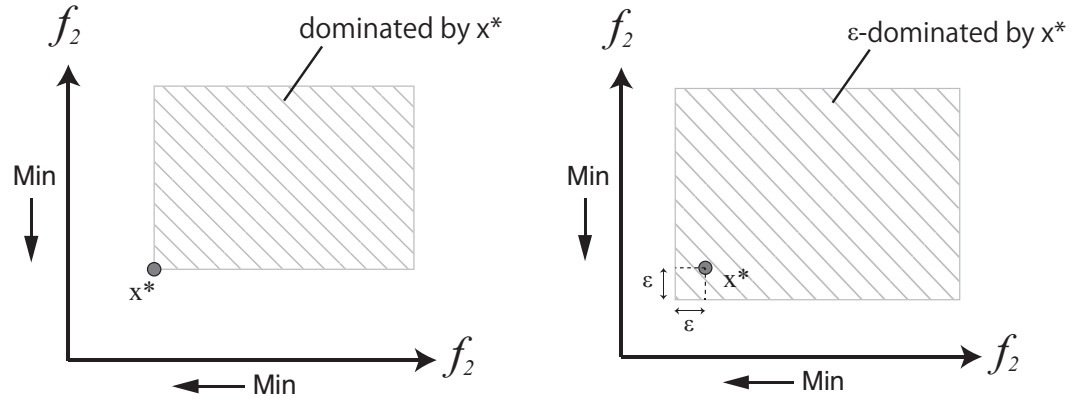
第3章　関連研究

本項では，EMOA における離散化に関連する研究について述べる．

3.1　目的関数の離散化に関する研究

ϵ-dominance

目的関数空間の離散化手法の一つに，Laumanns が提案した ϵ-dominance[?] がある．ϵ-dominance は，ある距離 ϵ ないのすべての点の中から一点を非劣解として採用することで，非劣解の数を削減することができる．図 3.1.1 に ϵ-dominance のを用いた場合の支配領域のイメージ図を示す．



(a) 通常の優越関係を用いた場合の支配領域　　(b) ϵ-dominance を用いた場合の支配領域

図 3.1.1: ϵ-dominance を用いた場合の支配領域の比較

図 3.1.1 のように，ϵ-dominance を用いることで解の支配領域が増え，さらに増えた領域内では非劣解を一点のみ採用するため，非劣解の数が削減される．削減した非劣解をアーカイブとして用いる手法が提案されており，多数目的最適化問題において収束性などの探索効率が向上することが示されている．

目的関数空間の異なるスケールを目的関数に持つ問題に関する研究

Ishibuchi らによって，目的関数空間の離散化具合による影響評価が行われている [?]．多目的組み合わせ最適化問題などでは，異なるケースの離散値を取る目的関数を持つことがあるが，この違いが進化にもたらす影響を評価し，粗い離散値を取る目的関数がある場合，その関数の最適化方向への探索効率が低下することが示されている．これは，粗い離散値を取る目的関数がある場合，非劣解の数が著しく減少してしまうからである．このような問題でも効率的に探索を行うため，Ishibuchi らは粗い離散値を取る目的関数値に擬似的に乱数を付与することで，非劣解の数を増加させる手法を提案している．

3.2 設計変数空間の離散化に関する研究

バイナリ型 GA における設計変数のビット長が進化に与える影響

設計変数空間の離散化に関する研究としては，主にバイナリ型遺伝的アルゴリズム（Binary-Coded Genetic Algorithms; BCGAs; バイナリ型 GA）において研究が進められている．バイナリ型 GA は実数値 GA と異なり，設計変数は 0，1 のバイナリビット列を持つアルゴリズムである．したがって，バイナリビット列のビット長がの長さにより，設計変数の取り得る値の数は減少するため，ビット長の長さを変えることが設計変数の粒度の変更と等価となる．バイナリ型 GA においては，設計変数のビット長を短くするほど解の収束速度が向上することが報告されている [?]．これは，設計変数のビット長を短くすることで，設計変数空間が粗く離散化され，探索空間の縮小に繋がるためだと考えられている．3.2.1 は，異なるビット長を用いた際の探索空間の違いを示しており，格子点が各ビット長を用いた際の探索点を表している．3.2.1 からわかるようにビット長を短くすることで，設計変数空間が粗く離散化され探索空間が減少していることがわかる．この性質を活かし，バイナリ型 GA ではビット長を探索の中で動的に変化させることで探索速度を向上させる手法が提案されている．

