

学位論文

ヒューリスティック手法による
宇宙探査機の機体構成と
惑星間軌道の複合領域最適化

2013 年 12 月

東京大学大学院 工学系研究科
航空宇宙工学専攻
今野 友和

Abstract

本論文は自然現象である銀河形成と台風の渦の中心を求めるアルゴリズムを有する新しいヒューリスティック最適化手法の提案を行っている。有効性を検証するための様々な種類の問題に適用し、実装可能なアルゴリズムを有していることを示している。

ここで簡単に本提案手法の概要を述べる。提案手法はヒューリスティックな法則によって多探索点を移動させる手法であり、銀河の渦と台風の渦が似ていることからアナロジーの着想を得たものである。手法の特徴となるポイントは大きく分けて4つある。1. 多探索点の局所集合アルゴリズムは重力の方程式を用いる。初期の天体の重力収縮による成長過程を模擬している。2. 局所解脱出アルゴリズムも備えている。天体の質量が大きくなるとジェットを噴出する天体現象を再現する。3. 台風の渦を再現するアルゴリズムもある。探索空間を強制的に回転させ、探索点にコリオリ力を発生させる。探索点群を渦状に遠回りに移動させることで大域的な最適解の発見確率を上げることが出来る。4. 探索空間の端を連結する。地球が球状であることを再現したものであり、計算上はトーラス面を探索点が移動する。

提案手法をテスト問題に適用して有効性の検証を行っている。テスト問題として不静定トラス問題、数学的多峰性関数問題、ポートフォリオ問題に適用している。提案手法の有効性は、テスト問題への適用を通して示されている。次に応用問題として数値シミュレーションによる宇宙探査機の機体構成と軌道の複合領域における同時最適化問題に適用した。ミッションは地球と火星を往復することを考え、推進装置は電気推進機を用いる。本章内では無人探査機と有人探査機の2つのパターンについて適用した結果を示している。宇宙機の機体構成は簡易重量推算式を用いて各サブシステムの重量と電力を見積もる。最適化問題上の設計変数は地球出発日時、火星でのランデブー時間、初期全電力、推力の方向角、スラストのノズル形状がある。目的関数は機体重量とミッション期間の2つあり、多目的最適化とした。問題設定を複雑にした場合および多目的最適化問題について提案手法を適用したときの有効性を示し、他最適化手法と比較して本手法の優位性を述べている。また、最適化計算結果から機体重量とミッション時間の相関関係を得ることが出来た。

以上から、本提案手法は多峰性を持つ解空間においても、コリオリ力による探索点の強制回転によって大域的最適解に到達しやすいという知見を得られている。また、宇宙工学の複合領域最適化問題にも応用可能であることを示しており、航空宇宙工学上に存在する最適化問題の解決に貢献できると考えている。

目次

Abstract	i
第1章 序論	1
1.1 本論文の構成	1
1.2 ヒューリスティック最適化手法	1
1.3 本研究の位置づけと過去の最適化手法に関する研究	3
1.4 本論文の目的	5
第2章 提案手法	6
2.1 概要	6
2.2 銀河形成過程を模擬	7
2.2.1 重力収縮 一局所収束アルゴリズム	7
2.2.2 宇宙ジェット 一局所解脱出アルゴリズム	8
2.3 台風の回転を模擬	9
2.3.1 コリオリ力 一大域解発見アルゴリズム	10
2.3.2 トーラス面構造 一探索点移動幅向上	13
2.4 定式化	14
2.4.1 運動方程式と離散化	14
2.4.2 最適化手法としての利点	15
2.5 最適化計算上のテクニック	15
2.5.1 計算テクニック	15
2.5.2 計算パラメータの指標（オーダー値）	17
2.5.3 ペナルティ関数法	18
2.5.4 多目的最適化	18
2.6 フローチャートとアルゴリズム	21
2.7 他のヒューリスティック手法	22
2.7.1 Particle Swarm Optimization	22
2.7.2 PSOのアルゴリズム	23
2.7.3 遺伝的アルゴリズム Genetic algorithm	24

2.7.4	GAのアルゴリズム	24
第3章	テスト関数へ適用	27
3.1	数学的テスト関数問題への適用	27
3.1.1	問題設定（多峰性問題） ^{[2][21]}	27
3.1.2	適用結果（多峰性問題）	28
3.1.3	コリオリ力を導入する場合としない場合の比較	31
3.1.4	Schwefel 関数の係数を変更した場合の比較	33
3.2	不静定トラス問題	36
3.2.1	問題概要	36
3.2.2	問題設定（トラス問題）	37
3.2.3	適用結果（トラス問題）	38
3.2.4	問題設定を制限した場合	41
3.3	ポートフォリオ問題へ適用	42
3.3.1	問題概要	42
3.3.2	問題設定	42
3.3.3	適用結果（ポートフォリオ）	44
3.3.4	他手法との比較	45
3.4	高次元数学的多峰性問題へ適用	47
3.4.1	問題概要	47
3.4.2	問題設定（テスト関数） ^[13]	47
3.4.3	適用結果（テスト関数）	49
3.4.4	提案手法のみ 100 次元と 1000 次元に拡張	51
第4章	宇宙探査機へ応用	53
4.1	はじめに 概要および目的	53
4.2	無人宇宙探査機の惑星間軌道計画の最適化	54
4.2.1	宇宙機の機体構成モデル ^[44]	55
4.2.2	惑星間軌道計算モデル	58
4.2.3	最適化問題設定	60
4.2.4	最適化計算結果	62
4.3	有人宇宙探査機の機体構成と惑星間軌道の複合領域最適化	72
4.3.1	居住モジュール	73
4.3.2	質量と電力の簡易推算式	73
4.3.3	MPDスラスト	75

4.3.4	軌道計画 [35][36][37][41]	82
4.3.5	最適化問題設定	84
4.3.6	最適化計算結果	86
第 5 章	結論	100
5.1	本論文の結論	100
5.2	今後の課題	101
5.3	本提案手法の適用傾向	101
	謝辞	104
	参考文献	105

目 次

2.1	銀河の渦形成 ^[3]	6
2.2	台風の渦 ^[4]	6
2.3	重力場を模擬した解空間	8
2.4	宇宙ジェット ^[5]	9
2.5	コリオリ力	10
2.6	評価値軸と変数軸	11
2.7	トーラス面	13
2.8	重心位置と最良解保存	16
2.9	二連星	16
2.10	提案手法のフローチャート	21
3.1	Schwefel 関数	28
3.2	Schwefel 関数の解析解位置	28
3.3	初期解、重心と最良解の履歴 (提案手法、数学関数)	29
3.4	各探索点の履歴 (提案手法、数学関数)	30
3.5	初期解、重心と最良解の履歴 (PSO、数学関数)	30
3.6	初期解、重心と最良解の履歴 (GA、数学関数)	31
3.7	コリオリ力の導入不導入の誤差比較	32
3.8	係数 $A = 0.5$ のとき	35
3.9	係数 $A = 2.0$ のとき	35
3.10	三部材不静定トラス	37
3.11	初期分布、重心履歴、各最良解 (提案手法、トラス問題)	39
3.12	探索点の移動履歴 (提案手法、トラス問題)	39
3.13	初期分布、重心履歴、各最良解 (PSO、トラス問題)	40
3.14	初期分布、重心履歴、各最良解 (GA、トラス問題)	41
3.15	初期分布、重心履歴、各最良解 (提案手法、制限トラス問題)	42
3.16	初期分布、重心履歴、各最良解 (提案手法、ポートフォリオ)	44
3.17	探索点群の移動履歴 (提案手法、ポートフォリオ)	44

3.18 初期分布、重心履歴、各最良解 (PSO、ポートフォリオ)	46
3.19 初期分布、重心履歴、各最良解 (GA、ポートフォリオ)	47
3.20 f6 - Rastrigin 関数	48
3.21 f7 - Rosenbrock 関数	48
4.1 はやぶさ (無人宇宙探査機) ^[42]	54
4.2 目的関数 (無人宇宙探査機)	64
4.3 設計変数 (無人宇宙探査機)	64
4.4 無人宇宙探査機の軌道計画	65
4.5 推力値 (無人宇宙探査機)	66
4.6 重量値の時間履歴 (無人宇宙探査機)	66
4.7 最悪解の軌道計画 (無人宇宙探査機)	67
4.8 地球→火星の軌道	68
4.9 ランデブー軌道	68
4.10 火星→地球の軌道	68
4.12 PSO 目的関数結果 (無人)	70
4.13 GA 目的関数結果 (無人)	70
4.14 PSO 設計変数結果 (無人)	70
4.15 GA 設計変数結果 (無人)	70
4.16 PSO 軌道結果 (無人)	71
4.17 GA 軌道結果 (無人)	71
4.18 PSO 推力値結果 (無人)	71
4.19 GA 推力値結果 (無人)	71
4.20 PSO 機体重量結果 (無人)	71
4.21 GA 機体重量結果 (無人)	71
4.22 有人宇宙探査機 (スピン型) のイメージ図 ^[43]	72
4.23 MPD スラスタ ^[47]	75
4.24 5つのコントロールポイントによるノズル形状表現	79
4.25 ノズル入口と出口に関する応答局面図	81
4.26 宇宙探査機の推力方向範囲	83
4.27 宇宙探査機の推力方向角コントロールポイント	83
4.28 目的関数 (有人宇宙探査機)	88
4.29 設計変数 (有人宇宙探査機)	88
4.30 有人宇宙探査機の軌道計画	89

4.31 推力値 (有人宇宙探査機)	90
4.32 推力方向角 (有人宇宙探査機)	91
4.33 重量値の時間履歴 (有人宇宙探査機)	91
4.34 MPD スラスタのノズル形状 (有人宇宙探査機)	93
4.36 PSO 目的関数結果 (有人)	95
4.37 G A 目的関数結果 (有人)	95
4.38 PSO 設計変数結果 (有人)	95
4.39 G A 設計変数結果 (有人)	95
4.40 PSO 軌道結果 (有人)	96
4.41 G A 軌道結果 (有人)	96
4.42 PSO 推力値結果 (有人)	96
4.43 G A 推力値結果 (有人)	96
4.44 PSO 推力方向角結果 (有人)	96
4.45 G A 推力方向角結果 (有人)	96
4.46 PSO 機体重量結果 (有人)	97
4.47 G A 機体重量結果 (有人)	97
4.11 無人宇宙探査機の三面図	98
4.35 有人宇宙探査機の三面図	99
5.1 本提案手法と他手法との適用傾向	103

目 次

3.1	各手法の設定パラメータ（数学関数）	28
3.2	コリオリ力を導入した場合としない場合の比較	32
3.3	係数 A 変更による各解析解の値	35
3.4	回転量 ω と平均移動幅（無次元）	36
3.5	係数 A と回転量 ω を変更した場合の到達率	36
3.6	各手法の設定パラメータ（トラス問題）	38
3.7	各投資先と運用利率	43
3.8	各手法の設定パラメータ（ポートフォリオ問題）	43
3.9	数学的多峰性関数式	48
3.10	数学的多峰性関数（解析解）	49
3.11	各手法の設定パラメータ（20 次元テスト関数）	49
3.12	各手法の計算結果（20 次元テスト関数）	50
3.13	提案手法の設定パラメータ（高次元テスト関数）	51
3.14	提案手法における高次元テスト関数の計算結果	52
4.1	宇宙機の質量に関する簡易推算式	58
4.2	宇宙機の電力に関する簡易推算式	59
4.3	最適化手法の設定パラメータ（無人宇宙探査機）	61
4.4	機体構成諸元（無人宇宙探査機）	68
4.5	PSO 機体諸元結果（無人）	72
4.6	GA 機体諸元結果（無人）	72
4.7	質量に関する簡易推算方程式（有人宇宙探査機）	74
4.8	電力に関する簡易推算方程式（有人宇宙探査機）	75
4.9	MPD スラスタの準一次元モデルでの計算条件	78
4.10	最適化手法の設定パラメータ（有人宇宙探査機）	86
4.11	機体構成諸元（有人宇宙探査機）	92
4.12	PSO 機体諸元結果（有人）	97
4.13	GA 機体諸元結果（有人）	97

5.1 最適化問題の設定傾向による本手法と他手法の比較	103
---------------------------------------	-----

第1章 序論

1.1 本論文の構成

本論文の構成は以下のようになっている。

第1章は本章であり、序論を述べている。ヒューリスティック最適化手法について、過去の研究と本研究の目的について述べている。

第2章では、本論文のメインテーマである、既存にはないヒューリスティック最適化手法を提案する。提案手法の特徴を述べ、定式化する。比較するために既存のヒューリスティック手法について述べている。

第3章では、提案手法をテスト問題に適用した。テスト問題として不静定トラス問題、数学的多峰性関数問題、ポートフォリオ問題に適用している。提案手法の有効性は、テスト問題への適用を通して示されている。

第4章では、応用問題として宇宙探査機の機体構成と軌道の複合領域最適化問題に適用した。無人探査機と有人探査機の2つのパターンについて適用した結果を示した。問題設定を複雑にした場合および多目的最適化問題について提案手法を適用したときの有効性を示す。

第5章には、本論文で得られた結果についてまとめている。

1.2 ヒューリスティック最適化手法

最適化問題において目的関数が多峰性を有する場合や探索空間が連続ではなく離散空間であった場合には、感度解析を行う線形・非線形計画法は適用が難しい。設計変数が規格などにより特定の値の中からは選択できない場合には最適化問題は有限の個数の要素からなる実行可能領域内で、目的関数が最大もしくは最小になる解を探索する組み合わせ最適化問題となる。計算量が爆発するような組み合わせ最適化問題での大域的最適解を探索する手法としてヒューリスティック手法（発見的探索手法）が注目されている。特に最近では組合せ最適化問題を対象とする場合が多かったヒューリスティクス手法でも連続かつ多峰性構造の非線形最適化問題に対する手法としても有用性を検証されている^{[1][2]}。ヒューリスティック手法では必ずしも実行可能性や最適性が保障されている場合ばかりではないが、近年の研究によってより良い解が実用的な計算コストで探索可能であることが分かってきており、一部の

ヒューリスティック手法は実用化されている。ヒューリスティック手法は多数の方法が存在し、これらの手法が線形・非線形計画法と異なる点として以下のポイントがある。1. 多点による探索方法、2. 過去の探索履歴から探索方法を学習する方法、3. 改悪を許可する場合もある方法、4. 経験的に解探索を構築する方法がある。ヒューリスティック手法はこれらのポイントを内包する場合が多い。また、ヒューリスティック手法の別の特徴として多くの手法が何らかのアナロジーを持つことがあげられる。代表的な例として、焼きなまし法は金属の焼きなまし現象から、遺伝的アルゴリズムは生物の遺伝学、粒子群最適化法は蜂や魚群の集団知能、タブーサーチは人間の記憶過程にアナロジーを持つ。特に生物の遺伝と進化を模擬した遺伝的アルゴリズムや遺伝的プログラミング、粒子群最適化法は進化型計算手法と呼ばれ現在も研究が進められている。次章から説明する本論文のメインテーマである新しく提案するヒューリスティック手法もこの進化型計算手法に属する。

ヒューリスティック手法の中でも探索点が多数あり解空間の探索を行う多点探索に関しては、工学問題によく設定される多目的最適化問題と相性が良い。現実世界の問題の多くは、複数の評価基準を持つ多目的最適化問題である。このとき、複数の評価基準は、一方を良くすれば他方が悪くなるようなトレードオフの関係にあることが多い。例えば、製造業において、製品の機能や性能の最大化と生産コストの最小化はトレードオフの関係にある。また、我々が日常的に購入する商品においても、商品の機能や性能と価格の間にトレードオフの関係が存在する。さらに、交通機関やホテルなどのサービスについても、サービスの質と価格の間にトレードオフの関係が存在する。我々はこのようなトレードオフの関係を総合的に考慮し、自分自身の都合に応じた商品を選択することになる。たとえば、航空券を購入する場合では目的地へのルートや航空会社の選択など代替案の数は多くても 100 件程度あるので、旅行者が個々の代替案を比較し意思決定を行うことも不可能ではない。この場合は価格と移動時間の間、価格と快適性の間でトレードオフの関係が存在する。しかし製品設計や生産計画などの問題では、すべての代替案を人間が比較することは不可能な場合が多い。例えば、40 個の機能から必要な機能だけを選択してシステムを構成するという単純なシステム設計問題でも、システム構成の可能な組合せは 1 兆程度になり、設計者が個々の代替案を比較することは不可能である。そのため、複数の評価基準を考慮し、有望な代替案を効率的に探索する多目的最適化手法が必要となる。システム設計問題から分かるように、実行可能な代替案の総数が指数関数的に増加するような問題も多い。そのため、複数の良好な代替案を発見するための効率的な探索手法が必要となる [2]。

事前情報として意思決定者の選択候補決定に関する優先情報が必要であるが優先情報の獲得は必ずしも容易ではない。そのため、最終的に獲得された候補解は意思決定者の真の決定思考に対応していない可能性もある。事前情報として優先情報を明示的に与えることに比較すると、複数の候補解から決定解を選択することは容易である。多目的最適化手法は、意思

決定者の選択優先情報を事前情報として利用し、優先情報を最もよく満たす最適解を提示する。事前に選択優先情報を与えることが困難な場合でも、具体的な代替案が提示されれば自分の都合に合わせて意思決定を行うことが可能な場合も多い。たとえば、東京とニューヨーク間の往復航空券を購入する場合は移動時間と価格に関する目標や制約条件を設定するよりも旅行代理店の担当者から提示された複数の候補解から最終的に購入する航空券を選択する方が容易である。ヒューリスティックな最適化手法では後者のアプローチが用いられる。すなわち、複数の候補解を意思決定者に提示するために、多目的最適化問題の多様なパレート最適解を発見することのできるアルゴリズム開発が行われている。一般的な最適化手法と比較した場合でのヒューリスティックな最適化手法の最大の特徴は、アルゴリズムの1回の実行で多数の最適解候補を同時に獲得することを明確に意図していることである [22]。

良好で多様な候補解を意思決定者に提示するために、ヒューリスティックな最適化手法では個体群の多様性とパレート最適解集合への収束性を同時に考慮する必要がある。個体群がパレート最適解集合から離れていれば、良解方向への探索を進めることで個体群をパレート最適解集合へ近づけることが重要であり、収束性の向上が重要である。一方、個体群がパレート最適解集合に近い状況では、多様性の向上が重要である。このように、多様性と収束性を同時に考慮することが必要であるが、多様性と収束性は互いに競合するためにアルゴリズムの設計では、両者のバランスを取ることが重要になる。なお、単一目的最適化問題に対する進化型最適化アルゴリズムの設計においても多様性と収束性のバランスは重要であるが、単一目的最適化と多目的最適化では多様性の役割が大きく異なることに注意が必要である。単一目的最適化では、局所解に収束することなく、大域的な最適解を発見するために、決定変数空間の広い範囲を探索するという意味で多様性が重要である。一方、多目的最適化では、多様なパレート最適解を発見するために、目的関数空間での多様性が重要になる。すなわち、単一目的最適化では大域的な最適解を発見するための手段として決定変数空間内での多様性が重要であるが、多目的最適化では目的関数空間内で多様なパレート最適解を獲得することが重要である [2]。

1.3 本研究の位置づけと過去の最適化手法に関する研究

近年のコンピュータを用いる進化的な自動最適化手法の理論および設計論の研究について概説する。コンピュータCPUの性能向上に伴い、進化論的最適化手法の実問題への応用が盛んに行われている。進化論的最適化手法は膨大な繰り返し計算を必要とするが、大域的な探索を行うことができ、対象とする問題に連続性や微分可能性を仮定する必要がないために幅広い分野で応用されている。代表例としては、遺伝的アルゴリズム (GA)、遺伝的プロ

グラミング (GP)、進化戦略 (ES)、進化的プログラミング (EP)、粒子群最適化法 (PSO)、アントコロニー最適化法 (ACO)、免疫的アルゴリズム (IA)、など自然進化や生物の振る舞いから発想を得た最適化手法では性能改善のための工夫にも中立進化の仕組みの導入や二倍体型の解表現など、生物とのアナロジーに基づくものが多く提案されている。また、タブーサーチ (TS) や Memetic アルゴリズム (MA) では、メタヒューリスティックスとしての有用性に特に重点が置かれ展開されている。シミュレーテッドアニーリング (SA) を代表とする、統計物理学におけるサンプリングテクニックを応用した最適化手法では、その最適化原理の理解に強い関心が置かれていた [6][7]。

この中において本研究では新しい最適化手法の提案を行う。本提案手法もこの進化論的最適化手法に分類される。2つの自然現象である、銀河形成と台風の渦に着想を得た最適化手法である。この提案手法の詳細は次章で述べるが、図 2.1 が銀河系の渦で図 2.2 が台風の渦のイメージ図であり、この2つの図のようにこの最適化手法は探索点を渦を巻くように移動させながら計算進行させる。探索点が多数点あり、個体同士の情報共有により速度を求めて移動先を決定するという観点から粒子群最適化法 (PSO) に類似したヒューリスティックな手法であると考えられる。本提案手法の独自性を述べるために、既存手法の改良版 PSO について述べる。PSO の代表的な改良版手法をいくつか挙げる。M.Clerc らが行った、Constriction Method^[8] は事前の安定解析に基づき基礎式の係数パラメータを収束性の弱い値に設定する方法である。さらに Random Inertia Weight Method^[9] や Linearly Decreasing Inertia Weight Method^[10] は最適化イタレーションによって PSO の基礎式の数値係数を変化させる手法である。以上の3つの改良版は係数パラメータ調整を行うことで良い計算解を得ることが出来ると参考文献 [8][9][10] で述べられている。しかし、これらの改良版は PSO の強い収束性を弱く遅くしているだけとも言え、根本的に大域的な探索アルゴリズムを有していないと考えることも出来る。上記以外には Activity Feedback PSO^{[11][12]} があり個体群の密集度を計測フィードバックすることにより、移動速度の計算式内の係数を変化させる手法である。密集度のスケジュール管理が難しいことが挙げられ、この手法も収束性を弱くしている方法といえる。

これらの問題点を解決するために探索点群の収束性の緩和と最適解の遠方での大域的探索のアルゴリズムの導入を図った。提案する手法はそれらのデメリットを改善するものであり、広域的な大域的探索のアルゴリズムを有する。計算者は大域的最適解を発見するために計算途中の個体収束過程において遠方の空間も探索したい。これを実現させるために移動計算に重力の式を用い、さらにコリオリ力を再現するために変数空間自体を回転させるというアイデアを組み合わせた。これらは本手法独自のものであり、他手法にはない新規性を有しているといえる。提案手法について詳しくは第2章で説明している。提案手法は多点探索を行い PSO に類似すると述べたが、既存の PSO 手法と異なるところは探索点の移動

方法である。移動幅を計算する基礎式に重力の式を用いている。これにより、最適解付近では強い引力が働き、遠方では引力が弱くなる。急速に個体同士が収束せずに探索することが出来る。さらに解探索をサポートするためにコリオリ力による渦状回転の概念を導入した。台風の渦から着想を得ている。これにより計算過程の暫定最良解の遠方を大域的に探索することができる。PSO では個体群の最良解から遠い個体ほど強い力を受けるので、急速に最適解に収束するメソッドであり、局所解に収束してしまう危険性を有している。提案手法の利点は最適解付近は強い収束性を有し、遠方は広範囲探索を行うという2つの優位性を兼ね備えている。探索点移動の計算過程は他に類似するものがなく本提案手法は独自性があるといえる。また、改良版 PSO の中では GA における突然変異の概念を取り入れた Vibrational mutation PSO^[13] も提案されており、乱数で個体を近傍に少し移動させて個体群がより評価値の良い場所に移行するというものである。局所解に陥ってしまった場合の脱出アルゴリズムと考えることが出来る。突然変異の概念は提案手法にも導入されていて第2章で詳しく述べている。

1.4 本論文の目的

本論文の目的は既存の手法にはない大域的最適化アルゴリズムを有する新手法を提案することにある。提案手法の基幹となるアイデアとアルゴリズムを説明しテスト問題に適用する。第3章のテスト問題での適用では単一目的最適化を行い、単純な工学的最適化問題と数学的多峰性関数問題の複数の最適化問題に適用する。本提案手法が実装可能な最適化アルゴリズムであることを確認し、さらに本手法のテスト問題に関する有効性を示す。第4章では応用問題として宇宙探査機の機体構成と軌道計画の統合領域最適化を実行する。無人宇宙探査機と有人宇宙探査機の2つのパターンについて計算を行う。無人宇宙探査機では単一目的最適化とし、設計変数は軌道計画要素のみとする。無人探査機の問題では有人探査機の最適化問題の前段階と位置づけて、最適化対象の解析モデルが妥当であることを検証する。次に宇宙探査機の最適化問題を無人から有人に拡張し、設計変数要素も機体構成と軌道計画の複合領域から決定する。有人宇宙探査機の最適化問題は多目的最適化問題と設定し、本提案手法を多目的解法用のアルゴリズムに変更する。応用問題として多目的最適化における有人宇宙探査機問題へ本提案手法を適用し、その結果から本手法の有用性を示す。

以上が本論文の目的となる。

第2章 提案手法

2.1 概要

この章では本論文のメインテーマであるメタヒューリスティックな最適化アルゴリズムを提案する。提案手法のアルゴリズムと特徴を述べた後に定式化と最適化計算上で使用した計算テクニックを説明している。本提案手法は銀河の形成と台風が渦を巻いている状態を模擬し、それを定式化・アルゴリズム化したものである。最後に比較対象としている他のヒューリスティック手法について述べている。比較とした手法は粒子群最適化手法（PSO）と遺伝的アルゴリズム（GA）を用いた。提案手法の着想は銀河と台風の渦の形が似ているというところから出発している。図 2.1 と図 2.2 に渦の様子を示す。本提案手法は銀河や天体の形成過程と台風の渦形成を模擬した最適化手法である。着想を最適化手法に変換する際に、銀河形成は重力による一点への収束を考え、台風の渦形成はコリオリ力と地球のように閉じた空間を考えている。手法特有のアルゴリズムは4つあり、次節以降に詳細を述べている。簡潔に提案手法を説明すると銀河や台風の渦の中心を求める最適化手法である。



図 2.1 銀河の渦形成 [3]

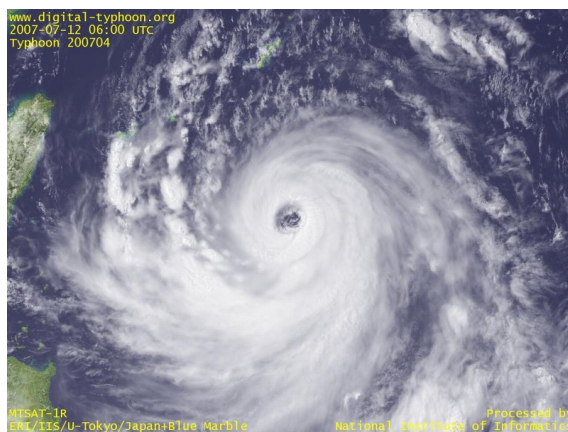


図 2.2 台風の渦 [4]

2.2 銀河形成過程を模擬

この節では提案手法の4つのポイントのうち2つについて説明する。銀河や天体の形成過程を模擬したものである。1つ目のポイントでは重力によって粒子が収束する現象を再現したものである。2つ目のポイントは巨大質量天体からジェットを噴出する現象を再現したものである。

2.2.1 重力収縮 一局所収束アルゴリズム

天体の原始状態では、分子や粒子がガス状態で空間に漂っている。この粒子たちが衝突を繰り返し、次第に集合体として集まっていく。このときに重力の偏りが生じることでさらに周囲の粒子を引き寄せる。実際の天体現象ではガス状態から長い年月を経て球体の状態へと進化を遂げる。この現象は重力によって粒子同士が引き合うことで集合体を形成するといえる。この自然現象を最適化手法へと応用することを本研究では考える。重力の方程式によって一点へと収束する様子を最適化手法で表す。着想を実現するために最適化手法においては多点探索とする。この重力収束と提案する最適化手法のアナロジーを箇条書きすると、ガス状態の粒子＝多点探索点、粒子の質量＝最適化問題設定上の評価値（目的関数値）となる。最適化上での解空間に多点の探索点を散布する。探索点上の評価値を計算しその値を質量として考える。その質量値を用いて重力の方程式を各探索点間で計算することで、多点探索点は重力の強いところへ集まっていく。つまり、最適化計算では評価値の高い点へ収束することになる。これが本研究で提案する最適化手法での局所収束ルーチンとなる。この重力収束が提案手法の柱となる考え方である。以上の過程を別の表現にすると、評価値空間を自然現象における重力ポテンシャル空間と捉えることでその重力場に多数の探索点を撒きその点の質量を計測すると考えることも出来る。重力場に探索点のサンプルを複数個に限定して撒くイメージである。

図2.3に重力ポテンシャル空間に探索点を分布させたイメージ図を載せる。この探索点同士が重力によって引き合う。最小化問題とするとこの図における深さは評価値と同等となる。最適化計算上ではこの評価値を質量とみなすことで重力を計算することができる。つまり、深い位置にいる探索点は強い重力で周りの探索点を引っ張ることになる。繰り返し計算で時間を経過させると探索点は深い位置に集まり、最適化計算上では最良解に探索点群が収束することになる。この重力アルゴリズムが最適化計算の局所収束アルゴリズムとなり探索点群を最良解へと収束させる。

ある探索点にかかる重力 F_g を示す。

$$F_g = \sum_k \left(G \frac{mM_k}{r_k^2} \right) \quad (2.1)$$

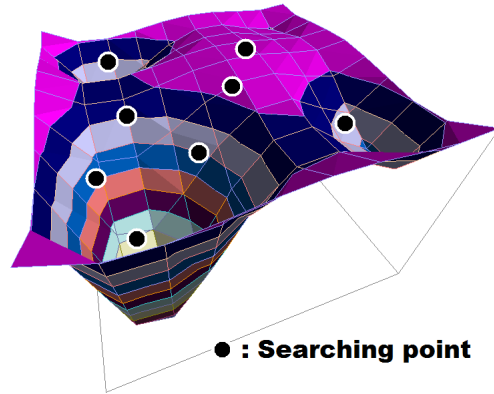


図 2.3 重力場を模擬した解空間

重力定数 G は設計者が決める定数である。問題設定により決める値は異なる。大まかな設定指針は別節に述べている。 m は重力を計算する中心探索点の評価値である。 M は各探索点の評価値である。 r は中心探索点と各探索点との距離である。添え字 k は中心探索点との重力を計算するある探索点を示す。中心探索点と周りにある各探索点との重力の総和を計算することで統合した重力を表している。この式を用いて探索点の運動方程式を求めている。詳細は別節の定式化において述べている。

2.2.2 宇宙ジェット 一局所解脱出アルゴリズム

天体现象における2つ目の自然現象と最適化手法のアナロジーとして採用したルーチンが宇宙ジェットの物理現象である。天体の成長過程において質量が巨大になると、内部の圧力や磁場の影響により天体から粒子を噴出する物理現象が実際に存在する。図2.4に宇宙ジェットのイメージ図を載せる。赤黄色く渦を巻いている中心に黒い球体がある。そこから白い物質が噴出している様子がわかる。これが宇宙ジェットの現象である。この自然現象を最適化計算上で模擬する。この天体の物理現象を最適化手法で再現すると、多点探索点がある範囲に収束した場合において、収束した探索点群から2つを選択しランダム方向に探索点を噴出させる。噴出に関しては探索点を問題設定によって決められた速度制限があり、その速度制限限界の速度で乱数で決められたランダム方向に移動させることである。

具体的に計算方法を述べる。後述する宇宙機の複合領域最適化問題へ適用した場合は全探索点の1/3の個体数がある領域へ収束した場合にジェットアルゴリズムを発動させる。ある領域とは最適化問題によって設計者が設定するものであり、その範囲を R_c とすると以下の

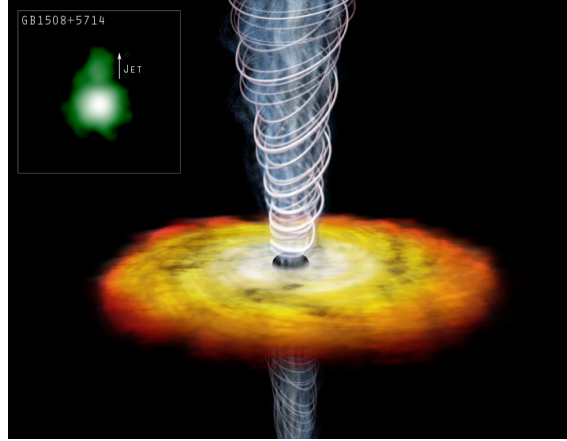


図 2.4 宇宙ジェット [5]

ようになる。

$$R_c = \frac{X_{lim,max} - X_{lim,min}}{C_{rc}} \quad (2.2)$$

分母の C_{rc} は設計者が決めるものであり、解空間に対してどのくらい縮小した範囲を採用するかというものである。参考に第 4 章の宇宙機問題での数値を述べると $R_c = 100$ とした。宇宙ジェットの噴出に関して述べると、宇宙機問題では個体数は 50 個と設定しているので 17 個以上が集まった時にジェットを噴出する。集合群から 2 個体をランダムに選出する。そして、その個体に速度 V を強制的に与える。速度 V は乱数によって与えられるものであり、局所的に収束してしまった個体群を真の大域的な最適解に移行する可能性を持っている。乱数 Ran は 0 から 1 の間を動く一様乱数である。

$$\begin{aligned} V_1 &= (X_{lim,max} - X_{lim,min}) Ran_1 + X_{lim,min} \\ V_2 &= (X_{lim,max} - X_{lim,min}) Ran_2 + X_{lim,min} \end{aligned} \quad (2.3)$$

この速度 V を用いて現在の位置 x_{old} から新しい位置 x_{new} へ探索点を移動させる。

$$\begin{aligned} x_{new,1} &= x_{old,1} + V_1 \\ x_{new,2} &= x_{old,2} + V_2 \end{aligned} \quad (2.4)$$

これによって新しい位置へ個体を移動させる事ができる。ただし、移動先が実行不可能領域の場合は実行可能領域に移動するまで乱数を変化させて計算をする。

2.3 台風の回転を模擬

この節では提案手法の 4 つのポイントのうち後半の 2 つについて説明する。台風の渦を模擬したもので、最適化計算上で再現するためのアルゴリズムを述べる。1 つ目は地球が回転

していることを再現したものであり、解空間を回転させてコリオリ力で多点探索点を強制的に移動させることで渦を巻くように探索させる。2つ目は地球が球形で端がないことを模擬したもので解空間をトーラス面とすることで探索点の移動の自由度を上げる。

2.3.1 コリオリ力 一大域解発見アルゴリズムー

台風の渦を再現するためにコリオリ力を導入する。この節では3番目のポイントであるコリオリ力の再現について述べる。解空間を回転させることで探索点が移動する際に回転半径方向に力を受ける。このような座標空間を回転する方法で渦を実現する。探索点の移動のイメージを図2.5に示す。地球が自転しているイメージを実現したものであり、設計変数空間を評価値軸を基準に回転させることで最適化アルゴリズムとする。各探索点の重心に向かうように重力が働くので、重心を中心とした円を描くように探索点が移動する。中心に向かう速度ベクトルを持つ探索点はコリオリ力を受けて渦を巻く。急速に中心に収束するのではなく、わざと大回りする様に移動する。これによって解空間の大域的最適解に到達する可能性を上げている。回転の量を表すパラメータである角速度パラメータ ω を設定することで回転量を調整することができる。パラメータ ω は設計者が決定するものである。最適化問題に依存するパラメータなので設定指針は別節において述べている。次に座標の回転方法につ

