jia, W. Liang, Z. Xu, and M. Huang, “Cloudlet load balancing in wireless metropolitan area networks,” in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, 2016.

[12] D. Zhang, Y. Ma, C. Zheng, Y. Zhang, X. S. Hu, and D. Wang, “Cooperative- competitive task allocation in edge computing for delay-sensitive social sensing,” in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 243–259, 2018.

[13] F. Facchinei, A. Fischer, V. Piccialli, “On generalized Nash games and variational inequalities,” *Operations Research Letters*, vol.35, pp.19-164, 2007.

[14] G. Scutari, D. Palomar, F. Facchinei, J. Pang, “Convex Optimization, Game Theory, and Variational Inequality Theory,” *IEEE Signal Processing Magazine*, vol.27, pp.35-49, 2010.

[15] T.Takine, “Lecture paper on Communication Network Engineering (Ver. 3.15), ” Osaka University, 2024. [Online]. Available: http://www2b.comm.eng.osaka-u.ac.jp/ takine/tmp/v3-15.pdf

研究実績

論文誌

[1] Y. Yokota, and S. Miyata, “Latency focused load balancing method between cloudlets using game theory,” *Nonlinear Theory and Its Applications, IEICE*, vol. 15, no. 2, pp. 473-484, 2024.

国際会議（査読あり）

[2] Y. Yokota, and S. Miyata, “Game Theoretic Approach for Non-Cooperative Load Balancing between Local Cloudlets,” *Proc. of NOLTA’23*, paper ID (6207), 2023.

国内会議（査読なし）

[3] 横田侑紀, 宮田純子, “スマートシティにおけるクラウドレットの優先順位付けジョブオフローディングシステムの提案,” 信学技報, vol. 123, no. 452, pp. 59-63, 2024 年.

[4] Y. Yokota, and S. Miyata, “Prioritized Job Offloading in Cloudlet System for Smart City,” IEICE GlobalNet Workshop 2024 in Hiroshima, Poster no.A–26, 2024.