令和元年度（第19回）電子航法研究所研究発表会
6/7/2019

空域編成に対する
2つの最適化アプローチ

筑波大学 猿渡康文
神奈川大学 伊豆永洋一
防衛大学校 鵜飼孝盛
航空交通管理領域 蔭山康太

- 背景と目的
- 解法のフレームワーク
  - 管制作業量和最小化
  - 管制作業量平準化
  - 可視化
- テストデータへの適用

我が国の航空交通

- 航空管制の年間取り扱い数
  - 1998年（197万）→2017年（310万）：1.6倍以上
  - 国際線/上空通過を中心とした需要の継続の予測
- 効率性/環境への意識の向上
  - 燃料効率の高い飛行の実現の要求

⇨ 航空交通システムの性能向上の必要

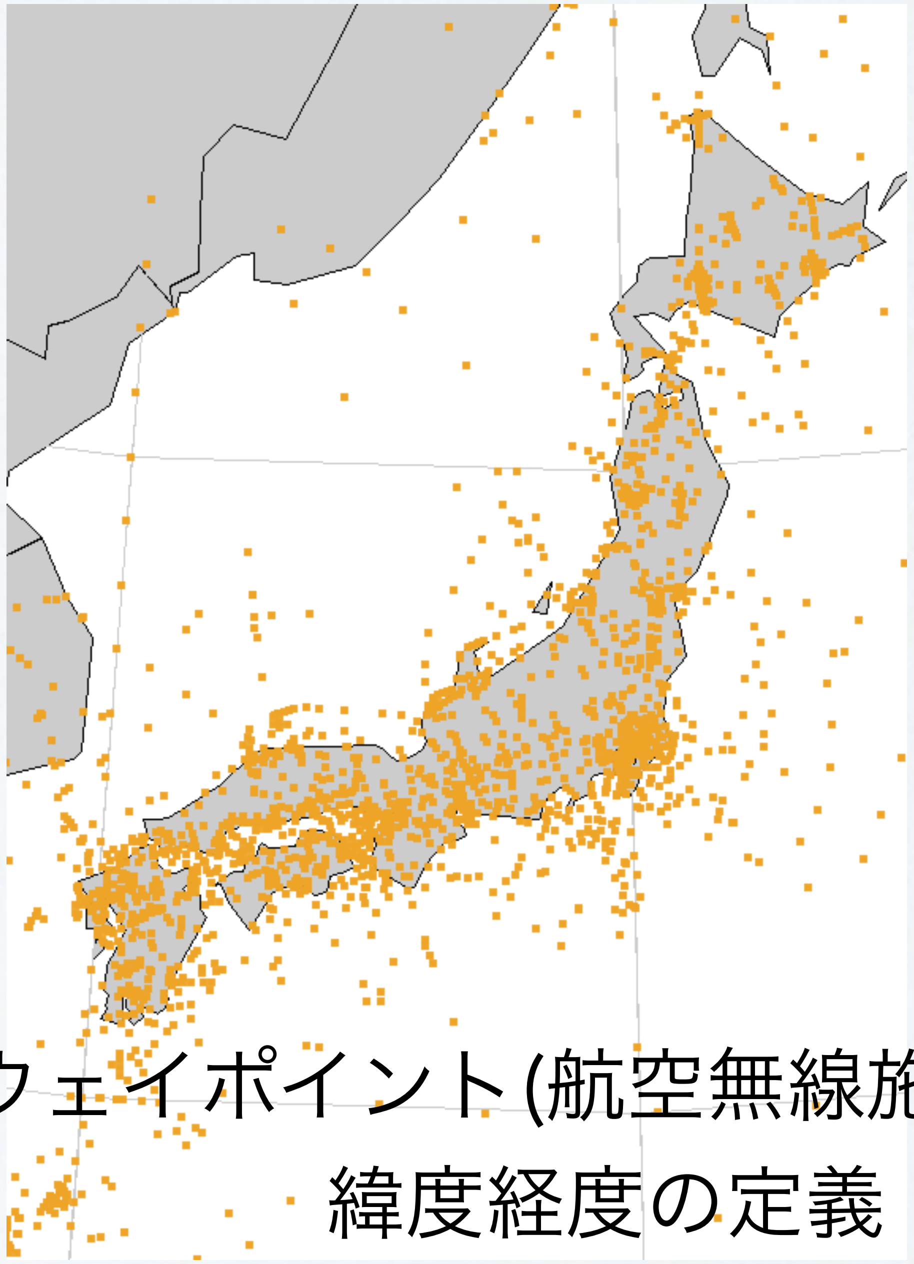
我が国の航空路

レーダ覆域（陸域）

http://www8.cao.go.jp/koutu/taisaku/h20kou_haku/genkyo/h3/h3s2_1.html

- 陸域において定義される
- 出発・目的空港の組合せごとの推奨経路
- 固定の経路構成

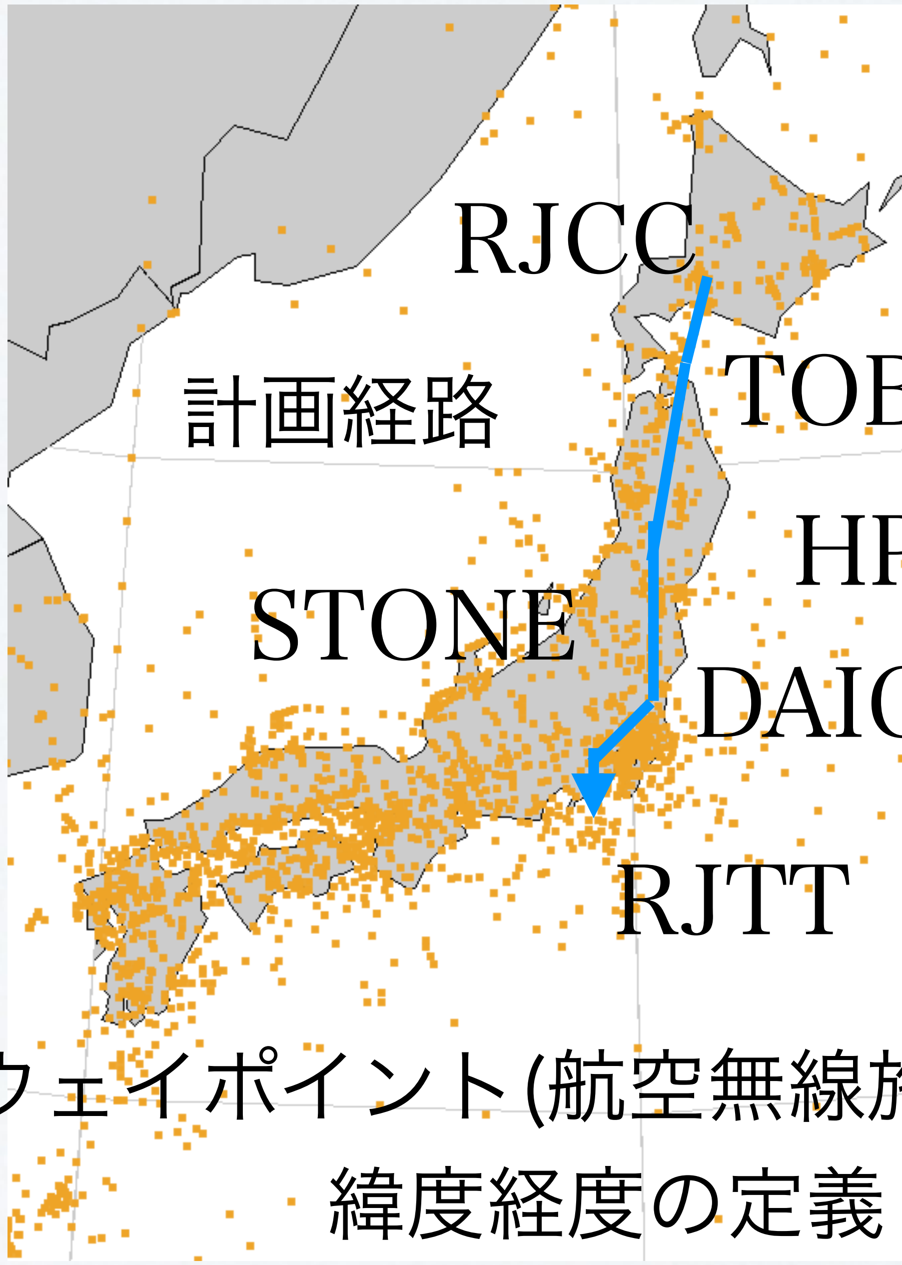
計画経路



ウェイポイント(航空無線施設/地点)

