

多目的最適化手法 NSGA-II における 等距離選択の効果について

広島修道大学商学部

阪井 節子 (Setsuko Sakai)

Faculty of Commercial Sciences, Hiroshima Shudo University

広島市立大学大学院 情報科学研究科 高濱 徹行 (Tetsuyuki Takahama)

Department of Intelligent Systems, Hiroshima City University

1 はじめに

社会的要請の多様化に伴い、相競合する評価尺度を持つ複数の目的関数を同時に最小化する解を求める、多目的計画法が注目されてきている。多目的計画法では、目的関数がベクトル値関数となり、全ての目的関数を最小化する完全最適解が必ず存在するとは限らない。このため、多目的計画問題においては、各目的関数値のトレードオフの扱い方が異なる数多くの解概念が提案されている。その中で最も代表的な解の1つが Pareto 最適解である。

従来、Pareto 最適解を求める方法としては、以下のような方法が提案されている [1, 2, 3]。

(a) スカラー化法

多目的最適化問題の各目的関数を組み合わせて、単一目的最適化問題に変換する方法である。荷重平均法、制約法、荷重 minimax 法などが含まれる。

(b) 目標計画法と妥協計画法

目的関数値ベクトルと意思決定者が達成しようと考えている目標値ベクトルの距離を最小化する方法である。

(c) 対話的手法

対話によって得られる意思決定者の局所的な選好情報に基づいて、実行可能解を最適解の方向へ改良する方法である。

しかし、(a)、(b) では多目的最適化問題から単一目的最適化問題への変換方法によっては、意思決定者の選好を十分に反映することが困難であり、(c) では意思決定者の局所的な選好を取り入れられるが、Pareto 最適解全体を比較するような大局的な選好を取り入れることは困難である、等の問題点がある。

多目的最適化においては、Pareto 最適解の全体集合を求め、意思決定者が Pareto 最適解の全体集合の中から最終的な選好解を選択することにより、意思決定者の選好をより適切に反映させることが非常に重要である。(a)、(b) でも、パラメータを変化させながら繰り返し問題を解くことによって Pareto 最適解の全体集合を求めることができるが、目的関数の定義域の次元が増加するにつれ、膨大な計算時間が必要となる。このため、多目的最適化問題を単一目的最適化問題に変換することなく、Pareto 最適解集合を直接求める手法を提案することが必要となっている [4, 5]。

Pareto 最適解集合を求める最適化手法の1つに、進化的アルゴリズム (Evolutionary Algorithms; EAs) を用いる方法がある。進化的アルゴリズムは、解集団による多点探索を行うため、解候補の多様性を維持し、真の Pareto 最適解集合への選択圧力を与えることにより、一回の実行によって複数の Pareto 最適解を発見することができる。進化的アルゴリズムによる多目的最適化手法は、多目的進化的アルゴリズム (Multiobjective Evolutionary Algorithms: MOEAs) と呼ばれる。最も初期の提案は Saffer [6] によってなされた。以来、数多くの研究がなされているが、NSGA-II [7] は、効果的かつ安定性の高い最も優れた多目的進化的アルゴリズムの1つである。

多目的進化的アルゴリズムにより得られる非劣解の数は、集団の個体数に依存する。そのため、真の Pareto 最適解集合全体を均等に被覆する非劣解集合を求めることが非常に重要である。NSGA-II は、“Simulated binary crossover (SBX)”と“parameter-based mutation (PBM)”を用いることによって、上下限制約で定義された探索空間内の親の近傍に子を生成する。しかし、境界近傍の非劣解を発見する確率が極端に小さい

ために、境界近傍にある Pareto 最適解を得ることは、ほとんど不可能である。また、NSGA-II は、親の選択と生存者選択において、目的関数値の空間における関数値ベクトルの混雑度を表す混雑距離を用いて解選択を行う。これにより、混雑度の低い解を選択し、解の多様性を維持し、かつ極端解を含めて広範囲に広がる解を得ることができる。しかし、混雑距離では混み合っている解は全て選択されないため、結果として均等に位置する解を得ることが困難である。

本研究では、境界付近の解を求め、かつ等間隔に広がる解を得るために、NSGA-II に対して、拡張 SBX と等距離選択を導入することを提案する。拡張 SBX では、子の生成範囲を拡張することによって、境界付近に解が生成されるようにする。等距離選択では、等間隔の位置にある解の混雑距離を増加させることによって、結果として選択された解が等間隔に配置されるようにする。拡張 SBX および等距離選択を用いた NSGA-II (改良版 NSGA-II) をいくつかの最適化問題に対して適用する。また、それらの結果を、従来の NSGA-II による結果と比較することによって、改良版 NSGA-II が従来の NSGA-II よりも良い非劣解集合を得られる多目的進化的アルゴリズムであることを示す。

以下、2. では多目的最適化問題を定義し、3. では、NSGA-II を説明する。4. では、改良版 NSGA-II を提案する。5. では制約なし多目的最適化問題に対して改良版 NSGA-II を用いた結果について述べ、NSGA-II、SBX のみを拡張 SBX に置き換えた NSGA-II を用いた結果と比較する。6. はまとめである。

2 多目的最適化問題

一般に、与えられた制約条件の下で、複数個の相競合する目的関数を最小化（あるいは最大化）する問題は多目的最適化問題（あるいは多目的計画問題）と呼ばれる。複数個の目的関数をベクトル値関数として形式化すれば、多目的最適化問題は、ベクトル値最小化問題として以下のように定式化できる。

$$(P) \quad \begin{array}{l} \text{minimize } f(\mathbf{x}) \\ \text{subject to } \mathbf{x} \in X = \{\mathbf{x} \in R^n | g(\mathbf{x}) \leq 0\} \end{array}$$

ここで、 $\mathbf{x} = (x_1, x_2, \dots, x_n)$ は n 次元決定変数ベクトル、 $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))$ は m 次元ベクトル値関数、 $g(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), \dots, g_k(\mathbf{x}))$ は k 次元ベクトル値制約関数である。