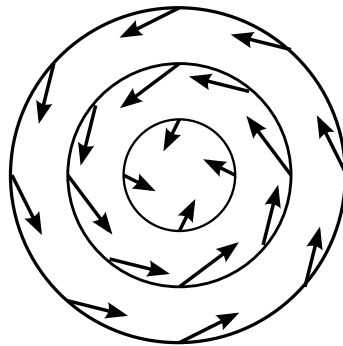


図 2.5 コリオリ力

いて詳細を述べる。まず、図2.6に示すように設計変数空間を定義する。上に向かう軸が評価値（目的関数値）軸である。横方向に設計変数軸を置く。この評価関数値軸と設計変数値軸によって定義された空間内を探索点が移動することになる。探索点の位置が設計変数を表し、評価値がその点の質を表す。探索点は重力の影響によって移動する。この時の速度に対してコリオリ力が掛かることになる。図2.6では評価値軸を回転量 ω で時計回りに回転させた場合の図である。この回転空間の中を探索点が回転の半径方向に速度成分を持つと図の方向にコリオリ力を受ける。設計変数が3つ以上の場合でも軸を置く概念は同様である。図示はできないが4次元空間以上でも、評価値軸を中心に回転させる構図になる。どの軸を回

転させるかによって探索点を受ける影響は異なるが本論文では評価値軸のみを回転させることにする。設計変数が2つの場合、解空間の表現は評価値軸と変数軸2つで合計3次元表示

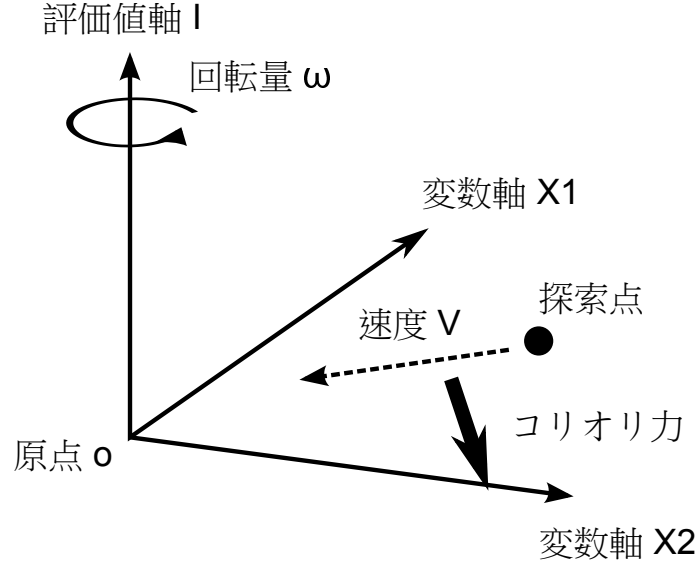


図 2.6 評価値軸と変数軸

になる。この最適化計算ではコリオリ力を求めるために外積計算を行う。解空間が3次元のときは外積計算を行うことは容易であるが、4次元以上になると難しくなる。そこで変数の数によって以下のように計算方法を変えている。

設計変数が2つの場合

設計変数が2つの場合はコリオリ力を以下のように定義する。

$$\mathbf{F}_c = -2m\boldsymbol{\omega} \times \mathbf{v} \quad (2.5)$$

3次元のベクトル成分表示をする。評価値軸を回転させるときの角速度 $\boldsymbol{\omega}$ と速度 \mathbf{v} は

$$\boldsymbol{\omega} = (0, 0, \omega)^T, \mathbf{v} = (v_x, v_y, 0)^T \quad (2.6)$$

となる。次にコリオリ力 \mathbf{F}_c の計算は以下の計算式によって行う。

$$\begin{aligned} \mathbf{F}_c &= 2\boldsymbol{\omega} \times \mathbf{v} \\ &= 2(0, 0, \omega)^T \times (v_x, v_y, 0)^T \\ &= (-2\omega v_y, \omega v_x, 0)^T \end{aligned} \quad (2.7)$$

また、変数が1つの場合には外積計算を行うことができないので、コリオリ力を考慮しない最適化計算を行う。

設計変数が3つ以上の場合

設計変数が3つ以上の場合にはコリオリ力を座標変換として次のように定義する。変数が3つ以上になると解空間の次元が4次元以上になるので外積計算が難しくなる。正確にはテンソルとして計算結果が出てくるので列ベクトルとの足し算が定義できなくなるので、この最適化計算内では計算結果が意味を成さない。なのでコリオリ力を求めるのではなく、別の方法で計算個体を移動させる。直交座標系から円柱座標系へ座標変換し、評価値軸を ω だけ回転させることを考える。これは速度がゼロの個体も移動してしまうためコリオリ力を受ける結果とは異なる。しかし、この最適化計算のメインアルゴリズムである解空間を周りながら探索するという意図には沿っている。また、4次元以上の回転座標を求めることも難しいため座標変換方式を採用した。以下に直交-円柱座標変換の式を示す。

直交座標系でのある個体の解空間内の座標は

$$\mathbf{X} = (I, x_1, x_2, \dots, x_n)^T \quad (2.8)$$

となる。 I は評価値である。これを円柱座標系にすると

$$\begin{bmatrix} I \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} I \\ r \cos \theta_1 \\ r \sin \theta_1 \cos \theta_2 \\ \vdots \\ r \sin \theta_1 \cdots \sin \theta_{n-2} \cos \theta_{n-1} \\ r \sin \theta_1 \cdots \sin \theta_{n-2} \sin \theta_{n-1} \end{bmatrix} \quad (2.9)$$

$$r = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

となる。次に評価値軸に角速度 ω で回転した後の座標 \mathbf{X}^* は

$$\begin{bmatrix} I \\ x_1^* \\ x_2^* \\ \vdots \\ x_{n-1}^* \\ x_n^* \end{bmatrix} = \begin{bmatrix} I \\ r \cos (\theta_1 + \omega) \\ r \sin (\theta_1 + \omega) \cos \theta_2 \\ \vdots \\ r \sin (\theta_1 + \omega) \sin \theta_2 \cdots \sin \theta_{n-2} \cos \theta_{n-1} \\ r \sin (\theta_1 + \omega) \sin \theta_2 \cdots \sin \theta_{n-2} \sin \theta_{n-1} \end{bmatrix} \quad (2.10)$$

$$r = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

となる。評価値軸の周りの回転を $\theta_1 \rightarrow \theta_1 + \omega$ と表現した。

2.3.2 トーラス面構造 ー探索点移動幅向上ー

地球が球体であることを再現するために解空間を閉じた空間とする。地球は端がなく地表面上を移動する際は地球の裏側まで自由に移動することが出来る。地球が球形なイメージを実現しており、設計変数空間を制約条件の上下限で単連結させることで最適化アルゴリズム上で再現する。従来の解空間であると制約条件により空間に上限下限が存在していたのでそれが壁の役割を果たしていた。上述のコリオリ力を再現する場合にはこの壁が障壁となり探索点が自由に移動できなくなる。通常の工学的な最適化問題では設計変数に制約がつくので空間上では端が存在する。これを解消するために設計変数空間（解空間）を制約条件の上下限で単純に連結させる。単純に端を連結するだけなので構造上は球面ではなくトーラス面となるが、探索点の移動とコリオリ力の再現には支障はない。図 2.7 はトーラス面のイメージ

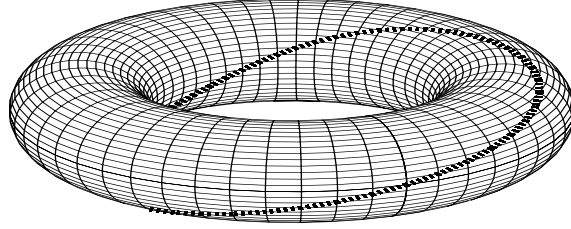


図 2.7 トーラス面

である。直交座標系で表された設計変数空間の端をつなぐことでドーナツ状の空間を再現する。計算プログラム上は探索点が制約上限に達した場合は下限に移動するというものである。探索点が制約条件という壁にぶつかることなく解空間を移動することができる。このことにより、探索点の最適解の発見確率が上がることを期待できる。設計変数の上限下限の制約条件のみならば単純に空間の端を連結するだけなので、最適化プログラム上のアルゴリズム構築も容易である。空間を横断するような制約条件の時は工夫が必要である。実行不可能領域にあたる部分の境界をつなげることで、実行不可能領域に侵入不可能な状態にすることが考えられる。これが数値プログラム構築上で困難な場合は、予測される実行不可能領域を跳躍可能な状態にするような各探索点の速度上限を設定することも考慮することもできる。プログラム上の式を示す。設計変数の各構成要素について以下の式を実行する。変数 x が速度 v によって、制約の上限 $X_{lim,max}$ または下限 $X_{lim,min}$ を超えた場合についてのみ以下の式を用いて変数位置を x_{old} から x_{new} へ移動させる。探索点は設計変数空間内をループする様に移動探索するようになる。

$$x_{new} = \begin{cases} x_{old} - (X_{lim,max} - X_{lim,min}) & (x_{old} > X_{lim,max}) \\ x_{old} + (X_{lim,max} - X_{lim,min}) & (x_{old} < X_{lim,min}) \end{cases} \quad (2.11)$$

2.4 定式化

上述の4つのポイントを定式化する。まずは探索点の運動方程式を定義し、数値計算するために離散表現に変換する。そして、コリオリ力を再現する際の座標変換について述べる。最適化の計算進行において重力とコリオリ力を考慮した運動方程式を解くことになる。以下のセクションで使う物理現象と最適化計算のアナロジーから使用する文字を箇条書きで示す。

- 質量 $M = \text{評価値 } I$
- 空間位置 $x = \text{設計変数 } x$
- 移動速度 v
- 重力定数 G (任意の値)
- 回転速度 ω (任意の値)

2.4.1 運動方程式と離散化

最適化計算をする上での個体の移動に関する運動方程式を示していく。個体にかかる力は重力 \mathbf{F}_g とコリオリ力 \mathbf{F}_c の2つである。

$$\begin{aligned} m \frac{d\mathbf{v}}{dt} &= \mathbf{F}_g + \mathbf{F}_c \\ &= \sum \left(G \frac{Mm}{r^2} \right) - 2m\boldsymbol{\omega} \times \mathbf{v} \end{aligned} \quad (2.12)$$

各文字を説明すると、運動方程式を考えている個体の質量 m 、ある別の個体の質量 M 、重力定数 G 、解空間の回転速度 $\boldsymbol{\omega}$ 、質点の速度 \mathbf{v} 、距離 \mathbf{r} は基準とする個体とある別の個体との幾何学的な距離である。分母のベクトル表現を分子側に移動すると次のように書ける。

$$\frac{d\mathbf{v}}{dt} = G \sum_k \left(\frac{M_k}{|r_k|^3} \mathbf{r} \right) - 2\boldsymbol{\omega} \times \mathbf{v} \quad (2.13)$$

質量は個体の評価値と同一として扱う。

次にプログラムで計算するように微分を差分に置き換える。ただし、最適化計算上の計算時間幅は $\Delta t = 1$ として考える。

$$\begin{aligned} \frac{d\mathbf{v}}{dt} &= \frac{\mathbf{v}_n - \mathbf{v}_{n-1}}{\Delta t} \\ &= \mathbf{v}_n - \mathbf{v}_{n-1} \end{aligned} \quad (2.14)$$

添字の n は計算ステップ数である。同様にして個体の位置座標 \mathbf{x} は

$$\mathbf{v}_n = \frac{d\mathbf{x}}{dt} = \mathbf{x}_{n+1} - \mathbf{x}_n \quad (2.15)$$

となる。以上をまとめると以下の方程式を得る。

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{x}_n + \mathbf{v}_n \\ \mathbf{v}_n &= \mathbf{v}_{n-1} + G \sum_k \left(\frac{M_k}{|r_k|^3} \mathbf{r} \right) - 2\boldsymbol{\omega} \times \mathbf{v}_{n-1} \end{aligned} \quad (2.16)$$

この方程式を用いて繰り返し計算を行う。

2.4.2 最適化手法としての利点

ここで最適化手法としてのメリットを述べる。以上までに手法の特徴を述べてきたが、本手法は大域探索と近傍探索を同時にかつ効率的に行うことが出来る最適化手法である。近傍探索については重力の方程式により各探索点が探索点群の質量中心（重心）に向かって探索が進む。大域探索はコリオリ力を導入することで実現しており、探索点をわざと回り道させることで広域探索を行う。これらから、大域的最適解から遠方の領域では重力の影響が薄いので収束が遅くなり、広くサーチすることになる。これは大域探索になる。ローカルな局所解への収束を防ぐように、最適解から遠い場所にある探索点は重力の影響が薄くゆっくり探索を進める。コリオリ力があるために大きく渦を作るのでローカルトラップに捕まっても脱出できる可能性を持っている。解空間の回転量パラメータ ω で回転させたほうが解空間を広く探索するので、最適解を見つけやすい。そして大域的最適解から近傍の領域では重力の影響が強いのので収束が早くなる。最適解に急速に集まることになり、最適解付近では局所探索をすることになる。重力の影響が強くなるためである。同一の空間において大域的探索と局所的探索の良い点を同時に満足していると言える。

2.5 最適化計算上のテクニック

2.5.1 計算テクニック

提案手法で示した最適化計算を円滑に進めるために以下のような制限を加えている。計算上のテクニックを箇条書きで示す。

- 初期個体の計算状態

解空間での初期位置 x は乱数によるランダム配置とする。初期速度 v に関してもランダム方向とした。

- 各計算回の最良値を保存

ヒューリスティック手法でよく採用されている最良解保存のアルゴリズムを使う。最適化計算を迅速に行うためである。

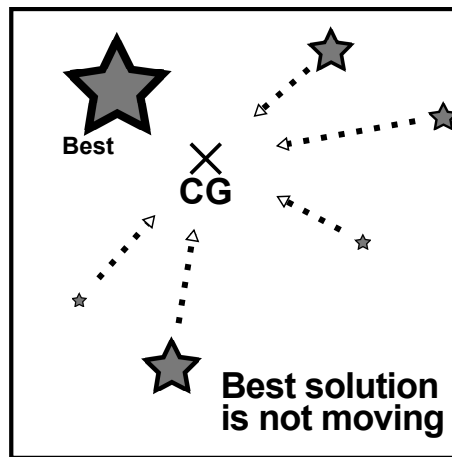


図 2.8 重心位置と最良解保存

- 全探索点の重心位置を計算

計算の進行は重心位置に引き込まれるように探索が進む。最適解付近で渦を巻くように回転してしまう場合でも重心位置の履歴を見ることで渦の中心を知ることが出来る。また、二連星のような状態になった場合も重心を知ることで回転の中心が分かる。探索回の最良解と重心位置の解を比較することで真に良い解を算出することが出来る。

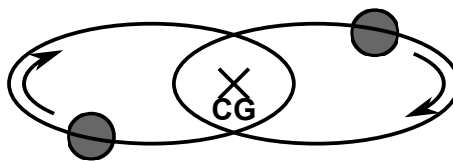


図 2.9 二連星

- 最大速度制限

探索点同士が近づきすぎると重力が発散し最適化計算上は不都合になる。探索点の移動が計算不可能になるので、これを解決するために探索点の速度に制限をつけることで安定した移動を行う。

- 近傍衝突仮定

ある範囲内に個体同士が近づくとその個体同士は衝突したと仮定して、同じ位置、

同じ速度を取るものとする。ある範囲とは問題によって変更することが妥当であるが、本論文中は式 2.2 によって与えられた範囲とする。

2.5.2 計算パラメータの指標（オーダー値）

最適化手法の調整パラメーターは角速度 ω と重力定数 G の 2 つである。角速度については、1 回の最適化計算で解空間を 1 周するようなパラメーター調整をすることが多い。具体的に式を使って表すと、

$$\omega = \frac{2\pi}{N_{cal}} = \frac{2\pi}{1000} \quad (2.17)$$

ただし、 N_{cal} は最適化計算内の解析ルーチン呼び出し回数である。遺伝的アルゴリズムにおける“個体数×世代数＝計算回数”といえる。ここでは $N_{cal} = 1000$ としたので 1000 回の計算でちょうど回転軸を 1 周することになる。

重力定数 G の大まかな決め方を以下に示す。

ある個体にかかる力 F は

$$F = mG \sum_n \frac{M_n}{r_n^2} \quad (2.18)$$

基準となる個体の質量 m （評価値）、別の個体との距離 r 、別の個体の質量 M （評価値）である。

運動方程式を考えると

$$\frac{dv}{dt} = G \sum_n \frac{M_n}{r_n^2} \quad (2.19)$$

個体の移動速度を考えて右辺を 1 のオーダーにする。 G を規定化し、倍率係数 n をかけると

$$G = \frac{r_{max}^2}{M_{max}} \times n \quad (2.20)$$

ここで、 M_{max} は初期個体群の中の最大質量（評価値）である。 r_{max} は解空間距離の平均をとって

$$r_{max} = \frac{\sqrt{\sum (X_{max,n} - X_{min,n})^2}}{2} \quad (2.21)$$

$X_{max,n}$, $X_{min,n}$ は設計変数の最大値、最小値である。これは制約条件により与えられている。次に倍率係数 n は

$$n = \frac{r_{max}}{n_{Grid}} \times \frac{1}{N_{Ind}} \quad (2.22)$$

と設定する。 n_{Grid} は解空間の移動距離を何分の一に制限するかというものである。 N_{Ind} は個体数である。よって G は

$$G = \frac{\{\sum (X_{max,n} - X_{min,n})^2\}^{\frac{3}{2}}}{M_{max}} \times \frac{1}{8} \times \frac{1}{n_{Grid}} \times \frac{1}{N_{Ind}} \quad (2.23)$$

となる。具体的に計算すると、テスト問題での値を使い以下のように計算することが出来る。

$$\begin{aligned}
 G &= \frac{\{\sum (X_{max,n} - X_{min,n})^2\}^{\frac{3}{2}}}{M_{max}} \times \frac{1}{8} \times \frac{1}{n_{Grid}} \times \frac{1}{N_{Ind}} \\
 &= \frac{\{(10-1)^2 + (10-1)^2\}^{\frac{3}{2}}}{6191.5} \times \frac{1}{8} \times \frac{1}{100} \times \frac{1}{25} \\
 &= 1.67 \times 10^{-5}
 \end{aligned} \tag{2.24}$$

となる。最適化計算上の定数なので、単位はない。物理学上の万有引力定数は $G = 6.67 \times 10^{-11} [\text{m}^3 \text{s}^{-2} \text{kg}^{-1}]$ である。

2.5.3 ペナルティ関数法

ペナルティ関数は目的関数、制約条件、設計変数を有する最適化問題のための解法の一つである。制約条件 $g(x)$ が存在する問題設定において、制約条件の許容領域外に設計変数 x が移動してしまう場合に非許容量 $\Phi(x, g(x))$ を算出する。非許容量 $\Phi(x, g(x))$ を目的関数 $f(x)$ に足すことで評価を改悪し、制約最適化問題を無制約最適化問題に変換することが出来る。たとえば評価値 I を最小化する問題設定のときは

$$I = f(x) + |\Phi(x, g(x))| \tag{2.25}$$

のように目的関数値 $f(x)$ を非許容量 Φ が評価を改悪し、最適化手法のアルゴリズムによって制約条件内に設計変数が移動するように作用する [26]。

2.5.4 多目的最適化

ここでは、多目的最適化に関連した基本的な概念の説明を行う。一般の k 個の目的をもつ多目的最小化問題について考える。 $f(x)$ は目的ベクトル、 x は設計変数ベクトルで構成される。最大化されるべき目的関数が存在する場合は、マイナスの符号を付けることで最小化の形式に変換することができるので、多目的最小化問題について最適解の概念を考える。もし、すべての目的関数を最小にする解が存在すれば、その解は多目的最小化問題の完全最適解である。しかし、通常は複数の目的関数の間にトレードオフの関係が存在するために、完全最適解が存在することは一般的ではない。そのため、多目的最適化の分野では、個々の解の間での優越関係に基づく解の概念が定義される。多目的最小化問題の実行可能解を x と y とし、どの目的関数に関しても解 x は解 y よりも大きくなく、少なくとも一つの目的関数に関して解 x は解 y より厳密に小さい場合、解 x は解 y を優越するという。この定義は、我々の直感に一致しており、この条件が満たされる場合は、明らかに解 x を選択する。したがっ

て、解 x は解 y よりも良い解であるといえる。しかし、必ずしもすべての解が優越関係で比較可能ではない。比較可能ではない多数の解が多目的最適化問題には存在する。多目的最適化では、すべての目的関数を同時に最適化する完全最適解が一般には存在しないので、ほかの解に優越されないという基準を用いて最適解が定義される。このような解はパレート最適解と呼ばれ、最終的に意思決定者により選択される選択解の候補になる。多目的最適化では互いに比較できない解が多数存在するので、多目的最適化問題は多数のパレート最適解を持つことになる。意思決定者はパレート最適解から最終的な優良解を選択することになる。したがって、最終的な選択解の候補として、多様なパレート最適解を意思決定者に提示する事が重要である。多目的のヒューリスティック最適化の分野では、確率的な多点探索という進化計算の特徴を有効に活用することにより、1 回の実行で多様なパレート最適解を発見する事が重要となる。

パレートランキング法

第 4 章の有人宇宙探査機の最適化問題では多目的最適化を行っている。この際に使用した目的関数の評価方法の一つにパレートランキング法がある。多目的最適化におけるヒューリスティック手法では、個体の評価は優越関係に基づき行われる。しかし、優越関係を用いて直接的に 2 個体の比較を行うと、多くの場合で比較不可能となる。たとえば、解空間の点 A と点 B を比較すると、明らかに点 B のほうがパレート最適解集合に近い良好な解であるとすると、優越関係では比較不可能となる。そのため、優越関係を用いて個々の個体にランクを割り当てるランキングという操作が行われることが多い。たとえば、多目的遺伝的アルゴリズムでは、現在の個体群の中で自分自身を優越する個体の数に 1 を加えたものがランクとなる。このランキング方法では、自分自身を優越する個体の数が多いほど悪いランクになる。ランキング操作を用いることで、パレート最適解集合に近い良好な個体に高いランクを与えることができる。したがって、個々の探索点に割り当てられたランクに基づき最適化計算進行を行うことで、良好な探索点を選択して次の探索点に移動するという進化型最適化の基本的な枠組みの実装が可能になる。優越関係に基づくランキングは、多目的最適化においてはほとんどのヒューリスティックなアルゴリズムで用いられている。

エリート保存戦略

パレート最適解集合への収束性を向上させるためには、多目的進化の途中で発見された良好な個体を保存しておくエリート保存戦略が重要である。1990 年に Zitzler と Thiele で明確

に示されたように、エリート保存戦略により、ヒューリスティック手法の探索性能が大きく向上する。これまでの最適化計算進行では、良い個体が存在したとしても計算過程で無条件に置き換えられることがある。このとき新しい解が悪くなる可能性もあり、その場合には、これ以降は良い個体が生成される見込みが少なくなる。これを防ぐために、これまでの計算進行とともにエリート保存と呼ばれる操作が付加されることがある。そのため、最近の研究では、エリート保存戦略が用いられることが多い。現在の探索点群の中から良好な個体を選択することによりエリート保存戦略を実現する。本論文ではエリート保存は探索点群中の良い解を1つエリートとして保存しておく。最適化計算進行でたまたまできの悪い解ばかりを生成しても、エリート保存により良い個体が保存されているので、最終的に良い解を得ることが期待できる。

シェアリング関数法

多様性の維持は、単一目的最適化の場合と同様に、密集している個体の適応度を相対的に小さくし、密集していない個体の適応度を相対的に大きくすることで実現される。たとえば、密集していない領域に存在する個体に対する適応度を相対的に大きくすることで多様性の維持が行われる。多様性の維持機能は、優越関係に基づくランキングと同様に、1990年代前半から現在までほとんどの多目的最適化アルゴリズムで用いられている。なお、多目的最適化計算では単一目的最適化とは異なり、目的関数空間で個体の密集度が測られることが多い。ニッチングカウントと呼ばれる値を求め評価関数値に加えることで評価を改悪する方法をとる。ニッチカウントは解空間における個体群同士の混雑度を示すものである。以下にニッチカウントの式を示す。

まずは、目的関数 f 上の空間の広さを定義する。以下の式内において、 $d_{i,j}$ は目的関数空間での個体間の距離を表し、 n は個体の番号、 N は全個体数、 i, j は目的関数 f の番号、 J は目的関数の総数である。

$$d_{i,j} = \sqrt{\sum_{n=1}^N \left(\frac{f_{i,n} - f_{j,n}}{f_{MAX,n} - f_{MIN,n}} \right)^2} \quad (2.26)$$

i 番目の目的関数のニッチカウント Nc_i は

$$Nc_i = 1 + \sum_{j=1}^J \left(1 - \frac{d_{i,j}N}{J} \right) \quad (2.27)$$

ニッチカウント Nc_i を目的関数と足し合わせることで、個体間同士の混雑度を含めた評価を行うことが出来る。

2.6 フローチャートとアルゴリズム

この節では実際の数値計算上の実行手順について説明する。以下に手順を箇条書きで示す。また、フローチャートも同時に示す。

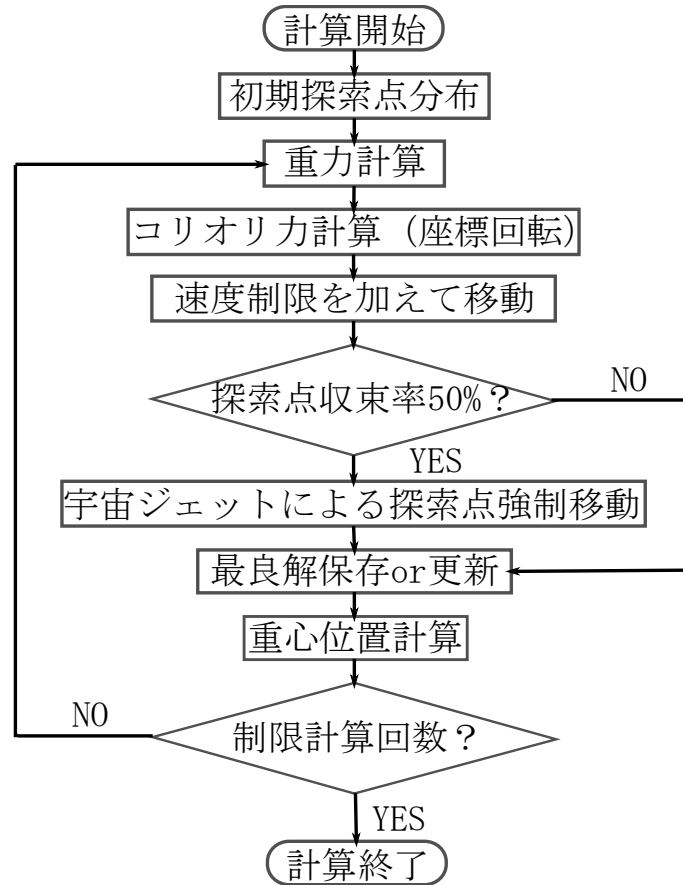


図 2.10 提案手法のフローチャート

提案手法の計算手順を以下の様に行う。

1. 本手法は多点探索であり、初期探索点を解空間に分布させる。
2. 探索点は解空間の位置 x から評価値 I が計算される。
3. 最適化計算上で評価値 I を重力計算用の質量 M とみなす。
4. 質量 M と探索点同士の距離 r が分かるのでお互いにかかる重力を計算する。
5. 運動方程式から次の点へ移動するための速度 v を求める。
6. コリオリ力を発生させるために解空間内の評価値軸を中心に角速度 ω で回す。

7. 座標変換でコリオリ力を求めた後に増減速分を速度 v に合成し真速度 v^* を求める。
8. 真速度 v^* を使って新しい座標位置 x_{new} を求める。
9. ステップ2へ。ただし収束条件を満たしていれば計算終了とする。

2.7 他のヒューリスティック手法

2.7.1 Particle Swarm Optimization

Particle Swarm Optimization(PSO) は、単純化された社会モデルのシミュレーションを通じて 1995 年に J.Kennedy と R.Eberhart により開発された発見的最適化手法（メタヒューリスティクス）の一つである。「Particle Swarm」は、位置や速度の概念をもつ探索点を表現した「Particle（微小粒子）」と、人工生命の研究における Swarm Intelligence で用いられている「Swarm(群れ)」とを合わせて作られた言葉である。PSO の基本的な概念は、鳥の群れが餌を探す行動研究によって導かれた「情報を群れ全体で共有している」という仮定に基づいている。すなわち、群れを構成する個体が独立に行動するのではなく、群れを構成する個体の個別情報と、群全体の共通情報を組合せ、一定の規則に従って行動するという概念である。最適化アルゴリズムとしての PSO の特徴は、探索点が複数個存在する多点探索アルゴリズムであることと多点間で互いに最良解に関する情報を共有し、それに基づいて解空間を探索する、ことがあげられる。また非常に簡単なアルゴリズムで構成され、基本的な算術演算しか用いてない点も特徴としてあげられる [14][15][16][23]。PSO は、もともとは二次元空間上で群れの動きをシミュレーションする過程から開発された手法であるが、最適化手法としての PSO は多次元空間に拡張することができる。 n 次元空間における一つの Particle（探索点）の位置は n 次元ベクトルによって $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{in})^T$ と表される。ここで i は Particle の番号であり、 x_{ij} は i 番目の Particle の位置ベクトルの j 次元成分を意味する。また、各 Particle は位置ベクトルの他に移動ベクトル $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{ij}, \dots, v_{in})^T$ を持っている。

さらに、各 Particle はそれまでの探索で発見したそれぞれの最良解 $\mathbf{pbest}_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{ij}, \dots, pbest_{in})^T$ と、その評価値 $f(\mathbf{pbest}_i)$ を記憶している。群としては、すべての Particle がこれまでの探索で発見した最良解 $\mathbf{gbest}_i = (gbest_{i1}, gbest_{i2}, \dots, gbest_{ij}, \dots, gbest_{in})^T$ と、その評価値 $f(\mathbf{gbest})$ を記憶している。

各 Particle は現在の位置 \mathbf{x}_i^k から、それぞれの最良解へ向かうベクトル $(\mathbf{pbest}_i^k - \mathbf{x}_i^k)$ 、群全体の最良解へ向かうベクトル $(\mathbf{gbest}^k - \mathbf{x}_i^k)$ 、および前回の移動ベクトル \mathbf{v}_i^k の重み付き線形結合として新たな移動ベクトル \mathbf{v}_i^{k+1} を生成し、次の位置 \mathbf{x}_i^{k+1} に移動する。 $k+1$ 回目

の移動における、 j 番目の直交座標に射影される、Particle i の移動ベクトル v_{ij}^{k+1} は次式で与えられている。

$$v_{ij}^{k+1} = w \cdot v_{ij}^k + c_1 \cdot rand_1()_{ij} \cdot (pbest_{ij}^k - x_{ij}^k) + c_2 \cdot rand_2()_{ij} \cdot (gbest_j^k - x_{ij}^k) \quad (2.28)$$

ただし、 $rand_1()_{ij}$ 、 $rand_2()_{ij}$ は、0 から 1 の間に分布する一様乱数、 w, c_1, c_2 はそれぞれの項に対する重みパラメータ、 k は反復回数である。PSO において移動ベクトル \mathbf{v}_i は慣例的に速度と呼ばれる。本節でもこれに従い、 \mathbf{v}_i を速度と呼ぶことにする。このとき Particle の移動は次式で与えられる。

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (2.29)$$

また、PSO では Particle の発散を防ぐために、Particle の速度上限 (V_{max}) を制約として課すことがある。

次節に一般的な PSO のアルゴリズムを示す。

2.7.2 PSO のアルゴリズム

- STEP1 初期化

各 Particle の初期位置 \mathbf{x}_i^0 と初期速度 \mathbf{v}_i^0 を与える。初期位置 \mathbf{x}_i^0 は実行可能領域内にランダムに、初期速度 \mathbf{v}_i^0 はランダムに与える。

$$pbest_i^0 = \mathbf{x}_i^0, \quad (i = 1, 2, \dots, m)$$

$$gbest^0 = pbest_{ig}^0$$

とおく。

- STEP2 \mathbf{v}_i 、 \mathbf{x}_i の更新

$$v_{ij}^{k+1} = w \cdot v_{ij}^k + c_1 \cdot rand_1()_{ij} \cdot (pbest_{ij}^k - x_{ij}^k) + c_2 \cdot rand_2()_{ij} \cdot (gbest_j^k - x_{ij}^k)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1}, \quad (i = 1, 2, \dots, m; j = 1, 2, \dots, n)$$

を計算する。

- STEP3 $pbest_i$ と $gbest$ の更新

各探索点において $f(\mathbf{x}_i^{k+1}) < f(pbest_i^k)$ が成り立つとき

$$pbest_i^{k+1} = \mathbf{x}_i^{k+1}$$

$$gbest^{k+1} = pbest_{ig}^{k+1}$$

とおく。

- STEP4 終了判定

$k = T_{max}$ ならば終了。さもなければ $k = k + 1$ として STEP2 へ行く。

2.7.3 遺伝的アルゴリズム Genetic algorithm

遺伝的アルゴリズムは 1960 年代に Holland によって導入された生物の進化過程を模擬するアルゴリズムであり、その後 1980 年代ごろから活発に研究されるようになってきた。このアルゴリズムは、発見的な最適化手法の中でも多くの研究者、実務家らに用いられているものの一つである。その特徴としては、複数の解候補を同時並行的に更新していく繰り返し計算型の解法であること、複数の解候補を組み合わせて新しい解候補を生成する交叉と呼ばれる独特の演算子が用いられること、アルゴリズムに柔軟性があり、多種多様な問題に適応可能であることなどがあげられる。

遺伝的アルゴリズムは生物の進化過程を模擬したものである。遺伝子の組み換えが生じる交叉や一部の遺伝子が増える突然変異などによって、その生物個体の周りの環境に適するものが残り、より多くの子を残す可能性が高い。逆に周りの環境に適していないものは淘汰されていく。進化過程はこのような選択、淘汰によって行われ、ある個体の生物としての繁殖能力は適応度と呼ばれる。遺伝的アルゴリズムは、このような生物の進化過程において個体を解候補に適応度を目的関数に対応させ、選択、交叉、突然変異に対応した演算子を実行することにより、目的関数のより良い解候補に次々と進化させて最適解を求めようとする解法である。このアルゴリズムは生物進化に端を発した解法であるが、実際の最適化の計算と上記の生物進化の知見とは基本的に関係がない。

2.7.4 GA のアルゴリズム

遺伝的アルゴリズムでは複数の個体が進化していく。一つの個体は複数の記号を用いて表現され、最適化問題の一つの解候補に対応させる。最も単純な表現は最適化問題の設計変数を 1 列に並べる方法であるが、必ずしもそのようにする必要はなく、交叉および突然変異がうまく働くように全く別の表現で個体を表し、その個体から解候補に変換できるようにしておいてもよい。本論文で使用した GA アルゴリズムでは遺伝子を設計変数の実数値で表現した実数値 GA を用いた^{[17][18][19][20]}。複数の個体からなる個体群に対して、以下の繰り返し計算を用いるアルゴリズムで解を得る。なお、遺伝的アルゴリズムでは 1 回の繰り返しを 1 世代と呼ぶ。一般には個体数と総世代数を大きくすると生成される個体数が増えるので、それだけ良い解が得られやすいが、その一方で計算時間が多く掛かることになる。個体数は数十から数百くらいの値がよく用いられる。総世代数は問題やアルゴリズムの構成に依存するが、利用可能な計算時間に合うように総世代数を設定する。遺伝的アルゴリズムでは交叉則を主に用いて大域的最適解を発見しようとする解法であるが、親個体に近い個体はなかなか生成されず、近傍探索には不向きである。従来からの近傍探索を行う解法と組み合わせてアルゴリズムを構築することもある。

- STEP1 初期個体群を生成する。世代を $g \leftarrow 1$ とする。
- STEP2 交叉則、突然変異則、選択則を現在の個体群に適応し、新しい個体群を生成する。
- STEP3 終了条件を満たせば計算を終了し、それまでに得られた最もよい個体を解として出力する。そうでなければ $g \leftarrow g+1$ として [STEP2] へ戻る。

STEP3 の終了条件は、あらかじめ定めておいた世代数に達した場合とすることが多い。そのほかには、何世代か続けて現最良個体よりも良い個体が生成されない場合とすることなどの方法がある。

次に STEP2 で用いられる交叉則、突然変異則、選択則を以下に説明する。これらの最適化計算進行則は自由に設計できるので、この設定次第でアルゴリズムの性能が良くなるし悪くなる。