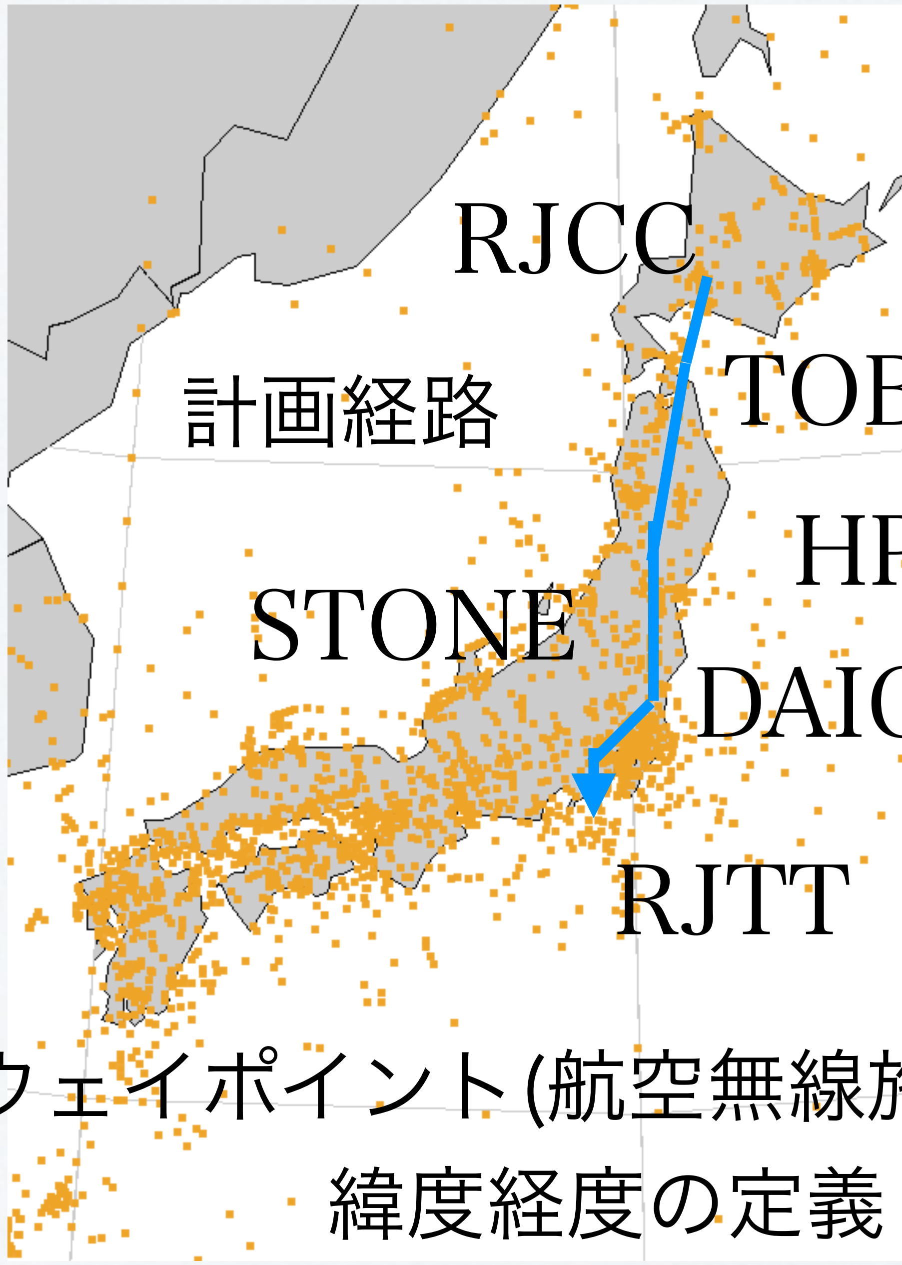
緯度経度の定義

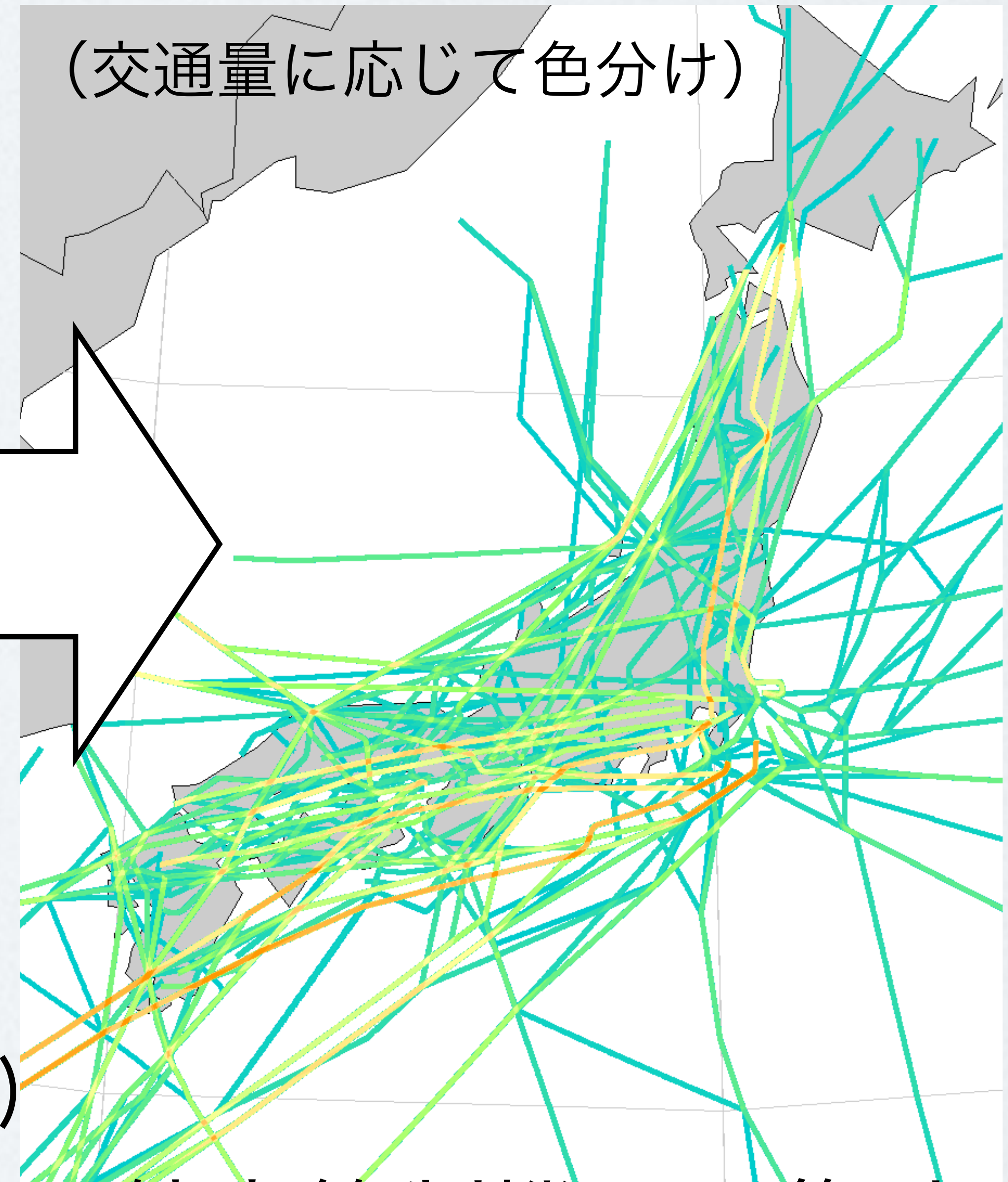
計画経路



- 運航者が飛行前に経路を計画
  - ウェイポイント（WP）の列

計画経路

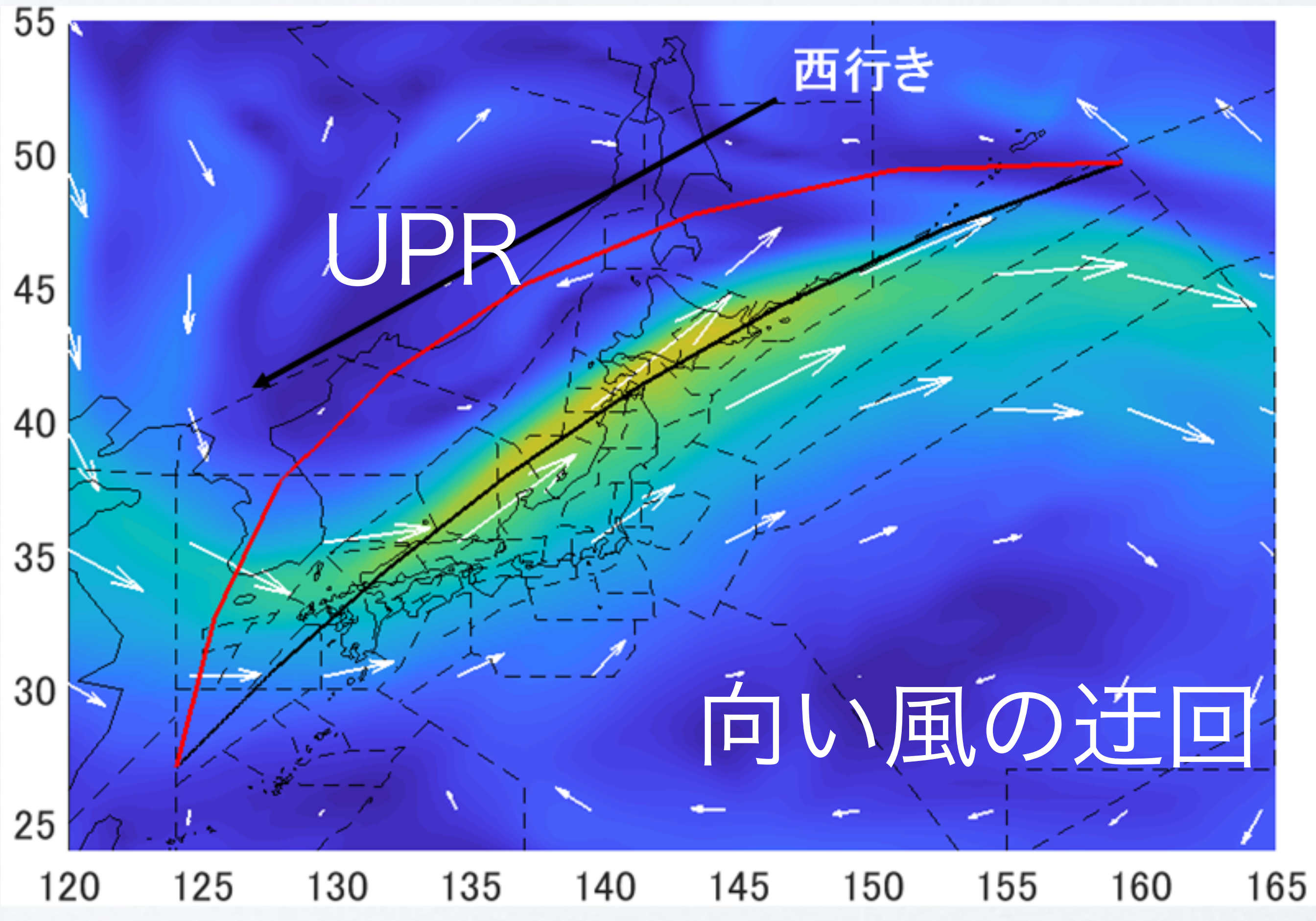




- 運航者が飛行前に経路を計画
  - ウェイポイント（WP）の列
- 航空管制機関で集計
  - 空域全体の飛行の流れ

利用者選択経路

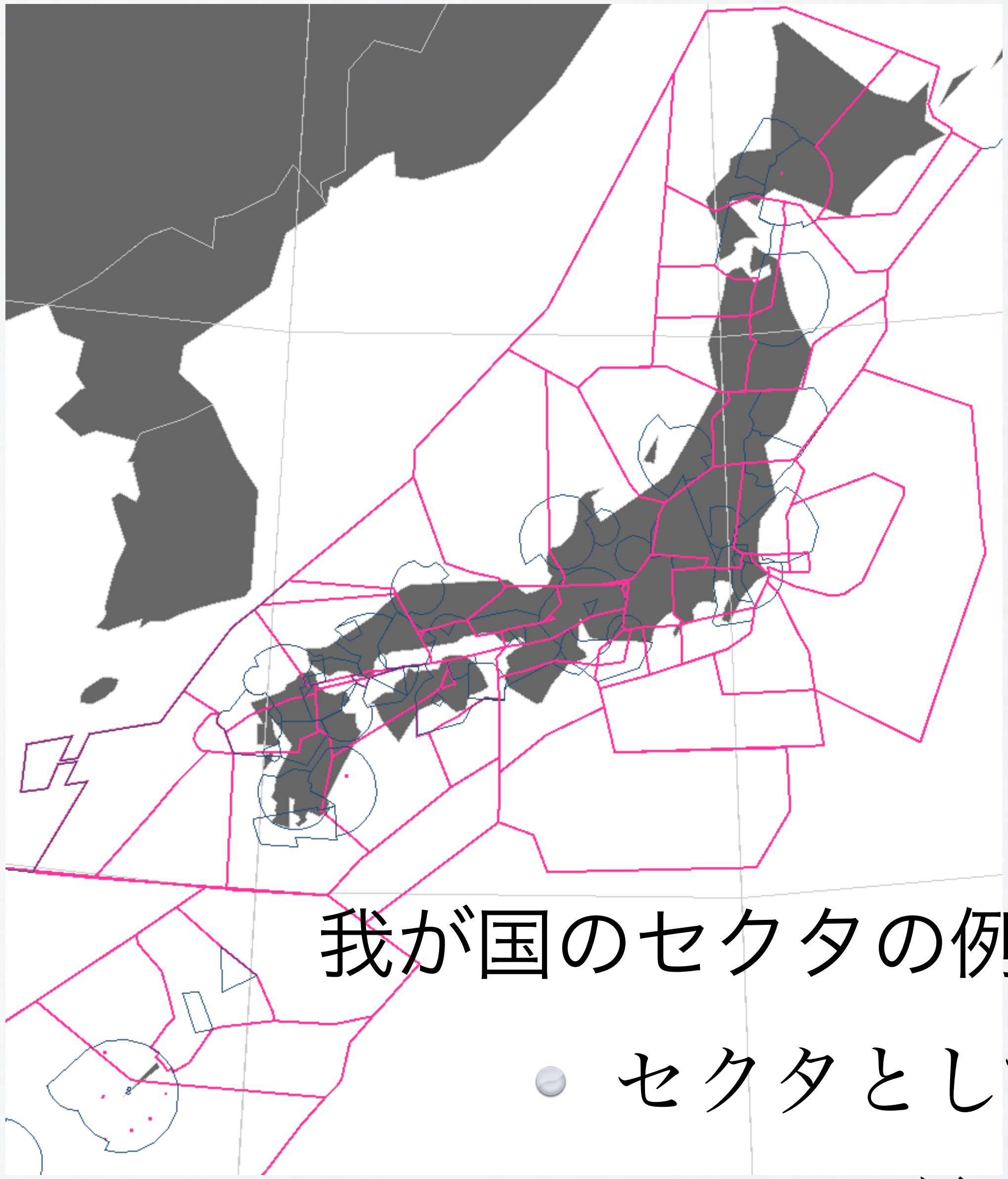
- User Preferred Route：UPR
- 運航者が燃料消費などを加味して計画した経路
  - 風向・風速に応じた様々な経路構成



- 太平洋上で運用
- 陸域への拡大計画

セクタ（空域の単位）

- セクタ ＝ 航空管制官の配置単位