BCGA における設計変数の可変ビット長に関する研究

2005 年に Jaimes が提案した Multiple Resolutions Multiobjective Optimization Genetic Algorithm (MRMOGA)[?] は，設計変数空間の離散化を取り入れた島モデル型

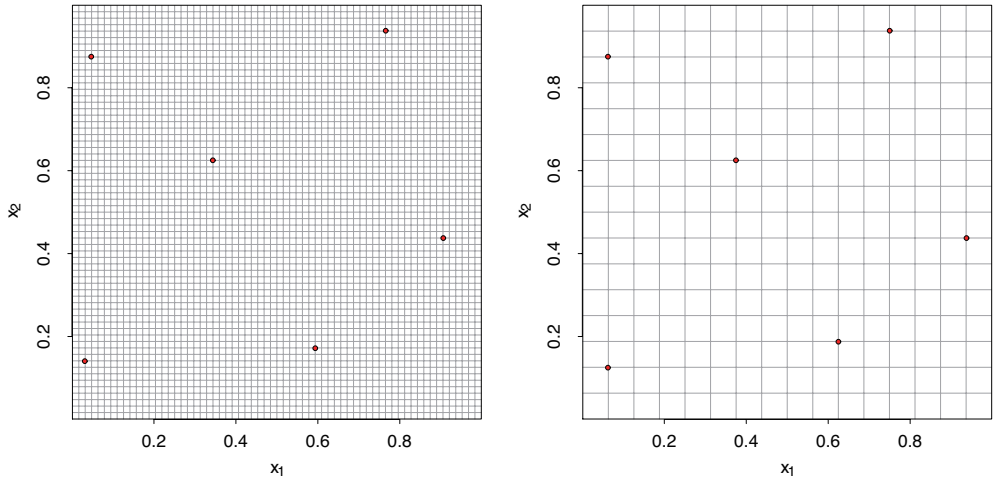


図 3.2.1: 離散化による探索空間のイメージ図 (左：ビット長 6 を用いた 0-1 の二変数の探索範囲　右：ビット長 4 を用いた 0-1 の二変数の探索範囲)

BCGA である．島モデル型の GA では，複数個の集団を並行して最適化を行い，ある条件下で各集団の個体を別の集団に移す（移民）することで，最適化を行う手法である．MRMOGA では，各島（集団）で異なるビット長で表現される設計変数空間を考え，進化初期では少ないビット長（細かい離散化）で最適化を行い，徐々に長いビット長（細かい離散化）に変化させることで，収束速度の向上と多様性の維持を実現している．

Kim らが提案した Variable Chromosome Length Genetic Algorithm (VCLGA)[?] も MRMOGA 同様にビット長を可変にした構造最適化アルゴリズムを提案している．VCLGA では，少ないビット長で解が収束した場合，その解をエリートとして保存し，複数個の保存した解と同様の個体とランダムに作成した個体を用いて，ビット長を増加させ，再度計算を行う．

MRMOGA，VCLGA ともに短いビット長から徐々に長いビット長に変化させることで効率的探索を目指している．

実数値 GA における設計変数の離散化の影響評価に関する先行研究

実数値 GA においては上述の MRMOGA や VCLGA のように動的に設計変数を離散化する手法の研究はおろか，設計変数の離散化が進化に与える影響についてもあまり議論されてこなかった．そこで，先行研究においては，実数値 GA において設計変

数の離散化が進化に及ぼす影響を NSGA-II と 3 つのベンチマーク問題を用いて評価した [?]．この研究では，設計変数の小数点以下の桁数を 2，4，6，8，16 桁に制御した場合の GD 値と IGD 値の推移を評価した．

