

Title	CARATS Open Data における航空機位置のセクター決定問題について
Author(s)	徳丸, 翔也
Citation	
Issue Date	2018-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/15188
Rights	
Description	Supervisor:平石 邦彦, 先端科学技術研究科, 修士 (情報科学)

修士論文

CARATS Open Data における 航空機位置のセクター決定問題について

1610132 徳丸翔也

主指導教員 平石邦彦

審査委員主査 平石邦彦

審査委員 緒方和博

上原隆平

浅野文彦

北陸先端科学技術大学院大学
先端科学技術研究科[情報科学]

平成 30 年 2 月

概要

2015年2月から国土交通省が CARATS Open Data という、日本の上空を飛ぶ航空機の情報が記載されているデータを提供開始した。これを機に国内ではそのデータを用いた研究が盛んに行われるようになった。

航空機が飛ぶ空域はセクターに分割されている。それぞれのセクターには、航空機の容量があり、人間の制御で問題なく航空の交通を処理するために、セクター容量が用いられる。セクター容量を超えると遅延や事故が発生する可能性が高まる。空域の需要能力のバランスを予測して、解決するためには、最適化と高速化技術が必要となる。

そこで本稿では、CARATS の実データを用いて、それぞれの航空機が任意の時間にどのセクターに存在するかという情報を付与する処理時間の短縮する手法を提案、評価を行う。その結果を比較することで、手法評価の考察を行う。

目次

第1章	はじめに	1
1.1	研究背景	1
1.2	研究目的	2
1.2.1	問題の定義 —任意の時間におけるスナップショット—	2
1.3	本論文の構成	3
第2章	準備	4
2.1	関連研究	4
2.2	今日の航空機について	5
2.2.1	航空機の巡航時の振る舞い	5
2.2.2	自動飛行システム	5
2.2.3	航空路の管制	5
2.3	CARATS Open Data	6
2.4	日本上空のセクター	7
2.5	CGAL ライブラリ—bounded_side_2—	8
2.6	全点探索による航空機のセクター情報の付与	10
第3章	提案手法	11
3.1	概要	11
3.2	パラメータ設定と安定区間	11
3.2.1	安定区間を決定するパラメータ	11
3.2.2	安定区間決定の流れ	13
3.3	安定区間のセクター情報決定	15
3.3.1	航空機の物理特性を利用した大まかな軌道の予測	15
3.3.2	安定区間のセクター情報決定の流れ	16
第4章	評価実験	18
4.1	実験内容と評価基準	18
4.2	実験環境	18

4.3 実験結果と考察	19
4.3.1 評価実験	19
4.3.2 パラメータ増加に伴う処理時間の増減	25
4.3.3 予想交点との誤差率	26
4.4 セクター探索時の直線を描く際に適した地図	28
第5章 おわりに	30
謝辞	31
研究業績	32
参考文献	33

第1章 はじめに

1.1 研究背景

2015年2月から国土交通省がCARATS Open Dataという、日本の上空を飛ぶ航空機の情報が記載されているデータを提供開始した。これを機に国内ではそのデータを用いた研究が盛んに行われるようになった。発展途上国を中心とした世界各国の航空旅客輸送数は増加傾向にある。先進国でも移動手段として航空機を利用する人口が増え、日本も例外ではない。旅客数や、輸送数の増加に対し、空港の整備や空域の再編、空域全体を取り仕切る管制の工夫によって、航空容量の拡大や効率的な制御が求められてきた。その中で、航空交通管理（Air Traffic Management：ATM）は、世界各国で近年注目を集めており、研究、開発が欧米を中心に進められている。日本では、2006年からATMセンターが福岡で運用開始され、航空交通流管理（Air Traffic Flow Management：ATFM）や協調的意思決定システムによる空域管理などの取り組みが実際に運用されており、今後もより効率化が期待されている[1]。

航空機の飛行計画は、混雑度や天候に応じて、管制官や操縦士によって修正されることがある。補強航空交通管理システム（Enhanced Traffic Management System：ETMS）は、現在飛行中の便や気象情報などをリアルタイムでモニターし、最短20分から最高15時間先までの交通量予測も行うシステムである。予測の正確さは、増加する予測時間によって減少する。将来の交通量予測は時間の遅延を軽減するためにとっても重要である[2]。

航空機が飛ぶ空域はセクターに分割されている。セクターとは、航空機を安全かつ効率的に航空交通管制部が管制できるよう、空港の位置や飛行経路の構成、交通量等を考慮して、適切な大きさ、範囲、形状を定めるものである。セクターが大きすぎると、管制官の扱う航空機の数が増えすぎてしまい、交通量を抑えなければならない。また、セクターの形状が適切ではないと、航空機がスムーズな運航を阻むような制限が求められ、管制の制御が非効率になってしまう。このような状態に陥ると、全体的な遅延が発生する。遅延を軽減するために、適切なセクターの配置が与えられる[3]。それぞれのセクターには、航空機の容量があり、人間の制御で問題なく航空の交通を処理する中で、安全に、そして最小限の遅延に抑えるためには、セクター容量内での制御が求められる。セクター全体での需要能力のバランスを予測して、解決するためには、最適化と高速化技術が必要となる。

1.2 研究目的

CARATS Open Data には、約 10 秒ごとの航空機の位置が記載されている。セクター容量に準じた混雑予測を考える際、それぞれの航空機の現在地を正確に知る必要は無く、それぞれの航空機が、いつセクターを移動したかをより高速に処理することがリアルタイムに混雑予測をする上で重要となってくる。また、航空機は、着陸と離陸付近以外は、直線に近い軌道を取ることから、軌道予測がしやすい。このことから、航空機が安定な飛行をしている場合、処理の削減を行うことで、より高速に航空機のセクター情報の取得を目指す。

1.2.1 問題の定義 —任意の時間におけるスナップショット—

本手法では、セクター予測モデルの開発に向けて、航空機の軌道データの中から、セクターをまたぐ点を探索する手法について述べる。セクターをまたぐ点とその時刻を用いることで実現できることの一つとして、任意の時間において、すべての航空機がどのセクターに存在するか(スナップショット)を求められる。ある時刻 t のスナップショットを求める際、航空機のセクターの境界点の時刻の関係が、 $a < t < b$ となるような時刻 a, b が存在する場合、その航空機は時刻 a で突入したセクター X のなかに存在するということがわかる。

本手法によって求められるスナップショットの例を、表 1.1 に示す。セクターには、含有関係があることから、複数セクター上に存在する航空機が存在する。航空の交通を制御、処理するためのセクター容量はこのスナップショットから判断可能である。

表 1.1 ある時刻の航空機のセクター配置の一部

セクター	航空機の名称							
["u'T12", " u'T12b", " u'T13"]	FLT0413	FLT0379	FLT0285	FLT0353	FLT0297	FLT0299	FLT0395	FLT0368
["u'T26"]	FLT0338	FLT0382	FLT0392	FLT0278				
["u'T23"]	FLT0252	FLT0313	FLT0263	FLT0268	FLT0411			
["u'T21"]	FLT0243	FLT0386	FLT0259	FLT0358	FLT0359	FLT0369		
["u'T11"]	FLT0239	FLT0242	FLT0178	FLT0171	FLT0172	FLT0134	FLT0132	FLT0216
["u'F05"]	FLT0336							
["u'F02"]	FLT0403	FLT0384	FLT0408	FLT0393				
["u'T01"]	FLT0401	FLT0325	FLT0320	FLT0391				
["u'T22"]	FLT0323	FLT0366	FLT0367	FLT0342	FLT0421	FLT0398	FLT0329	FLT0326
["u'F08"]	FLT0248	FLT0365	FLT0230	FLT0318	FLT0416			
["u'F01", " u'F11a"]	FLT0361							
["u'S02"]	FLT0330	FLT0284	FLT0269	FLT0296	FLT0346	FLT0397		
["u'T10", " u'T10a", " u'T15"]	FLT0295							
["u'F17"]	FLT0303							
["u'N02a"]	FLT0245	FLT0363	FLT0258	FLT0381				
["u'S01", " u'S31"]	FLT0275							
["u'T10"]	FLT0319							
["u'T09", " u'T09b", " u'T14"]	FLT0290	FLT0266	FLT0219					
["u'T09"]	FLT0234	FLT0310	FLT0226	FLT0340	FLT0309	FLT0279	FLT0304	
["u'T24"]	FLT0360	FLT0347	FLT0316					
["u'F03a", " u'F11a"]	FLT0412	FLT0417	FLT0228					
["u'T17"]	FLT0202	FLT0343	FLT0314					
["u'F01a", " u'F02a"]	FLT0322							
["u'F15"]	FLT0311							
["u'F03"]	FLT0351	FLT0415	FLT0218					

1.3 本論文の構成

本論文の構成は以下の通りである。第 2 章では、関連研究や本手法提案のための予備知識について述べる。第 3 章では、本研究の手法の提案として、CARATS Open Data から得られる航空機の経路を、安定区間と非安定区間に分け、安定区間内の処理時間の短縮を求める手法を提案する。第 4 章では、前章で述べた手法の実装及び考察を行う。第 5 章では、おわりに本研究のまとめと今後の課題について述べる。

第2章 準備

2.1 関連研究

航空機の混雑予測はレーダー管制官と調整管制官が制御している。現在の混雑予測、制御はセクターごとに、容量が設定されている。今日のセクター容量は、MAP: Monitor/Alert Parameter (モニター警報パラメータ) によって変動する。MAP 値は、天候、航空軌道の複雑さ、高い高度を飛ぶ航空機の数等、様々な要因で決定される。MAP 値をもとに、セクターの混雑度を予測する。例えば安全な順に、緑、黄、赤と表示させ、管制が監視しやすくする。セクター容量を高速に把握し、適切な管制を行うために提案されているセクター制御モデルを紹介する。

Jim らは、空域の混雑度を予測し、セクターに入る全体の遅延を、優先順位をつけることにより最小化する手法を、ヒューリスティックからのアプローチや、遺伝的アルゴリズムを応用したモデルを述べている [4]。

Banavar らは、過去の軌道データを、季節ごとにセクターフローモデルを作り、その複数個のモデルと実際の軌道データとを参照し、状況に適したモデルに改良するという提案をしている [5]。

セクターの予測モデルは、地域性を考慮することも重要である。日本は、先進国として、航空機の需要が高く、全国各地に空港が点在し、航空機の振る舞う軌道は複雑である。また、空港においても、広い面積をとれない等の理由から、機数の需要を満足できていない場合もある。平田は、羽田空港の4つの滑走路を、離陸と着陸それぞれの需要の変化に応じて、配分を変化することで空港の離着陸容量の最適化を提案している [6]。

日本のセクター予測モデルは、軌道を描く実データが公開されて日が浅いことから、日本の航空需要に適したモデルは開発途上である。本手法では、日本の航空需要に適したセクター予測モデル構築作成を目指し、その第一段階として、CARATS Open Data を用いて、セクター情報を決定する際の処理時間の短縮を提案する。