このように、多目的最適化問題は k 個の不等式制約条件の下で m 個の目的関数を同時に最小化する n 次元の決定変数ベクトルを求める問題として定式化される。なお、制約のない多目的最適化問題では $k = 0$ となる。ところが多目的最適問題では、目的関数がベクトル値関数であるため、目的関数値の間に半順序関係しか成り立たない。そのため、一般に全ての目的関数を同時に最小化する完全最適解は存在しない。そこで、多目的最適化問題の最適解として種々の解概念が提案されているが、その最も代表的な解が Pareto 最適解である。

Pareto 最適解 $\mathbf{x}^* \in X$ は、以下の条件を満足する解である。

$$f(\mathbf{x}) \leq f(\mathbf{x}^*) \text{ となる } \mathbf{x} \in X \text{ が存在しない。} \quad (1)$$

ここで、ベクトル値の大小関係として、以下の比較演算子を定義しておく。

$$\begin{aligned} f \leq f^* &\Leftrightarrow \forall i f_i \leq f_i^* \\ f \leq f^* &\Leftrightarrow \forall i f_i \leq f_i^* \text{ and } \exists j f_j < f_j^* \\ f < f^* &\Leftrightarrow \forall i f_i < f_i^* \end{aligned}$$

なお、本論文では、 X の部分集合 A において (1) を満足する解を、問題 (P) の Pareto 最適解と区別するために、特に A の非劣解と呼ぶことにする。

3 非優越ソーティングを行う遺伝的アルゴリズム (NSGA-II)

多目的進化的アルゴリズムでは、多目的最適化を実現するために、特別な個体の選択方法と集団の多様性を維持する方法が採用される。

個体選択は、真の Pareto 面へ探索を進めるために用いられる。Schaffer[6] は、各目的関数毎に独立にルーレット選択を使用することによって部分集団を生成するという選択方法を提案している。Goldberg[8] は、集団内のすべての非劣解にランク 1 を割当て、順次、ランク 1 から $i-1$ までのすべての非劣解を除いた集団内のすべての非劣解にランク i を割当てるといった選択方法を提案している。

集団の多様性を維持する方法は、解が集団内に一様に分布し、広範囲に広がる非劣解を求めるために用いられる。解の密度を推定するためには、ニッチ数、混雑距離、squeeze factor 等の評価尺度が提案されている。

本節で説明する NSGA-II は、解のランキングに Goldberg による Pareto ランク法を用い、ランク付けのためには高速非優越ソート (fast non-dominated sort) 法を用いる。また、個体選択には、Pareto ランクに基づいたタイブレークを持つバイナリ・トーナメント選択を採用し、適合度の割当てには混雑距離を用いる優れた多目的進化的アルゴリズムである。

3.1 混雑距離 (Crowding distance, CD)

NSGA-II では、1つの解の周りにおける解の密度を推定するために、その解の両隣にある2つの解の平均距離である混雑距離 (crowding distance) を用いる。

i 番目の解 $x^{(i)}$ の混雑距離は、 $x^{(i)}$ に関数値が最も近接する $i-1$ 番目と $i+1$ 番目の解 $x^{(i-1)}$ と $x^{(i+1)}$ によって次のように定義される：

$$CD(x^{(i)}) = \frac{1}{k} \sum_{j=1}^k |\bar{f}_j(x^{(i+1)}) - \bar{f}_j(x^{(i-1)})|, \quad (2)$$

ここで、 $\bar{f}_j(\cdot)$ は j 番目の目的関数 $f_j(\cdot)$ をその値域幅で正規化した目的関数である。

図1が示すように、混雑距離は両隣の解によって作られる四角形の周囲の長さの平均である。

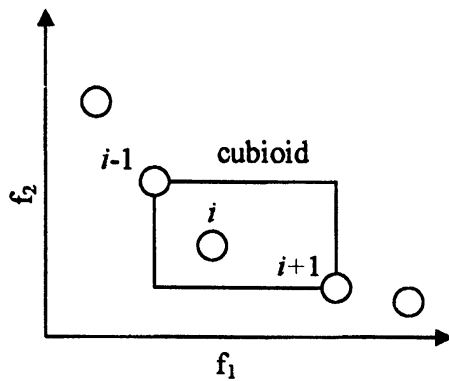


図1: 混雑距離の定義

混雑距離の計算アルゴリズムを以下に示す。

```
crowding_distance_assignment(I) {
  N = |I|;
  for(i = 1; i <= N; i++) CD[i] = 0;
  for(each objective j) {
    sort I according to the value of
      j-th objective value;
    CD[1] = CD[N] = ∞;
    for(i = 2; i < N; i++)
      CD[i] = CD[i] + 1/k * (I[i+1].j - I[i-1].j) /
        (f_j^max - f_j^min);
  }
}
```

ここで、 I は解の集合、 $CD[i]$ は i 番目の解の混雑距離である。 $I[i].j$ は、 i 番目の解の j 番目の目的関数値であり、 f_j^{\max} と f_j^{\min} はそれぞれ j 番目の目的関数の最大値と最小値である。ただし、極端解を保存するために各目的関数について最良解と最悪解の混雑距離は ∞ としている。

3.2 NSGA-II の全体アルゴリズム

NSGA-II のアルゴリズムの概要を以下に説明する。

1. 初期集団の生成

大きさ N の初期個体集団 P をランダムに生成する。

2. ランク付けと混雑距離の割当て
非優越ソートを用いて、 P の各個体にランクを付ける。さらに、各ランク毎に属する個体に混雑距離を割当てる。
3. 子の生成
 P に対して、選択、交叉、突然変異の遺伝子操作を行い、大きさ N の子集団 Q を生成する。
4. 世代交代
非優越ソートを用い、親と子の全体 $P \cup Q$ の各個体にランクを付ける。また、各ランク毎に個体をソートし混雑距離を割当てる。ランク上位 N 番目までの個体によって、集団 P を構成する。ただし、ランクが同じ個体については混雑距離が大きい個体を優先して選択する。
5. 3にもどる。

