

低軌道衛星キャッシュの人気度を 考慮した最適コンテンツ配置法

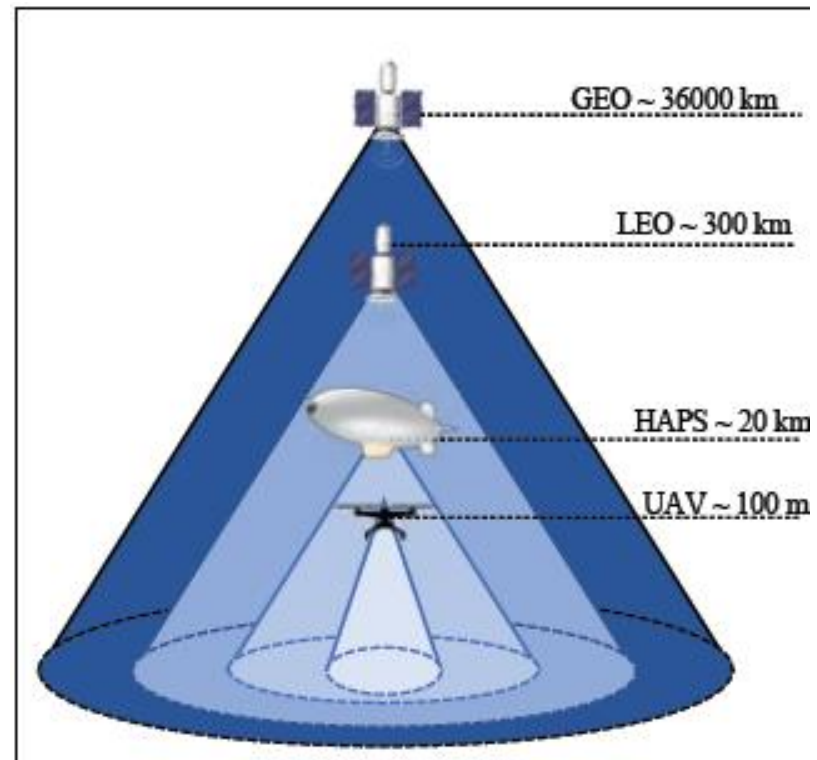
NS研究会

2025年 3月6日

立命館大学 情報理工学部
西嶋 舶

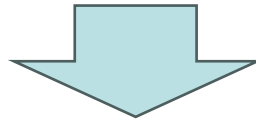
低軌道(LEO)衛星ネットワーク

- 高度2,000km以下の低軌道を周回する衛星を中継ノードとして用いたネットワーク
- 低電力・低遅延・大容量の通信が可能
- LEO衛星は高速で移動し、かつカバーエリアが狭いため、何千もの衛星が必要
- 有効な応用:
 - デジタルデバイドの解消
 - 空・海や過疎地域へのブロードバンド提供
 - バックアップ
 - 災害対策

