

管制運用に向けた航空交通流過密状態における
進化計算に基づく着陸順最適化

村田 暁紀

電気通信大学 大学院情報理工学研究科
博士(工学)の学位申請論文

2020年3月

管制運用に向けた航空交通流過密状態における
進化計算に基づく着陸順最適化

論文審査委員会

主査 高玉 圭樹 教授
委員 庄野 逸 教授
委員 大須賀 昭彦 教授
委員 柏原 昭博 教授
委員 佐藤 寛之 准教授

Copyright (C) 2020 Akinori MURATA All Rights Reserved.

Abstract

This thesis focuses on air traffic management under congested traffic around airport, coping with this issue by proposing the landing sequence optimization methods based on the free and fixed routes, and aims at investigating the effectiveness of these methods with the real flight data. For this purpose, this thesis employs evolutionary computation (EC) as one of metaheuristics methods and extends it as follows: (1) for the free route, the proposed method generates various landing routes of each aircraft and optimizes the landing sequence of all aircraft by combining the generated routes of each aircraft through EC; and (2) for the fixed route, the proposed method optimizes the landing sequence of all aircraft by repeating the cycle of dividing aircraft into the small number of clusters and reducing the conflict among aircraft in each cluster through EC. The intensive experiments of the proposed methods in the air traffic congestion of Haneda Airport and Charles de Gaulle Airport in France have been revealed the following implications: (1) for the free route, the proposed method can increase the total number of feasible landing route by combining the various landing routes, which contributes to reducing the total route length of aircraft. Since the proposed method evaluates routes from the both viewpoints of optimality (i.e., the total route length) and diversity (i.e., variety of the landing routes), it outperforms the methods that evaluates routes from the either viewpoint of optimality or diversity; and (2) for the fixed route, the proposed method solves the conflicts among aircraft by controlling the speed of the aircraft and guiding the aircraft to the detour routes within the cluster. In particular, the proposed method outperforms the horizon control as the conventional method in terms of the number of conflicts and the total delay time of aircraft.

概要

本研究では空港周辺の過密な航空交通流に対処可能な管制運用を目指し、自由・固定経路に基づく航空機着陸順最適化手法を提案し、その有効性を羽田空港及びフランス、シャルル・ド・ゴール空港の例題を通して有効性の検証することを目的とする。この達成に向け、管制負荷との単位時間当たりの着陸機数の量を示す着陸容量にはトレードオフがあることに着目し、管制負荷、着陸容量それぞれに対応する、自由・固定経路を想定した手法を提案する。自由経路に基づく着陸順最適化手法においては、最終的な着陸の際に適切な管制間隔が維持されるよう、複数の航空機に経路を独立に進化させ、経路の候補を複数生成する。その後、この多様な経路の組み合わせから構成される解候補の組み合わせを探索する多段階分散進化による着陸順最適化手法を提案する。各航空機が独立的に経路を生成することは最適化の処理時間の観点で有効であるが、全体化(着陸順の探索)にあたり他の航空機への解の考慮が必要となる。そこで解の良さを示す経路の評価と多様な経路を候補と保持することを意味する多様性の評価の二目的最適化において解決を目指す。この手法は多目的最適化進化計算の手法の一つである、NSGA-IIをベースとしており、その評価指標として、「最適化」を評価する解の良さと、「多様性」を評価するノベルティサーチを導入した後、(1) 評価値の高い解の近傍を重点的に探索する重点サンプリングと、(2) 評価値の高い解から離れる範囲を探索しない多様性制限を組み込むことで最適性と多様性のトレードオフの問題を改善した。次に固定経路においては、混雑環境に対応するため、実際の運用において使用されている固定経路に迂回経路を付加し、この迂回経路と速度の変更により時間調整することで航空機同士の間隔の維持を図る。この時、航空機の速度及び迂回経路により適切な航空機同士の間隔を維持するような着陸順生成のため、決定変数である、全航空機の速度と迂回量を最大限活用することが求められる。本研究では、最適化対象の航空機を分割し航空機群を並列的に最適化する処理を複数回繰り返す分割し、反復最適化手法を提案する。解の探索範囲を限定することで、実時間内で実行可能な着陸順の生成し、クラスタリングと最適化の反復により、分割群によるスケジュールの偏り(局所解への停滞)を脱出を図る。この手法実現のため、具体的には、全航空機をクラスタリング手法であるk-means法により、予想到着時刻を基準に複数の航空機群に分割し、分割した航空機群を個別に最適化を行うことで対応する。また、群同士の到着予定時刻に対す

る前後着陸順の関係を維持するため、最適化対象の航空機群の前後航空機群の到着時刻を固定することで前後の航空機間でのコンフリクトを防ぐ。自由経路における例題として進入禁止空域および、実際の入域地点を想定した羽田空港において経路・着陸順同時最適化手法を適用し以下の知見を得た。(1) 個体 (航空機) における経路評価を最短経路長にした場合に他の航空機との距離間隔が制約を満たさず、実行不可能な着陸順となるが、多様性の評価指標であるノベルティを導入することで複数航空機における経路生成及び着陸順の実行可能な解生成が可能であることがわかった。さらに、(2) 個体 (航空機) の評価指標である経路長と多様性の評価を二目的化し、着陸順最適化性能向上のため、重点サンプリングと多様性制限を導入したところ、現在の交通量を超える 1 時間に 48 機の着陸順最適化に成功し、最短経路のみの評価、ノベルティのみの評価それぞれの評価に比べ、0.5 % 水準で優位であることが明らかになった。次に固定経路における例題として、入域地点が 8 カ所あり、5 つの迂回経路を付加したシャルル・ド・ゴール空港における実際に定義される固定経路を用いて実験を行った。トラフィック量としては、実行不可能となるような制約が厳しい (コンフリクト解消が困難) ようなシナリオを提案手法である、クラスタリングに基づく分割反復最適化手法は従来の手法である Receding horizon control による最適化対象の航空機数を時間窓により分割し、逐次最適化する手法に適用し、性能比較を行ったところ以下の知見が得られた。(1) 分割による最適化により、従来の分割手法より最適性の高いスケジューリング (航空機の遅延時間が小さい) を獲得可能であること。(2) 分割を複数回繰り返すことでより良い航空機群の組み合わせを探索が出来る可能性があり、これを繰り返すことにより着陸順の最適化性能が向上し、(3) 全航空機がコンフリクトのない状態で航行可能な実行可能な着陸順が生成可能であるが明らかになり、分割と最適化を複数回行うことの効果が明らかになった。

目次

第 1 章	序論	1
1.1	研究背景	1
1.2	研究目的と方法	3
1.3	本論文の構成	4
第 2 章	航空交通管理と関連研究	5
2.1	航空管制	5
2.2	最適化と航空管制	13
2.3	Aircraft Landing Problem	16
2.4	ALP の拡張 (Merging Flow Optimization)	20
2.5	管制運用と自由・固定経路	23
2.6	研究の位置づけ	25
第 3 章	進化計算と機械学習	33
3.1	単目的遺伝的アルゴリズム	33
3.2	多目的遺伝的アルゴリズム : NSGA-II	36
3.3	ノベルティサーチ	40
3.4	クラスタリング	43
第 4 章	羽田空港とシャルル・ド・ゴール空港	46
4.1	羽田空港	47
4.2	シャルル・ド・ゴール空港	49
4.3	羽田空港におけるモデリング	53
4.4	シャルル・ド・ゴール空港におけるモデリング	55
第 5 章	自由経路による着陸順最適化	60
5.1	アプローチ	60
5.2	経路・着陸順同時最適化手法	62
5.3	最適性と多様性のトレードオフを考慮した経路・着陸順同時最適化手法	72

5.4	実運用での最適化	87
第 6 章	固定経路による着陸順最適化	105
6.1	アプローチ	105
6.2	進化計算に基づく固定経路による着陸順最適化	106
6.3	クラスタリングを用いた分割反復最適化手法	118
6.4	クラスタ数自動決定法	155
第 7 章	自由経路・固定経路と管制運用の親和性	160
7.1	自由経路における個体から全体の設計論的観点	160
7.2	実問題における GA のパラメータ設定指針	161
第 8 章	おわりに	163
8.1	まとめ	163
8.2	課題	165
参考文献		169

目次

1.1	羽田空港における滞留の例	2
2.1	航空管制の分類	5
2.2	航空管制業務の詳細	7
2.3	航空図	8
2.4	空港周辺における業務	9
2.5	羽田における標準到着経路の例	10
2.6	管制間隔	11
2.7	手法の比較	13
2.8	ALP の定式化	16
2.9	最適化における時間窓とステップの関係	18
2.10	迂回経路構造一例：Point Merge	21
2.11	自由経路における前提	23
2.12	固定経路における前提	24
2.13	経路と着陸容量の関係	25
2.14	本研究の全体の位置づけ	26
2.15	動的計画法に基づく燃料消費量最小化	27
2.16	降下点から滑走路への到着時間を指定した経路計画法	28
2.17	風を考慮した最短時間での経路生成法	29
2.18	光の伝搬モデルに基づく経路生成法	29
2.19	Mori らによる航空機合流法	30
2.20	固定経路における迂回経路を利用した手法	31
3.1	遺伝的アルゴリズムの流れ	34
3.2	パレート最適フロント	36
3.3	クラウディングディスタンス	38
3.4	NSGA-II ソート	40
3.5	ノベルティサーチを評価に用いた場合の解の分布	41

3.6	Fitness を評価に用いた場合の解の分布	42
4.1	羽田空港	46
4.2	ベクタリングの例	47
4.3	yokota ACC	48
4.4	Charles de Gaulle Airport (シャルル・ド・ゴール空港)	49
4.5	CDG における STAR の定義例	50
4.6	取り扱い機数のヒストグラム	53
4.7	グリッドマップへのモデリング	54
4.8	羽田空港におけるグリッドマップモデリング	55
4.9	シャルル・ド・ゴール空港における STAR	56
4.10	CDG における迂回経路	57
4.11	迂回経路	58
4.12	羽田空港の速度の推移	59
5.1	自由経路におけるアプローチ	61
5.2	自由経路における経路・着陸順最適化の概要	62
5.3	経路最適化手法における遺伝子構造	63
5.4	中継点平均方式	65
5.5	中継点連結方式	65
5.6	中継点組み換え方式	66
5.7	中継点追加	66
5.8	中継点移動	67
5.9	中継点削除	67
5.10	着陸順最適化部 (全体)	68
5.11	着陸順の遺伝子表現	69
5.12	多様性と経路長の 2 目的最適化	73
5.13	経路とその評価の関係性	74
5.14	範囲制限と重点サンプリング	75
5.15	重点サンプリング	76
5.16	多様性制限	77
5.17	自由経路における比較ケース	77
5.18	例題：羽田空港 (再掲)	78
5.19	生成された経路	80
5.20	着陸間隔	81
5.21	ノベルティを利用した場合の経路の多様性	82

5.22	重点サンプリングを使用した場合の結果	83
5.23	範囲制限を使用した場合の結果	84
5.24	β を変化させたときの総経路長の違い	85
5.25	W を変化させたときの総経路長の違い	85
5.26	β を変化させたときの経路集団の違い (ID1 から ID3)	86
5.27	連続環境下における航空機の入域	87
5.28	迂回経路の付加方法	88
5.29	2 回目の最適化における着陸順最適化	89
5.30	迂回経路の生成法	90
5.31	2 回目の最適化における Fitness	92
5.32	3 回目の最適化における Fitness	93
5.33	最適化毎の総経路長	94
5.34	迂回経路の利用を全機許容した場合と 2 機許容した場合の差異	95
5.35	ステップ毎での平均経路長 (3 機)	97
5.36	ステップ毎での平均経路長 (4 機)	97
5.37	ステップ毎での平均経路長 (5 機)	98
5.38	ステップ毎での平均経路長 (6 機)	98
5.39	ステップ毎での平均経路長 (7 機)	99
5.40	7 機での実行不可能解の総数	100
5.41	4 機の生成された総経路長の平均	100
5.42	6 機の生成された総経路長の平均	101
5.43	ステップ毎の各航空機の経路長	102
6.1	固定経路におけるアプローチ	106
6.2	遺伝子構造	107
6.3	最大違反量	111
6.4	平均違反量	112
6.5	解修復法における Fitness の推移	114
6.6	コンフリクトの推移	115
6.7	実行可能解到達世代の箱ひげ図	116
6.8	selected gene of infeasible individual	116
6.9	航空機全体の探索効率化の着眼点	118
6.10	固定経路における全体の構成	119
6.11	クラスタリングとスケジュールの関係	120
6.12	最適化部における GA の構成	121
6.13	分割した際の最適化の順番	122

6.14	実験環境 (再掲)	123
6.15	最適化対象の航空機	125
6.16	反復回数を 1 回にした場合の Fitness とコンフリクト数の結果	125
6.17	クラスタ数を 5 にした場合の Fitness とコンフリクト数の結果	126
6.18	反復回数を 20 回とした場合の Fitness の推移	127
6.19	2 試行目における Fitness の推移	128
6.20	最適化毎の所属クラスタの変化	128
6.21	変化の生じた航空機の詳細	129
6.22	最良シードにおける航空機の遷移	133
6.23	初期速度迂回無しでの遷移	134
6.24	各クラスタリング手法における Fitness の推移	136
6.25	世代ごとの Fitness の推移	138
6.26	初期世代と最終世代の性能差	139
6.27	各航空機の詳細 (反復回数 5, 世代数 200)	140
6.28	各航空機の詳細 (反復回数 10, 世代数 100)	141
6.29	各航空機の詳細 (反復回数 20, 世代数 50)	142
6.30	各航空機の詳細 (反復回数 25, 世代数 40)	143
6.31	各航空機の詳細 (反復回数 40, 世代数 25)	144
6.32	世代ごとの Fitness の推移	145
6.33	初期世代と最終世代の性能差	146
6.34	各航空機の詳細 (反復回数 5, 世代数 200)	147
6.35	30 機における各航空機の詳細 (反復回数 10, 世代数 100)	148
6.36	各航空機の詳細 (反復回数 20, 世代数 50)	149
6.37	各航空機の詳細 (反復回数 25, 世代数 40)	150
6.38	各航空機の詳細 (反復回数 40, 世代数 25)	151
6.39	クラスタリング数決定のためのアプローチ	155
6.40	クラスタ数の決定方法	156
6.41	単数ユニットと複数ユニットでの Fitness	158
6.42	複数ユニットにおける最終的に選択されたクラスタ数	159
7.1	自由経路における連続最適化環境	160
1	クラスタリング毎の Best Fitness (30 試行)	191
2	Median による航空機の詳細	191
3	Average による航空機の詳細	192
4	Centroid による航空機の詳細	193

5	Complete による航空機の詳細	194
6	Mcquitty による航空機の詳細	195
7	4 パッケージの中での最良評価値を持つクラスタ数のヒストグラム	196
8	4 パッケージの中での最良・最悪シードにおける Fitness の推移	197
9	迂回を使用しない場合における Iteration を変化させた場合の Fitness の 推移	197

表目次

2.1	航空管制の略称と名称	6
2.2	航空機のサイズによる管制間隔	12
2.3	航空機のサイズによる管制間隔 (再掲)	17
2.4	自由経路における研究の位置づけ	27
2.5	固定経路における研究の位置づけ	31
4.1	CDG における Waypoint の位置	51
4.2	入域地点毎の機数及び航空機サイズ	52
5.1	中継点の個数	64
5.2	経路着陸順最適化パラメータ	79
5.3	25 試行での総経路長の比較	80
5.4	パラメータ (再掲)	92
5.5	各最適化における迂回経路の利用回数	96
5.6	6 機における実行不可能解となるステップ数	104
6.1	GA におけるパラメータ	113
6.2	固定経路におけるシミュレーションのパラメータ	113
6.3	分割反復最適化におけるパラメータ	124
6.4	クラスタ数と総遅延時間と総コンフリクト数の関係	130
6.5	航空機の詳細スケジュール (seed2)	131
6.6	管制運用での実際の到着時刻	132
6.7	航空機の詳細スケジュール (seed2)	137
6.8	GA のパラメータ	152
6.9	19 機におけるコンフリクト数の推移	153
6.10	19 機のシナリオにおける航空機の詳細	154
6.11	各シナリオにおけるデータ	155
1	シャルル・ド・ゴール空港における航空機情報の詳細	182

2	自由経路における航空機の初期位置	182
3	16 機のシナリオにおける航空機の詳細	186
4	17 機のシナリオにおける航空機の詳細	187
5	18 機のシナリオにおける航空機の詳細	188
6	18 機のシナリオにおける航空機の詳細	189
7	19 機のシナリオにおける航空機の詳細	190

第 1 章

序論

1.1 研究背景

ライト兄弟が飛行機を発明し以来，人類の発展に航空機は多大に貢献しており，インターネット網が発達した現在においても人の輸送手段として航空機はその主たる手段として利用されている．国際民間航空機関（International Civil Aviation Organization : ICAO）によると，世界の航空需要 40.7 億人であり，今後も数十年以内で現在の 2 倍となる予測がある [1]．イギリスのヒースロー空港では 2 本の滑走路を保有しており，着陸と離陸を分離して運用することが可能である．この時の 1 時間当たりの着陸容量は 44 本されている [2]．世界でも特にスケジュール遅延の発生する，ロンドンヒースロー空港やアメリカのシカゴオヘア空港は多くの航空機が利用するため，しばしば，出発時間の遅延や空中待機などの制限がおこる．将来の航空需要に対応すべく，EU では SESAR プロジェクト [3]，アメリカでは NEXTGen [4] といった次世代航空交通システムプロジェクトが展開されており，日本では CARATS と呼ばれるプロジェクトの下，産学官が連動して航空交通システムの研究開発を進めている [5]．

近年，特に混雑する空港周辺においてはコンフリクトのない合流が必要となり，国や国際機関が連携し研究が展開されている．コンフリクトとは，航空機同士の間隔が短く，衝突の危険性のある状態である．現在の管制方式は First-Come First Serve (FCFS) を原則としており，この原則に従い航空管制官が航空管制レーダ画面で表示される航空機の順に着陸の指示を行う．このような管制方式は航行順に管制を行うために，管制官の直観として，理解及び処理を行い易い方式となっており，航空管制官は安全かつ効率的な管制の実現に尽力をしているが，これが遅延発生の主要な原因であり，多くの航空路から目的空港へ航行する航空機の合流の間隔付けのため，滞留が生じる．

図 1.1 は羽田空港周辺の混雑時における航空機を表している．黄色のシンボルは空域を航行している航空機を示し，赤い丸で囲った空域が羽田空港であり，到着機は空港の南側から進入している．南東の航空機の集団は成田空港への到着機を表している．航空管制の



図 1.1 羽田空港における滞留の例

中で、空港周辺が離発着機のため空域内の混雑が発生し、需要過多が見られる。この中でも特に着陸機の管制は、出発機とは異なり、到着機が目的空港へ航行を既に行っているため、到着時間を遅延させることには限度ある。また、航空機の機種によって航空機後方に発生する後方乱気流と呼ばれる気流の影響により着陸から一定の時間離発着不能となる。この関係は前方に大型の航空機、後方に小型の航空機の場合には時間間隔を最大にしなければならないため、着陸機において間隔を最小限に留める必要があり、将来の需要の増加に対応する手法が早急に求められる。

以上のような現状から、航空機の目的空港への着陸は航空機着陸問題 (Aircraft Landing Problem:ALP) と呼ばれ、多くの研究者によって取り組まれてきた [6]。この中で Besley らはこの問題を数理モデル化し、FCFS に従い生成された着陸順よりも短い時間でスケジュール対象の航空機の着陸が行えることを示したが、具体的な経路の仮定のないモデル化のため、航空管制官が直観的に指示できるものではなく、具体的な経路を仮定した着陸順の最適化が必要となる。そこで、近年では、空港周辺のコンフリクト解消を目的とした交通流合流問題がスケジュール問題として定義され、実際に管制官が運用で利用する到着経路を経路として仮定した着陸順最適化手法が提案されている [7]。ただし、これらの手法は管制官が取り扱えない制御となるため、実際の航空管制の運用では適用することが不可能であり、航空管制官の運用に資する方法論での着陸順最適化手法が求められる。

管制官で制御する場合にはベクタリングと呼ばれる航空機の速度、方位、高度をパイ

ロットに向かって指示する。管制官はセクタと呼ばれる一定の領域の航空機の管理を行っており、混雑状況においてはコンフリクト状態を避けるため、多くの航空機に指示を出す。この時、管制官にとってワークロードが高い状態となり、ヒューマンエラーを引き起こす可能性があるため、出来るだけ航空機に対して指示を減じるような管理を行うことが望ましい。この時に、航空機数と航空管制官のワークロードのトレードオフ関係となる。ここで、本研究ではベクタリングにおける方向制御に着目する。ベクタリングを用いて管制運用する場合には、航空機の機種方向を制御するため、各航空機が自由に航行できる自由経路を仮定する必要がある。ベクタリングを用いない場合には、全航空機が同じ経路に沿って航行を行う固定経路を仮定し、航空機を速度を制御する必要がある。本研究では、異なる経路の設定基準から交通流の過密状態における交通流の獲得を目指す。

航空機が自由に航行可能な状況においては、制御の柔軟性の観点から交通流過密状態において航空機の管理を行うことは可能であるが、管制官のワークロードを増大させる恐れがある。一方で、航空機が定められた経路に沿って航空機を航行させることは管制官のワークロードを減じることが可能であるが、過密状態において、多くの航空機を着陸に誘導することが困難である。

1.2 研究目的と方法

将来的に想定される航空交通需要の増加より、航空管制官に多大な負担がかかる。この航空需要と航空管制官への負担のトレードオフ関係があるため、同じ方向から空港周辺に入域する航空機は同一経路を利用する固定経路と全ての航空機が異なる経路を利用する自由経路という経路の仮定が異なる二つのアプローチにより、需要増加に伴う過密状態への対応と航空管制官の支援を目指す。この二つにより過密状態における管制運用に向けた着陸順最適化手法を構築する理由としては、実際の運用では、平時においては、固定経路を利用した着陸誘導を行い、混雑時においては各航空機を航空管制官が直接誘導を行うため、固定経路に沿って航行しない場合が多い。

過密状態における着陸順の最適化を達成するために、各航空機の経路を自由に設定することで混雑時の航空機の間隔を柔軟に変更する方法と、固定経路においては、到着時間毎に航空機群を形成し、着陸予定時間に基づく航空機の分割をすることによる探索空間の削減法を構築し、その有効性の検証を明らかにすることを目的とする。

本研究で提案する方法は二つの空港を例題に検証を行う。まず、自由経路における例題として、羽田空港の着陸問題を用いる。この問題は羽田空港周辺をグリッドマップにモデル化を行い、混雑時を想定した交通量における着陸順最適化を行う。固定経路における例題として、フランスシャルル・ド・ゴール空港における、航空機の合流・着陸問題を用いる。自由経路の場合には実際に運用の際に用いられる経路に基づいた着陸順の最適化を行う羽田空港は需要の特性として、空港の西側の空域において過密状態となり、しばしば管

制官のベクタリングによる運用を行う。このような状況下においては自由経路を仮定とした誘導が求められるため、自由経路における例題として採用する。固定経路においてシャルル・ド・ゴール空港を例題とした理由は、全方向に着陸経路が合計で8本設定されており、この経路を合流ための合流点も多数設定されているため、多数の合流を要求される環境が固定経路における最適化の環境としては適当である。

1.3 本論文の構成

本論文は8章から構成される。はじめに関連研究及び本研究で用いるアルゴリズムを説明し、その後、第5章以降において本論文で提案する方法について述べる。第1章では、航空管制における現運用とこれから航空需要の予測とそれに伴う現運用方式の限界について説明し、これからの管制システムに求められることを述べ、提案する過密状態における管制運用に向けた着陸順最適化手法の概要と研究目的を述べる。第2章では現在の航空交通管理の方式と航空管制官の業務や運用方法について述べ、ここから、現在の運用方法の問題点を整理する。その後、航空機着陸順問題について説明し、従来の着陸順最適化手法と経路最適化手法をまとめ、従来手法の問題点を指摘し、研究の位置づけを示した。第3章では、最適化手法の一つである遺伝的アルゴリズムと解集団の個体評価手法であるノベルティサーチ、データを分類手法であるクラスタリング手法について説明する。第4章では、本研究で提案する手法を適用する空港である羽田空港とシャルル・ド・ゴール空港に関する地理的特徴やトラフィックパターンについて説明する。その後、それぞれの空港におけるモデリング方法を説明する。第5章では、自由経路を想定した経路・着陸順最適化手法を提案する。はじめに着陸順を生成するためベースとなる経路生成法に関して説明し、その後、実行可能な着陸順生成のための着陸順生成法について説明した後、性能比較実験を行い、提案手法により実行可能な経路及び着陸順が可能であることを明らかにする。その後、経路の多様化に対する効果の考察を行う。第6章では固定経路を想定した場合の着陸順の効率を高めるためのクラスタリングに基づく分割反復最適化手法を提案し、フランスのシャルル・ド・ゴール空港において、実際のトラフィック量において実行可能な着陸順が獲得可能か検証を行う。はじめに、固定経路における進化計算による最適化方法について説明し、実行可能な着陸順を早期に獲得するための解修復法を提案し、その有効性の検証を行う。その後、混雑時における最適化性能を高めるため、分割反復最適化手法を提案し、その有効性の得検証を分割数、クラスタリングの回数を変化させ比較を行う。その後、その効果について考察する。第7章では、自由経路、固定経路の総括をおこない、二つの最適性の観点でのスケジュール性能の比較、実行可能なスケジュールを獲得可能な単位時間当たりの着陸機数と最適化における推奨航空機数について解析的に述べている。

最後に、第8章では上記の各章で得られた知見をまとめ、本研究の成果について述べるとともに、今後幅広い問題に適用する上での課題を述べる。

第 2 章

航空交通管理と関連研究

2.1 航空管制

2.1.1 航空管制の分類と業務

民間航空機は航空管制官の指示の下運行しており，航空管制官が日本の空の安全を担っている．航空管制業務は空域，飛行場における航空機が安全に運航出来るよう主に航空機同士の間隔の管理，気象情報の提供，交通情報の提供，搜索救難における関係機関への連絡，協力を行っている．

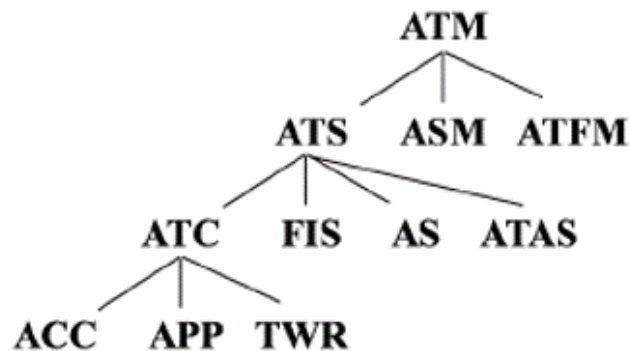


図 2.1 航空管制の分類

図 2.1 は航空管制の分類を示しており，それぞれの略語の名称は表 2.1 にて記載する．ATM 業務は ATS, ASM, ATFM の 3 つの業務に分類される．ASM は訓練空域を有効利用するため，自衛隊との連絡調整や特定の空域が混雑する状況下において，セクタを設定し直すことで，通常以上の処理を行えるよう調整することをいう．このような通常時や混雑時における空域の最適化を行う研究も存在する [10–12]．ATFM は飛行計画経路の管理，管制承認の発出，交通流制御の実施の 3 つが主な業務であり，日本では福岡県に

表 2.1 航空管制の略称と名称

ATM	Air Traffic Management	航空交通管理
ATS	Air Traffic Service	航空交通業務
ASM	Air Space Management	空域管理
ATFM	Air Traffic Flow Management	航空交通流管理
ATC	Air Traffic Control	航空交通管制
FIS	Flight Information Service	飛行情報業務
AS	Alert Service	警急業務
ATAS	Air Traffic Advisory Service	航空交通案内業務
ACC	Area Control Center	管制区管制所
APP	Approach Control	ターミナル管制所
TWR	Tower	飛行場管制所

ATFM センターが整備されている，この中で特に航空交通流が過密時に，出発航空機や飛行中の航空機に対し，出発の遅延や，空中待機（Holoding）を命じ，混雑する空港，空域に対して入域を制限するような指示を管制官が行う [13]．ATFM は現在の管制処理量に対し，航空需要が飽和すると見込まれる際に ATFM 業務における交通流制御が実施される．ATS は以下の業務が該当する．

1. 航空機相互の衝突を防止するため，適切な航空機間の間隔を維持すること
2. 空港において地上走行中の航空機，車両の衝突回避の管理を行うこと
3. 気象情報，管制空域における交通情報を提供すること

この中で，ATS 業務は ATC，FIS，AS，ATAS に細分化される．ATC は航空機の安全で効率的な運航を行うため，航空機に指示，支援情報を与える．FIS は運航に必要な情報を提供する業務を主としており，飛行情報区（FIR）内に配置されたセンターにおいて，以下の 6 つの情報や助言を行う．（1）異常気象情報（2）航法援助施設の運用状況と運用に関する情報（3）飛行場及び付属施設の状況に関するその他の情報，（4）航空安全に関する情報，（5）出発地，目的地，代替空港に関する気象情報，（6）管制区または管制圏内外の衝突予防情報が主たる業務となる．

次に AS は救難に必要な情報を伝達する業務を執り行う．具体的には，位置通報のない航空機，航行性能が悪化した航空機がある場合には救難調整本部（Search and Rescue Region）に通報し，SRR が搜索救難業務を遂行する．ATAS は空域での航空機同士の間隔が確保されるよう，他の航空機との関係の注意を喚起することが主たる業務となる．交通流のボトルネックとなるのが ATC 業務である．ATM 業務中には先述した通り，

安全に係る業務である情報支援業務や警急業務などの業務も含まれる。そのため、航空機の運航上の安全で効率的な支援を行う ATC 業務の詳細についてここから説明する。

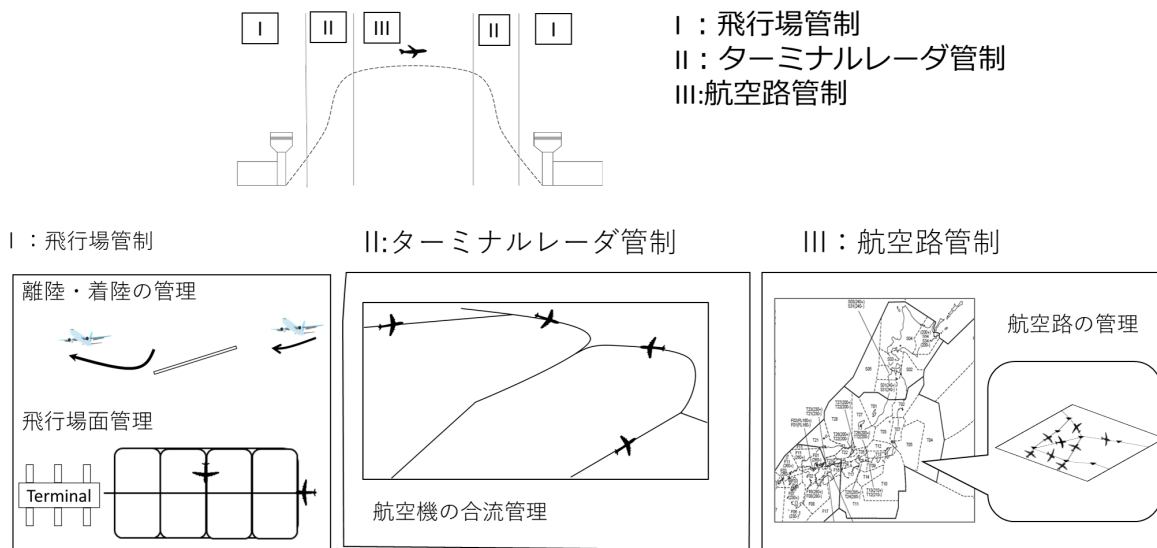


図 2.2 航空管制業務の詳細

図 2.1.1 は ATC は 3 業務に分類の説明を示している。I は TWR と呼ばれ、空港において滑走路を使用する航空機及び、管制圏内の航空機に対して指示を行う。航空機の離着陸の許可を出す。また、空港において滑走路以外の走行区域に関する管制も行うが、エプロン内の移動に関しては対象外となる。APP は II を表し、空港周辺の空域を航行する航空機に対して、進入、出発の順序、経路、着陸方式の指定、上昇や効果の指示を行う業務である。航空路から、空港周辺へ入域する際にはレーダを用いて、管制官が方位、速度を指示するベクタリングを行う。最後に、III の ACC は航空路管制と呼び、En-route と呼ばれる航空路を航行する航空機の管理を行う。

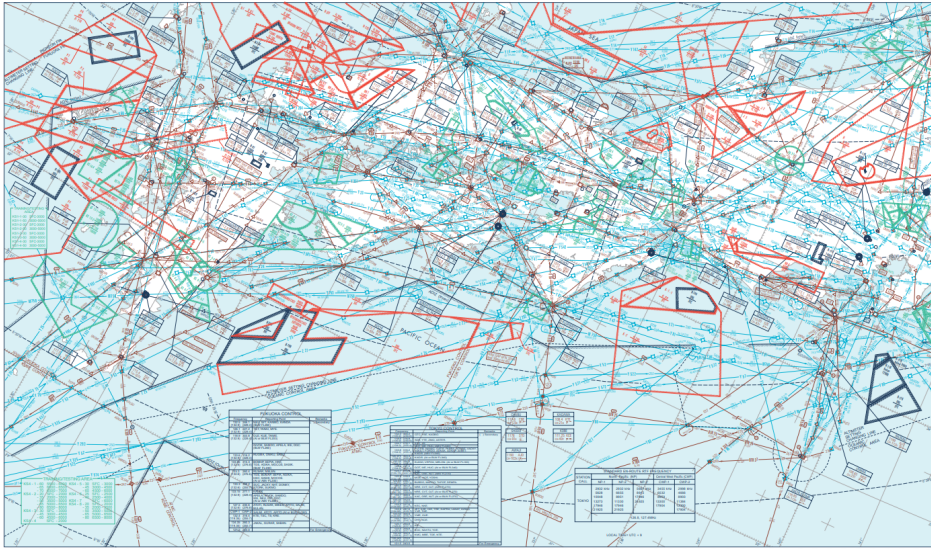


図 2.3 航空図

参考 : <https://aisjapan.mlit.go.jp/html/AIP/html/20200227/eAIP/20200227/pdf/JP-ENR-6.2-en-JP.pdf>

図 2.3 は日本の航空路の一部を示しており，この航空図（En-route chart）は国土交通省航空局が発行し，図の水色，茶色で示させる線が航空路を指す．緑，赤，青枠は飛行制限空域であり，空域によっては自衛隊の訓練，民間機の訓練を行う空域となる．この他に，航行援助施設（航空機の位置情報を提供する施設），無線周波数，計器進入方式等が記されている．この航空路において飛行経路，高度等の指示を行う業務を遂行する．また，航空路同士が交差している空域が多く存在し，航空機の交通量が増えると慢性的に航空機同士のコンフリクトが発生する．コンフリクトとは航空機同士の距離間隔が一定以下となる状態である．これを管制官が高度の変更，進路の変更により解消を図る．将来，航空需要が増えることで，潜在的にコンフリクトが起こる可能性が高くなるため，コンフリクトを検知し，それを解消する研究も盛んに行われている [14–18]．

図 2.4 は空港周辺業務の中でも，滑走路に関連する業務を表している．空港周辺業務には主に飛行場面管理（Airport surface management:S-man），出発管理（Departure management:D-man），到着管理（Arrival management:A-man）の 3 業務に分かれる．S-man は飛行場面における航空機の経路の最適化を指し，飛行場面に対してを飛行場面を移動する出発機，到着機等の航空機同士コンフリクトのない経路の管理を行う．D-man は出発機の出発タイミングおよび，出発順に関しての管理を行う．A-man は到着機の間隔及び着陸順に関する管理を行う．この中で S-man は飛行場面管理であるため，誘導路（駐機場と滑走路をつなぐ路）での遅延時間は空港の大きさにも依存するが，遅延量としてはそれほど大きくない．その上，交通流全体としてボトルネックとなっているのは航空路から滑走路へとつなぐ到着経路と滑走路であるため，ボトルネックへの影響は低い．次に D-man だが，出発機は滑走路を利用して目的空港へと航行するが，出発機に関する遅延

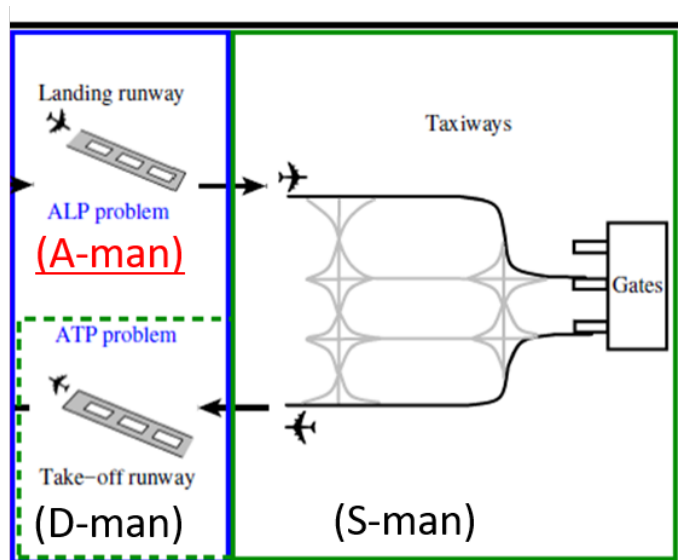


図 2.4 空港周辺における業務

は乗客による遅延と機材繰りによる影響が大きい。そのため、S-man と同様、ボトルネックの影響は限定的なものに留まる。A-man については、ボトルネックの影響が一番大きいといえる。これは、管制官は FCFS の原則で航空機の着陸順を生成するため、多くの航空機が着陸出来るよう順位付けと航空機間隔の維持が重要である。このような理由から、本論文では、着陸順最適化に取り組む。

2.1.2 標準出発方式及び標準到着方式

航空機が空港の滑走路を離陸後上昇し、航空路に合流するための経路及び飛行方式を標準計器出発方式と呼ぶ (Standard Instrument Departure: SID)。空港周辺という特性上、飛行する方式が細かく設定されている。例えば、出発時における上昇角、旋回方向や速度、空域制限がこれに当てはまる。次に、航空路から目的空港へ巡航時、空港周辺に到達すると着陸のため、高度を下げ、空港へと進入するが、この空港と航空路を接続する経路を標準到着方式 (Standard instrument arrival: STAR) と呼ぶ。STAR は航行するための飛行方式が定められており、これを遵守し航空機は航行しなければならない。例えば、通過すべき地点を示すウェイポイントや、飛行高度、巡航速度等が指定される。但し、管制官により指示がある場合には STAR に従わず航行を行う場合がある。

図 2.5 は実際の STAR を示しており、羽田空港南側からの到着経路を示している。いくつかの種類のマーカーが存在するが、ローマ字表記で示された文字が Waypoint と呼ばれるマーカーの名前を指している。四角で囲まれた情報には例えば、OSHIMA には VORTAC, TATEYAMA には DME, HANEDA には VOR/DME との表記がされてい

る。これは、VORTAC は方位と距離、DME は距離、VOR/DME は方位と距離を巡航中の航空機に情報を送信している。例えば、航空機が OSHIMA 付近を航行中に VORTAC はこの Waypoint と航空機の方位と距離関係を航空機に対して送信する。航空機はこの Waypoint 同士を結んだ線上を航行する。例えば、関西空港から羽田空港に向かう航空機は OSHIMA から UTIBO, UMUKI, KAIHO を通過し、羽田空港へ着陸する。また、OSHIMA にはホールディングポイントと呼ばれる空中待機空域が設定されており、空港の近隣において非常に混雑する場合にこのホールディングポイントを用いて、到着時間を遅延する処理を管制官は行う。

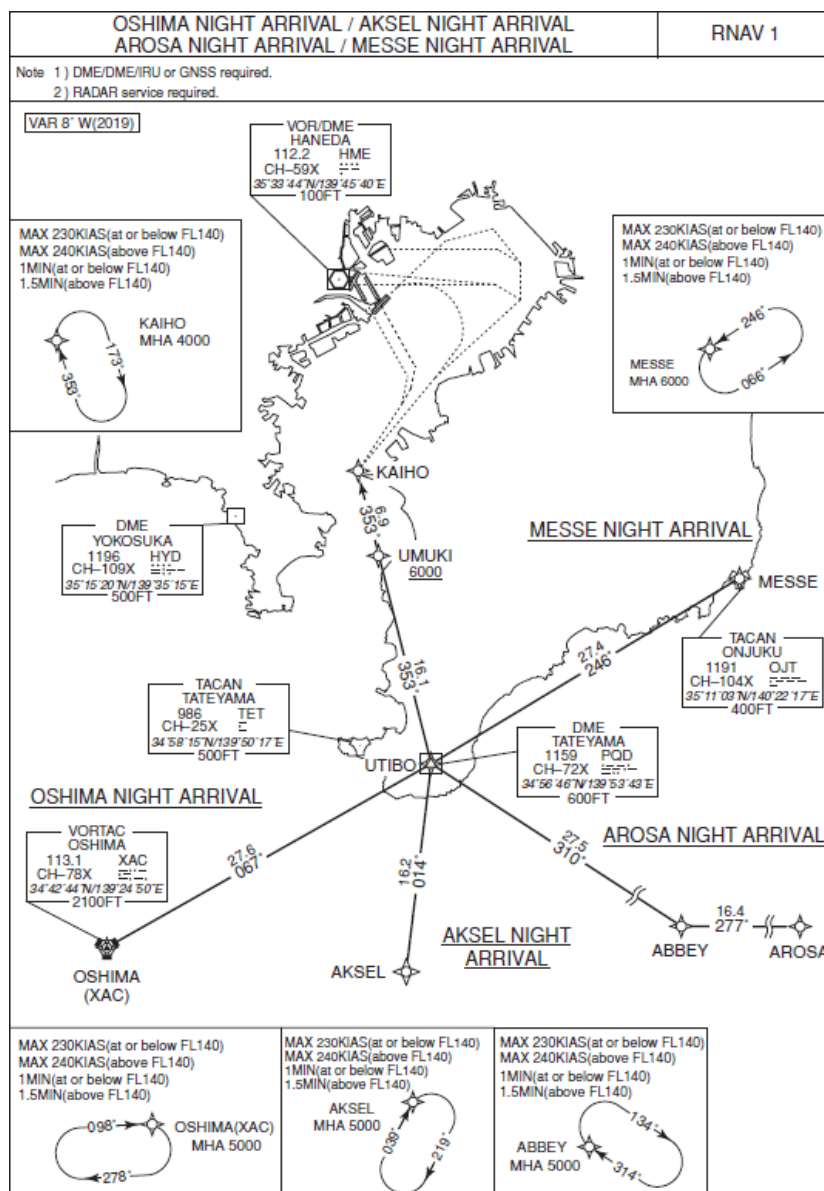


図 2.5 羽田における標準到着経路の例

参考：AIS JAPAN

2.1.3 管制間隔

管制間隔は縦間隔，横間隔及び垂直間隔の3種類ある。図 2.6 は左から，縦間隔，横間隔，垂直間隔を表しており，縦，横間隔は水平間隔であり，垂直間隔は航空機の航行における高度差を表す。具体的な間隔は航空機の機材使用する航法装置により適用できる間隔が異なる。

縦間隔においては，国内の航行はレーダを用いて航空機間の間隔を把握しているが，レーダの場合でも適用する区域により間隔が異なる。洋上の経路においては30から50NMの間隔が必要となる。航空路においては航空機の同士の間隔として10NM（18,520m）必要であるが，航空路から空港までの経路においては3NMから8NMの間隔が必要となる。

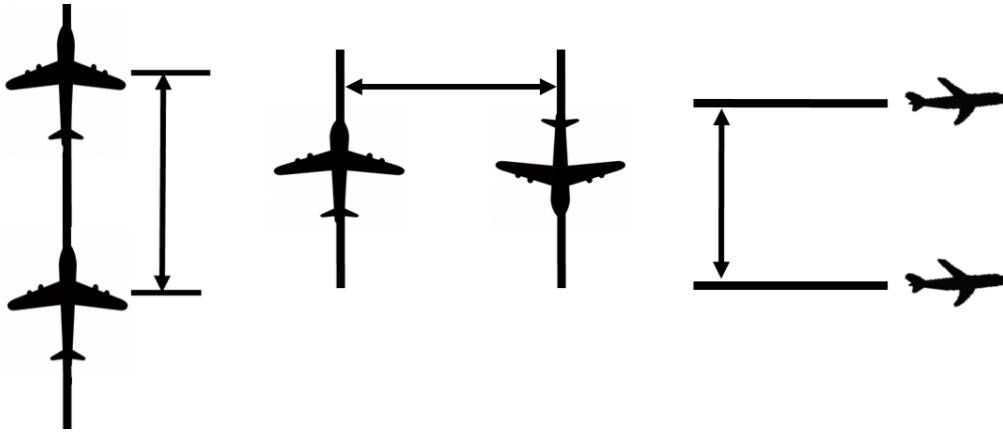


図 2.6 管制間隔

表 2.3 は航空機の分類ごとによる間隔を示しており，S.Heavy は super heavy を示している。S.Heavy に該当する航空機は 2019 年現在は AirbusA380，1 種類のみである。Heavy に該当する航空機は Boeing 社では B787，B747，B767 等が該当し，Airbus 社では A340，A330 等が該当する。Midium は B737，A320 等が該当する。Light に関しては，最大離陸容量が 7,000kg を上回らない機種が該当する。表における Trailing aircraft，Leading aircraft は航空機の前後関係を表し，Trailing が後ろを航行する航空機，Leading が前方を航行する航空機である。例えば，前方に A380 が航行し，その後方に B737 が航行する場合には 6NM の間隔が必要となる。

横間隔に関しては，洋上では Pacific Organized Track System(PACOTS) の経路間として 50NM が設定され，国内の航空路においては 10NM の間隔が設定されている。垂直間隔は 1000ft 毎に方向が逆の経路が設定されている。

表 2.2 航空機のサイズによる管制間隔

		Leading Aircraft			
Trailing Aircraft		S. Heavy	Heavy	Midium	Light
	S. Heavy	3	3	3	3
	Heavy	5	4	3	3
	Midium	6	5	3	3
	Light	8	6	4	3

2.1.4 航空機の着陸順と作業負荷

航空管制は航空機に対する航行支援であるため、エアラインは航行支援に関する優先権は保有せず、全航空機が平等に支援受けることができる。ただし、バードストライク等の緊急事態（エマージェンシー）時にはそれを宣言する航空機がすべてにおいて優先される。それ以外においては基本的には First-Come First-Servervice(FCFS) の原則において航空管制官は航空機を支援する。ただし、着陸時において、先行機と後続機の航空機のサイズの関係によって維持すべき航空機の間隔が異なるため、状況によっては着陸の効率が悪い可能性がある。

そこで、近年では航空機の着陸順の最適化（最適な順位付け）を行う研究が多くの研究者により展開されている [6]。着陸順の最適化は単位時間当たりの着陸本数を増やすことは有意義であるが、着陸本数を増やすに従い、航空機の着陸順を入れ替える必要があるため、管制官が航空機に対しての指示回数も増加する。多くの航空機が空港周辺に入域し、管制官の管制処理が許容量を著しく上回るような混雑が予測される状況においては、航空交通流管理（Air Traffic Flow Management:ATFM）業務内の交通流制御を実施する。この時に、大まかには二つの種類の制御があり、一方は航行前における出発時間の遅延、他方は航行中の混雑空域へ入域する時間の遅延である。交通流制御が実施される例として羽田空港南西部に位置する関東南 A セクタがある。この空域は主に羽田空港へ向けて航行する航空機が多く入域するエリアであり、この空域の混雑により福岡空港や新千歳空港から羽田空港に向かう航空機が出発遅延を行う。

以上のように離発着数を増やすことと同様に、混雑状況において管制官に負荷を軽減する支援をすることも重要である。管制官の作業不可は人間の感覚的な要素も含まれるため、航空管制官の管制負荷を客観的に数値化は困難であるが、安全で効率的な航空管制の実現に向けてシステム性能を適切に評価するような指標は今後求められる。このような指

標は 航空機の軌道やレーダの軌跡に基づいて推定する手法などが提案されている [19–21]. また, 管制官の管制処理を分析し, モデル化を行うことで, 管制官の負荷削減効果の客観的評価を行うアプローチもある [22–24]. このような手法は現時点でまだ, 実際の管制運用時に適用されてはならないが, 航空機が密集した状況において, 航空機同士がコンフリクトが発生する可能性がある状況はこの状況を改善するため多大な負荷がかかるため, これを支援するシステム有効である.

2.2 最適化と航空管制

2.2.1 メタヒューリスティックスと厳密解法

図 2.7 は最適化手法の比較を表している. ALP 及び空港周辺の合流問題は組み合わせ最適化問題に大まかには分類され, 組み合わせ最適化問題に対する求解のアプローチとして, 厳密解法とメタヒューリスティックスをがある. 厳密解法は問題の数理モデル化を行い, 解を得る. 数理モデル化によって解を求めた際には厳密解の保証が利点となり, 最適化処理が行われたならば, 最適な解を獲得することが可能である. ただ数理モデル化を行う過程において, 問題の前提を簡素化することが多く, 実問題として利用できないことがある. ALP においても, 厳密解法において, 空港の周辺における入域地点や周辺での航空機同士のコンフリクトが数理モデル化の際に消失されており, 着陸順として最適な順番の提示はできるが, 管制官にとっては順番のみでは運用が困難であるため, 実際の運用では活用できない. また, 解を得るまで多くの時間を費やす可能性がある. このような厳密解法の主要な方法論として, 分枝限定法 (Branch and Bound:BB) [41] や, 動的計画法 (DP) [42] やタブーサーチ [43] などがある.

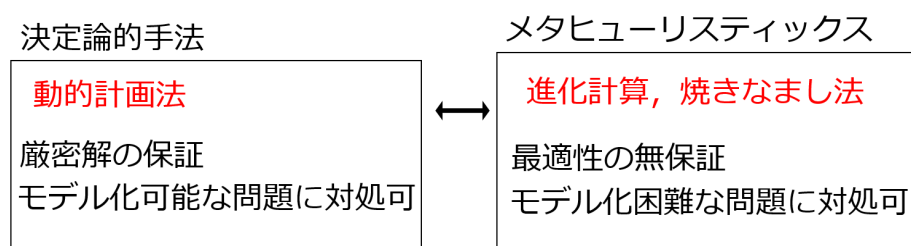


図 2.7 手法の比較

一方で, メタヒューリスティックの場合には, 評価値を得ることが出来れば, 解の探索が可能であるため, モデリングに柔軟さを含んでいる. 更に, 解を逐一更新するため, 更新の途中でも実行可能な解を取得することができる. しかし, 厳密解法とは異なり, 解の最適性の保証はない. また, 期待する解を得るためにはメタヒューリスティック内にあるパラメータ調整をする必要がある. メタヒューリスティックの手法としては進化計算手

法の一つである遺伝的アルゴリズム [44] や SA [39], 以上から, 本研究では管制運用の際に, 解の質的な観点より, 最適化の最中であっても解を保持している点と, 探索を行い際に, シミュレーションを用いて解を取得する観点でメタヒューリスティックを採用する.

2.2.2 経路最適化手法

経路最適化は航空機の問題に限らず, 車, ロボットなど様々な分野で取り組まれている [25, 26]. 車の問題ではあらかじめ道路の情報が与えられており, 最短経路となるよう, 交差点における分岐を選択することにより経路計画を行う問題がある. ロボット問題はあらかじめ与えられているマップから障害物等を避け, 最短経路を探索することを目的とし, 経路を探索するためグリッドマップ等の環境のモデリングを行う. ここで, 本研究は航空管制における経路生成法であるため, RTR [27] 等のリアルタイムで数秒の中で経路を逐一変化させる方法を想定しておらず, あらかじめ範囲の指定されているマップから経路生成する状況を想定する. これは, 航空機のパイロットが視認する範囲または, 航空機のアビオニクスで確認できる範囲で経路生成を行うのではなく, 航空管制官がレーダ画面を用いて着陸誘導を行うことを仮定しているからである. bezel curve を用いた手法として [28, 29] 等が航空機の軌道の最適化手法として提案されている.

安全で効率的な運航のため, 多くの研究者が経路最適化手法に取り組んでいる. 田島らは, 遺伝的アルゴリズムを用いた経路探索手法を提案し, グリッドマップベースの環境で航行に障害をきたす空域を避け, 目的地まで最短経路を生成する手法を提案した. [30]. 気象状況の変化に対応する経路生成法 [31], ガウス分布に基づくモデル予測とダイクストラ法を利用した経路生成法 [32], アントコロニーを用いた複数ロボットの経路生成法 [33], ホモトピー法を用いた経路計画法 [34], リアルタイムでコンフリクト予測に基づく軌道制御 [35], 複数機の経路制御法 [26], グリッドベースの経路最適化手法であり, 天候不順の空域を避け航空管制官が指示できるようベクタリング可能な経路を生成する手法である. 光の伝搬モデルを用いた Dougui らの手法は En-route での混雑時の環境において, 航空機同士のコンフリクトを解消する手法を提案し, その有効性を明らかにした [36]. また, BADA と呼ばれる航空機の物理モデルを用いて, 出発から到着まで燃料消費量を最小化する手法 [37] や, En-route から降下する経路の降下角度, 及び経路を考慮した着陸経路最適化手法が提案されている [38]. また, 空港周辺における SID 及び STAR の経路の集合を Simulated Annealing (以下: SA) [39] により生成し, シミュレーションにより最適な経路集合を獲得することを明らかにした [40].

2.2.3 GA と航空管制

航空管制の分野と GA は親和性が高く, 多くの手法が提案されている. 上記の表は従

提案者	年	問題	オペレーター の特徴	評価	Population Size	マシン スペック	世代数	処理時間	機数
Hansen	2004	ALP	有	無	-	-	-	-	20
Pinol and Beasley	2004	ALP	有	有	100	2GHz	5000	5850 sec	500
Cheng	1999	ALP	有	有	1000	-	6000	-	12
Ciesielski	1998	ALP	有	有	50	-	50000	30 sec	29
Hu and Chen	2005	ALP	有	無	70	300MHz	500	2-5 sec	30
Zuniga et al.	2011	MFO	有	有	200	3.0Ghz	200	-	50
Chida et al.	2016	MFO	有	有	100	2.4GHz	500	-	50
Ahmed et al.	2017	MFO	有	無	50	-	400	-	20
Xue and Zelinski	2013	MFO	無	有	800	2.66GHz	300	30 sec	15

来手法において GA を適用した手法に関してまとめている。左の列から、提案者、発表年数、適用される問題、GA における特徴的なオペレーションの有無、特徴的な評価例、問題に適用する母集団サイズ (Population Size)、最適化で使用するマシンスペック、収束判定に用いる世代数、実行時間、取り扱い機数をそれぞれ表す。オペレーションは選択、突然変異、交叉のなかで、ALP、MFO の最適化で用いられる知識を用いる。評価に関しても同様であり、問題の知識を評価関数に考慮するか表している。例えば、ALP において、突然変異を適用する場合には乱数により選択された航空機の順位を変更するが、このとき、その時点で変更の対象となっている順番で着陸する航空機の順番と順位の交換を行う。順番の最適化のため、このような突然変異がオペレーターとして適用される。

GA をベースとして提案されている手法の Population サイズは 50 個体から 1000 個体と幅があり、世代数においても同様であり、最大で 50000 世代と違いがあるが、近年の手法においては、500 世代程度で結果を出力する傾向にある。また、Hu らの手法においても交叉のオペレーションを改良することで通常より早い世代数で高い評価値の着陸順を獲得することが可能であることを明らかにした [45]。この手法は ALP における着陸順に関する改善に着目しており、遺伝子構造が着陸順をベースとした構造を保持しているため、本研究で適用することは難しい。また、本研究では過密状態に着目した航空機の合流及び着陸順最適化を行うため、実行不可能解に対するアプローチが必要になる。

GA において、実行不可能解が発生した際に、デスペナルティ法と呼ばれる実行不可能解となった場合に解の序列を最も低くする方法 [46] や評価に対し、どれくらい制約を違反しているか評価を割り引くことで実行不可能解を考慮する手法が提案されている [47]。ただし、このように評価に違反量を加えることは、違反量との調整が必要となるため、高濱らはこれを適応的に制御する手法を提案しこの有効性を明らかにした [48]。これらの手法は全体としての評価に対して実行不可能解に関する項目を考慮する手法であるため、GA におけるどの遺伝子が悪影響を及ぼしているのかを考慮していない。評価時に解が実行不可能解となった場合に解を修復する手法が存在する [49-51]。

また、田中らは実行可能解に対して違反量を個体に反映させるのではなく、違反量に基

づいて個体の解修復を施す手法を提案している [52]. 本研究でも実行不可能な解を生成した際には解修復を施すことで早い世代で実行可能な解集団を獲得する手法を提案する.

2.3 Aircraft Landing Problem

Aircraft Landing Problem(以下: ALP) は [6] に従来の取り組みが記載されているが, 航空機における滑走路の利用効率を高める問題とされており, 航空機の着陸時間, 着陸順の最適化を行う問題である. 最適化する際, 航空機の機種, 初期の着陸時間が与えられる. 初期の着陸時間は空港の STAR に基づき算出されるが, STAR における航空機同士のコンフリクトは考慮されておらず, 滑走路の間隔のみのコンフリクトを解消する. ここで, 航空機同士のとるべき間隔は航空機のサイズ (機種) に依存しており, 前方の航空機が大きく, 後方の航空機が小さい場合には大きな管制間隔が要求され, 前方の航空機が小さく, 後方の航空機が大きい場合には小さな管制間隔が求められるアシンメトリの関係があるため, 最適な着陸順を探索することは難しい.

2.3.1 ALP の問題定義

問題のモデル化により, 定義は異なるが, 各航空機毎に航空機のサイズ, 予定到着時刻, 最短到着時刻, 最長到着時間が前提として与えられる.

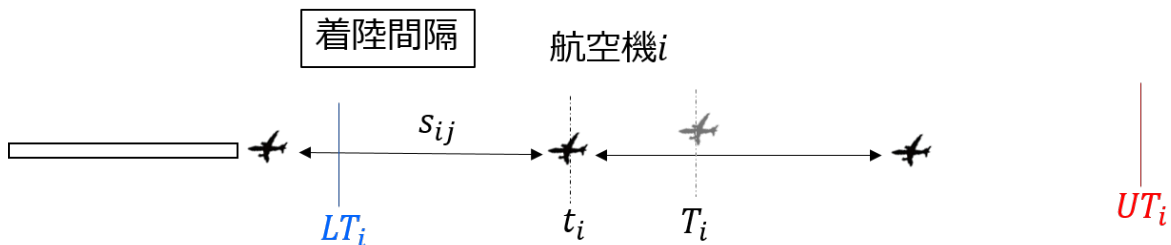


図 2.8 ALP の定式化

$$LT_i \leq t_i \leq UT_i \quad (2.1)$$

$$t_i + s_{ij} \leq t_j \quad \text{or} \quad t_j + s_{ji} \leq t_i \quad (2.2)$$

航空機の予定到着時刻の最短, 最長到着時刻は式 (2.1) で表され, 添え字の i は ID を表し, j は i 以外の航空機を表す. LT は最短到着時刻であり, UT は最長到着時刻を表す. t はこの時刻の間に到着することを制約とする.

式 (2.2) は制約として与えられる航空機の時間間隔を表す. 左の不等式は航空機 i が航空機 j よりも早く着陸する場合の不等式関係であり, 右の不等式は航空機 i が航空機 j よ

りも遅く着陸する場合の不等式関係をそれぞれ表す。 s_{ij}, s_{ji} はそれぞれの航空機の大きさの関係に合わせて与えられる時間間隔であり、大きさの関係によって異なる値を持つ。

図 2.8 は ALP の定式化の例を表しており、左に滑走路が存在し、これに沿って到着機が列を形成している。各航空機には最短到着時間と最長到着時間が設定されており、この範囲内で到着時間を管理し、多くの着陸を誘導するような着陸順を生成する。

表 2.3 航空機のサイズによる管制間隔 (再掲)

		Leading Aircraft			
		S. Heavy	Heavy	Midium	Light
Trailing Aircraft	S. Heavy	3	3	3	3
	Heavy	5	4	3	3
	Midium	6	5	3	3
	Light	8	6	4	3

航空機のサイズは上記の表の通り、4つに分類される。表の中は距離間隔であり、この距離を時間に変換し、 s として与えられる。

ALP の目的

航空管制の目的で ALP に関係する項目として、航空交通の安全性の向上、効率的な航空交通流の形成、航空交通容量の拡大、パイロット・管制官のワークロードの軽減が挙げられる。具体的には、航空管制に対しての目的としては安全で効率的な運用に向け、以下のような最大化・最小化が挙げられる。

- 最大化：滑走路容量，航空機に対するサービス受益に対する平等性
- 最小化：着陸前航行時間，航空管制官のワークロード，誘導路の移動時間，到着予定時刻の逸脱量，出発機数と到着機数のバランス

これらの中で従来手法で用いられている目的としては（1）到着予定時刻の平均遅延時間の最小化，（2）全航空機の着陸機に対する航空機間の間隔の総和の最小化の二つが挙げられる。ここで、遅延の定義は FCFS（航空機の入域時刻順）を基準にしており、この着陸順に伴う各航空機の予想到着時刻と最適化により獲得した各航空機の目的空港の着陸時刻との差分が基準となる。

2.3.2 従来手法

ALP において問題を解く際には動的，静的問題に大別される．航空機の前提によって問題の難しさが異なる．この前提は航空機を認識できる時間を表す．静的環境は最適化の対象となる航空機の着陸時間と機種をあらかじめ把握可能である状況を指し，その環境での一度のみ最適化を行う状況を言う．一方で，動的環境は最適化を行うその時間から，ある時間の範囲までを繰り返し最適化を行う環境を指す．

図 2.9 は最適化に関するタイミングチャートを示しており横軸は時間経過を示す．静的環境は time horizon が対象となる航空機のはじめから最後までを 1 回の最適化でおこなう．動的の場合には，計算機が最適化を行う際の現在時刻である Current Time からある一定時間の範囲内での最適化を行う．

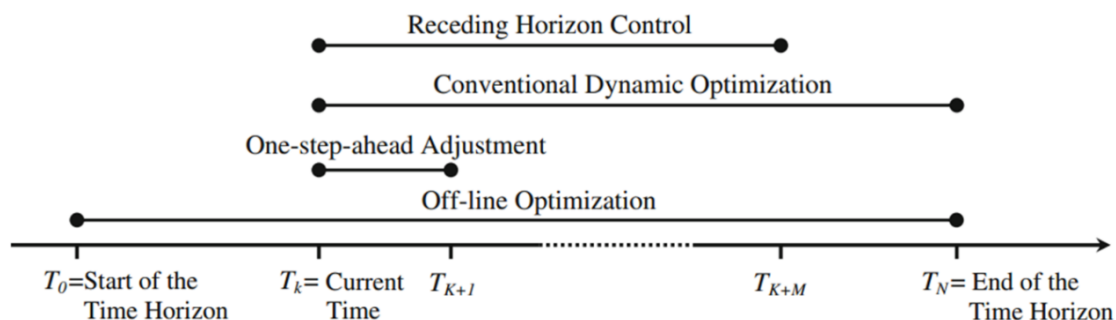


図 2.9 最適化における時間窓とステップの関係

従来手法としてのアプローチはメタヒューリスティックと厳密解法と大まかに二つのアプローチからそれぞれ多く提案されている．厳密解法におけるアプローチでは，Besley らは混合整数計画問題に定式化を行い，8つの例題にて定式化の有効性の検証を行った．この中で4つの滑走路を持ち50機の着陸順の最適化が実運用時間内で可能であることを明らかにした [53]．またメタヒューリスティックを利用した手法として Genetic Algorithm (以下：GA) を利用した Bianco らの手法 [54] や Ant Colony Optimization [55] を利用した手法や，Particle Swarm Optimization [56] を用いた手法もある [57–59]．ALP における複数の滑走路の割り当て問題においては，pinol らが，ヒューリスティック手法を用い，5つの滑走路における滑走路の割り当てが可能であることを明らかにした [60]．このように ALP において滑走路の割り当て数が多い場合，最適化対象の航空機が多い場合にはヒューリスティックを用いた手法が有用である．

GA のほかにもヒューリスティックの方法論として，Simulated annealing を用いた手法や，ACO を用いた手法がある [57, 58, 61]．整数計画問題として定式化した Fahle らは Simulated annealing を用いて着陸順を最適化することに成功した．航空機数が増加する

提案者	年	手法の分類	アプローチ	目的
Psaraftis [63]	2012	厳密解法	DP	遅延量 最終航空機の到着時間
Beasley [53]	2000	厳密解法	MILP	逸脱量
Jungai and Xu [58]	2012	メタヒューリスティックス	SA	遅延時間
Grish [59]	2016	メタヒューリスティックス	PSO	逸脱量
Hansen [64]	2004	メタヒューリスティックス	GA	遅延時間
Pinol and Beasley [60]	2004	メタヒューリスティックス	GA	逸脱量
Cheng [65]	1999	メタヒューリスティックス	GA	遅延時間
Ciesielski [66, 67]	1998	メタヒューリスティックス	GA	逸脱量
Hu and Chen [62]	2005	メタヒューリスティックス	GA	逸脱量
Fahle [61]	2003	厳密解法 メタヒューリスティック	SA SAT MIP HC	逸脱量
Balakrishnan and Chandran [68]	2006	厳密解法	DP	最大遅延時間 着陸間隔
Furini [69]	2014	厳密解法	DP	滑走路占有時間
松枝 高橋 [70]	2018	厳密解法	DP	逸脱量
Yu et al. [71]	2011	厳密解法 メタヒューリスティック	CA GA	逸脱量

と探索空間の増加により実時間内で最適化が困難である。Hu らはこの問題において、制御の分野で利用されている、Receding Horizon Control を用いた GA に基づく最適化手法を提案した [62]。

図 2.9 に表わされるように、最適化を行う現時点から一定の窓間隔を用意し、この窓内に入域する航空機の着陸順の最適化を行う。窓内の最適化が完了した後、開始時点のスライドし、最適化を行う範囲まで繰り返す。この時、最適化の窓は1時点後にシフトした際には、最適化対象の時間がオーバーラップをさせることで、最適化の実行時間を短縮可能であることを明らかにした。

上記に ALP における従来手法を表にまとめた。左の列から手法の提案者、論文の発表年、手法の分類、問題におけるアプローチ方法、最適化の目的をそれぞれ表している。手法の分類は厳密解法とメタヒューリスティックスで分類し、併記されている手法は両方ともに使用している手法を表す。アプローチ方法にはそれぞれ動的計画を表す Dynamic Programming (DP)、混合整数線形問題を表す Mixed Integer Linear Programming (MILP)、焼きなまし法を表す Simulated Annealing (SA)、粒子群最適化を表す Particle Swarm Optimization (PSO)、遺伝的アルゴリズムを表す Genetic Algorithm (GA)、充足可能性問題を表す Satisfiability problem (SAT)、混合整数計画問題を表す Mixed Integer Problem (MIP)、山登り法である Hill Climbing algorithm (HC) とそれぞれ略字で表す。

目的は、各航空機の予定到着時刻からの遅延の総和を表す遅延量、スケジューリング対

象の航空機の中で一番最後に到着する航空機の着陸時間の最小化目的とする最終航空機の到着時間，全航空機における予定到着時刻からの時間差を表す逸脱量，各航空機における前後の航空機の時間間隔の総和を表す着陸間隔，スケジューリング対象の航空機において，一番はじめに着陸する航空機と一番最後に着陸する航空機の時間差を表す滑走路占有時間の5つの目的が主な最適化対象となる．分類としては2種類だが，アプローチとしては多岐にわたるが，厳密解法においてはDPを用いた求解，メタヒューリスティクスにおいてはGAによるアプローチが多いことがわかる．

2.4 ALP の拡張 (Merging Flow Optimization)

2.4.1 MFO の問題定義と目的

Merging Flow Optimization は ALP を拡張した最適化問題であり，問題がまだ体系的にまとめられておらず，比較的新しい研究分野であり，共通される項目としては，最適化の対象は ALP と同様に着陸順の最適化であり，これに制約とし En-route から空港の滑走路へと結ばれる標準到着経路 (STAR) 上におけるコンフリクトも対象とする．先述したように，航空路は車の道路のよう航行する空域に張り巡らされており，目的空港へ着陸するにあたり，多くの航空路から標準着陸経路へ入域する．特に混雑空港に対しては航空機同士のコンフリクトの可能性が高まるため，経路上におけるコンフリクトを考えることは重要である．制約条件としては ALP よりも厳しくなるため，着陸時間を決定するための制御変数である入域速度が決定変数として追加される場合が多い．

下記に手法毎の特徴をまとめているが，最適化の観点以外でも航空機の合流を適切な間隔を満たしながら着陸まで誘導することは重要である．その際に，着陸経路をそのまま使用した場合には管制官がベクタリングを行った場合よりも混雑時のコンフリクトの対応力は低い．そのため，Boursier らはポイントマージと呼ばれる，混雑環境下におけるコンフリクトを解消できるような経路を提案した [72]．ヒューマンインザループシミュレーションにより，通常管制官が用いるベクタリングよりも迂回量及び燃料消費量を抑制できることを明らかにした．

また，蔭山らは AirTop と呼ばれるシミュレータを用いて，羽田空港において，ポイントマージと呼ばれる扇形状の経路構造が到着機の合流の有用性を検証した [73,74]．図 2.4.1 はポイントマージの例を表しており，上部の矢印から航空機は進入する．入域した航空機は他の航空機とのコンフリクトがない場合はそのまま丸マークが示す合流地点へ直進するが，コンフリクトの生じる間隔未満の場合には間隔を生むために扇型の湾曲部を航行する．このようにあらかじめ迂回経路を用意し，間隔付けのための経路構造を設定することで現在の管制の運用で標準的に用いるベクタリングに比べ管制負担を軽減可能であることが明らかにした．

また、実用化に向け、米国の NASA では、滑走路への最終進入間隔を航空機の変更にすることで設定するシミュレーション実験を行い、Savis らにより有効性が明らかになり、実際にこの機能を有する、CTAS (Center TRACON Automation System) に利用されている [75,76]. その他にも滑走路容量向上を導く経路構造 [77], TMA における航空機の合流の自動化法 [78], 不確かさを取り入れたマージ手法 [79–81] や、出発機、到着機の TMA エリア及び飛行場面のコンフリクトを解消するなど統合的な手法も提案されている [82].



