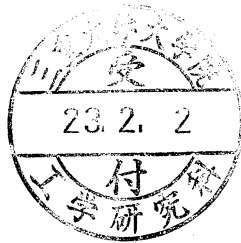


修士論文

多目的進化的アルゴリズムにおける
解の多様性向上に関する研究



平成 22 年度修了

三重大学大学院工学研究科

博士前期課程 電気電子工学専攻

市川 敦史

目次

第 1 章 はじめに	1
第 2 章 多目的進化的アルゴリズムについて	3
2.1 多目的進化的アルゴリズム	3
2.1.1 多目的最適化	3
2.1.2 進化的アルゴリズム	8
2.2 従来手法	11
2.2.1 NSGA-II	11
2.2.2 支配領域制御手法	15
第 3 章 提案法	19
3.1 従来法の問題点と考察	19
3.2 提案手法	23
第 4 章 実験	25
4.1 実験方法	25
第 5 章 まとめ	31
謝辞	32
参考文献	33
発表論文一覧	34

第 1 章 はじめに

我々の生活の中には，数多くの最適化問題がある．最適化問題とは，与えられた制約条件の下で，目的とする評価関数の値を最良にする解を求める問題である．現実の世界に現れる問題は多目的最適化問題であることが多く，ディーゼルエンジンの燃焼における，煤煙，窒素酸化物，燃料消費率などの最適化を目的とするディーゼルエンジンの燃焼噴射スケジューリング[1] や，航空機の主翼設計における，空力，構造，装備等の最適化を目的とする航空機最適化設計[2] など，様々なものがある．多目的最適化問題では，複数の目的が互いに相反する目的であることもあり，すべての目的を同時に最適にするような単一の解が存在しないことが多いので複数の解を求めることになる．そのため，求められた複数の解の多様性や収束性が重要となる．また，この多目的最適化問題を，進化的アルゴリズムによって解く方法があり，これを多目的進化的アルゴリズムという．進化的アルゴリズムとは，生物の進化を模倣しており，多様な解候補を複数保持しながら最適解の探索をおこなう手法である．多目的進化的アルゴリズムとしていくつかの手法が提案されている．その中でも NSGA-II[3] は収束性と多様性の面で優れた手法であるためよく知られている．

NSGA-II は，非支配ソートによる順位付けや解の集中を抑える混雑距離の評価を行うことにより解の成長を向上させる点に特徴がある．また，NSGA-II を基本としながら，支配領域を制御することによって，解の多様性と収束性をさらに高めることを目的とした支配領域制御手法[4] が提案されている．この方法は，個体の

支配領域を縮小させることにより多様性を向上させようとし、支配領域を拡大させることにより収束性を向上させようとしている。しかし、目的関数の特徴が異なるような多目的最適化問題において、支配領域制御手法では、本論文で議論するように、支配領域制御パラメータを変更し、支配領域を縮小させても、解の多様性を向上させるようには働かず、解の進化方向の違いとなるように働くことがある。本論文では、支配領域制御パラメータを変えることによって、解の進化方向が変わることを利用し、複数のパラメータを用いて進化方向の異なる解をそれぞれ求め、それらの解を統合し、統合した解をさらに進化させて、解の多様性を向上させる手法を提案する[a,b].

本論文の構成は以下のとおりである。2章では、多目的進化的アルゴリズムの枠組みと、本研究が対象とする支配領域制御手法に関して説明する。3章では、従来手法である支配領域制御手法の目的関数の特徴が異なる問題に対して解の多様性が向上できない問題点を分析するとともに、解の多様性を向上させるための方法を提案する。4章では、3章で述べた提案手法の性能評価実験を行い、その有効性を確認する。5章で、本研究をまとめる。

第2章 多目的進化的アルゴリズムについて

この章では，提案法を説明するのに必要となる概念・用語と，従来手法である NSGA-II[3] および支配領域制御手法[4] について述べる．

2.1 多目的進化的アルゴリズム

2.1.1 多目的最適化

多目的最適化とは，複数の目的を同時に最適化することである．そして，多目的最適化問題とは，「複数の目的関数を与えられた制約条件の中で最大化，または，最小化する問題」と定義される．一般に，多目的最適化問題は， n 個の個体をもつ m 個の目的関数

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

を，制約条件

$$\mathbf{x} \in F \quad (2.2)$$

のもとで最大化，または，最小化するような解 $\mathbf{x} = (x_1, x_2, \dots, x_n)$ を求めるという問題として定式化される．ただし， F は解空間 S の部分集合であり，制約条件を満たす解集合を表す．複数の目的関数を同時に最適化しようと試みると，ある一つの目的関数に対する最適解が，他の目的関数の最適解にはならないことが多い．そのため，多目的最適化では，制約条件を満たす他のどの解と比べても全ての目的関数値において劣っていないような解を求めることになる．多目的最適化では，このような解をパレート最適解と呼び，このような解の集合をパレート最適解集合と呼ぶ．よって，多目的

最適化の目標は、このパレート最適解集合を求めることである。

パレート最適解は、多目的最適化問題における解の支配関係により定義される。まず、解の支配関係を、最大化問題を例に説明する。いま、制約条件を満たす x, y という解があり、 m 個の目的関数をもつとする。すべての $i(i=1, 2, \dots, m)$ に対して、 $f_i(x) \geq f_i(y)$ のとき、 $f(x) \succeq f(y)$ と書く。また、 $f(x) \succeq f(y)$ であり、かつ $f_k(x) > f_k(y)$ であるような $k(1 \leq k \leq m)$ が少なくとも 1 つ存在するとき、 x は y を支配するといい、 $f(x) \succ f(y)$ と書く。もし、 x が y を支配するならば、 x の方が y より良い解である。そして、他のどの解にも支配されない解（非支配解）がパレート解である。パレート最適解集合 (Pareto optimal solution set (POS)) は次式で定式化される。

$$POS = \{x \in F \mid \neg \exists y \in F : f(y) \succ f(x)\} \quad (2.3)$$

このようにして、多目的最適化問題では、パレート最適解集合を求める。しかし、実際の多目的最適化問題では、パレート最適解集合を求められないことがある。求められない場合は、求めた解の中での非支配解の集合を求める。これをパレート近似解集合と呼ぶ。以下で 2 目的最大化問題を例に詳しく説明する。

いま、個体 x に対する 2 つの評価関数を $f_1(x), f_2(x)$ とする。 $f_1(x), f_2(x)$ を最大化する 2 目的問題を例にとり、多目的最適化では、どのような解を求めるのかについて説明する。図中の x, y は、ある世代での 2 つの個体を表す。まず、図 2.1 においては、 $f_1(x) > f_1(y)$, $f_2(x) > f_2(y)$ であるので、 x が y を支配しており、 x の方が y より良い解だと考えられる。次に、図 2.2 においては、 $f_1(x) > f_1(y)$, $f_2(x) < f_2(y)$ であり、 x が y を支配せず、かつ、 y が x を支配していないので、 x と y は、お互いに支配されていない。すなわち、非支配で