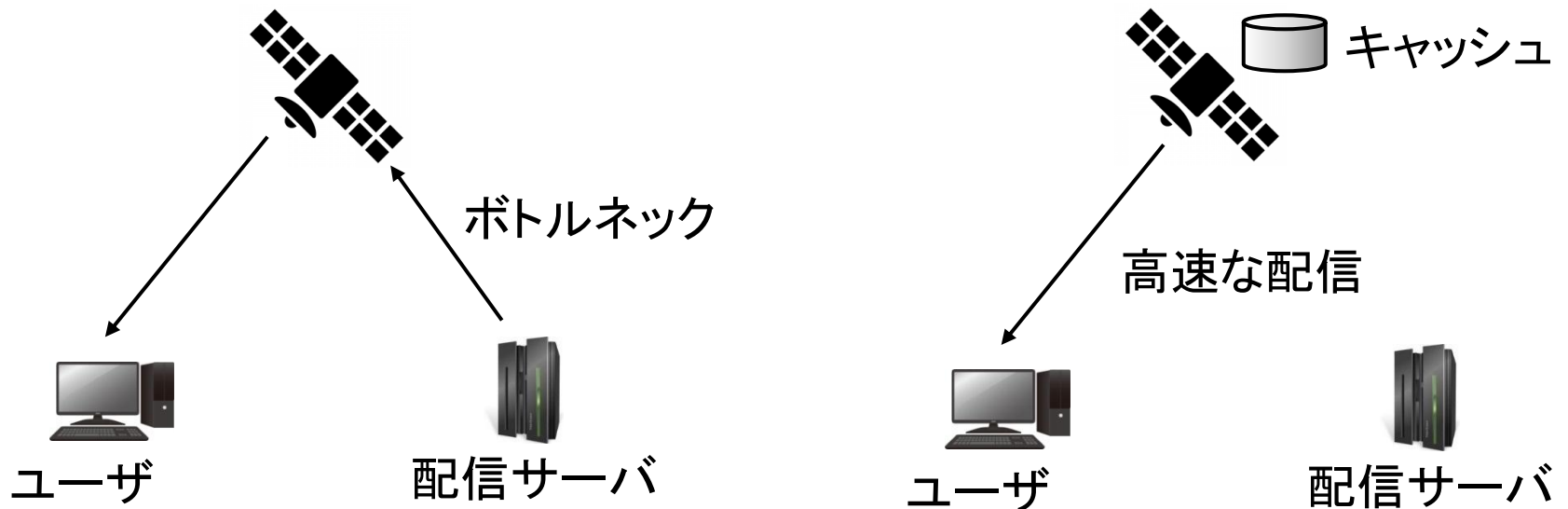


研究背景

- 低軌道衛星は上りのスループットが低い
 - 地上に存在するサーバから衛星経由で大容量のコンテンツを地上の端末に配信すると、上りのスループットがボトルネック



低軌道衛星にキャッシュを設置し、衛星からのキャッシュ配信が有効



研究目的

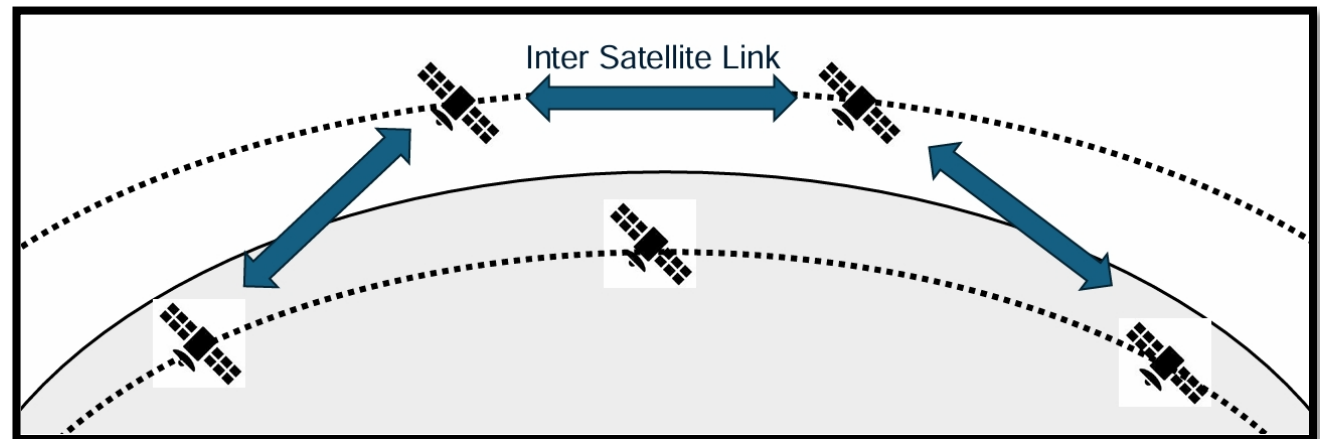
- 本研究の目的:コンテンツの低軌道衛星キャッシュへの最適配置法を提案
- 2つの方針:
 - 高人気のコンテンツを多数、配置することで、全体のキャッシュヒット率を向上
 - 同一コンテンツを固めずに分散配置することで、配信ホップ数を低減