2.2 今日の航空機について

この節では、現在の航空機の特性について述べる[7]。

2.2.1 航空機の巡航時の振る舞い

空港から離陸した航空機は、上昇し、角度を目的地に合わせた後に、高度は一定に保たれ、等速直線運動を行う。力の釣り合いとしては、上下では、揚力と重力、前後では、推力と抗力が釣り合っている状態である。また、航空機のある一定の燃料に対して最大の飛行距離を実現させたい場合、航続率を最大にする必要がある。航続率を求める式を(2.1)に示す。

$$\text{航続率} = \frac{\text{航空機の世界速度}}{\text{エンジンへの燃料流量}} \quad (2.1)$$

航空機の世界速度に関しては、あまり変化させることはできない。ここで、エンジンへの燃料流量に着目してみる。航空機は、飛行中絶えず燃料を消費し、その消費した分だけ機体の重量が軽くなる。そのため、はじめは揚力と重力が釣り合っている状態でも、次第に機体の重量が軽くなり、不釣り合いになって行く。これを解決するためには、速度は一定という条件下では、機体を取り巻く風圧を下げる必要がある。地球の大気の性質上、高度が上昇すればするほどその密度は小さくなり、その分風圧も下げることができる。そのため、安定し、一定の世界速度を保って運航している航空機でも、徐々に高度を上げてゆくことで上下の力の釣り合いを保っているということである。ただし、全ての航空機が自由に高度を上昇させると安全面が問題となるため、実際は高度を段階的に上昇させている（ステップ巡航）。この上昇の上限は、大気の温度が下がらなくなる成層圏（地上約 33000 フィート）である。

2.2.2 自動飛行システム

自動飛行システムは、操縦士の負担軽減や安全性を求め、設定された道のを自動で操縦するシステムで、3つのシステム（自動操縦装置、自動推力調整装置、自動離陸装置）がある。自動操縦装置は、現在の飛行状態を把握し、航空機の左右の揺れを抑え、設定した方向に飛行を保つ機能を持っている。自動推力調整装置は、操縦士や飛行管理計算機からの入力に従い、速度を制御することや、推力調整するために高度を変更する機能を持っている。自動離陸装置は、着陸までの滑走路への進入と接地の操作を自動に行う機能を持っている。このように航空機は半自動化が進み、さらに開発が進められているが、上空は天候の変化が激しいため、現在は操縦士や管制官との連携によって臨機応変に飛行ルートの変更が求められている。

2.2.3 航空路の管制

航空機には、航空路と呼ばれる、飛行経路定められている。様々な大きさや、安全性能をもつ航空機が飛行するため、それぞれの航空機が、安全のための高度や速度を必要とする。管制官は、全体の航空機の位置情報を確認しつつ、操縦士へ指示をする。ここで重要となるのは、航空機の位置情報、速度の正確さやその情報を更新する周期、そして更新までの遅れである。

日本の空域は、札幌、東京、福岡、那覇にある空港交通管制部がそれぞれの地域を管轄している。それぞれの管制部から、さらにセクターに分割され、それぞれのセクターには2, 3人の管制官が担当している。管制官は、担当するセクターを飛行する航空機に対して、操縦士と連絡を取り合い、飛行経路や高度に関する承認、指示、許可を行う。天候の悪化や急な乱気流を回避するための経路変更の指示も行う。天候悪化による経路変更について、課題の一つとなっているのが積乱雲である。積乱雲の中は上昇気流、下降気流が入り交じり、航空機がその中へ入ると多大な損傷を被る。また、局所的で短時間な場合が多く、予測が難しい。今日では、積乱雲を回避するために、万能ではないが、地上気象レーダーを用いて実況監視を行っている。

2.3 CARATS Open Data

CARATS : Collaboration Actions for Renovation of Air Traffic System (航空交通システムの変革に向けた協調的行動) とは、国土交通省が運営し、日本の経済、産業の成長を支えていく上で基盤となる社会インフラの中の航空サービスについて、量的な拡大と質的な向上を目指す機関である。産学官の協調を土台とし、運航者と管制機関、民間・自衛隊・米軍などの空域の共通利用者、地域社会、国際的航空交通など、航空に係る様々なステークホルダーとの協調を掲げている。

国際民間航空機関 (ICAO) の傘下には、日本の CARATS の他に、米国の NextGen、欧州の SESAR などがある。

今回用いる CARATS Open Data は、我が国の航空交通システムを発展させ、研究開発を促進していくことを目的として、航空管制データの一部を公開したものである。2012 年度の一定期間の航空機の位置・時刻情報等について、国土交通省航空局が保有するエンルート管制用レーダーのデータを基に、座標変換等の処理をしたものである。具体的なデータ期間は、以下に示す、2012 年度の各奇数月における 1 週間分 (合計 6 週間) のデータを採用している。

2012年 5月7日(月)から13日(日)
 7月9日(月)から15日(日)
 9月3日(月)から9日(日)
 11月5日(月)から11日(日)
 2013年 1月7日(月)から13日(日)
 3月4日(月)から10日(日)

また、データの形式は、CSV形式で、表 2.1 のように格納してある。

表 2.1 CARATS Open Data の格納例

時刻	便名	経度(度)	緯度(度)	高度(ft)	形式
8:00:00	FLT0001	42.413533	141.681313	6247	B763
8:00:00	FLT0002	33.877210	131.116394	11800	DH8D
8:00:01	FLT0003	42.275752	141.749411	20266	B773
8:00:02	FLT0004	35.508394	133.801179	7979	B738
...					
8:00:10	FLT0001	42.426521	141.656398	6417	B763

時刻は日本標準時(JST)とし、型式は国際民間航空機関(ICAO: International Civil Aviation Organization)の「DOC 8643 Aircraft Type Designators」の略号である。それぞれの航空機は約10秒間隔で格納されており、時間順に記載されている。

データの対象となる航空機は、計器飛行方式による定期便のみとなる。(有視界飛行方式による飛行、エンルート空域を通らない飛行、軍用機・自家用機などは含まれない。)

2.4 日本上空のセクター

日本上空のセクター情報は、AIS JAPAN: Japan Aeronautical Information Service Center(航空情報センター)が提供しているデータを用いた[8]。AIS JAPANは、国土交通省航空局が運用し、成田空港に存在する。基本的に航空関係施設、業務、方式と危険等に係わる設定や状態、変更等についての航空情報(NOTAM: Notice To Airmen)を管理、発行している。日本上空のセクター情報を図 2.1 に示す。セクターの含有関係については、それぞれを別セクターとして考える。

ただし、上図並びに実験では正距円筒図法を用いた。その理由については4.4節にて述べる。

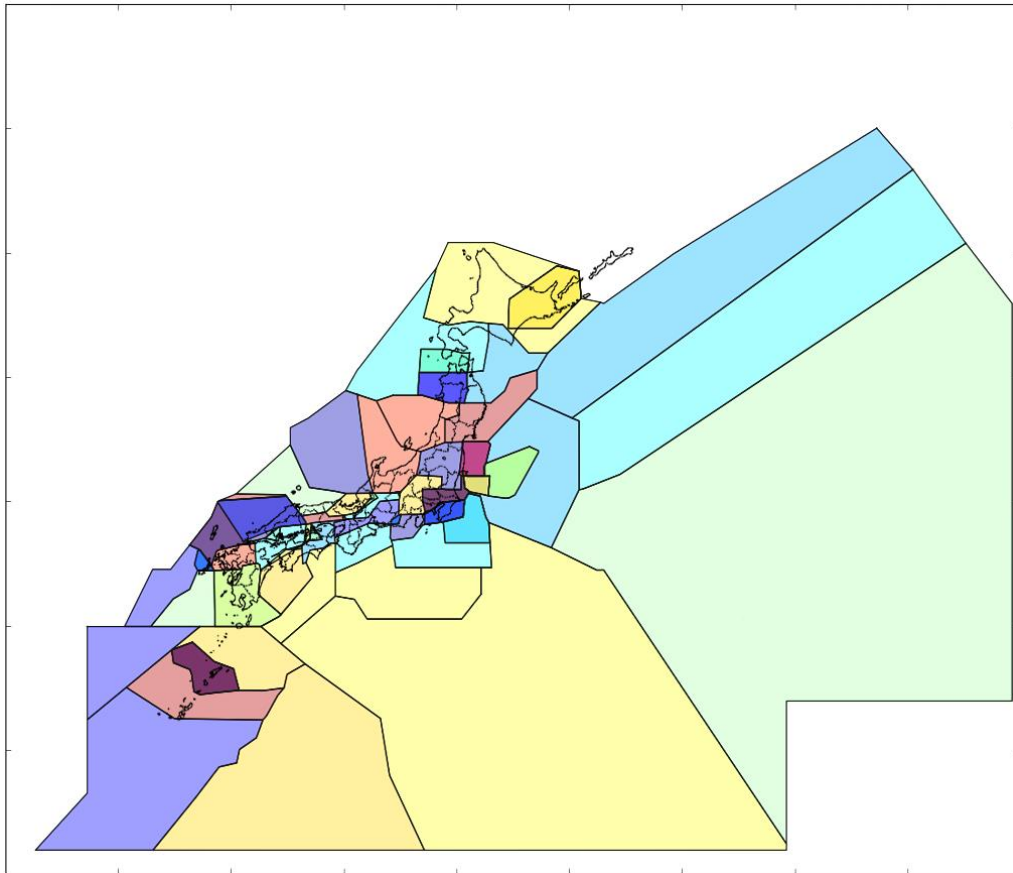


図 2.1 日本上空のセクター情報

2.5 CGAL ライブラリ —bounded_side_2—

本手法で、任意の点の所在地（セクター）の情報を与えるライブラリとして、CGAL (Computational Geometry Algorithms Library) の関数 `bounded_side_2` を用いる。[9]CGAL は、1996 年に始動したソフトウェアで、計算幾何学分野のアルゴリズムに効果的なアクセスを提供する。ヨーロッパやイスラエルが開発拠点で、多くの論文、学術会議、ジャーナルで利用されており、信頼性は高いといえる。

bounded_side_2 のアルゴリズム

任意の点 A が多角形 X の内側にあるかどうかを計算する。点 A からあらゆる方向に無限に広がる線を引き、与えたすべての多角形との交点をカウントする。その線と多角形 X との交点が、奇数の場合、点 A は多角形 X の内側に存在する。そうでない場合、外側に存在する。多角形の辺の上に存在する場合にも対応している。点 A から描く線と多角形 X との交点の数について、図 2.2、2.3 に示す。

