

高効率で省電力な IoT・ビッグデータ処理基盤

研究代表者 甲藤 二郎
(基幹理工学部 情報通信学科 教授)

1. 研究課題

各種センサ、スマートフォン、自動車、列車、移動ロボット等、IoT (Internet of Things) デバイスは多岐に渡り、またそのセンシングデータも、加速度、温度等の小容量データから高精細映像等の大容量データまで、あるいは、SNS (Social Networking Service) 等の遅延要求の緩いものから自動運転、Industry 4.0 等の遅延要求の厳しいものまで、これまた多岐に渡っている。また、災害時や大きなイベント時にはバースト的に大量のデータが発生するが、どのような環境下でも安全かつ高信頼なセンシングを確保すると共に、適切な情報を抽出できるデータ処理技術の確立が求められている。さらには、東日本大震災時の反省に基づき、電力供給が限られた中でも長時間動作可能な省電力化技術の確立が求められると共に、COVID-19 対策に貢献する情報通信処理技術の開発が期待されている。

2021 年度の研究成果としては、第一に、深層学習を活用した奥行き推定技術を開発し、360 度カメラで撮影した画像中の人物を検出しては、奥行き推定結果から撮影場所の歩行者密度を推定するレーダーマップシステムの開発を行った。さらに、提案システムの Kubernetes による並列実行を実現し、効率的なリソース割り当てに関する検討も行った。第二に、課題として Massive MIMO におけるパイロット信号の最適配置問題を取り上げ、max k-cut 問題として定式化を行い、量子アニーリングマシンであるデジタルアニーラによって最適解を求めることを試みた。その結果として、特に中規模～大規模な通信環境において、量子アニーリングによって良好な解が得られることを示した。

2. 主な研究成果

2.1. 360 度カメラ及び深層学習を利用した歩行者レーダーマップシステムの性能評価

2.1.1. はじめに

近年、インフラの老朽化問題や、地震などの自然災害などの影響から都市モニタリングの重要性が高まっている。また COVID-19 の感染拡大に伴い群集エリアの特定の需要が高まっている。群集レベルや歩行者密度を推定する手法として主に Wi-Fi の接続履歴や LiDAR を利用するセンサーベースのアプローチとコンピュータビジョンベースのアプローチが挙げられる。これらの研究へ貢献するため、都市モニタリングの新たなプロトタイプとして道路や歩道における歩行者密度レベルをレーダーマップとして可視化するシステムを提案する。またシステムを異なる実装手法によって実装し効率的な CPU リソース割り当て実現のための評価実験も行う。

2.1.2. 歩行者レーダーマップシステム

2.1.2.1. プロトタイプシステム

e-bike に搭載した 360 度カメラを用いた歩行者レーダーマップシステムを提案する。図 2. 1. 1 にカメラやセンサを取り付けた電動アシスト自転車（e-bike）とクラウドサーバを利用した道路インフラモニタリングシステムのプロトタイプ構成図を示す。道路や歩道上の画像をタイムスタンプや GPS による位置情報と共に収集していく。この時、e-bike によって収集された画像は LTE や 5G 通信を利用し、クラウドサーバへ送信され、人の密度推定、道路の損傷検出、段差検出など深層学習による画像処理技術をクラウドサーバにて実施する。