NSGA-II の全体アルゴリズムを以下に記述する:

```

NSGA-II() {
  t=0;
  P(0)=randomly generated N individuals;
  ranking P(0) using
    fast_nondominated_sort(P(0));
  while(終了条件が満たされない) {
    Q(t)=generate offspring using P(t);
    R(t)=P(t) ∪ Q(t);
    ranking R(t) using
      fast_nondominated_sort(R(t));
    P(t+1)=∅;
  }
  for(i=1; |P(t+1)| + |Fi| ≤ N; i++) {
    crowding_distance_assignment(Fi);
    P(t+1)=P(t+1) ∪ Fi;
  }
  if(|P(t+1)| < N) {
    crowding_distance_assignment(
      Fi[1 : N - |P(t+1)|]);
    sort Fi according to
      the crowding distance;
    P(t+1)=P(t+1) ∪ Fi[1 : N - |P(t+1)|];
  }
  t=t+1;
}

```

ここで、 $\text{fast_nondominated_sort}(\cdot)$ は高速非優越ソートで解集合をソートする関数、 F_i は、 i 番目の Pareto 面、 $F_i[n : m]$ は F_i に属する n 番目から m 番目までの要素で構成される F_i の部分集合である。

NSGA-II において、3. 子の生成で用いられる “Simulated Binary Crossover (SBX)”, “Parameter-based Mutation(PBM)” について次節で説明する。

3.3 Simulated Binary Crossover (SBX)

SBX では、次の多項式確率分布にしたがって 2 個の親から 2 個の子を生成する:

$$C(\beta) = \begin{cases} 0.5(\eta_c + 1)\beta^{\eta_c} & \text{if } \beta \leq 1, \\ 0.5(\eta_c + 1)\frac{1}{\beta^{\eta_c+1}} & \text{if } \beta > 1. \end{cases} \quad (3)$$

ここで、 η_c は非負の分布パラメータである。

親 x_1, x_2 ($x_1 < x_2$) から、子 c_1, c_2 ($x_1 \leq c_i \leq x_u, i = 1, 2$) を次のように生成する:

1. 区間 $[0, 1]$ 上の一様乱数 u を生成する。
2. 確率分布 $C(\beta)$ によって、 $\beta_{q,i}$ ($i = 1, 2$) を求める。

$$\beta_{q,i} = \begin{cases} (u\alpha_i)^{\frac{1}{\eta_c+1}} & \text{if } u \leq \frac{1}{\alpha_i}, \\ \left(\frac{1}{2-u\alpha_i}\right)^{\frac{1}{\eta_c+1}} & \text{otherwise,} \end{cases} \quad (4)$$

$$\alpha_i = 2 - \beta_i^{-(\eta_c+1)}, \quad (5)$$

$$\beta_1 = 1 + \frac{2(x_1 - x_l)}{x_2 - x_1}, \quad \beta_2 = 1 + \frac{2(x_u - x_2)}{x_2 - x_1}. \quad (6)$$

3. 子を生成する:

$$c_1 = 0.5\{(x_1 + x_2) - \beta_{q,1}(x_2 - x_1)\}, \quad c_2 = 0.5\{(x_1 + x_2) + \beta_{q,2}(x_2 - x_1)\}. \quad (7)$$

なお、ベクトルに対して SBX を適用するときは、通常、各成分について確率 $1/2$ で SBX を適用する。

3.4 Parameter-based Mutation (PBM)

PBM は、次のような多項式確率分布にしたがって親の近くに解を生成する:

$$C(\delta) = 0.5(\eta_m + 1)(1 - |\delta|)^{\eta_m}, \quad \delta \in [-1, 1] \quad (8)$$

ここで、 η_m は非負の分布パラメータである。親 x から子 c ($x_l \leq c \leq x_u$) を次のように生成する:

1. 区間 $[0, 1]$ 上の一様乱数 u を生成する。
2. 確率分布 $C(\delta)$ によって、値 δ_q を求める。

$$\delta_q = \begin{cases} \{2u + (1 - 2u)(1 - \delta_l)^{\eta_m + 1}\}^{\frac{1}{\eta_m + 1}} - 1 & \text{if } u \leq 0.5, \\ 1 - \{2(1 - u) + (2u - 1)(1 - \delta_u)^{\eta_m + 1}\}^{\frac{1}{\eta_m + 1}} & \text{otherwise,} \end{cases} \quad (9)$$

$$\delta_l = \frac{x - x_l}{x_u - x_l}, \quad \delta_u = \frac{x_u - x}{x_u - x_l}. \quad (10)$$

3. 解は次式で生成される:

$$c = x + \delta_q(x_u - x_l). \quad (11)$$

なお、ベクトルに対して PBM を適用するときは、通常、変異が確率的にある一つの成分で起こるように変異確率を $1/n$ とする。

4 改良版 NSGA-II

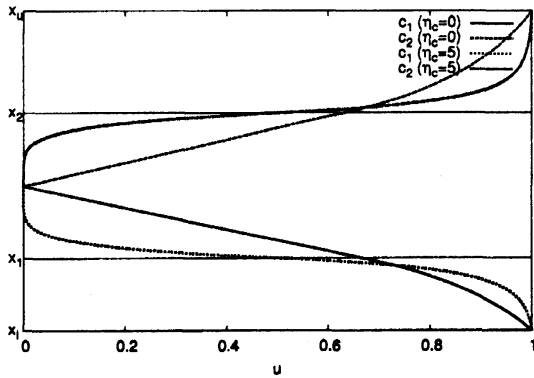
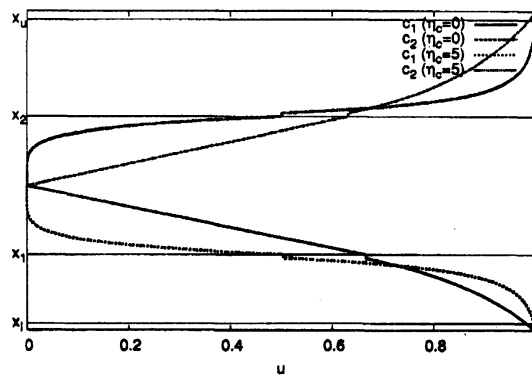
前述した NSGA-II は、多目的進化的アルゴリズムの中でも特に優れた手法の一つである。本研究では、SBX と混雑距離による個体選択を改良することによって NSGA-II を改良する。