1. 交叉則

複数の個体の遺伝子を組み換えて新しい個体を生成する操作である。生成に使用した個体を親、生成された個体を子と呼ぶ。生成された子は親と似た個体になっているので、親が良い個体であれば子もさらに良い個体であることが期待できる。また、一部の遺伝子がよい親同士を交叉した場合に、両親のそれぞれの良い部分が継承されてより多くの遺伝子が良い個体ができることも期待できる。交叉則は遺伝的アルゴリズム独特の演算子である。交叉が有効に機能していれば遺伝的アルゴリズムはほかの解法より良い解が得られる可能性が高く、逆に有効に機能していなければ他の解法より良い解が得られる可能性が低い。交叉則はすべての個体に対して必ず実行されるのではなく、通常はあらかじめ定められた確率の従って実行される。また、通常の交叉則では実行時に乱数が使用される。従って、同じ親に同じ交叉則を適用しても、その時どきによって異なる子が生成される。本論文では、2 個の親から 1 個の子を生成する場合の交叉則を使う。設計変数を 2 個の親同士で足し合わせ、2 で割る。単純に 2 個の親の midpoint を求める交叉方法となる。遺伝子型を実数値として表現しており最適化アルゴリズムとしては簡単なものとなる。

2. 突然変異則

通常は 1 個体の一部の遺伝子を変更して新しい個体を生成する操作である。なお、突然変異はタブーサーチなどの他の解法で新しい解候補を生成する場合に用いられる手続きと同じであるので、そのような手続きを突然変異として利用する事もできる。あらかじめ定められた確率で実行され、実行時には乱数が使用される。本論文では個体の設計変数値を完全に新しいものへと生まれ変わらせる。親個体の性質は遺伝せずに、乱数により設計変数を決定

し新個体を生成する。

3. 選択則

個体群中の個体、あるいは個体群中の個体と交叉則や突然変異則で生成されたこの中から今後の世代に必要な個体を選択する操作である。あらかじめ、各個体に対応した解の目的関数値や各個体の良さを示す適応度を計算しておき、それらの良い個体を選択する。交叉則や突然変異則でよい個体を生成するには、親が良くなければならない。そこで、選択則を用いて良い個体だけを選択するようにする。選択則は後々最適解を発見するのに適している個体を選択するように設計する。一般的には目的関数値のよい個体を選択することになるが、目的関数値が悪くても最適解に対応する個体の遺伝子の構成に近い個体であれば、そのような個体なども選択しておいたほうがよい。しかし、どの個体がそれにあてはまるかは未知であるので、通常は乱数を用いて目的関数値の良い個体を高確率で選択し、悪い個体も低確率で選択しておく手続きが用いられる。また、最も基本的な選択則では個体のよさを示す適応度が計算され、適応度を用いて選択が行われる。本論文では各個体の適応度を計算し、全個体中の一番適応度の低い個体を淘汰させる。計算進行の1世代で1個体の子個体が生成され、1個体の最悪解が淘汰される。各世代において全個体数は変化しない。

第3章 テスト関数へ適用

3.1 数学的テスト関数問題への適用

数学的なテスト関数問題として多峰性のある Schwefel 関数を取り上げる。変数は4つとされている。局所解にトラップされてしまう可能性が高い関数である。最適化手法として大域探索のアルゴリズムを有していないと解くことは難しい。

3.1.1 問題設定（多峰性問題）[2][21]

解空間を表現する関数 F は以下の通りである。

$$F = 2048 + \sum_{i=1}^4 -x_i \sin \left(\sqrt{|x_i|} \right) \quad (3.1)$$
$$(-512 \leq x_i < 512)$$

最適解は以下になる。空間の端に最適解が存在するので発見するのが難しい問題である。

$$\begin{aligned} \min F(x_1, x_2, x_3, x_4) \\ = F(420.97, 420.97, 420.97, 420.97) \\ = 372.07 \end{aligned} \quad (3.2)$$

Schwefel 関数についての概形図を以下に示す。図 3.1 は多峰性を有する関数であることを示すための図である。実際の解空間は5次元空間であるがここでは x_1, x_2 のみの相関関係を図示している。局所解が多数存在するのでローカルな局所解にトラップされてしまう可能性が高い。大域的な探索アルゴリズムを有していないと真の解析解に到達することは難しい。図 3.2 は解析解の場所を示した図である。空間の端の方に存在しているのが分かる。解析解を発見することは困難である。

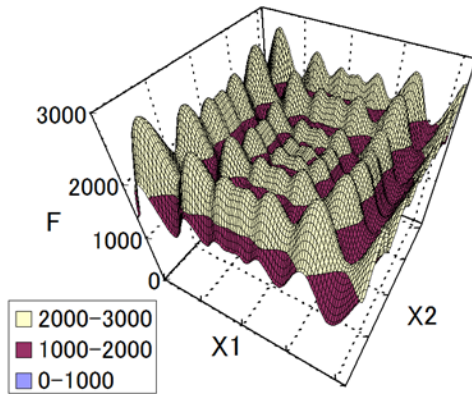


図 3.1 Schwefel 関数

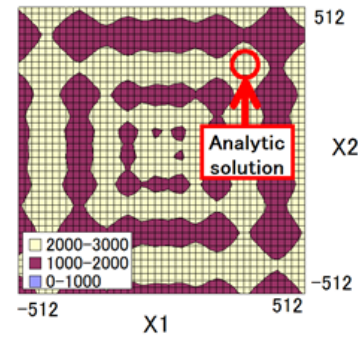


図 3.2 Schwefel 関数の解析解位置

各最適化手法の計算条件は表 3.1 の通りである。

表 3.1 各手法の設定パラメータ（数学関数）

	探索点 (個体数)	計算回数 (世代数)	評価関数 呼び出し回数	各手法独自の 設定パラメータ
提案手法	40	150	6040	$\omega = 2\pi/150, G = 10^8$
PSO	40	400	16040	$c_1 = c_2 = 2.0$ $w = 0.9 \rightarrow 0.4$
GA	40	50000	50040	$P_c = 0.2$

3.1.2 適用結果（多峰性問題）

数値計算によって得られた最適解は以下である。

$$\begin{aligned}
 & F^*(x_1^*, x_2^*, x_3^*, x_4^*) \\
 &= F^*(421.82, 420.84, 421.39, 419.79) \\
 &= 372.36
 \end{aligned} \tag{3.3}$$

計算による最良解と解析解の誤差は以下である。

$$\begin{aligned}
 \left| \frac{F^* - \min F}{\min F} \right| &= \bar{F}(0.2\%, 0.0\%, 0.1\%, 0.3\%) \\
 &= 0.08\%
 \end{aligned} \tag{3.4}$$

これをみると誤差が 0.08 % となっており高い精度で最適解を見つけ出していることが分かる。解空間をトーラス構造にすることで空間の端も探索することが出来るために効率良く最適解を見つけ出せたと考えられる。これにより設計変数空間をトーラス構造にすることは

最適化計算にとって有益であることが分かる。

次に結果グラフの説明を行う。図 3.3 に示したものは変数 x_1 と x_2 の空間である。問題に使用した変数は 4 つあるが、4 次元空間は図示できないので代表として変数 x_1 、 x_2 を用いる。菱形が初期点、四角が重心位置、白い三角が各計算ステップ時の最良解である。矢印で示した箇所が解析解となる。図 3.3 を見ると重心位置の履歴が最適解に一直線に到達する様子がハッキリと分かる。これは探索点が全体的に解析解へ向かって収束しているということである。最適化計算が有効に働いているといえる。他の変数の x_3, x_4 も同様の傾向を示しており、この提案した最適化アルゴリズムが有効に働いているといえる。解析解の値と数値計算結果の値は、式 3.4 をみると誤差は 0.08 % と高い精度で解けていることが分かる。

各探索点の動きをグラフにしたものが図 3.4 である。領域内を縦横無尽に動いている。これは空間の端を繋いだトーラス構造にしたお陰で、探索点が制限を受けずに動けるようになったためである。領域の隅に最適解があっても本手法にとっては関係なく、簡単に発見している。

コリオリの影響について述べる。このグラフについては $(x_1, x_2) = (0, 0)$ を手前に向かって通る Z 軸を時計回りに回転させている。これによって、探索点は半径方向外側に移動すると反時計方向に速度ベクトルを曲げられ、半径内側に移動すると時計回りの方向にコリオリ力を受ける。このグラフにおいて、ゆるやかに曲がっているものはコリオリの影響を受けていると推察される。急激に角度を変えたものは重力の影響が強いと考えられる。コリオリ力の解探索への影響力についての検討は次節でも述べている。

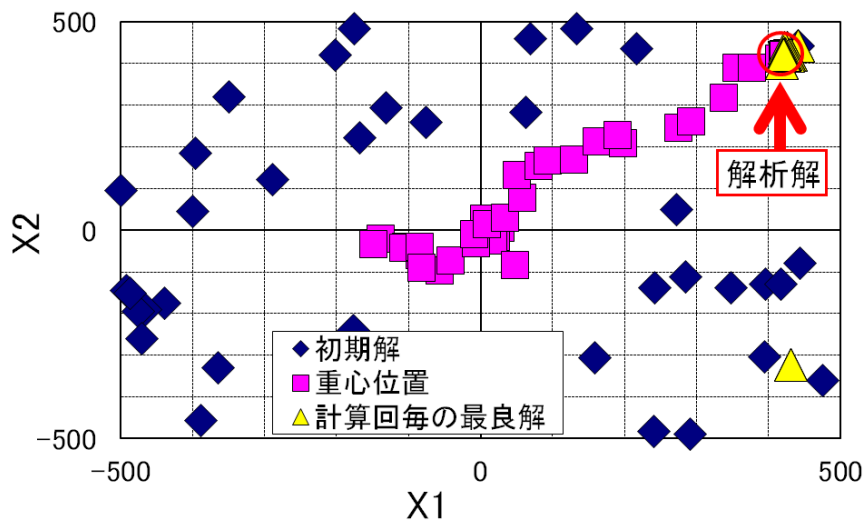


図 3.3 初期解、重心と最良解の履歴（提案手法、数学関数）

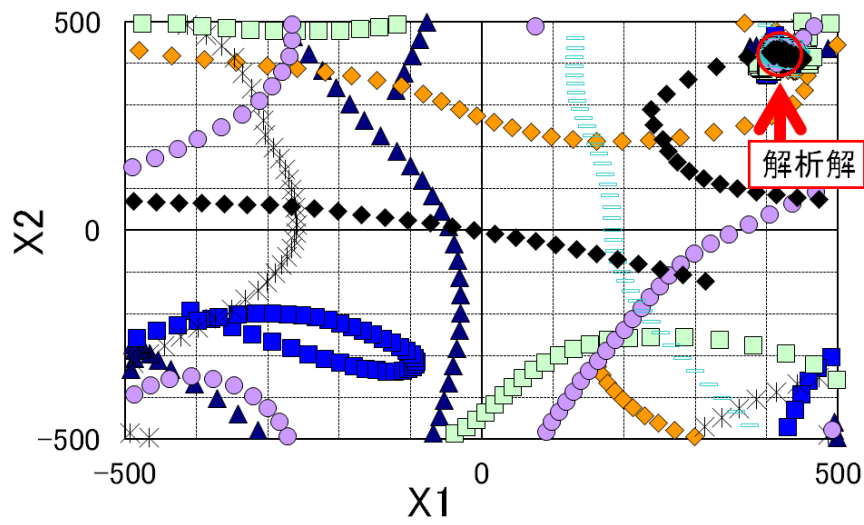


図 3.4 各探索点の履歴（提案手法、数学関数）

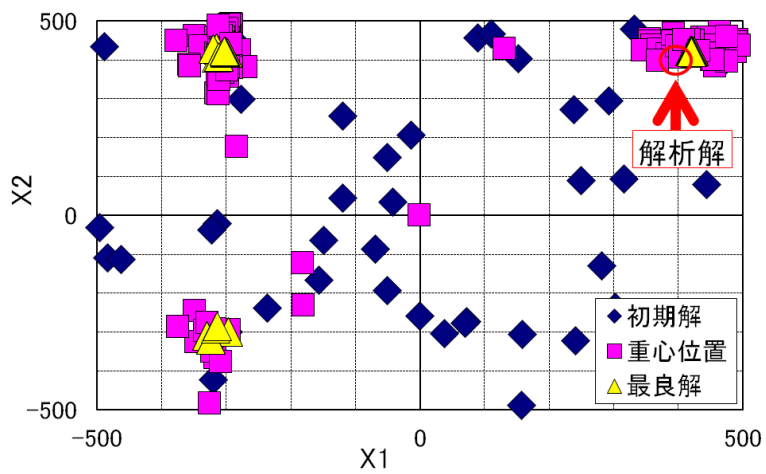


図 3.5 初期解、重心と最良解の履歴（PSO、数学関数）

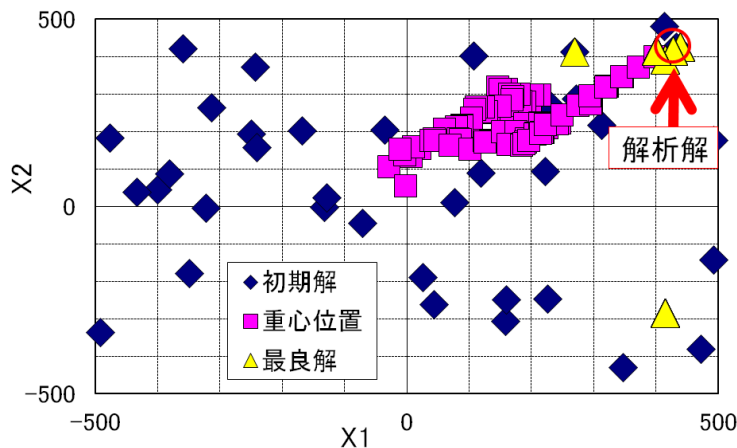


図 3.6 初期解、重心と最良解の履歴 (GA、数学関数)

最後に他の手法との比較である。前節と同様に図 3.5 は PSO、図 3.6 は GA を用いて解いたものである。PSO と GA は共に最適解に到達できている。大きく違うことは計算回数である。本手法を基準とすると PSO は 2.7 倍、GA は 333 倍という計算回数を要している。本手法は設計変数空間をトーラスとして空間の端を連結しているの、空間の隅にある最適解に素早く到達することが出来た。他 2 つの手法はトーラス構造としていないので、収束に時間がかかったと推測できる。解析解と計算結果の誤差は PSO は 0.01%、GA は 15.7% となった。PSO については重心位置の分布が飛び飛びとなっている。速度制限を緩和して個体の移動距離を空間幅の半分としたためである。個体をあちこちに飛び回るようにして初めて最適解に到達させることができた。これと比べて本手法と GA の結果では重心位置は連続的に変化している。この連続的な履歴は正常に最適化アルゴリズムが作動していると考えられる。飛び飛びの分布では、たまたま局所解に辿り着いたと言える。また、PSO は手法自体に局所解脱出のアルゴリズムを備えていないので、このような多峰性の関数では局所解に収束してしまう危険性がある。GA の交叉方法は実数値中点交叉とし、突然変異率は 20% とした。多峰性関数の探索において、突然変異に頼った探索をしているといえる。

3.1.3 コリオリ力を導入する場合としない場合の比較

本節の数学的テスト関数問題について提案手法のコリオリ力を導入した場合と導入しない場合の結果について述べる。探索点群の初期分布について乱数を変化させた場合の解析解への収束状況を比べる。探索点の初期位置を変更した場合でのコリオリ力の導入と不導入について比べる。乱数は一様乱数^[58]であり、この節では乱数を 100 回変化させる。初期分布以外の探索点数や重力定数などの最適化計算パラメータは固定している。解析解と計算解との

誤差が 10^{-3} 未満になるものについて評価する。また、初期値依存性についても評価する。下表における到達率とは初期分布を計算する乱数を 100 回変更して最適化計算を行なった場合に計算解の精度が 10^{-3} 未満になったものを解析解に到達したと評価し、到達数を計算実行数 100 回で割った値である。

表 3.2 コリオリ力を導入した場合としない場合の比較

		コリオリ力を導入した場合	コリオリ力を導入しない場合
到達率 (到達数/100 回)		10/100	1/100
最良解 (目的関数値) [解析解 $f = 372.07$]		372.09	372.38
誤差 (目的関数値)		6.05×10^{-5}	8.39×10^{-4}
最良解 (設計変数値) [解析解 $x_{1,2,3,4} = 420.97$]	x_1	421.28	420.37
	x_2	420.72	420.61
	x_3	420.79	419.81
	x_4	420.91	421.78
誤差 (設計変数値)	x_1	7.25×10^{-4}	1.43×10^{-3}
	x_2	5.95×10^{-4}	8.55×10^{-4}
	x_3	4.26×10^{-4}	2.77×10^{-3}
	x_4	1.34×10^{-4}	1.91×10^{-3}

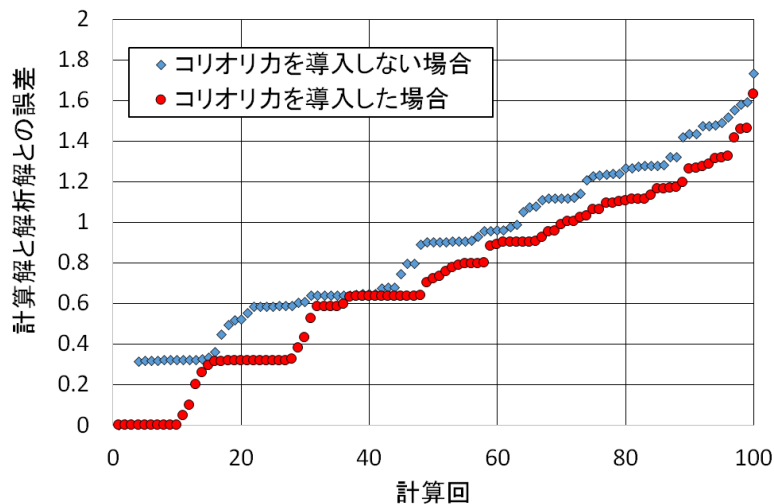


図 3.7 コリオリ力の導入不導入の誤差比較

到達率についてはコリオリ力を導入した場合は 1/10 の割合で解析解に到達していることが分かる。コリオリ力を導入していない場合は 1/100 の割合でしか解析解に到達していない。

図 3.7 については、コリオリ力を導入した場合としない場合の計算結果について示している。乱数を変えた 100 回の計算結果について表示しており、縦軸が目的関数の計算結果と解析解の誤差である。横軸は計算回としており、誤差の結果をもとに 100 回の計算結果を誤差値の昇順に並べ替えた結果を表示している。青いひし形のポイントがコリオリ力を導入しない場合、赤い丸のポイントがコリオリ力を導入した場合である。100 回の計算結果においてすべての回でコリオリ力を導入した場合の方が誤差が少ない結果となった。平均で約 18% の減少を結果から読み取れる。誤差減少によって計算上の解が解析解により近づくことができたことを示している。このことから、コリオリ力を導入した方が解析解に近づく可能性が高いといえる。理由としては、探索点群の位置の多様性を保つことが出来たためであると考えられる。探索点が重力による引力で速度ベクトルを得るので、それにしたがって進行するたびにコリオリ力の影響で速度ベクトルが方向を曲げられる。探索点群の初期位置分布の段階で全個体の重心位置に収束する様に探索が進むのだが、探索点が進行方向の左右に強制的に移動させられるので未探索の峰に到達することで最適解の候補を多数保有することができる。これによって解の多様性を確保することが出来るので、より評価の高い解に到達することが出来る。しかしながら、コリオリ力を導入した場合でも 1 割の確率でしか解析解に到達していない。重力定数や回転パラメータなどの設定によって探索点の移動履歴が異なる。この節のテスト関数は多峰性を有する数学的関数であり 10^{-3} 程度の設計変数の微小な変化に対しても目的関数値が数% 変化する。また、局所解が多数存在し解空間の対局に位置するなど多点探索を行う上で大域的最適解に到達することを妨げるトラップが存在する。ヒューリスティックな最適化手法は解析解に到達できるかは計算上のパラメータ設定に強く依存する。

3.1.4 Schwefel 関数の係数を変更した場合の比較

次に Schwefel 関数に係数を掛けて、その値を変化させた場合について本提案手法を適用した結果について示す。本提案手法にはコリオリ力を導入した場合のみ計算を行い初期値を生成する乱数を 100 回変えて計算を実行した。解析解に到達できたかどうかをカウントし表にまとめる。サイン関数の内部に係数 A を加え、それ以外の問題設定は同一とする。係数 A を $0.5 \leq A \leq 2.0$ の範囲で 0.1 刻みで変動させて、それぞれに提案手法を適用した。各計算について提案手法の設定パラメータは同一とした。設定パラメータの値は前節までの値を採用するが、重力定数は $G = 10^8$ 、最大計算回数は 500 回とした。図 3.8 と図 3.9 は係数 A を変更したときの山谷の図であり、図 3.8 は $A = 0.5$ のとき、図 3.9 は $A = 2.0$ のときを表

示したものである。表 3.3 の山幅とは解析解の谷底からすぐ隣の谷底までの距離を求めたものである。山幅（無次元）とは前述の山幅を解空間の上下限幅である 1024 で割ったものである。これにより解空間に対する山幅の占める割合が分かる。Schwefel 関数の係数 A を変化せることで解析解の位置が移動するので表 3.3 には各係数 A のときの解析解の位置 x とそのときの関数値 F を載せている。表 3.4 では本提案手法の設定パラメータである回転量 ω による平均強制移動幅を示している。平均移動幅の計算は $256 \sin \omega$ から求めている。回転中心である原点から上限までの距離 512 に $\sin \omega$ をかけて、さらに移動平均として $1/2$ をかけたものである。表の値はさらに 1024 で割って無次元化している。表 3.5 に本提案手法を 100 回適用した結果、解析解との誤差 10^{-3} 以下のときの回数である到達率を示す。数学的関数の境界上および境界付近に解析解がある場合は左右対称の解空間となり特殊な状況なので結果から省き、山幅と設定パラメータである ω との関係を見る。表 3.5 をみると山幅が狭くなる（係数 A が大きくなる）につれて到達率が減少する傾向があることが分かる。局所解が多数存在しローカルの局所解に探索点が捕らわれてしまうからである。一方、山幅が狭くなるときに回転量 ω を大きくする傾向にすると到達率があがっていることが分かる。これはコリオリ力によって強制回転することでローカルにトラップされた探索点が谷底から脱出したことで大域的最適解に到達できたと考えられる。また、回転によって動かす幅を大きくすることで解空間を探索点が走破し解群の多様性を保つことが出来るためである。具体的な数値で述べると、解空間の山幅に対して $1/10$ 以上の回転移動幅を持つことが出来れば解析解への到達率が少なくとも 1 つは存在することができたと言える。必ずしも山幅を超えるような回転移動幅を加える必要はないことが分かる。あくまでも局所収束は重力方程式によるもので、コリオリ力の強制回転はこの重力のサポート的な作用を行わせるものである。ただし、事前に解空間の山幅が分かっていることは稀であり、通常は未知の解空間に最適化手法を適用することが多い。解空間の山幅に対して $1/10$ という指標はトライアンドエラーを行い設定パラメータを調整する際の調整法則の 1 つとすることとする。

$$F = 2048 + \sum_{i=1}^4 -x_i \sin \left(A \sqrt{|x_i|} \right) \quad (3.5)$$

$$(-512 \leq x_i < 512)$$

表 3.3 係数 A 変更による各解析解の値

係数 A	山幅	山幅（無次元）	変数 x の解析解値	関数 F の解析解値
0.5	380	0.371	-499.317	81.98122
0.6	340	0.332	512	313.6017
0.7	300	0.292	415.9475	400.3003
0.8	270	0.263	-472.695	169.5965
0.9	260	0.253	512	3.173501
1.0	220	0.214	420.9687	372.0685
1.1	200	0.195	-462.106	206.1511
1.2	200	0.195	497.9625	61.68257
1.3	185	0.181	-512	186.0014
1.4	180	0.175	-456.491	226.106
1.5	175	0.171	485.3846	110.0074
1.6	170	0.166	-511.425	5.417753
1.7	140	0.136	-453.027	238.6538
1.8	137	0.133	477.1974	141.6747
1.9	130	0.126	-499.371	52.72736
2	122	0.119	512	90.42043

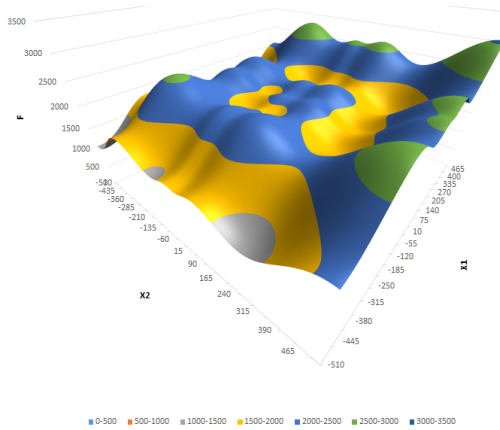
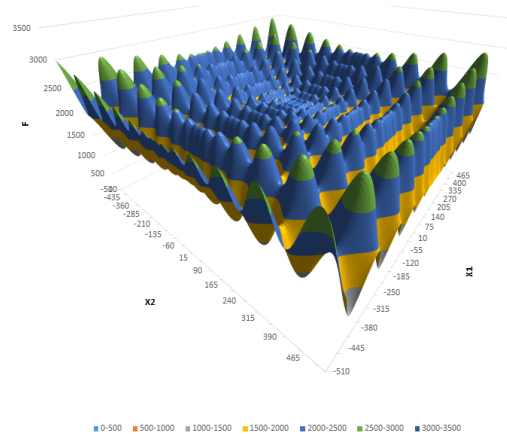

図 3.8 係数 $A = 0.5$ のとき

図 3.9 係数 $A = 2.0$ のとき

表 3.4 回転量 ω と平均移動幅（無次元）

回転量 ω	$2\pi/10$	$2\pi/50$	$2\pi/100$	$2\pi/250$	$2\pi/500$	$2\pi/750$	$2\pi/1000$
平均移動幅	0.14694	0.03133	0.01569	0.00628	0.00314	0.00209	0.00157

表 3.5 係数 A と回転量 ω を変更した場合の到達率

		回転量 ω						
		$2\pi/10$	$2\pi/50$	$2\pi/100$	$2\pi/250$	$2\pi/500$	$2\pi/750$	$2\pi/1000$
係数 A	0.5	0	1	0	0	0	0	0
	0.7	7	15	13	13	12	9	10
	0.8	6	14	11	11	8	4	4
	1.0	1	10	13	8	5	8	5
	1.1	1	5	2	6	7	7	3
	1.4	1	1	5	3	3	0	1
	1.5	1	0	1	0	0	0	0
	1.7	0	2	4	1	1	0	1
	1.8	1	0	0	0	0	0	0
	1.9	0	0	0	0	0	0	0

3.2 不静定トラス問題

3.2.1 問題概要

テスト問題として不静定トラス問題を解く。工学的最適化問題として解析解がある問題なのでテスト問題として数値計算解と解析解が比較可能である。制約条件があるので今後の応用問題に対する練習問題として有用である。

3つの部材から成るトラス構造を考える。最下点において角度 45° で力 P を加える。その時に部材に掛かる荷重が制約条件を満たしつつ、三部材の体積を最小とする問題である。この問題設定においては解析解を求めることが出来るので、数値計算の結果と比較し考察する。

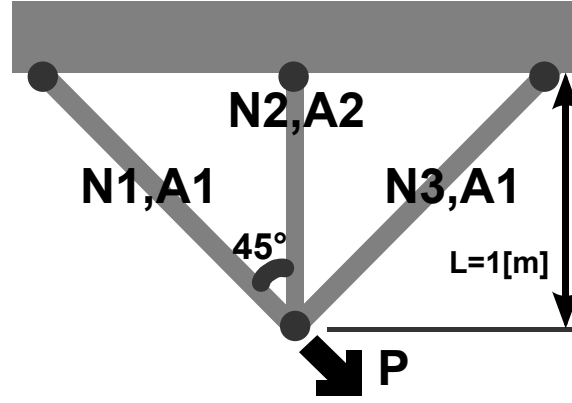


図 3.10 三部材不静定トラス

3.2.2 問題設定（トラス問題）

左右の部材断面積を A_1 、中央の部材断面積を A_2 とする。最小化する体積 V は

$$V = (2\sqrt{2}A_1 + A_2)l \quad (3.6)$$

となる。設計変数である断面積は A_1 、 A_2 の2つとなる。この断面積を制約条件の下で動かしていき、体積を最小化する問題となる。各部材にかかる力 N_1, N_2, N_3 は以下の通りである。

$$N_1 = \frac{2A_1 + \sqrt{2}A_2}{2A_1 + 2\sqrt{2}A_2}P \quad (3.7)$$

$$N_2 = \frac{A_2}{A_1 + \sqrt{2}A_2}P \quad (3.8)$$

$$N_3 = \frac{\sqrt{2}A_2}{2A_1 + 2\sqrt{2}A_2}P \quad (3.9)$$

次に制約条件を示す。上述の部材力に引張限界 $\sigma_Y A_1, \sigma_Y A_2$ 、座屈限界 $\sigma_K A_1$ の制約項が加わる。

$$N_1 \leq \sigma_Y A_1 \quad (3.10)$$

$$N_2 \leq \sigma_Y A_2 \quad (3.11)$$

$$N_3 \leq \sigma_K A_1 \quad (3.12)$$

部材力 N_1, N_2 は引張限界、部材力 N_3 は座屈限界を制約条件として制限されている。

また、設計変数 A_1, A_2 の存在可能範囲は以下である。

$$0 < A_{1,2} \leq \frac{2P}{\sigma_Y} \quad (3.13)$$

解析解はラグランジュの未定乗数法により求めている。最適解は以下である。

$$A_1 = \frac{P}{\sigma_Y + \sigma_K}, \quad A_2 = \frac{\sqrt{2}\sigma_Y P}{\sigma_Y^2 - \sigma_K^2} \quad (3.14)$$

部材等に関する各定数は以下の通りとした。

$$P = 10[\text{kN}], \sigma_Y = 90[\text{MPa}], \sigma_K = 30[\text{MPa}] \quad (3.15)$$

各パラメータを代入した解析解 $\min V$ は以下になる。

$$\begin{aligned} \min V(A_1, A_2) &= \min V(8.33 \times 10^{-5}, 5.89 \times 10^{-5}) \\ &= 2.95 \times 10^{-4} \end{aligned} \quad (3.16)$$

次に最適化問題設定を以下に示す。各手法の計算パラメータは以下の表の通りである。3つの手法について探索点数（個体数）、計算回数（世代数）、評価関数の呼び出し回数を示している。

表 3.6 各手法の設定パラメータ（トラス問題）

	探索点 (個体数)	計算回数 (世代数)	評価関数 呼び出し回数	各手法独自の 設定パラメータ
提案手法	10	100	1010	$\omega = 2\pi/100, G = 10$
PSO	10	150	1510	$c_1 = c_2 = 2.0$ $w = 0.9 \rightarrow 0.4$
GA	20	30000	66020	$P_c = 0.1$

3.2.3 適用結果（トラス問題）

以上の問題設定について、提案手法を適用した。以下の式 3.17 は数値計算によって得られた解である。

$$\begin{aligned} V^*(A_1^*, A_2^*) &= V^*(8.36 \times 10^{-5}, 5.80 \times 10^{-5}) \\ &= 2.95 \times 10^{-4} \end{aligned} \quad (3.17)$$

計算による最良解と解析解の誤差は以下である。

$$\left| \frac{V^* - \min V}{\min V} \right| = \bar{V}(0.4\%, 1.5\%) = 0.17\% \quad (3.18)$$

ほぼ誤差なく最適解を得られている。これによって提案した本手法は最適化問題に対して有効なアルゴリズムを備えていることが分かる。

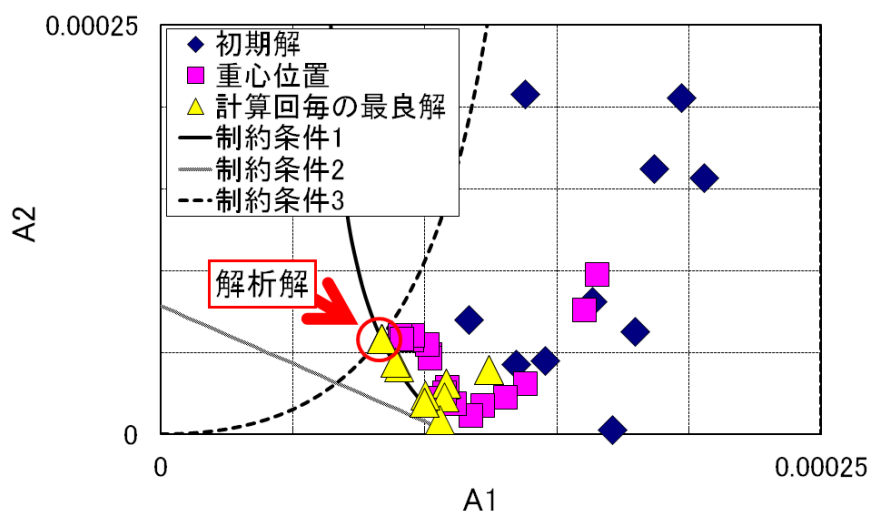


図 3.11 初期分布、重心履歴、各最良解（提案手法、トラス問題）

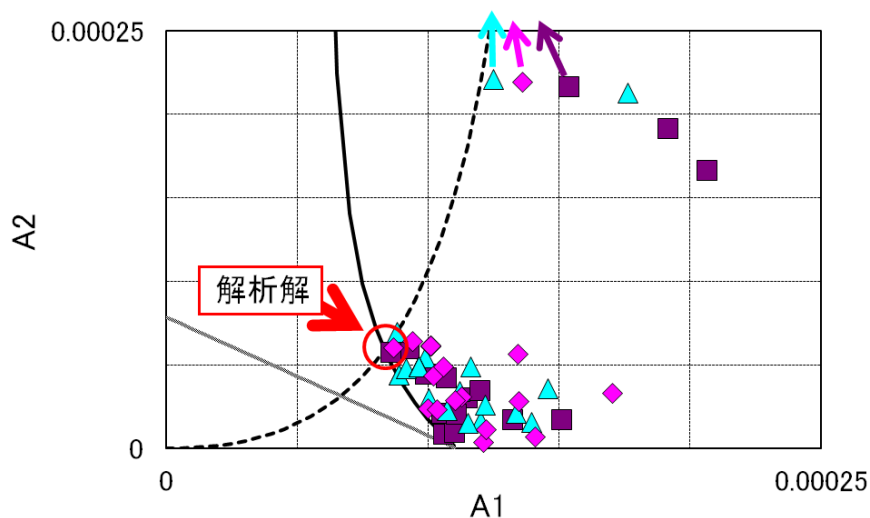


図 3.12 探索点の移動履歴（提案手法、トラス問題）

次にグラフを用いて計算結果を示していく。図 3.11 は探索点の初期分布と重心位置である。菱形が初期点、四角が重心位置、三角が各計算ステップでの最良解である。矢印で示した箇所が解析解である。図の重心位置が左下方向から左中央に進んでいるので、探索点が解析解に向かって動いているのが分かる。重心位置の移動は全体の探索点が収束しながら、ある一点に向かっていていることを表している。この結果から本手法のアルゴリズムが効果的に解空間を探索していると言える。解析解と数値計算結果の値は式 3.18 で、誤差を取ると 0.17 %で解析解とほぼ一致した結果を得る事が出来た。

次に代表的な探索点をピックアップした移動履歴が図 3.12 である。この問題では少ない計算ステップで解析解付近に到達した。図面下部に密集しているが探索が左中央方向に進んでいる。また、解空間がトラス構造になっているので、上方の探索点は上から進んで下から出てきている。これは探索点の移動距離が少なくて済むことを示している。コリオリ力のための回転軸は $(A_1, A_2) = (0, 0)$ にあり、反時計回りに回転させている。各探索点は半径方向に進むとコリオリ力を受ける。しかし、この問題では直ぐに収束してしまったので、コリオリ力の影響はほとんど現れなかった。

次に他の手法との比較を示す。本手法、PSO（粒子群最適化法）、GA（遺伝的アルゴリズム）と比較する。計算コストである評価関数の呼び出し回数は表 3.6 に示している。図 3.13 の PSO を見ると四角いポイントの重心位置が解析解に一直線に行っている。収束性が早いので一気に探索が進んでいる。本手法と PSO の誤差を比較すると 0.17 % と 0.60 % なので本手法は PSO とほぼ同一の精度で解けている。