図 2.10 迂回経路構造一例 : Point Merge

Hong らは MFO における目的関数を以下と定義している [83].

$$J = \sum_{F=1}^{N_F} \sum_{R=1}^{N_F} \{A_{f,r} T_{f,r,pF} + A_{f,r} (T_{f,r,pF} - T_{f,r,pr})\} \quad (2.3)$$

式 (2.3) が示すのは全航空機に対する入域時間と遅延時間の総和を示している. 式中の N_F, N_R はそれぞれ, 航空機と経路の総数を示し, $A_{f,r}$ は航空機 f が経路 r を通過する場合に 1 を取り, それ以外には 0 を取る. pF は最終進入地点を表しており, $T_{f,r,pF}$ 空港周辺に入域地点から最終進入地点に到着する時間である. $T_{f,r,pr}$ は空港周辺に入域地点から他の航空機の影響がなく, 遅延せずに最終進入地点まで到着した場合の時間である. 第 1 項で巡航時間の評価, 第 2 項で遅延量に関する評価を表す. 上記の評価式は, Hong らによる評価式であるが, 他の研究においても重み付けなどの差異はあるが, 評価としては, 遅延量と巡航時間が評価対象となる.

この他にも Liang らは FCFS と比較した時の着陸順から離れすぎないように評価を加

える。[84]。虎谷らは空港周辺へ入域し、最終進入地点までにおける全航空機の燃料最小化を目的とした [85]。この燃料算出のため BADA を用いて算出を行う [86]。

2.4.2 制約

ALP における制約は、航行時間に関する制約（最短航行時間，最長航行時間）と滑走路における管制間隔の 2 つが制約として与えられるが，MFO においては，滑走路における間隔に加え，経路間における間隔の制約が加わる。これにより，ALP では最終進入地点における到着時間で制約を満たすか否かを判断していたが，経路上での判断が必要となるため，制約を充足するか確認するための計算量が増加する。このように経路上のコンフリクトの算出のため，ALP とは異なるモデリングが行われる。算出方法は経路の仮定を自由とするか固定とするかでアプローチ方法論が異なる。自由とする場合には，虎谷らは時間と 2 次元空間を 3 次元で表現し，航空機の軌道の交叉をこの空間により算出した。

固定経路において多くの場合，リンク・ノードモデルを採用し，ノードを Waypoint，リンクを経路としてこのグラフ上を航行することを仮定とする。リンク・ノードモデルにおいてはリンク，ノードの点毎にコンフリクトを調べることでコンフリクト算出が可能であるため，多くの手法でこの算出方法が採用されている。他のコンフリクト算出方法として，航行時間毎に位置情報を記録し，位置情報から距離に基づいて判定する方法がある。

2.4.3 MFO における従来研究

MFO における従来研究の位置づけを以下の表にまとめた。表は固定経路における従来研究をまとめた。左の列から，提案者，論文を公表した年，提案手法の最適化方法の分類，具体的な最適化手法あるいはアプローチ，最適化対象の目的，航空路からの航空機の合流点の数，航空機における制御できる要素をそれぞれ示している。手法の分類においては，厳密解法とメタヒューリスティックス (MH) の 2 つに分類した。アプローチとしては，手法の分類における詳細を記載した。BB は blanch and bound, GA は Genetic Algorithm, MILP は Mixed Linear Interger Program をそれぞれ意味する。目的の詳細としては，遅延量，燃料消費量，コンフリクト数，滑走路占有時間，航続時間お 5 つがある。遅延量は予定していた最適化前における到着予想時刻からの遅れを表し，全航空機の合計の遅延量を最小化することを目的とする。燃料消費量は航空機の経路長ではなく，航空機の物理モデルにより算出した際の消費量において全航空機の燃料消費量を最小化することを目的とする。コンフリクト数は航空機同士の相対的な位置状態からコンフリクトを判定する。このコンフリクトを最小化する（コンフリクトができるだけない状態にする）ことを目的とする。滑走路占有時間においては，はじめに着陸する航空機の到着時間と最後に到着する航空機の差分により決定される時間間隔であり，この間隔を最小化すること

を目的とする。航続時間は航空機が進入地点から最終進入地点まで航行時間であり，全航空機の時間の総和を最小化することを目的とする。主にこの5つが目的関数を構成する要素となる。

提案者	年	手法の分類	アプローチ	目的	合流点	制御変数
Samà et al. [79]	2013	厳密解法	BB	遅延量	複数	Waypoint 毎の入域時間
Toratani et al. [85]	2015	厳密解法	MILP	燃料消費量	単点	速度，経路
Zuniga et al. [77,87]	2011	MH	GA	遅延量，コンフリクト数	複数	速度，経路，入域時間
Chida et al. [88]	2016	MH	GA	遅延量，コンフリクト数	複数	速度，経路，入域時間
Ahmed et al. [89]	2017	MH	GA	滑走路占有時間，航続時間	複数	速度，経路
Hong et al. [80]	2017	厳密解法	MILP	航続時間，遅延量	単点	速度，経路，入域時間
Xue and Zelinski [90]	2013	MH	GA	航続時間	複数	速度，経路，入域時間

2.5 管制運用と自由・固定経路

本研究では自由経路と固定経路という2つの異なる経路の運用方法から適切な交通流の生成を目指す。以下に自由経路と固定経路の前提条件を説明する。

図 2.11 は航空管制官が監視するレーダ画面を表している。黄色と緑の文字で表示されているシンボルが航空機を表し，航空機には航空機の識別番号，速度，高度情報を表示している。レーダ画面には複数の直線が表示され，航空路を表す。平時では航空機は航空路に沿って巡航する。混雑時には，航空機同士の間隔が密になるため，ベクタリングと呼ばれる指示を管制官は行う。ベクタリングは機首方向，高度，速度の指示を行い，適切な管制間隔の維持のため利用する。空港周辺における到着機は目的空港へ向けて降下を始めるため，高度情報は用いない。ここで，自由経路とは，航空路からベクタリングにより逸脱した経路のことを指す。

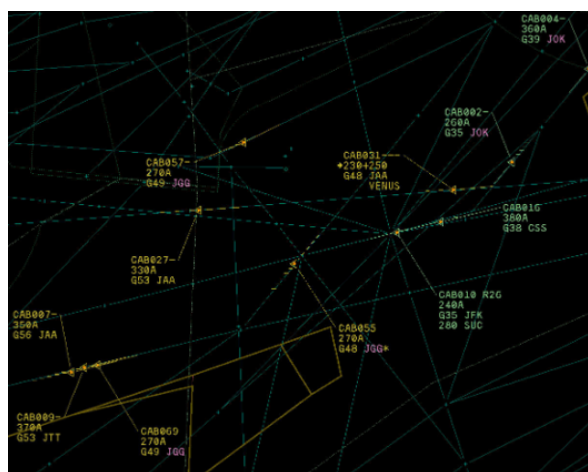


図 2.11 自由経路における前提

図 2.12 は空港周辺における標準到着経路を表す。空港周辺の経路は航空路とは異なり、高度差を用いて複数の高度帯を用いて同位置での航空機の巡航は許容せず、ある位置には必ず1つの航空機が巡航する。そのため、高度、最低巡航速度や最高巡航速度が記載される。また、複数の航空路からの合流点も設定され、図 2.12 においては、MOBRO が合流点に当たる。

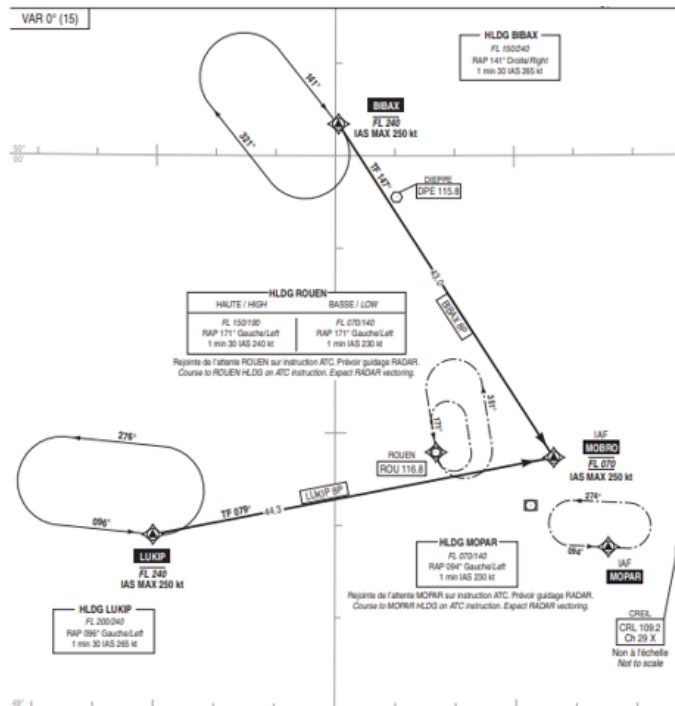


図 2.12 固定経路における前提

管制運用においては、交通流の状況により管制官の運用は異なる。

図 2.13 は経路の違いによる着陸容量と管制負荷の関係を示している。縦軸の着陸容量は単位時間あたりの航空機の着陸数、横軸は管制官負荷量を表す。横軸は右に行くほど負荷が小さくなり、縦軸は上に行くほど容量が大きくなる。グラフ中に2つの経路の関係が示されているが、左上に示されている経路が自由経路であり、4つ四角の枠にはある時間において入域した航空機が誘導される経路がそれぞれ示されている。

管制負荷のを定量的に表すことは難しいが、同一空域内におい同様の条件で航空機が入域した場合には固定経路を利用したほうが管制負荷が自由経路と比較した際には低いと言える。自由経路においては、各航空機の方向変換点である Waypoint は一意に定まっておらず、Waypoint があるだけ指示の必要がある。一方で固定経路の場合だと、予め経路同士の合流点や Waypoint が既知であるため、管制官は航空機に対して指示する回数が自由経路と比較した場合に比べ相対的に減少する。ただし、着陸への誘導の観点で比較した場合、通常、管制官は目的空港へ着陸する航空機を固定経路に沿って滑走路へと誘導する

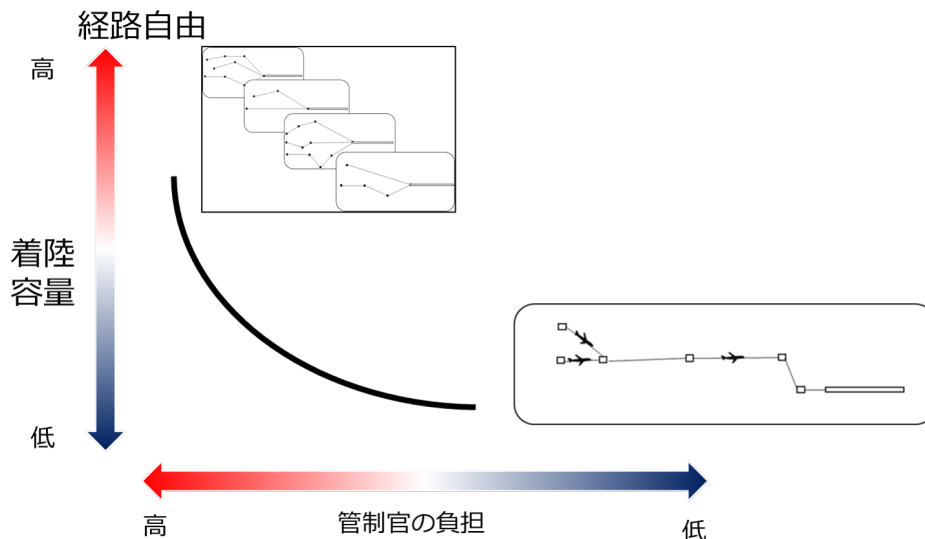


図 2.13 経路と着陸容量の関係

が、多く航空機が航空路から入域する状況においては固定経路ではなく管制官がレーダーベクタで航空機を滑走路へと誘導する。これは、固定経路のみを用いた場合においては、自由経路に比べ航空機間の間隔の制御の柔軟性が低いため、混雑時には利用することが困難である。

2.6 研究の位置づけ

図 2.14 は自由・固定経路を含めた場合の本論文の位置づけを示す。

ALP においては具体的な経路を仮定せず、滑走路に着陸する順番の最適化を行っており、航空管制にとって、どのような制御により順番を入れ替えるか示さない限りにおいては着陸順のみの提示のみでは管制官の支援に足りない。そこで、本研究では、自由経路及び固定経路を仮定した二つのアプローチにより着陸順最適化手法の構築を目指す。自由経路とは通常目的空港の着陸の際に用いる標準到着方式で定められている経路を用いず、航空管制官のベクタリングにより航空機の制御を行う方式である。一方で固定経路の場合は標準到着方式で定められる到着経路に基本的には航行することを仮定する。二つのアプローチを用いる理由としては、混雑状況において、固定・自由経路を用いた場合において着陸順の最適性と航空管制官へのワークロードの観点でトレードオフ関係となるからである。実際の航空管制においては混雑した時間帯（8：00頃や20：00頃）は標準到着方式で設定される経路のみを用いて適切な管制官を維持することは難しい。

表 2.14 の自由、固定経路は4種類にそれぞれ分割されており、初めに自由経路に関して説明する。表内は縦列が航空機の制御、横軸は最適化を軸としている。縦軸の航空機の

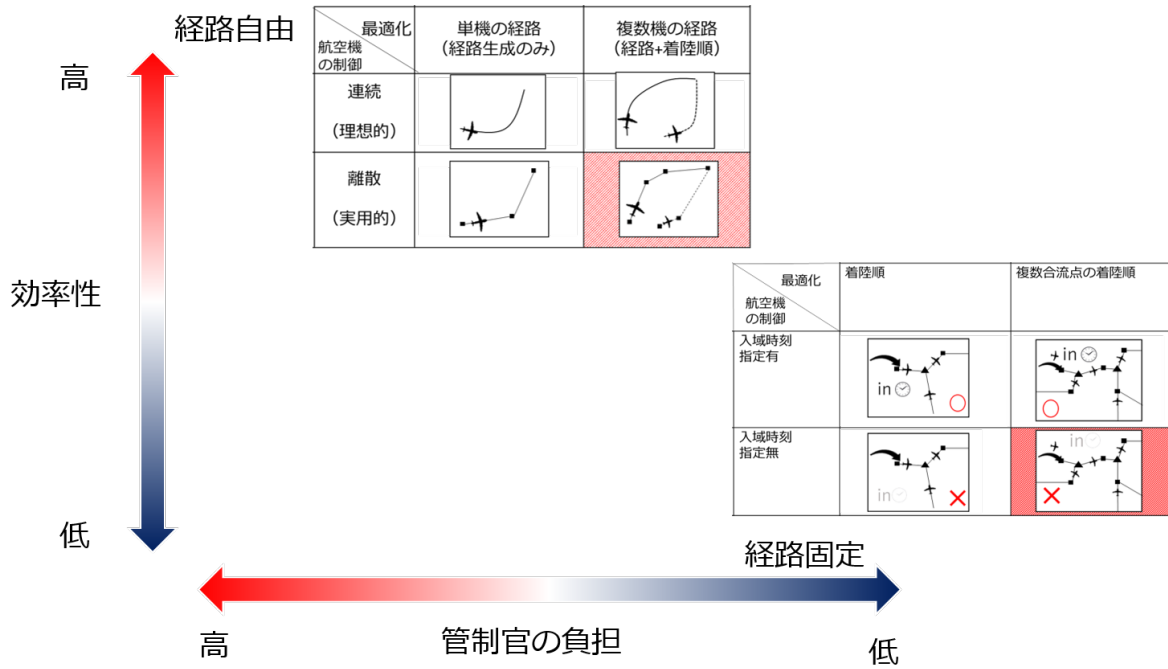


図 2.14 本研究の全体の位置づけ

経路は例であり，経路が滑らかに曲線を描くものと離散的に経路が結ばれるのかを図として表したものである．航空機が経路を航行する際に制御が連続的，離散的かが具体的に異なる点である．横軸のは最適化対象の航空機数で分類しており，1機であるか，複数機であるかを基準とした．最適化の対象は手法によって異なるが，複数の航空機を想定した着陸が考慮されているかを具体的な基準として定めた．

2.6.1 自由経路における位置づけ

自由経路における経路・着陸順の最適化は，大まかに4つに分類される．表 2.4 は自由経路における本論文での位置づけを示している．列は航空機の機数で分類し，行は管制官の運用の可否により大別する．ここで運用の可否について説明する．管制官は航空機をベクタリングによって制御するが，空港周辺においては，方向を変更する Waypoint を指定することや，方位あるいは航空機を速度を制御を行う．そのため，Waypoint を設定しない，連続的制御 (管制官が絶え間なく指示することが要求される)，あるいは降下による間隔制御は困難である．航空機の1機に対しては管制官のベクタリングを許容する経路生成手法は存在するが，1機のみの方論であり，着陸順及び着陸間隔を生成する手法としては不十分である．Cardenas らは Dubins パスを用いた経路計画手法を提案し有効性の検証を行った [91]．この手法は航空機の旋回はある程度の半径を持った弧が必要であり，Dubins パスは，旋回の際の弧を考慮した経路生成法である．この手法を UAV(Unmanned

Aerial Vehicle) において UAV の現在位置から目的地へ実行可能な経路を生成する手法である。

表 2.4 自由経路における研究の位置づけ

管制官	航空機数	単	複
	最適化 <small>航空機の制御</small>	単機の経路 (経路生成のみ)	複数機の経路 (経路+着陸順)
運用不可	連続 (理想的)	[Miyazawa et al.,2013] [Cárdenas et al., 2014]	[Dougui et al., 2013] [Mori et al., 2015] [Toratani et al., 2015]
運用可	離散 (実用的)	[Takeichi et al.,2018] [田島ら,1995] [Dey et al., 2014]	

図 2.15 は miyazawa らの動的計画法に基づく燃料消費量最小化を表す。航空機は物理モデルを仮定して、高度、速度を決定変数として燃料消費量最小化する。この時に、経路複数に細分化し、細分化された経路を初期位置から、順次決定する。図の左側は最適化により決定した高度と速度を表し、点一つ一つがある経路上での速高度、速度情報を表す。このような決定する点が多くなるほど、航空管制官の指示量が多くなり、これが複数機の場合には指示量が膨大となるため、この手法での経路・着陸順最適化は困難である。

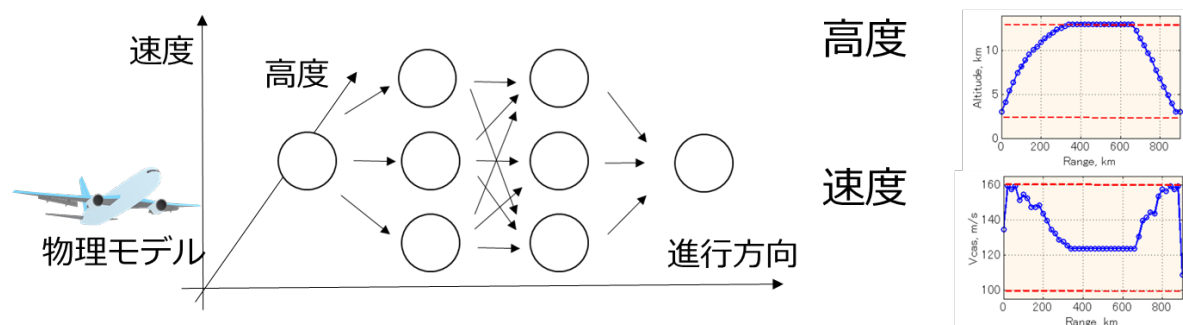


図 2.15 動的計画法に基づく燃料消費量最小化

参考文献：[37]

離散的な経路生成法においては、Takeichi らの手法、田島らの手法、Day らの手法など

がある。Takeichi らの手法はある地点から目的空港の滑走路までの基本の経路は定まっております。この経路において、ショートカットと迂回を用いた経路の変化と、降下地点の変更により到着時間を柔軟に変更が可能である。

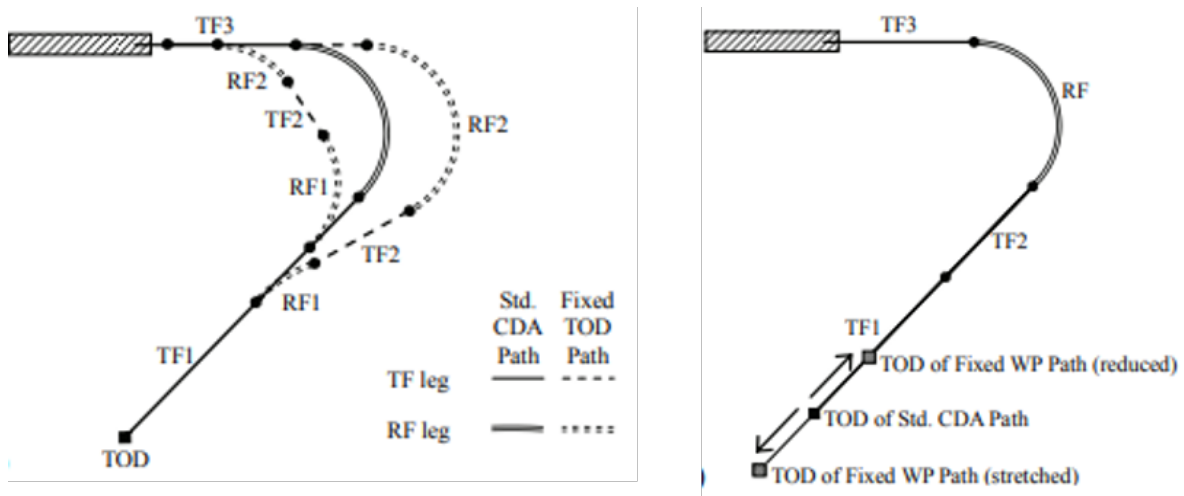


図 2.16 降下点から滑走路への到着時間を指定した経路計画法

参考文献：[38]

図 2.16 は降下開始地点から、滑走路までの経路計画の例であり、左側の図では基本経路とショートカットの経路、と迂回経路、右側の経路は基本経路での降下地点の変更点を表して。降下地点を後方側にすると、到着時間は遅くなり、前方にすると到着時間は早くなる。ただし、1つの経路を用いる前に航空機の合流を行う必要があり、設定している経路より前での調整が混雑時には特に重要となるため複数の航空機への適用は困難である。Dey らの手法は、現在位置から、目的地点までの最短経路を探索する手法であり、風向きを考慮した上で、目的地まで最短時間で到着するような経路をガウス課程を用いて生成する。

図 2.17 は Dey らの手法の有効性を検証するために用いたマップを表す。マップの左上のシンボルがゴール（目的地）、右下がスタート地点（現在位置）を示す。緑色のシンボルが天候情報の取得地点を表し、このシンボルの周辺の天候に関しては詳細のデータを取得することが可能である。黄色のシンボルが Waypoint に対応し、周囲の 8 近傍に遷移が可能である。この情報に基づいて最短時間となるような経路を生成する。



図 2.17 風を考慮した最短時間での経路生成法

参考文献：[32]

Dougui らは、航空機同士のコンフリクトや、悪天候の空域を避ける光の伝搬モデルを用いた経路生成法を提案した。一定距離毎に通過すべき Waypoint を定め、最終的には Waypoint を結び合わせた線が経路となる。実験結果により 30 秒以下で航空機同士のコンフリクトのない軌道を獲得可能なことを明らかにした。

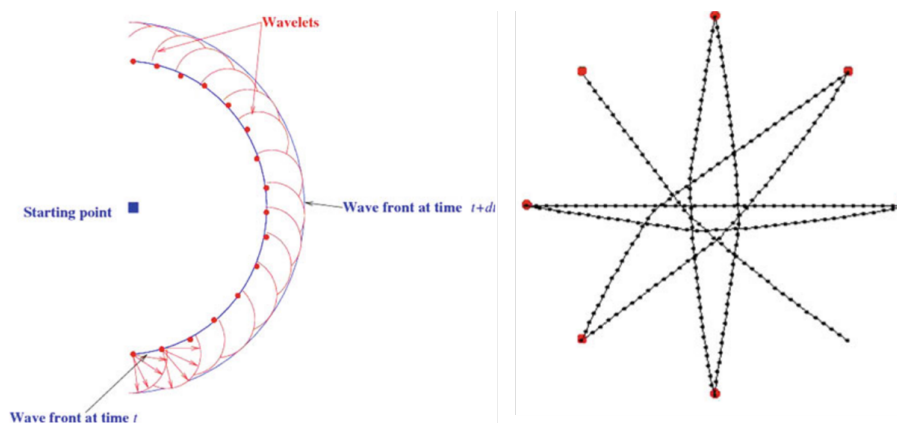


図 2.18 光の伝搬モデルに基づく経路生成法

参考文献：[36]

図 2.18 は光の伝搬モデルに基づく経路生成の例を示しており、左側は経路生成法の例を示し、右側は複数航空機の経路生成を表している。経路生成においては、円上に

Waypoint の候補が生成され、決定した Waypoint はまた、円の中心の基準となり、これを繰り返すことで1つの経路を生成する。右の図では赤が航空機の現在位置を示しており、中心に向かう方向で巡航する。これが、そのまま航行すると、コンフリクト状態となるため、全航空機のコンフリクトを解消するよう複数の経路を決定する。この時、コンフリクトのない、複数の経路生成が可能であるが、航空路におけるコンフリクトを解消する経路であるため、着陸順が考慮されておらず、混雑時の着陸容量を低下する可能性がある。

Mori らは航空機の降下時における燃料消費量の最小化を導く複数航空機の合流及び着陸順最適化手法を提案した [92]。航空機の、降下角を制御し、合流中のコンフリクトを解消し、かつ燃料消費量が最小となる経路生成を物理モデルにより行った。

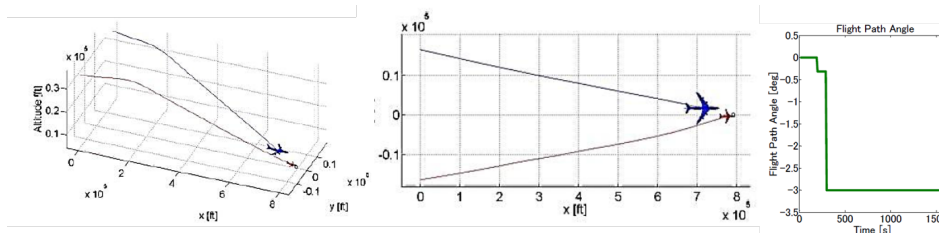


図 2.19 Mori らによる航空機合流法

参考文献：[92]

図 2.19 は 2 機による軌道生成の例を示しており、左の図は、高度と水平面での経路生成を表し、中央の図は、水平面上での経路を示す。右の図は、航空機の機首の角度を表している。2 機の航空機の合流を行うことが本手法では可能であるが、生成された軌道が連続的であるため、航空管制官によって指示が困難な軌道である。

複数機の経路生成に着目した場合には、ホモトピー法 [93] により、連続的に経路を生成し、混合整数計画問題として問題の定式化を行い最適化を行う手法が存在するが、管制官にとって指示が困難という問題がある。本研究は管制運用が可能であり、複数機の経路及び着陸順を生成する方法を構築する。

2.6.2 固定経路における位置づけ

固定経路における着陸順最適化も大まかに 4 つに分類を行った。こちらも縦軸は自由経路と同じく管制官に対する運用の可否を基準とし、具体的な基準は入域時刻の指定とした。横軸は単、複数の滑走路運用を基準とし、具体的な基準としては複数地点の合流が可能か否かを基準とした。表の中の詳細について説明する。初めに表の左上の 1 つの合流点による入域時刻の指定がある場合の手法については、Xue らは、1 つの入域地点での合流における全航空機の航行時間の最小化を最適化の対象とし、入域時刻の変更を許容する仮定を置き、最適化を実現した。

表 2.5 固定経路における研究の位置づけ

管制官	滑走路数	単	→	複
	最適化 航空機 の制御	着陸順		複数合流点の着陸順
運用 不可 ↓	入域時刻 指定有	[Xue et al., 2016] [Ahmed et al., 2017]		[Zuniga et al., 2011] [Liang et al., 2017] [Hong et al., 2017]
運用 可	入域時刻 指定無	[Kageyama et al., 2015] [Sama et al., 2013]		

図 2.20 はロサンゼルス空港における STAR を示しており，到着機は WayPoint の FIM から CORTY へと向かい滑走路へと着陸する．迂回経路は FIM から SADDE へと向かうものと，SMO から異なる旋回点を選択することで迂回を行うことのできる迂回経路（トロンボーン）のふたつが存在する．Xue らの適用した例題は到着便の経路と出発便の経路が交差した環境となっており，出発便のコンフリクトも対象となっている．最適においては到着便の最適化領域への入域時間も決定変数の範囲内となる．



図 2.20 固定経路における迂回経路を利用した手法

参考文献：[90]

Kageyama らは比較的空港に近い周辺での航行時間最小化を最適化の目的とした上で

羽田空港でのシミュレーション実験により手法の有効性を明らかにした。最適化の対象とする入域地点までは複数の合流点があるが、それらから航空機の入域の合流処理が完了していることが仮定されている。最後に Zuniga らは複数地点からの航空機の合流を実際の航空交通量を仮定した上で航空機の航行時間の最小化を目的とし、スペインのグラン・カナリア空港のモデル化及び最適化の対象の例題として手法の有効性を検証した。

以上のことから本章をまとめると、自由経路においては、複数の航空機における経路生成とその着陸順生成のため、経路に関しては航空管制官が運用 (ベクタリング) 可能な経路において複数の航空機の管制間隔が維持された着陸順の生成が可能な手法を提案する。固定経路においては、入域時刻の指定が無く、複数の入域地点からの航空機が入域する環境下での着陸順生成手法を提案する。

第 3 章

進化計算と機械学習

3.1 単目的遺伝的アルゴリズム

遺伝的アルゴリズム (Genetic Algorithm: GA) [44] は代表的な進化的アルゴリズム (Evolutionary Algorithm: EA) であり, 自然界において環境に適した生物が生き残り, 次の世代へと子を残し, より環境に適した生物へと進化する過程を工学的に模した学習的アルゴリズムであり, 特に目的関数が一つである GA を単目的 GA という. 乱数を用いて多くの初期解候補を生成し, この集団の解候補の情報を活用して, 新たな解候補を生成する最適化手法である. 多くの解候補を用いるものは多点探索型と呼び, 探索に乱数を用いることから, 解の最適性の保証が担保されないという特徴をもつ.

GA において, 生物を個体 (individual) とみなし, 個体は設計変数の値がコーディングされた染色体 (Chromosome) と呼ばれる文字列上で表現され, 複数の遺伝子座 (locus) に格納された遺伝子 (gene) から構成される. この個体の集合を母集団 (population) と呼び GA はこの母集団に対して, 選択 (selection), 交叉 (crossover) 突然変異 (mutation) の遺伝子操作を繰り返すことでより適した個体に進化させる. これらの一連のアルゴリズムのフローを図 3.1 に示し, その詳細を以下に述べる.

1. 初期化 (Initialize)

予め設定した N 個の個体からなる初期集団を発生させる. 通常は, 遺伝子に格納される値はランダムによって決定する.

2. 評価 (Evaluation)

それぞれの個体を問題に応じて評価し, 評価値 (Evaluation Value) を得る.

3. 選択 (Selection)

それぞれの個体評価に基づいて, 新たに生成する子孫 (Offspring) の親となる個体を選択 (Selection) する.

4. 交叉 (Crossover)

選択された 2 個体で各々の個体が持つルールを交換し、新しい個体を生成する。交叉は 3. 選択により選ばれた親が対象となる。

5. 突然変異 (Mutation)

交叉によって生成された子孫のうち、新しい染色体の遺伝子のある遺伝子座に対して一定の確率で変化を起こす操作を突然変異呼ぶ。

6. 世代交代 (Replacement)

交叉により生成された新しい子個体を母集団に追加し、追加された個体の数だけ評価の低い個体を削除する。

7. 終了判定 (Terminate Check)

予め定められた終了条件を満たすまで、2. から 7. の操作を繰り返す。

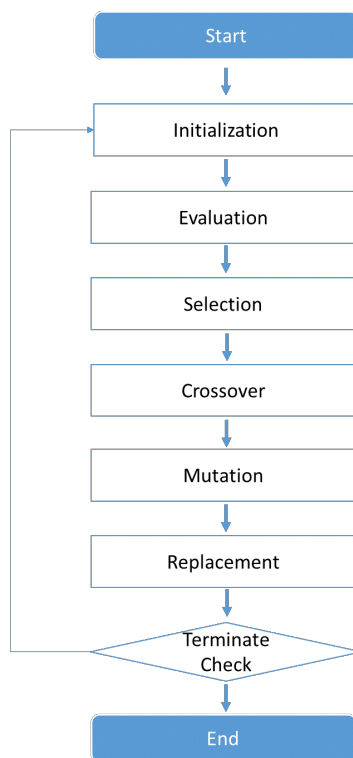


図 3.1 遺伝的アルゴリズムの流れ

3.1.1 オペレーター

上記のアルゴリズムの流れを示した通り，GAには選択，交叉，突然変異という進化オペレーターがあり，それぞれのオペレーションの組み合わせの違いにより性能の差異が生じる．以下では具体的にはオペレーションについて説明する．

選択

主な選択法として，ルーレット選択，トーナメント選択がある．ルーレット選択は個体の評価値に比例する確率で次世代の個体が選ばれる．確率を用いた選択方法であることから，評価値の低い個体を選択される場合もある．トーナメント選択の場合は解集団からランダムに選択された複数の個体 n 個をその中で最も評価値の高い個体が親個体として選択される．ランダムに選ぶ n 個体はトーナメントサイズといい，評価値の高い解が選ばれる確率を調整することが可能である． n が大きくなるほど局所探索性能は高まるが，局所解に陥った場合には脱出が出来ない可能性がある．

交叉

交叉は選択された親個体同士の遺伝子の一部を組み替え，子個体を生成する遺伝的な操作である．本研究では，遺伝子表現として連続値，バイナリ両方と用いており，バイナリでの主流の交叉方法である，1点交叉，多点交叉を説明し，連続値型の交叉は Simulated Binary Crossover (SBX) [94] について説明する．

1点交叉は交叉する位置をランダムにより1ヶ所決め，その前で一方の親の遺伝子，後ろでもう一方の遺伝子を子の遺伝子に受け継ぐ．1点の場合には点の前後により遺伝子を組み替えていたのに対して，複数点交叉の場合には点が複数あることから前後で組み換えを行う．SBXに関しては，各遺伝子に確率分布を付与し，この確率分布に従い遺伝子の値を決定する．

$$\beta_i = \begin{cases} \frac{1}{(2u_i)^{\eta_c + 1}} & \text{if } u_i \leq 0.5 \\ \left(\frac{1}{2(1-u_i)}\right)^{\eta_c + 1} & \text{otherwise} \end{cases} \quad (3.1)$$

式 (3.1) における u_i は $[0,1]$ の乱数の値であり， η_c はパラメータである．

突然変異

本研究で使用する突然変異は決定変数とし，実数値自然数関係なく，一様分布を仮定した乱数により値を選択する．今回使用する遺伝子の表現型は0，1のバイナリではなく，数値で表現する．通常の突然変異では，遺伝子長の確率でビットを0，1を反転する操作

が適用される。数値で表現した場合にも確率は同様で、通常は遺伝子長分の 1 の確率で作用する。

3.2 多目的遺伝的アルゴリズム : NSGA-II

3.2.1 パレート最適解

多目的（複数の目的関数を持つ問題）に拡張した GA を多目的 GA という。また、トレードオフ関係にある複数の評価基準（この基準を目的関数として表す）から最適値を求める問題を多目的最適化問題という。多目的最適化によって得られる解は単一ではなく、パレート解集合と呼ばれる複数のパレート解の集合が得られる。中でも優劣が付けられない解をパレート最適解といい、パレート最適解は多目的最適化問題における解の優越関係により定義される。全ての目的が最小であると仮定し、多目的最適化問題における解の優越関係を以下に定義する。

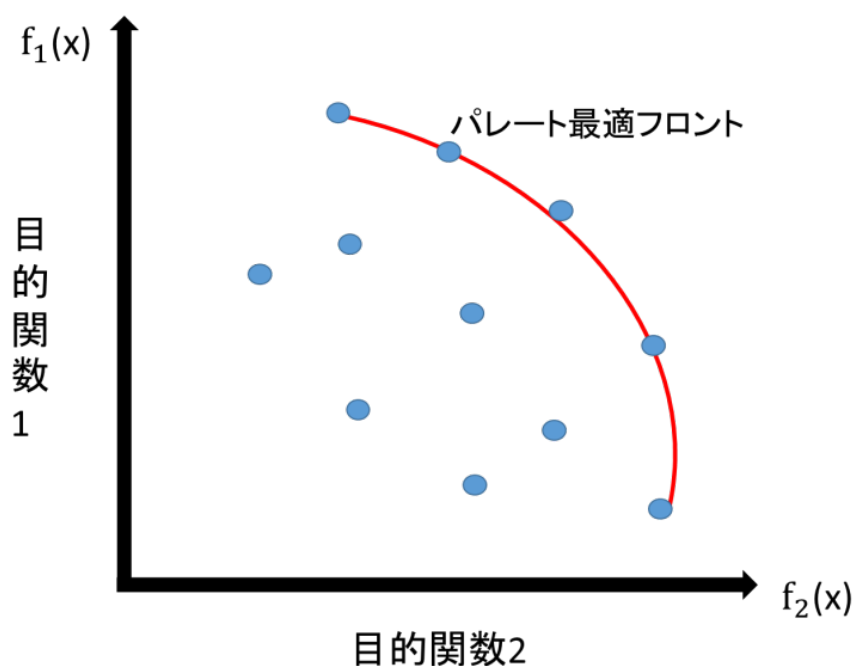


図 3.2 パレート最適フロント

図 3.2 はパレート最適フロントの例を示しており、2 目的最適化である。縦軸、横軸はそれぞれ 1 つ目と 2 つ目の目的をそれぞれ表す。青のシンボルは個体を示し、解に相当

する。赤の弧で囲まれた解集団（いずれも）がパレート最適フロントを形成する。

定義（優越関係）

$x^0 \in \mathcal{S}(x = (x_1, x_2, \dots, x_n))$ とする。

1. x^0 に強い意味で優越する $x \in \mathcal{S}$, x^0 を弱パレート最適解（Weak Pareto-optimal solution）という。
2. x^0 に優越する $x \in \mathcal{S}$ が存在しないとき, x^0 をパレート最適解（Pareto-optimal solution）という。

目的関数は2つの場合, パレート最適解の例を以下の図 3.2 に示す。

3.2.2 NSGA-II (Elitist Non-Dominated Sorting Genetic Algorithm)

多目的最適化の代表的な手法に NSGA-II がある。NSGA-II とは, NSGA の改良アルゴリズムとして Deb, Agrawal によって 2001 年に提案された手法である [8]。この手法は, (1) パレート面に近い個体を優先的に選ぶ非支配ランクソート (Non-Dominated Sort), (2) 解の多様性を維持するための混雑度トーナメント選択 (Crowded Tournament Selection Operator), (3) 次の世代へと適合度の高いエリート個体を残すアーカイブ保存の3つの特徴を持つ。

クラウディングディスタンス

ある個体の密度を評価するための手法に混雑距離 (Crowding Distance) がある。混雑距離とは, 同一ランクの解 (同一フロント内) で用いられ, 各目的関数軸において隣り合う個体間との距離を足し合わせた値である。混雑距離の概念を図 3.3 に示す。図 3.3 は概念図であり, 青と赤の各シンボルは個体を表す。青の解集団は赤の解集団に対して支配関係を持つのでこの場合には赤のクラウディングディスタンスの計算の場合にその対象とならない。赤の解集団はクラウディングディスタンスの計算対象となっており, 計算は隣接する個体との距離に基づき算出される。アルゴリズムを以下に示す。

Algorithm 1 の流れを説明する。初めに, 同フロント内に属する個体の初期化を行う。次に, 2行目では目的関数値 (Fitness) に基づきソートを行う。ソートの後, 3行目では, 一番両端の解 (目的に関する最大・最小) に関して値 ∞ を与える。それ以外に関しては前後隣接する個体から値を算出する。

NSGA-II における母集団の更新の概念を図 3.4 に示す。世代 t における優れた解を保存する母集団 P_t と, 交叉・突然変異といった遺伝的操作を用いて解を探索するための子母集団 Q_t を生成する。 Q_t に対して各遺伝的操作を施し Q_t に対して各遺伝的操作を施し, Q_t を更新する。次に, 親集団 P_t と各遺伝的操作を施した Q_t を組み合わせた

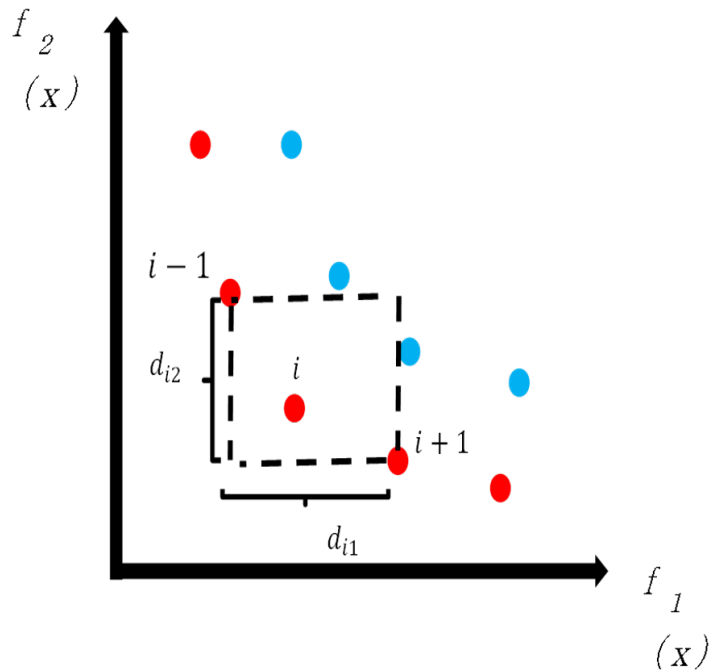


図 3.3 クラウディングディスタンス

Algorithm 1 混雑度距離の計算

- 1: フロント F に含まれる個体の数を変数 l に代入する: $l = |F|$. また, このフロントに含まれる各個体 i に対して初期値設定を行う: $d_i = 0$.
- 2: 各目的関数 $m=1,2,\dots,M$ に対して, 目的関数値が低い順に個体をソートする: $I_m = \text{sort}()$.
- 3: 各目的関数 $m=1,2,\dots,M$ に対して, まず境界個体 (目的 m の最大値と最小値の個体) に対して最大距離, もしくは無限距離を与える: $d_{I_1^m} = d_{I_l^m} = \infty$. さらに境界個体以外の全ての個体 ($j=2,\dots,l-1$) に対して以下の式に従い混雑度を計算する

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{I_{j+1}^m} - f_m^{I_{j-1}^m}}{f_m^{\max} - f_m^{\min}}$$

$R_t = P_t \cup Q_t$ を生成し, 選択操舵によって個体数 $2N$ の R_t から個体数 N の P_{t+1} を新たに選択し探索を進める. 実際のフローはアルゴリズム 1 に示す.

被支配ランクソート

NSGA-II における適合度割り当てについて説明する. 上述のように NSGA-II では非支配ランクソートと呼ばれる個体のランク付け方法を用いている. 非支配ランクソートは

個体が他個体よりも優越しているかどうかでランク付けする方式である。各個体の非支配ランクの概念を図に示す。縦軸と横軸はそれぞれ目的関数 $f_1(x)$, $f_2(x)$ である。各点はそれぞれ1つの解を示しており、(1) どの解にも支配されていない解を Rank 1, (2) Rank 1 の解を除く度の解にも支配されない解を Rank 2, (3) Rank 1, 2 の解を除く度の解にも支配されない解を Rank 3 とし、それ以降も同様にランクを決める。解は Rank 1, 2, 3 の順に適合度高い解である。

混雑度トーナメント

混雑度トーナメント選択 (crowding tournament selection operator) は、各ランク (r) により形成されるフロント (F_r) が一様に分布するよう解を選択する。トーナメントサイズ 2 のトーナメント選択に基づいた方法がある。混雑度トーナメント選択では、全ての解 i に対して次の2つの属性を持たせ、それらを選択基準に解を選択する。 i, j を各個体を識別する添え字とする。

1. 母集団における非支配ランク (i_{rank})
2. 母集団内の局所的混雑距離 ($i_{distance}$)

混雑度トーナメント選択では、 i と j の2個体の優越関係として以下のいずれかの条件を満たす場合に“ i は j より優れている”と定義する。

1. 個体 i のランクのほうが個体 j のランクよりも優れている： $(i_{rank} < j_{rank})$
2. 個体 i と j はともに同じランクであり、 i の混雑距離が j よりも優れている： $(i_{rank} = j_{rank}) \text{ and } (i_{distance} > j_{distance})$

図 3.4 はソートの例を表しており、非支配ランクソートと混雑度距離ソートの使い分けを示す。ソートにおいては非支配ランクソートからフロントに基づき序列を決定する。 F は非支配ランクソートに基づき決定した序列であり、数字が序列を示す。この時 F_1, F_2 の総量は半数に満たず、 F_3 で過半数を超える。 F_1, F_2 は次世代に引き続き、 F_3 の中で混雑度ソートにおいて序列の高い個体が次世代に引き継がれる。図中では赤での斜線内に属する個体は次世代に引き継がれず、淘汰されることを示す。

Algorithm 2 は親によって生成された子集団は親集団を含めソートを行う。ソートによりランクを決定する (1 行目に対応)。ランクの序列が高い個体から次世代の母集団に加えていくが、あるフロントで母集団以上の個体数となるので、このフロントに対してクラウディングディスタンスを計算する (2, 3 行目に対応)。最後にクラウディングディスタンスの値に基づき個体を母集団に加えていき、母集団数を満たすまで繰り返す (4 行目に対応)。

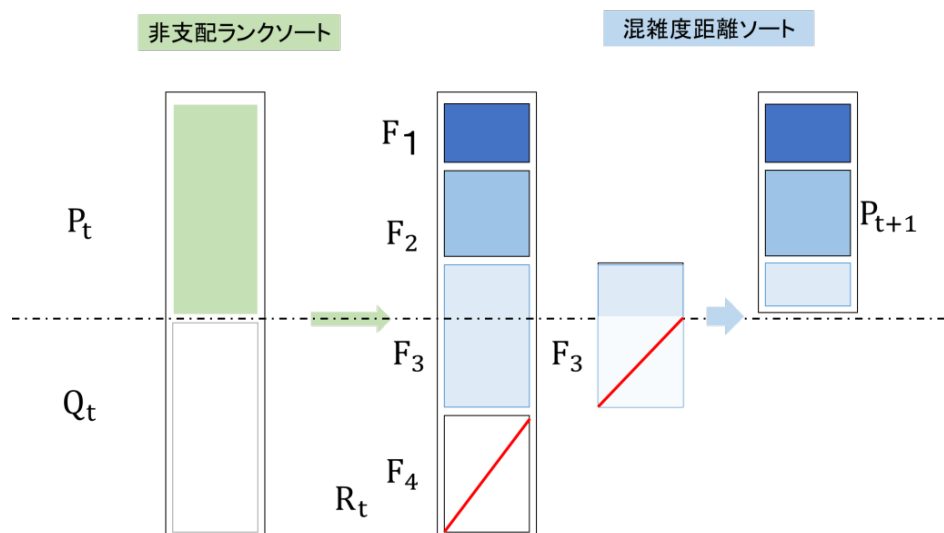


図 3.4 NSGA-II ソート

Algorithm 2 NSGA-II のアルゴリズム

- 1: 親母集団と子母集団を組み合わせて $R_t = P_t \cup Q_t$ を作成する. R_t に対して非支配ランクソートを実行し, 全個体をフロントごと (Rank ごと) に分類する: $F_i, i = 1, 2, \dots$, etc.
 - 2: 新たな母集団 $P_{t+1} = \emptyset$ を生成. 変数 $i=1$ とする. $|P_{t+1}| + |F_{t+1}| < N$ を満たすまで, $P_{t+1} = P_{t+1} \cup F_t$ と $i = i + 1$ を実行.
 - 3: 混雑度ソート (Crowding-sort) を実行し, 多様性の高い個体 $N - |P_{t+1}|$ 個を P_{t+1} に加える
 - 4: P_{t+1} を基に, 混雑度トーナメント選択, 交叉, 突然変異を用いて新たな子母集団 Q_{t+1} を生成する.
-

3.3 ノベルティサーチ

ノベルティサーチは Lehman が提案した個体の評価指標であり, Fitness にとらわれな
い新しい手法である [95]. Fitness は特定の評価に基づいた絶対評価であり, この絶対的
な基準を用いて個体の優劣が決定される. 一方でノベルティサーチは解集団の中での優劣
が決定し, 解の探索範囲を広げるように評価は作用するため, 大域的な探索が可能である.
個の評価法は他の個体と比較したとき, 違う探索点を探している場合に高い評価が得られ
るような評価式となっており, 局所解に陥ることなく, 網羅的に探索できる可能性が高
くなる. 機械学習などの最適化を行う手法には探索 (exploration) と活用 (exploitation)
の2つの側面が存在し, ノベルティサーチは活用に属する. 既存の活用の方法論として,

局所解探索 (Local Search) や欲張り法 (Greedy Method) がある。これらの手法の差異として、複数の解集団から算出される評価値を利用することから、集団での相対評価による個体の選択であるため、個体の位置関係や個体数に依存することや、評価値算出の際には評価する個体とそれ以外の解集団内の個体との評価値を算出しなければならないため、多くの計算量を要する。計算量削減のため、kd-tree を利用することで自身と距離関係が近い他の解を自身のノードの近くに配置すること計算量を削減することが出来る。

この手法は NEAT (The Neuro Evolution of Augumenting Topologies) と呼ばれるニューラルネットワークを進化的に構築する手法において novelty search の有効性が示された。ニューラルネットワークにおいてはノード同士の接続の組み合わせは膨大にあるため、この組み合わせ (トポロジー) を進化的に獲得するため NEAT が提案された。進化的に獲得すべき対象はノード間の重みであり、これが遺伝子として格納されている。遺伝子の評価値 (Fitness) はニューラルネットワークが最適化を行いたい問題である。例えば、関数の最小化問題であれば、その値が評価となる。ただし、Fitness のみの評価で最適解を得られない場合があり、novelty search を用いてこの問題の解決を図る。

図 3.5 は距離センサ及びモーターの制御をニューラルネットワークにより学習させる問題であり、ゴールまでの距離が評価値となる。ノベルティサーチを評価として利用した場合の解集団を表した例である。この例題は NEAT を用いて×印を示すスタート地点から★記号のゴール地点までを到達できるか実験を検証した例題である。黒い点は解集団を表している。図において、空間に均一に分布するように配置されているように見えるが、これは1つの解が周辺に分布している解との距離関係に基づいて評価を行っているため、解の分布が疎である部分に対して積極的に分布するよう作用するために図 3.5 のような分布となっている。

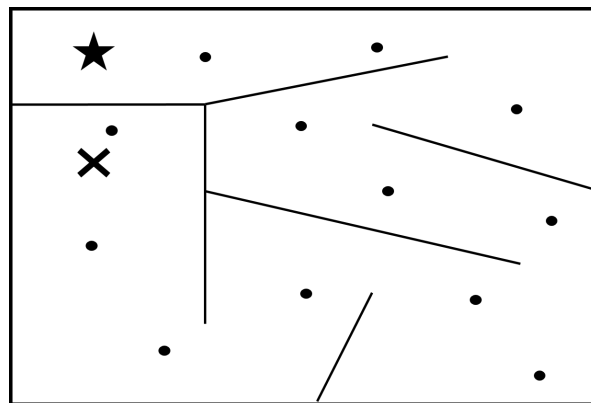


図 3.5 ノベルティサーチを評価に用いた場合の解の分布

文献 [95] を参考に

一方，図 3.6 は Fitness における解集団を表しているが，Fitness はゴール地点までの残り距離を評価とするため，行き止まりを越えた地点にゴールがある，スタートとゴールの中間地点に解が集まってしまうため，これ以降の解が進化することがなく局所解に陥った状況となるため，ゴールに到達することなく収束してしまう。

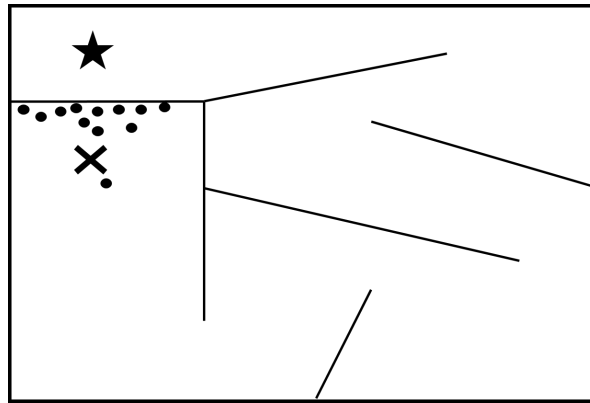


図 3.6 Fitness を評価に用いた場合の解の分布

文献 [95] を参考に

この他にも進化ロボティクス (Evolutionary-Robotics) などの分野で有効であり，マルチエージェントの協調タスクにおいてもノベルティサーチは有効であることが示されている [96–98].

ノベルティサーチは例えば迷路における行き止まりのような deceptive な問題に強いのが特徴であるが，解探索の特性上，探索空間が膨大になる場合には探索能力が下がる可能性がある [99]. その問題を解決するため，Fitness と併用して，線形的にスケールングして評価を行う手法や多目的の目的の一つとして評価を用いることで対応する [96]. この場合はどちらかの解に偏りが生じる恐れがあるが，パラメータの調整が必要であり，ノベルティにおいて多様性を維持しつつ有望な探索点を重点的に探索圧をかけることができない。そのため 有望な探索領域における探索力保持しつつ，多様な解生成を促すための制御必要となる。ノベルティに基づく評価式を以下に記述する。

$$\rho(x) = \frac{1}{K} \sum_{n=1}^k \text{dist}(x, \mu) \quad (3.2)$$

式 (3.2) における x は評価される解， μ は他の解を表しており， $\text{dist}(x, \mu)$ 2つの距離を求める関数となる。 k の値は近傍数を示しており，ノベルティの評価を用いる場合は 15 から個体数全てを用いることが多い。

3.4 クラスタリング

データの集合を部分集合にわけける方法論の1つであり、機械学習では教師なし学習に分類される。分類時に用いる部分集合はクラスタと呼ばれる。この時、同クラスタに属している場合には互いに似た特徴を持つ。クラスタ手法は大まかに階層的クラスタリングと分割最適化クラスタリングに分かれる。階層的クラスタリング最も近いデータを同クラスタに割り当てる方法であり、クラスタの過程を階層のようにあらわすことで可能であり、デンドログラムとして可視化することが出来る。

クラスタリングを行う際にはいくつかの距離測定方法を用いるが、主要な計測方法として、単リンク法、完全リンク法、群平均法、ワード法、重心法、重み付き平均法、メジアン法があり、これらの説明を以下に記載する。

3.4.1 単リンク法

最短距離法とも言い、クラスタ同士を併合するときには、併合前において、一番近い方のクラスタまでの距離を併合後のクラスタまでの距離とし併合する。以下に距離測定のために使用する式を記述する。

$$d_{kz} = \min \{d_{kx}, d_{ky}\} \quad (3.3)$$

式 (3.3) は既存のクラスタの x 、 y を併合し、新たなクラスタ z とした時の距離を計算する場合の条件を式を表しており、 z の式は既存のクラスタの距離である x における距離である d_{kx} と、 y における距離である d_{ky} の最小の値を採用することを表す。 k は既存のクラスタを表す。クラスタの距離はクラスタ間の最小値で示されるため、最短距離法と呼ばれる。

3.4.2 完全リンク法

complete linkage method:最長距離法とも言い、クラスタ同士を併合するときには、併合前において、一番遠い方のクラスタまでの距離を併合後のクラスタまでの距離とし併合する。

$$d_{kz} = \max \{d_{kx}, d_{ky}\} \quad (3.4)$$

式 (3.4) は既存のクラスタの x 、 y を併合し、新たなクラスタ z とした時の距離を計算する場合の条件を式を表しており、 z の式は既存のクラスタの距離である x における距離である d_{kx} と、 y における距離である d_{ky} の最大の値を採用することを表す。クラスタの距離はクラスタ間の最大値で示されるため、最長距離法と呼ばれる。

3.4.3 群平均法

group average method, 上記 2 つの計測法の中間的な計測法であり, 既存の 2 つのクラスタを併合し, 新たなクラスタを生成する際に, 2 つの距離の荷重平均により新たなクラスタの距離を定める. 以下に距離を表す.

$$d_{kz} = \frac{|C_a|}{|C_c|}d_{kx} + \frac{|C_b|}{|C_c|}d_{ky} \quad (3.5)$$

$|C|$ は所属するデータの総数を表す. k は先ほどと同様に既存の k 番目のクラスタを表す. x, y は併合されるクラスタ, z は併合された後の新しいクラスタを表す. d_{kx}, d_{ky} は併合前のクラスタ k との距離関係を表しており, この 2 つの荷重平均により, 新たなクラスタの値が決定する.

3.4.4 ウォード法

(Ward's method); 最小分散法とも呼び, 併合された後の分散と併合前のクラスタの和との差が最小となるようなクラスタを併合する. 以下に式を記載する.

$$d_{kz} = Var(C_k \cup C_z) - (Var(C_k) + Var(C_z)) \quad (3.6)$$

C_k は既存のクラスタを指し, C_z は新たに生成されたクラスタを指す. $Var(C)$ はクラスタの分散値を表す. 分散値を用いることで外れ値に強い特性を持つ.

3.4.5 重心法

(centroid method); それぞれのクラスタの重心により距離を決定する方法論を指す. 距離はユークリッド距離を用いて重心計算を行う.

$$d_{kz} = \frac{|C_x|}{|C_z|}d_{kx} + \frac{|C_y|}{|C_z|}d_{ky} - \frac{|C_x||C_y|}{|C_z|^2}d_{xy} \quad (3.7)$$

$|C|$ は所属するデータの総数を表す. d はユークリッド距離の 2 乗を表し, d_{kx} は既存クラスタである k と併合される前であるクラスタとの距離を示す.

3.4.6 重み付平均法

(weighted average method); 基本的な計算方法は群平均法と同様であるが, クラスタによって異なる重みづけ (クラスタ内のデータの総数と重みが一致しない) となるため, クラスタのデータの総数が大きい場合にはこの影響を受けにくい特徴をもつ.

$$d_{kz} = \frac{\frac{d_{kx}+d_{ky}}{2} + d_{kx+y}}{2} \quad (3.8)$$

d_{kx}, d_{ky} は併合前のクラスタと既存のクラスタ k の距離を表しており, $d(kx + y)$ は併合後クラスタ集団と既存クラスタの距離を表しており, 最終的にはこれらの平均を用いて算出される.

