

博士論文

遺伝的アルゴリズムによる
多目的最適化に関する研究

Genetic Algorithm
for Multi-Objective Optimization

2001

渡邊 真也

Shinya Watanabe

2003年3月

同志社大学大学院 工学研究科 知識工学専攻 博士論文

指導教員 三木 光範教授

知的システムデザイン研究室

目次

第 1 章 序論	1
1.1 本論文の目的	1
1.1.1 探索効率の優れた多目的遺伝的アルゴリズムの提案と実装	2
1.1.2 実問題に対するアルゴリズムの有効性の検証	3
1.2 本論文の構成	3
第 2 章 多目的最適化	5
2.1 まえがき	5
2.2 多目的最適化	5
2.2.1 多目的最適化問題の定義	6
2.2.2 パレート最適解	6
2.3 多目的最適化手法	7
2.3.1 重み係数法	7
2.3.2 制約法	8
2.3.3 辞書式配列法	8
2.4 まとめ	8
第 3 章 多目的遺伝的アルゴリズム	11
3.1 はじめに	11
3.2 遺伝的アルゴリズム	12
3.3 遺伝的アルゴリズムによるパレート最適解生成法	16
3.3.1 多目的遺伝的アルゴリズムの分類	16
3.3.2 VEGA	17

3.3.3	WBGA	19
3.3.4	MOGA	23
3.3.5	NSGA	26
3.3.6	NPGA	28
3.3.7	SPEA	30
3.3.8	NSGA-II	34
3.3.9	SPEA2	38
3.4	多目的遺伝的アルゴリズムの問題点	42
第 4 章	得られた非劣解集合に関する評価方法	45
4.1	はじめに	45
4.2	評価方法	46
4.2.1	パレート最適フロントに対する誤差 (error : I_{error})	46
4.2.2	被覆率 (cover rate: I_{cover})	47
4.2.3	各目的関数軸ごとの最大値, 最小値, 平均値 (Max and Min and Average: I_{MMA})	48
4.2.4	優越個体割合 (Ratio of Non-dominated Individuals: I_{RNI})	48
4.2.5	パレートフロントとサンプリング直線の交点にもとづく評価手法 (Sampling of the Pareto Frontier Lines of Intersection: I_{LI})	49
4.3	まとめ	51
第 5 章	多目的遺伝的アルゴリズムの分割母集団モデルの検討	53
5.1	はじめに	53
5.2	並列 GA モデル	54
5.2.1	マスタースレーブモデル	55
5.2.2	分割母集団モデル	55
5.2.3	近傍モデル	56
5.3	全体シェアリング遺伝的アルゴリズム	57
5.3.1	全体シェアリング GA のアルゴリズム	58
5.3.2	数値実験	60
5.3.3	実験結果および考察	61
5.3.4	まとめ	65
5.4	領域分割型多目的遺伝的アルゴリズム	65
5.4.1	領域分割型多目的 GA のアルゴリズム	65

5.4.2	数値実験	67
5.4.3	まとめ	72
5.5	分散協力型スキーム	72
5.5.1	分散協力型スキームの特徴	73
5.5.2	アルゴリズム	74
5.5.3	数値実験	76
5.5.4	結果	77
5.5.5	まとめ	80
5.6	まとめ	80
第 6 章 近傍培養型遺伝的アルゴリズム		83
6.1	はじめに	83
6.2	多目的遺伝的アルゴリズムにおける重要なスキーム	84
6.3	近傍培養型遺伝的アルゴリズムの概要	86
6.4	数値実験	88
6.4.1	対象問題	89
6.4.2	離散問題	91
6.4.3	GA の構成・GA パラメータ	91
6.4.4	各手法比較の結果	92
6.4.5	NCGA におけるソート基準の検討実験	106
6.5	まとめ	115
第 7 章 近傍培養型遺伝的アルゴリズムを用いた問題解決		117
7.1	はじめに	117
7.2	ディーゼルエンジン噴射スケジュールの多目的最適化	118
7.2.1	ディーゼル燃焼モデル	119
7.2.2	数値実験	121
7.2.3	結論	128
7.3	矩形ブロック最小面積配置の多目的最適化	128
7.3.1	矩形パッキング問題	129
7.3.2	数値実験	132
7.3.3	結論	142
7.4	まとめ	144

第 8 章 結論	145
8.1 本論文の成果	145
8.2 今後の課題	147
謝辞	149
参考文献	149
付録 A 研究業績	157

图 目 次

2.1	The concept of Pareto optimal solution	7
3.1	Schematic of GA	13
3.2	Flowchart of GA procedure	14
3.3	Schematic of GA search	15
3.4	Schematic of VEGA	18
3.5	Distribution of Pareto Solutions in VEGA	19
3.6	The concept of Pareto-ranking	24
3.7	The concept of Non-dominated Sorting method	27
3.8	The SPEA fitness assignment scheme	33
3.9	Schematic of the NSGA-II procedure	36
3.10	The crowding distance calculation	37
3.11	Comparison of fitness assignment schemes in SPEA and SPEA2	40
3.12	Archive truncation method used in SPEA2	42
4.1	Schematic of I_{cover}	47
4.2	An example of I_{MMA}	48
4.3	Schematic of $I_{RNI(X,Y)}$	49
4.4	Schematic of I_{LI}	50
5.1	Schematic of master slave model	55
5.2	Schematic of distributed population model	56
5.3	Schematic of neighborhood model	57
5.4	Schematic of the Total Sharing procedure	60
5.5	Pareto optimum individuals (T2)	62
5.6	Pareto optimum individuals (T4)	63
5.7	I_{error} of T2 and T4	63

5.8	I_{cover} of T2 and T4	64
5.9	Schematic of DRMOGA	66
5.10	Pareto optimum individuals(NDP)	68
5.11	Pareto optimum individuals(VB3)	69
5.12	Pareto optimum individuals ($f_2 - f_3$, VB3)	69
5.13	I_{error} of NDP	70
5.14	I_{cover} of NDP and VB3	70
5.15	Schematic of DC-Scheme	75
5.16	Pareto optimum individuals (ZDT4)	77
5.17	Pareto optimum individuals (KUR)	78
5.18	I_{error} of ZDT4	78
5.19	I_{cover} of ZDT4 and KUR	79
6.1	Pareto-optimal front of F_{discon}	90
6.2	I_{cover} and I_{error} of F_{discon} ($N = 10$)	92
6.3	I_{MMA} of F_{discon} ($N = 10$)	93
6.4	I_{RNI} and I_{LI} of F_{discon} ($N = 10$)	93
6.5	Pareto optimum individuals(F_{discon} ($N = 10$))	94
6.6	I_{cover} and I_{error} of F_{discon} ($N = 100$)	94
6.7	I_{MMA} of F_{discon} ($N = 100$)	95
6.8	I_{RNI} and I_{LI} of F_{discon} ($N = 100$)	95
6.9	Pareto optimum individuals(F_{discon} ($N = 100$))	96
6.10	I_{cover} and I_{error} of ZDT4	97
6.11	I_{MMA} of ZDT4	97
6.12	I_{RNI} and I_{LI} of ZDT4	97
6.13	Pareto optimum individuals(ZDT4)	98
6.14	I_{cover} and I_{error} of ZDT6	99
6.15	I_{MMA} of ZDT6	99
6.16	I_{RNI} and I_{LI} of ZDT6	100
6.17	Pareto optimum individuals(ZDT6)	100
6.18	I_{cover} of KUR	101
6.19	I_{MMA} of KUR	101
6.20	I_{RNI} and I_{LI} of KUR	102
6.21	Pareto optimum individuals(KUR)	102

6.22	I_{cover} of KP750 – 2	104
6.23	I_{MMA} of KP750 – 2	104
6.24	I_{RNI} and I_{LI} of KP750 – 2	104
6.25	Pareto optimum individuals(KP-2)	105
6.26	I_{cover} of p-NCGA	107
6.27	I_{error} of p-NCGA	108
6.28	I_{RNI} and I_{LI} of F_{discon}($N = 10$)	109
6.29	Pareto optimum individuals(F_{discon}($N = 10$))	109
6.30	I_{RNI} and I_{LI} of F_{discon}($N = 100$)	110
6.31	Pareto optimum individuals(F_{discon}($N = 100$))	110
6.32	I_{RNI} and I_{LI} of ZDT4	111
6.33	Pareto optimum individuals(ZDT4)	111
6.34	I_{RNI} and I_{LI} of ZDT6	112
6.35	Pareto optimum individuals(ZDT6)	112
6.36	I_{RNI} and I_{LI} of KUR	113
6.37	Pareto optimum individuals(KUR)	113
7.1	Optimization System	121
7.2	Coding method	123
7.3	Derived non-dominated solutions	124
7.4	Derived non-dominated solutions(SFC,NOx)	125
7.5	Derived non-dominated solutions(SFC,Soot)	125
7.6	Derived non-dominated solutions(NOx,Soot)	126
7.7	Concept of sequence-pair	130
7.8	Coding example of sequence-pair	130
7.9	Horizontal/Vertical Constrain graphs	131
7.10	Placement-based Partially Exchanging Crossover(PPEX)	133
7.11	Results of I_{LI} (33 blocks)	134
7.12	I_{MMA} of 33 blocks	135
7.13	Derived non-dominated solutions(33 blocks)	135
7.14	Results of I_{LI} (50 modules)	136
7.15	Results of I_{LI} (100 modules)	137
7.16	I_{MMA} of 50 modules	137
7.17	I_{MMA} of 100 modules	137

7.18 Derived non-dominated solutions(50 modules)	138
7.19 Derived non-dominated solutions(100 modules)	139
7.20 Results of I_{LI} (500 blocks)	139
7.21 I_{MMA} of 500 blocks	140
7.22 Derived non-dominated solutions(500 blocks)	140
7.23 The placement of the modules(33 modules)	142
7.24 The placement of the modules(100 modules)	143

表 目 次

3.1	A brief history of EMO	17
3.2	A example of weight vector Table	20
5.1	GA parameters	61
7.1	Specification of the target diesel engine	122
7.2	GA parameters	123
7.3	Cluster System	124
7.4	Calculation time	127
7.5	GA Parameter	133

第 1 章

序論

1.1 本論文の目的

最適化とは、与えられた制約条件下において何らかの評価指標を最良にすることである。この最適化の考えは、工学・産業・経済などの非常に多くの分野において深く関わる概念であり、我々の生活の中においても重要な役割を果たしている。

一般に、最適化とはある 1 つの評価 (目的) に対する最適化を行う単一目的最適化のことを意味する。しかしながら、本来多くの問題においてその評価基準は唯一とは限らない。例えば、ある製品を評価する場合、製品の機能、価格、外見、重量、大きさなど評価基準は複数に及ぶ。しかも、評価基準は何らかの形で互いに相反するトレードオフの関係にあることが多く、全ての評価基準が最適の製品は存在しない。このような複数の評価基準が存在し、評価基準が互いにトレードオフの関係にある問題を多目的最適化問題と呼ぶ。

多目的最適化問題では、単一目的の場合と異なり唯一の最適解を得ることは難しい。これは、複数の評価基準がトレードオフの関係にある場合に、一方の評価の改善が他方の改悪になってしまうからである。そのため、多目的最適化では、「パレート最適解」という概念を用いて解探索を行う¹⁾。パレート最適解とは「ある目的関数の値を改善するためには、少なくとも他の 1 つ目的関数の値を改悪せざるを得ないような解」と定義されており、複数、場合によっては無限に存在する。

従来の多目的最適化問題に対する手法として、複数の目的関数を任意の重み付けにより単一化する重みパラメータ法、ある 1 つの目的関数以外を全て制約条件化し単一目的化する ϵ 制約法などが提案されている。しかしながら、これらの手法は複数もしくは無限にあるパレート最適解集合の中のある 1 つの解しか求めることができず、何らかの形で各評価項目の優先度を定義する必要がある。

こういった問題点を解決するための新たな多目的最適化手法として、進化的計算を多目的最適化へ応用した進化的多目的最適化 (Evolutionary Multi-Objective Optimization:

EMO) が近年、非常に盛んに行われ大きな進歩を見せている²⁻⁹⁾。多くの EMO アルゴリズムでは、上記の問題点を解決しており、各評価項目の優先度を明示的に定義することなく 1 度の探索でパレート最適解集合を探索することが可能である。

この分野では、様々な進化的アルゴリズムが適用されているが、特に遺伝的アルゴリズム (Genetic Algorithm: GA) を多目的最適化問題に適用した多目的 GA は最も数多く研究されている²⁻⁹⁾。これは、多点探索という特徴を持つ GA では、探索対象となる複数のパレート最適解を一度の探索によって求めることができるためである。一方で、多目的 GA では、単一目的 GA の場合とは異なる個体の適合度の割り当てや母集団の多様性の保持といったメカニズムを新たに導入する必要がある。こういった解決すべき課題が数多く存在することも研究されている理由の 1 つである。

近年、様々な多目的 GA に関するアルゴリズムやその適用事例が盛んに報告されているが²⁻⁹⁾、それらの研究の中でも特に、Deb らの NSGA-II⁸⁾、Zitzler らの SPEA2⁹⁾ などは、それまでに提案されてきた多目的 GA のアルゴリズムに比べ良好な結果を示している。これらのアルゴリズムでは、探索途中で発見した優良解の保存、適切なパレート解候補の削減など探索において重要なメカニズムが実現されている。しかしながら、アルゴリズムにはまだ改良の余地が残されている上、適用されている例題がテスト関数のみ、もしくはある特定の実問題に限定されている場合がほとんどである。

そこで、本論文では、より高実用性を志向した多目的 GA アルゴリズムの提案とその有効性の検証を試みた。本論文は、以下の 2 つのアプローチに基づいている。

- 探索効率の優れた多目的 GA アルゴリズムの提案と実装。
- 実問題に対するアルゴリズムの有効性の検証。

以下、これら 2 つのアプローチについて説明する。

1.1.1 探索効率の優れた多目的遺伝的アルゴリズムの提案と実装

本研究では、これまであまり研究されていない多目的 GA の分割母集団モデルに対する様々な手法の検討を行った。これらの手法は、多目的 GA を分割母集団モデルへ適用する際の問題点を考慮するような手法である。我々は、これらのモデルの実験結果より、近傍付近の個体どうしで交叉を行う近傍交叉が多目的 GA の探索において大きな影響を与えることを発見した。

そこで、この近傍交叉とこれまでに提案されてきた優れた手法の持つ効果的なメカニズムを組み合わせた新たな多目的 GA、近傍培養型 GA (Neighborhood Cultivation GA : NCGA) の提案を行い、テスト関数を用いた数値実験によりその有効性の検証を試みた。

NCGA にて実装されている近傍交叉では、目的関数空間において隣り合う 2 つの個体を用いて交叉を行う。一般に、大域的な探索を行う多目的 GA では探索個体どうしの目的関

数空間距離が大きく離れ、効果的な交叉を行うことができない。しかし、近傍交叉を行うことにより、意味のない交叉を未然に防ぐことができ、結果として探索効率の向上を実現することができる。

1.1.2 実問題に対するアルゴリズムの有効性の検証

提案手法の高実用性を検証するために、2つの性質の異なるより実問題に近い対象問題に対して NCGA の適用を試みた。実験に用いたのは以下の2つの問題である。

i) ディーゼルエンジン噴射スケジュールの最適化

ディーゼルエンジン噴射スケジュール最適化問題は、ディーゼル燃焼を改善するためにディーゼルエンジン燃料の最適な噴射率を求める問題である。ディーゼル燃焼の改善のために考慮すべきことは、排気特性と熱効率であり、すなわち燃費率の効率化、NO_x、すすの低減化が目的となる。従来までの研究の多くが、これら3目的のうち1つの目的のみに注目した単一目的最適化だったのに対して、ここでは、燃費率、NO_x排出量、すす排出量の最小化を目的とする3目的最適化問題として扱った。

ii) 矩形ブロックの2次元空間上での配置面積の最小化

矩形ブロックの配置面積の最小化は、あらかじめ定められた複数の矩形部ブロックを2次元空間上に配置する問題であり、配置面積の縦、横の長さの最小化を目的とする2目的最適化問題として用いた。配置面積の最小化は、組み合わせ最適化問題の1つであり、扱うブロック数によって可能な組み合わせが指数的に増加するという特徴を持っている。配置面積の縦、横の長さを最小化することで、単に面積の最小化だけでなく解選考者に対して様々なアスペクト比を持つ最小面積を提示することができる。

上記の2つの問題は、性能評価のためのテスト問題と異なり、非常に複雑で解探索の困難な問題である。これら性質の異なるより実問題に近い対象問題に対して、NCGA を適用し、NCGA の高実用性に関する検証を行った。

1.2 本論文の構成

本論文の構成について述べる。本論文は、8章から構成されている。

第1章は序論であり、多目的GAの背景と本研究の位置づけについて説明している。

第2章では、多目的最適化問題およびパレート最適解に関する数学的な定義について概説し、多目的最適化問題に対する代表的なスカラー化手法について説明している。2章で取り上げたスカラー化手法は、重み係数法、制約法、および辞書式配列法の4手法である。

第 3 章では、多目的 GA および GA の概説と幾つかの代表的な多目的 GA の手法を取り上げ、その特徴について言及している。3 章では、代表的な手法として、Schaffer らの VEGA, Hajela らの WBGA, Fonseca らの MOGA, Horn らの NPGA, Srinivas らの NSGA, Deb らの NSGA-II, Zitzler らの SPEA および SPEA2 について説明している。

第 4 章では、多目的 GA により得られた解集合に対する評価方法の説明を行っている。多目的 GA では、得られる非劣解集合が複数存在する上、一意的に解の評価を行うことが出来ない。そのため、非劣解に対する評価方法が不可欠となる。4 章では、得られた解集合に求められる解の性質について述べるとともに、それらの性質を評価するための幾つかの方法について説明を行っている。なお、5 章以降の数値実験では、4 章において説明した評価手法が用いられている。

第 5 章では、これまであまり研究されていない多目的 GA の分割母集団モデルに対する様々な手法の検討を行っている。多目的 GA の並列モデルについての研究は幾つか行われているものの、その多くは単一目的における GA の並列化とほぼ同様に、並列化の際に多目的の特性を考慮しているモデルはほとんどない。そこで、5 章では、分割母集団モデルの並列多目的 GA として全体シェアリング、領域分散型 GA、分散協力型メカニズムの 3 つ手法について説明を行っている。また、これらの手法の性能を評価するため従来手法との数値実験および結果の考察についても述べている。

第 6 章では、新たな多目的 GA アルゴリズムとして NCGA の提案とその有効性の検証について述べている。6 章では、まず NSGA-II, SPEA2 といったこれまでに提案されてきた優れた多目的 GA に共通する探索に効果的なメカニズムを明らかにしている。その上で、これらの効果的なメカニズムと、近傍交叉という独自のメカニズムを合わせ持った新たなアルゴリズムとして、NCGA の提案を行っている。また、数値実験として、幾つかの代表的なテスト問題に対する NCGA の適用を行い、NSGA-II, SPEA2 との比較を試みた。

第 7 章では、6 章において提案された NCGA の実問題への応用についての検討を行った。7 章では、より実問題に近い対象問題として、ディーゼルエンジン噴射スケジュールの最適化と矩形ブロックの 2 次元空間上での配置面積の最小化を取り上げ、NCGA の適用を試みた。

第 8 章では、本研究のまとめとして結論を述べている。

第 2 章

多目的最適化

2.1 まえがき

一般に、最適化とはある 1 つの評価 (目的) に対する最適化を行う単一目的最適化のことを意味する。しかしながら、実世界に存在する様々な最適化問題を考えた場合、複数の評価基準を同時に考慮すべき問題は少なくない。このように、複数の評価基準が存在し、これらの評価基準を同時に考慮しながら最適解を探索する問題を多目的最適化問題と呼ぶ。

多くの多目的最適化問題では、評価基準の間に何らかのトレードオフの関係があり、単一の最適解を得ることは難しい。そのため、多目的最適化ではパレート最適解という別の概念を用いて解探索を行うことになる。従来より、パレート最適解を得るための手法として、何らかの方法により複数存在する評価基準を単一目的化するスカラー化手法が用いられてきた。

本章では、多目的最適化問題の定式化とパレート最適解の概念について解説する。さらに、パレート最適解を求めるために従来から用いられてきた幾つかのスカラー化手法について説明する。

2.2 多目的最適化

多目的最適化問題¹とは「複数個の互いに競合する目的関数を与えられた制約条件の中で何らかの意味で最小化 (最大化) する問題」と定義されている¹⁾。目的関数が互いに競合しあっているため、全ての目的関数の値が最良であるような最適解を求めることはできない。

そのため、多目的最適化では「ある目的関数の値を改善するためには、少なくとも他の 1 つの目的関数値を改悪せざるを得ないような解」を求めていく。多目的最適化では、このような解をパレート最適解 (Pareto-optimal solution) と呼んでいる。以下、多目的最

¹英訳「Multiobjective Optimization Problems」から MOPs とも呼ばれる。

適化問題およびパレート解の定義を示す.

2.2.1 多目的最適化問題の定義

一般に多目的最適化問題は, n 個の設計変数を扱う, k 個の互いに競合する目的関数

$$f_i(x_1, x_2, \dots, x_n) \quad (i = 1, 2, \dots, k) \quad (2.1)$$

を, m 個の不等式制約条件

$$g_j(x_1, x_2, \dots, x_n) \leq 0 \quad (j = 1, 2, \dots, m) \quad (2.2)$$

のもとで最小化 (最大化) する問題として定式化される¹⁾.

多目的最適化問題では, 一般に全ての目的関数 $f_i(x)$ を同時に最小化することはできない. これは, 目的関数間にトレードオフの関係が存在するためである. そのため, 多目的最適化問題では全ての目的において最良な値をとる最適解は一般には存在しない.

そこで, 多目的最適化問題では最適解の代わりに新たな解の概念として, パレート最適解 (Pareto-optimal solution) を用いる. このパレート最適解の概念は, 経済学者 Pareto によって初めて定義された概念である¹⁾.

2.2.2 パレート最適解

パレート最適解は, 多目的最適化問題における解の優越関係により定義される. 多目的最適化問題における解の優越関係の定義を以下に示す. ただし, 全ての目的が最小化であると仮定する.

定義 (優越関係): $\mathbf{x}^1, \mathbf{x}^2 \in \mathfrak{S}(\mathbf{x} = (x_1, x_2, \dots, x_n))$ とする.

- a) $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ ($\forall i = 1, \dots, k$) の時, $\mathbf{x}^1$ は $\mathbf{x}^2$ に優越するという.
- b) $f_i(\mathbf{x}^1) < f_i(\mathbf{x}^2)$ ($\forall i = 1, \dots, k$) の時, $\mathbf{x}^1$ は $\mathbf{x}^2$ に強い意味で優越するという.

もし, $\mathbf{x}^1$ が $\mathbf{x}^2$ に優越しているならば, $\mathbf{x}^1$ の方が $\mathbf{x}^2$ より良い解である. そのため, 多目的最適化では, このような他のどの解にも優越されないような解の探索を行う²⁾. 次にこの優越関係に基づくパレート最適解の定義について以下に示す.

定義 (パレート最適解): $\mathbf{x}^0 \in \mathfrak{S}$ とする.

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

²⁾このような他のどの個体と比較しても劣っていない解を非劣解と呼ぶ.

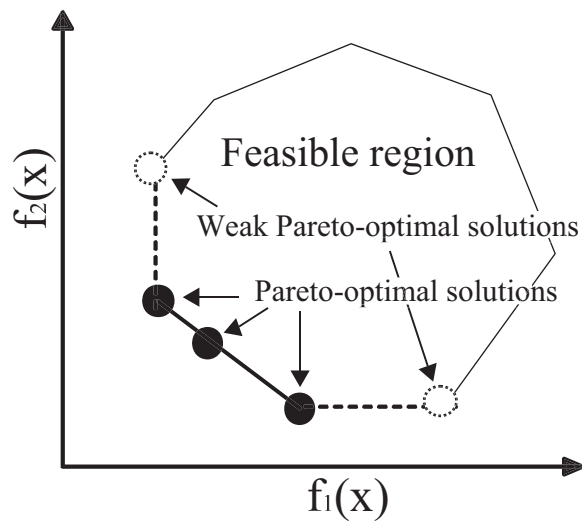


図 2.1 The concept of Pareto optimal solution

- b) $\mathbf{x}^0$ に優越する $\mathbf{x} \in \mathfrak{S}$ が存在しないとき, $\mathbf{x}^0$ をパレート最適解 (Pareto-optimal solution) という³.

目的関数が 2 つの場合におけるパレート最適解の例を図 2.1 に示す. 図中, 黒丸がパレート最適解を, 破線で描かれた白丸が弱パレート最適解をそれぞれ示している. 一般に, パレート最適解集合が形成する面のことをパレート最適フロントと呼ぶ (図 2.1 の実線部分).

2.3 多目的最適化手法

多目的最適化問題におけるパレート最適解は, 複数存在する目的関数を何らかの工夫により単一目的化することで求めることができる. これは, 単一目的化された目的関数の最適解をパレート最適解集合の 1 つとして対応づけすることができるためである. 一般に, この手法はスカラー化手法と呼ばれる. 以下, 代表的なスカラー化手法として, 重み係数法, 制約法, 重み付けミニマックス法, および辞書式配列法について説明する.

2.3.1 重み係数法

重み係数法 (weight method) は, 重み係数 w_i を用いて各目的関数に重みを設定し, 得られる加重和を単一の目的関数 $\mathbf{w}f(x)$ とし, パレート最適解の 1 つを求める手法である. なお, $\mathbf{w} = (w_1, w_2, \dots, w_k)$ である.

³弱パレート最適解と明確に区別するため, 強パレート最適解とも呼ばれる.

$$\min_{x \in X} \mathbf{w}f(x) = \sum_{i=1}^k w_i f_i(x) \quad (2.3)$$

$$w_i \geq 0 \quad \sum_{i=1}^k w_i = 1 \quad (2.4)$$

w を可変パラメータとして変化させながら解探索を繰り返すことにより，目的関数空間におけるパレート最適フロントの形状が凸の場合には，すべてのパレート最適解を得ることが可能である．しかし，非凸の場合にはギャップが生じ，すべてのパレート最適解を求めることができない¹⁾．

2.3.2 制約法

制約法 (または， ε 制約法) (constraint method) はある 1 つの目的関数以外の目的関数を制約条件に変換するというスカラー化手法である．すなわち，任意の $f_j(x)$ のみを目的関数とし，残りの $(k-1)$ 個の目的関数には上限値 $\varepsilon_i (i = 1, 2, \dots, k, i \neq j)$ を設定して， ε 制約とよばれる不等式制約に変換する．そして，以下の制約問題を解くことにより，パレート最適解を求める手法である．

$$\min_{x \in X} f_j(x) \quad (2.5)$$

$$\text{subject to } w_i f_i(x) \leq \varepsilon_i, i = 1, 2, \dots, k, i \neq j \quad (2.6)$$

i や ε_k を順次変化させることで，目的関数空間におけるパレート最適フロントの形状が非凸の場合でも，すべてのパレート最適解を得ることができる．

2.3.3 辞書式配列法

辞書式配列法では，目的関数に優先順位を付け，その優先順位に従って解探索を行う．すなわち， f_1 が一番優先される場合， f_1 のみによって順位付けを行い， f_1 が同じ場合には，その次に優先される f_2 により，さらに同じ場合には f_3 によるというように解を求める手法である．

この手法では，明らかに先に採用される目的関数が重視されるため，その優先順位の決め方が重要である．

2.4 まとめ

多目的最適化では，2.2.2 節において定義されているパレート最適解を求めることが第 1 の目標となる．しかし，単一目的の場合と異なり目的関数が複数存在するため，単一目的最

適化手法をそのまま多目的へ用いることはできない。そのため、従来より多目的を何らかの形で単一目的化して最適化を適用するスカラー化手法が提案され、適用されてきた。スカラー化手法には、複数存在する評価を重み和を用いて単一目的化する重み係数法、ある 1 つの目的のみを対象とし他の目的を制約条件として扱う制約法など様々な方法が存在する。

しかし、これらの手法に共通するのは 1 度の探索でパレート最適解集合の 1 つしか求めることができない点である。しかも、これらのスカラー化手法では各評価項目に対する解選考者の何らかの重み付け、もしくは順位付けを事前に決定する必要がある。一般に、多目的最適化では各評価項目を統合して扱うことができない。また、各評価項目の優先度を定義できない場合が多い。そのため、各評価項目の優先度をあらかじめ定義する必要がある、かつ 1 度の探索でパレート最適解集合の 1 つしか求められないスカラー化手法は、多目的最適化問題を解く上で、最適な手法とはいえない。

そこで、本研究では次章で説明する遺伝的アルゴリズム (Genetic Algorithm: GA) を用いた多目的最適化手法に注目した。GA を用いた多目的最適化では、各評価項目の優先度をあらかじめ定義する必要もなく、かつ 1 度の探索で複数のパレート最適解を求めることが可能である。

第 3 章

多目的遺伝的アルゴリズム

3.1 はじめに

パレート最適解集合を求めるための手法として考案されたスカラー化手法には、以下の 2 つの問題点が存在する。

- 各評価項目の優先度を定義する必要がある。
- 1 度の探索でパレート最適解集合の 1 つしか求められない。

それらの問題点を解決するための新たな多目的最適化手法として、進化的計算 (Evolutionary Computation: EC) を多目的へ応用した進化的多目的最適化 (Evolutionary Multi-Criterion Optimization: EMO) が近年、非常に盛んに行われ大きな進歩を見せている^{2, 6-9)}。多くの EMO アルゴリズムでは、上記の問題点を解決しており、各評価項目の優先度を明示的に定義することなく 1 度の探索で複数のパレート最適解を探索することが可能である。

この分野では、様々な進化的アルゴリズムが適用されているが、特に遺伝的アルゴリズム (Genetic Algorithm: GA) を多目的最適化問題に適用した多目的 GA は最も数多く研究されており、主要な研究の多くが多目的 GA を用いたものとなっている²⁾。

GA は自然界における生物の遺伝と進化をモデル化した最適化手法である^{4, 10)}。GA は多点探索であるため、多峰性のある問題においても最適解を探索でき、かつ離散的な問題にも対応できる非常に強力な最適化ツールの 1 つである。一方、基本となるアルゴリズム自体の仕組みはシンプルであるため、アルゴリズムの実装は比較的容易に行うことができる。

多目的 GA では、複数のパレート最適解を 1 度の探索によって求めることができる。しかし、多目的最適化問題では個体 (解候補) が複数の目的関数値を持っているため、単一目的最適化のように一意的に個体を評価することができない。この点について、これまで大きく 2 つのアプローチがとられてきた。1 つは、パレート最適解の概念に基づいて個体を

評価するパレートのアプローチであり、もう一方は、パレート最適解の概念を評価に利用しない非パレートのアプローチである。

また、その他にも探索過程で見つかった優れた個体の保存、個体群の多様性保持のためのメカニズム、各目的間のスケールの正規化など様々なメカニズムが提案され、実装されている。

以下では、まず GA について概説し、GA を用いた多目的最適化手法の分類、さらに代表的な手法のアルゴリズムについて解説する。

3.2 遺伝的アルゴリズム

遺伝的アルゴリズム (Genetic Algorithm: GA) は生物が環境に適合して進化していく過程を工学的に模倣した最適化アルゴリズムである。GA の研究は 1960 年代後半から 1970 年のはじめに Michigan 大学の Holland らによって始められ、その研究成果は 1975 年に “Adaptation in Natural and Artificial System¹⁰⁾” という題で出版されている。また、1989 年に出版された Goldberg の “Genetic Algorithms in Search, Optimization, and Machine Learning⁴⁾” 以降、日本でも盛んに研究が行われるようになり、現在では多方面に応用されている^{4, 11-17)}。

自然界における生物の進化過程においては、ある世代を形成している個体の集合、すなわち母集団の中で、環境に適合した個体がより高い確率で生き残り、次の世代に子を残す。このメカニズムをモデル化し、環境に対して最もよく適合した個体、すなわち目的関数に対して最適値を与えるような解を計算機上で求めようというのが GA の概念である。

GA において、個体 (Individual) は設計変数の値がコーディングされた染色体 (Chromosome) と呼ばれる文字列上で表現され、この染色体をデコーディングすることにより設計変数を読み出し、目的関数の値を計算する。このとき、染色体の構造のことを遺伝子型 (Geno Type)、これによって定まる個体の形質を表現型 (Pheno Type) と呼ぶ。また、個体の集団のことを母集団 (Population) と呼ぶ。GA はこの母集団に対して選択 (Selection)、交叉 (Crossover)、突然変異 (Mutation) などの遺伝的操作を繰り返し行うことによって解探索を行う。一般に、一連の遺伝的操作の繰り返しを世代と呼び、探索が終了するまでに必要となった世代を終了世代数と呼ぶ。図 3.1 に GA の概念図を示す。

ここで GA の流れを図 3.2 に示し、それぞれの操作について簡単に説明する。

- **Initialization** : 母集団の初期化

この操作は、あらかじめ設定された数だけランダムに個体を生成するものである。生成した個体の数のことを母集団サイズ (Population Size) や単に個体数と呼び、ここで生成した個体の集団を初期母集団とする。

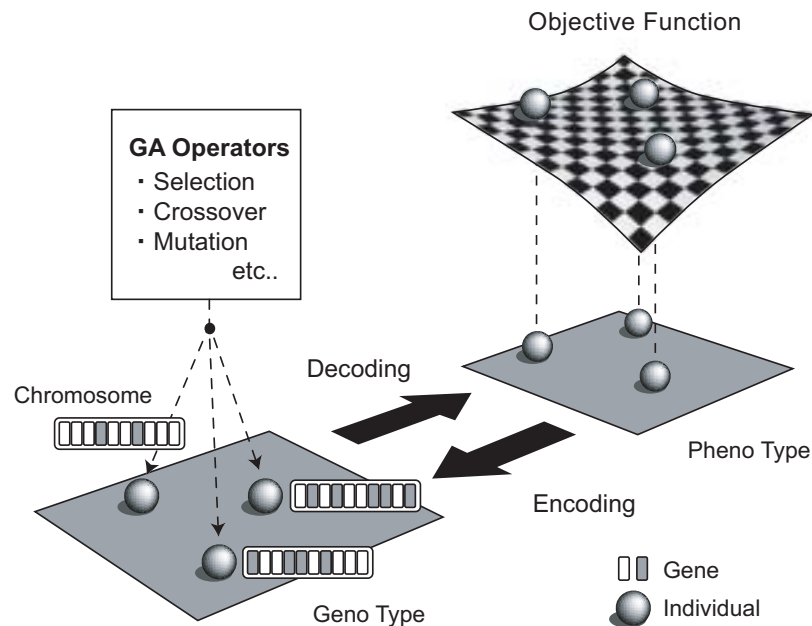


図 3.1 Schematic of GA

- **Evaluation** : 評価

この操作は、各個体の持つ染色体を問題空間にデコードして個体の評価値 (Evaluation Value) を求めるものである。一般に、この評価値を基に、個体の適合度値を決定する。適合度値は、個体とその環境にどの程度適合しているかを表す値であり、次世代への生き残りやすさを定量的に示している。そのため、適合度は選択操作の時に用いられる。適合度値が高いほど個体はその環境に適合していると見なす。

- **Selection** : 選択

この操作は、生物の適者生存を模倣したものである。この操作では、まず各個体の適合度値から次世代への生き残りやすさを求め、これに基づいて次世代の母集団を形成する。

- **Crossover** : 交叉

この操作は、生物の有性生殖を模倣したものである。この操作により、個体間で染色体情報が交換される。最適解を表す個体の一部分を持った個体どうしが交叉すればより最適解に近い個体が得られる可能性が高くなる。個体集団のうち何割の個体が交叉するかを交叉率 (Crossover Rate) と呼ばれるパラメータによって定める。

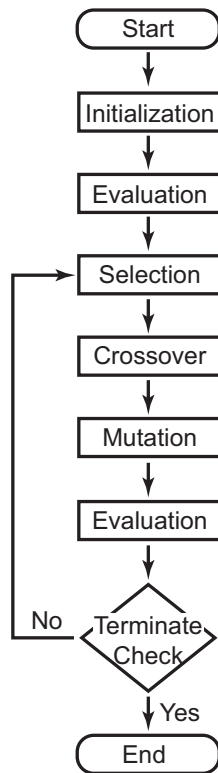


図 3.2 Flowchart of GA procedure

- **Mutation** : 突然変異

染色体は遺伝子 (Gene) を格納する複数の遺伝子座 (locus) から構成され、遺伝子座に入りうる遺伝子のことを対立遺伝子という。突然変異とは、染色体上の遺伝子座の遺伝子を別の対立遺伝子に置き換える操作のことであり、自然界における DNA 複写の際に起こるコピーミスにあたる。各遺伝子座に対して、何割の確率で突然変異が起きるのかを突然変異率 (Mutation Rate) と呼ばれるパラメータによって定める。

- **Terminate Check** : 終了判定

この操作は、あらかじめ定められた終了条件に基づいて GA を終了させるためのものである。

Goldberg によれば、GA は従来の最適化手法と比較して次の 4 つの特徴を持っている⁴⁾。

- 設計変数を直接操作せずにコード化した状態で扱う。
- 一点探索ではなく、多点探索である。
- サンプリングによる探索で、ブラインドサーチである。

- 決定論的規則ではなく、確率的オペレータを用いる探索である。

一方で、GA に関する研究が進むに従って以下のような問題点が指摘されるようになった。

- **高い計算負荷**

GA では、評価のために母集団サイズの分だけ目的関数を計算しなければならない。評価は毎世代行われるため、評価計算回数の合計は（母集団サイズ）×（終了までに要した世代数）となる。また母集団の中には同じ染色体を持つ個体が複数存在することもある。このため不要な評価計算が多くなり、1 点探索の最適化手法と比較して計算にかかる負荷が大きくなる。

- **早熟収束による局所解への収束**

他の個体に比べて非常に高い適合度値を持つ個体が存在した場合、その個体の遺伝子は急速に母集団内に広がる。一般に、このような減少を早熟収束という。早熟収束が起こると、母集団の多様性が減少し、局所解に収束する可能性が高くなることが知られている^{12,13,17)}。図 3.3 は GA における解探索の様子を示した図である。この図において最適解に最も近い個体は No.6 であり、もっとも適合度の高い個体は No.1 である。この図では、最適値に最も近い個体よりも適合度は高いが局所解に近い個体が存在するため、母集団全体が局所解に収束する可能性が高いと言える。

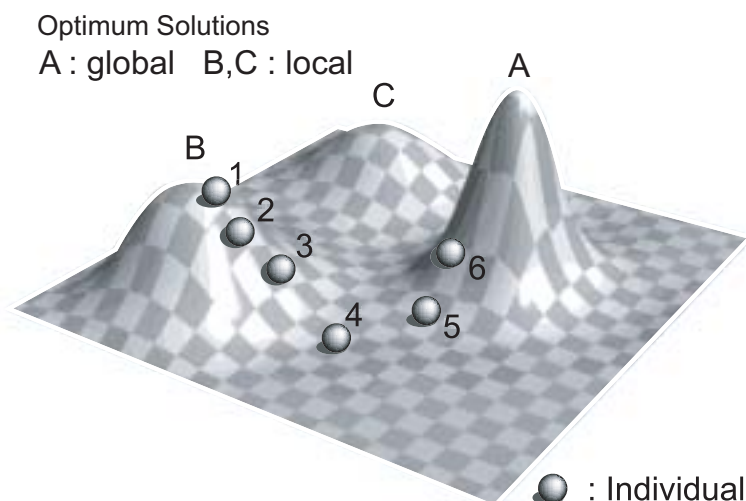


図 3.3 Schematic of GA search

- **パラメータ設定の複雑さ**

GA では、個体数や交叉率、突然変異率など設定すべきパラメータが多く、またこれ

らのパラメータは解探索能力に大きく影響する．さらに対象とする問題によって最適なパラメータの値が異なる．このため，最適なパラメータの値を知るためには多くの予備実験を行う必要がある．

3.3 遺伝的アルゴリズムによるパレート最適解生成法

Schaffer らの VEGA³⁾ によって始まった進化的多目的最適化に関する研究は，近年ますます盛んに行われるようになり大きな進歩を見せている．特に最近では，進化的多目的最適化に関する初めての国際会議 EMO'01(Conference on Evolutionary Multi-Criterion Optimization)¹⁸⁾ が開催されるなどこれまでにない盛り上がりを見せている．この分野では，様々な進化的なアルゴリズムが適用されているが，特に遺伝的アルゴリズム (Genetic Algorithm: GA) を多目的最適化問題に適用した多目的 GA は，最も主要な研究となっている¹⁸⁾．

本節では，これまでに提案された多目的 GA の大まかな分類を行った上で，代表的なアルゴリズムの解説，多目的 GA における問題点について述べる．

3.3.1 多目的遺伝的アルゴリズムの分類

多目的 GA では，設計領域内に遺伝子を生成し，交叉により新たな遺伝子を発生させ何らかの方法で選択することにより，パレート最適解集合を探索する．GA の探索過程における，その時点での最も良好な解，すなわち母集団全体の中で他のどの個体と比較しても優越されていない個体を，非劣個体または非劣解と呼び，非劣解集合をパレート最適解集合へ近づけることが多目的 GA の目的となる¹⁾．

一般に，GA の各世代における非劣解により形成される面を解の近似パレート最適フロントと呼ぶ²⁾．概念としては，世代が進むに従い個体の作り出す近似パレート最適フロントはパレート最適フロントに近づいていくものとして捉えることができる．

GA を多目的最適化問題に対して適用する場合，この非劣解集合を適切に評価し，次世代に残していくことがポイントとなる．従来の「1つの最適解」を求める単一目的の場合と異なり，多目的では他の解に劣っていない解 (パレート最適解) 全てが解候補となるため，単純に単一目的における適合度の割当て方法³⁾ をそのまま適応させることはできない．すなわち，複数の評価値を基に単一の適合度値を求める必要がある．その点に関して，従来

- 解の優越関係を用いない選択演算を行う (非パレートのアプローチ)

¹⁾非劣解とは，どの解にも支配されていない，劣っていない解という意味である．

²⁾パレート最適解集合が形成する曲面のことをパレート最適フロントと呼び，それと区別するため探索により得られた非劣解集合が形成する曲面を近似パレート最適フロントという．

³⁾一般に，多くの単一目的 GA では評価値をそのまま適合度値として用いている．

- 解の優越関係に基づいて選択演算を行う (パレートのアプローチ)

という2つ考え方に基づいて、種々の方法が提案されている²⁾。非パレートのアプローチは、1985年に初めて多目的にGAを適用したアルゴリズムであり、SchafferのVEGA(Vector Evaluated Genetic Algorithm)³⁾に始まり1990年代前半までに提案されたアルゴリズムに多く見られるアプローチ方法である²⁾。一方、パレートのアプローチは、2章において説明したパレート最適解の概念を用いた手法で1989年にGoldberg⁴⁾により提案された非優越ソートに始まり、1993年にFonsecaにより提案されたMOGA(Multi-Objective Genetic Algorithm)¹⁹⁾などが代表的である。近年提案されたアルゴリズムの多くはこのアプローチ方法に分類される²⁾。

表3.1には、代表的な多目的GAの手法とその特徴を年代順にまとめたものを示す。また、次節以降において、表3.1の各手法の概説を行う。

表 3.1 A brief history of EMO

Year	Name	Proposer(s)	Characteristic
1985	VEGA	Schaffer	The first multi-objective GA
1993	WBGA	Hajela and Lin	Weighted function
1993	MOGA	Fonseca and Fleming	Pareto-based selection
1993	NPGA	Horn and Nafpiliotis	Niching
1994	NSGA	Srinivas and Deb	Non-dominated Sorting
1999	SPEA	Zitzler and Thiele	Archiving + elitism
2000	NSGA-II	Deb	Crowding distance
2001	SPEA2	Zitzler and etc	Archive truncation + improved fitness assignment scheme

3.3.2 VEGA

Schafferは、1985年に初めて多目的へGAを適用したアルゴリズム、ベクトル評価遺伝的アルゴリズム(Vector Evaluated Genetic Algorithm: VEGA)を提案した³⁾。VEGAという名は、(スカラー目的関数の代わりに)各目的ベクトルを評価する手法であることに由来している。

VEGAは、非常にシンプルな手法であり、単一目的GAから多目的への単純な拡張アルゴリズムである。VEGAの概念図を図3.4に示す。図3.4からも分かるように、VEGAでは母集団を目的関数の数に等しいサブ母集団に分割し、サブ母集団ごと独立に個体を選択して新たなサブ母集団を生成する。そして、生成されたサブ母集団をすべて合わせて一つ

の母集団としたものに対して交叉，突然変異を行う。

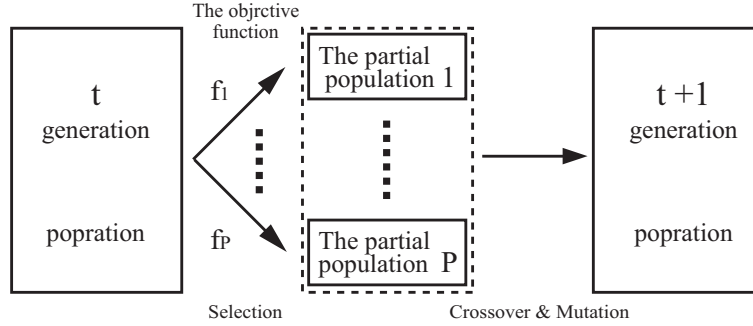


図 3.4 Schematic of VEGA

アルゴリズムの流れ

目的関数の数 M ，総個体数 N の場合の VEGA アルゴリズムの流れを示す。

Step 1 目的関数カウントを $i = 1$ とする．サブ母集団あたりの個体数を $q = N/M$ とする．

Step 2 サブ母集団内の各個体 $j = 1 + (i - 1) \times q$ から $j = i \times q$ に対して，以下の式に従って適合度割当てを行う．

$$F(x^j) = f_i(x^j) \quad (3.1)$$

Step 3 サブ母集団内の全ての q に対して選択を行い，母集団 P_i を生成する．

Step 4 もし， $i = M$ ならば Step 5 へ．そうでなければ， $i = i + 1$ として Step 2 へ．

Step 5 全てのサブ母集団の P_i を統合し P を生成する ($P = \cup_{i=1}^M P_i$). P に対して交叉，突然変異を実行し新たな母集団を得る．

利点

VEGA の利点は，アルゴリズムが非常にシンプルであり実装しやすいことである．単一目的 GA にわずかな変更を加えることにより VEGA を作成することができ，評価する目的も単一のままである．これは，他のアルゴリズムと比べて明確に優れた利点である．

欠点

VEGAにおいて各個体は、1つの目的関数においてのみ評価される。これは、VEGAの探索において、個体の評価はある一つの目的関数値のみから決定されることを意味する。そのため、個体は各目的の最適解付近、すなわちパレート最適フロントの端付近に集中する傾向が強く、パレート最適フロントの中間付近の解を得にくいという欠点がある(図 3.5)。

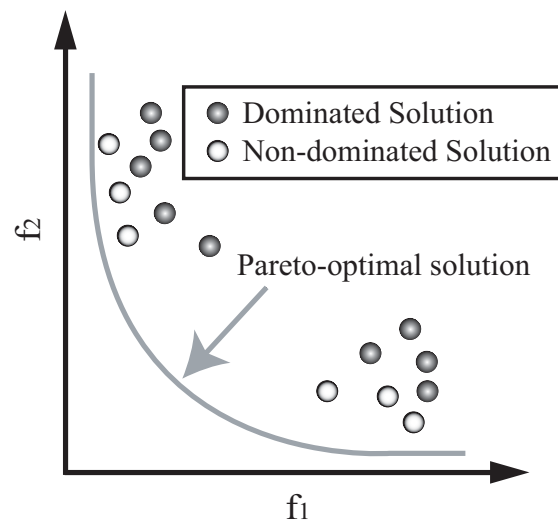


図 3.5 Distribution of Pareto Solutions in VEGA

3.3.3 WBGA

Hajela と Lin は 1993 年に WBGA(Weight-Based Genetic Algorithm) を提案した²⁾。この手法は従来までの重み係数法(2.3.1 章 参照)などの手法と異なり、GA 母集団の各個体は異なった重みベクトルを割当てられる。そのため、特定の重みに相当する1つのパレート最適解を探索するのではなく、GA 母集団で同時に複数の異なる重みベクトルを維持している。それゆえに、1度の探索によって複数のパレート最適解を見つけることができる。そのため、WBGAでは母集団中における重みベクトルの多様性が非常に重要となる。この点について2つの方法が提案されている。

- a) シェアリング関数を用いる方法
- b) ベクトル評価手法(前もって定義してある異なった重みベクトルを用いて評価する方法)

以下、それぞれの場合について説明する。

a) シェアリング関数を用いる方法

シェアリング関数を用いる WBGA では、個体は各目的を重みづけするための重みベクトルを持っている。ここでの各個体を持つ重みベクトルとは、 $F(x) = \sum_{i=1}^k w_i f_i(x)$ における w_i のことである。この方法では、個体の持つ各目的の重みベクトルに対してシェアリングを行い、個体の持つ重みベクトルの値が母集団全体として一様となることを目的とする。

この手法における個体 $\mathbf{x}^1$ の評価値は、以下の式により求まる。

$$F(\mathbf{x}^1) = \sum_{j=1}^M w_j^{x_w^1} \frac{f_j(\mathbf{x}^1) - f_j^{\min}}{f_j^{\max} - f_j^{\min}} \quad (3.2)$$

式 (3.2) のように、目的関数値のスケーリング値に重みを掛けた値の総和が評価値となっている。なお、式 (3.2) における x_w^i は、重みベクトルの割合を示す変数である。例えば、2 目的最適化問題において重みベクトルの割合を 9 種類に分割するという場合には表 3.2 のようになる。

表 3.2 A example of weight vector Table

x_w	1	2	3	...	9
Weight vector	(0.1,0.9)	(0.2,0.8)	(0.3,0.7)	...	(0.9,0.1)

この手法では、重みベクトルを表す変数を基にシェアリング関数 $Sh(d)$ を用いて評価値の変換を行う。そのため、重みベクトルを表す変数の距離を何らかの方法により計算する必要がある。2つの個体 i と j 間の重みベクトル変数間の距離 $d_{i,j}$ を次式に従い求める。

$$d_{i,j} = |x_w^i - x_w^j| \quad (3.3)$$

本手法では、上記の距離 $d_{i,j}$ を基に、シェアリング $Sh(d_{i,j})$ を行いニッチカウント (niche count) nc_i を求める。なお、 nc_i は $Sh(d_{i,j})$ の総和として求める。このニッチカウントを用いて $F'_i = F(\mathbf{x}^i)/nc_i$ の変換を行う。シェアリング $Sh(d_{i,j})$ とは、個体 i に対してある一定の範囲内 (σ_{share}) に別の個体 j が存在していた時にその距離に応じて 0 以上 1 以下の正の値を返す関数であり下記の式 (3.4) で表される。

$$Sh(d_{i,j}) = \begin{cases} 1 - \frac{d_{i,j}}{\sigma_{share}}, & \text{if } d_{i,j} \leq \sigma_{share}; \\ 0, & \text{otherwise.} \end{cases} \quad (3.4)$$

式 (3.4) から分かるように、シェアリング $Sh(d_{i,j})$ は個体間の距離 ($d_{i,j}$) が近ければ近いほど 1 に近い値を返し、 σ_{share} に近いほど 0 に近い値を返す。

また, $Sh(d_{i,j})$ から個体 i の nc_i を求める式を式 (3.5) に示す.

$$nc_i = \sum_{k=1}^N Sh(d_{i,k}) \quad (3.5)$$

適合度割当て

シェアリング関数を用いた場合における適合度割当ての流れを以下に示す.