図 3.14 における GA について述べる。この最適化計算に使った GA アルゴリズムは実数値 GA において交差のフェーズに実数値中点交叉アルゴリズムを採用した。表 3.6 より計算コストが本手法の 66 倍となり、重いコストが掛かる結果となった。解析解との誤差は 0.34% で他手法と同等の結果であった。

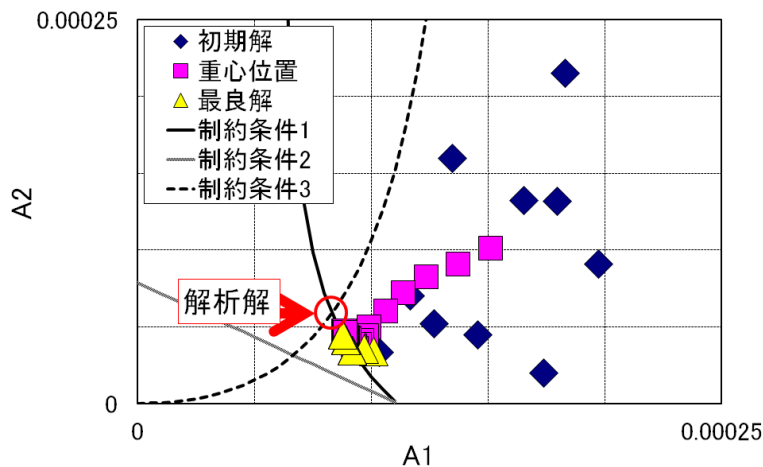


図 3.13 初期分布、重心履歴、各最良解 (PSO、トラス問題)

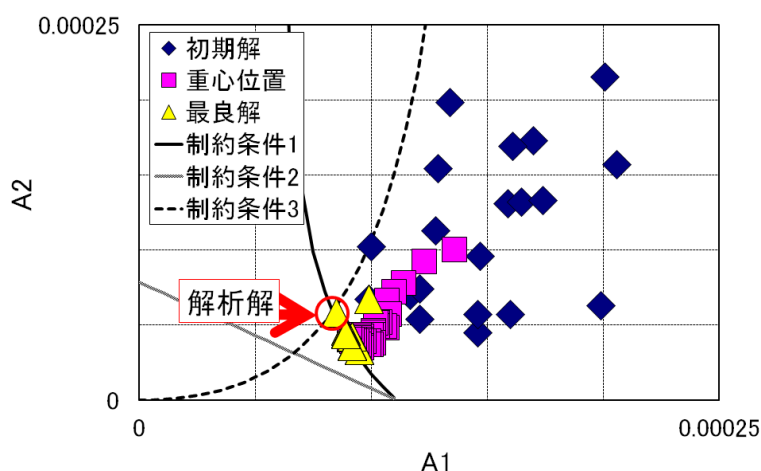


図 3.14 初期分布、重心履歴、各最良解 (GA、トラス問題)

3.2.4 問題設定を制限した場合

設計変数の取り得る範囲を制限した場合についての結果を示す。上記に示した材料の物性値を用いて、制約条件 1,2,3 は加えないものとする。設計変数である A_1, A_2 の範囲は以下とする。

$$\begin{aligned} 0.00015 &\leq A_1 \leq 0.00025 \\ 0.00015 &\leq A_2 \leq 0.00025 \end{aligned} \quad (3.19)$$

制約条件が加わらない単純な解空間に対して本提案手法が有効なアルゴリズムを有しているかどうかを検証するためである。解析解は $A_{1,2} = 0.00015$ となる。解空間の領域の端に解析解が存在する。この問題設定にした場合でも提案手法によって解析解が得られることを示す。図 3.15 では初期探索点群が青いひし形のポイント、重心がピンクの四角いポイント、各計算回の最良解が黄色い三角のポイントである。最終世代での最良解を提案手法では計算解として採用する。計算解は有効数字 3 ケタで四捨五入すると $A_{1,2}$ ともに $A_{1,2} = 0.000150$ となり解析解を得ることが出来た。図 3.15 の赤い円で囲まれている部分が解析解である。黄色い三角のポイントがたどり着いているのが分かる。設計変数の上限下限の境界上に解析解が存在する場合でも本提案手法が適用できるといえる。

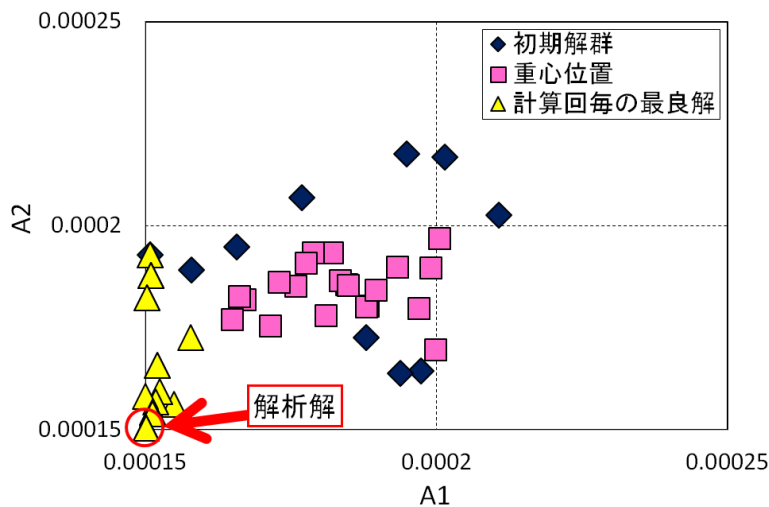


図 3.15 初期分布、重心履歴、各最良解 (提案手法、制限トラス問題)

3.3 ポートフォリオ問題へ適用

3.3.1 問題概要

経済学の中の簡単な問題としてポートフォリオ問題に適用した。最適化手法は工学だけでなく金融系の分野でも使用されるものなので、テスト問題として広い分野から取り上げる。

問題概要は 100 万円を初期費用として複数ある投資先にどのように資産を分配するかという問題である。各投資先には固定された利率があり、12 か月間資産を運用する。各投資先における初期投資額は 12 か月間変わらない。最初に分配した額で 12 か月間各投資先で運用することになる。各月で増額減額を計算し、翌月で利率を含めた複利計算を行う。最終月の全額から初期投資額を引いた純利率を目的関数とし、これを最大化する問題とする。表 3.7 に各投資先と運用利率を示している。投資先は 4 つあり、宝くじ、株式投資、信託投資、銀行預金がある。毎月の利益は運用利率によって左右される。表 3.7 の運用利率の列の数字は投資額の利益率を表しており、括弧内のパーセント数はその事象が起こる確率である。問題設定を簡易なものにするために運用利率の期待値をとり、今回の問題設定ではこの期待値を用いて利益を算出する。

3.3.2 問題設定

最適化問題としての変数は 3 つある。表 3.7 に示している通りに、宝くじの初期投資額を x 、株式投資の初期投資額を y 、信託投資の初期投資額 z とする。残りの初期資産はすべて銀行預金にまわすものとする。本来の最適化問題としては運用利率は乱数によって決定されるが、本節では最適化計算結果を最適化手法のアルゴリズムの性質によって比較したいので

表 3.7 各投資先と運用利率

	宝くじ	株式投資	信託投資	銀行預金
変数	x	y	z	a
運用利率	100.000(1%) 0.000(99%)	1.150(10%) 1.010(40%) 0.990(40%) 0.950(10%)	1.010(10%) 1.005(80%) 1.000(10%)	1.001(100%)
期待値	1.000	1.010	1.005	1.001

確率の要素を排除するためにこれらの運用利率の期待値をとる。この期待値を用いて毎月の利益を計算する。問題設定としては簡素なものになる。

銀行預金の額を a とし、初期資産を S_0 とすると関係式は以下になる。

$$S_0 = x + y + z + a \quad (3.20)$$

この問題設定での解析解は事前に求められており初期資産全額を株式投資に投入する場合である。純利率を δS とすると

$$\delta S = 12.07[\text{万円}] \quad (3.21)$$

となる。解析解での各変数値は

$$(x, y, z) = (0.00, 100.00, 0.00) \quad (3.22)$$

である。

次に最適化問題設定を以下に示す。各手法の計算パラメータは以下の表の通りである。3つの手法について探索点数（個体数）、計算回数（世代数）、評価関数の呼び出し回数を示している。

表 3.8 各手法の設定パラメータ（ポートフォリオ問題）

	探索点 (個体数)	計算回数 (世代数)	評価関数 呼び出し回数	各手法独自の 設定パラメータ
提案手法	10	100	1010	$\omega = 2\pi/300, G = 10$
PSO	10	100	1010	$c_1 = c_2 = 2.0$ $w = 0.9 \rightarrow 0.4$
GA	50	1000	2050	$P_c = 0.02$

3.3.3 適用結果（ポートフォリオ）

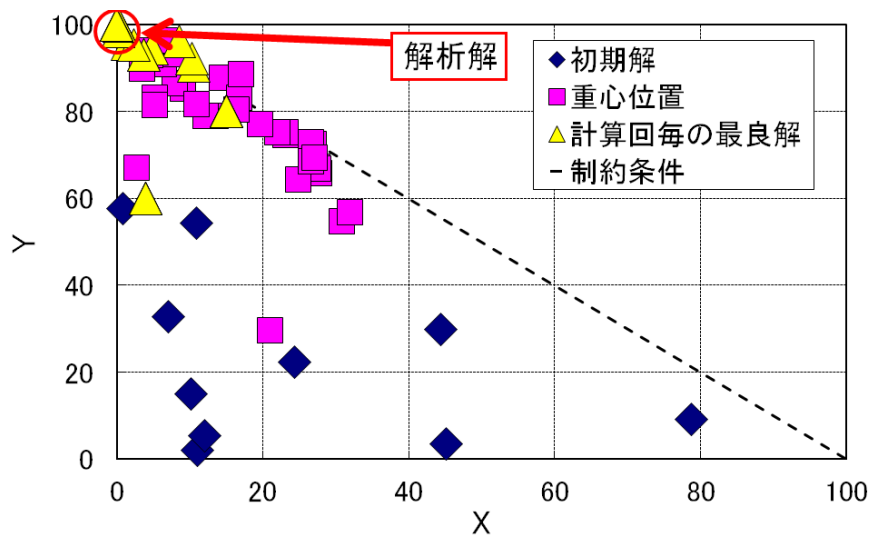


図 3.16 初期分布、重心履歴、各最良解 (提案手法、ポートフォリオ)

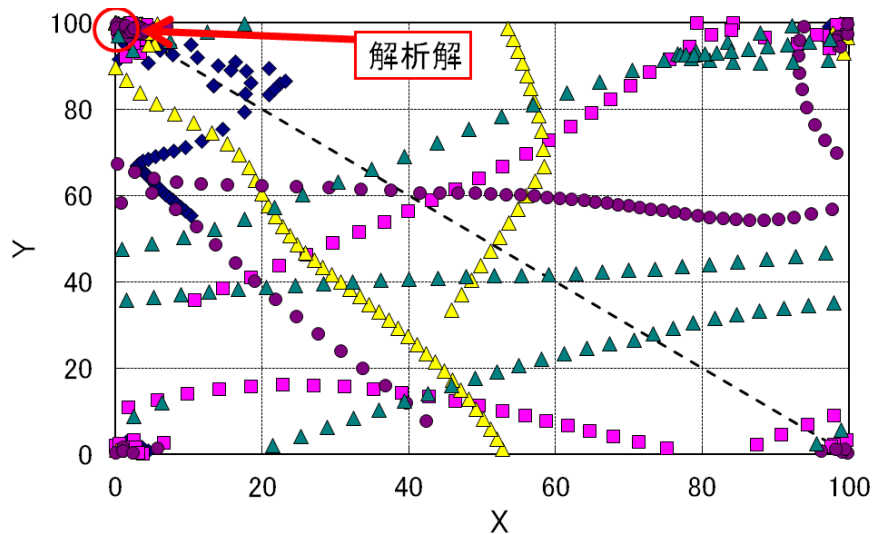


図 3.17 探索点群の移動履歴 (提案手法、ポートフォリオ)

適用結果としては図に示すような最適解に到達する結果となった。問題設定を簡単にしたために容易に解析解に到達することが出来た。参考文献での問題設定では乱数を用いた確率変動によって得られる報酬を計算するのだが、本節の目的は提案手法の有効性を示すことであり、また多手法との結果と比較をしなければならないので今回は期待値を取り報酬を計算した。確率変動の要素が入らないために多手法との比較が容易になったが、問題設定自体も平易になった。

解析解は解空間における端に位置している。従来の多点探索手法では制約条件などの境界上にある解析解の発見は難しかった。特に今回のような鋭角三角形の鋭い頂点に解析解が存在する場合は大域探索法のみではなく狭い範囲を探索する局所探索法のアルゴリズムも有効に働かなくては鋭角の頂点にある解析解に到達することはできない。

提案手法では解析解に到達している。重力による局所的な探索方法が有効に働いていることが分かる。また、トーラス構造にすることで端に解析解が存在する場合でも対局側に位置している探索点も端を飛び超えて最適解に移動することができる。本テスト問題では制約条件を超越し解析解に探索点群が収束する様子が分かる。このことから、重力による局所探索アルゴリズムが有効に稼働しているといえる。トーラス構造についても制約条件の壁を超えることで解析解に到達することが出来ているので有効なアルゴリズムであるといえる。

図 3.17 は探索点群の移動履歴を表したものである。図 3.16 の初期位置点から各探索点の位置を計算回ごとに表示した図である。設計変数の上限下限で空間を連結している。そのために探索点の移動は縦横無尽に解空間を移動している様子が分かる。式 3.20 の制約条件はペナルティ法により処理しているので、図 3.17 のように式 3.20 による制約条件外にも移動することが許容されている。ペナルティ法とは制約条件を満たしていない場合に満たしていない度合いに応じて重みをかけた量を評価関数値に加えて評価を改悪するという手法である。解空間を広く移動することができるので端にある最適解にも到達することができる。図の右から左に移動することで解析解に到達する探索点と左から右へ移動することで解析解に到達する探索点が存在するので多方面から解析解にアプローチすることが出来る。上下に関しても同様であり、さらに多次元になった場合も同様に多方面から解析解にアプローチすることができるので、トーラス構造が解空間探索に有効に働いている。

3.3.4 他手法との比較

この節では他手法によるポートフォリオ問題の最適化計算結果について示す。最適化問題設定は提案手法の設定と同一とする。各々の手法独自の設定パラメータは表 3.8 に示す。以下に結果を示す。図 3.18 は P S O による最適化結果、図 3.19 が G A による計算結果である。PSO については図 3.18 をみると最良解の履歴、重心位置の履歴ともに解析解に到達していることが分かる。ほぼ蛇行もなく直線的に収束している。本問題設定で使用している P S O のアルゴリズムは単純 P S O であるが計算条件を考えると 10 個体、100 世代での収束性の高さが分かる。大域的最適解を発見した後の収束性の早さは P S O の優位点である。しかし、大域的最適解が発見できない場合では局所解に収束しやすいというデメリットもある。この節での最適化問題設定は初期個体群の生成を制約条件内で行い、解析解が条件領域内の端にあるので単純 P S O では最適解に到達することは難しい。P S O は最良解に急速に収束する

性質があるので初期個体群外に移動する状況では最適解に到達することが困難になる。だが今回の問題設定でのPSOパラメータの設定値では解空間内の個体群と解析解の位置関係や速度ベクトルの向きがバランスよく調整され、大域的な最適解に到達することができた。ただ、今回のような実行可能領域内で移動する制限が無く大域的な最適解に到達することができる最適化問題ばかりではなく、実行不可能領域が存在している場合もある。このことから実行不可能領域を飛び越えるアルゴリズムや局所解にトラップされた場合の脱出アルゴリズムが必要になる。本提案手法は宇宙ジェットのアルゴリズムがそれに対応する。

GAは図3.19をみると最良解が初期世代群の領域から脱出できていない。重心位置の履歴を追うと初期値の最良解を計算回数を重ねるごとに赤い四角のポイントが近づいていることが分かる。初期解群の外にある大域的な最適解が探索過程で発見されずに初期解群内で交叉が進んだと考えられる。突然変異によって個体が移動したものがなく制約条件を満たす子個体が生成されずに次の計算回に進んでいるからである。突然変異での子個体生成はランダムに設計変数を決定し制約条件を満たさない場合はその子個体を破棄し次の計算回へと移行するように計算している。提案手法とPSOによって得られているような大域的な最適解を得られていないが世代数の繰り返し計算回数を増やすことで突然変異の回数を増やし大域的な最適解に到達する可能性はある。GAと提案手法の設計変数の関係図とを比較する。提案手法の結果をもとに考えると最適解への計算方向は左上の方向に進むことが良い。GAにおいては初期世代から脱出できておらず最終世代まで繰り返し計算が行われた。交叉の方法によって初期領域外に移動する方法が突然変異にのみ依存することが原因である。個体群の移動に速度の概念を導入することで局所探索においても初期個体群の外側に移動できる。本提案手法とPSOは速度の概念を有しており初期個体群の外側に移動できることが確認できた。

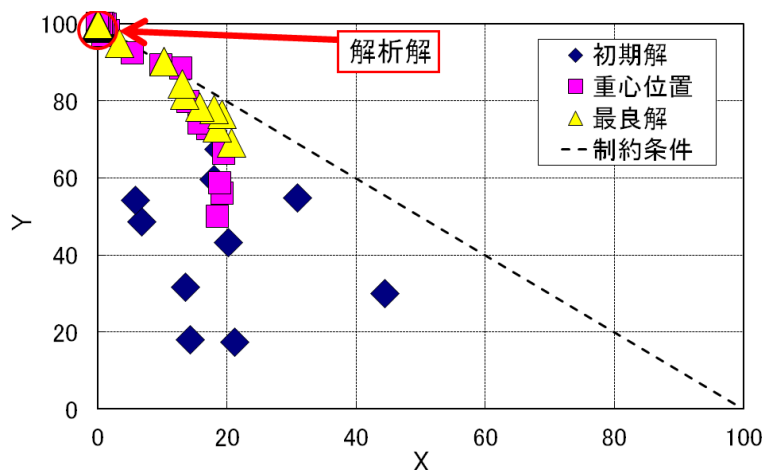


図 3.18 初期分布、重心履歴、各最良解 (PSO、ポートフォリオ)

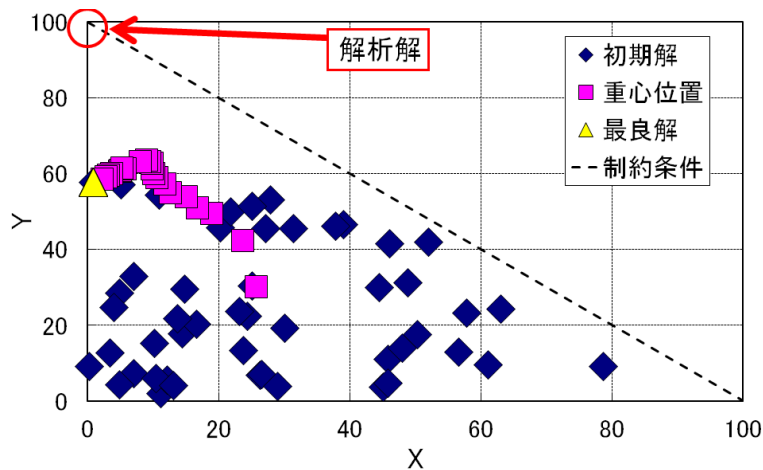


図 3.19 初期分布、重心履歴、各最良解 (GA、ポートフォリオ)

3.4 高次元数学的多峰性問題へ適用

3.4.1 問題概要

最後に高次元の数学関数における最適化問題に適用する。第一節に示したような数学的なテスト関数に3種類の最適化手法を用いる。複数のテスト関数に対して高い次元数の設計変数を設定し問題を解く。乱数を変化させて100回計算した平均値と最良値をとる。結果の比較は目的関数値のみによって議論する。

3.4.2 問題設定 (テスト関数) ^[13]

次はさらに設計変数空間を20次元まで拡張する。8つのテスト関数を用意し、各手法で計算条件を合わせて解いた結果を示す。前節と同様にGAとPSOの結果と比較する。表3.11のとおり各手法の最大計算回数は10万回、探索個体数は30個体とした。手法独自の設定パラメータの値も表3.11に示した。適用した8つのテスト関数を表3.9にまとめて示す。関数 $f(x)$ を最小化する問題である。表3.9において x^* は解析解、 $f(x^*)$ は解析解による最小値である。各手法とも乱数値を変化させて30回計算プログラムを実行した。その平均値と最良値を表3.12に示している。代表的な関数の概形を以下に2つ図示する。多峰性を有しており最適化計算で解析解を得ることが難しい数学的関数である。

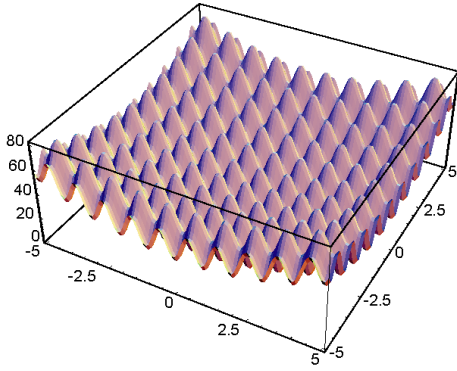


図 3.20 f6 - Rastrigin 関数

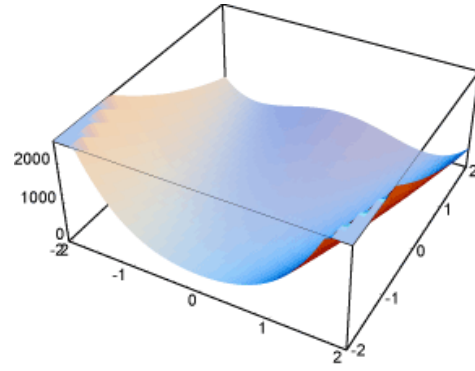


図 3.21 f7 - Rosenbrock 関数

表 3.9 数学的多峰性関数式

f	関数名	$f(x)$	x 範囲
f_1	Ackley	$-20 \exp \left(-0.2 \sqrt{\frac{1}{20} \sum_{i=1}^{20} x_i^2} \right) - \exp \left(\frac{1}{20} \sum_{i=1}^{20} \cos(2\pi x_i) \right) + 20 + e$	$-30 \leq x \leq 30$
f_2	Cosine mixture	$\sum_{i=1}^{20} x_i^2 - 0.1 \sum_{i=1}^{20} \cos(5\pi x_i) + 2$	$-1 \leq x \leq 1$
f_3	Ellipsoidal	$\sum_{i=1}^{20} (x_i - i)^2$	$-20 \leq x \leq 20$
f_4	Exponential	$1 - \exp(-0.5 \sum_{i=1}^{20} x_i^2)$	$-1 \leq x \leq 1$
f_5	Griewank	$1 + \frac{1}{4000} \sum_{i=1}^{20} x_i^2 - \prod_{i=1}^{20} \cos \left(\frac{x_i}{\sqrt{i}} \right)$	$-600 \leq x \leq 600$
f_6	Rastrigin	$10n + \sum_{i=1}^{20} [x_i^2 - 10 \cos(2\pi x_i)]$	$-5.12 \leq x \leq 5.12$
f_7	Rosenbrock	$\sum_{i=1}^{19} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$-30 \leq x \leq 30$
f_8	Zakharov	$\sum_{i=1}^{20} x_i^2 + (\sum_{i=1}^{20} \frac{i}{2} x_i)^2 + (\sum_{i=1}^{20} \frac{i}{2} x_i)^4$	$-5.12 \leq x \leq 5.12$

表 3.10 数学的多峰性関数（解析解）

f	x^*	$f(x^*)$
f_1	$[0, 0, \dots, 0]$	0
f_2	$[0, 0, \dots, 0]$	0
f_3	$[1, 2, \dots, 20]$	0
f_4	$[0, 0, \dots, 0]$	0
f_5	$[0, 0, \dots, 0]$	0
f_6	$[0, 0, \dots, 0]$	0
f_7	$[1, 1, \dots, 1]$	0
f_8	$[0, 0, \dots, 0]$	0

表 3.11 各手法の設定パラメータ（20 次元テスト関数）

	探索点 (個体数)	計算回数 (世代数)	各手法独自の 設定パラメータ
提案手法	30	100000	$\omega = 2\pi/10^3, G = 10^4$
PSO	30	100000	$c_1 = c_2 = 2.0$ $w = 0.9 \rightarrow 0.4$
GA	30	100000	$P_c = 0.02$

3.4.3 適用結果（テスト関数）

結果を表 3.12 に示す。評価値であるテスト関数の値のみを示し、各手法との比較を行っていく。各関数の平均値と最良値について、各手法の値と比べて最小のものを太字で示した。提案手法は平均値と最良値ともに、8つの関数のうち7つにおいて優位になる解を得ることが出来た。これらの関数は前節で解いた Schwefel 関数のように多峰性を持つ関数である。優位に解探索を進めることが出来た理由はコリオリ力を模擬した座標回転によって、探索点が大きな渦状に強制的に広範囲を移動することが出来たためである。急速に最適解に収束するのではなく遠回りすることで大域的最適解の峰に到達する確率を上げているのである。また、設計変数空間の端を連結することで座標面をトーラス面とすることで設計変数空間の端をなくしている。これは探索点の移動可能範囲の壁を無くしたのと同じなので、コリオリ力による探索点の強制移動において壁にぶつかる事が無い。この提案手法において、コリオリ力の座標移動のアルゴリズムと変数空間をトーラスにすることは相性の良い組み合わせである。また、その性能の相乗効果は結果にも反映されていて、表 3.12 の優位な結果を見ても

分かる。この節では表 3.12 のように最終的な評価関数値しか示さないが、探索点の移動は図 3.4 のように縦横無尽に変数空間を移動する結果を得ている。宇宙ジェットを模擬した局所解脱出アルゴリズムも作動した。特に提案手法における f_3 の平均値 10^2 のオーダーと最良値 10^{-4} のオーダーとの差が大きいので、この局所解脱出アルゴリズムが有効に働いたと推察される。

表 3.12 各手法の計算結果 (20 次元テスト関数)

		GA	PSO	提案手法
f_1	Average	$1.184 \times 10^{+1}$	2.057×10^0	5.929×10^{-3}
	Best	9.864×10^0	7.848×10^{-1}	7.510×10^{-4}
f_2	Average	1.774×10^0	3.048×10^{-2}	2.363×10^{-7}
	Best	1.387×10^0	5.862×10^{-4}	1.173×10^{-8}
f_3	Average	$1.086 \times 10^{+3}$	9.905×10^{-2}	$9.669 \times 10^{+2}$
	Best	$8.145 \times 10^{+2}$	1.556×10^{-1}	6.667×10^{-4}
f_4	Average	1.400×10^{-1}	1.019×10^{-3}	8.810×10^{-9}
	Best	6.681×10^{-2}	2.363×10^{-5}	1.289×10^{-9}
f_5	Average	$2.839 \times 10^{+1}$	5.165×10^{-2}	9.443×10^{-4}
	Best	$1.345 \times 10^{+1}$	1.303×10^{-4}	2.427×10^{-4}
f_6	Average	$1.280 \times 10^{+2}$	2.162×10^0	8.431×10^{-5}
	Best	$9.850 \times 10^{+1}$	1.044×10^{-2}	1.234×10^{-5}
f_7	Average	$6.755 \times 10^{+5}$	$1.793 \times 10^{+2}$	$1.867 \times 10^{+1}$
	Best	$2.952 \times 10^{+5}$	$2.381 \times 10^{+1}$	$1.858 \times 10^{+1}$
f_8	Average	$2.147 \times 10^{+1}$	4.598×10^{-6}	8.769×10^{-7}
	Best	$1.037 \times 10^{+1}$	2.730×10^{-7}	$1.129 \times 10^{e-7}$

PSO については関数 f_3 の平均値と f_5 の最良値について優位に立っている。今回は単純な PSO アルゴリズムで最適化計算を行ったが、 f_8 では最大で 10^{-7} のオーダーとなる解を得ている。 f_3 では平均値において提案手法より PSO が良い結果を得ている。30 回の計算試行をしているので的確に解析解に近づいている探索履歴であったと考えられる。急速に最良解に収束した方が良いのか、遠回りした方が良いのかは適用問題に依存するので、今回の結果を踏まえて提案手法のパラメータ調整の指針の参考にしたい。GA については今回は良い結果を得られなかった。個体数の不足などが考えられる。このことから提案手法が有効なアルゴリズムを備えている事が分かる。しかし、 f_3 では平均値の結果に大きな差がある。提案手法は 10^2 のオーダーで、提案手法の結果は局所解に収束してしまっているといえる。

重力定数 G と回転量 ω のパラメータ調整のバランスや局所脱出アルゴリズムの改善などが必要である。提案手法の得意な適用問題分野の調査や手法独自のパラメータである重力定数 G 、回転量 ω の最適化計算への感度解析などが必要である。

3.4.4 提案手法のみ 100 次元と 1000 次元に拡張

前節における問題設定をさらに高次元にする。設計変数を 100 次元と 1000 次元の場合について最適化計算する。本節では提案手法のみの結果を示す。この最適化計算結果から提案手法の高次元空間への対応性をみる。この節でも目的関数のみの結果を示す。提案手法の設定パラメータを表 3.13 に示す。乱数を変化させて同一問題について 100 回最適化計算を行い、平均値と最良解について表に示した。前節の数学的多峰性関数の 8 つの関数最適化問題について設計変数を 100 次元と 1000 次元にした場合の結果を表 3.14 に示す。

表 3.13 提案手法の設定パラメータ（高次元テスト関数）

探索点 (個体数)	計算回数 (世代数)	手法独自の 設定パラメータ
20	50000	$\omega = 2\pi/100, G = 10^4$

100 次元についての結果をみると提案手法では最良値で 10^{-3} 以下の計算解を、8 つの関数のうち 6 つにおいて最良解として得ている。前節での結果を踏まえた時に本提案手法が容易に多次元に拡張できていることが分かる。平均値と最良解値を比較するとオーダー値は同一か最良解が一桁低いという結果になっている。平均値が最良解に近いということは探索点群の全体が最良解近郊に存在している。近傍探索である重力方程式のアルゴリズムが有効に実装されているといえる。また、関数式 f_5 においては、平均値のオーダーが 1 なのに対して最良解は 10^{-4} となっており最良解が飛びぬけて良い。他の関数式の結果では平均値と最良解値の差倍率が 1/10 なのを考慮すると、 10^{-4} のオーダー差は本提案手法の宇宙ジェットのアルゴリズムによるものだと考えられる。宇宙ジェットは局所解に収束してしまった探索点群を脱出させるためのアルゴリズムであり、100 次元空間の最適化進行でも有効に働いたといえる。よって、宇宙ジェットによる脱出アルゴリズムは本提案手法に必要なアルゴリズムであるといえる。

次に 1000 次元に拡張した場合についてみると、最良値で 10^{-3} 以下の計算解を 8 つの関数のうち 6 つにおいて得ている。この 6 つの関数において平均値と最良値のオーダー値を比較すると 10^{-2} から 10^{-4} となっており差が開いている。1000 次元の解空間において最適解に近づくことは困難であるといえる。本計算において探索点が 20 点であることを考えると大域的探索において最良解が偶然的に発見され、その最良解に探索点群が局所探索過程で近づく

く方向に速度ベクトルが発生するが、1000 次元の変数の最適解との誤差が目的関数値に大きく反映してしまうので探索点が空間の多峰に阻まれてしまったと考えられる。本提案手法には大域的探索過程でのコリオリ力による移動アルゴリズムを有している。今回の計算では 8 つの関数においてすべて同じ回転量パラメータ $\omega = 2\pi/100$ を与えたが問題の次元数や解空間の特徴が事前に分かれば、回転量パラメータ ω と重力定数 G のパラメータ調整により探索点群の収束性を高めることはできる。

表 3.14 提案手法における高次元テスト関数の計算結果

		100 次元	1000 次元
f_1	Average	1.76×10^{-3}	6.52×10^{-4}
	Best	9.76×10^{-4}	1.81×10^{-4}
f_2	Average	2.82×10^{-7}	5.11×10^{-7}
	Best	8.01×10^{-8}	9.14×10^{-9}
f_3	Average	$1.79 \times 10^{+4}$	$3.33 \times 10^{+8}$
	Best	$1.21 \times 10^{+4}$	$3.30 \times 10^{+8}$
f_4	Average	1.19×10^{-8}	1.68×10^{-8}
	Best	1.69×10^{-9}	9.74×10^{-11}
f_5	Average	7.47×10^0	9.91×10^{-3}
	Best	4.81×10^{-4}	1.36×10^{-5}
f_6	Average	1.02×10^{-4}	1.92×10^{-4}
	Best	1.31×10^{-5}	8.94×10^{-6}
f_7	Average	$9.92 \times 10^{+1}$	$9.99 \times 10^{+2}$
	Best	$9.87 \times 10^{+1}$	$9.99 \times 10^{+2}$
f_8	Average	2.40×10^{-6}	2.92×10^{-4}
	Best	3.67×10^{-7}	2.32×10^{-8}

第4章 宇宙探査機へ応用

4.1 はじめに 概要および目的

宇宙機の機体構成と軌道計画は別に考えられることが一般的である。しかし、推力や打ち上げ日時と軌道計画には密接な相関関係が存在する。バラバラに計画を立てる事はロスが生じる。そこでミッション計画を円滑に進めるために計画初期段階での計画の方向性を与えるためにはミッションの指針となるものが必要である。この研究では初期段階での指針を与えるための材料となり得る統合的な複合領域最適化の手法を示す。複合領域最適化とは異なる分野にまたがっているサブシステム同士を統合して最適化を行うことである。一連の設計過程において、個別に実施されていた複数の領域の設計を初期の設計段階から統合して扱い、従来専門の異なる設計担当間で往復していた設計プロセスの無駄を除き、全体としてよりバランスのとれた設計を短時間に実現する上で極めて有用である。異分野を同時に扱うので問題設定が複雑であり、また考慮するシステムが巨大になりやすいという特徴を持つ。システム設計をさらに効率的に進めるために、構造だけではなく、制御、流体など他分野に関わる領域をも考慮して最適化を進める複合領域の最適設計が、航空宇宙分野を中心に 1990 年代から盛んになってきた。従来の航空機設計では、空気力学、構造、制御の順に設計を行ってきたが、これらの分野間のバランスが問題となり、旅客機全体として順に設計を行なった場合、必ずしも最適であるとはいえない。複合領域の最適設計は、これらの複数の領域を同時に考えて最適設計を目指すものである。たとえば、図に示す超音速旅客機の設計において、主翼の翼型および翼平面形の形状を決定する設計変数に対して、空力性能および構造強度を考慮して最適設計を行なっている。その他にも、空気力学、熱力学、構造力学、制御工学など多分野の領域を同時に最適化するモデル化など、多くの研究が行われている。並列計算機の発展により、このように複数の分野を統合して最適設計を行う研究が広がっている。大規模計算を伴う場合が多く、問題を解くことが非常に困難である。

宇宙観測や宇宙利用の技術が進歩し宇宙そのものが身近になりつつある現在、宇宙輸送システムの確立が求められている。しかし、宇宙機の設計は構造、推進、軌道、制御などのサブシステムが存在し強い相関関係がある。そのために一つの機体を設計するためには多大な労力を伴う。そのために、それらを統合したシステム全体を最適設計する必要が出てくる。本研究では宇宙機のプロトタイプ設計段階として、深宇宙探査機を考える。軌道計画として、地球か

ら火星へフライ・バイする宇宙機システムを考える。機体構成要素としては総質量の中でペイロード、電源部、推進機部などへの質量分配や電力分配の割合を決める。機体構成からスラスタの推力や重量が決まり、軌道を計算する。これらの設計要素を用いて最適化を行う。また目的関数を複数設定することにより多目的最適化問題とした。多目的最適化とは、複数の目的関数を同時に最適化することである。一般に最適化とは一つの評価基準に対する最適化を行うことがほとんどである。しかし、現実が存在する様々な最適化問題は多くの場合、複数の評価基準を同時に考慮する必要がある。例えばある工業製品を設計する場合、これを多目的最適化問題であると考え、性能、価格、外観、重量といった複数の評価ポイントが目的関数となり得る。そしてこれらの目的関数は多くの場合、トレードオフの関係にあり、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を得ることになる。設計者はパレート解集合の分布を見ることによって、その目的関数間のトレードオフを把握することが出来るようになり、これが多目的最適化の重要な役割となる。

