

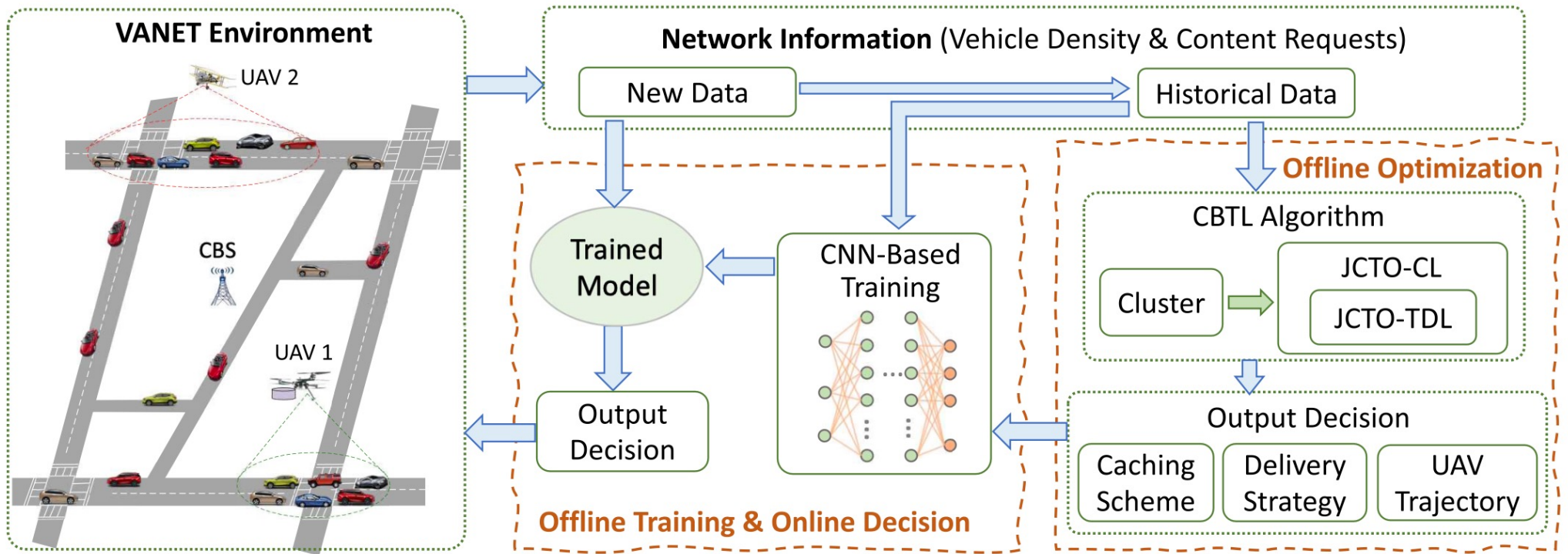
# Optimal UAV Caching and Trajectory in Aerial-Assisted Vehicular Networks: A Learning-Based Approach

---

Huaqing Wu, Feng Lyu, Conghao Zhou, Jiayin Chen, Li Wang, Xuemin Shen (2020)  
IEEE Journal on Selected Areas in Communications, Vol.38, No.12

交通・都市・国土学研究室B4  
内谷太亮

# Abstract



# Introduction

---

## 既存研究の限界

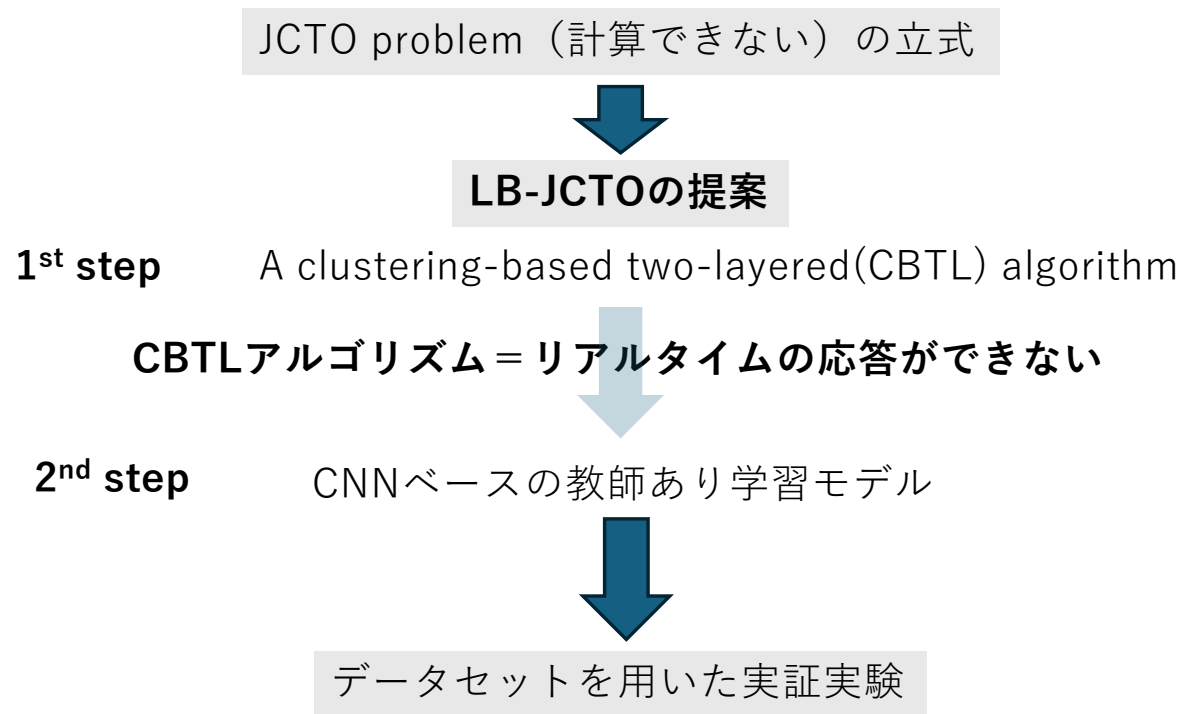
1. ネットワーク上でのユーザーの移動性が低い/全くない状況でのUAVのキャッシングを考えている
2. コンテンツ配置とUAVの軌道とコンテンツ配信を共同決定が十分に扱われていない
3. リアルタイムでの最適化を求められるが、対応できていない