  - 飛行の監視、指示、情報交換（管制作業）

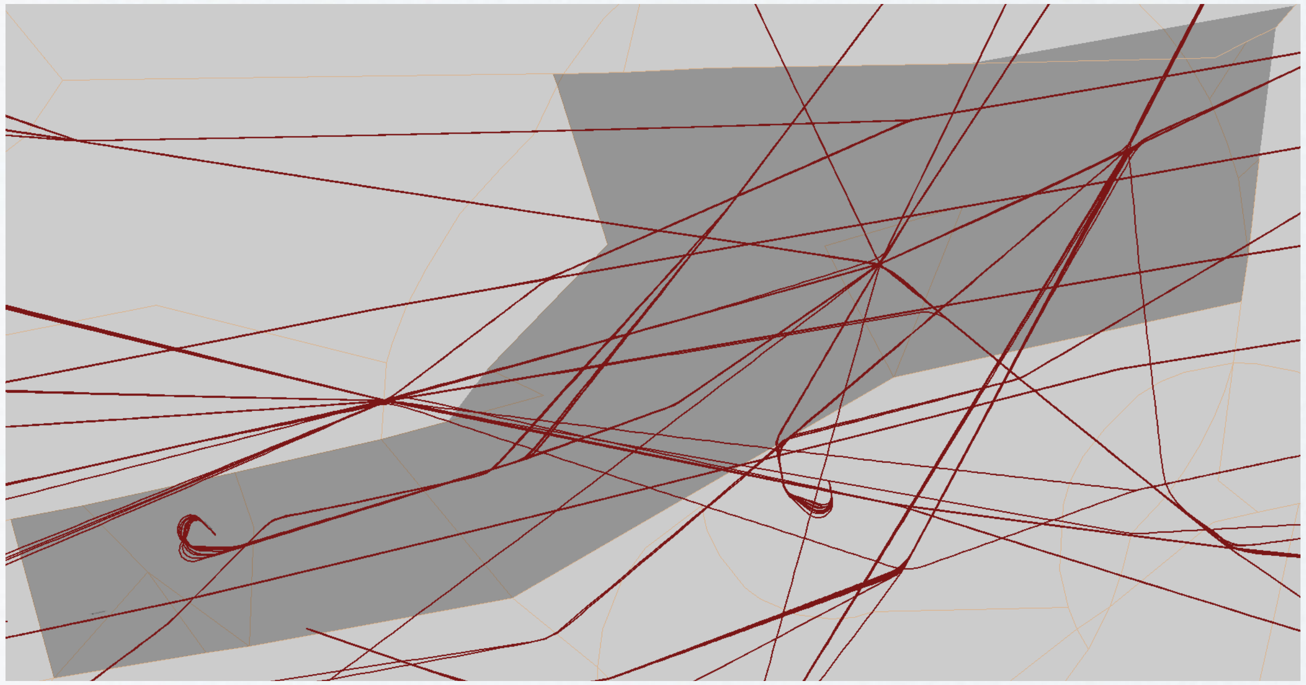


我が国のセクタの例

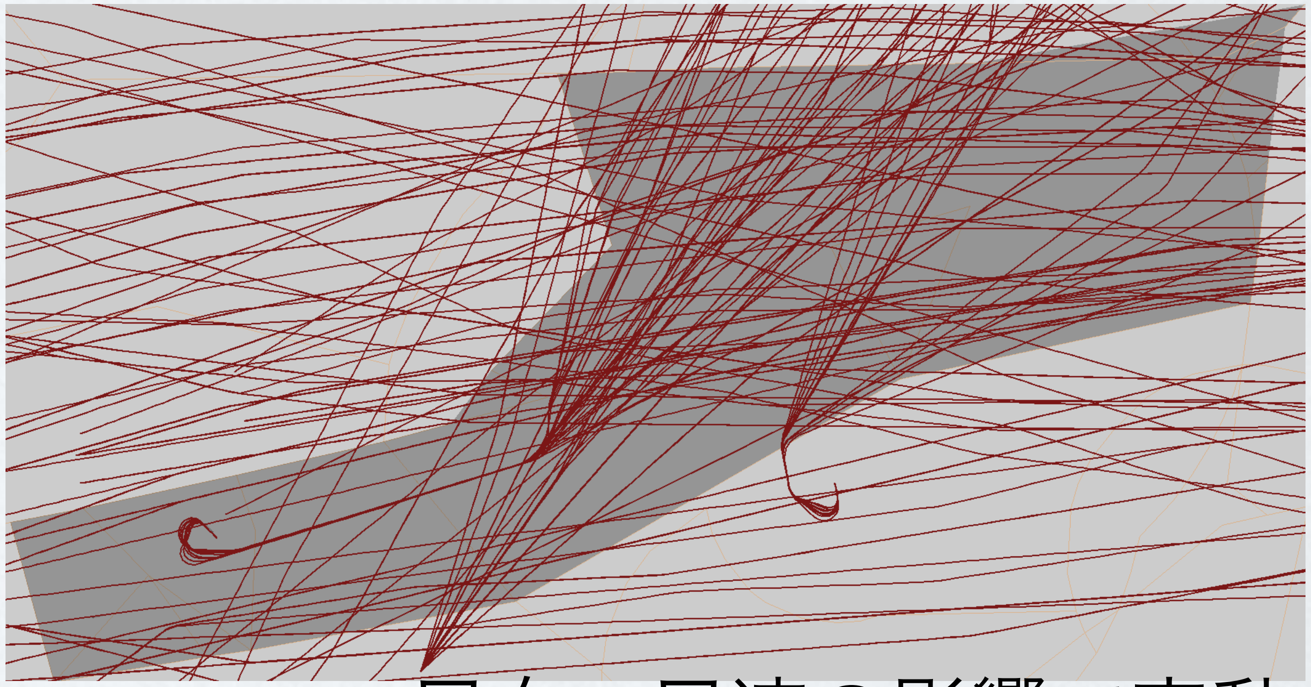
- セクタとしての条件（主要なもの）
  - セクタ最小滞在時間（管制指示時間の確保）
  - 境界線–WP間最小距離（管制指示時間の確保）
  - 凸性（同一航空機の複数回進入の回避）
  - 連結性（飛び地の回避）

効率的運航支援の重要性

現行の推奨経路



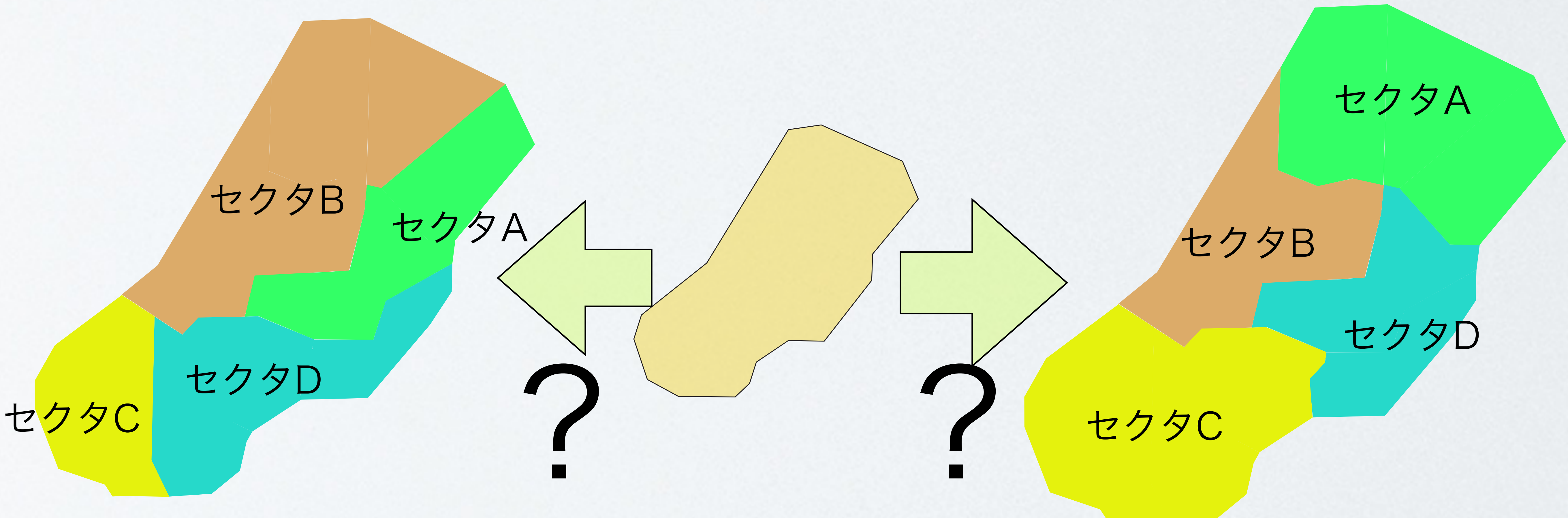
UPR経路の例



風向・風速の影響で変動

- 交通流（経路構成）の交差頻度／取り扱い数の増減
- 管制作業量：交通流に依存
    - セクタ間での管制作業量のばらつきの増大
- 効率的運航支援：経路構成に応じたセクタ境界線設定