4.1 拡張 SBX(Extended SBX)

SBX において、 η_c が 0 と 5 の場合、 u, x_i に対して生成された c_i を図 2 に示す。図 2 から分かるように、上限及び下限の境界付近の解の発見確率が極端に小さいため、境界上に解を生成するのは非常に困難である。この問題を解決するために、ここでは、SBX に対して拡張パラメータ α_c を導入し、境界上および外部に解を生成するために式 (4) を次のように変更する:

$$\beta_{q,i} = \begin{cases} (u\alpha_i)^{\frac{1}{\eta_c + 1}} & \text{if } u \leq \frac{1}{\alpha_i}, \\ (1 + \alpha_c)\left(\frac{1}{2 - u\alpha_i}\right)^{\frac{1}{\eta_c + 1}} & \text{otherwise.} \end{cases} \quad (12)$$

図 3 に、 $\eta_c = 0, 5, \alpha_c = 0.05$ とした場合の u, x_i に対して生成された c_i の間の関係を示す。図 3 では、小さい確率ながら探索空間の外部に点が生成されており、それらの解は境界上の解に変更される。

図 2: SBX における u , 親, 子の関係図 3: 拡張SBX における u , 親, 子の関係

4.2 等距離選択

NSGA-II では、混雑距離が大きい解、すなわち周囲が疎な解を選択することによって、解の多様性と解の分布の広がり維持している。混雑距離を用いた場合 (図 4 の左図)、混雑距離が大きい点を選択されるため、解が集中している場所にある解はほとんど選択されず、その領域が空白域となってしまう場合がある (●が選択された点である)。これを回避するために、ここでは等間隔に解を選択する等距離選択を提案する。等距離選択を用いた場合 (図 4 の右図)、ほぼ等間隔に点が選択される。

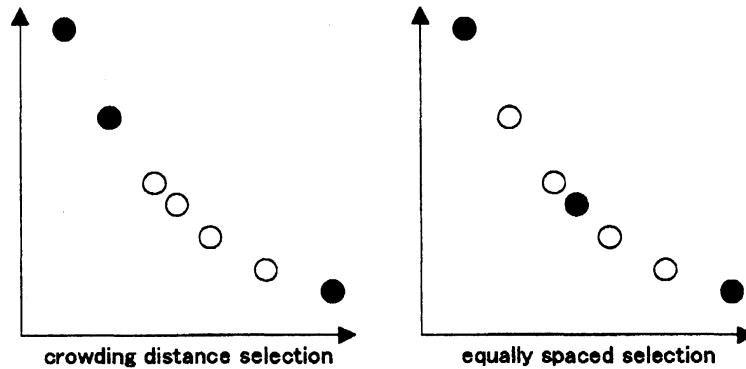


図 4: 混雑距離による選択と等距離選択の違い

等距離選択は次のように記述される。

1. 距離の初期化：目的関数値の辞書式順序によって解をソートする。1番目の解から l 番目の解までのバスの長さ $D[l]$ および最大混雑距離 CD_{max} (ただし、 ∞ を除く) を計算する。
2. 等間隔距離の計算：等間隔距離は、現在の解 (l 番目の解) から最後の解までを等間隔に分けた長さ $\frac{D_{max} - D[l]}{N_{sel} + 1}$ として与えられる。ここで、 D_{max} は第 1 番目の解から最後の解までの長さ、 N_{sel} は選択される解の個数である。
3. 解の選択：等間隔距離内にあり、各等間隔点に最も近い位置にある点を選択し、その点の混雑距離に CD_{max} を加える。これにより、この解が選択されることを保証する。また、その等間隔距離内に解が存在しない場合は、等間隔距離の点に最も近い解が選択される。
4. 十分な個数の解が選択されるか、最後の解が選択されるまで、2. に戻る。

等距離選択のアルゴリズムを以下に示す。

```

equally_spaced_distance_assignment(I)
{
  目的関数値によって I をソートする;
  CDmax=0; N∞=1; D[1]=0;
  for(k=2; k ≤ |I|; k++) {
    D[k]=D[k-1] + ||I[k-1] - I[k]||;
    if(CD[k] == ∞) N∞++;
    else if(CD[k] > CDmax)
      CDmax=CD[k];
  }
  Dmax=D[|I|];
  Nsel=N - |P(t+1)| - N∞;
  if(Nsel > 0) {
    dtarget=Dmax/(Nsel + 1);
    l=1;
    for(k=1; k ≤ Nsel; k++) {
      l0=l;
      while(l < |I| && D[l] < dtarget) l++;
      if(l > l0) l--;
      if(CD[l] != ∞) CD[l]=CD[l] + CDmax;
      if(l ≥ |I|) break;
      dtarget=D[l] + (Dmax - D[l])/(Nsel - k + 1);
      l++;
    } }
}

```

5 実験

本研究では、文献 [7] をはじめとするいくつかの研究で取り上げられている 7 個の多目的最適化問題を対象に、提案した拡張 SBX と等距離選択を用いた改良版 NSGA-II, NSGA-II, SBX のみを拡張 SBX に変更した NSGA-II を用いて最適化を行い、結果を比較検討する。