最適化手法には本論文における提案手法を用いて、多目的最適化問題を効率的に解く。機体構成と軌道計画の異なる分野にまたがって設計変数を動かしているのも複合領域最適化にもなっており、複雑な最適化問題に対しても有効であることを示す。以上のように最適化問題を設定し、最適化計算を行うことで最適解集合を求める。また比較のために粒子群最適化手法 (PSO) と遺伝的アルゴリズム (GA) でも同一の問題設定を解き結果を比較する。

4.2 無人宇宙探査機の惑星間軌道計画の最適化



図 4.1 はやぶさ (無人宇宙探査機)^[42]

ここでは単一目的の場合について最適化計算を行う。予想される最適解が得られるか検証するためである。目的関数はミッション期間を最小化する最適化問題とした。以下に機体重

量の計算方法と軌道計画について述べる。その後に最適化手法を適用した結果を示す。

4.2.1 宇宙機の機体構成モデル^[44]

イオンスラスタを搭載する宇宙機モデル

ここで考える宇宙機モデルは電気推進機であるイオンスラスタを用いて動力航行をすることを想定した機体構成である。宇宙機の機体構成は簡易推算式を用いて計算される。電力の確保は太陽電池パドルによるものとする。電力は太陽からの距離に依存し、式 4.1 に従うものとする。飛行中は常に MPD スラスタを稼働させている状態とし、推進機の寿命などは考慮していない。また、宇宙機のペイロード質量は 35[kg] として固定している^[27]。

$$P(r) = P_0 \left(\frac{r_0}{r} \right)^2 \quad (4.1)$$

宇宙機と太陽との距離 r 、初期全電力 P_0 、地球平均公転半径 r_0 である。次に宇宙機の質量と電力の簡易推算モデル式を示す。以下の式では質量 M 、電力 P と表す。 ΔV_{EP} は飛行時間から計算される。

・ 共通系

$$M_{COM} = 116 [\text{kg}], P_{COM} = 116 [\text{W}] \quad (4.2)$$

・ データレコーダ系 (設計値 : $V_{DR} = 64[\text{Mbit}]$)

$$M_{DR} = 0.01 \times V_{DR} [\text{kg}], P_{DR} = 0.02 \times V_{DR} [\text{W}] \quad (4.3)$$

・ 通信系 (設計値 : $R = 2.5[\text{AU}], B = 8[\text{kbps}], D = 1.6[\text{m}]$)

$$\begin{aligned} M_{HGA} &= 4.0 \times D^2 [\text{kg}] \\ P_{TX} &= 5.4 \times 10^{-3} \times \frac{BR^2}{D^2} [\text{W}] \end{aligned} \quad (4.4)$$

・ 姿勢制御系 (設計値 : $\Delta V_{CR} = 400[\text{m/s}], I_{CR} = 290[\text{s}]$)

$$\begin{aligned} M_{FL} &= M_{TOT} \left[1 - \exp \left(\frac{-\Delta V_{CR}}{g I_{CR}} \right) \right] [\text{kg}] \\ M_{RCS} &= 2.0 \times M_{FL}^2 [\text{kg}] \end{aligned} \quad (4.5)$$

・電気推進系 (設計値: $I_{SP} = 3000[\text{s}]$)

$$M_{PP} = M_{TOT} \left[1 - \exp \left(\frac{-\Delta V_{EP}}{g I_{SP}} \right) \right] [\text{kg}]$$

$$M_{EP} = 0.05 \times F [\text{kg}]$$
(4.6)

・バッテリー系 (設計値: $\tau = 3[\text{hours}]$, $V_{BUS} = 50[\text{V}]$)

$$M_{BAT} = \frac{0.54 (P_{COM} + P_{DR} + P_{TX}) \tau}{V_{BUS}} [\text{kg}]$$
(4.7)

・太陽電池系 (設計値: $L = 1.3[\text{AU}]$)

$$M_{SCP} = \frac{L^2 P_0}{38.2} [\text{kg}]$$
(4.8)

・構造系

$$M_{STR} = 0.094 \times M_{TOT} [\text{kg}]$$
(4.9)

・全機質量

$$M_{TOT} = M_{COM} + M_{DR} + M_{HGA} + M_{FL} + M_{RCS}$$

$$+ M_{PP} + M_{EP} + M_{BAT} + M_{SCP} + M_{STR} + M_{PL} [\text{kg}]$$
(4.10)

質量と電力の分配設計

電気推進を宇宙システム全体の一要素として捕らえ、他のサブシステムと比較しながら質量・電力の観点から俯瞰してみる。電気推進を含めたサブシステムの質量や電力特性を用いて軌道設計を行えば、より具体的な宇宙機概念設計が実施できる。また、電気推進と不可分な電源システムの発生電力と質量の関係を導入すれば、式を用いて、具体的な最適比推力を算定できる。

これまで主に科学観測や探査の分野で利用されてきた宇宙機の設計データから、構成要素の分類とそれぞれの特性値を明らかにして簡便な数式として表す努力がなされている。特性値は技術革新により年々向上するので、打ち上げ年次を関数に組み込んだ定量化モデルの要点を表にまとめる [44][28][29][34]。

宇宙機の質量、電力を推算するには、まず以下に示すオプションを選択し、項目ごとの質量と消費電力を合算する。

1. 搭載ミッション機器の質量 M 、電力 P を決定する。
2. 宇宙機方式を非スピン型深宇宙探査機とする。これによりセンサやリアクションホイールを含む姿勢制御、電源、通信、コマンドデータ処理系、熱制御系の共通系質量 M 、電力 P 、が決まる。
3. データレコーダの質量 M 、電力 P は書き込み容量 V の関数としている。
4. 通信のための消費電力 P は、伝送距離 R 、伝送レート B 、HGA パラボラアンテナの直径 D の関数として与える。 D より HGA 質量 M も決定される。技術革新を想定し、打ち上げ年次も独立変数としている。人工衛星の場合 HGA は不要であろう。この場合 P の算出には $D = 1 \text{ m}$ を代入する。
5. スピン型深宇宙機の場合はデスパン部を設ける場合がある。
6. リアクションホイールのアンローディング用のアクチュエータとして、小型人工衛星の場合には MTQ が利用できる。それ以外では RCS スラスタを用いる。制御量 V は 1 年当たり 10 m/s 、比推力 I は 200 秒が標準的である。軌道制御にも利用する場合に比推力 I を 300 秒とするオプションもある。制御量 V と比推力 I から、燃料質量 M と RCS 機構質量 M が決まる。
7. 電気推進を用いる場合には、制御量 V と比推力 I から推進剤質量 M 、電気推進機構質量 M 、消費電力 P が決まる。定式化については後述する。
8. バッテリーの質量 M は、供給電力 P 、作動電圧 V 、供給時間と打ち上げ年次の関数とした。一般的に電気推進をバッテリー駆動することはないので、本項の算出に電気推進の電力は含めない。
9. 全消費電力 P と太陽距離 L の関数として SCP 質量 M が決まる。人工衛星の場合 $L=1\text{AU}$ である。
10. 全質量 M から構造系質量 M が決まる。

打ち上げ年次の関数とした要素は、あまり後年を代入すると質量が負になり不条理となるので注意を要する。燃料や推進剤、構造系質量を得るのに M を必要とするために、初期値として M や P を与えて計算を始め、繰り返し計算により収束値を得る。

表 4.1 宇宙機の質量に関する簡易推算式

サブシステム系	入力	質量 [kg]
共通系	-	$M_{COM} = 116$
データレコーダ	V_{DR} [Mbit]	$K_{DRW} = 0.01[kg/Mbit]$ $M_{DR} = K_{DRW}V_{DR}$
通信系	$D[m]$	$K_{HGA} = 4.0[kg/m^2]$ $M_{HGA} = K_{HGA}D^2$
RCS	$M_{TOT}[kg]$ $\Delta V_{CR}[m/s]$ $I_{CR}[s]$	$M_{FL} = M_{TOT}[1 - \exp(-\Delta V_{CR}/gI_{CR})]$ $M_{RCS} = K_{RCS}(M_{FL})^{2/3}$ $K_{RCS} = 2.0[kg^{1/3}]$
電気推進	$M_{TOT}[kg]$ $\Delta V_{EP}[m/s]$ $I_{EP}[s]$	$M_{PP} = M_{TOT}[1 - \exp(-\Delta V_{EP}/gI_{EP})]$ $M_{EP} = K_{EPW}F + M_{EPO}$
バッテリー	$P_{TOT}[W]$ $P_{EP}[W]$ $V_{BUS}[V]$ $\tau[hour]$ $Y[year]$	$K_{BAT} = 1.0 - 0.027(Y - 1986)[kg/AH]$ $M_{BAT} = \frac{K_{BAT}(P_{TOT} - P_{EP})\tau}{V_{BUS}}$
太陽電池	$P_{TOT}[W]$ $L[AU]$ $Y[year]$	$K_{MT} = 1.0$ $K_{SCP} = 20.0 + 1.07(Y - 1986)[W \cdot AU^2/kg]$ $M_{SCP} = \frac{K_{MT}L^2P_{TOT}}{K_{SCP}}$
構造系	$M_{TOT}[kg]$	$K_{STR} = 0.295 - 0.0118(Y - 1986)$ $M_{STR} = K_{STR}M_{TOT}$
ミッション系 (ペイロード)	-	$M_{PL} = 35.0$

4.2.2 惑星間軌道計算モデル

軌道計画 [31][32][33][41]

ミッションは地球と火星の往復するものである。軌道計画は大きく分けて3つのフェーズから成る。1つ目は地球から火星へ向けて電気推進機を用いて宇宙機を増速させるフェーズ、2つ目は火星でのランデブーを行うフェーズ、3つ目は火星から地球へ帰還するために電気推進機を用いて減速させるフェーズである。これら3つについて以下で詳細を述べる。制約条件としては2014年1月1日から20年以内に出発する条件を与える。地球を出発し火星をランデブーする期間は5年以内とする。

表 4.2 宇宙機の電力に関する簡易推算式

サブシステム系	入力	電力 [W]
共通系	-	$P_{COM} = 116$
データレコーダ	V_{DR} [Mbit]	$K_{DRP} = 0.02[W/Mbit]$ $P_{DR} = K_{DRP}V_{DR}$
通信系	$B[bps]$ $R[AU]$ $D[m]$ $Y[year]$	$K_{TX} = 7.7 - 0.207(Y - 1986)$ $K_{LK} = 0.0013[W \cdot m^2 / bps \cdot AU^2]$ $P_{TX} = K_{TX}K_{LK}BR^2/D^2$
RCS	-	-
電気推進	F [N]	$P_{EP} = K_{EPP}F$
バッテリー	-	-
太陽電池	-	-
構造系	-	-
ミッション系 (ペイロード)	-	$P_{PL} = 0.0$

地球から火星への軌道

地球を出発し火星まで電気推進機を用いて増速する軌道を考える。本研究では太陽と宇宙機の2体問題として最適化問題設定を行っているので、地球と火星の重力による影響は考慮していない。

ランデブー軌道

火星へ到達した時に化学推進を噴射することで火星と宇宙機の相対速度をゼロとする。その後、宇宙機は火星と同じ軌道をとるものとする。ランデブー時の宇宙機は火星の衛星軌道をとっていると仮定しており、スラスターは停止状態とする。このランデブー期間を最適化問題上の設計変数とする。このランデブー期間を変更することで宇宙機は帰還のタイミングを計ることになる。

帰還軌道

帰還においては電気推進機を逆噴射して減速させる。火星軌道から徐々に減速させ、地球に帰還させる。その際に地球での化学インパルスによる減速は行わず、地球との相対位置をゼロとする条件を加えるのみとする。帰還軌道については相対位置ゼロとする条件で最適化

の数値計算を行う。地球上での再突入は空気による減速を考慮して燃料を用いたインパルス噴射は考慮しないものとする。

宇宙機の運動方程式

宇宙機と天体の運動は三次元とする。また、太陽と宇宙機の二体問題として取り扱う。推力方向は常に速度ベクトル方向になるように推力方向角を設定する。運動方程式の数値計算にはルンゲ・クッタ法を用いた。地球と火星の位置と速度はケプラー軌道要素から楕円軌道として計算した。

$$\frac{d}{dt} \begin{pmatrix} x \\ y \\ z \\ u \\ v \\ w \\ m \end{pmatrix} = \begin{pmatrix} u \\ v \\ w \\ -\frac{\mu x}{r^3} + \frac{K_p P}{m} \\ -\frac{\mu y}{r^3} \\ -\frac{\mu z}{r^3} \\ -\frac{K_p P}{g I_{SP}} \end{pmatrix} \quad (4.11)$$

位置 x, y, z 、速度 u, v, w 、機体質量 m 、電力と推力の特性係数 $K_p = 0.03[\text{mN/W}]$ とした。

4.2.3 最適化問題設定

以上の条件においての問題設定をここでまとめる。

目的関数

単一目的最適化としており、目的関数はミッション期間 T_{flight} の最小化としている。評価値の計算を行い最適化計算を行う。この際に飛行終了時点での宇宙機位置と目標天体位置との相対距離をペナルティ法により評価値を改悪する手法を加えている。ペナルティ法については第2章で説明している。評価値 I を以下に示す。ペナルティ法による改悪係数を $C_{penalty}$ 、宇宙機の火星到達時の火星との相対距離を R_{mars} 、宇宙機の地球到達時の地球との相対距離を R_{earth} とする。

$$I = T_{flight} + C_{penalty} \times (R_{mars} + R_{earth}) \quad (4.12)$$

今回の計算では $C_{penalty} = 10^{-5}$ とした。大まかな各項目のオーダー値は、 T_{flight} が 10^3 、 $C_{penalty} \times (R_{mars} + R_{earth})$ が 10^1 となっている。

設計変数

機体構成は固定しており設計変数はこの問題設定では存在していない。軌道計画変数に関しては惑星間軌道の計画要素である出発日時 T_{date} とランデブー期間 T_{rendez} がある。

以上をすべて足すと 2 個の設計変数がある。

制約条件

問題設定時に定義した制約条件を以下に示す。

- ・ 出発日時 T_{date} は 2014 年 1 月 1 日から 20 年以内に出発

$$T_{date} \leq 20[\text{years}] \quad (4.13)$$

- ・ ミッション期間 (飛行時間) T_{flight} は

$$T_{flight} \leq 20[\text{years}] \quad (4.14)$$

- ・ ランデブー期間 T_{rendez} は

$$T_{rendez} \leq 5[\text{years}] \quad (4.15)$$

となる。これらは宇宙機の軌道計画に関する制約条件である。

最適化手法の設定パラメータ

最適化手法に関する設定パラメータと制約条件を表に示す。

表 4.3 最適化手法の設定パラメータ (無人宇宙探査機)

	提案手法	PSO	GA
探索点数 (個体数)	20	20	20
最大計算回数 (世代数)	1000	1000	20000
固有パラメータ	重力定数 $G = 10^4$ 回転量 $\omega = 0.001$	収束係数 $w = 0.9$ から 0.1 速度係数 $c_1 = c_2 = 2.0$	突然変異率 0.02
解空間に対する 速度制限	$1/100$	$1/100$	-

4.2.4 最適化計算結果

目的関数の相関関係図と設計変数の解空間図を図示する。図 4.2 は本節における無人宇宙探査機の目的関数であるミッション期間と宇宙機全機重量との関係図を示したものである。ミッション期間を最小化する問題であるので図中のポイントが下側に移動するほど評価値が良いことになる。青い四角のポイントが提案手法における初期世代である。赤い三角のポイントが最終世代における解集合である。黄色い丸型のポイントが全計算履歴においての最良解である。初期世代から最終世代までの計算内での一番評価値が高い探索点ということである。最良解の具体的な数値は設計変数である出発日時は 5132 日、ランデブー期間は 62 日、目的関数であるミッション期間は 1336 日となった。機体質量は 529[kg] であった。はやぶさの諸元^{[38][39][40]}を元に問題設定を行っており、得られた最良解もはやぶさの機体構成と類似した解を得られている。最適化計算が良好に実行されたことを表している。さらにこの節での最適化計算では初期個体群に制限を設けておりミッション期間が 3000 日前後になるように初期探索点を生成している。初期探索点に制限を設けた理由は提案手法の探索能力の有用性を示すためである。この宇宙機の最適化問題は実行不可能領域が広く飛び飛びに分布している。実行可能領域の方が解空間に対して狭い領域をとっている。このような解空間を持つ最適化問題設定においては局所解にトラップされてしまう可能性が高い。局所解を脱出するアルゴリズムを備えている必要がある。提案手法は局所解脱出アルゴリズムとして宇宙ジェットを模擬した脱出アルゴリズムを用意している。このために探索点群が同一の実行可能領域内に存在した場合でもある探索点が計算不可能領域を飛び越えて新たな解候補領域にたどり着くことが出来る。今回の計算アルゴリズムは、ある範囲内に全探索点の 1/4 が集まった場合にそのグループ内から 2 個体をランダム方向に噴出するものである。ランダム方向に噴出するので目的関数値が悪くなる方向に飛び出す可能性もある。改悪される領域に移動した場合もその後の最適化計算ルーチンでの重力計算で改良される方向に移動する。図 4.2 においては青い四角の初期探索点群が中央にあり、ここから最適化計算ルーチンを実行するたびに目的関数であるミッション期間値を最小する方向に探索点が図の下方に移動する。この計算条件では計算回が 100 回くらいの早い段階である領域に探索点群が収束したので、そこから宇宙ジェットのアルゴリズムが発動し他の解候補群に移動した。事前の計算で図の点線部分に解のグループが存在することが分かっているので、この計算回での最終には 2 段階したの解グループに移動できたことが分かる。図 4.2 と図 4.3 における点線は初期値を制限しない事前の計算において、かつ計算上の乱数の発生値を変化させた計算を数十回行った結果によるものである。この実行可能領域を表した点線を飛び越えて移動している。これによって提案手法が実行不可能領域が多数存在する問題でも有効に最適化計算を実行することが出来る。また、多峰性を有する問題でも峰をジャンプすることで真の最適解に到達

する可能性を有していることが分かる。

図 4.3 は無人宇宙探査機の最適化問題設定における設計変数の関係図を示したものである。目的関数の関係図と同様に点線が事前計算で得られているパレート曲線である。今回の問題設定では 3 段階の解曲線が得られている。この中において提案手法は初期個体群からパレート曲線をジャンプすることで最良解にたどり着いている。図の右下に行くほど目的関数が改良される方向となっている。ミッション期間を最小化する単一目的関数が本節の問題設定なのでランデブー期間を少なくする方向に移動することで目的関数値を下げていく。初期個体群はランデブー期間が 1000 日程度の位置に存在しており、その下のパレート曲線では 350 日の位置にある。最適化ルーチンの繰り返し計算を行うとランデブー期間を短くするように計算が進み、最終的には図の下方にあるパレート曲線上にあるランデブー期間を 50 日程度の解へと到達している。目的関数であるミッション期間は地球から火星への航行期間、火星でのランデブー期間、地球への帰還期間を足したもので表している。機体の構成は固定しているので火星でのランデブー期間が直接的に目的関数に効いてくる。ここではランデブー期間は設計変数として変更することで軌道計画を定めているのでこの設計変数がゼロになるように移動していると理解できる。出発日時は 2014 年 1 月 1 日から何日後に出発するかを示したもので設計変数である。ランデブー期間を最小化するために地球と火星の位置を調整する様に変更されていることとなる。機体構成を固定しており推力が一定なので動力航行時間は出発のタイミングにのみ依存することとなる。目的関数であるミッション期間はランデブー期間の日数にほぼ決定されるので図 4.3 をみると最良解が図の最下限近くのランデブー期間 62 日の領域にまで到達していることを考えると最適解を得ているといえる。これらの結果から本解析モデルと最適化計算が正しく機能していると確認できた。

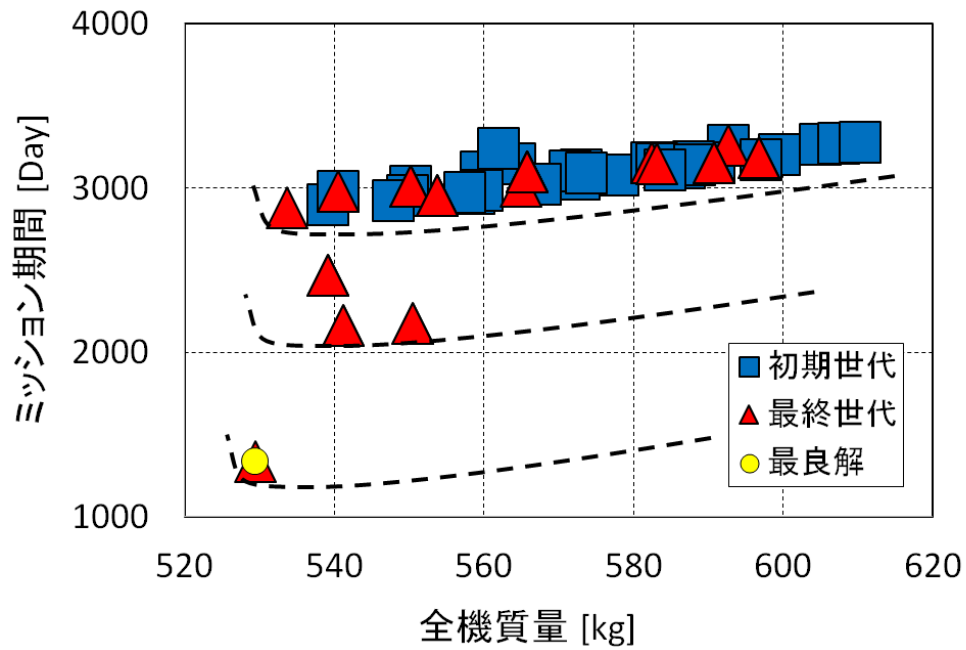


図 4.2 目的関数 (無人宇宙探査機)

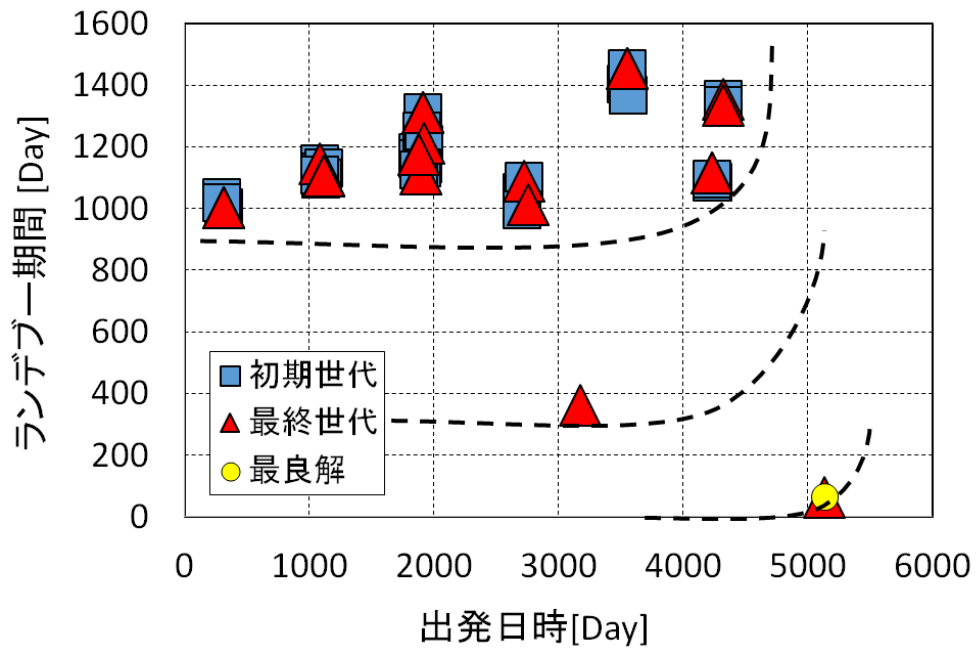


図 4.3 設計変数 (無人宇宙探査機)

惑星間軌道

この節では図 4.2 において示した最良解をピックアップして惑星間軌道などの計算結果を以下に示す。地球から火星への軌道を青い線で示している。青い丸は地球の位置を示している。青い線とつながっている方が出発位置である。赤い三角のポイントが火星到達位置である。灰色の点線がランデブー軌道を表している。赤い線が帰還軌道を表しており、赤い線とつながっている赤い三角が火星出発位置である。黒い点線は内側が地球軌道、外側が火星軌道である。小惑星探査機はやぶさは地球をスイングバイする特殊な軌道をとっているが本計算結果は単純なスパイラル軌道としている。はやぶさの当初の計画ではトータルのミッション期間は 4 年であり、本計算結果のミッション期間は 1336 日で 3.7 年となるので同等の結果を得ている。本節の無人宇宙探査機の最適化計算は提案手法の有効性を示すためのものであるので、小惑星探査機はやぶさを参考とした問題設定をした上でははやぶさと同等の機体構成と軌道を得ていることは最適化計算アルゴリズムが有効に働いているといえる。

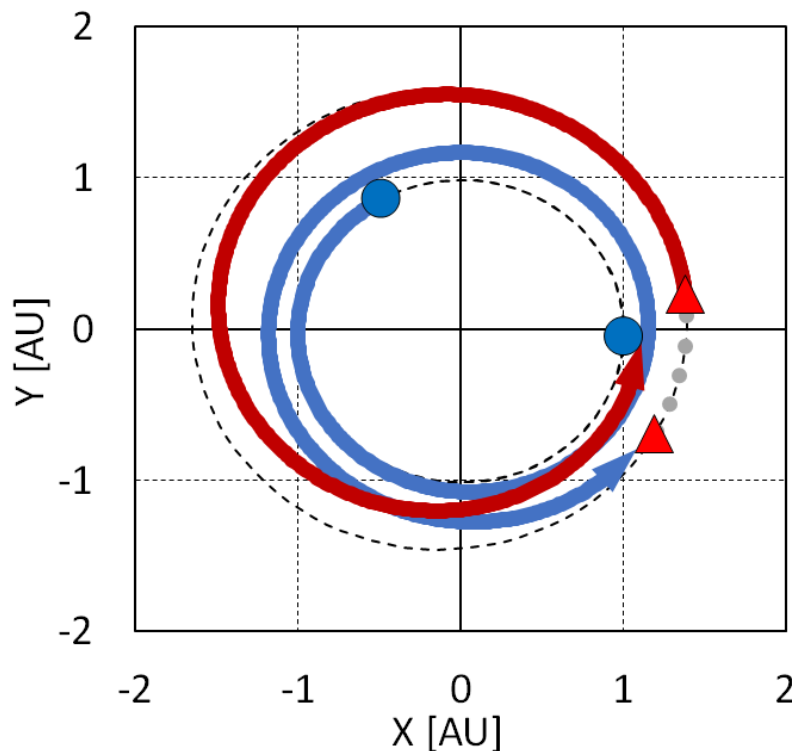


図 4.4 無人宇宙探査機の軌道計画

次に推力値の時間履歴と機体重量の時間履歴である。図 4.5 は探査機の推力値の変化を表すものである。推力値は太陽電池パネルから得られる電力に比例するので、太陽からの距離に依存する。宇宙機の機体構成は固定しており太陽電池パドルの大きさも最適化計算上の探

索点群によって変化しない。多点探索点における太陽電池パドルの大きさはすべて同じである。図 4.5 の縦軸は推力値であり、プラス側は地球から火星への往路の推力値である。推力方向は常に速度ベクトルと同一方向に向いているとする。地球から遠ざかるため推力値は減少している。中央の推力ゼロの区間はランデブー期間である。後半の推力値も太陽間距離に依存しており、速度ベクトル方向と逆向きに常に推力を発生させているのでマイナスの値としている。

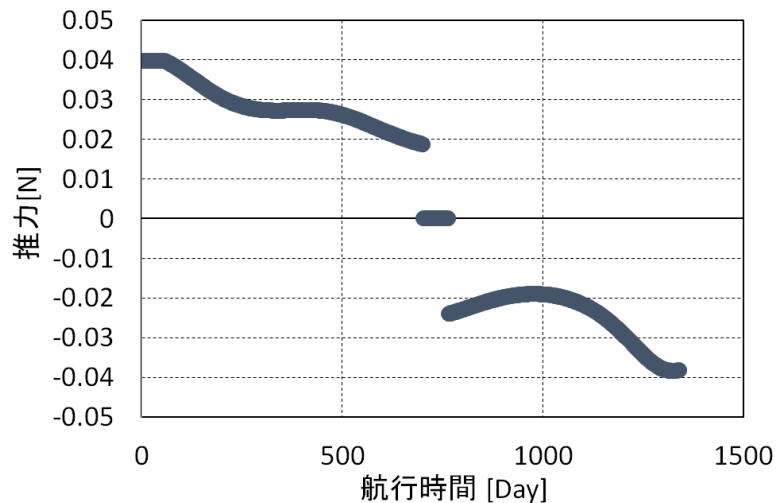


図 4.5 推力値 (無人宇宙探査機)

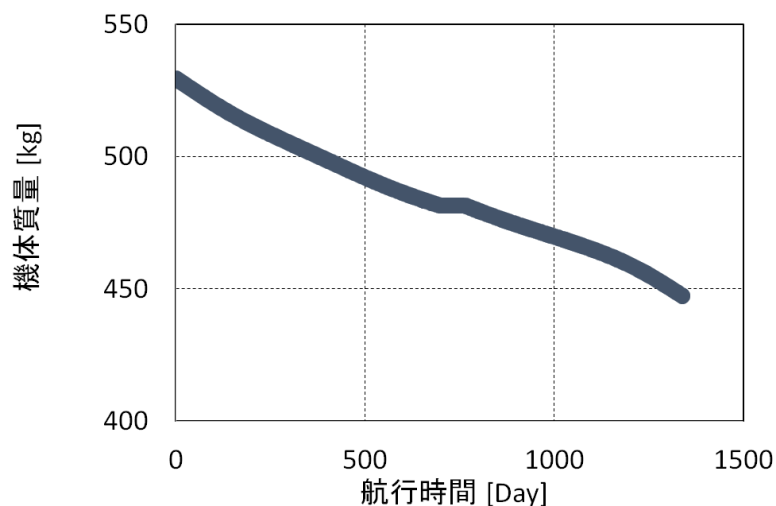


図 4.6 重量値の時間履歴 (無人宇宙探査機)

火星出発位置と太陽位置との距離の関係から推力値は減少してから増加しているが軌道を

考えると正常な変化である。はやぶさの推力値は $30[\text{mN}]$ であり、本計算結果の推力の平均値は $28.6[\text{mN}]$ となり同一のオーダーといえる。

図 4.6 は機体重量の変化を時間履歴で表示したものである。推進剤を徐々に使いながら航行している。計算上は推進剤をピッタリ使い切る量を搭載しており地球に帰還することとしている。中央の機体質量の減少が無い区間はランデブー期間である。また、参考に最終世代の最悪解を示す。最適化計算が有効に働いていることが分かる。最良解では太陽周りの周回数が 2.5 回転であったが、最悪解では約 6 回転であった。図 4.7 は最悪解の軌道である。軌道が重なっているのを分割したものを直下に示す。特にランデブー軌道が 1.7 周回しており帰還のタイミングを合わせるために無駄な待機時間をとられていることが分かる。提案手法の最良解を得られたことは最適化計算の宇宙ジェットアルゴリズムによって実行不可能領域を飛び越えて評価値の良い解群候補にジャンプすることが出来たためである。通常の最適化手法では周回数を事前に指定することで最適解付近の探索領域を制限することで計算を実行可能とすることがあるが、本提案手法は実行不可能領域を飛び越えて新しい解候補にたどり着くことが出来るので未知の解空間探索に適している手法であるといえる。

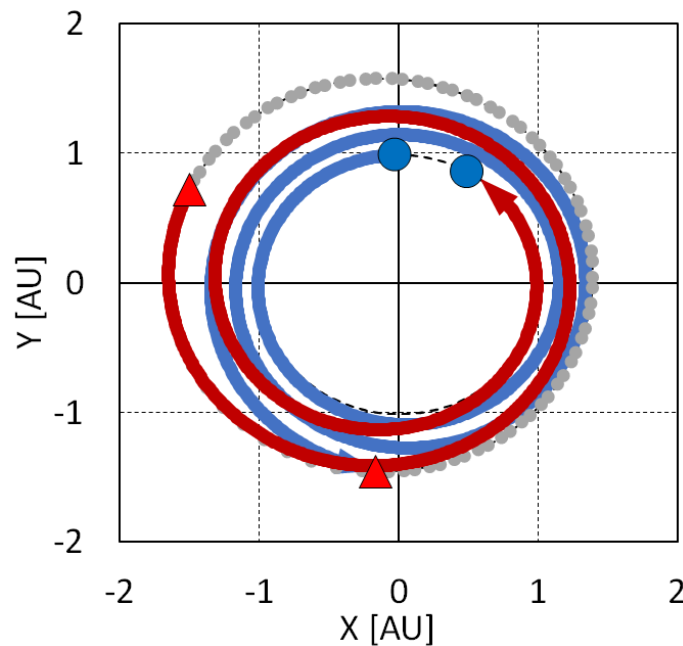


図 4.7 最悪解の軌道計画（無人宇宙探査機）

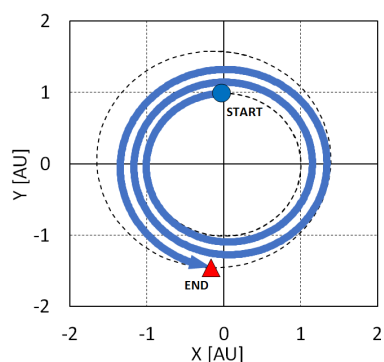


図 4.8 地球→火星の軌道

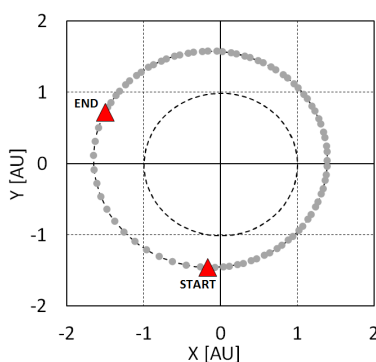


図 4.9 ランデブー軌道

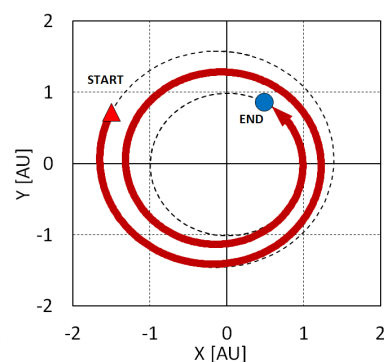


図 4.10 火星→地球の軌道

機体構成

提案手法によって得られた最良解の機体構成結果を表 4.4 に示す。各サブシステムの質量と電力の値を示している。ペイロードは 35[kg] で最適化計算上は固定している。電力値も固定としているので変化しているのは電気推進の推進剤である。ミッション期間によって使う推進剤量が変わるので全機重量も変化する。全機質量は 529[kg] となり、はやぶさは 502[kg] なので軌道計画が一致してはいないがはやぶさとの重量の誤差は 5% であった。

表 4.4 機体構成諸元（無人宇宙探査機）

	質量 [kg]	電力 [W]
ペイロード	35.0	-
共通	116.0	116.0
データレコーダ	0.64	1.28
通信	10.2	106.2
姿勢制御（RCS）	32.3	-
姿勢制御（燃料）	65.0	-
電気推進（機器部）	71.6	1190.6
電気推進（推進剤）	82.1	-
バッテリー	7.25	-
太陽電池	62.6	-
構造系	46.7	-
合計	529.4	1414.0

次に機体重量を基に面積や体積を逆算する。太陽電池パネルなどははやぶさの重量と面積の関係を参考とした。太陽電池パドルの $1[\text{kg}]$ 当たりの面積は $0.257[\text{m}^2/\text{kg}]$ とし、中央のメインユニットの質量体積比は $0.00427[\text{m}^3/\text{kg}]$ として計算した。この変換値をもとに各部の長さを算出し三面図に作画した。