第4章

羽田空港とシャルル・ド・ゴール 空港

羽田空港は東京の南東部に位置する空港であり，日本で最大の航空交通量を持ち，世界でも4番目の交通量を持つ混雑空港である．図4.1は羽田空港の空港場面を表しており，滑走路の構造として，4つの滑走路を有する．黒色はターミナルであり，上から二つが国内線，一番下のターミナルは国際線をそれぞれ示している．羽田空港は滑走路がそれぞれ並列に存在することから離陸と着陸の分離運用が可能である．日本の空港情報はAISJapanから入手できる [100]．

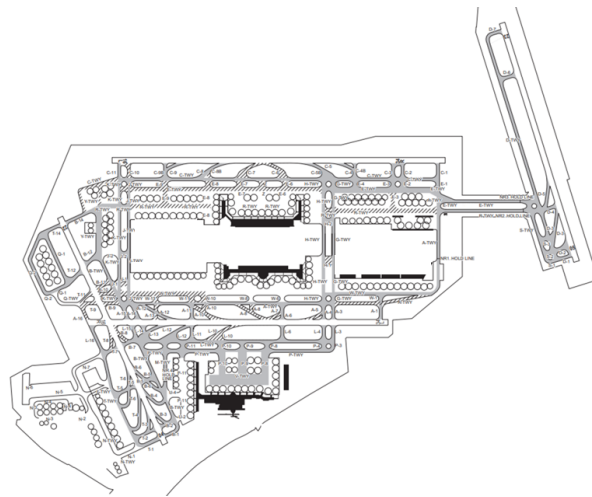


図4.1 羽田空港

出典：AIS JAPAN

4.1 羽田空港

図 4.2 は航空管制がある時間帯で行っているレーダベクタの例である。四角と点で構成されたマークは航空機を表しており矢印は管制官により指示された方位を示している。空港は黄色の四角い枠で表される。矢印の航空機はすべて羽田空港へ向かって航行している。東側の大島付近を航行する航空機は、縦に一系列に並んでいるが、管制官がベクタリングを行っている。矢印は左方向だが、この数分後、上向きに管制官が方向の指示を出す。左から 3 番目の矢印の航空機は南側へ航行していたが、北方向へ指示を受けて、旋回していることがわかる。これは、この航空機の北側を航行する 2 機の後ろに並べるために管制官が方向指示を行ったからである。このように羽田空港へ着陸する際には STAR に従わず、管制官の指示の下に着陸誘導することが多くある。北、東の経路に関しては、空港の北東部から航空機同士の合流を行う。

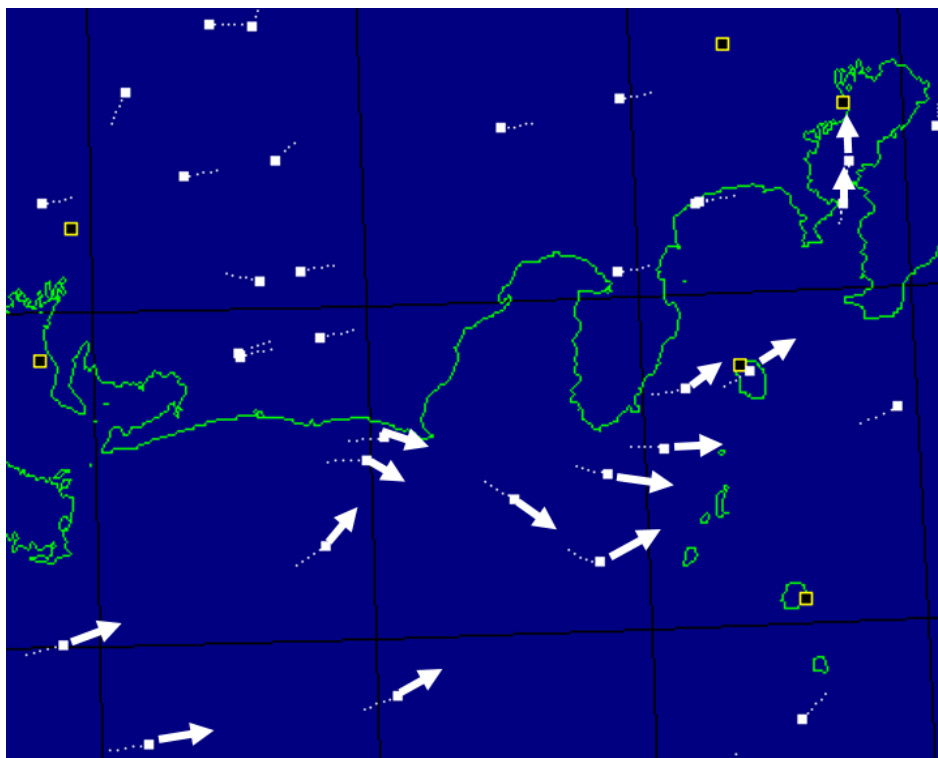


図 4.2 ベクタリングの例

図 4.3 は羽田空港東側に位置する横田進入管制空域を示している。FL は Flight Level の略称であり、数値に 100 倍した値が制限高度を示し、単位は f t である。左下の数値は緯度経度の情報であり、() 内の数字がエリアの外の数字に対応している。羽田空港は 8000 と表示された空域の東側に位置している。この空域は横田飛行場の管制空域であ

横田進入管制空域
Yokota Approach Control Area

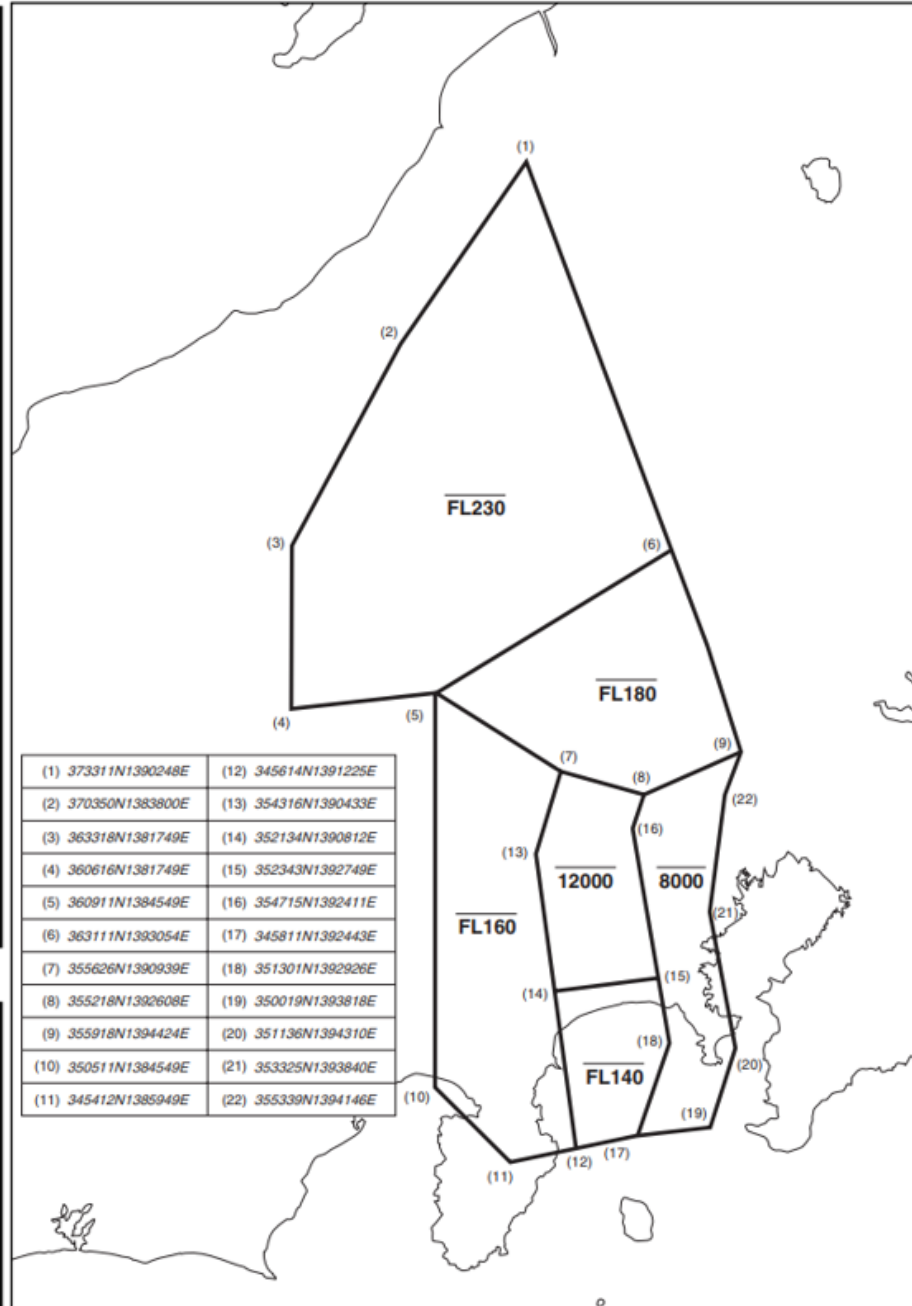


図 4.3 yokota ACC

るため、空域を利用するためには米軍との事前調整が必要であり、民間機の多くはこの空域を避ける（上空を通過）ことにより羽田空港から西日本や関西方面へ出域するために利用される。これをみてわかるように羽田空港に近いほど低い空域から制限があることがわかる。このため、出発機の場合には空域を回避するように急上昇するが、到着機の場合、西側の航空機は空港から南のほうへ南下し、その後滑走路に着陸するため航空機の着陸順が生成される。

4.2 シャルル・ド・ゴール空港

シャルル・ド・ゴール空港はフランス最大の空港であり、パリの中心から北東部に位置する。図 4.4 は空港場面を表している。滑走は並行に4つ持つが、それぞれ北と南二つの滑走路が併設されている。その間には航空機を駐機するスポットとターミナルが存在するため、北と南の同時運用が可能である。滑走路の着陸する向きは西側と東側の二方向が設定されており、風向きによって使用する滑走路の方向が異なる。

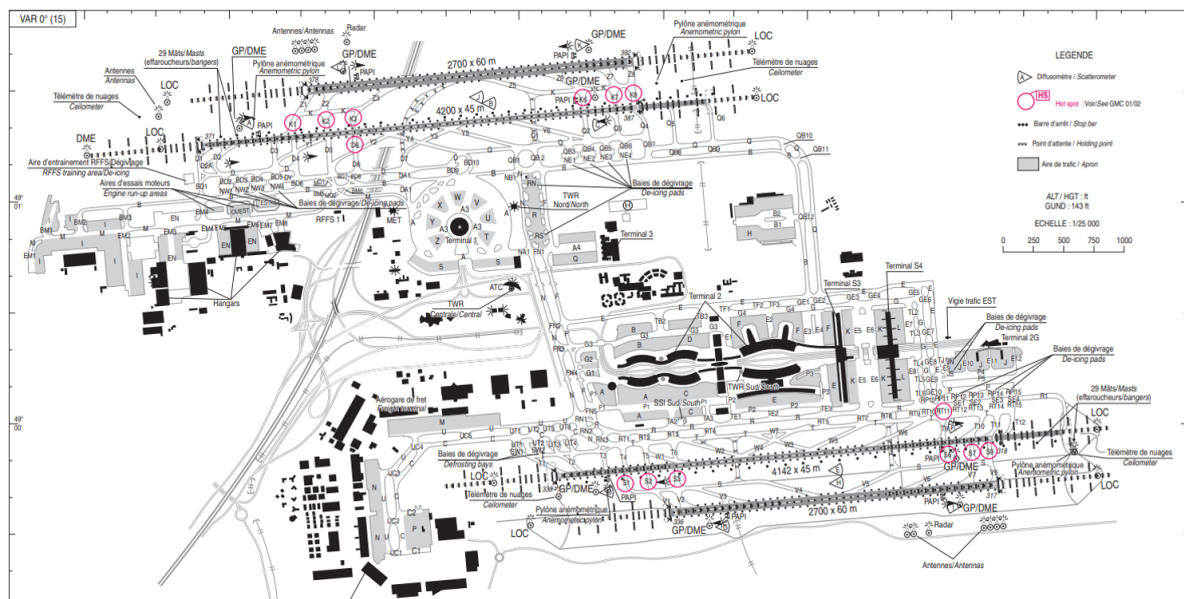


図 4.4 Charles de Gaulle Airport (シャルル・ド・ゴール空港)

出典：AIP FRANCE

シャルルドゴールにおける STAR の経路は北西、北東、南東、南西の4方向の航空路から航空機が進入し、それぞれの方角において2点の進入点を有する。全ての経路はWaypoint を結んだ形で定義される。このように4方位から空港へ向け航空機は巡航を行うが、主に北西はスペイン、北米便、イギリスからの航空機が、北東は日本をはじめとするアジア便やドイツや北欧をはじめとするヨーロッパ便が、南東はトルコやスイス、イタリアをはじめとするフランス以南のヨーロッパ地域からの航空機が、南西からは、アフリ

カ、ブラジルからの航空機がそれぞれ飛来する。

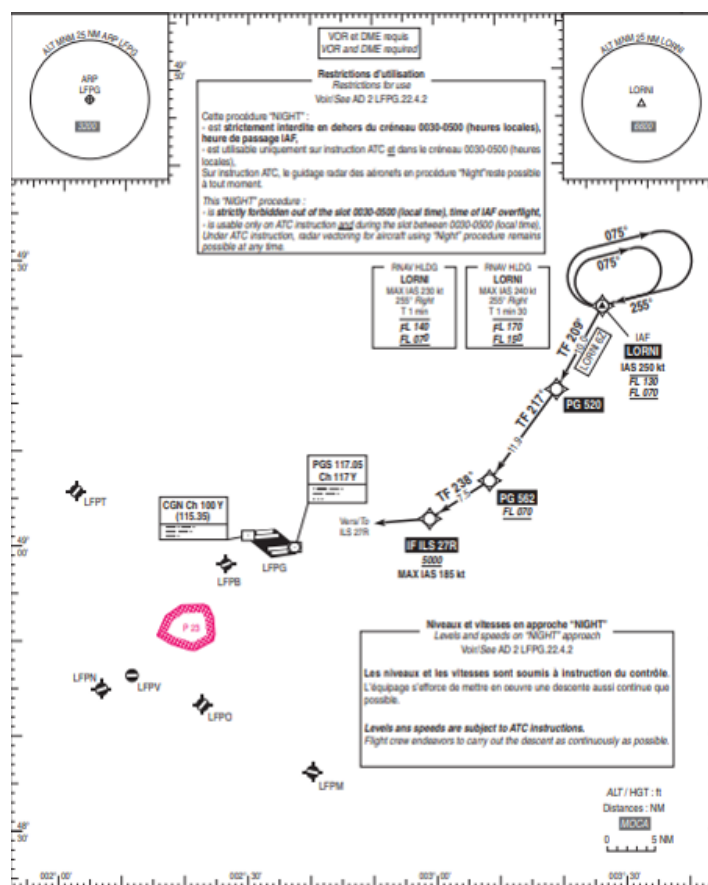


図 4.5 CDG における STAR の定義例

出典：AIP FRANCE

図 4.5 は滑走路までの経路の一つであり，日本と同様 Aeronautical Information Publication (AIP) に記載されている [101]。黒い四角で囲われている文字が Waypoint 名であり，その下に制限情報が記載されている。このチャートでは，北東に位置する LORNI から空港へ向けた経路を示している。具体的には，LORNI, PG520, PG562, IFILS27R を通過し，最終的にシャルル・ド・ゴール空港の滑走路に着陸する。この経路の Waypoint 情報を以下の表 4.1 にまとめた。

表 4.1 CDG における Waypoint の位置

Waypoint	latitude	longitude
KOVAK	47.99306	0.846667
BENAR	48.25306	0.744444
ROMGO	48.34658	0.844111
NERKI	48.51306	1.327139
BANOX	48.57419	1.5055
DOMUS	48.88808	1.963861
PG515	48.91622	2.459417
PG566	48.87097	2.729778
PG568	48.885	3.006389
IF_27R	49.04853	2.98425
LUMAN	47.93944	0.413333
GONEK	48.12	0.603333
PETAX	50.18656	0.848
VELOL	49.93861	1.286667
VEPET	49.71028	1.501111
KOLIV	49.29997	1.570556
MOPAR	49.29172	1.757278
PG534	49.26172	2.387972
PG536	49.23169	2.715
PG560	49.20139	2.967889
PESUK	49.21083	0.290833
PEXIR	49.20278	0.613333
PITAV	49.19472	0.933333
MOFIL	50.14778	4.107778
KENAP	49.95861	4.093889
XERAM	49.59667	4.067222
ENORI	49.47469	3.777944
DEVIM	49.45017	3.632222
LORNI	49.41942	3.451389
PG520	49.2745	3.325972
PG562	49.11492	3.144806
TALUD	49.60111	5.420556
RAPOR	49.59139	5.213056
VEDUS	49.59472	4.781389
BALMU	47.84672	4.639694
ERTIP	48.13444	4.406667
NANOP	48.26194	4.239444
URELO	48.53883	3.732028
OKIPA	48.61131	3.5635
PG522	48.7845	3.412722
PG564	48.95056	3.267111
RLP	47.90631	5.249167

表 4.1 は実際の経路の座標を示しており、緯度、経路情報が10進数で表示している。また、経路間、ウェイポイントの中には高度、速度の制限が与えられている。最終的にIF27Rから滑走路に向け着陸する。図、右上には円状の経路が表されているが、これはホールディングポイントと呼ばれる経路であり、滑走路へ向かう航空機が混雑した際に使用される経路である。この経路を使用し、空中待機することで、着陸を遅らせ、衝突を回避する役割を担う。ホールディングポイントは空港周辺に複数箇所設定されている。2017年のシャルルドゴール空港における到着機のデータを解析すると、1日の発着数が689機あり、1時間平均28.7機の着陸数がある。航空機のサイズは全体に対してHeavyが22%、Midiumが75%と2カテゴリーがほぼ全体を占めている。

表 4.2 入域地点毎の機数及び航空機サイズ

Num	Waypoint	機数	割合 (%)	S.H	H	M	L
1	KOVAK	74	10.7	0	7	67	0
2	LUMAN	32	4.6	0	9	23	0
3	PETAX	103	14.9	3	25	74	0
4	PESUK	39	5.7	2	30	7	0
5	MOPIL	71	10.3	0	12	59	0
6	TALUD	137	19.9	7	31	99	0
7	BALMU	220	31.9	1	39	179	1
8	RLP	13	1.9	0	2	11	0

表 4.2 は各エントリポイント毎における一日の航空機の機数とそのサイズの内訳を示している。左の列から順に進入点の識別番号、進入点におけるWaypoint名、1日当たりの航空機の機数、全航空機数に占める各エントリポイントでの航空機数の割合、Super Heavy (A380に相当)、Heavy、Medium、Lightの機数をそれぞれ示している。最終進入地点においては北西、北東、南西、南東の計4地点からの合流を行うが、南西からの合流が15%程度と他の入域地点と比べて交通流は少ないが、他はおおむね20%を超えていることがわかる。このことから、全体の合流、着陸順を考えるにあたり、他の3地点の合流も考慮しなければ適切な着陸順を生成出来ないことがデータからわかる。

図 4.6 は、時間毎の航空機の取り扱い機数を表している。縦軸が機数、横軸は時間帯をそれぞれ示す。例えば、横軸の1は0時から1時までにシャルル・ド・ゴール空港に入域した航空機数を表す。シャルルド・ゴール空港は2本の平行滑走路を有するため、2本の滑走路を利用した運用が可能である。そのため、時間によっては30機以上の着陸がある。一般的な航空機の着陸する航空機の混雑する時間帯とおおむね同じ傾向で、朝の7時にピークがある。1時間当たりの航空機数が50機を超える時間帯は、6時、7時、1

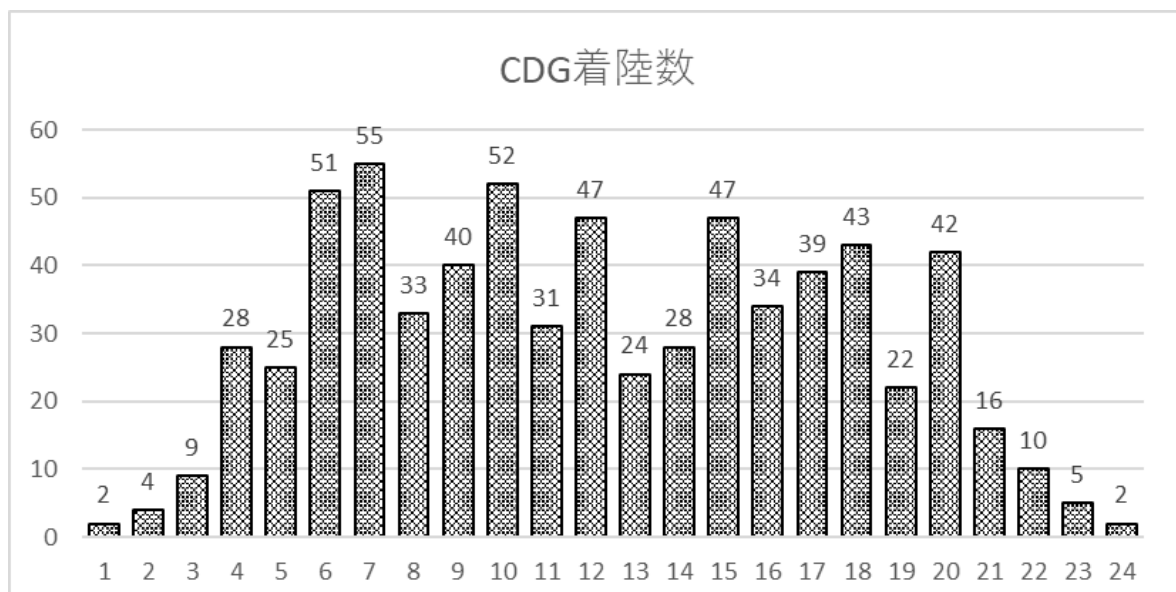


図 4.6 取り扱い機数のヒストグラム

0 時台が特に航空機が集中していることがわかる。

4.3 羽田空港におけるモデリング

4.3.1 自由経路

本研究では経路生成を行うために環境をグリッドマップとしてモデリングを行う。図 4.7 は最適化対象の地形をグリッドマップで表現した際の例である。例の中では高高度に存在する地理的障壁，気象擾乱を誘発する物体が存在する空域がグリッドマップ中で黒塗りで配置されている。このような空域を通過するにあたり，グリッド内ではコストを付加する。グリッドマップへとモデリングされる際には地形を 3 次元か 2 次元として扱うのかを決定する必要がある。航空機の経路，軌道の最適化の場合には，En-route におけるモデリングの場合には，高度変更を要する必要があるため，3 次元グリッドマップを用いてモデリングを行う [102]。ただし，空港周辺の空域において出発と到着の分離がされており，高度を用いた着陸順序の変更を行わないため，2 次元マップでのモデリングを行う。

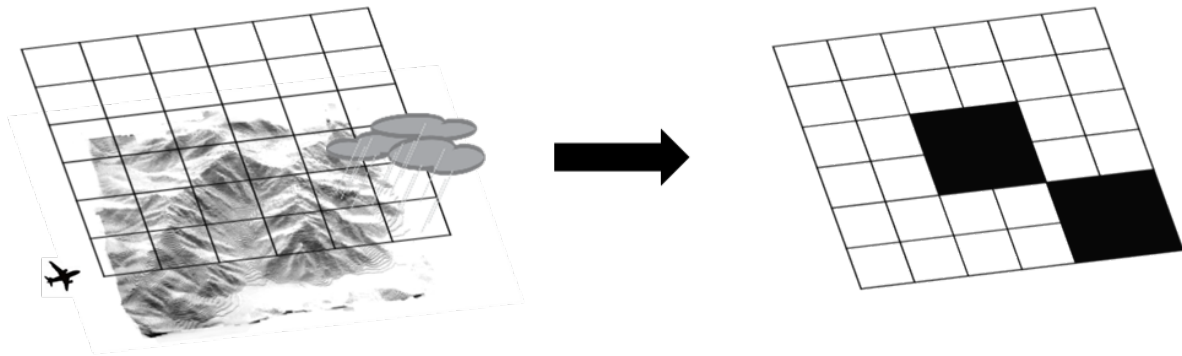


図 4.7 グリッドマップへのモデリング

図 4.8 はグリッドマップにモデリングされた羽田空港周辺を表す．図 4 で記載したように，羽田空港における到着機の割合は，西側からの入域が 7 割を占めているため，本研究は西側の到着機の最適化を行うように周辺のモデリングを行った [103]．着陸順最適化を行うための航空機の対象空域範囲は 250 km 四方の正方形であり，西側から入域し，およそ 30 分で着陸する範囲である．ここで，マップに示されている黒塗りは横田管制空域と標準出発方式の経路（STAR）が設定されているため，進入禁止エリアと定義した．左側の赤枠は航空機の入域エリアを表す．黄色の四角が滑走路における最終進入点を示し，最終進入点からは滑走路までは直進し，着陸へ向け降下する．水平，垂直にそれぞれ 5 本ずつ，直線が引かれているが，これが 5 マス毎の升目を示しており，グリッドは 25 マス四方となる．つまり，1 マス当たり 10 km となる．入域時の航空機の進入速度はおよそ 500 kt あり，分速に換算するとおよそ 15 km/min となるため，1 分当たり 1 マス以上進行することになる．方向展開を行う際にはこのグリッドのマス目が基準であり，1 分に 1 回程度，航空管制官が指示する機会をがあるため，マス目の解像度としては十分である．また，航空機の状態を把握するためのレーダカバレッジは空港は 100 km と本研究のモデルより短い範囲であるが，航空路で使用する航空路監視レーダーにおいては，300 km と大きな範囲で航空機の状態を把握することが可能である．ただし，300 km だと，航空機の降下を始めるタイミングよりだいぶ早い状況であるため 250 km と設定した．

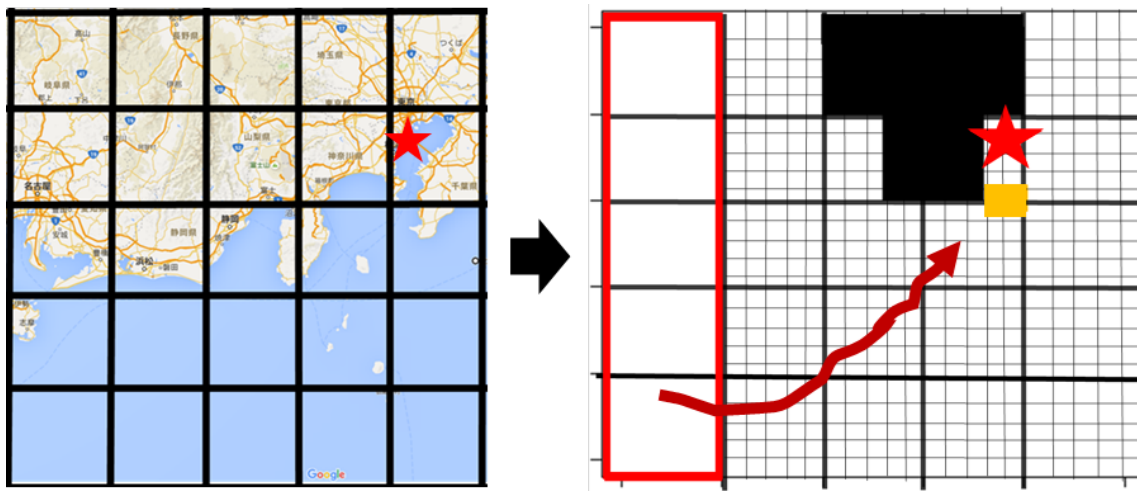


図 4.8 羽田空港におけるグリッドマップモデリング

4.3.2 シミュレーション

本研究で扱う航空機のサイズは同一とし、巡航速度も同一の平均速度 320 kt とした。空港周辺の航空機の速度は航空機毎に大きな変化はなく、同様の傾向を示している。そのため、航空機毎の速度の違いはないものとした。最適化を行う時間間隔は 5 分に 1 回とし、1 度最適化を行った後には再度航空機の位置情報が更新される。この時、更新された位置情報において、新たな航空機が最適化を対象とする範囲内入域した場合には新たな最適化対象の航空機として経路の生成を行う。

4.4 シャルル・ド・ゴール空港におけるモデリング

4.4.1 固定経路

空港周辺における着陸経路においては、STAR と呼ばれる経路が、多方面から設定されており、複数の着陸経路から最終的には滑走路へ向け一つの経路に合流する。実際の運用と同様、空港のレーダ範囲である 60 NM から 80 NM 程度航空機を視認することが可能である。航空機は航空管制官が視認可能な範囲である進入点である EntryPoint に進入する。レーダからは航空機の速度、高度、機種が取得可能であり、4 秒に 1 回の頻度で情

報の取得が可能である。この情報を基に航空管制官が目的空港に誘導する。この時、航空機同士の間隔が保持されるよう制御を行う。この時行われる制御がベクタリングであり、通常は、高度、速度、方位の指示を行い、コンフリクトの解消を図るが、空港周辺の到着機の場合、高度の制約があるため、速度、方位の指示により対応する。

4.4.2 航空機 の速度調整

速度の上限、下限は Romaric らの研究に従って設定した [104]。速さの制御は進入点でのみ変更可能であり、以下の式に従って行われる。

$$v'_k = v_k(1 + \Delta v), \Delta v \in \psi \quad (4.1)$$

v_k, v'_k はそれぞれエントリーポイントにおけるスピード、変更後における速度を表す。速度の変化は ψ の範囲で行われるが、目的空港が近いため、 $[-20, 5\%]$ の範囲で速度の変化が可能である。これはシャルル・ド・ゴール空港における実際 の速度変化に基づき設定された速度範囲である。

次に経路に関しては、今回例題として適用するフランス、シャルル・ド・ゴール空港を例に説明する。図 4.9 はシャルルドゴール空港周辺到着経路を表しており、それぞれ、8つの進入地点から、1つの滑走路の最終アプローチ点へと向かう。最終アプローチ地点での速さは航空機のサイズと各空港に依存するが、本研究対象であるシャルル・ド・ゴール空港では Small の航空機がないため、Midium, Heavy に対して 180 kt, 230 kt である。なお、これらの入域点、固定経路、最終進入点は定義され、固定されている。

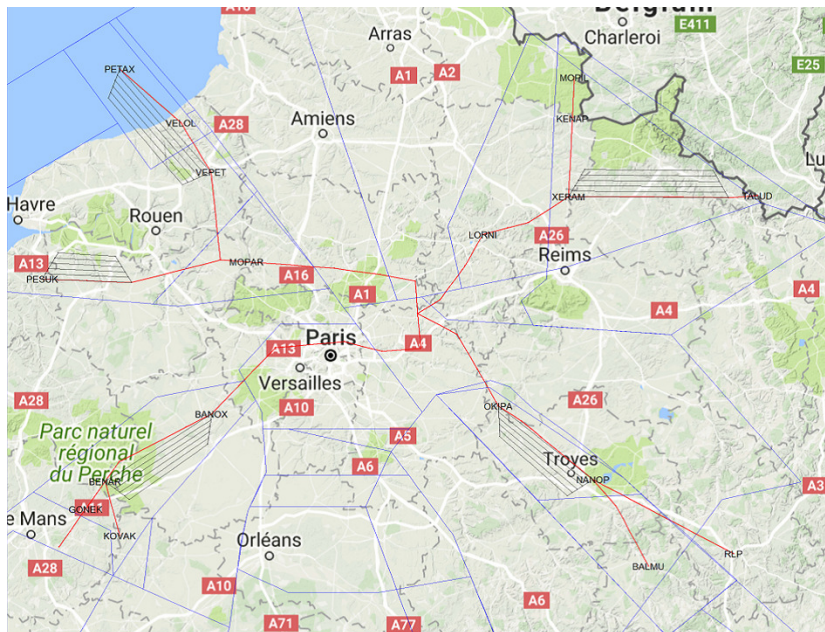


図 4.9 シャルル・ド・ゴール空港における STAR

4.4.3 迂回経路

本章で最適化を行う対象はシャルル・ド・ゴール空港でありホールディングポイントを有する。ホールディングポイントを利用した空中待機は着陸時間の遅延時間を大きくすることは可能であるが、遅延時間の大きさは周回の大きさに依存するため、より細かな遅延時間を航空機に与えることが不可能である。そのため、台形上の迂回経路本研究では付加する。図 4.10 は Detour を配置したシャルル・ド・ゴール空港周辺である。迂回経路は計 5 か所あり、北西に 2 ヶ所、残りはそれぞれ 1 ヶ所に設定されている。



図 4.10 CDG における迂回経路

図 4.11 は 1 つの迂回経路を表しており、台形の形状を採用する。この台形の左側の辺に沿って迂回を開始し、どの地点からでも右へ旋回可能であり、旋回後は上底と下底と平行して航行する。台形の形状を採用する理由は、旋回後、上底と下底に平行して航行するため、三角形の構造よりも航空機の乱雑度が低くなる。乱雑度は以下の式に従い計算される [105]。

$$N_c = \frac{2N_s \sqrt{V_{lj}^2 - V_{lj}V_{ij} \cos \theta_{ijl} + V_{ij}^2}}{V_{lj}V_{ij} \sin \theta_{ijl}} f_{ij} f_{lj} \quad (4.2)$$

N_s は正規化を行うためのノルム、 f_{ij}, f_{lj} は Waypoint t である i, j と l, j の間の経路を示す。 V_{ij}, V_{jl} は、経路 f_{ij} と f_{jl} における航空機の平均速度を表している。 θ_{ijl} は、 f_{ij}, f_{lj} における角度を表す。式 (4.2) の意味としては、コンフリクトの可能性の大きい航空機が存

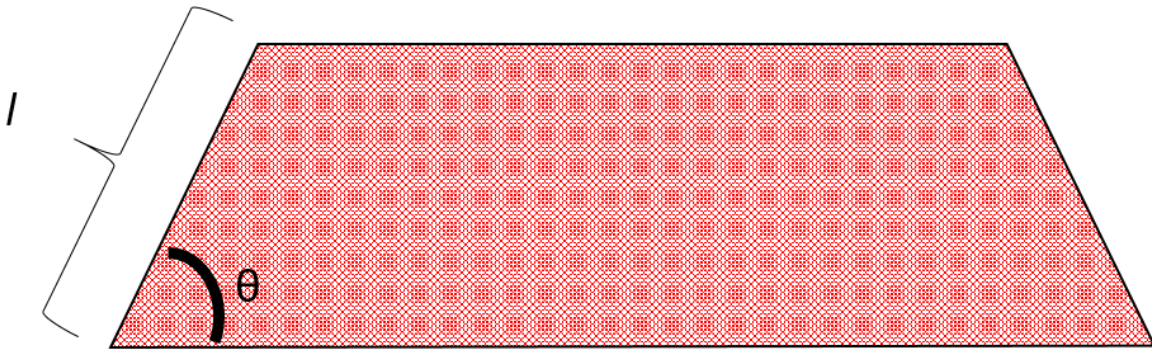


図 4.11 迂回経路

在する場合に、値が大きくなるように算出される。

迂回経路の範囲は STAR における Waypoint 間が下底の長さとなるが、側面に関しては l と θ によって変更が可能である。 l は最大迂回経路長を示し、 θ は迂回を始める際の迂回角度を示す。この 2 つの値が大きいほど、最大迂回経路長が大きくなる。

4.4.4 シミュレーション

航空機は入域の進入速度を決定した後、線形的に速度を落としながら目的空港の滑走路へ向け、降下を行う。航空機の速度の制御則の想定はいくつかあるが、例えば入域速度を最終進入地点まで維持する方法、燃料消費量を最小化するような航行方法などがある。空港周辺においては、速度は空港に近づくにつれ、減速していく。

図 4.12 は 2012 年 7 月 9 日における、羽田空港へ向かう、航空機の速度の推移を表している。機種は B 7 6 7 であり、サイズのカテゴリでは Heavy に属する。縦軸は速度で、横軸は時刻を示す。12 時 20 分付近で速度を徐々に落としていることがわかる。それまでは、一定速度で巡航しているが、速度を落とす地点が、航空路 (En-route) から標準到着経路へと入域する地点であるため、ここから高度を下げていく。このように速度の傾向として、降下時には多少の分散があるように見えるが、おおむね一定速度の航行の後、線形的に速度を下げていくように見える。他の航空機にも同様の傾向があるため、本研究では入域から線形的に速度が減少していくことを仮定とした。

この時の航行は Δt 毎に位置を記録する。この位置の記録はコンフリクトの検出として利用する。コンフリクトの検出は基本的に、航空機の経路の端点である Waypoint であるノードを通過する時刻、経路である、リンクを入域し、出域する時刻を記録し、ノードとリンクそれぞれで航空機間の距離を算出することによりコンフリクトを検知する [106]。サブルート範囲内においては Δt の位置情報を利用して、コンフリクトの有無を検知する [107]。

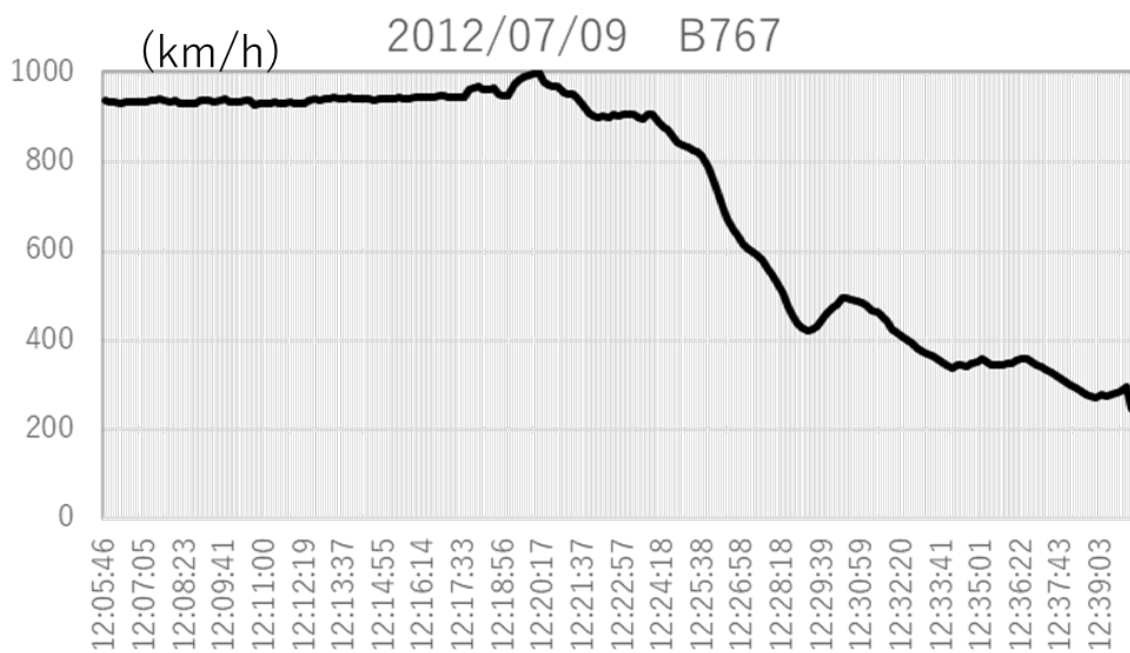


図 4.12 羽田空港の速度の推移

第 5 章

自由経路による着陸順最適化

5.1 アプローチ

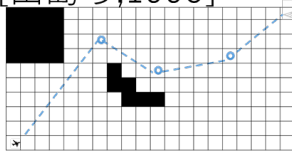
自由経路に基づく経路及び着陸順最適化を行う場合には、大まかに全体を一元的に経路を生成するか、航空が個別に経路の候補を生成するかの二つ考えられる。全体としての方法論は全ての航空機の位置関係や大きさを考慮しながら経路を生成していくため、探索が進むと条件を満たす経路及び着陸順の探索が可能だが、膨大な探索を伴うことが考えられる。そのため、全体の経路を生成する手法は探索空間の削減のため、経路を簡素化するアプローチをとることにより、経路の柔軟性が低くなる可能性がある。そこで、本研究では、個体である航空機が個々に経路を生成し、この経路の組み合わせを用いて、適切な着陸間隔を満たす経路を選択する。個別に経路生成を行う利点としては、経路選択の自由度を高められる点にある。仮に選択された経路に対して変更の要求がある場合には制約を満たす経路の組み合わせがあれば、変更が可能である。また、航空機の航行は混雑時ほど予定した巡航を行わない場合や、突然の気象変化により経路変更を余儀なくされる可能性があるため、修正すべき航空機のみ経路を生成し直すことでこのような場合にも対応が可能である。

図 5.1 は位置づけとアプローチを表す。自由経路における手法として求められる要件は複数の航空機の経路生成の方法論であり、航空管制官が運用可能な方法論である。そこで、本研究では表内の左下に分類されている、田島らの手法を基本アルゴリズムとする。田島らの手法 [30] は後に詳細にアルゴリズムの説明をしていくが、田島らの手法を採用する理由としては、入域地点と、最終進入点の間に Waypoint を生成し、これを結ぶ形で経路を生成する。そのため、Waypoint ごとに方位、速度の指示が可能でそこから直進するため、管制官の基本的なオペレーションであるベクタリングが可能である。ただし、田島らの手法は 1 機の航空機の経路生成手法であるため、複数機の経路を生成し、着陸順として適用した場合には、適切な間隔を保てないため、これを解決するために 1 機の経路多様化することで解決を目指す。

管制官	航空機数	単	複
	最適化 <small>航空機の制約</small>	単機の経路 (経路生成のみ)	複数機の経路 (経路+着陸順)
運用不可	連続 (理想的)	[Miyazawa et al., 2013] [Cárdenas et al., 2014]	[Dougui et al., 2013] [Mori et al., 2015] [Toratani et al., 2015]
運用可	離散 (実用的)	[Takeichi et al., 2018] [田島ら, 1995] [Dey et al., 2014]	本研究①

アプローチ

[田島ら, 1995]



+ 複数航空機の合流 (他機の影響)

- 離散値による経路生成が可能
(ベクタリング可能)
- 進化計算を用いた経路探索

図 5.1 自由経路におけるアプローチ

図 5.2 に提案手法の概要を示す。図 5.2 の下側が経路最適化部（個体（＝各航空機の経路）の最適化を担当）を表し、上側が着陸順最適化部（複数の個体（＝複数の航空機の経路集合＝群）の最適化を担当）となっており、提案手法はこの2つから構成される。この図において、各航空機の地図上にある線の一つ一つが経路の候補を表しており、経路計画では各航空機が独立に経路の候補を X 個生成し、生成された X 個の解から上位 Y 個（経路長の短いものから Y 個）の候補を選択する。その後、到着順最適化では複数個体の Y 個の経路の候補から適切な経路の組み合わせ（群に相当）を探索し、着陸間隔の制約を満たす適切な着陸順を最適化することにより、全ての航空機の経路および到着順を決定する。

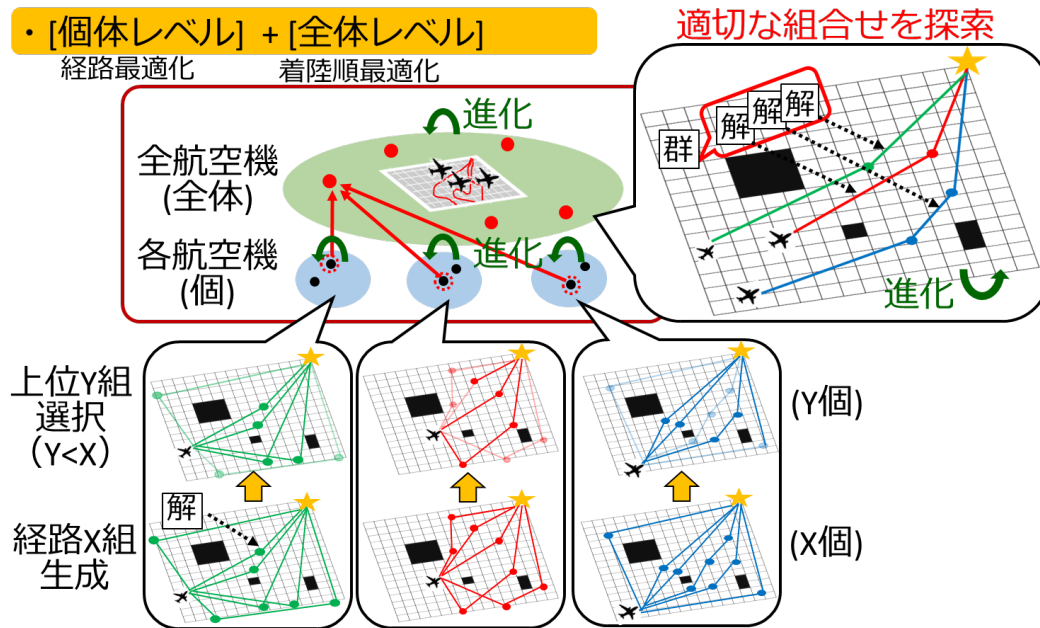


図 5.2 自由経路における経路・着陸順最適化の概要

5.2.1 節では従来手法である経路計画手法を説明する。

5.2 経路・着陸順同時最適化手法

5.2.1 経路最適化（個体レベル）

古典的な探索方法では通常メッシュ状（均等あるいは階段層的）に環境を分割して隣接するメッシュへの移動を動作要素（primitive action）として定義し、これらの動作要素の連続系列により経路を表現する。そして、A*などの古典的な探索法を用いて最適な動作系列を探索する [108]。このような形式では経路の折曲りを最適性評価に考慮しない表現方法では得られる経路はジグザグとなる。

この欠点を解消するために現在位置と目的地点とを結ぶ複数の極力少ない個数の中継点（Waypoint）により経路を表現する。航空機は悪天候の空域や自衛隊の訓練空域などの侵入禁止空域を避けて航行しなければならない。これに対応する手法として経路を探索する従来手法にGA（Genetic Algorithm）を用いた手法がある。GA に関しては第3章で詳細を説明する。GA はいくつかの特徴を持つが、魅力的な1つの特徴は探索過程で常に何らかの解を出力し、その解が何世代かの自然淘汰を繰り返すにより、順最適解に収束していくことである。すなわち、与えられた猶予時間を最大限利用し、与えられた時間が短い場合も何らかの解を算出し、長い場合には進化を続け、良質な解を算出する。このように長い時間を費やせば、ゴールへの解への充足性が高まり、短い間でも何らかの解を算出することから、リアルタイム性での利点があるといえる。経路は航空機が通過する座標

(waypoint) の組み合わせにより決定し、GA の遺伝子構造は図 5.3 のように表現される。遺伝子座は 1 つの waypoint に対応し、括弧内の数値は地図上の x, y 座標を表す。遺伝子張は可変であり、長いほど、方向転換が多い経路となる。

Waypoint	1	2	3	4	5
	(0,3)	(1,5)	(5,10)	(x,y)	(15,15)

図 5.3 経路最適化手法における遺伝子構造

経路最適化アルゴリズム

経路は GA のアルゴリズムに準拠して生成される。以下にそのステップを示す。

Algorithm 3 は経路の最適化のアルゴリズムのステップを示している。個体は 1 つの経

Algorithm 3 経路最適化のアルゴリズム

- 1: 全個体を初期化
 - 2: 全個体の評価値を算出
 - 3: **for** i=1 から全個体分 **do**
 - 4: 交叉により子個体を生成評価された経路案の中から、その評価値に応じた確率により任意に二つのものを選択する。それらの経路案の情報を交叉させ。新たな経路案を複数個生成し、経路案プールに蓄積する。
 - 5: 突然変異を適用
生成された経路案を適当な頻度でランダムに新たな中継点を追加する。
 - 6: 子個体の評価値を算出
 - 7: 親個体より子個体が優れていれば入れ替え
 - 8: **end for**
 - 9: 終了条件を満たさなければ 3. に戻る
-

路を表す。GA は一番初めに解の初期化を行うが、経路最適化手法も解の初期化を行う。初期の解は可変長の遺伝子がすべて一つずつ格納されている。その後、個体毎の経路の評価を行うが、経路の長さが短いほど高い評価値を得るが、航行禁止エリアに経路が存在している場合には経路の評価値は最悪値となる。

この評価値に基づいて親個体を選択され、4 行目の交叉、5 行目の突然変異を子個体に適用する。その後、子個体の評価を行い、ステップ 3 からステップ 8 までを繰り返す。終了条件は経路案のコストがある閾値以下であることや、ある世代数を越えた場合とすることが多い。終了地点で経路プールの中の経路案の中で、最も評価値の高い解として出力する。

交叉方式

Algorithm 3 のステップ 4 に対応する交叉について説明する。2つの個体をその評価値に応じて選択し、それらの特徴を継承した個体を生成する。以下に示す3つの方式を2つに個体の中継点の個数に応じて使い分けて用いる。

1. 中継点平均方式

それぞれの経路案からランダムに一つずつの中継点間を取得し、その位置的な平均をとった中間点を新規中継点とする方式

2. 中継点連結方式

二つの経路案に存在する中継点を全て連結していく方式

3. 中継点組み換え方式

二つの経路案を複数個（各経路における分割数が同じ）に分割し、分割された経路を組み替えて新規経路を生成する方式

なお、選択した二つの経路案を Parent-1, Parent-2 とすると、個数に応じての関係は次のようになる。

1. 中継点の個数が Parent-1, Parent-2 のいずれか、あるいはいずれもが1個の場合は中継点平均方式を用いる。
2. 中継点の個数が Parent-1, Parent-2 いずれもが複数個保持している場合には中継点組み換え方式を用いる。

これらをまとめたものを表 5.1 中継点の個数と交叉の関係として示す。

表 5.1 中継点の個数

Parent-1中継点の個数	Parent-2中継点の個数	交叉方式
1	1	(1),(2)
1	n(>1)	
n(>1)	1	
m(>1)	n(>1)	(3)

中継点平均方式

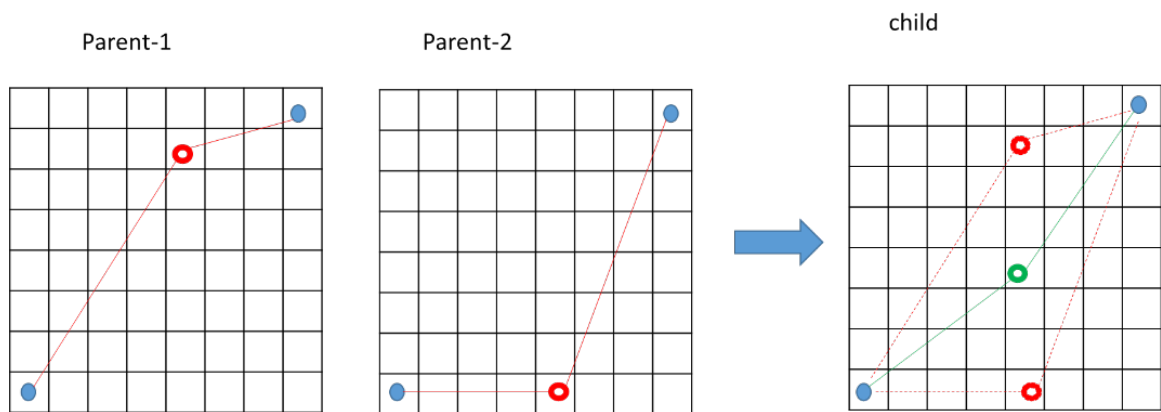


図 5.4 中継点平均方式

中継点連結方式

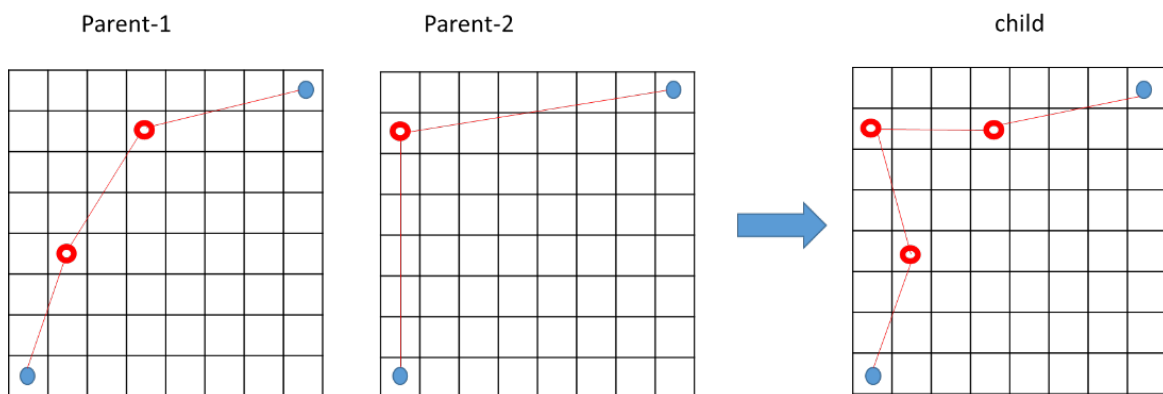


図 5.5 中継点連結方式

突然変異の方式

Algorithm 3 のステップ 5 に対応する突然変異について説明する。ある確率 (突然変異確率) で個体 (中継点) の一部の値を変化させる。局所解回避の効果をもつ。以下に示すような方式をランダムに用いる。

1. 中継点追加：中継点を一定確率でランダムな位置に一つ追加する
2. 中継点移動：中継点を一定確率でランダムな位置に一つ移動する
3. 中継点削除：中継点を一定確率で一つ削除する

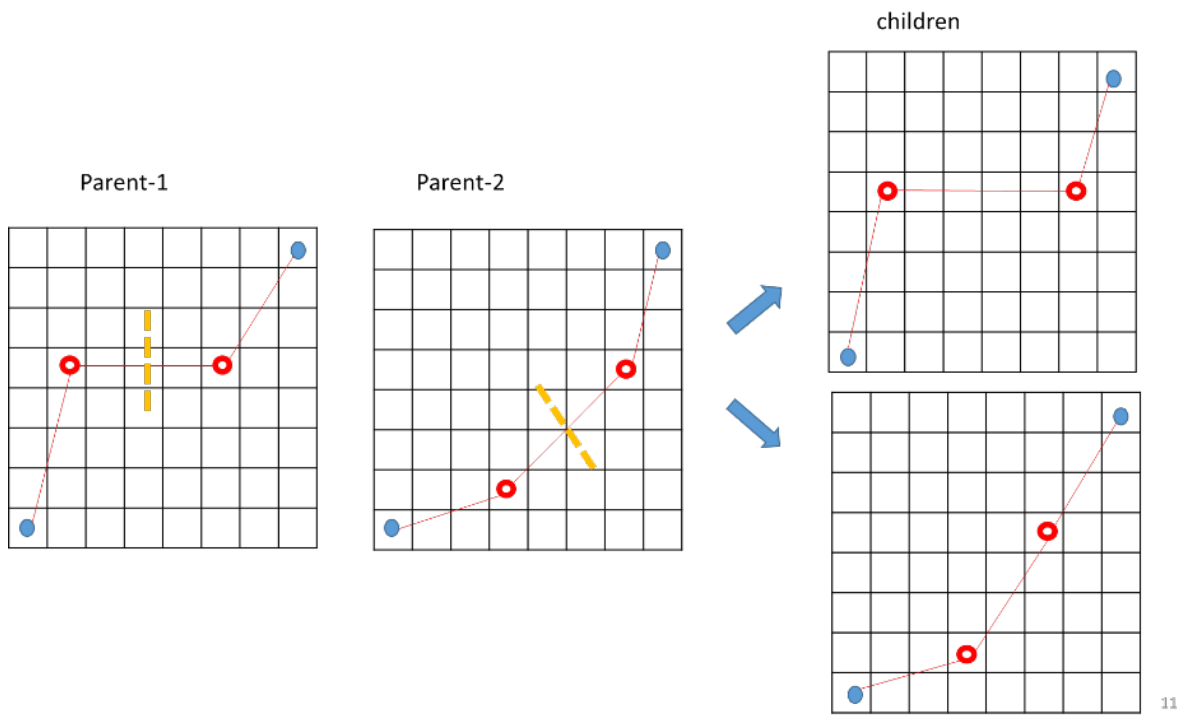


図 5.6 中継点組み換え方式

中継点追加



図 5.7 中継点追加

中継点移動

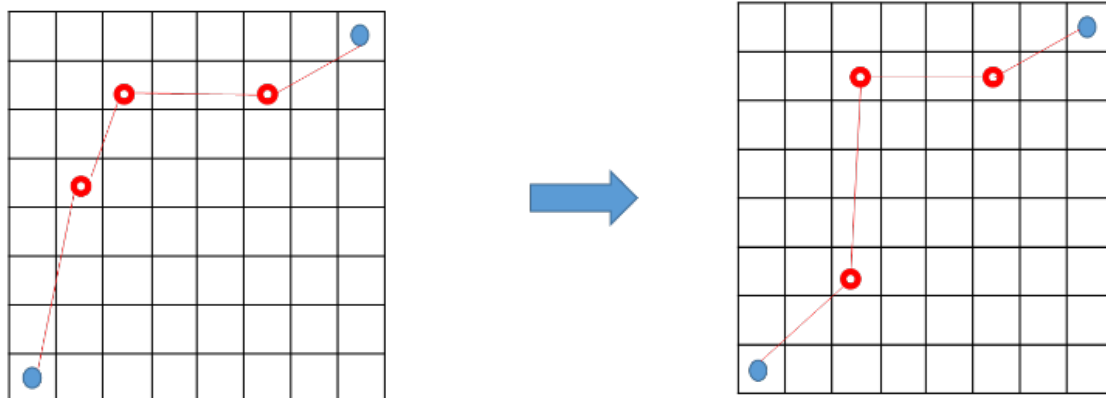


图 5.8 中継点移動

中継点削除

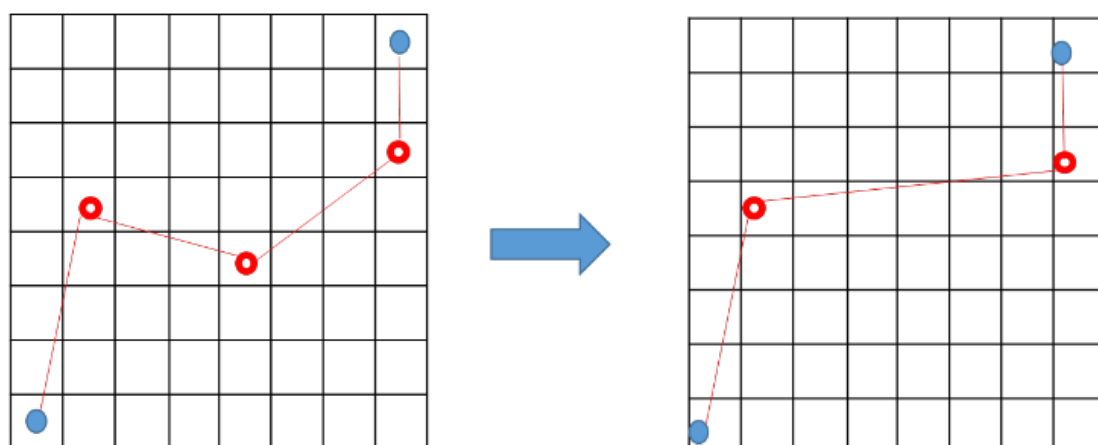


图 5.9 中継点削除

評価基準

経路のコスト算出するために、経路プランニングに関する区域を均一の正方形のメッシュに分割する。各メッシュには該当地点のコストが割り振られる。各中継点はメッシュを最小単位として認定する。各経路案の評価値は、経路の移動コストと中継点の個数を評価項目とし、それらの総和とする。

5.2.2 着陸順最適化（全体レベル）

図 5.10 に提案手法の概要を示す。図 5.10 の左側が経路最適化部（個体（＝各航空機の経路）の最適化を担当）を表し、下側が着陸順最適化部（複数の個体（＝複数の航空機の経路集合＝群）の最適化を担当）となっており、提案手法はこの2つから構成される。この図において、各航空機の地図上にある線の一つ一つが経路の候補を表しており、経路計画では各航空機が独立に経路の候補を X 個生成し、生成された X 個の解から上位 Y 個（経路長の短いものから Y 個）の候補を選択する。その後、到着順最適化では複数個体の Y 個の経路の候補から適切な経路の組み合わせ（群に相当）を探索し、着陸間隔の制約を満たす適切な着陸順を最適化することにより、全ての航空機の経路および到着順を決定する。

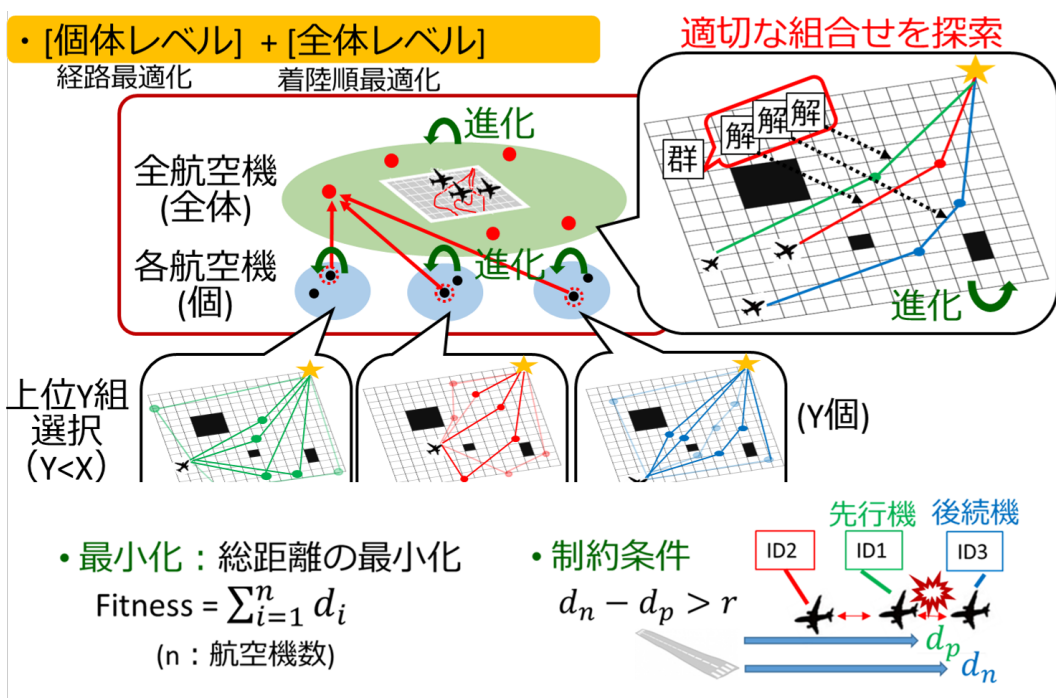


図 5.10 着陸順最適化部（全体）

遺伝子表現

着陸順における遺伝子表現を以下の図 5.11 に示す。遺伝子内には整数の値が格納されており、遺伝子における遺伝子座は各航空機の経路候補の識別番号に対応している。図 5.11 の最上位に位置する遺伝子を例に説明をすると、左から ID 1 から ID 5 の航空機の経路候補が格納されている。ID 1 は経路の候補の中の配列 1 の候補、ID 2 も配列 1 の候補、ID 3 は配列 2 の候補、ID 4 は配列 3 の候補、最後に ID 5 は配列 2 の候補を選択しており、この選択された経路の長さに基づき制約条件を見満たした解であるか判定を行い、実行可能解である場合には評価値を算出する。それゆえ、遺伝子の長さが、最適化すべき航空機の総数となる。

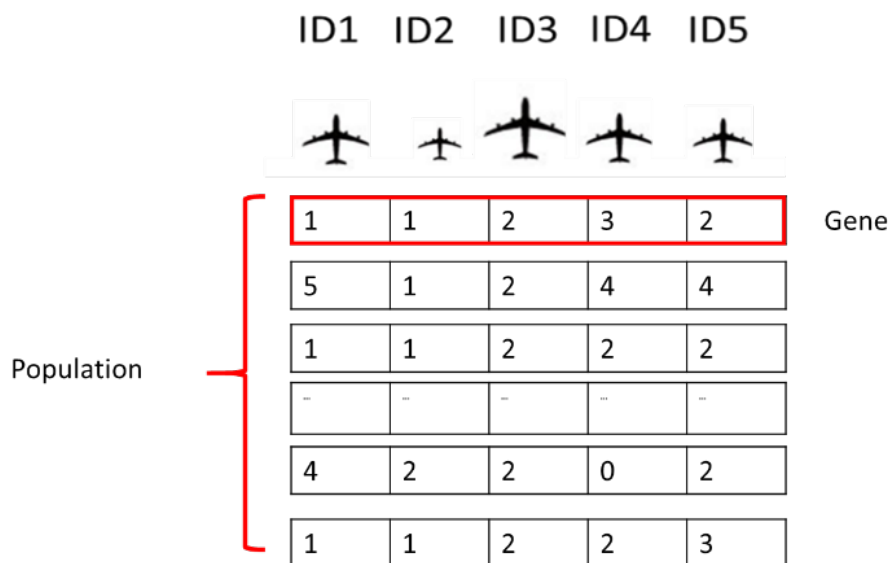


図 5.11 着陸順の遺伝子表現

着陸順最適化アルゴリズム

着陸順生成は GA を用いて解を進化させ、最適解を探索する。以下にその流れを示す。

Algorithm 4 着陸順生成アルゴリズム

- 1: 全個体を初期化
 - 2: 全個体の評価値を算出
 - 3: **for** $i=1$ から全個体分 **do**
 - 4: 交叉により子個体を生成評価された着陸順の案の中から, その評価値に応じた確率により任意に二つのものを選択する. それらの着陸順の情報を交叉させ. 新たな着陸順を複数個生成し, 着陸順プールに蓄積する.
 - 5: 突然変異を適用
生成された着陸順のある航空機の経路を適当な頻度でランダムに別の経路候補に変更する.
 - 6: 子個体の評価値を算出
 - 7: 親個体より子個体が優れていれば入れ替え
 - 8: **end for**
 - 9: 終了条件を満たさなければ 3. に戻る
-

経路未決定時における経路選択

着陸順を生成するために各航空機で生成された経路群から経路選択を行う. 経路が未決定の時には経路群から経路が選び出され, 着陸順を決定する.

経路決定後の着陸経路

着陸順が決定し, 経路選択された航空機に対しては経路群の中に格納された多重経路を利用し, 着陸順を再度決定する. 経路決定し, その経路を航行する他の経路は航行不可能となり, 経路が固定されるため, 柔軟な着陸順生成が難しくなる. そのため, 経路決定後は多重化された経路を航行する経路の候補として経路選択を行う.

制約条件

複数の航空機の経路群が生成された後, 着陸の順番を決定する. 順番を決める際に目的空港までの距離のみで評価をしてしまうと, 航空機同士の間隔が狭くなり空港に着陸できない状況が生じる. そこで, 着陸順が実行不可能解とならないように制約条件として, 以下の航空機同士の距離間隔を条件として付加する.

$$d_n - d_p > r \quad (5.1)$$

式 (5.1) は先行する航空機とその後ろを航行する航空機の距離間隔の関係性を示している. d_p はその前を航行する航空機における目的空港までの距離であり, d_n は次に着陸する航空機の目的地までの距離を表している. 先行機と後続機の着陸空港までの距離関係は常

に $d_n - d_p > 0$ が成立する. この関係が r 以上の間隔がなければ後方乱気流の影響で航行に支障をきたす恐れがあるのでこの条件を満たすように経路選択を行う. 着陸順は GA を用いることにより評価値を基に着陸順を生成する.

評価基準

着陸順の評価は, 航空機同士に優劣の関係を付けず, 全ての航空機が対等な関係で着陸順の最適化を行う. この時, 着陸順の最適性は全ての航空機の出来るだけ早く到着するため, 各航空機の経路の合計により評価する. 評価式を以下示す.

$$Fitness = \sum_{i=1}^n d_i \quad (5.2)$$

式 (5.2) は, 順番付けされる航空機が n 機存在するとき, 各航空機の目的空港までの距離の総和を示している. d_i は i 番目の航空機の経路長を表しており各航空機の総和が Fitness として表される.

自由経路における経路・着陸順に関する全体のアルゴリズム

以下にアルゴリズムを記述する. まず 1 行目と 2 行目は一機の航空機の経路の初期集団の初期化を行い, 3 行目からは 1 機の航空機の経路の進化を行う.

M, T はそれぞれ航空機数と世代数を表す.

4 行目は 1 機の航空機の経路の集団を表し, その経路の集団の中から親選択を行い, 交叉, 突然変異を一定の確率に基づいて行い, 子を生成する.

5 行目は 4 行目で生成した子である経路の経路長と経路の評価を算出する. これを世代 T 行い, M 回 (航空機数分) 繰り返す (6 行目から 9 行目までに対応).

10 行目からは着陸順最適化のアルゴリズムを表しており, 経路の組み合わせを探索するための GA である初期母集団 P'_0 を用意する.

12 行目は親個体を選択し, 交叉, 突然変異を一定の確率に基づいて行い, 13 行目で着陸順の評価値を算出する.

14 行目では着陸順の評価値に基づいて新たな親集団を生成, T' まで繰り返す.