**UAVのバッテリー制約**のもとで、ネットワーク全体のスループットを最大化するように、**リアルタイム**の共同最適化問題への最適解を見つける

# Introduction

## 本研究の流れ



# System Scenario and Problem Formulation

---

# Scenario Description

集合 $\mathcal{K}$	キャッシングストレージを持っているK機の回転翼UAV
$\mathcal{F}$	リクエストされるファイルFの集合
$s_f$	コンテンツファイルのサイズ
$C_k$	UAV $k$ のキャッシングストレージの容量
$H$	UAVが飛ぶ高さ（一定）

時間 $t$ において、グリット $v$ にいる利用者が、ファイル $f_f$ をリクエストする確率 $r_{v,t,f}$

$$r_{v,t,f} = p_f \frac{g(\phi_{v,t}, \vartheta_f)}{\sum_{v' \in \mathcal{V}} g(\phi_{v',t}, \vartheta_f)}, \quad (1)$$

$\phi_{v,t}, \vartheta_f$ はそれぞれ、グリット $v$ とファイル $f_f$ の特徴量

$p_f = \frac{1/f^\xi}{\sum_{m \in \mathcal{F}} 1/m^\xi}$  は、コンテンツ $f_f$ の人気度を表す（Zipf distribution）

$g(\phi_{v,t}, \vartheta_f) = (1 - |\phi_{v,t} - \vartheta_f|)^{\frac{1}{\alpha^3} - 1}$  は、グリット $v$ とファイル $f_f$ の相関を表す（kernel function）

# Scenario Description

前提：単一のCBS(Cellular BS)でカバーされた長方形のエリアで考える

- エリアを幅 $w$ の小さい正方形のグリッドで区切る
- それぞれのグリッドはグリッドの中心点  $(i, j)$  で表現される

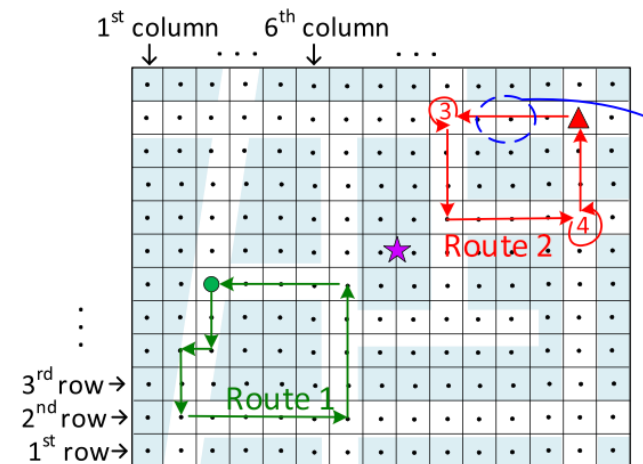
トポロジーグラフ  $G = (\mathcal{V}, \mathcal{E})$  の設定

$\mathcal{V}$       正方形グリッドの中心点の集合

$\mathcal{E}$       中心点を結ぶエッジの集合

UAVの航行持続時間を $T_U$ タイムスロットに離散化し、それぞれの長さを $\Delta_t$ とする

UAVの出発点と終着点は同じ（充電の関係）



# Notation Definition

$$\mathbf{D}_{N_{\text{row}} \times N_{\text{col}} \times T_U}$$

時空間に変動する車両密度

$d_{i,j,t}$  は, タイムスロット  $t$ , グリッド  $(i,j)$  における平均車両数

$$\mathbf{R}_{N_{\text{row}} \times N_{\text{col}} \times T_U \times F}$$

空間と時間に変動するコンテンツのリクエスト分布

$r_{i,j,t,f}$  は, タイムスロット  $t$ , グリッド  $(i,j)$  におけるファイル  $f_f$  のリクエスト確率

$$\mathbf{X}_{K \times T_U} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K] \quad K \text{台のUAVの飛行軌跡}$$

$$\mathbf{A}_{K \times F} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_K] \quad \text{UAVにコンテンツファイルがキャッシングされているか示す指標行列}$$

$\mathbf{a}_k = [a_{k,1}, a_{k,2}, \dots, a_{k,F}]$  は, UAV  $k$  のキャッシング状況を表す

$$\mathbf{S}_{K \times T_U} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_K] \quad \text{UAVの軌跡に沿ってUAVがコンテンツ配信するかどうかを決定}$$

$s_{k,t} = 1$  の時は, 時間  $t$  において UAV  $k$  がリクエストを飛ばしてきた車両にコンテンツを配信



# Communication Models

## U2V Communications

**UAVのカバレッジ半径** (UAVが提供できる通信やサービスの範囲を示す半径)

$$r_{UAV} = \min \left\{ \frac{H}{\tan \left( a_1 - \frac{1}{a_2} \ln \left( \frac{1 - \xi_{LoS}}{a_1 \xi_{LoS}} \right) \right)}, \sqrt{\left( \frac{c \gamma_{max}}{4\pi f_c} \right)^2 - H^2} \right\}, \quad (2)$$

$a_1, a_2$ : 環境によって変化する定数  
 $f_c$ : キャリア周波数  
 $c$ : 光速  
 $\xi_{LoS}$ : LoSリンクの確率  
 $\gamma_{max}$ : U2V freespaceパスロスの閾値