5.1 テスト問題

以下に示す 7 つの問題 [7] をテスト問題として使用する。

問題	n	変数領域	目的関数	Pareto 最適解	Pareto 面
SCH	1	$[-10^3, 10^3]$	$f_1(\mathbf{x}) = x^2, f_2(\mathbf{x}) = (x-2)^2$	$x \in [0, 2]$	凸集合
FON	3	$[-4, 4]$	$f_k(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^3 \left(x_i + \frac{(-1)^k}{\sqrt{3}}\right)^2\right)$ ($k = 1, 2$)	$x_1 = x_2 = x_3$ $\in [-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]$	非凸集合
ZDT1	30	$[0, 1]$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - \sqrt{x_1/g(\mathbf{x})}\right]$ $g(\mathbf{x}) = 1 + 9 \left(\sum_{i=2}^n x_i\right) / (n-1)$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \dots, n$	凸集合
ZDT2	30	$[0, 1]$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - (x_1/g(\mathbf{x}))^2\right]$ $g(\mathbf{x}) = 1 + 9 \left(\sum_{i=2}^n x_i\right) / (n-1)$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \dots, n$	非凸集合
ZDT3	30	$[0, 1]$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - \sqrt{x_1/g(\mathbf{x})} - (x_1/g(\mathbf{x})) \sin(10\pi x_1)\right]$ $g(\mathbf{x}) = 1 + 9 \left(\sum_{i=2}^n x_i\right) / (n-1)$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \dots, n$	凸集合 非連結
ZDT4	10	$[0, 1]$ $[-5, 5]$ $i = 2, \dots, n$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - \sqrt{x_1/g(\mathbf{x})}\right]$ $g(\mathbf{x}) = 1 + 10(n-1) + \sum_{i=2}^n [x_i^2 - 10 \cos(4\pi x_i)]$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \dots, n$	非凸集合
ZDT6	10	$[0, 1]$	$f_1(\mathbf{x}) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $f_2(\mathbf{x}) = g(\mathbf{x}) \left[1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2\right]$ $g(\mathbf{x}) = 1 + 9 \left[\left(\sum_{i=2}^n x_i\right) / (n-1)\right]^{0.25}$	$x_1 \in [0, 1]$ $x_i = 0$ $i = 2, \dots, n$	非凸集合 分布に 偏り有り

5.2 実験条件と性能指標

すべての問題に対して、SBXに関するパラメータは、交叉率 $P_c = 0.9$ 、分布パラメータ $\eta_c = 20$ とし、各変数は、確率 0.5 で SBX によって変化する。PBMに関するパラメータは、突然変異率 $P_m = 1/n$ 、分布パラメータ $\eta_m = 20$ とする。ここで、 n は変数の数である。集団サイズ (探索点数) $N = 100$ 、最大世代数 T は、問題 SCH, FON では 50, ZDT1, ZDT2, ZDT3 では 100, ZDT4, ZDT6 では 200 とした。以上の条件の下で、試行を 10 回行った。

異なったアルゴリズムによって得られた非劣解集合を比較する尺度としては、GD 尺度 [9]、 Δ 尺度 [7]、SC 尺度 [10]、支配尺度 [11] を用いた。

1. GD 尺度 (Generational Distance metric)

GD は非劣解集合 S の真の Pareto 最適解集合 P^* への収束度を評価する距離で、次式で定義される。

$$GD(S, P^*) = |S|^{-1} \sum_{\mathbf{x} \in S} d(\mathbf{x}, P^*), \quad (13)$$

$$d(\mathbf{x}, P^*) = \min_{\mathbf{y} \in P^*} \|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| \quad (14)$$

GD は、値が小さいほど S が真の Pareto 最適解集合 P^* に収束していることを意味する。 $S = P^*$ のとき、 $GD(S, P^*) = 0$ である。

2. Δ 尺度 (Δ metric)

Δ 尺度は、非劣解集合 S の多様性を評価するための距離で、次式で定義される。

$$\Delta(S, P^*) = \frac{\sum_{i=1}^m d(\mathbf{e}_i, S) + \sum_{\mathbf{x} \in S} |d(\mathbf{x}, S) - \bar{d}|}{\sum_{i=1}^m d(\mathbf{e}_i, S) + |S|\bar{d}} \quad (15)$$

$$d(\mathbf{x}, S) = \min_{\mathbf{y} \in S \setminus \{\mathbf{x}\}} \|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\|, \quad \bar{d} = |S|^{-1} \sum_{\mathbf{x} \in S} d(\mathbf{x}, S) \quad (16)$$

ここで、 \mathbf{e}_i は f_i のみを最適化した極端解であり、 $S \setminus \{\mathbf{x}\}$ は S から 1 点 \mathbf{x} を除いた集合である。

解が一様に分布すれば $d(\mathbf{x}, S) = \bar{d}$ となり、広範囲に分布すれば極端解 \mathbf{e}_i が S に属するようになる。よって、 Δ 尺度は、値が小さいほど、解が一様にかつ広範囲に分布していることを意味する。

3. SC 尺度 (Set Coverage metric)

2つのアルゴリズム (A), (B) による非劣解集合がそれぞれ A, B であるとき、 B に対する A の SC 尺度 $SC(A, B)$ は次式で定義される。

$$SC(A, B) = \frac{|\{b \in B | \exists a \in A, \mathbf{f}(a) < \mathbf{f}(b)\}|}{|B|} \quad (17)$$

SC 尺度は非対称であるため、 $SC(A, B)$ と $SC(B, A)$ の両方を求める必要がある。 $SC(A, B)$ は、 B における A の解に優越されている解の割合である。よって、 $SC(A, B)$ が $SC(B, A)$ より十分大きいということが繰り返し観測されれば、(A) の方が (B) よりも優れたアルゴリズムであると考えられる。

4. 支配尺度 (Domination Metric)

2つのアルゴリズム (A), (B) による非劣解集合がそれぞれ A, B であるとき、 B に対する A の支配尺度 $Dom(A, B)$ は、次式で定義される：

$$Dom(A, B) = \frac{dom(A, B)}{dom(A, B) + dom(B, A)} \quad (18)$$

$$dom(X, Y) = \sum_{\mathbf{x} \in X} |\{y \in Y | \mathbf{f}(\mathbf{x}) < \mathbf{f}(\mathbf{y})\}| \quad (19)$$