表 3.2.1 には NSGA-II の計算条件を，表 3.2.2 には先行研究で用いたベンチマーク問題と各問題で設定した集団サイズと世代数をまとめた．

表 3.2.1: 先行研究で用いた共通パラメータ

パラメータ	数値
交叉率	1.0
突然変異率	1/n
η_c	30
η_m	20
試行回数	10

表 3.2.2: 先行研究で使用したベンチマーク問題ごとの計算条件

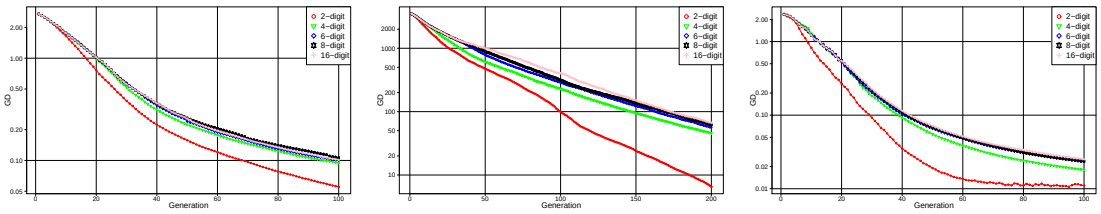
	DTLZ2	DTLZ3	DTLZ4
集団サイズ	100	500	300
世代	100	200	100

図 3.2.2 は，DTLZ2，3，4 それぞれの GD の世代ごとの推移を示している．図 3.2.2 より，いずれの問題においても，赤色の線で示した最も粗く離散化されている 2 桁の結果が良い結果となっていることが分かる．この結果からも，DTLZ 問題では粗く離散化するほど収束性が向上する傾向があることが分かる．

図 3.2.3 は，DTLZ2，3，4 それぞれの IGD の世代ごとの推移を示している．図 3.2.3 より，DTLZ2，3 は GD の結果と同様に，最も粗く離散化されている 2 桁の結果が良い結果となっていることが分かる．一方で，DTLZ4 の結果は，2 桁の結果が最も悪く，細かい粒度を持った 8 桁や 16 桁の結果が最も良い結果となっていることが確認できる．

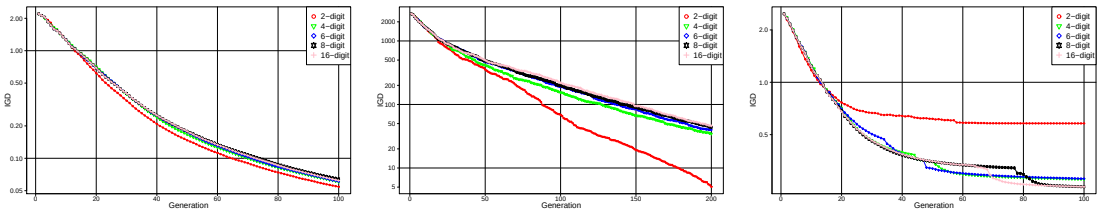
図 3.2.4，3.2.5，3.2.6 は，2 桁と 16 桁を用いた場合に DTLZ2，3，4 それぞれで最終的に得られた非劣解の分布である．図 3.2.4 から，DTLZ2 では，2 桁と 16 桁で得られる非劣解の分布には大きな違いはないことが分かった．図 3.2.5 から，DTLZ3 では，2 桁のほうが明らかに最適化方向である原点方向に進化が進んでおり，明らかに収束性が高いことが分かる．また，16 桁の結果では，極付近に解が集中して分布していることが分かる．これらの解は dominance-resistantsolutions (DRSs)[?] と呼ばれ，効率的な探索を妨げることが報告されている．DRSs の詳細については後述する．一方で，2 桁の結果では，DRSs はほとんど発生していないことが分かる．図 3.2.6 から，DTLZ4 では，2 桁は局所的にしか解が得られていないことが分かる．対照的に 16 桁は真のパレートフロントを覆うように非劣解が得られていることが分かる．このこと