ある．このような支配，非支配の関係に基づいて，多目的最適化では，図 2.3 の水色で示すパレート近似解集合を求める．

また，多目的最適化問題を解き，パレート最適解集合の近似解であるパレート近似解集合を求める場合，多様性と収束性が重要である．多様性については，図 2.4 で，(b) よりも(a) というように，より多様な解を求められたものを良いとする．収束性については，図 2.5 で，(a) と(b) を同じ世代数でのパレート解集合とすると，(b) よりも(a) というように，少ない世代数で，より良いパレート解集合を求めたものを良いものとする．

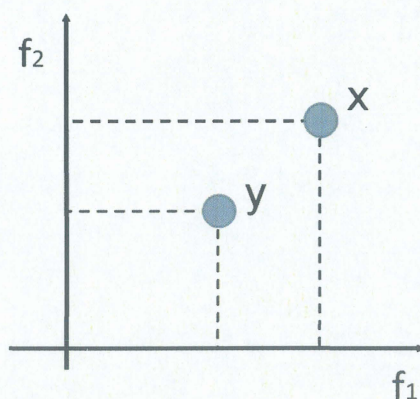


図 2.1: 支配関係の例

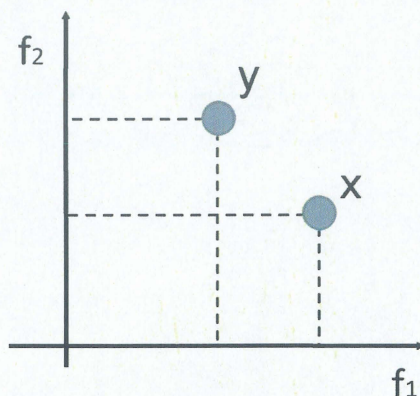


図 2.2: 非支配関係の例

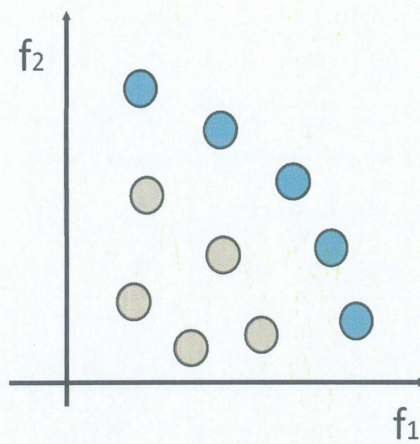
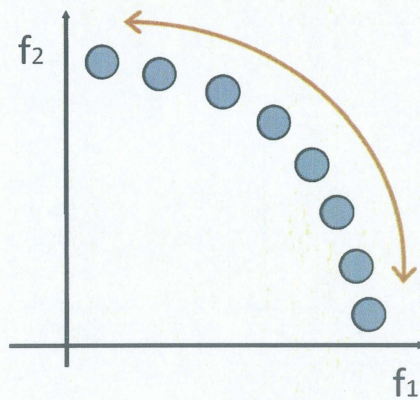
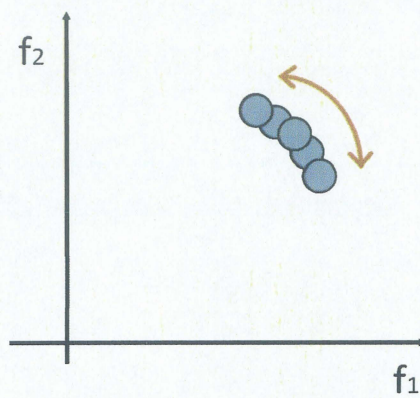


図 2.3: パレート解集合

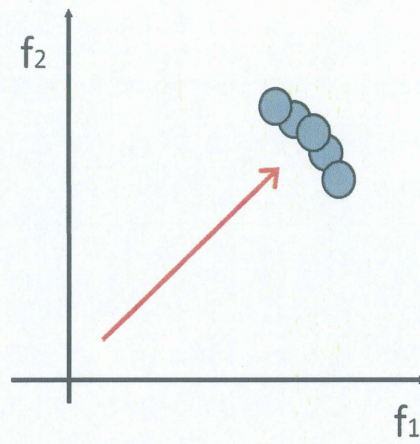


(a) 多様性の良い例

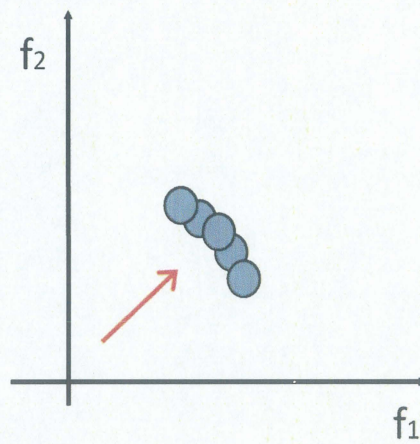


(b) 多様性の悪い例

図 2.4: 多様性の良し悪しの例



(a) 収束性の良い例



(b) 収束性の悪い例

図 2.5: 収束性の良し悪しの例

2.1.2 進化的アルゴリズム

進化的アルゴリズムとは、生物の進化の過程をモデル化したものであり、ある最適化問題に対する最適解を求めるための探索アルゴリズムの一種である。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群の中で、環境への適応度の高い個体が高い確率で生き残る。さらに、交叉や突然変異によって、次の世代の個体群が形成されていく。それをまた次の世代、そのまた次の世代へと行っていく。進化的アルゴリズムでは、このような遺伝的操作を繰り返し行い、最適化問題において良い解を見つける。解の候補を一つの個体とみなし、それを遺伝子の並びである染色体を用いて表現する。そして、遺伝的操作を各染色体にほどこすことで、解の候補を進化させる。

以下に、進化的アルゴリズムの操作である、初期集団生成、評価、選択交叉、突然変異について簡単に述べる。

(1) 初期集団の生成

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。これを初期集団とする。一般には、各個体の染色体に遺伝子をランダムに割り当てることで行う。

(2) 評価

各目的に対して、あらかじめ定めておいた評価関数によって、各個体の評価値を求める。次に計算された評価値によって、各個体の評価を行う。進化的アルゴリズムにおいては、評価値に基づいて選択、淘汰などが行われるので、評価関数は、個体の成長に大きな影響を与える。したがって、評価関数をどのように与えるかは非常に重要である。

(3) 淘汰

各個体の評価値に応じて、次世代に残す個体を決定する。すなわち、環境に適応しているもの（評価が良いもの）が生き残り、適応していないもの（評価が悪いもの）は、消滅し、世代交代する。また、淘汰する際に、エリート保存と呼ばれる手法がある。エリート保存とは、個体群の中で最も評価値が高いいくつかの個体を無条件でそのまま次世代に残すことである。これにより、その時点で最も評価値の高い個体が、交叉や淘汰、突然変異により消えて、進化が後退する（退化する）のを防ぐことができる。