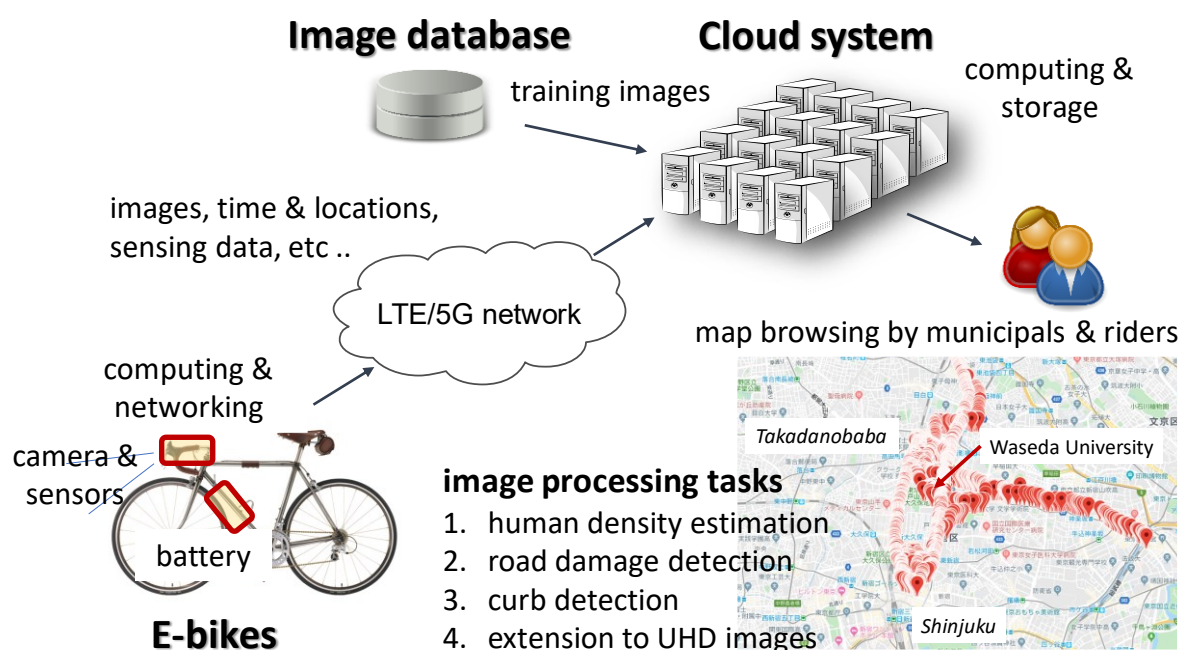


図 2. 1. 1: プロトタイプ構成図

2.1.2.2. 歩行者レーダーマップへの可視化

プロトタイプでは、路上に存在する歩行者の位置を 360 度レーダーマップとして可視化する。本システムでは、1 枚の 360 度画像からこのレーダーマップの自動生成を行う。その処理の流れを以下に示す。

1. **前処理:** 360 度画像は通常、歪みを持った正距円筒図法と呼ばれる形式で保存されているため、通常の 2 次元画像に変換する必要がある。そこで前処理として、まず 360 度画像を水平視野角 90 度ごと（前、後、左、右方向の画像）に分割しそれぞれも画像を Gnomonic projection を用いて変換を行う。
2. **人物検出:** 前処理の適用後、分割された画像（前後左右方向の画像）ごとに人物検出を行い、バウンディングボックス座標の抽出を行う。人物検出アルゴリズムとして本プロトタイプでは YOLOv3 を利用している。

3. **距離推定：** 次にカメラから歩行者までの距離を推定する。距離推定は、図 2.1.2 に示すように、Disnet[1]、Ama-Net[2]、VGG-16[3]の三つの既存のモデルを組み合わせた独自の深層学習ベースの距離推定モデルを適用する。バウンディングボックスのサイズや画像のぼけ、画像特徴量の3つの特徴量を用いて回帰を行う。
4. **角度推定：** 前の処理と同時に、このステップでは、カメラからの歩行者までの方向を推定する。画像上におけるバウンディングボックスの x 座標を用いて、極座標空間における歩行者の位置 (d, θ) における θ を推定する。なお、距離 d は前のステップで推定するものとする。角度推定はSVR (Support Vector Regression)を用いて行い、入力バウンディングボックスの x 座標、出力は角度 θ としている。
5. **レーダーマップ可視化：** 最後に、検出された歩行者の極座標 (d, θ) を 360 度のレーダーマップとして可視化する。このレーダーマップから、式(1)によって密度を算出する。

$$\text{密度} = [\text{検出された歩行者数}] / [\text{レーダー円の面積}] \quad (1)$$

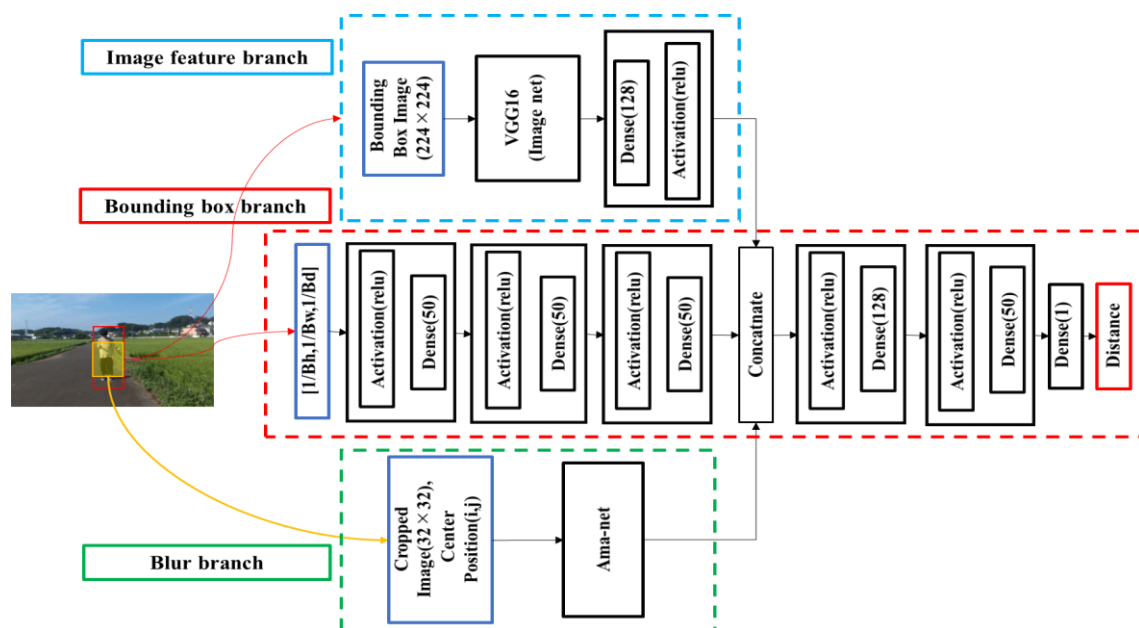


図 2.1.2: 距離推定ネットワーク

2.1.2.3. 歩行者レーダーマップシステム精度評価実験

KITTI データセットを用いて、歩行者レーダーマップと歩行者密度推定の精度評価を行う。データセットにおける 340 枚の時系列画像を評価用データセットとして利用する。歩行者密度は式(1)を用いて算出を行う。図 2.1.3 に生成された歩行者レーダーマップの一例を示す。また図 2.1.4 に 340 枚の時系列画像を対象とした密度推定結果を示す。一部の画像において False-Positive や False-Negative が見受けられたが、群集レベルの傾向を大まかに把握することができることを確認した。