から，DTLZ4 においては，粗い離散化をしすぎると解の多様性が失われてしまうことが考えられる．



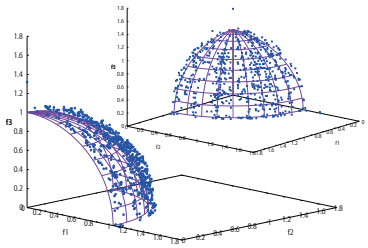
(a) DTLZ2　(b) DTLZ3　(c) DTLZ4

図 3.2.2: DTLZ2-4 における GD の推移（NSGA-II）

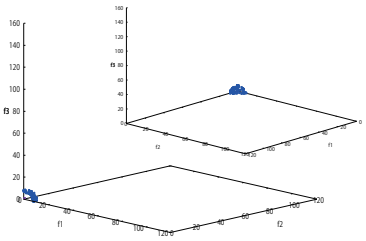


(a) DTLZ2　(b) DTLZ3　(c) DTLZ4

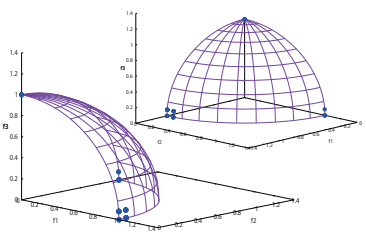
図 3.2.3: DTLZ2-4 における IGD の推移（NSGA-II）



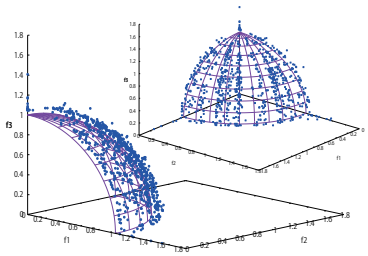
(a) 2-digit



(a) 2-digit

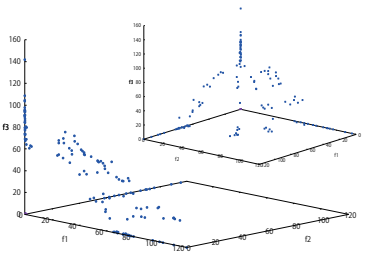


(a) 2-digit



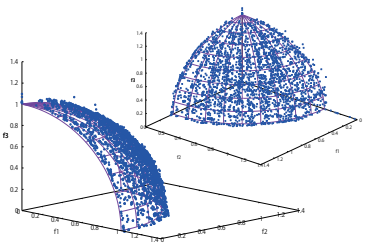
(b) 16-digit

図 3.2.4: DTLZ2 における非劣解の分布（NSGA-II）



(b) 16-digit

図 3.2.5: DTLZ3 における非劣解の分布（NSGA-II）



(b) 16-digit

図 3.2.6: DTLZ4 における非劣解の分布（NSGA-II）

先行研究では，制御する桁数の違いにより進化に影響が出る原因を調べるため，さ

らなる検証が行われた．離散化する粒度の違いにより結果に違いが生まれる一つの要因として交叉（SBX）と突然変異（polynominal mutation）による分布の違いが考えられる．そこで，桁数の違いによる SBX，polynominal mutation の分布を比較した．

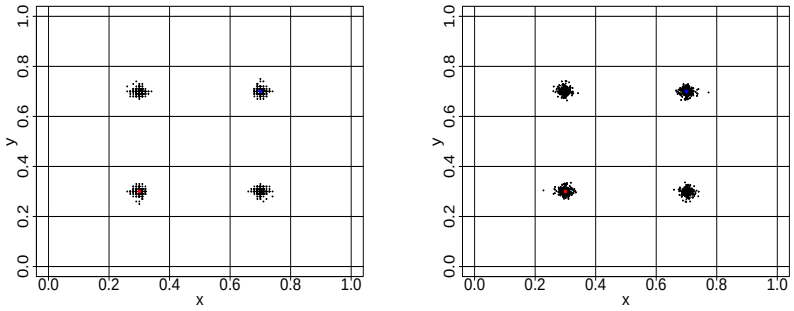
SBX に関しては $\eta_c = 30$，親個体を (0.3,0.3)，(0.7,0.7) に設定し，子個体がどのように分布するか 1000 回試行し，分布の違いを評価する．polynomial mutation に関しては，$\eta_m = 20$，親個体を (0.5,0.5) とし，SBX と同様に 1000 回試行し，分布の違いを評価する．

シミュレーション結果を図 3.2.7，図 3.2.8 に示す．また，図 3.2.9 には，生成される子個体の頻度ヒストグラムを示す．これらの図より，頻度に大きな違いが見られるが，取り得る値には大きな違いは見られない．それぞれの x 値，y 値ごとに Kolmogorov-Smirnov 検定を行い，分布の正規性を調べた結果，正規性であるという帰無仮説は棄却された．したがって，それぞれの分布には正規性がないと仮定し，Wilcoxon の順位和検定を行った．その結果，各分布に統計的有意な差があるという結果は得られなかった．故に，SBX と polynomial mutation には，有効桁数の違いによる影響はないと考えられる．

