

# 多目的最適化問題における探索難易度を考慮した目的関数に対する重み付けの検討

A Study of Weighting on Each Fitness Function based on Difficulty of Search in Multi-objective Optimization Problems.

岸上 利裕      吉川 大弘  
Kishigami Toshihiro      Yoshikawa Tomohiro

名古屋大学大学院工学研究科  
Graduated School of Engineering, Nagoya University

The methods using the concept of weight vector or reference point, for example MOEA/D and NSGA-III, gather attention in multi-objective optimization problems. These methods need to normalize each fitness value in order to deal with the difference of order of magnitude. However, conventional normalizations using the max / min value of obtained individuals may cause unbalanced search because they do not consider difficulty of searching on each fitness function. So we examine the weighting method based on “update” width in this paper. The update width is an index of difficulty of searching on each fitness function. The proposed weighting method emphasis the fitness values that are not searched well. And we confirmed the significance of the proposed weighting method by experiment using original benchmark problems based on DTLZ2.

## 1. はじめに

多目的最適化問題において、MOEA/D[Zhang 07]に代表される、評価値空間におけるベクトルを用いた手法などでは、オーダーの異なる目的関数に対応するため、評価値の正規化を必要とする。特に実問題においては、目的関数間での単位や値のオーダーが異なることが多く、さらにベンチマーク問題とは異なり、評価値の最大値/最小値は未知の場合がほとんどである。

これに対し従来の多目的探索手法では、探索過程で得られた各評価値の最大値/最小値を用いて、各評価値を0~1に正規化する方法などが一般的である[Zhang 07]。しかし、(特に実問題において)探索を行う上では、各目的関数がとり得る値の範囲と、探索により更新され得る値の範囲とが大きく異なる場合がある。さらに、その評価値の更新され易さは、その評価値の最大値付近と最小値付近とで大きく異なることが一般的である。そのような問題を対象とした際、一般的な最大値/最小値による正規化では、例えば、とり得る値の範囲(最大値 - 最小値)は大きい、更新され得る値の幅が小さいとき、更新された評価値の量が過小に評価されるなどの問題が生じる。そしてこれにより、目的関数ごとの探索のされやすさに偏りが生じる。

本稿では、多目的最適化問題における、従来の最大値/最小値に基づいた正規化方法の問題点に注目する。更新幅という指標を定義し、上述の問題点を持つ問題に対して、更新幅を用いた局所的な重み付けを行うことにより、探索の性能が向上することを確認する。

## 2. 従来の評価値正規化方法

### 2.1 評価値正規化方法例

ここでは、ベクトルを用いた代表的な手法の1つである、MOEA/D[Zhang 07]の評価値の正規化方法を例として挙げ

る。本稿における“正規化”とは、以下の式(1)などにより、評価値を0~1の範囲に変換することをいう。

$$\bar{f} = \frac{f - f_{min}}{f_{max} - f_{min}} \quad (1)$$

ここで  $f$  は正規化をする対象の評価値で、 $f_{max}$ ,  $f_{min}$  はそれぞれ値の上限値, 下限値を表す。MOEA/Dでは、式(2)(以降の式は全て最小化問題を前提とする)により評価値の正規化を行う。

$$\bar{f}_i = \frac{f_i - z_i}{z_i^{nad} - z_i} \quad (2)$$

各変数における添え字  $i$  は、対象の目的関数  $i$  を表しており、 $z_i$  は探索で得られた目的関数  $i$  での最小評価値、 $z_i^{nad}$  は現個体群の目的関数  $i$  の最大評価値を指している。この正規化の意味するものは、評価値空間全体の均一化ではなく、現個体群の存在する空間の均一化である。これは、探索の目的が、評価値空間全体の均一な探索ではなく、パレートフロントの均一な探索としていることから来ているものと考えられる。

### 2.2 従来正規化の問題点

ベクトルを用いる探索の場合、全ての目的関数の影響力を均一にする必要がある。そのため、手法ごとに具体的な正規化の方法は異なっている[Zhang 07][Deb 14]が、正規化そのものは必須の前処理となっている。これらの正規化方法では共通して、何かしらの最大値/最小値により上限下限を設定しているものが多い。これは、評価値空間を均一にする際には、元空間の幅(最大値 - 最小値)を用いるのが直接的であり、簡易的な計算で変換が可能であることが理由であると考えられる。実際、先に挙げた論文[Zhang 07][Deb 14]での実験では、目的関数間での偏りなく、均一に探索が行われており、一見特に問題が無いように思われる。しかし、これらの実験で用いられている対象問題は、共にDTLZ[Deb 02b]などのベンチマーク問題である。ベンチマーク問題は、真のパレートフロントが既知であり、評価にかかる計算時間も短いため、手法の性能評価を行うにはとても有用な問題である。ただし、正規化する以前に、最初から目的関数ごとにオーダーが統一されているもの

連絡先: 岸上 利裕, 名古屋大学大学院工学研究科, 052-789-2793, kishigami@cmplx.cse.nagoya-u.ac.jp  
吉川 大弘, 名古屋大学大学院工学研究科, yoshikawa@cse.nagoya-u.ac.jp