Location A (KITTI)



Location B (KITTI)

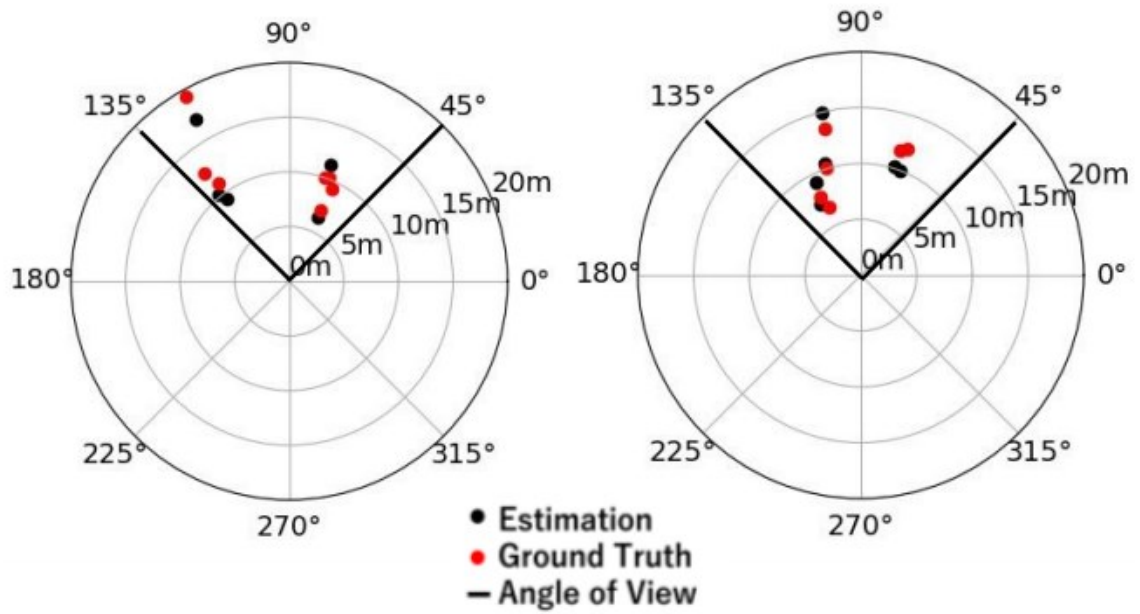


図 2.1.3: 歩行者レーダーマップ推定結果 (KITTI データセット)



図 2.1.4: 密度推定結果 (西早稲田キャンパス周辺)

2.1.3. CPU リソース割り当て効率化のための評価実験

2.1.3.1. 処理ブロック分割

はじめに、前節で提案した歩行者レーダーマップシステムを表 2.1.1 のように 6 つの処理に分割を行った。

表 2.1.1: 歩行者レーダーマップシステムにおける全プロセス

PID	名前	処理内容
1	Image	VGG を用いて画像特徴量を抽出
2	Bbox	歩行者画像のサイズから特徴量を抽出
3	Blur	Ama-net を用いてぼけ特徴量を抽出
4	Regression	3 つの特徴量を concat し距離を回帰
5	Angle	カメラから歩行者までの角度を推定
6	Map	推定結果からレーダーマップを出力

2.1.3.2. 実装手法

次に、分割した 6 つの処理ブロックを Docker によってコンテナ化し、Kubernetes 上で次の 2 パターンによる手法でデプロイを行った。

1) Microservice

Microservice による実装では各処理を別々の Node (物理マシン) 状に pod としてデプロイする。各 pod は Pub/Sub モデル (Apache Kafka) を用いてデータのやりとりを行う。Microservice ではそれぞれの pod に対して割り当てるリソース量を柔軟に変えることができる。またそれぞれの処理が独立して実行されるため並列処理が可能となる。レーダーマップを出力するまでの合計処理時間 T_{mi} と合計使用 CPU リソース量 R_{mi} は (2) ~ (4) 式によって表すことができる。ここで PID= j 番のプロセス処理時間 t_j は事前実験によって求めたリソース量 r_j と処理時間のプロファイリング結果 f_j を用いて算出する。また N_d は検出された歩行者の数を示している。

$$T_{mi} = N_d * \max(\max(t_1, t_2, t_3) + t_4, t_5) + t_6 \quad (2)$$

$$t_j = f_j(r_j) \quad (3)$$

$$R_{mi} = \sum_{j=1}^6 r_j \quad (4)$$

$$1 \leq j \leq 6, j \in pid, r_j \in resource$$

2) Monolith

Monolith による実装では 6 つの全ての処理ブロックを 1 つの Node に対してデプロイを行うため各 pod に割り当てられるリソース量がすべて等しくなる、Monolith 実装における合計処理時間 T_{mo} と合計使用 CPU リソース量 R_{mo} は (5) ~ (7) 式によって表される。

$$T_{mo} = N_d * \sum_{i=1}^5 t_i + t_6 \quad (5)$$

$$t_j = f_j(r_j) \quad (6)$$

$$R_{mo} = r_j \quad (7)$$

$$1 \leq j \leq 6, j \in pid, r_j \in resource$$

2.1.3.3. 実装手法によるリソース割り当て評価実験

1) Microservice における最適リソース割り当てパターンの算出

まず Microservice における各 pod への CPU 割り当てパターンのうち最も効率的な組み合わせの探索を行う。そこで式(2)～(4)を用いて数値計算により CPU リソース量の合計 R_{mi} と処理時間 T_{mi} の関係をシミュレーションする。その結果を図 2.1.5 に示す。この結果より最も効率的なリソース割り当てパターンを求めた(赤線)。