セクターは凹な多角形も存在するため、あらゆる二次元の多角形に対応するこの関数を用いた。実行時間は、与えた多角形の頂点数に対し線形である。

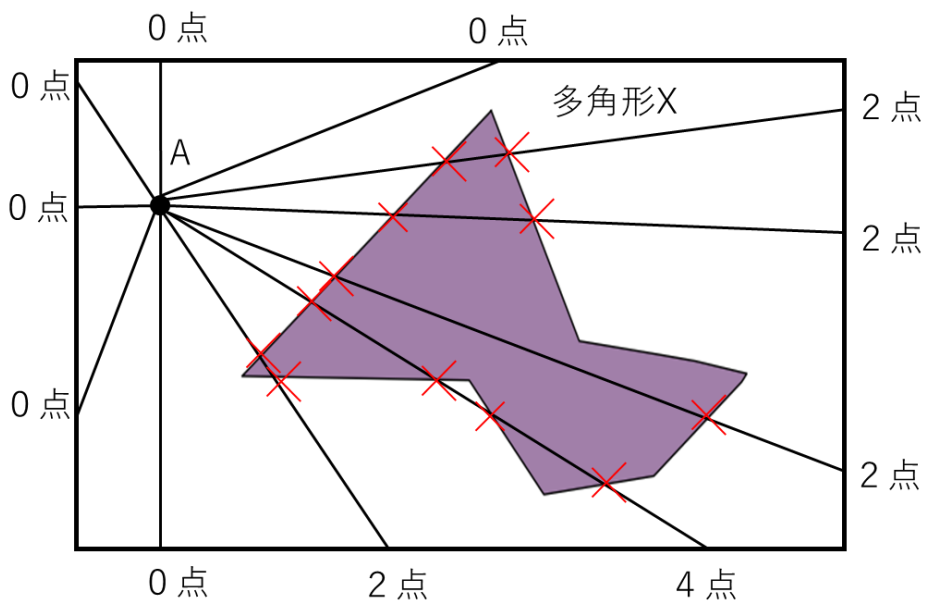


図 2.2 点 A が多角形 X の外側時の交点数

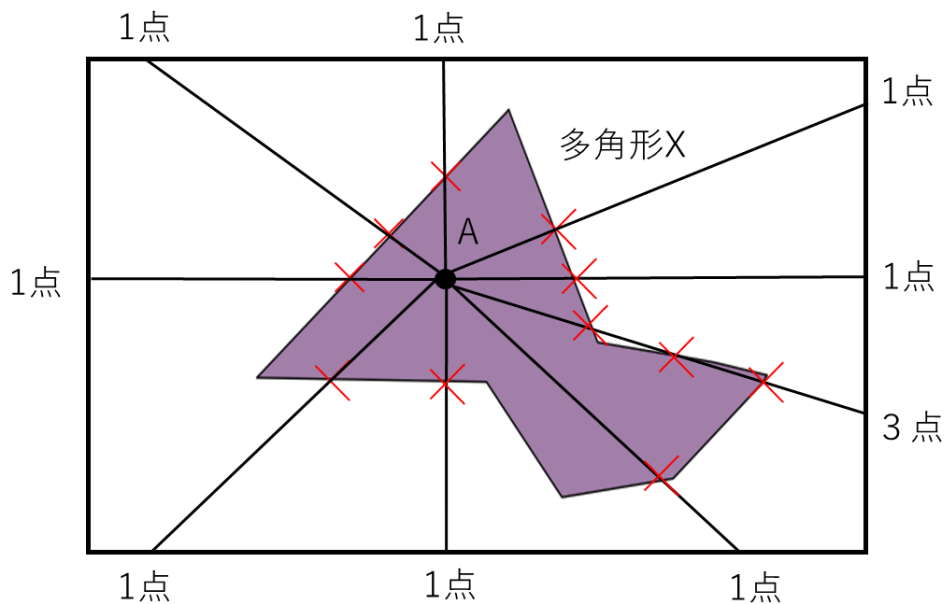


図 2.3 点 A が多角形 X の内側時の交点数

2.6 全点探索による航空機のセクター情報の付与

本手法の高速化の程度を計測するために、CARATS Open Data に格納されているデータを全点探索する。約 10 秒間隔で記載されている航空機のデータを一つ一つ時間順に、bounded_side2 によりセクター情報を付与し、セクターの変化した点（交点）を求める。その処理時間を計測し、単純なセクター情報決定プログラムを基に、本手法との比較対象とする。実験環境等の条件は、4.2 節で述べる本手法の環境と同等とする。

この処理時間は 10594.1909921[s]であった。

第3章 提案手法

3.1 概要

それぞれの航空機が任意の時間にどのセクターに存在するか、実際の飛行情報である CARATS Open Data を用いて付与する。航空機は離着陸の際、安全の為に旋回をし、目的地への軌道を調整することがある。逆に、その他の区間は高度や方向の転換が少なく、大まかな予測が簡易である。本手法では、離着陸付近の状態を非安定区間、その他の安定した飛行がされていると考えられる区間を安定区間と定義し、安定区間内のセクター情報を決定する際の処理時間を短縮するための手法を提案する。

3.2 パラメータ設定と安定区間

本節では、航空機の軌道を安定区間と非安定区間に分けるためのパラメータと、その流れについて述べる。

3.2.1 安定区間を決定するパラメータ

安定区間と非安定区間を分別するために、いくつかのパラメータを用意して選別する。航空機が安定している場合の条件はいくつか存在するが、今回は処理時間をできるだけ短縮する為に、特に重要であると考えられる3つのパラメータを用いる。また、CARATS Open Data には、任意の航空機の情報は、時刻、便名、緯度、経度、高度、型式のみの記載となっている。今回はその中の高度を安定区間の決定材料として利用する。

最低限の高度 (Min_height)

航空機が安定飛行をするためには、ある程度の高さが必要となる。安定した飛行をする為の高度は、33000 フィート付近が多いが、これは、エンジン効率を求めつつ、航空機が重力と揚力を保つ最大の高度である。航空機の性能やその時の気候、飛行距離によって変化する。このパラメータは、あらゆる航空機の最低限の高度を設定することで、安定区間とは考えにくい低い高度の点を除外する機能を持っている。Min_height によって決定される安定区間の候補の構図を図 3.1 に示す

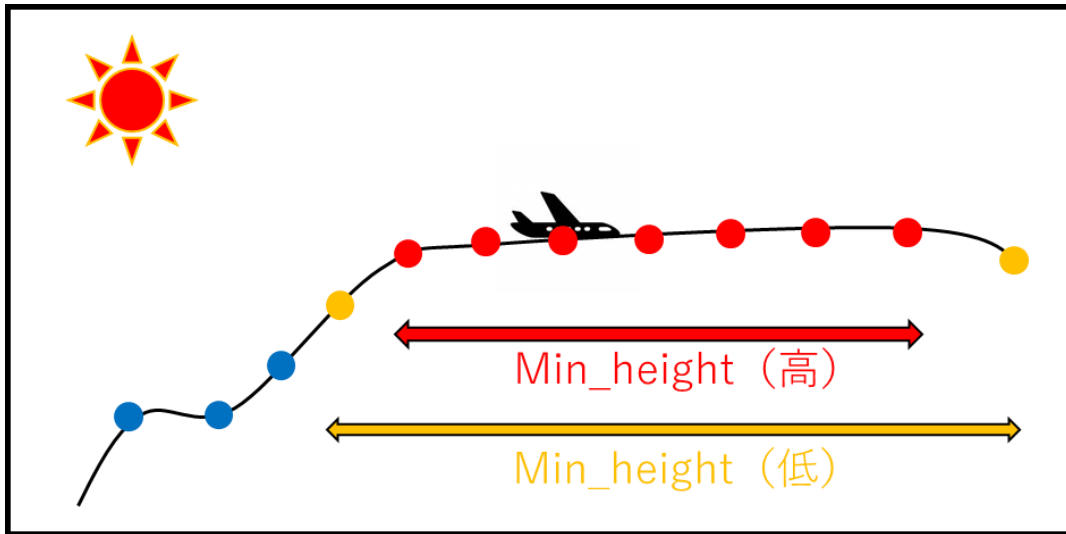


図 3.1 Min_height の構図

高度の変化量 (Height_range)

安定区間に達したと考えられる航空機の高度は、変化が小さいことが考えられる。このパラメータは、航空機の高度の変化量の基準を設定し、変化量の大きい高度の点を除外する機能を持っている。Height_range によって決定される安定区間の候補の構図を図 3.2 に示す。

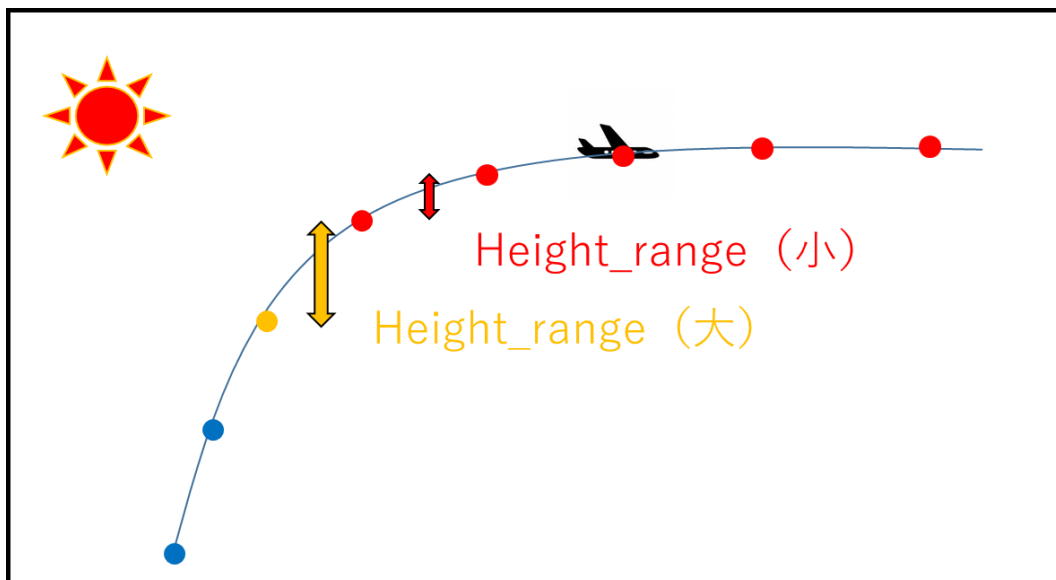


図 3.2 Height_range の構図

比較対象の点の幅 (Safety_step)

Height_range は基準となる点の他に、それと比較対象となる点が必要となる。今回の手法では、Safety_step のパラメータを設け、基準点から Safety_step と 2 Safety_step だけ離れた点を比較対象とする。Safety_step によって決定される安定区間の候補の構図を図 3.3 に示す。

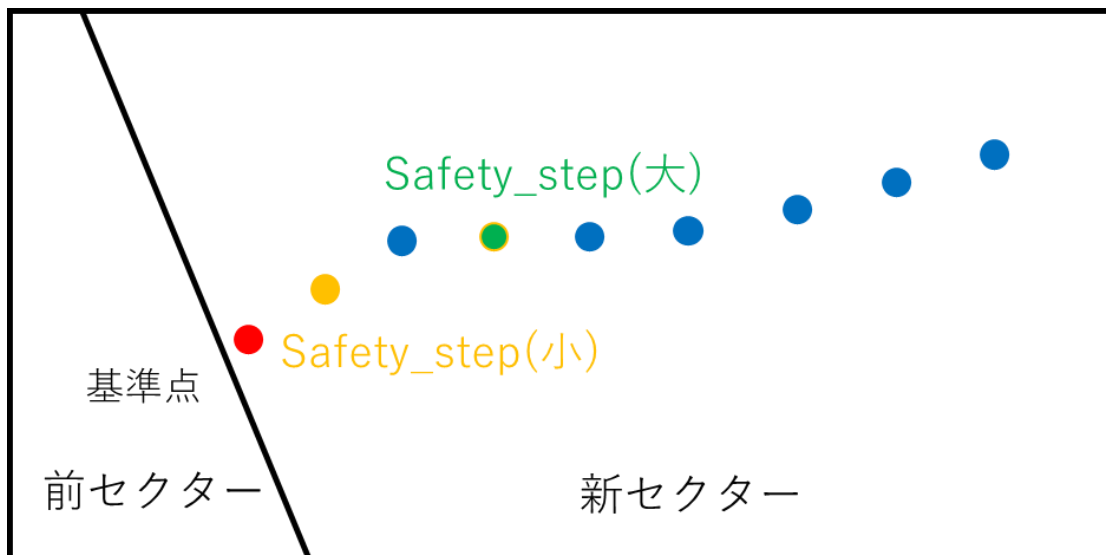


図 3.3 Safety_step の構図

3.2.2 安定区間決定の流れ

航空機データが上述のパラメータにより分別されるフロー図を図 3.4 に示す。実時間順に並べられた航空機データを、昇順に処理を行う。Min_height を満たす点はさらに Safety_step によって得られる比較対象と Height_range を満たすことができればその点を安定区間の始点とする。パラメータ条件を満たさなかった場合はさらに次の点を基準点として処理を行う。同様に、降順でも条件を満たす点を検索し、終点となる点を決定する。安定区間に達しても高度は一定量変化することから、始点と終点の間に存在する点を安定区間、それ以外を非安定区間とする。