空域編成問題（本研究での定義）



（入力）計画経路（WPの列）の集合

（出力）セクタ数とセクタ境界線

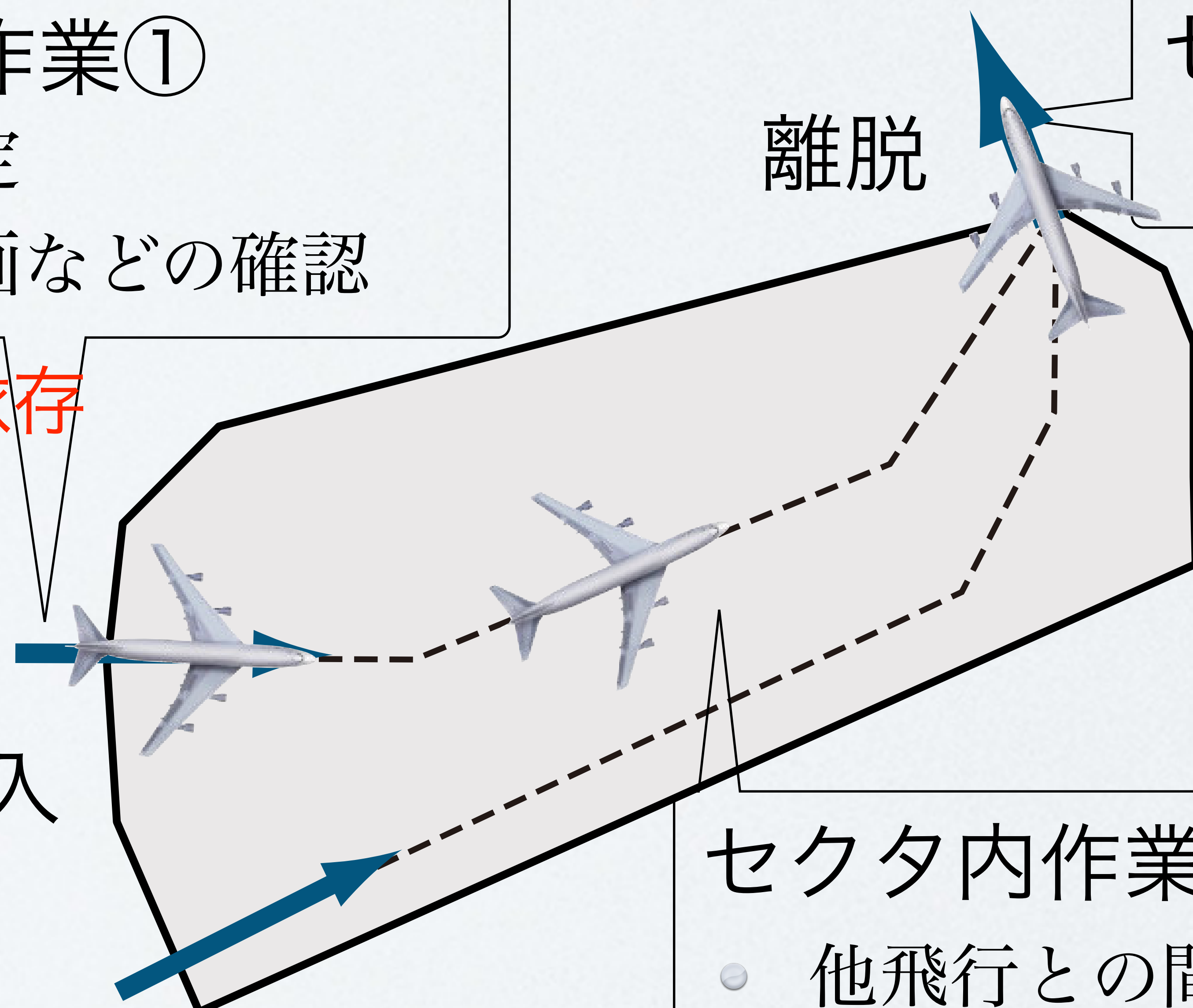
（条件）セクタとして満たすべき条件

各セクタでの管制作業量の平準化 → 最適化

管制作業量のモデル化

セクタ間作業①
- 通信設定
- 飛行計画などの確認

飛行数に依存



セクタ間作業②
- 通信設定

飛行数に依存

飛行距離に依存

セクタ内作業
- 他飛行との間隔などの確認

管制作業量 =

セクタ内作業 + λ (セクタ間作業① + セクタ間作業②)

数理最適化アプローチ

■最適化のターゲット：

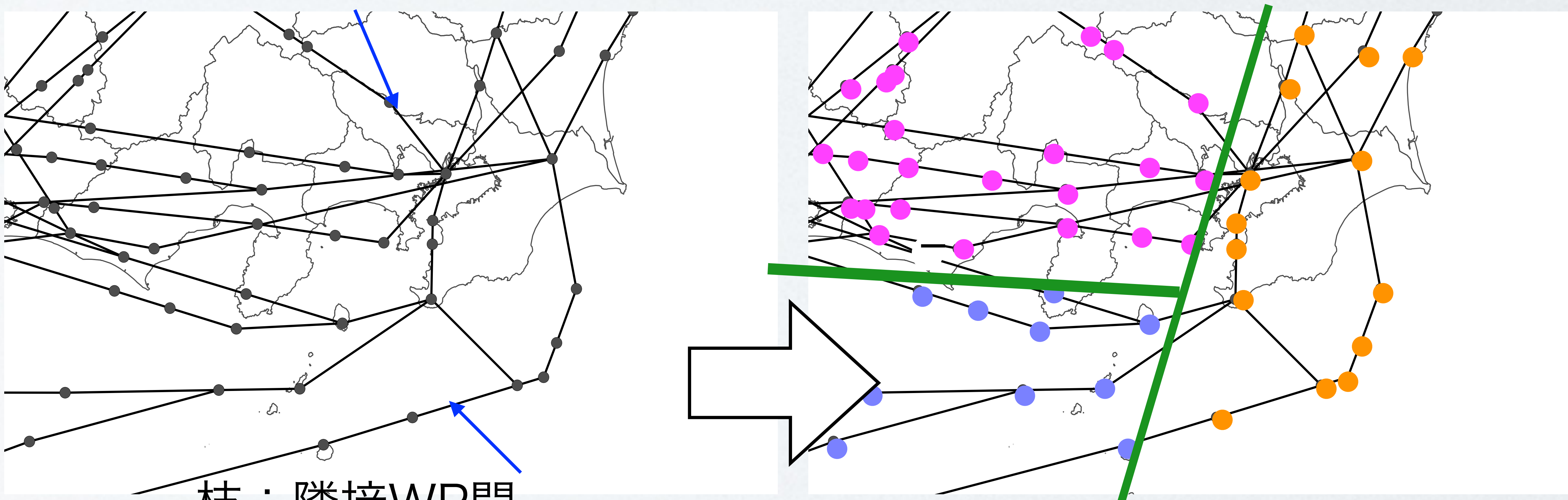
管制作業量の平準化

⇒ 航空機の飛行距離と機数の扱いがポイント

■求解のアプローチ：

	管制作業量和最小化	管制作業量平準化
飛行距離	そのまま	機数で近似
機数	そのまま	そのまま
求解	ハード	比較的容易
意図	・管制作業量の和の最小化を、各セクタでの管制作業量の最小化で実現 ・セクタ間での管制作業量のばらつきを上下限を設定して抑制	そのまま