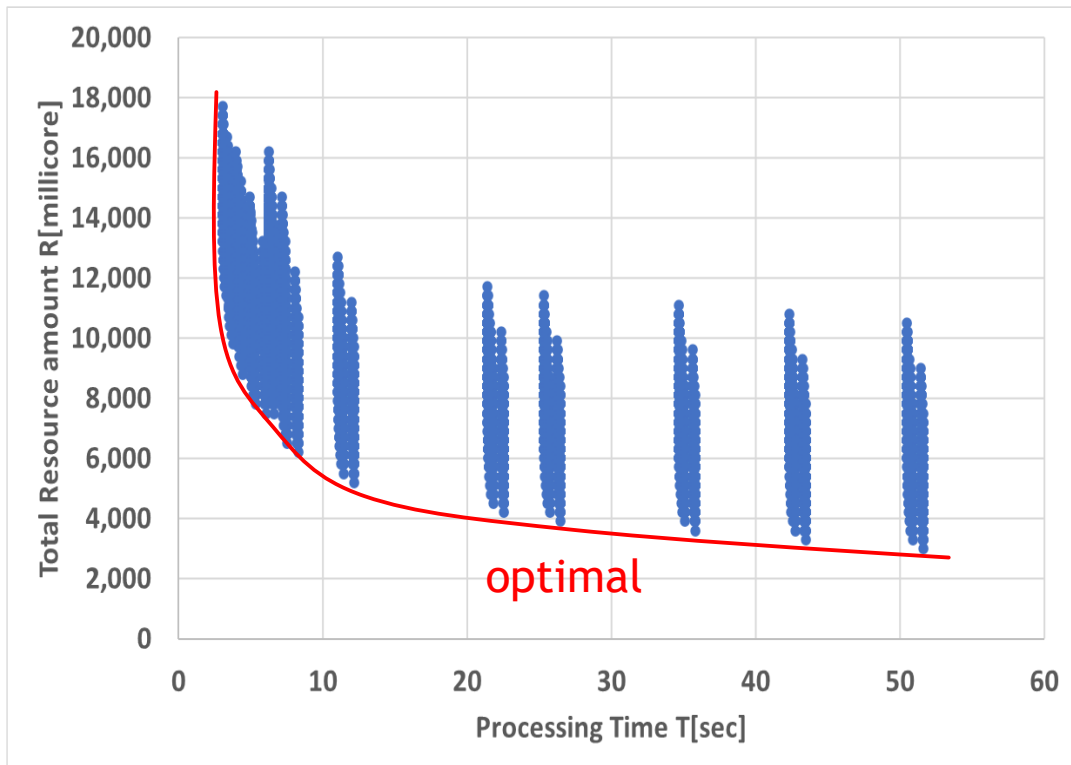


図 2.1.5: CPU リソース全探索

2) 数値計算による CPU リソース比較

次に(5)～(7)式を用いて Monolith における CPU リソース量の合計と処理時間を数値計算によりシミュレーションし、Microservice(図 2.1.5 の赤線)との比較を行った。その結果を図 2.1.6 に示す。図 2.1.6 より特性変化点より左側の領域では Microservice によって処理時間、リソース量を抑制できていることを確認した。またパイプライン処理による効果を考慮すると、より Microservice による抑制効果が表れると考えられる。