支配尺度では、 $Dom(B, A) = 1 - Dom(A, B)$ より、 $Dom(A, B)$ を求めるだけで十分である。 $Dom(A, B)$ は、優越関係が決定できる A と B の解の対の中で、 A の解が B の解を優越する対の割合である。よって、 $Dom(A, B)$ が十分大きいということが繰り返し観測されれば、(A) の方が (B) よりも優れたアルゴリズムであると考えられる。

5.3 実験結果

図 5-11 は、それぞれ SCH, FON, ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 に対する 5 回目と 6 回目の試行で得られた非劣解である。ここで、NSGA-II, extended, uniform はそれぞれ NSGA-II, 拡張 SBX を用いた NSGA-II(拡張 SBX 版 NSGA-II と表す), 拡張 SBX と等距離選択を用いた NSGA-II(改良版 NSGA-II と表す) による非劣解集合を表す。NSGA-II, 拡張 SBX 版 NSGA-II, 改良版 NSGA-II, いずれの手法でも, Pareto 最適解集合に近い非劣解集合を発見できている。しかし, ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 では, 改良版 NSGA-II の方が NSGA-II よりも良い解を発見できている。

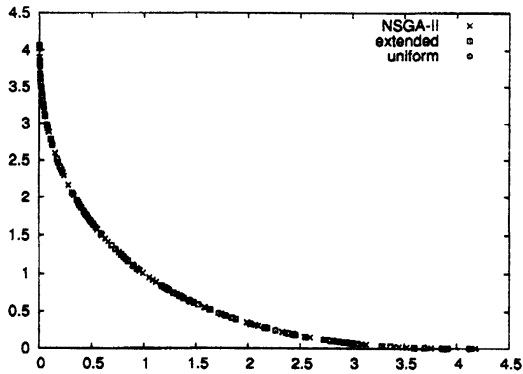


図 5: SCH の非劣解集合

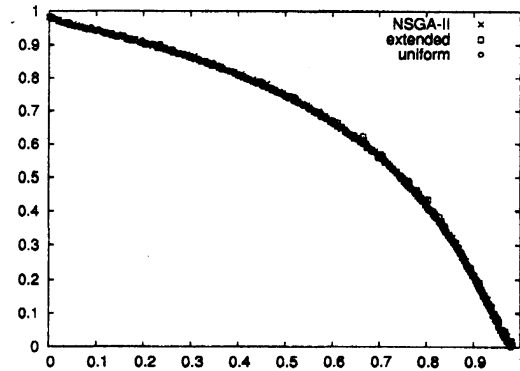


図 6: FON の非劣解集合

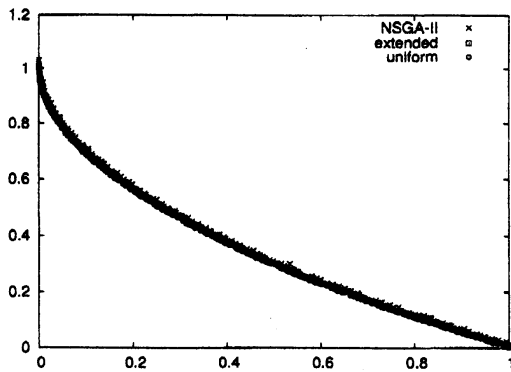


図 7: ZDT1 の非劣解集合

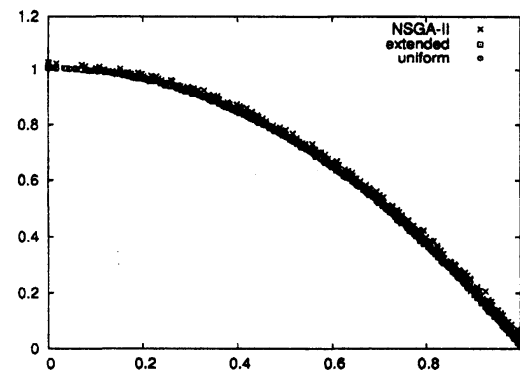


図 8: ZDT2 の非劣解集合

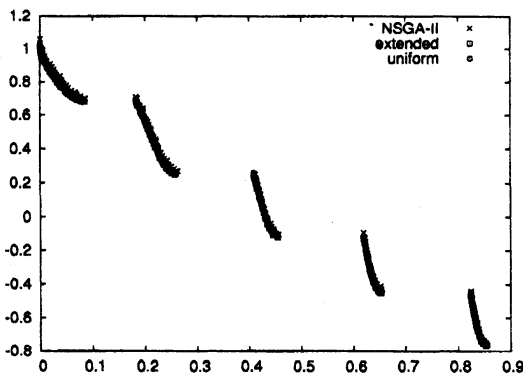


図 9: ZDT3 の非劣解集合

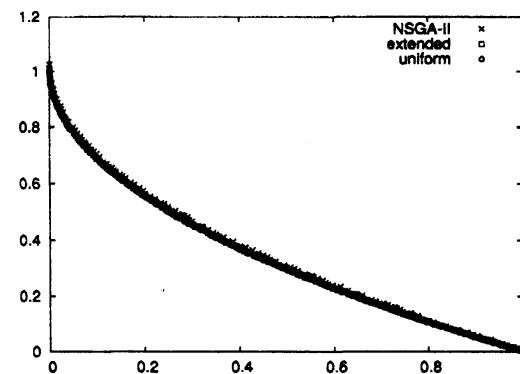


図 10: ZDT4 の非劣解集合

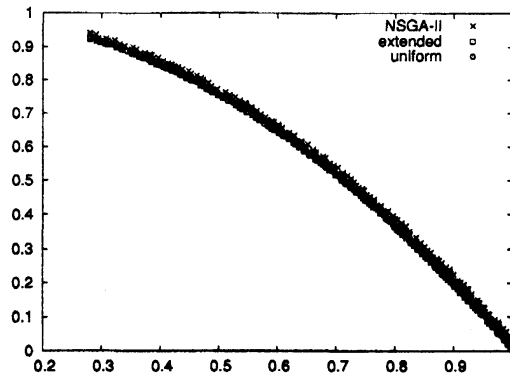


図 11: ZDT6 の非劣解集合

より正確に各アルゴリズムの性能を比較するために、GD、 Δ 尺度、SC 尺度、支配尺度を用いた比較を行った。

表 1 は GD の結果である。ここで、average, std. はそれぞれ 10 回の試行による GD の平均値、標準偏差を表す。平均値については、SCH, ZDT3, ZDT6 では拡張 SBX 版 NSGA-II が、SCH, FON, ZDT1 ZDT2, ZDT4 では改良版 NSGA-II が最良の結果を出している。以上より、改良版 NSGA-II は NSGA-II より明らかに優れている。一方、拡張 SBX 版 NSGA-II と改良版 NSGA-II の差は明瞭ではないが、多くの問題に対して改良版 NSGA-II の方が良い結果を得ている。この結果より、拡張 SBX と等距離選択を導入した改良版 NSGA-II は、真の Pareto 最適解集合への収束度の高い結果を得られる方法であるといえる。