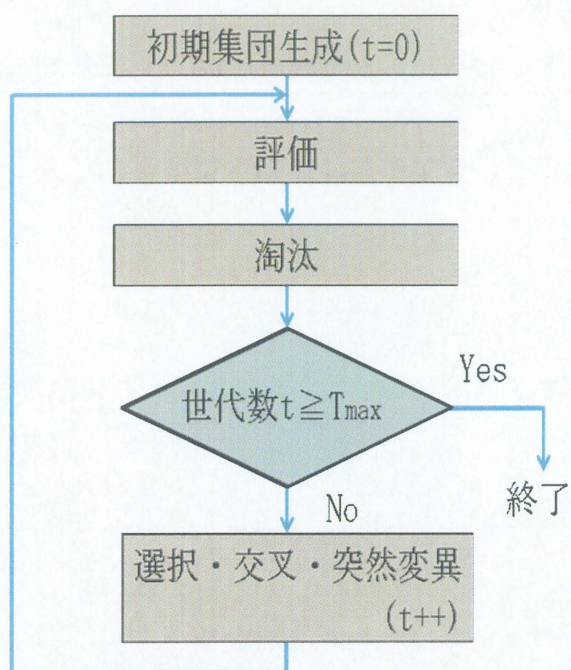
(4) 選択・交叉

2つの個体（親）を選び、それらの染色体を組み替えて新しい個体（子）を生成する。本研究では、4つの個体からトーナメント選択により2つの個体（親）を選ぶ。交叉には、一点交叉や多点交叉など様々な方法があるが、本研究では、広くよく用いられている一様交叉を用いる。これは、一点交叉などでは、新しい個体（子）を作るときに用いる切断箇所によっては、偏りが生じやすいからである。一様交叉は、遺伝子の長さ分の桁を持つマスクビットをランダムに作り、そのマスクビットにしたがって交叉を行う。また、交叉により生成された子が、制限条件を満たさない個体（致死遺伝子）となることを防ぐため整形を行うこともある。

(5) 突然変異

ある確率で、ある個体の染色体の一部の値を変える。もともと交叉だけでは、親の個体に依存する限られた子しか生成することができない。突然変異は、交叉だけでは生成が不可能な子をつくり、個体群の多様性を維持する働きをする。

(2)～(5) をあらかじめ設定した世代数だけ繰り返し，より良い解を求める．進化的アルゴリズムの流れを図 2.6 に示す．



現在の世代： t

あらかじめ設定した世代数： T_{\max}

図 2.6: 進化的アルゴリズムの流れ

2.2 従来手法

2.2.1 NSGA-II

多目的進化的アルゴリズムでは，複数の個体を維持しながら，多様性や収束性が良い個体の集団を求める．そのため，各個体間の支配関係や個体間の密度などを考慮して，評価を行う必要がある．NSGA-II は，非支配ソートや混雑距離を用いた評価方法や探索過程における非支配解の保存などの特徴をもつ多目的進化的アルゴリズムである．NSGA-II の特徴である非支配ソートや混雑距離は，前述した進化的アルゴリズムの評価操作の中に組み込まれ，エリート保存は，淘汰操作で用いられる．これらを用いることにより，個体の局所解への集中をふせぎ，多様性を維持したり，現世代における各目的の最良値をもつ個体が次世代へ保存されるなど，パレート解の適切な保存ができ，より良い個体の成長を実現する．以下に，非支配ソートと混雑距離の評価について説明する．

2.2.1.1 非支配ソート

各々の個体について，支配されるかどうかの支配関係を全ての他の個体と比較し，その支配関係のレベルによって，個体をフロントに分ける．まず，他のどの個体からも支配されていない個体（非支配な個体）からなる集合を第 1 フロントとする．それから，第 1 フロントにある個体を除いて，残った個体の中から，非支配な個体を探し，それを第 2 フロントとする．そして，この上述した手順を繰り返して，個体がなくなるまで，行う．これによって，全ての個体が，第 β フロント ($\beta = 1, 2, \dots$) のどれかに入るように分類される．フロント番号が小さい個体ほど，支配関係からみて，良い個体である．

2.2.1.2 混雑距離

混雑距離とは、ある世代での集団において、フロントごとに計算され、そのフロントに含まれる個体に関する個体の密度に関する尺度である。混雑距離は、そのフロントに含まれる個体に対して、各目的関数の軸において、隣り合う個体との距離を足し合わせるにより計算される。この値が小さいほど、周りに個体があり、似たような個体があるので、多様性の見地から、評価が低いものとなる。以下に、あるフロントに含まれる個体に対する混雑距離の求め方を示す（図 2.7）。

- (1) フロントに含まれる個体を求める。その個体の数を l とする。
- (2) 手順(1) のフロントに含まれる各個体 $\mathbf{x}^{(k)}$ ($k = 1, 2, \dots, l$) に対して、混雑距離の初期値 $d_k = 0$ とする。
- (3) 各目的関数 f_i ($i = 1, 2, \dots, m$) について、以下を繰り返す。
 - (3.1) f_i の値が良い順に個体をソートし、それらを、あらためて $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(l)}$ とする。
 - (3.2) 各目的関数 f_i ($i = 1, 2, \dots, m$) に対して、まず、目的 f_i の最大値と最小値である両端の個体 $\mathbf{x}^{(1)}$ と $\mathbf{x}^{(l)}$ に対して、無限距離を与える ($d_1 = d_l = \infty$)。さらに、両端の個体以外の全ての個体 ($k = 2, 3, \dots, l-1$) に対して、以下の式に従った混雑距離の計算を行う。

$$d_k \leftarrow d_k + \{f_i(\mathbf{x}^{(k-1)}) - f_i(\mathbf{x}^{(k+1)})\} \quad (2.4)$$

続いて、非支配ソートと混雑距離の評価を用いた順位づけを説明する。この順位付けは、個体の良し悪しを比較する際に利用す

る．これによって，選択・交叉や淘汰などでより適切な個体の比較が行え，局所解に集中するのを防ぎ，多様性の向上につながる．非支配ソートだけでは，同一フロント間の比較が行えないが，そこに混雑距離を用いることによって，より多様性を考慮した個体の比較ができる．例えば， x と y の 2 個体があるとする．そして，個体 x のフロント番号を $xfront$ ，混雑距離を $xdistance$ と表す．以下のようなとき，個体 y よりも個体 x の方が優れている．

(a) $(xfront < yfront)$

(b) $(xfront = yfront) \quad \text{and} \quad (xdistance > ydistance)$

NSGA-II では，エリート保存の方法にも特徴がある．通常の進化的アルゴリズムでは，一般に，数個の個体を次世代に保存する．一方，NSGA-II では，大きさ N の親集団から大きさ M の子集団を生成し，これらを併合して大きさ $(N+M)$ の集団を作り，評価の良いものから，大きさ N だけ次世代に保存し，これを次世代の親集団とする．これは，多目的最適化問題では，パレート最適解集合を求めることが目標であり，その集合が淘汰されてしまわないように，通常のものとは比べ，多くのエリートを取り，次世代に保存する．また，このアルゴリズムの流れを図にしたものを図 2.8 で示す．