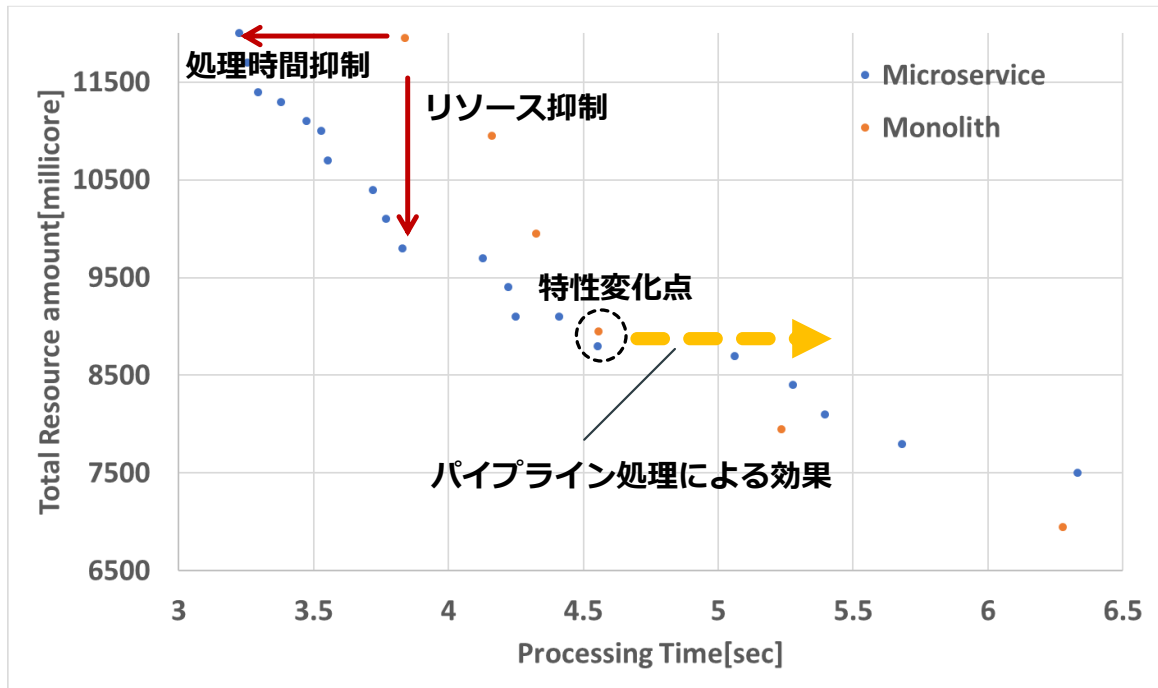


図 2.1.6: 数値計算による Monolith、Microservice 比較結果

2.1.4. まとめ

本稿では歩道上の人密度を推定するための歩行者レーダーマップシステムの提案を行った。またシステムを異なる手法により実装し、効率的なCPUリソース割り当てを実現するための比較評価を行った。

2.1.5. 参考文献

- [1] M. A. Haseeb, et al., “Disnet: A novel method for distance estimation from monocular camera,” Planning Perception and Navigation for Intelligent Vehicles, IROS 2018.
- [2] M. Kashiwagi, et al., “Deep Depth From Aberration Map”, ICCV 2019.
- [3] K. Simonyan, et al., “Very deep convolutional networks for large-scale image recognition,” ICLR 2015.

2.2. イジングマシンを用いた Multi-cell Massive MIMO におけるパイロット配置の最適化

2.2.1. はじめに

Massive multiple-input multiple-output (MIMO) では、パイロット汚染の課題が顕在する。パイロット汚染とは、所望セル内の端末のみならず、近傍セルの端末からのパイロット信号を不必要に受信してしまうことにより、本来受信すべき端末のパイロット信号が干渉を被る現象である。パイロットの汚染が強い場合、基地局は端末とのチャネル状態を正しく推定できず、アップリンク信号の復調やダウンリンク通信でのビームフォーミングの精度が落ちてしまう。近年、パイロット汚染を権限するための研究として、パイロット配置を組み合わせ最適化問題として捉え、端末の通信容量を最大化する手法が多く提案されている。その一方で、それらの手法の多くは局所解に収まりやすく、特に大規模な通信シナリオでは十分な最

適配置が行えない。

本節では、大規模な通信シナリオでのパイロット汚染を解決するために、組み合わせ最適化問題に特化したイジングマシンを用いてパイロット配置を行う手法を提案する。シミュレーション実験の結果、提案手法は従来手法よりも最適なパイロット配置を実行し、パイロット汚染を軽減できることが示された。特に、端末数と基地局数の多い大規模な場合において、従来手法よりも端末の通信容量を大幅に改善することが確認できた。

2.2.2. 関連研究

パイロット汚染を軽減するため、パイロット配置手法に関する研究が盛んに行われている。例えば、Smart Pilot Assignment (SPA) [1]は、セル内でチャネル品質が悪い端末から順に、セル間干渉の小さいパイロット信号を割り当てる。また、近年は、グラフモデルに基づくパイロット配置手法が提案されている。[2]では、重み付きグラフ彩色問題に基づくパイロット配置手法 (WGC-PD: Weighted-Graph-Coloring-based Pilot Decontamination) を提案している。この手法は端末間のパイロット汚染の潜在的な強度をグラフの重み付きエッジにて表現している。さらに、Fractional pilot reuse and max k-cut-based pilot decontamination scheme (FPR-MKC) [3]でも同様に重み付きグラフにモデリングを行なっている。[3]では、max k-cut 問題に基づいて目的関数が定式化される。Max-k cut は、切断されたエッジの重みの合計値が最大となるようにグラフを分割する問題である。Max k-cut 問題を応用することにより、強度のパイロット汚染が発生し得るリンクは切断し、端末同士で異なるパイロット信号を使用するようにグループ分割される。この手法では、切断するエッジの重みの合計値が最大化することを目的として、最適化問題を解いている。

2.2.3. 提案手法

本稿では、イジングマシンを活用したパイロット配置手法を提案する。本手法では、セル数を L 、各セル内の端末数を K 、パイロット系列長を τ とし、 $L \times K$ の端末に対して τ 個のパイロット信号を最適に割り当てることを目的とする。提案手法は、図 2.2.1 に示すような 3 つの処理によって構成される。