Algorithm 5 経路・着陸順アルゴリズム

```
1: for  $m \leftarrow 0$  to  $M - 1$ 
2: 初期母集団  $P_0$  の生成
3: for  $t \leftarrow 0$  to  $T - 1$ 
4: 親集団  $P_t$  から親個体を選択し、交叉、突然変異を実行し子集団の  $Q_t$  を生成する.
5: 子集団と親集団の評価値を算出し、評価に基づき解をソートし、新たな親集団  $P_{t+1}$ 
   を選択する.
6:  $t = t + 1$  とする.
7: end for
8:  $m = m + 1$  とする.
9: end for
10: 全体最適化のため初期母集団  $P'_0$  を生成する.
11: for  $t' \leftarrow 0$  to  $T' - 1$ 
12: 親集団  $P_t$  から親個体を選択し、交叉、突然変異を実行し子個体を生成する.
13: 子集団の評価値を算出
14: 親個体より子個体の評価値が高ければ入れ替えを行う.
15:  $t' = t' + 1$  とする.
16: end for
```

5.3 最適性と多様性のトレードオフを考慮した経路・着陸順 同時最適化手法

最適性と多様性を同じ割合で探索すると、短い経路長を見出すことが困難となる。なぜなら、ノベルティは特性として評価される解が他の解と比べて離れれば離れるほど評価が高くなるため、より長い経路長が多様性としての評価が高まり、そのような経路が生成され易くなるからである。この問題を解決するために、(1) 評価値の高い解の近傍を重点的に探索する重点サンプリングと (2) 評価値の高い解から遠い範囲を探索しない多様性制限を提案する。

図 5.12 はノベルティおよび経路長の評価の関係を示しており、解の関係性は NSGA-II の支配関係に基づいてトレードオフの関係となっている。この多様性の評価は 3 章に基づいた評価であるが、従来のノベルティサーチの評価は空間の網羅性を加味した評価となる。本研究ではノベルティサーチの空間の距離関係ではなく経路長の差分により評価を行う。経路長の差分を多様性として評価する理由としては本研究で対象とする問題が経路に基づいて着陸順を決定するため、経路長の干渉が問題となる。干渉を解決するためには干

渉を回避するための適切な間隔を保てるような経路長が必要になる。このため、経路長の差を評価することで一定の間隔を保った解の生成が可能となる。

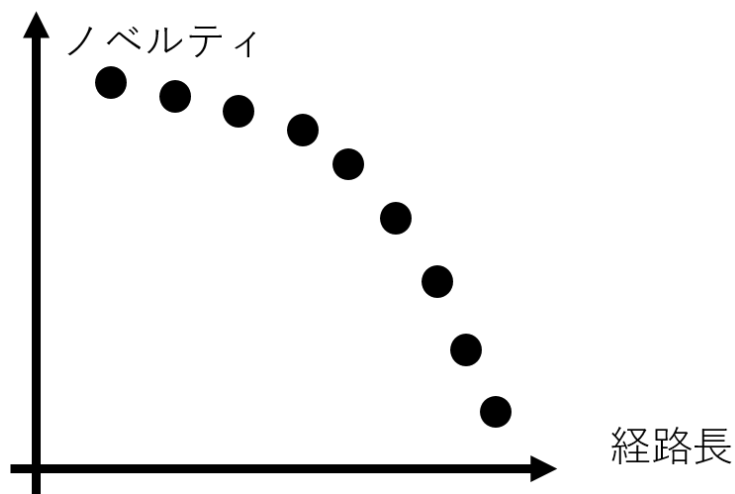


図 5.12 多様性と経路長の 2 目的最適化

5.3.1 ノベルティに基づく多様性の導入

航空機の経路は、田島らの経路計画手法と同様に GA を用いて最適化する [30]。具体的には、航空機の経路を表す個体は遺伝子座にウェイポイントと呼ばれる方向変換点から構成され、現在地からウェイポイントと目的空港までを結ぶことで 1 つの経路が生成される (なお、ウェイポイントの数は決まっていないので、可変長の個体となる)。このとき、最適性としての最短経路だけでなく、多様性としての異なる経路を見出すために、多目的最適化手法の一つである NSGA-II を採用し、図 5.13 (右) に示すような経路長とノベルティの 2 つの目的関数をもとに、航空機の経路を進化させる。なお、図 5.13 は各経路 (同図 (左)) とその評価 (同図 (右)) の対応関係を同じ色で示している。

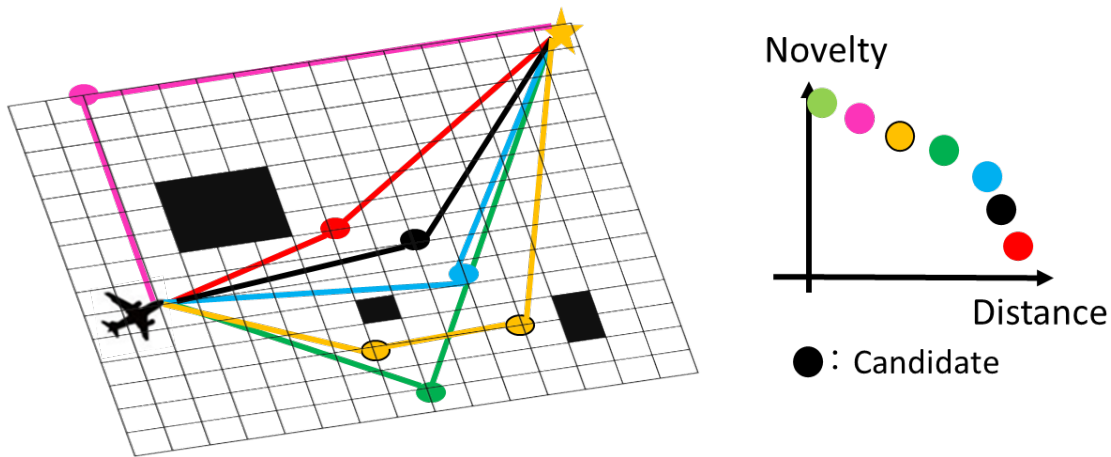


図 5.13 経路とその評価の関係性

本研究で用いるノベルティに基づく評価式を以下に記述する.

$$\rho(x) = \frac{1}{K} \sum_{n=1}^k \text{diff}(x, \mu) \quad (5.3)$$

式 (5.3) における x は評価される経路, μ は他の経路を表しており, $\text{diff}(x, \mu)$ 経路長の差分を求める関数となる. 違いは, $\text{dist}(x, \mu)$ を $\text{diff}(x, \mu)$ と経路最適化用に変更しただけで, 本質的に違いはない. 厳密に言えば, 従来のノベルティサーチの評価である $\text{dist}(x, \mu)$ は空間の網羅性を考慮しているが, 本研究で用いる $\text{diff}(x, \mu)$ は経路長の差分として評価する. この理由としては, 着陸順を決定する際に経路長によって決まる航空機の干渉が問題になるからである. そこで, 経路長の差を評価することで一定の適切な間隔を保てるような経路長を見出し, 干渉を回避する.

経路を選択する際には, 経路最適化によって, 各航空機の経路がパレートフロントとして複数 (図 5.13 では X 個) 生成され, 経路長の短いものからいくつか (図 5.13 では Y 個) が選択される. その後, 先述したようにその経路の組み合わせを複数航空機の経路の集合 (=群) として最適化する.

5.3.2 最適性の高い多様な解獲得へ向けた手法

最適性と多様性を同じ割合で探索すると, 短い経路長を見出しにくくなる. なぜなら, ノベルティは他と比べて離れば離れるほど評価が高くなるため, より長い経路長が多様性としての評価が高まり, そのような経路が生成され安くなるからである. この問題を解決するために, (1) 評価値の高い解の近傍を重点的に探索する重点サンプリングと (2) 評価値の高い解から遠い範囲を探索しない多様性制限を提案する. 図 5.14 は重点サンプリングと多様性制限を合わせたときの効果を表しており, ある範囲に制限した上でその中でも偏りを持った経路候補の生成を可能とする.

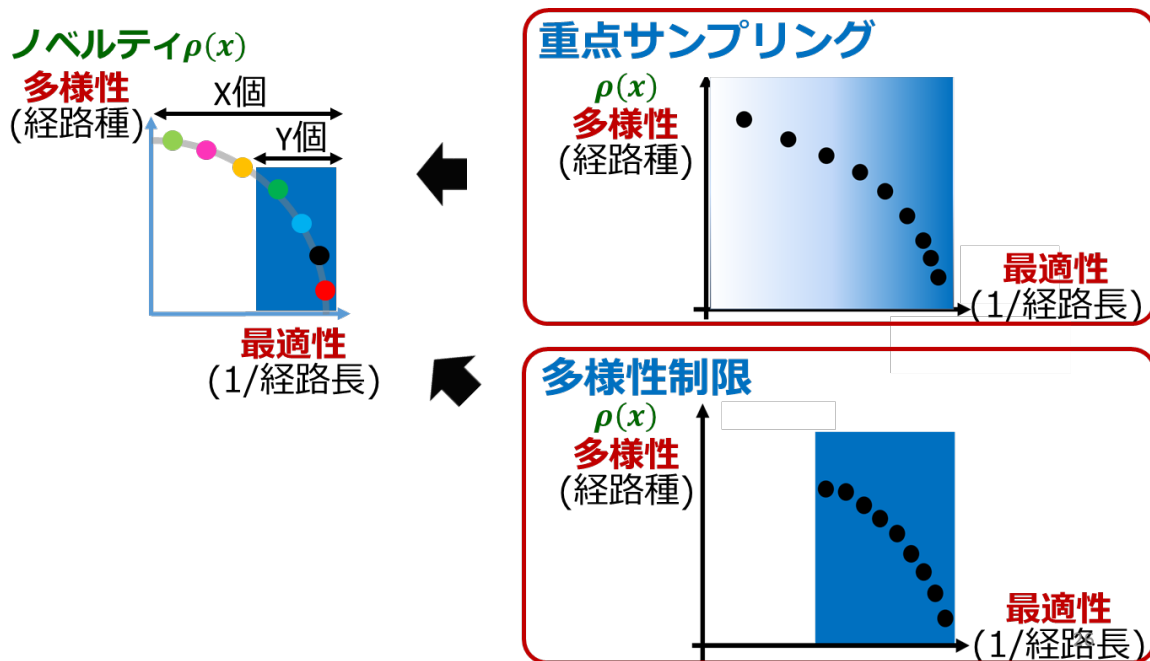


図 5.14 範囲制限と重点サンプリング

重点サンプリング

航空機着陸問題では、やみくもに経路長の長いものを多様性として探索する必要はなく、ある範囲の経路長を重点的に探索する戦略が重要である。航空機の目的としてはできるだけ短い経路長で目的空港に到着することが重要であるため、最短経路に周辺の経路長を探索すべきである。そこで、最短経路を基準として重点的にサンプリングする手法を提案する。

具体的には、図 5.15 に示すように最短経路の重みを最大とし、そこから指数的に重みが変わるようにサンプリングする割合を変更する。重点的に探索するために通常ノベルティの評価としては以下の式 (5.4) を式 (5.5) に導入する。

$$\alpha = W^{\left(\frac{\text{distance}}{d_{\min}} - 1\right)} \quad (5.4)$$

d_{\min} は個体（航空機 1 機）の最短経路であり、 distance は評価される経路長である。 W はパラメータであり、 W の値が大きいほど、重みに対する偏差が大きくなるこのようなサンプリング手法を導入することで、最短からの経路長が短いものを優先的にサンプリングすることが可能となる。図 5.15 を例にとると W が小さい場合に一つ一つの経路候補の間隔が広いが、 W が大きい場合には最短経路周辺で密な経路を分布を保持する。

$$\rho(x) = \frac{1}{K} \sum_{n=1}^k \text{diff}(x, \mu) \alpha \quad (5.5)$$

重点サンプリング：最短の近傍経路長の候補をより密に探索

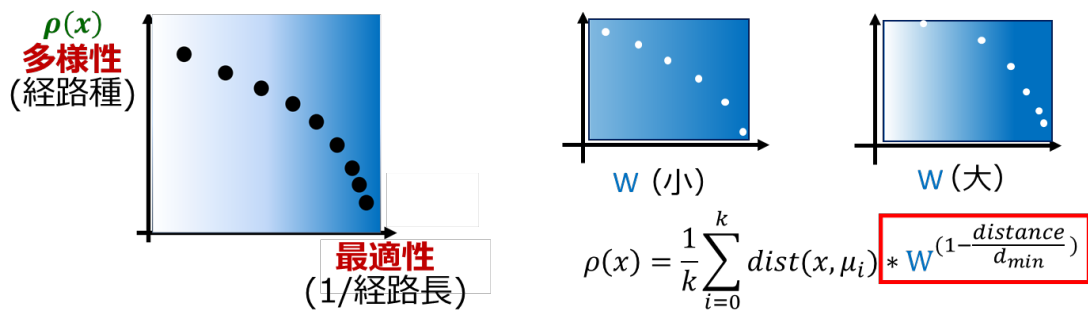


図 5.15 重点サンプリング

多様性制限

複数の航空機の干渉を避けるためには、各航空機は多様な経路が必要であるが、5.3.2で述べたように過剰に多様な経路は最適性の観点から必要ない。特に、提案手法では長い経路のノベルティ評価が高くなるため、経路長を制限することが重要となる。そこで、最短経路から一定以内の経路のみを評価し、その範囲から逸脱した経路は評価しないように制限をする方法を提案する。具体的には、図 5.16 に示すように、最短経路から大幅に外れた経路は探索しないように範囲を制限し、枠で囲った部分の経路を評価するために、下記の式 (5.6) を式 (5.5) に加える。

$$\rho(x) = 0 \text{ if } distance \geq d_{min}\beta \quad (5.6)$$

式 (5.6) は探索すべき経路長の範囲を示しており、最短経路長の長さに依存するパラメータが付加されている。これにより、評価される経路長 ($distance$) が世代毎の最短経路 (d_{min}) の β 以上であれば探索せず、ノベルティの値を 0 とする。図 5.16 を例に説明すると、 β の値が小さい場合には経路長の存在領域を狭め、 β の値が大きい場合には存在領域を広くし、長い経路長を利用し、航空機の混雑した状況に対応する。

重点サンプリング法と範囲制限法を組み合わせた時は上記のようにまとめられる。範囲制限により多様性は維持しつつも最短経路に近い経路を探索させ、さらに重点サンプリングにより最短経路の近傍をより多く探索することで更なる経路の最適化が期待できる。

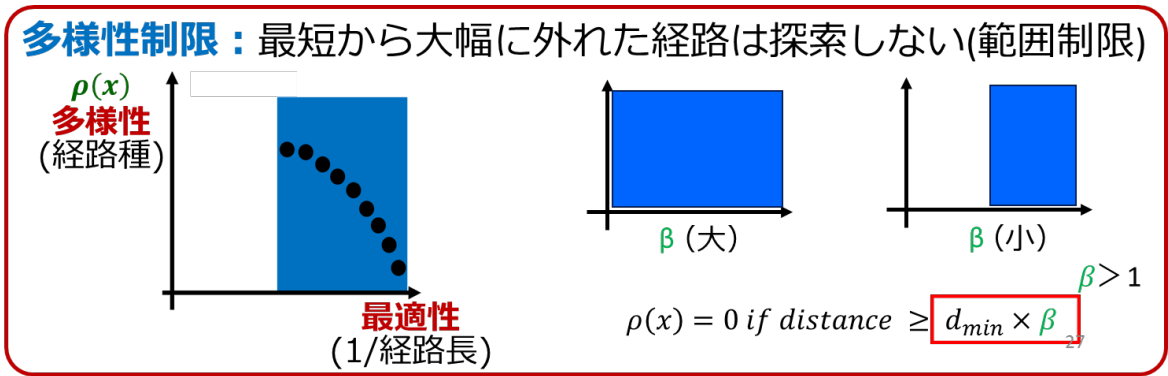


図 5.16 多様性制限

5.3.3 実験：最適性と多様性のトレードオフを考慮した経路・着陸順同時最適化

実験内容

提案手法の有効性を検証するため、羽田空港における航空機着陸問題に適用し、提案した多群探索型進化計算の有効性を検証する。評価基準としては全航空機の総経路長を採用し、生成された着陸順も評価する。比較対象としてはノベルティ単目的最適化、経路長単目的最適化、提案手法(ノベルティおよび経路長の2目的最適化)の3手法を比較する。

下記に示す図 5.17 は比較されるケースを表し、比較するケースにおける評価の有無を示している。最小化が示すのが経路長の最小化を示し、短いほど良い評価を示す。多様性はノベルティに基づく評価であり、この有無からケース分けをした。ケース1は多様性の評価に基づく経路生成法、ケース2は最適性に基づく評価、ケース3は最適性と多様性の両方を考慮した2目的の評価をそれぞれ示す。

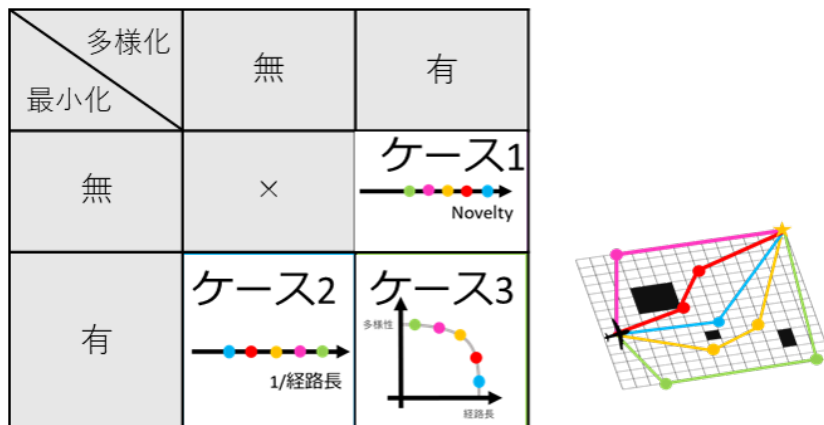


図 5.17 自由経路における比較ケース

羽田空港における着陸問題

本実験では図 4.8 に示すような西側から到着する到着機に着目した航空機着陸問題を扱う。着陸間隔は 9.26 km 以上の間隔を保つ必要があることが決められており、本実験でも同様の設定とする。さらに、羽田空港の最大の到着機数は 1 時間あたりに 30 機程度であり、本実験では 5 分に 1 回最適化を行うことを想定すると、1 回の最適化での出現機数は最大で 3 機となる。図 4.8 の赤枠に航空機をランダムに配置し着陸順の最適化を行う。この環境は 1 時間に 36 機と多くの航空機が目的空港へと着陸する環境である。

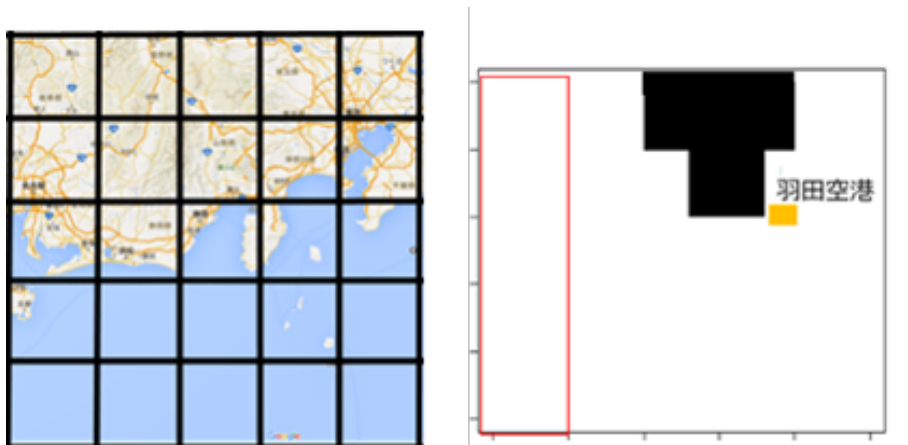


図 5.18 例題：羽田空港（再掲）

評価基準とパラメータ

本実験で用いるパラメータを表 5.2 にまとめる。まず、各航空機の経路最適化に関しては、集団サイズ (population) を 100 とし、世代数 (generation) を 500、交叉率 (crossover rate) を 1.0、突然変異率 (mutation rate) を 0.7、ノベルティの近傍数を 15、重点サンプリングの重みを 10、多様性制限の β をなしとした。着陸順最適化に関しては集団サイズ (population) を 100 とし、世代数 (generation) を 1000、交叉率 (crossover rate) を 1.0、突然変異率 (mutation rate) を 0.3 とした。また、3 手法すべて 25 試行実験し、その汎用的な結果がどうかを調べる。

表 5.2 経路着陸順最適化パラメータ

Path Planning		Arrival Schedule Sequence	
parameter	value	parameter	value
populationsize	50	population size	100
generation	500	generation	1000
crossoverrate	1	crossover rate	1
mutationrate	0.7	mutation rate	0.3
k	15	constraint	9.26
W	10		
β	なし		

結果

表 5.3 は提案手法（ノベルティおよび経路長の 2 目的最適化）、ノベルティ単目的最適化、経路長単目的最適化の 3 手法の総経路を示す。表 5.3 から経路長単目的最適化では実行不可能解が 5 試行で生成されていることから、1 機目と 2 機目の間隔が制約条件を満たすことができないことがあるのに対し、ノベルティ単目的最適化提案手法ではすべて実行可能解であることから、航空機同士が適切な間隔を維持しながら着陸順が生成できていることがわかる。また 25 試行中 11 試行で提案手法が経路長単目的最適化手法より優位である結果が得られ、ノベルティ単目的最適化手法と比較しても 19 試行で有意である結果が得られ、ウィルコクソンの符号順位和検定でも 0.5% の有意水準を得られた。

表 5.3 25 試行での総経路長の比較

	Novelty+ Distance	Distance	Novelty
1	669.40	675.29	669.25
2	611.01	Infeasible	613.44
3	624.12	630.61	636.40
4	682.99	Infeasible	692.90
5	655.91	655.91	657.57
6	561.02	561.59	572.46
7	595.16	598.58	595.16
8	634.51	634.51	637.86
9	636.65	Infeasible	639.26
10	598.71	598.71	600.27
11	604.07	604.33	603.76
12	603.85	Infeasible	607.20
13	545.79	545.79	547.60
14	589.76	589.76	591.11
15	568.34	568.34	571.83
16	655.17	655.17	655.26
17	579.51	579.51	579.53
18	540.69	540.69	540.69
19	592.98	592.98	594.15
20	597.30	597.30	601.08
21	590.65	Infeasible	589.72
22	553.70	548.71	551.83
23	543.78	543.78	551.44
24	720.99	720.99	765.28
25	609.18	609.52	611.48

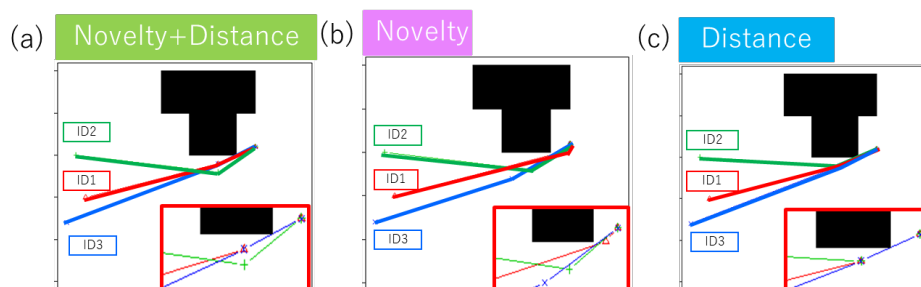


図 5.19 生成された経路

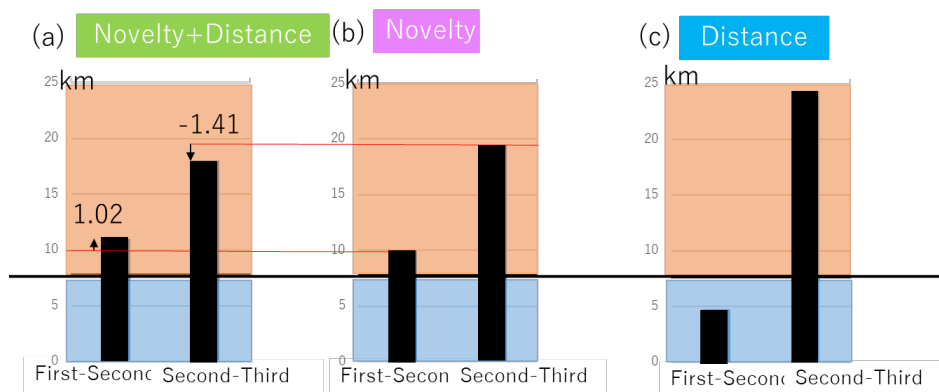


図 5.20 着陸間隔

次に、このような結果が得られた理由を明確にするために、経路長単目的最適化手法にて着陸順が実行不可能解となった 2 試行目に焦点をあてる。図 5.19 は (a) から (c) において生成された経路を示しており、ID 1 から ID 3 の各航空機の経路を示している。マップの上部にある、航行禁止空域を全ての方法において避けて航行しているが、生成されている経路が各手法により異なることがわかる。特に図 5.19 (c) の経路は禁止空域に近い経路を選択し、(a)、(b) はそれよりも遠い経路を選択していることがわかる。特に (a) においては ID 2 が (c) よりも遠い経路を選択しており、(b) においては ID 3 と ID 2 の経路が (c) よりも遠い経路を選択していることがわかる。

別の観点からみると、図 5.20 における (a) から (c) は選択された経路と航空機同士の間隔を縦軸に示しており、横軸に 1 機目と 2 機目の間隔、2 機目と 3 機目の間隔を示し、その間隔の制約である、9.26 km で線を引いて上下で色分けをしている。まず図 5.20 (c) に関しては 1 機目と 2 機目に着陸する航空機の間隔が制約を満たしていないため実行不可能解となった。これは図 5.19 (c) で生成された経路を見てもより経路長が短い経路しか生成されていないことが起因している。次に図 5.19 (a) と (b) の間隔を比較すると 1 機目と 2 機目の間隔は (b) の方が (1.02 km だけ) 短くなっているが、2 機目 3 機目を比べると (a) の方が (1.41 km だけ) 短く、トータルとして提案手法の方が (0.39 km だけ) 着陸間隔が短く、着陸順として最適性の高い解が生成されている。

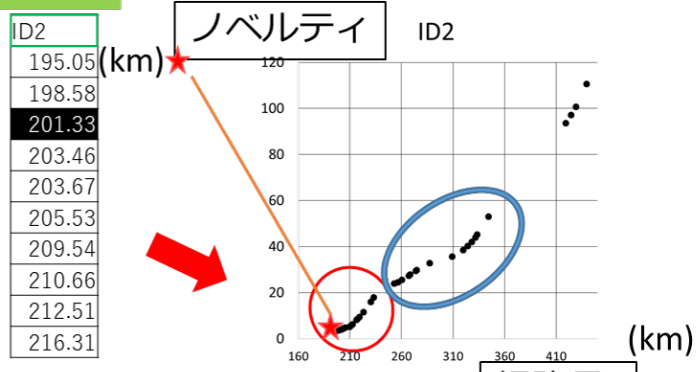
5.3.4 考察

考察 1-1：最適性と多様性を考慮した 2 目的化の効果

本節では 2 目的化により航空機 1 機の獲得された経路の解集団について考察する。

ケース3 Novelty+Distance

最短経路の獲得 →
経路評価による効果



ケース2 Novelty

最短経路の獲得できず →

着陸順として
選択された解

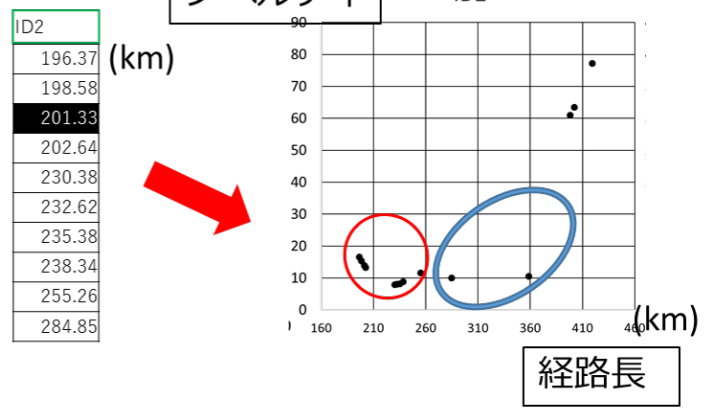


図 5.21 ノベルティを利用した場合の経路の多様性

図 5.21 は同試行での多様性の評価指標であるノベルティをのみを評価した場合の解と経路長とノベルティの2目的化の評価で生成した場合の解集団について表している。上側の解集団は2目的で得られた解集団，下側の解集団はノベルティ単目的獲得した解集団をそれぞれ表す。右側の縦軸はノベルティの値，横軸は経路長であり，点一つ一つが，解(経路)を表している。左側の表は経路を短い順に並べた場合の経路の長さ示す。表の一番最上位は最短経路を表しているが，提案手法は最短経路を獲得することが出来ているが，一方でノベルティの単目的に関しては最短経路を獲得できていない。このように2目的にすることで，航空機単体として最適な経路が獲得可能であることがわかる。次に，解集団の違いについて考察する。2目的の解集団の場合は，上位10個の経路の間隔は21 kmで，前後の経路長の間隔は最大で4 km程度と短い経路に関して密に分布されている。次に，ノベルティの解集団に着目すると，上位10個の解集団の間隔は90 km程度と2目的に比べ大きな間隔を持っており，一つの一つの間隔を見ると，上位4つ目までの間隔は2目的と同程度であるが，その次の解は230 kmとその前の解と30 km程度離れている。これ以外にもリストの後ろにいくにつれ，前後の間隔が大きくなっていることがわか

る。右側の解集団全体の分布を見ても同様の傾向がある。赤くマークされている箇所は、比較的、両手法共に密に解集団を獲得できている一方で、ノベルティの単目的のほうは経路長が長くなるにつれ、解集団が疎になるため、一つ一つの経路長の間隔が大きくなるが、2目的で獲得した経路集団は後方でも単目的よりも密に解が分布されていることがわかる。このように経路長の評価とノベルティの2目的で評価すると、多様で密な経路長の獲得が可能であることがわかった。

考察 1-2：重点サンプリング及び範囲制限手法の有効性

本章では、多目的最適化の改良手法の有効性を検証するために、第4章で説明した例題を適用した多目的最適化手法を用いる実験において、異なるパラメータで生成された経路の解集団の比較を本節ではおこなう。重点サンプリングおよび、範囲制限のパラメータを変化させた場合の考察を行う。

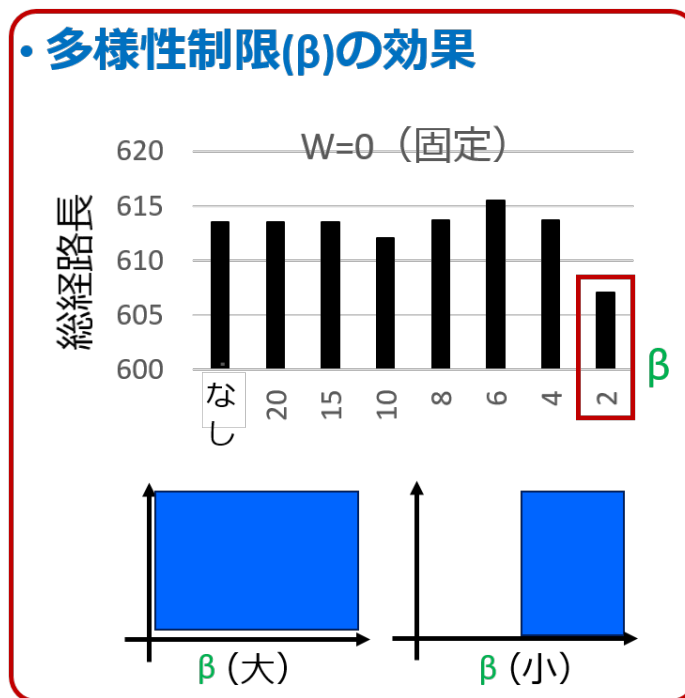


図 5.22 重点サンプリングを使用した場合の結果

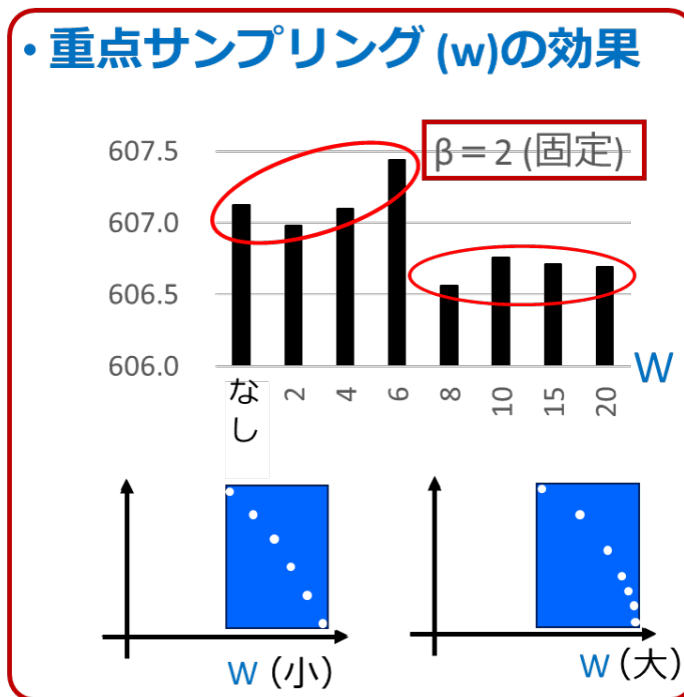


図 5.23 範囲制限を使用した場合の結果

図 5.22 に重点サンプリングを用いた場合の結果を示す。縦軸は全航空機（3 機）の総経路長の 25 試行の平均を表しており、横軸は重点サンプリングを用いた場合のパラメータ W の値を表している。図 5.22 の横軸が 0 は範囲制限および重点サンプリングを使用しない場合での結果を表しており、全てのパラメータにおいて適切な間隔を満たす着陸順が生成されていることがわかる。パラメータが 3 以上である時は全て総経路長の平均 W がなし、2 のときに比べ低いことがわかる。また、 W が 3 以上の場合には総経路長の変化が小さく、パラメータの変化に対しての安定的であると図からわかる。次に、図 5.23 は範囲制限を用いた場合の結果を表しており、縦軸は図 5.22 と同様であり、横軸は範囲制限のパラメータ β の値を表している。図 5.22 に比べ総経路長の分散が大きく、 β が 6 の値の時にはどちらも使用しない場合（横軸が 0 の時）に比べ、総経路長が大きいことがわかる。ただし、 β を下げていくと総経路長が下がっていくことがわかる。これは範囲制限によって解の探索範囲を狭めているため、有効な解の候補を保持できていると考えられる。

次に実験で使用したパラメータにおける β および W を変化させた場合の結果を以下に示す。

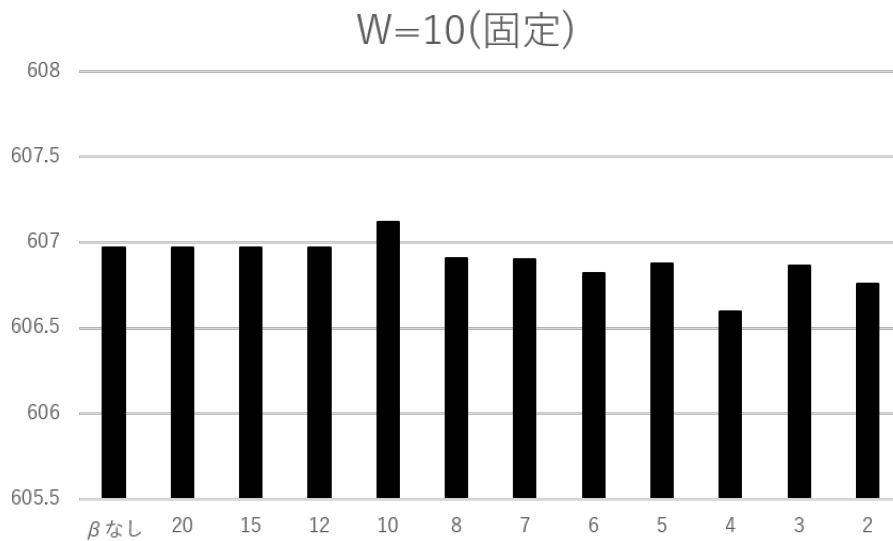


図 5.24 β を変化させたときの総経路長の違い

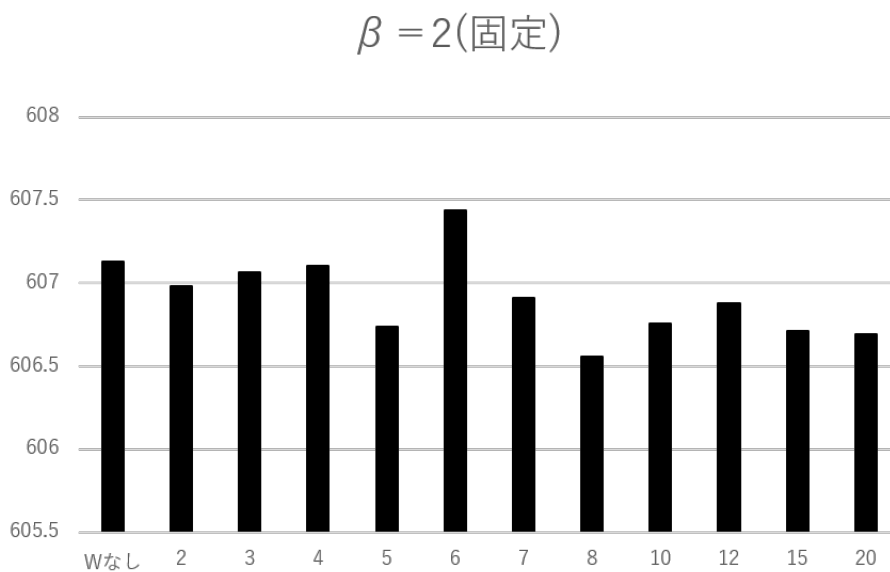


図 5.25 W を変化させたときの総経路長の違い

図 5.24 は W を実験の時と同じ数値の 10 として β を変化させたときの総経路長の変化を表している. W を 10 とした理由は, W の重み付けは指数となっており, 10 も 20 もほぼ同じ重みとなるため 10 を採用した. 縦軸は 25 試行での総経路長の平均を表しており, 横軸は β の値を示している. β を変化させた場合に, β が 10 時を除いては全て

総経路長が減少していることがわかる。特に範囲を 4 以下にした時に効果が表れていることがわかる。

図 5.25 は総経路長が短くなった $\beta = 2$ の場合に W を変化させたときの総経路長の変化を示す。縦軸は 25 試行での総経路長の平均を表しており、横軸は W の値を示している。 β が 6 の時を除いては総経路長が重点サンプリングを使用しない場合に比べて短い総経路長を見出すことに成功していることがわかる。

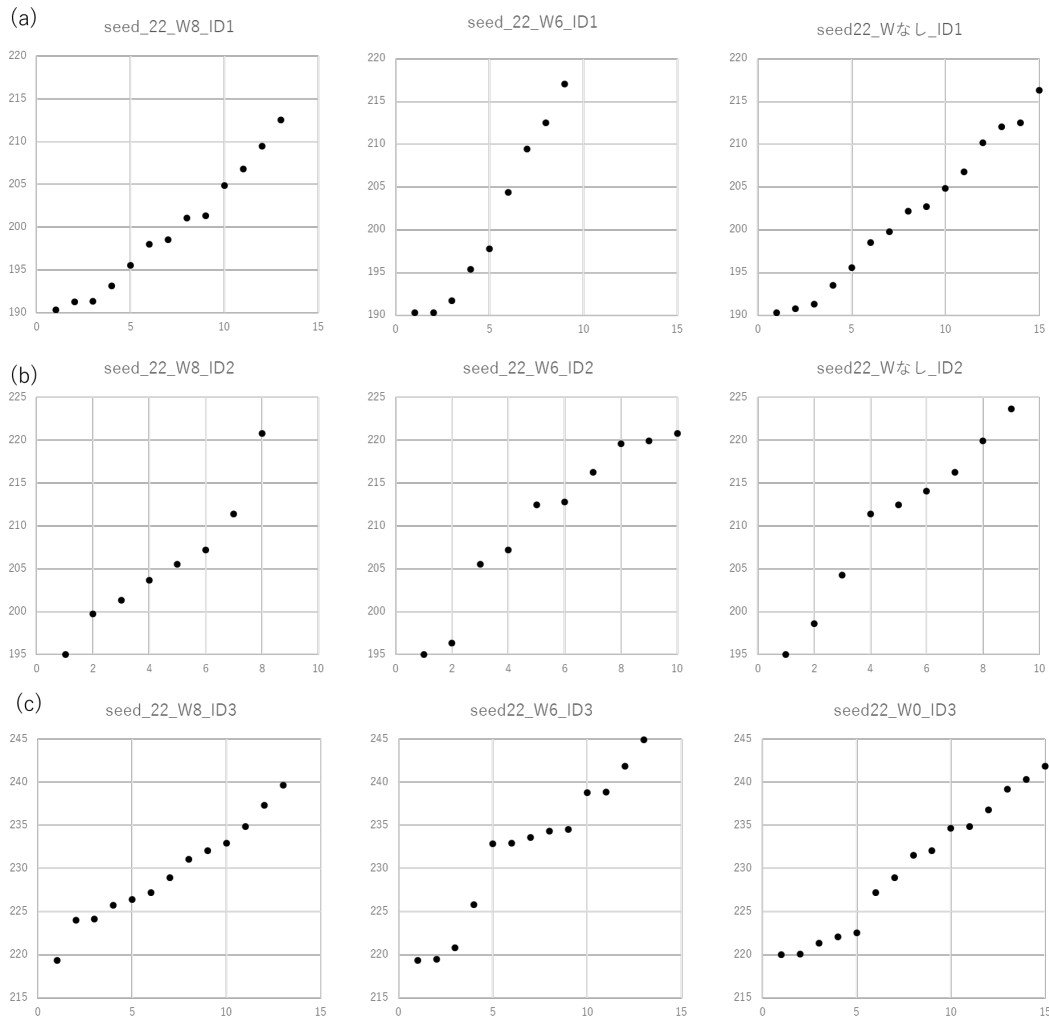


図 5.26 β を変化させたときの経路集団の違い (ID1 から ID3)

図 5.3.4(a) は 2 2 試行目における ID 1 における経路の集団の分布を表している。左から W が 8, 6, なしの順で図示されている。縦軸は経路長を表しており、横軸は経路長を昇順で並べたときの番号を示している。なお、同じ経路長が重複している場合も 1 つとして数え上げている。図 5.3.4(b), 図 5.3.4(c) においても同様であり、それぞれ ID 2, ID 3 の経路の分布を表している。図 5.25 において W が 6 での最短経路近傍での解の分布が疎であることが W が 8 と W なしと比較しても明らかである。特に ID 2 と ID 3 の

経路の分布では顕著に表れており、ID3 においては 5 番目の解と 4 番目の解の偏差が大きいことがわかる。一方で、 W が 8 と W なしにおいては最短経路近傍においての分布が同じようにみえるが、(b) の ID2 においては W が 8 においては密度が W なし比べ密であることがわかる。ただし、 W を大きくすると最短経路近傍において密に解を獲得することが可能になるが、重複する解も増える。 W が 8 において最短経路と重複している解が 50 個中 30 個最短となっていたため、近傍へ集中的に探索する探索圧がかかりすぎるため最短経路長に多くの経路が重複してしまうことが明らかとなった。それにより重複解近傍についてのノベルティの評価が見積もられない可能性があり、これに対する対策が必要になる。以上のことから、重点サンプリングと多様性制限を組み合わせた場合には更に総経路長を短縮することが可能であり、 W を大きくすると解の多様性が損なわれる可能性もあることがわかった。

5.4 実運用での最適化

5.4.1 連続環境下における最適化手法

先ほど説明した経路・着陸順最適化手法は 1 回の経路・最適化手法における運用方法であり、実際の運用を考えると、航空機が連続的に入域する環境に対応しなければならない。特に、航空機は経路を 1 度決定すると、その経路に沿って航行を行ため、混雑状況によっては前のステップで定めた経路によっては次ステップの最適性能が劣化する可能性がある。この問題に対応するため、経路選択後の次ステップ後においても経路の変更が行える迂回経路を付加し、最適化ステップ中では先ほど同様に経路生成及び経路選択を行い、次のステップでは付加された迂回経路が選択肢として追加される。

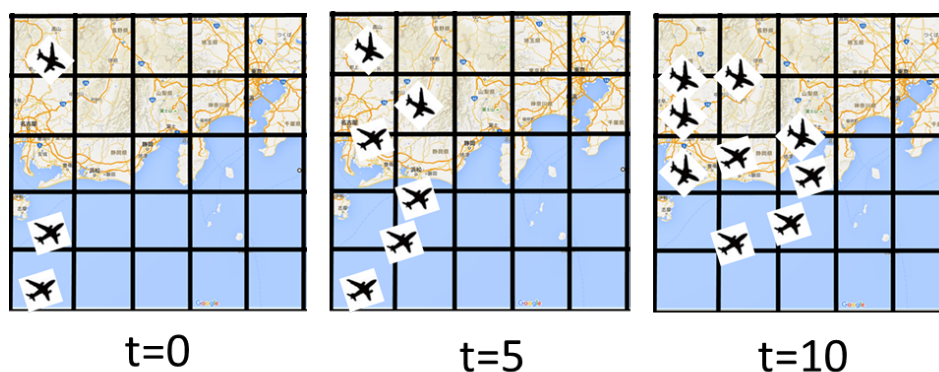


図 5.27 連続環境下における航空機の入域

図 5.27 は連続環境下 (航空機混雑) において、時間推移による最適化対象空域内における入域する航空機の推移をあらわす。左から 0 分、5 分、10 分後の航空機の状態を表

し、このように逐一経路と着陸順の生成する。

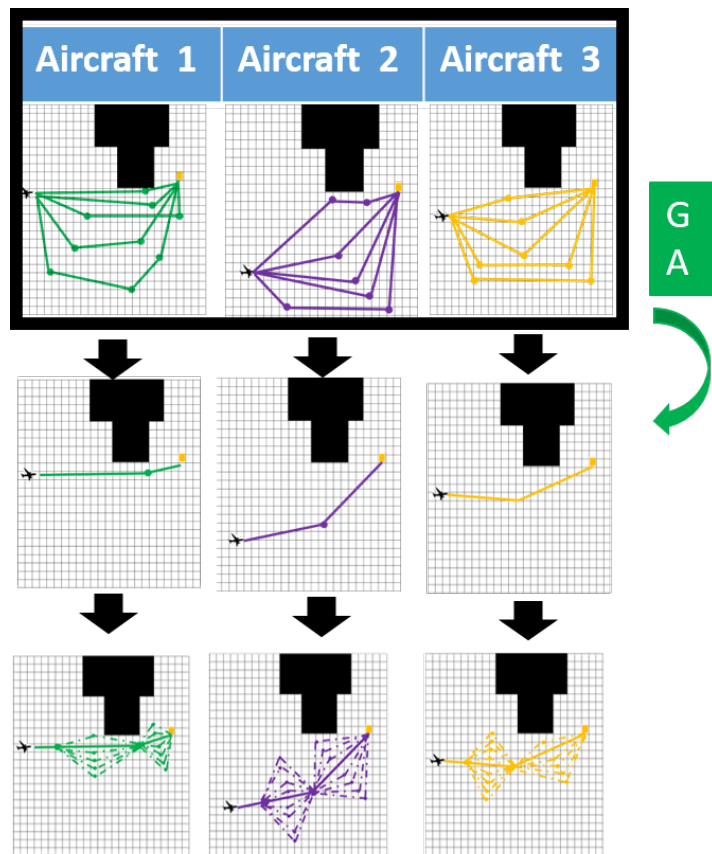


図 5.28 迂回経路の付加方法

図 5.28 は生成のフローのイメージを表しており、はじめに 3 機が入域した最適化の過程の様子を表している。初めに各航空機が経路候補を生成する (図の太枠内に相当)。その後、各航空機の経路の組み合わせにより着陸順を決定する (図内中段に対応)。着陸順を生成した後は決定した経路に迂回経路を付加する (図下段に対応)。各航空機には一つの経路と複数の迂回経路が生成されていることがわかる。

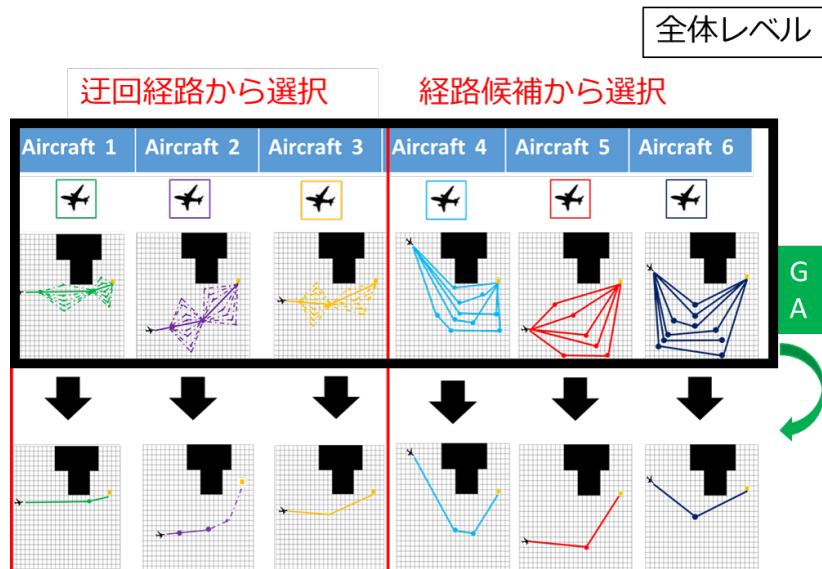


図 5.29 2 回目の最適化における着陸順最適化

図 5.29 は 2 回目の最適化においては、3 機が新たに入域しており、図中の Aircraft 4 から 6 までが該当する。この航空機に対しては通常の経路生成を行った後にこの経路の組み合わせを探索する。Aircraft 1 から 3 に対しては 1 回目の最適化後に生成した迂回経路を生成した後にこの組み合わせを探索することで Aircraft 1 から 6 までの着陸順を決定する。3 回目以降は Aircraft 1 から 3 までの航空機に関しては経路の変更を行わない。次に迂回経路の生成法について説明する。

図 5.30 は迂回経路生成法を示しており、ヒューリスティックなルールに基づいて生成する。迂回経路が付加される場所は次ステップにおいて航空機が巡航する位置座標を視点として、迂回を行う Waypoint を終点とする。新たに追加する Waypoint は視点と終点の midpoint とする。Waypoint の配置は始点と終点を結ぶ直線に垂直かつ midpoint を通る直線上に迂回経路の迂回点が設定される。ただし、迂回経路を利用するタイミングは次回の最適化までは決定した経路に沿って航行するため、現時点を迂回地点とした場合には、迂回時には既に迂回地点を通過する。そのため、現時点から迂回経路を生成する場合には、次回最適化の際の航行地点を迂回点の基準とする。迂回経路の本数としては、航行するための経路を直線が基準の角度となり、これから θ までの角度を最大として迂回経路を等距離に Waypoint を配置する。迂回の本数及び最大 θ 角はパラメータである。

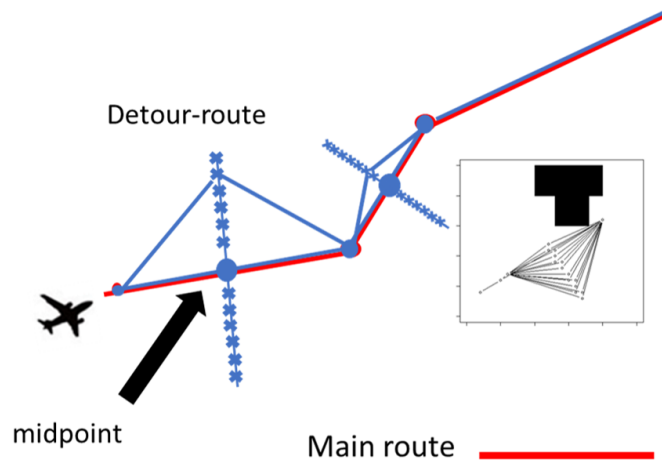


図 5.30 迂回経路の生成法

以下に連続最適化時のアルゴリズムを記載する.

Algorithm 6 自由経路における迂回経路生成を含むアルゴリズム

- 1: **for** $m \leftarrow 0$ to $M - 1$
 - 2: Generate $P_{ind(0)}$ as an Initial Population.
 - 3: **for** $t \leftarrow 0$ to $T - 1$
 - 4: Choose individuals from $P_{ind(t)}$ and run crossover and mutation operators.
 - 5: Evaluate fitness and novelty of all solution
 - 6: Update $P_{ind(t)}$ replace if the individual in child population high evaluation value than the parentpopulation
 - 7: $t = t + 1$
 - 8: **end for**
 - 9: $m = m + 1$
 - 10: **end for**
 - 11: To optimize group (multiple aircrafts), $P_{all(0)}$ as an initial population is generated.
 - 12: **for** $t' \leftarrow 0$ to $T' - 1$
 - 13: Select individuals, run crossover and mutation operators.
 - 14: Evaluate all individuals in child population
 - 15: Update $P_{all(t')}$ replace if the individual in the child population high evaluation value than the parent population
 - 16: $t' = t' + 1$
 - 17: **end for**
-

Algorithm 7 自由経路における連続最適化アルゴリズム

```
1: for  $t_o \leftarrow 0$  to  $T_o - 1$ 
2: (if 航空機は一度経路を決定し かつ 迂回経路を決定していないか?)
3: リストに航空機を加える
4:   else if(航空機はセクタ内に入域しているか?)
5:   経路生成メソッド ()
6:   着陸順生成メソッド ()
7:  $t_o \leftarrow t_o + 1$ 
8: end for
```

5.4.2 実験：連続環境下における手有効性の検証

実験内容

実験は羽田空港を模擬したシミュレーション環境において、連続的に入域する航空機における実行可能な経路生成の可用性の検証を目的とする。提案する最適化手法の最適化のタイミングは5分とした。実験では常時3機が空港の東側から入域することを想定する。3機が入域する箇所については入域範囲からランダムで位置を決定する。初期実験では3機の航空機が入域し、経路・着陸順を最適化が行われることを3度繰り返す。比較対象としては、提案手法である、多様性および最適性の2目的により経路を生成した場合と多様性の単目的により生成した方法、最適性の単目的により生成した方法とする。評価指標は着陸順の評価である航空機の総経路長とした。本実験では3機入域が3回（15分に相当）の最適化における着陸順の性能を検証する。

実験設定

最適化に関わるパラメータは5.4節と同様であるが、再掲する。

表 5.4 パラメータ (再掲)

Path Planning		Arrival Schedule Sequence	
parameter	value	parameter	value
populationsize	50	population size	100
generation	500	generation	1000
crossovertime	1	crossover rate	1
mutationrate	0.7	mutation rate	0.3
k	15	constraint	9.26
W	10		
β	3		

結果

	Novelty+ Distance	Novelty	Distance
1	1116.43	1114.938	Infeasible
2	1032.002	1058.271	Infeasible
3	1094.674	1107.043	Infeasible
4	1173.23	1202.435	Infeasible
5	1125.084	1155.742	Infeasible
6	1005.118	1015.657	1005.689
7	1044.658	1048.837	Infeasible
8	1141.02	1149.008	Infeasible
9	1152.8	1156.018	Infeasible
10	1052.586	1055.607	1056.849
11	1049.572	1051.495	Infeasible
12	1047.554	1056.078	Infeasible
13	997.524	1008.826	997.524
14	1059.067	1067.752	Infeasible
15	1029.767	1036.082	Infeasible
16	1112.088	1111.212	Infeasible
17	1018.98	1026.197	Infeasible
18	993.1018	1005.933	993.1018
19	1103.404	1106.308	1103.404
20	1097.995	1102.969	1097.995
21	1082.359	1098.465	Infeasible
22	933.8621	932.8318	Infeasible
23	1009.831	1017.88	1009.831
24	1178.491	1365.582	Infeasible
25	1046.844	1052.765	Infeasible

図 5.31 2 回目の最適化における Fitness

	Novelty+ Distance	Novelty	Distance
1	1428.59	1431.046	Infeasible
2	1326.05	1349.523	Infeasible
3	1394.07	1415.978	Infeasible
4	1458.31	1523.676	Infeasible
5	1426.83	1477.327	Infeasible
6	1300.34	1313.671	Infeasible
7	1320.44	1370.337	Infeasible
8	1444.83	1477.677	Infeasible
9	1465.76	1481.229	Infeasible
10	1353.62	1364.115	Infeasible
11	1318.35	1319.307	Infeasible
12	1314.43	1328.994	Infeasible
13	1278.422	1311.79	1278.664
14	1358.534	1369.903	Infeasible
15	1355.68	1371.043	Infeasible
16	1382.79	1398.041	Infeasible
17	1399.73	1417.99	Infeasible
18	1349.673	1365.866	Infeasible
19	1406.542	1413.687	Infeasible
20	1484.789	1489.763	1484.789
21	1479.84	1497.015	Infeasible
22	1252.05	1251.893	Infeasible
23	1382.936	1392.363	1382.866
24	1529.12	1777.499	Infeasible
25	1399.14	1403.643	Infeasible

図 5.32 3 回目の最適化における Fitness

図 5.31, 5.32 はそれぞれ、各目的により生成した経路に基づいて生成した着陸順の総経路長を表し、左から提案手法の評価、多様性に基づく評価、最適性に基づく評価をそれぞれ表している。行番号が試行番号に対応し、オレンジの枠が同試行において、総経路長が低い手法を指し、青枠が総経路長が大きい手法を指す。黄色は”Infeasible”と表示され、実行不可能（制約を満たさない）な着陸順を示す。多様性が考慮された経路生成に基づく着陸順最適化は 25 試行全てにおいて 2 回目、3 回目の最適化において実行可能な着陸順を獲得可能であることを示している。一方で最適性のみを考慮した手法では 2 回目の最適化の時点で実行不可能な着陸順が 30 試行中 18 試行において発生しており、3 回目の最適化では 3 試行以外ですべての着陸順が実行不可能な着陸順となっている。提案手法と多様性の単目的のみを考慮した経路生成法を比較すると、2 回目の最適化において、30 試行中 27 試行で性能向上が見られ、3 回目では 29 試行で優位性を示した。このように、最適性と多様性二つを考慮することは、連続的に航空機が入域する環境においては、実行可能な着陸順生成を促し、最適性の高い着陸順が生成可能であることが明らかになった。

考察 2-1：管制官のワークロード低減のために

実験設定

実験は先述した実験環境同様に 5 分に一度の最適化を想定する。各航空機に対しては 2 回の経路選択の機会がある。1 度目は経路多様化手法により生成された経路に関して最適な着陸順を生成するよう経路の組み合わせを選択する。2 度目の経路選択の際には、1 度

目に選択された経路に付加された迂回経路が経路選択の対象となる。ここで、迂回経路を使用する航空機は1度目の経路長よりも長い経路を選択するが、このような長い経路が選択されるのは全ての航空機の経路の組み合わせにより評価値が良くなる場合である。2度目の経路の選択機会を終えた航空機の経路は現在航行する経路に沿って航行する。経路変更の可能性が1度ある場合には航空管制官の指示により経路変更が必要になるため、全航空機のうち、複数の航空機の経路の変更は許容しない、部分的に航空機を固定することで管制官のワークロードの負担と着陸順の最適性のトレードオフとなることを確認する。航空機は5分に3機追加する。

結果

図 5.33 は固定機数を変えた場合の航空機の総経路長を表している。横軸はステップ数で、1回の最適化を表し、縦軸は総経路長 (k m) を表す。1ステップ毎に各航空機は50 k mずつ目的空港へ近づくため、50 k m減少する。すべての航空機を固定した場合は固定しない場合、部分固定 (1ステップ毎に1機を固定) する場合よりも総経路長が増加するため、最適化性能が劣化していることがわかる。また、すべてのステップにおいて実行可能な着陸順の生成が可能であり、迂回経路を利用する場合、総迂回経路の削減が可能であるが、それと同時に迂回により指示回数も増えることがわかる。

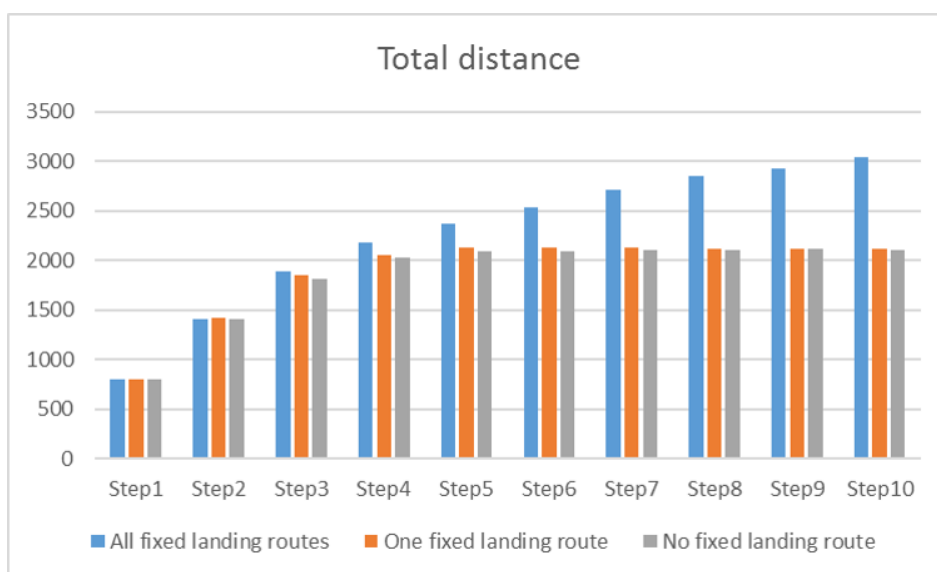


図 5.33 最適化毎の総経路長

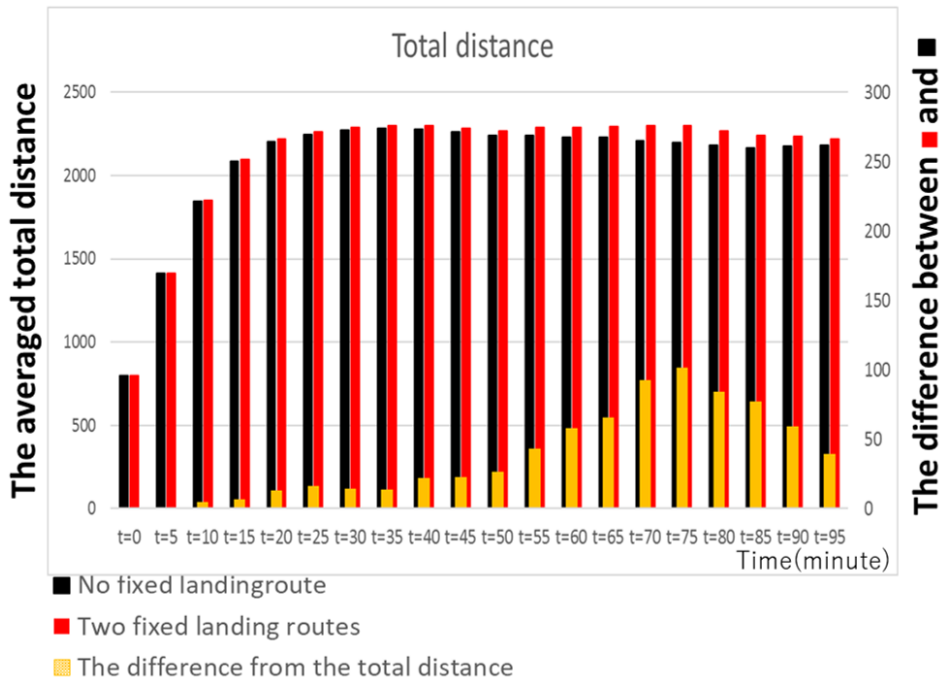


図 5.34 迂回経路の利用を全機許容した場合と 2 機許容した場合の差異

図 5.34 は 2 機固定した場合と全航空機の迂回指示を許容した場合の差を表している。横軸は先ほど同様に 5 分を 1 ステップとした場合の総経路長の比較を行っており、赤が 2 機を迂回を許容した場合の総経路長、黒は全航空機の迂回を許容した場合の総経路長の合計を表している。黄色のグラフは赤と黒の差分の距離を棒グラフとして表している。このように、迂回経路は総経路長減少に貢献することがこのグラフからわかる。毎ステップ、航空機を追加することで、6 ステップまでは全体の総経路長は増加しているが、その後、いくつかの航空機は着陸し、全体の航空機数がステップで大まかに同じ機数となるため、総経路長がステップの径かにより変化があまりないことがわかる。ただし $t = 75$ の時に、二つの差が最大となっているが、100 km 程度の差であり、総経路長の割合に比べ 5 % 程度である。

表 5.5 各最適化における迂回経路の利用回数

Time	One fixed landing routes	No fixed landing route
t= 0	0	0
t= 5	0	0
t= 10	0	0
t= 15	0	1
t= 20	0	3
t= 25	0	0
t= 30	1	2
t= 35	0	0
t= 40	1	3
t= 45	0	0
total	2	9

表 5.5 は使用した迂回経路の回数を示している。total は $t = 0$ から $t = 45$ までの合計を $t = 45$ までの迂回経路利用回数はすべての航空機に対し迂回を許容した場合は 9 回に対し、一機のみ迂回を許容した場合には 2 回となっており、大幅に削減可能である。これは迂回経路の利用回数を調整することで航空管制官への負荷を調整可能であることを示唆している。

考察 2-2:連続最適化における適用範囲の検証

本章では提案手法の実用の可能性を検討するため、ステップ数を増加した時においても提案手法が適用可能であるかを検証する。本章での実験はステップ毎に固定数の航空機を追加し、ステップ毎で実行可能な経路生成および着陸順を評価する。実験のケースとしては以下のケースで実験を行う。今回実験を行うケースは 3 機から 7 機を毎ステップ追加するケース（計 5 ケース）で実験を行う。この時、すべての実験ケースで 10 ステップで実験を行い、試行回数は 10 試行で実験を行う。

3 機での実験結果

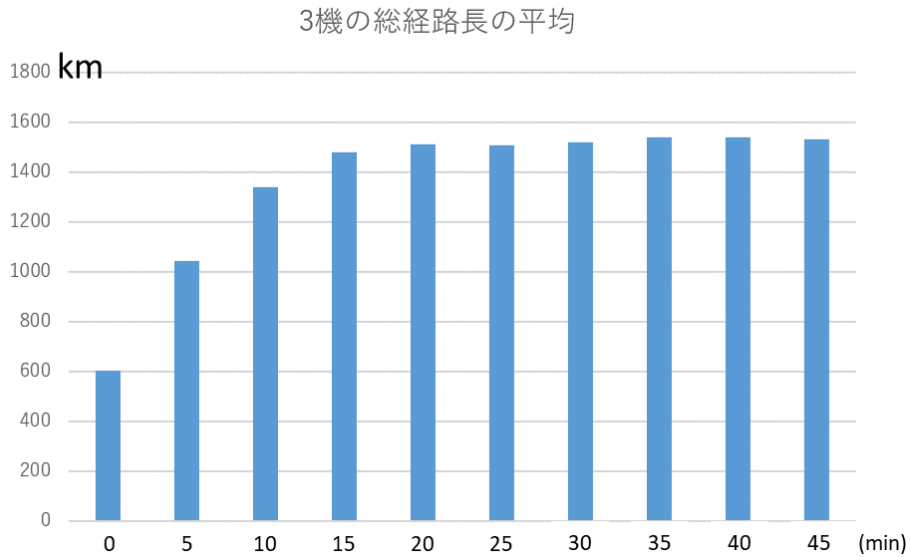


図 5.35 ステップ毎での平均経路長 (3 機)