その他の原因を探るため，制御する桁数によって収束性に最も大きな差が見られる DTLZ3 に注目して検証する．図 3.2.2(b) より，2 桁と 16 桁では，20 世代目にはすでに GD に大きな差がついていることが分かる．

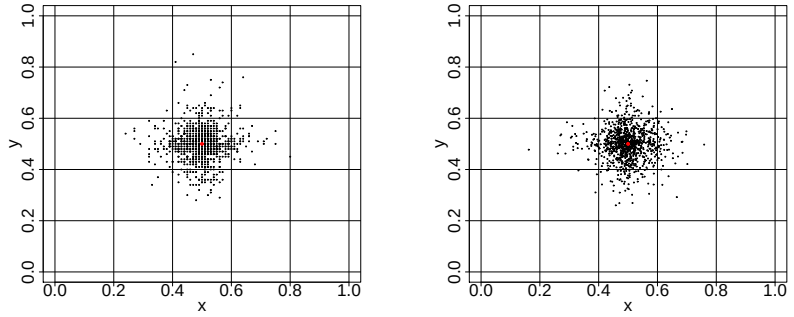
ここで，20 世代目における変数 x_1，x_2 の累積頻度ヒストグラムをそれぞれ図 3.2.10(a), (b) に示す．DTLZ 問題は，目的関数空間において解の位置を司る変数と，距離を司る変数に分けられている．x_1，x_2 はいずれも，目的関数空間において解の位置を司る変数であり，本研究ではこれらの変数のことを位置変数と呼ぶこととする．図 3.2.10(a), (b) を見ると，どちらも 0 もしくは 1 付近に解が集中して分布していることが分かる．これは，位置変数が 0 もしくは 1 付近に分布した場合，目的関数空間において極付近に解が生成されるため，非劣解になりやく，親子体として選択されることが多くなるためであると考えられる．これらの解は DRSs と呼ばれ，効率的な進化の妨げになることが報告されている．一方で，図 3.2.10(a)，(b) を比較してみると，2 桁のもののほうが 0 もしくは 1 付近の解が少ないことが分かる．したがって，2 桁は DRSs の発生を抑制することができる可能性があり，DRSs の発生を抑制することができるため

に解の収束性が向上していることが考えられる.