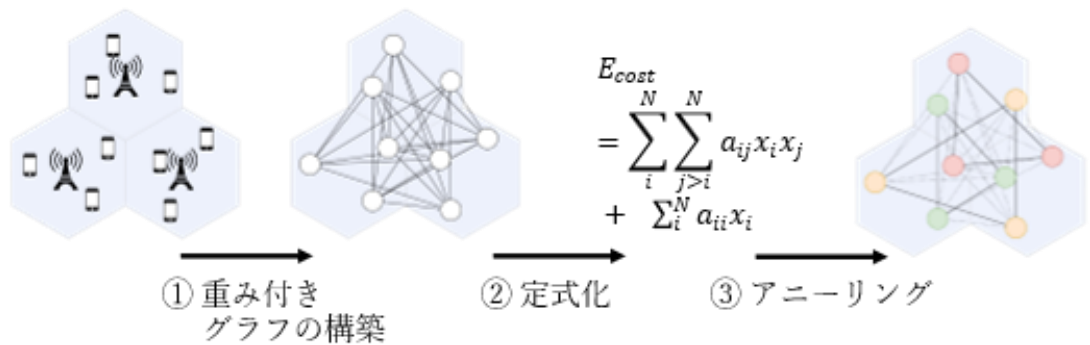


図 2.2.1: 処理フロー図

重み付きグラフの構築

[2, 3]に基づき、パイロット汚染の潜在的な強度をグラフで表現する。ただし、ノードは端末

を、重み付きエッジはパイロット汚染の強度を示している。重みの値が大きいほど、端末同士が非直交なパイロット信号を使用した際に深刻なパイロット汚染を引き起こし、通信容量を低下させる。端末 $UT_{(j,k)}$ と端末 $UT_{(j',k')}$ のエッジの重みは式(1)で計算される。ただし、 $\beta_{(j,k),l}$ はセル j 内の端末 k とセル l 内の基地局の large-scale fading coefficient を示している。

$$w_{(j,k),(j',k'),j \neq j'} = \frac{\beta_{(j',k'),j}^2}{\beta_{(j,k),j}^2} + \frac{\beta_{(j,k),j'}^2}{\beta_{(j',k'),j'}^2} \quad (1)$$

Max k-cut に基づいたパイロット配置の定式化

本節では、[3]に従い、パイロット配置を max k-cut 問題に基づく組み合わせ最適化問題としてモデル化を行う。イジングマシンでこの問題を解くためには、コスト関数を Quadratic Unconstrained Binary Optimization (QUBO) 形式に定式化する必要がある。定式化したコスト関数を式(2)に示す。ただし、 $x_{s,(j,k)} \in \{0,1\} | 1 \leq s \leq \tau, 1 \leq j \leq L, 1 \leq k \leq K\}$ は、端末 $UT_{(j,k)}$ に s 番目のパイロット信号が割り当てられる場合に1、割り当てられない場合に0となる。また、 $\alpha_{\{1,2,3\}}$ は各項のペナルティ係数を表す。

$$\begin{aligned} E_{cost}^{proposal} &= \alpha_1 \sum_p \sum_{(j,k) \langle j',k' \rangle, j \neq j'} w_{(j,k),(j',k'),j \neq j'} x_{s,(j,k)} x_{s,(j',k')} \\ &+ \alpha_2 \sum_{(j,k)} \left(\sum_s x_{s,(j,k)} - 1 \right)^2 \\ &+ \alpha_3 \sum_s \sum_j \left(\sum_k x_{s,(j,k)} - 1 \right)^2 \end{aligned} \quad (2)$$

イジングマシンを用いたアニーリングによる最適解の導出

イジングマシンを用いて、式(2)を最小化する二次決定変数 $x_{s,(j,k)}$ の組み合わせを求める。本節では、イジングマシンとして富士通製のデジタルアニーラ[4]を用いる。デジタルアニーラによって得られた $x_{s,(j,k)}$ から、端末とパイロット信号の組み合わせに復元する。その後、各端末の属するセルの基地局との SINR と通信容量を組み合わせ結果から算出し、提案手法の評価を行う。

2.2.4. 評価実験

本節では、シミュレーションにおいて、提案手法と従来パイロット配置手法による端末の通信容量の比較評価を行う。

通信容量ヒートマップの比較評価

実験シナリオにおける各端末のアップリンク通信容量の可視化結果を図 2.2.2 に示す。こ

ここで、プロットは端末を表し、色の濃いプロットはパイロット汚染によって通信容量が小さくなってしまっている端末を表している。この結果より、従来手法がセルの端に位置する端末を中心にパイロット汚染を引き起こしているのに対し、提案手法は最適なパイロット配置によって通信容量が小さくなる端末を減らしていることがわかる。