計画経路のグラフによる表現



重み付きグラフ

$$G = (V, E)$$

（枝への重み付けなどにより
作業量の表現が可能）

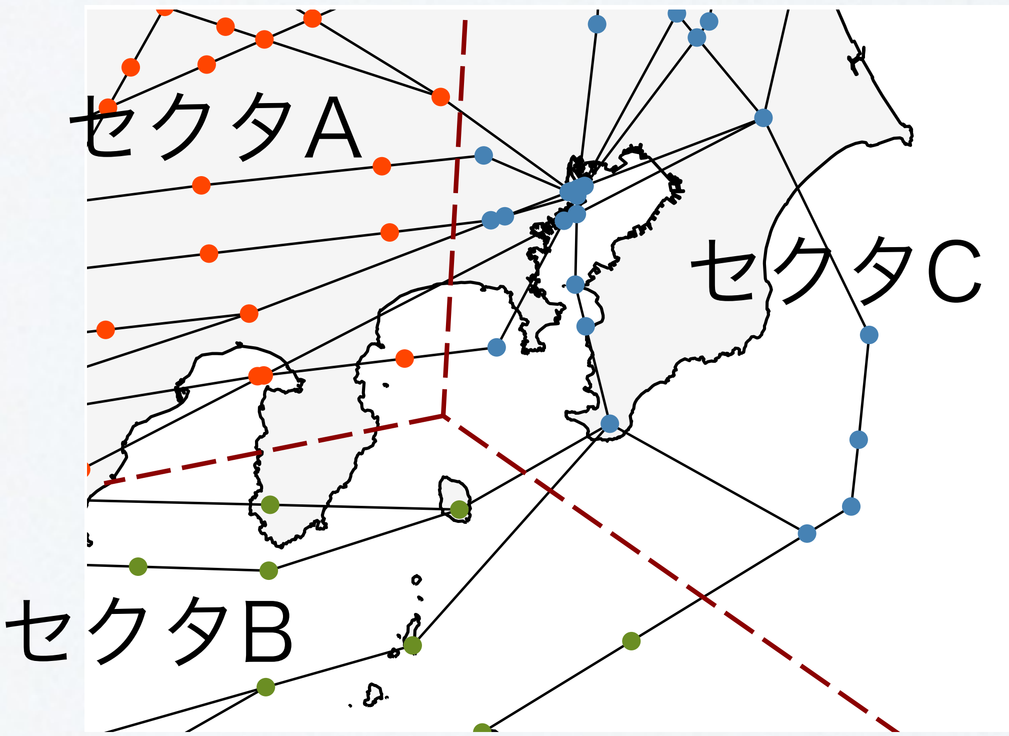
$$G_1, G_2, \cdots, G_k$$

- 部分グラフへの分割
  - 各セクタに対応
  - 大まかな領域

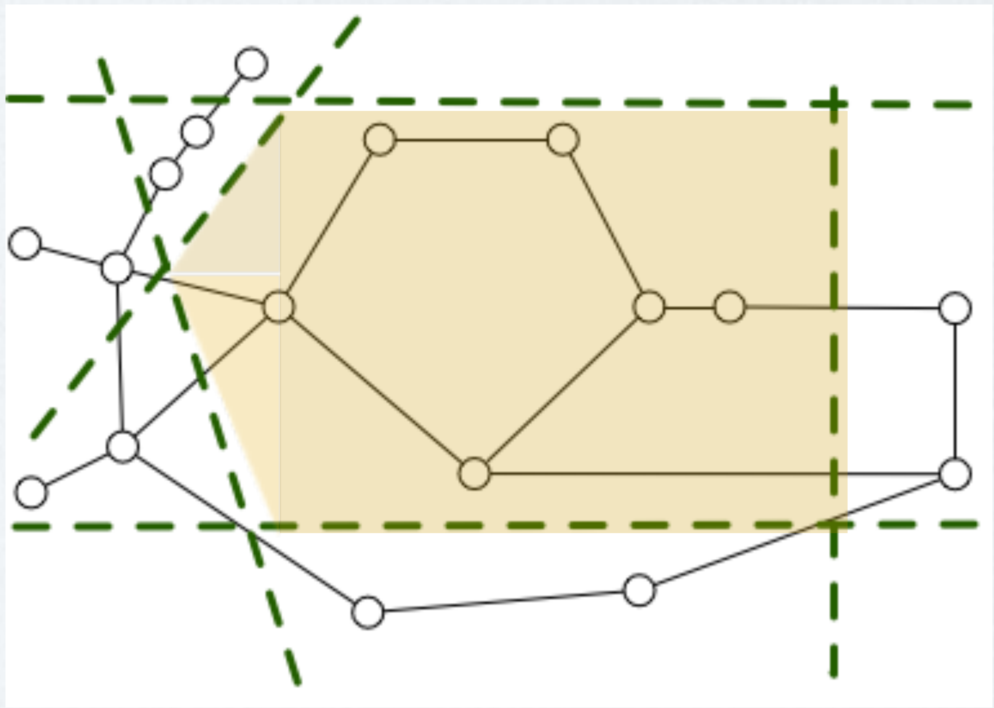
- 背景と目的
- 解法のフレームワーク
  - 管制作業量和最小化
  - 管制作業量平準化
  - 可視化
- テストデータへの適用

管制作業量和最小化問題と列生成アプローチ

～解法構築のアイデア



計画経路とセクタの例



- 空域編成
  - WP全体をグループ分けすることで実現

- セクタ
  - グループ内のWPが必ずつながっているもの

⇒ グラフの木構造を利用可能

管制作業量和最小化問題と列生成アプローチ

～集合分割問題

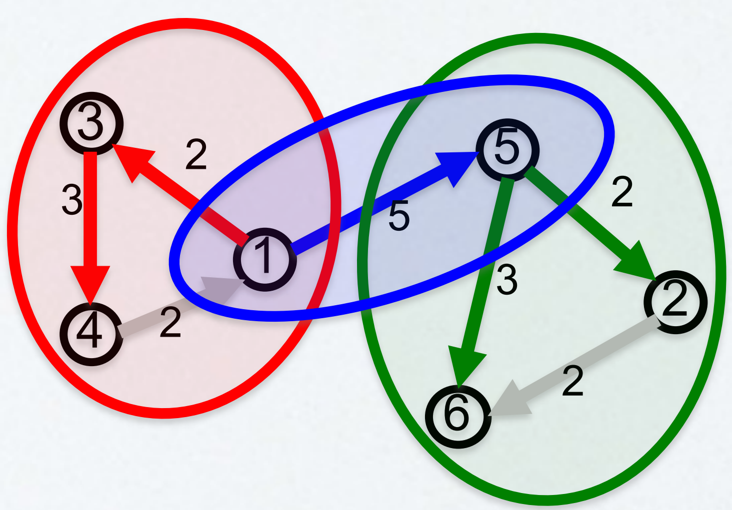
$$
\begin{aligned}
\text{min.}\quad & \boxed{\begin{bmatrix} c_1 & c_2 & c_3 & \dots & \dots & \dots & \dots & c_n \end{bmatrix}} \\
\text{s.t.}\quad & \begin{bmatrix} 1 & 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 1 & 0 & 1 & \dots & \dots & \dots & \dots & 1 \\ \vdots & \vdots & \vdots & \dots & \dots & \dots & \dots & \vdots \\ 0 & 1 & 1 & \dots & \dots & \dots & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}
\end{aligned}
$$

← 管制作業量

セクタ

意思決定変数：選ぶ(1)／選ばない(0)

- セクタを全列挙
  ⇒ 困難
- 列生成法の利用
  ⇒ 効率的に列挙



WPと計画経路

管制作業量

$$
\begin{bmatrix} \boxed{5} & \boxed{7} & \boxed{10} & \cdots\cdots\cdots \end{bmatrix}
$$

$$
\begin{array}{c}
1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6
\end{array}
\begin{bmatrix}
1 & 1 & 0 & \\
0 & 0 & 1 & \\
1 & 0 & 0 & \\
1 & 0 & 0 & \cdots\cdots\cdots \\
0 & 1 & 1 & \\
0 & 0 & 1 &
\end{bmatrix}
$$

セクタ

管制作業量和最小化問題と列生成アプローチ

～アルゴリズム

列生成アルゴリズム

(限定フェーズ)

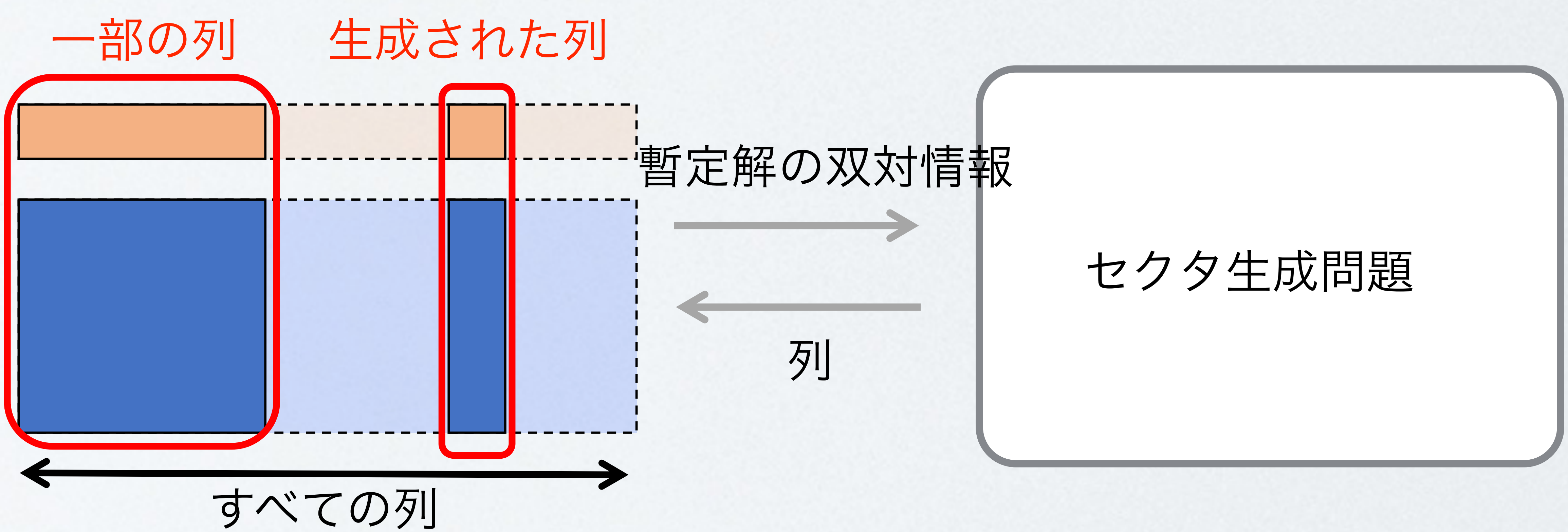
1. （すでに生成した）一部の列だけに限定した部分問題を解き、暫定解を得る

(列生成フェーズ)