Step 1 各目的関数 f_j の, 最大値 $f_j^{\max}$, 最小値 $f_j^{\min}$ を求める.

Step 2 各個体 $i = 1, 2, \dots, N$ に対して全ての個体との距離 $d_{i,k} = |x_w^i - x_w^k|$ を求める. そして, 式 (3.4) を用いてシェアリング関数 $Sh(d_{i,k})$ を計算する.

その後, 個体 i のニッチカウント nc_i を式 (3.5) により求める.

Step 3 各個体 $i = 1, 2, \dots, N$ に対して $F'_i = F(\mathbf{x}^i)/nc_i$ を行う.

利点

WBGA は単一目的 GA として使用するため, 単一目的 GA から WBGA への変換へ多大な労力を必要としない. また, 変数 $\mathbf{x}_w$ が増加するものの, アルゴリズム自体は他の多目的 GA と比較してシンプルである.

欠点

WBGA は, シェアリング関数を用いて評価値を減少させるという方法を用いているため, 原理的には最大化問題にしか対応することができない. そのため最小化問題を扱うためには, 評価の部分において変換を行う必要がある. 特に, 最大化と最小化の混在する問題に対して適用が困難となる.

また, 一般に重みベクトルに基づくアプローチは, 非凸のパレート最適フロントを持つ問題においてパレート最適解集合を探索することが困難である. そのため, WBGA はこのような問題において解探索が困難である²⁾.

一方, 一様に分散した重みベクトルの集合が必ずしも一様に分散したパレート最適解集合を表しているとは限らないため, 一様に広がった非劣解集合を得ることができない可能性がある.

b) ベクトル評価手法

ベクトル評価手法は、VEGA に類似した手法である。本手法では、まず K 個の異なる重みベクトル $w^{(k)} (k = 1, 2, \dots, K)$ の集合が選択される。そして、全ての個体 N に対して式 (3.2) を適用し各重みベクトル $w^{(k)}$ に対する評価値を計算する。母集団 N の中で、各重み $w^{(k)}$ に対する最良の N/K 個体を選択し、それらを重み $w^{(k)}$ のサブ母集団としてグループ化する。その上で、各グループ内において選択、交叉、突然変異などの遺伝的操作を行い探索を進める。

上記の手順から全部で K のサブ母集団が存在することが分かる。また、1つの個体は1つ以上の(重みベクトル)グループに属することが許されている。そのため、中間的な解は一つ以上のサブ母集団に含まれやすくなる。

このアルゴリズムでは、複数の非劣解が明確に定義された K 個の重みベクトル $w^{(k)} (k = 1, 2, \dots, K)$ によって探索することができる。それゆえ、先ほどのシェアリングを用いたアプローチと異なり追加的なニッチング操作が必要ない。1つのサブ母集団は、対応する重みベクトルによって処理され、遺伝的操作はその重みベクトルのベクトル軸上の最良の解を探索する。

アルゴリズムの流れ

ベクトル評価手法のアルゴリズムの流れを示す。ただし、 K 個の重みベクトルの集合が既知であるとの前提で行う。また、 M は目的関数の数を表している。

Step 1 重みベクトルカウンタ $k=1$ にセットする。

Step 2 重みベクトル $w^{(k)}$ を持つ各個体 $x^{(i)}$ の適合度 F_j を以下の式を用いて見つける。

$$F(x^{(i)}) = \sum_{j=1}^M w_j^{x^{(i)}} \frac{f_j(x^{(i)}) - f_j^{\min}}{f_j^{\max} - f_j^{\min}} \quad (3.6)$$

適合度の値 F に関連して最良の個体 N/K を選択。これらの個体をサブ母集団 P_k にコピーする。

Step 3 P_k において選択、交叉、突然変異を実行。 N/K 個の新たな母集団を生成する。

Step 4 もし $k < K$ ならば、 k に 1 増加 ($k = k + 1$) して Step2 へ。そうでなければ全てのサブ母集団を統合して、新たな母集団 $P = \cup_{k=1}^K P_k$ を生成する。もし $|P| < N$ ならば、ランダムに個体を生成し母集団のサイズを N とする。

Step 2では、 N 個の個体の評価が K 回実行されている。ゆえに、全体としては $K \times N$ 回の評価が必要となる。もし、 K が N に比べて十分に小さければ、このアルゴリズムの各世代における複雑度は、 $O(N)$ となる。

利点

この手法では、シェアリング関数を用いていないため 2 個体間の距離の測定などを行う必要がなく、より複雑な問題においてはシェアリング関数を用いた方法よりも良好な結果を得ることができる。また、重みベクトル変数といった追加的な変数を個体情報に加える必要もない。

欠点

全ての重みに基づく方法と同様に、この手法においても重みベクトルの設定が重要となる。これは、前もって定義してある重みベクトル軸に得られる解集合の分布が強く依存するためである。また、GA 操作が各サブ母集団に独立に行われるため、各重みベクトルに相当するパレート最適解を探索するためには、適切なサブ母集団サイズが必要となる。

3.3.4 MOGA

1993 年に Fonseca により提案された MOGA (Multi-Objective Genetic Algorithm)¹⁹⁾ は、パレートの概念を探索に用いたアルゴリズムである。MOGA では、個体の評価としてパレートランキング法を用いた評価方法を行っている。

パレートランキング法では、個体 X_i が n_i 個の個体に優越されているとき、 X_i のランク $r(X_i)$ を

$$r(X_i) = 1 + n_i \quad (3.7)$$

のように定めることにしている。この手続きによるランキング例を図 3.6 に示す。なお、図 3.6 は最小化問題におけるランキングの例である。

このランキング法を用いた選択手法としては、ランクの値を適合度に変換し用いるルーレット選択、各世代で非劣個体 (ランク 1 の個体) のみ残すパレート最適個体保存選択などがある。

また、非劣解集合の多様性保持のための手法として、Fonseca と Fleming は各ランク間においてニッチングを提案している。このニッチングは、3.3.3 節において用いたシェアリング関数 (式 3.4) を用いる方法である。このシェアリング関数では、設計変数空間での距離ではなく、目的関数空間での距離を用いる。あるランクにおける個体 i と j 間の距離は、次式により求まる。

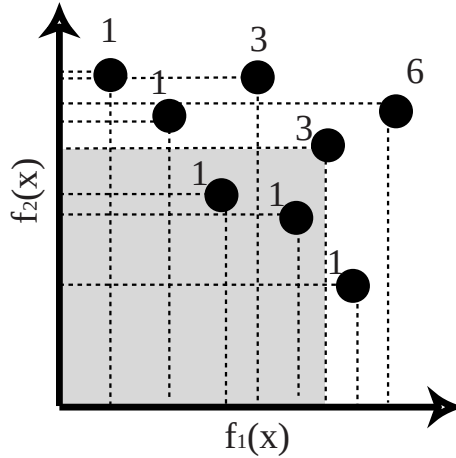


図 3.6 The concept of Pareto-ranking

$$d_{ij} = \sqrt{\sum_{k=1}^M \left(\frac{f_k^i - f_k^j}{f_k^{\max} - f_k^{\min}} \right)^2} \quad (3.8)$$

式 (3.8) の $f_k^{\max}, f_k^{\min}$ は, k 番目の目的関数値の最大値と最小値である. 式 (3.4) を用いて $Sh(d_{ij})$ の値を求める. その後, ニッチングカウントを下記の式によって求める.

$$nc_i = \sum_{j=1} \mu_{r_i} Sh(d_{i,j}) \quad (3.9)$$

式 (3.9) における μ_{r_i} は, ランク r_i における個体の数を表している.

適合度割当て

MOGA における適合度割当ての流れを以下に示す.

Step 1 各変数の初期化を行う ($\mu(j) = 0 (j = 1, \dots, N) : i = 1$). 変数 $\mu(j)$ はランク j に属する個体数を保持する変数である.

Step 2 個体 i を支配している個体の数 n_i を数え, 個体 i のランク r_i を $r_i = 1 + n_i$ として計算する. また, ランク r_i が属する $\mu(r_i)$ を更新する ($\mu(r_i) = \mu(r_i) + 1$).

Step 3 $i < N$ ならば, $i = i + 1$ を行い Step 2 へ戻る. そうでなければ Step 4 へ進む.

Step 4 $\mu(r_i) > 0$ を満たす r_i からランクの最大値を求め r^* とする. ランク値を基準に個体のソートを行い, 全ての個体に対する平均適合度割当てを式 (3.10) に従い求める

$$F_i = N - \sum_{k=1}^{r_i-1} \mu(k) - 0.5(\mu(r_i) - 1) \quad (3.10)$$

ランク $r_i = 1$ を持つ個体 i に対して、式 (3.10) は $F_i = N - 0.5(\mu(1) - 1)$ の適合度を割当てて、この F_i 値は、 $\mu(1)$ の平均値であり、 N から $N - \mu(1) + 1$ までの連続する整数である。ランクカウンタを $r_c = 1$ に定める。

Step 5 ランク r_c の各個体 i に対して、同じランクを持つ自分以外の個体からニッチカウントを計算する。この計算は、式 (3.9) を用いる。ニッチカウントを用いて $F'_j = F_j / nc_j$ の変換を行う。同じ平均適合度を維持するため、次式のように割当て適合度をスケール化する。

$$F'_j \rightarrow \frac{F_j \mu(r_c)}{\sum_{k=1}^{\mu(r_c)} F'_k} F'_j \quad (3.11)$$

Step 6 もし $r_c < r^*$ ならば、 $r_c = r_c + 1$ を行い Step 5 へ。そうでなければ、この処理は終了する。

このように、ランク r_i の値が 1 に近い (低い) 個体ほど高い適合度が割当てられ、ランク r_i の値が大きい (高い) 個体ほど低い適合度が割当てられる。

利点

MOGA は、適合度割当てスキームがシンプルである。ニッチングが目的関数空間で行われるため、MOGA では連続問題だけでなく離散的な組み合わせ問題などに対しても容易に適用することができる。また、目的関数空間でのニッチングを行う MOGA は、目的関数空間における非劣解の広がりを求める場合に適した手法であるといえる。

欠点

個体間の支配関係が適合度割当てに用いられているものの、(ランク 1 の非劣個体を除いて) 特定のランクに属する個体全てに同じ適合度を割り振る必要はない。これは、ある探索領域において幾つかの個体方向へ望まない偏りが生じる危険性を持っている。特に、このアルゴリズムではパレート最適フロントの形、探索空間の個体密度に探索が影響されやすい²⁾。

また、MOGA における適合度の割当て計算では、よりランクの悪い解が、よりランクの良好な解に対して常により悪い適合度を割当てるとは限らない。そのため、より良好なラ

ランクの個体が混み合っている場合には、これらの個体に対するニッチカウントが大きくなり、ランクの低い個体の適合度がランクの高い個体の適合度よりも高くなる可能性がある。もし、このような逆転現象が生じた場合、より良好なランクを持つ全ての個体に対して適切な選択圧がかからなくなり、結果として収束の遅延、それ以上の探索が不可能となる。

3.3.5 NSGA

非優越ソート GA (Non-dominated Sorting Genetic Algorithm: NSGA) は、1994 年に Deb, Srinivas らによって提案されたアルゴリズムである²⁰⁾。その最大の特徴は、1989 年に Goldberg⁴⁾ により提案された個体のランク付け方法の 1 つである非優越ソート (non-dominated sorting) の概念に基づいて設計されている点である。また、MOGA に比べ NSGA は非劣個体をより重要視する適合度割当てを行っており、個体の多様性の維持のために設計変数空間での距離に基づくシェアリングを用いている。NSGA における設計変数空間に基づく距離を求める式を式 (3.12) に示す。

$$d_{i,j} = \sqrt{\sum_{k=1}^{P_1} \left(\frac{x_k^i - x_k^j}{x_k^{\max} - x_k^{\min}} \right)^2} \quad (3.12)$$

適合度割当て

まず、NSGA における適合度割当てについて説明する。上述のように NSGA では非優越ソートと呼ばれる個体のランク付け方法を用いている。非優越ソートに基づくランキングの手続きを以下に示す。

Step 1 ランク $r = 1$ とする。

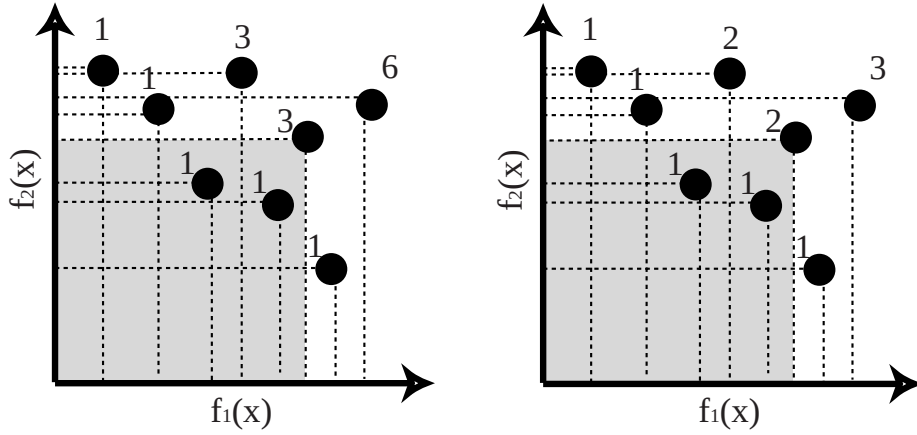
Step 2 個体群 (P) の中から非劣個体を求め、これらの個体をランク r とする。

Step 3 得られた非劣個体群を個体群 P から除き、 $r = r + 1$ とする。

Step 4 全ての個体がランク付けされるまで (個体群 P が空になるまで)、Step 2 および Step 3 を繰り返す。

同じ分布において MOGA におけるパレートランキングを適用した場合と非優越ソートを適用した場合の個体のランク付け例を図 3.7 に示す。

NSGA の非優越ソートに基づく適合度の割当てアルゴリズムを以下示す。



(a) Pareto Ranking method(MOGA) (b) Non-dominated Sorting method(NSGA)

図 3.7 The concept of Non-dominated Sorting method

Step 1 シェアリングパラメータ σ_{share} と正の小さな変数 ϵ を決定し, $F_{\min} = N + \epsilon$, $j = 1$ とする.

Step 2 非優越の定義に従って母集団 P を非優越ソートのランクに従い分類する:

$$(P_1, P_2, \dots, P_\rho) = \text{Sort}(P, \preceq)$$

Step 3 各サブ母集団 P_j に含まれる個体 q に対して

Step 3a $F_j^{(q)} = F_{\min} - \epsilon$ を用いて適合度割当てを行う.

Step 3b 式 (3.4) を用いて P_j の個体間におけるニッチカウント nc_p を計算する.

Step 3c $\hat{F}_j^q = \frac{F_j^q}{nc_q}$ を用いて適合度の再計算を行う.

Step 4 サブ母集団 P_j に含まれる個体の最低の適合度値を求める $F_{\min} = \min(\hat{F}_j^q : q \in P_j)$.

Step 5 $j = j + 1$ を行い, もし母集団 P_j のランクが最大ランク P_ρ 以下 ($j \leq \rho$) ならば Step 3 へ. そうでなければ適合度割当て終了.

利点

NSGA の利点は, 非優越集合に従った適合度割当てを行っている点である. よりランクの高い個体が優位な適合度値が割当てられるようになっている. そのため, NSGA の探索は, パレート最適フロント方向へと必ず進んでいく. さらに, 表現空間での距離に基づく

シェアリングを行っているためより多様な非劣解集合が得られる。また、シェアリング距離は目的関数空間に基づく方法を用いることも可能である。

欠点

シェアリング関数を用いた手法では、シェアリングパラメータ σ_{share} を固定する必要がある。NSGA の性能がこの σ_{share} 値によって大きく影響を受けることが、幾つかの研究によって指摘されている²⁾。

3.3.6 NPGA

NPGA(Niched Pareto Genetic Algorithm) は、1994 年に Horn, Nafpliotis によって提案された手法であり、MOGA, NSGA と同様、非優越の概念に基づいたアルゴリズムである²¹⁾。NPGA の最大の特徴は、パレート最適解の概念を用いたバイナリトーナメント選択という独自の手法を用いている点である。バイナリトーナメント選択は、単一目的最適化において、他の選択手法に比べ理論的に優れていることが確認されている¹¹⁾。NPGA では、バイナリトーナメント選択にパレート最適解の概念とシェアリングの概念を取り入れ多目的への適用を試みている。

NPGA のバイナリトーナメント選択

NPGA のバイナリトーナメント選択について説明する。まず、個体数 N の親母集団 P から 2 つの個体 i と j をランダムに選択する。次に、優越関係のテスト用に母集団から $t_{dom} (\ll N)$ 個の個体を比較集合として選択する。そして、2 つの個体 i と j と比較集合の全ての個体の優越比較を行う。

その結果、もし一方が非優越であり他方が少なくとも 1 つの比較集合に優越されているという場合、前者が選択される。一方、 i と j の両方の個体が非優越である、もしくは比較集合に優越されているという場合には、現在の子個体母集団に対する i と j のニッチカウントを計算し、より小さいニッチカウントを持つ個体を選択する。

選択に選ばれた 2 つの個体を i と j 、現在の子母集団を Q 、とした場合の NPGA におけるバイナリトーナメント選択の手順を以下に示す。

$$\text{winner} = \text{NPGA-tournament}(i, j, Q)$$

Step T1 母集団 P から t_{dom} 個の比較個体 T_{ij} を選択する。

Step T2 i を優越している T_{ij} の数 α_i を計算する。また、同様に j を優越している T_{ij} の数 α_j についても計算する。

Step T3 もし、 $\alpha_i = 0$ かつ $\alpha_j > 0$ ならば、 i が選択され選択操作は終了となる。

Step T4 逆に $\alpha_i > 0$ かつ $\alpha_j = 0$ ならば、 j が選択され選択操作は終了となる。

Step T5 上記の条件に当てはまらず、もし $|Q| < 2$ であるならば、 i と j はお互いに5割の確率でどちらかが選択され選択操作は終了する。また、 $|Q| \geq 2$ ならば、現在の小母集団 Q における i と j のニッチカウントがお互い独立に計算される。ニッチカウントは、 i からの距離がニッチパラメータ σ_{share} 内にある子個体($k \in Q$)の数となる。目的関数空間における個体 i と j のユークリッド距離 d_{ik} は、目的関数空間において下記のように求められる。

$$d_{ik} = \sqrt{\sum_{m=1}^M \left(\frac{f_m^i - f_m^k}{f_m^{\max} - f_m^{\min}} \right)^2} \quad (3.13)$$

式(3.13)における $f_m^{\max}$ と $f_m^{\min}$ は m 番目の目的関数空間における探索領域の上限と下限を示している。

Step T6 もし、 $nc_i \leq nc_j$ ならば i が選択され、そうでなければ、 j が選択される。

このようにNPGAでは、まず優越に基づく比較によって選択を行い、それが行えない場合には、個体の混雑度に基づく選択を行っている。つまり、前者はパレート最適フロントの方へ探索を進めることを、後者は解の多様性維持を目的としている。この2つのポイントは、多目的GAにおける重要かつ相反する目的であり、NPGAは選択においてこの2つのポイントを考慮しているのが分かる。

アルゴリズムの流れ

次に、NPGAでは上記のバイナリトーナメント選択がどのように探索サイクルの中で用いられているのか考察する。NPGAの探索サイクルを以下に示す。

Step 1 母集団 P をランダムに並び替え、 $i = 1, Q = \emptyset$ とする。

Step 2 上述のトーナメント選択を行う。第1の親を選択する $p_1 = \text{NPGA-tournament}(i, i + 1, Q)$ 。

Step 3 $i = i + 2$ を行い第2の親を選択する $p_2 = \text{NPGA-tournament}(i, i + 1, Q)$ 。

Step 4 p_1 と p_2 の交叉を行い子個体 c_1 と c_2 を生成する。子個体 c_1 と c_2 に対して突然変異を行う。

Step 5 子母集団 Q の更新を行う $Q = Q \cup \{c_1, c_2\}$ 。

Step 6 $i = i + 1$ を行う. もし $i < N$ ならば Step 2 へ戻る. そうでなく $|Q| = N/2$ ならば, P をランダムに並び替え $i = 1$ として Step 2 へ戻る. そうでなければ探索は終了する.

利点

上記の選択方法からも分かるように, NPGA では明示的な適合度割当てを行っていない. 一般に, VEGA, NSGA や MOGA などの多目的 GA における適合度割当てには, 何らかの独自の主観, 解釈が入り込んでしまう危険性がある. しかし, 適合度割当てを行っていない NPGA ではそのような危険性は全くない.

また NPGA の複雑性は, もしサブ母集団のサイズ t_{dom} が N に比べかなり小さな値であった場合, NPGA の複雑性は目的関数の数 M に依存しなくなる. これは, NPGA の比較における複雑性は $O(Mt_{\text{dom}})$ となり, 混雑度に基づく計算では $O(N^2)$ となるためである. そのため, 目的の多い問題を解く場合, NPGA の処理時間は他のアルゴリズムよりも短い可能性がある.

欠点

NPGA では, σ_{share} と t_{dom} の重要な 2 つのパラメータを設定する必要がある. 特に, NPGA では, 他の多目的 GA である NSGA, MOGA などのシェアリングを用いる手法よりも σ_{share} パラメータの解探索へ与える影響は大きい. これは, NPGA におけるニッチカウントが個体 i から距離 σ_{share} 内にある母集団個体の数によって決まるからである. NSGA, MOGA におけるシェアリングを表す式 (3.4) では, 個体 i との距離が $Sh(d)$ の評価値に大きく影響するため NPGA ほど直接的に解探索へ影響しない. そのため, NPGA ではより適切な σ_{share} の設定が重要となる. また, t_{dom} も優越に基づく比較結果に大きく影響するパラメータであるため, 解探索への影響は非常に大きく慎重に決定する必要がある.

3.3.7 SPEA

1999 年に Zitzler, Thiele によって提案された SPEA は, それまでに提案されてきた手法の幾つかのメカニズムに独自のメカニズムを組み込んだ手法である⁶⁾. SPEA において実装している既に提案されている手法から取り入れたメカニズムを以下に示す.

a) 探索過程において見つかった非優越解の保存 (エリート主義アプローチ)

これは, 単一目的 GA におけるエリート保存に相当する. 探索過程において発見した非劣解を保存する方法. 多くの手法では, 探索個体群とは別に非劣解を保存するための個体群を使用する^{2, 9, 21)}. 一般に, この個体群のことをアーカイブ個体群という.

b) パレート最適解の概念に基づく適合度割当て

c) 非劣解集合の削減方法

非劣解の保存を行う場合、必要以上の非劣解集合が得られた場合、それらを削減する必要がある。対象となる全ての個体が非劣解であるため、一般には多様性の観点から優越を決定する。

一方、オリジナルの手法として以下のメカニズムの提案を行っている。

- 独自の適合度割当て
- 多様性維持のための新たなニッチ手法
- 保存している非劣解集合の選択への参加

アルゴリズムの流れ

SPEA の全体的なアルゴリズムの流れを示す。

Step 1 初期探索母集団 P を生成、そして空のアーカイブ個体群 $\bar{P}$ を生成。

Step 2 P における非劣個体を $\bar{P}$ にコピーする。

Step 3 $\bar{P}$ において、ランク 1 以外の個体を削除。

Step 4 もしも $\bar{P}$ における個体数が、前もって与えられた最大個体数 $\bar{N}$ を越えていた場合、クラスタリングの手法によって $\bar{P}$ の個体を削減する。

Step 5 $\bar{P}$ における場合と同様に P における各個体の適合度を計算する。

Step 6 $P + \bar{P}$ の母集団より個体を選択する。この操作は、必要個体数に達するまで続けられ、新たな P を生成する。SPEA では、バイナリトーナメント選択を用いる。

Step 7 交叉や突然変異を P に対して適用する。

Step 8 もし、世代が終了世代に到達していたならば終了、そうでなければ Step2 へ。

上記から分かるように、SPEA ではアーカイブ個体群としてそれまでの探索において非優越である個体を保存し、毎世代ごとにアーカイブ個体群を更新しながら探索を進めていく。

SPEA では、適合度の割当てとクラスタリングに関してオリジナルの手法を提案、実装している。以下、適合度の割当てとクラスタリングについて説明する。

適合度の割当て

SPEA の適合度の割当て手順は、2つの段階により構成されている。

まず、アーカイブ個体群 $\bar{P}$ における個体のランク付けを行い、その後、母集団 (探索個体群) P のランク付けを行う。

Step 1 アーカイブ個体群の各個体 $i \in \bar{P}$ に対しては、適合度値として $s_i \in [0, 1)$ の実数値が割当てられる⁴。 s_i は、探索母集団 P の総個体数 N および個体 i が支配している探索母集団 $j \in P$ の数 n により決定する ($s_i = \frac{n}{N+1}$)。 個体 i の適合度値は、 s_i と同値にする ($f_i = s_i$)。

Step 2 探索母集団群の各個体 $j \in P$ の適合度値は、 j を優越している全てのアーカイブ個体 $i \in \bar{P}$ の s_i を足し合わせ求める：

$$f_j = 1 + \sum_{i, i \succ j} s_i \quad \text{where } f_j \in [1, N) \quad (3.14)$$

式 (3.14) において、左辺に 1 を足しているのは、 P のメンバーよりも $\bar{P}$ の適合度値がより良いものとするためである。

上記の手順から分かるように、SPEA では適合度が最小化されている。そのため、より小さな適合度値を持っている個体が高い確率で選択される。図 3.8 に、SPEA における適合度割当ての概念図を示す。

図 3.8 から分かるように、この手法ではパレート最適フロントへの近さと個体間の密度の双方が考慮された適合度の割当てが実現されている。特に、適合度シェアリングとの違いについて見た場合、密度を求めるのに距離の概念を全く用いていないことがあげられる。そのため、 σ_{share} のような追加的なパラメータを用いる必要はない。

クラスタリングを用いたパレート 集合の削減

SPEA では、アーカイブ個体群 $\bar{P}$ の個体数 $\bar{N}$ を越える非劣解が見つかった場合にクラスタリングという手法を用いて非劣解の個体数が $\bar{N}$ を越えないようにしている。

今、非劣解集合 $\bar{P}$ の個体数 $\bar{N}'$ を $\bar{N}$ へ削減する場合を考える ($\bar{N}' > \bar{N}$)。

まず、 $\bar{P}$ における各個体は、異なるクラスタに属するものとする⁵。従って、最初は $\bar{N}'$ 個のクラスタが存在する。それから、全てのクラスタどうしの *cluster-distance* を測定す

⁴SPEA では適合度の概念に Strength という単語を用いており、各個体に Strength を割当て、それを適合度として用いている。

⁵ここでのクラスタとは、まとまり、グループの意味である。

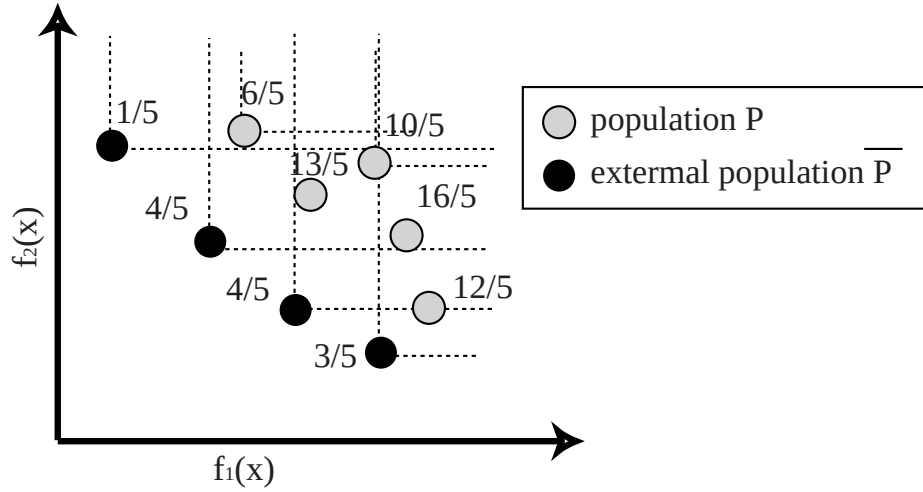


図 3.8 The SPEA fitness assignment scheme

る。一般に、クラスタ C_1 と C_2 の距離 d_{12} は、解の全てのペアのユークリッド距離によって求まる ($i \in C_1, j \in C_2$)

$$d = \frac{1}{|C_1| \cdot |C_2|} \cdot \sum_{i \in C_1, j \in C_2} d(i, j) \quad (3.15)$$

SPEA では、目的関数空間を基に $d(i, j)$ を計算している。式 (3.15) を用いた SPEA のクラスタリングの流れを以下に示す。

このクラスタ統合プロセスをクラスタの総数が $\overline{N}$ 個になるまで繰り返す。クラスタの総数が $\overline{N}$ 個になった時点で、各クラスタ内で他の個体に対して最小平均距離を持つ個体を選択し、残りの個体を全て削除する。

Step 1 クラスタ集合 C を生成： $\overline{N}$ 個のアーカイブ個体群の個体 $i \in \overline{P}$ は、それぞれ別のクラスタを構成している： $C = \{C_1, C_2, \dots, C_{\overline{N}}\}$

Step 2 もし $|C| \leq \overline{N}$ ならば Step 5 へ、そうでなければ Step 3 へ。

Step 3 全てのクラスタ間の距離を測定する (全てのペアは、 $\frac{|C|}{2}$ ある)。

Step 4 最小距離 d を持つクラスタ C_1 と C_2 を決定する；選択したクラスタどうしを統合し、クラスタの総数 C を 1 削減する。Step 2 へ。

Step 5 各クラスタ内で他の個体に対して最小平均距離を持つ個体を選択し、その他の個体を削減する。

利点

SPEA の最大の利点は、探索性能が優れていることである。SPEA では、探索で見つかった優れた非優越解を失うことなく、探索へ反映させている。すなわち、探索において発見した非劣解の扱いが非常に優れている。これは、SPEA におけるクラスタリング手法とアーカイブ個体群の探索への参加に起因している。SPEA のクラスタリングでは、得られた非劣解集合に対してより一様に広がっているものを保存するようにしている。クラスタリングの効果については、幾つかの研究によりクラスタリングが解探索へ良い影響をもたらすことが確認されている²⁾。また、それらの非劣解集合を探索へ参加させることによりパレート最適フロントに対する収束性も向上させている。

欠点

SPEA では、新たなパラメータとしてアーカイブ個体の数 $\overline{N}$ が追加された。現在の探索個体群の数 N とアーカイブ個体の数 $\overline{N}$ のバランスは、SPEA の探索の成否に大きく影響する。もし、 N に対して非常に大きな $\overline{N}$ を用いた場合、エリートを選択圧が大きくなり SPEA はパレート最適フロントに対して収束しなくなる可能性がある。一方、 N に対して $\overline{N}$ が小さすぎた場合、エリート保存の効果が失われる。さらに、母集団における多くの解が、どのアーカイブ個体群の個体からも優越されず等しい適合度が割り振られる。SPEA の研究者らは、保存個体群と探索個体群のサイズ比を 1:4 として用いている。

また、SPEA における適合度割当てでは、多くの個体を優越しているアーカイブ個体が悪い適合度となる。このような割当ては、非劣個体の近くに優越されている個体が集中している場合に対する適切な適合度割当てとなる。しかし、このような場合は非常にまれであり、本来、混雑度を重視するのはクラスタリング手法に限定するべきである。そのため、この適合度割当ては、結果として非劣個体に対して不適切な選択圧を生む可能性がある。

3.3.8 NSGA-II

NSGA-II(Elitist Non-Dominated Sorting Genetic Algorithm) は、NSGA の改良アルゴリズムとして Deb, Agrawal らによって 2001 年に提案された⁸⁾。NSGA-II では、NSGA と比較して次のような特徴が追加されている。

- 探索過程において見つかった非優越解の保存 (アーカイブ個体群の利用) の導入。
- 混雑度トーナメント選択 (Crowded Tournament Selection Operator) の導入。
- 混雑距離 (Crowding Distance) の導入。

また、NSGA-II の NSGA からの変更点を以下に示す。

- 解の多様性の基準を設計変数空間から目的関数空間へ変更.
- 適合度割当て方法の変更.
- 選択方法の変更.

上記の通り、NSGA-IIはアルゴリズムの重要な部分の多くが NSGA から変更されている。逆に、非優越ソートを用いて個体のランキングを行っている点が唯一の共通点と言っても良い。以下、NSGA-II のアルゴリズムの流れ、混雑度トーナメント選択、混雑距離について説明する。

アルゴリズムの流れ

NSGA-IIでは、保存する母集団 P_t と交叉・突然変異といった遺伝的操作を用いた探索を行うための母集団 Q_t の2つの独立した母集団を用いて解探索を進めていく。NSGA-IIでは、非劣個体を保存する母集団 P_t を親母集団として、探索を行うための母集団 Q_t を子母集団として用いることにより解探索を行っている。具体的には、まず世代 t における親母集団 P_t から遺伝的操作を用いた探索を行うための子母集団 Q_t を選択する。 Q_t に対して各遺伝的操作を行い Q_t を更新する。次に、各遺伝的操作を行った Q_t と親母集団 P_t を組み合わせ $R_t = P_t \cup Q_t$ を生成し、選択操作によって個体数 $2N$ の R_t から個体数 N の P_{t+1} を新たに選択し探索を進めていく。

以下、NSGA-II のアルゴリズムの流れを示す。

Step 1 親母集団と子母集団を組み合わせ $R_t = P_t \cup Q_t$ を生成する。 R_t に対して非優越ソートを行い、全個体をフロント毎 (ランク毎) に分類する： $F_i, i = 1, 2, \dots, etc.$

Step 2 新たな母集団 $P_{t+1} = \emptyset$ を生成。変数 $i = 1$ とする。

$|P_{t+1}| + |F_i| < N$ を満たすまで、 $P_{t+1} = P_{t+1} \cup F_i$ と $i = i + 1$ を実行。

Step 3 混雑度ソート (Crowding-sort) を実行し、最も多様に広がっていた個体 $N - |P_{t+1}|$ 個を P_{t+1} に加える。

Step 4 P_{t+1} を基に、混雑度トーナメント選択、交叉、突然変異を用いて新たな子母集団 Q_{t+1} を生成する。

このように NSGA-IIでは、親母集団 P_t と子母集団 Q_t を組み合わせ R_t の上位 N 個体を選択し、次世代の親個体 P_{t+1} としている。また、探索個体 (子個体) Q_t は、親個体 P_t から混雑度トーナメント選択を用いて選抜されており、親個体 P_t のより優れた個体を用いて各遺伝的操作を用いた探索が行われている。

一方，常に優良個体を保存する親母集団 P_t と探索を行う子母集団 Q_t を分けて保持することにより，それまでの探索で発見した優れた解が欠落するのを防いでいる．NSGA-II の親母集団 P_t 更新の概念図を図 3.9 に示す．

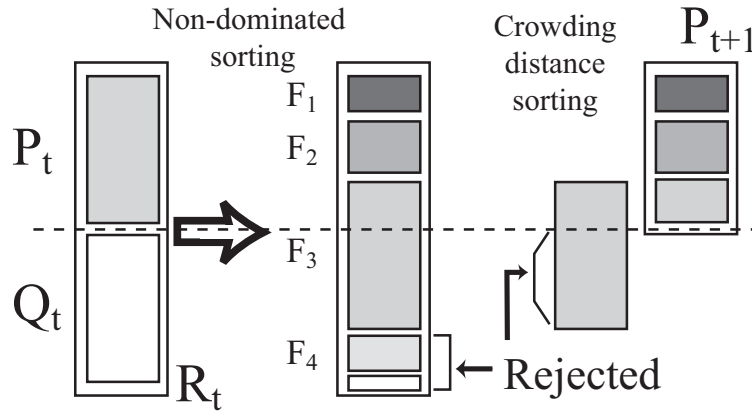


図 3.9 Schematic of the NSGA-II procedure

混雑度トーナメント選択

混雑度トーナメント選択 (Crowded Tournament Selection Operator) は，各ランク (r) により形成されるフロント (F_r) が一様に分布するように選択を行っており，トーナメントサイズ 2 のトーナメント選択にもとづいた方法である．この選択方法では，全ての解 i に対して次の 2 つの属性を持たせ，それらを選択基準に選択している．

- 1) 母集団における非優越ランク (i_{rank})
- 2) 母集団内の局所的混雑距離 ($i_{distance}$)

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

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

混雑距離

混雑距離 (Crowding Distance) とは, ある個体 i の周りにおける個体の密度を評価するための手法である. 混雑距離は, 同一ランクどうし (同一フロント内) で用いられ, 各目的関数軸において隣り合う個体間との距離を足し合わせたものである. 混雑距離の概念図を図 3.10 に示す.

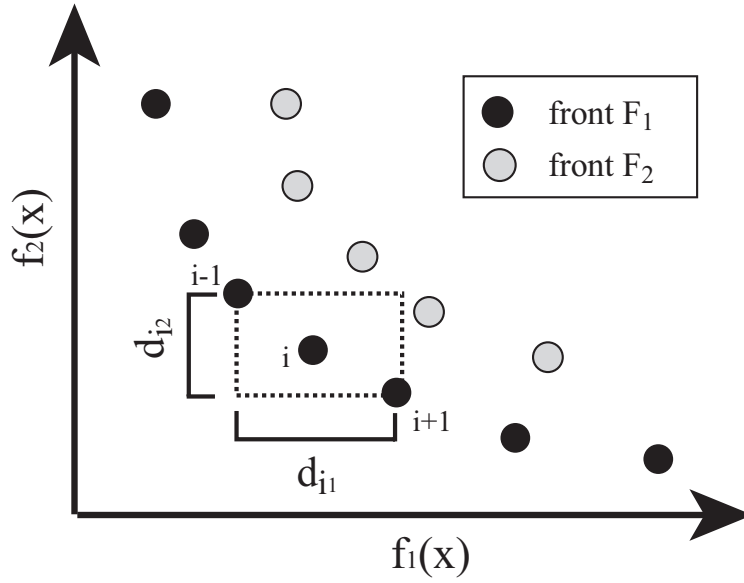


図 3.10 The crowding distance calculation

以下に混雑度距離を計算するためのアルゴリズムを示す.

Step C1 フロント F に含まれる個体の数を変数 l に代入する: $l = |F|$. また, このフロントに含まれる各個体 i に対して初期値設定を行う: $d_i = 0$.

Step C2 各目的関数 $m = 1, 2, \dots, M$ に対して, 目的関数値が悪い順に個体をソートする: $I^m = \text{sort}(f_m, >)$.

Step C3 各目的関数 $m = 1, 2, \dots, M$ に対して, まず境界個体 (目的 m の最大値と最小値の個体) に対して最大距離, もしくは無限距離を与える: $d_{I_1^m} = d_{I_l^m} = \infty^6$. さらに境界個体以外の全ての個体 ($j = 2, \dots, l-1$) に対して以下の式に従った混雑度計算を行う.

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{I_{j+1}^m} - f_m^{I_{j-1}^m}}{f_m^{\max} - f_m^{\min}} \quad (3.16)$$

⁶ I_j は, ソートした個体の j 番目の個体を意味する.

利点

最大の特徴は、探索性能の良さである。適切なアーカイブが行われている上、必ず非劣解の各目的軸に対する両端が次世代へ保存される仕組みであるために、得られている解集合がパレート最適フロントの一部に収束することはない。NSGA-IIにおいて用いられている混雑度計算では、近傍間の個体密度が用いられており、 σ_{share} などの余分なニッチングパラメータを必要としない。

欠点

混雑度距離にもとづく混雑度比較が母集団内の非劣個体の数を制限するために用いられている時、すなわち親母集団 P_t と子母集団 Q_t を組み合わせた $R_t = P_t \cup Q_t$ 内に N 個以上の非劣解集合が存在し、混雑度比較を用いた非劣解集合の削減を行った場合、アルゴリズムは収束の特性を失う。これは、探索の後半になり個体制限 N 以上の非劣解集合が存在した時に、それまでの探索で得られた非劣解が欠落する可能性があるためである。

3.3.9 SPEA2

SPEA2(Strength Pareto Evolutionary Algorithm 2)は、SPEAの改良アルゴリズムとして Zitzler らによって 2001 年に提案された⁹⁾。SPEAは1999年に提案された比較的新しい探索性能に優れたアルゴリズムである。しかし、1999年から2001年までの間には、NSGA-IIを初めとして重要なアルゴリズム、メカニズムが提案され、SPEAには幾つかの改良点があることが明確になった。そこで、さらに効率の良い高性能な探索を行うため、幾つかの重要な改良を加え提案されたのが SPEA2である。SPEAと比較した SPEA2の主な特徴を以下に示す。

- 改良した適合度割当て手法

これは、各個体に対してどれだけ多くの数の個体をその個体が支配しているか、支配されているかを考慮した手法である。この適合度割当てには、個体の優越度合いと密集度の両方が考慮されている。

- 新たなアーカイブ端切り手法

この手法は、アーカイブ個体群を適切な数に削減するために用いられる。この手法では、得られた非劣解集合の各目的軸における端の個体が保存されることを保証している。

- アーカイブ個体群と探索個体群の扱い

SPEA2では、常にこれまでの探索で発見した $\overline{N}$ 個体の優良解をアーカイブに保存

する。また、NSGA-IIと同様にアーカイブ個体群から探索個体群をさらに選択して、アーカイブ個体群のより優れた個体を再抽出して各遺伝的操作を用いた探索を行っている。

以下、SPEA2のアルゴリズムの流れ、適合度割当て手法、アーカイブ端切り手法を用いたアーカイブ個体群の更新について解説する。

アルゴリズムの流れ

SPEA2のアルゴリズムの流れはNSGA-IIに非常に似ている。これは、2つの母集団(アーカイブ母集団 $\overline{P}$, 探索母集団 P)を用いて探索を行う点、アーカイブ母集団から探索母集団を選択する点、探索母集団を用いたアーカイブ母集団の更新を行う点などアルゴリズムとして共通する部分が多いからである。

Step 1 **初期化**: 初期母集団 P_0 を生成する。空のアーカイブを生成する: $\overline{P}_0 = \emptyset$. Set $t = 0$.

Step 2 **適合度割当て**: P_t と $\overline{P}_t$ における個体適合度を計算する (3.3.9 節を参照)。

Step 3 **環境選択**: P_t における全ての非劣個体を $\overline{P}_t$ へコピーし、 $\overline{P}_{t+1}$ とする。ただし、 $|\overline{P}_{t+1}| > \overline{N}$ の場合には、端切りオペレータを用いて $\overline{P}_{t+1}$ に削減する。また、 $|\overline{P}_{t+1}| < \overline{N}$ の場合には、 P_t における優良個体 $\overline{N} - |\overline{P}_{t+1}|$ 個分を $\overline{P}_{t+1}$ へコピーし、 $\overline{P}_{t+1}$ の個体数を $\overline{N}$ にする。

Step 4 **終了判定**: もし $t \geq T$ もしくはその他の終了条件が満たされた場合、 $\overline{P}_{t+1}$ の中の非劣個体群が最終的な解として出力され探索は終了する。そうでなければ、Step 5へ進む。

Step 5 **メイティング選択**: $\overline{P}_{t+1}$ からバイナリトーナメント選択によって N 個分の P_{t+1} を選択する。

Step 6 **変化**: P_{t+1} に対して交叉と突然変異オペレータを実行する。

上記の環境選択 (Environmental Selection), メイティング選択 (Mating selection) について説明する。一般に、新たに得られた探索個体群 P_t を用いて保存しているアーカイブ個体群 $\overline{P}_t$ を更新する場合、探索個体群 P_t とアーカイブ個体群 $\overline{P}_t$ から選択操作を用いて新たなアーカイブ個体群 $\overline{P}_{t+1}$ を生成する。この選択操作のことを環境選択と呼ぶ。

一方、アーカイブ個体群から探索個体群を選択することをメイティング選択という。交叉 (Mate) などの遺伝的操作を行う個体群の選択という意味でメイティング選択と呼ばれる。これは、探索をより優れた個体のみで行うためである。SPEA2, NSGA-II などではバイナリトーナメント選択 (binary tournament selection) が用いられている。

適合度割当て手法

SPEA2では、まず全ての個体 i に対して支配している個体の数 $s(i)$ を求める。次に、支配されている個体の適合度値 $f(i)$ は、その個体を支配している個体の持つ $s(i)$ を全て足し合わせた値となる。そのため個体 i が非劣解の場合、適合度は 0 となる。

SPEA と SPEA2 における適合度割当ての比較例を図 3.11 に示す。

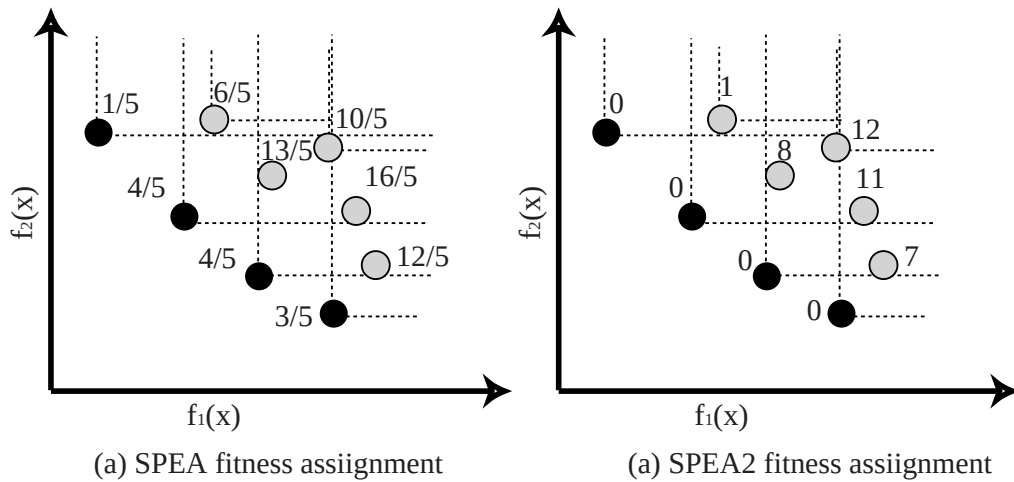


図 3.11 Comparison of fitness assignment schemes in SPEA and SPEA2

図 3.11 から分かるように、SPEA2では、より多くの個体を支配している非劣個体の $s(i)$ は、高くなる。そのため、そのような非劣解に支配されている個体の適合度は悪くなる。一方、SPEA の場合と異なり SPEA2では、非劣解に対しては同等の最も高い適合度が割当てられている。そのため、SPEA に比べパレート最適フロントに対する収束性がより強くなっていることが分かる。

環境選択

SPEA2でのアーカイブ母集団の更新は、SPEA と比較して次の 2 点において異なっている。

- (i) アーカイブ母集団の個体数が常に一定である。
- (ii) 過剰な非劣解の削減方法として、端切り手法 (archive truncation method) という境界の個体を取り除くことを防ぐ手法を用いている。

SPEA2における環境選択では、最初のステップとして全ての非劣個体をコピーする。すなわち、アーカイブ $\overline{P}_t$ と探索母集団 $\overline{P}_t$ から最良の適合度値 0 を持つ個体を $\overline{P}_{t+1}$ へコピー

する.

$$\overline{P}_{t+1} = \{i | i \in P_t + \overline{P}_t \wedge F(i) < 1\} \quad (3.17)$$

もし、非劣個体の数がアーカイブ母集団サイズの上限と同値であった場合 ($|\overline{P}_{t+1}| = \overline{N}$), 環境選択のステップは終了する. 一方, もし非劣解の数がアーカイブ母集団サイズの上限以下の場合, 上限以上の場合もある.

前者の場合, 前のアーカイブにおける最良の $\overline{N} - |\overline{P}_{t+1}|$ 個の優越個体と母集団が新しいアーカイブとしてコピーされる. 具体的には, まずアーカイブ母集団と探索母集団を足し合わせた個体集合 $P_t + \overline{P}_t$ を生成する. 個体集合 $P_t + \overline{P}_t$ に対して適合度の最良値順にソートを行い, ソート個体の $F(i) \geq 1$ となる初めの個体から $\overline{N} - |\overline{P}_{t+1}|$ 個の個体 i を $\overline{P}_{t+1}$ にコピーすることにより実行される.

また, 後者の上限以上の場合, 母集団の個体を 1 個ずつ削減するアーカイブ端切り手法を $|\overline{P}_{t+1}| = \overline{N}$ となるまでを繰り返し行うことにより, アーカイブ母集団 $\overline{P}_{t+1}$ の個体数を $\overline{N}$ としている.

アーカイブ端切り手法のアルゴリズムを以下に示す.

Step A1 全ての非劣個体に対して, 目的関数空間において最も隣り合う距離が最小の i と j を選択する. 変数 $k = 2$ とする.

Step A2 個体 i と j に対して k 番目に最も接近している個体との距離 $(\sigma_i^k, \sigma_j^k)^7$ を求め比較する.

Step A3 もし, $\sigma_i^k > \sigma_j^k$ もしくは $\sigma_i^k < \sigma_j^k$ の場合には, 距離の短い方を削減対象とする. また, $\sigma_i^k = \sigma_j^k$ の場合には, $k = k + 1$ として Step A2 へ戻る.

このアーカイブ端切り手法の特徴は, 非劣個体群の各目的の端に存在する個体が必ず削減対象とならないことである. また, 必ず最も隣り合う距離の最小の個体が削減されるため妥当性をもっとも高いアーカイブの削減方法である. さらに, パラメータを用いない点も利点といえる.

アーカイブ端切り手法の概念図を図 3.12 に示す.