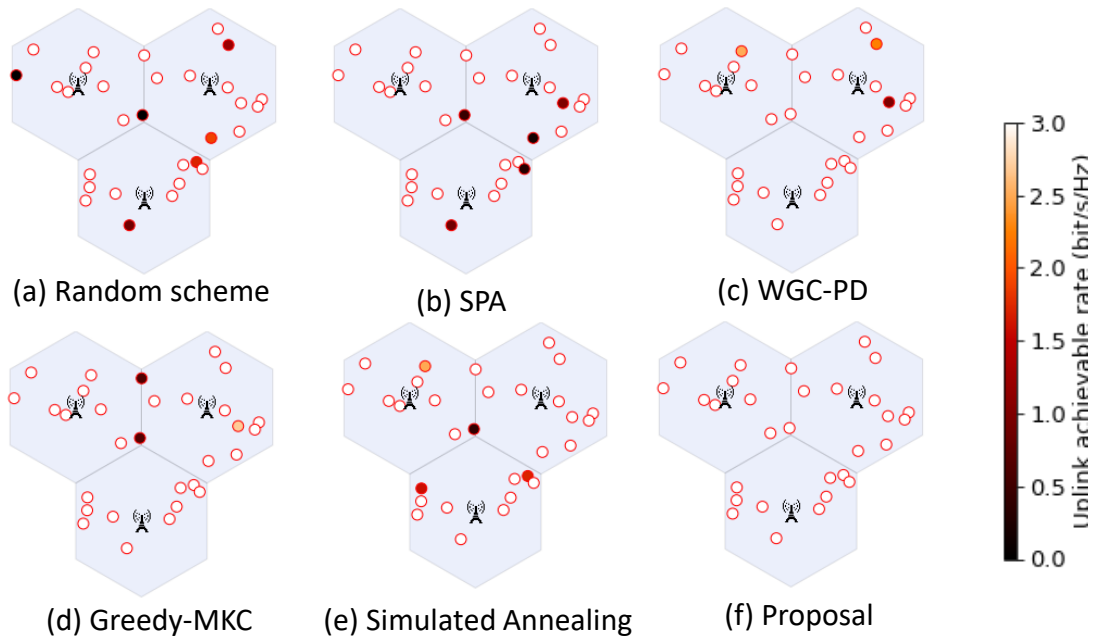


図 2.2.2: 各端末の通信容量のヒートマップ

通信容量ヒートマップの比較評価

最小通信容量の累積分布関数 (CDF: Cumulative Distribution Function) を図 2.2.3 に示す。この結果より、提案手法が従来手法よりも最小通信容量の値が安定して大きいことが分かる。特に、実験規模が大きくなるほど他手法との差は大きくなり、提案手法が大規模な最適化問題に頑健であることが分かる。

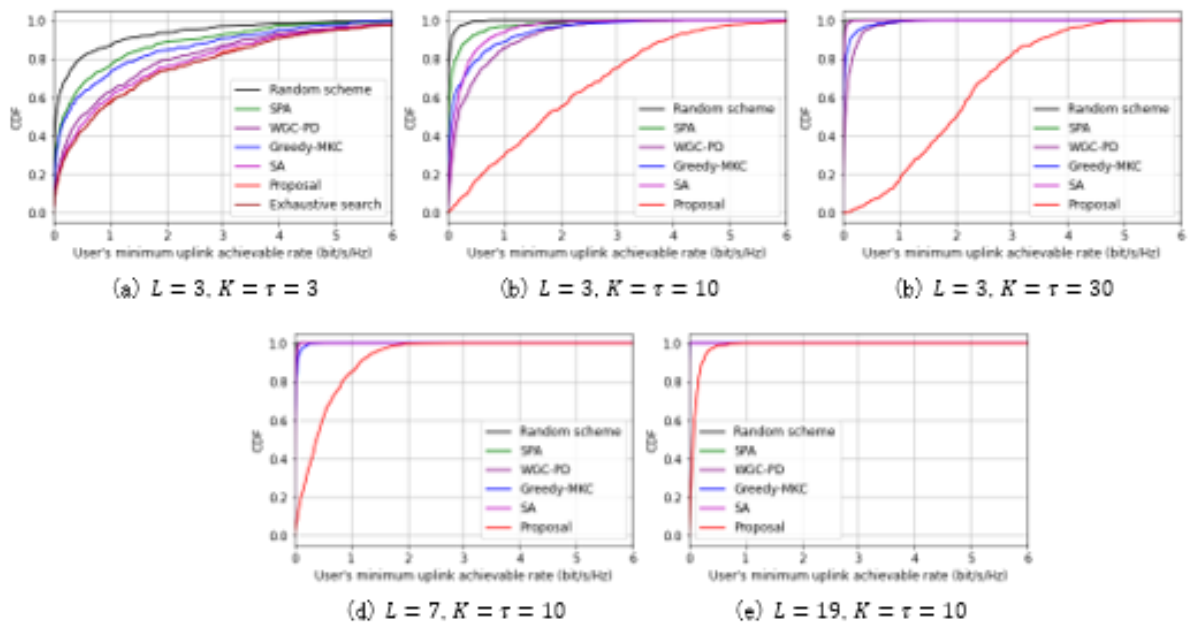


図 2.2.3: 最小通信容量の累積分布関数

2.2.5. おわりに

本節では、Multi-cell Massive MIMO におけるパイロット配置の最適化手法を提案した。本手法は、max k-cut 問題に基づきモデル化され、イジングマシンを用いたアニーリング法によって最適解が導出される。シミュレーション実験の結果、強度のパイロット汚染を引き起こす可能性のある端末に対して、汚染を回避するようにパイロット信号が割り当てられていることを確認した。特に、大規模な端末数とセル数を想定した実験シナリオで本手法は従来手法と比較して高い最適化効果を発揮した。

2.2.6. 参考文献

- [1] X. Zhu, Z. Wang, L. Dai, and C. Qian: “Smart Pilot Assignment for Massive MIMO,” IEEE Communications Letters, vol.19, no.9, pp.1644–1647, Sep. 2015.
- [2] X. Zhu, L. Dai, Z. Wang and X. Wang: “Weighted-Graph-Coloring-Based Pilot Decontamination for Multicell Massive MIMO Systems,” IEEE Transactions on Vehicular Technology, vol.66, no.3, pp.2829–2834, Mar. 2017.
- [3] W. Zeng, Y. He, B. Li and S. Wang: “Fractional pilot reuse and max k - cut based pilot decontamination scheme for multicell TDD massive MIMO systems,” IET Communications, vol.15, no.8, pp.1109–1122, Mar. 2021.
- [4] Technology introduction: FUJITSU LABORATORIES LTD., [online] Available: <https://www.fujitsu.com/jp/group/labs/en/about/resources/tech/techintro.html>.