2. 必要に応じて列（セクタの候補）を逐次生成する

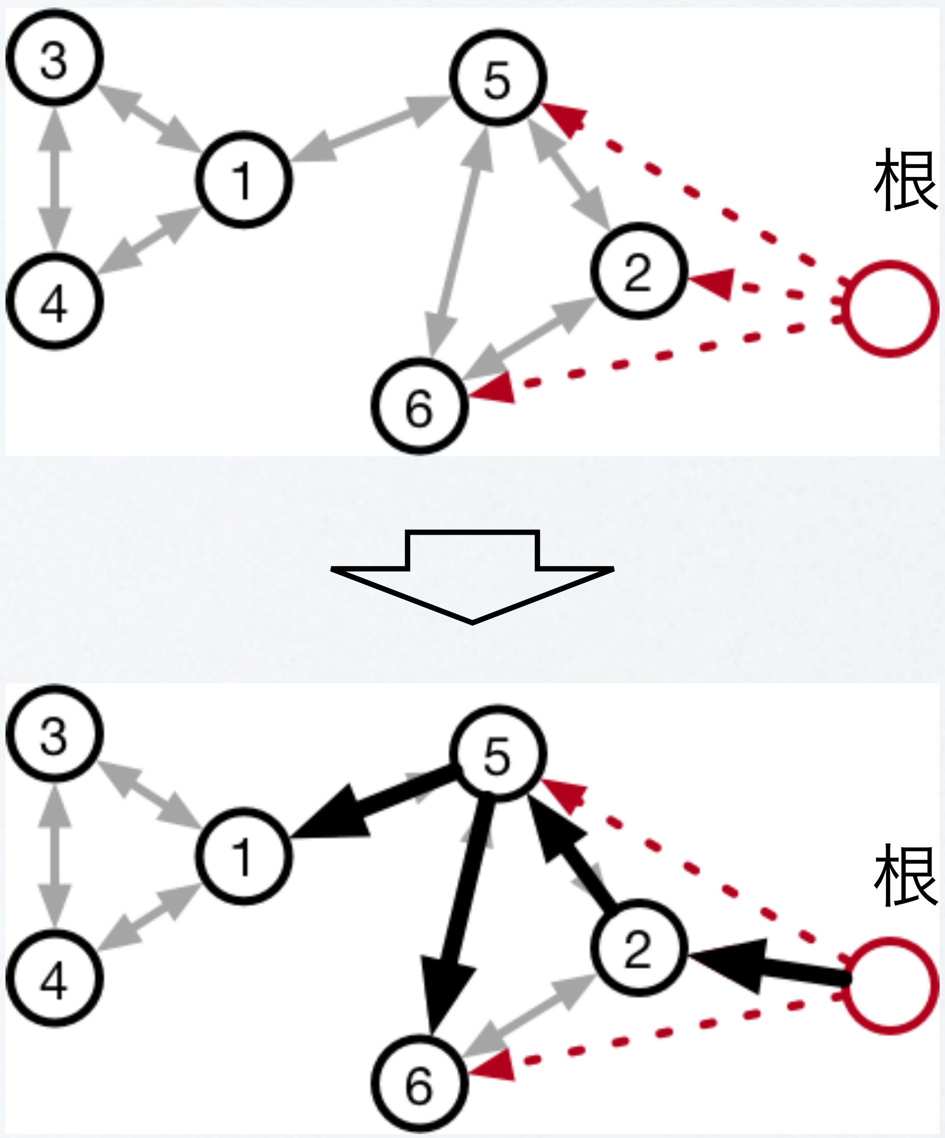
■追加すべき列が存在しないならば終了する

■有用な列が存在するならば、部分問題に追加し、1へ戻る



管制作業量和最小化問題と列生成アプローチ

～列生成



根付き有向木

セクタ（根付き有向木）生成問題

$$
\begin{aligned}
\text{min.} \quad & (1-\theta)\sum_{(i,j)\in E} w_{ij}(1-y_i)y_j + \theta \sum_{(i,j)\in E} w_{ij} d_{ij} z_{ij} - \sum_{i\in V} \lambda_i^* y_i \\
\text{sub. to} \quad & \sum_{j\in\delta(0)} z_{0j} = 1, \\
& \sum_{j\in\rho(i)} z_{ji} = y_i \quad \text{for } i \in V, \\
& \sum_{j\in\delta(i)} z_{ij} \le |V| \times y_i \quad \text{for } i \in V, \\
& l_i + 1 \le l_j + |V|(1 - z_{ij}) \quad \text{for } (i,j) \in E, \\
& \sum_{(i,j)\in E} z_{ij} = \sum_{i\in V} y_i - 1, \\
& y_i \in \{0,1\} \quad \text{for } i \in V, \\
& z_{ij} \in \{0,1\} \quad \text{for } (i,j) \in \bar{E}, \\
& l_i \in \mathbb{R} \quad \text{for } i \in V.
\end{aligned}
$$

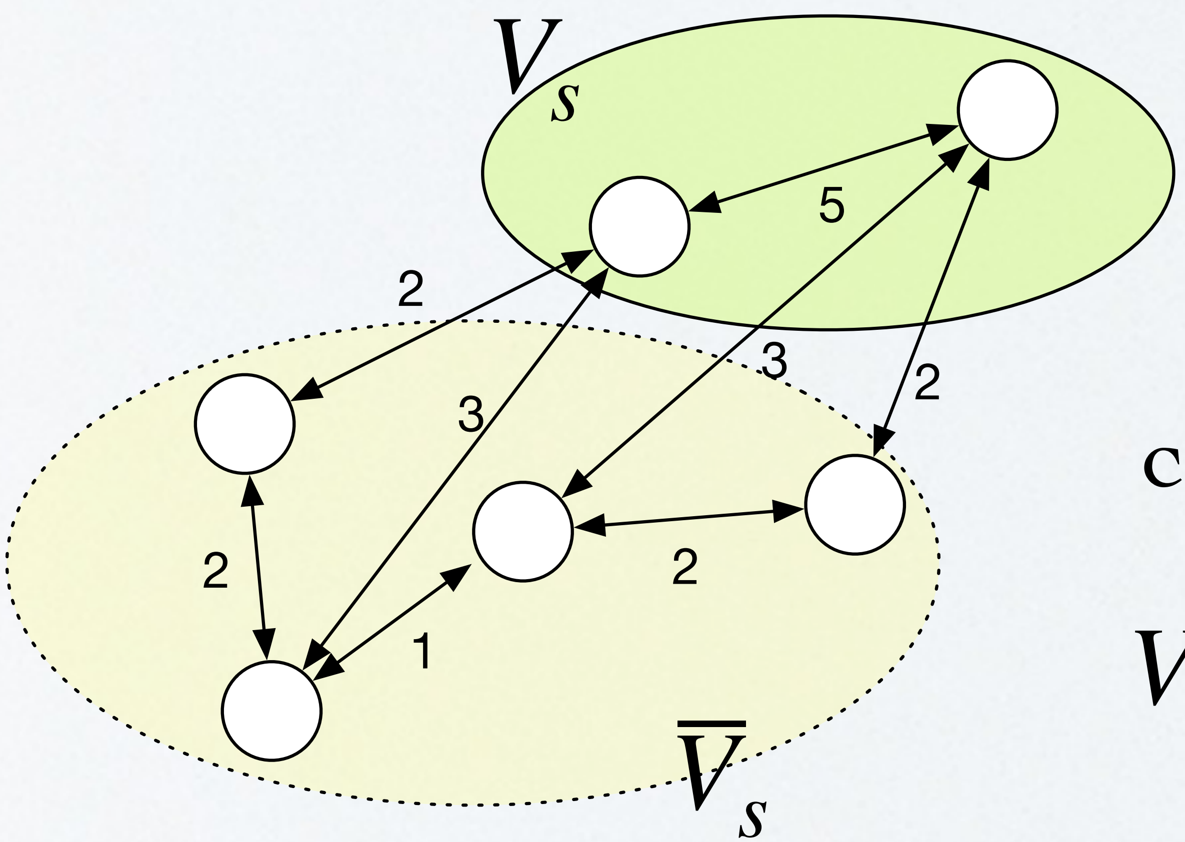
$$
y_i = \begin{cases} 1 & \text{waypoint } i \text{ in the tree} \\ 0 & \text{otherwise} \end{cases}
$$

$$
z_{ij} = \begin{cases} 1 & \text{segement } (i,j) \text{ in the tree} \\ 0 & \text{otherwise} \end{cases}
$$