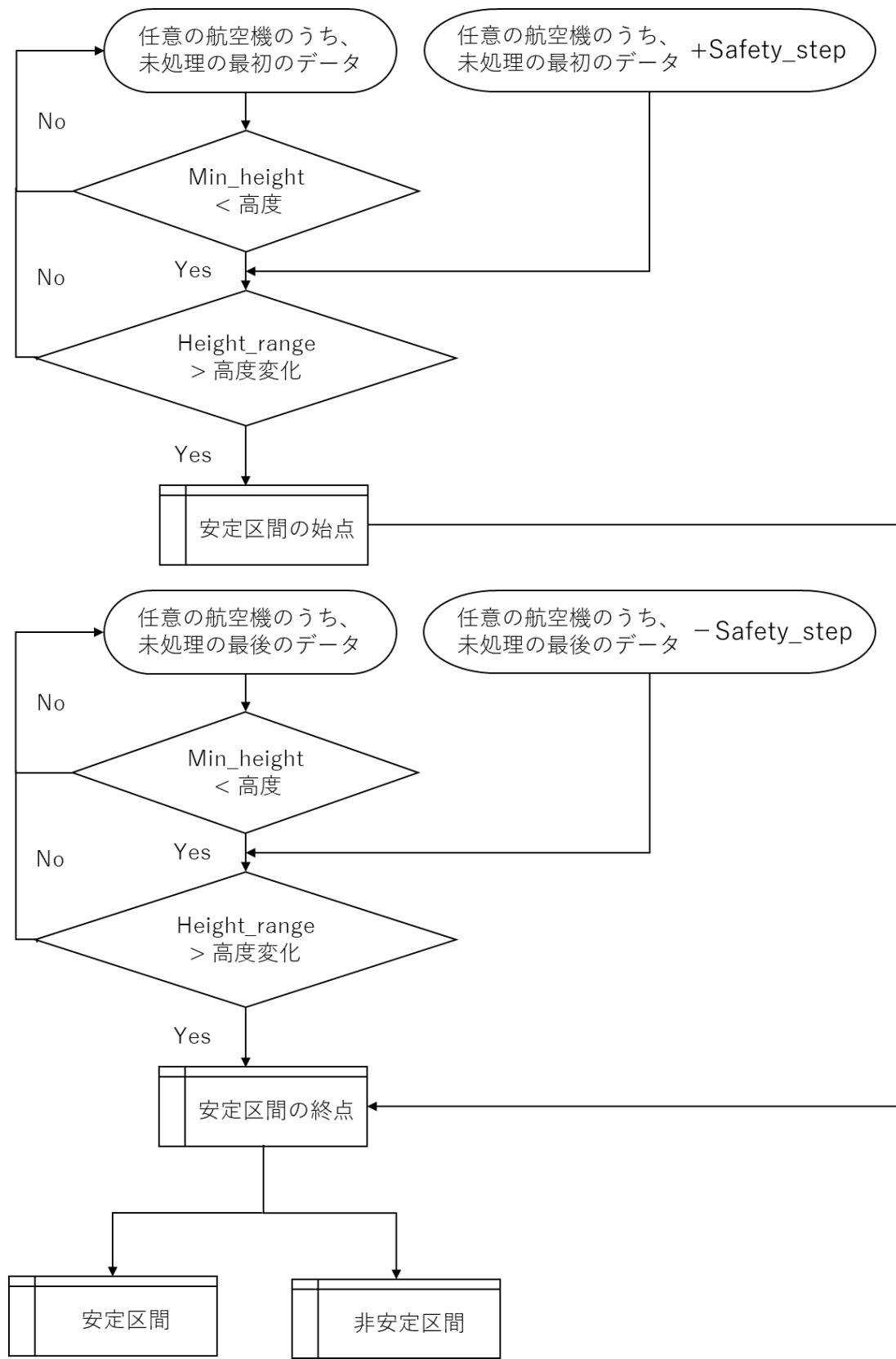


図 3.4 安定区間決定のフロー図

3.3 安定区間のセクター情報決定

この節では、安定区間内のセクター情報を決定する際に、処理時間を短縮するための手法を記す。3.2節で安定区間を取り出し、できるだけ参照するデータ数を減らし、処理時間の短縮を目的とする。

3.3.1 航空機の物理特性を利用した大まかな軌道の予測

安定区間内の航空機の軌道は、直線的な軌道をとるため、飛行角度や速度はあまり変化しないことが考えられる。これを利用し、基準点から直線を形成し、現在のセクターの辺と交差する予測時間を検索する。直線を形成するための点は、基準点とパラメータ Step を用いて、基準点+Step の2点とする。基準点と基準点+Step の二点から直線を描き、予測時間を計測するモデルを図 3.5 に示す

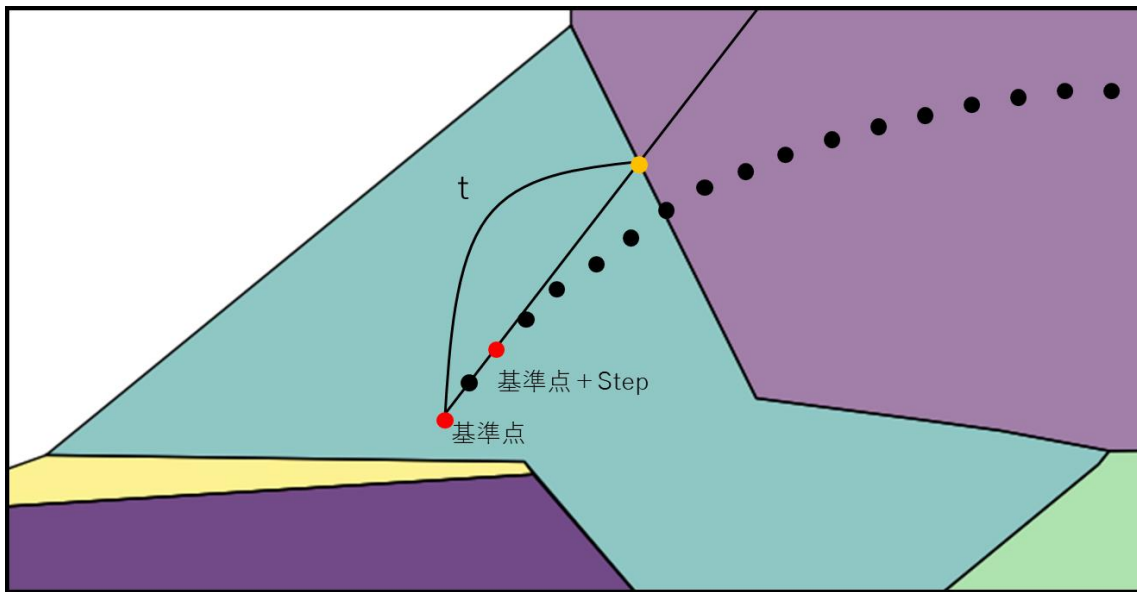


図 3.5 基準点と基準点+Step の二点から予測時間を計測するモデル

3.3.2 安定区間のセクター情報決定の流れ

安定区間内のセクター情報決定のフロー図を図 3.6 に示す。直線を描く際の基準点となる点は、最初は安定区間の始点とする。以下にフロー図の詳細を述べる。

1. 安定区間の始点を基準点として、セクター情報を与える。
2. 基準点と、基準点+Step を通る直線を引く
3. 2の直線と、セクターの辺との交点を求め、それを予測交点とする。
4. 基準点と基準点+Step から速度を求め、予測点まで等速直線運動をしたと仮定したときの予測交点の時間を求め、それと近い実データの点を求める。
5. 4の実データの点のセクター情報を求め、基準点と同じセクターであるかを判定する。
6. 基準点と4の点と同じ場合、実データのセクターの境界点が大きく遠くないことが考えられるため、セクターが更新されるまで昇順に実データの点を探索する。
7. 基準点と4の点が違う場合、実データのセクターの境界点が予測交点と離れている可能性も考えられる為、基準点と4の点の中を、セクターの境界点を検出するまで二分探索する。
8. 求められたセクター境界点を新しい基準点として、2に戻る。

また、この手法は、直線を引くことでセクターの予測交点を求めるが、実際のセクター境界点と誤差があったとしても、5から7の手順のように、予測交点から実際の境界点を求めるため、最終的には実際のセクターの境界点が求められる。したがって、得られる境界点の結果は正しいことは明らかである。