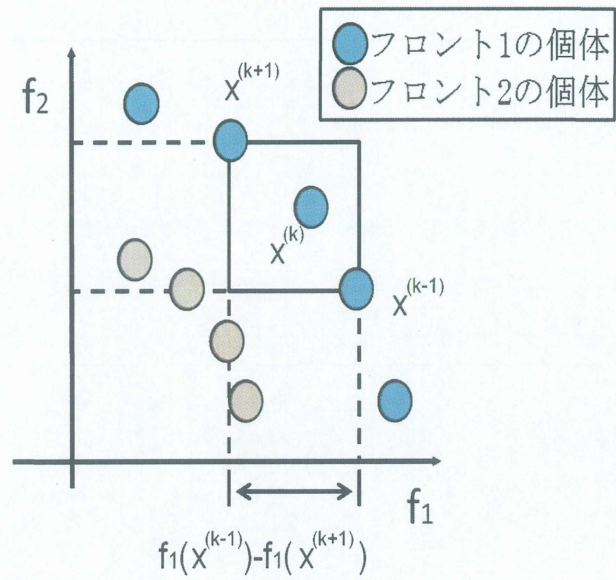
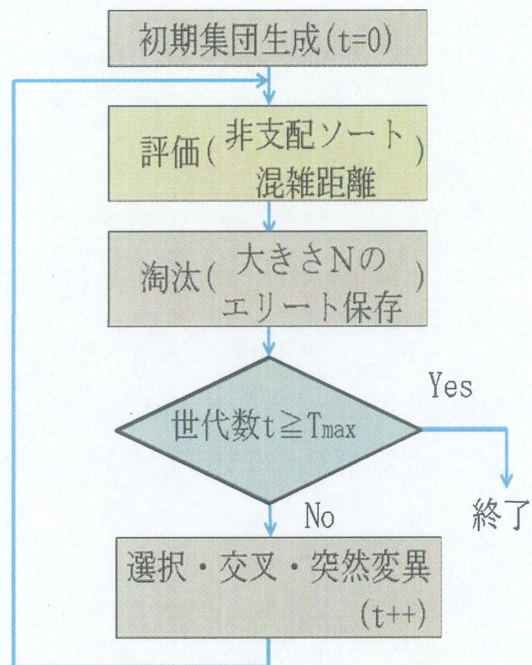


図 2.7: 混雑距離の算出



現在の世代: t
 あらかじめ設定した世代数: T_{\max}

図 2.8: NSGA-II の流れ

2.2.2 支配領域制御手法

NSGA-II では、最適化問題の制約条件を満たす解空間が大きい場合、非支配ソートや混雑距離での順位付けが効果的でないときがある。それは、解空間が広がると、非支配ソートでのフロントの数が少なくなりやすくなるので、ほとんどの個体が、非支配解となる。そのために、収束性が失われやすくなる。また、求めた解が局所解に集中してしまいやすくなるような複雑な問題では、フロントの数が多くなる場合がある。すなわち、1つのフロントにある個体の数が少なくなる。そのため、多様性が失われることがある。これらを改善するために、支配領域制御手法[4] が考えられた。

支配領域制御手法は、多目的最適化手法の一つである NSGA-II に支配領域を制御するパラメータを導入したものである。パラメータの値によって、支配領域を制御し、多様性、または、収束性を向上させることを意図している。文献[4] では、ある個体 \mathbf{x} に対する m 個の目的関数（ここでは最大化問題とする）の値を $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, ..., $f_m(\mathbf{x})$ とするとき、支配領域制御のためのパラメータ S_i ($0 < S_i < 1$) ($i=1, \dots, m$) によって、 $f_i(\mathbf{x})$ を、

$$f'_i(\mathbf{x}) = r \sin(\omega_i + S_i \cdot \pi) / \sin(S_i \cdot \pi) \quad (i = 1, 2, \dots, m) \quad (2.5)$$

に写像し、写像された $f'_i(\mathbf{x})$ によって支配関係を判定することにより多様性と収束性を制御している。ここで、 r はベクトル $\mathbf{r} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))$ の長さ、 ω_i は \mathbf{r} と第 i 軸のなす角度である。通常は、 S_i の値を i によらず同じ値 S とし、収束性を高めたいときには $S < 0.5$ 、多様性を高めたいときには $S > 0.5$ である適当な値に決める。以下に 2 目的最大化問題を例に図を用いて、支配領域の

制御について説明する．図 2.9 のように， f_1 と f_2 を最大化する 2 目的最大化問題において，個体 A, B, C があるとする．また， $\phi = S * \pi$ とする．図 2.9 のように， $S = 0.5$ のときには，NSGA-II と同じ支配領域になり，NSGA-II そのものとなる．この場合には，A が C を支配しており，A と B がパレート解である．図 2.10 のように， $S = 0.65$ のときには，支配領域が縮小し，A, B, C のすべての個体がパレート解となる．この場合は，多くのパレート解が残りやすくなり，多様性が向上する．一方，図 2.11 のように， $S = 0.4$ のときには，支配領域が拡大し，A が B, C を支配するので，A のみがパレート解である．この場合は，1 つのフロントにある個体の数が少なくなるので，フロントの数が増加しやすくなり，収束性が向上する．

このようにして， S の値をどう設定するかによって，支配領域を制御することができ，多様性，または，収束性を向上させることができるという点で，支配領域制御手法は有効な手法である．しかし，目的関数の特徴がお互いに異なるような多目的最適化問題においては，支配領域制御手法では，解の多様性において良い解が得られないことがある．このことは，目的関数の特徴の違いにより，残りやすくなる解に偏りが生じることが一因であると思われる．