**UAVが提供する平均スループット** (各ゾーンの面積加重平均をとっている)

$$\bar{R}_{UAV} = \rho \left( \frac{\sum_{j=1}^{j=L} (R_j [(l_j + l_{j-1})^2 - l_{j-1}^2])}{\left( \sum_{j=1}^{j=L} l_j \right)^2} \right), \quad (3)$$

L個のゾーンに分割  
j番目ゾーンは幅 $l_j$ で、データレート $R_j$   
 $\rho$ : Wi-Fiスループット係数



$$N_{U,max} = \lfloor \bar{R}_{UAV} / R_{req} \rfloor. \quad (4)$$

$N_{U,max}$ : 同時にUAVからコンテンツを提供できる最大台数

# Communication Models

## C2V Communications

C2Vチャネルゲイン (基地局から車両への信号強度)

$$h_{i,C} = \rho_{i,C} d_{i,C}^{-\alpha},$$

$P_{C,max}$  CBSの利用可能な最大送信電力

$B_C$  セルネットワーク全体で利用可能な総セル  $h_{v,C}$ : クトル帯域幅

セル単独での全体のネットワークスループット

$$R_C(t) = \frac{B_C}{\sum_{v \in \mathcal{V}} d_{v,t}} \sum_{v \in \mathcal{V}} d_{v,t} \log \left( 1 + \frac{P_{v,C}(t) h_{v,C}}{\sum_{v \in \mathcal{V}} d_{v,t}} \right), \quad (5)$$

$P_{v,C}(t)$  送信電力

$h_{v,C}$  CBSから車両までの平均チャネルゲイン

$\sigma^2$  ノイズスペクトル密度

# Communication Models

## C2V Communications

$$\begin{aligned} \sum_{v \in \mathcal{V}} d_{v,t} P_{v,C}(t) &= P_{C,\max}, \\ P_{v_1,C}(t) h_{v_1,C} &= P_{v_2,C}(t) h_{v_2,C}, \quad \forall v_1, v_2 \in \mathcal{V}. \end{aligned} \quad (6)$$

送信電力の強さ

$$n_{C,k,t}^{\mathbf{A},\mathbf{X},\mathbf{S}} = (1 - s_{k,t}) d_{x_{k,t},t} + s_{k,t} d_{x_{k,t},t} \left( 1 - \sum_{f_f \in \mathbf{a}_k} r_{x_{k,t},t,f} \right). \quad (7)$$

CBSによって提供される  
コンテンツを必要とする  
車の平均台数

$$B_{C,t}^{\mathbf{A},\mathbf{X},\mathbf{S}} = \frac{B_C}{\sum_{k=1}^K n_{C,k,t}^{\mathbf{A},\mathbf{X},\mathbf{S}} + \sum_{u \in \mathcal{V}, u \notin \mathcal{V}_t} d_{u,t}}. \quad (8)$$

# Communication Models

$$R_{C,t}^{\mathbf{A},\mathbf{X},\mathbf{S}} = B_{C,t}^{\mathbf{A},\mathbf{X},\mathbf{S}} \left( \sum_{u \in \mathcal{V}, u \notin \mathcal{V}_t} d_{u,t} \log \left( 1 + \frac{P_{u,C}(t) h_{u,C}}{B_{C,t}^{\mathbf{A},\mathbf{X},\mathbf{S}} \sigma^2} \right) + \sum_{k=1}^K n_{C,k,t}^{\mathbf{A},\mathbf{X},\mathbf{S}} \log \left( 1 + \frac{P_{x_{k,t},C}(t) h_{x_{k,t},C}}{B_{C,t}^{\mathbf{A},\mathbf{X},\mathbf{S}} \sigma^2} \right) \right),$$

CBSのアベレージスループット

$$R_{U,k,t}^{\mathbf{A},\mathbf{X},\mathbf{S}} = \bar{R}_{UAV} \cdot \varepsilon \left( d_{x_{k,t},t} \cdot s_{k,t} \cdot \sum_{f_f \in \mathbf{a}_k} r_{x_{k,t},t,f} \right), \quad (9)$$

UAVのアベレージスループット

$$R(\mathbf{A}, \mathbf{X}, \mathbf{S}) = \sum_{t=1}^{T_U} \left( R_{C,t}^{\mathbf{A},\mathbf{X},\mathbf{S}} + \sum_{k=1}^K R_{U,k,t}^{\mathbf{A},\mathbf{X},\mathbf{S}} \right). \quad (10)$$

システム全体のスループット

# UAV Energy Consumption Models

## Propulsion Energy

$$\begin{aligned} P(V) &= P_0 \left( 1 + \frac{3V^3}{U^2} \right) + P_1 \left( \left( 1 + \frac{V^4}{4v_r^4} \right)^{\frac{1}{2}} - \frac{V^2}{2v_r^2} \right)^{\frac{1}{2}} \\ &\quad + \frac{1}{2}AV^3, \\ E_p(x) &= \frac{x}{V}P(V) + \max \left\{ \Delta_t - \frac{x}{V}, 0 \right\} \cdot (P_0 + P_1), \end{aligned} \quad (11)$$

$x \in \{0, w, \sqrt{2}w\}$  is the flying distance within one time slot determined by the UAV trajectory planning,

$P_0$ ,

$P_1$ ,

$U$ ,

$v_r$ , and

$A$  are constant parameters related to the UAV's weight, wing area, air density, etc.