表 2 は Δ 尺度で評価した結果である。ただし、average, std. はそれぞれ 10 回の試行における Δ 尺度の平均値、標準偏差である。平均値については、すべての問題において、改良版 NSGA-II の方が優れており、SCH においては拡張 SBX 版 NSGA-II も同様の結果を得ている。また、ZDT6 を除くすべての問題において、拡張 SBX 版 NSGA-II の方が NSGA-II より優れている。よって、 Δ 尺度の意味において、改良版 NSGA-II は明らかに NSGA-II よりも優れており、拡張 SBX 版 NSGA-II よりも優れている。このように、改良版 NSGA-II は均一にかつ広範囲に広がった解を得られる方法である。

表 1: GD 尺度

Prob.	NSGA-II		拡張 SBX 版		改良版	
	average	std.	average	std.	average	std.
SCH	0.002353	0.002328	0.001797	0.001603	0.001797	0.001603
FON	0.002540	0.000209	0.002599	0.000179	0.002502	0.000323
ZDT1	0.011338	0.001585	0.005810	0.001200	0.005229	0.001031
ZDT2	0.017748	0.003538	0.006566	0.001522	0.006267	0.001461
ZDT3	0.016807	0.007924	0.010752	0.001265	0.014116	0.009460
ZDT4	0.006236	0.003430	0.004804	0.003117	0.004705	0.001986
ZDT6	0.018758	0.001751	0.009320	0.001053	0.009861	0.000525

表 2: Δ 尺度

Prob.	NSGA-II		拡張 SBX 版		改良版	
	average	std.	average	std.	average	std.
SCH	0.762491	0.068801	0.740230	0.108301	0.740230	0.108301
FON	0.409797	0.044612	0.400073	0.040098	0.304353	0.026959
ZDT1	0.367034	0.031527	0.357016	0.048696	0.306647	0.033919
ZDT2	0.482167	0.206755	0.387577	0.029093	0.322953	0.034329
ZDT3	0.427152	0.072924	0.405702	0.041157	0.308901	0.039056
ZDT4	0.616521	0.201929	0.430642	0.066840	0.407773	0.095551
ZDT6	0.367328	0.029725	0.368907	0.060475	0.323761	0.076795

表 3 は、SC 尺度の結果である。ただし、average, std. は、それぞれ 10 回の試行における全ての集合対、すなわち 10×10 の集合対に対する SC 尺度の平均値、標準偏差である。平均値については、FON を除く全ての問題に対して、拡張 SBX 版 NSGA-II が NSGA-II より優れており、改良版 NSGA-II は全ての問題において NSGA-II より優れている。さらに、改良版 NSGA-II は SCH を除く全ての問題において拡張 SBX

版 NSGA-II より優れている (SCH においては, 同じ結果を得ている). よって, SC 尺度によれば, 改良版 NSGA-II は NSGA-II よりも明らかに優れており, 拡張 SBX 版 NSGA-II よりも優れている. 以上の結果より, 改良版 NSGA-II は精度の高い非劣解を得られる方法である.

表 3: SC 尺度

Prob.	NSGA-II × 拡張 SBX 版		拡張 SBX 版 × NSGA-II	
	average	std.	average	std.
SCH	0.013778	0.014698	0.015297	0.015880
FON	0.183300	0.039270	0.176700	0.039422
ZDT1	0.009800	0.014763	0.666700	0.161413
ZDT2	0.001600	0.004841	0.911200	0.096315
ZDT3	0.010800	0.017011	0.651900	0.160173
ZDT4	0.181800	0.251035	0.270870	0.288020
ZDT6	0.000000	0.000000	0.956800	0.041744

表 4: 支配尺度

Prob.	NSGA-II × 拡張 SBX 版		拡張 SBX 版 × NSGA-II		Prob.	NSGA-II × 改良版		改良版 × NSGA-II	
	average	std.	average	std.		average	std.	average	std.
SCH	0.491667	0.395021	0.508333	0.395021	SCH	0.491667	0.395021	0.508333	0.395021
FON	0.512689	0.093236	0.487311	0.093236	FON	0.494713	0.102059	0.505287	0.102059
ZDT1	0.019353	0.036592	0.980647	0.036592	ZDT1	0.009970	0.021178	0.990030	0.021178
ZDT2	0.001464	0.004598	0.998536	0.004598	ZDT2	0.000355	0.002019	0.999645	0.002019
ZDT3	0.019694	0.045585	0.980306	0.045585	ZDT3	0.067454	0.180717	0.932546	0.180717
ZDT4	0.422661	0.460626	0.577339	0.460626	ZDT4	0.412855	0.438684	0.587145	0.438684
ZDT6	0.000000	0.000000	1.000000	0.000000	ZDT6	0.000000	0.000000	1.000000	0.000000