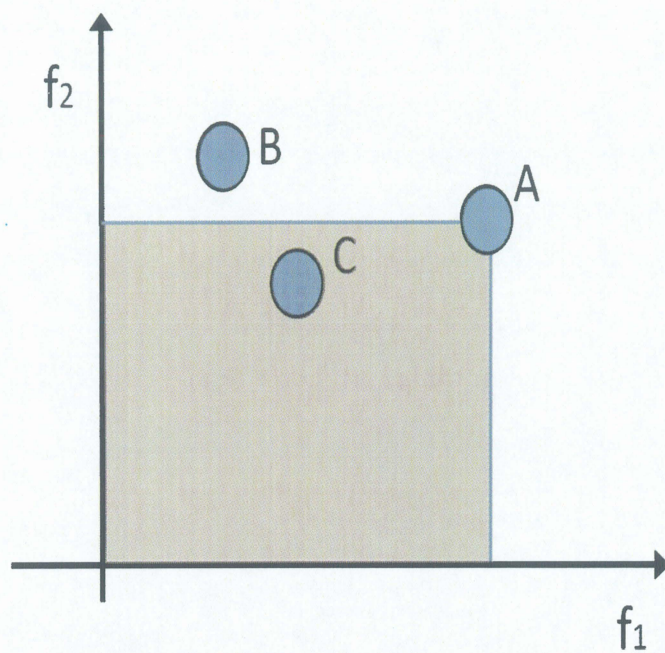


図 2.9: NSGA-II ($S=0.5$) における支配領域の制御

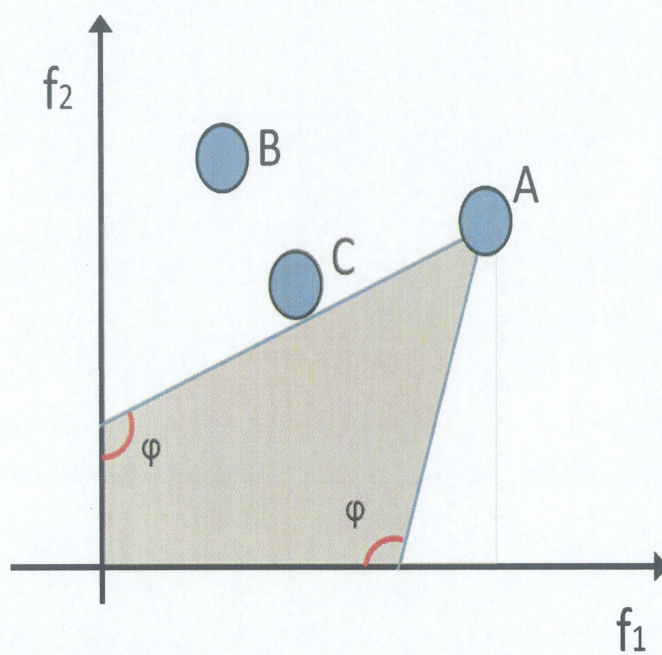


図 2.10: 支配領域の縮小の例

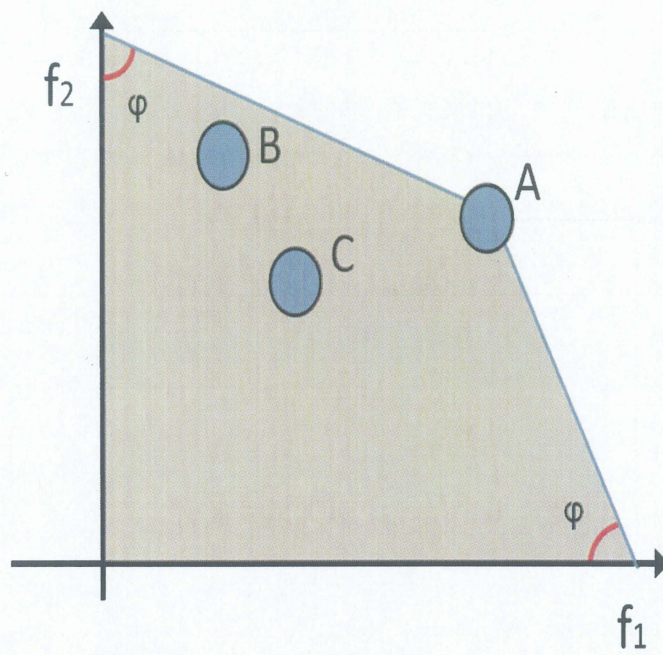


図 2.11: 支配領域の拡大の例

第 3 章 提案法

この章では，目的関数の特徴が異なる問題に対して解の多様性を向上できないことがあるという従来手法（支配領域制御手法）の問題点について考察する．そして，その問題点の分析結果から得られた知見をもとに，解の多様性を向上させる提案手法を述べる．

3.1 従来法の問題点と考察

従来手法では，各目的関数の特徴が似ている場合には，図 3.1 のように解の多様性を向上させることができる．しかし，各目的関数の特徴が大きく異なるときには，(2.5) 式の写像が多様性の向上に向けた写像とはならず，図 3.2 のように解の多様性を向上させることができないことがある．

すなわち，目的関数の値の各個体による分布（平均，分散）が各目的で大きく異なる場合など，目的関数の特徴が異なる場合には，進化途中での集団内の個体の分布が偏るため，パラメータ S が多様性向上や収束性向上には必ずしも結びつかないことがある．

いま説明例として，進化させる途中のある世代での個体が図 3.3 のようになっているものとする． $S=0.5$ の場合，図 3.3 のような支配領域となる．多様性を向上させようとした $S>0.5$ の場合，図 3.4 のように，左側の範囲の非支配個体が多くなる．つまり，左側の範囲の解が生き残りやすくなる．親子併合集団から次世代の親集団を作るときの親集団内個体数は一定であるため，左側の範囲の解が生き残りやすくなると，右側の解が淘汰されやすくなる．こ

のような支配関係のまま，何度も進化するため，より左側の解が生き残り．その方向がほぼ進化の方向となる．

一方，収束性を向上させようとした $S < 0.5$ の場合，図 3.5 のように，非支配個体が少なくなるので，非支配個体として残った個体によって，進化の方向がほぼ定まる．

以上のことから，目的関数の特徴が異なる問題では， $S > 0.5$ として進化させても解の多様化には結びつかないことがあるが， S の値によって進化する方向を変えることができる場合がある．