# UAV Energy Consumption Models

## Communication Energy

時間 $t$ において、グリッド $v$ の上にUAV  $k$ が飛んでいて、 $n$ 台の車がファイル $f_f$ をリクエストする確率

$$\Pr(f_f, d_{v,t}, n) = \binom{d_{v,t}}{n} r_{v,t,f}^n (1 - r_{v,t,f})^{d_{v,t}-n}. \quad (12)$$

リクエストを出した車へコンテンツ提供を行う際の平均コミュニケーション消費エネルギー

$$E_c(\mathbf{a}_k, x_{k,t}) = \sum_{n=1}^{m \cdot d_{v,t}} \Pr(\mathcal{F}_k, d_{x_{k,t},t}, n) P_U \min \left\{ \Delta_t, \frac{n \cdot \zeta_f}{\overline{R}_{UAV}} \right\}. \quad (14)$$

# Problem Formulation

$$\text{(JCTO)} : \max_{\mathbf{A}, \mathbf{X}, \mathbf{S}} R(\mathbf{A}, \mathbf{X}, \mathbf{S}) \quad (15)$$

$$\text{s.t.} \quad \sum_{f_f \in \mathcal{F}} a_{k,f} \leq C_k, \quad \forall f_f \in \mathcal{F}, \forall k \in \mathcal{K}, \quad (15a)$$

UAVにキャッシュされる  
ファイルの最大数

$$\sum_{t=1}^{T_U} E_p(\|x_{k,t} - x_{k,t-1}\|) + E_c(\mathbf{a}_k, x_{k,t}) \cdot s_{k,t} \leq E_{k,\max}, \quad (15b)$$

UAVの最大バッテリー消費可能量

$$x_{k,1} = x_{k,T_U} = v_{0,k}, \quad (x_{k,t-1}, x_{k,t}) \in \mathcal{E}, \quad (15c)$$

UAVはエッジしか飛べない  
出発点に戻ってくる必要がある

$$a_{k,f} = \{0, 1\}, \quad s_{k,t} = \{0, 1\}, \quad (15d)$$

**非凸で実際に解くことができない**

# Design of LB-JCTO

---



# Design of LB-JCTO

---

UAV control center/edge server

UAV

Offline Optimization

ネットワーク情報が与えられている



Offline Model Training

CBTLアルゴリズムで得られた結果を  
教師データとしてCNN



よく訓練されたモデルを元に  
新しいネットワーク情報から判断する

# Offline Optimization

まず、**K+1**のクラスターに分ける

その時に、①セル性能の類似性、②物理的距離の類似性、③コンテンツの好みの類似性の3つの指標を組み合わせる

次に、問題を**2つの層に分離**する

- **Caching-Layer(CL) Optimization** → 粒子群最適化で解く

$$\max_{\mathbf{A}} \sum_{t=1}^T R(\mathbf{A}, \mathbf{X}, \mathbf{S}) \quad (16)$$

$$s.t. \text{ Constraint (15a)}. \quad (16a)$$

- **Trajectory-and-Delivery-Layer(TDL) Optimization** → 時間ベースのグラフ分解法

$$\max_{\mathbf{X}, \mathbf{S}} \sum_{t=1}^T R(\mathbf{A}, \mathbf{X}, \mathbf{S}) \quad (17)$$

$$s.t. \text{ Constraints (15b-15d)}. \quad (17a)$$

# Offline Model Training and Online Decision

---

正確な予測の難しさ  
リアルタイムな適応  
CBTLアルゴリズムの制限



## CNNの利用

重要な特徴の抽出に長ける  
データ次元を削減しながら主要な特徴を残す

# CBTL-Based Offline Optimization

---

2024/5/27

Optimal UAV Caching and Trajectory in  
Aerial-Assisted Vehicular Networks

# Determining the Number of UAVs

CBSによって提供される車両が受信する信号強度

$$\gamma_C = P_{C,\max} / \left( \sum_{v \in \mathcal{V}} \frac{d_{v,t}}{h_{v,C}} \right). \quad (18)$$

$$\frac{B_C}{\sum_{i=1}^{N_C} d_{v_i,t}} \log \left( 1 + \frac{P_{C,\max} \sum_{i=1}^{N_C} d_{v_i,t}}{B_C \sigma^2 \sum_{i=1}^{N_C} \frac{d_{v_i,t}}{h_{v_i,C}}} \right) \geq R_{\text{req}}. \quad (19)$$

車両のスループットリクエストをクリアする最低限度のドローンの台数

$$K_{\min} = \left\lceil \frac{\sum_{v \in \mathcal{V}} d_{v,t} - \sum_{i=1}^{N_{C,\max}} d_{v_i,t}}{N_{U,\max}} \right\rceil. \quad (20)$$

# Vehicle Clustering

K台のUAVがある時, 車両はK+1個のクラスターに分けられる

## ①セルの性能の類似性

$$\text{sim}_{u,v,1} = \min\{R_{C,v}/R_{C,u}, R_{C,u}/R_{C,v}\} \in [0, 1].$$

## ②物理的距離の類似性

$$\text{sim}_{u,v,2} = 1 - \frac{\text{dist}_{u,v}}{\max_{u,v \in \mathcal{V}} \text{dist}_{u,v}} \in [0, 1],$$

## ③コンテンツの好みの類似性

$$\text{sim}_{u,v,3} = \frac{1}{T_U} \sum_{t=1}^{T_U} \frac{\mathbf{r}_{v,t} \cdot \mathbf{r}_{u,t}}{\|\mathbf{r}_{v,t}\| \cdot \|\mathbf{r}_{u,t}\|},$$



$$\text{sim}_{u,v}^{all} = \text{sim}_{u,v,1}^{\alpha_1} \cdot \text{sim}_{u,v,2}^{\alpha_2} \cdot \text{sim}_{u,v,3}^{\alpha_3}, \quad (22)$$

---

### Algorithm 1 K-Means-Based Vehicle Clustering

---

Let  $\bar{d}_v = \frac{1}{T_U} \sum_{t=1}^{T_U} d_{v,t}$ ,  $\bar{\mathbf{r}}_v = \frac{1}{T_U} \sum_{t=1}^{T_U} \mathbf{r}_{v,t}$ ,  
 $\alpha_1, \alpha_2, \alpha_3 = 1$ .