利点

SPEA2 の最大の特徴は, 探索性能の良さである. SPEA と比較しても, その探索性能は格段に向上している⁹⁾. アルゴリズム自体は NSGA-II に非常に似ているが, 探索途中において発見した非劣解の削減方法などは NSGA-II のものよりも妥当性の高い方法を用いて

⁷⁾ σ_i^k は, 個体 i において k 番目に最も接近している個体との距離を意味する

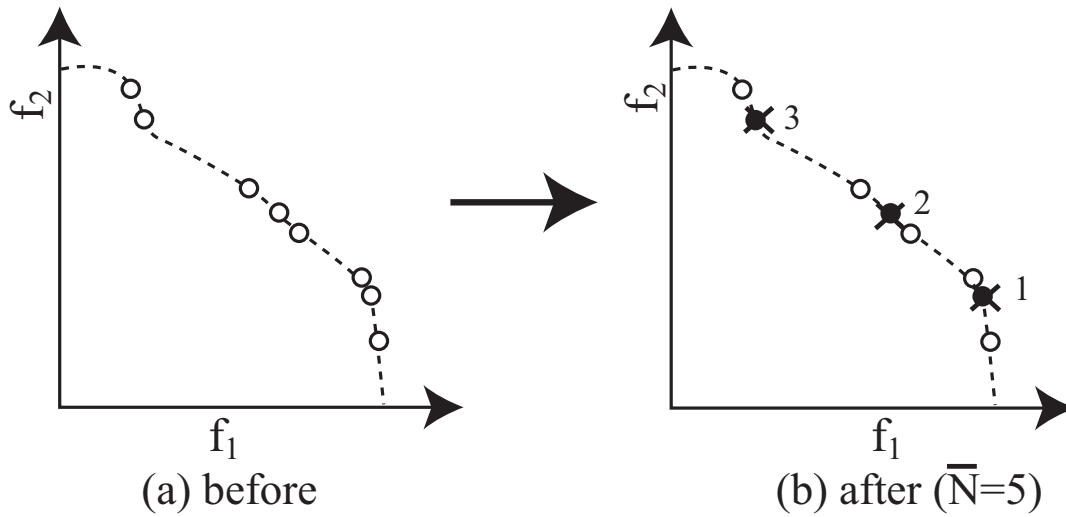


図 3.12 Archive truncation method used in SPEA2

いる。また、適合度割当てに関しても非劣解が必ず優遇されるような仕組みになっており、SPEA の場合の適合度割当て方法の欠点が改善されている。

欠点

SPEA2において用いられている、端切りオペレータは非常に妥当性の高いものであるが、1度の試行において削減対象の個体を1つしか発見できない上、最も近接している2つの個体の σ^k に関して差が生じるまで個体間の比較を行う必要がある。そのため、その複雑性は高くなる。最悪の場合、端切りオペレータの複雑性は $O(M^3)(M = N + \bar{N})$ となる。ただし通常は、第2もしくは第3の近傍に関しては異なっているので、平均の複雑性は $O(M^2 \log M)$ 以下となる。

3.4 多目的遺伝的アルゴリズムの問題点

上述の通り、多目的GAにおいてこれまで提案されてきたアルゴリズムは非常に膨大であり、特に1999年から2002年度の間において非常に大きな進歩を遂げているのが分かる。特に、NSGA-IIやSPEA2は他のアルゴリズムと比較して非常に優れた結果を残している^{2,9)}。

一方、多目的GAは以下のような理由により一般に膨大な計算が必要となる。

- 評価する目的の数が複数存在。

- 求める解が複数存在する上、探索領域がより広範囲となるため、単一目的に比べ探索が困難.

そのため、実問題を対象とした場合、実時間内で良好な結果を得るためには以下のいずれかを実装する必要がある.

- i) 探索効率の良いアルゴリズムの適用.
- ii) 並列処理による計算時間の短縮.
- iii) 探索効率の良いアルゴリズムと並列処理の組み合わせ.

このうち、探索効率の良いアルゴリズムについて見た場合、NSGA-II, SPEA2 など非常に探索効率の良いアルゴリズムが既に提案されている. しかし、大域的な探索を行う多目的 GA では、探索個体どうしの目的関数空間距離が大きく離れ効果的な交叉を行うことができないといった問題点が存在する. これを防ぐためには、交叉のためのペア個体を選ぶ際、個体どうしの近接度合を何らかの方法により考慮する必要がある. しかし、多目的 GA における研究では、この近接度合いを考慮した交叉に関する議論はほとんど行わなわれておらず、従来までに提案されてきたアルゴリズムには改良の余地があると思われる.

また、多目的最適化 GA の並列化に関する研究を見た場合、いくつかの研究が見られるが²²⁻²⁴⁾、その数は多くない. また、そこで使用されている計算モデルは単一目的における GA の並列化、例えばマスタースレーブ型モデルや²³⁾、分割母集団モデル^{22, 24)} とほぼ同様である.

しかし、単一の最適解を探索する単一目的 GA と複数のパレート最適解を探索する多目的 GA では、求められる性能が異なる. そのため、単純に単一目的 GA における分割母集団モデルを多目的へ適用しても良好な結果を得ることはできず、多目的 GA を並列実装するためには、アルゴリズム自体に変化のないマスタースレーブ型モデルを選ばざるを得ない.

一方で、並列計算機上での実装を想定した場合には分割母集団モデルが有利であると考えられる. すなわち、分割母集団モデルでは通信負荷が低いため、プロセッサ数が多い並列機や PC クラスタのようなネットワーク性能の低い並列計算機には適している. そのため、分割母集団モデルに適した多目的 GA アルゴリズムは非常に重要な意味を持っている.

そこで、本研究では実用的な多目的 GA アルゴリズムの実現という観点から次の2つの取り組みを行った.

1. 多目的の特性を考慮した分割母集団モデルの多目的 GA アルゴリズム
2. 近接度合いを考慮した交叉を実装した新たな多目的 GA アルゴリズム

次章以降では，上記に関する本研究の取り組み，新たな多目的 GA アルゴリズムの提案とその有効性の検証について説明する．

第 4 章

得られた非劣解集合に関する評価方法

4.1 はじめに

得られた非劣解集合に対する評価は、適用したモデルの性能を評価するうえで不可欠である。特に、多目的最適化では一意的に解の評価を行うことができない。そのため、非劣解を評価する何らかの方法が必要となる。

一般に、多目的 GA を用いて探索した非劣解集合には、次の 3 点の性質が求められる。

- パレート最適フロントに対する近接度合い，精度。
- パレート最適フロント全体に対する幅広さ。
- パレート最適フロント全体に対する隙間のなさ，分散。

そのため、非劣解集合に対する評価も上記の 3 点を考慮したものでなければならない。これまで、上記のいずれか、もしくは上記の幾つかの性質を考慮した評価手法が複数、提案されている^{6, 22, 25, 26)}。これらの評価手法は、得られた非劣解集合から絶対的な評価を行う手法と、2 つ、もしくはそれ以上の非劣解集合の比較に基づいて相対的な評価を行う手法の 2 つに分類することができる。

パレート最適フロントの形状が既知である場合、絶対的な評価を行うことができる前者の手法は特に有効である。これは、絶対的な評価を行ういくつかの手法では、得られた非劣解集合とパレート最適フロントとの比較により評価値を決定しているからである。しかし、問題によってはパレート最適解が未知であるため、必ずしも絶対的に非劣解集合を評価できるわけではない。

一方、2 つ以上の非劣解集合の比較にもとづく評価手法では、パレート最適解の未知・既知に関わらず適用することができる。特に、複数のアルゴリズムの優越を比較したい場合には、有用である。

本論文で取り上げる評価方法を以下に示す.

- i) パレート最適フロントに対する誤差
- ii) 被覆率
- iii) 各目的関数軸ごとの最大値, 最小値, 平均値
- iv) 優越個体割合
2つ以上の非劣解集合の比較にもとづく評価手法であり, 相手に対して非劣である解の数を比較する単純な比較手法である. この評価手法を用いることにより, 解集合の精度に関する評価を行うことができる.
- v) パレートフロントとサンプリング直線の交点にもとづく評価手法
2つ以上の非劣解集合の比較にもとづく評価手法であり, 精度, 幅広さ, 隙間のなさなどを総合的に評価する手法である.

以下, 上記で示した評価手法について説明する.

4.2 評価方法

4.2.1 パレート最適フロントに対する誤差 (error : I_{error})

誤差 (I_{error}) は, 得られた非劣解を絶対的に評価する手法であり, 精度の評価を行う. パレート最適解が既知の場合, 各非劣解とパレート最適解とのユークリッド距離の平均は誤差とみなせる. 誤差の値は, 探索した非劣個体のパレート最適解集合への近接度合いを表している. しかし, 3 目的以上の場合, 得られた解に対する最近接点を計算することが困難となる. そこで, 本研究では, 以下のような簡易的な精度測定法を用いた.

得られた個体 k を $x^k \cong (x_i^k, \dots, x_m^k)$ とし, パレート最適解が $f(x_1, \dots, x_m) = C$ で表されるものとする. 簡易誤差 $e(x^k)$ を, 式 (4.1) に示す.

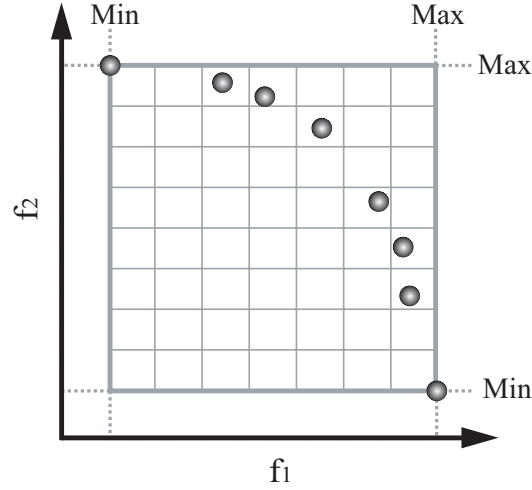
$$e(x^k) = C - f(x_i^k, \dots, x_m^k) \quad (4.1)$$

また, パレート最適解集合の誤差 I_{error} は $e(x)$ の平均値として求めた.

$$E = \frac{1}{N} \sum_{k=1}^N e(x^k) \quad (4.2)$$

なお, 式 (4.2) における N は非劣解の数を意味する.

ただし, この評価基準はパレート最適解が既知の場合でなければ使用できない.

図 4.1 Schematic of I_{cover}

4.2.2 被覆率 (cover rate: I_{cover})

被覆率 (I_{cover}) は，得られた非劣解を絶対的に評価する手法であり，パレート最適フロントに対する幅広さ，隙間のなさの評価を行う．本研究では，比屋根により提案された被覆率²²⁾を簡略化した以下のような方法を用いた．

各目的関数に対する最大値と最小値の間を任意数に分割し，分割された領域をどれだけカバーしているかを評価値として用いた．具体的な定式を以下に示す．

ある目的関数 f_k における被覆率 I_{cover_k} は，

$$I_{cover_k} = \frac{N_k}{N} \quad (4.3)$$

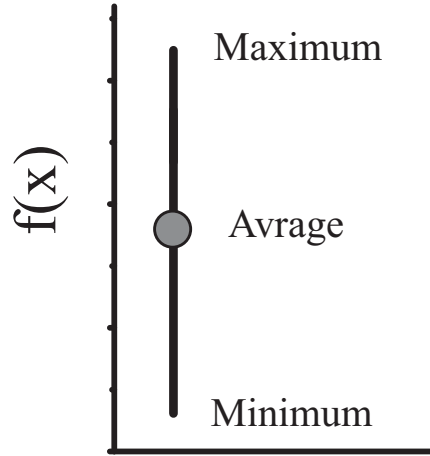
により求められる．なお，式 (4.3) における N は分割数， N_k は非劣解を含む小領域の数を意味する．

また，非劣解の被覆率 I_{cover} は各目的関数における被覆率 I_{cover_k} の平均により表される．

$$I_{cover} = \frac{1}{N} \sum_{k=1}^N I_{cover_k} \quad (4.4)$$

ゆえに，被覆率の最大値は 1，最小値は 0 となり，最大値 1 に近いほど良い解であるといえる．被覆率の評価方法に関する概念図を図 4.1 に示す．

なお，各目的関数の最大値と最小値に関して，パレート最適フロントが既知の場合にはパレート最適フロントから求め，パレート最適フロントが未知の場合には経験的に得られた最大値と最小値を用いた．

図 4.2 An example of I_{MMA}

4.2.3 各目的関数軸ごとの最大値, 最小値, 平均値 (Max and Min and Average: I_{MMA})

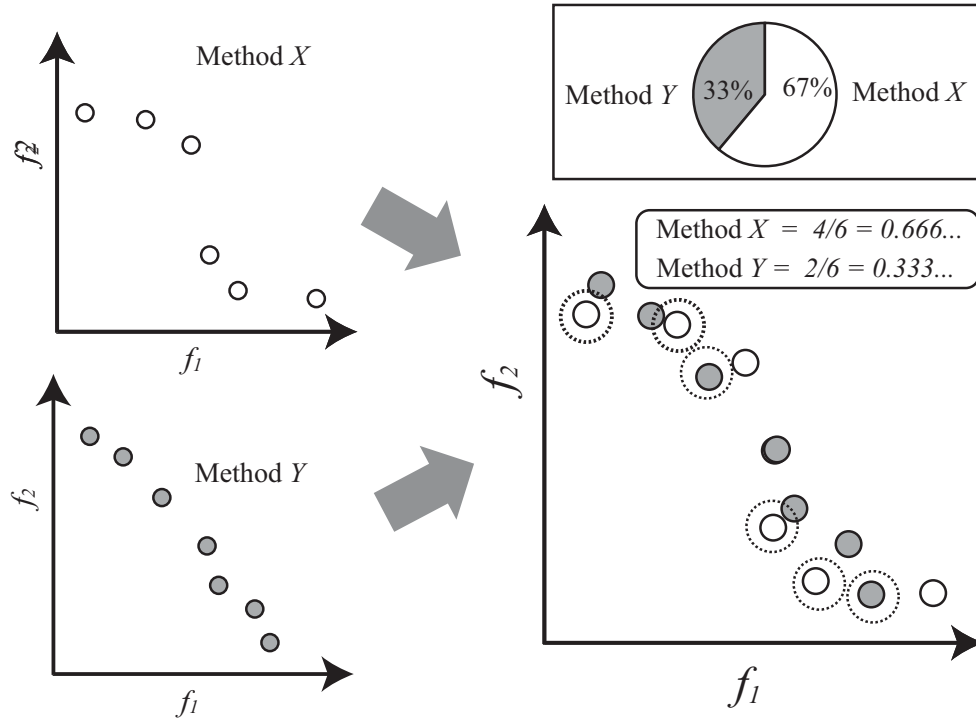
この評価手法は, 得られた非劣解を絶対的に評価する手法であり, パレート最適フロントに対する幅広さの評価を行う手法である. 得られたパレート最適解集合における各目的関数軸の最大値と最小値および平均値を求めることにより, 得られた解の幅広さについて評価することができる. 本論文では, これらの値を示したものを I_{MMA} として用いる. I_{MMA} の例を図 4.2 に示す. この図から分かるように, I_{MMA} では, 各目的の最大値, 最小値がプロットされており, 平均値が円印にて示されている.

4.2.4 優越個体割合 (Ratio of Non-dominated Individuals: I_{RNI})

I_{RNI} は, 2つの非劣解集合を比較し, 相手に対して非劣である解の数を求める比較手法である. この手法では, 精度に関する評価を行う. この手法は, Tan らによって用いられていた手法²⁶⁾を2つの非劣解集合の比較へと拡張したものである. 本手法の比較手順を以下に示す.

まず, 2つの手法で得られた解集合 X と Y の和集合をとり S^U とする. 次に, S^U の中から, どの解にも優越されない解のみを選び出し, 選ばれた解集合を S^P とする. そして, S^P の各手法の割合を $I_{RNI(X,Y)}$ として導き出すというものである. 図 4.3 に $I_{RNI(X,Y)}$ の例を示す.

そのため, この割合は最大値の 100 %に近いほど, もう一方の手法を優越している, すなわち, より真の解に近い解が得られているものと判断することができる. ただし, 本手法では非劣解集合の優越関係しか考慮していないため, 得られた非劣解の評価を行うため

図 4.3 Schematic of $I_{RNI}(X,Y)$

には、非劣解の幅広さに関する何らかの評価基準が必要となる。

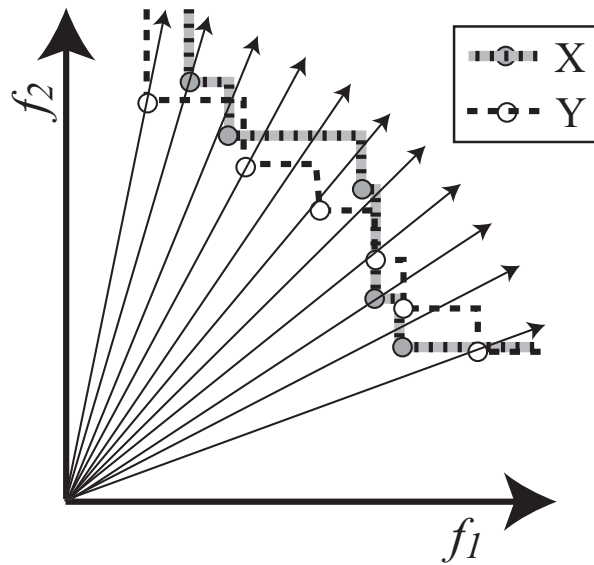
4.2.5 パレートフロントとサンプリング直線の交点にもとづく評価手法 (Sampling of the Pareto Frontier Lines of Intersection: I_{LI})

) この手法は, Knowels と Corne により提案された 2 つ以上の非劣解集合の比較を行うための手法²⁵⁾である。この手法では、複数の方向ベクトル軸上における非劣解集合同士の優越比較を行い、どちらの非劣解がどの程度優れているのかを判断する。

2 つの非劣解集合 (X, Y) を対象とした場合のこの手法の概念図を図 4.4 に示す。

まずこの手法では、得られた非劣解集合から形成される到達フロント (図 4.4 における点線のライン) を計算する。次に、パレート領域を一樣にサンプリングするようなサンプリング線を決定し、このサンプリング線と各非劣解の到達フロントの交点を求め、比較する。2 つの比較集合 (X, Y) がある時、 $I_{LI}(X, Y)$ は、交点の比較において X が Y を優越していた割合を表す。それゆえ、 X にとって最良の結果は $I_{LI}(X, Y) = 1$ であり、 $I_{LI}(Y, X) = 0$ となる。

本手法の最大の弱点は、サンプリングする線の選択に結果が大きく依存することである。そのため、本研究では、この $I_{LI}(X, Y)$ に対して以下の 3 つの改良を行っている。

図 4.4 Schematic of I_{LI}

- i) 非劣解集合の各目的の最大値・最小値を用いて目的関数値のスケールを行っている。

対象とする問題によっては、非劣解集合の分布する目的関数軸のスケールが大きく異なる場合が考えられる。最大値・最小値を用いたスケールを行うことで、このような目的関数ごとのスケールの差がなくなる。

- ii) 非劣解集合の各目的の最大値・最小値の間に一様なサンプリング線を設定する。

これは、サンプリング線がパレート最適解集合の分布している領域に限定されるようにするためである。

- iii) サンプリングの線を多く用いる。

サンプリングの線が 10, 100 程度では線を選択の仕方により結果が異なることが考えられるため、1000 本のサンプリング線を使用する。

上記の 3 つの改良を行うことにより、サンプリング線に依存しにくい評価を行うことができる。さらに、ある非劣解集合 X が別の非劣解集合 Y を完全に優越している場合には、必ず $I_{LI}(X, Y) = 1, I_{LI}(Y, X) = 0$ となるため、得られた結果に対する信憑性も向上するものと思われる。

4.3 まとめ

本節では、得られた非劣解集合そのものを絶対的に評価する手法として誤差 (I_{error}), 被覆率 (I_{cover}), 各目的関数軸ごとの最大値・最小値・平均値 (I_{MMA}) の3つの評価手法について説明した。また、2つ以上の非劣解集合の比較に基づいて相対的な評価を行う手法として、優越個体割合 (I_{RNI}), パレートフロントとサンプリング直線の交点にもとづく評価手法 (I_{LI}) の2つの評価手法について説明した。

これらの評価手法のうち、誤差はパレート最適フロントが既知でなければ求めることができないが、その他の評価手法はパレート最適フロントの既知、未知に関わらず使用することができる。これらの手法は、非劣解の評価性質から大きく次の3つに分類することができる。

1) パレート最適フロントに対する近接を評価する手法。

これは、パレート最適フロントとの誤差にもとづく評価手法、もしくは何らかの形でパレート最適解の概念にもとづく評価手法が該当する。そのため、誤差 (I_{error}), 優越個体割合 (I_{RNI}) がこの分類に含まれる。

2) パレート最適フロント全体に対する幅広さと隙間のなさを評価する手法。

この分類には、被覆率 (I_{cover}), 各目的関数軸の最大値・最小値・平均値 (I_{MMA}) といった非劣解の幅広さ、隙間のなさのみを評価する手法が分類される。被覆率は、得られた非劣解の隙間のなさについて、各目的関数軸の最大値・最小値・平均値は、非劣解の幅広さについての評価を行っている。

3) 総合的に非劣解を評価する手法

これは、上記の両方の評価を何らかの形で含んだ評価手法が分類され、パレート最適フロントとサンプリング直線の交点にもとづく評価手法 (I_{LI}) が該当する。この手法では、非劣解により形成される近似パレート最適フロントの幅広い範囲においてベクトル軸上での優越比較判定を行っている。そのため、近似パレート最適フロントに対する近接度合いと幅広さ、隙間のなさを総合的に評価しているといえる。

本論文では、次章以降における数値実験において、本節の各種評価手法を用いて得られた非劣解の評価を行っている。

第 5 章

多目的遺伝的アルゴリズムの分割母集団モデルの検討

5.1 はじめに

3 章において説明した通り，GA の多目的最適化への応用に関する研究は非常に数多く行われており，幾つかの研究では非常に優れた結果が得られている²⁾．一方，多目的 GA は複数の目的関数および制約条件の値を繰り返し評価する必要がある，膨大な計算時間が必要となる．このため，並列処理により計算時間を短縮することは重要な課題となる．

GA のアルゴリズムは他の最適化手法と比較して，優れた並列性を持っている．これは，GA の評価，選択，交叉，突然変異といった遺伝的操作は全て並列に処理することが可能であるためである．GA の並列化に関する研究は単一目的を中心に活発に行われており²⁷⁻²⁹⁾，多くのモデルが考案され，その検証が行われている．

その中でも，評価値の計算を並列化するモデルであるマスタースレーブモデル，母集団をいくつかのサブ母集団¹に分割しそれぞれのサブ母集団内で GA を行う分割母集団モデルが代表的である．また，分割母集団モデルは通信負荷が低いだけでなく，単一母集団モデルと比較して，解を求めるために必要な計算量そのものが減少することが知られている³⁰⁾．

一方で，多目的最適化 GA の並列化に関する研究はいくつか見られるが²²⁻²⁴⁾，その数は多くない．また，そこで使用されている計算モデルは単一目的における GA の並列化とほぼ同様で，例えばマスタースレーブモデルや²³⁾，分割母集団モデルである^{22, 24)}．

単一目的における GA の場合，探索初期段階では，各個体は探索領域全体を探索する大域的探索を行い，探索が進むにつれて，各個体は 1 つの解に収束し，局所探索を行うという振る舞いを示す．そのため，少ない個体数の場合には初期収束を起こし，個体数が必要以

¹一般に，母集団を分割した分割母集団のことをサブ母集団という．また，サブ母集団の単位として島を用いるのも一般的である．そのため，4 つのサブ母集団を用いた探索のことを，4 島による探索という．

上に多い場合には余分な計算が必要となる。それに対して、多目的における GA では、パレート最適解集合が探索領域内の広範囲に広がっている場合が多いため、探索の初期段階においても、最終段階においても、探索領域全体に対する大局的探索が必要となる。また、それと同時に、各個体はパレート最適解へ近付く必要性があるので、局所探索も必要となる。これらのことから、個体数は多いほど、より広い範囲での精度の良いパレート最適解が求められることになる。すなわち、単一目的の場合と多目的の場合とでは、GA に求められるメカニズムが異なる。

通常の分割母集団モデルを多目的問題に適用した場合、各サブ母集団内の個体数は、単一母集団のものと比較してサブ母集団数分の 1 と減少するため、各サブ母集団で求められる非劣解は劣化する。また、各サブ母集団内では非劣解であっても、全体の解集合と比較した場合には、非劣解ではなくなる場合が考えられるため、探索効率も良いとはいえない。さらに、探索個体全体を把握することが難しいため、全体としての多様性維持も困難となる。そのため、単一母集団モデルであるマスタースレーブモデルと分割母集団モデルとを比較すると、同一の精度の解を求めるには、前者のモデルの方が有利となる。一方、並列計算機上での実装を考えた場合、後者の分割母集団モデルの方が通信負荷が低いため、プロセッサ数が多い並列計算機や PC クラスタのようなネットワーク性能の低い並列計算機には適している³¹⁾。

これらの点から、並列計算機を用いて GA を行うには、分割母集団モデルで、かつ、単一母集団モデルと同等の解探索能力を持つ、新しいモデルが望まれる。そこで本研究では、そのような多目的最適化 GA に適した分割母集団モデルとして異なる特徴を備えた以下の 3 つのモデルの提案を行う。

- 全体シェアリング GA (Total Sharing Genetic Algorithm: TSGA)
- 領域分割型 GA (Divided Range Multi-Objective Genetic Algorithm: DRMOGA)
- 分散協力型スキーム (Distributed Cooperation Scheme: DC-Scheme)

以下、GA における代表的な並列モデルの紹介を行った後、それぞれの手法のアルゴリズムと数値実験を用いた従来手法との比較について述べる。

5.2 並列 GA モデル

本節では、代表的な並列 GA モデルとして以下の 3 つのモデルを取り上げて説明する。

- マスタースレーブモデル
- 分割母集団モデル

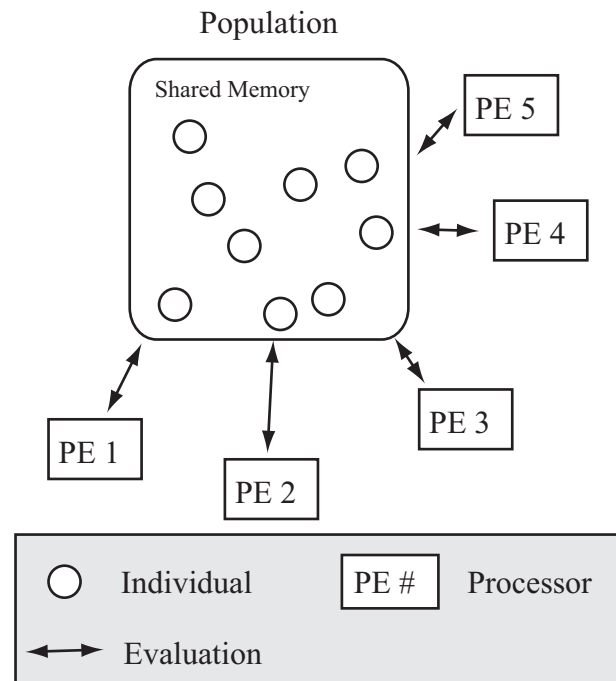


図 5.1 Schematic of master slave model

- 近傍モデル

それ以外の並列モデルについても、そのほとんどが上記の3つの並列モデルを変形したモデルとなっている。以下、上記の3つのモデルについて説明する。

5.2.1 マスタースレーブモデル

マスタースレーブモデル (master slave model) では母集団の分割は行わず、選択と交叉を大域的に行い、個体の評価のみ複数のプロセッサを使用する。そのため、最終的に得られる結果は単一母集団モデルと差がなく、用いるスレーブマシンの台数による解への影響も全くない。しかし、マスタースレーブ間の通信負荷が大きく、評価計算の負荷が軽い場合にはマスターのみに計算負荷が集中してしまうという欠点がある。

そのため、対象問題としては評価計算負荷が高い問題に向いており、用いるアーキテクチャ環境としては複数のプロセッサが単一のメモリを共有可能な、共有メモリ型の並列計算機に向いているといえる。図 5.1 に概念図を示す。

5.2.2 分割母集団モデル

分割母集団モデル (distributed population model) では母集団は、使用するプロセッサ分のサブ母集団 (subpopulation) に分割される。広義では、母集団をプロセッサ分のサブ

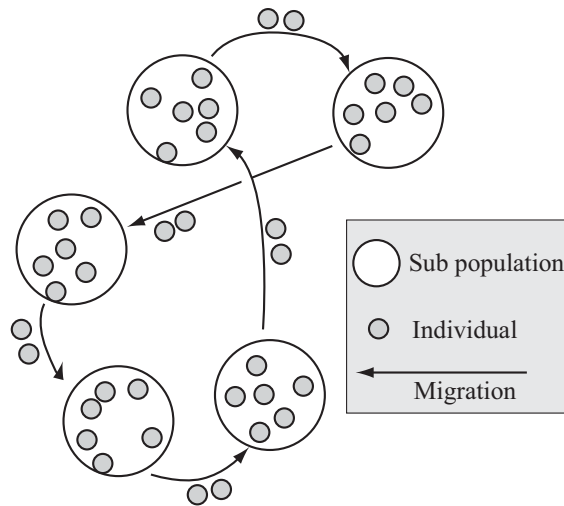


図 5.2 Schematic of distributed population model

母集団に分割するモデルの総称を分割母集団モデルと呼ぶが、分散 GA^{27,28)} と呼ばれる分割母集団モデルの 1 つを意味する場合にも用いられる。サブ母集団は並列計算機上の各プロセッサに分配され、各サブ母集団において GA が行われる。早熟収束の回避と多様性の維持を行うために、一定間隔ごとにサブ母集団間で個体の交換を行う。この操作は移住 (Migration) と呼ばれる。移住を行う世代周期を移住間隔 (Migration interval) と呼び、サブ母集団内全体に対する移住個体の割合を移住率 (Migration rate) という。移住の概念図を図 5.2 に示す。

分割母集団モデルでは、移住の頻度、移住先の選択法、移住個体の選択法といった移住に関わるパラメータ、メカニズムが探索に大きく影響する。また、移住の頻度については各サブ母集団が一斉に移住を行う同期移住や、各サブ母集団が個別の判断により移住を行う非同期移住などがある。この他にも様々な移住のメカニズムが提案されている³²⁾。

このモデルは、一定間隔ごとに行われる移住操作以外に通信の負荷がかからないため、マスタースレーブモデルと比べ通信負荷が低い。そのため、PC クラスタのようなネットワーク性能の低い並列計算機に適している。

5.2.3 近傍モデル

近傍モデル (neighborhood model) では、プロセッサ 1 つに割当てられる個体は 1 つまたは比較的少数であり、プロセッサ間通信のオーバーヘッドを減らすために、近傍のプロセッサの個体とのみ交叉を行う。このモデルは、個体の配置状態が 2 次元格子状空間の区分単位であるセル上に配置されているように見えることから、セルラー GA とも呼ばれる。こ

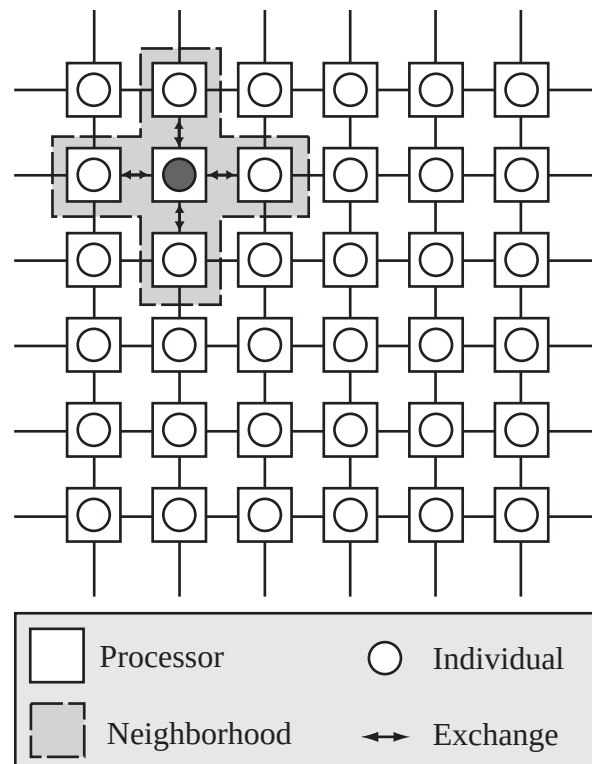


図 5.3 Schematic of neighborhood model

のモデルは、低い性能のプロセッサを極めて多数用いる並列計算機に多く見られる。

図 5.3にその概念図を示す。このモデルは、プロセッサ間通信のオーバーヘッドが少なく極めて効率的ではある。しかし、部分集団の適当な近接構造の設定が困難であること、また、その近接構造の決定の際に並列計算機のアーキテクチャに依存することが問題となる。

5.3 全体シェアリング遺伝的アルゴリズム

5.2節で示したように、これまで大きく3種類のモデルがGAの並列モデルとして提案されている。これら3種類のうち、特に分割母集団モデルは、単一母集団モデルと比較して探索効率が良いことが知られている³⁰⁾。しかしながら、複数のパレート最適解を探索する必要のある多目的GAをそのまま分割母集団モデルへ適用しても良好な探索を行うことは難しい。多目的GAを分割母集団モデルへ適用する際の主な問題点を以下に示す。

- 1) 各サブ母集団の探索領域の重なりに起因する非効率な探索
- 2) サブ母集団あたりの個体数減少にともなう探索能力の低下

全体シェアリング遺伝的アルゴリズム (Total Sharing GA: TSGA) は、上記 1) の問題点、探索領域の重なりに注目した多目的 GA である。この問題は、選択がサブ母集団内でのみ行われていることに起因する。すなわち、選択操作が個体全体を用いて行われていないため、単一母集団の場合と比較して以下の事柄を実現していない。

- 適切な個体間の優越関係の決定
- 適切な個体密度の測定

上記の問題点を解決するために、TSGA では非定期的に全てのサブ母集団の個体、すなわち探索個体全てを用いた選択操作 (全体シェアリング選択) を行う。究極的には、世代ごとに全個体を用いた選択を行うことが理想であるが、膨大な通信負荷が必要となり、分割母集団モデルの利点が失われる。そのため、TSGA では探索中の非劣解の総数が一定数を超えた場合にのみ、全個体を用いた選択操作を行っている。

5.3.1 全体シェアリング GA のアルゴリズム

TSGA は非劣解の総数が一定数を越えた場合、母集団全体を一カ所に集計し、母集団全体を用いた選択操作を行う。本研究では、この選択操作のことを全体シェアリング選択と呼ぶ。これは、全てのサブ母集団の個体情報、すなわち全個体の情報にもとづいてシェアリング操作、および個体の適合度付け、選択を行うためである。

TSGA の特徴を以下に示す。

- i) パレート最適個体保存選択³³⁾の使用。
- ii) 非定期的に母集団全体を用いる選択を実行。

アルゴリズムの流れ

TSGA を用いたアルゴリズムでは、個体数 N の母集団 P は交叉などの遺伝的オペレータによって新たに N 個の個体が生成され総個体数は $2N$ となる。パレート最適個体保存選択によって N 個が選択されることにより、探索の 1 サイクルが終了する。TSGA は、分割母集団モデルに対応しているためこのサイクルは各サブ母集団内で行われる。

なお、上記のパレート最適個体保存選択では全ての非劣個体を選択し、次世代に残すという選択方法を用いている。そのため、個体数が上限の N を越えることも許可している。逆に、非劣個体 N_S が一定数 N に満たなければ ($N_S < N$)、Fonseca らによるランク付け¹⁹⁾をもとに適合度割当てを行い、非劣個体以外から $N - N_S$ 個の個体を選択し、母集団サイズを N としている。

TSGA アルゴリズムにおける探索サイクルを以下に示す．なお，サブ母集団数 M ，総個体数 N ，サブ母集団あたりの個体数を $q = N/M$ とする．

Step 1 世代 t における各サブ母集団の交叉，突然変異，評価，選択，の一連の操作後，各サブ母集団の持つ非劣解集合の数の和 S^t を求める．

Step 2 もし， $S^t > N$ ならば Step 3 へ．そうでなければ，全体シェアリングの操作を終了し，各サブ母集団での探索へ戻る ($t = t + 1$)．

Step 3 各サブ母集団の個体 (P_i^t) を，1つの母集団 (P^t) に統合する ($P^t = \cup_{i=1}^M P_i^t$)．

Step 4 P に対して全体シェアリング選択を行い，新たな P として N 個の個体を選択する．

Step 5 各サブ母集団に対して P から q 個ずつ個体を分配する． $t = t + 1$ を行い，サブ母集団ごとの探索へ戻る．

上記の Step 4 における全体シェアリング選択を以下に示す．

Step TS1 Fonseca らによるランク付け¹⁹⁾をもとに全個体 P の適合度を設定する²． P の中から非劣個体のみを選択し， P' とする．もし，選択した劣個体の個体数 $|P'|$ が $N > |P'|$ ならば，非劣個体以外をルーレット選択より $N - |P'|$ 個非復元抽出し， P' に加え， P' を P にコピーして終了する． $N < |P'|$ ならば，Step TS2 へ．

Step TS2 P' から重複している個体を削減する． $N > |P'|$ ならば， P の非劣個体以外をルーレット選択より $N - N_s$ 個非復元抽出し P' に加え， P' を P にコピーして終了する． $N < |P'|$ ならば，Step TS3 へ．

Step TS3 3.3.3 節において用いられていたシェアリングを使用し，適合度の再割当てを行う．新たな適合度をもとにルーレット選択により N 個を非復元抽出し， P とする．

上述のように全体シェアリング選択では，3段階の選択操作から成り立っている．全体シェアリング選択は，サブ母集団ごとの非劣解の総数が N を越えた段階で行われ，探索全体として優れた解を選択し，そうでない解を削減する処理を行っている．全体シェアリング選択を用いた場合の概念図を図 5.4 に示す．

ここで，通常の分割母集団モデルと比較した場合の TSGA の利点を以下に列挙する．

- 探索効率の向上
- サブ母集団間における個体数，探索能力の均一化

²ランク分の 1 の値を適合度として設定する．

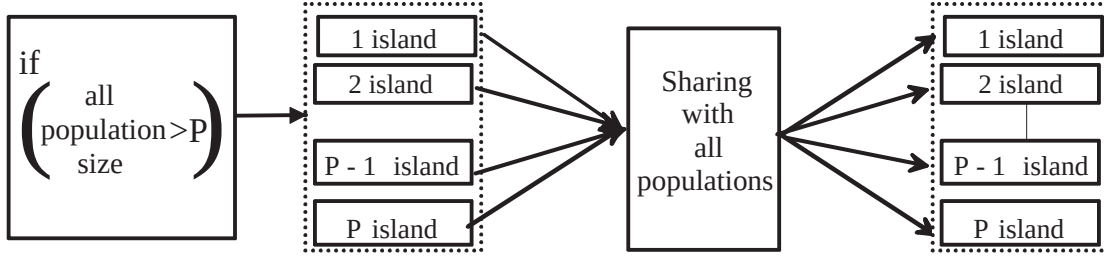


図 5.4 Schematic of the Total Sharing procedure

- 移住の必要性減少

上記に共通した事項として，各サブ母集団が独立して探索を行っているにも関わらず，各サブ母集団の個体は単一母集団として探索している場合と非常に近い選択が行われている．すなわち，全体シェアリング選択を行うことにより分割母集団モデルにおける問題点であった探索の全体的な視野を各個体に与えることが期待でき，より効率的な探索が実現されるものと考えられる．

5.3.2 数値実験

ここで，単一母集団 GA (Single population GA: SGA)，分散 GA (Distributed GA: DGA)^{27, 28)} と全体シェアリング GA (TSGA) の数値結果の考察を行う．

例題

対象例題としては，以下に示すような単純な関数による 2 目的の多目的最適化問題を用いた．これらの例題は，玉置らの使用した関数³⁴⁾であり，いずれも最小化問題である．なお，例題の問題名 **T2**，および **T4** は玉置らの使用した関数番号に対応している．

T2

$$\mathbf{T2} : \begin{cases} \min f_1(x) = x_1^2 - x_2 \\ \min f_2(x) = -\frac{1}{2}x_1 - x_2 - 1 \\ s.t. \\ g_1(x) = \frac{1}{2}x_1 + x_2 - \frac{13}{2} \leq 0 \\ g_2(x) = \frac{1}{2}x_1 + x_2 - \frac{13}{2} \leq 0 \\ g_3(x) = \frac{1}{2}x_1 + x_2 - \frac{13}{2} \leq 0 \\ g_4(x) = x_1 \geq 0 \\ g_5(x) = x_2 \geq 0 \end{cases} \quad (5.1)$$

T4

表 5.1 GA parameters

	SGA	DGA	TSGA
Terminal generation	250		
Total population size	100		
Crossover rate	1.0		
Mutation rate	0.0		
Number of islands	-	5	
Migration interval (sort interval)	-	5	
Migration rate	-	0.1	-

$$\mathbf{T4} : \begin{cases} \min f_1(x) = -2x_1 + x_2 \\ \min f_2(x) = x_2 \\ s.t. \\ g_1(x) = x_1^2 - x_2 \leq 0 \\ g_2(x) = x_1 \geq 0 \\ g_3(x) = x_2 - 1 \leq 0 \end{cases} \quad (5.2)$$

アルゴリズムの構成

本研究において用いるアルゴリズムは、分散 GA を基本としている。今回実装したアルゴリズムの特徴を以下に示す。

- 遺伝子としてビット列ではなく、実数値ベクトルを用いている。
- 交叉方法として、重心を用いた正規分布交叉³⁵⁾を採用している。
- 突然変異を用いない。

GA のパラメータ設定

数値実験には、SGA、DGA と TSGA の 3 種類の GA を用いている。使用したパラメータを表 5.1 に示す。

5.3.3 実験結果および考察

$\mathbf{T2}$ 、および $\mathbf{T4}$ に対する各手法の非劣解集合を図 5.5、図 5.6 に示す。なお、図 5.5、図 5.6 における各手法のプロットは、10 試行の試行ごとに得られた全ての非劣解集合を示し

ている．また，パレート最適フロントに対する得られた非劣解の誤差と被覆率を図 5.7，図 5.8 に示す．

T2

T2は，比較的パレート最適解の求めやすい 2 目的の問題である．図 5.5 から，どの手法においても良好な近似パレート最適フロントが得られているのが分かる．また，図 5.8 から，得られた非劣解の幅広さを表す被覆率においても，どの手法も良好な値を示しているのが分かる．

しかし図 5.7 から，精度面において，DGA が SGA および TSGA の手法に比べて大きく劣っているのが分かる．また，わずかながら TSGA の方が SGA よりも良好な結果を得ているのが分かる．

T4

T4は，**T2**と比較するとパレート最適解を求めるのが困難な問題である．図 5.6 から，DGA における非劣解集合だけがパレート最適フロントの端部分が求まっていない様子が分かる．これは，被覆率を示す図 5.8 において DGA が他手法に比べ，値が悪いことから推測することができる．また，図 5.7 において先ほどの **T2**と同様，DGA が SGA およ

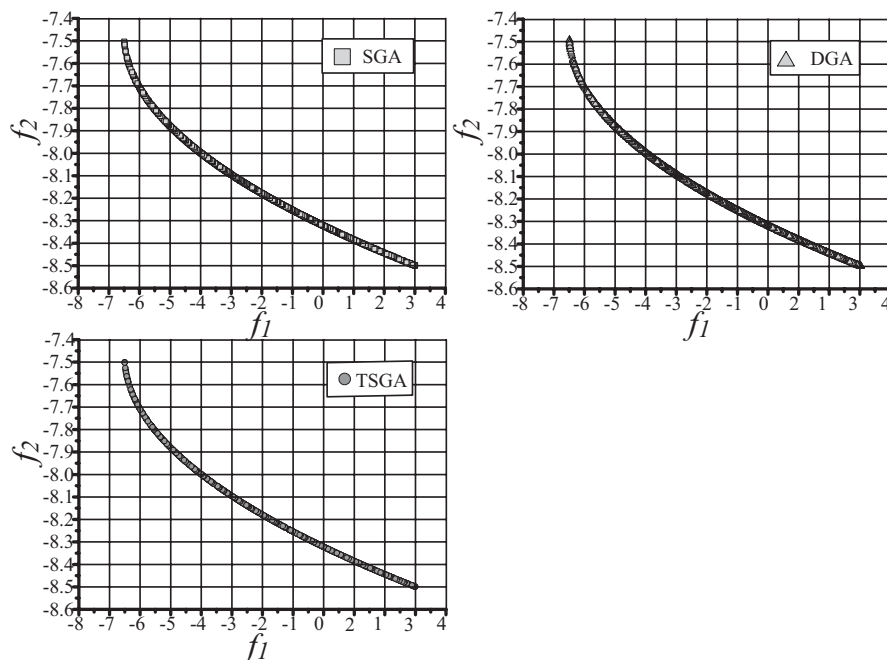


図 5.5 Pareto optimum individuals (**T2**)

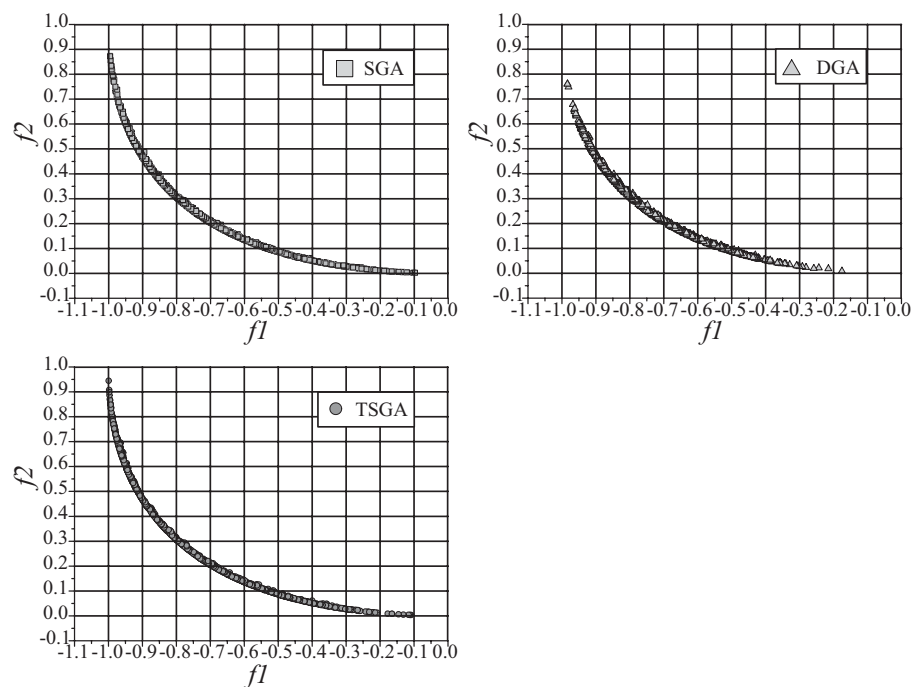
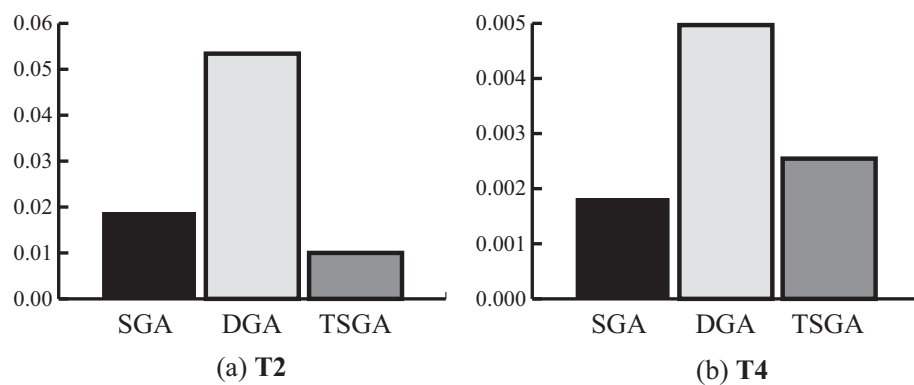
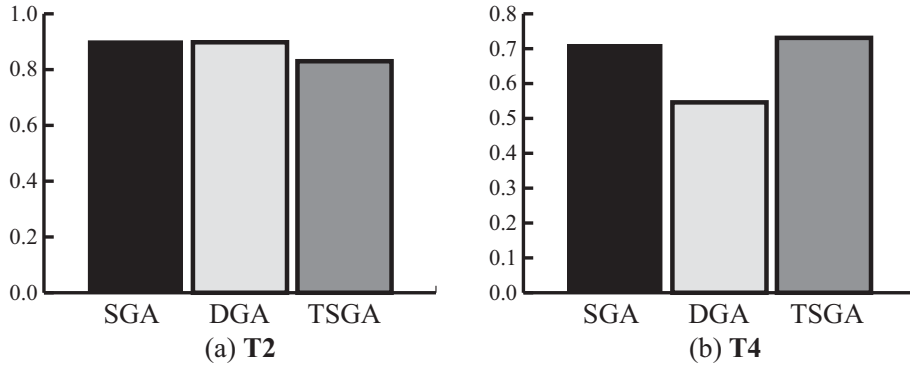


図 5.6 Pareto optimum individuals (T4)

図 5.7 I_{error} of T2 and T4

図 5.8 I_{cover} of T2 and T4

び TSGA の手法に比べて大きく劣っているのが分かる。

結果の考察

DGA は、各サブ母集団間の (移住以外の) 情報交換をせずに探索を行う。そのため、サブ母集団ごとのパレート最適フロントに対する探索箇所が重なり、パレート最適フロント全体を一様に探索することが難しいと思われる。特に、少ない個体数を用いてパレート最適フロントの探索が難しい問題を探索した場合、パレート最適フロント全体へ個体が広がりにくくなるため、その影響が大きく出る。そのため、簡単な問題では DGA の被覆率は他手法と変わらなかったが、T2 に比べ難易度の高い T4 では、被覆率、解の広がりにおいて他手法に劣っていた。

TSGA では、サブ母集団あたりの個体数は DGA と同様少ないものの、非定期的に全個体を用いた選択を行うため、各サブ母集団の探索箇所が重なる危険性が減少し、パレート最適フロント全体を一様に探索することができたものと思われる。

一方、サブ母集団ないの探索個体数はパレート最適フロントに対する精度にも影響する。本実験におけるシェアリングは、非劣個体の数が一定数を越えた段階においてシェアリングを用いた選択を行い、一定数以上の非劣個体の削減を行っている。その際、シェアリングによりパレート最適フロントの一部に固まっていた非劣個体が削減され、削減後の非劣解集合はパレート最適フロントに対して一様に分布する。そのため、サブ母集団における探索個体数が少ない場合、シェアリング後の個体のパレート最適フロントに対する分布は、探索個体数が多い場合に比べ疎となる。

しかしながら、パレート最適フロントに対する精度を求めるためには、ある程度個体が近傍に存在する必要がある。これは、個体間の距離が離れている状態での交叉よりも距離のより近い状態での交叉の方がより良い子個体を生成しやすいからである。そのため、SGA

に比べパレート最適フロントに対する個体の分布が疎となる DGA では、得られる精度が SGA よりも劣ってしまう。

この点において、TSGA では母集団全体を用いたシェアリングしか行わないため、DGA に比べパレート最適フロントに対する個体の分布が疎にならず、SGA とほぼ同程度の精度を求めることができる。

5.3.4 まとめ

TSGA では、母集団内の非劣個体の数が一定数を越えた段階において全個体を用いたシェアリングを行い、非劣個体の削減を行っている。

数値実験より TSGA は、DGA に比べ良好な結果を得ることができた。これは、DGA にはなかった全個体での優越関係の決定、シェアリングを用いた非劣解の削減が分割母集団モデルの探索にとって効果的であったためと考えられる。一方、SGA と DGA を比較した結果、SGA の方が 2 つの例題でともに勝っていたことから、多目的 GA では単純に母集団を分割するだけでは探索性能が低下することを確認することができた。