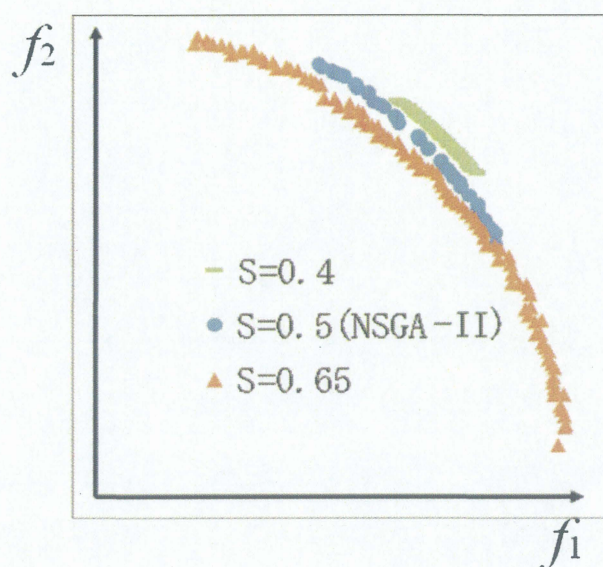


図 3.1: 特徴が似ている問題の場合

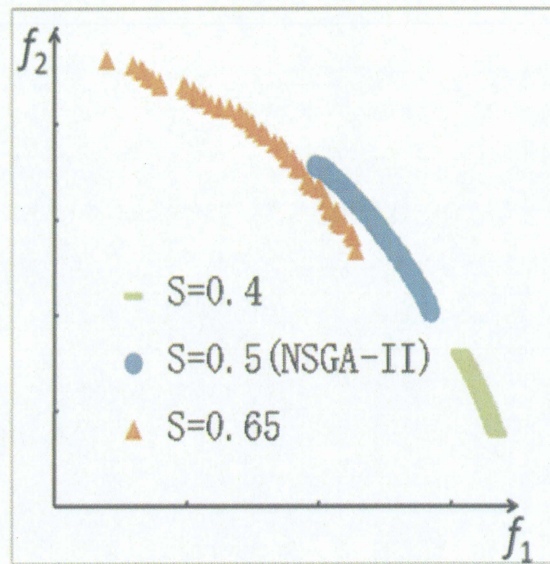


図 3.2: 特徴が異なる問題の場合

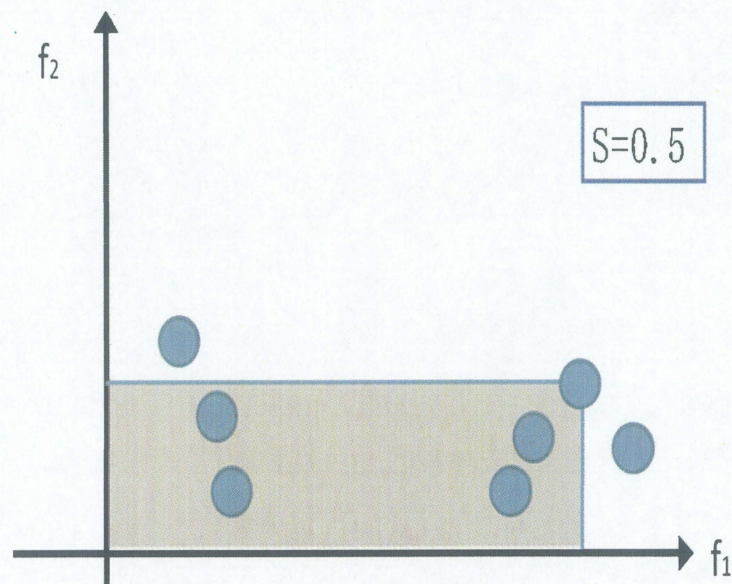


図 3.3: 各目的関数の特徴が異なる場合 ($S=0.5$)

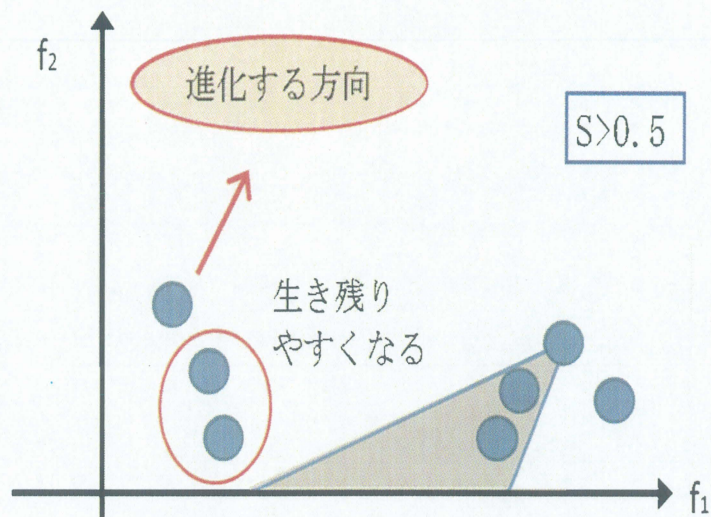


図 3.4: 各目的関数の特徴が異なる場合 ($S > 0.5$)

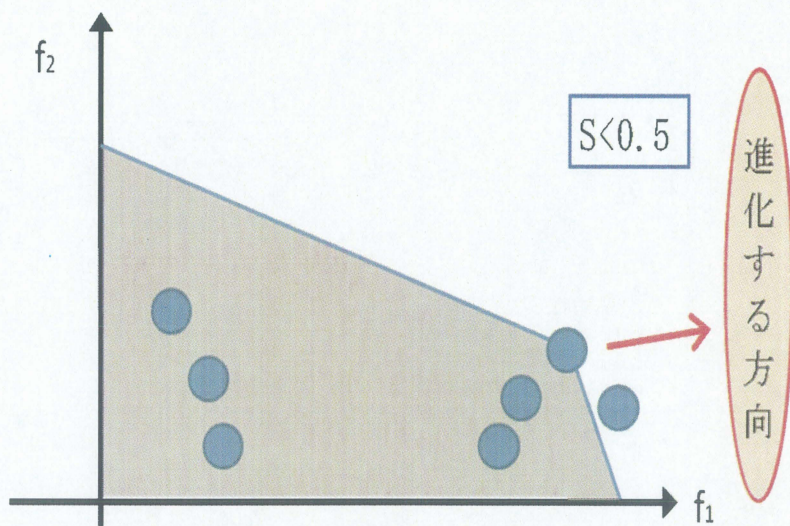


図 3.5: 各目的関数の特徴が異なる場合 ($S < 0.5$)

以上の分析は、支配領域制御のパラメータをいくつか組み合わせることによって、各目的関数の特徴が異なるような最適化問題に対しても、多様性を確保できる手法を実現できる可能性があることを示唆している。

3.2 提案手法

前節での考察をもとに、目的関数の特徴が異なる問題に対しても、解の多様性を向上させるために、パラメータ S の値を複数個用意し、各 S 値によって初期集団を異なる方向に一定世代進化させた後、それらの進化後の集団を統合して解候補集団を作る。ただし、進化後の集団を単に統合しただけでは、各パラメータで決まる進化方向のみに個体が偏る可能性がある。そこで、よりバランスのよい多様な解集合の生成をするために、統合した解候補集団を初期集団としてさらに進化させることにした。以下にその提案手法を示す。

[提案手法の手続き]

まず、制御パラメータの値 S_1, S_2, \dots, S_K を K 個と、各 S_k に対して進化させる世代数 $T_k (k=1, 2, \dots, K)$ 、および、統合後の制御パラメータ S' と進化させる世代数 T' を与えておき、制御パラメータ $S=S_k$ とし、ランダム初期集団から T_k 世代進化させてできるパレート近似解集合 P_k を求める。次に、多様性の観点でバランスのよい解となるように、求めた P_1, P_2, \dots, P_K を併合し、そこから評価の良い個体を選んで初期集団を作り、制御パラメータ $S=S'$ として、この集団を T' 世代進化させ、最終の解集合を求める。

提案手法においては、 K 個の制御パラメータ S_k と、その進化世代数をどのように決めるかが重要であり、厳密にはその決め方は難しい。しかし、上記の提案手法は、進化方向の異なる多様なパレート近似解の元となる個体を大ざっぱに作り、それをもとに、

進化方向を保持・補間しながらさらに進化させるので、趣旨は異なるがとりあえず文献[4]と同様に、 S の値として、0.4, 0.5, 0.65の3種類を用いるのも有効と思われる。また、 S の値が小さい場合の方が収束性は向上する、つまり、進化速度が速く、進化する方向もすぐに定まるため、進化世代数をやや少なめにしても良いと考えられる。逆に、進化速度が遅い S の値が大きい場合には、進化する方向はすぐに定まらないため、進化世代数をやや多めにするのが良いと考えられる。例えば、 S の値として、0.4, 0.5, 0.65の3種類を用いたとすると $S=0.4$ ならば $T_k=500$, $S=0.5$ ならば $T_k=1000$, $S=0.65$ ならば $T_k=1500$ といった進化世代数が考えられる。

第 4 章 実験

この章では、提案手法の有効性を確認するための実験方法と実験結果を示す。

4.1 実験方法

提案手法の有効性を確認するため、目的関数の特徴が異なる最適化問題に対する性能評価実験を行った。比較するのは、従来手法である支配領域制御手法と提案手法である。また、本研究では、2 目的ナップサック問題を例としてとりあげる。よく知られている単一目的ナップサック問題では、評価値（値段）は 1 つのみであるが、多目的最適化をおこなうため、本研究では、2 つの評価値（値段）をもつ 2 目的ナップサック問題を用いる。2 目的ナップサック問題とは、制限重量内でできる限り 2 つの評価値が高くなるような荷物の組合せを求める問題のことである。