他手法との比較（無人宇宙探査機）

この節では他手法による無人宇宙探査機の最適化計算結果について示す。最適化問題設定は前節の設定と同一とする。各々の手法独自の設定パラメータは表に示す。以下に結果を示す。図列の左側が P S O による最適化結果、右側が G A による計算結果である。

PSO については図 4.12 をみると最終世代が初期世代群の領域から脱出できていない。今回使用した P S O のアルゴリズムは基本的な単純 P S O であり、脱出アルゴリズムを備えていない P S O を用いたために初期世代群から抜け出せなかった。この節での最適化問題設定は初期個体群の生成を制限したので単純 P S O では最適解に到達することが出来なかった。P S O は最良解に急速に収束する性質があるので初期個体群を制限されている状態では最適解に到達することが困難になる。もちろん P S O も解空間と解析解の位置関係や速度ベクトルの向きによって初期世代群を制限されている状態でも大域的な最適解に到達する可能性はある。ただ、今回のような実行不可能領域が広く飛び飛びの領域を有しているときは大域的最適解に到達することはできなかった。このことから局所解にトラップされた場合の脱出アルゴリズムを備えることは必須であるといえる。次に説明する G A は突然変異アルゴリズムが脱出アルゴリズムとなり、本提案手法は宇宙ジェットのアルゴリズムがそれに対応する。GA は図 4.13 をみると初期世代群から脱出していることが分かる。赤い三角のポイントが下方に 4 つ移動していることが分かる。これは突然変異によって個体が移動したものである。突然変異での移動はランダムに位置を移動させるものとして計算している。提案手法によって得られているような大域的な最適解を得られていないが世代数の繰り返し計算回数を増やすことで大域的な最適解に到達する可能性はある。しかしその方法はランダムサーチと変わらないものとなる。P S O と G A の設計変数の関係図と提案手法とを比較する。提案手法の結果をもとに考えると最適解への計算方向は右下の方向に進むことが良いと考えられる。P S O においては初期世代から脱出できておらず最終世代まで繰り返し計算が行われた。G A については初期世代群から脱出できており特にランデブー期間 100 日、出発日時 5000 日付近に解が移動している。最適解付近に移動してはいるが目的関数であるミッション期間に変換すると図の黄色い丸のポイントが最良解になる。G A の目的関数結果をみると最終世代の 4 つのポイントが横並びになっている。この 4 つのミッション期間の最小差は 19 日であり、この差が設計変数空間では出発日時の差が 3000 日と解空間の $1/2$ の領域差として現れ

ている。設計変数差と目的関数差との隔たりがあるために最適化計算の最良解への進行を妨げている要因ともなっている。今回のGAの計算結果では図の右下の領域へは到達することが出来なかった。参考までにPSOとGAの計算結果における最終世代の最良解をピックアップして軌道などの結果を示す。どちらの解ともランデブー期間が長いために全機体質量が20[kg] 前後重くなっている。

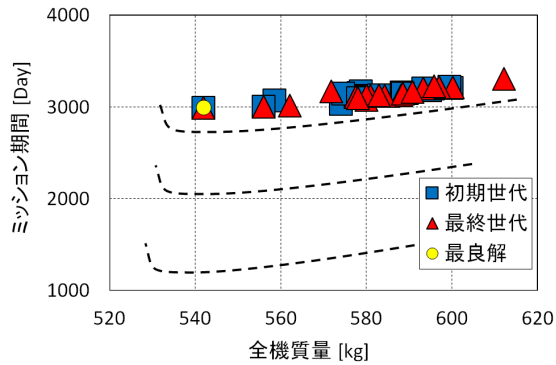


図 4.12 PSO 目的関数結果 (無人)

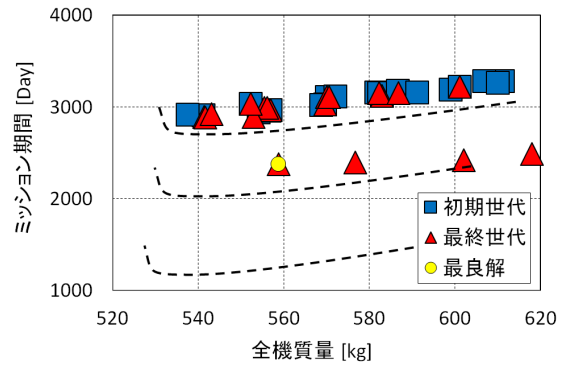


図 4.13 GA 目的関数結果 (無人)

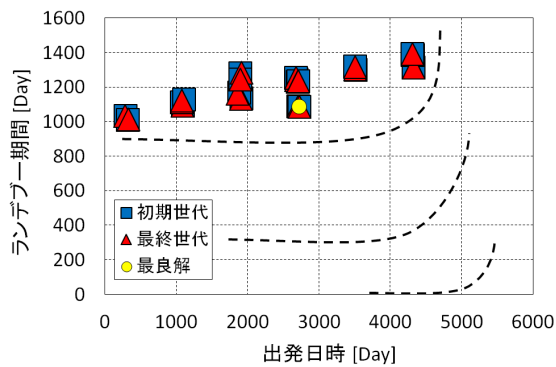


図 4.14 PSO 設計変数結果 (無人)

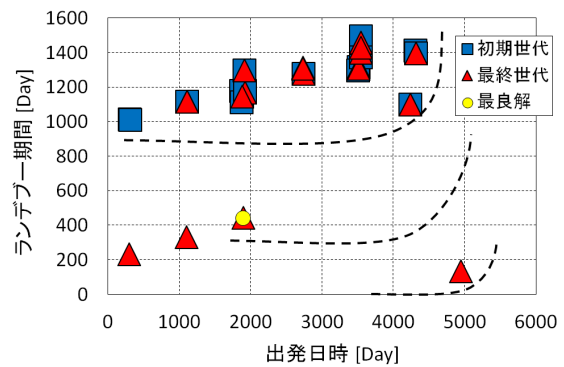


図 4.15 GA 設計変数結果 (無人)

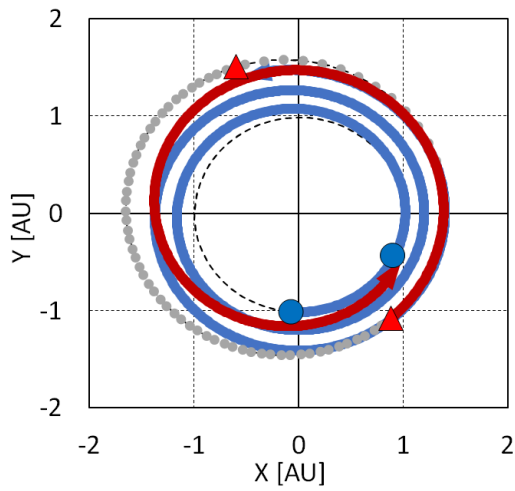


図 4.16 PSO 軌道結果（無人）

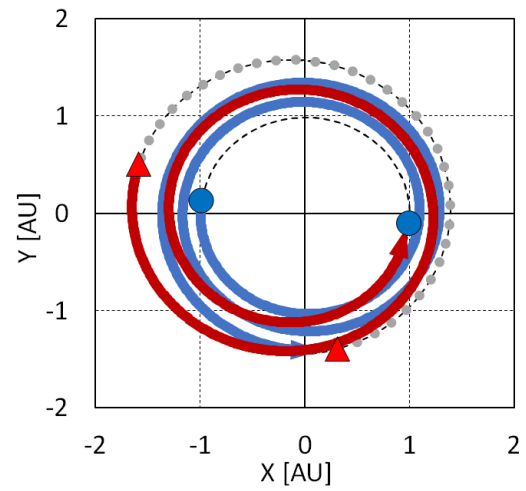


図 4.17 GA 軌道結果（無人）

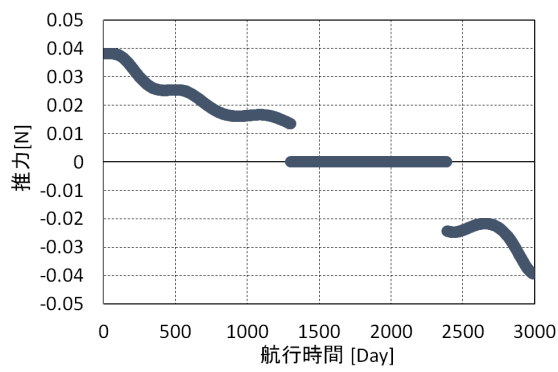


図 4.18 PSO 推力値結果（無人）

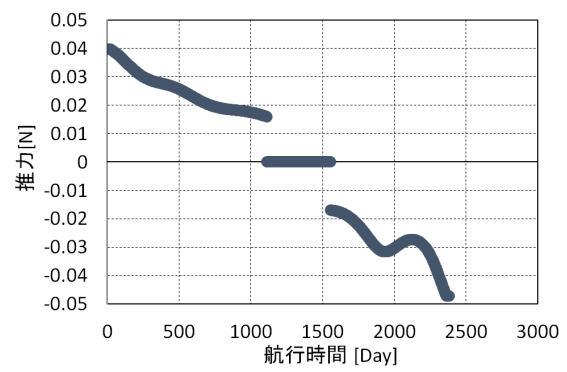


図 4.19 GA 推力値結果（無人）

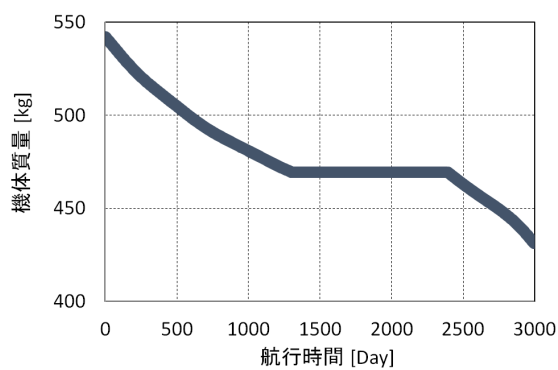


図 4.20 PSO 機体重量結果（無人）

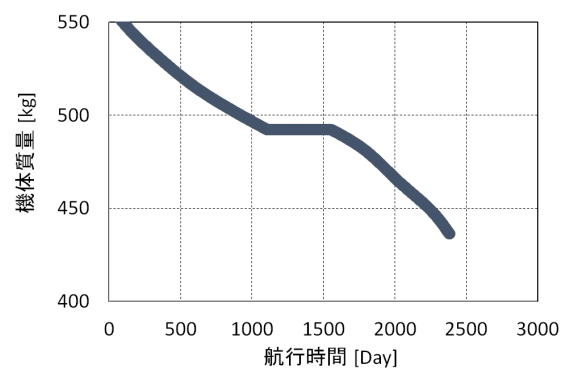


図 4.21 GA 機体重量結果（無人）

表 4.5 PSO 機体諸元結果（無人）

	質量 [kg]	電力 [W]
ペイロード	35.0	-
共通	116.0	116.0
データレコーダ	0.64	1.28
通信	10.2	106.2
姿勢制御（RCS）	32.9	-
姿勢制御（燃料）	66.7	-
電気推進（機器部）	71.6	1190.6
電気推進（推進剤）	110.8	-
バッテリー	7.25	-
太陽電池	62.6	-
構造系	47.9	-
合計	542.0	1414.0

表 4.6 GA 機体諸元結果（無人）

	質量 [kg]	電力 [W]
ペイロード	35.0	-
共通	116.0	116.0
データレコーダ	0.64	1.28
通信	10.2	106.2
姿勢制御（RCS）	33.6	-
姿勢制御（燃料）	68.9	-
電気推進（機器部）	71.6	1190.6
電気推進（推進剤）	122.5	-
バッテリー	7.25	-
太陽電池	62.6	-
構造系	49.4	-
合計	558.7	1414.0

4.3 有人宇宙探査機の機体構成と惑星間軌道の複合領域最適化

前節の無人宇宙探査機をさらに発展させて有人探査機を考える。このときに無人探査機の計算条件にいくつかの新しい条件を加えることで有人探査機の最適化計算に設定を発展させる。

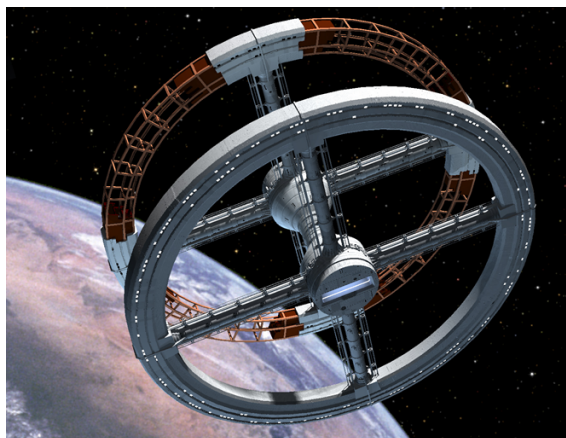


図 4.22 有人宇宙探査機（スピン型）のイメージ図 [43]

居住モジュールと食糧、水、空気を搭載することで有人化を再現する。簡易推算式に居住モジュールの項目を加えることで有人探査機の重量推算を行う。

搭乗員は2人とする。宇宙飛行士を2人の乗組員として搭乗することを仮定する。さらにMPDスラスタを搭載した場合の宇宙機の最適化問題として考えていく。宇宙機の質量計算モデルは簡易推算式を用いる。ここでは多目的の場合について最適化計算を行う。目的関数はミッション期間と機体質量を最小化する最適化問題とした。以下に機体重量の計算方法と軌道計画について述べる。その後に最適化手法を適用した結果を示す。

4.3.1 居住モジュール

居住モジュールはロシアのスベズェタという国際宇宙ステーションの实在の居住モジュールの諸言を参考にしている^[30]。本来はスベズェタは6人で使用するものだが、ここでの最適化問題設定は2人としているのでスベズェタの重量と必要電力を単純に1/3として最適化計算の重量推算を行っている。簡易推算式へ加える項目は以下の式を使う。

食糧、水、空気に関しては人間が一日に使用する平均量を参考とした。探査機に搭載する量は全ミッション期間分とする。一日に使用した量はその日のうちに探査機外に廃棄するものとする。人間が必要とする物質量は以下のとおりである。

$$\begin{aligned} \text{食糧の必要量/日} &= 2.53[\text{kg/day}] \\ \text{水の必要量/日} &= 1.97[\text{kg/day}] \\ \text{空気の必要量/日} &= 0.84[\text{kg/day}] \end{aligned} \tag{4.16}$$

必要量の合計は $5.34[\text{kg/day}]$ となり、一日に廃棄する量も $5.34[\text{kg/day}]$ となる。

4.3.2 質量と電力の簡易推算式

ここでは最適化計算に用いた宇宙機の質量と電力に関する簡易方程式を表にまとめた^[44]。各項目詳細は前述の無人宇宙探査機の節で述べている。

表 4.7 質量に関する簡易推算方程式（有人宇宙探査機）

サブシステム系	入力	質量 [kg]
共通系	-	$M_{COM} = 116$
データレコーダ	V_{DR} [Mbit]	$K_{DRW} = 0.01[kg/Mbit]$ $M_{DR} = K_{DRW}V_{DR}$
通信系	$D[m]$	$K_{HGA} = 4.0[kg/m^2]$ $M_{HGA} = K_{HGA}D^2$
RCS	$M_{TOT}[kg]$ $\Delta V_{CR}[m/s]$ $I_{CR}[s]$	$M_{FL} = M_{TOT}[1 - \exp(-\Delta V_{CR}/gI_{CR})]$ $M_{RCS} = K_{RCS}(M_{FL})^{2/3}$ $K_{RCS} = 2.0[kg^{1/3}]$
電気推進	$M_{TOT}[kg]$ $\Delta V_{EP}[m/s]$ $I_{EP}[s]$	$M_{PP} = M_{TOT}[1 - \exp(-\Delta V_{EP}/gI_{EP})]$ $M_{EP} = K_{EPW}F + M_{EPO}$
化学推進	ΔM_{TOT}	$M_{CHE} = 0.07 \times \Delta M_{TOT}$
バッテリー	$P_{TOT}[W]$ $P_{EP}[W]$ $V_{BUS}[V]$ $\tau[hour]$ $Y[year]$	$K_{BAT} = 1.0 - 0.027(Y - 1986)[kg/AH]$ $M_{BAT} = \frac{K_{BAT}(P_{TOT} - P_{EP})\tau}{V_{BUS}}$
太陽電池	$P_{TOT}[W]$ $L[AU]$ $Y[year]$	$K_{MT} = 1.0$ $K_{SCP} = 20.0 + 1.07(Y - 1986)[W \cdot AU^2/kg]$ $M_{SCP} = \frac{K_{MT}L^2P_{TOT}}{K_{SCP}}$
構造系	$M_{TOT}[kg]$	$K_{STR} = 0.295 - 0.0118(Y - 1986)$ $M_{STR} = K_{STR}M_{TOT}$
居住モジュール (有人生命維持)	$T_{MIS}[Day]$	$M_{SYS} = 6120.0 + (1.0 + 0.5 + 0.01)T_{MIS}$

表 4.8 電力に関する簡易推算方程式（有人宇宙探査機）

サブシステム系	入力	電力 [W]
共通系	-	$P_{COM} = 116$
データレコーダ	V_{DR} [Mbit]	$K_{DRP} = 0.02[W/Mbit]$ $P_{DR} = K_{DRP}V_{DR}$
通信系	$B[bps]$ $R[AU]$ $D[m]$ $Y[year]$	$K_{TX} = 7.7 - 0.207(Y - 1986)$ $K_{LK} = 0.0013[W \cdot m^2/bps \cdot AU^2]$ $P_{TX} = K_{TX}K_{LK}BR^2/D^2$
RCS	-	-
電気推進	F [N]	$P_{EP} = K_{EPP}F$
化学推進	-	-
バッテリー	-	-
太陽電池	-	-
構造系	-	-
居住モジュール (有人生命維持)	T_{MIS} [Day]	$P_{SYS} = 10000.0$

4.3.3 MPDスラスタ

この節では最適化計算に使用したMPDスラスタのモデル化について述べる [46][48][49][50][51]。

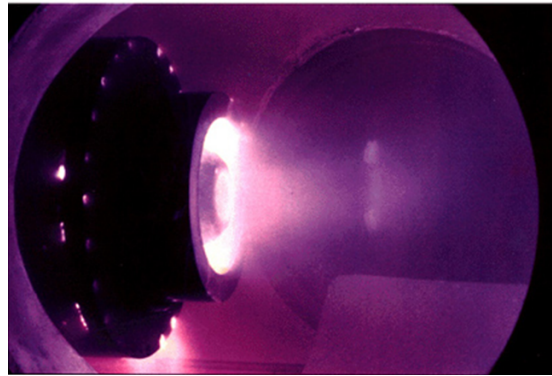


図 4.23 MPDスラスタ [47]

準一次元条件下でのモデリング

MPD スラスタのノズル内部では、電離や解離、ホール効果やシース等、様々な物理現象が複雑に絡み合っているため、厳密な流れ場の解析は非常に難しい。そこで本研究では基本的な電磁特性やおおまかな設計指針を得ることを目標とし、電離や粘性などによるエネルギー損失を考慮しない単純な完全電離のプラズマ流体モデルを用いる。計算を行う際に仮定した主な条件を以下に示す [46][48]。

- 流速、磁場、電流はそれぞれ一方向のみに制限される（準一次元モデル）
- 完全電離を仮定する
- 粘性、熱伝導、シースの影響は無視する
- 電気伝導度は一定とする

方程式

準一次元条件下での流体・磁場の方程式を以下に示す。以下の方程式はすでに無次元化されているものを示す。

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \mathbf{H} = \mathbf{0}$$

$$\mathbf{Q} = \begin{bmatrix} \rho A \\ \rho u A \\ e A \\ B A \end{bmatrix}, \mathbf{F} = \begin{bmatrix} \rho u A \\ \rho u^2 A \\ (e + p) u A \\ B u A \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 0 \\ A \frac{\partial p}{\partial x} - j B A \\ -j E A \\ -\frac{1}{R_m} \frac{\partial}{\partial x} \left(A \frac{\partial B}{\partial x} \right) \end{bmatrix} \quad (4.17)$$

$$* \begin{cases} e = \frac{1}{2} \rho u^2 + \frac{p}{\gamma - 1} & (4.18a) \\ j = -\frac{\partial B}{\partial x} & (4.18b) \\ E = \frac{j}{R_m} - u B & (4.18c) \end{cases}$$

となる。さらに、CIP 法を適用するために保存系で書かれている方程式を、以下のように

移流項を括りだした形に書き直す。

$$\frac{\partial Q}{\partial t} + u \frac{\partial Q}{\partial x} = S$$

$$Q = \begin{bmatrix} \rho A \\ \rho u A \\ e A \\ B A \end{bmatrix}, S = \begin{bmatrix} -\rho A \frac{\partial u}{\partial x} \\ -\rho u A \frac{\partial u}{\partial x} - A \frac{\partial p}{\partial x} + j B A \\ -e A \frac{\partial u}{\partial x} - \frac{\partial p u A}{\partial x} + j E A \\ -B A \frac{\partial u}{\partial x} + \frac{1}{R_m} \frac{\partial}{\partial x} \left(A \frac{\partial B}{\partial x} \right) \end{bmatrix} \quad (4.19)$$

第2章に書かれている CIP 法の解法に従って時間更新を重ねていき、物理値が変動しなくなった状態を定常解として採用する。

無次元化式

磁気レイノルズ数は

$$R_m = \mu_0 \sigma u_m L \quad (4.20)$$

磁気音速は

$$u_m = \frac{B_0^2 A_0}{\mu_0 \dot{m}} \quad (4.21)$$

全電流は

$$J = \frac{w B_0}{\mu_0} \quad (4.22)$$

である。無次元化に用いた式を以下に示す。便宜的に以下に示している式 (3-9) の無次元数のみチルダで表記している。

$$\begin{aligned} \tilde{x} &= x/L, \quad \tilde{A} = A/A_0, \quad \tilde{\rho} = \rho u_m A_0 / \dot{m}, \quad \tilde{u} = u/u_m, \quad \tilde{p} = p A_0 / \dot{m} u_m, \\ \tilde{B} &= B/B_0, \quad \tilde{V} = V w / A_0 u_m B_0, \quad \tilde{j} = j \mu_0 L / B_0, \quad \tilde{E} = E / u_m B_0 \end{aligned} \quad (4.23)$$

有次元での値を以下に示す。

$$\begin{aligned} J &= 12.5[kA], \quad \dot{m} = 1.25[g/s], \quad p_0 = 40[Pa], \quad L = 0.1[m], \\ A_0 &= 0.001[m^2], \quad w = 0.1[m], \quad u_m = 15.7[km/s], \quad \sigma = 5071[1/\Omega m] \end{aligned} \quad (4.24)$$

計算条件

モデルによる計算での境界条件と初期条件を以下に示す^[46]。入口の境界条件を

$$u_{in} = 0.5, \quad \rho_{in} = \frac{1}{u_{in} A_{in}}, \quad p_{in} = 0.002, \quad B_{in} = 1 \quad (4.25)$$

と定める。無次元化されている状態での値である。

出口境界条件での磁場は $B_{end} = 0$ とする。CIP 法では1次外挿により出口での物理値を外挿する。入口と出口での磁場の値が固定されており、二点境界値問題となっている。

計算開始の初期値条件について、CIP法を用いた場合の初期状態の値を示す。流速 $u(x) = u_m$ ，密度 $\rho(x) = 1/u_m A(x)$ ，圧力 $p(x) = p_m$ ，磁場 $B(x) = -x + 1$ とする。また，時間ステップは $dt = 0.001$ とし，計算グリッド分割数は $N = 30$ とする。

以上をまとめたものを表に示す。

表 4.9 MPD スラスタの準一次元モデルでの計算条件

入口境界条件	入口流速	$u_{in} = 0.5$
	入口密度	$\rho_{in} = \frac{1}{u_{in} A_{in}}$
	入口圧力	$p_{in} = 0.002$
	入口磁場	$B_{in} = 1$
出口境界条件	出口磁場	$B_{end} = 0$
初期状態値	流速	$u(x) = u_m$
	密度	$\rho(x) = 1/u_m A(x)$
	圧力	$p(x) = p_m$
	磁場	$B(x) = -x + 1$
磁気レイノルズ数	$R_m = 10$	
時間ステップ	$dt = 0.001$	
計算グリッド分割数	$N = 30$	

推進効率 η を評価の指針とする^[45]。無次元化すると， $\dot{m} = 1, J = 1$ である。

$$\begin{aligned}
 \eta &= \frac{F^2}{2\dot{m}P} \\
 &= \frac{(\dot{m}u_{end} - \dot{m}u_{in})^2}{2\dot{m}VJ} \\
 &= \frac{(u_{end} - u_{in})^2}{2V}
 \end{aligned} \tag{4.26}$$

ノズル形状（設計変数）

ノズル形状は以下により表現される。流れ場は円筒座標系の中心軸を通る平面で考え、さらに上下対称であるとして図のようにノズルを表現する。流体は一次元流れを考えているが、ノズル形状は二次元としている（準一次元モデル流れ）^[48]。

x 方向を 4 分割した等間隔の各点をノズル形状設計のコントロールポイントとする。よって、設計変数は 5 つとなる。コントロールポイントは上下のみに動くことができ、 $1.0 \leq A \leq 10.0$ の範囲を移動するという制約条件を設ける。

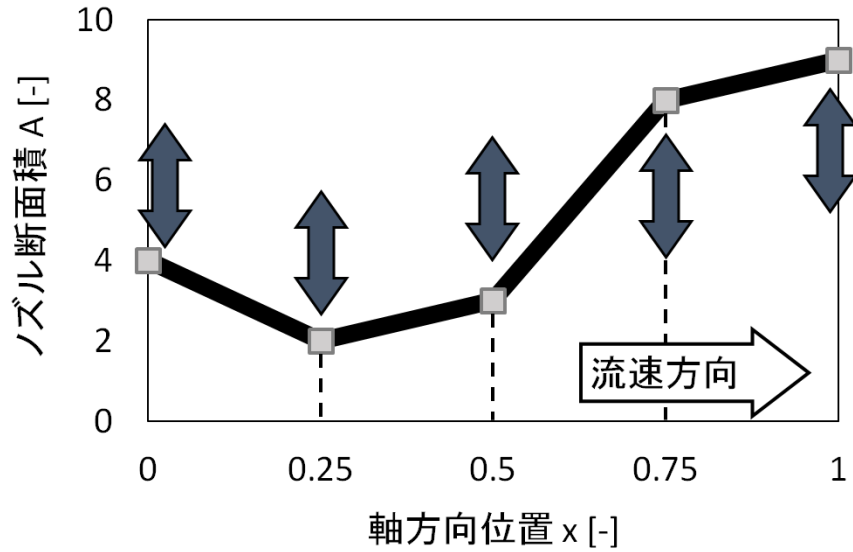


図 4.24 5つのコントロールポイントによるノズル形状表現

以上により、MPD スラスタのモデル化を行い事前に応答曲面を作成する。この作成した応答曲面の関数式を用いて、宇宙機の機体設計モデルでの推力などを計算する。

CIP

数値流体計算について CIP 法を用いる。MPD スラスタの準一次元モデルでの解析には CIP 法 (Cubic-Interpolated Pseudo-Particle Scheme) を用いた^[57]。これは、波の波形とその微分量も伝播させることで、移流前の形を崩さずに伝播させられる。また、人工粘性を加えることで数値的な振動を抑えた。

方程式

外力項が加わっている移流方程式は以下である。

$$\frac{\partial Q}{\partial t} + u \frac{\partial Q}{\partial x} = S \quad (4.27)$$

この方程式を移流を計算する部分 (advection phase) とその他の部分 (non-advection phase) に分けて考える。これは計算上のテクニックである。

non-advection phase

$$\frac{\partial Q}{\partial t} = S \quad (4.28)$$

advection phase

$$\frac{\partial \mathbf{Q}}{\partial t} + u \frac{\partial \mathbf{Q}}{\partial x} = \mathbf{0} \quad (4.29)$$

まず，移流項以外の部分 (non-advection phase) を更新することで，予因子を求める．

$$\begin{aligned} \mathbf{Q}^n &= \bar{\mathbf{Q}} \\ \frac{\partial \mathbf{Q}^n}{\partial x} &= \frac{\partial \bar{\mathbf{Q}}}{\partial x} \end{aligned} \quad (4.30)$$

具体的な式は以下である．

$$\begin{aligned} \bar{\mathbf{Q}}_i &= \mathbf{Q}_i^n + \mathbf{S}_i^n \Delta t \\ \frac{\partial \bar{\mathbf{Q}}_i}{\partial x} &= \frac{\partial \mathbf{Q}_i^n}{\partial x} + \frac{1}{2\Delta x} (\bar{\mathbf{Q}}_{i+1} - \mathbf{Q}_{i+1}^n - \bar{\mathbf{Q}}_{i-1} + \mathbf{Q}_{i-1}^n) \end{aligned} \quad (4.31)$$

次に予因子から移流項部分 (advection phase) を更新して，修正子を求める．

$$\begin{aligned} \bar{\mathbf{Q}} &= \mathbf{Q}^{n+1} \\ \frac{\partial \bar{\mathbf{Q}}}{\partial x} &= \frac{\partial \mathbf{Q}^{n+1}}{\partial x} \end{aligned} \quad (4.32)$$

詳細な式を以下に示す．

$$\begin{aligned} \mathbf{Q}_i^{n+1} &= \{(\mathbf{a}X + \mathbf{b})X + \bar{\mathbf{Q}}_{x,i}\}X + \bar{\mathbf{Q}}_i \\ \mathbf{a} &= \frac{-2(\bar{\mathbf{Q}}_i - \bar{\mathbf{Q}}_{im})\text{sgn}(u_i) + (\bar{\mathbf{Q}}_{x,i} - \bar{\mathbf{Q}}_{x,im})\Delta x}{(\Delta x)^3} \\ \mathbf{b} &= \frac{-3(\bar{\mathbf{Q}}_i - \bar{\mathbf{Q}}_{im}) + (2\bar{\mathbf{Q}}_{x,i} + \bar{\mathbf{Q}}_{x,im})\Delta x \cdot \text{sgn}(u_i)}{(\Delta x)^2} \end{aligned} \quad (4.33)$$

傾きの時間更新の式は以下の通りである．

$$\begin{aligned} \hat{\mathbf{Q}}_{x,i} &= 3\mathbf{a}X^2 + 2\mathbf{b}X + \bar{\mathbf{Q}}_{x,i} \\ \mathbf{Q}_{x,i}^{n+1} &= \hat{\mathbf{Q}}_{x,i} - \frac{u_{i+1} - u_{i-1}}{2\Delta x} \hat{\mathbf{Q}}_{x,i} \Delta t \end{aligned} \quad (4.34)$$

ただし， X, im は以下で定義される．

$$X = u\Delta t \quad (4.35)$$

$$im = i - \text{sgn}(u)$$

$$\text{sgn}(u) = \begin{cases} 1 & (u \geq 0) \\ -1 & (u < 0) \end{cases} \quad (4.36a)$$

$$(4.36b)$$

以上より，時間更新 $\mathbf{Q}^n \mathbf{Q}^{n+1}$ が求まる．

応答曲面法

応答曲面法とは実験データなどの値をもとに解空間を関数近似する方法である。実験や計算負荷の高い数値計算では最適化するための十分なサンプル数が取れないことが困難点であったが、関数で近似することにより、少ないサンプリング数で最適化計算できる利点がある [59]。この応答曲面を宇宙機の機体構成モデルの計算フェーズに適応する。MPD スラスタのノズル形状決定での最適化計算ルーチンに用いる。これによって、CIP 法での流体計算の計算負荷を軽減することが出来る。今回は二次モデルで近似された関数空間を PSO を用いて最適化する。関数式自体は単純な多項式である。作成した関数式を以下に示す。 A は各コントロールポイントのノズル断面積を表し、添字は各点の x 方向位置を表す。評価関数として推進効率 η の式を用いる。電力 P の式は宇宙機の機体構成モデルから要求される電力を満たすための制約条件の式として使う。

実際に使用した関数を以下に示す。応答局面法により作成した局面を表す方程式である。式が煩雑にならないように係数は有効数字を考えて四捨五入している。図 4.25 は代表として方程式におけるノズル入口 A_0 とノズル出口 A_4 での関係図を示したものである。

$$\begin{aligned}
 \eta = & 0.81 - 0.002A_0 - 0.011A_1 - 0.017A_2 - 0.009A_3 + 0.042A_4 \\
 & + 0.0001A_0^2 + 0.001A_0A_1 - 0.00003A_0A_2 - 0.0004A_0A_3 \\
 & - 0.0002A_0A_4 + 0.0005A_1A_1 + 0.0001A_1A_2 - 0.001A_1A_3 \\
 & + 0.0001A_1A_4 + 0.0003A_2^2 + 0.0004A_2A_3 + 0.0006A_2A_4 \\
 & + 0.0003A_3^2 + 0.001A_3A_4 - 0.003A_4^2
 \end{aligned} \tag{4.37}$$

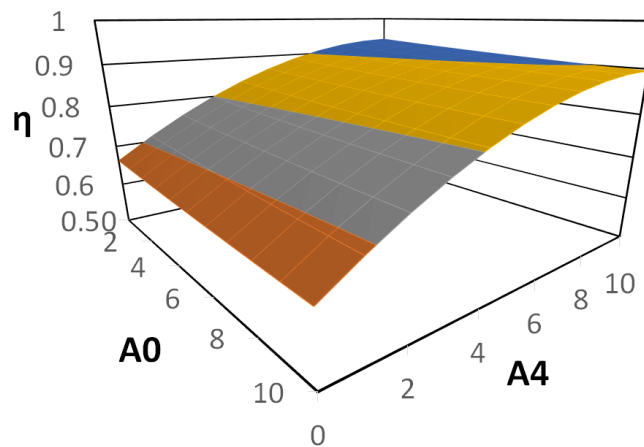


図 4.25 ノズル入口と出口に関する応答局面図

4.3.4 軌道計画 [35][36][37][41]

ミッションは地球と火星の往復するものである。軌道計画は大きく分けて3つのフェーズから成る。1つ目は地球から火星へ向けて電気推進機を用いて宇宙機を増速させるフェーズ、2つ目は火星でのランデブーを行うフェーズ、3つ目は火星から地球へ帰還するために電気推進機を用いて減速させるフェーズである。これら3つについて以下で詳細を述べる。制約条件としては2014年1月1日から10年以内に出発する条件を与えて、宇宙機が火星に向かう間の飛行時間は5年以内にするという制約も加える。地球を出発し火星をランデブーする期間は5年以内とし、火星から地球への帰還は5年以内に行うものとする。

地球から火星への軌道

地球を出発し火星まで電気推進機を用いて増速する軌道を考える。本研究では太陽と宇宙機の2体問題として最適化問題設定を行っていいので、地球と火星の重力による影響は考慮していない。

ランデブー軌道

火星へ到達した時に化学推進を噴射することで火星と宇宙機の相対速度をゼロとする。その後、宇宙機は火星と同じ軌道をとるものとする。ランデブー時の宇宙機は火星の衛星軌道をとっていると仮定しており、スラスターは停止状態とする。有人宇宙探査機においてはランデブー時においても、水食料空気の生活廃棄物は放出し続けるものとする。このランデブー期間を最適化問題上の設計変数とする。このランデブー期間を変更することで宇宙機は帰還のタイミングを計ることになる。

帰還軌道

帰還においては電気推進機を逆噴射して減速させる。火星軌道から徐々に減速させ、地球に帰還させる。その際に地球での化学インパルスによる減速は行わず、地球との相対位置をゼロとする条件を加えるのみとする。帰還軌道については相対位置ゼロとする条件で最適化の数値計算を行う。地球上での再突入は空気による減速を考えて燃料を用いたインパルス噴射は考慮しないものとする。

推力方向角

推力方向角を設計変数とする。基本的に宇宙機の世界速度ベクトル方向に推力方向を向けるものとする。ここでの推力方向角は速度ベクトル方向からどのくらいの角度を傾けるかという

ものである。図 4.26 を用いて方向角の向きを示す。ただし、帰還フェーズにおいては速度ベクトルと逆向きに推力方向角をとる。この場合も速度ベクトルの逆ベクトルからどのくらい傾けるかという方向角のパラメータを導入する。同様に図 4.27 に示す。