5.4 領域分割型多目的遺伝的アルゴリズム

前節の結果から、単一目的 GA の分割母集団モデルを単純に適用した場合、多目的 GA としての性能が劣化することが確認できた。その 1 つの解決策として提案した、非定期的に全ての個体を用いて選択操作を行う TSGA は単一目的 GA の分割母集団モデルに比べ良好な結果を示した。しかしながら、TSGA では非定期的にしか全体としての選択操作を行わない上、広範囲のパレート最適フロントを探索する多目的の特性が全く考慮されていない。

そこで、本研究ではより多目的の特性に特化した分割母集団モデルとして領域分割型多目的遺伝的アルゴリズム (Divided Range Multi-Objective Genetic Algorithm: DRMOGA) を提案する。本モデルは、サブ母集団ごとに探索する目的関数空間の領域を定めることにより、探索の重なりを減らし、全体として効率的な探索を実現しようとするモデルである。

5.4.1 領域分割型多目的 GA のアルゴリズム

以下に提案する DRMOGA の流れを説明する。ここで、GA の総個体数を N 、分割数を K とし、目的関数は f_1 から f_M まで M 個存在するものとする。

Step 1 N 個の個体をランダムに生成する。これらの個体が表現する設計変数は全ての制約条件を満足するものとする。

Step 2 得られた解のうち、非劣解のみを選択する。

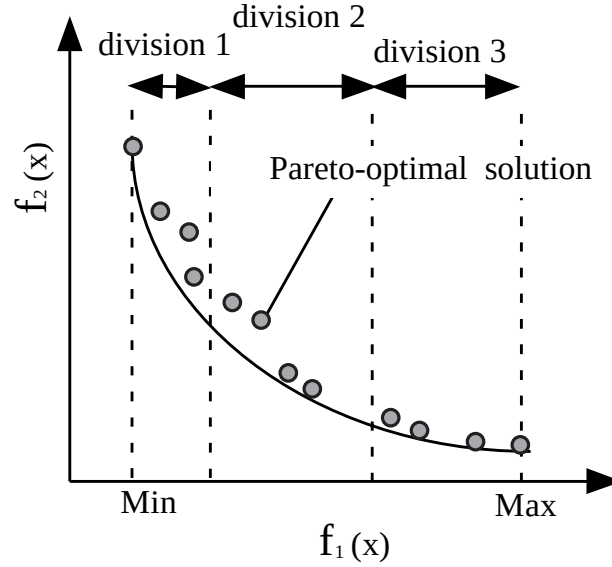


図 5.9 Schematic of DRMOGA

Step 3 基準となる目的関数 f_i の値に従って各個体のソートを行う．本研究では，基準とする目的関数 f_i はランダムではなく f_1 から f_M まで順に変更することとしている．また，基準とする目的関数の最大値から目的関数値の順に N/K 個ずつ個体をサブ母集団へ分配し， N/K 個の個体からなる K 個のサブ母集団を形成する．

Step 4 サブ母集団ごとに多目的 GA を行う．本研究で行う多目的 GA は次節で詳しく説明する．また，世代ごとに終了判定を行い，条件を満たす場合には終了する．終了判定で，条件を満たさない場合は，Step 5 に進む．

Step 5 各母集団で多目的最適化 GA が s 世代行われたら Step 3 に戻る．この世代数をソート間隔と呼ぶ．

本研究では，分割数 K およびソート間隔 s はあらかじめ決定しておくものとする．図 5.9 には 2 目的の場合 ($M = 2$) に目的関数 f_1 に沿って 3 分割 ($K = 3$) している概念図を示す．なお，図 5.9 は最小化の場合の例である．

この DRMOGA に対して分散メモリ型並列計算機を用いて並列処理を行うことにより，DGA と同等の処理速度の向上や SGA と同等，もしくはそれ以上の解の精度を持った解集合を求めることが期待できる．

5.4.2 数値実験

DRMOGA の有効性を検証するため、DGA, SGA との比較実験を行う。なお、これら全ての多目的 GA は Fonseca らの MOGA¹⁹⁾ である。

例題

DRMOGA の有効性を検証するために **NDP**, **VB3** の2つの例題に対して実験を行う。

NDP

$$\mathbf{NDP} : \begin{cases} \min f_1 = -x_1 \\ \min f_2 = -x_2 \\ \vdots \\ \min f_{n-1} = x_{n-1} \\ \min f_n = x_n \\ s.t. \\ g_j = -x_j (j = 1, 2, \dots, n) \\ g_{n+k} = x_k - 6 (k = 1, 2, \dots, n) \\ g_{2n+1} = 1 - x_1 \times x_2 \times \dots \times x_n \end{cases} \quad (5.3)$$

式 (5.3) に表される **NDP** は、非常に単純な関数である。この関数の最大の特徴は、 n 次元への拡張が容易に行え、問題の難易度を設定することができることである。また、式 (5.3) の制約式 g_{2n+1} がパレート最適フロントを形成するため、得られた非劣解の誤差を求めることができる。

VB3

$$\mathbf{VB3} : \begin{cases} \min f_1(x) = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2) \\ \min f_2(x) = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15 \\ \min f_3(x) = \frac{1}{x_1^2 + x_2^2 + 1} - 1.1 \exp(-x_1^2 - x_2^2) \\ s.t. \\ g_1(x) = x_1 \geq -3 \\ g_2(x) = x_2 \leq 3 \end{cases} \quad (5.4)$$

VB3 は式 (5.4) で表され、Veldhuizen らの論文で用いられていた 3 目的 2 変数の最小化問題である³⁶⁾。この例題の最大の特徴は、3 つ目の目的関数式にある。3 つ目の目的関数式における最小値は、 $(x_1, x_2) = (0.0, 0.0)$ のとき、 $f_3(x) = -0.1$ であるが、設計変数の値が非常に大きくなった場合（例えば、 $(x_1, x_2) = (\infty, -\infty)$ ）にも、最小値に近似の $f_3(x) = 0.0$ という値を得る。そのため、初期段階、および探索の中盤段階でこのような設計変数の値

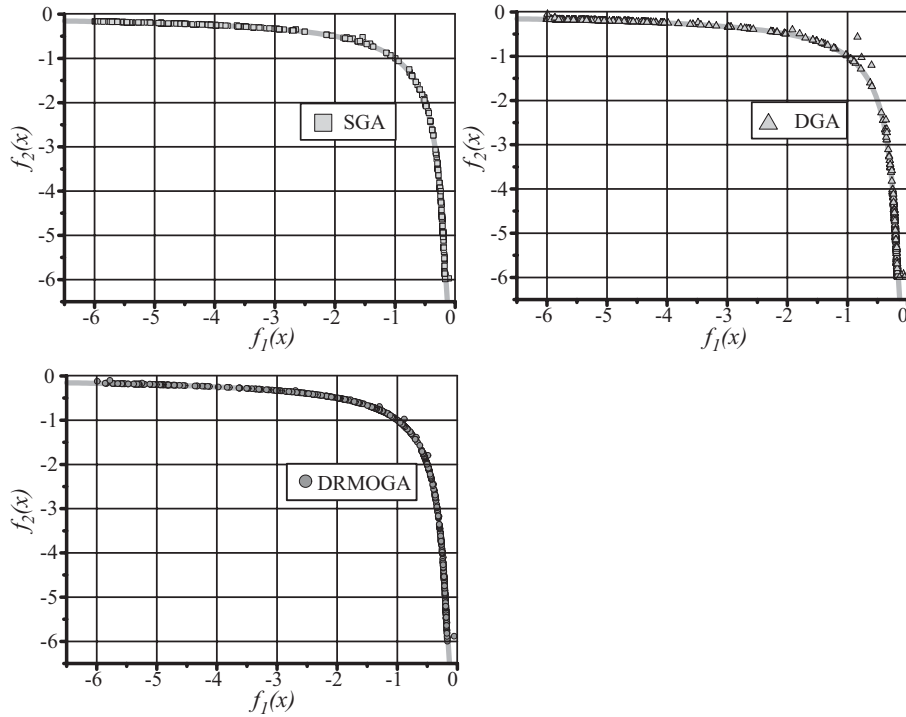


図 5.10 Pareto optimum individuals(NDP)

ばかりに個体が優越された場合，最終的に非劣解は広がらず，パレート最適解に近似した値を得ることもできない。

さらにもう 1つの問題として，パレート最適解全体が偏っていることが挙げられる．目的関数 2 ($f_2(x)$) は， $f_2(x) \in [15, 17.5]$ までの値をとるにも関わらずパレート最適解全体としては $f_3(x) = 15$ 付近に集中するような形態を持っているのである．そのため $f_2(x) \in [15, 17.5]$ の区間のパレート最適解を得ることが困難なのである．

なお，用いたパラメータは先ほどと同様の表 5.1 を使用した．なお，DRMOGA におけるソート間隔 s は，表 5.1 における移住間隔と同値を用いた．

結果

得られた結果として，10 試行の試行ごとに得られた全ての非劣解集合のプロットを図 5.10，図 5.11 に示す．**VB3** は特に $f_2(x)$ 軸に関する探索が難しいという特徴を持つため， $f_2(x)$ を横軸， f_3 を縦軸にとったプロット図を図 5.12 に示す．また，**NDP** における誤差を図 5.13 に³，**NDP** および **VB3** の被覆率を図 5.14 に示す．

³**VB3** のパレート最適解は未知であるので，得られた非劣解とパレート最適解との誤差は評価できない。

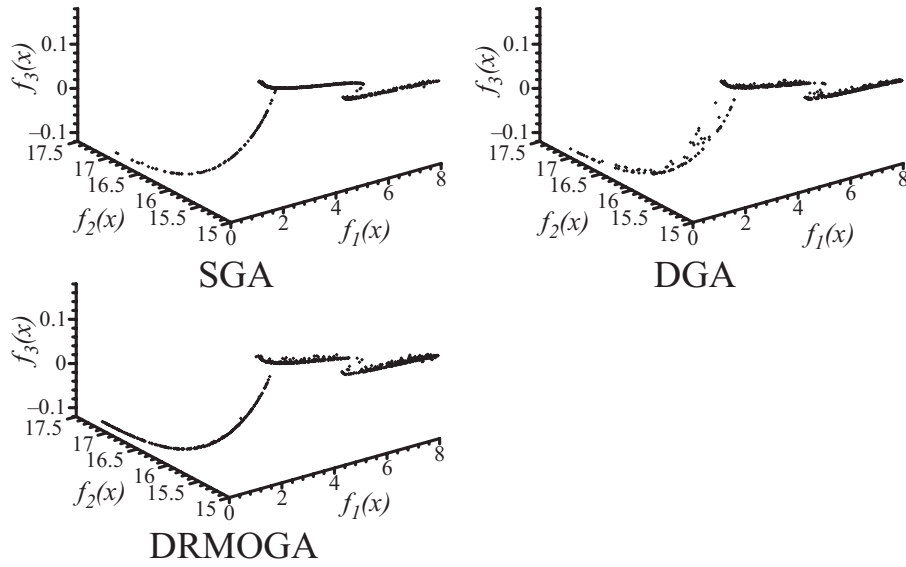
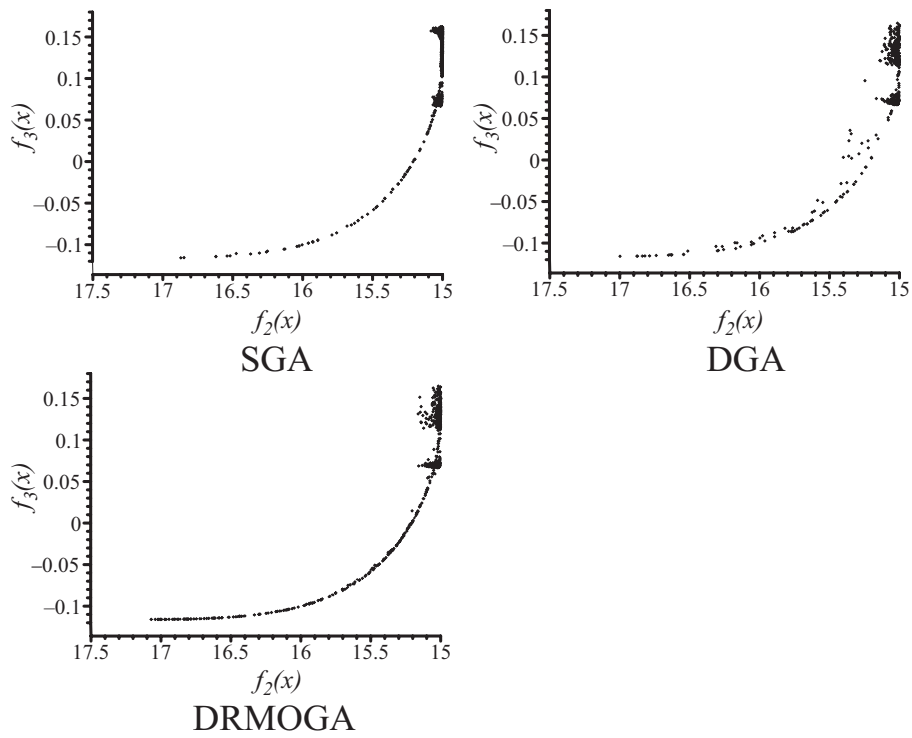
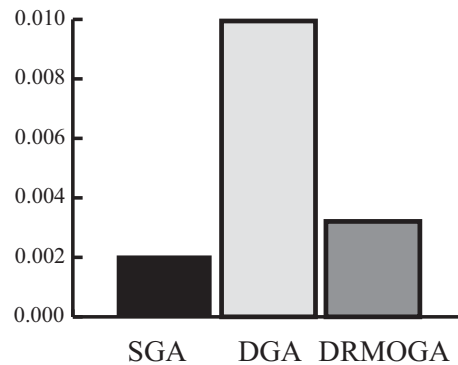
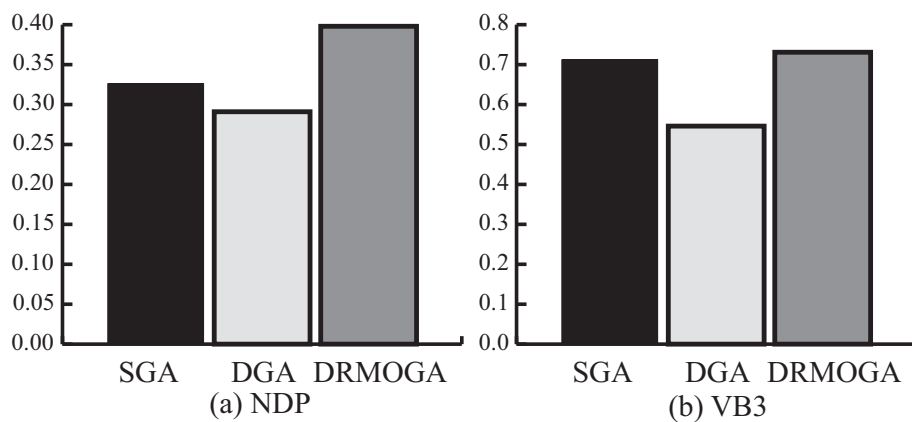


図 5.11 Pareto optimum individuals(VB3)

図 5.12 Pareto optimum individuals ($f_2 - f_3$, VB3)

図 5.13 I_{error} of NDP図 5.14 I_{cover} of NDP and VB3

NDP

図 5.10 のプロットから，SGA，DRMOGA に比べ DGA の近似パレート最適フロントに隙間が多いことが分かる．一方，被覆率を示す図 5.14 では，どの手法の値においても 4 割を切っており，全ての手法において一度の試行ではパレート最適フロントの十分な範囲を探索できていないことが分かる．しかしながら，DRMOGA は SGA や DGA と比較して最も良好な結果が得られている．このことから，DRMOGA は他の手法と比較してよりパレート最適フロントの広い範囲の探索が行えているといえる．

また，図 5.13 の得られた非劣解の精度について見た場合，DRMOGA は SGA ほどではないもののかなり良い精度で解探索が行えているのが分かる．対して，DGA は他の 2 手法と比較して大きく解の質が劣っている．

VB3

式 (5.4) に示される例題は、パレート最適解集合を求めるのが困難な 3 目的の問題である。3 次元の目的関数空間にも関わらずパレート最適フロントは、曲面ではなく曲線に近い形状を持っている。本例題は $f_2(x) = 15$ から $f_2(x) = 17$ にパレート最適解集合が分布するような問題であり、この部分の探索が非常に困難である。

図 5.11 から、DRMOGA, SGA により得られた非劣解集合の方が DGA により得られたものに比べ幅広いことが分かる。このことは、被覆率を示す図 5.14 から読みとることができる。

また、 $f_2 - f_3$ 平面での分布を表す図 5.12 から SGA と比較しても DRMOGA は、一様な幅広い非劣解集合が得られている様子が分かる。SGA および DGA が $f_2(x) = 15$ 付近に個体が固まっているのに対して、DRMOGA はより一様な個体分布を示している。

結果の考察

2 つの例題ともに、DRMOGA はパレート最適フロント全体へ探索が広がっていた。特に、問題の特性として求めることが難しい **VB3** の $f_2 - f_3$ 平面平面に関して最も良好な近似パレート最適フロントを得ることができた。

これは、DRMOGA では目的関数空間の領域ごとに母集団を分割し探索を進めているため、探索個体全体として局所的な収束が起こりにくいと考えられる。SGA などの全個体を用いた選択を行う場合には、個体全体が一部の良好な個体に影響を受け、パレート最適フロントの一部に探索個体が偏ってしまう危険性がある。これに対して、DRMOGA では母集団が個体の目的関数空間での分布をもとにサブ母集団ごとに完全に分割されるためこのような危険性がほとんどない。

一方、DGA を多目的に適用した場合、母集団はサブ母集団ごとに完全に分割されているものの、個体の分布が異なっているとは限らないため、全てのサブ母集団においてパレート最適フロントの一部に個体が偏る可能性がある。さらに、パレート最適フロントの特定の一部付近における探索が他の部分に比べて容易な問題の場合、単一母集団の場合よりもさらに探索個体がパレート最適フロントの一部に集中しやすくなる。そのため、DGA は SGA よりも得られた非劣解集合の分布領域が狭くなりやすい。

また、解の精度についても DRMOGA は DGA と比較して良好な結果を得ることができた。5.3 節で述べたように DGA では、シェアリングおよび探索個体数の影響により SGA ほどの精度を求めることが難しい。対して、DRMOGA は目的関数空間の領域ごとにサブ母集団を分割し探索を行っているため、サブ母集団ないの個体の数が SGA に比べ少ないものの、探索する領域もサブ母集団の数の分だけ狭くなる。また、シェアリングの影響に

についても同様であり、探索する領域自体がサブ母集団数の分だけ狭くなるためパレート最適フロントに対する分布は DGA ほど疎になることはない。さらに、DRMOGA では母集団内の個体どうしが目的関数空間において近いため近傍どうしによる交叉を実現しており、探索効率を向上させている。これらの複合的な効果により、DRMOGA では DGA に比べ精度の良い個体が得られていると思われる。

5.4.3 まとめ

本節では多目的 GA を並列処理するための分割母集団モデルとして、得られる非劣個体群を領域によって分割するモデルである、領域分割型多目的 GA (Divided Range Multi-Objective Genetic Algorithm: DRMOGA) を提案した。DRMOGA では、個体の目的関数値をもとにサブ母集団への分配が行われるため、サブ母集団ごとの探索が、パレート最適フロントに対して重なり合わない。また、母集団内の個体どうしがパレート最適フロントのほぼ同じ部分の探索を行っているため、より効果的な交叉を実現することができる。すなわち、DRMOGA では通常の分散モデルに比べ、より効率的な探索を実現することができる。

数値実験により、以下のことが明らかになった。

1. 多目的 GA では、多くの場合、単一目的の場合とは異なり、DGA よりも SGA によって良好な解が得られるが、DRMOGA によって得られる解はそれと同等以上である。
2. パレート最適解全体が求めにくい問題に対して DRMOGA は特に有効である。

5.5 分散協力型スキーム

TSGA および DRMOGA は、多目的 GA を分割母集団に適用する際の問題点を解決するために考案されたアルゴリズムである。すなわち、これらのアルゴリズムは母集団を分割して探索を行うことにより生じる多目的特有の問題に対処することに主眼が置かれたアルゴリズムといえる。

対して、ここで提案する分散協力型スキーム (Distributed Cooperation Scheme: DC-Scheme) は、より広範囲の近似パレート最適フロントを得ることに主眼が置かれた多目的 GA の枠組みである。

多目的最適化において広範囲な探索領域の中から、パレート最適フロントと同等の広がりを持った非劣解集合を得ることは非常に重要である。しかしながら、各目的間のバランスをとりながら探索を進める多目的 GA では、必ずしも各目的関数における最適値、すなわちパレート最適フロントの端の部分まで探索が進むとは限らない。そのため、何らかの

幅広い近似パレート最適フロントを得るための仕組みが必要である。この1つの方法として、明示的にパレート最適フロントの端部分を探索する仕組みを探索アルゴリズムに組み込む方法が考えられる。

DC-Scheme では、明示的に各目的関数における最適解を探索する仕組みとして、多目的 GA とは別に、各目的の単一目的最適化を行う単一目的 GA (Single Objective Genetic Algorithm: SOGA) を探索に組み込む方法を用いている。そのため、DC-Scheme では目的関数の数が M の問題において、各目的ごとに単一目的最適化を行う SOGA が M 個、多目的最適化を行う多目的 GA (Multi Objective Genetic Algorithm: MOGA) が 1 個の計 $M+1$ 個の探索母集団が必要となる。

また、DC-Scheme はアルゴリズムそのものではなく、多目的最適化に対して GA を適用する際の枠組みである。そのため、SOGA および MOGA のアルゴリズムには、基本的にどのような手法でも使用することができる。

以下、DC-Scheme の詳細な仕組みについて説明する。

5.5.1 分散協力型スキームの特徴

本節では、多目的最適化において各目的関数の最適解の探索と近似パレート最適フロントの前進を同時に行う、DC-Scheme の提案を行う。

DC-Scheme は、代表的な多目的 GA と単一目的 GA を組み合わせ、それらが協調して探索を行う。DC-Scheme では、多目的最適化におけるパレート最適解の探索を MOGA で、各目的関数における最適解の探索を SOGA で行っている。さらに、それぞれで得られている最良解を一定期間ごとに情報交換し、探索個体数を動的に調整することで協調探索を実現している。これらの仕組みにより、MOGA 単独では実現できない幅広い範囲の解探索の実現を期待できる。DC-Scheme の目標としては、単独の MOGA と同じ計算量で、より広範囲に分布した非劣解集合を得ることである。

分散モデル ($M+1$ 個体群)

DC-Scheme では、MOGA と SOGA の両方を用いて探索を行うため、複数の探索個体群が必要となる。

単一目的 GA は各目的関数の最適解の探索を行うため、目的関数が M 個であれば、 M 個の探索個体群が必要となる。さらに、SOGA 個体群とは別に多目的 GA でパレート最適解の探索を行う MOGA 個体群を用いる。すなわち M 目的関数の場合、 $M+1$ 個の個体群が必要となり、 M 個の SOGA 個体群と 1 個の MOGA 個体群が存在することになる。

協調探索

DC-Scheme では、多目的 GA と単一目的 GA を用いて探索を行っている。しかし、それぞれが個々に探索するのではなく、協力することにより高い探索能力の実現を目指している。そのため、最良解の交換と各個体数の調整という 2 つの方法により協調探索を実現している。

最良解の交換（移住）

各個体群で解探索を行うが、ある一定期間ごとに最良解の交換（移住）を行っている。これは各個体群で得られている最良解の情報を共有することでより少ない計算での探索が期待できるためである。

動的な個体数の調整

解交換を実行する際、交換する最良解を比較し、その結果をもとに各個体群の個体数の調整を動的に行っている。これにより、SOGA 個体群と MOGA 個体群の探索の進み具合に偏りが生じた場合、その差を減少させる効果を期待できる。

協調探索は次のように実行される。

目的関数 $F_i(x)$ $i = 1, 2 \dots M$ の場合、MOGA 個体群には各目的関数における最も良い値を持つ個体が M 個存在する。ここではこれらを B_{M_i} とする。それに対し、 N 個の SOGA 個体群にはそれぞれ決められた目的関数に関する最良解が存在し、それらを B_{S_i} とする。

一定期間ごとに MOGA 個体群の B_{M_i} と各 SOGA 個体群の B_{S_i} を交換する。その際、それらの個体を比較し、 $B_{M_i} > B_{S_i}$ であれば、MOGA 個体群の数個体を i 番目の SOGA 個体群に移動させる。一方、 $B_{S_i} > B_{M_i}$ であれば、逆に SOGA 個体群の個体を MOGA 個体群に移動させる。ただし、各個体群の個体数を増減させるが、総個体数は一定である。

パレートアーカイブ

近年、代表的な多目的 GA の手法にはパレートアーカイブを用いるものが多い^{6,8,9)}。これはパレートアーカイブを導入する事により、探索個体とは別に一度得た非劣解集合を保存することができるためである。

提案する DC-Scheme においてもパレートアーカイブを導入し、各個体群で得られた非劣解集合を探索個体とは別に保存している。

5.5.2 アルゴリズム

N 目的の多目的最適化問題の場合、DC-Scheme は次のように探索を進める。

Step 1 総個体数分の個体をランダムに発生させる。

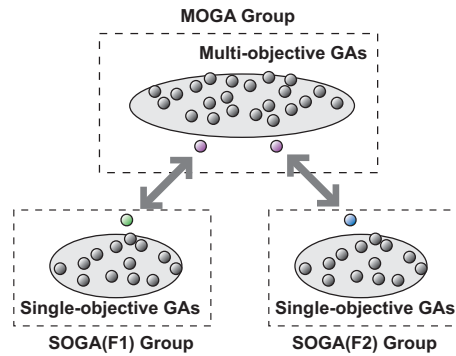


図 5.15 Schematic of DC-Scheme

Step 2 全個体を MOGA 個体群と各目的関数 F_i の最適解を探索する M 個の SOGA 個体群に分割する.

Step 3 各個体群が設定されている GA に従って解探索を行う.

Step 4 各個体群でエリートアーカイブ, パレートアーカイブの更新を行う.

Step 5 一定期間ごとに各個体群の間で解交換を行う.

- MOGA 個体群は各目的関数におけるその時点で最も良い解（最良解） B_{M_i} を目的関数 F_i の探索を行う各 SOGA 個体群に送信する.
- 目的関数 F_i の探索を行う各 SOGA 個体群はグループ内の最良解 B_{S_i} を MOGA 個体群に送信する.

Step 6 各個体群の最良解である B_{M_i} と B_{S_i} を比較する.

$B_{S_i} > B_{M_i}$ SOGA 個体群の数個体を MOGA 個体群に移動させ, MOGA 個体群の探索個体数を増加させる.

$B_{M_i} \geq B_{S_i}$ 上記を逆の操作を行う. つまり, MOGA 個体群内から数個体を SOGA 個体群内に追加する.

Step 7 終了条件に満たない場合は, 3. に戻り, 解探索を続ける.

DC-Scheme の 2 目的の場合の概念図を図 5.15 に示す.

5.5.3 数値実験

数値実験より提案手法である DC-Scheme の有効性を検証する. DC-Scheme の多目的 GA 部分に Fonseca らの提案したランキング⁵⁾を用いた MOGA, Zitzler らの SPEA2⁹⁾, Deb らの NSGA-II⁸⁾を組み込み, 各手法単独で行った場合と DC-Scheme を組み込んだ場合の比較を行う. なお, 本実験で組み込んだ MOGA は, 非劣解の保存 (アーカイブ個体群の利用) を組み込んだエリート主義にもとづく MOGA であり, Fonseca らの提案した MOGA⁵⁾とは厳密には異なる.

また, 単一目的 GA 部分には単一目的 GA においては非常に性能が優れているといわれる DGA^{27, 28)}を用いた. なお, 実験では表 5.1 のパラメータを用いた.

対象問題

本実験では, Zitzler, Deb らの数値実験において使用された **ZDT4**⁴⁰⁾ および Kursawe の数値実験により使用された **KUR**³⁷⁾を用いた.

ZDT4

$$\text{ZDT4} : \begin{cases} \min f_1(x) = x_1 \\ \min f_2(x) = g(x) \left[1 - \sqrt{\frac{x_1}{g(x)}} \right] \\ s.t. \\ g(x) = 91 + \sum_{i=2}^{10} [x_i^2 - 10 \cos(4\pi x_i)] \\ x_1 \in [0, 1], x_i \in [-5, 5], i = 2, \dots, 10 \end{cases} \quad (5.5)$$

この問題は, 10 設計変数よりなる 2 目的の多峰性を有する問題である. この問題は, x_1 以外の設計変数値のとり範囲が広いためパレート最適解 $x_i = 0.0 (i = 2, \dots, 10)$ を見つけだすことが難しいという特徴を持っている. また, 局所的な収束域が多数存在するため, 探索能力の違いが得られた解の精度へに反映されやすい問題である.

KUR

$$\text{KUR} : \begin{cases} \min f_1 = \sum_{i=1}^n (-10 \exp(-0.2 \sqrt{x_i^2 + x_{i+1}^2})) \\ \min f_2 = \sum_{i=1}^n (|x_i|^{0.8} + 5 \sin(x_i)^3) \\ s.t. \\ x_i \in [-5, 5], i = 1, \dots, n, n = 100 \end{cases} \quad (5.6)$$

この問題は, $f_1(x)$ において連続する 2 変数間の相互作用を持ち, $f_2(x)$ において多峰性を有する問題である. また本実験では, この問題を 100 個の設計変数を持つ問題として扱っ

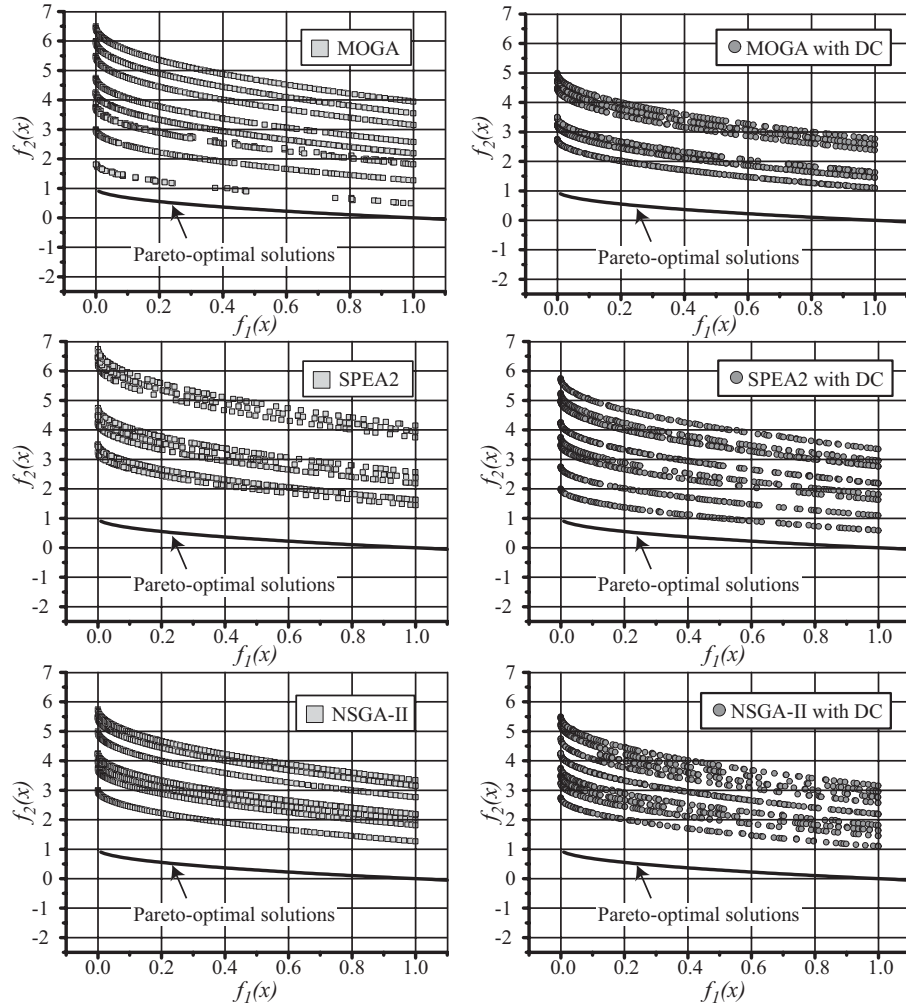


図 5.16 Pareto optimum individuals (ZDT4)

た. そのため, 上述の **ZDT4** の問題と比べパレート最適解の探索には膨大な計算が必要となり, より探索が困難な問題となっている.

5.5.4 結果

得られた結果のうち, **ZDT4**, **KUR** の非劣解集合のプロットを図 5.16, 図 5.17 に示す. また, **ZDT4** の誤差を図 5.18 に, **ZDT4** と **KUR** の被覆率を図 5.19 に示す⁴.

⁴KUR のパレート最適解は未知であるので, 得られた非劣解のパレート最適解との誤差は評価できない.

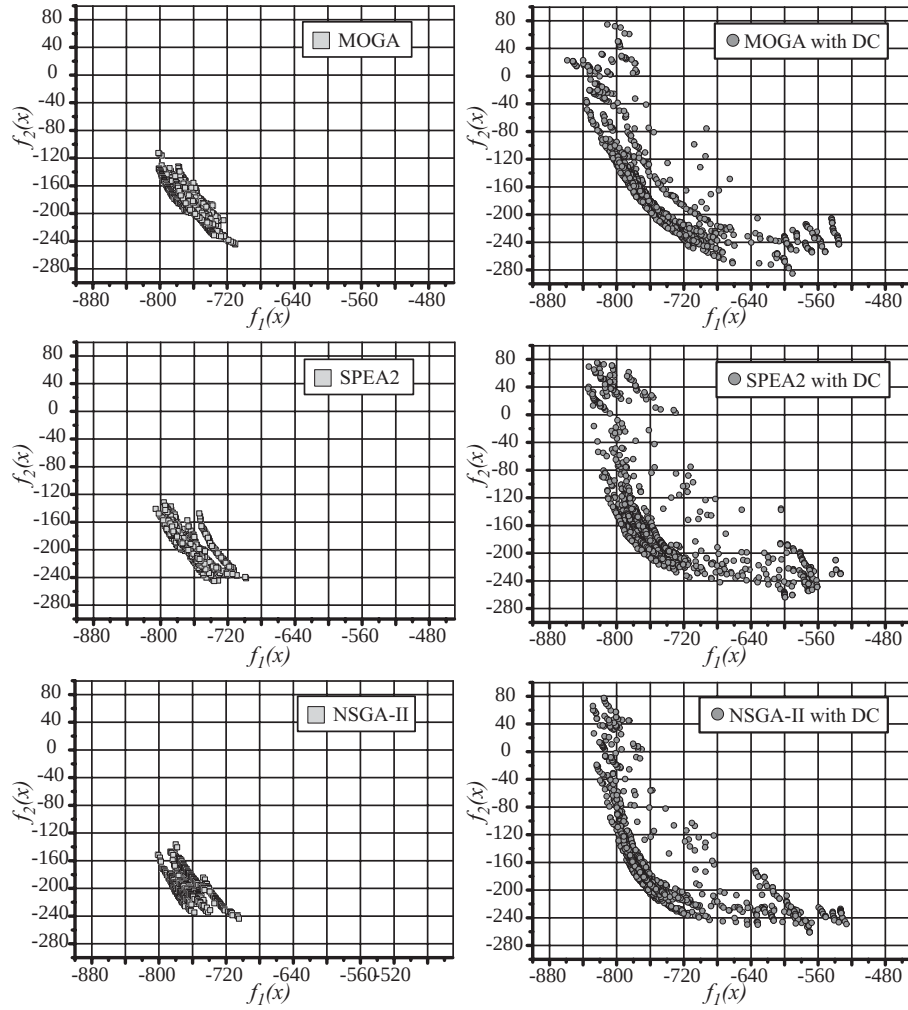
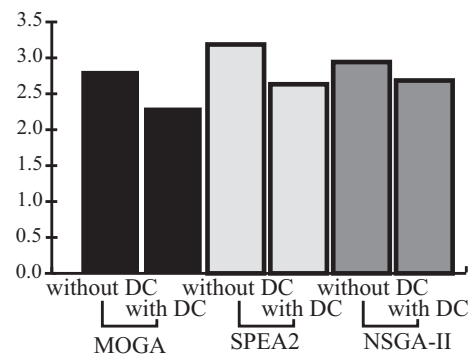
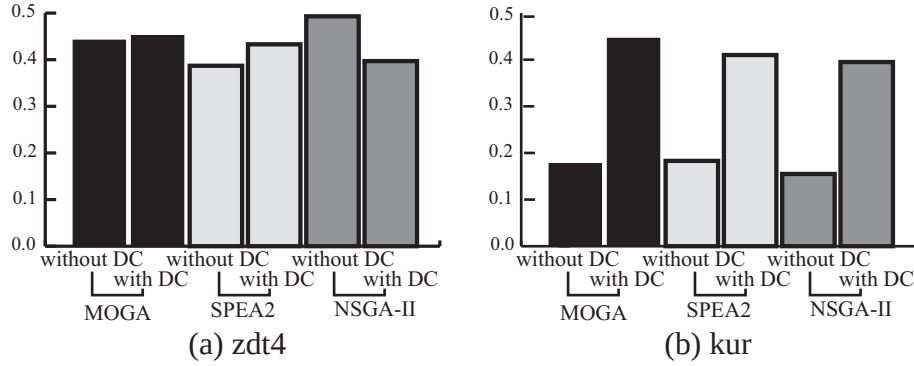


図 5.17 Pareto optimum individuals (KUR)

図 5.18 I_{error} of ZDT4

図 5.19 I_{cover} of **ZDT4** and **KUR**

ZDT4

ZDT4は、 f_1 が x_1 と等しい関係にあるため DC-Scheme にとってあまり有利な問題ではない。これは、DC-Scheme の f_1 に関する単一目的最小化が完全に無駄となり、探索効率が減少するためである。しかも、本例題では f_1 軸において容易に広範囲な非劣解を得ることができるため DC-Scheme の利点の 1 つである広範囲に分布する効果を発揮することができない。そのため、図 5.19 から分かるように解の幅広さを表す被覆率では DC-Scheme の効果は確認することができない。

しかしながら、図 5.18 では DC-Scheme を組み込んだ手法の方が、各手法単独の場合よりもパレート最適解に対する近さ、精度の点において優れているのが分かる。これは、局所解が多数存在する **ZDT4**において、各サブ母集団が独立に探索を進める DC-Scheme が効果的に影響したためと考えられる。

KUR

本例題は、設計変数の数が 100 と多いためパレート最適フロントに到達しにくい探索の困難な問題である。また、広範囲な解を得るためには多様な母集団を保持しながら探索を行う必要があるため比較的 DC-Scheme の利点を活かしやすい問題であるといえる。

得られた結果のうち、図 5.17 および図 5.19 から DC-Scheme が非常に効果的に探索に影響しているのが分かる。DC-Scheme を組み込んだ手法は、非常に広範囲に分布しているのに対して、各手法単独の場合にはいずれの手法においてもパレート最適フロントの一部の部分に解集合が固まっているのが分かる。

結果の考察

本実験結果より、DC-Schemeを組み込むことにより、**KUR**のような広範囲の非劣解集合を得にくい問題に対して、効果的な探索を実現することが確認できた。特に、DC-Schemeを用いない場合にはパレート最適フロントの中央付近に非劣解集合が偏ってしまうのに対して、DC-Schemeを用いた場合には広範囲に分布し偏りのない非劣解集合を求めることができた。

DC-Schemeでは、単一目的GAで得た各目的関数の最良解を活用することにより、多目的GAにおいて広範囲の探索が実現できる。また、**ZDT4**のような多峰性のある問題に対して効果的な探索を実現していることから、DC-Schemeは単に広範囲の探索を実現するだけでなく、探索の局所的収束を防ぐ効果もあることが分かった。

5.5.5 まとめ

本研究では、より広範囲の非劣解集合を得るための新たな枠組みとして分散協力型スキーム (Distributed Cooperation Scheme: DC-Scheme) の提案を行った。DC-Schemeは、目的ごとの単一目的最適化を行う単一目的GAと多目的GAを用いて多目的最適化を行う枠組みである。そのため、多目的GAにDC-Schemeの枠組みを用いた場合、多目的GA単独で探索を行う場合に比べ、より多様な非劣解集合の探索が可能となる。

提案するDC-SchemeをMOGA⁵⁾、SPEA2⁹⁾、NSGA-II⁸⁾の3つの多目的GAに組み込み、各手法を単独で用いた場合との比較を行った。その結果、多目的GAにDC-Schemeの枠組みを用いることにより、より広範囲の多様な非劣解集合を得ることができた。

5.6 まとめ

本章では、分割母集団モデルを多目的GAへ適用するための新たなアルゴリズム、枠組みとして以下の3つの手法を提案した。

- 全体シェアリングGA (Total Sharing Genetic Algorithm :TSGA)
- 領域分割型GA (Divided Range Multi-Objective Genetic Algorithm :DRMOGA)
- 分散協力型スキーム (Distributed Cooperation Scheme :DC-Scheme)

上記のうち、TSGAおよびDRMOGAは多目的GAを分割母集団に適用する際の問題点を解決するために考案されたアルゴリズムである。すなわち、多目的GAを分割母集団に適用する際の問題点である以下の事柄に対応したアルゴリズムである。

- 1) 各サブ母集団の探索領域の重なりによる探索の無駄

2) サブ母集団あたりの個体数減少に伴う探索能力の低下

TSGA では、上記の内、重なりによる探索の無駄を非定期的に全ての個体を用いて選択操作を行うことにより、その影響を小さくするという方法を用いた。全体シェアリング選択という、全ての個体を用いて選択操作を行うという非常に単純な操作を加えることにより、DGA などの分割母集団モデルよりも良好な結果を得ることができた。

また、DRMOGA は目的関数空間の領域ごとにサブ母集団を分割し探索を進めるという方法を用いることにより、上記の両方の問題点について対応していた。DRMOGA では、分布する目的関数空間の位置によって個体をサブ母集団に分配されるため、各サブ母集団の探索の重なりがほとんど起こらない。また、個体数の減少にともなう探索能力の劣化についても、探索するパレート最適フロントの面積が減少するため、この影響を受けることはない。

その他、DRMOGA では大きく 2 つの特徴を実現していた。1 つは、サブ母集団ごとに異なるパレート最適フロントを探索することにより、母集団全体の多様性が維持されやすく、収束が起こりにくいという点であり、もう 1 つは、近傍どうしの個体が交叉することによる探索効率の向上である。この近傍どうしの個体による交叉は、目的関数空間の広域に個体が分布する多目的 GA 特有の問題点を解決する 1 つの有効な手法であり、分割母集団モデルだけでなく全ての多目的 GA へ応用できる特徴である。例題を用いた数値実験により、DRMOGA の分割母集団モデルに対する有効性を確認することができた。

一方、DC-Scheme は目的ごとの単一目的最適化を行う単一目的 GA を任意の多目的 GA に組み込む枠組みであり、より広範囲の近似パレート最適フロントを求めることを目的としている。目的ごとに単一目的 GA での探索を行っているために、パレート最適フロントの端付近を確実に得ることができる。また、得られた端付近の個体を多目的 GA 探索に活用することにより、多目的 GA のより広範囲の探索を実現することができる。数値実験の結果より、より広域のパレート最適フロントを探索することができた。

第 6 章

近傍培養型遺伝的アルゴリズム

6.1 はじめに

3章で述べた通り、近年の数多くの多目的 GA の手法に関する研究が行われ良好な結果が得られている。その最も代表的なものとしては、Deb らの NSGA-II⁸⁾、Zitzler らの SPEA2⁹⁾、Erickson らの NPGA2⁷⁾、Fonseca らの MOGA⁵⁾ などがあげられる。これらのアルゴリズムでは、探索途中で発見した優良解の保存、適切な適合度割り当て、適切なパレート解候補の削減など探索において重要なスキームが共通して実現されている。

一方、本研究では 5.4 節において分割母集団モデルの多目的 GA として DRMOGA を提案し、個体を目的関数空間での位置情報によってサブ母集団に分配する方法が多目的最適化において効果的であることを確認した。この原因の 1 つとして、近傍の個体間での交叉があげられる。大域的な探索を行う多目的 GA では、探索個体群の目的関数空間での距離が大きく離れ、効果的な交叉を行うことができない場合がある。これに対して、明示的に目的関数空間の距離が近い個体間での交叉を行うことにより、探索効率が向上させることができる。本研究では、このような目的関数空間での距離の近い個体間で行われる交叉を近傍交叉と呼ぶ。

しかしながら、これまでの多目的 GA における研究では、上記のような近傍交叉に関する議論は全く行われていない。そこで本研究では、これまでに提案されてきた代表的な多目的 GA から探索に効果的なスキームを考察し、それらの探索に重要と思われるスキームに近傍交叉を組み合わせた新たな多目的 GA のアルゴリズムとして近傍培養型 GA (Neighborhood Cultivation GA: NCGA) の提案を行う。提案手法の有効性を検証するため、幾つかの代表的なテスト問題に対して、Deb の NSGA-II⁸⁾、Zitzler らの提案する SPEA2⁹⁾ との比較実験を行い、提案手法の解探索性能の比較検証を行った。

6.2 多目的遺伝的アルゴリズムにおける重要なスキーム

これまで数多くの多目的 GA の手法に関する研究が行われてきたが、特に 1999 年度以降になり SPEA⁶⁾、NPGA2⁷⁾、NSGA-II⁸⁾、SPEA2⁹⁾ など非常に優れた手法が数多く提案されるようになった。

これは、多目的 GA における複数の探索に重要なメカニズムが明らかとなり、様々な手法でこれらのメカニズムを組み込んだからである。そのため、これらのアルゴリズムでは、適合度の割り当て方法などに違いがあるものの、アルゴリズム全体の仕組みにおいて大きな違いはない。

これらの手法に共通するメカニズムが多目的 GA の探索において重要な効果を持つことは、これらの手法の実験結果より明らかである⁶⁻⁹⁾。共通するメカニズムについてまとめたものを以下に示す。

a) 探索した非劣解の保存

非劣解の保存は、単一目的におけるエリート主義 (elitism) に相当する。多くの手法では、保存対象となる非劣解集合を探索個体群とは別の個体群に保存する^{8,9)}。一般に、この保存する個体群は探索個体群と区別してアーカイブ個体群と呼ばれる。NSGA-II、SPEA2 などの手法では、現世代における非劣解だけでなく任意に設定した個体数分を優良個体として保存する方法を用いている^{8,9)}。これは、後述するようにアーカイブ個体群を次世代の探索個体群として直接活用するためである。

b) 保存している優良解の探索への反映

保存しているアーカイブ個体群を探索へ反映させるためには、アーカイブ個体群を選択へ参加させる必要がある。アーカイブ個体群の選択への参加は、より優れた個体を用いた探索による探索の高速化を目的として行われる。

NSGA-II、SPEA2 などの手法に実装されており、次世代の探索個体を作成するもととしてアーカイブ個体群を利用している^{8,9)}。一般に、アーカイブ個体群に対して選択手法を用いて探索個体群を選び出す選択のことをメイティング選択 (mating selection) と呼ぶ。

c) アーカイブ個体群の削減

アーカイブ個体群の削減は、あらかじめ設定したアーカイブ個体群の個体数を超えて非劣解が存在するときに任意の設定数まで個体を削減することである。問題によって、探索途中の非劣個体が非常に膨大となり、設定していたアーカイブ個体群の個体数を超えて存在する場合がある。この場合には、選択の対象がすべて現世代における非劣解であ

るため個体の近接度合い、密集度合いといった多様性にもとづく選択を行い非劣個体の削減を行う必要がある。

アーカイブ個体群の削減における最大の問題は、解のパレート最適フロントへの収束を妨げる可能性があることである。全ての非劣解は、それまでの探索により得られた最良の1個体であり、探索の最も進んだ個体の1つである。非劣解を削減することにより、その後の探索により、この削減された非劣解よりも劣った解が非劣解として存在する可能性が生じるため、最悪の場合、パレート最適フロントに対してアーカイブ個体群が後退することが考えられる。実際に、これまでの探索により得られた非劣解集合を削減することにより解のパレート最適フロントへの収束の保証は失われる。そのため、アーカイブ個体群の削減は非常に慎重に行われる必要がある。

多様性にもとづく代表的な選択手法として、シェアリングを利用する方法がある²¹⁾。しかし、シェアリングを用いた選択では、シェアリングに関するパラメータが新たに必要となる上、各目的関数軸における最良個体を保存する保証がない。各目的関数軸における最良個体の損失は、探索領域を狭めることにつながり大域的な探索の弊害になる恐れがある。

一方、シェアリングの持つ問題点を考慮した新たなアーカイブ個体群の削減方法として、SPEA2において用いられている端切手法(truncation method)⁹⁾、NSGA-IIにおいて用いられている混雑度計算(crowding distance calculation)⁸⁾を用いた手法などがある。これらの削減方法は、パラメータフリーであり、非劣解集合の各目的軸における最小値、最大値が必ず保存される仕組みになっている。このため、これらの手法では、これまで得られた非劣解集合の各目的関数軸に対する幅広さが失われることはない。つまり、非劣解集合の分布領域は、新たな非劣解によりそれまで端にいた非劣解が優越されない限り、縮小することはない。

d) 探索個体に対する適合度割り当て

前述のように、多目的GAでは「多様性」を維持しながら「精度」の高い解を探索する必要がある。そのため、各個体間の優越関係と目的関数空間での近接度合いを考慮した適合度割り当てを行う必要がある。これまでに提案されてきた代表的な手法として、MOGAで用いられているランキングとシェアリングを組み合わせた方法、SPEA2において用いられている独自の適合度割り当て方法、NSGA-IIにおいて用いられている非優越ソートと混雑度計算を組み合わせた方法などがある⁷⁻⁹⁾。例外的にNPGAでは、2個体比較にもとづく独自の選択手法を用いることにより適合度という概念が存在しない²¹⁾。

適合度割り当てにおいて重要なことは、非劣個体とそうでない個体が存在する場合、必ず非劣個体の方がより良好な適合度値を割り当てられなければならないことである。これは、非劣個体よりも高い適合度値を持つ非劣でない個体が存在した場合、非劣個体よりも非劣個体以外の個体の方が選択されることになり、母集団のパレート最適フロントに対する探索が後退するためである。一般に、このようにパレート最適フロントに対する探索が後退する可能性がある場合、パレート最適フロントに対する（母集団の）収束性が失われていると言う。逆に、個体群が探索によってパレート最適フロントから後退する可能性がない場合、パレート最適フロントに対する収束性がある言う。

この条件を満たした適合度割り当てとしては、非優越ソートとシェアリングを組み合わせた NSGA の方法、非優越ソートと混雑度ソートを組み合わせた NSGA-II の方法、SPEA2 において用いられている適合度割り当て方法がある。

e) 各目的スケールの正規化

個体の各目的関数値によるスケールを正規化しなければ、目的関数空間において測定する個体間の距離などに弊害が生じる恐れがある。これは、扱う目的関数のスケールがほぼ同等の場合には全く問題とならないが、各目的関数のとりうる値のスケールが異なる場合には重要となってくる。

この問題に対する有効な方法として、NPGA2 において用いられているスケーリング手法がある⁷⁾。このスケーリングでは、個体群の持つ各目的関数値の最大値と最小値を用いて、各評価終了後（もしくは適合度割り当て前）に各個体の持つ目的関数値をもとに以下の式に従いスケーリングを行っている。

$$O'_i = \frac{O_i - O_{i,\min}}{O_{i,\max} - O_{i,\min}} \quad (6.1)$$

なお、式 (6.1) における O_i は、目的関数 i における値を意味しており、 O'_i は目的関数値 O_i のスケーリング後の値を意味する。また、 $O_{i,\max}$ および $O_{i,\min}$ は現個体群中における目的関数 i の最大値、最小値の値を意味している。

6.3 近傍培養型遺伝的アルゴリズムの概要

本章で提案するアルゴリズム、近傍培養型遺伝的アルゴリズム (Neighborhood Cultivation GA: NCGA) では 6.2 節において説明した探索に重要な効果をもたらすメカニズムがすべて考慮されている。これらのメカニズムに目的関数空間における近傍どうしの個体を用いて交叉する近傍交叉のメカニズムを取り入れたものが NCGA である。

以下に、本モデルの流れを示す。

Step1

初期個体を生成する．世代 $t = 1$ とする．各個体の評価を行い，これらの初期個体群をアーカイブ個体群 (A_t) とする．

Step2

アーカイブ個体群 (A_t) を探索個体群 (P_t) にコピーし， P_t を 1 つの目的関数値を基準にソートし並び替える．この際，基準となる目的関数 $f_j(x)$ は，世代 t を目的関数の数 M で割った余りから求める ($t \equiv j \pmod{M}$)．また，Step 3, 4 において用いる変数 i を 0 に初期化する．

Step3

Step2 によりソートされた探索個体群 (P_t) に対して i 番目と $i + 1$ 番目の隣り合った 2 個体をペア個体群として選択する．

Step4

選択された 2 個体のペア個体群を用いて交叉，突然変異，評価を行い，新たなペア個体群を生成する．得られた新たなペア個体群を Step3 により選択した 2 個体と入れ替える．変数 i に 2 を加え， i が個体数と等価になるまで Step3, Step4 を繰り返す．この結果，探索個体群がすべて更新される (P_{t+1})．

Step5

探索個体群 (P_{t+1}) とアーカイブ個体群 (A_t) との比較を行い，アーカイブ個体群を更新する (A_{t+1})．この際，アーカイブ更新の方法として SPEA2 における環境選択 (environmental selection) ¹ の手法を用いる．

また，SPEA2, NSGA-II などと同様，アーカイブ個体群はあらかじめ設定した個体数 N 個分の個体を優良個体として保存している．

Step6

終了条件を満たすかどうか判定を行う．終了条件を満たせば終了，満たさない場合には，世代 $t = t + 1$ を行い，Step2 へ戻る．

このように提案する NCGA は，探索個体群に対してある 1 つの目的関数を基準としたソートを行い，隣り合う 2 つの個体をペア個体群として選択し，遺伝的操作を行っている．ソートされた個体群における隣り合う個体どうしは，比較的，目的関数空間における個体間の距離が近いので，隣り合う 2 つの個体を用いて交叉を行うことにより近傍交叉を実現することができる．

ただし，ここでのソートを常にある定まった目的関数を基準に行うと繰り返し同じペア

¹一般に，新たに得られた探索個体群を用いて保存しているアーカイブ個体群を更新する場合，探索個体群とアーカイブ個体群の和集合から選択操作を用いて新たなアーカイブ個体群を生成する．この選択操作のことを環境選択と呼ぶ．

どうしでの交叉になる恐れがあるため以下のような方法を用いている.