図 5.35 は 9 試行でのステップ毎の総経路長の平均を表しており、縦軸は総経路長 (km)、横軸はステップを表している。

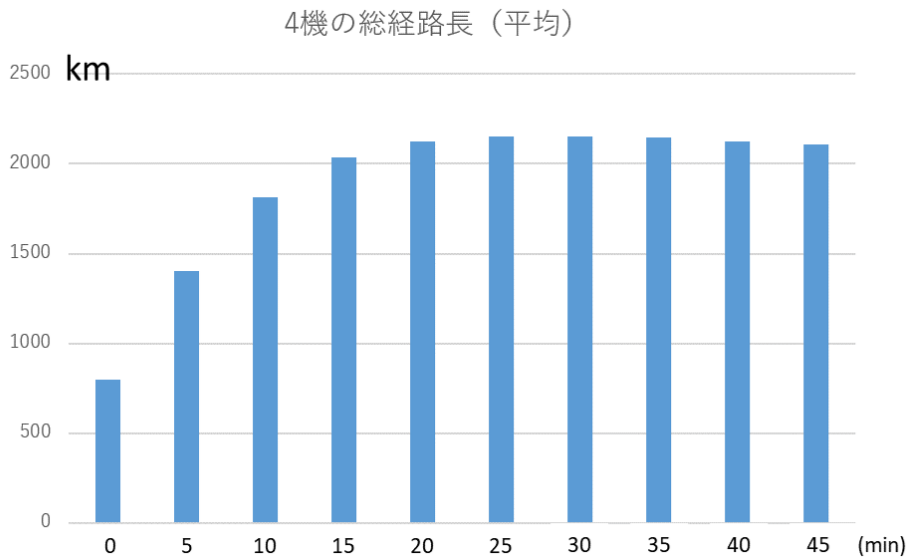


図 5.36 ステップ毎での平均経路長 (4 機)

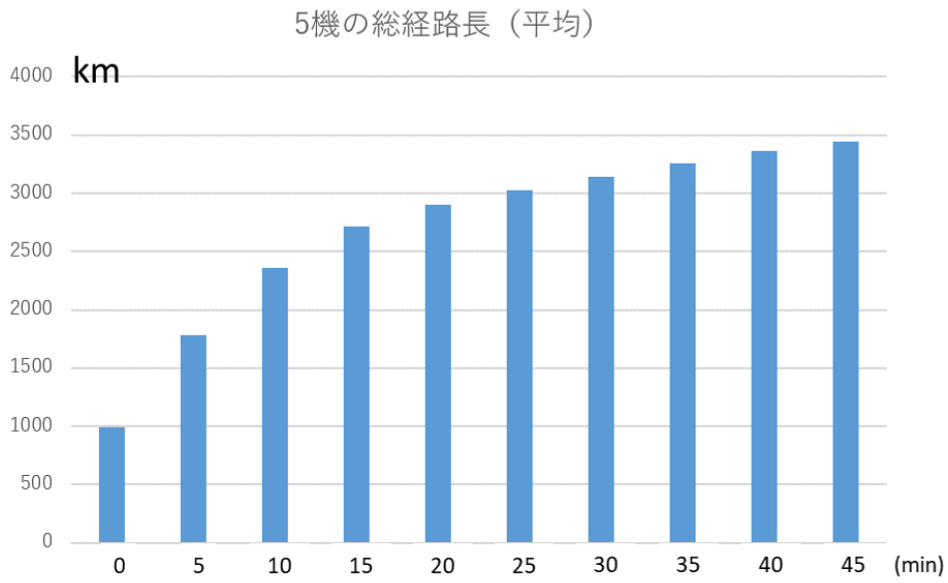


図 5.37 ステップ毎での平均経路長（5 機）

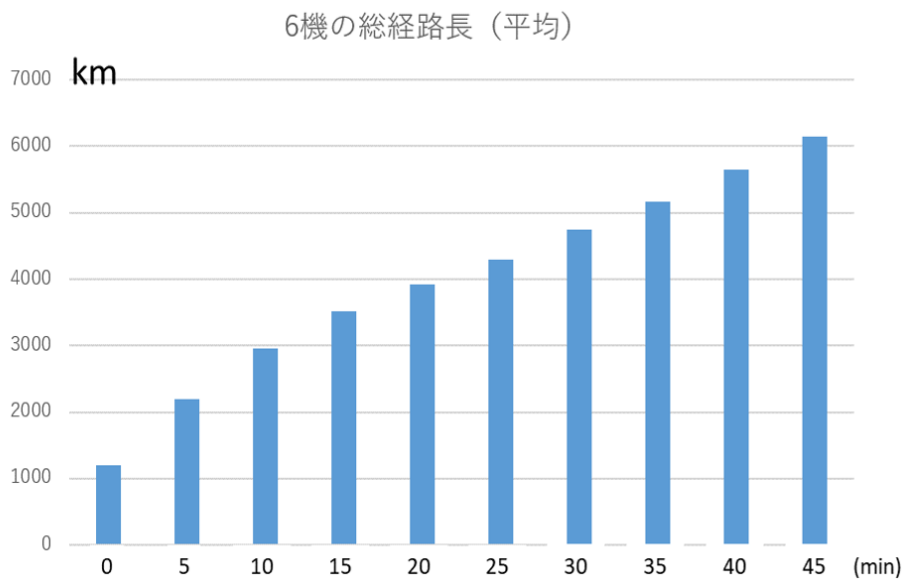


図 5.38 ステップ毎での平均経路長（6 機）

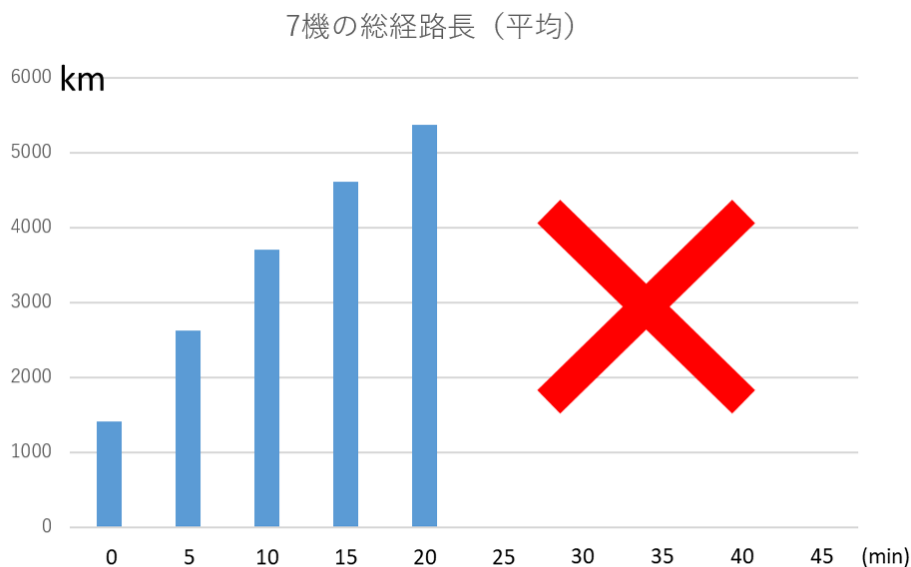


図 5.39 ステップ毎での平均経路長（7機）

今回の実験は3機から6機までのケースはそれぞれ全てのケースで実行可能解を生成することが可能であることが明らかになった。3機追加するケース、及び4機追加するケースは5ステップ目から収束していることが図 5.35, 図 5.36 からわかる。そして、図 5.37, 図 5.38 からは収束せず、10ステップでも総経路長が増加していることがわかる。

次に、図 5.40 は7機のケースに関するステップ毎の実行不可能試行数を表しており、縦軸は実行不可能解の個数であり、横軸は1ステップを表している。5ステップまで9シード全て実行可能であったが、その後、徐々に実行不可能解が増え、最終的に10ステップ目ではすべてのケースにおいて実行不可能解となっていることが図からわかる。このように過密であればあるほど実行不可能解になりやすく、ステップが増えれば増えるほど、航空機数が増えるため、着陸間隔の調整が難しくなることがわかる。

実行不可能の総数

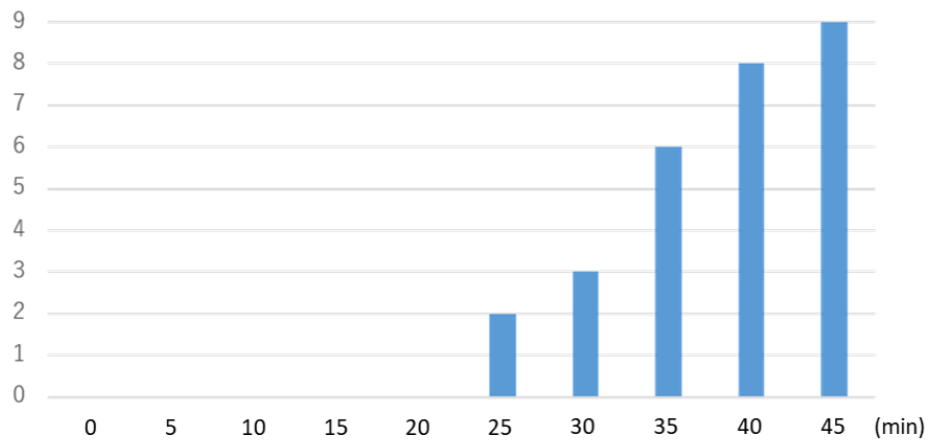


図 5.40 7機での実行不可能解の総数

ステップ毎の経路長の考察

個々では各ステップで生成された着陸順および経路長を考察する。

出現した航空機の経路長の平均 (4機)

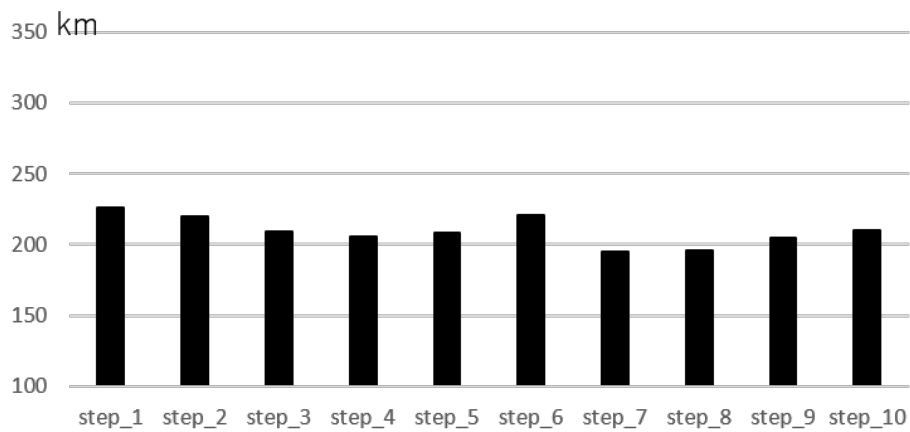


図 5.41 4機の生成された総経路長の平均

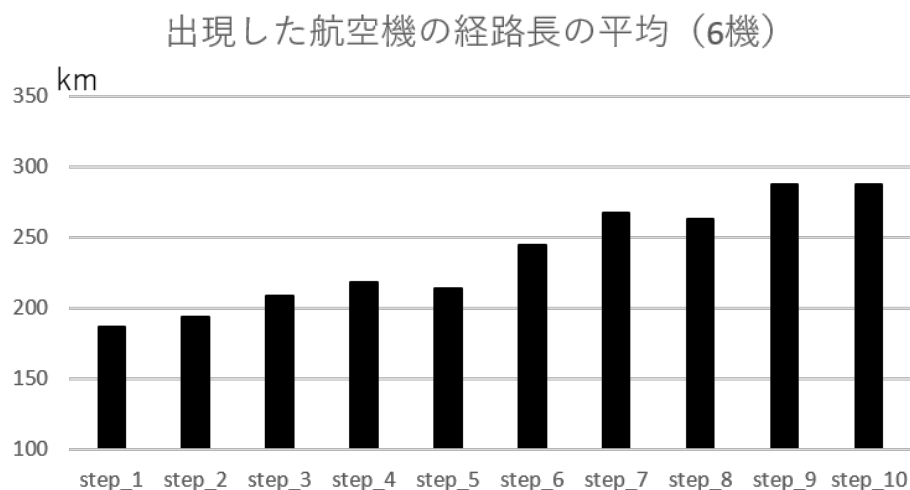


図 5.42 6 機の生成された総経路長の平均

図 5.41, 図 5.42 はステップ毎に追加された経路長の平均を表している. 図 5.41 は 4 機の平均の経路長, 図 5.42 は 6 機の平均の経路長を表している. この二つを比較すると, 4 機の平均経路長はステップが増えても平均の経路に変化はないが, ステップ毎の追加が 6 機の場合にはステップ毎に 6 基の平均経路長が増加していき, 10 ステップ目では平均が 300km 近くまで増加していることがわかる. このことから, 10 ステップ以降も増加する可能性はあるが, 10 ステップ目でも着陸順の生成が可能であったため, 6 機での着陸順生成は可能であると考えられる.

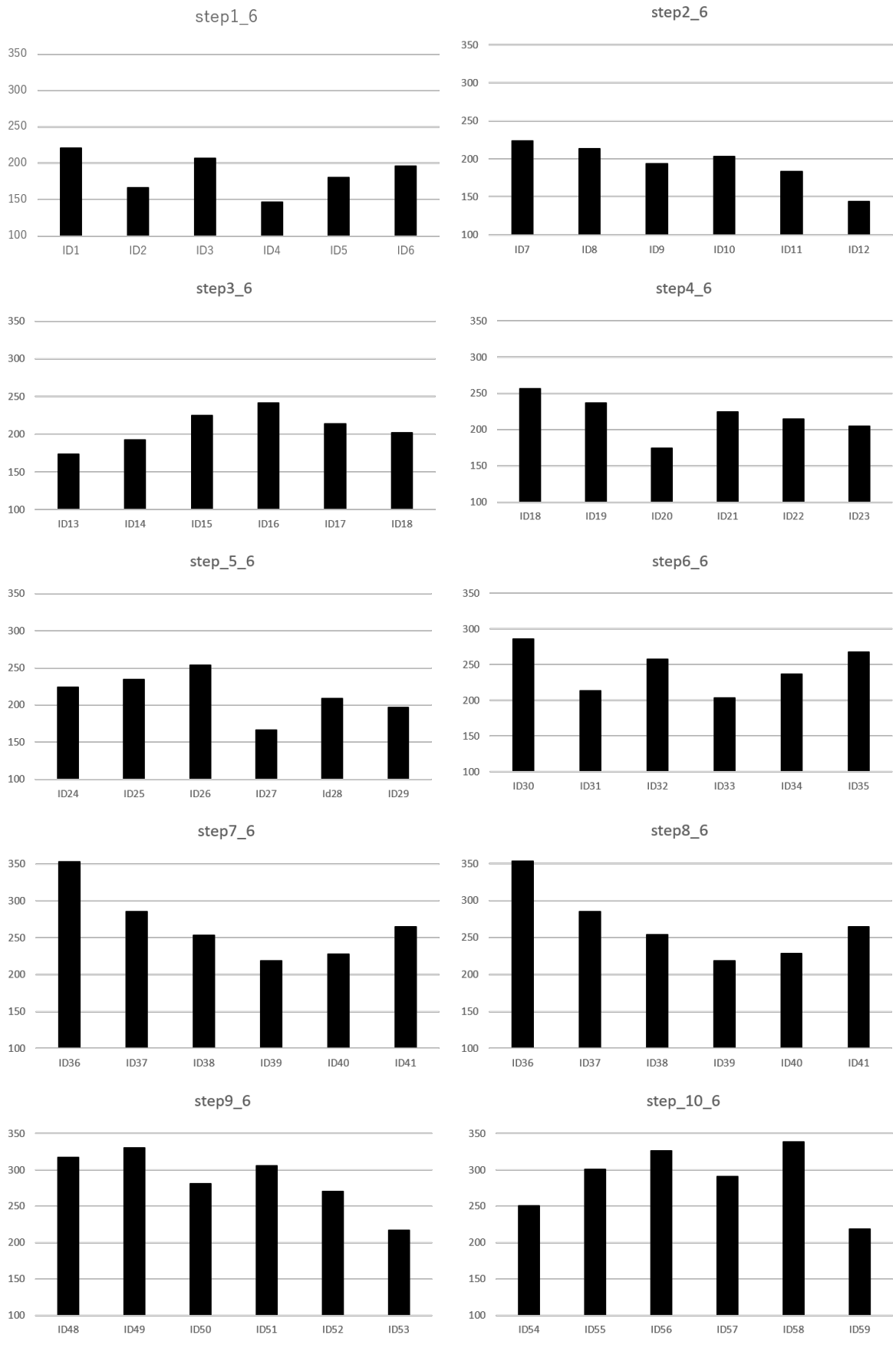


図 5.43 ステップ毎の各航空機の経路長

図 5.43 は各航空機の初期経路長を表しており、縦軸は経路長、横軸は航空機番号 (ID) を示している。初期位置はランダムで配置されるため、各航空機の経路長には差はあるが、図 5.42 から平均はステップ毎に増加していたため、ステップ毎の最小の経路と最大の経路は初期は 150km 以下の経路が生成され、最大でも 250km の経路だが、9 ステップ目、10 ステップ目では最小の経路でも 200km を越え、最大で経路が 300km を超えている。これは、ステップ毎に航空機が追加しているため、先に航行している航空機が滞留をしているため、その後ろを航行するために経路長が増加していることがわかる。

考察 2-3：手法の限界と理論的な航空機数に関して

理想状態における航空機の着陸容量は全航空機が管制官の制約同様に並んだ場合である。自由経路においては最終進入点における航空機間の間隔を 5 NM と設定しており、この間隔を時間に換算し、1 時間の比で表すと着陸容量を概算できる。以下が概算式となる。

$$N = \frac{T}{t} \quad (5.7)$$

T は集計期間、 t は 2 機の航空機間隔を時間間隔に変換したものである。今回、集計期間は 1 時間とし、 t における管制間隔は 5 NM と実験において設定している。速度に関する計算はモデリングの章で説明をしたが、250km 四方の最適化範囲において 5 分に 50km の巡航を仮定しており、1 時間当たり平均速度は 600km/h となる。この速度を海マイルに変換する (1.8562 を除する) と $324Nt$ と算出される。このことから、 t は速度、距離の関係から、管制間隔から速度を除することで求めることが出来る。

これによる計算結果として、64.8 機となる。第 5 章において、継続的な最適化による限界量を求めたところ、5 分毎に 7 機入域する場合において、5 回目の最適化までは実行可能な着陸順を獲得することができたが、それ以降は実行不可能解の試行回数が増えています。5 分毎に 6 機を追加する場合においても、10 回の航空機の追加において、最後の 10 回目の最適化の総経路長は増加しており、これは、後に追加された航空機はより長い経路を選択しているためである。ここから、10 回目以降の最適化を実行した場合に実行不可能な着陸順となる。5 分 6 機が入域する場合において、経路長が安定するので、ここが手法の限界点であると言え、これは単位時間換算で 72 機となることから、予想限界とほぼ一致していることがわかる。

実際に 10 ステップ目以降の 6 機が実行不可能な着陸順となったステップ数を以下の表 5.6 にまとめた。表は 6 機が連続的に入域する環境において最適化した場合の実行不可能解となるステップ数である。5 分に 6 機入域する環境において、適切な間隔を満たさない着陸順をステップ数の列で表される。10 試行全て 10 ステップまで実行可能な着陸順を獲得可能であることを確認したが、これが破綻する状況が 11 ステップ目から生じる。少なくとも 14 ステップ目にあたる 70 分の最適化において 10 試行全てで実行不可能解となることを確認した。このように 6 機連続して入域する環境において実行不可能解とな

ることが実験的にも明らかになった。

表 5.6 6機における実行不可能解となるステップ数

試行番号	ステップ数
1	12
2	11
3	11
4	13
5	14
6	12
7	12
8	13
9	11
10	14

第6章

固定経路による着陸順最適化

6.1 アプローチ

図 6.1 は本研究の位置づけをアプローチを表す。固定経路の位置づけとしては分類基準としては、入域時刻の指定の有無と合流点の個数により分割した。本研究では入域時刻の指定のない複数合流点における着陸順の生成が可能な手法の提案を固定経路における本研究の位置づけとする。本研究では右上に分類された Zuniga らの手法を基本にする。図 6.1 の左下には Zuniga らが適用したスペイングランカナリア空港における3つ固定経路とそれに付加された迂回経路、そして二つの合流点が表されている。このマップにおいて航空機の合流及び着陸順の生成を実現するために航空機に対する迂回量、航空機の速度、入域時刻を探索し、実行可能な着陸順を獲得する。ここで、通常の航空管制官は入域時刻の指定することが出来ないため、運用上においては、入域時刻の指定がないことが前提の最適化が必要となる。しかし、入域時刻の指定が無の場合には実行可能な着陸順の獲得が困難となるため、本研究ではこの探索を最大限活用するため、スケジュールを分割し、探索空間を狭めた上で効率的な探索を行う。

従来はこれに加え、入域時刻の変更が可能であるが、実際に、航空管制官は入域時刻の変更が出来ないため、本章では迂回経路と速度制御のみで交通流過密状態における着陸順最適化を目指す。

固定経路に対する最適化手法として、混合整数計画問題に数理モデル化を行い求解する、決定論的方法を用いたアプローチや、焼きなまし法 (Simulated Annealing:SA) などのメタヒューリスティックスを用いた方法論、この二つをハイブリッドするアプローチなどがある。

管制官		合流地点の数	
		単	複
		着陸順	複数合流点の着陸順
運用不可	入域時刻指定有	[Xue et al., 2016] [Ahmed et al., 2017]	[Zuniga et al., 2011] [Liang et al., 2017] [Hong et al., 2017]
運用可	入域時刻指定無	[Kageyama et al., 2015] [Sama et al., 2013]	本研究②

複数合流点への対応

アプローチ

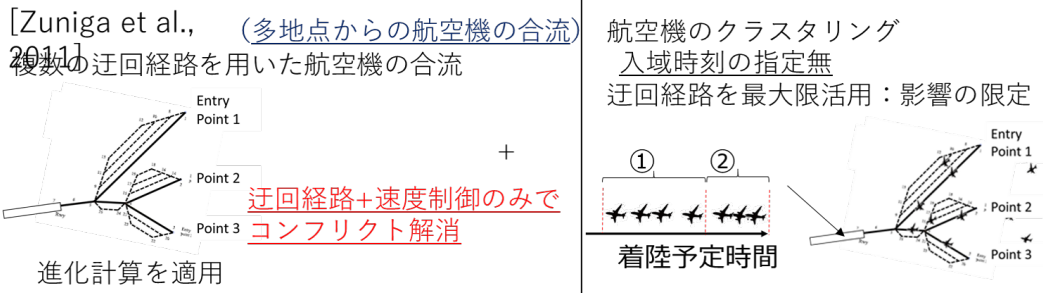


図 6.1 固定経路におけるアプローチ

6.2 進化計算に基づく固定経路による着陸順最適化

本研究では最適化手法として進化計算の一種である GA を採用する。従来手法のメタヒューリスティックアプローチとして SA が用いられているが、この方法論は一つの解の決定変数（本章では、初期速度と迂回量）を変動させ、解の値を変えるか否かを示す遷移を確率的に決定する。一方で GA は解を表す個体が複数存在し、各個体の解として良さを表す評価値に基づいて自分とは異なる他の解の情報を利用し、解が最適解に近づくように決定変数を変化させる。本研究ではシミュレータにより個体の評価を獲得するため、アルゴリズムの処理速度はこのシミュレータによる解の評価値の獲得に依存する。本研究で扱う問題は固定経路による混雑環境下における着陸順の最適化であるため、航空機同士のコンフリクトが非常に多く発生しやすい環境である。そのため、SA のような単点で探索する方法論よりも GA のような探索法適用した場合のほうが実行可能な着陸順を発見しやすく、その分解の進化を促進し、より最適性の高い解（着陸順）を獲得できる可能性が高い。

6.2.1 遺伝子構造と決定変数

図 6.2 は固定経路における GA を適用する場合の遺伝子構造を示している。Individual と呼ばれる個体（1つの解に相当）であり、個体の中には全ての航空機である N 機の決定変数が格納されている。決定変数は迂回量を制御するための α と初期入域速度の変更量を制御する v がそれぞれの航空機毎に格納されているが従来手法においてはこれに加え、入域時間の秒数の計三変数を仮定する。迂回量は実数値であり、0 から 1 までの範囲をとり、入域速度は初期入域速度の増加量の上限と下限の範囲の整数値を取る。遺伝子に値を格納する際に、インデックスの低い遺伝子には入域時刻の低い航空機を格納する。

初期化において生成される遺伝子については、入域速度に関しては乱数により選択された値を持つが、迂回量に関しては初期化においては 0 の値を持つ。この理由として、迂回を利用せず速度変更のみで混雑を解消できるのであれば、固定経路から逸脱がなく、管制官に対しての負荷を抑えることが可能なため、初期の迂回は利用しない上で速度変更のみでコンフリクトのない着陸順の獲得を目指すよう設計した。また、各航空機毎にコンフリクトカウント c を持っており、他の航空機との距離が管制間隔以下であればコンフリクトと判定されこのカウントが 1 付加される。

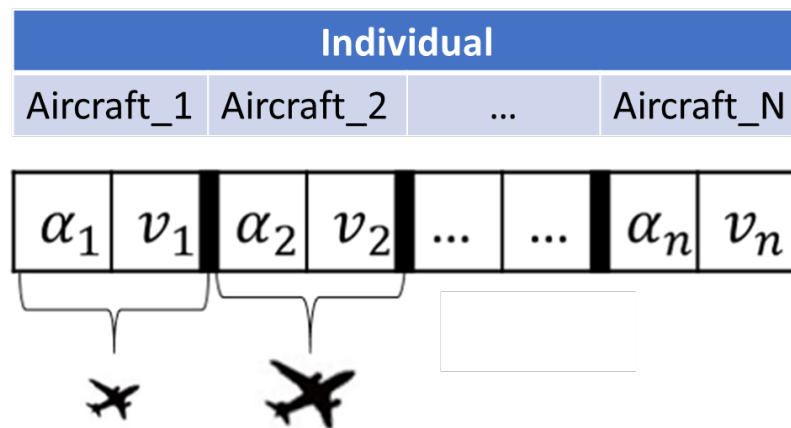


図 6.2 遺伝子構造

6.2.2 オペレーション

GA における選択、交叉、突然変異のオペレーションは従来手法である Zuniga らのオペレーションを用いる [87]。以下にこれらの手法の詳細について説明する。ただし、迂回経路は実数値制御であるため、既存の実数値における交叉、突然変異の方法を参考にした。

交叉 : crossover

はじめに、速度に関する交叉方法に関して説明する。交叉オペレーションは通常の一様交叉ではなく、コンフリクト数に基づく一様交叉を行う。ここで、コンフリクト数の算出に関してはシミュレータを介して算出される。シミュレータは Δt 毎に各航空機の位置を記録する。すべての航空機のシミュレーションが終了すると、記録された時刻における航空機同士の距離関係を算出し、管制間隔以下であれば航空機自身のコンフリクトカウントに1が付加される。つまりこのコンフリクトカウントが大きければ大きいほど他の航空機とコンフリクト状態であることを示し、更に実行不可能な着陸順であることも同時に示している。子個体に対して交叉オペレーションによる値の受け渡しは以下の確率によって行われる。常に近い位置で航空機同士が航行している場合にはコンフリクトカウントが大きくなる。

$$S = C_{p1} + C_{p2} \quad (6.1)$$

$$P_c = 1.0 - \frac{C_{p1}}{S} \quad (6.2)$$

式 (6.1) は交叉対象として選択された親個体である $p1$, $p2$ の全航空機のコンフリクトカウントの総和を示している。 C_{p1}, C_{p2} はそれぞれ、親個体 $p1$, $p2$ における全航空機コンフリクトの合計を表す。この値を利用して、式 (6.2) は子に対するデータの引継ぎ確率を示す。確率であるため、0 から 1 の範囲内の値をとる。 $\frac{C_{p1}}{S}$ は式 (6.1) に占める親個体 $p1$ のコンフリクトの割合を表している。これは確率 P_c で $p1$ の個体の遺伝子座の確率を引き継ぐことを表す。従来手法ではこの選択された遺伝子の値を両方の子個体に引き継ぐが、本研究では、一方の子個体はコンフリクトの少ない遺伝子の値を引き継ぎ、もう一方はコンフリクトの多いほう遺伝子の値を引き継ぐ。このようにすることで、多様性を維持し、局所解に陥ることを防ぐ。コンフリクトがある場合の交叉に関する確率は上記に従い実施するが、 $S = 0$ となる場合には上記の式が成立しないため、親個体 $p1$, $p2$ の値を等確率で選択する。

次に、迂回量の交叉について説明する。迂回量は実数値を扱うため、速度制御と同様の交叉を行った場合、十分な探索が行えない可能性がある。そのため、迂回量の制御は十分な探索を行うため、SBX 交叉法を適用する。SBX 法の式は 3.1.1 小節で説明した。

突然変異 : mutation

速度変更における突然変異も交叉と同様にコンフリクトに基づき突然変異を適用する遺伝子を決定する。以下の式により突然変異確率を定める。

$$P_m(k) = \frac{\sum_{i=1}^{i=k} c_i}{C} \quad (6.3)$$

式 (6.3) は k 番目の遺伝子座に関する突然変異変数を表している。 C は突然変異の対象となる個体の総コンフリクト数、 c_i は遺伝子座の中の i 番目におけるコンフリクト数をそれぞれ表す。式は k 番目までのコンフリクトの累積割合を表している。突然変異が適用されるのは 0 から 1 までの一様分布をもつ乱数 p が $P_m(k)$ 以上の場合に突然変異が適用される。ただし、コンフリクトカウントが 0 である遺伝子座に関しては突然変異は適用しない。この突然変異にはスケジューリングの知識を用いている。コンフリクトを解消するためには到着の早い航空機から決定変数を変更していくことが重要である。そのため、突然変異の確率を入域時間の早い航空機に対して大きくすることで積極的にコンフリクトの解消に作用させる。

次に迂回量の制御変数に関してだが、一般的な突然変異確率である遺伝子長の逆数をある遺伝子座に関する突然変異確率として定義する。突然変異確率は 0 から 1 の均一分布を想定した実数値を乱数により選択する。

6.2.3 遺伝子内の最大制約量と平均制約量を活用した解修復法

実行可能解が見つからない場合、最適化性能の低下を招く恐れがあり、探索の早い段階で実行可能解を獲得する必要がある。適切な決定変数に対して解修復を行うため、本研究では最大制約違反量と平均制約違反量に着目する。本研究では例題である航空機着陸問題に従い、各航空機に対する最小の航空機間の間隔とコンフリクト数をそれぞれ最大制約違反量と平均制約違反量と定義する。従来のアプローチの異なる点は、提案手法が個体の Fitness だけでなく各遺伝子内の評価も考慮することである。このような違いは、提案された方法が実現可能な解をより早期に実行可能な着陸順の探索に貢献する。本研究では、(I) : 最大制約違反に基づく修復、(II) : 平均制約違反に基づく修復、および (III) : (I) 及び (II) を考慮した制約違反のランキングに基づく修復の 3 種類の修復法を提案する。以下にこれらの方法を説明する。

手法 (I) 最大制約違反量に基づく解修復法

手法 (I) では最大制約違反量に基づき遺伝子内の特定の遺伝子座に対して解の修正を施す。航空機の位置と速度を考慮し、コンフリクト状態の航空機間の間隔が維持されるよう、実行不可能解を修正する解修復法を提案する。具体的には、全コンフリクトが記録さ

れたコンフリクトデータの中から、航空機の間隔が最小である航空機 2 機の組み合わせを取り出し、これが修正対象となる。この時、2 つの航空機の 1 つの間隔の組み合わせが選択されているのが、以下の式により速度の変更を行う。

$$\begin{cases} v_{\max} \geq v'_w \geq v_w (w \text{ is leading aircraft}) \\ v_w \geq v'_w \geq v_{\min} (w \text{ is trailing aircraft}) \end{cases} \quad (6.4)$$

v'_w , v_{\max} , v_{\min} 及び v_w はそれぞれ遺伝子内 (全航空機内) の最大違反量 (最小の管制間隔) を保持している遺伝子座における最高及び最低速度と解修正前の速度をそれぞれ示している。ここで、航空機同士の間隔は 2 機によって定められるため、式 (6.4) に示す通り、先行機と後続機が最大違反量を保持しており、この 2 機の解修復を行う。先行機の場合には違反 (コンフリクト) 時点での速度から上げるよう作用し、後続機の場合には速度を下げるよう作用させる。この時の速度値の変化は一様分布でランダムな値を選択する。このように作用することで、最大違反量を起こした航空機同士の間隔を大きくし、制約違反を解消する効果が期待できる。これを図 6.3 において最大制約違反量に基づく解修復法の例を示す。図 6.3 の上部の表は 2 つの航空機間においてコンフリクトが発生している航空機識別番号とその距離をまとめたコンフリクトデータベースを示しており、表の左からデータ番号、コンフリクト状態の先行機の識別番号、後続機の識別番号、2 機の距離間隔を示している。データは番号の低い順から航空機同士の間隔が狭いデータが格納される。この例では ID l_1 と t_1 が間隔が小さいことから、この遺伝子が解修復の対象となり、それぞれの速度を式 (6.4) に従い変更する。図の下部は選択された航空機に関する速度とコンフリクト数を表す。速度 v_1, v_2 は先述した式に沿って速度を変更する。

手法 (II) 平均制約違反量に基づく解修復法

手法 (II) は遺伝子内の平均制約違反量に基づいた修復法であり、違反の最大量ではなく、どのくらい他の航空機に影響を与えているかという平均量的な観点を考慮する。本手法では各航空機のコンフリクト数を平均制約違反量として定義し、遺伝子内の平均制約違反量が最大の遺伝子を解の修正の対象とする。修正はコンフリクト状態である航空機同士の中で一番短い間隔を保持している部分に着目する。この間隔において、対象の航空機が先行機である場合には速度を早めるよう作用させ、後続機である場合には速度を遅くするよう作用させる。以下に各航空機の合計のコンフリクト数の算出法を示す。

$$C_i = \sum_{t=1}^T \sum_{j=1, j \neq i}^n c_{i,j}^t \quad (6.5)$$

$$c_{i,j}^t = \begin{cases} 1 & (i, j \text{ are in conflict}) \\ 0 & (\text{otherwise}) \end{cases} \quad (6.6)$$

$$C_{gene} = \text{Max} \{C_1, \dots, C_n\} \quad (6.7)$$

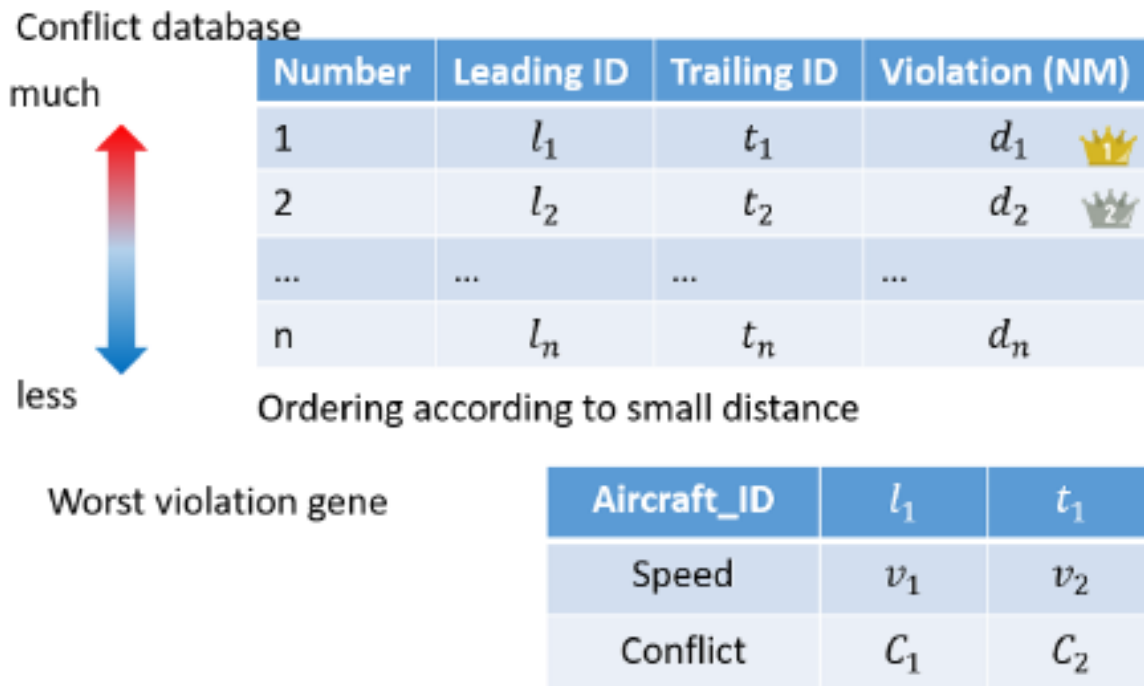





図 6.3 最大違反量

式 (6.5) はコンフリクトの算出法を表し、すべての期間におけるコンフリクトの総和となっている。式 (6.5) 中の C_i , C_{gene} は i 番目の遺伝子座における航空機における合計のコンフリクトと遺伝子全体の中で合計コンフリクトが一番大きい航空機におけるコンフリクト数を表している。 t はコンフリクトを算出するための単位時間であり Δt に該当する。これを T は全単位時間数であり、例えば、10 秒毎にコンフリクトを記録しているのであれば、全スケジュール期間が $10T$ 秒となる。先述の通り、 C_{gene} の値を持つ航空機 (遺伝子座) が解修正の対象となり、解修正が行われた後にシミュレーション部により評価値を得る。図 6.4 は平均制約違反量に基づく解修復法の例を示しており、上部の表は個体の遺伝子情報を表しており、速度とコンフリクト数が格納されている。この中で、ID 1 が一番コンフリクトがあるため、解の修正の対象となる。対象となった航空機 (遺伝子) の解の変更の仕方は、修復対象である航空機 (遺伝子) における、航空機の中で一番航空機同士の間隔の狭いコンフリクト状態を解消する。図の例では先行機が ID 1、後続機は t_1 である。後続機の選定基準は、全航空機の中で最大のコンフリクトを持つ航空機が修正の対象となり、対象の航空機のコンフリクト関係にある航空機の中で一番間隔の短い航空機を解修復の対象として選択する。このほかに後続機が他の航空機が対象となるが、この 2 機が速度の変更の対象となる。図においては、最小間隔 d_1 の関係を持つ航空機が選択された。

Most conflicts in gene is modified

Gene	Aircraft	ID1	ID2	ID3	...	ID N
Speed		v_1	v_2	v_3	...	v_N
Conflict		 C_1	 C_2	C_3	...	 C_N

↓

Number	Leading ID	Trailing ID	Violation (km)
1	ID1	t_1	d_1

図 6.4 平均違反量

手法 (III) 最大制約違反量及び平均制約違反量を活用した解修復法

先述した二つの解修復法はそれぞれ最大制約違反量あるいは平均制約違反量の一方を考慮した手法であるが、本手法はどちらも考慮した手法である。これを実現するために、ランキングに基づいた制約違反量を新たに適用する。このランキングは遺伝子内の序列を決定するものであり、ランキングの最低順位は着陸順を行う航空機数であり、遺伝子長に対応する。一つ目のランキングは最大制約違反量に基づいたランキングであり、違反量が多ければ大きいほど遺伝子内の順位が高くなる。この時の違反量は航空機の間隔の大きさを表しており、間隔が小さいほど違反量が大きいことを意味する。以下の式 (6.8) に従って遺伝子内の相対価値を決定する。

$$\text{value}_{\text{hybrid}} = \text{rank}_{\text{violation}} + \text{rank}_{\text{total}} \quad (6.8)$$

式 (6.8) の $\text{rank}_{\text{hybrid}}$, $\text{rank}_{\text{violation}}$ と $\text{rank}_{\text{total}}$ はそれぞれ平均と最大制約違反量を相対的に評価した値、最大制約違反量における遺伝子内順位、平均制約違反量における遺伝子内の順位をそれぞれ示している。手法 (III) の順位は $\text{rank}_{\text{hybrid}}$ に従い、この値の小さい遺伝子が解修復の対象となる。

6.2.4 実験

実験内容

提案手法の有効性を検証するため、フランスのシャルル・ド・ゴール空港をシミュレーション実験の対象空港として選定した。シャルル・ド・ゴール空港は 8 つのエントリーポ

表 6.1 GA におけるパラメータ

GA	Value
population size	100
generation	250
Simulation mutation rate	0.8
crossover rate	1

表 6.2 固定経路におけるシミュレーションのパラメータ

Simulation	value
aircraft	29
period	3600 (sec)
δt	10 (sec)

イント有した環境である。8つのエントリーポイントは経路は赤線で表す標準到着経路と呼ばれる着陸飛行場の滑走路へと導き、各空港ごとに設定されているものである。航空管制官はこの経路に基づき航空機の制御を行う。標準到着経路にはサブルートと呼ばれる迂回経路本研究の実験では空港における到着機の混雑時において本研究の実験では空港における到着機の混雑時において提案手法の有効性を検証する。具体的な実験設定は以下の表に記載する。表 6.1 は GA における Population サイズ，世代数，突然変異率，交叉率とスケジューリングを行う航空機の機数，スケジュール期間，シミュレーションにおける位置情報を記録する周期を表す。本研究の航空機は実際のシャルル・ド・ゴール空港におけるレーダデータを使用しており，エントリーポイントにおける初期進入速度，エントリーポイントにおける進入時間はレーダデータに基づいた値を使用する。比較対象として，提案手法で述べた，(I) 最大制約違反量を考慮した解修正法，(II) 平均制約違反量を考慮した解修正法，(III) 最大・平均制約違反量の両方を考慮した解修正法そして解修正法を入れない従来手法 4 つをそれぞれ比較した。評価基準として，コンフリクト数と Fitness 及び，実行可能解を獲得する世代数とし，各 31 試行を行った。

結果

図 6.5 は各世代ごとの Fitness の推移を表しており，縦軸は Fitness，横軸は世代数をそれぞれ表している。対応する結果は conventional は解修復法を用いない従来手法, Hybrid は，提案手法 (III) である，最大・平均違反量を考慮した手法を指し，Total conflicts は手法 (II) 平均制約違反量を考慮した解修復法，最後に Violation は手法 (I) 最大制約違反量を用いた解修復法を表している。図 6.6 は世代毎の最良解のコンフリクト数を表してお

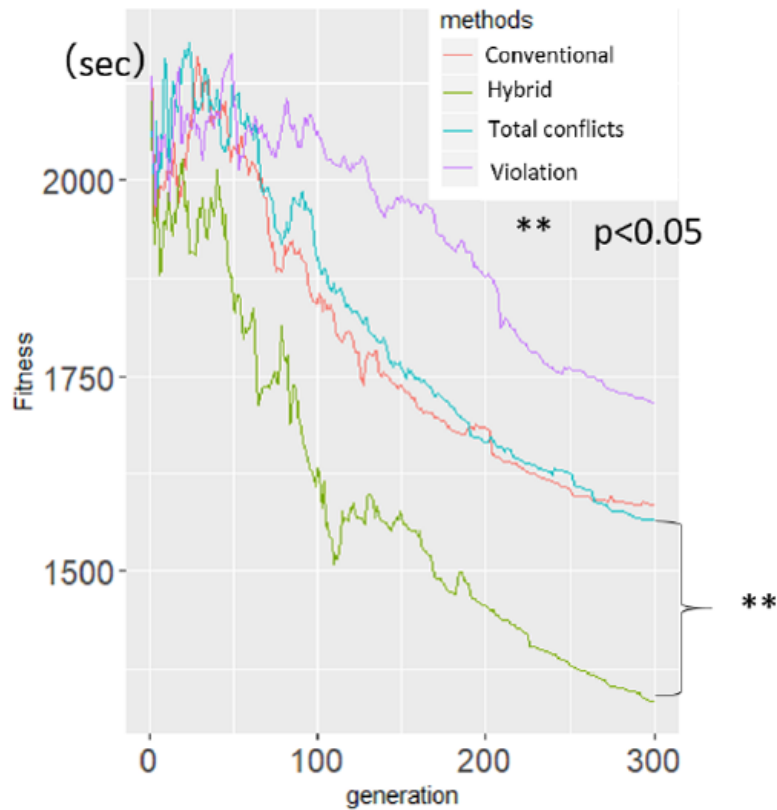


図 6.5 解修復法における Fitness の推移

り，縦軸は個体の総コンフリクト数，横軸は世代数をそれぞれ表している．Fitness での結果は明らかに (III) の最大・平均を考慮した手法が一番良好な結果を示している．最大制約違反量と平均制約違反量それぞれ一方での修正は限定的なものであり，最適性の向上が見られなかった．特に (I) 手法の最大制約違反量のみの評価では Fitness(総遅延時間)の観点では従来手法に比べて，悪化していることがわかる．次に図 6.6 の制約条件である総コンフリクトの世代ごとの変位を見ると，従来手法以外の 3 つの手法ではコンフリクト状態を 300 世代までにほとんどの試行数で解消することが可能だが，最大制約違反量のみを考慮した手法の場合には 1 試行で実行可能解を獲得できない場合があった．平均制約違反量を考慮することで総コンフリクトを解消に貢献することがわかるが，(II) 手法である平均制約違反量のみを考慮した場合の手法は Fitness の観点からみると最適性の向上には寄与していないことがわかる．

6.2.5 考察

図 6.7 は一番初めに実行可能解を獲得した時の世代の全試行における箱ひげ図を表す．左から，従来手法，提案手法 (III),(II),(I) をそれぞれ表す．これより，(II) と (III) 手

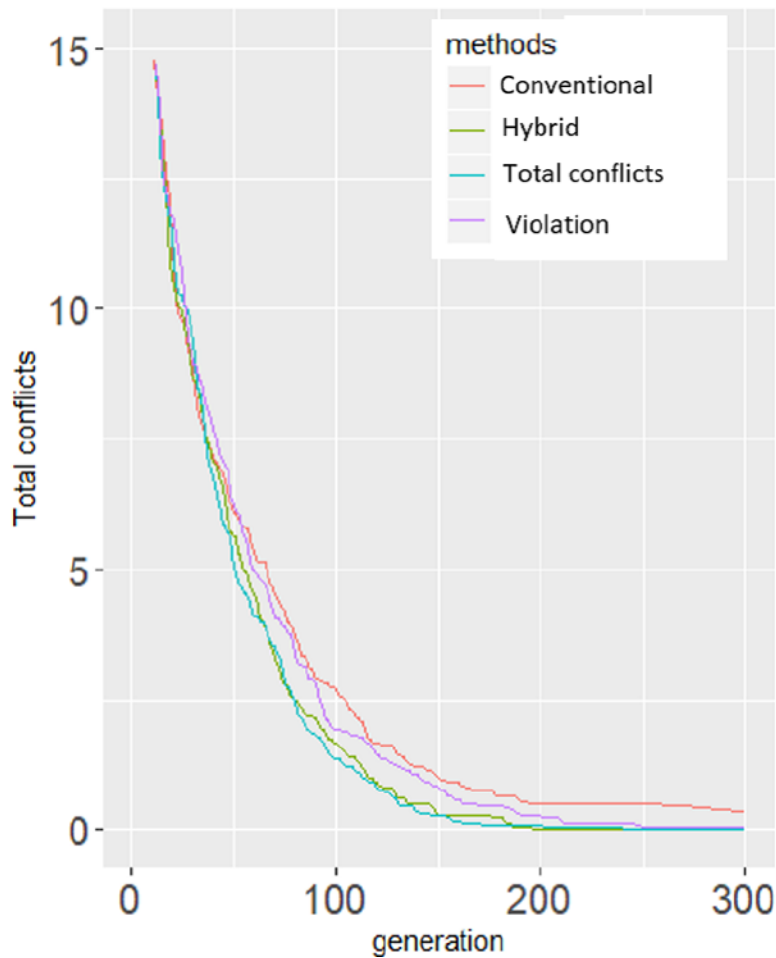


図 6.6 コンフリクトの推移

法が他の2手法に比べ早くに実行可能解を獲得していることがわかる。ただし、(II)と(III)の間には有意差が見られなかった。これにより、早さの観点では平均制約違反量を導入することで従来手法より早く実行可能解を獲得できることが明らかになった。次に、図 6.8 は交叉において、解修復の対象である劣個体において、遺伝子内のどの遺伝子座が修正の対象となったのかを数え上げ、ヒストグラムとして表したものである。左から、手法 (I)、(II) そして (III) のヒストグラムを表しており、縦軸が解修復として選択された回数、横軸は遺伝子座を表している。この図において、手法 (I)、(II) には選択回数に偏りがあることがわかる。しかし、手法 (I) に比べ、手法 (II) は他を選択する頻度も大きくなっているために、実行可能解を獲得が従来手法及び最大制約違反量の手法に比べ早くなったと考えられる。また手法 (III) のヒストグラムに着目すると、選択される解が、他の2手法に比べ、全体が偏りなく選択される可能性が高くなっていることがわかる。これは Population 内の個体の遺伝子が類似することを防ぐ効果があると考えられる。このように、平均制約違反量および最大制約違反量を考慮した評価をすることは解の多様性を導

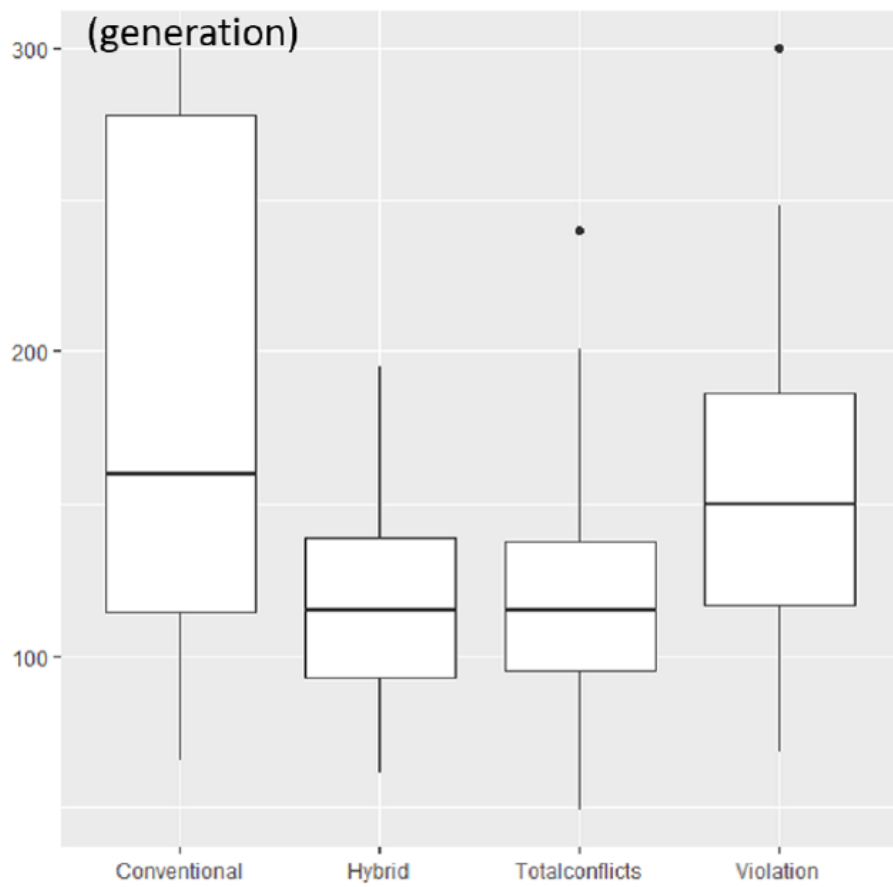


図 6.7 実行可能解到達世代の箱ひげ図

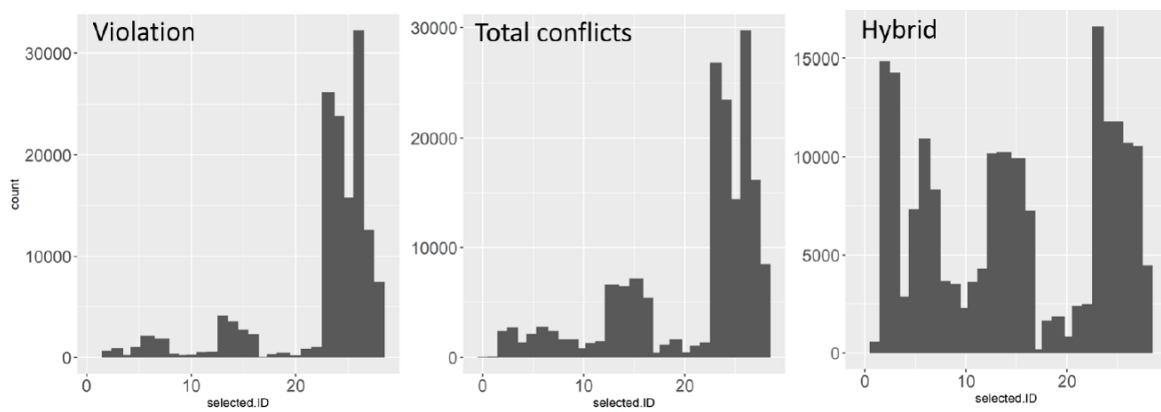


図 6.8 selected gene of infeasible individual

くことで局所解を防ぎ，評価（Fitness）も向上することがわかった．これは，一つの指標による評価ではなく，複数の視点から評価を行うことの重要性が示唆される．

6.3 クラスタリングを用いた分割反復最適化手法

固定経路を用いた場合には、自由経路のように経路の伸縮性に制限があるため、スケジュールの調整力に関しては自由経路に比べて調整能力が低い。そのため、従来手法では、通常の標準到着経路（STAR）に迂回経路を付加することに加え、航空機毎に入域時刻と速度制御の調整が可能であり、これを利用し、混雑時に実行可能な着陸順の獲得及び航空機の合流が可能である。しかし、実際の運用では航空機の種類、方位の制御は行われない。入域時刻の調整は迂回経路や速度調整に比べ、着陸時刻の調整量が大きいことが特徴である。特に入域時刻の遅延に対しては5分以上の遅延を認めている。この調整がなくなることは航空機同士の間隔調整が出来ず、実行可能な着陸順を獲得することが難しくなる。

6.3.1 航空機全体の最適性の向上のための分割反復法

図 6.9 は本研究のアプローチを示しており、本研究では第 2 章の研究の位置づけで述べた固定経路の分類において説明した Zuniga ら [87] の手法を基礎とする。この手法は図 6.9 はクラスタリング部にスケジュール全体の制御を担い、スケジューリング対象の航空機の分割を決定する（群の生成）。着陸予定時刻最適化部においては個体の制御を担い、GA を用いて分割された航空機群毎に最適化を行う。この時、スケジュールの前後関係に矛盾が生じないように、最適化対象の航空機群の前後の群の航空機は固定しておく。この群の最適化が終わった後、残りの航空機群の最適化を行う。全ての群の最適化の終了後、全体のスケジュールを統合する。

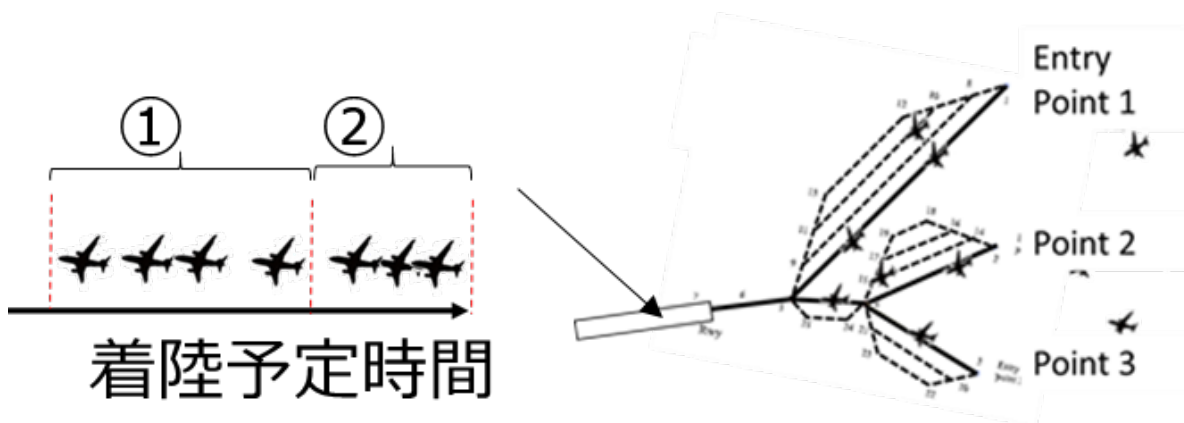


図 6.9 航空機全体の探索効率化の着眼点

図 6.10 は提案手法における全体の構成を示しており、クラスタリング部、航空機群最

適化部に分かれる。固定経路においても全体と個体の考え方として、全体がクラスタリング部、個体が航空機群最適化部となる。クラスタリング部は航空機全体として、クラスタリングを行い、全航空機を複数の群に分割し、スケジュールの探索範囲を削減する。個体における役割としては、分割された航空機群に対して、迂回量と速度の調整を行う。

全体を複数の個体群に分割+個体群ごとに最適化

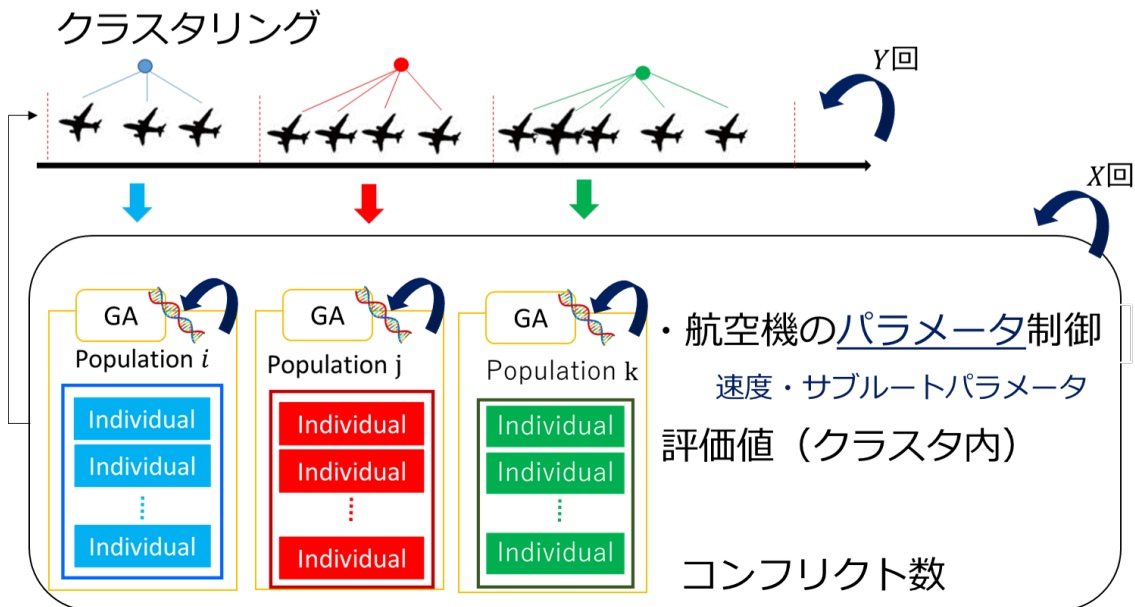


図 6.10 固定経路における全体の構成

6.3.2 クラスタリング部 (全体レベル)

従来手法ではスケジューリング期間が長くなる場合や混雑時など航空機数が多くなる場合、探索範囲が大きくなるため、最適化性能が低下する。この問題に対応するために提案手法ではスケジュール期間を一定に区切らず、複数の航空機毎に群を作り、この群毎に並列的に最適化提案する。この際、重要になるのがどのように群を作るかである。本手法では機械学習の手法の一つであるクラスタリング手法に着目し、この中でも k-means 法を利用する [9]。k-means 法は、はじめに分割数を決め、各個体 (各航空機に相当) が属する群はセントロイド (重心) (各航空機の重心位置に相当) によって決定する。セントロイドはランダムに初期化し、属する個体によって重心が移動する。重心が移動した後、再度個体が属する群を決定し、群での移動がなくなるまで繰り返す。

図 6.11 はクラスタリング部詳細の挙動を表している。上部は初期のクラスタリング、下部は 2 回目以降のクラスタリングについての詳細を表す。航空機のクラスタリングの基準として使用する変数は滑走路に着陸する時間を基準とする。最適化前の段階において

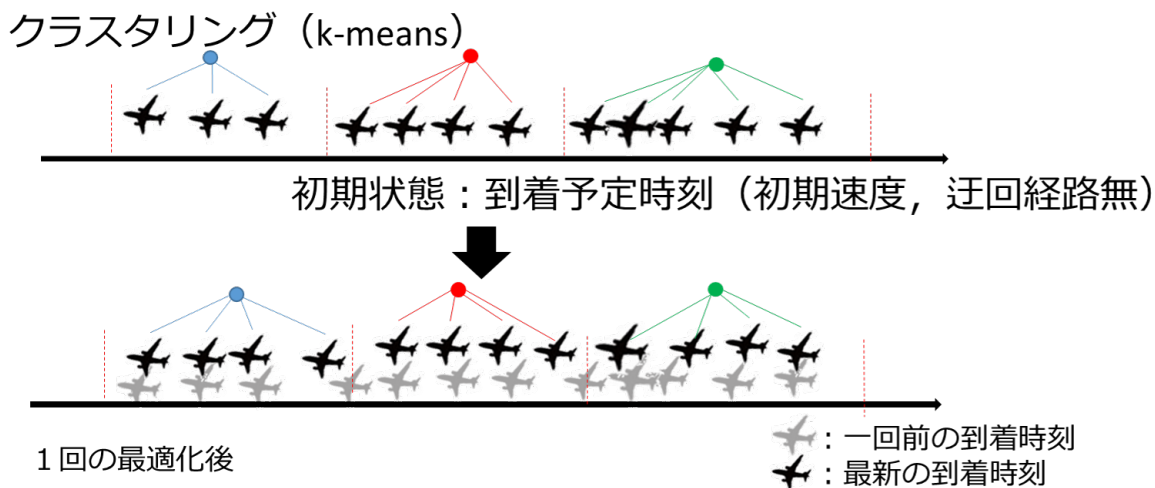


図 6.11 クラスタリングとスケジュールの関係

は迂回量や、速度が未決定であるが、航空機の入域速度は既知であるため、この入域速度を基準として、迂回経路を利用しない場合における着陸時間をクラスタリングの基準とする。

基準航空機の決定

スケジュールリング対象の全航空機を最適化する場合には全ての航空機をシミュレータで航行し、一定時間に航空機の位置を記録し、コンフリクトを算出するが、航空機を複数に分割した最適化するには、この算出に問題が生じる。全航空機を2分割した最適化を例とすると、他方の航空機群よりも早く着陸する群をA群、もう一方をB群とする。もし、A群とB群を同時に最適化を想定すると、A群とB群共に最適化する際にシミュレーションを通した評価が必要になる。この時、A群、B群が共に独立的にシミュレータを用いた場合には、B群における一番早く着陸する航空機がA群における一番遅く着陸する航空機の位置を考慮していないため、コンフリクトが発生する恐れがある。そのため、特に群の先頭と後方に位置する航空機に関しては群の前後の航空機の位置情報が必要になる。つまり、A群を最適化している際にはB群の航空機の情報も固定する必要がある。

そこで提案手法では、群の前後の航空機の情報を用いるため、偶数群と奇数群を逐次的に最適化を行うアプローチをとる。具体的には、はじめに、入域時間の一番早いのは奇数群の一番初めの航空機であるため、奇数群から最適化を行う。この時、偶数群の航空機は最適化は行わず、制御変数である、迂回量と速度は最適化が行われた時の値を用いる。なお、一番初めのは最適化を行っていないため、初期進入速度を与え、迂回量は0とする。偶数群における航空機はこのような制御変数を用いて、奇数群の最適化を行う。GAによ

る機数群の最適化の終了条件を満たした次は偶数群の航空機の最適化を行う。偶数群における航空機の最適化の際には、偶数群の航空機は最適化を行った制御変数を用いてシミュレーションを行う。従来手法では GA によって全航空機の最適なスケジュールを探索するが、提案手法では GA 適用前に、クラスタリングにより航空機を群に分け、これを GA によって最適なスケジュールを探索する。この時、クラスタリングによって獲得した群が適切ではない可能性があるため、GA による世代数を何度か区切り、区切った地点で再度クラスタリングを行う。このような新たな群を生成することで、適切ではない組み合わせの群での探索よりも良いスケジュールを見つける可能性が高まる。

6.3.3 航空機スケジューリング最適化部（個体レベル）

航空機スケジューリング最適化部の構成

図 6.12 はクラスタにより分割された群を最適化する場合の GA の分割を表す。図は、分割数が二つの場合を例とした GA と群の構成を合わせている。N は航空機の総数を示している。群はそれぞれ、Population i, j を持ち、 i には航空機のインデックスの 1 番目から R 番目、 j は R+1 番目から N 番目までの航空機の決定変数が格納されている。この例の分割では R 番目を境に二つに分割されていることがわかる。このケースでは、奇数群には Population i 、偶数群には Population j が所属することになる。