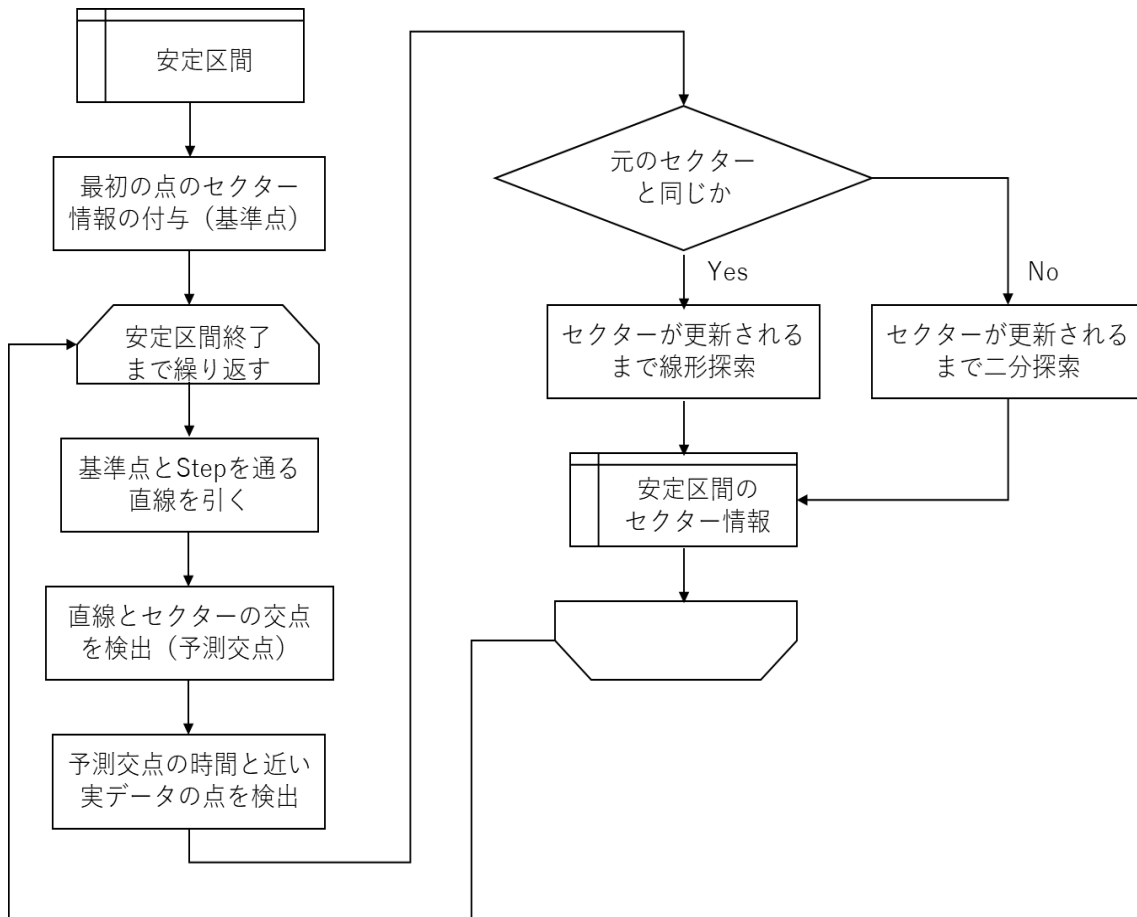


図 3.6 安定区間内のセクター情報決定のフロー図

第4章 評価実験

4.1 実験内容と評価基準

本研究では、CARATS Open Data の 12 時間分を用いて、3 章で示した提案手法を実装する。2.4 節の全点探索によって得られるセクター情報の実装時間と比較し、どの程度処理時間が短縮されたかを本手法の評価基準とする。

また、本手法に用いるセクターとの予測交点と、実際の予測交点との誤差率をカウント誤差率、距離誤差率で評価する。カウント誤差率、距離誤差率の定義をそれぞれ(4.1)、(4.2)に示す。

$$\text{カウント誤差率} = \left| \frac{\text{新しい基準点の番号} - \text{予測交点と時間的に最も近い点の番号}}{\text{新しい基準点の番号} - 1 \text{ つ前の基準点の番号}} \right| \times 100[\%] \quad (4.1)$$

$$\text{距離誤差率} = \left| \frac{\text{新しい基準点と予測交点の距離}}{\text{新しい基準点と 1 つ前の基準点の距離}} \right| \times 100[\%] \quad (4.2)$$

また、距離誤差率を求める際の分母、分子それぞれの距離に関して、CARATS Open Data は位置情報として経度と緯度を提供している。経度と緯度の距離を求める式を(4.3)に示す。

地球の赤道半径 $r = 6378.127[\text{km}]$ 、地点 A(経度 x_1 , 緯度 y_1)、地点 B(経度 x_2 , 緯度 y_2) とすると、二点間の距離 d は、

$$d = r \cos^{-1}(\sin y_1 \sin y_2 + \cos y_1 \cos y_2 \cos \Delta x) \quad [10](4.3)$$

4.2 実験環境

本システムの開発、及び実験環境については、Microsoft 社の Surface Pro 3 上の、仮想コンピュータ VMware を用いた。Surface Pro 3 のメモリは 4 G で、CPU は Intel (R) Core(TM) i5-4300U 1.9GHz が搭載されている。VMware のメモリは 1.5G で、プロセッサのコア数は 4 個である。プログラムの作成には、Atom editor を使用し、言語は python2.7 で行った。

4.3 実験結果と考察

3章で述べた提案手法を、パラメータの値を変更し、その振る舞いを考察する。また、その処理時間の結果が良いパラメータ値に固定し、新たにパラメータを設定し、安定区間の条件を厳しくしたときの処理時間を計測し、考察する。さらに、4.1節で述べたように、本手法の安定区間の予測点の誤差率についても評価、考察する。

4.3.1 評価実験

3.2節で述べたように、本実験では、安定区間を検出するためのパラメータを用いる。パラメータに設定した数値を以下に示す。パラメータ値に関しては、CARATS Open Dataのデータを参照し、航空機の特性に適していると思われる値を設定した。

Safety_step = [1,2,3]

Hight_range = [200,400,600][ft]

Min_height = [5000,7500,10000][ft]

また、3.3.1で述べた、安定区間内のセクター予測交点を求める際に用いるパラメータに設定した数値を以下に示す。

Step = [1,2,3]

パラメータ設定したセクター決定の処理時間を Step と Safety_step ごとに示したものを図 4.1 から図 4.9、に示す。また、この結果から得られた最適なパラメータ値と推定される処理時間の上位を表 4.1 に示す。