- ・ 毎世代毎に基準となる目的関数を変化
- ・ ソート後の探索個体群に対して母集団サイズの 1 割程度の幅において近傍シャッフルを実行

ソートの基準となる目的関数 $f_i(x)$ は, 世代 t を目的関数の数 M で割った余りにより決定する ($t \equiv i \pmod{M}$). そのため, 毎世代ごとに基準が変化し, 交叉を行うペアがより変化しやすくなっている. 上記における近傍シャッフルとは, ある一定の範囲内で個体をランダムに並び替えるものであり, ある目的関数を基準に個体をソートした後に実行される. この近傍シャッフルにより, ソートされた個体群の並びは僅かに乱れたものとなり, 結果, 個体ペアは毎世代ごとに変化することになる. 例えば, 100 個体の個体群が対象である場合, 各個体は幅最大 10 の範囲で乱数を用いたシャッフルが行われる. また, この近傍シャッフルを行うことにより, 各個体間の情報交換も活発化されるものと思われる.

また, パレート個体群 (A_t) の更新には SPEA2 における適合度割り当ておよび端切手法を用いた環境選択の方法を用いている⁹⁾(3.3.9 節参照). SPEA2 における環境選択の大まかな手順を以下に示す.

Step1 探索個体群 (P_{t+1}) とアーカイブ個体群 (A_t) の和集合 R_t を作成する.

Step2 R_t に対して SPEA2 における適合度割り当てを行う (3.3.9 節 参照).

Step3 割り当てられた適合度と端切手法を用いて R_t からアーカイブ個体群の個体数分の個体を選択し, 新たなアーカイブ個体群 A_{t+1} を生成する.

一方で, SPEA2, NSGA-II などの手法が用いているアーカイブ個体群 (A_t) から探索個体群 (P_t) を選択するメイティング選択 (mating selection)² を行っていない. メイティング選択では, アーカイブ個体群の中から特に優れた個体を重複して選択する. しかし, NCGA では同じ個体が複数存在すると近傍交叉に悪影響を及ぼす可能性があるためこの操作を行っていない. そのため, NCGA ではアーカイブ個体群 (A_t) のコピーを探索個体群 (P_t) として用いている.

6.4 数値実験

本章では, 提案した手法を実際に幾つかの例題へ適用し, 従来手法との比較を通じて提案手法の解探索性能の検討を行う.

提案する NCGA の比較対象アルゴリズムとして Zitzler らによって提案された SPEA2, Deb らによって提案された NSGA-II, 近傍交叉を用いない NCGA(non-NCGA) を用いた.

²アーカイブ個体群から探索個体群を選び出す選択操作をメイティング選択と言う. メイティング選択は, アーカイブ個体群の中でも特に優れた個体を重複して選択し探索個体群を生成することにより, より効率の良い探索を実現している.

non-NCGA とは、近傍交叉を行わずに、ランダムに交叉の親個体を選択する NCGA である。NCGA と non-NCGA を比較することにより、近傍交叉の効果を明確にすることができる。

さらに、NCGA の他のアルゴリズムに対する有効性の検証とは別に、NCGA におけるソート基準についての検討も行った。NCGA は、6.3 節でも説明したように目的関数空間を基準に個体間の近傍の定義を行っている。しかし、個体間の近傍の定義、すなわち NCGA のソート基準としては、設計変数空間を用いた方法も考えられる。そのため、本節では設計変数空間で近傍定義と目的関数空間で近傍を定義した場合について結果の比較および考察を行った。なお、設計変数空間を用いた場合の NCGA では、6.3 節の目的関数空間でのソートと同様に、ある 1 つの設計変数を基準にソートする方法を用いた。

6.4.1 対象問題

数値実験には、異なる特徴を持つ 4 つの連続テスト関数³⁷⁻³⁹⁾ と離散問題として多目的ナップザック問題^{6,9)} を用いた。以下、対象問題について説明する。

連続テスト関数

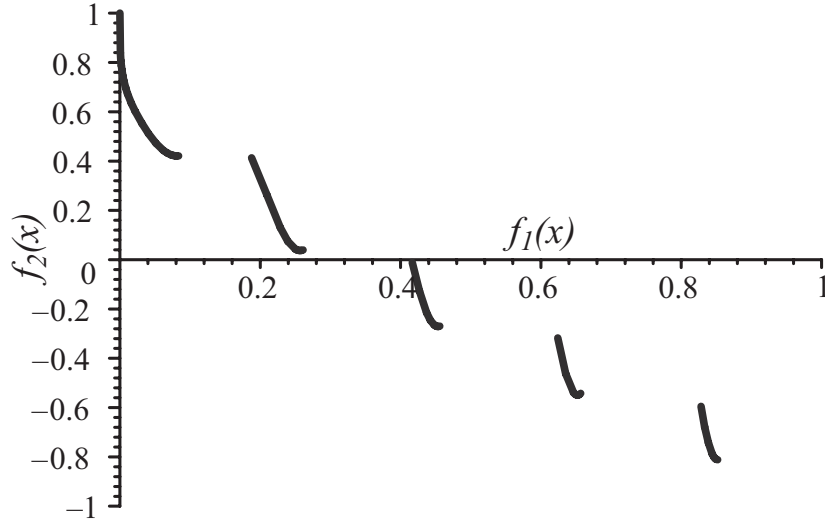
本研究では、幾つかの特徴の異なる連続テスト関数を数値実験に用いた。使用した関数はすべて 2 目的の最小化問題である。本実験では、Deb の考案した非連続パレート問題 ($\mathbf{F}_{\text{discon}}$)³⁹⁾、Zitzler, Deb らの数値実験において使用された **ZDT4** および **ZDT6**⁴⁰⁾、さらに Kursawe の数値実験により使用された **KUR**³⁷⁾ の計 4 つの問題について実験を行った。

以下、非連続パレート問題 ($\mathbf{F}_{\text{discon}}$)、**ZDT6** を示す。**ZDT4** および **KUR** の詳細は、5.5.3 節において既に説明している (式 (5.5), 式 (5.6) 参照)。

非連続パレート問題 ($\mathbf{F}_{\text{discon}}$) :

$$\mathbf{F}_{\text{discon}} : \begin{cases} \min & f_1 & = x_1, \\ \min & f_2(x) & = g(x) \times h(f_1, g) \\ & g(x_2, \dots, x_N) = 1 + 10 \frac{\sum_{i=2}^N x_i}{N-1}, \\ & h(f_1, g) = \left(1 - \left(\frac{f_1}{g} \right)^{0.25} - \frac{f_1}{g} \sin(10\pi f_1) \right) \\ s.t. & & x_i \in [0, 1], \quad i = 1, \dots, N \end{cases} \quad (6.2)$$

全ての変数は $[0,1]$ の値をとりパレート最適解は $x_i = 0$ for $i = 2, \dots, N$ かつ $[0,1]$ の範囲における不連続な x_1 の値である。

図 6.1 Pareto-optimal front of $\mathbf{F}_{\text{discon}}$

この問題のパレート最適解は、次式により表され、図 6.1 に示すように非連続なフロントの形状を持っている。

パレート 最適解：

$$\begin{aligned} f_1(x_1) &= x_1([0, 1]) \\ f_2(x_i) &= 1 - x_1^{0.25} - x_1 \sin(10\pi x_1) \end{aligned}$$

ZDT6

$$\text{ZDT6: } \begin{cases} \min f_1 = 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\ \min f_2 = g(x) \times \left(1 - \left(\frac{f_1}{g}\right)^2\right) \\ s.t. \\ g(x) = 1 + 9 \left(\frac{\sum_{i=2}^N x_i}{N-1}\right)^{0.25} \leq 0 \\ x_i \in [0, 1], i = 1, \dots, 10 \end{cases} \quad (6.3)$$

この問題の特徴は、 $f_1(x)$ と x_1 の間に偏りが存在することである。具体的には、 $f_1(x)$ のある一定範囲の値 $[0.28, 0.65]$ は、 x_1 のある微少な範囲 $([0.05, 0.125])$ の値によって決定される。そのため、この x_1 の微少な範囲 $([0.05, 0.125])$ を詳細に探索しなければ $f_1(x)$ のある一定範囲 $[0.28, 0.65]$ のパレート最適解を一様に求めることができない。

6.4.2 離散問題

本実験では、離散テスト問題として複数種類の多目的ナップザック問題を用いた。この問題は、非常にシンプルで実装しやすい反面、問題自体は探索が非常に困難である。また、前節で説明した連続関数の問題と異なり多目的最大化問題として定式化されている。ここでは、2 目的 750 荷物問題に対する適用を行った。

KP750 – 2

$$\text{KP750 – 2 : } \begin{cases} \max f_i(x) = \sum_{j=1}^{750} x_j \cdot p_{(i,j)} \\ s.t. \\ g_i(x) = \sum_{j=1}^{750} x_j \cdot w_{(i,j)} \leq W_i \\ 1 \leq i \leq k, k = 2 \end{cases} \quad (6.4)$$

上式における $p_{(i,j)}$ および $w_{(i,j)}$ は、それぞれ i 番目の評価値を計算する際の j 番目の荷物に付随する利益値と重み値を表している。また、 W_i は i 番目の評価値計算を行う際の重み値の総和に対する制約値 (上限値) である。

なお、KP750 – 2 に関するパラメータは Zitzler らによる文献^{6,9)} と同じ値を使用した。

6.4.3 GA の構成・GA パラメータ

各個体の表現には、すべての問題においてビットコーディングを用いた。幾つかの文献では、連続関数の問題に対してビットコーディングではなく実数値コーディングを用いて UNDX といった表現型空間での交叉方法を用いた方が、より良好な値を得られやすいことが指摘されている⁴¹⁾。しかし、本研究では各手法の探索能力の差について議論を行うために、ビットコーディングを用いた。同様の理由により、交叉、突然変異に関しても 1 点交叉、ビット反転を用いた。

また、交叉率、突然変異率は Zitzler や Deb の文献^{8,9)} を参考にそれぞれ 1.0、ビット長分の 1 と設定した。各問題におけるビット長は、ナップザック問題に関しては荷物数である 750 ビット、連続関数問題においては 1 変数あたり 20 ビットを用いた。

ナップザック問題において用いた個体数および評価計算回数は、文献⁹⁾ を参考に、250 個体、500000 回とした。一方、連続関数テスト問題に対しては文献⁸⁾ を参考にすべての問題において個体数 100、終了世代数 250 世代とした。

また、評価方法としては 4 節において説明した以下の 5 つの方法を用いた。

- 各目的関数軸ごとの最大値、最小値、平均値を表す I_{MMA} .
- パレート最適解と得られた非劣解との誤差を示す I_{error} .
- 2 つ以上の非劣解集合の比較を行う I_{RNI} , I_{LI} .

6.4.4 各手法比較の結果

本節では、4つの代表的な連続テスト関数と多目的ナップザック問題に対する各手法 (NCGA, SPEA2, NSGA-II, non-NCGA) の比較結果およびその考察を行う。対象とした各問題ごとに以下に示す。

F_{discon}

非連続のパレートフロントを持つこの問題は、**ZDT4**, **ZDT6**と同様、全ての設計変数のうち、 x_1 以外の設計変数値が 0 のとき、パレート最適解が得られるように設計されている。そのため、これらの問題では、 x_1 の多様性を保持しつつ、いかに x_1 以外の値を 0 へ近づけるかがポイントとなる。

一方、後述する **ZDT4**, **KUR** に比べて F_{discon} の探索は容易である。これは、 F_{discon} には多峰性や隣り合う変数の相互作用などが無いためである。

そのため本実験では、この問題を設計変数の数が 10 の場合 ($N = 10$) と 100 の場合 ($N = 100$) の 2つの場合について実験を行った。これは、 $N = 10$ 程度では各手法に大きな差が見られなかったためである。この問題を含め、今回用いた全ての問題では、設計変数の数が多いほど、探索の難易度は高くなる。

以下、それぞれの場合に分けて結果を考察する。

$F_{\text{discon}}(N = 10)$ 設計変数の数が 10 の場合 ($N = 10$) の結果を考察する。得られた結果の内、 I_{cover} と I_{error} の結果を図 6.2 に、 I_{MMA} を図 6.3 に示す。また、各手法で得られた非劣解を比較して求める I_{RNI} および I_{LI} を図 6.4 に示す。得られた非劣解に対するプロット図を図 6.5 に示す。

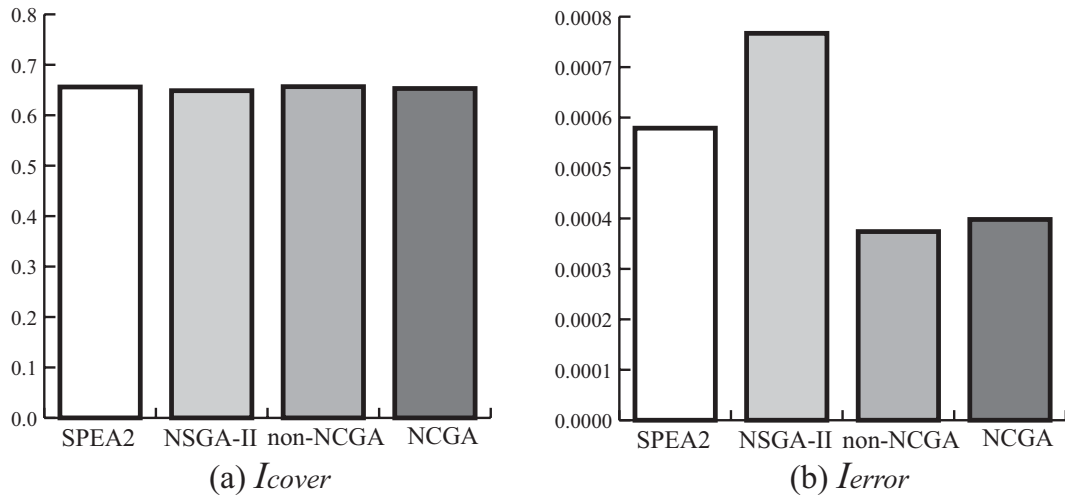


図 6.2 I_{cover} and I_{error} of $F_{\text{discon}}(N = 10)$

この問題は、他の問題と比較して探索が困難ではない。そのため、全ての手法において良好な結果が得られているのが分かる (図 6.2-6.5)。このことは、各手法により得られた非劣解集合のプロット図である図 6.5 において、どの手法もパレート最適フロントに均一な非劣解集合を得ていることが視覚的に読みとることができる。

また、2つの非劣解集合の比較結果である図 6.4 では、non-NCGA, NCGA の値が SPEA2, NSGA-II を勝っているもののその差は僅かであり、ほぼ同程度の解の質であるということが分かる。

$\mathbf{F}_{\text{discon}}(N = 100)$ 設計変数の数が 100 の場合 ($N = 100$) の結果を考察する。得られた結果の内、 I_{cover} , I_{error} の結果を図 6.6 に、 I_{MMA} を図 6.7 に示す。また、各手法で得られた非劣解を比較して求める I_{RNI} および I_{LI} を図 6.8 に示す。得られた非劣解に対するプロット図を図 6.9 に示す。

先ほどの $N = 10$ の場合と異なり、各手法の結果に差が見られる。特に、誤差および非

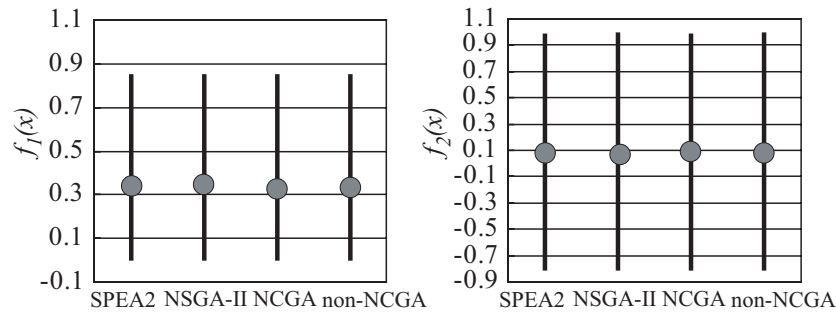


図 6.3 I_{MMA} of $\mathbf{F}_{\text{discon}}(N = 10)$

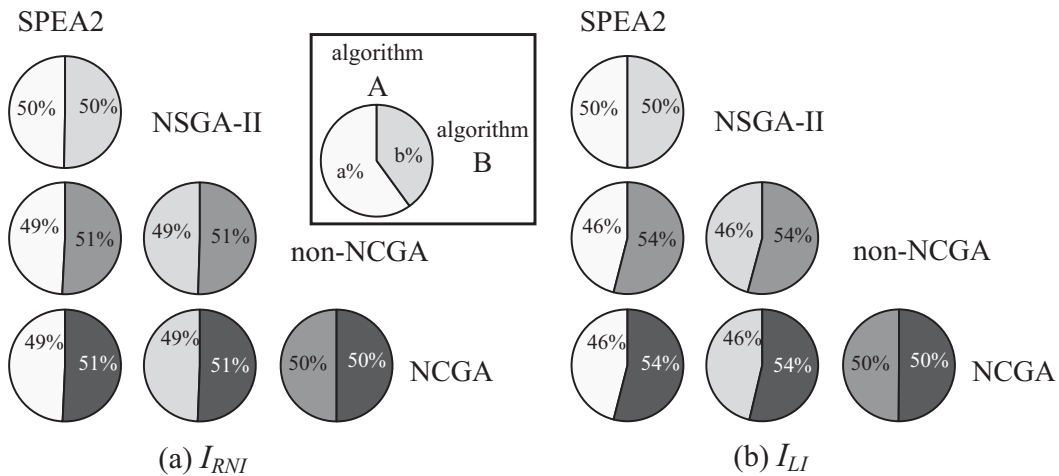


図 6.4 I_{RNI} and I_{LI} of $\mathbf{F}_{\text{discon}}(N = 10)$

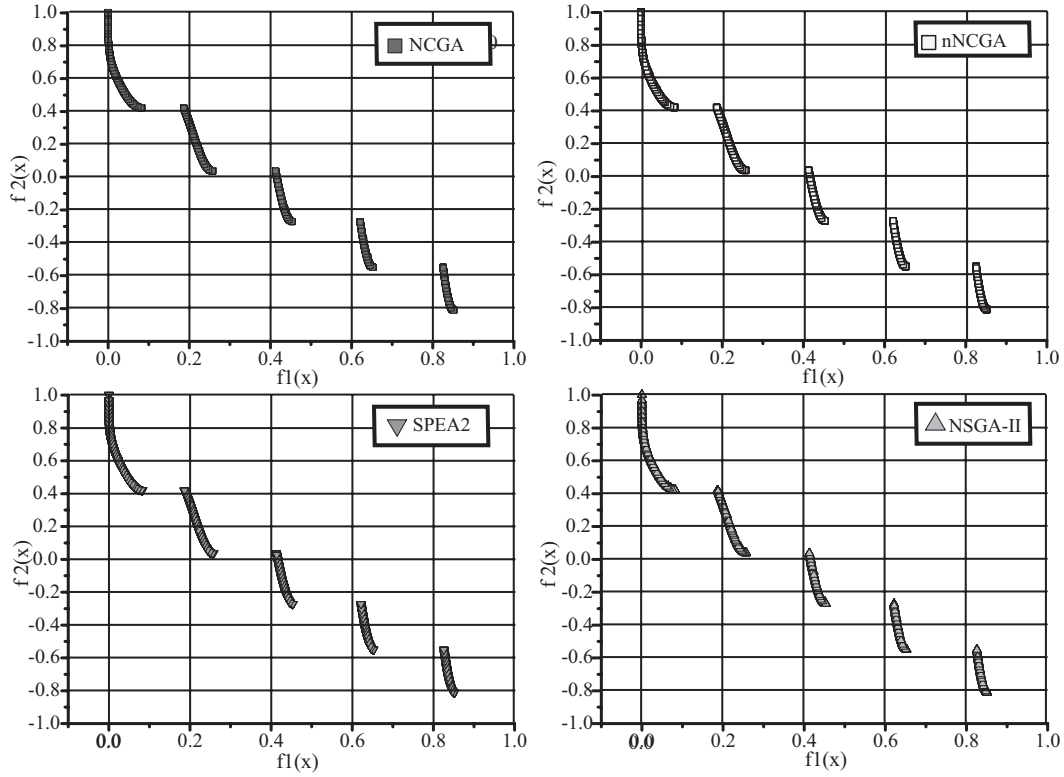


図 6.5 Pareto optimum individuals($\mathbf{F}_{\text{discon}}(N=10)$)

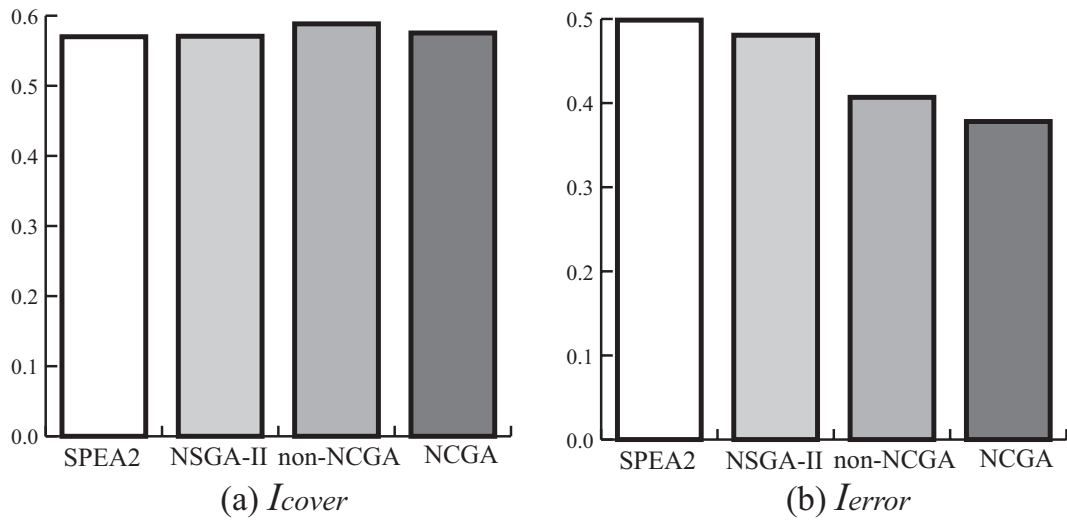


図 6.6 I_{cover} and I_{error} of $\mathbf{F}_{\text{discon}}(N=100)$

劣解集合の比較の結果において差が見られる．図 6.6(b) から，NCGA が他の手法よりも得られた非劣解の誤差が少ないことが分かる．すなわち，NCGA の探索が最もパレート最適フロント付近まで進んでいる．また，NCGA ほどではないものの近傍交叉を用いない non-NCGA においても SPEA2，および NSGA よりも良好な結果が得られている．これは，探索個体を生成するメイティング選択において，より優れた個体を重複して選択する方法を用いているため，探索が局所的に偏りやすいためであると考えられる．

一方，解の幅広さについて見た場合，図 6.6(a) に示される I_{cover} および近似パレート最適フロント図 6.9 から，各手法とも大きな差がみられないことが分かる．これは， x_1 の値により全てのパレート最適フロントが形成されているというこの問題の特徴に起因している．すなわち，この問題では多様な x_1 の値を保持していれば幅広い解を得ることができるのである．この特徴は，上述のように，**ZDT4**，**ZDT6** についてもあてはまる．

また，図 6.9 において各手法の非劣解が，パレート最適フロントに対して垂直方向に幅広い分布を示している．これは，各手法により得られる非劣解の質が試行ごとによって異

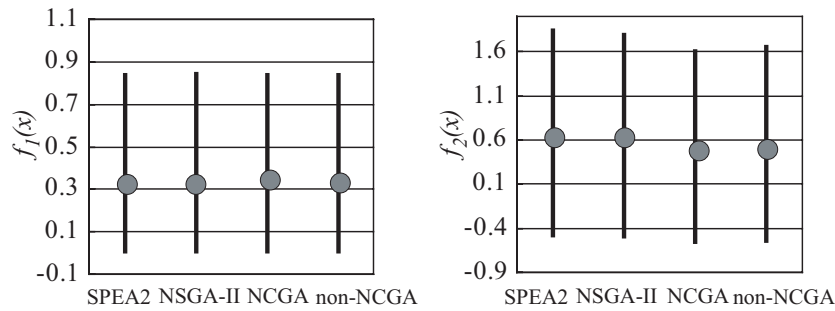


図 6.7 I_{MMA} of $\mathbf{F}_{discon}(N=100)$

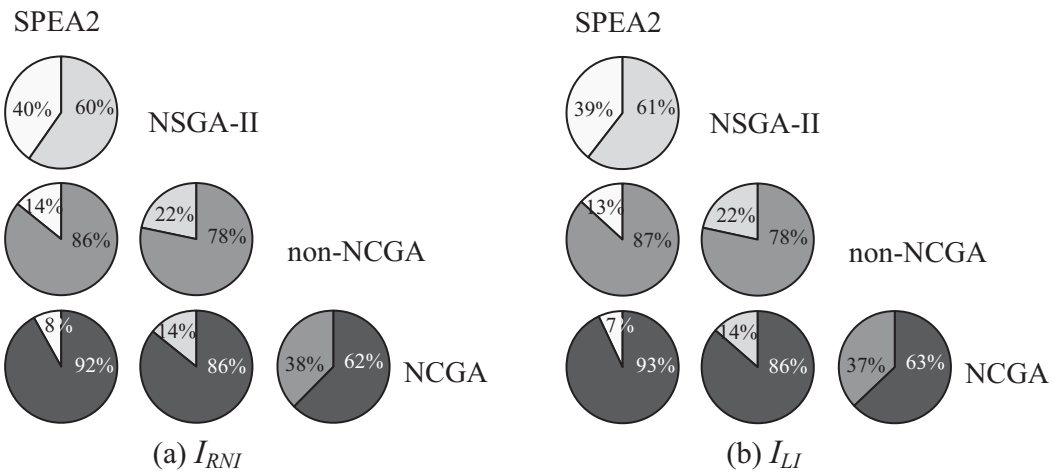


図 6.8 I_{RNI} and I_{LI} of $\mathbf{F}_{discon}(N=100)$

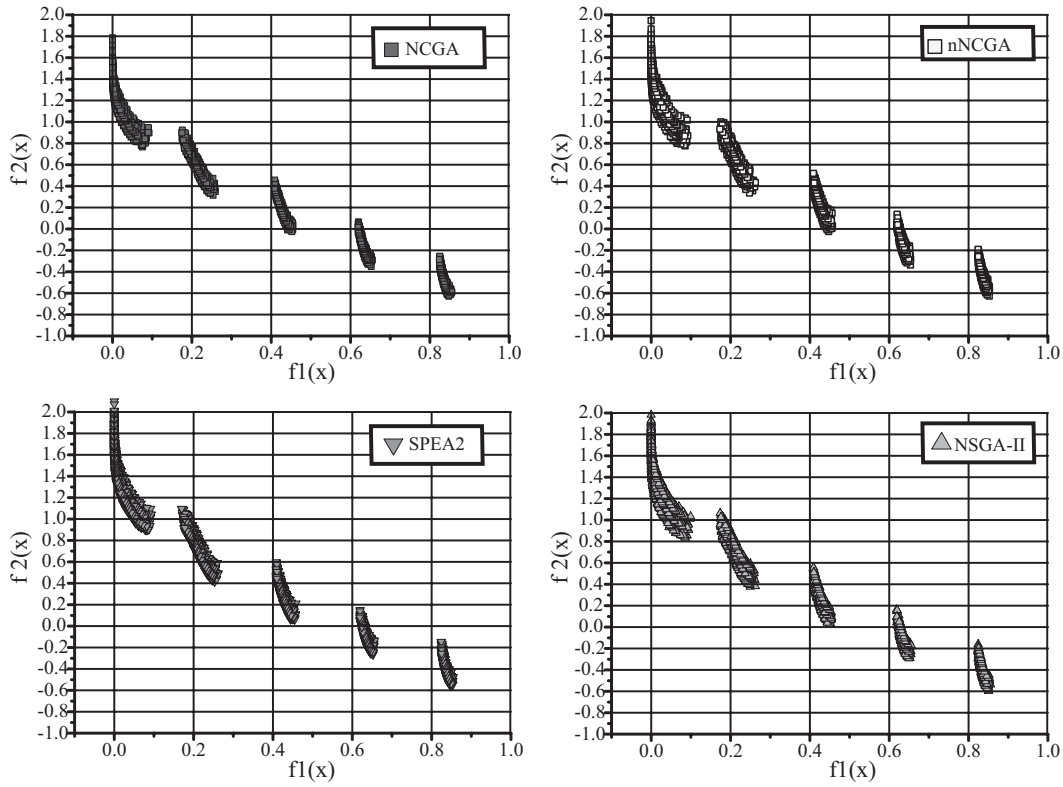


図 6.9 Pareto optimum individuals($\mathbf{F}_{\text{discon}}(N = 100)$)

なっていることを意味している。すなわち、試行によってパレート最適フロントに対する探索の接近度合いが異なっているのである。このことより、多くの手法では 250 世代の段階において探索が収束していないことが分かる。

ZDT4

この問題は多峰性を有しているため、いかに局所解から抜け出し、パレート最適フロントへ探索を進めていくかが問題となる。

得られた結果のうち、 I_{cover} と I_{error} の結果を図 6.10 に、 I_{MMA} を図 6.11 に示す。また、各手法で得られた非劣解を比較して求める I_{RNI} および I_{LI} を図 6.12 に示す。得られた非劣解に対するプロット図を図 ?? に示す。

ZDT4 は多峰性を有する比較的探索の難しい問題である。そのため、いかに局所解から抜け出し、よりパレート最適フロントに近づくことができるかが大きなポイントとなる。

図 6.10(b)、図 6.11 から分かるように、NCGA および non-NCGA は SPEA2, NSGA-II と比較して、よりパレート最適フロントに近い領域を探索している。そのため、手法の比較結果である図 6.12 においても NCGA, non-NCGA は、SPEA2, NSGA-II に対して非常に良好な値を得ている。NCGA, non-NCGA の双方ともに SPEA2, NSGA-II に対して

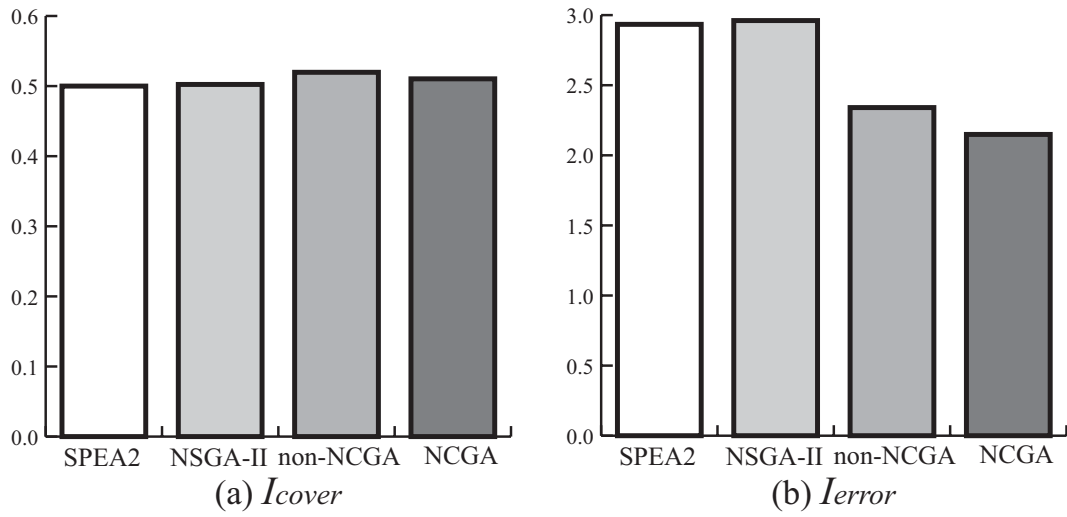


图 6.10 I_{cover} and I_{error} of ZDT4

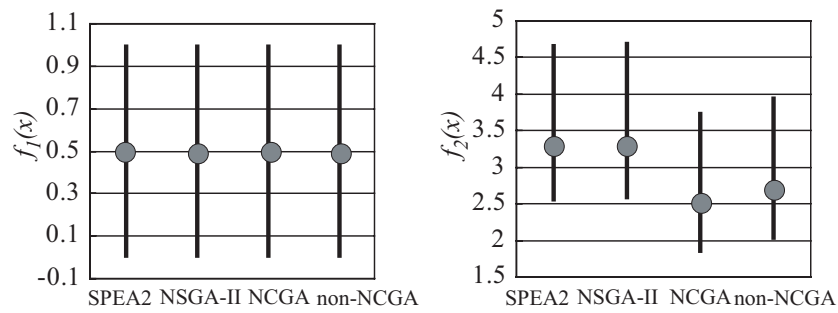


图 6.11 I_{MMA} of ZDT4

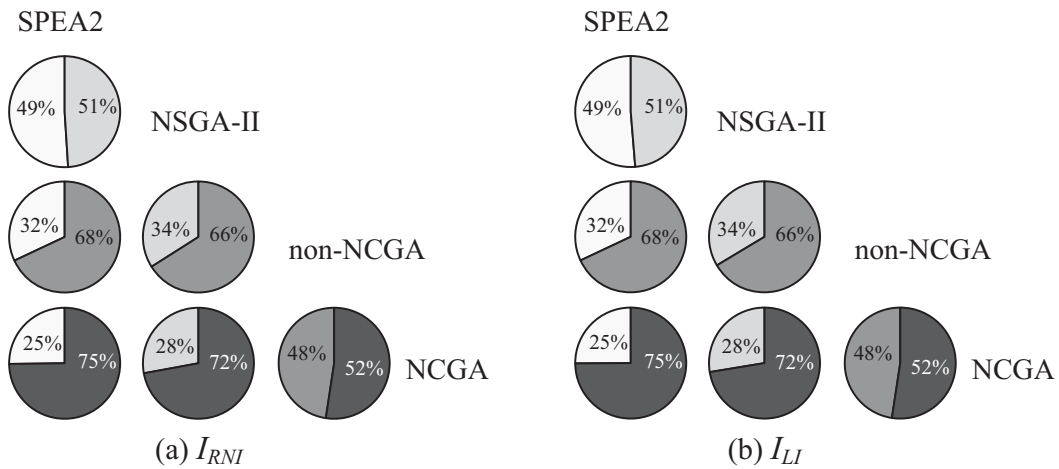


图 6.12 I_{RNI} and I_{LI} of ZDT4

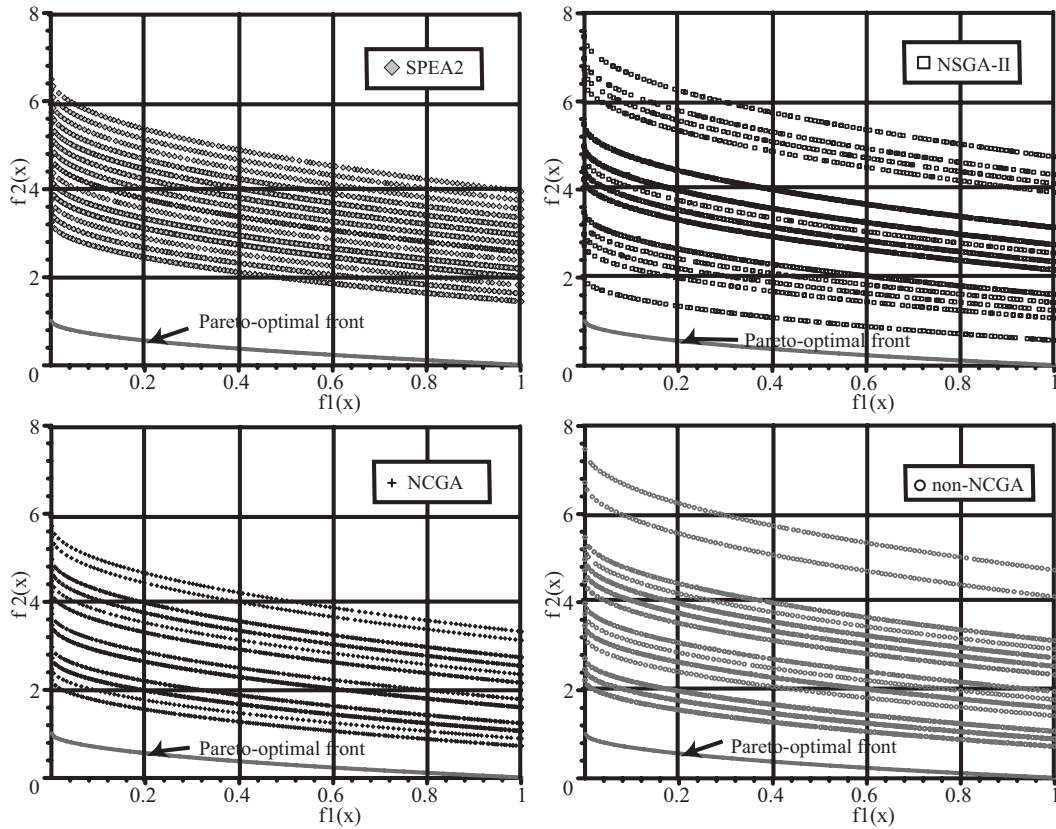
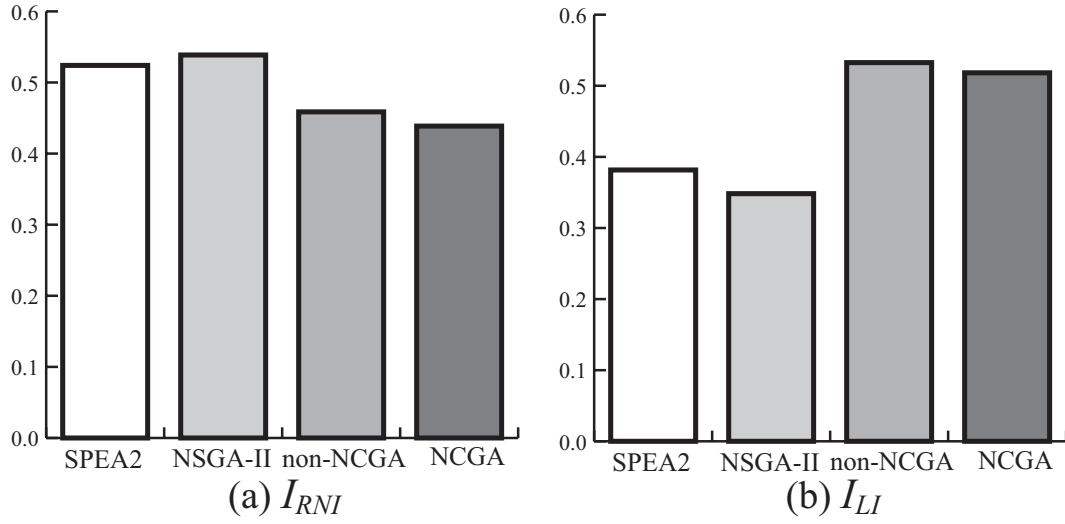


図 6.13 Pareto optimum individuals(ZDT4)

明確に優位であることから、これらの結果が探索個体群の生成方法の違いに起因していると考えられる。NCGA および non-NCGA では単純にアーカイブ個体群を次世代の探索個体群としてそのまま用いているのに対して、SPEA2, NSGA-II ではアーカイブ個体群に対してメイティング選択を用いて次世代の探索個体群を生成するという方法を用いている。SPEA2, NSGA-II におけるメイティング選択では、それぞれアーカイブ個体群から復元抽出で選択され探索個体群を生成している。このように次世代探索個体群の生成において選択圧をかけることにより、探索個体群は精錬され探索性能が向上するものと考えられる。しかし、ZDT4 のように多峰性のある問題では局所解に陥りやすくなる。そのため、母集団の多様性を保持しながら探索を行っている NCGA や non-NCGA よりも誤差が大きくなっていると思われる。

一方、幅広さについて見た場合、図 6.10(a) および図 6.11 から、各手法とも大きな差がないことが分かる。これは、非劣解集合のプロット図である図 6.13 における各手法の非劣解集合の分布からも確認することができる。

ZDT6

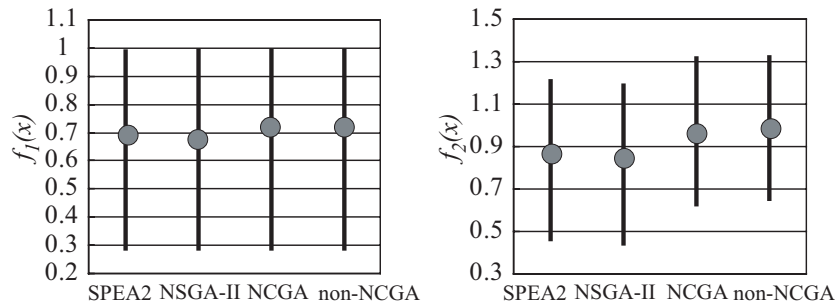
図 6.14 I_{cover} and I_{error} of **ZDT6**

この問題は、 $f_1(x)$ と x_1 の間に偏りが存在する．そのため、 $f_1(x)$ に関して多様な非劣解が得られるかどうかポイントとなる．

得られた結果の内、 I_{cover} と I_{error} を図 6.14 に、 I_{MMA} を図 6.15 に示す．また、各手法で得られた非劣解を比較して求める I_{RNI} および I_{LI} を図 6.16 に示す．得られた非劣解に対するプロット図を図 6.17 に示す．

ZDT6 は、**ZDT4** のように多峰性はないため、比較的パレート最適フロントまで探索が進みやすい．また、この問題の特徴である $f_1(x)$ と x_1 の間に偏りに関して見た場合、図 6.17 および図 6.15 より、どの手法においても $f_1(x)$ の広い範囲において解が得られていることが分かる．

得られた結果のうち、図 6.16 から分かるように NCGA および non-NCGA が SPEA2, NSGA-II に比べて明確に劣っている．これは、次のような理由によるものと考えられる．SPEA2, NSGA-II は NCGA と比較してパレート最適解付近へ接近しようとする力が強い

図 6.15 I_{MMA} of **ZDT6**

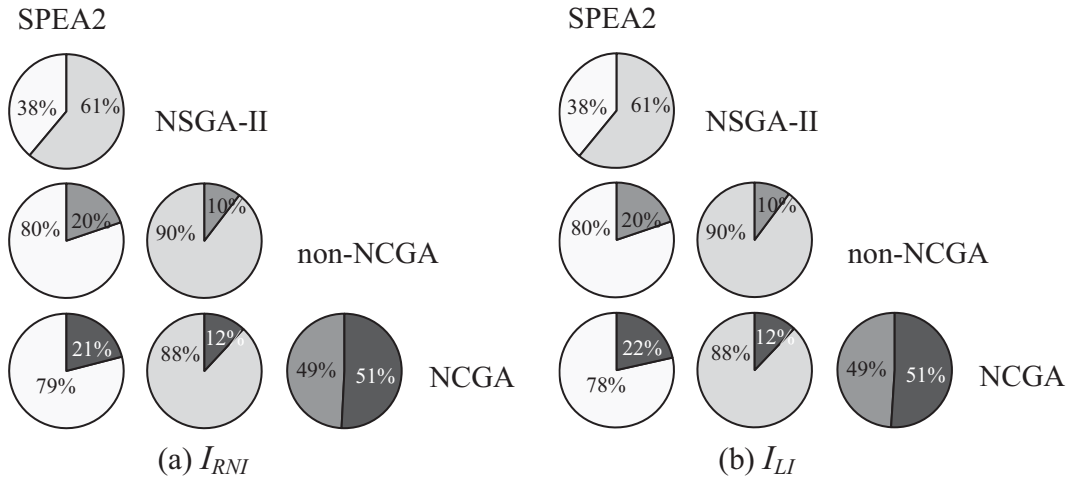
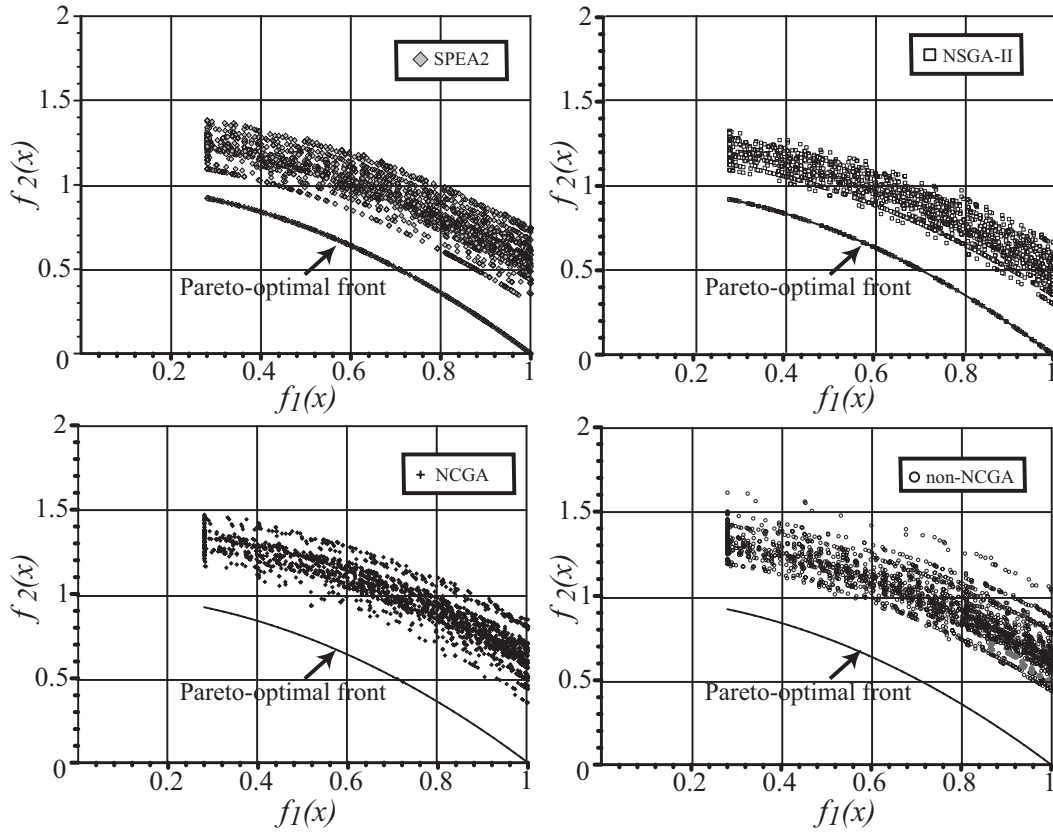
図 6.16 I_{RNI} and I_{LI} of ZDT6

図 6.17 Pareto optimum individuals(ZDT6)

という特徴を持っている．一方，この例題 **ZDT6** は **ZDT4** と比較して局所解なども存在せず，探索の容易な問題である．そのため，パレート最適解へ接近しようとする力の強い SPEA2，NSGA-II が良好な結果な結果を得ることができたと考えられる．このパレート最適解への接近しようとする力の差は，探索個体群の生成方法の違いに起因している．これは，上述の **ZDT4** において述べた理由と同じである．**ZDT4** のような多峰性が存在する問題では，強い探索の収束性は，マイナスの効果をもたらすのに対して，**ZDT6** のような多峰性もなく比較的探索が容易な問題では，強い探索の収束性が良好な結果をもたらしている．