配置方針

- 低軌道衛星は高速で地球上空を周回
 - 約90分～120分で地球を1周
 - 世界の様々な地域から配信要求を受ける
- コンテンツの世界的な人気度に着目
 - 全世界で人気の高いコンテンツは要求される回数が多い
- 世界的に人気の高いコンテンツを衛星キャッシュに多数配置することで、全体的なキャッシュヒット率を向上

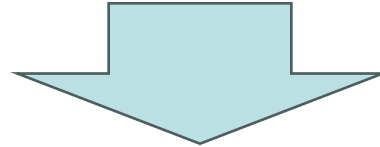
配置方針

- 衛星間にレーザを用いた通信リンクを確立する衛星間リンクが考えられている
 - 今後の衛星コンステレーションで運用予定
- 要求先の衛星キャッシュに目的コンテンツが存在しない場合衛星間通信を用いて配信
- 目的コンテンツをキャッシュしている衛星が近くにあれば通信遅延が削減



提案する配置方式

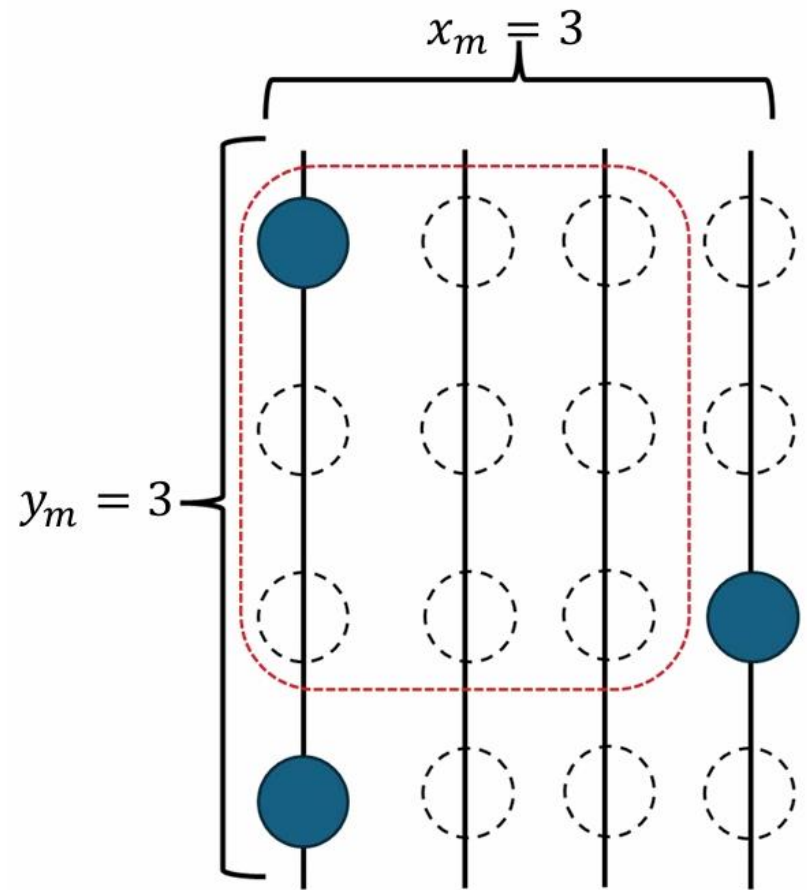
- 高人気コンテンツをできるだけ多くキャッシュし、かつコンテンツを分散させて配置