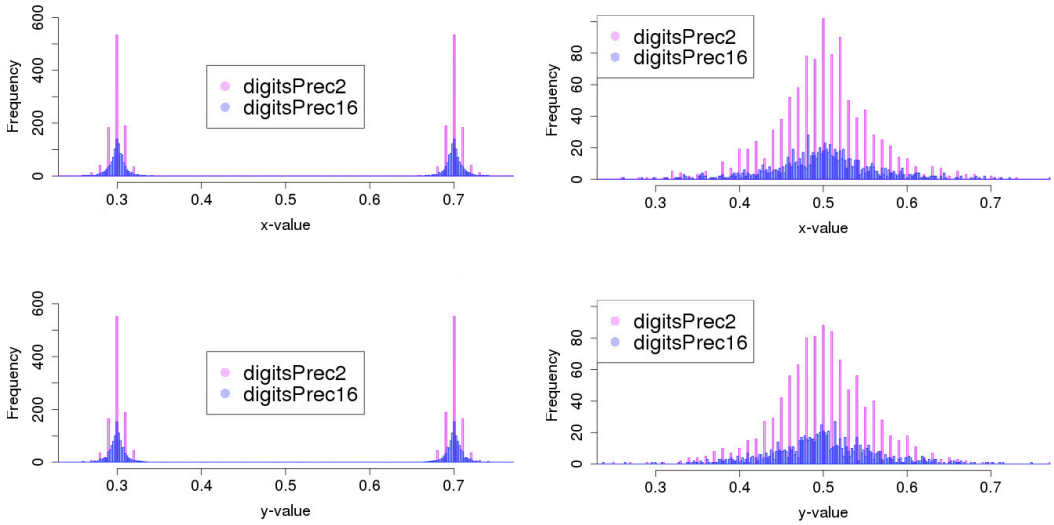
(a) 2 桁　　(b) 16 桁

図 3.2.7: 制御桁数の違いによる SBX の分布比較



(a) 2 桁　　(b) 16 桁

図 3.2.8: 制御桁数の違いによる polynomial mutation の分布比較



(a) SBX　　(b) polynomial mutation

図 3.2.9: SBX，polynominal mutation で得られる子個体の頻度ヒストグラム

この仮説を確認するため，DTLZ3 の 16 桁において，位置変数が 0 もしくは 1 付近に生成された場合に，その位置変数の桁数を 2 桁にすることで，DRSs の発生が抑制され進化が効率よく行われるか試みた (Modified Method)．今回は，位置変数 x_1，x_2 が 0.1 以下もしくは 0.9 以上の時に桁数を 2 桁に落とすこととした．計算条件は表 3.2.1 と同様である．

図 3.2.10(c) は，Modified Method を用いた場合の 20 世代目における位置変数 x_1，x_2 の累積頻度ヒストグラムを示したものである．また，図 3.2.11(a)，(b) には各桁数と Modified Method を用いた場合の GD，IGD の推移を，(c) には Modified Method を使用した場合に最終的に得られた非劣解の分布を示す．

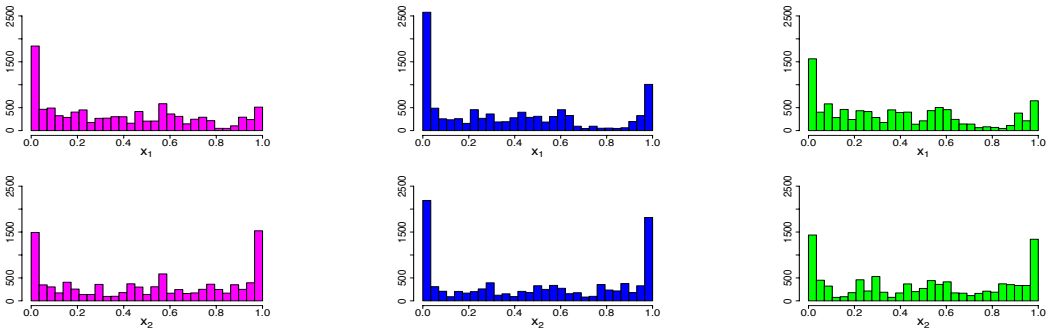
図 3.2.10 より，Modified Method は 16 桁と比較し，明らかに 0 もしくは 1 付近の解の分布の頻度が減少していることが分かる．このことから，DTLZ3 においては，桁数を小さくする（粗く離散化する）ことが DRSs の発生を抑制することに繋がっていることが分かった．また，図 3.2.11(a)，(b) より，Modified Method は 2 桁には及ばないものの，16 桁と比較し GD・IGD いずれも小さい値となっていることが分かる．このことからも，桁数を小さくすることで DRSs の発生が抑制されたため，効率的に進化が進み収束性や多様性が向上したものと考えられる．さらに，図 3.2.11(c) より，やや DRSs は残っているものの，16 桁（図 3.2.5(b)）と比較すると減少していることが分かる．

以上より，先行研究においては，NSGA-II と 16 個の多目的ベンチマーク問題を用いて設計変数の離散化の影響を評価した結果，粗く離散化するほど解の収束速度が向上することが分かった．一方で，DTLZ4 等の非線形性が非常に強い問題では，粗く離散化しすぎると解の多様性が失われてしまう可能性があることが分かった．また，DTLZ3 においては，粗く離散化することで DRSs の発生が避けられ，解の収束性が向上したことが確認できた．これらの結果から，問題に応じて各設計変数を適切に設定することで，効率的に探索を進めることができる可能性があることが分かった．さらに，動的に離散化する大きさを変更することが，効率的探索に繋がる可能性を示唆した．

しかしながら，先行研究では NSGA-II のみでしか影響が評価されておらず，ベンチマーク問題も DTLZ 問題のみであったため，他の実数値 GA やベンチマーク問題にお