KUR

この関数における結果のうち， I_{cover} と I_{error} を図 6.18 に， I_{MMA} を図 6.19 に示す．また，各手法で得られた非劣解を比較して求める I_{RNI} および I_{LI} を図 6.20 に示す．得られた非劣解に対するプロット図を図 6.21 に示す．

この問題は， $f_1(x)$ において隣り合う変数間において相互作用があり， $f_2(x)$ において多峰

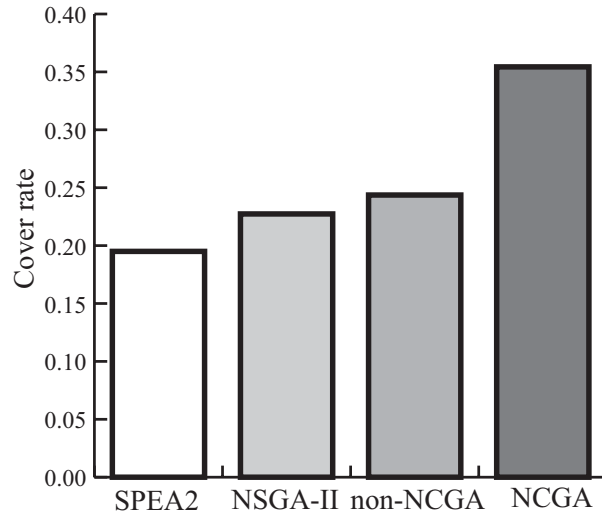


図 6.18 I_{cover} of **KUR**

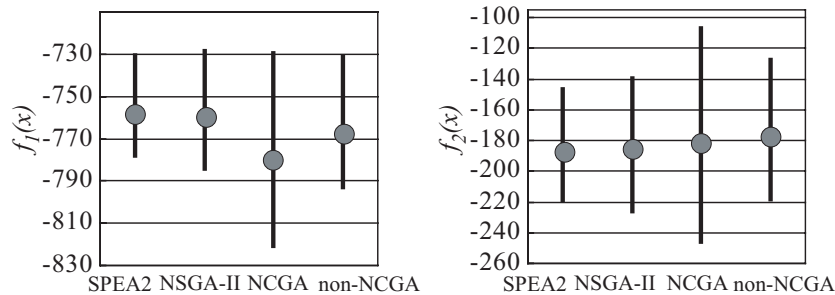


図 6.19 I_{MMA} of **KUR**

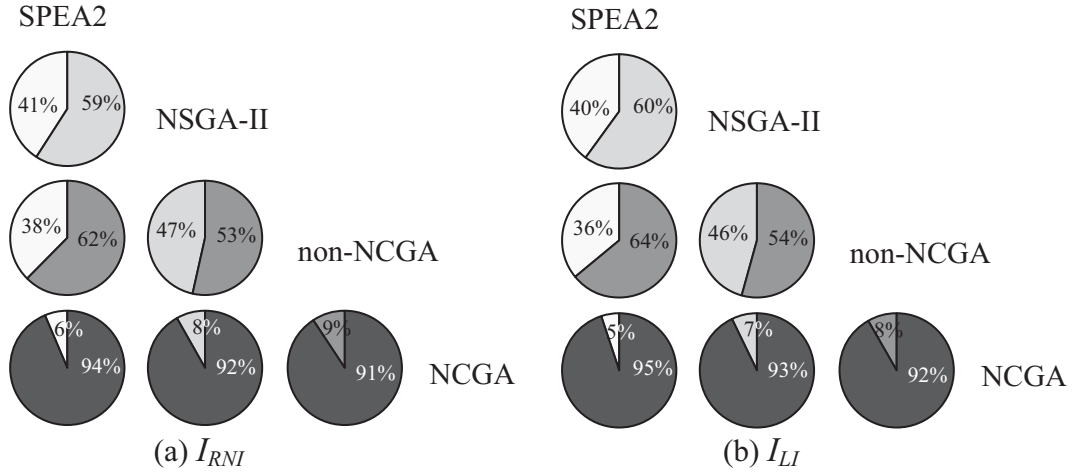
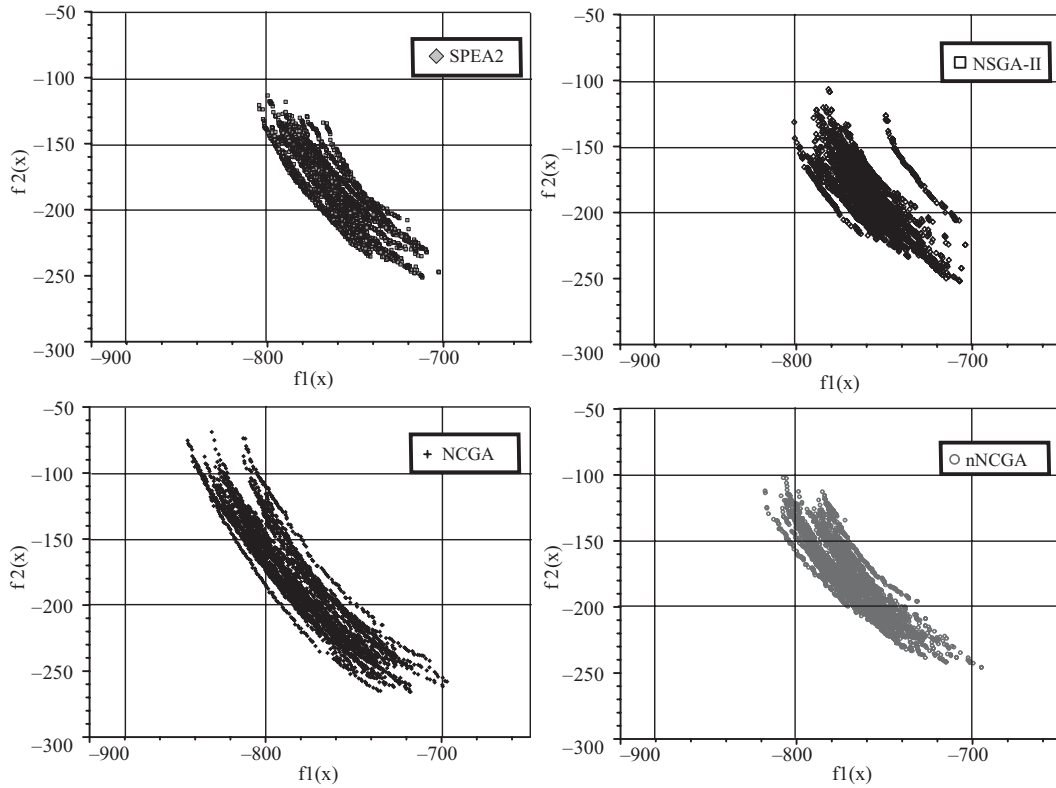
図 6.20 I_{RNI} and I_{LI} of KUR

図 6.21 Pareto optimum individuals(KUR)

性を有する問題である。また、設計変数の数が 100 と多いため、10 変数の **ZDT4**, **ZDT6** と比べ探索がより困難な問題であると言える。

図 6.20 の結果から、NCGA は他の手法よりも優位な結果を示しているのが分かる。さらに、図 6.19 から、各目的関数軸において最も広範囲に探索個体が分布しているのが分かる。この主な理由として、以下の 2 つが挙げられる。

- i) アーカイブ個体群からの探索個体群の生成方法。
- ii) 近傍交叉の効果。

上述のように、NCGA, non-NCGA と SPEA2, NSGA-II の探索個体群の生成方法は異なっている。SPEA2, NSGA-II では、アーカイブ個体群から復元抽出の選択を行い探索個体群を生成している。そのため、探索個体のパレート最適解に対する接近は早くなるものの、母集団は多様性を失いやすくなり、局所解にも陥りやすい。そのため、**KUR** のように多峰性のある問題において良好な解を得ることが難しい。しかも、**KUR** は $f_1(x)$ において隣合う変数間での相互作用を持っており、設計変数空間における多様性を保持していなければ幅広い近似パレート最適フロントを得ることはできない。そのため、SPEA2, NSGA-II は、NCGA, および non-NCGA と比較して限られたパレート最適フロントの部分しか探索が行えていない。

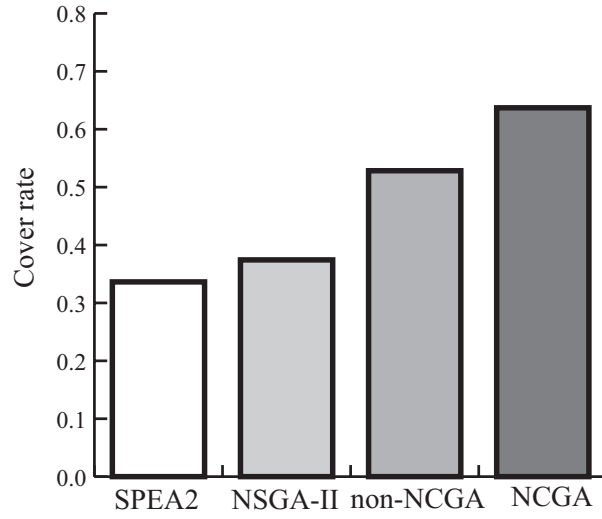
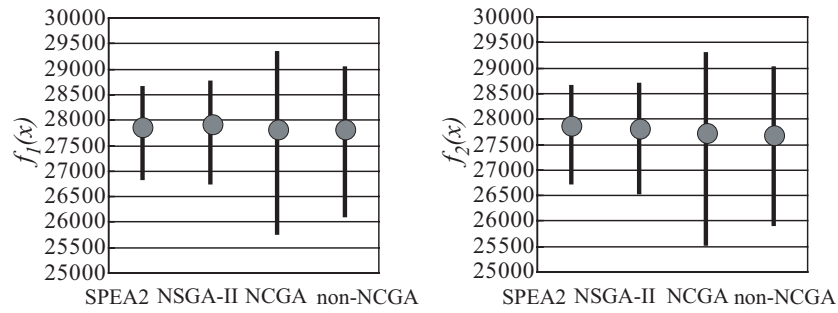
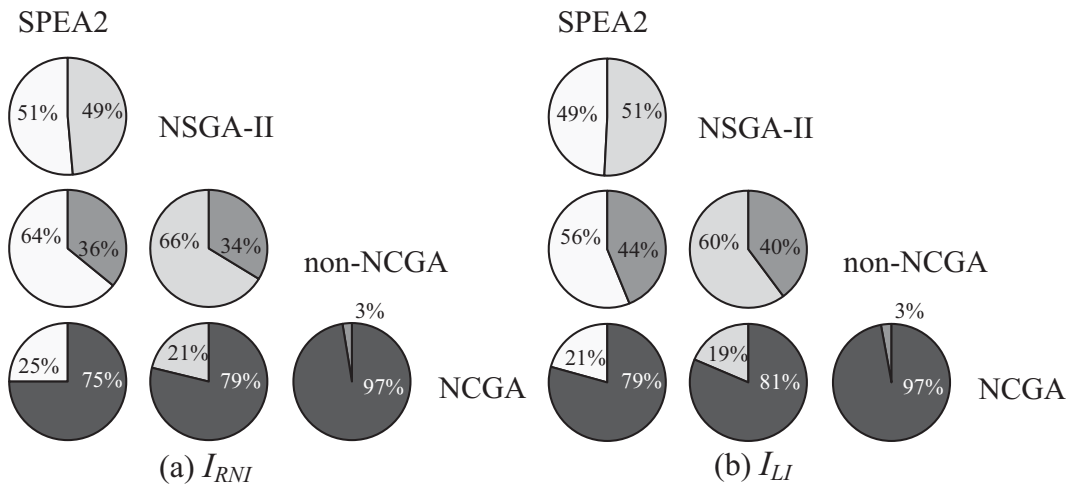
また、NCGA と non-NCGA を比較した場合、NCGA が大きく勝っていることから近傍交叉が探索に非常に効果的であることが分かる。**KUR** は、 x_1 の値によって非劣解の幅広さが決定する **ZDT4**, **ZDT6** と異なり、パレート最適フロントを形成する探索個体の設計変数空間における多様性は非常に高い。つまり、パレート最適フロントの両端では個体の設計変数値も大きく異なっていると思われる。そのため、近傍交叉を行うことにより探索の効率是非常に向上すると考えられる。

KP750 – 2

2 目的 750 荷物のナップザック問題に対する結果のうち、 I_{cover} の結果を図 6.22 に、 I_{MMA} を図 6.23 に示す。また、各手法で得られた非劣解を比較して求める I_{RNI} および I_{LI} を図 6.24 に示す。得られた非劣解に対するプロット図を図 6.25 に示す。

本実験において用いた 2 目的 750 荷物ナップザック問題は、総組み合わせ数が非常に膨大であり探索が困難な問題である。また、前節までの連続関数の例題と異なり多目的最大化問題として定式化される。

図 6.22 および図 6.23 から、NCGA はその他の手法と比較してより広範囲に解が分布しているのが分かる。これは、プロット図である図 6.25 から確認することができる。**KP750 – 2** は、非常に幅広いパレート最適フロントを持つ問題であり、パレート最適フ

図 6.22 I_{cover} of **KP750-2**図 6.23 I_{MMA} of **KP750-2**図 6.24 I_{RNI} and I_{LI} of **KP750-2**

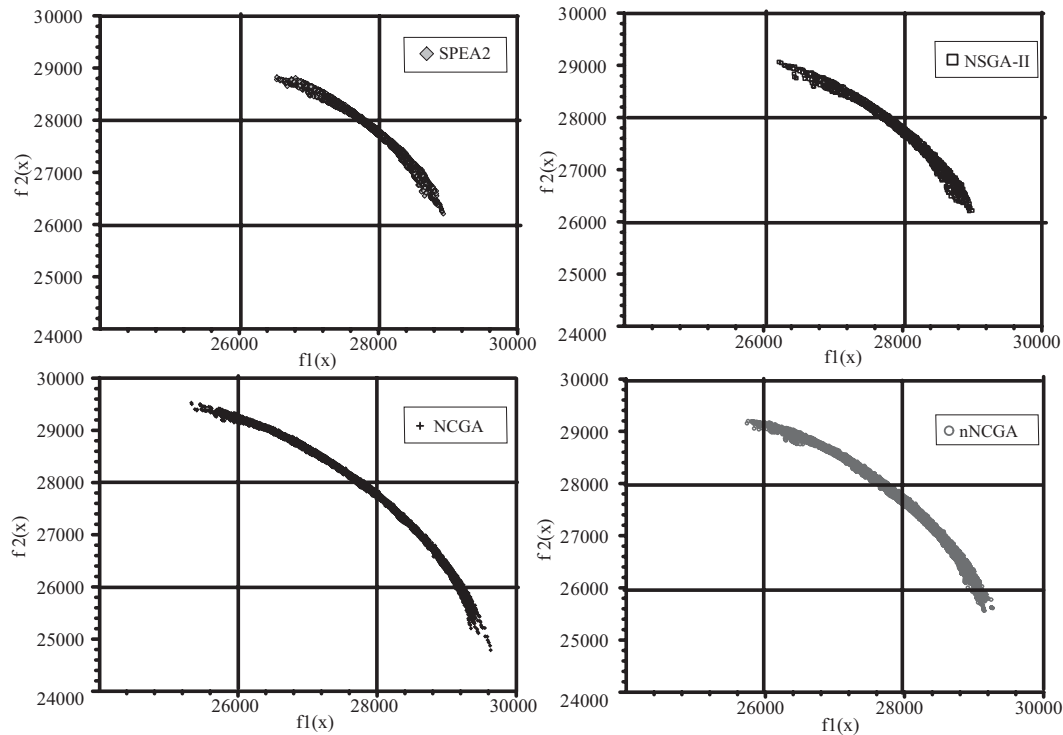


図 6.25 Pareto optimum individuals(KP-2)

フロントを形成するパレート最適個体の設計変数値も非常に多様性に富んでいる。そのため、**KUR**の場合と同様、幅広いパレート最適フロントを探索するためには、母集団の多様性が非常に重要となってくる。NCGA および non-NCGA では、SPEA2, NSGA-II に比べ多様性を保持しながら探索を行うため、結果として幅広いパレート最適フロントが得られている。

また、解の精度についても、図 6.24 から分かるように、NCGA は NSGA-II, SPEA2 よりも良好な解を得ているのが分かる。一方、non-NCGA は NSGA-II, SPEA2 よりも悪い結果を示している。さらに、NCGA と non-NCGA の比較では、NCGA が勝っていることが分かる。これらのことから、**KP750-2**においては、近傍探索が非常に効果的であるといえる。これは、探索母集団に多様な個体が存在する場合には、パレート最適フロントの同じ部分(もしくは近傍の部分)を探索している個体どうしでなければ設計変数値の異なりが大きくなり、結果として効果のない交叉が起きやすくなってしまうためだと考えられる。

手法の比較実験のまとめ

手法の比較実験として、4種類の連続テスト関数と多目的ナップザック問題の計5つの例題に対して提案手法である NCGA と NSGA-II, SPEA2, 近傍交叉を行わない non-NCGA の4つの手法の適用を試みた。

得られた結果は、以下に示すような例題の特徴によって傾向が異なっていた。

1. 問題の多峰性の有無.
2. パレート最適フロントを形成するパレート最適個体の多様性の有無.

問題に多峰性がある **ZDT4**, **KUR** のような場合、母集団の多様性が非常に重要となってくる。すなわち、探索個体群の局所的収束を避ける仕組みを持つことが重要となる。このような問題では、NCGA および non-NCGA は、SPEA2, NSGA-II に対して非常に優れた結果を得ることができた。しかし、多峰性がない **ZDT6** では、SPEA2, NSGA-II の方が良好な結果を得る場合もあった。このような NCGA, non-NCGA と SPEA2, NSGA-II の違いは、探索個体群の生成方法の違いに起因している。

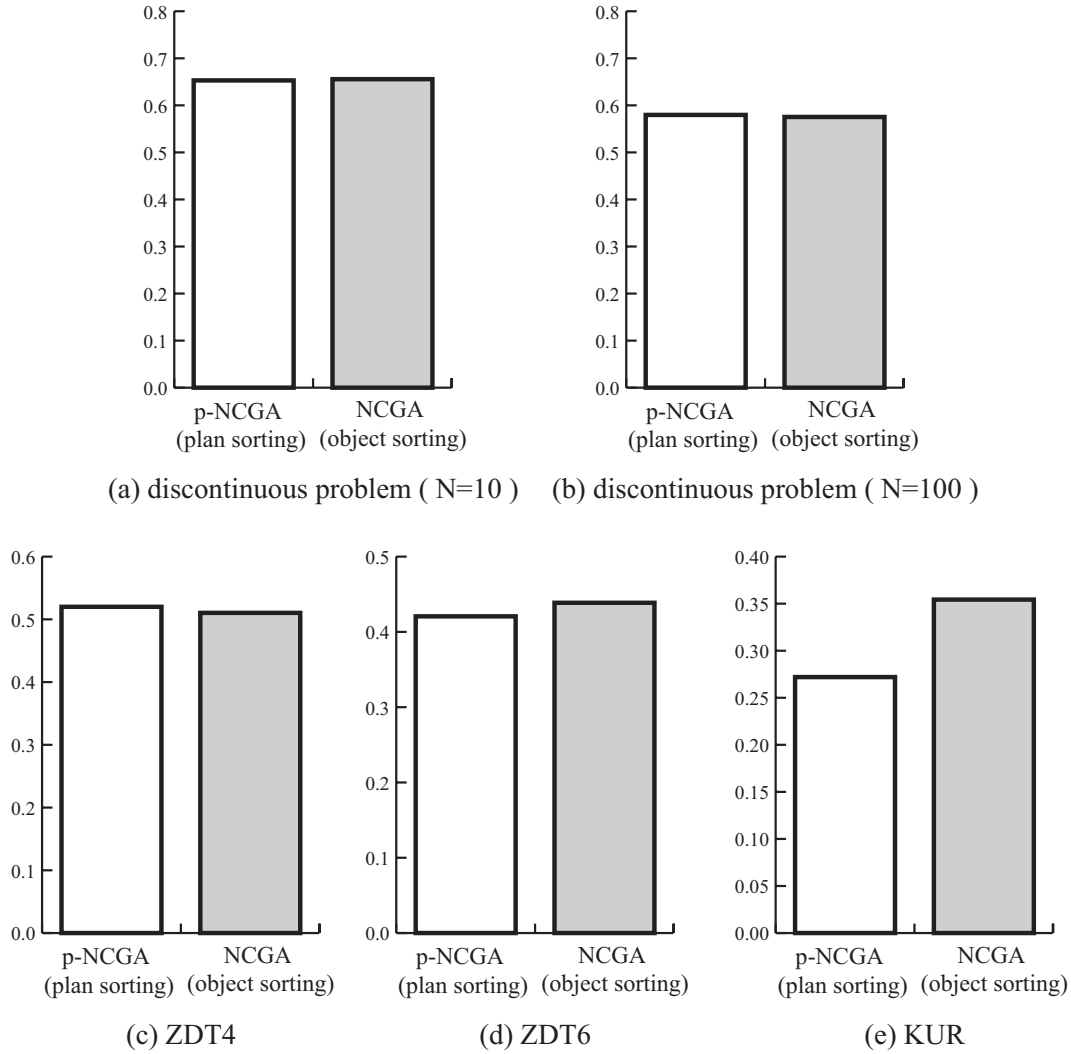
また、**KUR** や **KP750-2** のようにパレート最適フロントを形成するパレート最適個体の設計変数値が多様性に富んでいる場合、近傍交叉が非常に探索に効果的であることが分かった。このような問題においては、NCGA と non-NCGA の結果に顕著な差が認められた。このことは、 I_{RNI} および I_{LI} において明確に表れている。逆に、**F_{discon}**, **ZDT4**, **ZDT6** のような、 x_1 の値のみがパレート最適フロントの幅広さに関連しているような問題では、NCGA と non-NCGA の間にそれほど顕著な差は認められなかった。

近傍交叉は、**KUR** や **KP750-2** のように多様な母集団を用いて探索を行う必要のある問題において、パレート最適フロントの大きく異なる個体間の交叉を禁止することにより探索効率を向上させていた。一方、母集団全体の多様性が不要ない問題において近傍交叉は、あまり意味がないと言える。

また、上記以外として、**F_{discon}** ($N=100$), **KUR** のように設計変数の数が多い場合にも、母集団の多様性が非常に影響していたことが挙げられる。そのため、これらの問題においては、NCGA, non-NCGA の方が SPEA2, NSGA-II よりも良好な結果を得ることができた。これは、長い遺伝子長を用いる問題において、探索の初期、中盤の段階まで母集団がある一定のビット列に収束すると最適なビットの組み合わせを探索できないためであると思われる。

6.4.5 NCGA におけるソート基準の検討実験

本節では、NCGA におけるソート基準についての検討を行う。具体的には、ソート基準として目的関数空間を用いた場合と設計変数空間を用いた場合についての比較実験を行った。設計変数空間を用いた場合の NCGA では、6.3 節の目的関数空間でのソートと同様に、基準となる設計変数値 x_j は、世代 t を設計変数の数 P で割った余りから求める ($t \equiv j \pmod{P}$)。

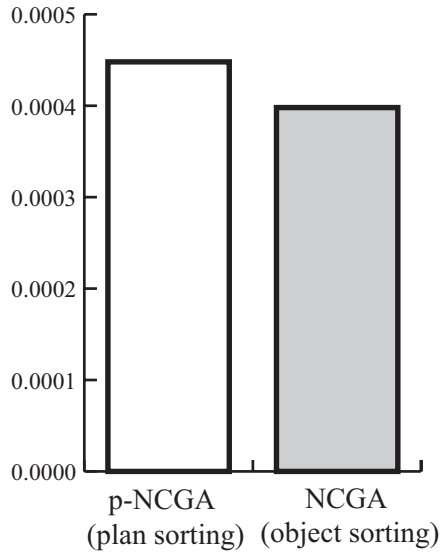
図 6.26 I_{cover} of p-NCGA

比較実験として、目的関数空間でのソーティングを用いた NCGA(以下, 単に NCGA と記す) と設計変数空間でのソーティングを用いた NCGA(以下, p-NCGA と記す) の実験を行った. 対象問題としては, 連続関数テスト問題を用いた. 多目的ナップザック問題は, 変数にもとづくソーティングを実行することができないため, 本実験では用いていない.

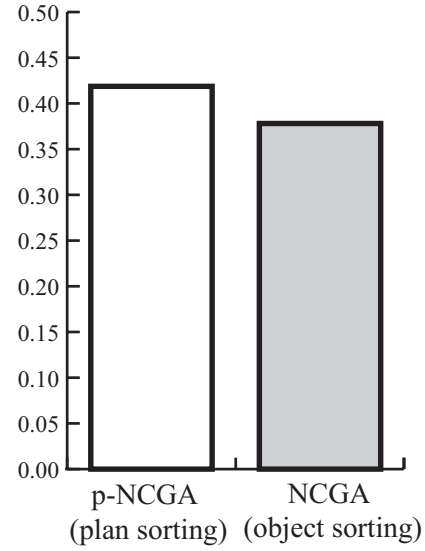
得られた結果のうち, 全ての対象例題における I_{cover} および I_{error} を図 6.26, 図 6.27 に示す³.

$\mathbf{F}_{discon}(N = 10)$ $\mathbf{F}_{discon}$ の設計変数の数が 10 の場合 ($N = 10$) の結果を考察する. 得られた結果のうち, I_{RNI} および I_{LI} を図 6.28 に, 得られた非劣解に対するプロット図を図

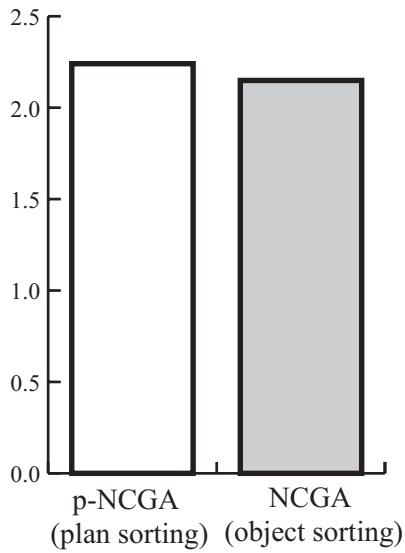
³ただし, **KUR** は I_{error} の計測が不可能なため図 6.27 には含まれていない.



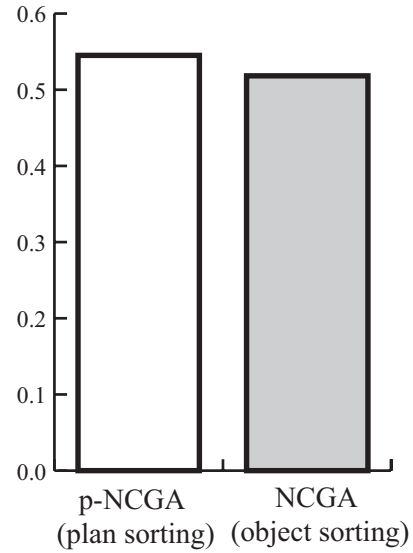
(a) discontinuous problem (N=10)



(b) discontinuous problem (N=100)

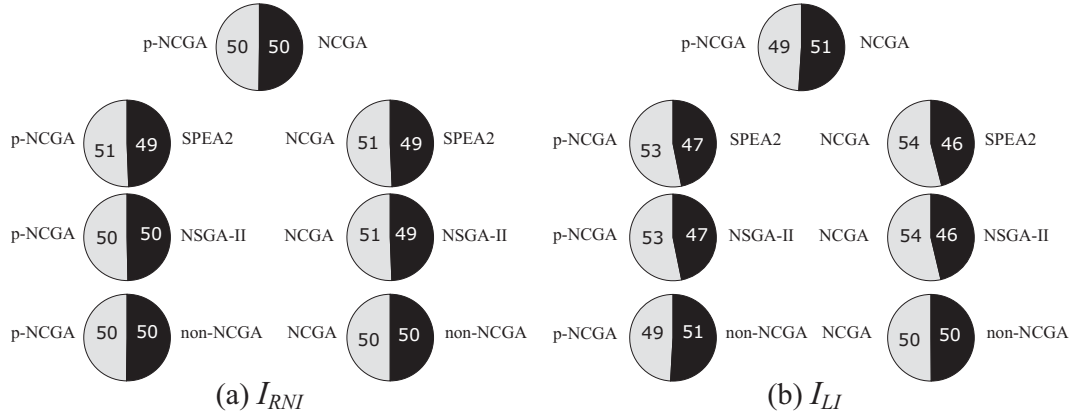
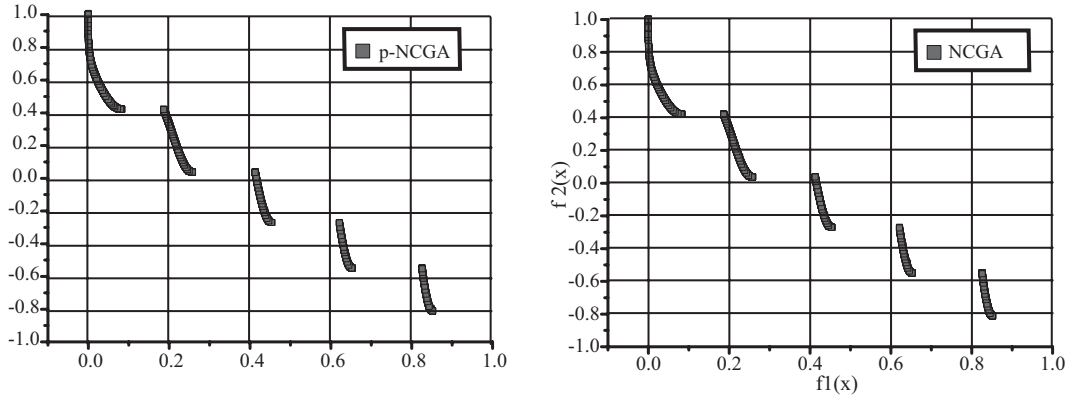


(c) ZDT4



(d) ZDT6

図 6.27 I_{error} of p-NCGA

図 6.28 I_{RNI} and I_{LI} of $\mathbf{F}_{\text{discon}}(N = 10)$ 図 6.29 Pareto optimum individuals($\mathbf{F}_{\text{discon}}(N = 10)$)

6.29 に示す。

6.4.4 節の各手法の比較の場合と同様，p-NCGA，NCGA とともにパレート最適解付近の非劣解が得られている．比較の結果である図 6.28 から，両手法にあまり差がないことが分かる．これは，図 6.26 の I_{cover} ，図 6.27 の I_{error} から分かるように両手法ともパレート最適解に近く幅広い良好な解が得られているためである．

$\mathbf{F}_{\text{discon}}(N = 100)$ 次に， $\mathbf{F}_{\text{discon}}$ の設計変数の数が 100 の場合 ($N = 100$) の結果を考察する．得られた結果のうち， I_{RNI} および I_{LI} を図 6.30 に，得られた非劣解に対するプロット図を図 6.31 に示す．

図 6.30 より，設計変数の数 N が 100 の場合には，NCGA に比べ p-NCGA が劣った結果を示していることが分かる．しかし， I_{error} について見た場合，p-NCGA の値は NCGA に劣っているものの差はそれほど大きくない．このことから，p-NCGA は NCGA に劣っているものの，得られた非劣解の精度においては，それほど大きな差がないことが分かる．

これらの結果の最大の原因は、設計変数の数が 100 と多いためである。p-NCGA では、100 設計変数空間においてある 1 つの設計変数値を基準にしたソーティングを行い、それらの個体は、近傍にいるものとして近傍交叉を行う。しかし、100 設計変数空間においてある 1 つの設計変数値にのみ着目した近傍の定義は、本質的な近傍の定義にはなりにくい。対して、2 目的の目的関数空間での個体ソーティングを用いる NCGA では、目的関数空間 2 目的のうち 1 つの目的関数値を基準にするため比較的、本質的な近傍となりやすい。

特に、この $\mathbf{F}_{\text{discon}}(N = 100)$ は、**ZDT4**、**ZDT6** と同様に、設計変数 $x_1 (= f_1)$ 以外の設計変数値をいかに 0 に近づけるかが解探索において重要となる。そのため、 f_2 を基準としたソーティングを行い近傍交叉を行うことによって、非常に効率的な探索を実現することができる。これは、より 0 に近い設計変数値どうしの組み合わせを実現することが期待できるためである。対して、p-NCGA では設計変数空間を基準としたソーティングでの近傍交叉であるため、NCGA と比較して 0 に近い設計変数値どうしを組み合わせるような交

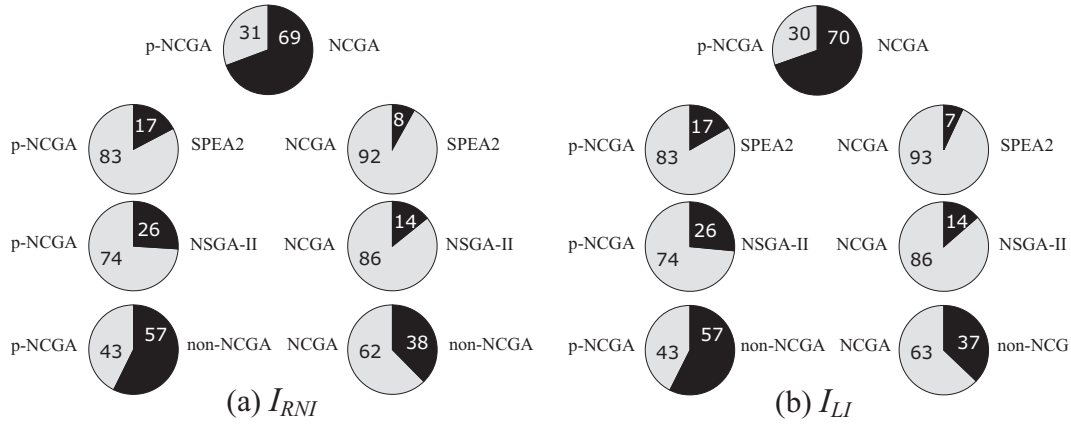


図 6.30 I_{RNI} and I_{LI} of $\mathbf{F}_{\text{discon}}(N = 100)$

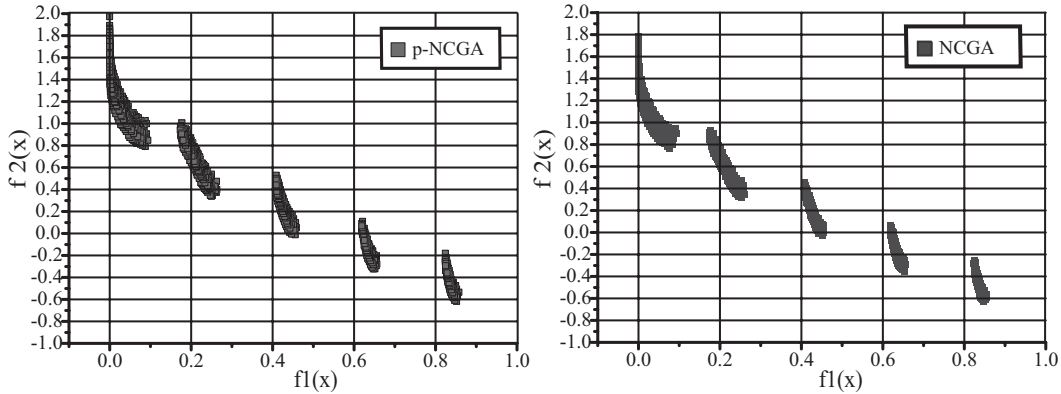


図 6.31 Pareto optimum individuals($\mathbf{F}_{\text{discon}}(N = 100)$)

叉を実現することは難しい．さらに，0に近い設計変数値を組み合わせさせて探索を進めるとい
うことに関して，p-NCGAはランダムに親個体を選択する交叉方法と比較しても不利にな
ると思われる．そのため，p-NCGAは近傍交叉を行わない non-NCGA よりも結果が劣っ
ていたと思われる．このことは，特に設計変数の数が多い場合に生じやすいものと考えら
れる．

ZDT4 ZDT4の結果のうち， I_{RNI} および I_{LI} を図 6.32 に，得られた非劣解に対するプ
ロット図を図 6.33 に示す．

比較結果である図 6.32 から，**ZDT4**において NCGA と p-NCGA はあまり差がないこ
とが分かる．このことは， I_{cover} や I_{error} における結果にも表れている．この問題は，先ほ
どの $\mathbf{F}_{discon}(N = 100)$ に比べ設計変数の数が少ない．また，この **ZDT4**は，多峰性を有
する問題であり，母集団の設計変数空間における多様性が重要となる．設計変数空間を基
準にした近傍交叉を行っている p-NCGA では，設計変数空間において多様な解を得ること

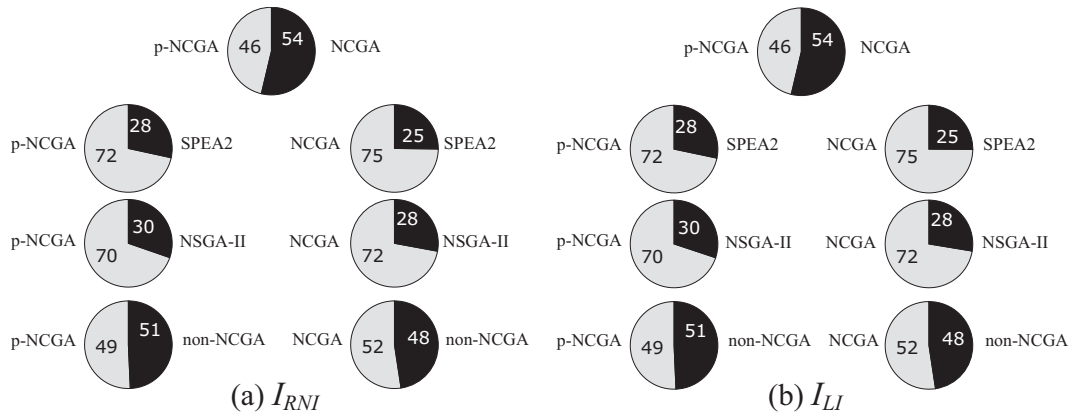


図 6.32 I_{RNI} and I_{LI} of ZDT4

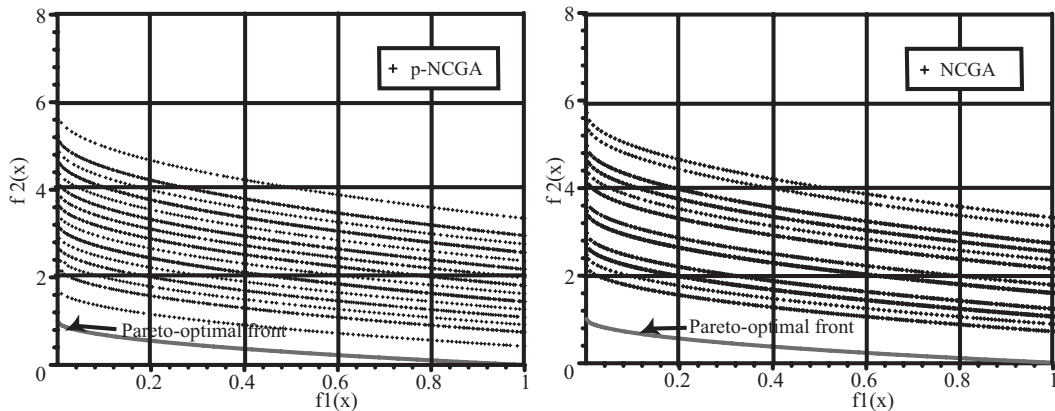


図 6.33 Pareto optimum individuals(ZDT4)

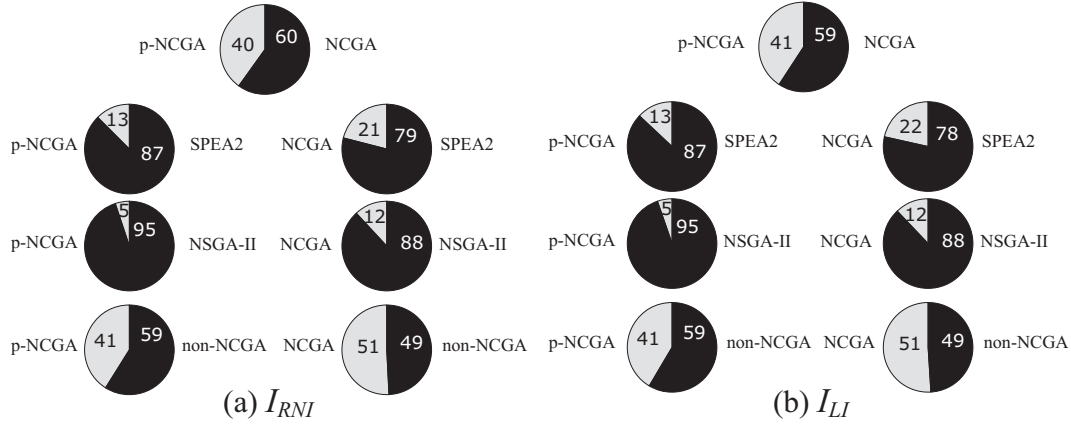
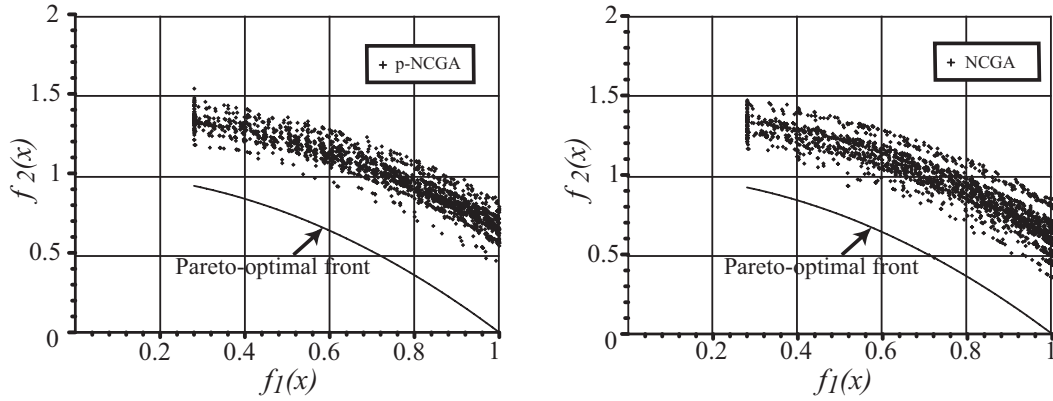
図 6.34 I_{RNI} and I_{LI} of ZDT6

図 6.35 Pareto optimum individuals(ZDT6)

ができると思われる。一方, p-NCGA は, 僅かながら NCGA および近傍交叉を行わない non-NCGA よりも劣っている。これは, $\mathbf{F}_{\text{discon}}(N = 100)$ で述べたとおり, ZDT4 が設計変数 $x_1 (= f_1)$ 以外の設計変数値をいかに 0 に近づけるかという問題であることが原因と考えられる。

ZDT6 ZDT6 の結果のうち, I_{RNI} および I_{LI} を図 6.34 に, 得られた非劣解に対するプロット図を図 6.35 に示す。

ZDT6 は, $\mathbf{F}_{\text{discon}}(N = 100)$ 程ではないものの ZDT4 の場合よりも NCGA と p-NCGA の差が大きい。比較結果である図 6.34 から, NCGA が p-NCGA よりも良好な結果を得ていることが分かる。これは, ZDT6 が ZDT4 よりも探索のしやすい問題であり多峰性のない問題であるためと思われる。 $\mathbf{F}_{\text{discon}}(N = 100)$ において述べたとおり, 本質的に $\mathbf{F}_{\text{discon}}$, ZDT4, ZDT6 において p-NCGA はランダムに親個体を選択する方法よりも探索に不利である。 ZDT4 では, 問題の特性として多峰性があり, 設計変数空間においてよ

り多様な母集団が探索に有利であったため p-NCGA は、それほど NCGA と差がなかった。しかし、**ZDT6** のように探索しやすく多峰性のない問題では、**ZDT4** の場合よりも、より大きく NCGA と p-NCGA の差が生じた。

KUR **KUR** の結果のうち、 I_{RNI} および I_{LI} を図 6.36 に、得られた非劣解に対するプロット図を図 6.37 に示す。

比較結果である図 6.36 より、p-NCGA は NCGA と比較して大きく劣っているものの、他の手法と比較して優位な結果を示していることが分かる。まず、NCGA に比べ大きく劣った理由としては、設計変数の数 N が 100 と多いことがあげられる。つまり、 $\mathbf{F}_{\text{discon}}(N = 100)$ の場合と同様に、100 設計変数空間においてある 1 つの設計変数値を基準にしたソーティングでは、本質的な近傍の定義にはなりにくいことが原因である。

また、p-NCGA が他の手法と比較して優位な結果を示している最大の原因は、**KUR** が異なり隣り合う設計変数間に依存関係を持つためである。つまり、ある 1 つの設計変数値

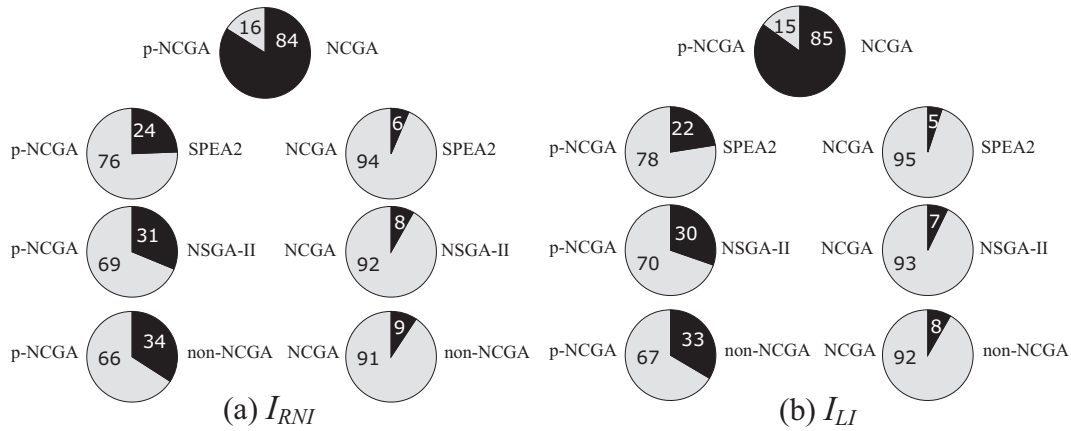


図 6.36 I_{RNI} and I_{LI} of **KUR**

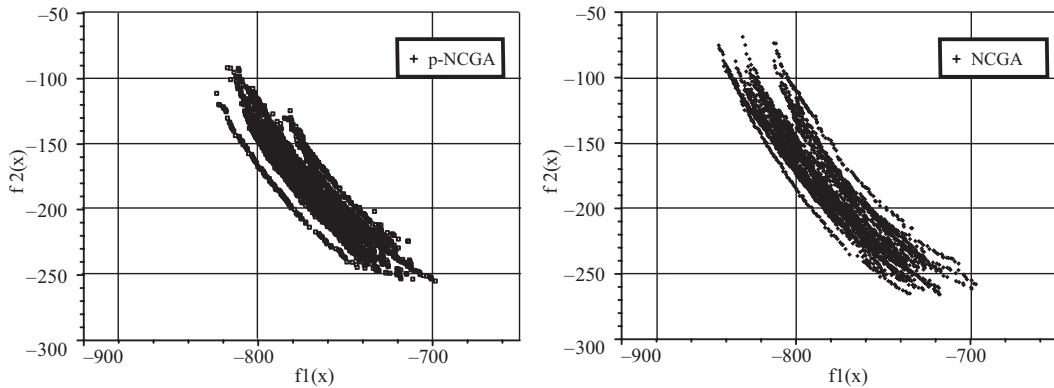


図 6.37 Pareto optimum individuals(**KUR**)

を基準としたソートを行った場合、隣り合う個体は近傍どうしとなる傾向が強いのである。そのため、p-NCGA は、近傍交叉を用いない non-NCGA と比較して良好な結果が得られたものと思われる。

NCGA におけるソート 基準の検討実験のまとめ

本節では、NCGA におけるソート基準として設計変数空間を用いた場合 (p-NCGA) と目的関数空間を用いた場合 (NCGA) の比較実験を行った。これらの 2 つの NCGA に対して、連続関数テスト問題である $\mathbf{F}_{\text{discon}}(N = 10)$, $\mathbf{F}_{\text{discon}}(N = 100)$, $\mathbf{ZDT4}$, $\mathbf{ZDT6}$, $\mathbf{KUR}$ の 5 つのテスト関数を用いた。

これらの結果より、以下のことが明らかとなった。

- i) 設計変数の多い問題では、p-NCGA は NCGA と比較して劣っていた。これは、扱う次元の数が多くなるにつれ近傍の定義が難しくなるためである。つまり、 $\mathbf{F}_{\text{discon}}(N = 100)$ の場合、設計変数空間では 100 次元空間において近傍定義を行わなければならないが、目的関数空間では 2 次元空間において近傍定義を行う。特に、本実験で用いた NCGA はある 1 つの基準値にもとづくソーティングにより近傍の定義を行っていたため、顕著にこの影響がでたものと思われる。
- ii) $\mathbf{KUR}$ のように、隣り合う設計変数間に依存関係があり、多様な設計変数値がパレート最適フロントに存在する問題において、p-NCGA, NCGA とともに効果的な探索を行うことができた。これは、このような問題において p-NCGA の近傍の定義が、実際の近傍に近いためである。逆に、 $\mathbf{F}_{\text{discon}}$, $\mathbf{ZDT4}$, $\mathbf{ZDT6}$ において p-NCGA は、近傍交叉を行わない non-NCGA よりも僅かに悪い結果を示した。これは、これらの問題では x_1 以外の設計変数値に関して、0 に近い設計変数値が組合わさるような交叉が探索に効果的であったのに対して、設計変数空間での近傍の定義にもとづく p-NCGA ではこのような交叉の実現が困難であったためである。

今回の実験では、設計変数の数が目的関数の数に比べ多い問題のみを用いていた。そのため、ある 1 つの基準値を用いてソーティングを行い近傍の定義を行っている NCGA では、次元数の少ない目的関数空間でのソーティングの方が有利であった。今後の課題として、以下のことがあげられる。

- 目的関数の数と設計変数の数が同程度の場合の問題に対する比較実験。
- ある 1 つの基準値にもとづくソーティングにより近傍の定義を行うのではなく、より多くの基準値を用いた近傍の定義を用いる。

6.5 まとめ

本論文では、新たな多目的GAにおけるアルゴリズムとして近傍培養型GA(Neighborhood Cultivation GA: NCGA)を提案した。本モデルは、これまでに提案されてきたアルゴリズムの探索に有効なメカニズムを考慮し、近傍交叉という独自のメカニズムを取り入れていたものとなっている。

また、提案する手法の解探索性能の検討を行うために、幾つかのテスト関数を用いてDebらにより提案されたNSGA-IIおよびZitzlerらにより提案されたSPEA2といった代表的な手法との比較を行った。