- 各コンテンツを低軌道衛星上に等間隔で配置
- 平均キャッシュヒット率を最大化するよう、各コンテンツの配置間隔を最適化
 - 最適化に遺伝的アルゴリズムを使用

配置方式の概要

- 地球全土をカバーする衛星ネットワークを想定
- 各コンテンツに対し、それぞれ緯度方向と経度方向の配置間隔を決定
- $(x_m * y_m)^{-1}$ を推定キャッシュヒット率として計算
- キャッシュ容量を超えないよう推定ヒット率が最大化する配置間隔を導出



遺伝的アルゴリズム(GA)

- 生物の進化過程を模倣した近似解を探索するアルゴリズム
- GAの流れ
 - 初期個体をランダムに生成
 - 各個体が持つ遺伝子の適応度を計算
 - 適応度に基づく選択確率に従い、親となる個体を選択
 - 交叉確率に基づいて子個体を生成
 - 突然変異確率に従い、遺伝子を突然変異させる
 - 以上の操作を任意の世代数繰り返す

GAの設定

- 遺伝子構造は各コンテンツの配置間隔
 - コンテンツ m に対し、それぞれ配置間隔を設定
 - (m, x_m, y_m) の形式
- 適応度は推定キャッシュヒット率に人気度をかけた値
 - $(x_m * y_m)^{-1}$ で計算される推定キャッシュヒット率に、Zipf分布により計算される各コンテンツの人気度をかけた値の合計をその個体の適応度とする
 - 適応度の計算式: $\sum_{m=1}^M (\frac{1}{x_m * y_m} * q_m)$
 - 人気度の計算式: $q_m = \frac{1/k^\theta}{\sum_{j=1}^M 1/j^\theta}$

GAにおける制約条件

- 各コンテンツの配置間隔を決めるとき、キャッシュ容量を考慮する必要がある
- GAにおける適応度計算時に制約条件を加える
 - 制約条件はキャッシュの総容量
 - 適応度にペナルティを与えることで制約を考慮
- 各コンテンツの配置間隔から、大まかなキャッシュの使用量を計算
- キャッシュ使用量から適応度を与えるペナルティ係数を計算
- $$P = \frac{C_{used} - C_{max}}{C_{max}}$$