3. 共同研究者

佐藤 拓朗（早稲田大学・特任教授）

亀山 渉（早稲田大学・教授）
金井 謙治（早稲田大学・次席研究員）
孫 鶴鳴（早稲田大学・次席研究員）
山崎 恭（北九州市立大学・准教授）
市野 将嗣（電気通信大学・助教）

4. 研究業績

4.1 学術論文

【論文誌】

- [1] Chao Liu, Heming Sun, Jiro Katto, Xiaoyang Zeng and Yibo Fan: “An Efficient Low-Complexity Convolutional Neural Network Filter,” IEEE Multimedia, Mar.2022.
- [2] Hidehiro Kanemitsu, Kenji Kanai, Jiro Katto & Hidenori Nakazato: “A containerized task clustering for scheduling workflows to utilize processors and containers on clouds,” Journal of Supercomputing, Apr.2021.

【査読付き国際学会】

- [1] Joi Shimizu, Heming Sun and Jiro Katto: “Forward and Backward Warping for Optical FlowBased Frame Interpolation,” ICAIIC 2022, Feb.2022.
- [2] Shota Hirose, Naoki Wada, Jiro Katto and Heming Sun: “Research and examination on implementation of super-resolution models using deep learning with INT8 precision,” ICAIIC 2022, Feb.2022.
- [3] Jinming Liu, Heming Sun and Jiro Katto: “Learning in Compressed Domain for Faster Machine Vision Tasks,” IEEE VCIP 2021, Dec.2021.
- [4] Bo Wei, Hang Song and Jiro Katto: “Adaptive Video Transmission Strategy Based on Ising Machine,” ACM SenSys 2021, Nov.2021.
- [5] Suphakit Awiphan, Jakramate Bootkrajang, Kanin Poobai and Jiro Katto: “Mitigating Cold Start Problem of Experience-Based Adaptive Streaming in NDN,” IEEE GCCE 2021, Oct.2021.
- [6] Xin Wang, Heming Sun, Jiro Katto, Yibo Fan: “A Hardware Architecture for Adaptive Loop Filter in VVC Decoder,” IEEE ASICON 2021, Oct.2021.
- [7] Kazuki Murayama, Kenji Kanai, Masaru Takeuchi, Heming Sun, Jiro Katto: “Deep Pedestrian Density Estimation for Smart City Monitoring,” IEEE ICIP 2021, Sep.2021.
- [8] Liuhong Chen, Heming Sun, Jiro Katto, Xiaoyang Zeng and Yibo Fan: “Fast Object Detection in HEVC Intra Compressed Domain,” EUSIPCO 2021, Aug.2021.
- [9] Heming Sun, Lu Yu and Jiro Katto: “Learned Image Compression with Fixed-point Arithmetic,” PCS 2021, June 2021.
- [10] Bo Wei, Hang Song and Jiro Katto: “High-QoE DASH Live Streaming using Reinforcement Learning,” IEEE IWQoS 2021, poster, June 2021.
- [11] Shota Hirose, Naoki Wada, Jiro Katto and Heming Sun: “ViT-GAN: Using Vision Transformer as Discriminator with Adaptive Data Augmentation,” IEEE ICCCI 2021, June

2021.

[12] Bo Wei, Hang Song and Jiro Katto: “FRAB: A Flexible Relaxation Method for Fair, Stable, Efficient Multi-user DASH Video Streaming,” IEEE ICC 2021, June 2021.

[13] Yixuan Zeng, Heming Sun, Jiro Katto and Yibo Fan: “Approximated Reconfigurable Transform Architecture for VVC,” IEEE ISCAS 2021, May.2021.

4.2 総説・著書

なし

4.3 招待講演

[1] 甲藤二郎: “画像圧縮・画像通信に対する深層学習応用と研究開発事例,” 画像電子学会年次大会, June 2021.

4.4 受賞・表彰

[1] Best Paper Award: Joi Shimizu, Heming Sun and Jiro Katto, ICAIIC 2022, Feb.2022.

4.5 学会および社会的活動

[1] 総務省 放送システム委員会委員

[2] 経済産業省 総合資源エネルギー調査会 省エネルギー小委員会委員

5. 研究活動の課題と展望

今後の研究課題としては、以下の研究開発を進める。(1) 深層学習を活用した映像情報の未来予測技術、(2) 高フレームレート映像の低遅延圧縮・伝送技術、(3) ネットワーク仮想化の低遅延スライスにおけるオーケストレーション（リソース管理）、(4) システム情報とユーザ情報を統合したオーケストレーション、(5) 5G/Beyond 5G の実験用小規模テストベッドの作成、等。