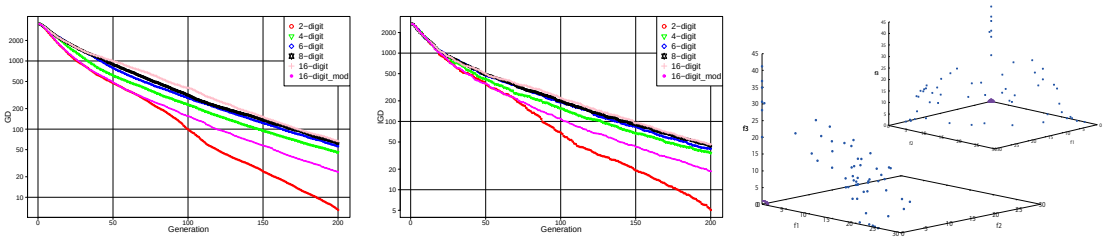
いても同様の現象が確認できるかなど，未だ不明確な部分も多い．また，設計変数の離散化の活用法についても，研究の余地が残されている．

そこで本研究では，先行研究の NSGA-II に加え，代表的実数値 GA の一つである MOEA/D での検証を行い，様々な特徴を持つ 19 個のベンチマーク問題を用いて影響を評価する．また，設計変数の離散化の活用法を考察し，効率的探索に繋がる手法の提案を行う．



(a) 2-digit　(b) 16-digit　(c) modified 16-digit

図 3.2.10: 2 桁，16 桁，Modified Method における位置変数 x_1，x_2 の累積頻度ヒストグラム



(a) History of GD　(b) History of IGD　(c) modified 16-digit

図 3.2.11: Modified Method における GD，IGD の推移と非劣解集合

DRSs

NSGA-II などの，解の優越性に基づいて個体を評価する最適化アルゴリズムでは，解空間の極付近に存在する解が，非劣解として生存しやすくなる．このような解は，他の個体に支配されにくく，他の非劣解より収束が悪くても生存しやすい．DRSs とは，非劣解の中でも収束性が悪いにも関わらず，長い世代淘汰されずに残り続ける解のことである．DRSs のイメージを図 3.2.12 に示す．

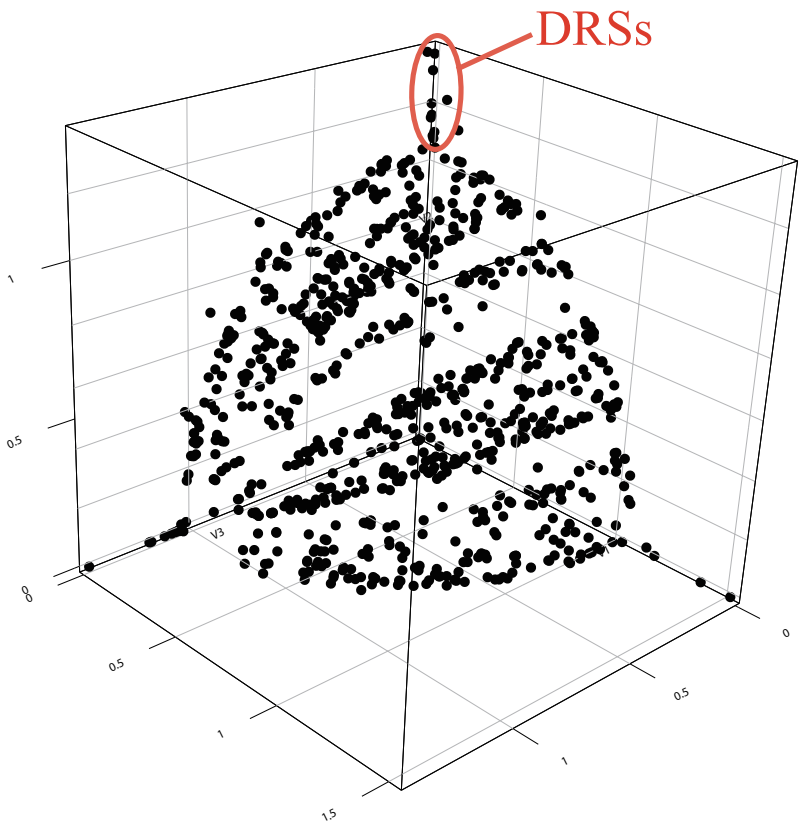


図 3.2.12: DRSs のイメージ図

DRSs が多く存在すると，その周辺に次の個体が多く生成されるため，進化の効率が悪くなってしまうことが知られている．