なお、図中のパラメータは、Step : st、Hight_range : hr、Min_height : mh、Safety_step : ss と表している。

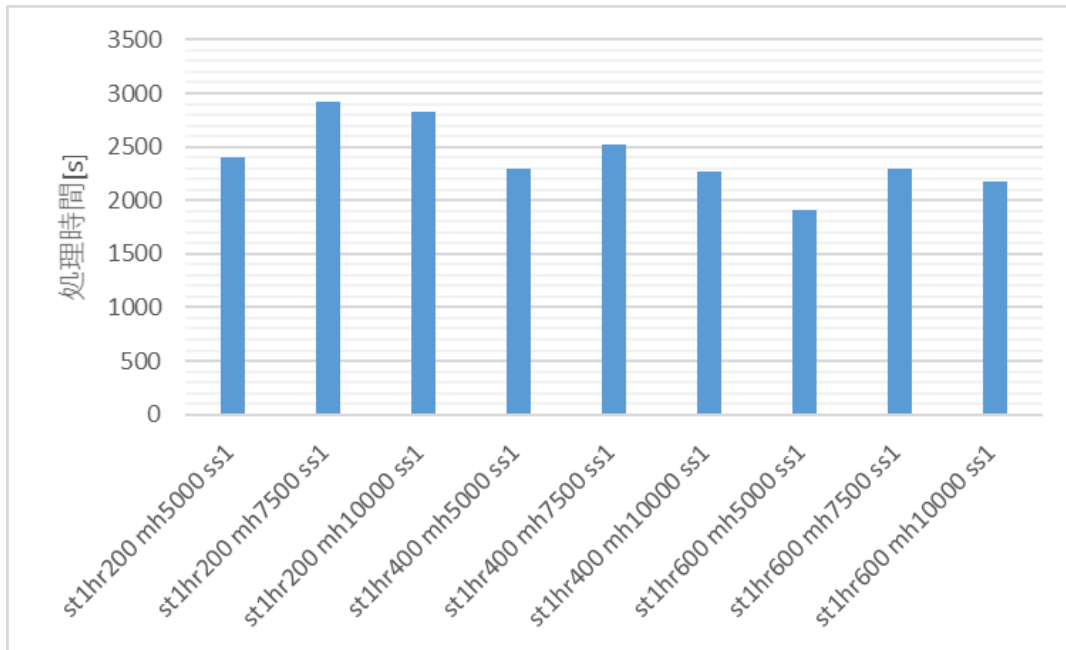


図 4.1 Step = 1、Safety_step=1 とした処理時間

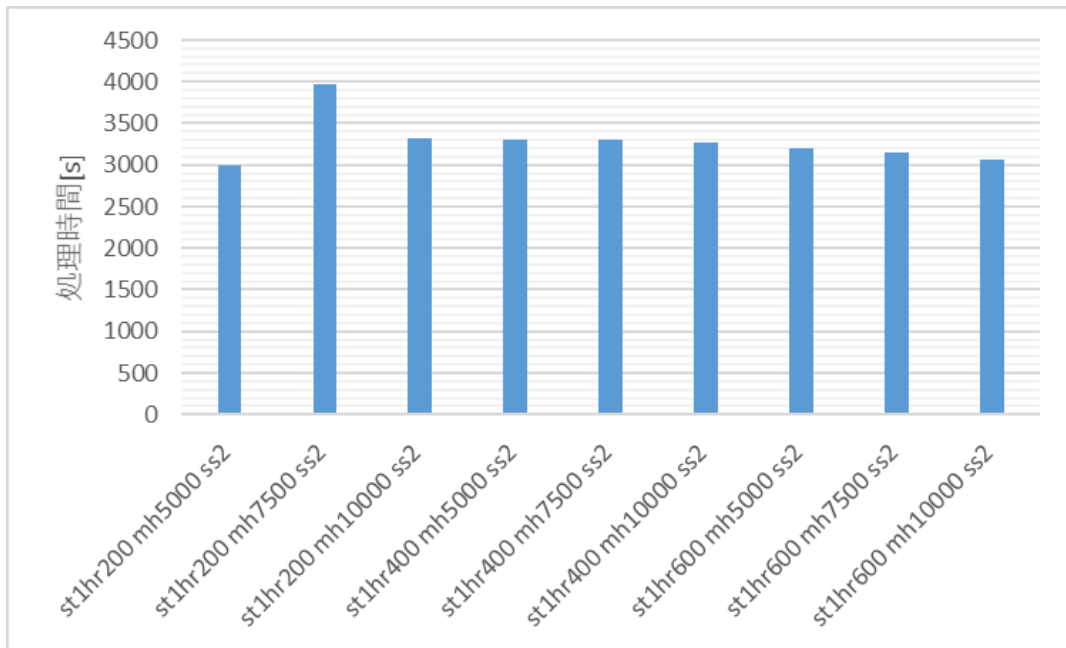


図 4.2 Step = 1、Safety_step=2 とした処理時間

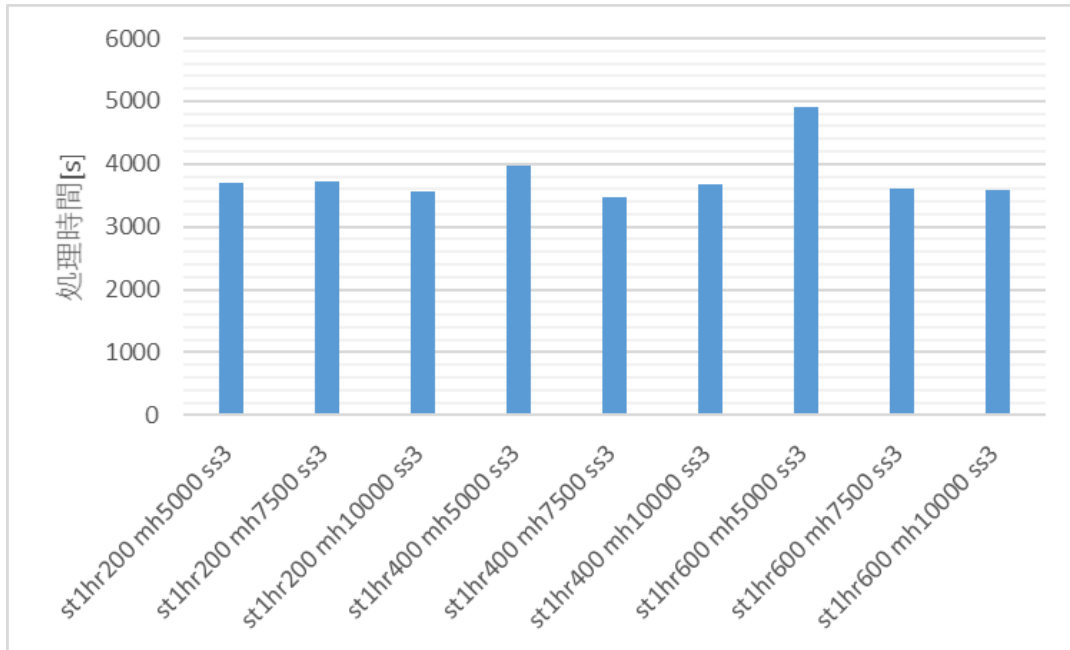


図 4.3 Step = 1、Safety_step= 3 とした処理時間

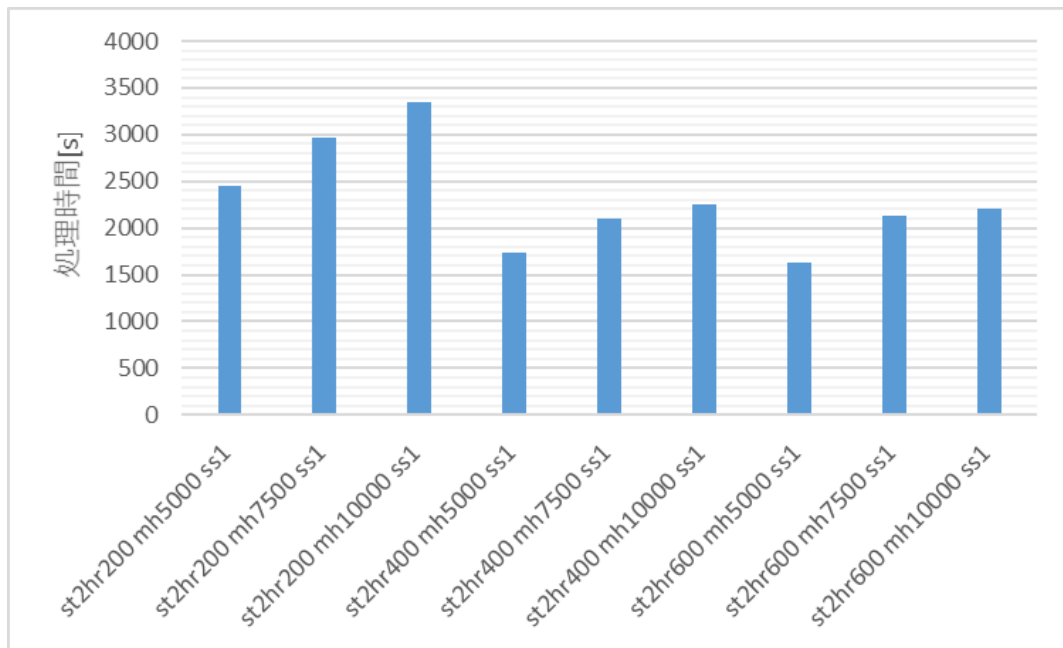


図 4.4 Step = 2、Safety_step=1 とした処理時間

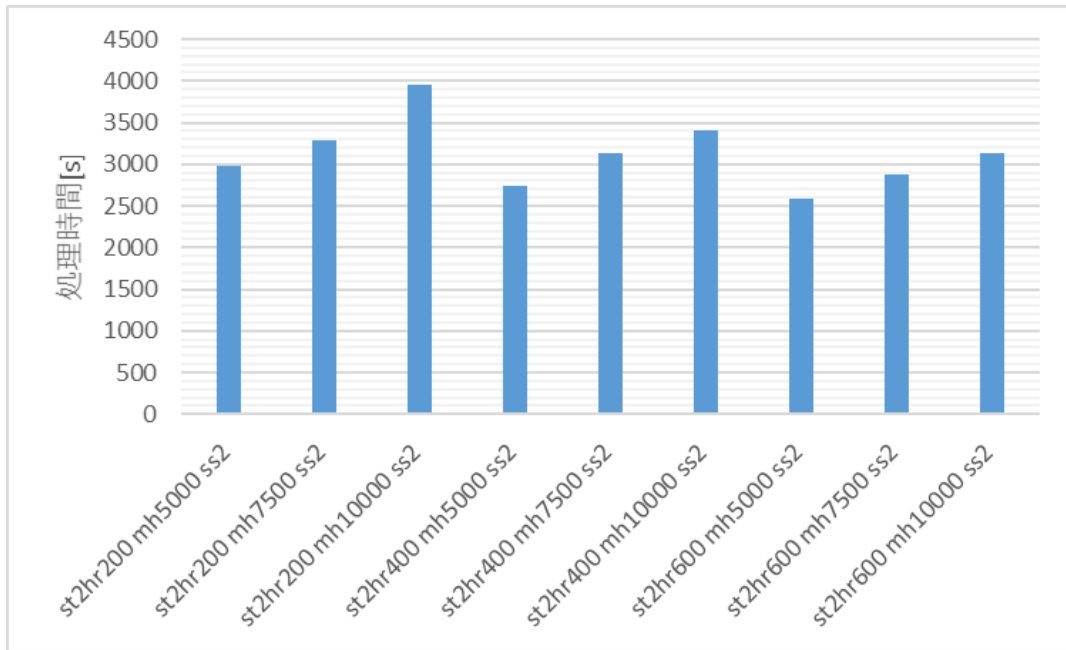


図 4.5 Step = 2、Safety_step= 2 とした処理時間

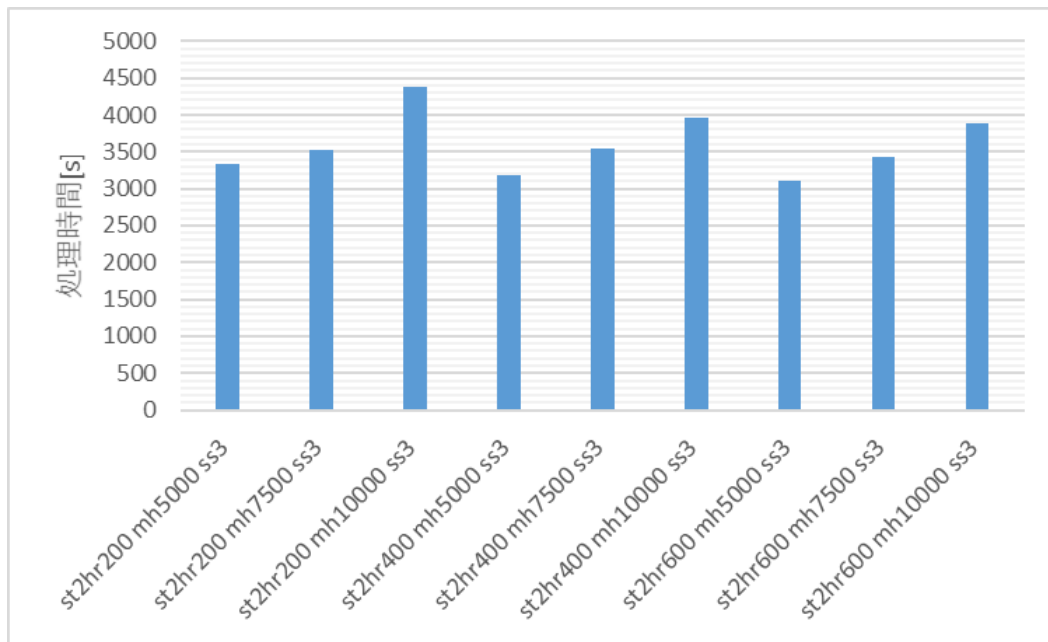


図 4.6 Step = 2、Safety_step= 3 とした処理時間

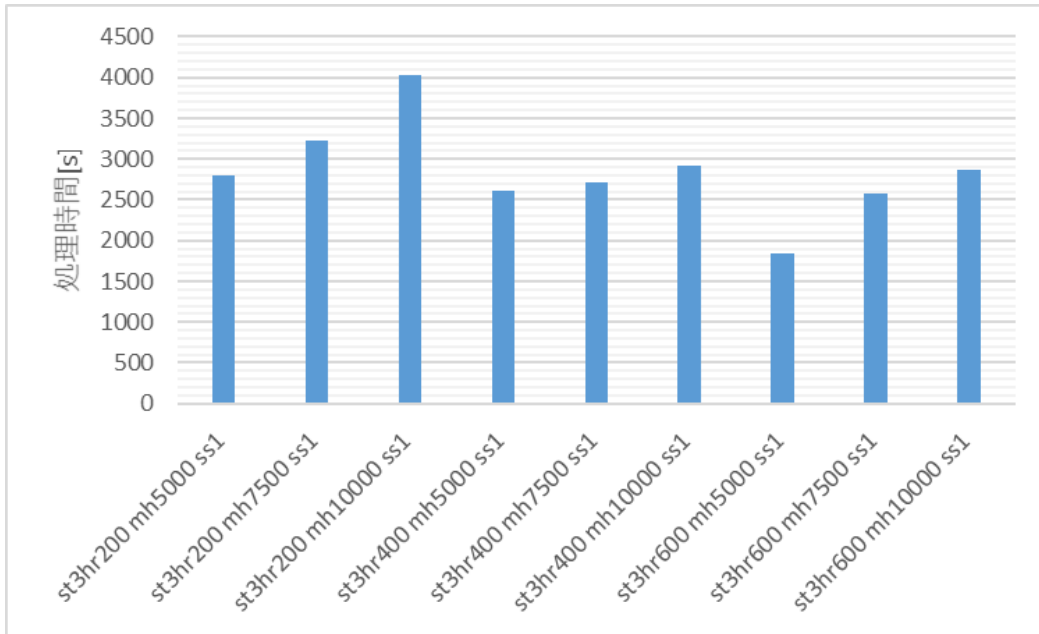


図 4.7 Step = 3、Safety_step=1 とした処理時間

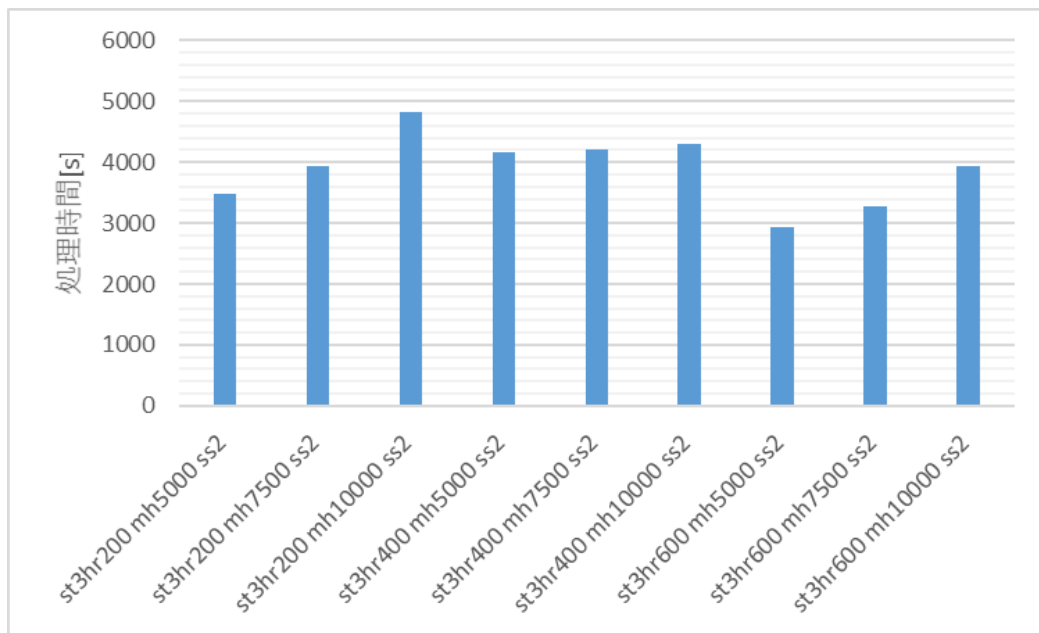


図 4.8 Step = 3、Safety_step=2 とした処理時間

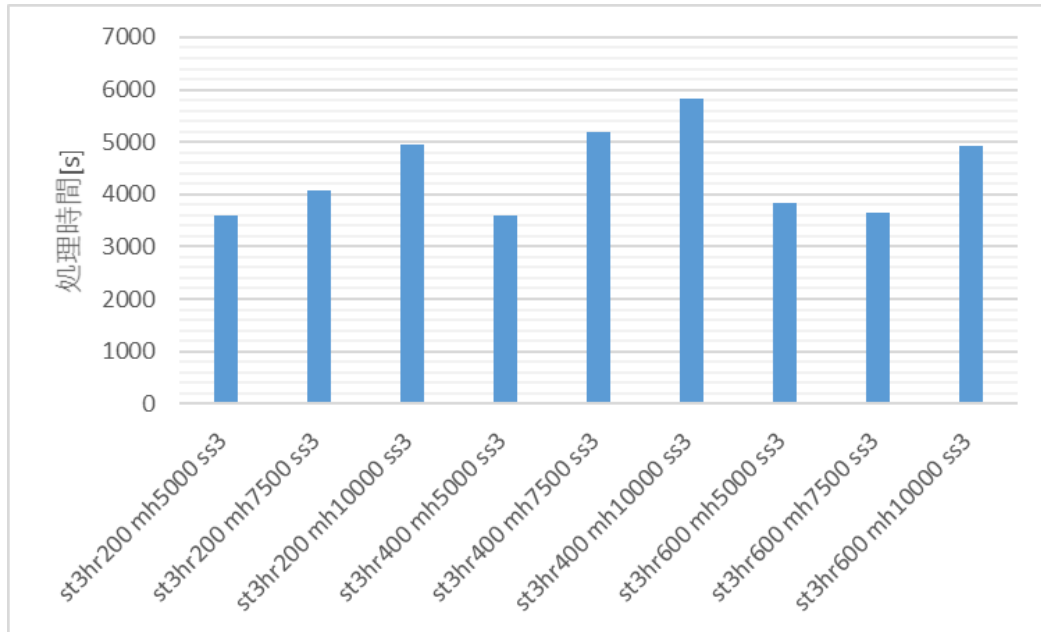


図 4.9 Step = 3、Safety_step= 3 とした処理時間

表 4.1 処理時間の短い上位 10 組

順位	パラメータ	処理時間[s]
1	st2hr600 mh5000 ss1	1622.493470
2	st2hr400 mh5000 ss1	1730.632583
3	st3hr600 mh5000 ss1	1843.502328
4	st1hr600 mh5000 ss1	1906.184860
5	st2hr400 mh7500 ss1	2094.287663
6	st2hr600 mh7500 ss1	2127.614539
7	st1hr600 mh10000 ss1	2177.191824
8	st2hr600 mh10000 ss1	2211.482621
9	st2hr400 mh10000 ss1	2248.589505
10	st1hr400 mh10000 ss1	2269.663453

処理時間は、全点探索で実装した 10594.1909921[s]から大幅に短縮することができた。パラメータ値は、Safety_step=1 で良い結果となる傾向があることが分かった。また、Min_height は、値が小さいほうがより処理時間が短くなる。Step や Hight_range については、値を変化させることによって処理時間が増減するという傾向はみられなかった。

4.3.2 パラメータ増加に伴う処理時間の増減