適応度関数

- キャッシュ使用量がキャッシュの総容量を超えていない場合
以下の式で適応度を計算

- $$\sum_{m=1}^M \left(\frac{1}{x_m * y_m} * q_m \right)$$

- キャッシュ使用量がキャッシュの総容量を超えている場合
以下の式で適応度を計算

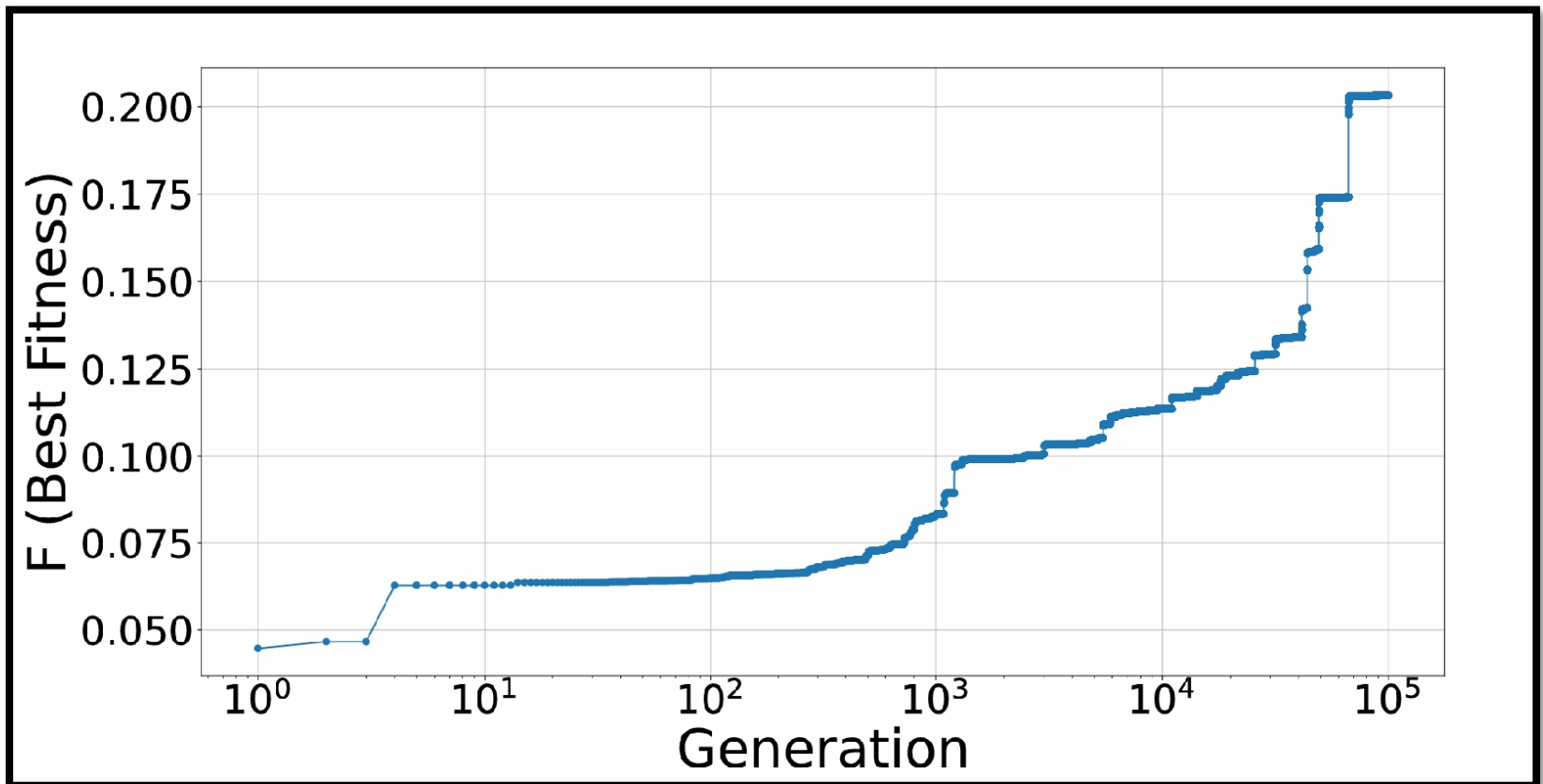
- $$\sum_{m=1}^M \left(\frac{1}{x_m * y_m} * q_m \right) * (1 - P)$$

- ペナルティにより、制約条件を満たさない配置パターンは次世代に残りずらい

性能評価

■ GAの結果

- 個体数10個、交叉80%、突然変異確率0.1%、10万世代
- 推定キャッシュヒット率は約20%



性能評価

- 簡易的なシミュレーションでの性能評価条件
- 極座標コンステレーションを想定
 - OneWebを参考にして衛星数を決定
 - 軌道数12個、軌道内の衛星数49個
 - 各衛星のキャッシュ容量はコンテンツ10個分
- コンテンツ数は1000個に設定
- 各コンテンツの人気度は同一のZipf分布で与える

性能評価

- GAで最適化した配置間隔に従って配置を行った場合と、貪欲法による配置を行った結果を比較
- コンテンツ配置の方法
 - まずすべてのコンテンツをランダムな衛星に一つずつ配置
 - GAで決定した配置間隔に従って、人気上位のコンテンツから配置を始める
 - ランダムに衛星を選び、その衛星から等間隔に配置する
- 貪欲法による配置でも同様