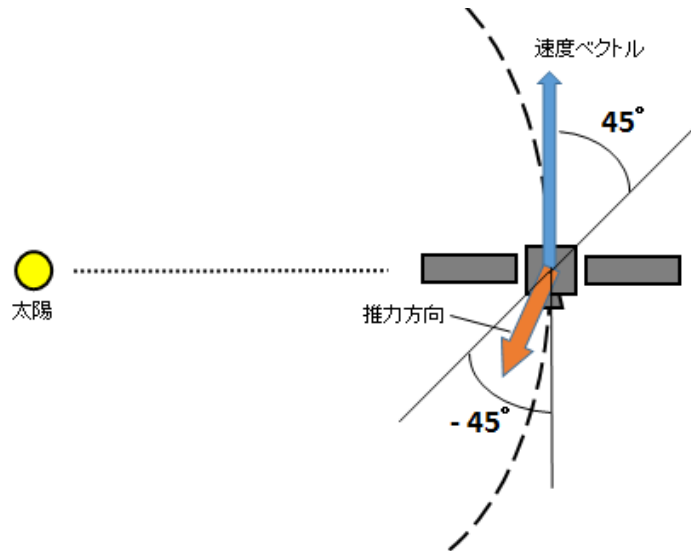


図 4.26 宇宙探査機の推力方向範囲

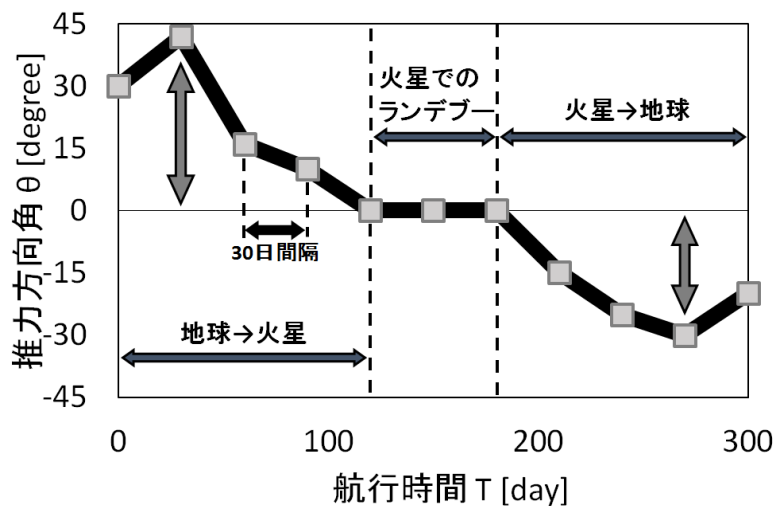


図 4.27 宇宙探査機の推力方向角コントロールポイント

推力方向角 θ_{th} を表現するために図 4.27 のような四角い灰色で示されたコントロールポイントを用いる。計算結果によるミッション期間の平均やコントロールポイントの総数を考慮して、30 日間隔でコントロールポイントを配置する。トータルのコントロールポイント

数がおおよそ 180 点となるようにした。コントロールポイント間は直線で結び、徐々に方向角が変化する様にする。地球から火星間でのコントロールポイントの移動範囲は正として $0 < \theta_{th} < 45$ とする。火星でのランデブー期間は推力は停止している状態とするので推力方向角もゼロとする。火星から地球への帰還では推力方向角は負として $-45 < \theta_{th} < 0$ とする。

宇宙機の運動方程式

宇宙機と天体の運動は三次元とする。また、太陽と宇宙機の二体問題として取り扱う。推力方向は常に速度ベクトル方向になるように推力方向角を設定する。運動方程式の数値計算にはルンゲ・クッタ法を用いた。地球と火星の位置と速度はケプラー軌道要素から楕円軌道として計算した。

$$\frac{d}{dt} \begin{pmatrix} x \\ y \\ z \\ u \\ v \\ w \\ m \end{pmatrix} = \begin{pmatrix} u \\ v \\ w \\ -\frac{\mu x}{r^3} + \frac{K_p P}{m} \sin \theta_1 \cos \theta_2 \\ -\frac{\mu y}{r^3} + \frac{K_p P}{m} \sin \theta_1 \sin \theta_2 \\ -\frac{\mu z}{r^3} + \frac{K_p P}{m} \cos \theta_1 \\ -\frac{P^2}{g F I_{SP}} \end{pmatrix} \quad (4.38)$$

位置 x, y, z 、速度 u, v, w 、機体質量 m 、推力方向角 θ_1, θ_2 、電力 P 、推力値 F とした。

4.3.5 最適化問題設定

以上の条件においての問題設定をここでまとめる。

目的関数

多目的最適化としており、目的関数は全機質量の最小化とミッション期間の最小化としてある。これらの要素をパレートランキング法により順位をつけ評価値の計算を行い最適化計算を行う。

加えてニッチング法により、探索点集合の個体間の混雑度を解消するように評価値の修正している。また、飛行終了時点での宇宙機位置と目標天体位置との相対距離をペナルティ法により評価値を改悪する手法を加えている。パレートランキング法、ニッチング法、ペナルティ法については第 2 章で説明している。評価値 I を以下に示す。パレートのランクを $Rank_{pareto}$ 、ニッチング法による修正値は N_{sh} 、ペナルティ法による改悪係数を $C_{penalty}$ 、宇宙機の火星

到達時の火星との相対距離を R_{mars} 、宇宙機の地球到達時の地球との相対距離を R_{earth} とする。

$$I = Rank_{pareto} + N_{sh} + C_{penalty} \times (R_{mars} + R_{earth}) \quad (4.39)$$

今回の計算では $C_{penalty} = 10^{-7}$ とした。大まかな各項目のオーダー値は、 $Rank_{pareto}$ が 10^0 、 N_{sh} が 10^{-1} 、 $C_{penalty} \times (R_{mars} + R_{earth})$ が 10^{-1} となっている。

設計変数

機体設計変数に関しては宇宙機の機体構成要素である初期全電力 P_0 、推進効率に関する項目にノズル形状を表すコントロールポイントが5つある。軌道計画変数に関しては惑星間軌道の計画要素である出発日時 T_{date} とランデブー期間 T_{rendez} 、推力方向角を表すコントロールポイントが最大で183個ある。

以上をすべて足すと最大で191個の設計変数がある。推力方向角を除くと8つの変数が存在する。

制約条件

問題設定時に定義した制約条件を以下に示す。

- ・宇宙機の初期全電力 P_0

$$50.0 \leq P_0 \leq 220.0[\text{kW}] \quad (4.40)$$

- ・ノズル形状を表す無次元化されたコントロールポイント A

$$0.0 \leq A \leq 10.0[-] \quad (4.41)$$

- ・出発日時 T_{date} は2014年1月1日から20年以内に出発

$$T_{date} \leq 20[\text{years}] \quad (4.42)$$

- ・ミッション期間(飛行時間) T_{flight} は

$$T_{flight} \leq 10[\text{years}] \quad (4.43)$$

- ・ランデブー期間 T_{rendez} は

$$T_{rendez} \leq 5[\text{years}] \quad (4.44)$$

- ・推力方向角 θ_{th} については

$$-45 \leq \theta_{th} \leq 45[\text{degree}] \quad (4.45)$$

となる。これらは宇宙機の機体構成と軌道に関する制約条件である。

最適化手法の設定パラメータ

最適化手法に関する設定パラメータと制約条件を表に示す。

表 4.10 最適化手法の設定パラメータ（有人宇宙探査機）

	提案手法	PSO	GA
探索点数 (個体数)	10	10	10
最大計算回数 (世代数)	1000	1000	10000
固有パラメータ	重力定数 $G = 10^4$ 回転量 $\omega = 0.01$	収束係数 $w = 0.9$ から 0.1 速度係数 $c_1 = c_2 = 2.0$	突然変異率 0.02
解空間に対する 速度制限	$1/300$	$1/300$	-

4.3.6 最適化計算結果

この節では有人宇宙探査機の機体構成と軌道の最適化計算結果を示す。目的関数はミッション期間と全機質量の2つとしている。設計変数は前節で述べた通りである。図 4.28 は目的関数同士の関係図を示した図である。縦軸にミッション期間、横軸に全機体質量をとっている。双方とも目的関数であり最小化する対象である。図において左下への方が最適化の進行方向となる。青い四角のポイントが探索点の初期世代群である。赤い三角のポイントが繰り返し計算が終了した最終世代の探索点群の位置となる。探索点数は10点となるが解が重なっており見かけ上は5つのように見えている。先節の無人宇宙探査機の最適化計算では初期探索点群に制限をかけて位置を制約した生成を行っていたが今節の有人宇宙探査機の最適化計算では初期探索点群に制約を設けておらず、制約条件に示した通りの設計変数の上限下限での制約とした生成としている。最良解は黄色い丸のポイントとなり、ミッション期間は1239日、全機質量は144.4[ton]となった。参考としてヨーロッパ宇宙機構が計画している有人宇宙探査船の概要値を示すと全体質量が100[ton]程度、往復期間が3年としている。本計算結果と比較すると機体質量は44%増、ミッション期間は20%増となる結果となった。化学推進と電気推進との推進機構の違いがあり単純比較は難しいが機体質量の増加が現実的な最適化計算結果を得るための障害となっている。特に電気推進用の推進剤の質量が全体質量の42%を占めており、この推進剤の削減が現実的な解を得るための必須事項となる。図 4.29 は設計変数の解空間を図示する。設計変数は複数あり代表として縦軸にランデブー期間、横軸に初期全電力をとった。目的関数に直接的に関わる2つをとった。ミッション期間は設計変数であるランデブー期間に依存し、全機質量は初期全電力に依存する。

ミッション期間を最小化するためにランデブー期間も制約下限値まで下がる傾向をみせるはずだが、結果は初期個体群から離れた上方向に移動した。これはランデブーによって宇宙機が帰還のタイミングを図るために上方に移動したと考えられる。ヨーロッパ宇宙機構による有人宇宙船の試算によると全機質量は 100[ton] 程度まで軽減することが出来るので、問題設定における宇宙機のシステム構成や重量推算式を変更することでさらなる良解を得ることが出来ると考えられる。この図 4.29 を最適化計算手法の観点からみると、最良解は 1 つだけグループ内から離れており宇宙ジェットのアルゴリズムが発動して実行不可能領域を飛び越えることで図の最良解の位置にたどり着いたと考えられる。重力とコリオリ力による移動だけではローカルの解群の中を脱出し他のグループに移動するまではいかなかった。重力は探索点群の重心に移動する方向に力が働くので、宇宙ジェットによってより良い解に 1 つだけ到達しても残りの 9 つの探索点の重力による全体重心がグループ外まで移動しなければ重力によって最良解にグループ全体が移動することは難しい。重力定数 G の値を大きくとるなどの対策をとる必要がある。しかし、重力によってグループがバラバラになるくらいの強い力を受けると解探索が逆に難しくなり速度制限を設けることで最適化計算の進行をスムーズにする。今回の問題設定においては実行不可能領域が広く飛び飛びに分布しているので、本計算の速度制限である設計変数の上限下限の $1/300$ とした制限では図 4.29 の初期解群領域から最良解付近に移動することはなかった。このような問題に対応するために本提案手法はコリオリ力による重力移動の補助をするシステムが存在する。最良解を中心に解空間を回転させることでコリオリ力が働き重力による移動をサポートすることになる。もちろんコリオリ力による速度ベクトル方向の方向修正は改良する方向にも改悪する方向にも動き得る。今回の問題設定では実行不可能領域に探索点が移動した場合は元の位置に戻すというルーチンをとっているので、重力またはコリオリ力によってランデブー期間が 100 日以上の領域に入っても計算をやり直すために本来のコリオリ力による移動の効果を得ることが難しかった。問題設定を改変し実行不可能領域を無くすか手法のパラメータである重力定数 G と回転量 ω のバランス調整を精密に行う必要がある。

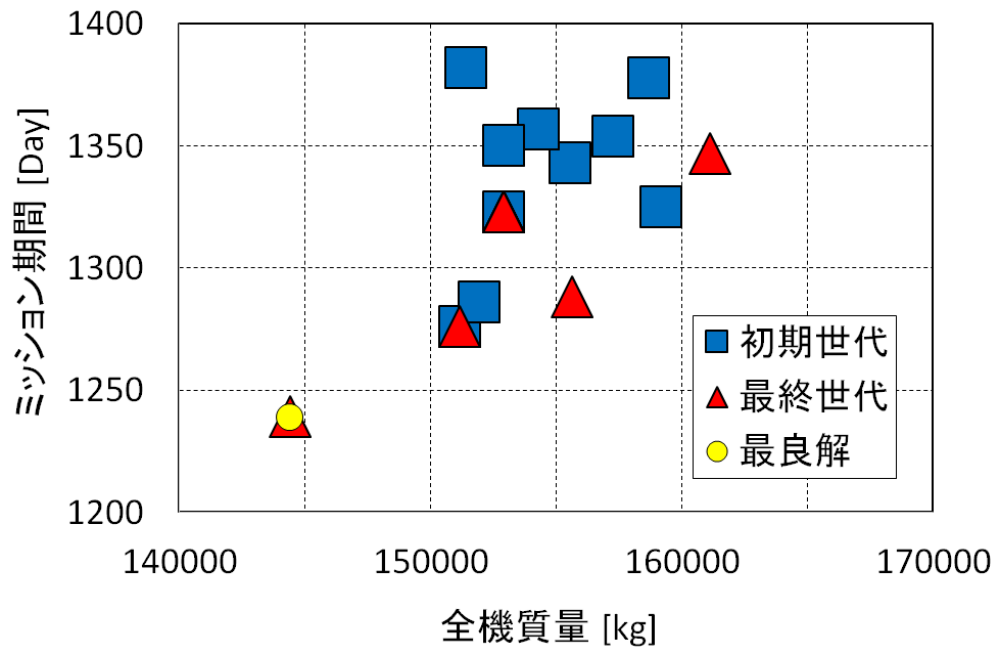


図 4.28 目的関数 (有人宇宙探査機)

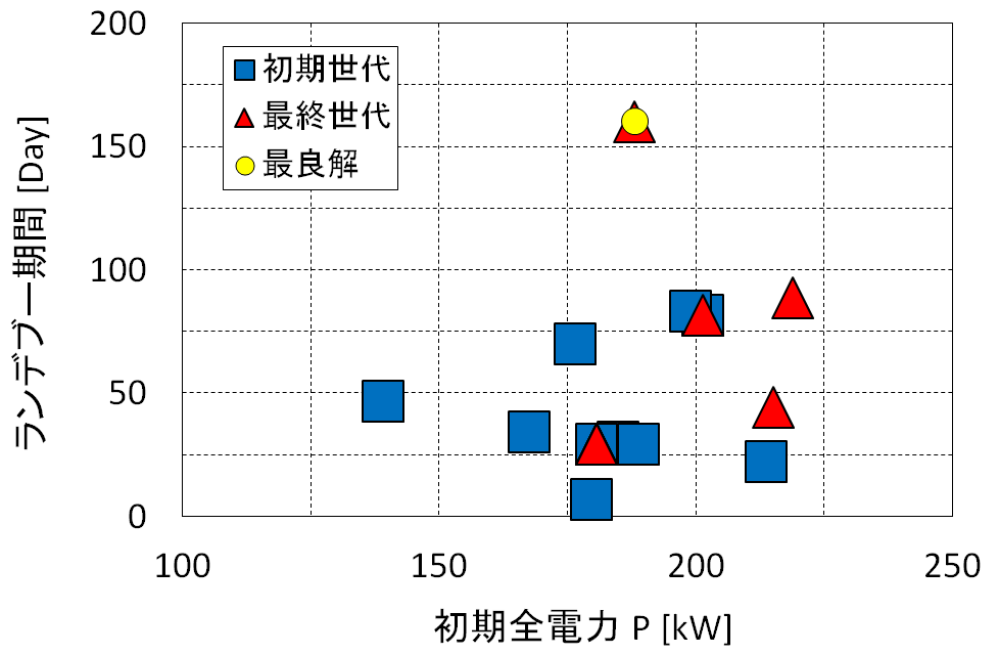


図 4.29 設計変数 (有人宇宙探査機)

惑星間軌道

最適化計算によって得られた最良解から惑星間軌道を示す。図 4.30 は最良解をピックアップして地球と火星の往復軌道の計算結果を示す。先節の無人宇宙探査機とポイントやラインの説明は同様であるが重ねて述べる。地球から火星への軌道を青い線で示している。青い丸は地球の位置を示している。青い線とつながっている方が出発位置である。赤い三角のポイントが火星到達位置である。灰色の点線がランデブー軌道を表している。赤い線が帰還軌道を表しており、赤い線とつながっている赤い三角が火星出発位置である。黒い点線は内側が地球軌道、外側が火星軌道である。本計算結果において宇宙探査機は単純なスパイラル軌道としている。最適化計算結果のミッション期間は 1239 日である。ヨーロッパ宇宙機構による有人宇宙船の概要によるミッション期間設定は 1095 日程度なので比較すると半年ほどの増加分があるが電気推進を用いたスパイラル軌道をとっていることを考えると同等のオーダーであるといえる。先節に引き続き有人宇宙探査機の最適化計算は提案手法の有効性を示すためのものであるので、今回の問題設定内で機体構成と軌道を得ていることは最適化計算アルゴリズムが有効に働いているといえる。

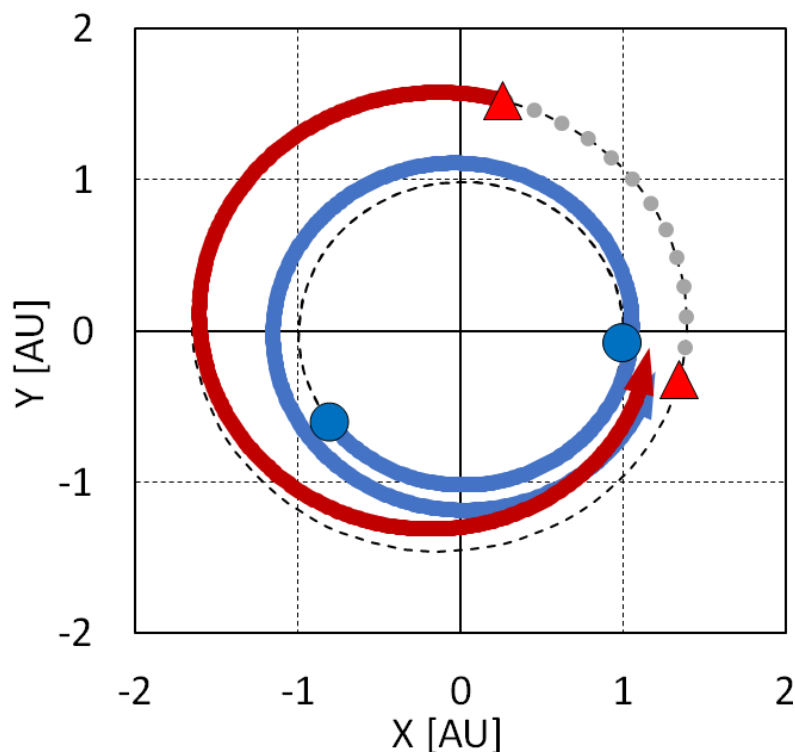


図 4.30 有人宇宙探査機の軌道計画

次に推力値、制御変数である推力方向角、機体重量の時間履歴を図示する。図 4.31 は探

査機の推力値の変化を表すものである。本計算で有人宇宙機はスピン型であり機体の側面に太陽電池セルを張り付けることを想定している。推力値は太陽電池パネルから得られる電力に比例し、太陽からの距離に依存する。図 4.31 の縦軸は推力値であり、プラス側は地球から火星への往路の推力値である。推力方向は常に速度ベクトルと同一方向に向いているとする。地球から遠ざかるため推力値は減少している。中央の推力ゼロの区間はランデブー期間である。後半部分のマイナス値については、速度ベクトル方向と逆向きに常に推力を発生させているので推力をマイナスの値と表現している。火星出発位置と太陽位置との距離の関係から推力値は減少してから増加しているが軌道进行を考えると正常な変化である。平均推力値は 8.2[N] であり、電気推進としてはかなり大きい値であるが本問題設定での電気推進機に投入する電力使用量が 177[kW] と大きいためにこのような平均推力値となっている。国際宇宙ステーションの全電力量は 110[kW] であることを考えると、今回の計算結果を実現するような構造物を建築するためには将来の技術発展を望むことになる。図 4.32 は推力方向角の時間履歴である。制御変数であるコントロールポイントは 30 日おきに設定してあり、30 日ごとに推力方向を変化させることになる。上下限は 45°としており、プラス値は速度ベクトルの外側に推力方向を傾け、マイナス値は速度ベクトル方向の内側に推力を傾ける。特に帰還部分の後半において推力方向角が上限値に近づいている値をとっており宇宙機が最終的な位置調整を行うような軌道をとっていると理解できる。図 4.33 は機体重量の変化を時間履歴で表示したものである。推進剤を徐々に使いながら航行している。計算上は推進剤をピッタリ使い切る量を搭載しており地球に帰還することとしている。中央の機体質量の減少が無い区間はランデブー期間である。また、中央区間の機体質量が不連続に降下している部分は化学推進によるランデブー操作による質量使用量分である。

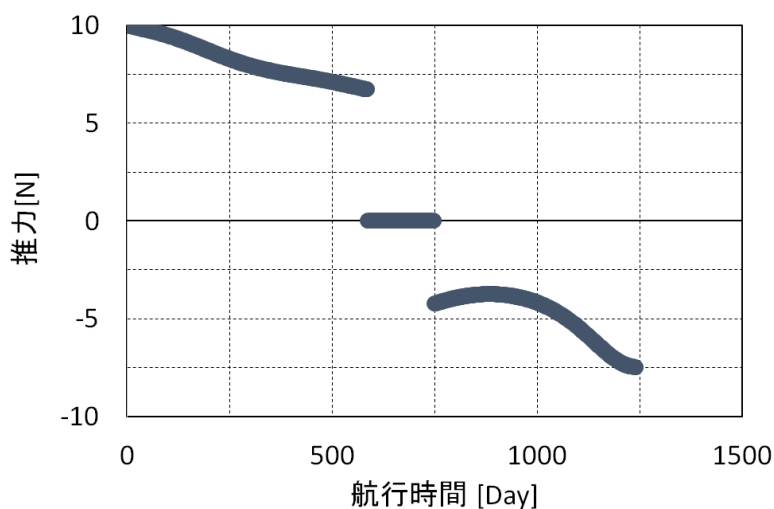


図 4.31 推力値 (有人宇宙探査機)

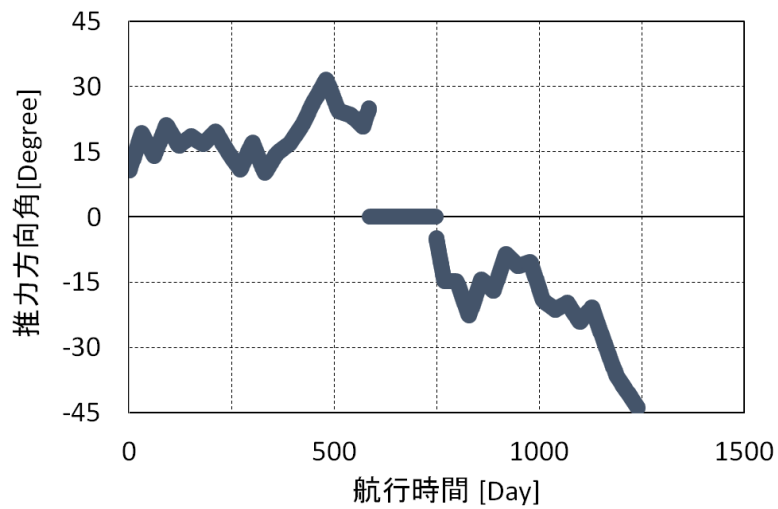


図 4.32 推力方向角 (有人宇宙探査機)

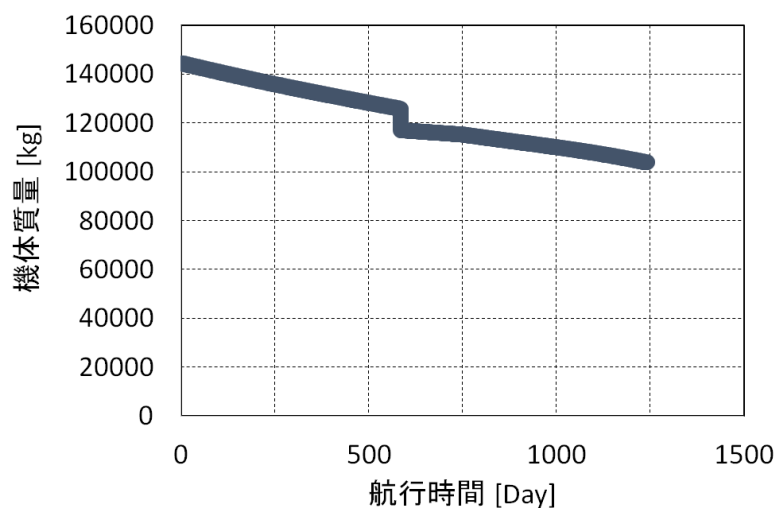


図 4.33 重量値の時間履歴 (有人宇宙探査機)

機体構成

提案手法によって得られた有人宇宙探査機の最良解の機体構成結果を表 4.11 に示す。各サブシステムの質量と電力の値を示している。前節の無人宇宙探査機の簡易推算式に居住モジュールと化学推進系を加えたものとなる。特徴として全体質量の約 4 割を推進剤に割り振られている。ミッション期間によって使う推進剤量が変わるので全機重量も変化する。推進剤質量が 62[ton] あるためにそれを支える構造系と推力のための電力を生み出す太陽電

池系が計 24[ton] あるので今回の問題設定の上では推進剤使用量を減らすことが全機質量の減少につながる事が分かる。推進剤の重さがネックとなっているので、実現可能性を考えると推進剤質量を軽減する方法を取り入れなくてはならない。全機質量は 144[ton] となり、ヨーロッパ宇宙機構が試算している有人宇宙探査機の 100[ton] と比べると質量の誤差は 44% であった。

表 4.11 機体構成諸元（有人宇宙探査機）

	質量 [kg]	電力 [W]
共通	106.0	109.0
データレコーダ	0.64	1.28
通信	10.2	106.2
姿勢制御（RCS）	1423.9	48.3
姿勢制御（燃料）	18995.8	-
電気推進（機器部）	479.5	177940.4
電気推進（推進剤）	62031.1	-
化学推進系	8186.4	-
バッテリー	148.8	-
太陽電池	19990.0	-
構造系	13632.4	-
居住モジュール	19341.8	10000.0
合計	144410.6	188104.0

次にMPDスラスタのノズル形状最適化結果について述べる。図 4.34 は有人宇宙探査機の最適化問題設定における電気推進機のノズル形状最適化結果である。有人宇宙探査機での推進装置はMPDスラスタを使うものとする。事前に応答局面を作成し提案手法の最適化計算ルーチン内部で形状の設計変数を変化させている。最終的な最適形状は図のような釣り鐘型のノズル形状となった。入口付近は狭く、出口付近で大きく開くと推進効率が良いことが応答局面を作成した段階で分かっているので、入口と出口の面積比が大きくなるような釣り鐘型の形状が計算結果として出てきている。このような形状が最適形状と今回の問題設定における計算結果では出ており、推進流体の圧力膨張を考慮したノズル形状では後部が大きく開いているので妥当な形状であると考えられる。また、推進効率は $\eta = 0.90[-]$ 、比推力は $Isp = 3399[s]$ となった。一次元流れの数値流体計算を行っているので、エネルギー損失が無く高い推進効率となっている。

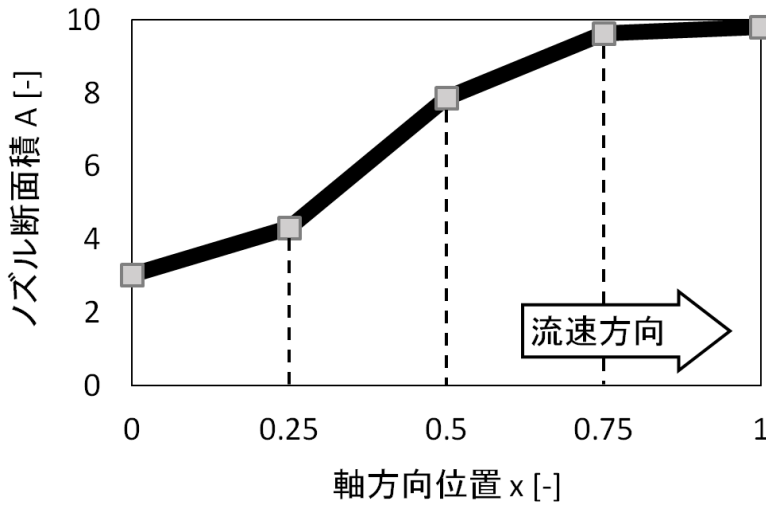


図 4.34 MPD スラスタのノズル形状 (有人宇宙探査機)

次に機体重量を基に面積や体積を逆算する。太陽電池パネルなどははやぶさの重量と面積の関係を参考とした。太陽電池パドルの $1[\text{kg}]$ 当たりの面積は $0.257[\text{m}^2/\text{kg}]$ とし、中央のメインユニットの質量体積比は $0.00427[\text{m}^3/\text{kg}]$ として計算した。この変換値をもとに各部の長さを算出し三面図に作画した。有人宇宙探査機はスピン型を想定しており、1 分間に 10 回転するような角速度で宇宙機は回転するものとする。宇宙機の円形部分の外形部での遠心力が地球上での重力加速度の 70% を生み出すような構造半径を持つものとして機体半径を決定する。宇宙飛行士は外形部に滞在することで 7 割の重力を得ることが出来る。

他手法との比較 (有人宇宙探査機)

本節では他手法による有人宇宙探査機の最適化計算結果について示す。最適化問題設定は提案手法の場合の設定と同一とする。各々の手法独自の設定パラメータは表に示す。以下に最適化計算結果を示す。図列の左側が P S O による最適化結果、右側が G A による計算結果である。

PSO に関して図 4.36 をみると最終世代が初期世代群の領域から脱出できていない。これは前述の無人宇宙探査機の最適化結果でも同様であった。有人宇宙探査機における問題設定においても単純 P S O アルゴリズムでは脱出アルゴリズムを備えていないために初期世代群から抜け出せていない。有人探査機での最適化問題設定は設計変数が 70 個近くあり、初期個体群の生成自体に計算時間がかかった。このことから実行可能領域が無人機問題に比べて格段に狭くなったと予測できる。その狭い実行可能領域において単純 P S O では初期個体群を

脱出することが出来なかった。今回のP S Oの計算アルゴリズムにおいて最良解を保存するというルーチンを入れていないため最良解は全計算回における目的関数が一番良い解を選んでいる。P S Oの性質として計算途中の最良解に集まりやすい傾向があるので初期個体群の領域内を探索する状態となり初期個体群領域の外に最適解が存在する場合は発見することが困難になる。この節の有人探査機の問題設定では実行可能領域が狭いと考えられるのでP S Oの解群の位置関係や初期速度ベクトルの向きによって領域の中心に移動する進行方向になった。図 4.38 は最良解がランデブー期間 100 日以上に位置にあり、全個体群の移動方向が図の上昇方向になることはなかったが、初期全電力は 200[kW] 付近に集まっている傾向がある。提案手法の結果と比較するとP S Oは一所に集まろうとする傾向が見て取れるが、提案手法はコリオリ力によって初期個体群の中でも速度ベクトルを修正されて移動するので狭い実行領域内でも解の多様性を保っていると考えられる。大域的最適解に到達するためには局所解にトラップされにくい必要があるので、コリオリ力によってわざと遠回りする速度ベクトル方向の修正ルーチンは必要であるといえる。

GA は図 4.37 をみると初期世代群から脱出できていないことが分かる。最終世代の赤い三角のポイントが3箇所に移動して収束していることが分かる。GAは突然変異アルゴリズムが脱出アルゴリズムとなり、新たな領域に到達する可能性を有しているが本問題設定では初期解群の領域を脱出することはなかった。GAの交叉方法にもよるが今回使用したGAの交叉方法は2点間位置の中心をとるものであり初期解群の中心に移動する傾向がある。突然変異でのランダム移動は実行可能解が生成されなければならず今回の計算回数では突然変異は起こらなかった。提案手法における宇宙ジェットのアルゴリズムでも同様のことが起こり得る。大域的な最適解を得るために世代数の繰り返し計算回数を増やすことで最適解に到達する可能性はある。

P S OとGAの設計変数の関係図と提案手法とを比較する。提案手法の結果をもとに考えると最適解への計算方向は中央上部の方向に進むことが良いと考えられる。P S Oにおいては初期世代から脱出できておらず最終世代まで繰り返し計算が行われた。GAについても初期世代群から脱出はなかった。GAの設計変数結果である図 4.39 をみると提案手法の最良解であるランデブー期間 150 日、初期全電力 175[kW] 付近に解が存在している。しかし、提案手法の最適解付近に解が存在してはいるが目的関数に変換すると図の黄色い丸のポイントが最良解になる。ランデブー期間と初期全電力以外にも設計変数は変化しているので一概に目的関数への影響を言及出来ないが、無人宇宙探査機での結果と同様に設計変数差と目的関数差との隔たりがあるために最適化計算の最良解への進行を妨げている要因ともなっている。GAの計算結果では図の中央上部の領域へは到達することが出来なかった。

本節でもP S OとGAの計算結果における最終世代の最良解をピックアップして軌道などの結果を示す。双方とも初期個体群から最良解が選ばれているので、特にミッション期間が

1250 日と似たような値となったために軌道に関しては似通った軌道経路となった。どちらの解ともランデブー期間が提案手法より長いために全機体質量が3 から 7[ton] 程度重くなっている。

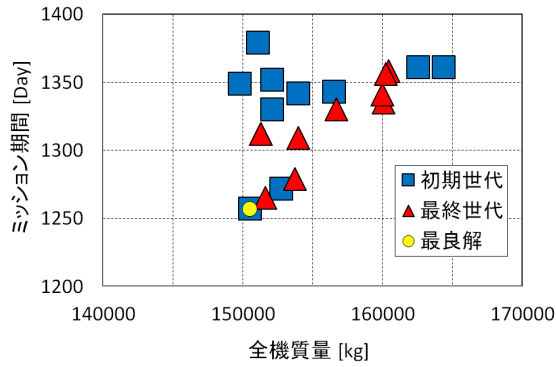


図 4.36 PSO 目的関数結果（有人）

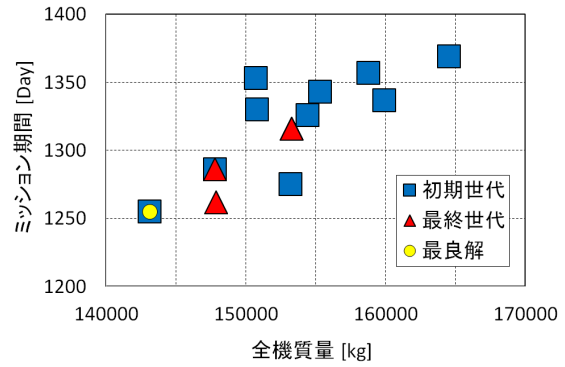


図 4.37 GA 目的関数結果（有人）

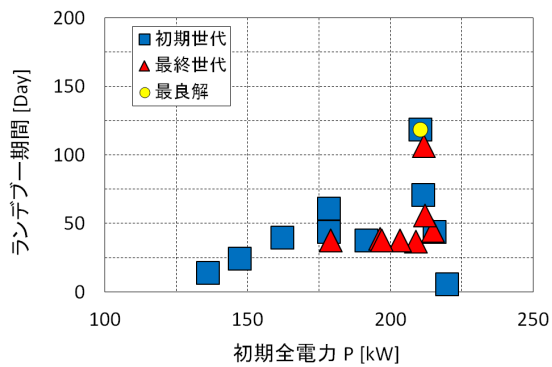


図 4.38 PSO 設計変数結果（有人）

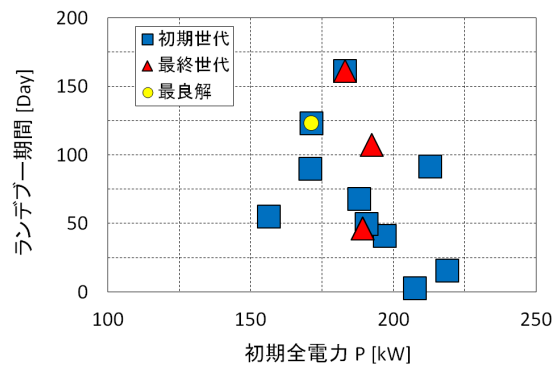


図 4.39 GA 設計変数結果（有人）

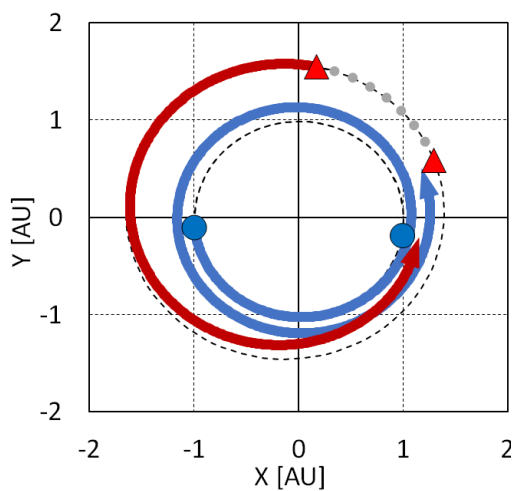


図 4.40 PSO 軌道結果 (有人)