本手法ではいくつかのパラメータを用いて安定区間を定めたが、CARATS Open Data から得られる情報をもとに安定区間を決定するためには、様々なパラメータを用いることで安定区間内の処理を高速にできる可能性がある。反対に、非安定区間の処理が増加し、処理が遅くなる可能性もある。この節では、3つのパラメータの他に新しいパラメータ vec を設定し、処理時間の増減を考察する。

航空機の方向 (vec)

安定区間は、直線に近い軌道をとることを想定し、処理を行った。このパラメータは、軌道の曲がり具合を判定する機能を持つ。

基準点と、パラメータ $Safety_step$ によって得られる2点のそれぞれとの単位ベクトルの内積を求め、 Vec と比較することで判定する。 Vec によって決定される安定区間の候補の構図を図 4.10 に示す

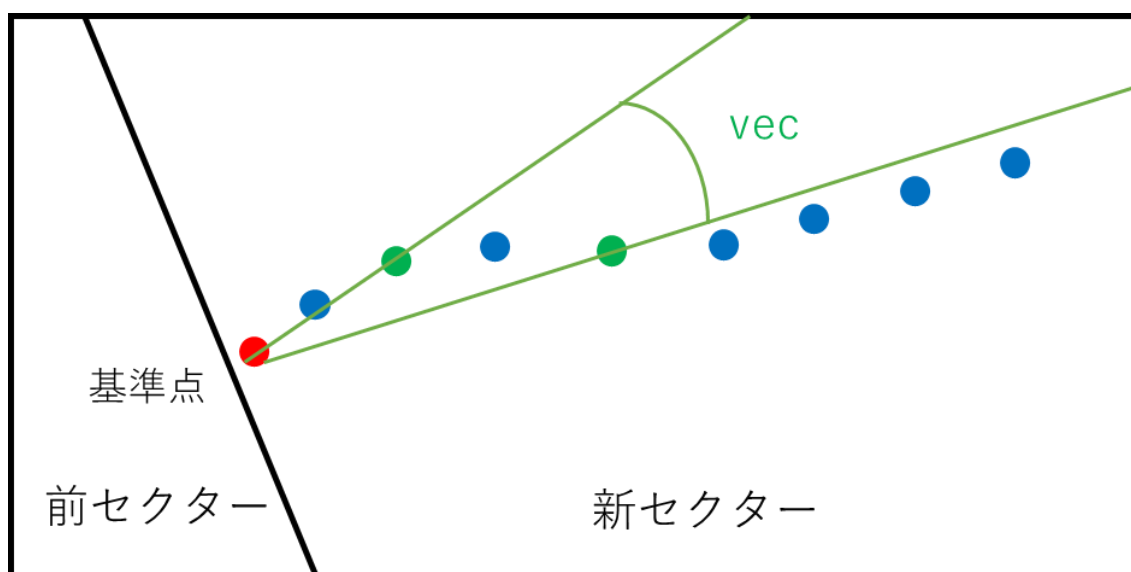


図 4.10 Vec の構図

また、パラメータ Vec の値は、以下のように設定する。

$$Vec = [0.8, 0.9, 0.95]$$

パラメータ Vec を用いた処理時間と、4.3.1 の実験結果から得られた処理時間の比較を表す表を、表 4.2 に示す。なお、その他のパラメータ値は、4.3.1 で最も処理時間の短い結果となった、 $Step=2$ 、 $Height_range=600$ 、 $Min_height=5000$ 、 $Safety_step=1$ と固定した。

表 4.2 Vec を用いた処理時間

Vec	処理時間[s]
設定なし	1622.49347
0.80	1946.28145
0.90	2077.90237
0.95	2190.61145

Vec は、単位ベクトルの内積のため、Vec=1 が最高となり、安定区間となる条件が最も厳しくなる。Vec の条件を厳しくすると処理時間は増加するという結果となった。これは、安定区間が減少し、非安定区間が増加したため、安定区間のデータはより直線に近い軌道となることが考えられるが、非安定区間が全点のセクターを検索するため増加してしまったと考えられる。

4.3.3 予測交点との誤差率

(4.1)、(4.2)を用いて、安定区間内の予測直線の誤差を考察する。パラメータ値は、処理時間の短い結果となった、Step=2、Height_range=600、Min_height=5000、Safety_step=1 とする。データの分布を表現するために、箱ひげ図を用いる。図 4.11 にその結果を示す。なお、箱ひげ図の箱中の線が中央値、箱の上下がそれぞれ、第 3 四分位数、第 1 四分位数、外れ値は、第 1 四分位数から第 3 四分位数までの長さの 1.5 倍以上と設定している。

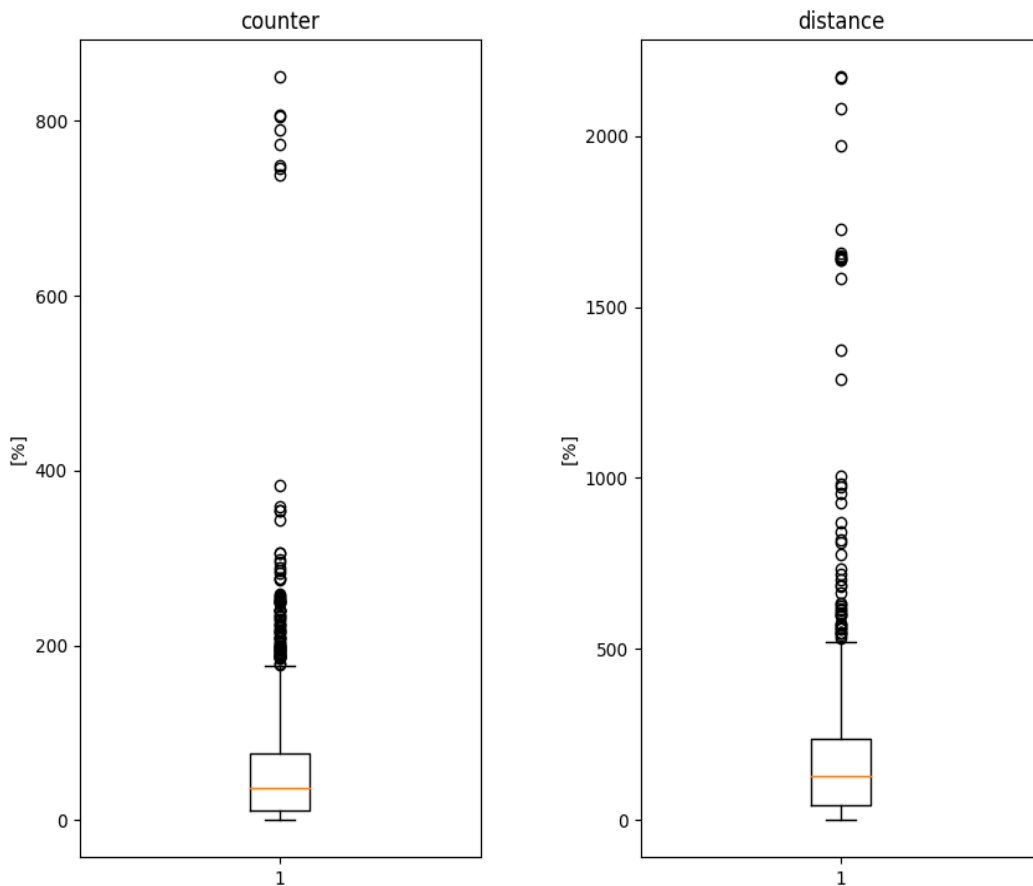


図 4.11 処理時間の短い実装時のカウント、距離誤差率

図 4.11 を見てわかることは、大きい誤差が数十存在する。これは、予測直線が大きく外れた位置を示していることを意味している。

本手法では、セクターの予測境界点を求める際に、参照している点が、基準点と基準点 + Step の二点となる。そのため、その二点が航空機の軌道性質から多少の軌道のずれを生じさせてしまっている場合、予測境界点とは別のセクターの辺が実際のセクター境界点となり得るため、大きく誤差になってしまうことがあると考える。