貪欲法

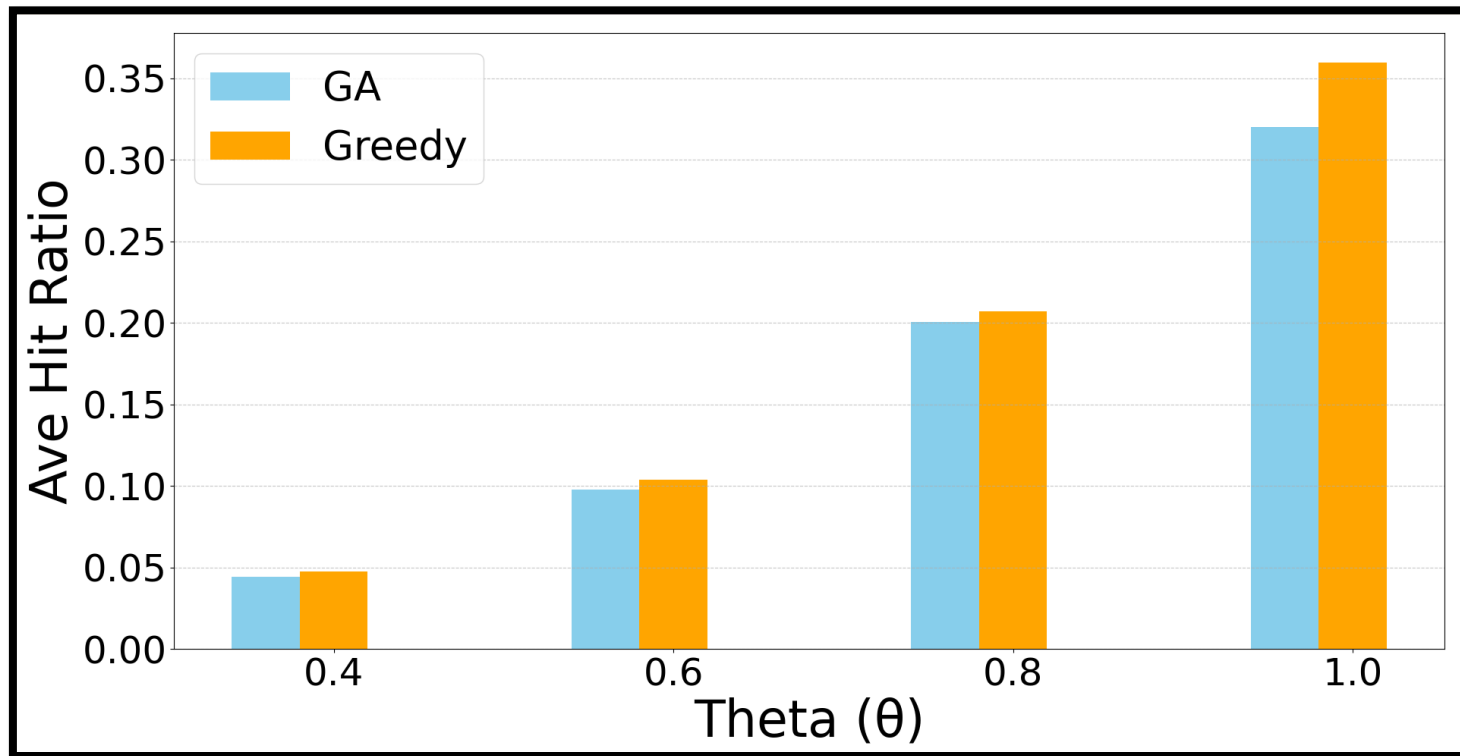
- 人気上位のコンテンツから順に、配置間隔を緯度方向、経度方向ともに1に設定
- その時点でのキャッシュ使用量を計算し、次のコンテンツ配置で容量を超える場合、そのコンテンツの配置間隔を+1する
- 全コンテンツの配置間隔が決まるまで繰り返し
- 最終的に、人気上位8位までのコンテンツが間隔1となり、人気下位のコンテンツの配置間隔は上限値となる