**Step 1: Centroid Initialization:** The first cluster centroid  $v_C^0$  is the grid where the CBS is located. Besides, randomly choose  $K$  grids, denoted by  $v_C^1, \dots, v_C^K$ , as the centroids for the remaining  $K$  clusters.

**Step 2: Grid Clustering:**  $K + 1$  clusters, denoted by  $\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_K$ , are created by associating every grid with the centroid with maximum similarity based on (22).

**Step 3: Centroid Update:** Update the  $K + 1$  cluster centroids:

$$v_C^0 = v_C^0, \quad v_C^k = \frac{1}{\sum_{v \in \mathcal{C}_k} \bar{d}_v} \sum_{v \in \mathcal{C}_k} \bar{d}_v v,$$

$$R_{C,v_C^k} = \frac{1}{\sum_{v \in \mathcal{C}_k} \bar{d}_v} \sum_{v \in \mathcal{C}_k} \bar{d}_v R_{C,v}, \quad \mathbf{r}_{v_C^k} = \frac{1}{\sum_{v \in \mathcal{C}_k} \bar{d}_v} \sum_{v \in \mathcal{C}_k} \bar{d}_v \bar{\mathbf{r}}_v.$$

**Step 4:** Repeat Steps 2-3 until converging.

**Step 5:** Repeat Steps 1-4 and choose the best for multiple runs.

---

# JCTO-TDL Optimization in CBTL Algorithm

Resource constrained shortest path(RCSP) algorithmがそのままでは使えない



有向グラフなどを用いて表現することで適用できるように

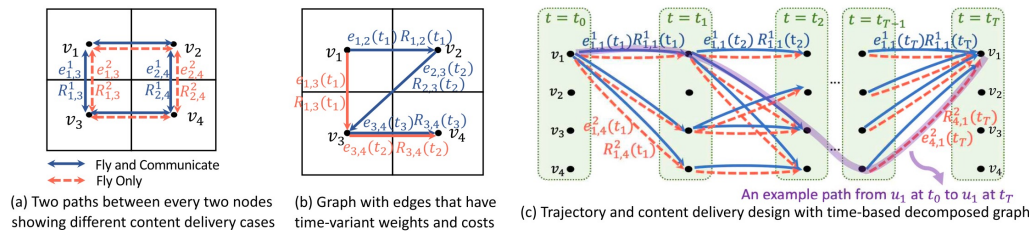


Fig. 4. A simple example of trajectory and content delivery design with time-based graph decomposition. ( $R_{i,j}^1(t)$  and  $R_{i,j}^2(t)$ : achievable throughput when flying from  $v_i$  to  $v_j$  at time  $t$  with and without content delivery.  $e_{i,j}^1(t)$  and  $e_{i,j}^2(t)$ : corresponding energy consumption.)

---

## Algorithm 2 JCTO-TDL Optimization in CBTL Algorithm

---

$v_k$ : the starting and ending point of UAV  $k$ .

**Step 1:** For any  $(u, v) \in \mathcal{E}$  and  $t \in [1, T_U]$ , calculate the weights (throughput) and costs (energy consumption).

**Step 2:** Use shortest path (SP) algorithms (e.g., Dijkstra's algorithm) to find the path with smallest cost. Let  $E_{t_0, t_T}$  denote the sum cost from source to  $v_k$  at time slot  $t_T$ . Find  $t_T$  such that  $E_{t_0, t_T} \leq E_{k, all}$  and  $E_{t_0, t_{T+1}} > E_{k, all}$ . Let  $t_T^{max} = t_T$ .

**Step 3:** Use SP algorithms to find the path with largest cost. Find  $t_T$  such that  $E_{t_0, t_T} \leq E_{k, all}$  and  $E_{t_0, t_{T+1}} > E_{k, all}$ . Let  $t_T^{min} = t_T$ .

**for**  $t_T = [t_T^{min}, t_T^{max}]$  **do**

Construct time-based decomposed graph as shown in Fig. 4(c).

Apply RCSP algorithms to find the optimal path in the graph. Record the best path which leads to the maximum achievable network throughput.

**end**

Output the recorded best path.

---

# JCTO-CL Optimization in CBTL Algorithm

## 粒子群最適化の導入

$$\begin{aligned} \nu^\ell(t+1) &= \phi \nu^\ell(t) + c_1 \phi_1 (\varpi_{\text{local}}^\ell(t) - \varpi^\ell(t)) \\ &\quad + c_2 \phi_2 (\varpi_{\text{global}}(t) - \varpi^\ell(t)), \\ \varpi^\ell(t+1) &= \varpi^\ell(t) + \nu^\ell(t+1), \end{aligned} \tag{23}$$

[View Source](#) 

where

$\phi$  determines convergence speed,

$c_1$  and

$c_2$  are local and global learning coefficients, and

$\phi_1$  and

$\phi_2$  are positive random variables. The iteration terminates when a termination criterion (e.g., reaching the maximum iterations or minimum error criteria) is met.