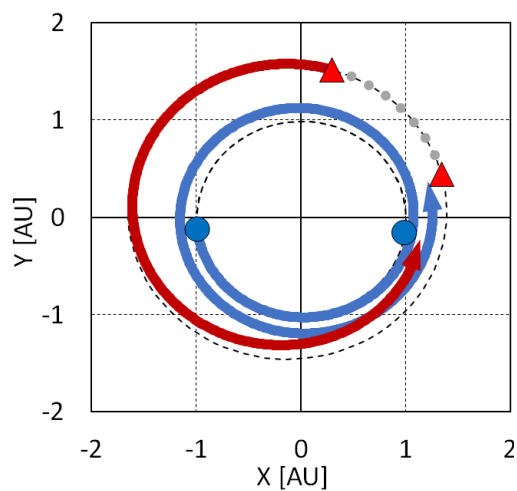


図 4.41 GA 軌道結果 (有人)

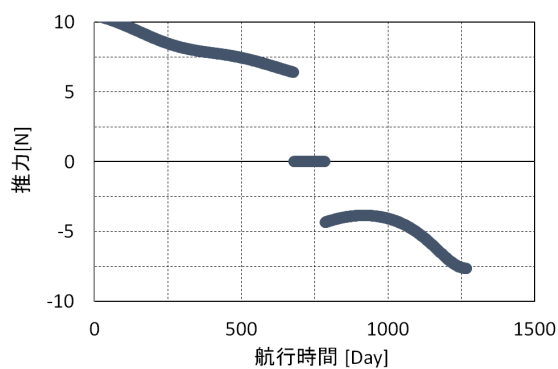


図 4.42 PSO 推力値結果 (有人)

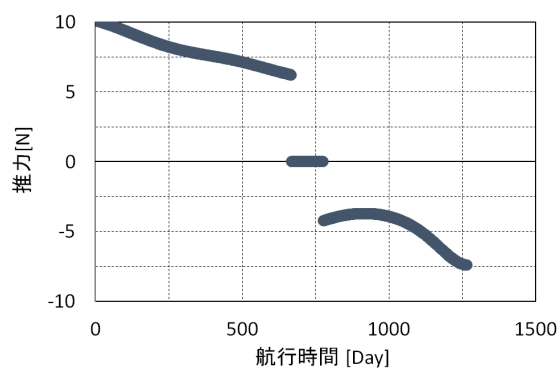


図 4.43 GA 推力値結果 (有人)

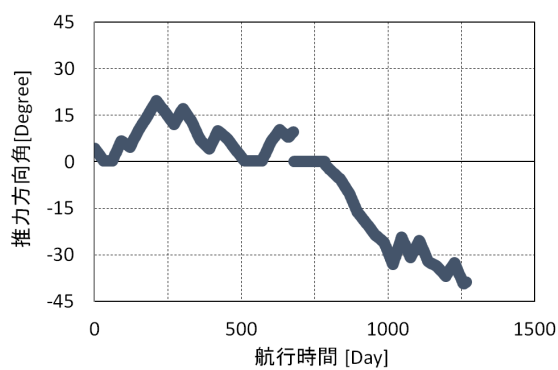


図 4.44 PSO 推力方向角結果 (有人)

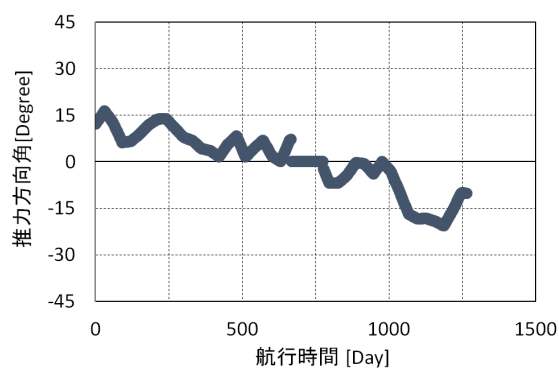


図 4.45 GA 推力方向角結果 (有人)

4.3 有人宇宙探査機の機体構成と惑星間軌道の複合領域最適化

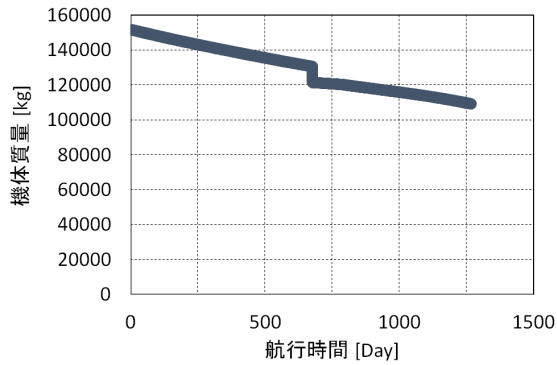


図 4.46 PSO 機体重量結果（有人）

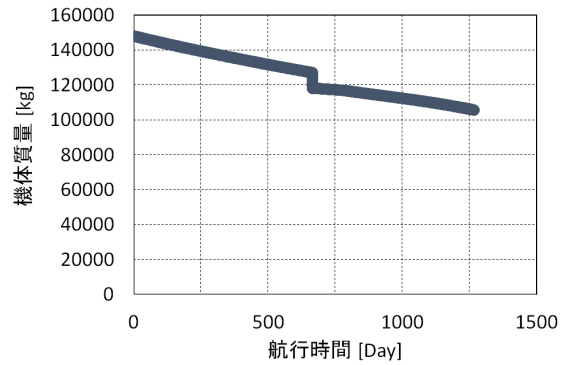


図 4.47 GA 機体重量結果（有人）

表 4.12 PSO 機体諸元結果（有人）

	質量 [kg]	電力 [W]
共通	106.0	109.0
データレコーダ	0.64	1.28
通信	10.2	106.2
姿勢制御（RCS）	1470.8	48.3
姿勢制御（燃料）	19941.8	-
電気推進（機器部）	498.8	201498.6
電気推進（推進剤）	64508.0	-
化学推進系	8493.8	-
バッテリー	148.8	-
太陽電池	22493.5	-
構造系	14311.3	-
居住モジュール	19630.2	10000.0
合計	151602.3	211662.2

表 4.13 GA 機体諸元結果（有人）

	質量 [kg]	電力 [W]
共通	106.0	109.0
データレコーダ	0.64	1.28
通信	10.2	106.2
姿勢制御（RCS）	1446.6	48.3
姿勢制御（燃料）	19452.2	-
電気推進（機器部）	477.1	182193.8
電気推進（推進剤）	64019.7	-
化学推進系	8264.6	-
バッテリー	148.8	-
太陽電池	20442.0	-
構造系	13959.9	-
居住モジュール	19608.8	10000.0
合計	147880.3	192357.4

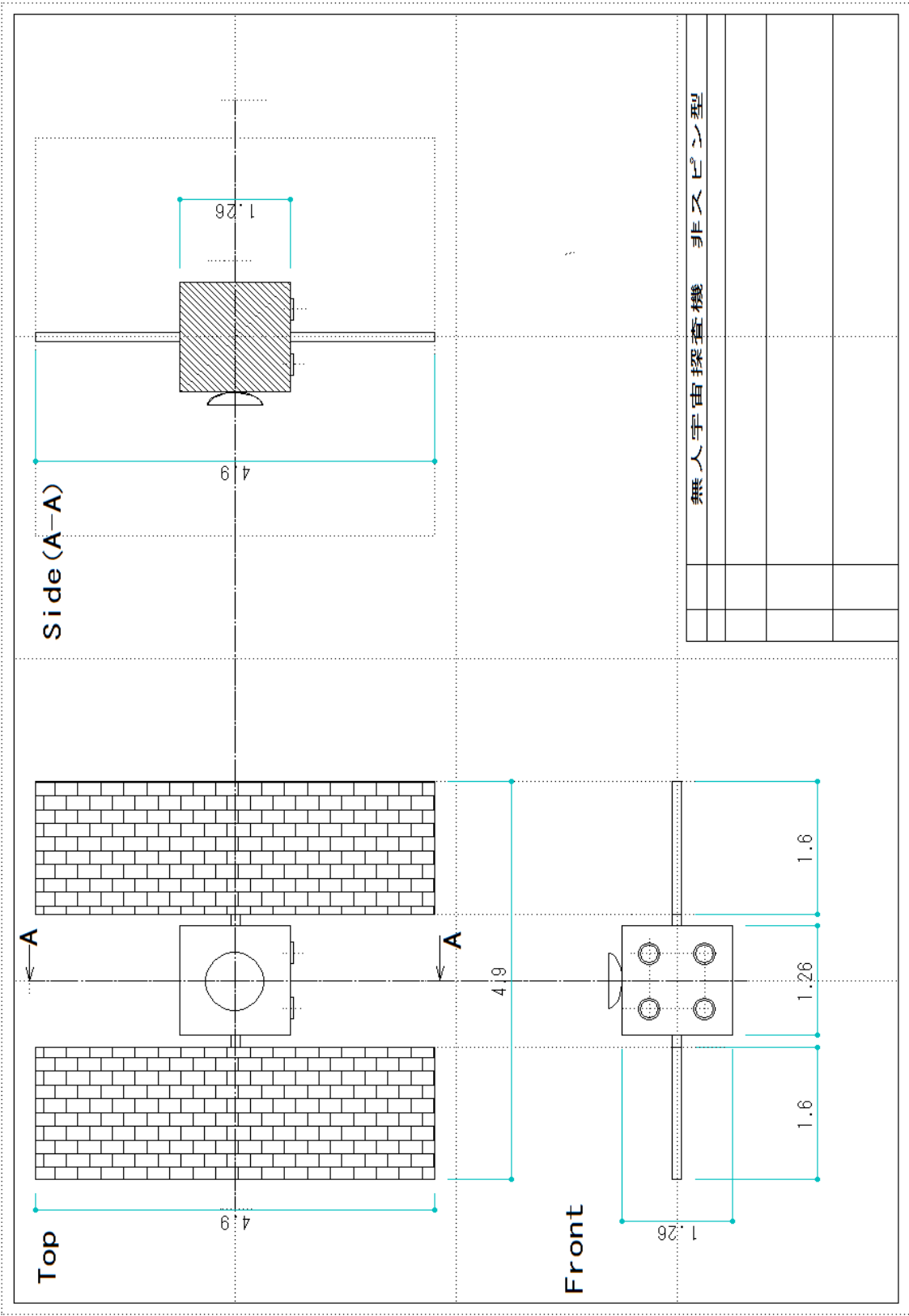


図 4.11 無人宇宙探査機の三面図

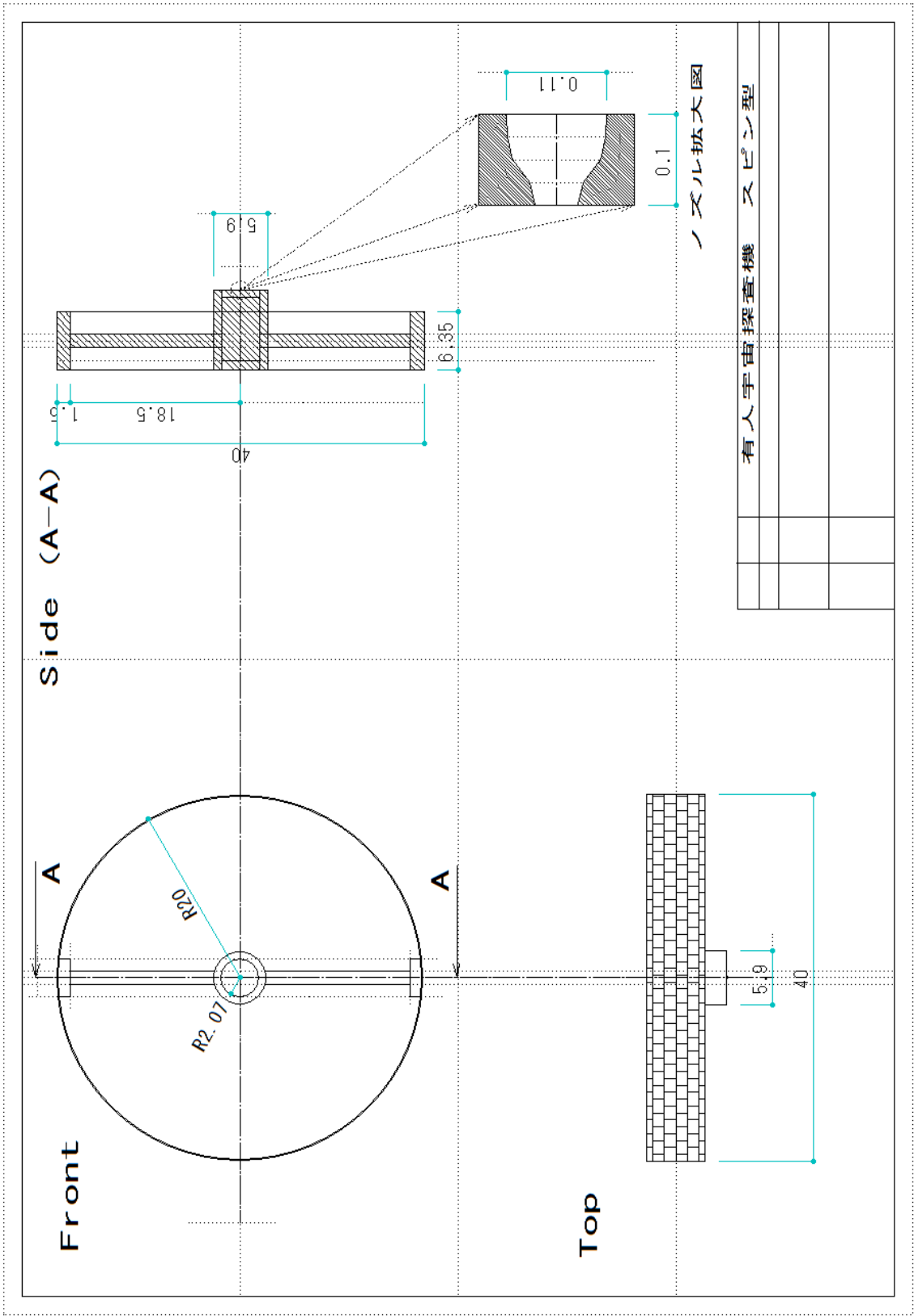


図 4.35 有人宇宙探査機の三面図

第5章 結論

5.1 本論文の結論

本論文では、ヒューリスティックな最適化計算手法について、従来の手法にはない計算アルゴリズムを用いる最適化手法の提案を行った。

提案した手法は銀河の渦と台風の渦の2つからヒントを得ている手法である。本提案手法は4つの特徴となるポイントがある。簡潔にポイントを述べると1. 局所探索は重力収束、2. ローカル脱出は宇宙ジェット、3. コリオリ力での強制移動、4. トーラス面構造。これらのアルゴリズムを実装し、テスト問題を通して提案手法の有効性を示した。

第3章において、以上のアルゴリズムを用いた本提案手法をいくつかのテスト関数に適用した。不静定トラス問題では解析解との誤差0.17%で計算解を得ることができた。基本的な最適化計算アルゴリズムは有効に実装されていることが確認された。数学的テスト関数問題では最適化手法にコリオリ力を導入した場合としない場合について比較を行い、コリオリ力を導入した方が最大で0.3%の範囲で解析解との誤差を改善した。計算アルゴリズムに重力方程式とコリオリ力を用いることで探索点群の移動に曲線的な移動をもたらすことで探索点群に多様性を保持したまま計算を進行できることを確認した。

第4章では、本提案手法を宇宙探査機の機体設計と軌道計画の同時最適化問題に適用した。無人宇宙探査機と有人宇宙探査機の2つのパターンについて計算を行った。無人宇宙探査機ではイオンスラスタを搭載した探査機を地球-火星間を往復する問題として解いた。初期解群の生成領域に制限を設けており、提案手法のローカル解脱出方法である宇宙ジェットのアルゴリズムの有効性を検証した。初期解群から実行不可能領域を飛び越えて新しい実行可能領域に移動することが出来たために、目的関数であるミッション期間を1700日近く改善した。最良解として得られた探査機重量に関する結果は小惑星探査機はやぶさと比較すると重量誤差5%という結果を得られており、宇宙ジェットのアルゴリズムが作用したおかげで有効に最良解を得ることが出来た。有人宇宙探査機では、MPDスラスタを搭載した探査機が地球-火星間を往復する問題とした。上述の無人探査機の設定と比べて有人機は設計変数にスラスタのノズル形状と推力方向角の項目を追加し変数次元数は最大で191次元とした。また、目的関数をミッション期間と全機体質量の2つとし多目的最適化問題とした。提案手法を多目的最適化用に改良し現実的な応用問題にも適用可能であることを示した。

以上の結果から本提案手法が実装可能なアルゴリズムを持ち、現実的な工学的最適化問題に有効に働くことが確認された。本論文を要訳すると、自然現象である銀河形成と台風の渦の中心を求めるアルゴリズムを有する新しいヒューリスティック最適化手法の提案を行った。有効性を検証するための様々な種類の問題に適用し、実装可能なアルゴリズムを有していることを示した。また宇宙工学の複合領域最適化問題にも応用可能であることを示しており、航空宇宙工学上に存在する最適化問題の解決に貢献できると考えている。

5.2 今後の課題

今後の課題として以下の2つを挙げる。提案手法の設定パラメータに関する項目とコリオリ力での座標変換方法に関する項目である。1番目の設定パラメータに関する課題について述べる。本論文では設定パラメータについての指標を提示しているが、実際の計算ではその指標値をもとに数値を前後させて問題によってパラメータ調整を行っている。よりの確なパラメータ指標の提示が必要であると考えられる。また、重力定数 G と回転量パラメータ ω のパラメータ値のバランスも重要項目である。問題設定によって設計変数幅は変わるので、重力定数 G と回転量パラメータ ω の値も変化させる必要がある。パラメータ値の指標を提示する際に設計変数幅が変わる場合でも対応できるように設計変数空間を事前に無次元化することで解決できると考える。無次元化された空間において重力定数 G と回転量パラメータ ω のバランスを探ることでの確なパラメータ指標の提示が行えるようになる。2番目はコリオリ力での座標変換方法に関する課題について述べる。本論文ではコリオリ力を表現するのに極座標変換を行っているが、本来のコリオリ力を再現しきれていない。回転空間における半径方向の速度によってコリオリ力の大きさは決定されるが、極座標変換のみでは半径方向の速度に依存しない。よりコリオリ力の表現に近づけるために、座標変換方法もしくは数学的に直接コリオリ力を計算する方法などを考える必要がある。

5.3 本提案手法の適用傾向

本論文により得られた提案手法の最適化問題に対する適用傾向を表 5.1 と図 5.1 に示す。表 5.1 では各手法の最適化問題の種類に対する得意不得意を表示した。○が得意とするもの、△が普通もしくは適用できなくはない、×は不得意であり適用するには工夫がいるものである。

局所探索に関しては多点探索点同士で速度（移動幅）を求めて次の位置に移るので提案手法と PSO は局所探索に優れている。GA は交叉によって新しい個体を生成するので局所探索は不得意である。多峰性空間の性質を持つ問題に対してはコリオリ力によって多点探索点

群を強制回転させることで解空間を走破させるアルゴリズムを有するので提案手法は多峰性に有効であるといえる。GAは交叉生成で個体が空間を広く移動するので多峰性問題は得意である。PSOは個体群の収束性が早いことが特徴であるが逆に局所解にトラップされやすい。実行不可能領域が広い、飛び飛びに存在する場合は、提案手法とGAは局所解脱出アルゴリズムを有するので適用できなくはない。ただし、問題によって工夫が必要となる。PSOは脱出アルゴリズムを有しないので適用することは難しい。提案手法、PSO、GAは共に多点探索のヒューリスティック手法なので容易に多目的最適化と設計変数が多数の問題に応用可能である。もちろん解析解を求めるには設定パラメータの調整や計算コストの問題がある。論文中には触れなかったが参考として数理計画法の1つであるSQP（逐次二次計画法）も示す。表の◎は数学的証明による最適化計算結果があるものである。数理計画法は目的関数の傾きを使用するために局所探索に特に優れている。ただし、多峰性空間や実行不可能領域が存在する問題には向かない。

図5.1は図的に得意不得意を表示したものである。縦軸に大域的問題に強い局所的問題に強いを示し、横軸に精度が良いか悪いかを示したものである。図の赤い丸にNewと表示しているものが本提案手法である。本論文結果より、PSOの局所探索精度を持ちGAと同等の大域的探索能力があるといえる。局所探索精度に関しては数理計画法が優れているといえる。図中のSAはシミュレーテッドアニーリングでありヒューリスティック手法の1つである。内点法と変分法は数理計画法の種類である。

今回の論文では提案手法が大域的最適化問題に適用できるかを主眼に議論したが、本論文の結果を基に将来的には設計変数同士が強い相関関係を持っている複合領域最適化問題に適用していきたい。

表 5.1 最適化問題の設定傾向による本手法と他手法の比較

	局所的探索	多峰性空間	実行不可能領域 多	多目的最適化	設計変数 多
提案手法	○	○	△	○	○
P S O	○	△	×	○	○
G A	×	○	△	○	○
S Q P	◎	×	×	×	×

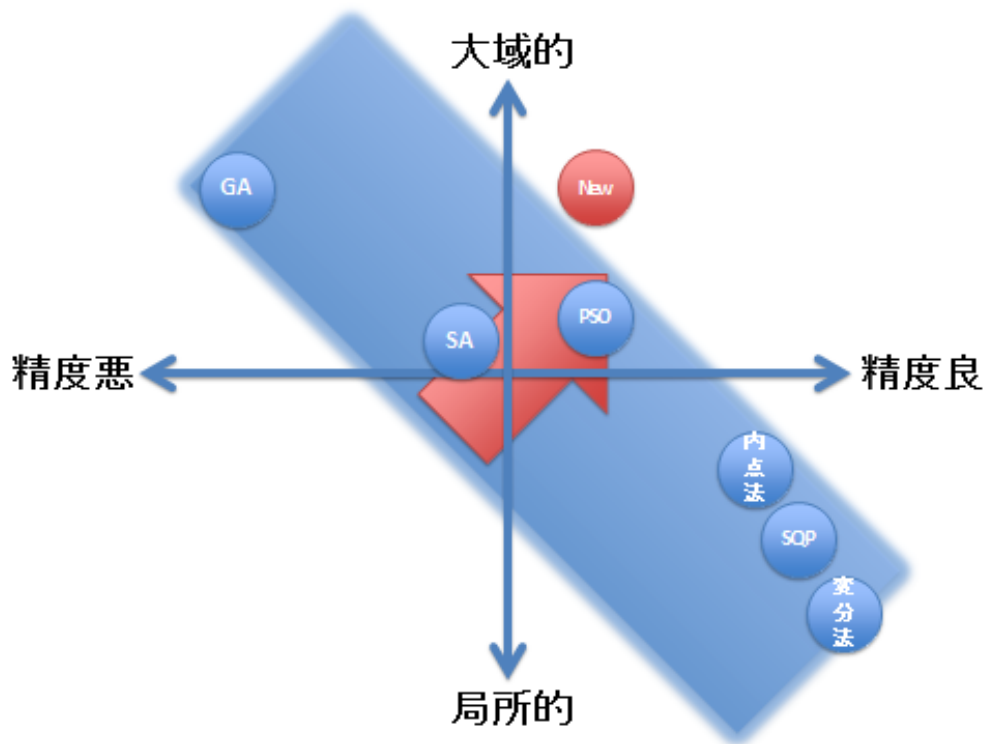


図 5.1 本提案手法と他手法との適用傾向

謝辞

本研究を進めるに当たり多数の人に感謝を述べなくてはなりません。

特に指導教授である土屋武司准教授にはお世話になりました。私のことを見捨てずに丁寧にご指導していただきました。ここで深い謝意を表したいと思います。同じ研究室である鈴木真二教授にもお世話になりました。いくつかのプロジェクトを通して様々なことを教えていただきました。ありがとうございます。

本論文の副査を担当していただいた、東京大学の中須賀真一教授、船瀬龍准教授、宇宙航空研究開発機構の船木一幸教授にも感謝したいと思います。貴重なご意見をありがとうございました。

柄沢研治助手には研究室内の事務系の仕事を担当して頂いたおかげで、研究室生活を円滑に進めることが出来ました。ありがとうございました。

最後に研究室の同期、先輩、後輩にも感謝の意を表したいと思います。私に色々なことを教えて下さりありがとうございました。

参考文献

- [1] 山川宏ほか：最適設計ハンドブック, 朝倉書店, 2003.
- [2] 相吉英太郎, 安田恵一郎（編著）：メタヒューリスティクスと応用, 電気学会, 2007.
- [3] <http://apod.nasa.gov/apod/ap060302.html>
- [4] <http://agora.ex.nii.ac.jp/digital-typhoon/wallpaper/typhoon/>
- [5] <http://apod.nasa.gov/apod/ap031128.html>
- [6] 小野田 崇, 染谷 博司,”ニューラルネット研究の温故知新と最適化手法の研究動向”, 電気学会論文誌C, Vol. 130, No. 1, P 2-5, 2010.
- [7] 半田 久志, 川上 浩司, 片井 修,”進化論的計算手法の進化の系譜”, 電気学会論文誌C, Vol. 128, No. 3, P 334-339, 2008.
- [8] M.Clerc, J.Kennedy:”The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space”, IEEE Trans. Evol. Comput., pp.58-73,2002.
- [9] Yuhui Shi:”Particle Swarm Optimization”,IEEE Connections,2004.
- [10] Yuhui Shi, Russell C. Eberhart:”Empirical Study of Particle Swarm Optimization”,IEEE Evolutionary Computation, 1999.
- [11] K.Yasuda, N.Iwasaki:”Velocity Feedback Adaptive Particle Swarm Optimization”, the 6th Metaheuristics International Conference,pp.947-952, 2005.
- [12] Zhi-Hui Zhan, Jun Zhang, Yun Li, Henry Shu-Hung Chung: ”Adaptive Particle Swarm Optimization”,Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions, 2009.
- [13] Y.Volkan Pehlivanoglu, Oktay Baysal:”Improved Particle Swarm Optimization: Catching the Big Wave on the Surf”, IEEE World Congress on Computational Intelligence, 2012.

-
- [14] Angel E. Munoz Zavala, Arturo Hernandez Aguirre, Enrique R. Villa Diharce: "Constrained Optimization via Particle Evolutionary Swarm Optimization Algorithm (PESO)", GECCO Proceedings of the 2005 conference on Genetic and evolutionary computation, 2005.
- [15] M.R.AlRashidi, M.E.El-Hawary: "A Survey of Particle Swarm Optimization Applications in Electric Power Systems", IEEE Evolutionary Computation Transactions, Vol 13, Issue 4, 2009.
- [16] J. Chen, Z. Ren, and X. Fan, "Particle swarm optimization with adaptive mutation and its application research in tuning of PID parameters", 1st International Symp. on Systems and Control in Aerospace and Astronautics, pp.990-994, 2006.
- [17] 小野 功, 山村 雅幸, 喜多 一: "実数値 GA とその応用", 電気情報通信学会技術報告, AI-2000, pp.61-68.
- [18] H.Bersini, M.Dorigo, S.Langerman, G.Seront and L.Gambardella: "Result of the First International Contest on Evolutionary Optimization", IEEE International Conference on Evolutionary Computation, pp.611-615(1996)
- [19] H.Ishibuchi, T.Yoshida and T.Murata: "Balance between Genetic Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling", IEEE Trans. Evol. , 7, 2, pp.204-223, 2003.
- [20] 榎本 保之, 黒澤 貞男, 中村 高紀, "多目的遺伝的アルゴリズムと実験計画法による水車形状の最適化", 最適化シンポジウム講演論文集 2010(9), "124-1"-"124-5", 2010.
- [21] 河本敏孝, 廣安知之, 三木光範: 5つのテスト関数における自作 GA と ga2k との性能比較, ISDL Report No. 20040810007, 2004.
- [22] J. ホロムコヴィッチ (著), 和田幸一, 増澤利光, 元木光雄 (訳): 計算困難問題に対するアルゴリズム理論, 丸善出版, 2012.
- [23] Zhou Yongsheng, Yang Haoxiong, Zhang Hao, "A Trade-off Pareto Solution Algorithm for Multi-objective Optimization", Computational Sciences and Optimization (CSO), 2012 Fifth International Joint Conference, 2012.
- [24] Alastair Murray, Bjorn Franke, "Compiling for automatically generated instruction set extensions", CGO '12 Proceedings of the Tenth International Symposium on Code Generation and Optimization, Pages 13-22, 2012.

-
- [25] Saeed Ghadimi and Guanghui Lan. , "Optimal Stochastic Approximation Algorithms for Strongly Convex Stochastic Composite Optimization", Shrinking Procedures and Optimal Algorithms. SIAM Journal on Optimization 23:4, 2061-2089 , 2013.
- [26] N. L. Bolland, A. C. Eberhard, F. Engineer, and A. Tsoukalas, " A New Approach to the Feasibility Pump in Mixed Integer Programming ", SIAM Journal on Optimization, 22(3), 831-861, 2012.
- [27] D.M. Goebel, M. Martinez-Lavin, A. Bond and A.M. King 『Performance of XIPS Electric Propulsion in On-Orbit Station Keeping of the Boeing 702 Spacecraft』 (AIAA-2002-4348)
- [28] 狼嘉彰, 富田信之, 中須賀真一, 松永三郎『宇宙ステーション入門 第2版』 (東京大学出版会, 2008 年)
- [29] 岩崎信夫, 的川泰宣『宇宙工学』(日経印刷社, 2010 年)
- [30] 狼嘉彰, 富田信之, 堀川康, 白木邦明『宇宙ステーションと支援技術』 (コロナ社, 2004 年)
- [31] Hargraves, C. R. and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation", Journal of Guidance, Control and Dynamics, Vol.10, No.4, pp.338-342, 1987.
- [32] Kluever, C. A. and Oleson, S. R., "Direct Approach for Computing Near-Optimal Low-Thrust Earth-Orbit Transfers", Journal of Spacecraft and Rockets, Vol.35, No.4, 1998.
- [33] Wu, A. K. and Miele, A., "Sequential Conjugate Gradient-Restoration Algorithm for Optimal Control Problems With Non-Differential Constraints and General Boundary Conditions", Part 1, Journal of Optimization and Applications, Vol.1, pp.69-88, 1980.
- [34] S. Bhaskaran, J. E. Riedel, S. P. Synnott and T. C. Wang, "The Deep Space 1 Autonomous Navigation System: A Post-Flight Analysis", AIAA-2000-3935, 2000.
- [35] L. Rider, "Optimized Polar Orbit Constellations for Redundant Earth Coverage", Journal of the Astronautical Sciences, Vol. 33, No. 2, pp. 147-161, 1985.
- [36] R.Killinger, R.Kukies, M.Surauer and G.Saoccia:"Final Reports on the ARTEMIS Salvage Mision using Electric Propulsion",AIAA-2003-4546, 2003.

-
- [37] S.Bhaskaran, J.E.Riedel, S.P.Synnott and T.C.Wang:””The Deep Space 1 Autonomous Navigation System: A Post-Flight Analysis”,AIAA-2000-3935, 2000.
- [38] MUSES-C 計画概要, 宇宙科学研究所, 1996.
- [39] 宇宙開発事業団特別報告, NASDA SPP-940002-940003, 1994.
- [40] 小惑星探査計画 提案書, 宇宙科学研究所 小惑星探査計画ワーキンググループ, 1995.
- [41] Stephen Kembler:”Interplanetary Mission Analysis and Design”,Springer, 2006.
- [42] <http://www.isas.jaxa.jp/j/forefront/2004/kuninaka/03.shtml>
- [43] <http://k-hiura.cocolog-nifty.com/blog/2009/07/post-5e06.html>, (2001:a space odyssey).
- [44] 國中均, 中山宜典, 西山和孝『イオンエンジンによる動力航行』(コロナ社, 2006 年)
- [45] 栗木恭一, 荒川義博『電気推進ロケット入門』(東京大学出版会, 2003 年)
- [46] Kyoichiro Toki『Optimal Control of Quasi One Dimensional Self Field Magneto plasma dynamic Arcjet Flowfields』(Journal of Propulsion and Power, vol.13, No.1, pp.157-161, 1997)
- [47] <http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-522-space-propulsion-spring-2004/>
- [48] 中根昌克, 今野友和, 池谷弦, 石川芳男, 船木一幸, 都木恭一郎:“ソフト最適化手法による MPD スラスタのノズル形状最適化”, 日本航空宇宙学会論文集, Vol. 54, pp.360-366,(2006).
- [49] 窪田健一, 船木一幸, 奥野喜裕:“MPD スラスタ内における電熱・電磁加速モード間の遷移に関する研究”, 日本航空宇宙学会論文集, Vol.54, pp.535-541,(2006).
- [50] A.C.Malliaris, R.R.John, R.L.Garrison, D.R.Libby: ” Performance of Quasi-Steady MPD Thrusters at High Powers ”, AIAA Journal, Vol.10, No.2.
- [51] Kuriki.K, Suzuki.H: “Thrust measurement of a quasisteady MPD arcjet”, AIAA-1976-1002.
- [52] Coralia Cartis, Nicholas I. M. Gould, and Philippe L. Toint, “On the Evaluation Complexity of Composite Function Minimization with Applications to Nonconvex Nonlinear Programming”, SIAM Journal on Optimization, 21(4), 1721-1739, 2011.

-
- [53] Sebastian Sager, Carola M. Barth, Holger Diedam, Michael Engelhart, and Joachim Funke, “Optimization as an Analysis Tool for Human Complex Problem Solving”, SIAM Journal on Optimization, 21(3), 936-959, 2011.
- [54] Luigi Di Puglia Pugliese and Francesca Guerriero, “Dynamic programming approaches to solve the shortest path problem with forbidden paths”, Optimization Methods and Software, Volume 28, Issue 2, pages 221-255, 2013.
- [55] Zhaosong Lu, “Robust portfolio selection based on a joint ellipsoidal uncertainty set”, Optimization Methods and Software, Volume 26, Issue 1, pages 89-104, 2011.
- [56] O. P. Ferreira, S. Z. Nemeth, “Generalized projections onto convex sets”, Journal of Global Optimization, Volume 52, Issue 4, pp 831-842, 2012.
- [57] 矢部 孝, 尾形 陽一, 内海 隆行: “CIP 法”, 森北出版, 2003.
- [58] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery(著), 丹慶勝市, 奥村晴彦, 佐藤俊郎, 小林誠(訳): “NUMERICAL RECIPES in C”, 技術評論社, 1993
- [59] 大島 裕子, 小泉 孝之, 辻内 伸好, 足立 渡, “応答曲面による最適設計のための関数近似と実験計画法”, 最適化シンポジウム講演論文集 2010(9), ”201-1”-”201-6”, 2010.