性能評価

■ シミュレーション結果

- GAで最適化した配置 : GA
- 貪欲法による配置 : Greedy

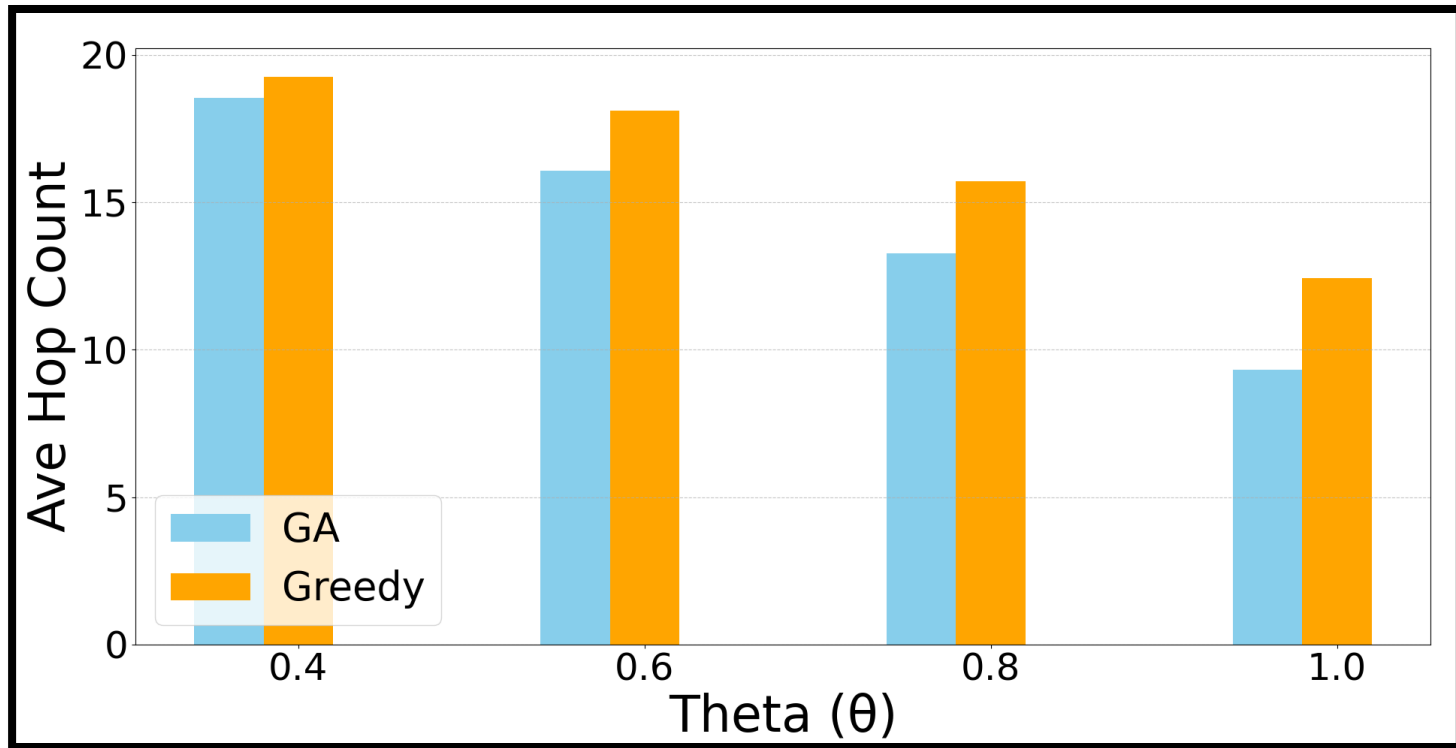
■ キャッシュヒット率の評価



性能評価

■ 平均ホップ数

■ GAによる配置の方がわずかにホップ数が少ない



まとめ

- 各コンテンツを低軌道衛星上に等間隔で配置する配置法を提案し、配置間隔の最適化方法としてGAと貪欲法を比較した
- 今後の課題
 - 衛星のカバー可能範囲や時間による移動を考慮したより厳密なシミュレーションを行う
 - GAの適応度関数の改良
 - 提案した配置法と異なる配置法との性能比較