今回は、より簡素なアルゴリズムを用いることで複雑な計算を少なくしようと考えた。誤差率が大きくても、その誤差から実データの境界点を探索するため、境界点の結果は正確ではあり、処理時間は、全点探索と比べて十分短縮できたが、誤差率をより減少させることで、さらに処理時間は減少できると考える。

4.4 セクター探索時の直線を描く際に適した地図

本手法では、セクターをマッピングする地図に正距円筒図法を用いた。正距円筒図法の世界地図を図 4.15 に示す。

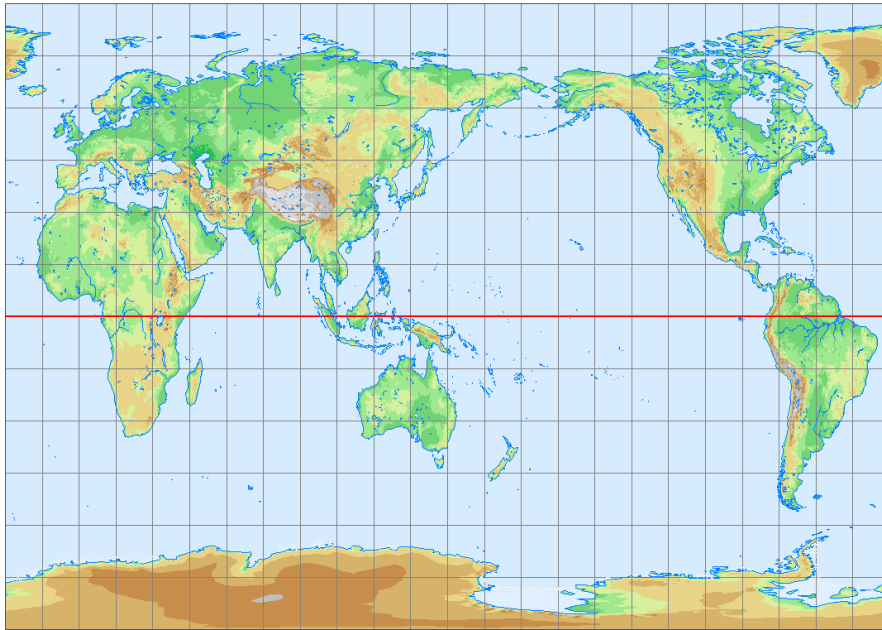


図 4.15 正距円筒図法によって描かれた世界地図[11]

この図法は、緯度・経度をそれぞれ地図の縦・横をそのまま読み替えたもので、標準経線上と縦方向に関して正距である。次のセクターの交点を求める際、この地図上で直線を描くのは、球形の地球では正確な直線とは言えない。正確に境界との直線を描く場合、正距方位図法が適している。正距方位図法の世界地図を図 4.16 に示す。正距方位図法は、中心からの距離と方位が正しく記され、地球全体が真円で表される投影法である。基準点を地図の中心とすることで、すべての方向に正確な直線を描くことができる。

今回正距円筒図法を用いた理由として、まず、本手法は正確な直線を必要としていないことである。飛行軌道を近似的に表すために直線を用いたが、そこから実際の境界点を探る。つまり、直線は境界点を直接的に求める訳ではなく、境界点の付近を求めるものである。正距方位図法で求めた方がより境界点の近くを求められ、調整の処理時間が短縮される可能性はあるが、処理時間問題については以下に示す。

次の理由として、処理時間問題を考察する。正距円筒図法では、セクターをあらかじめマッピングしておいて、その上で航空機の軌道を描くため、地図作成にかかる処理時間はオフラインとして考えられる。それに対し正距方位図法は、中心となる点が基準点となるため、処理ごとに地図作成を行わなければならない。その処理時間は、

航空機の数 × 各航空機のセクターを通る数 × セクターの数 × 各セクターの頂点の数
となり、 $O(n^4)$ となる。より高速に処理することを目的とする本手法では、正距方位図法を用いることは不適切と考えた。

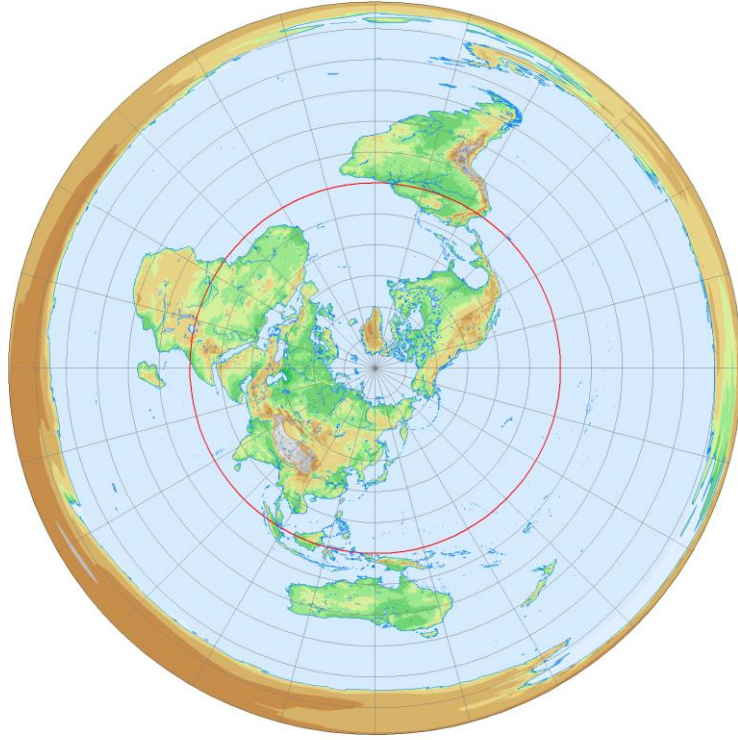


図 4.16 正距方位図法によって描かれた世界地図[11]

第5章 おわりに

本研究では、航空機軌道の実データを用いて、日本の航空サービスに適したセクター予測モデルの開発に向けて、その第一段階として、セクターの情報を付与するために、航空機の軌道の特性を利用して、処理時間短縮についての提案を試みた。航空機の軌道を高度情報や高度変化情報から、安定区間と非安定区間に分けて、安定区間内のセクター付与の処理時間短縮モデルを開発した。

安定区間を決定するパラメータや、セクターの境界点を予測するためのパラメータを用いてシミュレーションを行い、線形探索と比較し、十分な時間短縮を実現することができた。また、パラメータを適切に設定することで、よりよいセクター探索モデルになることを示した。

至らなかった点として、本研究では、Microsoft社の surface pro3 を用いたが、環境構築上、VMware で仮想コンピュータを起動しての実装となった。そのため、同じ環境下であれば、相対的な処理時間の比較は可能だが、絶対的な比較は困難である。また、私の技術上、プログラム言語は python2.7 を利用したが、これも高速化やリアルタイム処理を目指す本手法には C 言語など、より高速に処理が可能な言語もあると考える。

さらに、本手法は、大胆に直線を描いて、それを航空機の軌道の予測線としたが、これは、日本のセクターは狭く、アメリカなどの広大な土地に配置されているセクターに比べ、現在のセクターから次のセクターへの間隔が狭いため、直線での予測にもある程度正確性が保たれるであろうことと考え実装に至ったが、これを検証する実験までには至らなかった。

また、4.3.3 で述べたように、誤差率がとても大きい数値を出してしまったので、本手法をさらに改善することで、より処理時間の短縮ができるのではないかと考える。

今後の課題

将来的には、実際の航空機の軌道の予測をリアルタイムで行うことを考えている。そのため、今後の目標としては、シミュレーションを行う際、現在時刻と仮定した時刻より先のデータを用いない研究を行いたいと考えている。また、リアルタイム予測の際には、実際の天候情報やその後の天候の予報を加味した実装や、航空需要密度の高い日本の航空情報に適したモデルの実装も行いたいと考えている。また、航空機の軌道予測を行う際に、本手法の多少の軌道のずれが生じて誤差が生じないようなアルゴリズムの提案と実装を行いたい。

謝辞

本研究を行うにあたり、未熟な私に対し、手厚くご指導いただきました平石邦彦教授には深く感謝致します。

また、審査員を引き受けていただきました、緒方和博教授、上原隆平教授、浅野文彦准教授には、本研究に対し多くのご助言をいただき、深く感謝致します。

副テーマ指導教員である白井清昭教授には、わかりやすいご指導をしていただきました。厚く御礼申し上げます。

最後に、JAISTでの学生生活を共に過ごし、共に切磋琢磨した友人、そして、生活面や精神面で支えてくれた家族へ心から感謝致します。

研究業績

口頭発表

1. 徳丸翔也, 平石邦彦

“CARATS Open Data における航空機位置のセクター決定問題について”, システム
数理と応用研究会 (MSS), 2018 年 3 月 13 日 (発表予定)

参考文献

[1]平田輝満, 米国における航空交通流管理の現状—Ground Delay Program に焦点をあてて—, 第 42 回土木計画学研究発表会・講演集, 2010.

[2]Banavar Sridhar, Neil Chen, Santa Cruz, Hok Ng, Santa Cruz, An Aggregate Sector Flow Model for Air Traffic Demand Forecasting, 9th ATIO, 2009

[3]国土交通省, <http://www.mlit.go.jp> (2018/02/05)

[4] Craig Wanke, Lixia Song, Stephen Zobell, Daniel Greenbaum, and Sandeep Mulgund, Probabilistic Congestion Management, The MITRE Corporation

[5]James S. DeArmon, Craig R. Wanke, Daniel P. Greenbaum, Lixia Song, Sandeep Mulgund, Steve Zobell, Neera Sood, Solving Probabilistic Airspace Congestion: Preliminary Benefits Analysis, 6th ATIO, 2007

[6]平田輝満, 羽田空港の滑走路運用特性に起因した航空機遅延の軽減方策に関する研究, 土木学会論文集 D3 (土木計画学), Vol.69, No.5 (土木計画学研究・論文集第 30 巻),I_869-I_880, 2013.

[7]青木隆平, 阿部和利, 上野誠也, 長岡栄, 中島陸博, 星次郎, 李家賢一, 渡辺紀徳, (2009) 『航空機の百科事典』, 丸善株式会社

[8] AIS JAPAN, Japan Aeronautical Information Service Center, <https://aisjapan.mlit.go.jp/>

[9] The Computational Geometry Algorithms Library, <https://www.cgal.org/>

[10] keisan, <http://keisan.casio.jp>

[11]地図投影法学習のための地図画像素材,
<http://keisan.casio.jp/exec/system/1257670779>

使用データ：国土交通省 CARATS Open Data