- 背景と目的
- 解法のフレームワーク
  - 管制作業量和最小化
  - 管制作業量平準化
  - 可視化
- テストデータへの適用

管制作業量の定式化①

セクタ間の作業量



$$\mathrm{cut}(V_s, \overline{V_s}) = (2+3) + (3+2) = 10$$

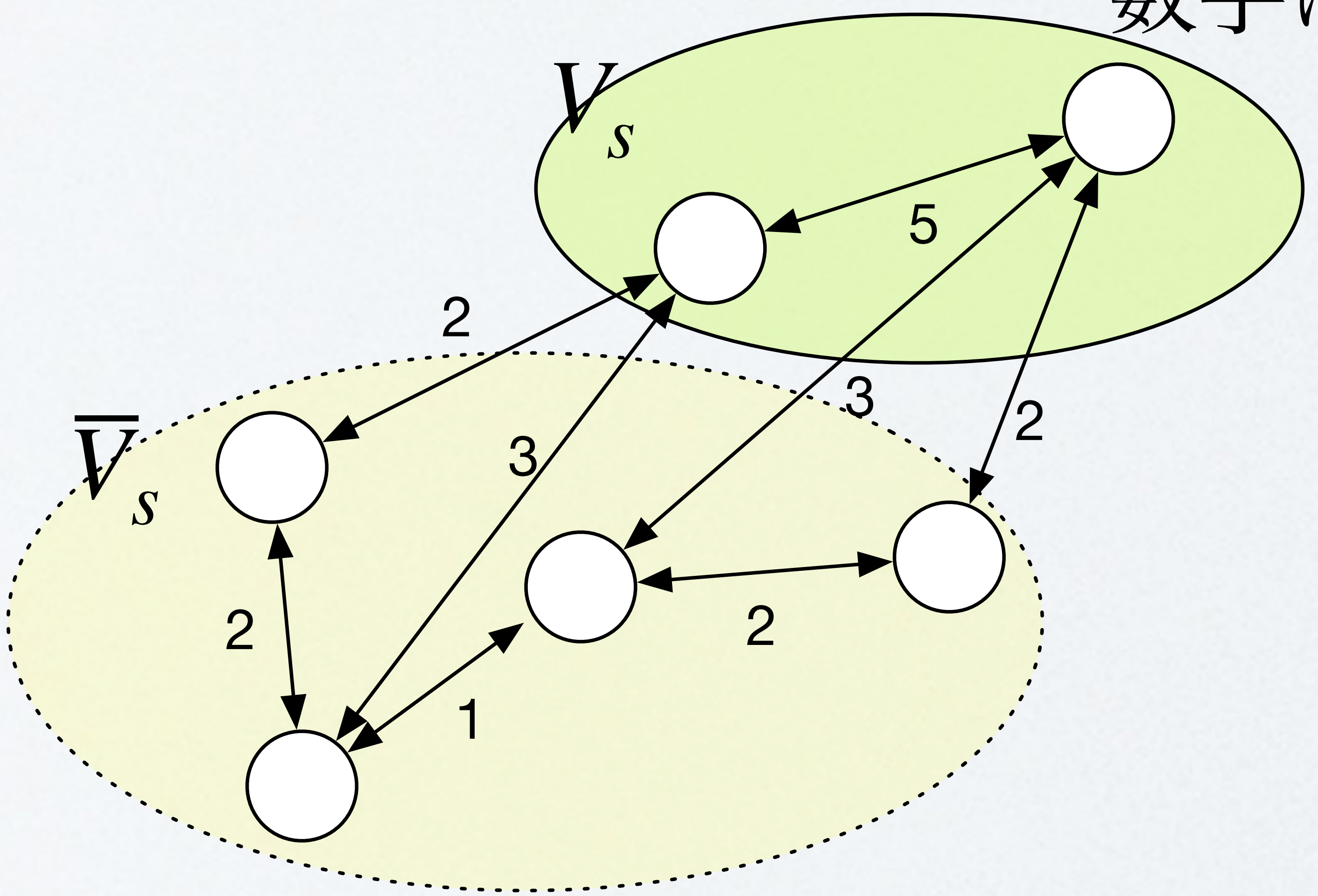
V_s に関するカット枝の重み和

数字は通過機数を表す

管制作業量の定式化②

セクタ内の作業量

数字は通過機数を表す



$$\mathrm{vol}(V_s) = (2+3+5) + (5+3+2) = 20$$

V_s に含まれる頂点の重みつき次数

Normalized-cutの定義

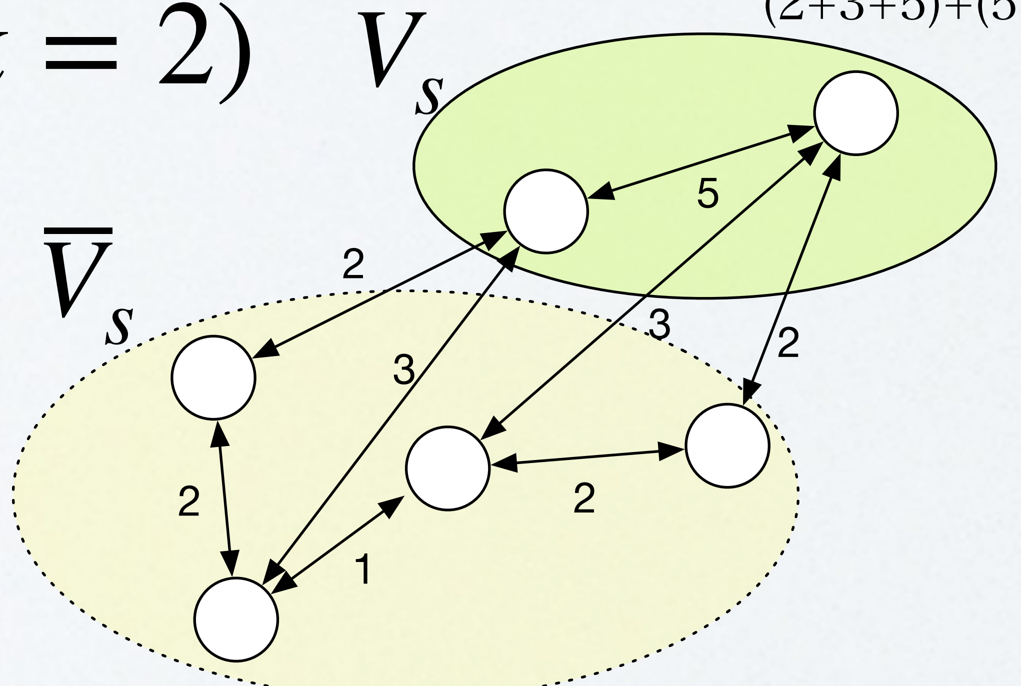
$$\sum_{s=1}^{k} \frac{\mathrm{cut}(V_s, \overline{V_s})}{\mathrm{vol}(V_s)}$$

$\mathrm{cut}(V_s, \overline{V_s})$ ← V_s に関するカット枝の重み和

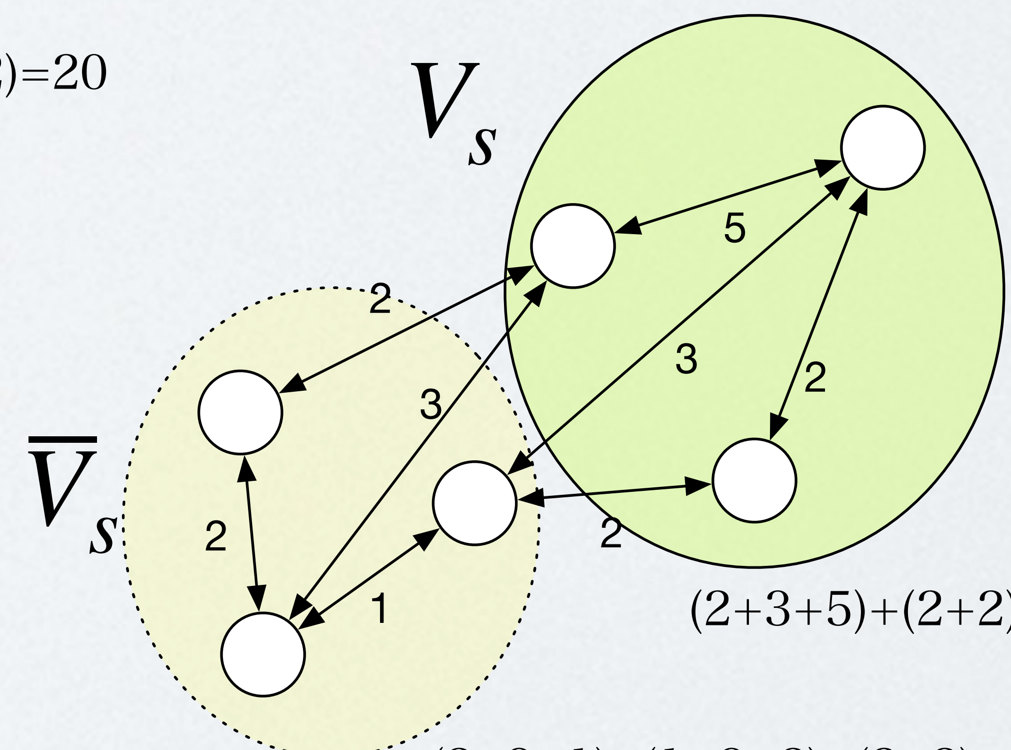
$\mathrm{vol}(V_s)$ ← V_sに含まれる頂点の重み付き次数の和

$(k = 2)$

(2+3+5)+(5+3+2)=20



(2+2)+(2+3)+(1+3+2)+(2+2)=19



(2+3+5)+(2+2)+(5+3+2)=24

(2+3+1)+(1+3+2)+(2+2)=16

$$\sum_{s=1}^{k} \frac{\mathrm{cut}(V_s, \overline{V_s})}{\mathrm{vol}(V_s)} = \frac{10}{20} + \frac{10}{19} = 1.02$$

$$\sum_{s=1}^{k} \frac{\mathrm{cut}(V_s, \overline{V_s})}{\mathrm{vol}(V_s)} = \frac{10}{24} + \frac{10}{16} = 1.12$$

算出の手順

- 評価関数Normalized-cutの最小化
  - セクタ間作業量を小さく（分子）
  - セクタ内作業量を平準化（分母）

$$\sum_{s=1}^{k} \frac{\mathrm{cut}(V_s, \overline{V_s})}{\mathrm{vol}(V_s)}$$

- スペクトル法の適用

1: ラプラシアン L に関する一般化固有値問題 $L\boldsymbol{u} = \lambda D\boldsymbol{u}$ を解き，小さい方から k 個の固有値 λ_j と対応する固有ベクトル $\boldsymbol{u}^j$ を求める．$\mathrm{LB} \leftarrow \sum_{j=1}^{k} \lambda_j$.
2: **for** $i \in V$ **do**
3: $\quad \boldsymbol{z}^i \leftarrow (u_i^1, u_i^2, \ldots, u_i^k) \in \mathbb{R}^k$
4: データ $\{\boldsymbol{z}^1, \boldsymbol{z}^2, \ldots, \boldsymbol{z}^n\}$ に対して k-means クラスタリングを実行し，分割 $\{V_1, V_2, \ldots, V_k\}$ を得る．
5: $\mathrm{UB} \leftarrow \mathrm{Ncut}(V_1, V_2, \ldots, V_k)$ として LB と UB を出力する．

(Shi J., Malik J. 2000)

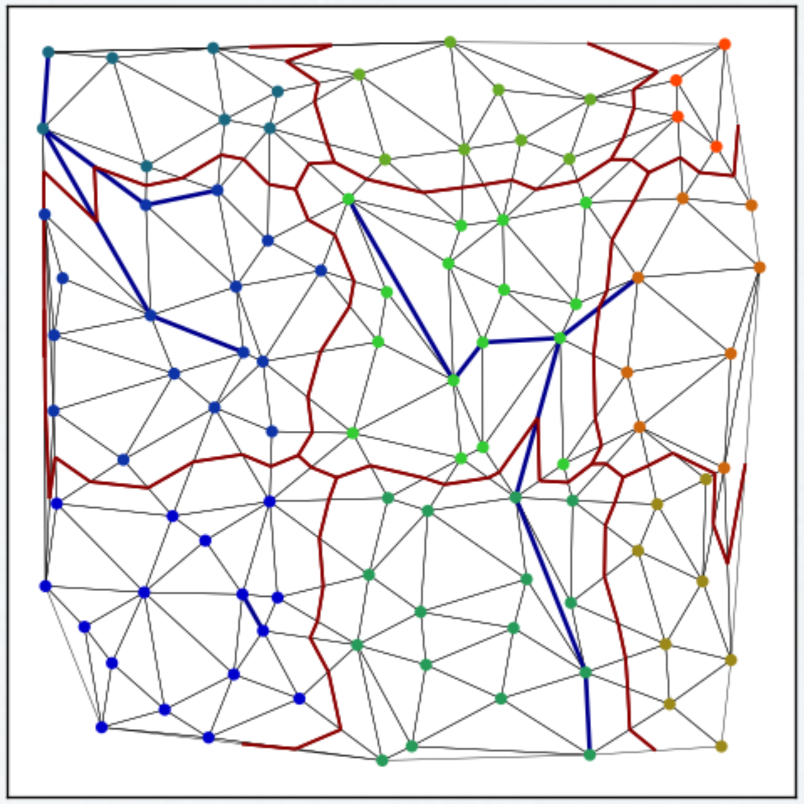
- 背景と目的
- 解法のフレームワーク
  - 管制作業量和最小化
  - 管制作業量平準化
  - 可視化
- テストデータへの適用