# **CNN-Based Learning For Online Decision**

---

2024/5/27

Optimal UAV Caching and Trajectory in  
Aerial-Assisted Vehicular Networks

# Image-Like Input Data

JCTO問題を動的なネットワーク情報を用いて調査する

Dは三次元配列

Rは四次元配列であるから、 $\phi_{v,t}$ を使用してグリットvの時間tにおけるコンテンツ要求分布を表す

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}. \quad (24)$$

また、Dと $\phi$ は異なるスケールなので、(24)で正規化を実施



**学習プロセスが容易に**

# CNN-Based Model Training

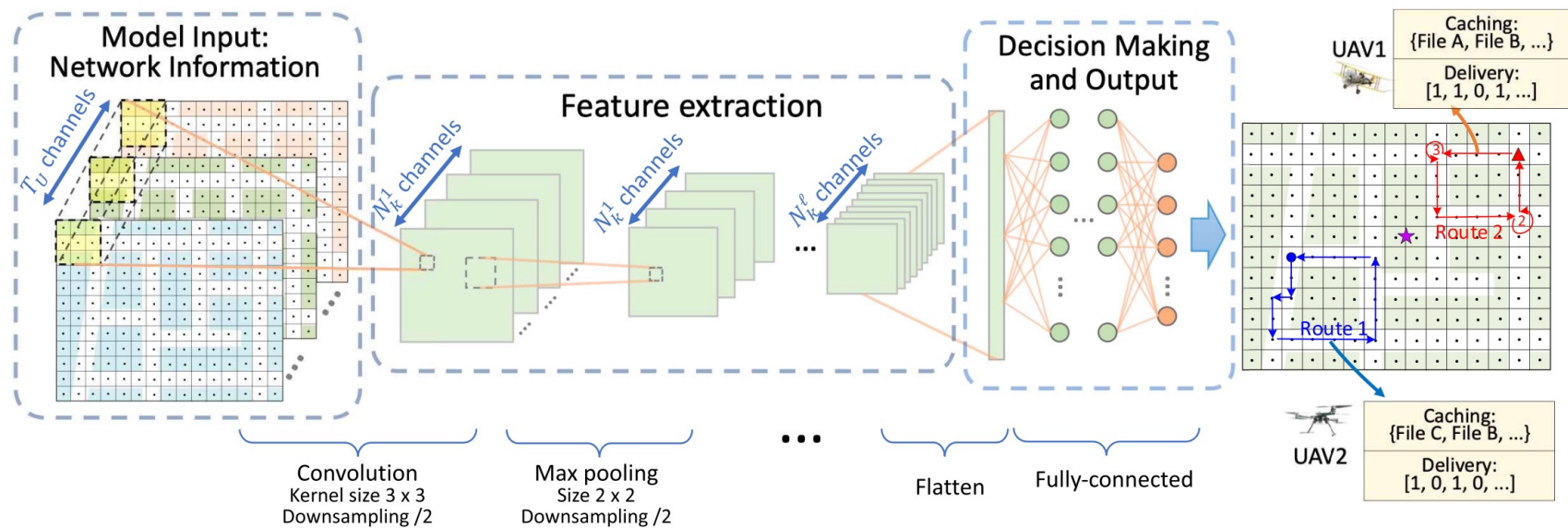


Fig. 5. Structure of the CNN-based deep supervised learning model.

# Performance Evaluation

---

2024/5/27

Optimal UAV Caching and Trajectory in  
Aerial-Assisted Vehicular Networks

# Experiment Settings

## シミュレーションの概要

データセット：DiDi Chuxing GAIA Initiative datasetを使用。データは西安の第二環状道路上のタクシーのGPSトレースを含み、車両の位置、ID、及び対応するタイムスタンプなどのキー属性を記録。

シミュレーション範囲：2000m×2000m

UAV関連パラメータ： $K=2$ ,  $H=30\text{m}$ ,  $V=15\text{m/s}$ ,  $E_{k,max} = 50\text{KJ}$ ,  $\xi_{LoS} = 0.99$ ,  $\gamma_{max} = 37.5\text{dB}$

セルラー通信パラメータ： $\alpha = 3$ ,  $B_C = 20, \text{MHz}$ ,  $P_{C,max} = 20\text{W}$ ,  $\sigma^2 = 10^{-15}\text{W/Hz}$

Zipf指数： $\xi=0.7$ ,  $\Delta_t = 5\text{s}$

## 学習モデルの構築

レイヤの構成：11層の学習モデルを実装

畳み込み層：チャンネルサイズは16, 32, 64。各畳み込み層のカーネルサイズとストライドは(3,3)と(2,2)

プーリング層：各畳み込み層の後に最大プーリング層を追加。プールサイズは(2,2)

全結合層：1024, 1024, 512, 256ニューロンを持つ4つの全結合層を追加。最後に出力層を追加

活性化関数：各層の後にReLU活性化関数を追加して非線形性を導入し、学習能力を向上させる

# Evaluation of CBTL-Based Offline Optimization

## Benchmark scheme

しらみつぶし探索 (ES)

貪欲法 (Greedy)

## PSO- vs ES-Based algorithms

- 図6a : PSOベースの方法はESアルゴリズムと**ほぼ同等のスループット性能**を提供
- 図6b : PSOベースの方法はESアルゴリズムよりも**実行時間が短い**
- 結論 : CBTLアルゴリズムは学習の監督者として利用され、**データラベリング時間が節約**される

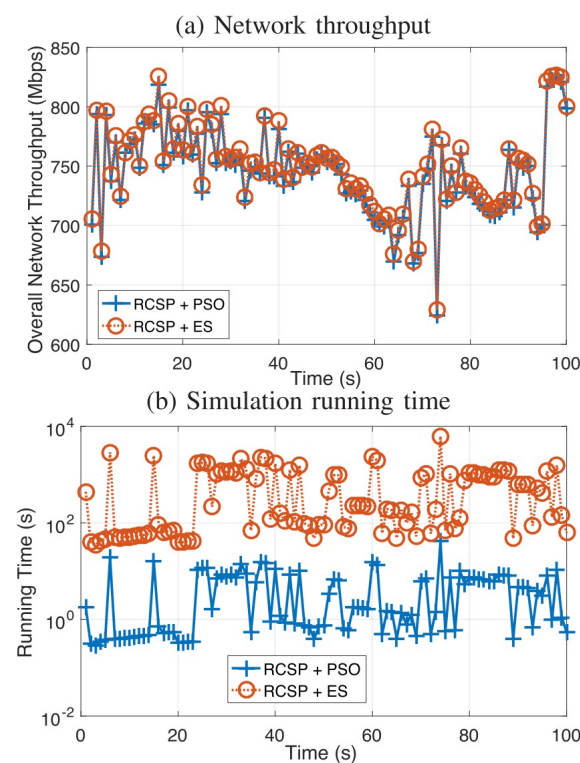


Fig. 6. Comparison between PSO- and ES-based algorithms.