Prob.	NSGA-II × 改良版		改良版 × NSGA-II		Prob.	拡張 SBX 版 × 改良版		改良版 × 拡張 SBX 版	
	average	std.	average	std.		average	std.	average	std.
SCH	0.013778	0.014698	0.015297	0.015880	SCH	0.011129	0.012865	0.011129	0.012865
FON	0.175500	0.044031	0.188500	0.040703	FON	0.172300	0.042892	0.194200	0.040104
ZDT1	0.006300	0.011105	0.741100	0.133813	ZDT1	0.113300	0.110146	0.204500	0.147176
ZDT2	0.000300	0.001706	0.937900	0.084998	ZDT2	0.170300	0.149194	0.219200	0.172393
ZDT3	0.020900	0.046370	0.691500	0.160651	ZDT3	0.143300	0.102236	0.167000	0.121651
ZDT4	0.125400	0.174817	0.264034	0.278930	ZDT4	0.213400	0.250429	0.244300	0.277208
ZDT6	0.000000	0.000000	0.972100	0.031632	ZDT6	0.071400	0.091017	0.133100	0.107932

Prob.	拡張 SBX 版 × 改良版		改良版 × 拡張 SBX 版		Prob.	拡張 SBX 版 × 改良版		改良版 × 拡張 SBX 版	
	average	std.	average	std.		average	std.	average	std.
SCH	0.011129	0.012865	0.011129	0.012865	SCH	0.500000	0.350793	0.500000	0.350793
FON	0.172300	0.042892	0.194200	0.040104	FON	0.474921	0.096102	0.525079	0.096102
ZDT1	0.113300	0.110146	0.204500	0.147176	ZDT1	0.384258	0.317661	0.615742	0.317661
ZDT2	0.170300	0.149194	0.219200	0.172393	ZDT2	0.455178	0.341444	0.544822	0.341444
ZDT3	0.143300	0.102236	0.167000	0.121651	ZDT3	0.506716	0.317638	0.493284	0.317638
ZDT4	0.213400	0.250429	0.244300	0.277208	ZDT4	0.483578	0.450073	0.516422	0.450073
ZDT6	0.071400	0.091017	0.133100	0.107932	ZDT6	0.361970	0.372039	0.638030	0.372039

支配尺度の結果を表 4 に示す. ただし, average, std. は, それぞれ 10 回の試行における全ての集合対, すなわち 10×10 の集合対に対する支配尺度の平均値, 標準偏差である. 平均値については, FON を除く全ての問題において, 拡張 SBX 版 NSGA-II が NSGA-II より優れており, 改良版 NSGA-II は全ての問題において, NSGA-II より優れている. さらに, 改良版 NSGA-II は, ZDT3 と SCH を除くすべての問題において, 拡張 SBX 版 NSGA-II よりも優れており, ZDT3 と SCH においてもほぼ同等の結果を得ている. よって, 改良版 NSGA-II は, 明らかに NSGA-II よりも優れており, 拡張 SBX 版 NSGA-II よりも高い性能を示している. この結果より, 改良版 NSGA-II は, 支配尺度で判断しても, 精度の高い非劣解が得られる方法である.

6 終わりに

多目的最適化問題に対する Pareto 最適集合を求める手法として, 交叉に拡張 SBX, 選択に等距離選択を採用した改良版 NSGA-II を提案した. 拡張 SBX は, 子を生成する範囲を拡張するように定義することによって境界上への子の生成を保証した. 等距離選択では, 等間隔の位置にある解の混雑距離を増やすことによって, 次世代に生き残りやすいようにした.

様々なタイプの 7 種類の多目的最適化問題に改良版 NSGA-II を適用した. また, 拡張 SBX のみを用いた NSGA-II, NSGA-II の結果と比較することによって, 改良版 NSGA-II が, これら 2 つの方法に比べて精度の高い探索性能を有することを示した. これによって, 拡張 SBX による交叉および等距離選択, 特に等距離選択の導入が多目的最適化問題の Pareto 最適集合を求めるために非常に有効であることを示した.

しかし, 拡張 SBX と等距離選択にはいくつかの問題点がある. 拡張 SBX については, 図 2 のように, 親集合に近い領域が取り落とされる場合があり, よりなめらかな拡張を採用する方が望ましい. 等距離選択については, ZDT3 のように Pareto 最適集合が非連結である問題において, 大きな解のない空領域が存在するために等間隔距離が正しく与えられない場合があり (図 9 を参照), 領域全体の長さを計算する際に空領域の長さを考慮する必要がある. 今後は, これらの問題に取り組む予定である.

参考文献

- [1] 西川緯一, 三宮信夫, 茨木俊秀: 岩波講座 情報科学 最適化, 岩波書店 (1982).
- [2] 坂和正敏: 非線形システムの最適化, 森北出版 (1986).
- [3] 北野宏明編: 遺伝的アルゴリズム 2, 産業図書 (1995).
- [4] 阪井節子, 高濱徹行: 多目的非線形最適化手法の一提案 —Vector Simplex 法—, 数理解析研究所講究録, Vol. 1174, pp. 169–178 (2000).
- [5] 高濱徹行, 阪井節子: 多目的非線形最適化手法 Vector Simplex 法による多目的ファジィ制御ルールの対話的学習, 情報処理学会論文誌, Vol. 42, No. 11, pp. 2607–2617 (2001).
- [6] Schaffer, J.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms, in *Proc. of the 1st ICGA*, pp. 93–100 (1985).
- [7] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T.: A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II, *IEEE Trans. on Evolutionary Computation*, Vol. 6, No. 2, pp. 182–197 (2002).
- [8] Goldberg, D.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley (1989).
- [9] Veldhuizen, D. A. V. and Lamont, G. B.: On Measuring Multiobjective Evolutionary Algorithm Performance, in *Proc. of Congress on Evolutionary Computation 2000*, Vol. 1, pp. 204–211 (2000).
- [10] Zitzler, E., Deb, K. and Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, *Evolutionary Computation*, Vol. 8, No. 2, pp. 173–195 (2000).
- [11] Rajagopalan, R., Mohan, C. K., Mehrotra, K. G. and Varshney, P. K.: An Evolutionary Multi-Objective Crowding Algorithm (EMOCA): Benchmark Test Function Results, in *Proc. of the 2nd Indian International Conference on Artificial Intelligence*, pp. 1488–1506 (2005).