テスト関数を用いた数値実験により以下の点が明らかとなった。

- **ZDT6**を除く全ての問題において、NCGAは他の手法と比較して良好な結果が得られた。これは、探索個体群の多様性と近傍交叉による効果であると思われる。SPEA2、NSGA-IIでは、探索個体群はアーカイブ個体群から復元抽出により選択されるのに対して、NCGAではアーカイブ個体群のコピーがそのまま探索個体群に用いられるためより多様な個体を用いた探索が実現されている。その結果、より幅広い非劣解集合が得られた。また、近傍交叉によりパレート最適フロントに対する効率の良い探索が実現され、よりパレート最適フロントに近い非劣解集合を得ることができた。
- NCGAと近傍交叉を行わないnon-NCGAの比較の結果、全ての問題においてNCGAが優れた結果を得た。そのことから、近傍交叉は多目的GAの探索において効果的であることが分かった。特に、設計変数の数が多くより探索の難しい問題において近傍交叉の効果が強く認められた。
- NCGAにおけるソート基準として設計変数空間を用いた場合と目的関数空間を用いた場合の比較実験を行った結果、後者の方法が全ての例題に対して良好な結果を示した。これは、対象とした問題における目的関数の数と設計変数の数が大きく異なっていたためである。NCGAでは、ある1つの値を基準としてソートを行い近傍の定義を行っているため、対象となる空間の次元数が多くなるほど本質的な近傍の定義と異なってくる。一般に、扱う目的関数の数に比べ設計変数の数は膨大となる。また、各変数の持つ最適な値の組み合わせを求めるような問題において、設計変数空間を基準としたNCGAでは、良好な結果を得ることができず、逆に目的関数空間を基準としたNCGAでは、効果的な交叉を実現していた。

以上の点から、ソート基準を目的関数空間とした近傍交叉を用いるNCGAは多目的GAの探索において有効なアルゴリズムであると言える。

第 7 章

近傍培養型遺伝的アルゴリズムを用いた問題解決

7.1 はじめに

本章では、6 章において提案された近傍培養型 GA (Neighborhood Cultivation GA: NCGA) の実問題への応用についての検討を行う。6 章において、複数の連続関数テスト問題および多目的ナップザック問題における NCGA の有効性が検証され、良好な結果が得られた。

しかしながら、これらの結果は必ずしも実問題に対する NCGA の有効性を証明するものではない。実際に対象とする実問題の多くは、6 章において用いたテスト問題のように単純な関数式、もしくは評価式ではなく、複雑で離散的である。そのため、NCGA の多目的最適化問題への有効性を証明するためには、より実問題に近い複数の問題に対する有効性を検証する必要がある。そのため、本章では以下の特徴の異なる 2 つの問題に対して NCGA の有効性の検証を行った。

- i) ディーゼルエンジン噴射スケジュールの最適化
- ii) 矩形ブロックの 2 次元空間上での配置面積の最小化

ディーゼルエンジン噴射スケジュールは、ディーゼルエンジン燃料の最適な噴射率を求める問題である。ここでは、燃費率、NO_x 排出、すす排出の最小化を目的とする 3 目的最適化問題として扱った。経験的に、これら 3 目的間には何らかのトレードオフの関係があることが分かっている。

また、矩形ブロックの配置面積の最小化は、あらかじめ定められた複数の矩形部ブロックを 2 次元空間上に配置する問題であり、配置面積の縦、横の長さの最小化を目的とする 2 目的最適化問題として用いた。配置面積の縦、横の長さを最小化することで、単に面積の最小化だけでなく解選考者に対して様々なアスペクト比を持つ最小面積を提示すること

ができる。本章では、上記の問題に対する NCGA の適用と得られた解の有効性の検証を行う。

7.2 ディーゼルエンジン噴射スケジュールの多目的最適化

ディーゼルエンジンをはじめとするディーゼル機器は、燃費、耐久性といった特徴のため小型から大型まで幅広く用いられている。しかしながら、近年、東京都の排ガス規制に見られるように、これらのディーゼル機器に対して環境面からの厳しい要求が高まっている。そのため、これらのディーゼル機器の排出する窒素酸化物 (NO_x)、すすなどの排出量を減少させることは急務となっている。

排ガス中のこうした物質を削減するための対策としては、大きく分けて次の 3 方向の研究開発がある⁴²⁾。

- 1) ディーゼル燃焼の改善
- 2) 使用燃料の検討
- 3) 後処理装置の開発

本節では、このうちディーゼル燃焼の改善に着目した排気特性の最適化を試みる。ディーゼル機関の燃焼改善のための対策には、噴射の特性設定が行われる。この噴射の特性設定には、多くのパラメータのチューニングとそれに伴う膨大なシミュレーション実験が必要となるため、一般には計算機上のシミュレーションを用いた最適設計が行われる。

ディーゼル燃焼の改善のために考慮すべきことは、排気特性と熱効率であり、すなわち燃費率の効率化、 NO_x 、すすの低減化が目的となる。しかしながら、ディーゼル燃焼のシミュレーションには多くの計算が必要となる上、多くの場合、最適パラメータ設計のための明確な方法論も確立されていない。そのため、これまでに行われてきたディーゼル機関の最適パラメータ設計の多くは、3 目的のうち 1 つの目的のみに注目し、その他の目的を制約条件とする単一目的最適化であった⁴³⁻⁴⁵⁾。

そのため、本節では燃費率、 NO_x 排出、すす排出の最適化を同時に行う 3 目的最適化を試みる。実験では、多段階噴射を想定し、燃料噴射期間内に噴射率を任意に変化させることが可能であるものとした。噴射率を変更することにより、燃費率、 NO_x 排出量、すす排出量が最小となる設定の探索を行う。経験的に、これら 3 目的間には互いにトレードオフの関係があることが知られている。例えば、 NO_x は高温の下で完全燃焼に近づくほど発生量が増加するのに対して、すすは低温で不完全燃焼状態になると生成されるという全く正反対の性質を持っているため、 NO_x とすすの両方に関する最適値を得ることはできない。

上記の3目的ディーゼルエンジン噴射スケジュール最適化に対し、6章において述べたNCGAの適用を試みた。また、計算負荷を軽減するために32台のPCにより構成されるクラスタ計算機を用いた並列計算を行った。

7.2.1 ディーゼル燃焼モデル

ディーゼル、ガソリンに限らず自動車などの大型機械のエンジンのほとんどは、吸入・圧縮・燃焼・排気の4つの行程をくり返す、4サイクルエンジンである。ただし、この行程をさらに細かく分類した場合、ディーゼルとガソリンに2つの違いをあげることができる。それは燃料と空気を混合するタイミングと、混合気を爆発させる方法である。

ディーゼルエンジンは空気のみを燃焼室に送り込んで圧縮し、その後燃料と混合する。これに対して多くのガソリンエンジンでは、あらかじめ燃料と空気を混合して燃焼室に送り込み、圧縮する。ディーゼルエンジンのように空気のみを圧縮すると、圧縮時の自然発火が原因となるノッキング¹が起こらないため、圧縮比²をより高めることができる。さらにノッキングがないとシリンダ径の制限も受けないため、排気量を増やして出力を高めることも可能となる。

また、混合気を爆発させる方法について見た場合、ディーゼルエンジンは圧縮した空気に高圧力で燃料を噴射し自然発火させる。対して、ガソリンエンジンは圧縮した混合気に点火プラグで着火する。ディーゼルエンジンのような自己着火の場合、多点で同時に燃焼するため、可燃範囲が広く、点火系の故障に起因するミスファイヤがない。しかし一方で、騒音が大きくなるという問題もある。

これら2つの異なりに起因するディーゼルエンジン燃焼のガソリンエンジン燃焼と比較した場合の利点、欠点を以下に示す。

利点 CO₂、CO、HCの排出量が少ない。

ディーゼルエンジンは、燃料に対して空気の量が十分な状態で燃焼するため酸素が不足による不完全燃焼が起こりにくい。そのためCO、HCの排出量が少ない。また、燃費効率に優れているため、CO₂の排出量もガソリンエンジンと比較して少ない。

欠点 NO_x・PM(Particulate Matter)を多く排出する

ディーゼルエンジンは点火装置を使わず自己着火によって燃焼を行う構造上、燃焼段階が「着火遅れ期間」「予混合燃焼期間」「拡散燃焼期間」「後燃え期間」の4つに分かれる。この中の「予混合燃焼期間」に爆発的な燃焼で高い熱発生を起こし、シリン

¹ガソリンエンジンで、圧縮時に点火プラグから離れた部分の混合気が自然発火し、燃焼室全体のガスが瞬間的に燃焼し、キンキンやカリカリなどの音が発生する現象。

²燃焼室の圧縮前の最大容積と、圧縮後の最小容積の比。圧縮比が高いほど出力が大きくなる。

ダ内の温度が急激に上昇する。この時、 NO_x が生成される³。一方、ディーゼルエンジンでは、空気と燃料の混合気体を用いるのではなく、シリンダ内に燃料を噴射して燃料と空気を混ぜ合わせるため混合状態が不均一になり易い。さらに、「拡散燃焼期間」と「後燃え期間」ではシリンダ内の温度と圧力が低下するため、そのため、低温状態の不完全燃焼において発生しやすい PM が多く発生する。

上記のように、ディーゼル燃焼モデルにはガソリン燃焼モデルと比較して NO_x 、PM が多量に発生するという問題点がある。特に、 NO_x 、PM は条例などの排ガス規制の対象となっており、その削減は急務である。

以下、 NO_x および PM の簡単な説明を示す。

NO_x NO 、 NO_2 、 N_2O 、 N_2O_2 などの酸化窒素の総称。高温下で窒素と酸素が結合して発生する。特性として、高温で完全燃焼に近づくほど発生量が増加するため、発生量をおさえるには燃焼温度を下げる必要がある。

PM Particulate Matter：粒子状物質の略。排出される炭素粒子のまわりに、燃料の燃焼中間生成物など、さまざまな化合物が付着したもので、黒煙（Soot）も PM の一種である。PM は、低温で不完全燃焼状態になると生成される。

一方、熱発生率の予測と NO_x および PM などの排気有害成分の予測を行うためには、ディーゼル燃焼現象のモデル化が必要となる。しかしながら、ディーゼル燃焼現象のすべてを数学的に記述することは現在のところ不可能である。これまでに、多くのディーゼル燃焼のモデルが提案されているが、それらは次の 3 つに分類できる⁴²⁾。

1) 熱力学的モデル (Thermodynamic Model)：

熱発生率のみを予測するもので、燃焼室内で行っている事象を全体として把握し、その時間的推移を追う。

2) 現象論的モデル (Phenomenological Model)：

ディーゼル燃焼現象の基本構成要素について、実験的に求めた実験式を主として計算に用いる。比較的短い時間で解析が可能である。

3) 詳細多次元モデル (Detailed Multidimensional Model)：

物理的・化学的課程を支配するすべての微分方程式を解くことでエンジンの特性の三次元構造を明らかにする。解析には膨大な計算が必要となる。

³ NO_x は、ガソリンエンジンでも排出されるが、後処理技術の確立により低減されている。

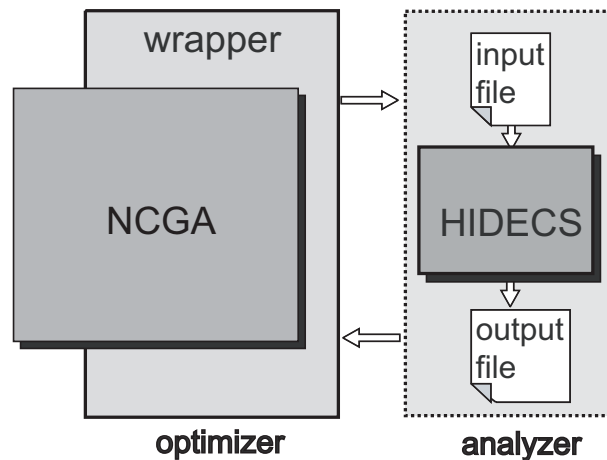


図 7.1 Optimization System

本研究では，広島大の廣安らによって開発された現象論的モデルの1つであるディーゼルエンジンシミュレータの HIDECS⁴⁶⁾ を用いる．HIDECS の最大の特徴は，必要とする計算負荷が非常に軽いことが挙げられる．詳細多次元モデルとしては，KIVA code⁴⁷⁾ が有名であるが，これは1試行の解析に非常に大きな計算負荷を必要とする．

HIDECS は，噴射率 (Injection Rate)，大気状態 (Atmosphere Condition) などをもとに，ディーゼル燃焼をシミュレートし，燃費や炭素化合物の生成量を求めることができる．ここでは，噴射率についての最適化を行う．噴射率は，ディーゼルエンジンの燃焼室の中にどのような時間的割合で燃料を噴射するかを表している．本研究では，総噴射量が一定のもと，噴射期間を 18° に固定し，36 分割に分割された単位時間⁴ ごとの燃料噴射を燃費効率 (Specific Fuel Consumption : SFC)，窒素酸化物 (NO_x) 排出量，すす (Soot) 排出量が最小となるように最適化した．

7.2.2 数値実験

最適化は，最適化を行う最適化部と解候補の評価を行う解析部の2つの組み合わせと考えることができる．実験には，NCGA を最適化部，HIDECS を解析部として使用した．実験に使用した最適化システムの概念図を図 7.1 に示す．

問題設定

HIDECS では，ボア径，噴射開始クランク角，噴射スケジュールなど複数の変数を扱うことができる．本実験では，このうち噴射率のみを設計変数として扱い最適化を行った．噴

⁴単位時間の基準は， $18^\circ/36 = 0.5^\circ$ である．

表 7.1 Specification of the target diesel engine

Injected Fuel Volume	40mg/st
Injection Timing	ATDC−5.0°
Injection Duration	18.0°

射に関する開始クランク角，期間などを表 7.1 に示す．

表 7.1 の各設計変数について説明する．ディーゼルエンジン内では，シリンダーのついたクランクシャフトが回転しながらその中で噴射・圧縮・爆発が行われている．本実験では，クランクシャフトが ATDC−5.0°⁵ から燃料の噴射が開始され，そこから 18.0° の間に燃料の噴射が行われるよう設定している．また，噴射期間の最小間隔である単位時間を 0.5° と設定しているため，本実験では合計 36 個に分割された噴射スケジュールを最適化の対象として扱う．

また，HIDECS により得られる出力を以下に示す．

- Specific Fuel Consumption (g/kWh)
- Indicated Mean Effective Pressure (kPa)
- Indicated Power (kW)
- NO_x (g/kWh)
- Soot (g/kWh)

本研究では上記の内，燃費 (Specific Fuel Consumption : SFC) ，窒素酸化物排出量 (NO_x) ，すす排出量 (Soot) の最適化を行った．

噴射率のコーディング

本実験では，36 個の単位噴射率からなる噴射スケジュールを HIDECS に入力する．しかし，実際のシステムにおいて実現できる噴射スケジュールはもっと単純である必要がある．

そこで，実験では問題を次のように単純化している．図 7.2 に示すように噴射率を最大でも 6 つの矩形からなるものとし，各区分の高さ (噴射率) および幅 (噴射期間) の 12 の設計変数について最適化を行う (図中の a から l) ．

⁵ATDC とは，After Top Dead Center の略であり．ピストンの上死点後を示す．そのため，ATDC−5.0° はピストンが一番上に上がった状態からクランクを 5 度戻した位置のことである．

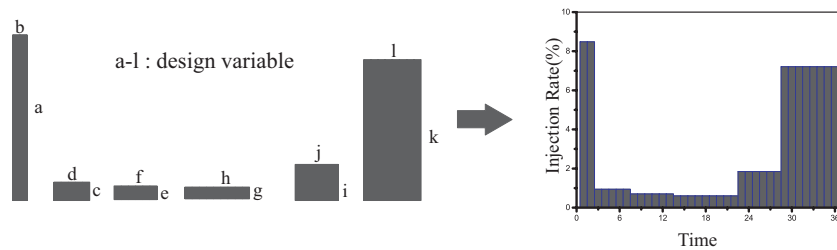


図 7.2 Coding method

次節で行うシミュレーションにおいて、総燃料および噴射期間は一定としているので、各区分の総面積および幅の合計は一定となるように、得られた設計変数値を正規化する必要が生じる。

そこで、本実験では矩形の縦・横の長さより各矩形の面積を足し合わせた総面積を求め、総面積が 100 となるように各矩形の幅（噴射期間）を表す設計変数値をスケーリングしている。なお、本研究ではそれぞれのパラメータを 10bit の gray コーディングによって表現する。

実験パラメータ

表 7.2 のパラメータを用いて実験を行った。

表 7.2 GA parameters

population size	100
crossover rate	1.0
crossover method	1X
total length of the chromosome	12 × 10 bit
mutation rate	1/120
total evaluation number	20000

また、計算環境として表 7.3 に示す PC クラスタシステムを使用した。なお、本クラスタシステムのネットワークには、一般的な FastEthernet および安価な Switching Hub を使用している。

実験結果

得られた非劣解集合のプロットを図 7.3 に示す。また、図 7.3 を各 2 次元平面に投影した非劣解集合を図 7.4、図 7.5、図 7.6 に示す。なお、非劣解集合のプロット図中における

表 7.3 Cluster System

CPU	Pentium III (1GHz)*32
Memory	512 Mb
OS	Linux2.4.4
Network	FastEthernet
	TCP/IP
Communication library	LAM

番号と非劣解プロット図の下に示される噴射率形状の番号は対応している。

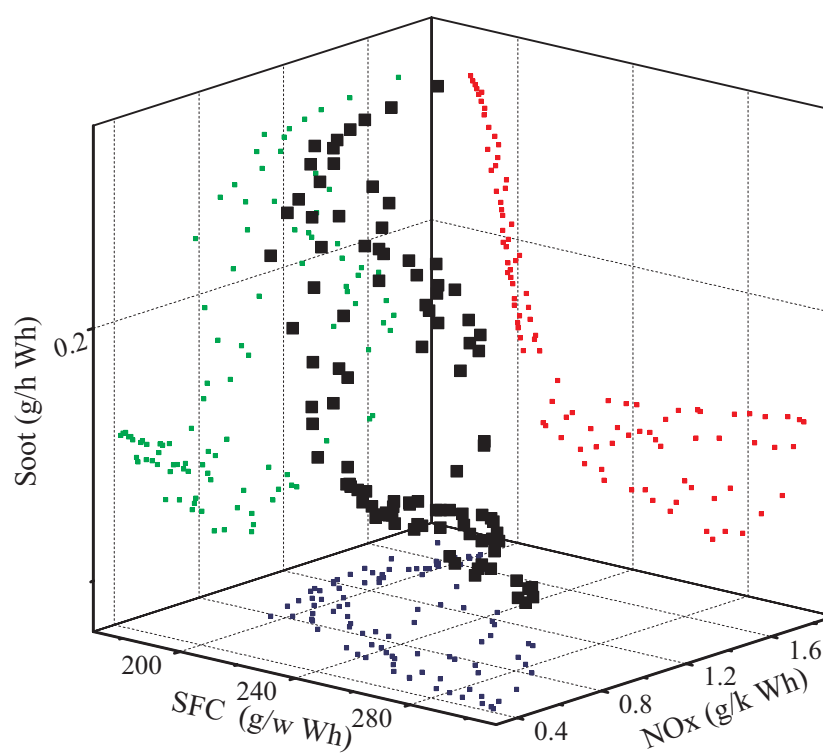


図 7.3 Derived non-dominated solutions

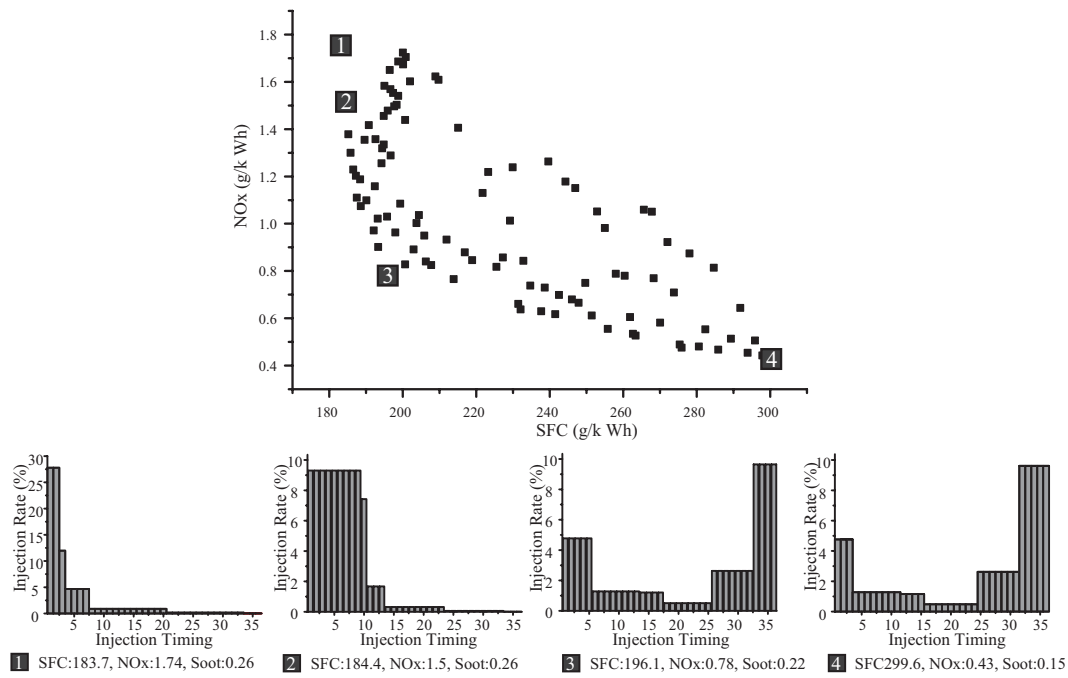


図 7.4 Derived non-dominated solutions(SFC,NOx)

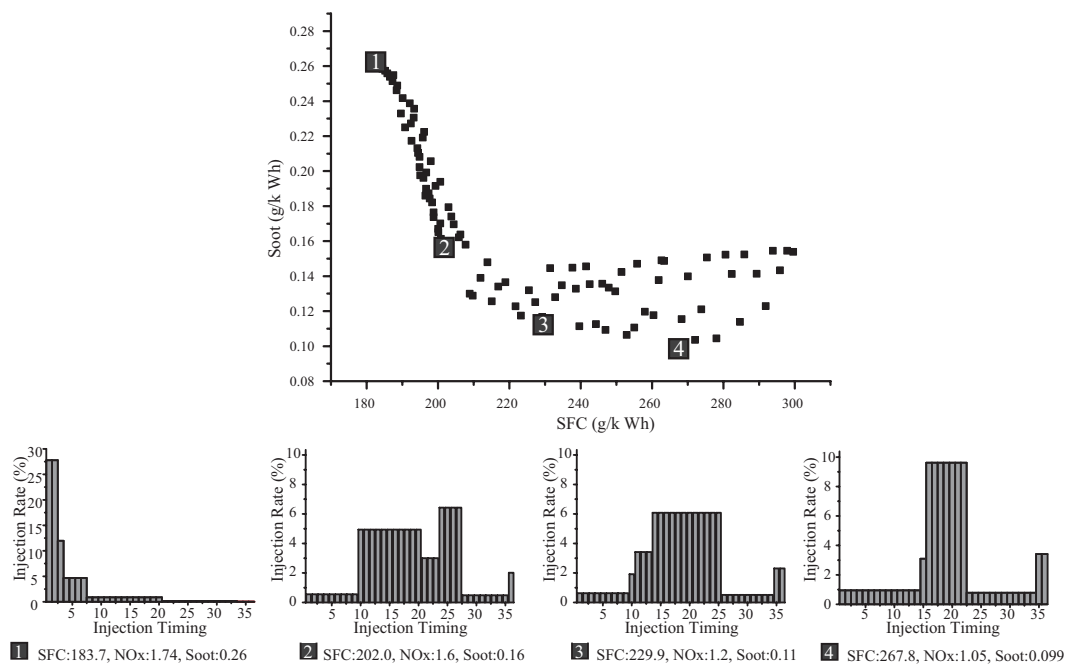


図 7.5 Derived non-dominated solutions(SFC,Soot)

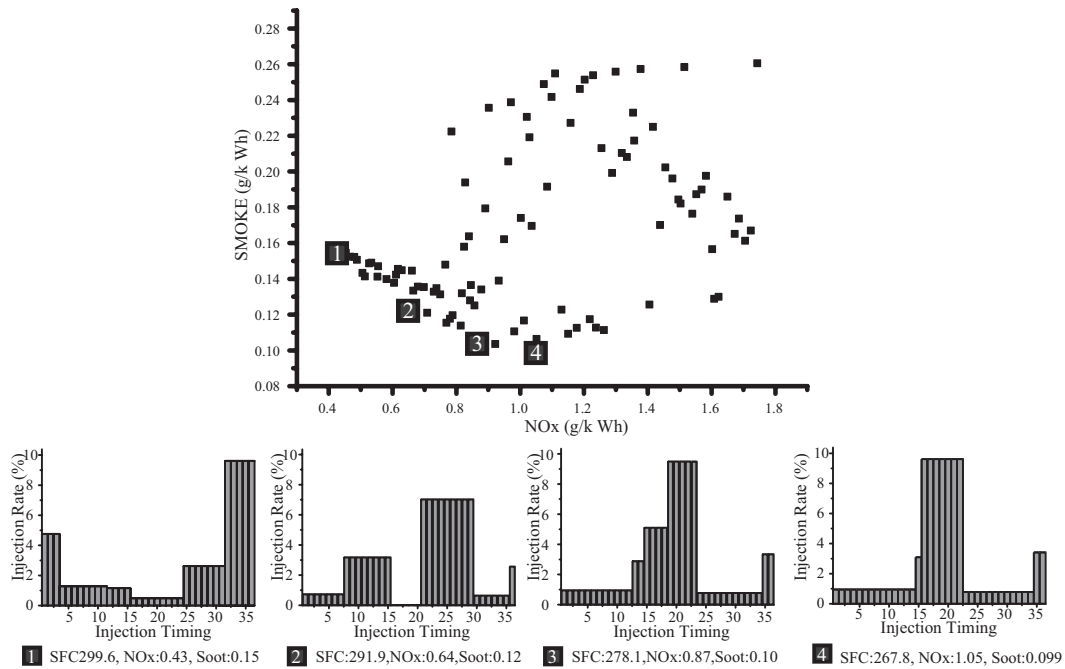


図 7.6 Derived non-dominated solutions(NOx,Soot)

2次元平面に投影した非劣解集合図 7.4, 図 7.5, 図 7.6から, 燃費率 (Specific Fuel Consumption: SFC), NOx, すず (Soot) の各目的にはトレードオフの関係があることが分かる。

得られた結果のうち, 図 7.4に示される燃費率-NOxの関係について考察する。図 7.4における, 解1は燃費率における最小値を表しており, 解4はNOxにおける最小値を表している。燃費率最小である解1の噴射率形状では, 多段噴射において前半部分に集中して燃料を噴射する形状となっている。対して, NOx最小である解4の形状では, 前半部分と後半部分の2段階に分けて噴射する形状となっている。この2段階噴射はいわゆる Pilot 噴射として知られているものである。解1から解4の解候補の噴射率形状から, 徐々に燃費率最小の噴射率形状からNOx最小の噴射率形状に推移している様子が分かる。解2の噴射率形状では, 解1の燃費率最小の場合と比較して, 前半部分のごく最初の部分に集中していた燃料噴射が, やや全体へと分散しているのが分かる。また, 解3の噴射率形状では, ほぼ解4のNOx最小の場合と同じ前半部分と後半部分の2段階に分かれた形状となっている。

つぎに, 図 7.5に示される燃費率-すずの関係について考察する。図 7.4と同様, 図 7.5中の解1の噴射率形状は, 燃費率における最小値を表しており, 解4の解候補はすずにおける最小値を表している。解4のすず最小を表す噴射率形状は, NOx最小の噴射率形状(図 7.4の解4の噴射率形状)の場合とは反対に, 中盤の辺りに集中して噴射する形状となって

表 7.4 Calculation time

Total calculation time(sec)	11425 (About 3 hours)
The average execution time of one trial of HIDECS(sec)	11.86
The execution time of only master node running(sec)	525

いる。また、解1から解4の噴射率形状の推移について見た場合、徐々にすす最小の噴射率形状である中盤辺りに集中する噴射率形状となっている様子が分かる。特に、解2から解4にかけては、より中盤辺りに噴射が集中するように噴射率形状が変化しているのが分かる。

最後に、 NO_x -すすの関係について図 7.6 から考察する。前述の通り、図 7.6 における解1の NO_x 最小の噴射率形状と解4のすす最小の噴射率形状は全く正反対の性質を持っている。このことは、解1から解4の噴射率形状の推移にも現れている。一般に、 NO_x とすすはトレードオフの関係であることが知られているが、本実験においても図 7.6 に見られるように、トレードオフの関係があることが分かった。

このように、NCGA を用いたディーゼルエンジン噴射スケジュールの多目的最適化では、1 度の探索により非常に多様な非劣解集合を得ることができた。特に、各目的の最小値を示す非劣解だけでなく各目的のバランスのとれた非劣解も得ることができた。一方、各目的の最小値を示す非劣解の噴射率形状が、これまで経験的に得られた噴射率形状と非常に類似していることから、得られた解集合の妥当性は高いといえる。よって、ディーゼルエンジン噴射スケジュールにおいて NCGA は有効な手法であるといえる。

計算時間

最後に、計算時間について付記する。本実験では、最適化手法として並列 NCGA を用いている。並列 NCGA は、マスタースレーブモデルであり、解析部分の HIDECS のみを並列で処理している。本実験にかかった計算時間を表 7.4 に示す。

表 7.4 におけるのマスターのみで動いていた時間とは、NCGA における並列化されていない部分を表している。NCGA では、マスターのみの操作として非劣解の生成などの操作が行われているが、全体の計算時間に対するマスターのみの処理に要した計算時間は、5%程度であることから、問題とならないことが分かった。つまり、NCGA を用いた全計算のうち、95%以上が並列化されていた。また、総計算時間が約3時間程度であることから、十分に現実的な計算時間内に結果が得られていることが分かる。これは、比較的計算負荷の軽い HIDECS を用いたためである。このことより、繰り返し計算がある程度必要となる

GA と計算負荷の軽い HIDECS を用いることにより、現実的な時間内に良好な解を得ることができることが分かった。

7.2.3 結論

本実験では、ディーゼル燃焼改善のためのディーゼルエンジン噴射スケジュールの多目的最適化を行った。シミュレーターとしては、現象論的モデルの 1 つであるディーゼルエンジンシミュレータの HIDECS を用いた。また、最適化アルゴリズムとしては、NCGA をマスタースレーブ型とした並列 NCGA を用いた。実験条件として、多段階噴射可能なディーゼルエンジンを想定し、噴射率を最適化することにより、燃費率、NO_x、すすの最小化を試みた。その結果、以下の点が明らかとなった。

- 多段階噴射による、燃費率、NO_x、すすの制御の可能性
- NCGA のディーゼルエンジン噴射スケジュールに対する有用性
- 燃費率、NO_x、すすのトレードオフの関係
- 燃費率、NO_x、すすの各噴射率特性
- 多目的 GA において HIDECS を用いる必要性

今後の課題を以下に示す。

- より実用的な多段噴射の設定
- 噴射開始時期、噴射期間などをパラメータとして考慮

7.3 矩形ブロック最小面積配置の多目的最適化

本節では、複数の矩形ブロックを最小面積で配置する矩形パッキング問題に対する NCGA の適用を試みる。この矩形パッキング問題は、超大規模集積におけるチップ面積最小化を目的とするフロアプランニング⁴⁸⁾、プラントなど設備や職場の配置など幅広い分野に応用されている⁴⁹⁾。矩形パッキング問題は、組み合わせ最適化問題の 1 つであり、扱うブロック数によって可能な組み合わせが指数的に増加するという特徴を持っている。

ここでは、この矩形パッキング問題を、最終的に得られた配置図形の縦と横の長さをそれぞれ目的にとり、2 目的最適化問題として定式化を行う。これは、面積の最小化だけでなく、実際に各ブロックを配置する図形のアスペクト比を解選考者が選択できるようにするためである。また、ブロック配置表現としてシーケンスペア⁵⁰⁾を、交叉法として PPEX⁵¹⁾を採用している。

数値計算例では、多目的矩形パッキング問題に対して提案手法、近傍培養型 GA (Neighborhood Cultivation Genetic Algorithm : NCGA)⁵²⁾ の適用を試み、SPEA2, NSGA-II との比較を通じて本手法の有効性の検証を行った。

7.3.1 矩形パッキング問題

矩形パッキング問題は、超大規模集積 (LSI) におけるチップ面積最小化を目的とするフロアプランニング^{48, 50)} プラントなど設備や職場の配置など幅広い分野に応用されている。⁴⁹⁾ 矩形パッキング問題は、組み合わせ最適化問題の 1 つであり、扱うブロック数によって可能な組み合わせが指数的に増加するという特徴を持っている。

また、本実験では、この矩形パッキング問題を単純に面積最小とするのではなく、最終的に得られた配置図形の縦と横の長さをそれぞれ目的にとり、多目的最適化問題として扱った。これは、実際に各ブロックを配置する図形のアスペクト比を解選考者が選択できるようにするためである。以下、問題の設定に大きく関わってくるデータ構造の表現、および問題の定式化について説明する。

データ構造表現

矩形パッキング問題において最も重要な問題の一つは解の表現方法、すなわちデータの構造表現である。矩形パッキング問題では、このデータ構造の表現によりブロックの配置方法が決まるだけでなく、実際の探索空間の規模、最終的に得られる解の精度といった問題の本質的な部分までが決定する。

この点に関して、従来より様々な方法が提案されてきたが^{49, 50, 53)}、LSI のフロアプランニングの分野において開発されたシーケンスペア (Sequence-Pair)⁵⁰⁾、BSG⁵³⁾ などの登場により有限な解空間において最適なパッキングを求めることが可能となった。

ここでは、BSG に比べより効率的な解探索を実現することのできるシーケンスペアを採用した。以下、具体的なシーケンスペアの仕組みについて述べる。

シーケンスペア シーケンスペアは配置の対象となるブロック名から構成される 2 つの順列 (Γ_-, Γ_+) の並びに基づいて方形配置を表現する方法である。各ブロック名は (Γ_-, Γ_+) の順列に基づいて任意の 2 つの方形の相対的な位置関係が指定される。図 7.7 にシーケンスペアの概念図を示す。

図 7.7(a) は、任意の 2 つのブロック X, Y の (Γ_-, Γ_+) に関する順列とそれに対応するブロック X, Y の位置関係を表している。つまり、ブロック順列 $(\Gamma_-, \Gamma_+) = (XY, XY)$ は、 Γ_- および Γ_+ に関して X が Y よりも先に並んでおり、このような順序関係の場合には X は Y の左にあるということを意味している。基本的に、 Γ_- 軸がブロック間の上下関係、

Γ_+ 軸がブロック間の左右関係を表している．すなわち， Γ_- は下から上への配置， Γ_+ は左から右への配置を決定している．図 7.7(b) は同図 (a) を視覚的に示している．

また，シーケンスペアは最適性が保証されているため，任意の系列対 (Γ_-, Γ_+) に対してこれらの位置関係を満たす方形パッキングは必ず存在する.⁵⁰⁾

ブロック数 6 の場合でのシーケンスペアの例を図 7.8 に示す．図 7.8(c) の個体例は， (Γ_-, Γ_+) の順列および各ブロックの向き θ を表している．なお，本実験では，配置する各ブロックの向きは縦・横のどちらかに限定するものとする．この図 7.8(c) の (Γ_-, Γ_+) 順列よりブロック間の相対位置を表す図 7.8(b) を作成する．相対位置を表す図 7.8(b) から，図 7.7

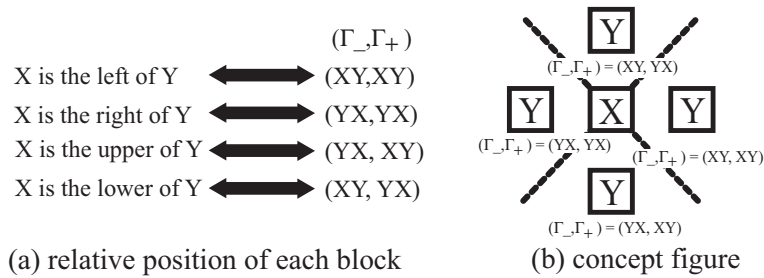


図 7.7 Concept of sequence-pair

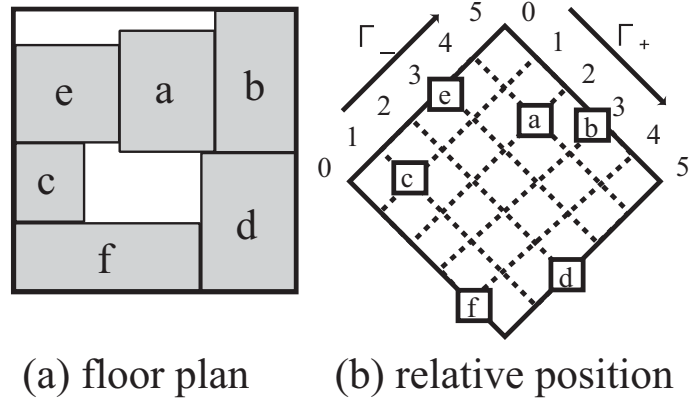


図 7.8 Coding example of sequence-pair

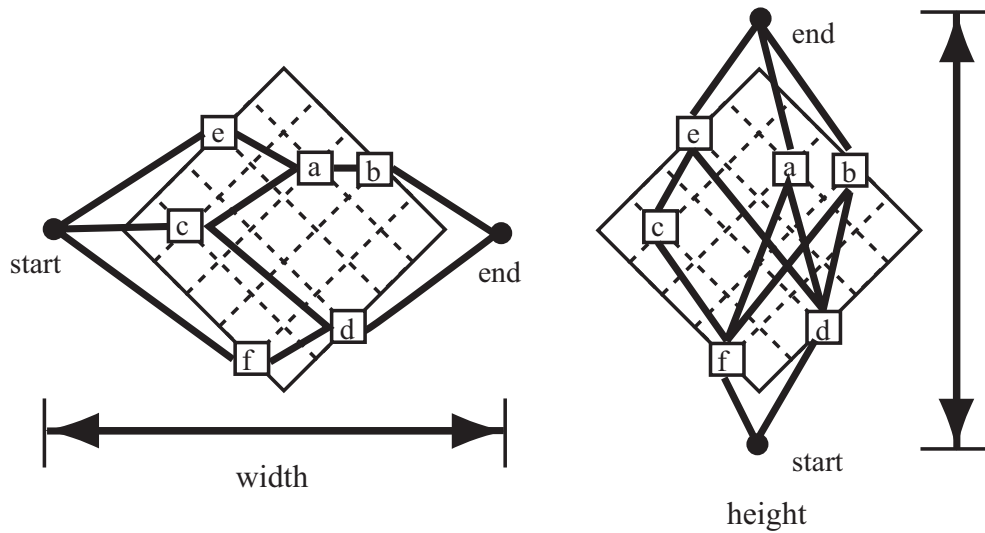


図 7.9 Horizontal/Vertical Constrain graphs

に基づく全ブロックの相対位置を求め、垂直、水平制約グラフ図 7.9 を作成することができる。垂直、水平制約グラフでは、それぞれのブロックの幅、高さが辺の重みとして与えられ、始点から終点までの最長経路を求める。それぞれの最長経路上の辺の重みの総和がチップ幅、チップ高さに相当する。これにより、各ブロックの配置図 7.8(a) を求めることができる。

問題の定式化

数値実験において設定した目的は、以下の 2 つである。

$$\min f_1(x) = width \quad (7.1)$$

$$\min f_2(x) = height \quad (7.2)$$

なお、上式における $width$ は横幅、 $height$ は高さを意味するものとする。

上記 2 つの目的には明示的なトレードオフの関係があるものの、両目的に対する最適化が行われることにより、結果としてブロックの占める面積の最小化も実現されるものと考えられる。また、多目的化されているため解の選考者はパッキング面積の縦横比を自由に選択することができる。

ブロック配置後の全領域の横幅、高さは、上述の垂直、水平制約グラフ図 7.9 から求めることができる。

7.3.2 数値実験

本論文では、上記で説明した矩形パッキング問題に対する NCGA の有効性の検証を行った。また、NCGA の比較対象として Zitler らに提案された SPEA2, Deb らに提案された NSGA-II, 近傍交叉を用いない NCGA(non-NCGA) を用いた。

GA 解法

交叉法 数値実験では、矩形パッキング問題に対する交叉手法として配置依存部分交換交叉 (Placement-based Partially Exchanging Crossover : PPEX) を用いた。⁵¹⁾ 本手法は、シーケンスペアの相対配置に基づく方法であり、配置上の近いブロックに関して局所的な交叉を行うことが可能となる。

PPEX のアルゴリズムを以下に示す。

Step 1: それぞれの親から乱数によりブロックを 1 個ずつ選択。

Step 2: 選択したブロックを中心に窓領域を生成し、窓に含まれるブロック集合を交叉対象 M_c とし、それ以外のブロック集合を M_{nc} とする。

M_c の各ブロックを Γ_-, Γ_+ で相手の親の順に入れ替え、子にコピーする。

Step 3: M_{nc} の各ブロックはそのまま子にコピーする。

上記における窓領域とは、シーケンスペアの相対配置を示す斜め格子⁶ 上の連続した部分領域を意味する。本論文では、この窓領域を正方形の部分領域として扱った。

図 7.10 に PPEX が窓領域の一辺を 4 とした場合の例を示す。この例では、親 1 でランダムに生成された窓領域に含まれるブロック a, b が交叉対象ブロック (M_c) となる。交叉前は、 $(\Gamma_-, \Gamma_+) = (da, da)$ の順であるが、交叉後には親 2 の順 $(\Gamma_-, \Gamma_+) = (ad, ad)$ に変更している様子が示されている。また、交叉対象ブロック以外 (M_{nc}) については、親 1 から子 1 へそのままコピーされているのが分かる。なお、向きについても同様に相手の親の向きに変更するものとする。

突然変異 突然変異の手法として、ランダムに選ばれたブロックの向きを逆に変換するという方法を用いた。

例題

我々は、ブロック数の異なる 4 種類 (33, 50, 100, 500 ブロック) の例題に対して NCGA を適用し、他手法との性能比較を行った。

⁶ オブリークグリッド (oblique grid), 図 7.8(b) を参照。

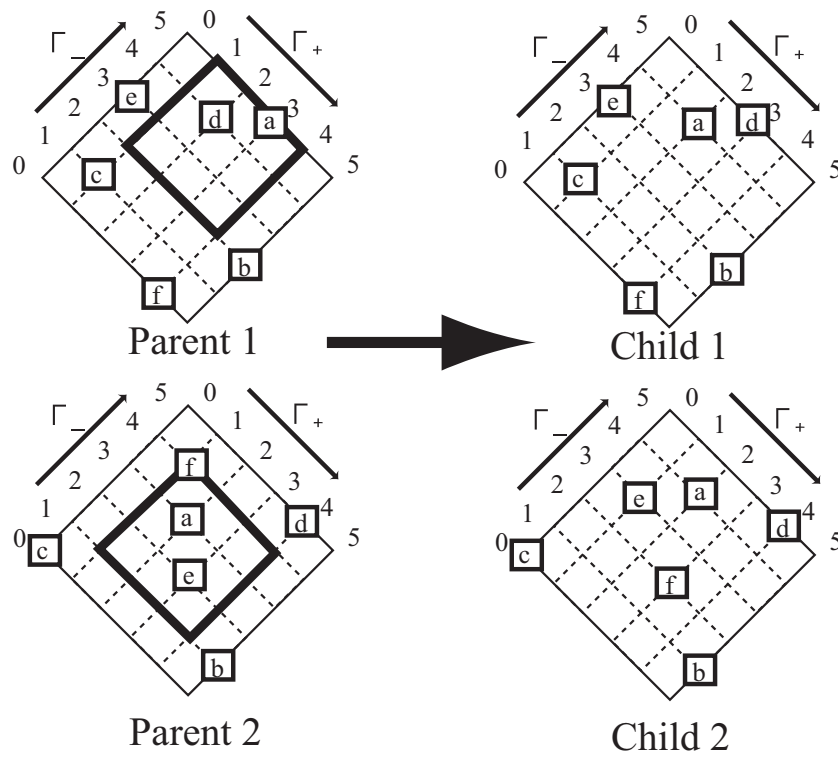


図 7.10 Placement-based Partially Exchanging Crossover(PPEX)

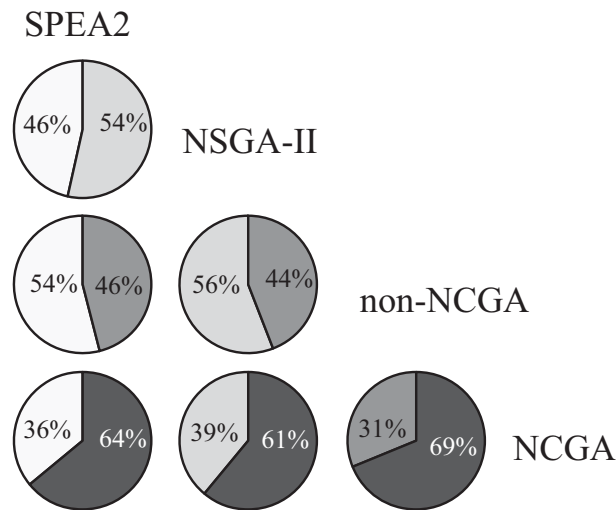
本実験で用いた手法は, Zitzler らに提案された SPEA2⁹⁾, Deb らに提案された NSGA-II²⁾, 本論文で提案している NCGA の 3 手法に近傍交叉を行わない non-NCGA(non-NCGA)を加えた 4 手法である.

GA パラメータ

本実験で用いた GA のパラメータを表 7.5 に示す.