実験には、目的関数の特徴が異なる最適化問題を用いるが、このような問題は多数存在する、ここではまず、荷物数 1000 で、価値 1 の値の平均が価値 2 の値の平均の 2 倍になっており、価値の分布が図 4.1 のようになっている 2 目的ナップサック問題を用いる。

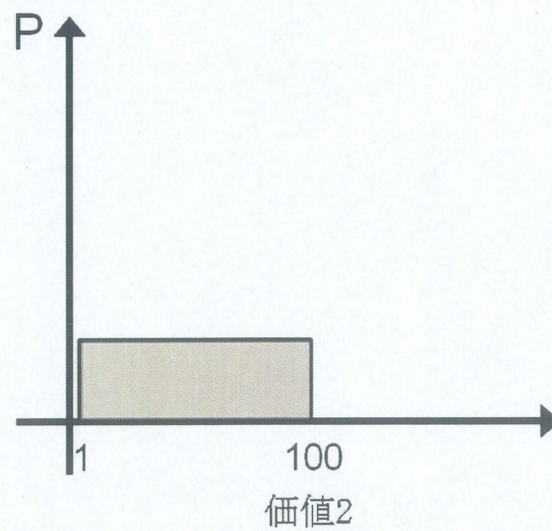
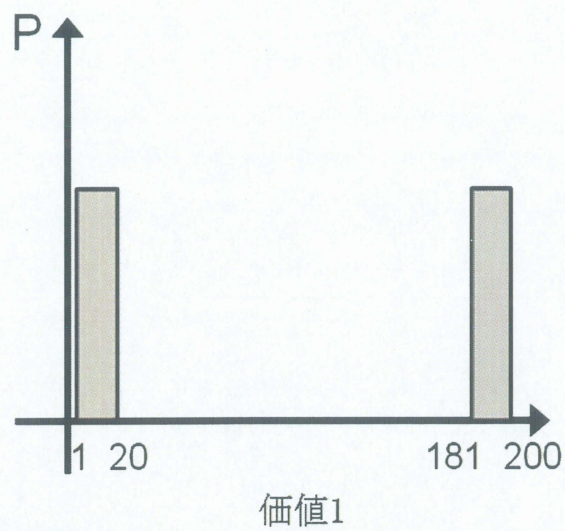


図 4.1: 価値の分布

また，提案手法と，従来法である支配領域制御手法[4]の収束性が良いとされる $S=0.4$ ，多様性が良いとされる $S=0.65$ ，NSGA-II と同じである $S=0.5$ の各場合とを，最終的に得られたパレート近似解によって比較する．

4.2 実験結果

図 4.2 は、従来法[4] による結果であり、 $S=0.4$, $S=0.5$, $S=0.65$ のそれぞれについて、親集団の個体数 100 でそれぞれ 4000 世代まで進化させるという条件で行なった実験結果を示している。図 4.2 から、従来法での多様性が得られると考えられた $S=0.65$ の場合でも解の多様性が確保されておらず、 S の値の変化が解の進化方向の違いとなって現れているということがわかる。

図 4.3 は、提案法による実験結果を示している。この実験では、制御パラメータの数である K の値を $K=3$ とし、 $S=0.4$ で 500 世代、 $S=0.5$ で 1000 世代、 $S=0.65$ で 1500 世代進化させ、それぞれで得られたパレート近似解から評価の良い個体 100 個を選んで初期集団を作り、 $S=0.5$ でさらに 1000 世代進化させ、合計 4000 世代進化させた。

図 4.2 と図 4.3 を比較すると従来手法の $S=0.65$ の場合よりも提案法がより多様性のある解を求められているということがわかる。また、従来手法の $S=0.4$ の場合のパレート近似解と $S=0.5$ の場合のパレート近似解の間にある部分の解も提案法により求められていることがわかる。これは、提案法が多様性の観点でバランスの良い解を求めるために最後に進化させたことが結果としてあらわれたと考えられる。

また、見た目だけの多様性ではなく、数値として多様性を評価できるように被覆率を用いた評価を行った。被覆率とは、いかに解を幅広くかつ隙間なく求めているかに関する評価尺度である。被覆率の計算の仕方は図 4.4 のように各目的関数に対して領域を任意に等分割し、各領域に解が存在すれば+1、なければ何もしな

い，という計算を行い，その合計を分割数で割ることにより被覆率を得る．領域の分割数が個体数と同じ場合，一番厳しい被覆率の評価基準となるので，今回の実験では，個体数 100 であることから 100 分割とした．その結果，図 4.5 の被覆率を得た．図 4.5 から，従来手法での多様性が得られると考えられていた $S=0.65$ の被覆率は，各目的関数ともそれぞれ約 40%であり，提案法の被覆率は，それぞれ約 80%であった．このことから，提案法は，目的関数の特徴が異なる問題に対して，被覆率においても良い結果を得ることができた．