セクタ境界線の決定

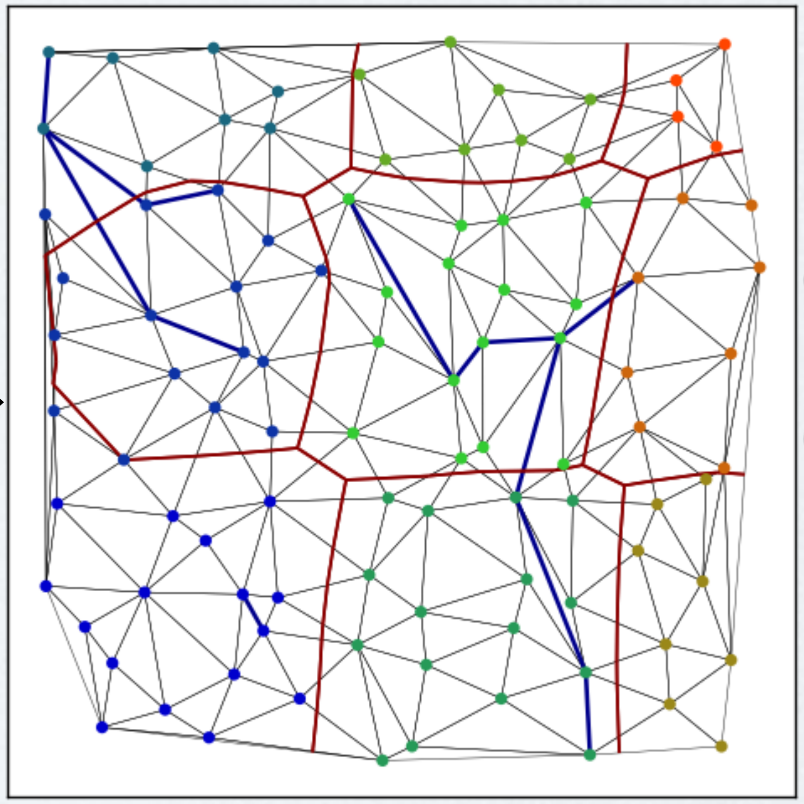
- アルゴリズムの適用結果＝WP全体のグループ分け
  - 各セクタ境界線の形状は未決定

⇨ 境界線決定アルゴリズムの開発（2段階）

初期分割（制約付き三角形分



補正（競合学習法）



- 凹凸の低減
- 凸に近い

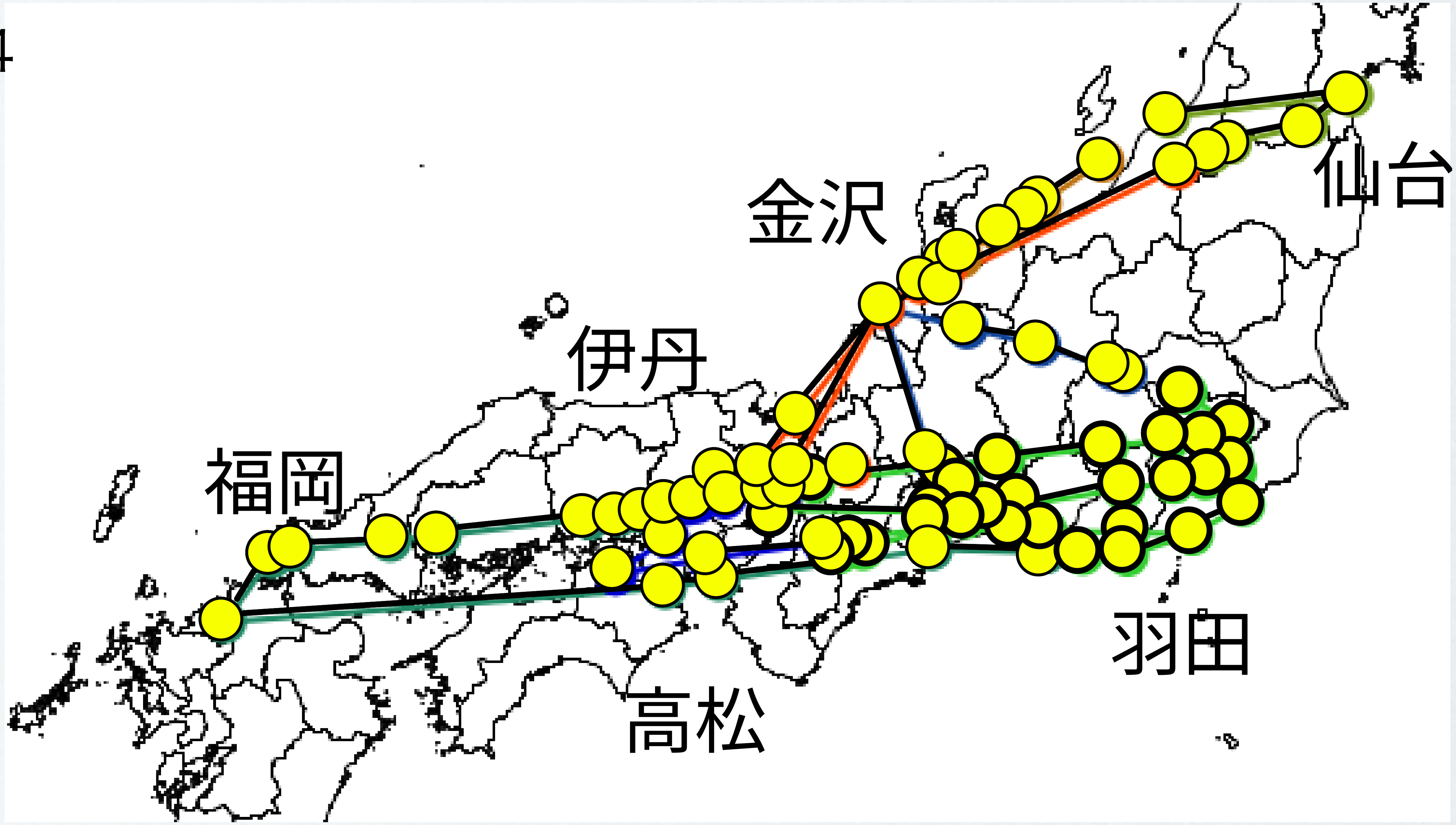
- 背景と目的
- 解法のフレームワーク
  - 管制作業量和最小化
  - 管制作業量平準化
  - 可視化
- テストデータへの適用

テストデータの作成

- 便数が多い路線の選択
- 発着空港が適度に分散

部分的な空域モデル

- WP数：84
- 枝数：99



管制作業量和最小化の結果

（実装）Python3.6.3, Gurobi Optimizer 8.0.1（最適化）

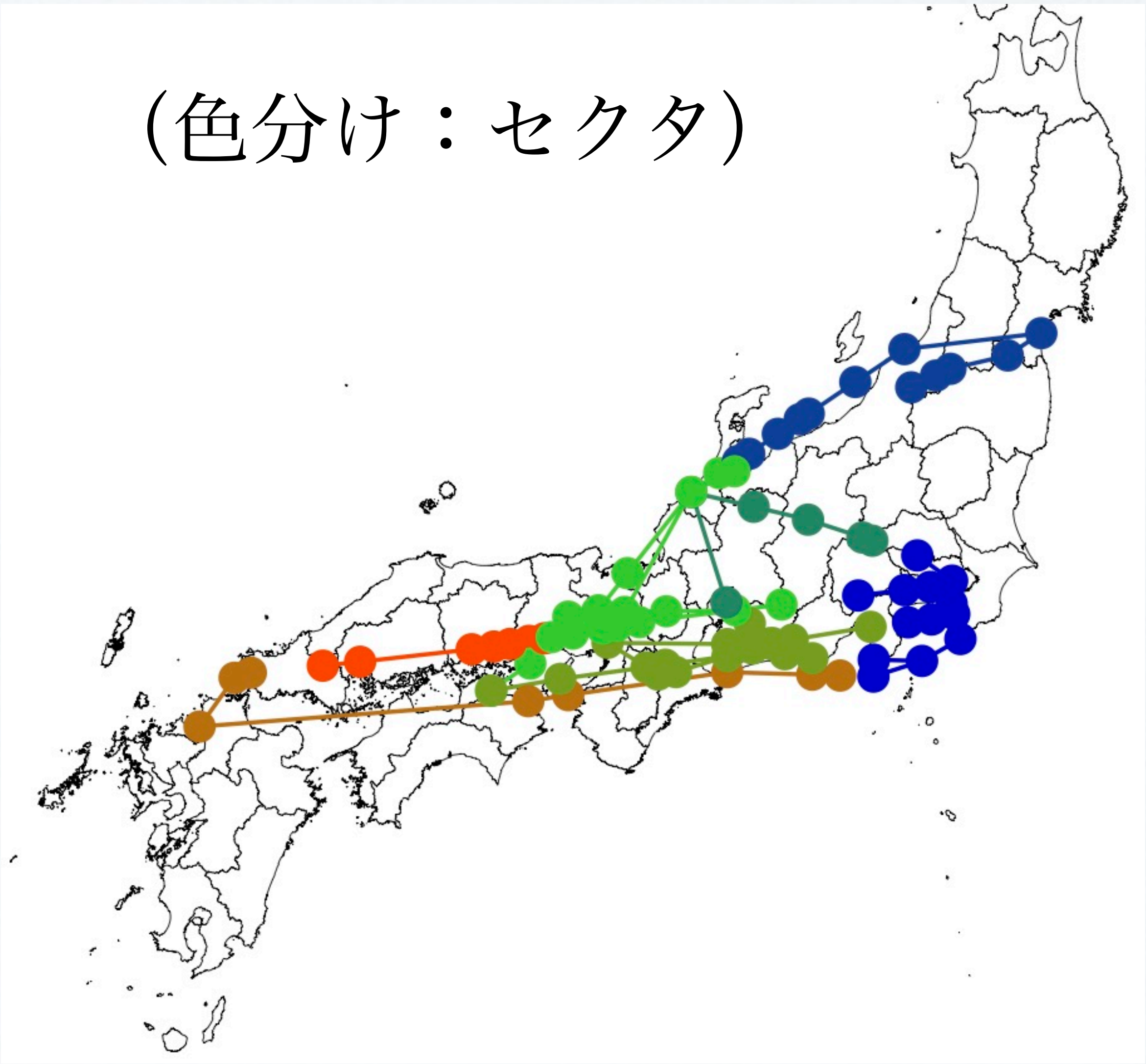
セクタ番号	WP数	作業量
#1	5	1.91
#2	3	2.05
#3	3	1.00
#4	5	1.76
#5	15	1.40
#6	40	4.15
#7	13	2.41

作業量：最大作業量との比

- 作業量のばらつき：最大で4倍程度

管制作業量平準化の結果

（実装）Python3.6.3, SciPy（線形計算）+ scikit-learn（k-means）



セクタ番号	WP数	作業量
#1	19	1.38
#2	19	2.00
#3	12	1.00
#4	12	1.63
#5	6	2.13
#6	8	1.63
#7	8	1.63

作業量：最大作業量との比

- 作業量のばらつき：最大で2倍程度

おわりに

空域編成に対する2つの最適化アプローチ

- モデル化と解法の提案
- 解法の実装
  - テストデータへの適用

今後の課題

- 我が国全体の空域モデルへの適用


Tokyo International Airport
35
FLIGHT CATERING
TFK
JAPAN AIRLINES
JAL