表 7.5 GA Parameter

number of blocks	33, 50, 100, 500
population size	200
crossover rate	1.0
mutation rate	1/bit length
terminal generation	400
number of trial	30

図 7.11 Results of I_{LI} (33 blocks)

結果

我々は、ブロック数 33, 50, 100, 500 の 4 つの矩形パッキング問題に対する実験を行った。用いた問題のうち、ブロック数 33 の問題は VLSI レイアウト CAD 用の MCNC ベンチマーク ami33 のブロックサイズを用いている。また、ブロック数 50 の問題は 1 から 50 の乱数を用いて、ブロック数 100 の問題は 1 から 50 の乱数を用いて、ブロック数 500 の問題は 1 から 500 までの乱数を用いてブロック数サイズを決定した。

これらの例題に対して、NCGA, SPEA2, NSGA-2, non-NCGA の 4 つの手法をそれぞれ 30 試行行った。non-NCGA と NCGA を比較することにより、近傍交叉の影響を明確にすることができる。

ブロック数 33 の場合

ブロック数 33 における結果の内、 I_{LI} を図 7.11 に、 I_{MMA} を図 7.12 に示す。また、各手法により得られた非優越解集合のプロット図を図 7.13 に示す。なお、図 7.13 におけるプロット図では各試行毎の非優越解集合がまとめてプロットされている。

図 7.11 および図 7.13 から分かるように、NCGA はどの手法と比較しても、よりパレート最適解に近い範囲でかつ幅広い解を得ているのが分かる。解の幅広さについて見た場合、図 7.12 から NCGA, non-NCGA がより幅広い解領域に非優越解が分布しているのが分かる。しかし、non-NCGA は他手法と比べパレート最適解への近さという面において劣っていることがわかる。これは、幅広い解を得ているにも関わらず図 7.11 の I_{LI} において、non-NCGA

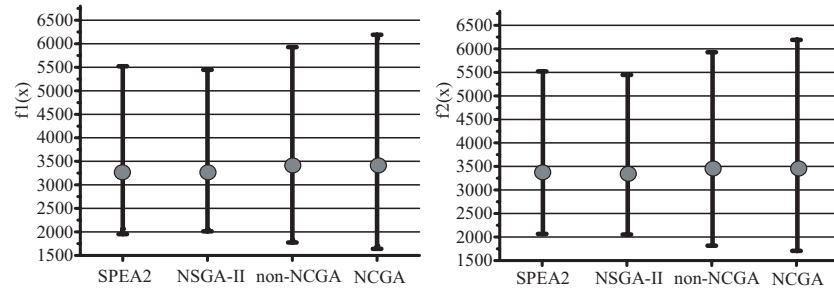
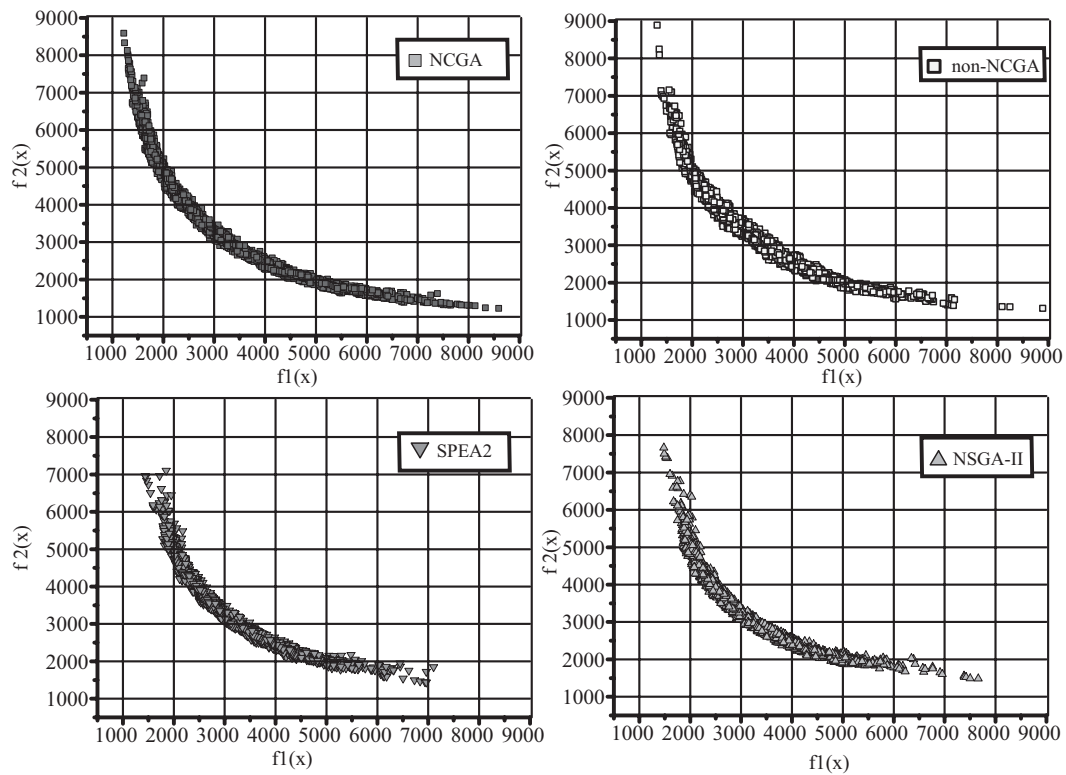
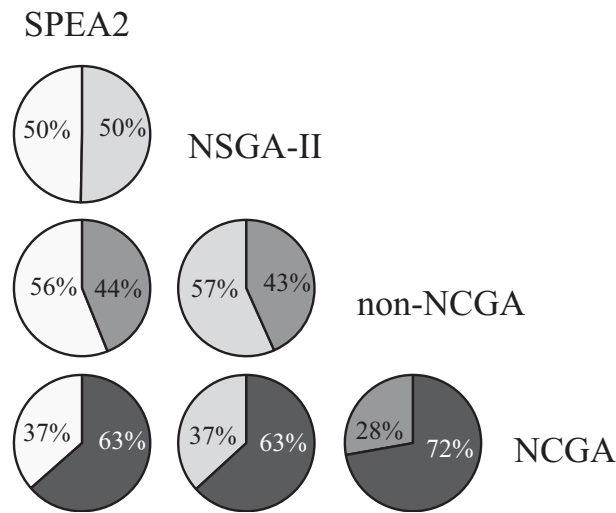
図 7.12 I_{MMA} of 33 blocks

図 7.13 Derived non-dominated solutions(33 blocks)

図 7.14 Results of I_{LI} (50 modules)

の値がどの手法よりも劣っていることから判断することができる。

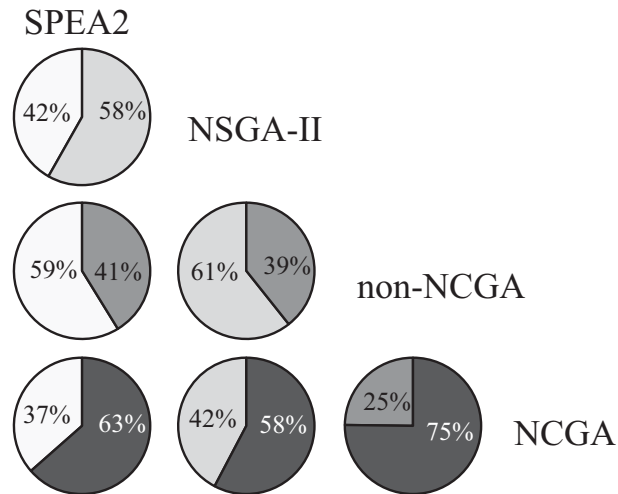
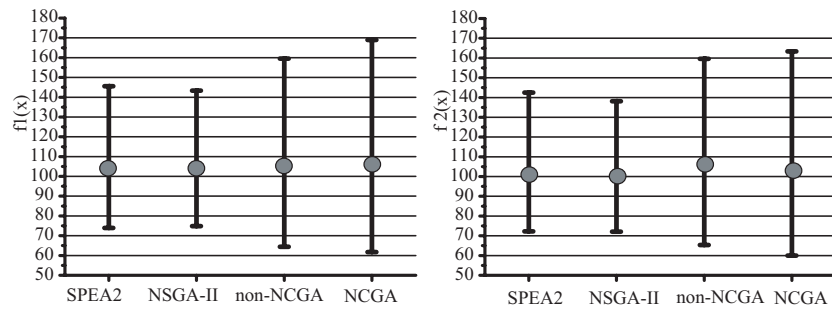
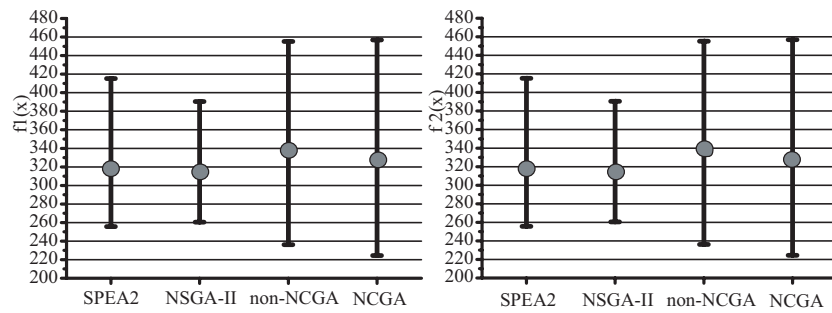
このことより、NCGA における近傍交叉は矩形パッキング問題においても非常に探索に効果的であることが分かる。

50,100 ブロック数の場合

ブロック数 50 および 100 の場合における結果について考察する。50 および 100 の場合における、 I_{LI} の結果を図 7.14, 図 7.15 に示す。また、 I_{MMA} の結果を図 7.16, 図 7.17 に示す。両問題における各手法により得られた非劣解を図 7.18, 図 7.19 に示す。

図 7.14 - 7.19 から、ブロック数 50 および 100 の場合の結果は、ブロック数 33 の場合における結果とほぼ同じ傾向を持っていることが分かる。すなわち、これらの問題においても NCGA は、他の手法よりもよりパレート最適解に近い範囲に幅広く分布している。このことは、プロット図よりも図 7.14 - 7.17 の I_{LI} , I_{MMA} の値において顕著に示されている。

しかし、先ほどのブロック数 33 の非優越プロット図 7.13 に比べブロック数 100 の図 7.19 では、非優越フロントの線幅が全ての手法において太くなっており、各試行毎のばらつきが大きくなっていることが分かる。これは、ブロック数 33 の場合に比べ、ブロック数 100 の場合には問題の難易度が高くなっているためであると思われる。

図 7.15 Results of I_{LI} (100 modules)図 7.16 I_{MMA} of 50 modules図 7.17 I_{MMA} of 100 modules

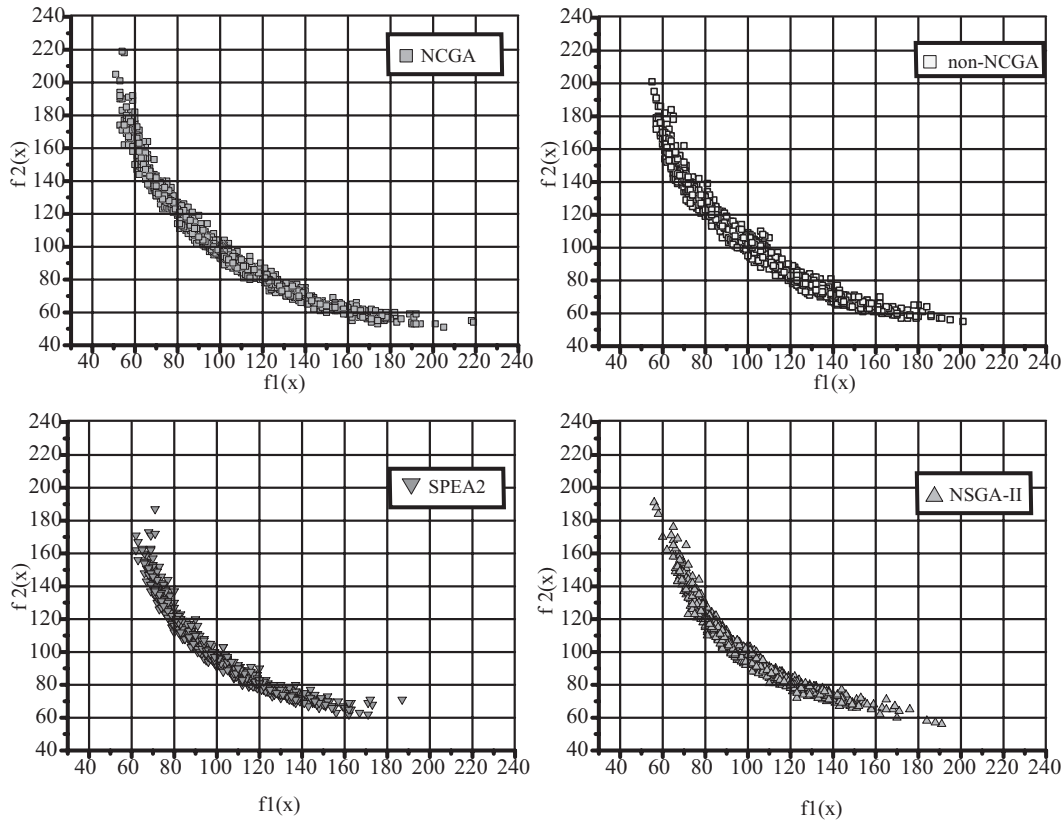


図 7.18 Derived non-dominated solutions(50 modules)

ブロック数 500 の場合

ブロック数 500 の場合における結果をそれぞれ, I_{LI} について図 7.20 に, I_{MMA} の結果を図 7.21 に示す. 各アルゴリズムの非優越フロントについても, 図 7.22 に示す.

先ほどのブロック数 33 の問題における結果と比べ, 図 7.20 において I_{LI} (NCGA, SPEA2) と I_{LI} (NCGA, NSGA-II) の値が, (50%, 50%) に近くなっていることが分かる. この I_{LI} は, パレート解への近さ, 幅広さという解の質そのものを表している. そのことから, これらの結果は, 両者の差がなくなっているということを意味しているのが分かる.

また, 図 7.22 においては, 以下の点においてこれまでの問題と異なっている.

- 1) SPEA2 と NSGA-II の非優越解がパレート最適フロントのある一部に固まっている.
- 2) パレート最適フロントの中心部において, SPEA2 と NSGA-II の非優越解は NCGA のそれを優越している.

ブロック数 500 の問題は, ブロック数 33 の問題と比べ難易度が非常に高くなる. これは, ブロック数 500 の場合における可能な順列の総組み合わせ数が, ブロック数 33 の場合と比較して 10^{2000} 倍以上となるためである. このような膨大な総組み合わせ数の問題に対して,

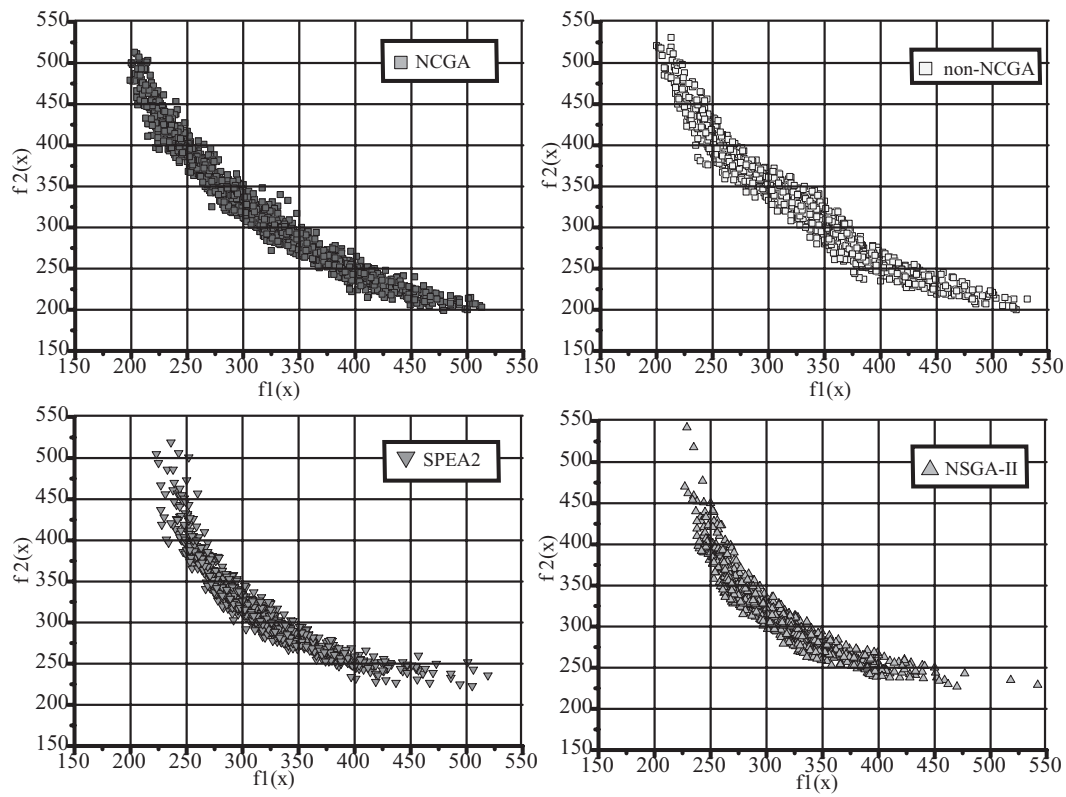
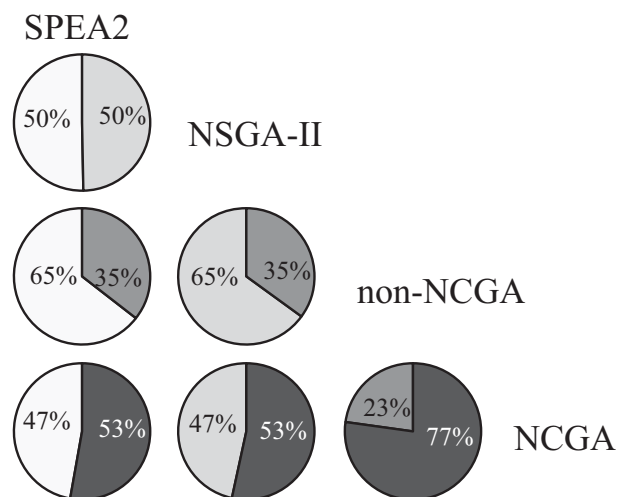


図 7.19 Derived non-dominated solutions(100 modules)

図 7.20 Results of I_{LI} (500 blocks)

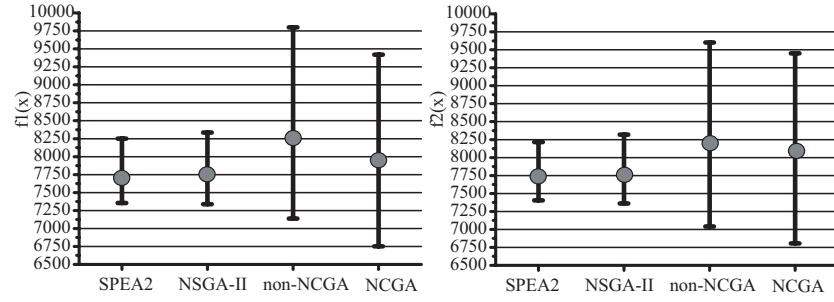
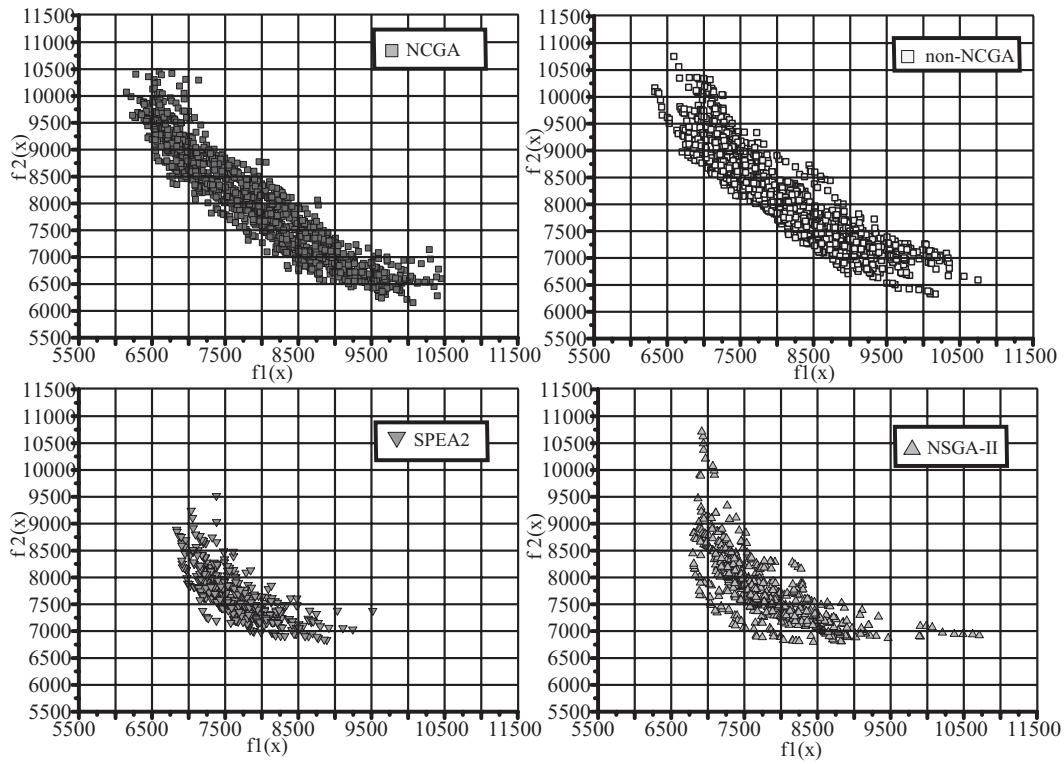
図 7.21 I_{MMA} of 500 blocks

図 7.22 Derived non-dominated solutions(500 blocks)

今回用いた個体の評価回数では十分な探索を行うことはできない。

得られた結果から、このような問題において、SPEA2, NSGA-IIは局所的な探索を行うようになり、より幅広い解を求めようとする NCGA, non-NCGA よりも、ある一部分においては良好な値が出やすくなっていることが分かる。

しかし、特筆すべき点は、このような困難な問題においても NCGA, non-NCGA は非常に高い多様性を保っている点である。特に、NCGA は一部の非優越フロントにおいて SPEA2, NSGA-II に優越されているものの、相対的には同等以上の解を得ている。

パッキング図

ここで、NCGA により得られた解の実際のレイアウト図について考察を行う。上述の実験結果より、ほぼすべての場合において NCGA の解が最も良好であることが示された。そこで、本論文ではブロック数 33 の場合、ブロック数 100 の 2 つの場合において NCGA が得た非優越解のレイアウト図の妥当性について検討を行った。

最終的に解を選択する選考者は、面積-アスペクト比の関係から解を選択すると思われる。そこで、非劣解集合のプロット図として、面積の縦幅-横幅をプロットの軸にとるのではなく、面積-アスペクト比を軸として用いた。以下、ブロック数 33, 100 の場合における、NCGA の面積-アスペクト比を軸としたプロット図とプロット図の中から選んだ幾つかのプロット点に対応したレイアウト図をそれぞれ図 7.23, 7.24 に示す。

ブロック数 33 のレイアウト結果である図 7.23 から、ほぼ全ての解が隙間の少ない妥当なレイアウトであることが分かる。また、アスペクト比が 0 から 1 の範囲まで幅広い解が得られているのが分かる。

図 7.23 の各レイアウト図について見てみると、アスペクト比の大小に関わらず、“A”から“E”における各レイアウトの面積がほぼ一定である。このことより、アスペクト比と配置図の空白(レイアウトの総面積)には強い因果関係はないことが分かる。また、“A”から“E”のうちアスペクト比の軸における両端である“A”および“E”において、若干の空白が見られるが、これは、高さもしくは幅に関する最適なレイアウトに近づくほど空白が多くなりやすくなるためであると思われる。しかし、“A”および“E”を含む各レイアウトは、視覚的にほぼ最適なレイアウトに近く、十分妥当なレイアウトであると言える。

一方、ブロック数 100 のレイアウトである図 7.24 では、ブロック数 33 の場合と比べ全体的に空白が多いことが分かる。このことより、NCGA により得られた解がパレート最適解から多少離れていることが分かる。

これは、矩形パッキング問題の特性の 1 つであるブロック数の増加に伴う可能組み合わせ数の指数的増加が原因である。ブロック数 100 の矩形パッキング問題をシーケンスペアを用いて解く場合、可能な順列の総組み合わせ数は、 $100! \times 100! \times 2^{100} (> 1.0 \times 10^{339})$ となる。

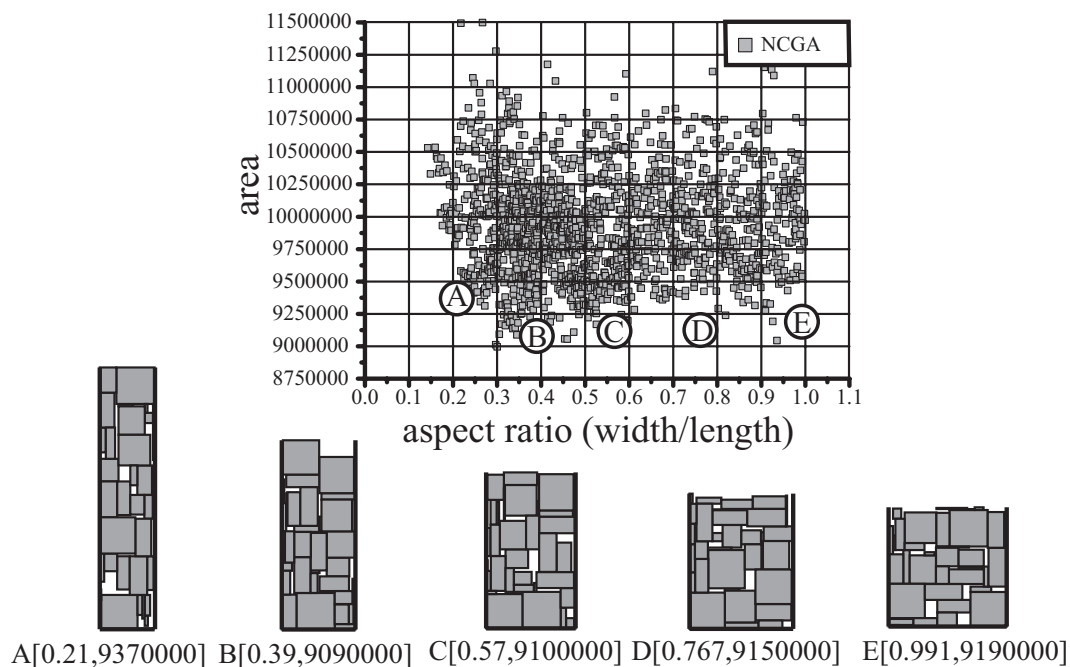


図 7.23 The placement of the modules(33 modules)

本論文では、全ての問題において 8000 評価回数 (200 個体, 400 世代) しか行っていないため、この問題規模に対しては十分な探索が行えなかったものと思われる。

しかしながら、ほとんどの解において、最適なレイアウトではないものの、感覚的に大きく妥当性を欠いたものではない。特に、問題の組み合わせ規模と評価回数の関係을考えた場合、十分に良好な解が得られていると言える。

したがって、これら 2 つの結果より NCGA により得られた解は、最適なレイアウトではないものの、良好なレイアウト結果を示していると言える。

7.3.3 結論

本論文では、職場配置問題や LSI 設計問題で知られている矩形パッキング最適化問題の多目的化を行い、この問題に対して NCGA の適用を行った。

数値実験として、ブロック数 33, 50, 100, 500 の 4 つの矩形パッキング問題に対する NCGA, 多目的 GA の代表的手法である SPEA2, NSGA-II の結果を比較、考察した。また、近傍交叉による矩形パッキング問題への探索の有効性を検証するために近傍交叉を行わない non-NCGA と NCGA の比較についても検討した。

数値実験より、以下の点が明らかとなった。

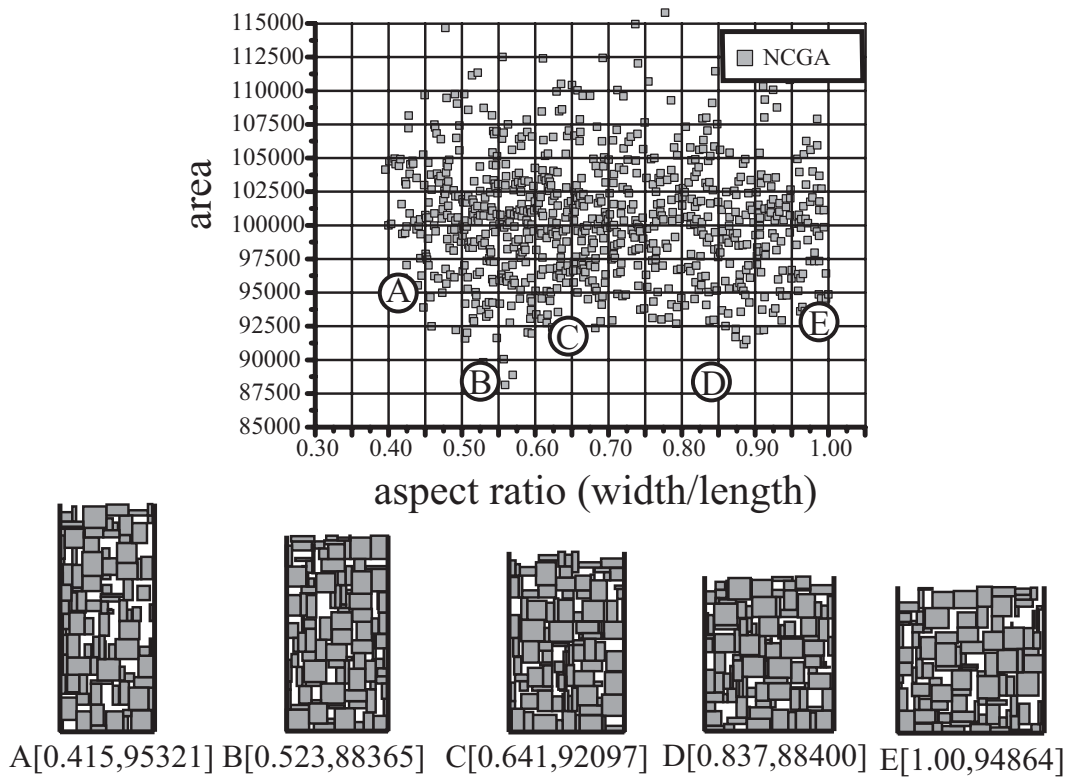


図 7.24 The placement of the modules(100 modules)

- 全てのブロック数の場合において、NCGA は、他の手法と比較して良好な結果が得ることができた。このことより、NCGA は多目的矩形パッキング問題において有効な手法であると言える。
- 近傍交叉を行った場合と行わない場合の NCGA を比較した結果、全ての問題において前者が非常に良好な結果を示した。よって、近傍交叉は多目的矩形パッキング問題の探索においても有効であると言える。
- ブロック数の多い、より複雑な問題において SPEA2 や NSGA-II ではパレートフロントの中心に解が集中する傾向が見られたが、NCGA および non-NCGA では多様な解を得ることができた。このことより、多目的矩形パッキング問題において、NCGA における多様性保持の効果は問題の規模に依存しないことが分かった。
- NCGA において得られた解の実際のレイアウト図について見た結果、得られた解は視覚的に見ておおよそ妥当なものであることが分かった。そのため、NCGA は多目的矩形パッキング問題に対して良好な結果を得ることができると言える。

また、今回採用した問題の定式化およびパッキングの表現方法、交叉などの有用性につ

いての検討は今後の課題である。

7.4 まとめ

本章では、6章において提案された近傍培養型 GA(Neighborhood Cultivation GA: NCGA) の実問題に近い問題に対する有効性の検証として、以下の問題に対する NCGA の適用を試みた。

- i) ディーゼルエンジン噴射スケジュールの最適化
- ii) 矩形ブロックの 2 次元空間上での配置面積最小化

ディーゼルエンジン噴射スケジュールでは、シミュレーターとして現象論的モデルの 1 つであるディーゼルエンジンシミュレータの HIDECS を用いて燃費率、NO_x、すすの最小化を試みた。ディーゼルエンジンシミュレータとしては負荷の軽い HIDECS であるが、1 試行あたり平均 10 秒程度の計算負荷が必要となる。そのため、NCGA をマスタースレーブ型とした並列 NCGA を用いて最適化を行った。この結果、約 3 時間程度の総計算時間で良好な非劣解集合を得ることができた。得られた非劣解集合から、燃費率、NO_x、すすの噴射率特性および、それぞれのトレードオフの関係を見つけることができた。

また、矩形ブロックの配置面積最小化では、配置図形の縦と横の長さをそれぞれ目的とする 2 目的問題として最適化を試みた。ブロック配置表現としては、シーケンスペア⁵⁰⁾を、交叉法として PPEX を用いた。扱うブロック数の異なる 4 つの問題に対して実験を行い、NCGA における結果と SPEA2, NSGA-II, 近傍交叉を行わない non-NCGA における結果との比較を行った。比較実験の結果、NCGA は全ての問題において、どの手法と比較しても良好な結果を得ることができた。このことから、NCGA は多目的矩形パッキング問題において有効な手法であることが確認することができた。また、NCGA により得られた解の実際のレイアウト図について見た結果、得られた解は視覚的に見ておおよそ妥当なものであることが分かった。そのことから、NCGA は多目的矩形パッキング問題に対して良好な結果が得られることを確認できた。

以上のように、NCGA は 2 つの特徴の異なる問題に対して良好な結果を得ることができた。そのため、NCGA は数学的な連続テスト関数および多目的ナップザック問題だけでなく、実際の実問題に対しても有効な手法であると思われる。

第 8 章

結論

8.1 本論文の成果

本論文では、多目的 GA における高実用性を目指した新たなアルゴリズムとして、近傍培養型 GA (Neighborhood Cultivation GA :NCGA) の提案を行い、その有用性の検証を行った。NCGA は、これまでに提案されてきたアルゴリズムの探索に有効なメカニズムと、近傍交叉という独自のメカニズムを合わせ持った新たなアルゴリズムである。NCGA の近傍交叉では、目的関数空間において隣り合う 2 つの個体を用いて交叉を行う。一般に、大域的な探索を行う多目的 GA では探索個体同士の目的関数空間距離が大きく離れ、効果的な交叉を行うことができない。しかし、近傍交叉を行うことにより、意味のない交叉を未然に防ぐことができ、結果として探索効率の向上を実現することができる。

NCGA の有効性を検証するために、幾つかのテスト関数を用いて Deb らにより提案された NSGA-II および Zitzler らにより提案された SPEA2 といった代表的な手法、近傍交叉を行わない NCGA との比較実験を行った。また、NCGA におけるソート基準として設計変数空間を用いた場合と目的関数空間を用いた場合の検討も行った。これらの数値実験より得られた結論を以下に示す。

- (1) ほとんどの問題において、NCGA は他の手法と比較して良好な結果が得られた。これは、探索個体群の多様性と近傍交叉による効果であると思われる。このことより、NCGA は他の優れた多目的 GA 手法と比較して少なくとも同等以上の性能を持っていることが分かった。
- (2) NCGA と近傍交叉を行わない non-NCGA の比較の結果、全ての問題において NCGA が優れた結果を得た。そのことから、近傍交叉は多目的 GA の探索において効果的であることが分かった。特に、設計変数の数が多くより探索の難しい問題において近傍交叉の効果が強く認められた。

(3) NCGA におけるソート基準として設計変数空間を用いた場合と目的関数空間を用いた場合の比較実験を行った結果、後者の方法が全ての例題に対して良好な結果を示した。これは、対象とした問題における目的関数の数と設計変数の数が大きく異なっていたためである。また、各変数の持つ最適な値の組み合わせを求めるような問題において、設計変数空間を基準とした NCGA では、良好な結果を得ることができず、逆に目的関数空間を基準とした NCGA では、効果的な交叉を実現していた。このことから、NCGA のソート基準としては、目的関数空間の方が良好であることが分かった。

また、2つの性質の異なるより実問題に近い対象問題に対して NCGA を適用し、NCGA の高実用性に関する検討を行った。対象とした問題は以下の 2 つである。

i) ディーゼルエンジン噴射スケジュールの最適化

ディーゼルエンジン噴射スケジュール最適化問題では、燃費率、NO_x 排出、すす排出の最小を目的とする 3 目的最適化問題とした最適化を行った。シミュレータとしては、現象論的モデルの 1 つであるディーゼルエンジンシミュレータの HIDECS を使用し、最適化アルゴリズムとしては、NCGA をマスタースレーブ型とした並列 NCGA を用いた。この結果、約 3 時間程度という現実的な時間内で良好な非劣解集合を得ることができた。また、得られた非劣解集合から、燃費率、NO_x、すすの噴射率特性および、それぞれのトレードオフの関係を見つけることができた。

ii) 矩形ブロックの 2 次元空間上での配置面積の最小化

矩形ブロックの配置面積最小化では、配置図形の縦と横の長さをそれぞれ目的とする 2 目的問題として最適化を試みた。ブロック配置表現としては、シーケンスペア⁵⁰⁾を、交叉法として PPEX を用いた。扱うブロック数の異なる 4 つの問題に対して実験を行い、NCGA における結果と SPEA2、NSGA-II、近傍交叉を行わない non-NCGA における結果との比較を行った。比較実験の結果、NCGA は全ての問題において、どの手法と比較しても良好な結果を得ることができた。このことから、NCGA は多目的矩形パッキング問題において有効な手法であることが確認することができた。また、NCGA により得られた解の実際のレイアウト図について見た結果、得られた解は視覚的に見ておおよそ妥当なものであることが分かった。そのことから、NCGA は多目的矩形パッキング問題に対して良好な結果が得られることを確認できた。

これらの結果から、NCGA は数学的な連続テスト関数および多目的ナップザック問題だけでなく、実問題に対しても良好な結果が得られることが確認された。

8.2 今後の課題

本論文の今後の発展として、次のことが挙げられる。

i) ディーゼルエンジン噴射スケジュール問題のさらなる検討

より実用的な多段噴射の設定を行うように、扱う設計変数のコーディング方法の再検討を行う必要がある。また、噴射開始時期、噴射期間などを新たにパラメータとして考慮する最適化へ拡張する必要がある。

ii) LSI のモジュール配置、配線問題に対する NCGA の適用。

本論文では、矩形ブロックの 2 次元空間上での配置面積の最小化を行っていた。この問題は、同じ矩形ブロックを最適に配置する LSI のモジュール配置、配線問題への応用が可能である。矩形ブロック配置面積の最小化において NCGA が良好な結果を得られたことから、LSI のモジュール配置、配線問題においても NCGA は有効であると思われる。

iii) その他、様々な実問題への NCGA の適用。

上記以外の対象として、フローショップスケジュール、ジョブショップスケジューリングなどのスケジュール組み合わせ最適化問題への NCGA の応用を行う必要がある。

謝辞

本研究を遂行するにあたり，多大なる御指導，そしてご協力を頂きました同志社大学 工学部 知識工学科 三木光範教授，廣安知之講師に心より感謝いたします。

また，本研究を遂行するに辺り助言を頂いた，大阪大学大学院 情報科学研究科 助手 坂主 圭史さん，上智大 理工学部 機械工学科 管理工学講座 助手 白井 裕さんに深く感謝します。

さらに，本研究の実施にあたり惜しみない協力をしてくれた同志社大学大学院 工学研究科 知識工学専攻 知的システムデザイン研究室 近藤 健史君，奥田 環君，上浦 二郎君に本当に感謝します。また，既に卒業された日本アクティ・システムズ（株）の岡田 靖男君にも感謝しています。そして，同志社大学 工学部 知識工学科 金 美和君にも協力して貰いました。ありがとう。

また，博士論文音読会に参加してくれた，青井君，花田君，澤田君，輪湖君，斉藤君，釘井君，米澤君，谷口君，米田君，上村君ありがとう。

そして，知的システムデザイン研究室の研究室の皆さん，1期生から6期生までの先輩後輩の皆さんには本当にお世話になりました。皆さんのおかげで5年間を非常に有意義に過ごすことができました。ここに感謝の意を表したいと思います。

最後に，5年間の研究室生活を含む8年間の学生生活を理解と寛容を持って見守り，経済的援助，精神的な支えとなった両親に感謝します。

参考文献

- [1] 坂和正敏. 離散システムの最適化. 森北出版, 2000.
- [2] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, UK : Wiley, 2001.
- [3] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of 1st International Conference on Genetic Algorithms and Their Applications*, pp. 93–100, 1985.
- [4] D. E. Goldberg. *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.
- [5] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th international coference on genetic algorithms*, pp. 416–423, 1993.
- [6] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp. 257–271, 1999.
- [7] M. Erickson, A. Mayer, and J. Horn. The Niche Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems. In *In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello and David Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, Springer-Verlag. Lecture Notes in Computer Science No. 1993*, pp. 681–695, 2000.
- [8] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *KanGAL report 200001, Indian Institute of Technology, Kanpur, India*, 2000.

- [9] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Performance of the Strength Pareto Evolutionary Algorithm. In *Technical Report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich*, 2001.
- [10] John H. Holland. *Adaptation In Natural and Artificial Systems*. University of Michigan Press, 1975.
- [11] D. E. Goldberg and K. Deb. A Comparison of Selection Schemes Used in Genetic Algorithms. In *Foundations of Genetic Algorithms 1 (FOGA-1)*, pp. 69–93, 1991.
- [12] 嘉数侑昇, 三上貞芳, 皆川雅章, 川上敬, 高取則彦, 鈴木恵二. 遺伝的アルゴリズムハンドブック. 森北出版, 1994.
- [13] 伊庭斉志. 遺伝的アルゴリズムの基礎. オーム社, 1994.
- [14] 北野宏明. 遺伝的アルゴリズム. 産業図書, 1993.
- [15] 北野宏明. 遺伝的アルゴリズム 2. 産業図書, 1995.
- [16] 北野宏明. 遺伝的アルゴリズム 3. 産業図書, 1997.
- [17] 北野宏明. 遺伝的アルゴリズム 4. 産業図書, 2000.
- [18] First International Conference on Evolutionary Multi-Criterion Optimization. <http://www.tik.ee.ethz.ch/emo/>.
- [19] Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms (ICGA'93)*, pp. 416–423, 1993.
- [20] N. Srinivas and Kalyanmoy Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, Vol. 2, No. 3, pp. 221–248, Fall 1994.
- [21] J. Horn, N. Nafpliotis, and D. E. Goldberg. A Niche Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Vol. 1, pp. 82–87, 1994.

- [22] 比屋根. 並列遺伝的アルゴリズムによる多目的最適化問題のパレート最適解集合の生成法と定量的評価法. 第9回自律分散システムシンポジウム, pp. 295–300, 1997.
- [23] B.R. Jones, W.A. Crossley, and A.S. Lyrintzi. Aerodynamic and Aeroacoustic Optimization of Airfoils via a Parallel Genetic Algorithm. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. 1–11, 1998.
- [24] D.Q. Vicini. Sub-population policies for a parallel multiobjective genetic algorithm with applications to wing design. In *Proceedings of International Conference on Systems, Man and Cybernetics*, pp. 3142–3147, 1998.
- [25] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, Vol. 8, No. 2, pp. 149–172, 2000.
- [26] K.C.Tan, T.H.Lee, and E.F.Khor. Incrementing Multi-objective Evolutionary Algorithms: Performance Studies and Comparisons. In *First International Conference on Evolutionary Multi-Criterion Optimization*, pp. 111–125, 2001.
- [27] Reiko Tanese. Distributed Genetic Algorithms. In *Proceedings of 3rd International Conference on Genetic Algorithms (ICGA'89)*, 1989.
- [28] Theodore C. Belding. The Distributed Genetic Algorithm Revisited. In *Proceedings of the 6th International Conference on Genetic Algorithms (ICGA'95)*, pp. 114–121, 1995.
- [29] L. Nang and K. Matsuo. A Survey on the Parallel Genetic Algorithms. *J.SICE*, Vol. 33, No. 6, pp. 500–509, 1994.
- [30] E. Cantu-Paz. Topologies, Migration rates and Multi-Population Parallel Genetic Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, Vol. 1, pp. 91–98, 1999.
- [31] T. Hiroyasu, M. Miki, and Y. Tanimura. Characteristics of Models of Parallel Genetic Algorithms on PC Cluster systems. In *Proceedings of 1st International Workshop on Cluster Computing (IWCC '99)*, 1999.

- [32] Y.Takai M.Munemoto and Y.Sato. An Efficient Migration Scheme for Subpopulation-Based Asynchronously Parallel Genetic Algorithms. In *in Proceedings of the Fifth International Conference on Genetic Algorithms*, p. 649, 1993.
- [33] 森泰樹, 山村雅幸, 小林重信. 遺伝的アルゴリズムによるポートフォリオ選択. 計測自動制御学会 第19回 知能システムシンポジウム, pp. 17–22, 1994.
- [34] 玉置, 森, 荒木. 遺伝アルゴリズムを用いたパレート最適解集合の生成法. 計測自動制御学会論文集, Vol. 31, No. 8, pp. 1185–1192, 1995.
- [35] 三村泰成, 廣安知之, 三木光範. 実数コーディングを用いた遺伝的アルゴリズムのトラス構造物への適用. 計算工学会講演論文集, 第4巻, pp. 457–460, 1999.
- [36] D. A. V. Veldhuizen and G. B. Lamont. Multiobjective Evolutionary Algorithm Test Suites. In *Proceedings of the 1999 ACM Symposium on Applied Computing*, pp. 351–357, 1999.
- [37] Frank Kursawe. A Variant of Evolution Strategies for Vector Optimization. In *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, Vol. 496, pp. 193–197, 1991.
- [38] K. Deb and T. Meyarivan. Constrained Test Problems for Multi-Objective Evolutionary Optimization. *KanGAL report 200005, Indian Institute of Technology, Kanpur, India*, 2000.
- [39] K. Deb. Construction of Test Problems for Multi-Objective Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 1999.
- [40] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. In *Evolutionary Computation*, Vol. 8(2), pp. 173–195, 2000.
- [41] I. Ono and S. Kobayashi. A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distributed Crossover. pp. 246–253, 1997.
- [42] H.Hiroyasu. Diesel Engine Combustion and its Modeling. *International Symposium on Diagnostics and Modeling of Combustion in Reciprocating Engines*.
- [43] M. P. Walsh. Global Trends in Diesel Emissions Regulation-A 2001 Up Date. *SAE Paper 2001-01-0183*, 2001.

- [44] T. V. Johnson. Diesel Emission Control in Review. *SAE Paper 2001-01-0184*, 2001.
- [45] P. K. Senecal and R. D. Reitz. Optimization of Diesel Engine Emissions and Fuel Efficiency using Genetic Algorithms and Computational Fluid Dynamics. *Eighth ICLASS, Pasadena, CA.*, 2000.
- [46] H. Imanishi, T. Yoshizaki, and H. Hiroyasu. Simulation Study of Effects of Injection Rate Profile and Air Entrainment Characteristics on D.I. Diesel Engine. *SAE Paper 962059*, 1996.
- [47] Kiva. <http://gnarly.lanl.gov/secondlevel/codes/kiva.htm>.
- [48] 野島隆志, 坂主圭史, 高橋篤司, 梶谷洋司. 配線可能性を保証する sequence-pair を用いた配置手法. 電子情報通信学会技術研究報告, 第 101 巻, pp. 59–65, 2001.
- [49] Y. Shirai and N. Matsumoto. Performance Evaluation of ES Type Genetic Algorithms for Solving Block Layout Problems with Floor Constraints. In *PTransaction of JSME (C)65-634*, pp. 296–304, 1999.
- [50] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani. VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair. In *IEEE Transactions on Computer Aided Design*, Vol. 15, pp. 1518–1524, 1996.
- [51] 中矢伸吾, 若林真一, 小出哲士. 適応的遺伝的アルゴリズムとシーケンスペアに基づくフロアプランニング手法. 情報処理学会研究報告, 第 99 巻, pp. 119–126, 1999.
- [52] 渡邊真也, 廣安知之, 三木光範. 近傍培養型遺伝的アルゴリズムによる多目的最適化. 情報処理学会論文誌「数理モデル化とその応用」へ投稿中.
- [53] S. Nakatake, H. Murata, K. Fujiyoshi, and Y. Kajitani. Module placement on bsg-structure and ic layout applications. In *Proc. of International Conference on Computer Aided Design '96*, pp. 484–491, 1996.

付録 A

研究業績

1. 学術論文

1. Tomoyuki HIROYASU, Mitsunori MIKI, Shinya WATANABE, “Distributed Genetic Algorithms with a New Sharing Approach in Multiobjective Optimization Problems”, IEEE Proceedings of the 1999 Congress on Evolutionary Computation, pp.69-76,1999
2. Tomoyuki HIROYASU, Mitsunori MIKI, Shinya WATANABE, “Divided Range Genetic Algorithms in Multiobjective Optimization Problems”, Proceedings of International Workshop on Emergent Synthesis, IWES’ 99, pp.57-66,1999
3. 廣安知之, 三木光範, 渡邊真也:“ 領域分散型多目的遺伝的アルゴリズム”, 情報処理学会論文誌「数理モデル化と応用」tom3 号, pp.79-89,2000
4. Shinya WATANABE, Tomoyuki HIROYASU, Mitsunori MIKI, “Parallel Evolutionary Multi-Criterion Optimization for Block Layout Problems”, The 2000 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA2000), pp.667-673, 2000
5. Tomoyuki HIROYASU, Mitsunori MIKI, Shinya WATANABE, “The New Model of Parallel Genetic Algorithm in Multi-Objective Optimization Problems - Divided Range Multi-Objective Genetic Algorithms“, IEEE Proceedings of the 2000 Congress on Evolutionary Computation, pp.333-340,2000
6. Shinya WATANABE,Tomoyuki Hiroyasu and Mitsunori Miki, “PARALLEL EVOLUTIONARY MULTI-CRITERION OPTIMIZATION FOR MOBILE TELECOMMUNICATION NETWORKS OPTIMIZATION”, Proceedings of the EUROGEN2001 Conference, Athens. Greece, September 19-21, pp.167-172,2001

7. Tamaki Okuda, Tomoyuki Hiroyasu, Mitsunori Miki, Shinya Watanabe, “DC-MOGA: Distributed Cooperation model of Multi-Objective Genetic Algorithm”, Advances in Nature-Inspired Computation: The PPSN VII Workshops, pp. 25-26, 2002
8. Shinya WATANABE, Tomoyuki Hiroyasu and Mitsunori Miki, “Neighborhood Cultivation Genetic Algorithm for Multi-Objective Optimization Problems”, SEAL’02, 2002 (採録決定)
9. 渡邊 真也, 廣安 知之, 三木 光範, “近傍培養型遺伝的アルゴリズムによる多目的最適化”, 情報処理学会論文誌「数理モデル化と応用」, 2002 (採録決定)
10. 渡邊 真也, 廣安 知之, 三木 光範, “多目的矩形パッキング問題に対する進化的手法の有効性”, 電子情報通信学会, 2002 (投稿中)
11. 廣安 知之, 上浦 二郎, 三木 光範, 渡邊 真也, “適応的重みを有する多目的最適化のための分散遺伝的アルゴリズム”, 情報処理学会論文誌「数理モデル化と応用」, 2002 (採録決定)
12. 廣安 博之, 廣安 知之, 三木 光範, 上浦 二郎, 渡邊 真也, “現象論モデルと遺伝的アルゴリズムによるディーゼルエンジン燃料噴射率の多目的最適化”, 自動車技術会, 2002 (投稿中)
13. Shinya WATANABE, Tomoyuki Hiroyasu and Mitsunori Miki, “Multi-Objective Rectangular Packing Problem and Its Applications”, EMO 2003 Conference, 2003 (投稿中)

2. 一般講演

14. 廣安 知之, 三木 光範, 渡邊 真也, “多目的最適化のための分散遺伝的アルゴリズムにおける広域ランドスケープ形成スキーム”, 情報処理学会 数理モデルと問題研究会報告, 1999, pp.7-12
15. 廣安 知之, 三木 光範, 渡邊 真也, “全体シェアリングによる多目的分散遺伝的アルゴリズム”, 第43回システム制御情報学会研究発表講演会, 1999, pp.381-382
16. 廣安 知之, 三木 光範, 渡邊 真也, “領域分散型遺伝的アルゴリズムの提案”, 情報処理学会 数理モデルと問題研究会報告, 北海道札幌市 1999, pp.37-40
17. 廣安 知之, 三木 光範, 渡邊 真也, “領域分散型多目的遺伝的アルゴリズムの検討”, 情報処理学会 99年度秋期全国大会, 1999, pp.149-150

18. 廣安 知之, 三木 光範, 谷村 勇輔, 渡邊 真也, 金子 美華, 吉田 純一, 川崎 高志, 小掠 真貴, “分散遺伝的アルゴリズムによる各種クラスタのベンチマーク”, 日本機械学会 日本機械学会第 13 回計算力学講演会講演論文集, pp.497-498,,2000.10.
19. 三木 光範, 廣安 知之, 近藤 武史, 渡邊 真也, “多目的最適化における新たな離散的テスト問題の提案 -巡回セールスマン問題の多目的化-”, 2001,
20. 廣安 知之, 三木 光範, 渡邊 真也, “進化的計算手法による多目的最適化”, 計測自動制御学会 第 13 回自律分散システムシンポジウム, pp.295-300,2001
21. 廣安知之, 三木光範, 奥田環, 渡邊真也, “多目的最適化問題のための多目的GAと単一目的GAの分散協力型モデル”, 日本機械学会 2001 年度 年次大会, pp.26-27,2001.8
22. 廣安知之, 三木光範, 奥田環, 渡邊真也, “多目的 GA における分散協力型モデルの提案”, 計測自動制御学会 第 11 回 FAN インテリジェント・システム・シンポジウム, pp.235-238,2001.9.
23. 渡邊 真也, 廣安 知之, 三木 光範, “局所的培養型マスタースレーブモデルの多目的最適化への応用”, 第 8 回 MPS シンポジウム -進化的計算シンポジウム 2001-, pp.401-408,2001.10.
24. 渡邊 真也, 廣安知之, 三木光範, “局所的培養型 GA の多目的最適化への応用”, 第 46 回システム制御情報学会, pp 597-598,2002.5.
25. Shinya WATANABE, Tomoyuki Hiroyasu and Mitsunori Miki, “NCGA : Neighborhood Cultivation Genetic Algorithm for Multi-Objective Optimization Problems”, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002) LATE-BREAKING PAPERS, pp.458-465,2002.8.
26. T. Hiroyasu, M. Miki, J. Kamiura, S. Watanabe, H. Hiroyasu, “Multi-Objective Optimization of Diesel Engine Emissions and Fuel Economy using Genetic Algorithms and Phenomenological Model ”, SAE 2002 Powertrain & Fluid Systems Conference, 02FFL-183, 2002.10.

3. 研究会

27. 渡邊 真也, 廣安 知之, 三木 光範, “多目的最適問題における進化的アルゴリズムの並列化”, システム制御情報学会 第 8 回創発システム研究交流会講演会,2001

4. 学内論文

28. 廣安 知之, 三木 光範, 渡邊 真也, “多目的分散遺伝的アルゴリズムにおけるシェアリング, 収束判定, 及び解の評価手法”, 同志社大学理工学研究報告, Vol.40 No.4, 2000, pp.19-30
29. 三木 光範, 廣安 知之, 近藤 武史, 渡邊 真也, “多目的遺伝的アルゴリズムにおけるパレート解の補間”, 理工学研究報告, Vol. 41, No. 2, 2000, pp.139-146
30. 廣安 知之, 三木 光範, 奥田 環, 渡邊 真也, “多目的遺伝的アルゴリズムの分散協力型モデル”, 同志社大学理工学研究報告 第42巻 第3号, pp.1-12, 2001
31. 廣安知之, 三木光範, 渡邊真也, 迫田岳志, 上浦二郎, “多目的遺伝的アルゴリズムにおける各手法の比較”, 同志社大学理工学研究報告 第43巻 第1号 pp.41-52, 2002

5. ポスター

32. Shinya WATANABE, Tomoyuki Hiroyasu and Mitsunori Miki, “LCGA : Local Cultivation Genetic Algorithm for Multi-Objective Optimization Problems”, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002), pp.702, 2002

以 上