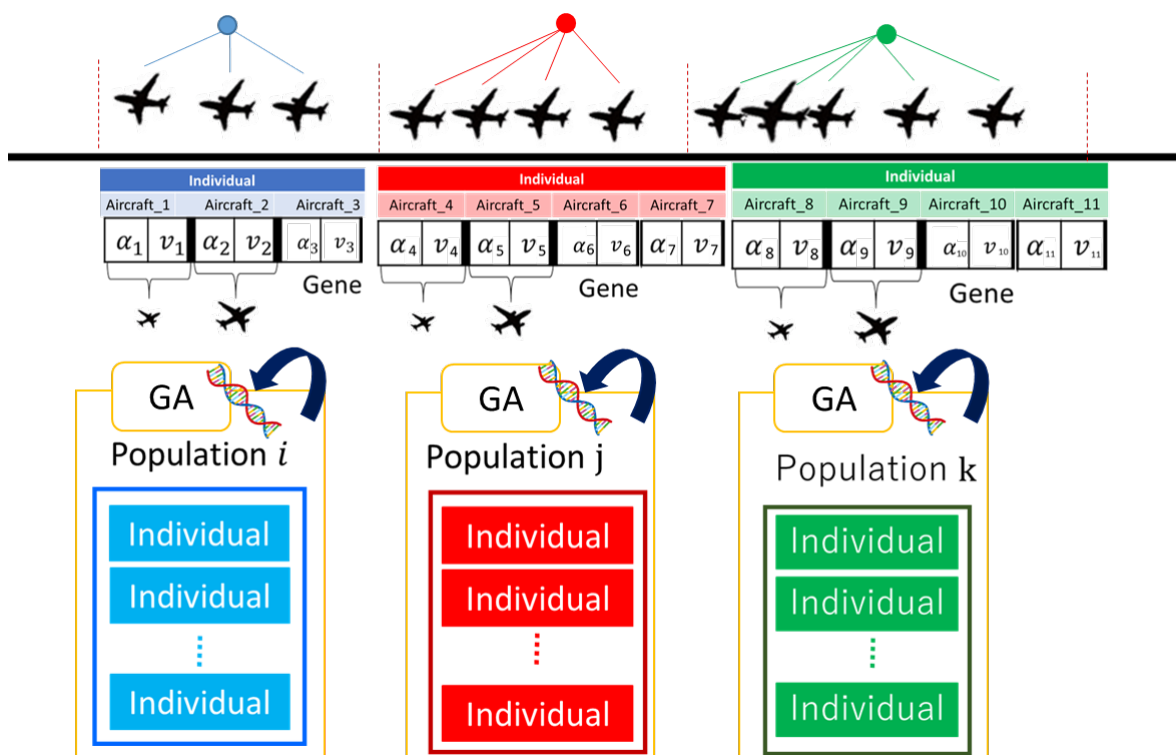


図 6.12 最適化部における GA の構成

6.3.4 航空機スケジューリング最適化部（全体 + 個体レベル）

分割反復最適化における GA の初期解

群生成 (クラスタリング) と最適化は複数回繰り返すが、GA の初期解も同回数必要となる。1 回目の最適化においては、迂回量は 0 とし、速度は乱数により選択された値を利用する。2 回目以降の GA の初期解も一回目と同様の初期化を行うが、解集団の内、一番初めの解を前回最適化を行ったスケジュールの決定変数 (迂回量, 進入速度) を引き継ぐ。最良解の引継ぎは解の改悪を防ぎ、他の解に対して、評価の良いスケジュールの情報を受け継ぐ可能性がある。

分割にかかる最適化の順番

分割反復最適化においては全航空機の群を複数に分割し、奇数群と偶数群に分け、それぞれの群で同時に最適化を実行する。

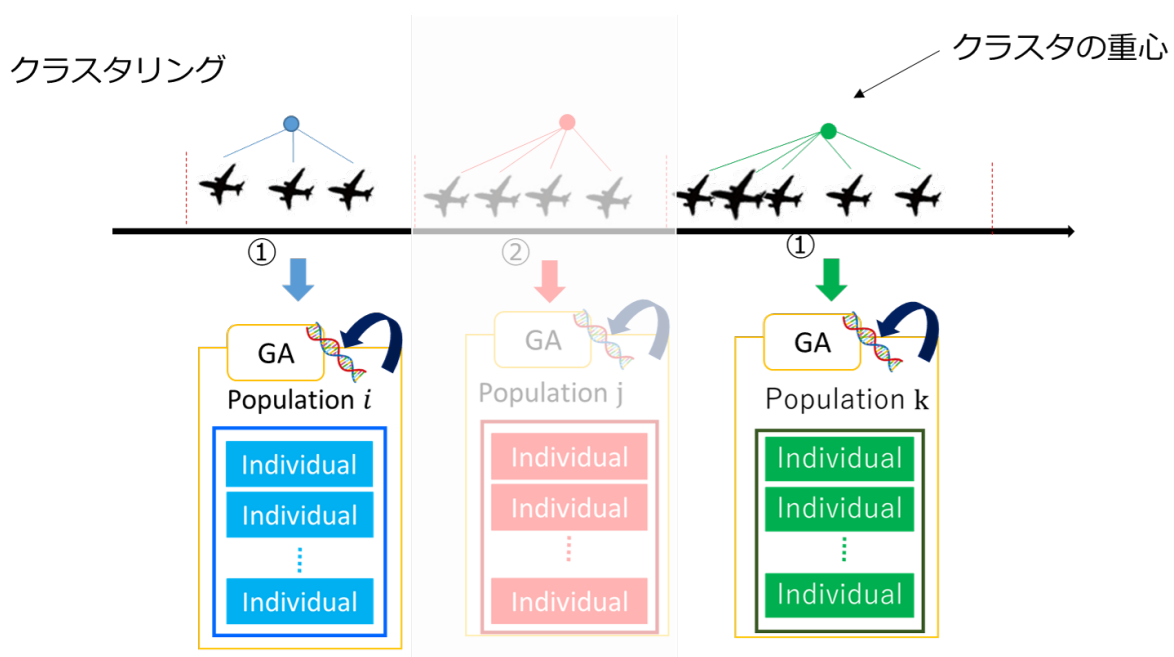


図 6.13 分割した際の最適化の順番

図 6.13 は分割最適化の例を表し、水色と緑色の群が奇数群、赤が偶数群をそれぞれ表す。それぞれの群には少なくとも一つ以上の航空機が所属することになる。分割された場合に、図中の例でははじめに水色と緑の航空機群がそれぞれ独立に最適化が行われる。赤の航空機は待機状態であり、最適化は行われませんが、青と緑の航空機群に対して航空機の決定変数の情報である、迂回量と速度情報を与える。この時、赤の航空機は最適化前の段階であるため、変化することはない。仮に偶数群、奇数群すべてが同時に最適化された

場合には分割の基準点前後の異なる群に所属する航空機がコンフリクトとなる可能性がある。

6.3.5 実験

実験内容

提案手法の有効性を検証するため、フランスのシャルル・ド・ゴール空港をシミュレーション実験の対象空港として選定した。図 4.10 はシャルル・ド・ゴール空港周辺の経路を表しており、8つのエントリーポイント有した環境である。8つのエントリーポイントは経路は赤線で表す標準到着経路と呼ばれる着陸飛行場の滑走路へと導き、各空港ごとに設定されているものである。航空管制官はこの経路に基づき航空機の制御を行う。標準到着経路にはサブルートと呼ばれる迂回経路を5つ設定するが、北西からのエントリーポイントには2つ、そのほかの方角からのエントリーポイントはそれぞれ一つずつサブルートを設定した。

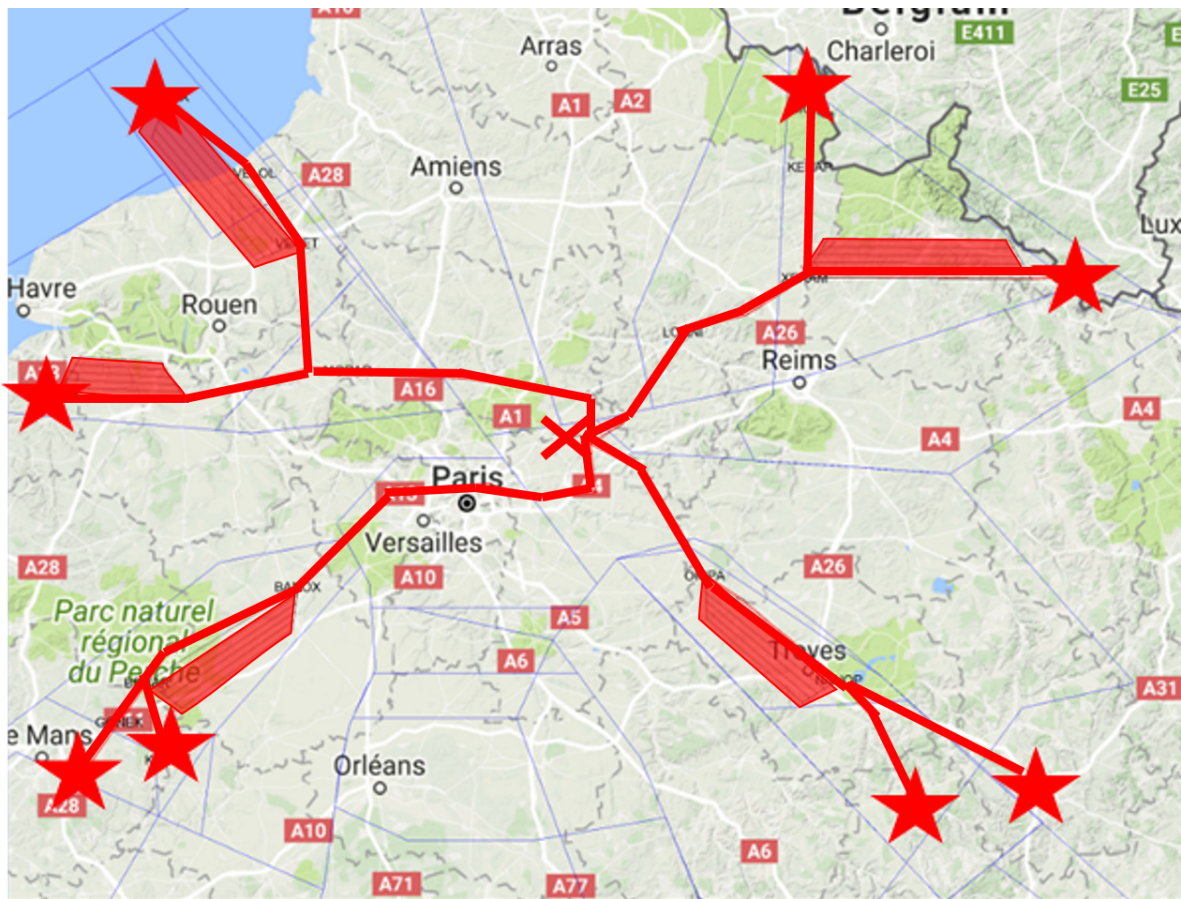


図 6.14 実験環境 (再掲)

表 6.3 分割反復最適化におけるパラメータ

Parameter of GA	value	Simulation	value
population size	100	aircraft	15
generation	250	period	1810 (sec)
mutation rate	0.8	Δt	10 (sec)
crossover rate	1		

実験設定

本研究の実験では空港における到着機の混雑時において提案手法の有効性を検証する。具体的な実験設定は以下の表に記載する。

表 6.3 は GA における Population サイズ，世代数，突然変異率，交叉率とスケジューリングを行う航空機の機数，スケジュール期間，シミュレーションにおける位置情報を記録する周期を表す。このパラメータにおいて，実験を 2 ケース実施した。

- ケース 1 は，分割数による最適化性能に影響があるか検証することを目的とし，分割回数は 1 回，群の分割数をそれぞれ，3,4,5 回として実験を行った。実験の試行回数は 10 回とした。
- ケース 2 は，分割を繰り返すことによる最適性向上のための効果を検証を目的とし，群の分割回数を 1, 3, 5 回とし，群の分割数は 5 回と固定し，試行回数をケース 1 と同様に 10 回として実験を行った。

GA による総評価回数は同じとして，性能評価を行った。例えば，分割回数を 2 回としたとき，毎回の世代数は 125 回となり，分割回数を 5 回にした場合には 50 回を毎回の世代数とした。比較手法とは，スケジュールを一定区間で分割したスライディングウィンドウとし，区切る区間は 500 秒とした。本研究の航空機は実際のシャルル・ド・ゴール空港におけるレーダデータを使用しており，エントリーポイントにおける初期進入速度，エントリーポイントにおける進入時間はレーダデータに基づいた値を使用する。本研究で用いる評価基準は各航空機の予定到着時刻と実際の到着時刻の差分の総和（全航空機分）と全航空機のコンフリクト数とした。スケジュール対象の航空機は以下の図 6.15 に記載した。

結果

ケース 1：分割回数を固定した場合

分割数を変えた場合の結果を以下に示す。

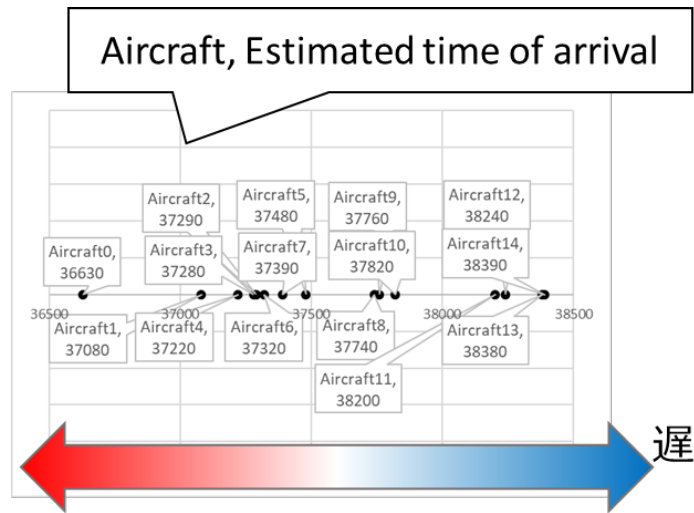


図 6.15 最適化対象の航空機

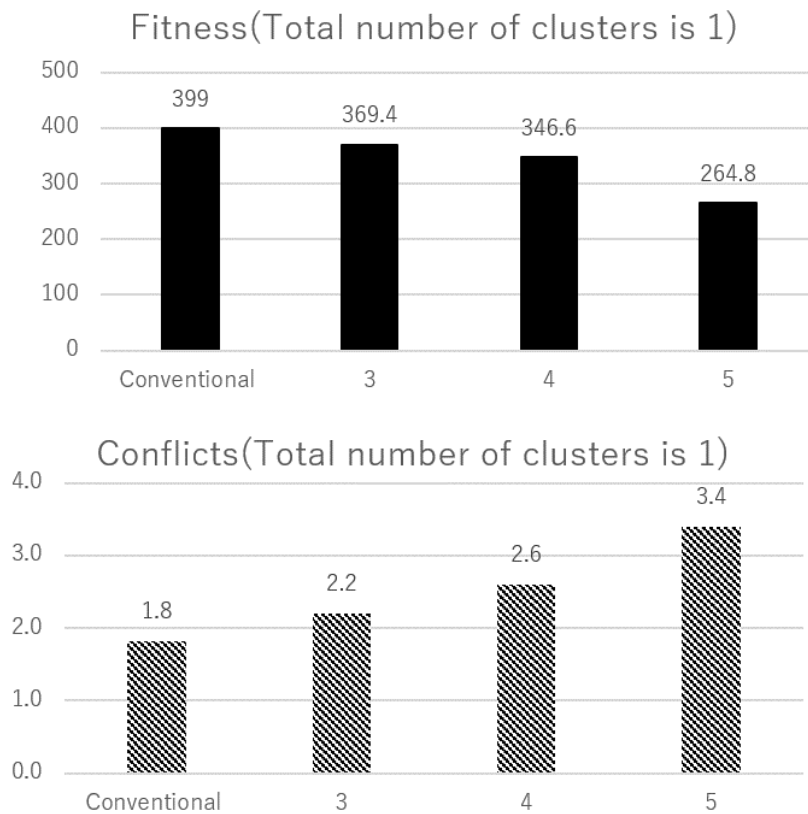


図 6.16 反復回数を 1 回にした場合の Fitness とコンフリクト数の結果

図 6.16 の上部は従来手法と分割する群を 3, 4, 5 分割とした時の Fitness の値であり下部はコンフリクト数である。Fitness は先述した通り、全体のスケジュールにおける

遅延について評価をしており、10試行の平均を表している。分割数を大きくした時に、Fitnessの値は減少していくことがこの結果より確認できる。これは分割した航空機数が少なくなればなるほど、一つの航空機の良いスケジュールを探索する可能性が高まるため、スケジュールの最適性能が上昇したと考えられる。一方で、コンフリクトにおいては分割数が増えるほど大きくなることがわかる。これは、分割数が増えると、一つの群当たりの航空機が少なくなるため、群の中の航空機の組み合わせはコンフリクトを解消することができないため、コンフリクトの値が大きくなっていることが考えられる。

ケース2：分割数を固定した場合

次にケース2における分割の繰り返し回数を変えた時の結果を以下に示す。

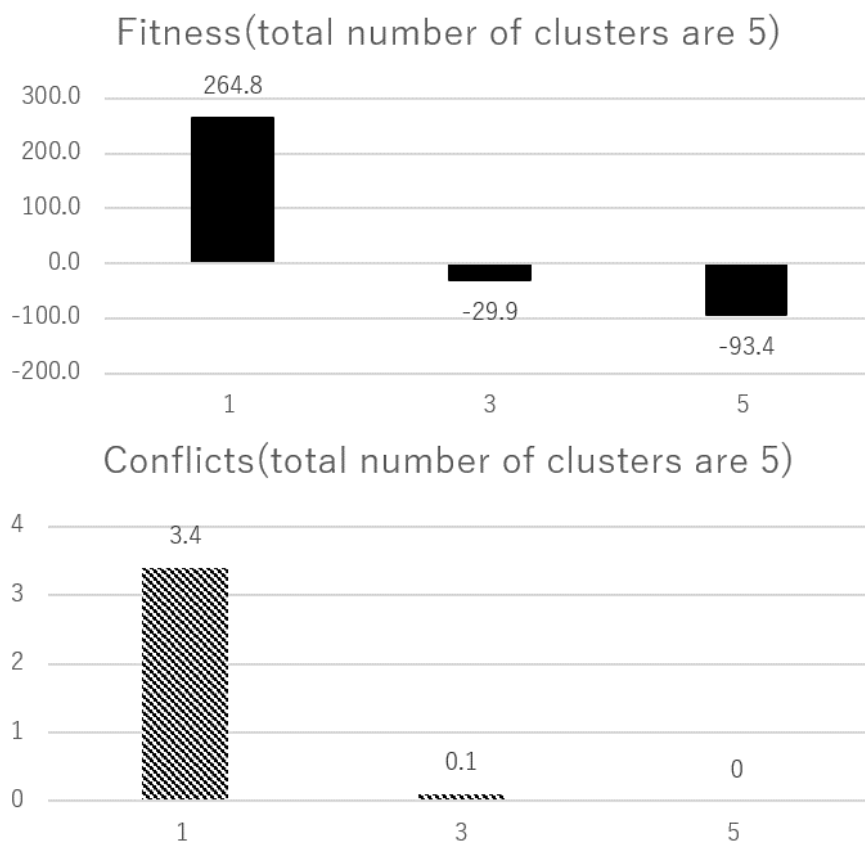


図 6.17 クラスタ数を5にした場合の Fitness とコンフリクト数の結果

図 6.17 の横軸は分割回数の違いであり、左から、3,4,5 回の分割回数における Fitness を表し、下部はその際のコンフリクト数を表す。ケース1と同様に、分割回数を大きくした場合にスケジュールの改善が見られた。分割回数の違いは先ほどの群の数と同様にスケ

ジューリングの性能差を生むことが分かった。分割回数を大きくしたときに、スケジューリング性能が大きくなり、分割数が5回の時に一番スケジューリング良い結果を得られた。特に分割回数を5回とした時、すべての試行回数においてコンフリクトがないスケジュールが得られ、従来手法よりスケジュールの遅延の少ないスケジュールが獲得できることが明らかになった。分割回数を変えた時には群の航空機の組み合わせに変化が生じる。組み合わせの変化によりスケジューリングが困難な場合もあるため、出来るだけ分割数を増やすことで良質なスケジューリングの探索の可能性を高めるように作用したと考えられる。

6.3.6 考察 1-1：分割および反復における最適化への効果

分析として総試行数の Fitness の平均値の推移、最良シードにおける Fitness の関係と航空機群の変化を示す。実験設定は分割の総世代数は 1000 世代とし、分割の回数は 20 回として再度実験を行い、分析を行った。

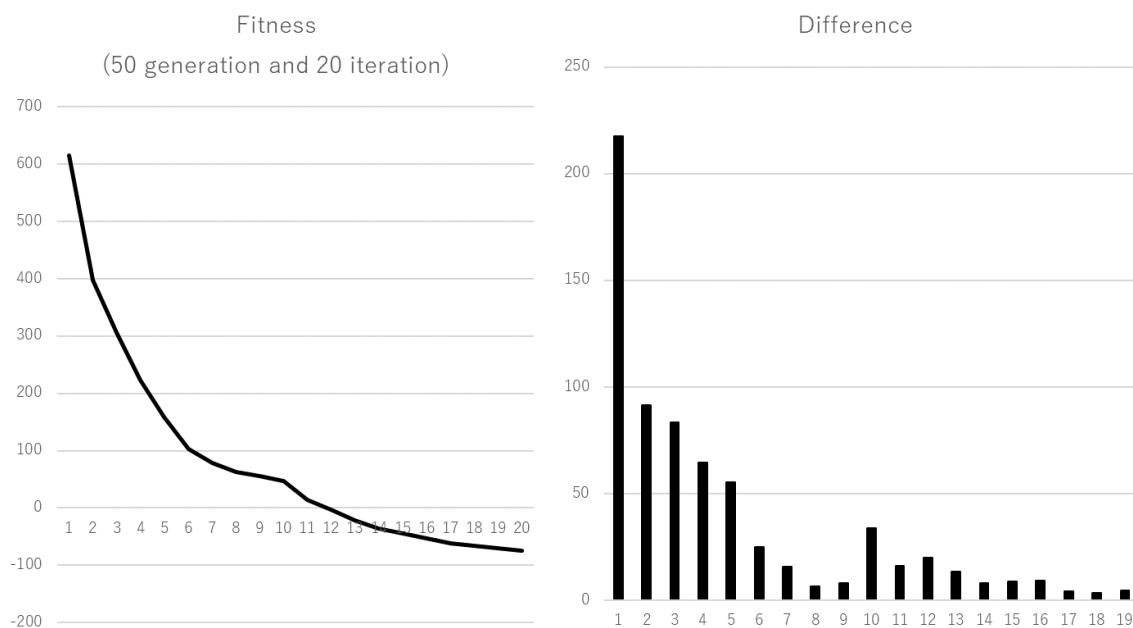


図 6.18 反復回数を 20 回とした場合の Fitness の推移

分割回数を 20 回とした場合には世代数は 50 世代であり、Fitness の推移を示した。図 6.18 の左のグラフは 10 試行の Fitness の平均の結果を示しており、右のグラフは一つ前の分割における Fitness と分割によって得られた Fitness との差を表し、スケジュールの改善の度合を示す。横軸の左の数字から 1 回目の分割と 2 回目の分割による Fitness の差を示しているため、数字は最大で 19 となっている。この結果からの初期世代は、その

時の世代よりも良いスケジュールを見つける可能性が、高いが、後半のほうは改善も軽微なものになっていくことがわかる。

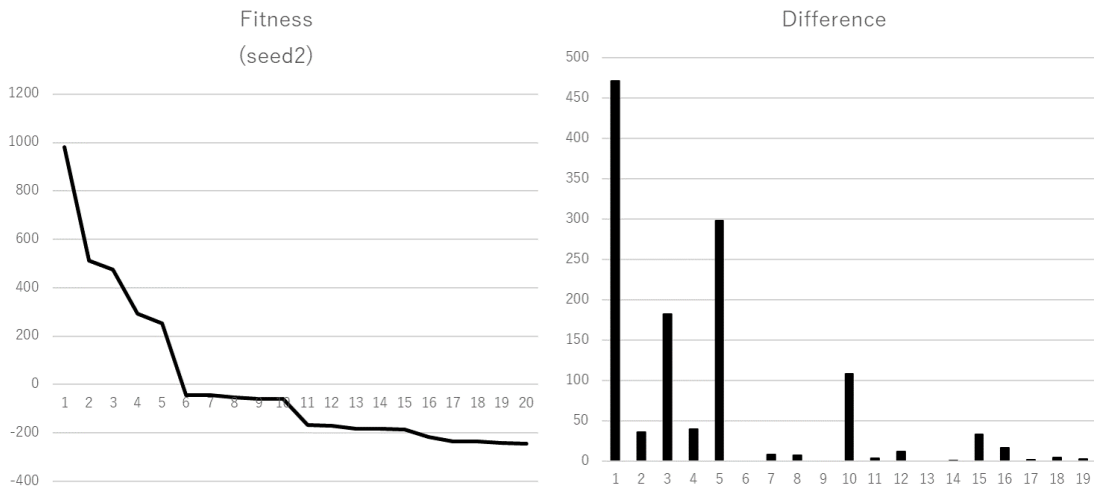


図 6.19 2 試行目における Fitness の推移

図 6.19 は最良シードにおける Fitness の推移及び、各回における Fitness の改善度を示す。図の見方は先ほどと同様であり、縦軸は Fitness，横軸は分割回数を示す。後半の世代は改善が見られる場合、見られない場合があることが確認できる。この時の群内の変化を以下に表す。

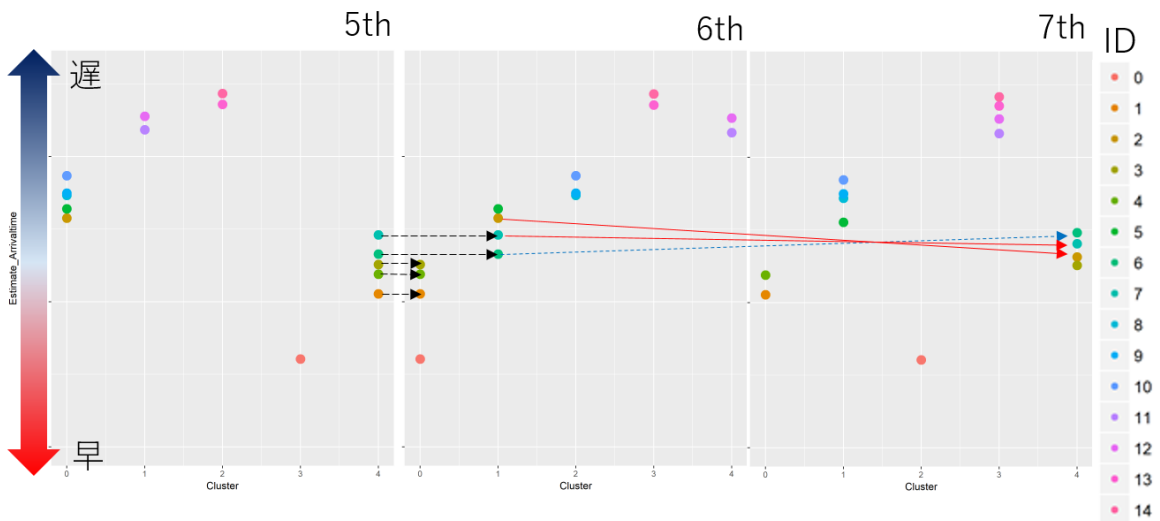


図 6.20 最適化毎の所属クラスタの変化

図 6.21 は具体的な着陸順の変化が生じ航空機の詳細データを表し、着陸順の変化が生

じた Aircraft 2, 6, 7 に関する予定到着時刻と実際の到着時刻, その時間差も表されている. 到着時刻を早める (Fitness の向上に相当) ためには航空機がどのクラスタに所属しているかが重要である. 図の右側の 5 回目と 6 回目の最適化においては, 6 回目のクラスタで Aircraft 2 が到着時刻が遅くなり, 4 分程度到着予定時刻から遅れたスケジュールを獲得している. このような変化は 4 回目におけるクラスタでは Aircraft 2 が Aircraft 6, 7 と異なるクラスタに属しているため, 着陸順の変化が生じえない. これが, 次の最適化においては, Aircraft 2, 6, 7 の航空機が同じクラスタに属し, 着陸順の変化が生じたため, 他のクラスタに所属する航空機の着陸予定時刻を早められた. したがって, 反復によるクラスタの変化が Fitness の向上に貢献することがこの例により示唆された.

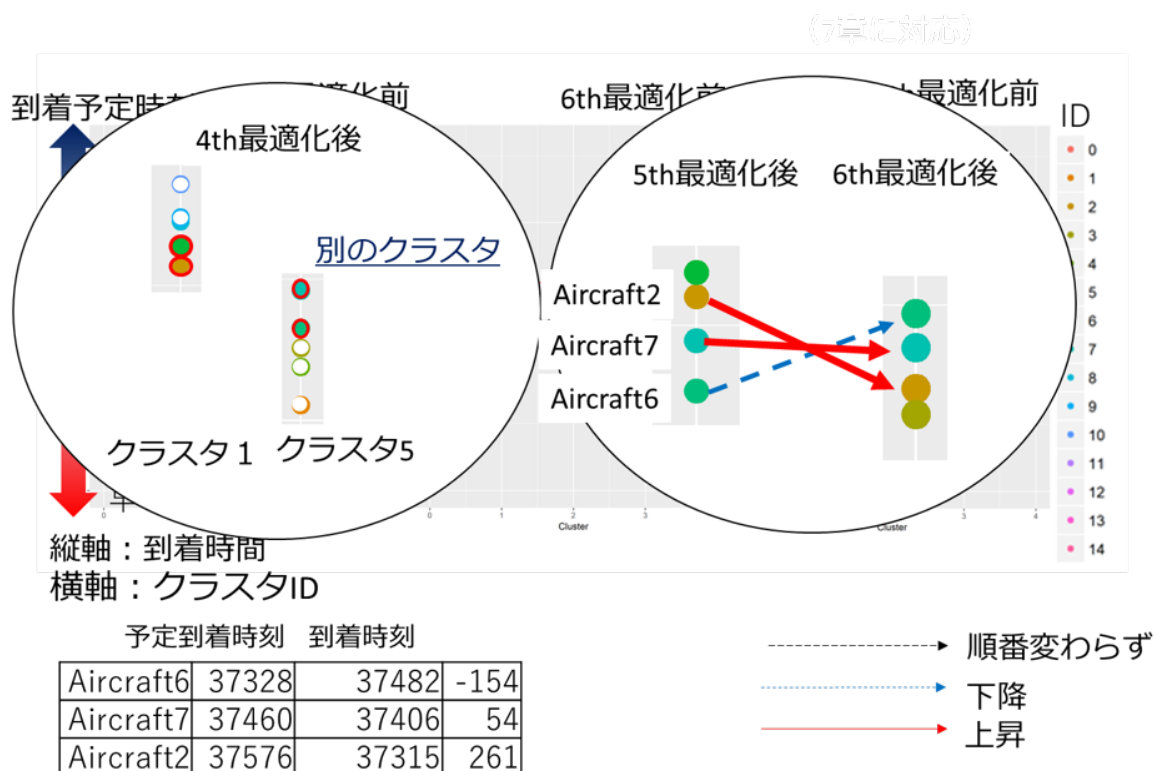


図 6.21 変化の生じた航空機の詳細

次に反復回数を 3 とした場合における分割数を 6 以上とした場合の結果を以下の表 6.4 に表す. 左列から, クラスタ数, Fitness(総遅延時間), 総コンフリクト数を表す. Fitness に関してはコンフリクトの生じない試行回数を母数として平均を求めた. コンフリクト回数に関しては, 10 試行におけるコンフリクトの平均回数を示す. 最適化対象の航空機は 15 機となっているが, クラスタ数がおおよそ半分である場合にはコンフリクトの生じる試行回数が存在していることがわかる. ただし, クラスタ数が 9 や 11 の場合にはコンフリクトを解消できるようなクラスタ数も存在するが, 傾向としてクラスタ数が最適化対象

の航空機の半数以上になった場合には実行可能な着陸順が生成されない恐れがある。

表 6.4 クラスタ数と総遅延時間と総コンフリクト数の関係

Cluster num.	Fitness	Conflict
6	-32.7	0
7	-104.556	0.1
8	-127.222	0.1
9	-169.8	0
10	-185.5	1.4
11	-189.1	0
12	-195.125	0.6

6.3.7 考察 1-2 : 分割反復最適化手法の解析

図 6.20 はスケジュールの改善が見られた分割 6 回目とその前後の分割された群内の航空機を表す。分析対象とした理由は、全試行での平均推移での 6 回目から 7 回目の変化に比べ、大きく変化が見られたからである。横軸の数字はクラスタの番号を示しており、縦軸は航空機の到着時間を表す。この図を見ると 5 回目から 6 回目にかけて群内の航空機の変化はなくスケジューリングの改善も最低限に留まる。一方で 6 回目から 7 回目にかけての変化は線で示す通り、群の分割に変化が生じていることがわかる。特に 6 回目から 7 回目の変化においては図 6.20 の真ん中の図に着目すると、クラスタ番号 1 における ID6 の航空機は ID7 と ID2 の後に着陸するスケジュールとなり、着陸順に変化が生じていることがわかる。これは、5 回目のクラスタリングによって分割された群ではこの航空機 ID6,7,2 が同じ群に所属していないため順位に着陸順の変動が生じず、新たなクラスタリングによって航空機の組み合わせに変化が生じたためにスケジュールが改善されたことがわかる。

表 6.5 は左から全航空機 15 機の ID, スケジュール前と後での着陸時刻と初期速度と最適化によって獲得した速度, サブルートのパラメータ値をそれぞれ示している。この表より、航空機のスケジュールは 15 機中 11 機は初期速度による着陸時刻よりも早く着陸している。遅延した 4 機をみると 1 機のみ 2 分の遅延となっているが、2 分の遅延は十分運用可能な範囲内であるため、獲得されたスケジューリングは実行可能なスケジューリングであることが言える。これによりスケジュールの改善が見られる場合には群の中の航空機の組み合わせに変化が生じ、この変化により、探索し得なかった組み合わせをによりスケジュールが変化したことがわかり、分割し最適化を繰り返すことの効果を確認すること

が出来た。次に，着陸順の観点から考察を行う。表の Order は 15 機の最適化後の着陸順を表している。表の ID は航空機番号を示し，この ID の順番が FCFS の順番となる。最良シードによる順位は，ID 3 と ID 5 の序列，ID8 と ID6 の序列に変化が生じていることがわかる。通常であれば，ID 3 は 3 番目に着陸を行うが，本シードでは 5 番目に着陸し，ID 5 は逆に FCFS の順番では 5 番目に着陸予定の順番が，3 番目に着陸を行い，順番の変更が起こっていることがわかる。また，ID 6 に関しては，FCFS の順番で 6 番目に対して，実際の順番では 8 番目となり，ID 8 に関しては実際の順番では 6 番目となっている。

表 6.5 航空機の詳細スケジュール (seed2)

ID	Actual_Time	Estimate_Time	Initial_Speed	Actual_Speed	Detour	Order
0	36606	36651	347	364	1.29E-06	1
1	37047	37086	390	409	0.008723	2
2	37317	37312	473	494	0.423684	5
3	37257	37296	361	377	0	4
4	37193	37229	438	459	0.054821	3
5	37482	37498	473	476	0.00321	8
6	37442	37328	442	404	0.618643	7
7	37389	37400	416	432	0.182076	6
8	37712	37760	434	455	0.068162	9
9	37749	37781	367	380	0.011175	10
10	37832	37828	363	361	0.00106	11
11	38177	38212	506	525	0.093813	12
12	38221	38246	356	366	0.0014	13
13	38350	38392	456	477	0.057146	14
14	38406	38405	520	529	0.165019	15

表 6.6 は実際の最適化対象の航空機の実際の到着時刻と着陸順を表しており，列の左から航空機の ID，実際の到着時刻，航空機のサイズ，着陸順を示す。実際の到着時刻に関しては，シミュレーターの精度に依存するため，比較することが困難だが，同じような着陸順を獲得できている。特に 9 番目以降の着陸順においては，ID 13 と ID 14 以外では一致している。この 2 つの航空機の順番が客店している理由は，ID11 の航空機サイズが Heavy カテゴリであり，サイズが Heavy の航空機が Mideum の前におる場合にはより大きな時間間隔が必要となる。このケースでは ID 14 と ID 15 の管制間隔が大きくあり，ID13 と ID 14 の間隔はそれほど大きくなかったため。ID 13 は ID14 との順番は変更したが，ID13 と ID 15 との順番の変更は起こらなかった。本来は FCFS の原

則に則るため順番の変更は生じにくいですが、本ケースでは順番の変更が確認された。

表 6.6 管制運用での実際の到着時刻

ID	Actual Time	size	Order
0	36785	M	1
1	37409	M	2
3	37636	M	6
4	37543	M	4
5	37588	M	5
6	37463	M	3
7	37746	M	7
8	37831	H	8
9	37924	M	9
10	38056	M	10
11	38203	H	11
12	38264	M	12
13	38581	H	14
14	38470	M	13
15	38727	M	15

図 6.22 は最良シードにおける全航空機の航行履歴を示している。一番はじめに入域する ID0 の航空機が空港周辺内で航行を開始する $t = 0$ から一番最後に入域する ID14 が滑走路に着陸する時間である $t = 3110$ までの履歴のうち、それぞれ t が 1000,1250,1500,1750,2000,2250 の時の航跡データを示している。 $t = 1000$ 時点においては、一番初めに着陸する ID 0 の航空機が最終合流地点に到着前であり、いくつかの ID 8 までの航空機が航行を始めている。 $t = 1750$ 時点になると、最終進入地点を中心にかんがりの混雑が見られる。 ID2 から ID8 までの 7 機が 4 地点から合流を待っている。この集中的な状況において合流地点での間隔を維持するため、ID6 や ID 7 は積極的に迂回を利用する。

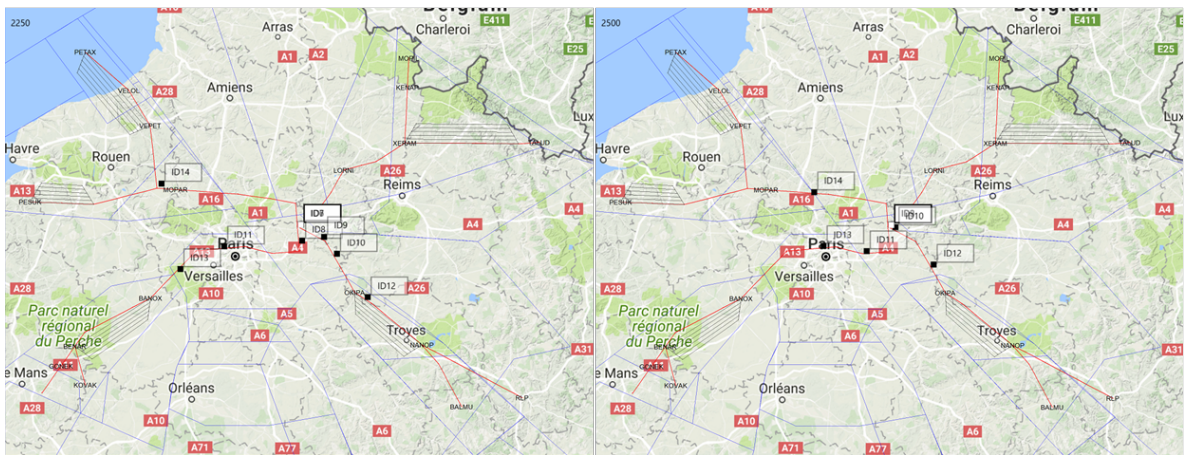
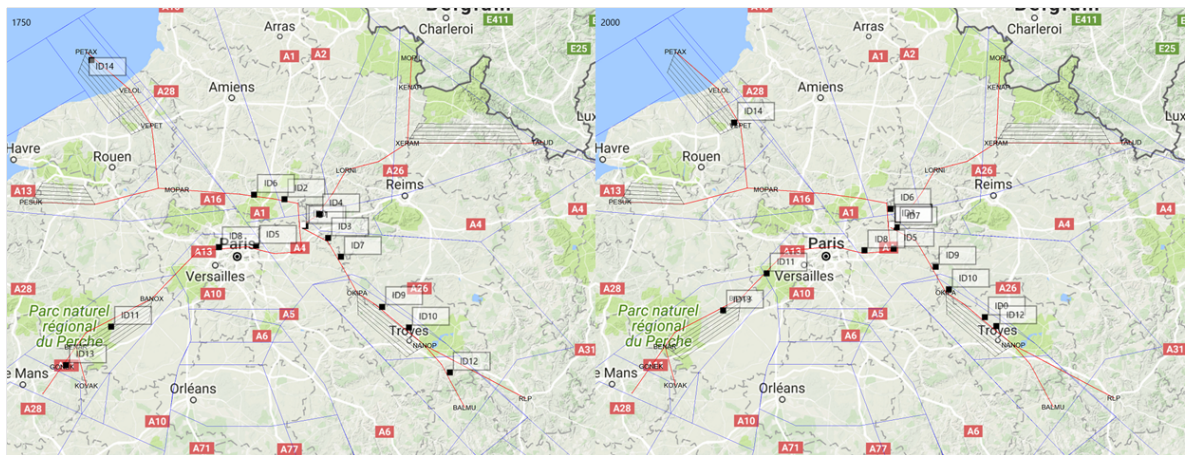
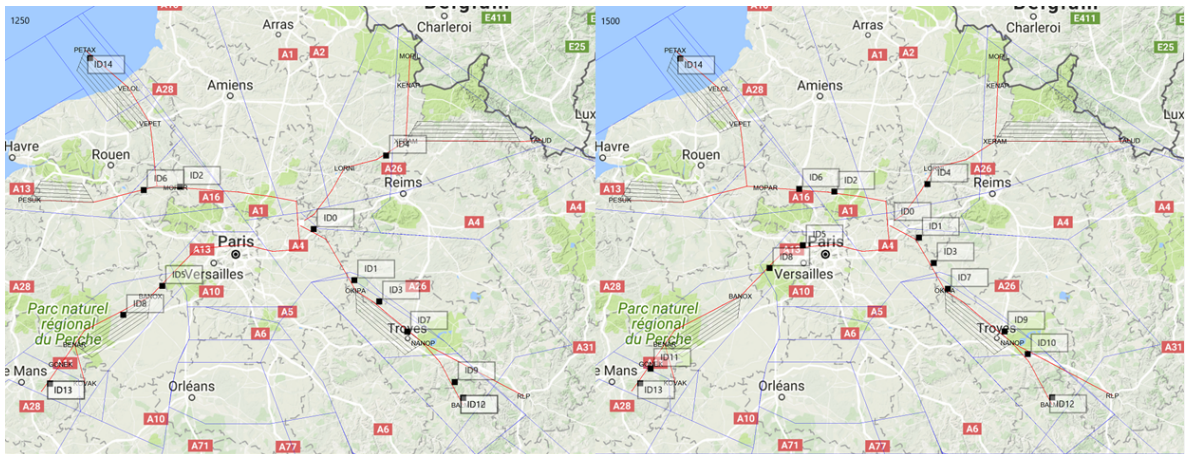


図 6.22 最良シードにおける航空機の遷移

図 6.23 は初期進入速度を変更せず、迂回経路を利用しなかった場合の航跡を表している。先ほどの図と同じくそれぞれ t が 1000, 1250, 1500, 1750, 2000, 2250 の時の航跡データを示している。 $t = 1750$ を見ると ID 2 と ID 6 の間隔が短く、初期速度ではこの 2 機にコンフリクトが生じる可能性がある。このように初期速度を変更しない場合にはコンフリクトが生じるため、過密状態での迂回経路と速度変更が有効である。

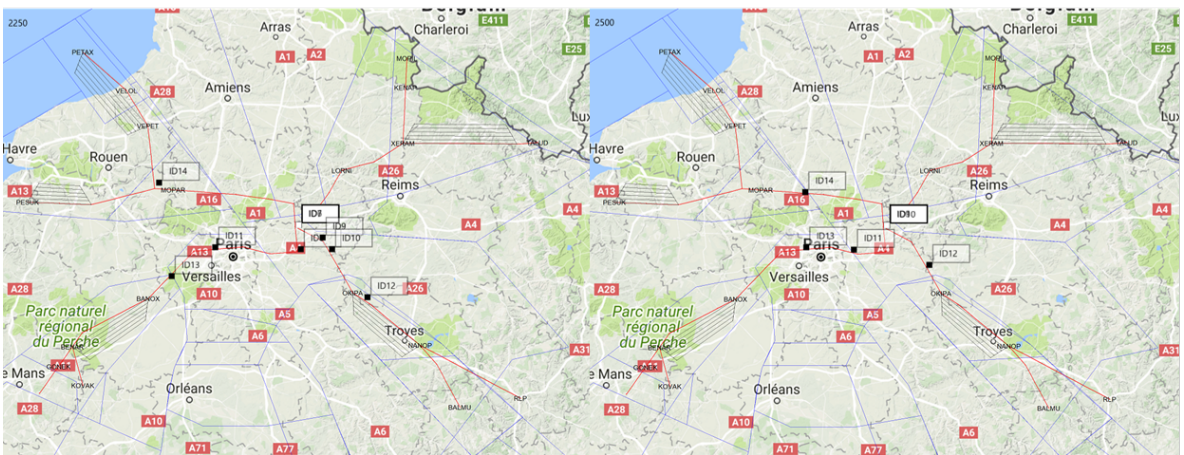
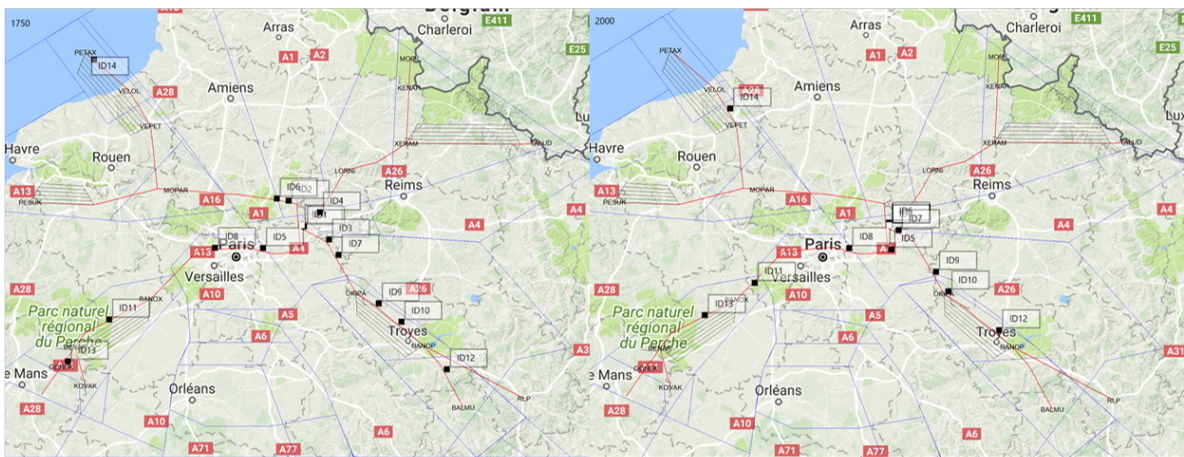
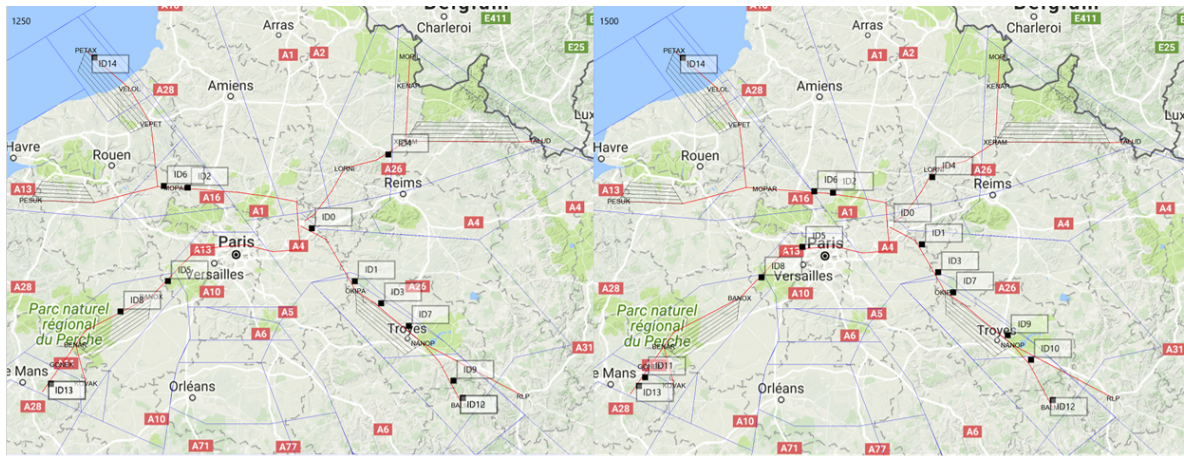


図 6.23 初期速度迂回無しでの遷移

6.3.8 考察 1-3 : 反復最適化によるクラスタの変化

図 6.24 は主要なクラスタリング手法を適用した場合の最適化性能を示している。縦軸は Fitness (全航空機からスケジュール逸脱量の合計)、横軸は最適化の反復回数合計を示している。それぞれの手法は第 3 章で説明した 7 つの手法を用いており、上から、重み付き平均法、重心法、完全リンク法、k-means 法、McQuitty 法、メディアン法、単リンク法である。それぞれのグラフは 31 試行の平均値であり、1 回の最適化がい終わるごとに全航空機のスケジュール逸脱量 (Fitness) を算出しプロットしたグラフである。Fitness の推移をみると初期の最適化では、淡い緑色の線である k-means 手法が他のクラスタリング手法に比べ大きな値を示していることがわかる。しかし、5 世代目以降で他の手法と同等の性能を維持し、その後、10 世代以降では平均値が他の手法より上回っていることがわかる。手法の差異を評価するため、k-means 手法と 2 番目に性能が良いメディアン法を統計的検定を行った。検定方法としては、ウィルコクソンの順位和検定 (マン・ホイットニー検定) を適用したところ、5 % の統計的有意差が確認された。

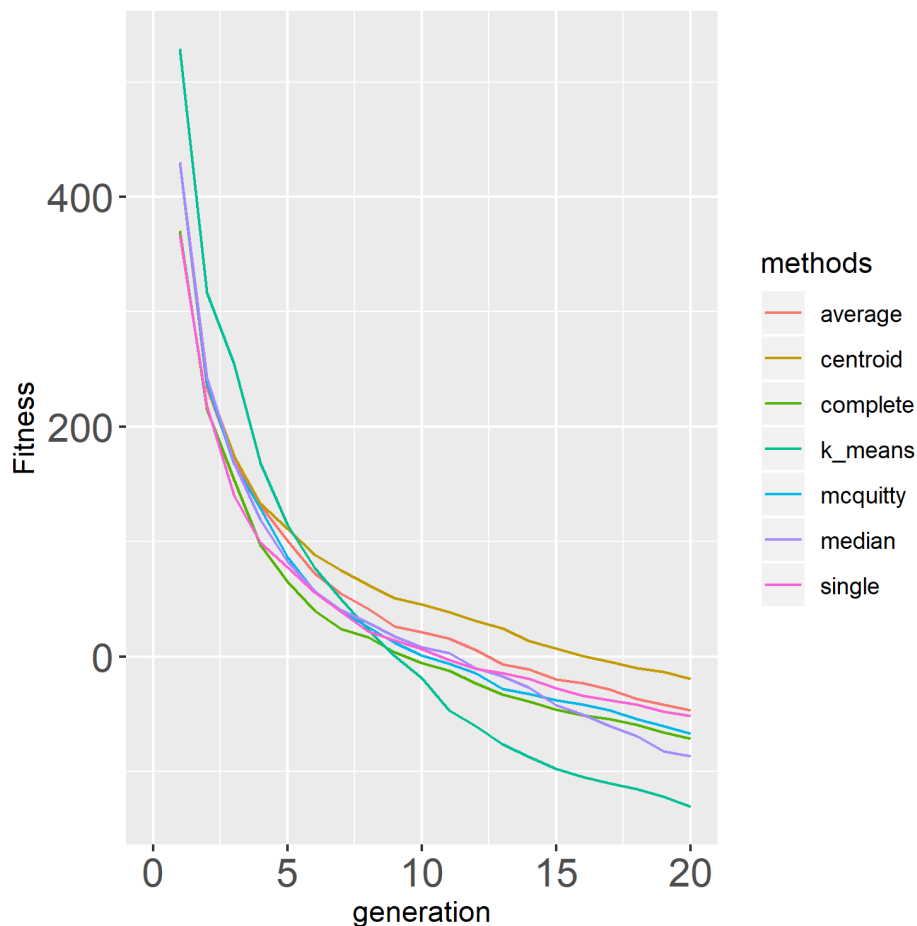


図 6.24 各クラスタリング手法における Fitness の推移

6.3.9 考察 1-4：反復回数と世代の性能に与える影響に関する考察

本章の結果として、反復回数を多くすれば、Fitness である総遅延時間を減少させることに貢献するが、最適化の経過に関しても同様に重要である。また、航空機の機数、ランダム (試行) における性能の影響に関しても実運用においては重要な要素となるため、本節で考察する。実験設定としては以下を想定する。航空機のシナリオとしては、15機と30機の2パターン、反復回数と世代数の関係を以下の表にまとめた。15機は混雑時を想定した交通量における性能評価であり、30機としたのは、同様の交通流において、2倍の航空機数で提案手法の性能に変化がないかを検証するため30機とした。

クラスタリングと反復回数の結果

ここでは、クラスタ数が同様の場合における反復回数と世代数における最適化性能の違いについてまとめる。ここでは合計の評価回数を1000回として、5パターンパラメータ

を変更し実験を行った。本実験においては、クラスタ数を5で固定した場合の結果をまとめる。

表 6.7 航空機の詳細スケジュール (seed2)

クラスタ数	反復回数	世代数
5	5	200
5	10	100
5	20	50
5	25	40
5	40	25

結果

15 機の場合

図 6.25 は毎回の最適化後の Fitness の推移を表しており、it は iteration, gen は generation を示す。Fitness の推移は平均値であり、30 試行の平均である。グラフの推移をみると反復回数を多くすれば多くするほど、Fitness の値が小さくなることを示している。そのため、15 機のシミュレーションにおいては、反復回数を増やせば増やすほど、最適化性能の向上を見込むことができる。

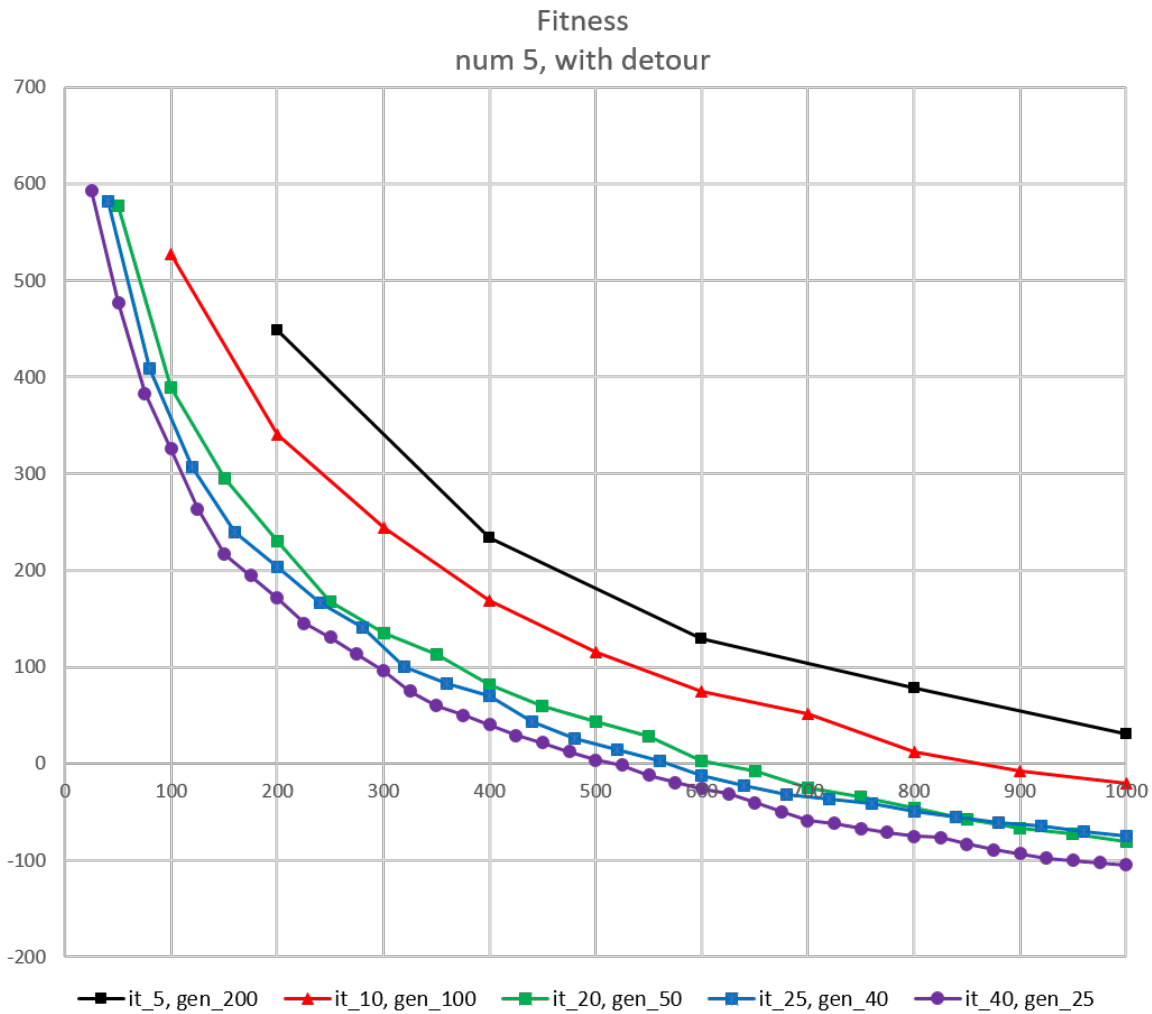


図 6.25 世代ごとの Fitness の推移

図 6.26 は 15 機における初期と最終世代における最適化終了後の全試行の Fitness の箱ひげ図を表している。縦軸は Fitness である，全航空機の到着予定時刻との差分の合計，それぞれ二つの箱ひげ図は初期の最適化後の性能と最終最適化後の性能の箱ひげ図をそれぞれ表す。反復回数を少なくした場合には初期世代における性能は大きくなり，箱ひげ図を四分位の幅が広くなり，試行におけるばらつきが大きくなるため，早期に安定した解を獲得したい場合には反復回数を少なくした方が良好な性能の着陸順を得ることができる。

一方で最終世代の性能を見ると反復回数を大きくするとより小さい Fitness を獲得することが可能であるが，反復回数を 25 以上にした時の性能差は四分位の幅の広さを比較しても大まかに変化が生じないため，ある程度の反復回数の増加は Fitness の改善に貢献するが，20 回程度の反復回数で性能に変化がなくなることがわかる。

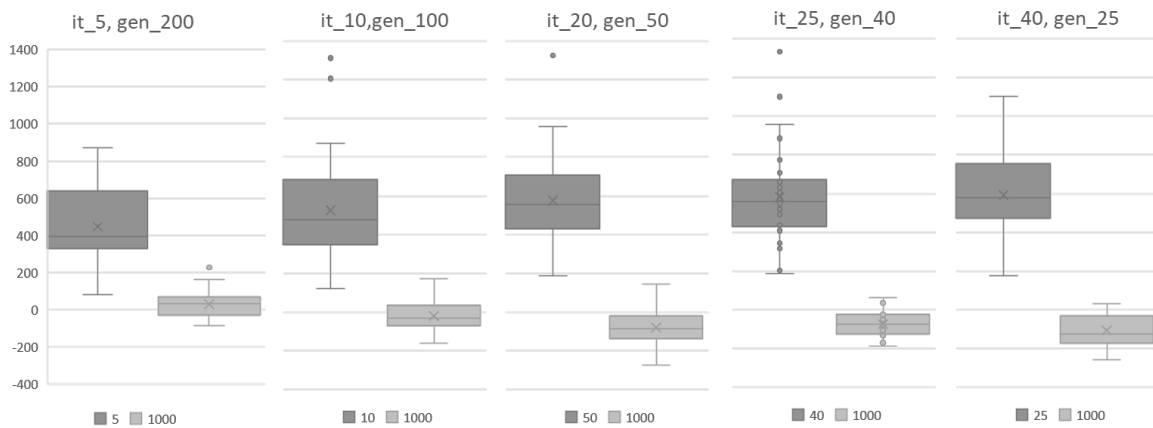


図 6.26 初期世代と最終世代の性能差

図 6.27～6.31 は各航空機における速度，初期速度と比較した時の偏差，到着予定時間，予定到着時刻との偏差の全試行の箱ひげ図をそれぞれ表している．全体のおおまかな傾向は反復回数に依存せず一致していることがわかる．ただし，速度，到着時刻の箱ひげ図をみると，航空機によっては到着時間に分散を持つ航空機があることがわかる．例えば，ID3，ID 4 に着目すると航空機の到着時刻はほとんど同じであるため，通常の場合では ID 3 のほうが順位が早いケースがあるが，ID 4 と順位が入れ替わるケースがいくつか見られる．図 6.29 においては，全試行のうち 1 試行で ID 4 の到着時刻が他の試行に比べ遅くなるケースがあることがわかる．このように着陸順の安定度の観点では順位変動があるため，安定度の観点では課題が残る．

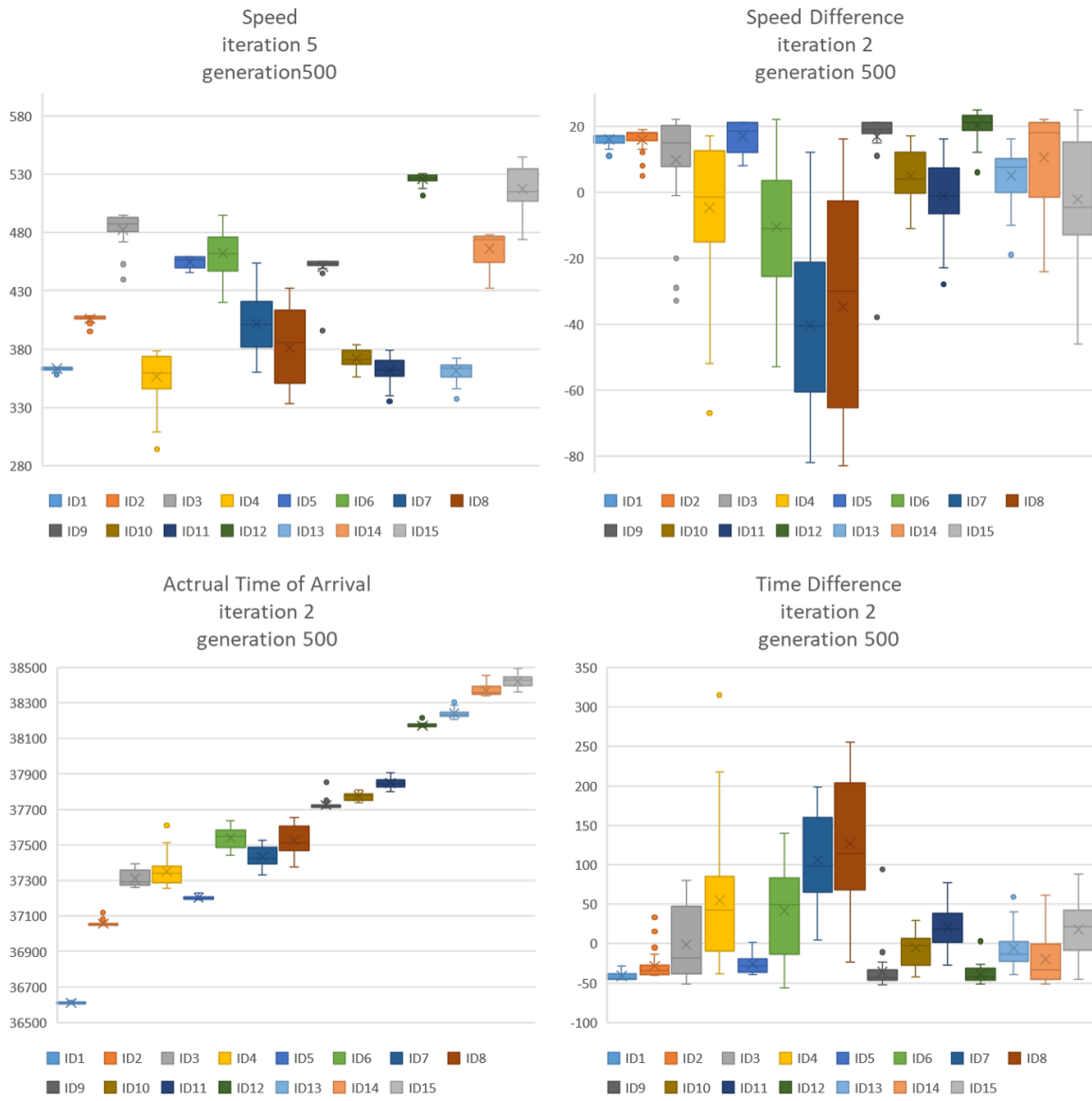


図 6.27 各航空機の詳細 (反復回数 5, 世代数 200)

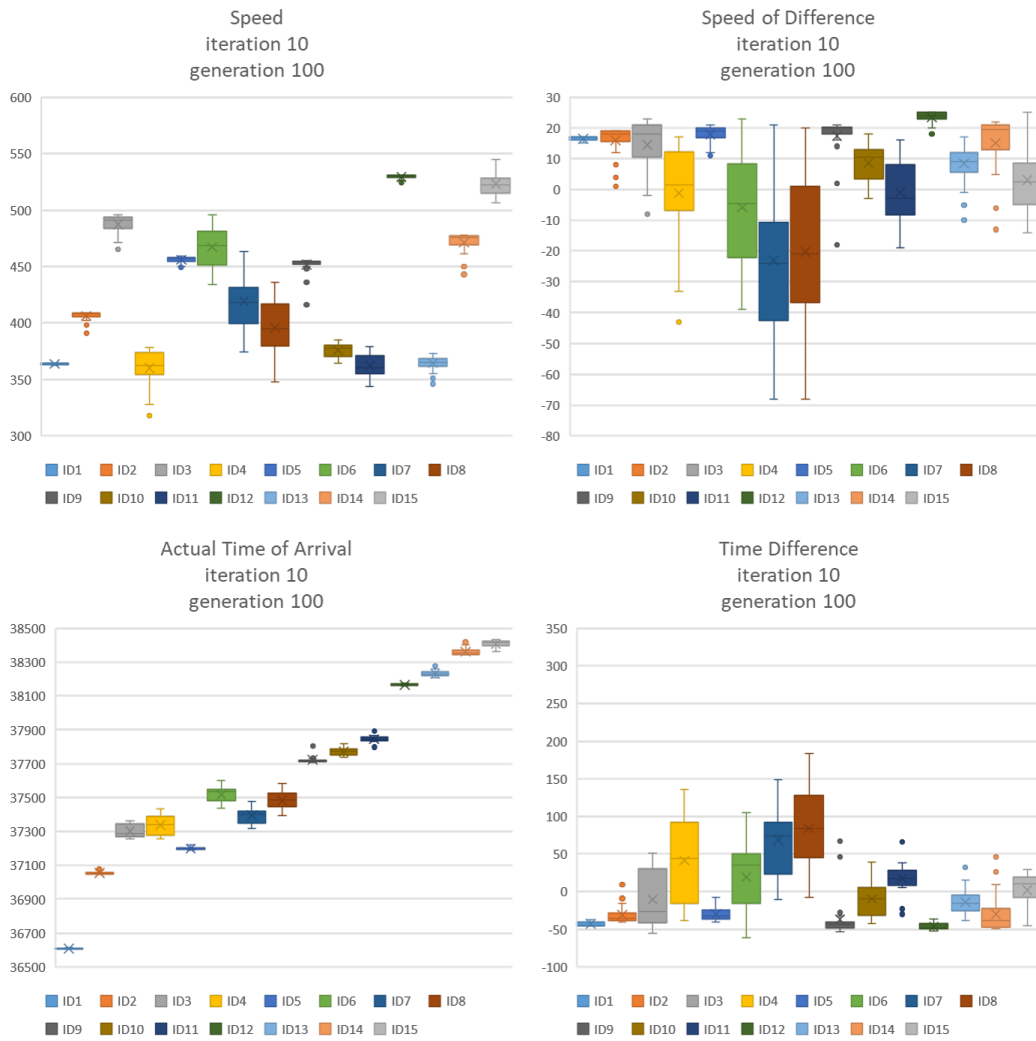


図 6.28 各航空機の詳細 (反復回数 10, 世代数 100)