# Evaluation of CBTL-Based Offline Optimization

## RCSP- vs Greedy-Based algorithms

- 図7a：RCSPベースの方法は、**貪欲アルゴリズムよりも優れた性能**を提供
- 図7b：RCSPベースの方法は、特に小さい $\Delta t$ とHの場合に**貪欲アルゴリズムよりも実行時間がかかる**
- 結論：CBTL最適化アルゴリズムは、貪欲ベースのアルゴリズムよりもわずかに実行時間がかかるが、**ほぼ最適なネットワークスループット性能を達成**できる

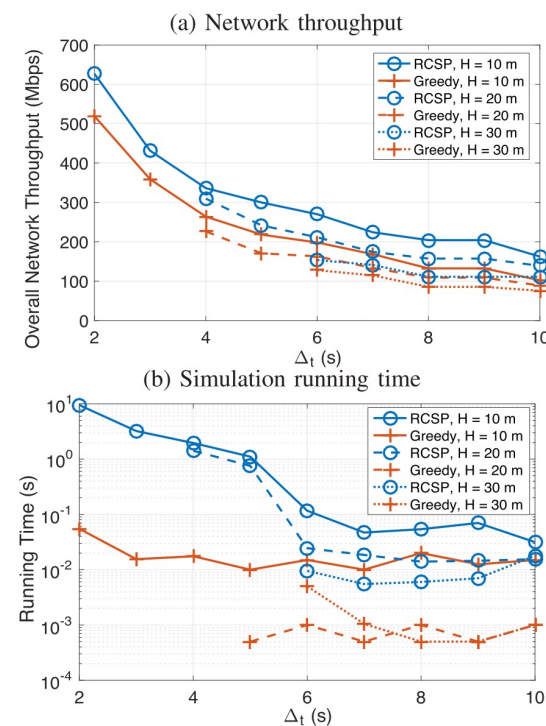


Fig. 7. Comparison between RCSP- and greedy-based algorithms.

# Evaluation of CBTL-Based Offline Optimization

## Throughput Performance vs. $K$ and $E_{K,max}$

- 図8a：システムに入るUAVの数が増えると、ネットワークスループットが増加
- 図8b：より大きな $E_{K,max}$ は、UAVのシステム内での滞在時間を長くするため、より高いネットワークスループットをもたらす
- 結論：UAVの軌道設計は、UAVの配置設計よりも効果的であり、ネットワークの変動に適応してエネルギーを効果的に利用できる

## 結論

提案されたCBTL最適化アルゴリズムは、高い性能と適応性を持ち、オフラインでの最適化において有効

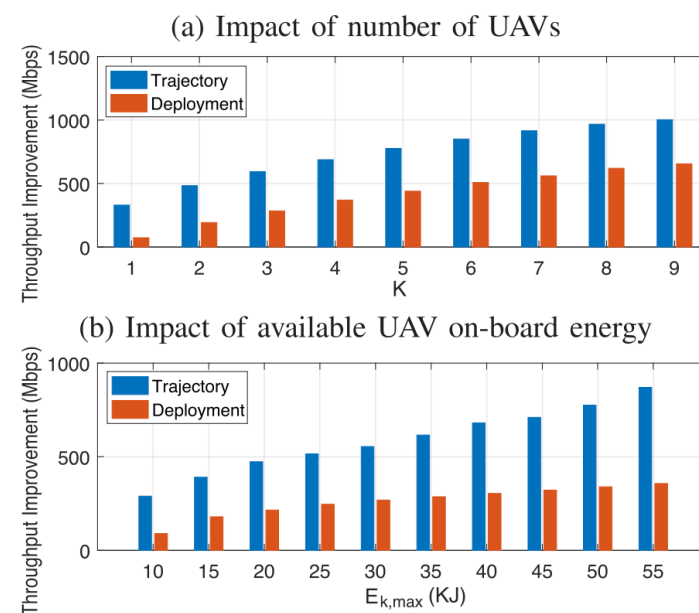


Fig. 8. Throughput performance vs.  $K$  and  $E_{k,max}$ .



# Evaluation of EI-Based CNN Learning Model

## モデルのトレーニング

複数のモデルを、異なる車両トレースデータのセットを使用してトレーニング

(例: “[20, 50]”, “[50, 80]”, “ $\geq 80$ ”, “General”など、異なる車両密度の範囲でトレーニングされたモデル)

## Performance for CNN-Based online decision model

- 図9b: 一般的なモデルを使用した場合、ネットワークスループットはRCSPベースのCBTLアルゴリズムよりも優れていない
- 図9c: “Model:  $\geq 80$ ”を使用した場合、車両密度が80を超えると、RCSPベースの最適化アルゴリズムとほぼ同等のネットワークスループットが提供される
- 図9d: CNNベースの学習モデルは、CBTLベースのオフライン最適化アルゴリズムと比較して、実行時間がはるかに短い