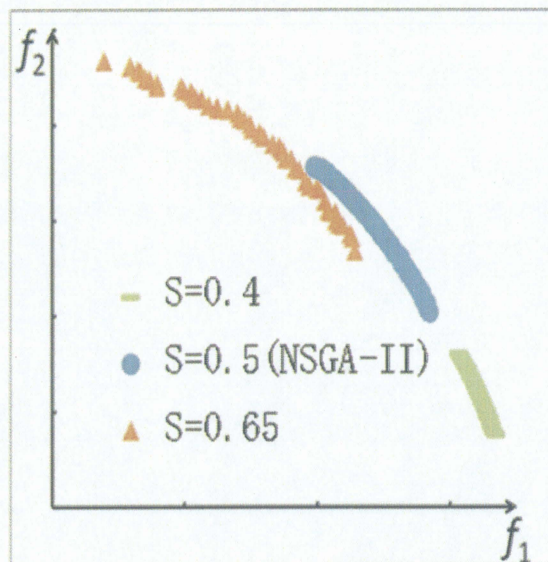


図 4.2: 従来法の 3 つの実験結果

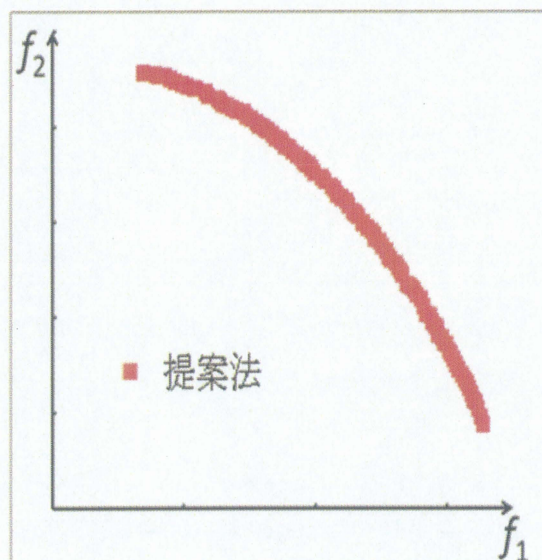


図 4.3: 提案法の実験結果

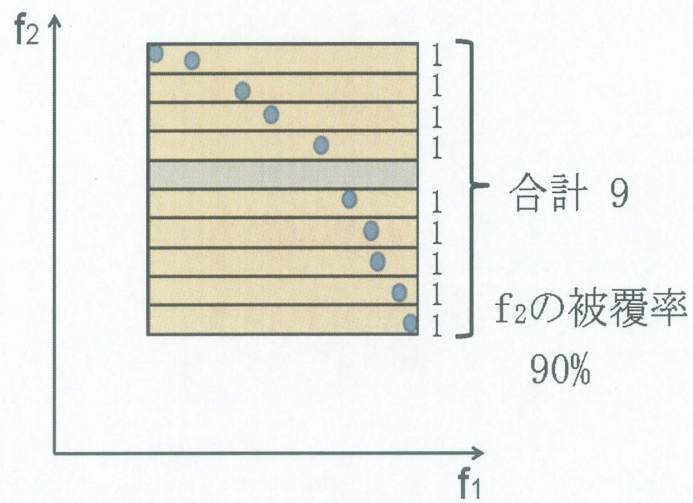
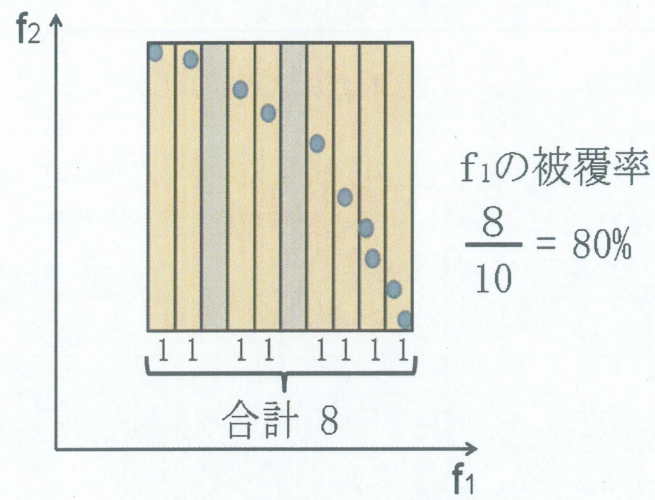


図 4.4: 被覆率の計算例

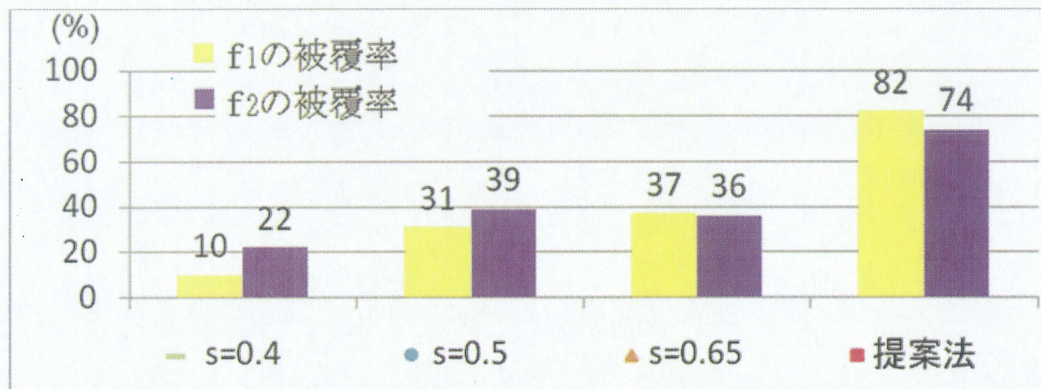


図 4.5: 各目的間数の被覆率

第 5 章 まとめ

多目的最適化問題を効率良く解くために，従来からいくつかの多目的進化的アルゴリズムが提案されており，中でも NSGA-II[3] はよく知られている．この NSGA-II をベースにし，これに個体間の支配関係を制御するパラメータ S を導入した支配領域制御手法 [4] は，支配領域を制御することで，目的関数の特徴がお互いに似ている問題に対して，多様性を向上させることができる．

しかし，目的関数の特徴がお互いに異なる問題については，多様性を向上させることができないことがある．そこで，支配領域制御手法が解の多様性を向上できない原因について調査を行った．その結果，このような問題に対しては，支配領域制御パラメータ S の値の違いによる個体間の支配関係の違いが，解の進化の多様性に影響するのではなく，解の進化方向の違いとなるように働く場合があるということを明らかにした．

本論文では，この現象を利用し，複数個のパラメータを用いて進化方向の異なる解をそれぞれ求め，それらの解を統合し，統合した解をさらに進化させて，多様性を向上させる手法を提案した．そして，2 目的ナップサック問題に対する適用実験により，その有効性を確認した．

今後の課題として，3 目的以上の多目的ナップサック問題や現実の問題に対しても，提案法が有効であることを確認することなどが挙げられる．

謝辞

本論文は，著者が三重大学大学院工学研究科電気電子工学専攻に在籍中におこなった研究をまとめたものである．本研究を進めるにあたり，懇切丁寧な御指導と御督励を賜った三重大学大学院工学研究科電気電子工学専攻，北英彦准教授，林照峯特任教授に感謝いたします．また，本論文の副査をしていただいた残間忠直准教授に感謝いたします．

最後に，本研究論文をまとめるにあたり，助言，討論など，お世話になった計算機工学研究室の皆様方，他すべての人に感謝いたします．

参考文献

- [1] 中山靖一, 廣安知之, 三水光範, 渡邊真也, 廣安博之, SPEA2+
によるディーゼルエンジンの燃料噴射スケジューリング問題
の多目的最適化(設計と最適化 V), 設計工学・システム部門講
演会講演論文集 Vol.2004, No.14(20041129), pp.239-242, (2004)

- [2] 大林茂, 航空機の多目的最適設計, 人工知能学会誌, Vol.18,
No.5(20030901), pp.495-501, (2003)

- [3] Kalyanmoy Deb, A Fast Elitist Non-Dominated Sorting Genetic
Algorithm for Multi-Objective Optimization: NSGA-II, KanGAL
Report No.200001, (2002)

- [4] Hiroyuki Sato, et al.: Controlling Dominance Area of Solutions in
Multiobjective Evolutionary Algorithms and Performance Analysis
on Multiobjective 0/1 Knapsack Problems. IPSJ Digital Courier 3,
pp703-708, 2007.

発表論文一覧

本論文に関するもの

[a] 市川敦史，林照峯，北英彦，多目的進化的アルゴリズムの解の多様性向上に関する一考察，平成 22 年度電気関係学会東海支部連合大会講演論文集， D4-4， 2010

[b] 市川敦史，林照峯，北英彦，多目的進化的アルゴリズムにおける解の多様性向上に関する考察，平成 22 年度三重地区計測制御研究講演会講演論文集， A-8， 2010

本論文と直接には関しないもの

[c] 市川敦史，林照峯，北英彦，BIST 支援スキャンテストにおける LFSR 状態制御に基づくテスト効率向上について，平成 21 年度三重地区計測制御研究講演会講演論文集， B-15， 2009