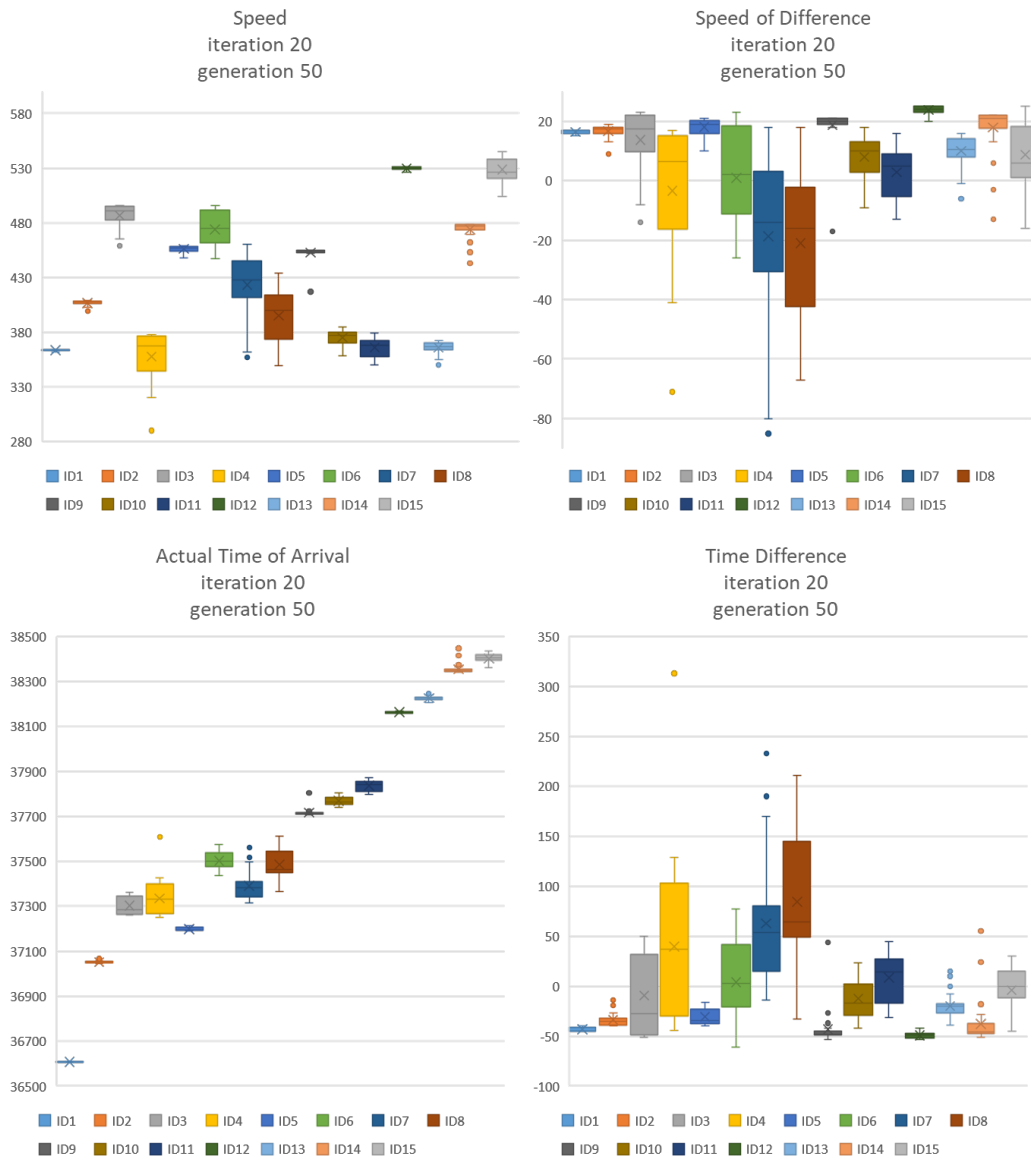


図 6.29 各航空機の詳細 (反復回数 20, 世代数 50)

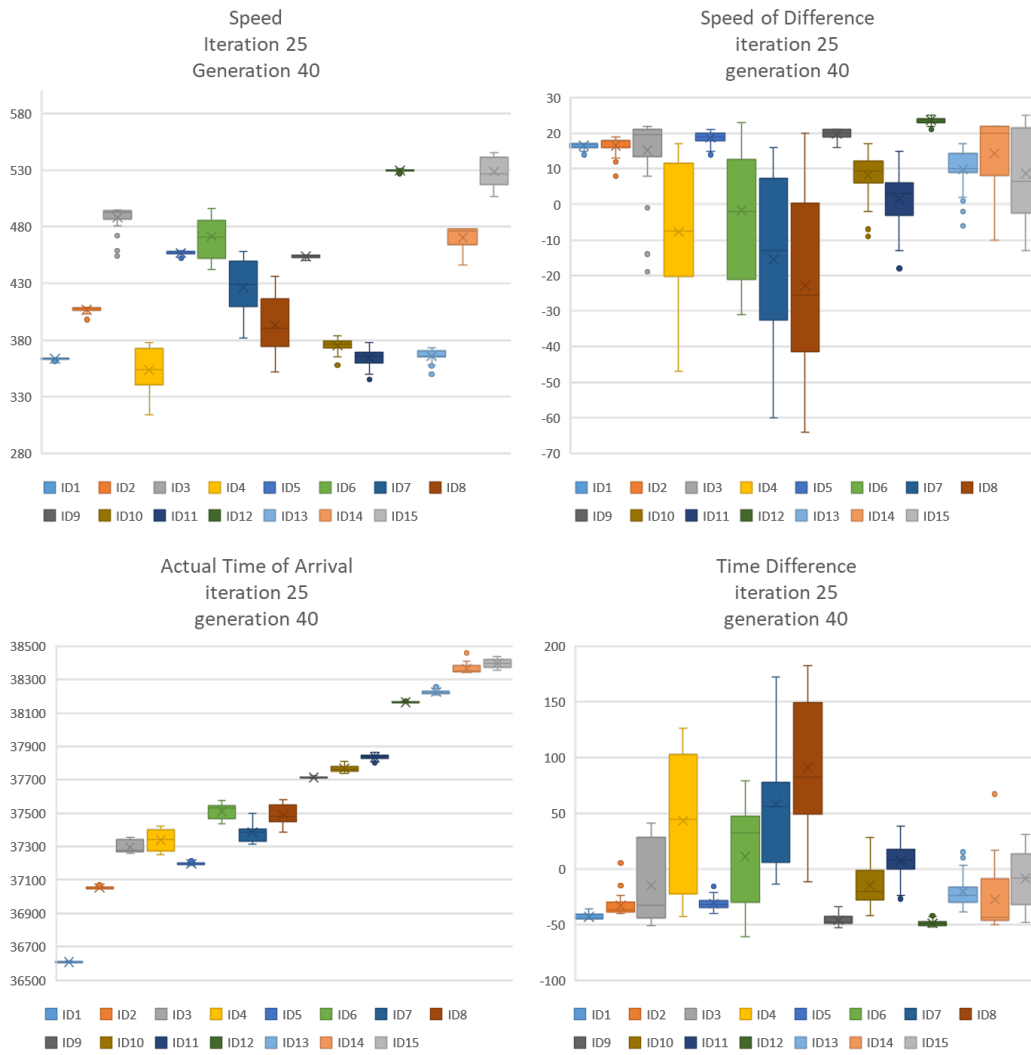


図 6.30 各航空機の詳細 (反復回数 25, 世代数 40)

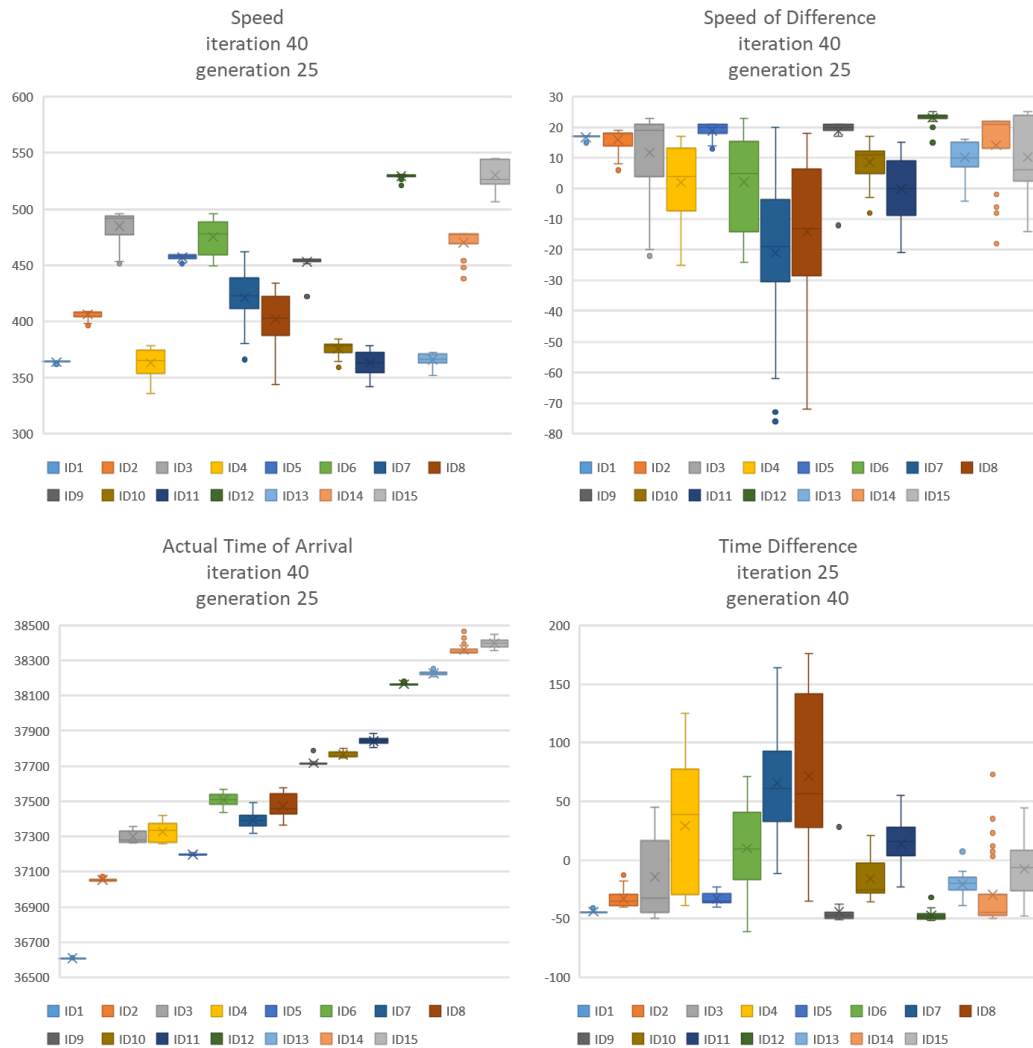


図 6.31 各航空機の詳細 (反復回数 40, 世代数 25)

30 機の場合

図 6.32 はスケジューリング対象の航空機が 30 機における Fitness の推移を表しており、5 つのグラフの違いは反復回数と世代数の違いであり、15 機のケースと同様に評価回数を同様にし、反復回数をそれぞれ、5 回、10 回、20 回、25 回、40 回とした場合の世代ごと性能を示す。このグラフは、1 回の最適化が終わった後の Fitness をプロットし、それを結び合わせた結果であり、31 試行の平均値の値である。

Fitness の性能は 15 機とほとんど同じような推移を示しており、評価回数と同じ場合には反復回数が多いほど良好な結果である。ただし、反復回数を性能の向上の大きさは小さくなっていくことがわかる。

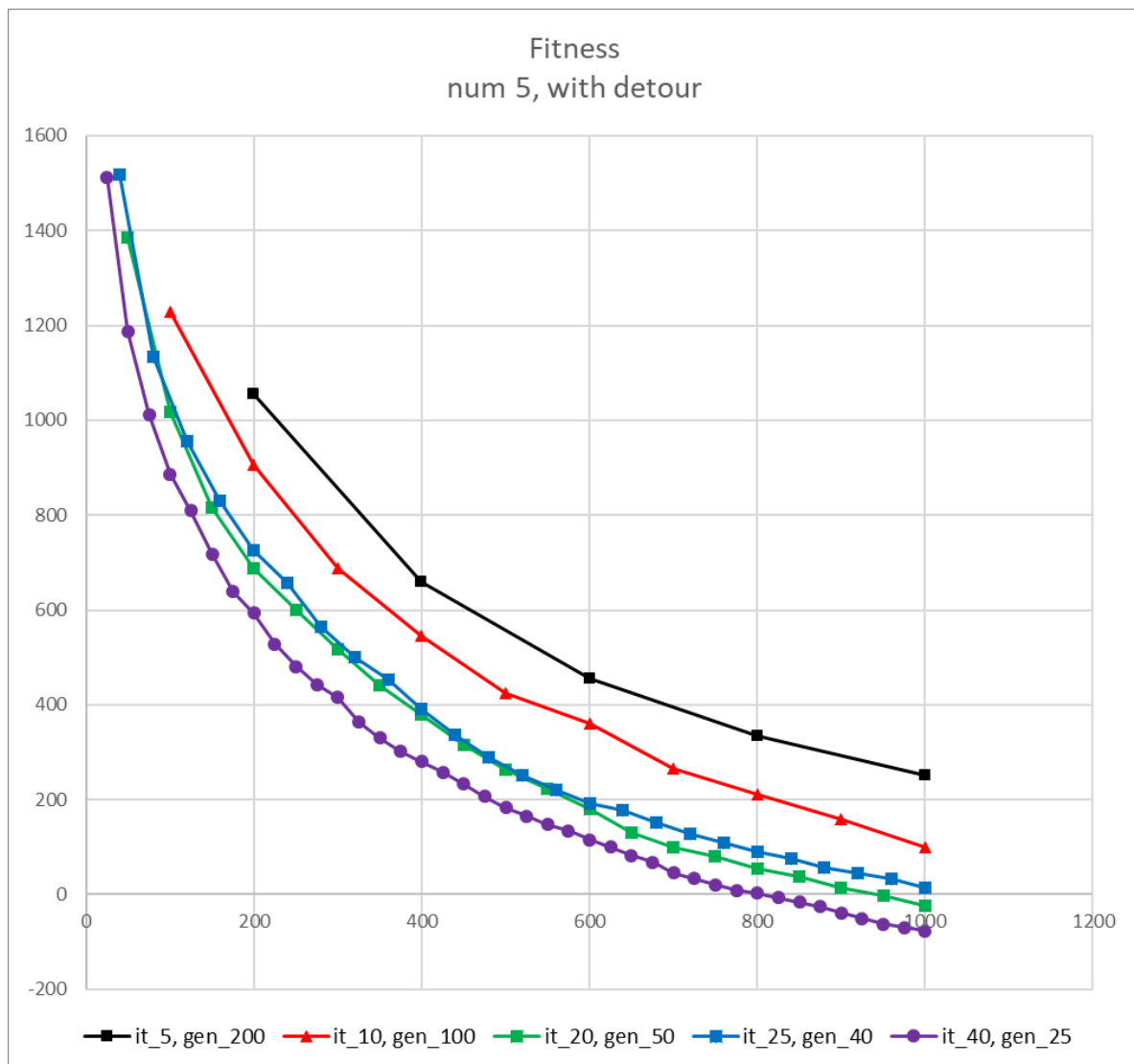


図 6.32 世代ごとの Fitness の推移

図 6.33 は初期及び、最終世代における最適化後の Fitness を表し、箱ひげ図は 31 試行の Fitness をまとめたものである。図の下に数字が表されてるが、これは、初期の最適化の終了後の世代数と反復回数が最後の最適化における Fitness を示している。30 機においても 15 機の同様の傾向を示しており、反復回数を多くすると初期の最適化における Fitness は反復回数の少ない場合のほうが良い結果を示していることがわかる。一方で最終世代の Fitness の推移をみると反復回数の多いほうが最終世代での Fitness が良くなり、分散の幅も小さくなる。つまり、少ない世代での反復回数を多くすることは最適化安定度の増加に貢献することがわかる。

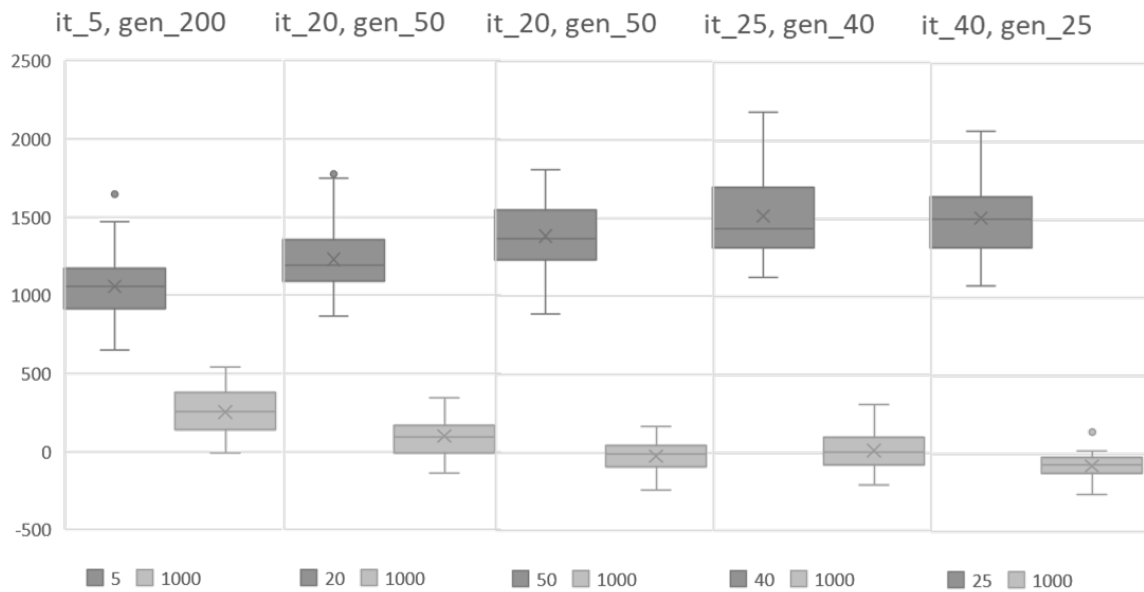


図 6.33 初期世代と最終世代の性能差

図 6.34～6.38 は各反復最適化後の航空機の詳細を表しており、4 つのグラフから構成される。左上の図は速度の変化、右上の図は、初期速度との差分、左下の図は実際の到着する時刻、右下は到着時間に対する差分をそれぞれ示している。

全体の傾向として、到着時間の分散が 15 機のスケジュールに比べて大きくなっていることがわかる。反復回数が分散が 1 分以上となる航空機が 8 機あり、これを見ると、反復回数が多くても 3 割程度の航空機において到着時間の分散を持つことがわかる。このことから、スケジュール対象の航空機数の航空機数が増えるほど各航空機の到着予定時間の分散が大きくなることが明らかとなった。

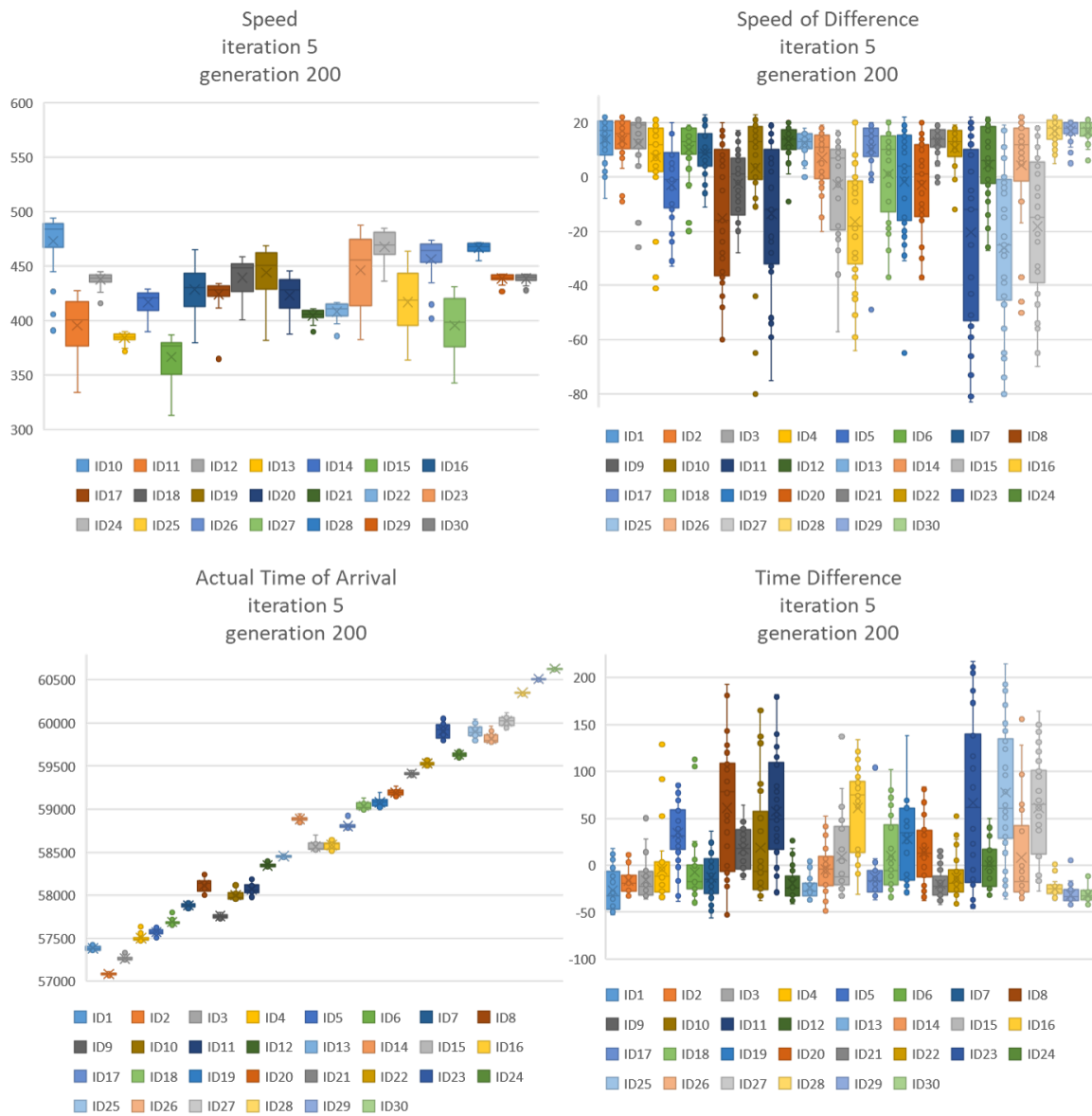


図 6.34 各航空機の詳細 (反復回数 5, 世代数 200)

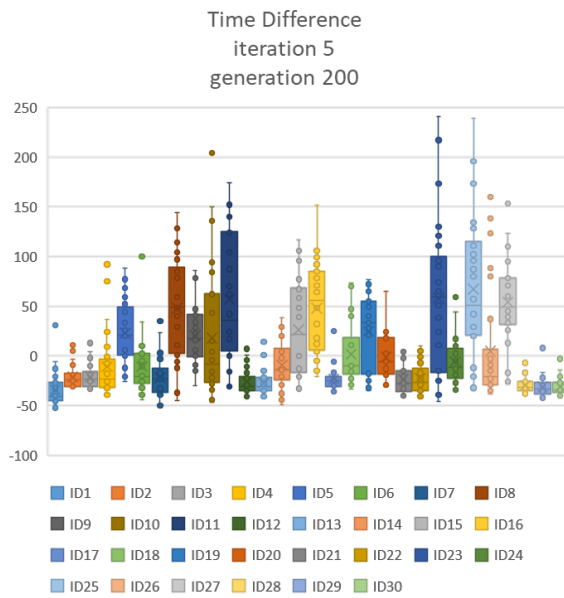
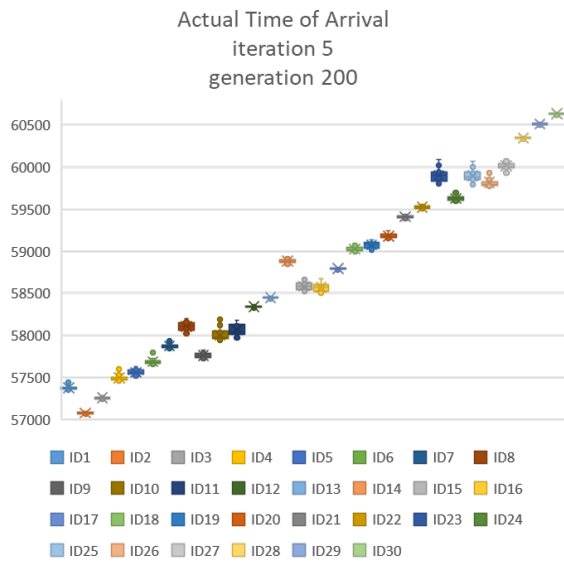
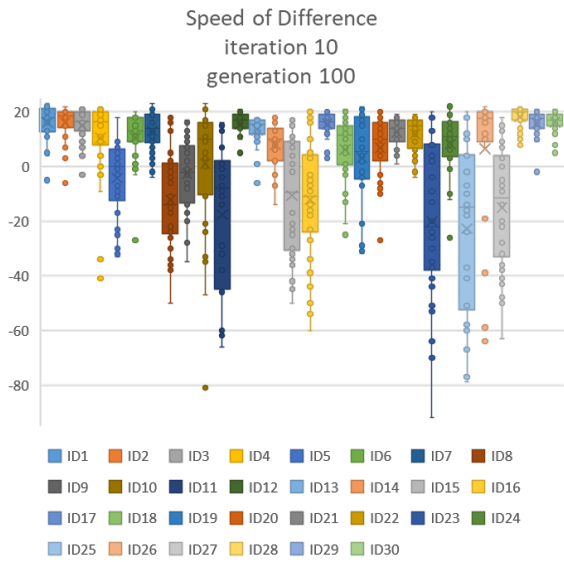
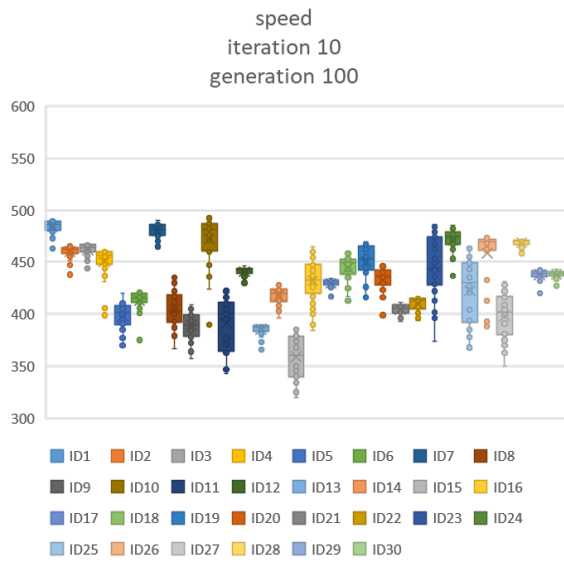


図 6.35 30 機における各航空機の詳細 (反復回数 10, 世代数 100)

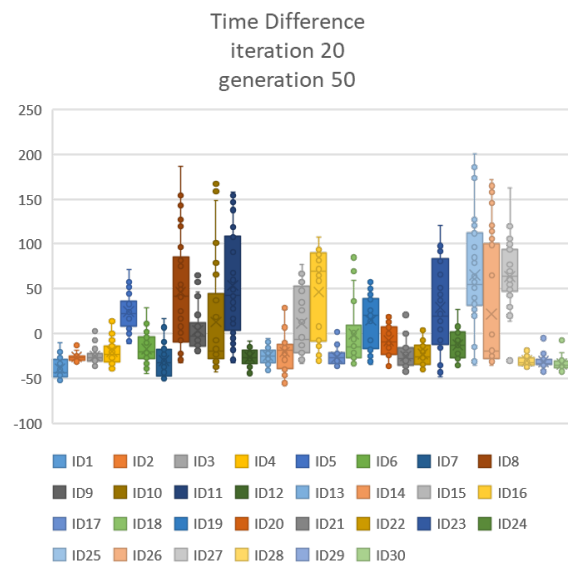
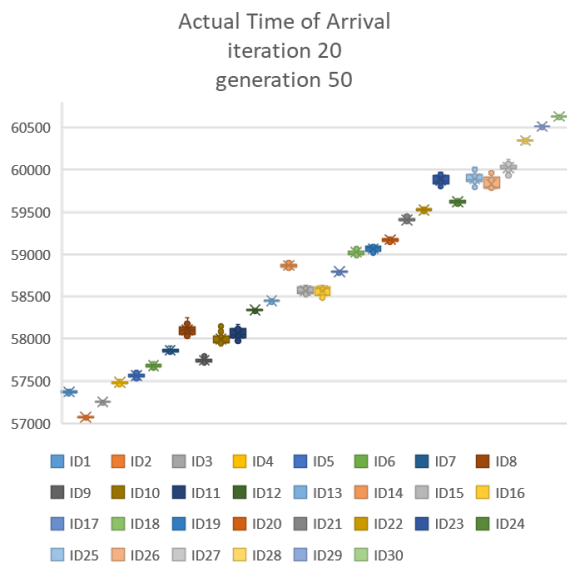
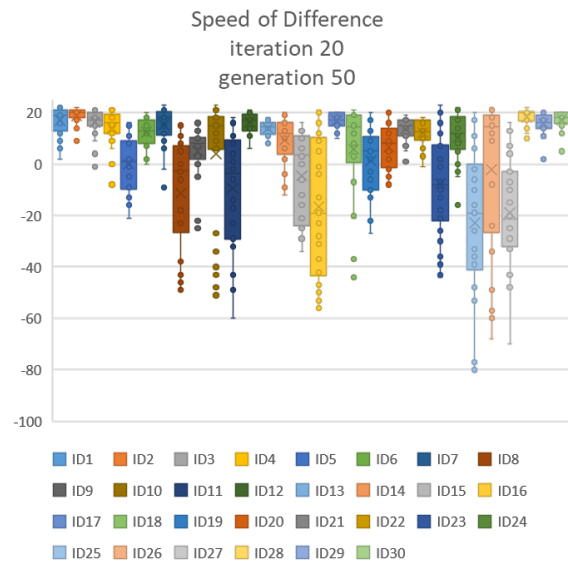
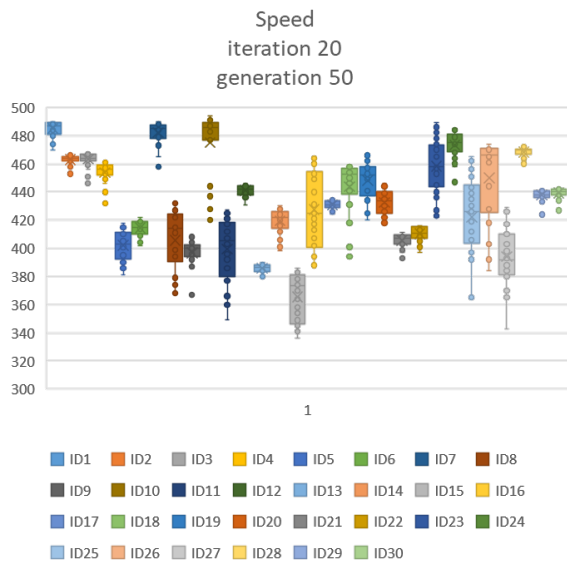


図 6.36 各航空機の詳細 (反復回数 20, 世代数 50)

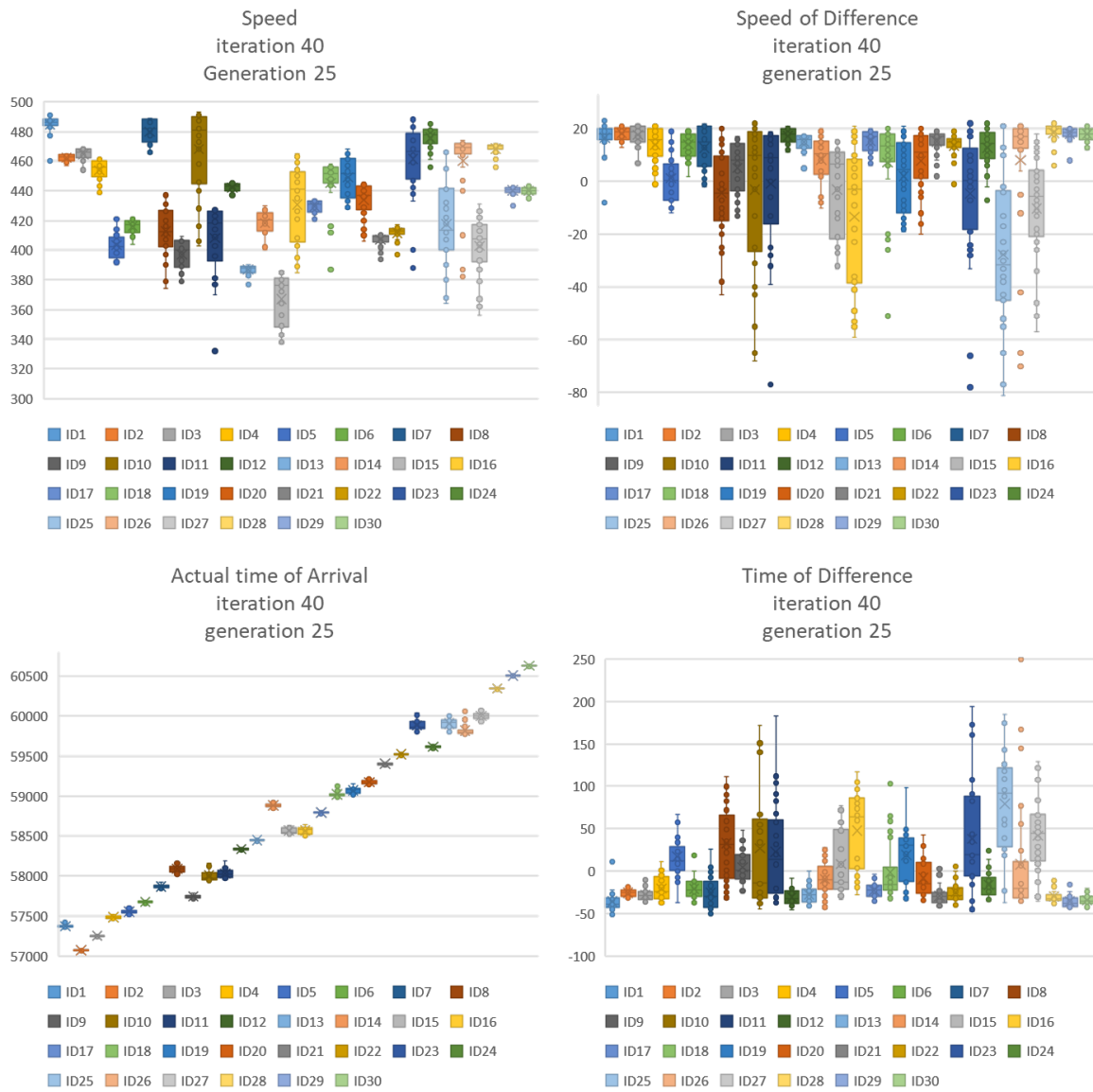


図 6.37 各航空機の詳細 (反復回数 25, 世代数 40)

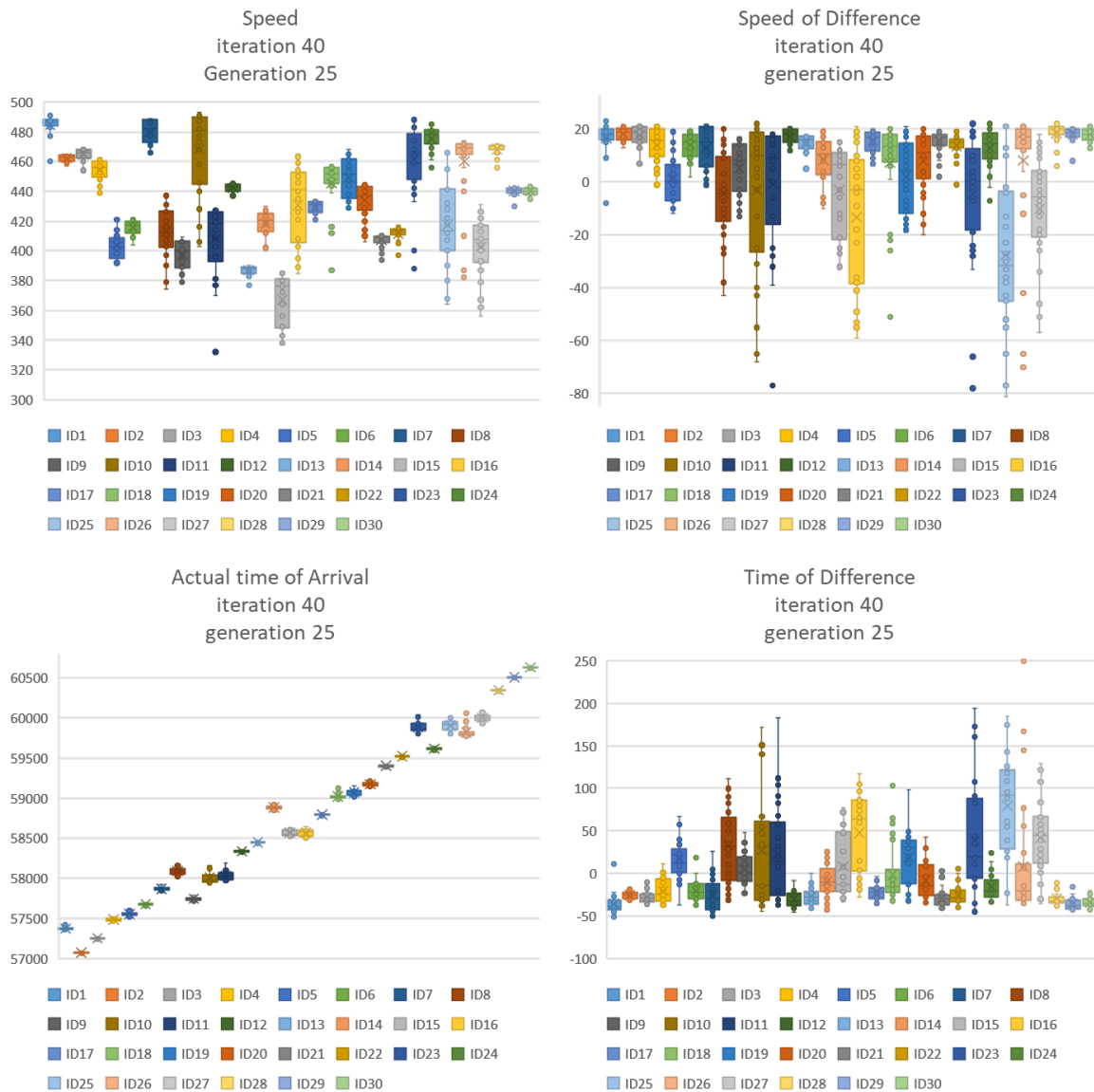


図 6.38 各航空機の詳細 (反復回数 40, 世代数 25)

6.3.10 考察 1-5 : 固定経路による分割反復最適化手法の限界

実験

実験内容固定経路の理論的な着陸数の導出は難しく本論文では実験的に限界量を算出する。この時の実験条件としては、解析を用意するために入域地点は 2 地点とし、北西方向の入域地点から航空機が入域する。スケジュール対象の航空機はすべて同じサイズの Medium であり、入域時間は同一入域地点に対する航空機の時間間隔は最低でも 50 秒以上間隔を保つ。これは入域の時点において初めからコンフリクト状態を防ぐためである。また、全航空機は 1000 秒以内に全航空機が入域するよう乱数により 10 秒単位で入

域時間を選択する。入域地点における初期速度の範囲は4 4 2 Nt から5 2 1 Nt の範囲から乱数により選択する。これは実際のシャルル・ド・ゴール空港におけるトラフィックデータの入域地点の速度の最大値と最小値である。評価基準は全航空機の総遅延時間と総コンフリクト数、そして1時間あたりに換算した着陸機数とする。

最適化手法のパラメータを表 6.8 に示す。

表 6.8 GA のパラメータ

GA	Value	分割反復最適化手法	Value
Population Size	100	Num. of Cluster	3
Generation	50	Iteration	10
Mutation Rate	0.8		
Crossover Rate	1.0		

結果

1 5 機における最適化は可能であるため、1 6 機以降のシミュレーション結果を記載する。表 6.10 は試行毎のコンフリクト数の推移を表しており、列ごとに1 試行の結果、行は反復最適化回数を表している。このシナリオは初期の最適化において、コンフリクトが最大で1 3 2 回、最小でも2 0 回程度発生しており、スケジュール遅延よりもコンフリクトの解消をターゲットとすべき交通量となっている。1 0 回目の最後の最適化ではコンフリクト数が一桁となっているが、すべてのコンフリクト状態が解消されたわけではなく、最適化後においても実行可能な着陸順が獲得できていないことを表している。このシナリオにおいては、1 0 試行全てでコンフリクトが解消不可能であるため、手法としても実行可能な着陸順の獲得が困難であり、現実的にも制御が難しいとみなした。

1 9 機のシナリオ以外でも複数のシナリオで実験しているが、1 6 機の場合には1 0 試行中3 試行、1 7 機の場合には1 0 試行中2 試行でコンフリクトが解消されるシナリオが存在し、1 8 機では1 0 試行中1 試行、2 0 機においても1 0 試行中1 試行でコンフリクトが解消される結果を得た。最適化手法としては実行可能な着陸順を得ることは必須な条件であり、このような環境においては、手法において最適化が困難な状況であることが示唆される。

考察

表 6.10 は1 9 機のシナリオにおける航空機の詳細情報を示しており、左の列からコンフリクト数、最適化により獲得した到着予定時刻、初期における到着予想時刻、初期速度、最適化によって得た速度、迂回量をそれぞれ表している。行は航空機の ID に対応する。ID 1 6、ID 1 8 の2 つがコンフリクト状態にあることがわかる。到着予想時刻では2 機の時刻はほとんど同じであるが、最適化後の到着予定時刻には差があり、2 機ともに迂回を利用していることがわかる。このシナリオでは1 5 0 0 秒と2 1 0 0 秒の周辺にお

表 6.9 19 機におけるコンフリクト数の推移

	0	1	2	3	4	5	6	7	8	9
1	82	49	131	20	121	88	60	94	132	47
2	76	33	121	16	54	42	52	44	105	47
3	76	32	97	7	25	17	40	43	31	42
4	57	20	97	7	25	8	14	23	23	15
5	55	18	60	7	25	8	13	23	23	15
6	37	14	39	7	25	7	11	21	20	15
7	37	14	22	7	25	7	11	18	14	15
8	32	14	8	7	25	7	11	14	14	14
9	29	14	3	7	25	4	6	8	13	14
10	22	14	3	7	24	4	6	7	13	14

いて航空機の混雑がある。そのため、コンフリクト状態にある 2 機は当初の予定到着時刻であった 2000 秒周辺での着陸ではなく、それよりも遅い時刻で着陸している。

また、全体の航空機の傾向として、実際に選択した入域速度が初期に設定された入域速度に比べ遅くなっていることがわかる。本シナリオにおいては、19 機中 16 機が速度を下げており、車の渋滞と同じように混雑時には滞留が生じることがわかる。

このシナリオにおける着陸容量は最初と最後に到着した航空機の時刻は 1338 秒と 2761 秒であり、2つの時間差は 1423 秒となる。この時間内に 19 機到着していることから、この時間を 3600 秒（1 時間）に換算すると 48 機が 1 時間当たりの着陸機数となる。

表 6.10 19 機のシナリオにおける航空機の詳細

ID	Conf. num.	Actual Time	Estimate Time	Initial Speed	Actual_Speed	Detour
0	0	2242	1894	444	457	0.891436
1	0	2441	2173	506	466	0.515172
2	0	1536	1469	515	474	0.043074
3	0	1387	1362	475	459	0.000277
4	0	1940	1851	496	462	0.247338
5	0	1783	1419	476	474	0.858057
6	0	1476	1418	499	465	0.199058
7	0	2761	2348	489	402	0.633986
8	0	2013	1979	465	444	0.250617
9	0	2506	2100	502	496	0.970506
10	0	2155	1998	489	497	0.575251
11	0	1683	1675	494	486	0.027919
12	0	1865	1858	461	452	0.250338
13	0	2368	2029	514	481	0.707428
14	0	1338	1357	453	459	0.12534
15	0	1614	1590	486	470	0.139546
16	3	2303	2020	513	511	0.782941
17	0	2079	1938	492	421	0.015197
18	3	2668	2079	520	421	0.842581

表 6.11 は航空機を 16 機から 19 機までのシナリオでの最良試行における交通流の詳細を示している。左の列から、最適化対象の航空機数、最適化によって得られた 1 時間当たりの着陸容量、全航空機の初期進入速度と最適化後の入域速度差の平均 (Nt)、最適化対象の航空機における初期の入域速度と最適化後の入域速度差の最大値、スケジュール対象の航空機の到着予測時刻からの平均遅延時間と最大遅延時間 (秒)、迂回量の平均と最大をそれぞれ表している。なお、迂回量は最小で 0、最大で 1 の値を取る。4 つのシナリオ全てにおいて、10 試行中いくつかの試行で実行不可能解を生成するため、過密状態である。この過密状態では航空機の入域速度を下げ、迂回経路を積極的に利用していることがわかる。平均の遅延時間に着目すると、当初の到着予想時刻に比べ、大きく遅れていることがわかる。全航空機の平均遅延時間はおおよそ 3 分程度となっているが、18 機、19 機の最大遅延時間に着目すると 10 分程度の遅延が発生している。このような過密状態においては、コンフリクトの解消が主眼が置かれるため、遅延時間を抑えることが困難であることが考えられる。また、着陸容量が大きいほど、滞留が発生しやすい状況になっており、平均速度差もより負の値に大きくなる傾向がある。

次に比較的交通量に余裕のある航空機が 10 機の場合と比較すると、着陸容量に関しては 16 機以上の着陸容量と比較しても大きく変化はないが、他の項目には大きな変化が見

られる。例えば、平均速度が大きく上昇し正の値を取り、平均迂回量も0.06と他の航空機数とくらべ大きく減少していることから、迂回せず、速い速度で巡航するため、多くの航空機が予想到着時刻通りに航行が行えたことがわかる。このことから交通流が限界に近い状態においては速度が減少し、多くの航空機が迂回経路を利用することがわかる。また、着陸容量の限界は48機から49機であると推察される。

表 6.11 各シナリオにおけるデータ

航空機数	着陸容量	平均速度差	最大速度差	平均遅延時間	最大遅延時間	平均迂回量	最大迂回量
10	41.4	12.8	-28	-39.5	23	0.06	0.22
16	44.2	-14.3	-86	108	510	0.26	0.97
17	43.6	-8.1	-45	155	433	0.43	0.95
18	42.3	-16.8	-91	176	691	0.39	0.87
19	48.1	-25.9	-99	190	589	0.43	0.97

6.4 クラスタ数自動決定法

6.4.1 スケジュール最適化の並列化に基づくクラスタ数の自動決定

クラスタリングを適用する際には、クラスタ数を予め与えなければならない。これを自動的に決定するために、異なるクラスタ数を定めた分割反復最適化手法を並列に実行し、個々の最適化性能によりクラスタ数の評価を行い、評価に従い、クラスタ数の変更を行う。

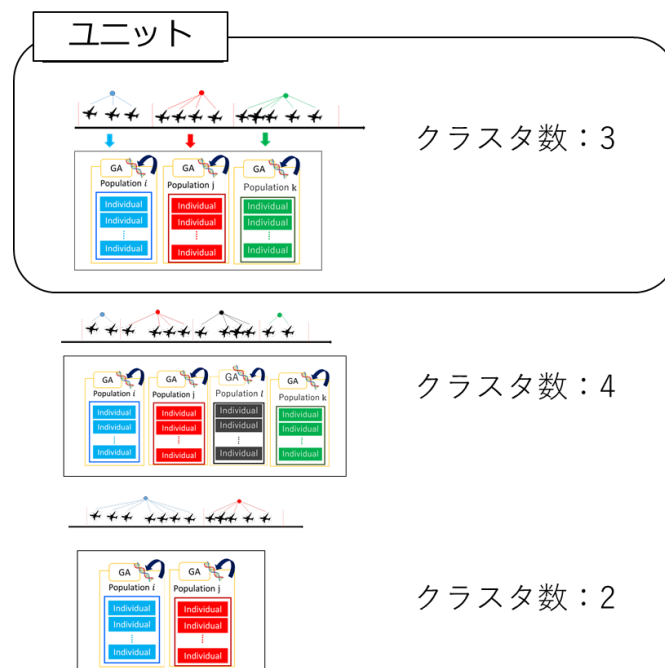


図 6.39 クラスタリング数決定のためのアプローチ

図 6.39 はクラスタ数の自動決定のアプローチを示しており、図のように複数の分割反復最適化を行うユニットで構成され、このユニット一つ一つ個別にクラスタ数が付与される。このクラスタリング部、最適化部、クラスタ数の構成が一つのユニットとして定義する。付与されたクラスタははじめ、ランダムに与えられ、このクラスタ数に基づいて、分割反復最適化を行う。分割反復最適化を行う回数としては、規定数の最適化における GA の世代数とクラスタの反復回数を実行する。

本手法におけるクラスタ数の決定は複数の最適化の処理を並列的に行うことで適応的にクラスタ数を決定するため、6.3 節で議論したクラスタ数のパラメータを考慮しなくてよい。ただし、並列化のためにどのようにクラスタ数を決定し、最適化の性能を向上するかが重要となる。そこで、クラスタ数の決定は、反復して最適化を行う特性を生かした方法で行う。次にこの方法論の詳細を説明する。

6.4.2 評価に基づくクラスタ数の決定法

個々のユニットは並列に独立して最適化を実行するが、最適化後に個々のユニットは評価値を得る。それぞれのユニットは子の評価値に基づき、クラスタ数を変更するか否かを決定する。ここではこのクラスタ数の変更方法について説明する。

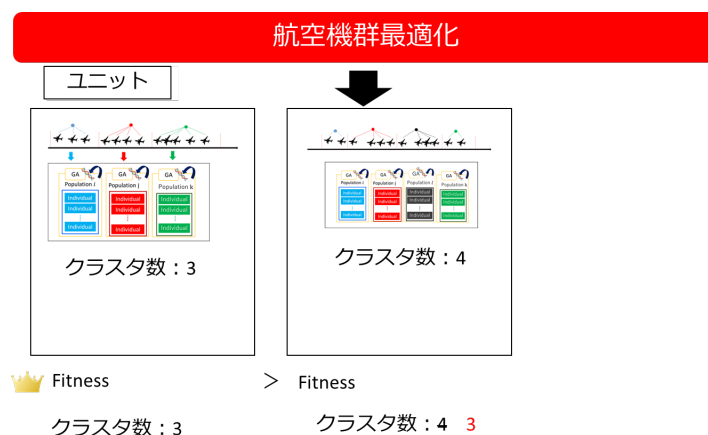


図 6.40 クラスタ数の決定方法

図 6.40 は航空機群最適化を行った後のユニット同士の評価値の比較とクラスタ数の変更例を表している。Fitness は最適化で得られる評価値を表している。クラスタ数の変更の際には複数のユニットの中から二つのユニットがランダムに選択される。選択されたユニットはそれぞれの評価値である Fitness を比較する。この評価が高いユニットはクラスタ数を変更せず、次の最適化が実行される際には同じクラスタ数を利用する。一方で、評価値の低いユニットのクラスタ数は評価値の高いほうのクラスタ数の値を引き継ぎ、そのクラスタ数に従い、クラスタリングを行い、最適化を実行する。このように高い評価値

を得たクラスタ数を利用することで、最適化の効率を図ること可能な一方で、得られた評価値が偶然（乱数に依存する要因）獲得されたものである可能性があることや、初期に選択されたクラスタ数に大きな影響を受ける。ユニットの初期化の際にはクラスタ数がランダムによって選択されるため、選択されなかったクラスタ数はその数でクラスタリングが行われない可能性が高い。

この問題を解決するために、GA における突然変異のフレームワークを活用する。突然変異は集団が局所解に陥ることを防ぐためのオペレーションであり、GA においては遺伝子内の値を乱数で変化を与える。本手法でも同様に同じクラスタ数を持つユニットにおける局所解での停滞を防ぐ目的がある。詳細の方法を以下に示す。

突然変異オペレータの対象としては、ユニット内でクラスタ数を比較される際に選択されたユニットとする。選択された二つの中で低い評価値を持つユニットは一定確率 r_m で突然変異オペレータを作用させる。突然変異させる場合には、クラスタ数を乱数により選択する。

6.4.3 実験

実験内容

ユニットによるアプローチによるクラスタリング数の自動決定の有効性を検証するため、シャルル・ド・ゴール空港における固定経路の実験を行った。実験におけるユニット数は4とし、最適化の反復数は20とした。突然変異によるクラスタ数の変更確率は0.4とした。GA のパラメータは先の実験と同様である。

6.4.4 結果

図 6.4.4 は Fitness の推移を表しており、横軸はユニットの比較回数、縦軸は Fitness を表す。Fitness は第6章と同様に初期速度、遅延量0（迂回経路を使用しない）と比較した際の遅れを表す。赤、青はそれぞれ単数、複数ユニットの Fitness の推移である。単数における結果はクラスタリング数を常に5にした場合であり、複数の場合は4つのユニットを用いてクラスタリング数を変化させた結果である。Fitness の推移をみると初期推移は単数における性能のほうが Fitness は低い値を取っているが、その後、最終的には同じような Fitness の評価を得ていることがわかる。

6.4.5 考察

図 6.42 は初期世代と最終世代において選択されたクラスタ数のヒストグラムを表しており、最良評価値を30シードを集計した結果である。初期世代における最良解におけるクラスタ数は分割数2から分割数6までまでカウントされており、最終世代のヒストグラ

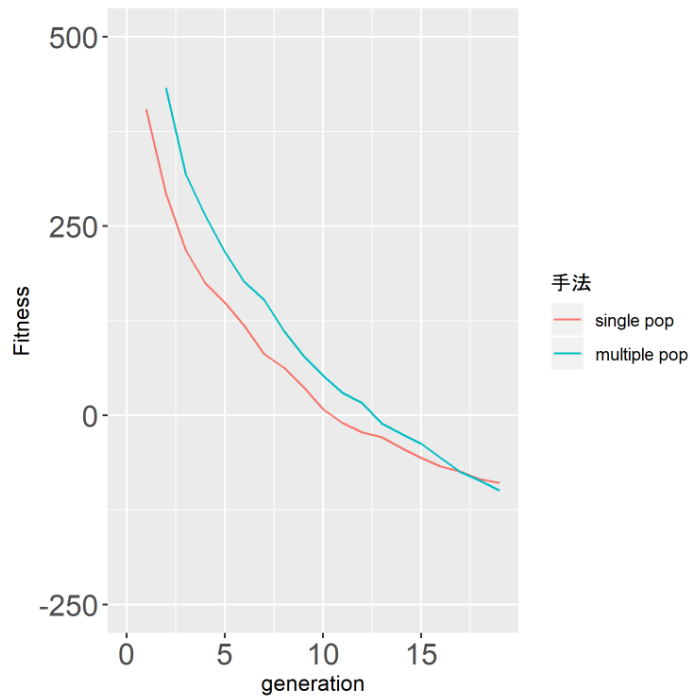


図 6.41 単数ユニットと複数ユニットでの Fitness

ムとは異なり，幅広くクラスタ数が分散されていることがわかる．ヒストグラムの最頻値は分割数 3 と分割数 6 の 9 で，初期世代ではコンフリクト解消のため，ある程度群としてのまとまりが必要になる．それ故クラスタ数の値の小さいものが選択され，最良解を獲得したと考えられる．最終世代の結果としては，クラスタリングにおける分割数は 4 から 6 までに偏っており，特に分割数 6 が 30 シード中 17 シードとなっている．性能を上げるためにはある程度分割数を大きくすることが重要であることは 6.6.1 で示唆したが，これを証明する結果となっていることがわかる．

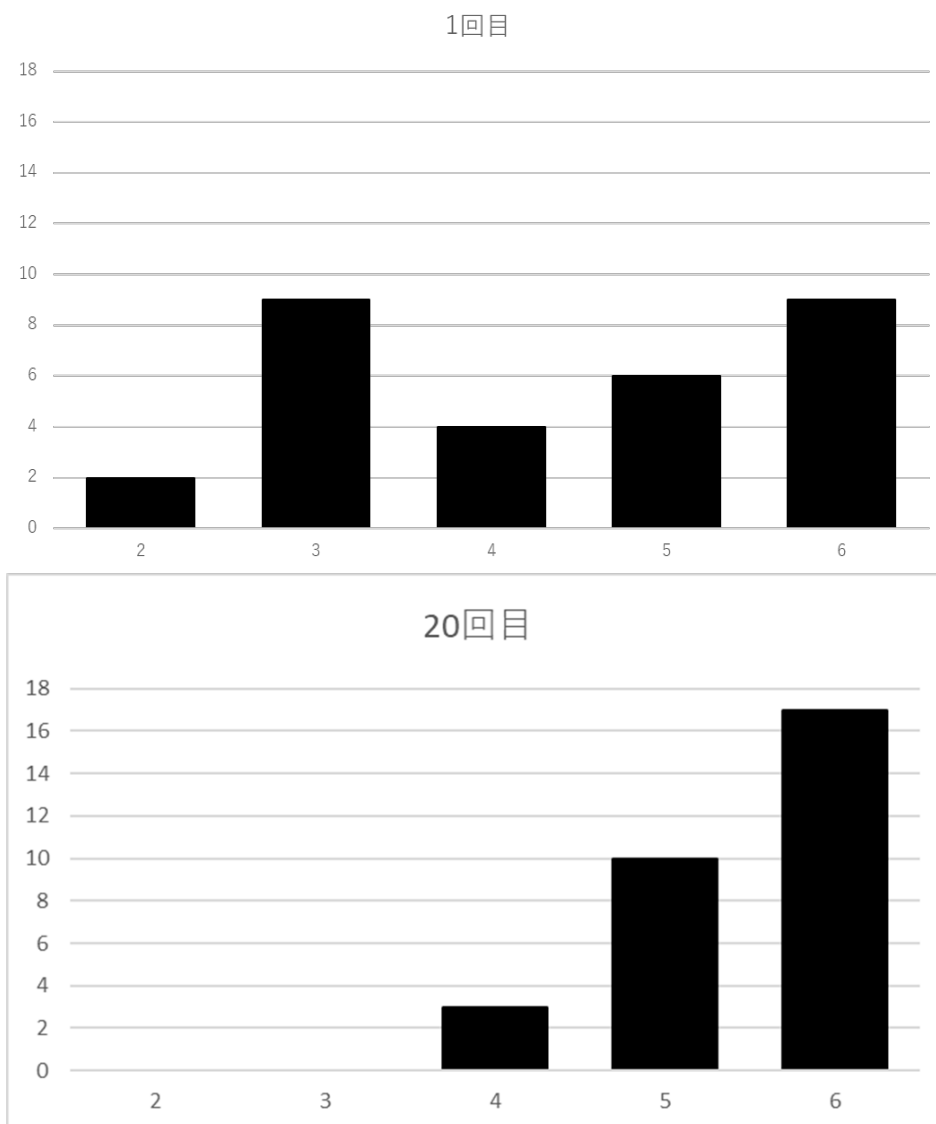


図 6.42 複数ユニットにおける最終的に選択されたクラス数

第 7 章

自由経路・固定経路と管制運用の親和性

7.1 自由経路における個体から全体の設計論的観点

本研究では、航空管制官が可能な経路着陸順最適化手法を提案したが、これは最適化手法の運用が管制官が業務を遂行する状況と同じ環境を想定する。航空機が航行する経路に関しては Waypoint と呼ばれる通過点に基づいて生成されるが、空港周辺の標準到着経路 (STAR) や管制官が混雑時用いるベクタリングは Waypoint ベースで、機首方向をある通過点に向けて指示を行うため、経路の生成法は親和性の高いものである。

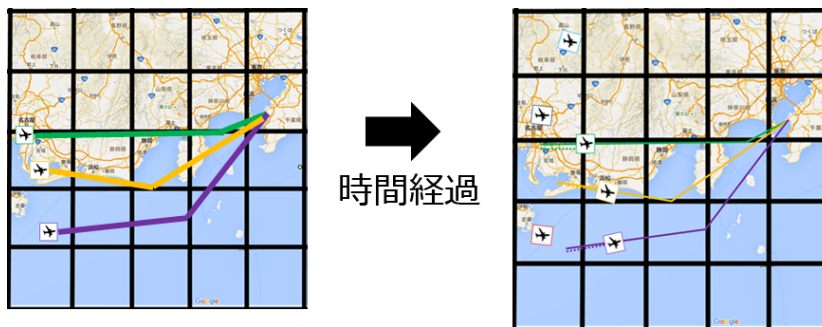


図 7.1 自由経路における連続最適化環境

図 7.1 は複数の航空機が入域した場合における決定した経路とその後新たに入域した航空機の例を示している。管制官が監視可能なレーダ範囲内に入域した航空機の安全管理を行うため、レーダ画面外の状態を基本的には把握することが出来ない。左側図ではじめに入域していた 3 機の航空機に関しては決定した経路に沿って航行するが、新たな航空機が入域した際、以前決定した経路を大幅に変更し新たな経路を決定することは管制官の混乱を誘引する。管制官は FCFS の原則で航空機を管理するため、先に入域した航空機を

先に着陸へと誘導する。この観点から、一度決定した経路を変更することは好ましくないが、過密状態においては、着陸間隔は制約を満たす上で出来るだけ短くすることが好ましいため、本研究では迂回経路より、経路の微小な変化を許容した。

自由経路の提案手法は各航空機が独立に経路候補を複数生成し、この経路候補の組み合わせから着陸順を生成する個体が主体となる分散的な方法論である。分散的アプローチの具体的な利点としては、2つあり、1つ目は悪天候における環境変化に伴う経路変更への対応と、2つ目は航空機の混雑時における柔軟な対応である。悪天候における環境変化への対応は、目的空港へ航空機が航行する際に、悪天候により経路変更が余儀なくされた場合に、航空機全体を勘案して各航空機の経路を生成する場合には全体としての経路の再設計が必要となり、多くの計算コストを伴う。本提案手法は経路生成においては各航空機が独立に生成しているため、悪天候空域を通過する航空機を対象に経路を再生成すれば良く、最小限の変更で対応することが可能である。環境変化における個体から全体での設計論に対する手法の有効性は村田らが羽田空港における悪天候が発生した場合を例題として手法の有効性を検証した [109]。

2つ目の混雑時における対応だが、本論文においても自由経路手法の連続的な最適化の有効性を検証する際、迂回経路の利用により多くの着陸の誘導が可能であることを明らかにした。混雑時においては多くの航空機に対する指示が必要であるため、管制官にとって多くの指示が必要でありワークロードが大きくなる。そのため、本提案手法はこのような混雑環境下において、特定の航空機に対して管制官が直接管理し、それ以外の航空機に関してはシステムが管理するようなフレームワークを利用することが可能である。このように航空管制官のみで交通流の管理が困難な状況でシステムの支援が必要な状況においては、分業が可能なシステムは人とシステムの親和性が高いと言える。

7.2 実問題における GA のパラメータ設定指針

本研究で、使用したパラメータは自由経路においては、個体と全体の GA のパラメータは、個体である航空機の経路生成にかかる GA の Population size と、世代数はそれぞれ、50 と 500 であり、全体である着陸順生成にかかる経路の組み合わせを探索するための GA の Population size と世代数は、100 と 1000 である。固定経路においては、Population size は 100、世代数は 250 である。3章の表でまとめたように、本研究に関わる従来研究で最適化手法として GA を使用する手法の Population と世代数はおよそ、50 から 200,200 から 1000 となっているため、範囲内であるといえる。関連研究はそれぞれ、実際の運用に資するようなパラメータ設定であるため、本研究における Population size と世代数は妥当であると言える。ここで、Population size と世代数に着目した理由としては、この二つのパラメータが最適化の実行処理に関わる項目だからである。Population size が多くなるほど、評価回数が多くなり、世代数も同様に増加すると処理時間が大きく

なる。ただし、一概に Population size と世代数が関連研究と同程度の評価回数であれば良いとことを述べるだけでは不十分であり、評価値を獲得するまでの時間や、固定経路においては、クラスタリングや GA の初期化を複数回行うことからこのような処理時間も考慮に入れなければならない。