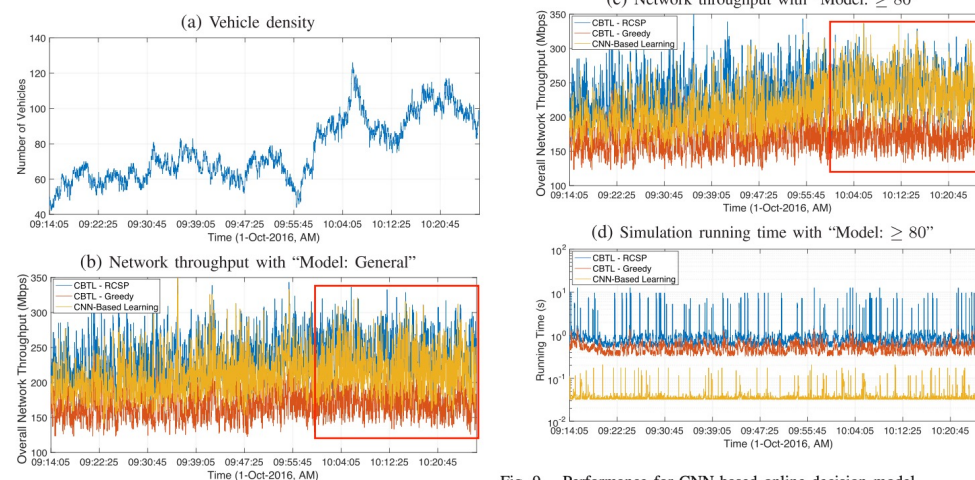
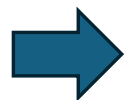


Fig. 9. Performance for CNN-based online decision model.

# Evaluation of EI-Based CNN Learning Model

## Network Throughput with different methods of training data section

- 複数の密度指定モデルをトレーニングし、対応するシナリオに適用することで、モデルパフォーマンスを向上させる方法を示す
- しかし、モデルのトレーニングの微細化は、多くのトレーニングとストレージコストを伴う



最適な学習モデルの数は、スループット要件やデバイスの計算及びストレージ容量、データの利用可能性などに基づいて決定されるべきである

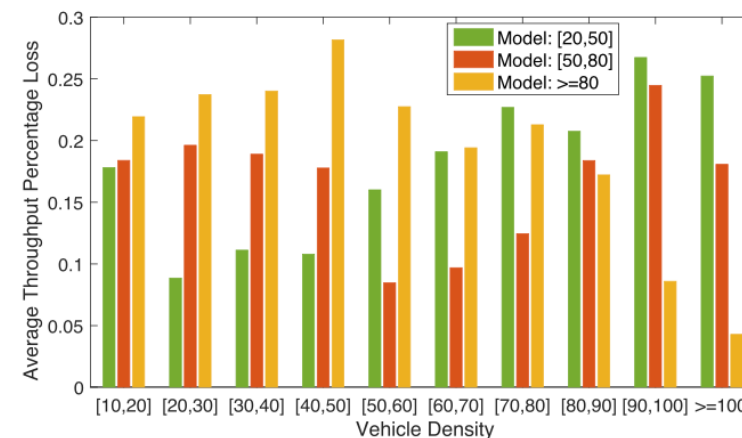


Fig. 10. Throughput performance with density-related CNN models.

# Evaluation of EI-Based CNN Learning Model

## Network Throughput with different methods of training data section

- モデルのパフォーマンスに与えるトレーニングデータの影響を示す
- より多くのデータを使用すると、一般的にネットワークの特徴や規則性に関する情報が学習され、スループット性能が向上する
- トレーニングデータ選択の方法によってパフォーマンスが異なり、特に**連続的なデータの場合**、最悪の性能が得られる可能性がある

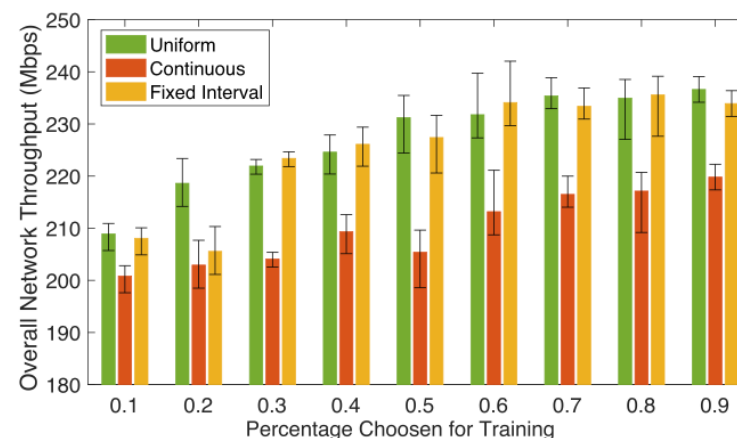


Fig. 11. Network throughput with different methods of training data selection.

## 結論

- CNNベースのオンライン決定モデルは、**適切にトレーニングされると、満足できるネットワークスループット性能と低複雑性**を提供できる
- 異なるネットワーク条件に適用するモデルを選択し、微調整する方法についてはさらなる調査が必要である

# Conclusion and Future Work

## 高度な動的な車両ネットワークにおけるUAVキャッシングと軌道の共同設計について

### 問題の概要

JCTO問題は非凸であり、リアルタイムに解決するのが困難

### 提案手法 LB-JCTO

- オフラインの段階でJCTO問題を最適化するために**CBTLアルゴリズム**を使用
- オンラインの段階では、エッジで学習モデルをトレーニングしてオンライン決定を行う
- **CNNベースの教師付き学習モデル**を使用し、CBTLアルゴリズムを学習

### 実験と結果

- トレース駆動型の実験を実施して**LB-JCTOの効率を示す**
- JCTOの問題設定とCBTLの最適化プロセスは、**キャッシングを用いたUAVシステム**に関連する将来の研究の**理論的基礎**を提供

# Conclusion and Future Work

---

## 将来の展望

- CNNベースのモデルを最適化して学習パフォーマンスを向上させ、異なる車両環境に適応可能にする予定
- オフライン最適化とオンライン上での決定の学習原理は、将来の異なる種類または空気地一体型車両ネットワークにおける他の複雑なリソース管理にも有用であると考えられる

# 所感

---

この研究でやりたいことは何となく見えたものの、数式やアルゴリズムを追うだけの基礎がなく、論文を読むのに苦労した

今回は時間がなかったなので、もう一回じっくりと腰を据えて、機械学習への理解を深めながら読んでいきたい

# Appendix

---