固定経路における実行処理時間について考察する。本論文における実行時間計算のための設定は、計算機スペックは、CPU Core i7-4790 3.60Ghz, メモリ 16GB の計算機をした。実験における航空機数は 15 機とし、最適化に係るパラメータは分割反復における分割数(クラスタ数)を 4, 反復回数を 5 回, 世代数を 50 世代とした。この時の実行時間は 201 秒となった。ここで、最適化にかかる並列化において議論する。本研究で使用する最適化アルゴリズムにおいて並列化可能である部分主に 2 つあり、GA の個体を評価する際には処理の並列化を行うことが可能であり、スケジュールを分割した際の奇数, 偶数群での奇数群, 偶数群の同群内に関して並列処理可能である。クラスタを 4 つとすると、偶数群と奇数群のクラスタがそれぞれ、2 つずつ所属することになる。GA において実時間で解を得るための設計の方針としては、実時間内で解を得る場合には問題固有の知識を最適化手法に入れ込むことが重要である。本研究においても、知識を用いて解探索を効率化している。例えば、航空機同士が干渉(コンフリクト)し、着陸順として実行不可能解となる場合、コンフリクトに直接的な影響を与える航空機を修正することで最適化実行処理を始めてから早期に実行可能の獲得を図ることで、7 章で述べたように、実時間内の実行可能な着陸順の確保と最適化性能の向上が可能である。

また、航空機数が多くなるほど探索空間が大きくなるため、実行時間も増加する恐れがある。そのため、本研究ではスケジューリング対象の航空機を複数の群に分割することで解の探索範囲を限定するアプローチを採用した。これは遺伝的アルゴリズムの遺伝子構造の観点で言えば、遺伝子を複数に切断し、その切断した遺伝子内で進化を実行することに該当する。例えば、車の設計に関する最適化を行う際に、決定変数にはシャーシやエンジン、タイヤなどの車のパーツが格納されること想定すると、分割をした場合には全体としての評価を獲得することは困難であり、全体としての評価が必要となる。一方で、本研究では最適化後のスケジュールあるいは評価を全群の総和から見積もることができるため、分割というフレームワークが適用可能である。このように問題の前提をうまく活用することは実問題の最適化においては非常に重要であることが示唆される。

第 8 章

おわりに

8.1 まとめ

本研究では空港周辺の超過密状況を想定した，航空機同士のコンフリクト（取り決められた間隔未満の状態）の解消と航空機遅延を最小限にすることを目的とし，航空管制官が運用可能なシステム向け，混雑度合いの違いによる自由・固定経路の異なる二つのアプローチにより目的の達成を目指した．

自由経路の場合には進化計算に基づく経路・着陸順同時最適化手法を提案しその有効性の検証を行った．具体的には，最終的な着陸の際に適切な管制間隔が維持されるよう，複数の航空機をの経路独立に進化させ，多様な経路をの組み合わせから構成される解候補の組み合わせを探索する多段型分散進化により，着陸順を最適化を実現した．この手法は多目的最適化進化計算手法の一つである二目的の NSGA-II をベースとしており，その評価指標として「最適性」を評価する解の良さと，「多様性」を評価するノベルティサーチを導入した後 (1) 評価値の高い解の近傍を重点的に探索する重点サンプリングと (2) 評価値の高い解から離れる範囲を探索しない多様性制限を組み込むことで最適性と多様性のトレードオフの問題を改善した．

自由経路の提案手法である最適性と多様性のトレードオフを考慮した経路・着陸順最適化手法の有効性を検証するため，羽田空港を模擬したシミュレーションマップにおいて，実際のトラフィック量を想定した航空機数における経路・着陸順の最適化の性能検証を行った．シミュレーションは 5 分に 1 回の最適化を想定し，3 機がレーダ画面に入域することを想定する．実験は大まかに二つの分かれており，一つ目は 3 機のみでの経路・着陸順の最適化．二つ目は混雑時における交通流を想定し，10 回の最適化で連続的な最適化の可用性を検証した．

はじめに事前実験として，航空機の有用な経路候補を生成するための重点サンプリングのパラメータである W と多様性制限のパラメータである β の効果を確認した．重点サンプリングの値に関しては指数のオーダーで評価値に重み付けされるため，1 から 5 程度ま

での値は着陸順の最適性の向上は見られなかったが、10以上になると性能向上がみられた。次に多様性制限は大きい値を持つほど長い経路長を許容するが、この値が10程度の大きな値だと着陸順の効果が見られない。これが、2程度だと最大の効果が出ることが見られた。これは、経路長が最大でも2倍程度で、重みも10倍程度の場合に着陸順として効果的な経路候補を持つことが言える。

1つ目の実験においては、経路生成における評価値の算出を最適性、多様性それぞれの単目的、最適性と多様性の2目的により生成された経路の候補から生成した着陸順を比較した。実験はそれぞれ異なる初期位置を持つ航空機で25試行実験を行ったところ以下の結果が得られた。まず、最適性のみの評価により生成した経路は30試行中で5試行が着陸間隔を満たせず、実行不可能解となった。多様性のみの経路評価に基づく着陸順最適化と最適性と多様性の2目的に基づき生成した着陸順は30試行全てにおいて実行可能な着陸順が獲得可能であることを明らかにした。更に、2目的により経路生成した着陸順は25試行で他の2手法に比べて25試行で高い性能の着陸順を獲得した。

2つ目の実験において、入域する航空機を1回につき3機から7機までの範囲で同時に入域し、10回連続で同数の航空機が入域する過密状況において、提案手法の適用範囲の限界を検証した。結果として、6機同時に入域した際には10回の最適化でも実行可能な着陸順を生成することができたが、7機の場合には5回目以降で実行不可能解な着陸順となった。

固定経路の場合には増加する航空需要に対応する為、航空管制システム高度化され、進化計算や動的計画法を用いた手法により着陸順最適化が実現されているが、これらの手法の問題点として航空機数が増えれば増えるほど最適化を行う時間が膨大になり、最適性能が劣化するという問題が生じる。この問題に対応するため、本研究では着陸問題における航空機のスケジュールにおける反復分割並列最適化手法を提案した。具体的には、クラスタリング手法の一つである、k-means法を用いて全体のスケジュールを複数に分解し最適化し、この手順を複数回繰り返すことでスケジュールリングの探索範囲を狭め、優良なスケジュールリングを探索しつつ、複数回局所解にならないよう、複数分割を行うことで解決を図った。提案手法の有効性を検証するために、フランスのシャルル・ド・ゴール空港におけるトラフィックデータに基づく着陸シミュレーション実験を行った。このシミュレーション実験において以下の知見が得られた。(1)分割による最適化により、従来の分割手法より最適性の高いスケジュールリングを獲得可能であること。(2)分割を複数回繰り返すことでよりよい群の組み合わせを探索出来る可能性があり、分割を複数回行うことで複数回行うことによる最適性の向上が明らかになった。

次に、過密状態においてはコンフリクトのない状態の着陸順を生成することが難しい状況に対し、進化計算は実行不可能解の割合が増えるほど解探索の停滞を誘発し、最適化性能の低下を招く。この問題に対し、実行不可能解内の遺伝子の解修復に着目し、各遺伝子

毎の制約違反量を算出し、最大・平均制約違反量に基づく解修復法を提案し、手法の有効性の検証を行った。具体的には、評価を行う着陸順の解の航空機の間隔が一番小さい間隔を最大違反量、航空機他の航空機とどれだけコンフリクト状態なのかを表す指標を平均違反とし、この二つの違反量を遺伝子内で順位付けし、それぞれの順位の和により順位付けを行う。相対ランキングによって遺伝子内の解修復対象を定め、修復を施すことで実行不可能解に影響する遺伝子を直接的に見つけ出すことが出来る。シミュレーションにおいて、(1) 最大制約違反量に基づく手法は Population を実行可能解へと導く効果は高いが、最適化性能は平均制約違反量に基づく手法よりも低い。(2) 平均制約違反量に基づく手法は高い最適化性能を導くが、最大制約違反に基づく手法より実行可能解へ導くまで時間がかかる。(3) 最大・平均違反量を考慮した手法は実行可能解に到達する世代数及び最適化性能を示す Fitness の値どちらも一方のみを考慮した手法よりも良好な結果を示すことが明らかになった。

8.2 課題

8.2.1 最適化手法における課題

不確かさ

実際の航空機の運用は風の影響や、巡航する経路の違いなどから実際予定していた入域時刻に差異が生じる場合がある。特に最適化を実行する際には実行処理時間が必要であり、これが、次回実際の経路着陸順に影響を与える可能性がある。本論で計算した最適化の実行処理時間は 201 秒であり、この値は最適化の時間としては十分ではあるが、最適化を実行する前の状況と後の状況に変化が生じている可能性がある。このような変化を最小限にとどめるためには最適化の時間を短くし、航空機の情報に常時更新できる状況が理想的である。本研究でのシミュレーションでは、風などで発生する不確かさのについては考慮しなかったため、これに対する対応が必要がある。

最適化に係るパラメータ

本論文で使用したパラメータはシミュレーション時において実験中に有効なパラメータを発見し利用したが、これらの値に再考の余地がある。例えば自由経路において、生成された経路の候補数を本研究においては 50 個の経路を着陸順生成のための候補としたが、この候補数に関しては重点サンプリングと多様性制限のパラメータである W と $beta$ は着陸順の最適化性能に大きな影響を与える。固定経路においては分割反復最適化手法における最適化の反復回数や全航空機に対する分割数は最適化対象の航空機数に依存するため、この関係性を明らかにする必要がある。また、1 回の最適化における GA の世代数や交叉方法や突然変異確率も再考する必要がある。

8.2.2 ALP における課題

統合的な飛行場の管理

空港周辺におけるオペレーションは到着機のコンフリクトや着陸順に関するマネジメントのみならず、出発機の管理、飛行場面の管理を行わなければならない。出発機に関する最適化も多くの研究者により取り組まれており、最適化手法としては、動的計画法や GA、Branch and Bound などの手法が提案されている [110–115]。また、到着機、出発機、飛行場面における統合的な手法も提案されており、航空機の過密状況における手法の有効性を比較する必要がある。 [116–118]

経路形状と着陸順との関係

固定経路に基づく最適化において、固定経路においては実際に運用で定義された経路を適用しているが、付加した迂回経路に関しては本論文で定義した経路を適用している。本論文で利用した迂回経路は台形の形状をしており、迂回量もそれほど大きくしなかったが、迂回経路を付加する位置、台形の長さは着陸順に大きな影響を与えるため、迂回経路の設定事項は今後の課題である。また、形状も台形以外の形状も考えられ、例えば Point Merge と呼ばれる扇状の形状は羽田空港で最近運用を始めた。このように形状も着陸順の最適化に大きな影響を与えるため、空港の地理的条件に見合う形状の条件を見出すことが重要である。

8.2.3 システムと航空管制官との親和性における課題

多目的意思決定

航空機の運航には多数のステークホルダーが関わるため、航空管制官ためだけではなく、エアラインや、気象庁、訓練空域を管理する防衛省等の協調的な意思決定が必要である。意思決定を行うためには、多様な要求を満たす意思決定を行うための候補が必要となる。例えば、空港においてはできる限り多くの航空機を滑走路に着陸させることをエアラインは望むが、このような過密状況においては、管制官のワークロードが増加するため、これを低減させるよう、着陸する時間を遅延させることを望む状況が考えられ、このような解集団候補は多目的のトレードオフとなる。本論文では多目的最適化に関しては取り扱っていないため、意思決定を行うための多出発機最適化への拡張を行う必要がある。

複数滑走路・出発機

羽田空港のような混雑空港においては、複数の滑走路を同時に運用することが可能であり、本研究では出発機と到着機を分離併用を仮定しているが、実際の運用においては、二

つの滑走路を出発機と到着機両方を運用すると更に効率的に滑走路利用することが可能となる [119]. Bianco et al. らは着陸順と出発順の混合運用において、ジョブショップ・スケジューリング問題にモデル化を行い、ヒューリスティックなルールに基づく手法を提案し、ローマ、ミランの空港における実際の航空交通流を想定した例題において検証し、FCFS に比べ 40% 程度の性能向上を実現し、手法の有効性を明らかにしている。また、Liang らは北京国際空港において到着機の合流及び到着機と出発機のコンフリクトを解消し、離発着数の最大化を目的に、SA において手法の有効性を検証し、通常よりも 26% の性能向上が明らかになった [84]。その際に、出発機に対する、決定変数と到着機に対する制御変数が異なるため、二つの探索に偏りが発生する恐れがあるため、二つの探索バランスを調整する必要がある。

提示した着陸順の解釈性

厳密解法、メタヒューリスティックスという分類に関わらず、最適化手法において出力された解（本研究においては着陸順）はある評価指標に基づき生成されているが、実際に航空管制官によって導き出された着陸順及び経路は異なる場合が多くある。管制官は業務処理規定と担当空域の研修を得て、実際の業務に当たるため、ある程度の基準や規則に従い指示を出す。経験値や経験則に従って航空機に指示を送るため、それぞれの癖や基準を持つ。これらのルール生成や言語化は難しく、エキスパートシステムによるルール化においても矛盾を含んだルールが生成されるなど人が理解しうる着陸順や交通流を生成するには課題がある。

謝辞

本研究をまとめるにあたり，多大なるご指導とご教示をいただいた主任教員である高玉圭樹教授，指導教員である庄野逸教授，佐藤寛之准教授に心より感謝の意を表します．特に高玉圭樹教授には研究指導以外にも研究者としての姿勢やあるべき姿をご自身の経験を語ってくださり，私が今後，研究者として歩むための指針を示していただきました．

また，博士論文審査をしていただいた大須賀昭彦教授，柏原昭博教授に感謝いたします．予備審査において多くの御指摘や意見をいただき，本論文の意義や貢献がより明確になりました．重ねて感謝いたします．

特別研究員制度によって研究費の面で助成いただいた日本学術振興会様，日々の学生生活の支援をしてくれた家族には深く感謝申し上げます．本研究を遂行できたのは，特別研究員制度による研究費の援助と生活費の支援があるからこそだと感じております．

修士2年以降，継続的に専門的なご助言をしていただいた，電子航法研究所の研究員のみなさまにも感謝申し上げます．当時，インターンシップ制度により電子航法研の航空交通管理領域での受け入れを承諾し，研究テーマを決定するために多大なご助言をいただいた福島副領域長，研究におけるシミュレーション面での御意見や進捗に関するアドバイスをいただいた瀬之口主任研究員に初めに感謝申し上げます．インターンシップ後，高玉研究室との共同研究契約の締結にご尽力いただき，定期的な研究に関するディスカッションをしていただいたブラウン上席研究員，航空管制に関する専門知識の観点から航空管制官との親和性に関する多大なアドバイスをいただいた平林主幹研究員に重ねて感謝申し上げます．

また，虎谷研究員にはインターンシップ時から，研究に関するアドバイスをはじめ，留学先の Daniel 先生を紹介していただき，私の研究者としての礎を築いていただいといっても過言ではありません．深く感謝申し上げます．

I would like to express my gratitude Prof. Daniel Delahaye for the continuous discussion and support during my ph.D study.

Then, I would like to thanks the following ENAC stuff and students: Ji Ma, Tambet Treimuth, Huo Ying, Gilles, Serge for technical and for all the funs we had in the lanch and coffee breaks.

参考文献

- [1] ICAO annual report 2017. <https://www.icao.int/annual-report-2017/Documents/>, 最終閲覧日:2019年9月24日.
- [2] 屋井鉄雄, 平田輝満, 山田直樹. 飛行場管制からみた空港容量拡大方法に関する研究. 土木学会論文集 D, Vol. 64, No. 1, pp. 122–133, 2008.
- [3] SESAR. <https://www.sesarju.eu/>, 最終閲覧日:2019年9月24日.
- [4] NextGen. <https://www.faa.gov/nextgen/>, 最終閲覧日:2019年9月24日.
- [5] Collaborative actions for renovation of air traffic systems (CARATS). <http://www.mlit.go.jp/common/000123890.pdf>, 最終閲覧日:2019年9月24日.
- [6] Julia A. Bennell, Mohammad Mesgarpour, and Chris N. Potts. Airport runway scheduling. *JOR*, Vol. 9, No. 2, pp. 115–138, 2011.
- [7] Man Liang, Daniel Delahaye, and Pierre Maréchal. Integrated sequencing and merging aircraft to parallel runways with automated conflict resolution and advanced avionics capabilities. *Transportation research. Part C, Emerging technologies*, Vol. 85, pp. 268–291, 2017.
- [8] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, Vol. 6, No. 2, pp. 182–197, 2002.
- [9] John A. Hartigan and Manchek A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 28, No. 1, pp. 100–108, 1979.
- [10] Daniel Delahaye and Stéphane Puechmorel. 3D airspace design by evolutionary computation. In *Proceedings of 2008 IEEE/AIAA 27th Digital Avionics Systems Conference*. IEEE, 2008.
- [11] Rafal Kicingier and Arash Yousefi. Heuristic method for 3d airspace partitioning: genetic algorithm and agent-based approach. In *Proceedings of 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS)*, 2009.

- [12] Shannon Zelinski. A comparison of algorithm generated sectorizations. *Air Traffic Control Quarterly*, Vol. 18, No. 3, pp. 279–301, 2010.
- [13] ATFM. https://www.mlit.go.jp/koku/15_bf_000346.html, 最終閲覧日:2019年9月24日.
- [14] Maxime Gariel and Eric Feron. 3D conflict avoidance under uncertainties. In *Proceedings 2009 IEEE/AIAA 28th Digital Avionics Systems Conference*. IEEE, 2009.
- [15] Nicolas Durand, Jean-Marc Alliot, and Olivier Chansou. Optimal resolution of en route conflicts. In *Proceedings of 1st USA/Europe Seminar*, 1997.
- [16] Nicolas Durand and Nicolas Barnier. Does atm need centralized coordination? autonomous conflict resolution analysis in a constrained speed environment. *Air Traffic Control Quarterly*, Vol. 23, No. 4, pp. 325–346, 2015.
- [17] Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. pp. 3–19. Springer, 2011.
- [18] Géraud Granger, Nicolas Durand, Jean-Marc Alliot, et al. Token allocation strategy for free-flight conflict solving. In *Innovative Applications of Artificial Intelligence Conference (IAAI)*, pp. 59–64, 2001.
- [19] ブラウンマーク長岡栄. 航空交通管理における航空交通の複雑性指標について. *日本航海学会誌*, Vol. 204, pp. 37–45, 2018.
- [20] Banavar Sridhar, Kapil S Sheth, and Shon Grabbe. Airspace complexity and its application in air traffic management. In *Proceedings of 2nd USA/Europe Air Traffic Management R&D Seminar*, pp. 1–6, 1998.
- [21] Daniel Delahaye and Stéphane Puechmorel. Air traffic complexity: towards intrinsic metrics. In *Proceedings of 3rd USA/Europe Air Traffic Management R & D Seminar*, 2000.
- [22] 狩川大輔, 青山久枝. 航空路管制業務における到着間隔設定プロセス可視化手法の機能評価. *ヒューマンインタフェース学会論文誌*, Vol. 18, No. 3, pp. 275–288, 2016.
- [23] Daisuke Karikawa and Hisae Aoyama. Analysis of controllers’ working methods supporting safe and efficient air traffic operations. *International Federation of Automatic Control (IFAC)-PapersOnLine*, Vol. 49, No. 19, pp. 319–324, 2016.
- [24] Esa M. Rantanen and Ashley Nunes. Hierarchical conflict detection in air traffic control. *The International Journal of Aviation Psychology*, Vol. 15, No. 4, pp. 339–362, 2005.
- [25] Imen Châari, Anis Koubaa, Hachemi Bennaceur, Sahar Trigui, and Khaled Al-Shalfan. smartpath: A hybrid aco-ga algorithm for robot path planning. In

- Proceedings of 2012 IEEE Congress on Evolutionary Computation (CEC2012)*, pp. 1–8. IEEE, 2012.
- [26] Seiji Sugiyama and Sadao Akishita. Optimal horizontal guidance for vehicle traffic control. *Transactions of the Society of Instrument and Control Engineers*, Vol. 35, pp. 1207–1212, 1999.
- [27] Steven M. LaValle. Rapidly-exploring random trees: A new tool for path planning. pp. 8–11, 1998.
- [28] 杉原有理花, 高橋正樹. ベクタリングによる管制間隔の調整と騒音の広域分担を目的とした着陸航空機の軌道計画. *日本航空宇宙学会論文集*, Vol. 66, No. 2, pp. 53–60, 2018.
- [29] 齊藤有紀, 吉村翔, 井上正樹, 高橋正樹, 杉原有理花. 騒音の広域分担を目的とした航空機の最適軌道計画. *システム制御情報学会 第 61 回自動制御連合講演会*, pp. 1724–1728. 自動制御連合, 2018.
- [30] Toshiyuki Tajima, Kan Nakano, Masaya Ichikawa, and Hiroshi Maeda. A real-time path planning using genetic algorithms. *Journal of Japanese Society for Artificial Intelligence*, Vol. 10, No. 1, pp. 94–104, jan 1995.
- [31] 田中雅晴, 溝口正信, 丹下利明, 高見勲, 鈴木真二. 遺伝的アルゴリズムを用いた複数飛行経路の生成法. *電子情報通信学会論文誌 D*, Vol. 85, No. 8, pp. 767–775, 2002.
- [32] Debadeepta Dey, Andrey Kolobov, Rich Caruana, Ece Kamar, Eric Horvitz, and Ashish Kapoor. Gauss meets canadian traveler: shortest-path problems with correlated natural dynamics. In *Proceedings of 2014 international conference on Autonomous agents and multi-agent systems(AAMAS2014)*, pp. 1101–1108. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
- [33] Juanping Zhao and Xiuhui Fu. Improved ant colony optimization algorithm and its application on path planning of mobile robot. *Journal of Computers*, Vol. 7, No. 8, pp. 2055–2062, 2012.
- [34] Soonkyum Kim, Subhrajit Bhattacharya, and Vijay Kumar. Path planning for a tethered mobile robot. In *Proceeding of 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1132–1139. IEEE, 2014.
- [35] Jan-Peter Calliess, Michael Osborne, and Stephen J Roberts. Conservative collision prediction and avoidance for stochastic trajectories in continuous time and space. In *Proceedings of 2014 international conference on Autonomous agents and multi-agent systems*, pp. 1109–1116, 2014.
- [36] Nourelhouda Dougui, Daniel Delahaye, Stéphane Puechmorel, and Marcel Mongeau. A light-propagation model for aircraft trajectory planning. *Journal of*

- Global Optimization*, Vol. 56, No. 3, pp. 873–895, 2013.
- [37] Yoshikazu Miyazawa, Navinda Kithmal Wicramasinghe, Akinori Harada, and Yuto MIYAMOTO. Passenger aircraft optimal trajectory with an arrival time constraint. *Space Technology Japan*, Vol. 12, pp. 31–38, 2013.
- [38] Noboru Takeichi, Daigo Inami, Masahiro Kudo, and Naoki Fujii. Arrival time controllability of a constrained tailored arrival path and its optimization. In *Proceedings of 27th International Congress of the Aeronautical Sciences (ICAS)*, 2010.
- [39] Peter J.M. Van Laarhoven and Emile H.L. Aarts. Simulated annealing. In *Simulated annealing: Theory and applications*, pp. 7–15. Springer, 1987.
- [40] Jun Zhou, Sonia Cafieri, Daniel Delahaye, and Mohammed Sbihi. Optimization of Arrival and Departure Routes in Terminal Maneuvering Area. In *Proceedings of 6th International Conference on Research in Air Transportation(ICRAT2014)*, 2014.
- [41] Ailsa H. Land and Alison G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, Vol. 28, No. 3, pp. 497–520, 1960.
- [42] Richard Bellman. Dynamic programming. *Science*, Vol. 153, No. 3731, pp. 34–37, 1966.
- [43] Fred Glover. Tabu search—part I. *ORSA Journal on computing*, Vol. 1, No. 3, pp. 190–206, 1989.
- [44] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1st edition, 1989.
- [45] Xiao-Bing Hu and Ezequiel Di Paolo. Binary-representation-based genetic algorithm for aircraft arrival sequencing and scheduling. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 9, No. 2, pp. 301–310, 2008.
- [46] Efrén Mezura-Montes and Carlos A. Coello Coello. Constrained optimization via multiobjective evolutionary algorithms. In *Proceedings of Multiobjective problem solving from nature*, pp. 53–75. Springer, 2008.
- [47] Frank Hoffmeister and Joachim Sprave. Problem-independent handling of constraints by use of metric penalty functions. pp. 289–294, 1996.
- [48] Tetsuyuki Takahama and Setsuko Sakai. Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites. In *Proceedings of 2006 IEEE International Conference on Evolutionary Computation(CEC2006)*, pp. 1–8. IEEE, 2006.
- [49] Andrzej Jaszkiewicz. On the performance of multiple-objective genetic local

- search on the 0/1 knapsack problem—a comparative experiment. *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 4, pp. 402–412, 2002.
- [50] Hisao Ishibuchi and Shiori Kaige. Effects of repair procedures on the performance of emo algorithms for multiobjective 0/1 knapsack problems. In *Proceedings of 2003 Congress on Evolutionary Computation(CEC2003)*, Vol. 4, pp. 2254–2261. IEEE, 2003.
- [51] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, Vol. 3, No. 4, pp. 257–271, 1999.
- [52] Mariko Tanaka, Yuki Yamagishi, Hidetoshi Nagai, and Hiroyuki Sato. Infeasible solution repair and MOEA/D sharing weight vectors for solving multi-objective set packing problems. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion(GECCO2018)*, pp. 71–72. ACM, 2018.
- [53] John E. Beasley, Mohan Krishnamoorthy, Yazid M. Sharaiha, and David Abramson. Scheduling aircraft landings—the static case. *Transportation science*, Vol. 34, No. 2, pp. 180–197, 2000.
- [54] Lucio Bianco, Paolo Dell’ Olmo, Stefano Giordani. Scheduling models and algorithms for tma traffic management. In *Modelling and simulation in air traffic management*, pp. 139–167. Springer, 1997.
- [55] Marco Dorigo and Mauro Birattari. *Ant colony optimization*. Springer, 2010.
- [56] Yuhui Shi and Russell C. Eberhart. Empirical study of particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation(CEC1999)*, Vol. 3, pp. 1945–1950. IEEE, 1999.
- [57] Marcus Randall. Scheduling aircraft landings using ant colony optimisation. In *6th International Conference Artificial Intelligence and Soft Computing(IASTED)*, pp. 129–133, 2002.
- [58] Tian Jungai and Xu Hongjun. Optimizing arrival flight delay scheduling based on simulated annealing algorithm. *Physics Procedia*, Vol. 33, pp. 348–353, 2012.
- [59] B.S. Girish. An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft landing problem. *Applied Soft Computing*, Vol. 44, pp. 200–221, 2016.
- [60] H. Pinol and John E. Beasley. Scatter search and bionomic algorithms for the aircraft landing problem. *European Journal of Operational Research*, Vol. 171, No. 2, pp. 439–462, 2006.
- [61] Torsten Fahle, Rainer Feldmann, Silvia Götz, Sven Grothklags, and Burkhard

- Monien. The aircraft sequencing problem. pp. 152–166. Springer, 2003.
- [62] Xiao-Bing Hu and Wen-Hua Chen. Genetic algorithm based on receding horizon control for arrival sequencing and scheduling. *Engineering Applications of Artificial Intelligence*, Vol. 18, No. 5, pp. 633–642, 2005.
- [63] Harilaos Psaraftis. A dynamic programming approach to the aircraft sequencing problem. 2012.
- [64] James V. Hansen. Genetic search methods in air traffic control. *Computers & Operations Research*, Vol. 31, No. 3, pp. 445–459, 2004.
- [65] Victor H.L. Cheng, Lara S. Crawford, and P.K. Menon. Air traffic control using genetic search techniques. In *Proceedings of 1999 IEEE International Conference on Control Applications*, Vol. 1, pp. 249–254. IEEE, 1999.
- [66] Vic Ciesielski and Paul Scerri. An anytime algorithm for scheduling of aircraft landing times using genetic algorithms. *Australian Journal of Intelligent Information Processing Systems*, Vol. 4, No. 3/4, pp. 206–213, 1997.
- [67] Vic Ciesielski and Paul Scerri. Real time genetic scheduling of aircraft landing times. In *Proceeding of 1998 IEEE International Conference on Evolutionary Computation(CEC1998)*, pp. 360–364. IEEE, 1998.
- [68] Hamsa Balakrishnan and Bala Chandran. Scheduling aircraft landings under constrained position shifting. In *Proceeding of Guidance, Navigation, and Control Conference and Exhibit*, 2006.
- [69] Fabio Furini, Martin Philip Kidd, Carlo Alfredo Persiani, and Paolo Toth. State space reduced dynamic programming for the aircraft sequencing problem with constrained position shifting. In *Proceeding of International Symposium on Combinatorial Optimization*, pp. 267–279. Springer, 2014.
- [70] 松枝友佳, 高橋里司. 航空滑走路スケジューリング: 複数滑走路を考える. *オペレーションズ・リサーチ*, Vol. 63, No. 11, pp. 693–699, 2018.
- [71] Sheng-Peng Yu, Xian-Bin Cao, and Jun Zhang. A real-time schedule method for aircraft landing scheduling problem based on cellular automation. *Applied Soft Computing*, Vol. 11, No. 4, pp. 3485–3493, 2011.
- [72] Ludovic Boursier, Bruno Favennec, Eric Hoffman, Aymeric Trzmiel, François Vergne, and Karim Zeghal. Merging arrival flows without heading instructions. In *Proceeding of 7th USA/Europe air traffic management R&D seminar*, pp. 1–8, 2007.
- [73] AirTop. <http://airtopsoft.com/en-route-simulation/tmatracon/>, 最終閲覧日:2019年9月24日.

- [74] 蔭山康太, 秋永和夫宮津義廣. 高速シミュレーションによる到着機処理の比較. 第 15 回電子航法研究所研究発表会, pp. 15–18. Electronic Navigation Research Institute(ENRI), 2015.
- [75] CTAS. <https://www.aviationsystemsdivision.arc.nasa.gov/research/foundations/index.shtml>, 最終閲覧日:2019 年 9 月 24 日.
- [76] Thomas J Davis, Heinz Erzberger, Steven M Green, and William Nedell. Design and evaluation of an air traffic control final approach spacing tool. *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 4, pp. 848–854, 1991.
- [77] Catya Zuniga and Daniel Delahaye. An efficient landing route structure to enhance airport capacity. In *Proceeding of 4th ENRI International Workshop on ATM/CNS*, 2015.
- [78] Anastasios Nikoleris and Heinz Erzberger. Autonomous system for air traffic control in terminal airspace. In *Proceedings of 14th AIAA Aviation Technology, Integration, and Operations Conference*, 2014.
- [79] Marcella Samà, Andrea D’ Ariano, Dario Pacciarelli. Rolling horizon approach for aircraft scheduling in the terminal control area of busy airports. *Procedia-Social and Behavioral Sciences*, Vol. 80, pp. 531–552, 2013.
- [80] Youkyung Hong, Byunghun Choi, Somang Lee, Keumjin Lee, and Youdan Kim. Optimal and practical aircraft sequencing and scheduling for point merge system. *International Federation of Automatic Control-PapersOnLine*, Vol. 50, No. 1, pp. 14644–14649, 2017.
- [81] Youkyung Hong, Byunghun Choi, Keumjin Lee, and Youdan Kim. Dynamic robust sequencing and scheduling under uncertainty for the point merge system in terminal airspace. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, No. 9, pp. 2933–2943, 2017.
- [82] Ji Ma, Daniel Delahaye, Mohammed Sbihi, Paolo Scala, and Miguel Antonio Mujica Mota. Integrated optimization of terminal maneuvering area and airport at the macroscopic level. *Transportation Research Part C: Emerging Technologies*, Vol. 98, pp. 338–357, 2019.
- [83] Youkyung Hong, Byunghun Choi, Keumjin Lee, and Youdan Kim. Dynamic robust sequencing and scheduling under uncertainty for the point merge system in terminal airspace. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, No. 9, pp. 2933–2943, 2017.
- [84] Man Liang, Daniel Delahaye, and Pierre Marechal. Conflict-free arrival and departure trajectory planning for parallel runway with advanced point-merge

- system. *Transportation Research Part C: Emerging Technologies*, Vol. 95, pp. 207–227, 2018.
- [85] Daichi Toratani, Ueno Seiya, and Takehiro Higuchi. Simultaneous optimization method for trajectory and sequence for receding horizon guidance in terminal area. *SICE Journal of Control, Measurement, and System Integration*, Vol. 8, No. 2, pp. 144–153, 2015.
- [86] Angela Nuic. User manual for the base of aircraft data (BADA) revision 3.10. *Atmosphere*, Vol. 2010, , 2010.
- [87] Catya Zuniga, Daniel Delahaye, and Miquel A Piera. Integrating and sequencing flows in terminal maneuvering area by evolutionary algorithms. In *Proceedings of 30th IEEE/AIAA Digital Avionics Systems Conference (DASC)*. IEEE, 2011.
- [88] Hidenori Chida, Catya Zuniga, and Daniel Delahaye. Topology Design for Integrating and Sequencing Flows in Terminal Maneuvering Area. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*.
- [89] M. Shohel Ahmed, Sameer Alam, and Michael Barlow. An evolutionary optimization approach for path planning of arrival aircraft for optimal sequencing. In *Proceedings of Intelligent and Evolutionary Systems*, pp. 1–16. Springer, 2017.
- [90] Min Xue and Shannon Zelinski. Optimal integration of departures and arrivals in terminal airspace. *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, pp. 207–213, 2013.
- [91] Israel Lugo-Cárdenas, Gerardo Flores, Sergio Salazar, and Rogelio Lozano. Dubins path generation for a fixed wing uav. In *2014 International conference on unmanned aircraft systems (ICUAS)*, pp. 339–346. IEEE, 2014.
- [92] Adriana Andreeva-Mori, Shinji Suzuki, and Eri Itoh. Scheduling of arrival aircraft based on minimum fuel burn descents. *ASEAN Eng J*, Vol. 1, No. 1, 2011.
- [93] Eugene L. Allgower and Kurt Georg. *Numerical continuation methods: an introduction*, Vol. 13. Springer Science & Business Media, 2012.
- [94] Kalyanmoy Deb, Ram Bhushan Agrawal, et al. Simulated binary crossover for continuous search space. *Complex systems*, Vol. 9, No. 2, pp. 115–148, 1995.
- [95] Joel Lehman and Kenneth O Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the 11th International Conference on Artificial Life (ALIFE XI)*, pp. 329–336, 2008.
- [96] Jorge Gomes, Pedro Mariano, and Anders Lyhne Christensen. Devising effec-

- tive novelty search algorithms: a comprehensive empirical study. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation (GECCO2015)*, pp. 943–950. ACM, 2015.
- [97] Joel Lehman and Kenneth O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 2011 annual conference on Genetic and evolutionary computation (GECCO2011)*, pp. 211–218. ACM, 2011.
- [98] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, Vol. 10, No. 2, pp. 99–127, 2002.
- [99] Giuseppe Cuccu and Faustino Gomez. When novelty is not enough. In *Proceedings of European Conference on the Applications of Evolutionary Computation*, pp. 234–243. Springer, 2011.
- [100] Transport Ministry of Land, Infrastructure and Japan Tourism. AIS JAPAN (Japan Aeronautical Information Service Center). <https://aisjapan.mlit.go.jp>, 最終閲覧日:2019年9月24日.
- [101] Service de l’information Aéronautique. eAIP FRANCE. <https://www.sia.aviation-civile.gouv.fr/>, 最終閲覧日:2019年9月24日.
- [102] Imen Dhief, Nour Houda Dougui, Daniel Delahaye, and Nouredine Hamdi. A new trans-atlantic route structure for strategic flight planning over the nat airspace. In *Proceedings of 2017 IEEE Congress on Evolutionary Computation (CEC2017)*, pp. 1511–1518. IEEE, 2017.
- [103] Terumitsu HIRATA. Mitigation of the delay caused by the special features of runway operation of haneda airport. *INFRASTRUCTURE PLANNING REVIEW*, Vol. 69, No. 5, pp. I-869–I-879, 2013.
- [104] Romaric Breil, Daniel Delahaye, Laurent Lapasset, and Eric Féron. Multi-agent systems to help managing air traffic structure. *Journal of Aerospace Operations*, Vol. 5, No. 1-2, pp. 119–148, 2017.
- [105] Daniel Delahaye and Stéphane Puechmorel. *Modeling and optimization of air traffic*. Wiley Online Library, 2013.
- [106] Ji Ma, Daniel Delahaye, Mohammed Sbihi, and Marcel Mongeau. Integrated optimization of terminal manoeuvring area and airport. In *6th SESAR Innovation Days*, 2016.
- [107] Supatcha Chaimatanan, Daniel Delahaye, and Marcel Mongeau. A hybrid meta-heuristic optimization algorithm for strategic planning of 4D aircraft trajectory-

- ries at the continental scale. *IEEE Computational Intelligence Magazine*, Vol. 9, No. 4, pp. 46–61, 2014.
- [108] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, pp. 100–107, 1968.
- [109] Akinori Murata, Masaya Nakata, Hiroyuki Sato, Tim Kovacs, and Keiki Takadama. Optimization of aircraft landing route and order: An approach of hierarchical evolutionary computation. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pp. 340–347, 2016.
- [110] Hamsa Balakrishnan and Bala Chandran. Efficient and equitable departure scheduling in real-time: new approaches to old problems. In *Proceedings of 7th USA-Europe Air Traffic Management Research and Development Seminar*, pp. 2–5, 2007.
- [111] Justin Montoya, Sivakumar Rathinam, and Zachary Wood. Multiobjective departure runway scheduling using dynamic programming. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15, No. 1, pp. 399–413, 2013.
- [112] Gautam Gupta, Waqar Malik, and Yoon Jung. A mixed integer linear program for airport departure scheduling. In *Proceedings of 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS)*, 2009.
- [113] Jason A.D. Atkin, Edmund K. Burke, John S. Greenwood, and Dale Reeson. Hybrid metaheuristics to aid runway scheduling at london heathrow airport. *Transportation Science*, Vol. 41, No. 1, pp. 90–106, 2007.
- [114] Ioannis Anagnostakis and J.Paul Clarke. Runway operations planning: a two-stage solution methodology. In *Proceedings of 36th Annual Hawaii International Conference on System Sciences*. IEEE, 2003.
- [115] Waqar Malik and Yoon C Jung. Exact and heuristic algorithms for runway scheduling. In *16th AIAA Aviation Technology, Integration, and Operations Conference*, p. 4072, 2016.
- [116] Michael Joseph Frankovich. Air traffic flow management at airports: a unified optimization approach. 2012.
- [117] Christabelle Bosson, Min Xue, and Shannon Zelinski. Optimizing integrated arrival, departure and surface operations under uncertainty. 2015.
- [118] Dag Kjenstad, Carlo Mannino, Tomas Eric Nordlander, Patrick Schittekat, and

Morten Smedsrud. Optimizing aman-smam-dman at hamburg and arlanda airport. *Proceedings of the SID, Stockholm*, 2013.

- [119] Gordon F Newell. Airport capacity and delays. *Transportation Science*, Vol. 13, No. 3, pp. 201–241, 1979.

関連論文の印刷公表の方法および時期

1. 全著者名 : Akinori Murata, Hiroyuki Sato, Keiki Takadama
論文題目 : Towards Adaptive Aircraft Landing Order with Aircraft Routes Partially Fixed by Air Traffic Controllers as Human Intervention
印刷公表の方法および時期 : SICE Journal of Control, Measurement, and System Integration, Vol. 11, Issue 2, pp. 105-112, 2018. (第 5 章に関連)
2. 全著者名 : Akinori Murata, Hiroyuki Sato, Keiki Takadama
論文題目 : Exploring Tradeoff Between Distance-minimality and Diversity of Landing Routes for Aircraft Landing Optimization
印刷公表の方法および時期 : SICE Journal of Control, Measurement, and System Integration, Vol. 11, Issue 5, pp. 409-418, 2018. (第 5 章に関連)
3. Akinori Murata, Daniel Delahaye, Keiki Takadama
論文題目 : Merging Flows and Optimizing Aircraft Scheduling in Terminal Maneuvering Area Based on GA
印刷公表の方法および時期 : International Conference for Research in Air Transportation (ICRAT 2018), Barcelona, Spain, 2018. (第 4 章に関連)
4. Akinori Murata, Hiroyuki Sato, Keiki Takadama, Towards Adaptive Aircraft
論文題目 : Landing Order with Aircraft Routes Partially Fixed by Air Traffic controllers as Human Intervention
印刷公表の方法および時期 : The 19th International Conference on Human-Computer Interaction (HCII), Human Interface and the Management of Information, Part II, Lecture Notes in Computer Science, Vol. 10274, pp. 422-433, 2017. (第 5 章に関連)
5. Akinori Murata, Hiroyuki Sato, Keiki Takadama
論文題目 : Optimization of Aircraft Landing Route and Order Based on Novelty Search

- 印刷公表の方法および時期：The 20th Asia-Pacific Symposium on Intelligent and Evolutionary Systems (IES), Proceedings in Adaptation, Learning and Optimization, Vol. 8, Springer, pp. 291-304, 2016（第5章に関連）。
6. Akinori Murata, Masaya Nakata, Hiroyuki Sato, Tim Kovacs, Keiki Takadama,
論文題目：Optimization of Aircraft Landing Route and Order: An approach of Hierarchical Evolutionary Computation, The 9th International Conference on Bio-inspired Information and Communications Technologies (BICT), 2015.
 7. 村田 暁紀, Daniel Delahaye, 高玉 圭樹
論文題目：実行不可能解における遺伝子内の最大制約違反量と平均制約違反量を活用した解修復法
印刷公表の方法および時期：計測自動制御学会, システム・情報部門学術講演会 2018, SS04-12, 2018.（第6章に関連）
 8. 村田 暁紀, Daniel Delahaye, 佐藤 寛之, 高玉 圭樹
論文題目：航空機着陸問題におけるクラスタリングを用いた分割反復最適化手法
印刷公表の方法および時期：電気学会, 第28回インテリジェント・システム・シンポジウム (FAN2018), ST-18-117, pp. 167-172, 2018.（第6章に関連）
 9. 村田 暁紀, 佐藤 寛之, 高玉 圭樹
論文題目：最適性と多様性のトレードオフを考慮したノベルティサーチに基づく多目的進化計算
印刷公表の方法および時期：情報処理学会, 第112回数理モデル化と問題解決 (MPS) 研究会, 1A, 2017.（第5章に関連）
 10. 村田 暁紀, 佐藤 寛之, 高玉 圭樹
論文題目：到着機スケジューリングにおける最適性と多様性のトレードオフを考慮した進化計算
印刷公表の方法および時期：計測自動制御学会, システム・情報部門 学術講演会 2016, O2-4,SS02-5, pp. 437-442, 2016.（第5章に関連）
 11. 村田 暁紀, 森本 紗矢香, 神馬 隆博, 原田 智広, 佐藤 寛之, 高玉 圭樹
論文題目：階層的進化計算を用いた動的航空機着陸経路スケジューリング
印刷公表の方法および時期：計測自動制御学会, 第42回 知能システムシンポジウム, F-12, 2015.（第5章に関連）

付録

フランス, シャルル・ド・ゴール空港 (固定経路) の実験におけるスケジュール対象の航空機

表 1: シャルル・ド・ゴール空港における航空機情報の詳細

ID	waypoint	entry time	altitude	<i>speed</i>	size
1	BALMU	35280.7	347.34	170.299	M
2	BALMU	35822.9	390.232	212.745	M
3	PETAX	35900.2	472.632	260	M
4	BALMU	35955.8	360.755	193.541	M
5	TALUD	35963.5	437.713	310	M
6	LUMAN	35976	473.291	274.98	M
7	PESUK	36085.3	441.996	243.393	H
8	BALMU	36194.2	416.162	201.857	M
9	KOVAK	36233.8	434.03	270.183	M
10	BALMU	36461.4	367.163	209.132	M
11	BALMU	36612.9	362.963	200	H
12	LUMAN	36769.6	505.811	220	M
13	BALMU	36900.3	355.737	215.972	M
14	LUMAN	36950.6	456.152	280.25	H
15	PETAX	37090.5	519.822	257.397	M

表 2: 自由経路における航空機の初期位置

seed	1	2	3	4	5	6	7	8	9	10										
ID1	2	12	4	25	0	12	3	25	5	13	1	0	3	13	5	0	2	13	4	1
ID2	2	16	4	4	5	17	0	5	2	18	3	6	4	19	6	7	1	20	3	8
ID3	5	17	5	12	6	8	0	4	0	25	1	21	1	16	1	12	1	8	2	4
ID4	2	4	1	2	6	24	5	22	4	20	2	18	1	15	6	13	5	11	3	9
ID5	0	22	4	7	2	17	6	2	4	11	2	21	6	6	3	16	1	1	5	11
ID6	2	9	1	15	6	22	4	3	3	9	2	16	0	22	5	3	4	10	2	16
ID7	3	9	5	4	2	24	4	19	0	13	2	8	5	3	1	23	3	17	5	12

次のページに続く

	1	2	3	4	5	6	7	8	9	10										
ID8	6	8	0	8	1	8	1	8	2	8	3	8	3	8	4	8	5	8	5	8
ID9	3	4	3	22	2	15	2	8	1	1	1	18	0	11	5	4	5	22	4	14
ID10	4	16	3	23	3	5	3	12	2	20	2	2	2	9	1	16	1	23	0	5
ID11	4	24	3	4	1	9	5	14	4	18	2	23	0	3	5	8	3	13	2	17
ID12	4	18	1	18	3	17	0	16	3	16	0	15	3	15	6	14	3	14	6	13
ID13	1	17	2	2	3	12	5	23	6	8	1	18	3	3	4	13	5	24	0	9
ID14	4	16	3	24	1	7	5	15	3	22	1	5	6	13	4	20	2	3	0	11
ID15	4	25	3	11	3	22	2	8	1	20	0	6	6	17	5	3	4	15	3	1
ID16	1	9	5	21	4	8	3	20	1	7	6	19	4	6	3	18	1	5	6	17
ID17	3	13	3	8	4	4	5	24	5	19	6	15	1	10	1	6	2	1	3	21
ID18	3	14	1	7	5	24	3	17	1	9	5	2	3	19	1	12	5	4	2	22
ID19	0	3	4	24	2	20	1	16	5	12	3	8	1	4	5	25	3	21	1	17
ID20	2	25	1	18	0	10	6	3	5	21	4	14	3	7	2	24	1	17	0	10
ID21	6	24	2	4	4	9	1	15	3	20	5	1	2	6	4	11	0	17	3	22
ID22	4	11	5	6	6	1	0	20	1	15	2	10	2	4	3	24	4	19	4	14
ID23	1	9	2	7	3	5	3	3	4	1	4	23	5	21	6	19	0	17	1	15
ID24	0	14	5	1	5	13	4	0	3	12	2	24	1	11	6	24	5	11	4	23
ID25	5	4	1	9	2	13	3	17	5	21	6	25	1	4	2	8	4	12	5	16
ID26	5	9	4	6	2	2	1	23	5	20	4	16	3	13	1	9	6	5	4	2
ID27	4	10	1	19	3	3	6	11	2	20	5	4	1	12	4	21	0	5	3	13
ID28	3	9	4	14	5	18	6	22	1	2	2	6	3	11	4	15	5	19	6	24
ID29	5	16	5	23	6	4	6	10	6	16	6	23	0	4	0	10	0	17	0	23
ID30	3	4	5	20	1	12	3	3	4	19	0	10	2	1	4	17	6	8	2	24
ID31	6	19	0	6	1	18	2	5	3	17	4	5	4	17	5	4	0	16	1	3
ID32	3	1	3	6	3	11	3	16	4	21	4	1	4	6	4	11	5	16	5	21
ID33	4	0	2	8	0	15	4	23	2	5	6	13	3	21	1	3	5	11	3	18
ID34	6	0	0	14	1	3	2	17	3	6	4	20	5	9	5	23	0	12	1	1
ID35	5	9	6	19	1	4	1	14	2	24	3	10	4	20	5	5	6	15	0	0
ID36	2	11	4	13	6	15	1	16	3	18	5	20	1	22	3	23	5	25	1	2
ID37	5	16	4	21	3	2	2	8	1	13	0	19	6	25	5	5	4	11	3	17
ID38	4	20	4	7	3	20	2	8	2	21	1	8	1	21	6	9	5	21	5	9
ID39	1	4	2	21	2	14	3	6	3	24	4	16	4	8	5	1	5	18	6	11

	1	2	3	4	5	6	7	8	9	10										
ID40	3	20	2	2	1	10	1	17	0	24	5	7	5	14	4	22	3	4	2	11
ID41	4	5	1	10	4	15	2	21	5	1	3	6	6	12	3	17	1	22	4	3
ID42	4	13	5	6	6	24	0	16	1	9	1	1	2	19	3	12	3	4	4	22
ID43	1	19	5	17	3	14	1	12	5	9	3	7	2	4	6	2	4	24	2	22
ID44	4	0	4	19	5	13	5	8	5	2	6	21	0	15	1	9	1	4	1	23
ID45	0	4	4	16	2	3	0	15	4	2	2	14	6	1	4	13	2	25	6	12
ID46	4	9	2	20	0	6	4	17	2	3	1	14	5	25	3	11	1	22	5	8
ID47	4	5	1	10	4	16	1	21	4	1	2	7	5	12	2	18	5	23	2	3
ID48	0	5	4	2	1	25	5	22	2	20	6	17	3	15	1	12	4	10	2	8
ID49	3	18	4	15	5	13	0	11	1	8	2	6	3	4	4	1	5	24	0	22
ID50	4	6	5	1	0	21	1	15	2	10	3	5	4	25	5	20	6	15	0	9
ID51	2	1	1	9	6	16	4	23	3	5	2	12	0	19	5	2	4	9	2	16
ID52	4	14	0	9	2	4	5	24	1	19	3	14	5	10	1	5	3	25	5	20
ID53	5	10	4	2	3	20	3	12	2	4	1	22	6	14	5	6	4	24	3	16
ID54	3	1	3	6	3	11	3	16	3	21	3	0	3	5	3	10	3	15	3	20
ID55	4	17	1	8	4	25	0	16	3	7	6	24	2	15	5	7	2	23	5	14
ID56	1	18	1	1	2	9	2	17	3	0	4	9	4	17	5	25	5	8	6	16
ID57	1	18	3	18	5	18	2	19	4	19	0	19	2	20	5	20	1	20	3	20
ID58	3	19	4	17	4	15	5	13	5	11	6	9	0	7	1	5	1	3	2	1
ID59	6	23	2	6	5	15	1	23	3	7	6	15	2	24	4	7	1	15	3	24
ID60	5	24	2	17	6	11	3	4	6	23	3	16	6	10	3	3	6	21	3	15
ID61	2	23	4	8	5	17	0	1	2	11	3	20	5	4	0	13	2	23	3	7
ID62	6	17	5	8	5	25	4	16	4	8	3	24	2	16	2	7	1	24	1	15
ID63	4	24	2	20	1	17	5	14	3	11	2	8	0	5	4	2	3	24	1	21
ID64	5	15	2	6	6	22	3	13	6	4	3	20	0	11	4	2	1	18	4	9
ID65	3	13	1	13	5	13	3	13	1	13	5	12	3	12	1	12	5	12	2	12
ID66	2	14	4	18	6	22	2	2	4	6	6	11	2	15	4	19	6	24	2	3
ID67	3	6	6	8	3	10	1	13	4	15	1	17	4	19	2	22	5	24	2	1
ID68	2	14	5	21	2	3	6	10	3	16	6	23	3	5	0	12	3	19	0	0
ID69	3	5	2	12	1	19	0	2	6	9	5	16	4	24	3	6	2	14	2	21
ID70	4	18	3	18	2	18	1	18	0	18	6	18	5	18	4	18	3	18	2	17
ID71	1	15	1	12	0	9	5	5	5	2	4	24	4	20	3	17	2	14	2	10

	1	2	3	4	5	6	7	8	9	10										
ID72	2	10	4	9	6	9	1	9	3	9	5	9	1	9	3	9	5	9	1	9
ID73	0	24	3	8	5	17	1	1	3	10	6	19	2	3	4	12	1	21	3	4
ID74	1	23	2	11	2	24	2	12	3	25	3	12	4	0	4	13	4	1	5	14
ID75	4	8	5	24	5	16	0	7	1	24	2	15	3	6	3	23	4	14	5	6
ID76	1	15	4	11	1	6	4	2	1	23	4	19	2	15	5	11	2	7	5	3
ID77	5	19	1	4	4	14	6	25	2	10	5	20	1	6	3	16	6	2	2	12
ID78	5	12	3	22	6	7	3	17	1	3	4	13	1	23	5	8	2	18	5	3
ID79	1	1	3	18	5	11	1	3	4	20	6	12	2	4	5	21	1	14	3	6
ID80	5	23	6	22	0	22	1	21	1	21	2	20	2	20	3	19	3	18	4	18

以上

固定経路のシナリオ 表 3 は 16 機の航空機の詳細スケジュールを示しており、左の列から入域時間、航空機のサイズ、入域場所、航空機の色をそれぞれ表している。以下に 16 機から 19 機までのシナリオを表す。入域時間は 10 秒単位で設定され、航空機のサイズは Medium を 2 で統一し、エントリーポイントは 2ヶ所設定され、入域速度はノット表記である。

表3 16機のシナリオにおける航空機の詳細

Time	Size	Entry Point	Speed
800	2	4	471
750	2	3	511
130	2	4	514
580	2	4	479
480	2	4	518
240	2	4	454
530	2	3	504
850	2	4	482
90	2	3	457
480	2	3	506
690	2	3	516
640	2	4	520
10	2	4	446
1000	2	4	487
710	2	4	513
910	2	4	506

表 4 17 機のシナリオにおける航空機の詳細

Entry Time	Size	Entry Point	Speed
230	2	3	447
220	2	4	483
890	2	3	461
90	2	3	505
840	2	3	493
360	2	3	473
520	2	3	517
290	2	3	497
660	2	3	514
470	2	4	465
990	2	4	478
760	2	4	481
930	2	4	502
140	2	4	467
580	2	4	505
830	2	4	449
390	2	4	448

表 5 18 機のシナリオにおける航空機の詳細

Entry Time	Size	Entry Point	Speed
370	2	4	482
330	2	3	455
520	2	4	453
960	2	4	485
60	2	4	488
430	2	3	507
430	2	4	513
970	2	3	496
770	2	3	470
170	2	3	443
560	2	3	485
40	2	3	456
720	2	3	453
610	2	3	468
230	2	4	503
830	2	4	489
890	2	4	481
120	2	3	473

表 6 18 機のシナリオにおける航空機の詳細

Entry Time	Size	Entry Point	Speed
560	2	4	444
960	2	4	506
270	2	4	515
90	2	4	475
620	2	4	496
150	2	4	476
60	2	3	499
970	2	3	489
550	2	3	465
750	2	3	502
620	2	3	489
440	2	4	494
420	2	3	461
830	2	4	514
40	2	4	453
340	2	4	486
690	2	3	513
700	2	4	492
890	2	4	520

表 7 19 機のシナリオにおける航空機の詳細

Entry Time	Size	Entry Point	Speed
560	2	4	444
960	2	4	506
270	2	4	515
90	2	4	475
620	2	4	496
150	2	4	476
60	2	3	499
970	2	3	489
550	2	3	465
750	2	3	502
620	2	3	489
440	2	4	494
420	2	3	461
830	2	4	514
40	2	4	453
340	2	4	486
690	2	3	513
700	2	4	492
890	2	4	520

クラスタリング手法の結果詳細

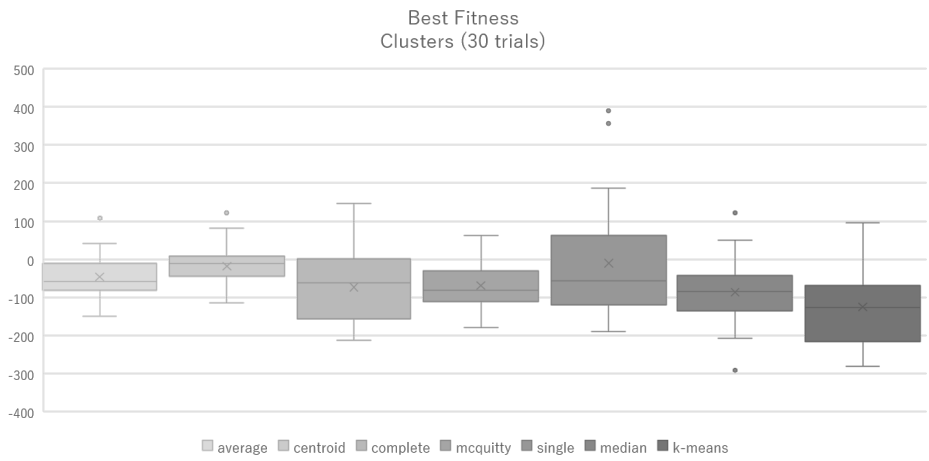


図1 クラスタリング毎の Best Fitness (30 試行)

クラスタリングを使用した場合の航空機の詳細

図 2-6 は各手法の実際の着陸予定時刻と 10 試行における各航空機の着陸時刻の分散, 各航空機の 10 試行における速度の分散及び速度の分布を表している。



図2 Median による航空機の詳細

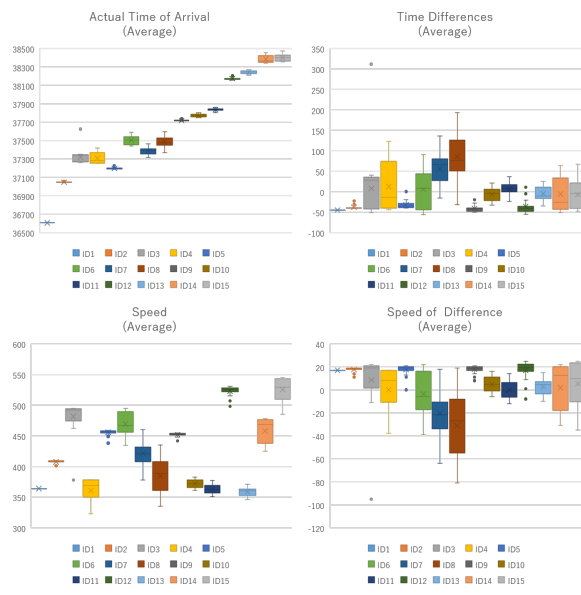


図3 Average による航空機の詳細

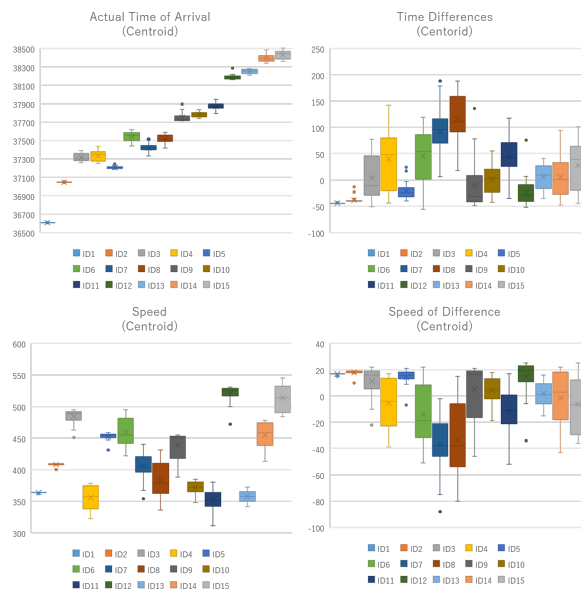


図4 Centroidによる航空機の詳細

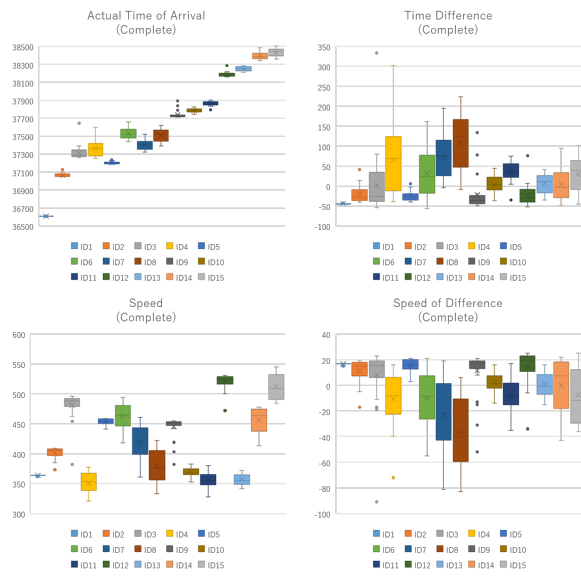


図5 Completeによる航空機の詳細

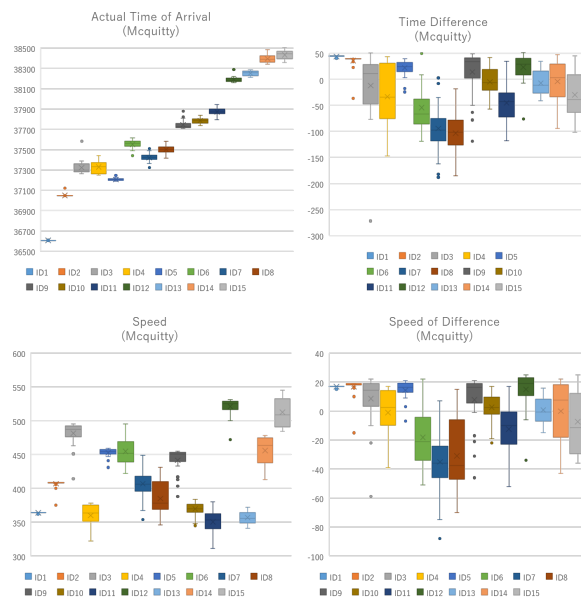


図6 Mcquitty による航空機の詳細

複数パッケージにおける結果

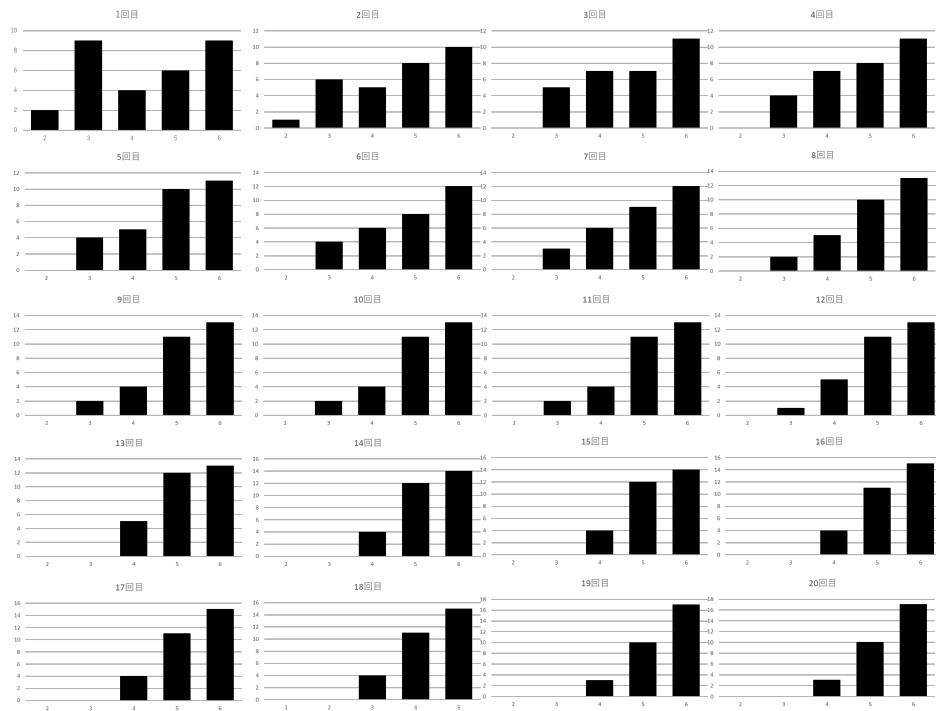


図7 4パッケージの中での最良評価値を持つクラスター数のヒストグラム

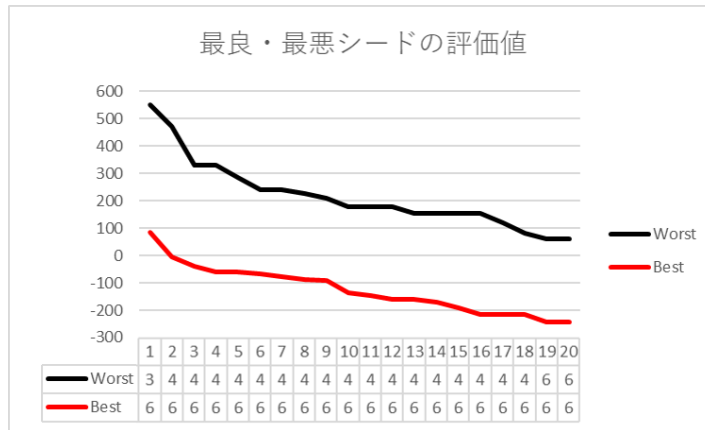


図8 4パッケージの中での最良・最悪シードにおける Fitness の推移

迂回経路を使用しない場合の結果

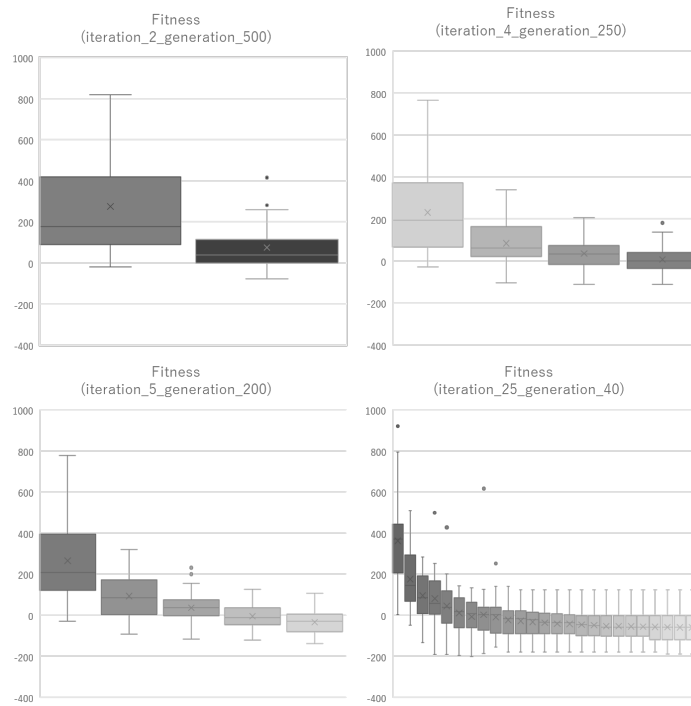


図9 迂回を使用しない場合における Iteration を変化させた場合の Fitness の推移