も多く、また、目的関数ごとに特徴の差の無いものも少なくない。そのような問題においては、そもそも正規化といった前処理の必要性が少なく、正規化方法に対して深く検討する必要がない。

しかし、実問題においては、目的関数ごとにオーダーが大きく異なり、また目的関数間の依存関係も様々であることで、目的関数によって評価値の更新のされ易さが大きく異なる場合が多い。筆者らは、そのような問題に対して、評価値の最大値/最小値のみを用いて正規化を行ったとしても、適切に探索が行われず、目的関数によって偏ったパレート解が得られるなどの問題が生じることを確認している [岸上 15]。これは、前述の通り、従来の正規化では評価値空間を均一に扱うことが目的であり、目的関数ごとの評価値の更新量や更新のし易さの違いを考慮に入れていないことが原因であると考えられる。そのため、目的関数間での評価値空間の広さと、探索による更新の幅の比に格差がある場合、更新の量を適切に捕らえることができず、偏った探索を行うこととなる。

### 3. 重み付けによる探索の偏りの軽減

2.2 で述べたように、従来の評価値の最大値/最小値のみを用いた正規化を行った場合、目的関数間での評価値空間の広さと、探索による更新の幅の比に格差がある問題においては、偏った探索となってしまう。そこで本稿では、評価値の“更新幅”を定義し、重み付けという形でこの更新幅を用いることで、探索されにくい目的関数に対しても、適切に探索が行われるようにする。

#### 3.1 更新幅

従来の最大値/最小値を用いた正規化の問題点は、目的関数ごとにどれほど探索が行われ易いかを考慮に入れていないことである。そこで、目的関数ごとの更新のされ易さ、すなわち更新量を正規化の計算に組み込むため、“更新幅 (UW)” を以下 (式 (3)、図 1) のように定義する。

$$UW_i = \frac{z_i^{max} - z_i^{min}}{\bar{z}_i^{nad} - z_i} \quad (3)$$

$$\bar{z}_i^{max} = \max\{PS_{j,i}^{max} : j = 0, \dots, J_{now}\}$$

$$\bar{z}_i^{min} = \min\{PS_{j,i}^{max} : j = 0, \dots, J_{now}\}$$

$$PS_{j,i}^{max} = \max\{PS_{j,n,i} : n = 0, \dots, N\}$$

ここで、 $PS_{j,n,i}$  は  $j$  世代目の  $n$  番目のパレート解 (個数  $N$ ) の目的関数  $i$  の値を示すものとする。多くの目的関数は他の目的関数とトレードオフの関係にあり、1つの目的関数に注目すると、その目的関数が最良値を取っている解においては、他目的関数は悪い値となっており、最悪値を取っている解については他の目的関数は良い値となっているのが一般的である。つまり、評価値の最大値/最小値による幅は、他の目的関数の評価値が悪化することを許容した上での (更新可能) 幅である。しかし、パレート解の収束性を高めるためには、極力他の目的関数を悪化させることなく、目的とする目的関数の評価値を更新 (改良) する必要がある。式 (3) により定義した更新幅は、他の目的関数を悪化させることによる更新ではなく、他の目的関数を悪化させずに更新した幅 (図 1) を指標化している。つまり、この値が小さい目的関数ほど、探索が難しいことを表し、強調すべき目的関数であると考えられる。

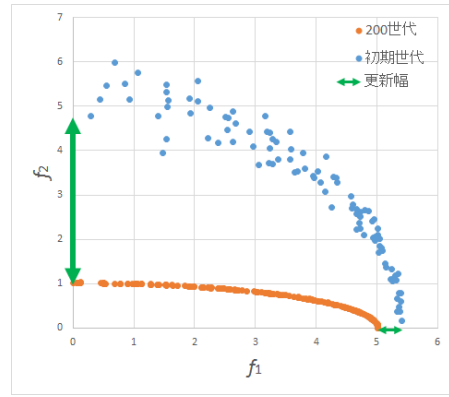


図 1: 更新幅

#### 3.2 更新幅による局所的重み付け

ここでは、3.1 で定義された更新幅を探索に反映させる方法を考える。3.1 で示した通り、更新幅は各目的関数の最悪値付近での更新の量を示した局所的な指標であるのに対して、式 (1) や式 (2) などの最大値/最小値を用いた正規化は、空間全体を伸縮させるものであるため、そのまま組み合わせることは困難である。また局所的な情報による更新幅のみでは、空間全体に適用することは適切ではない。そこで本稿では、目的関数ごとの重み付けを局所的に行うことにより、更新幅の概念を探索に取り入れることを考える。本稿で提案する重み付け方法のアルゴリズムの要点は 2 つである。1 つ目は、更新した量に対する重み付けであり、2 つ目は、局所的な重み付けを行うことである。以下その 2 点に関して説明する。

##### 3.2.1 更新量への重み付け

3.1 で述べたように、更新幅が小さいほどその目的関数は更新されにくいことを示しており、更新量を強調して評価をすべきである。そのため、更新幅が小さい目的関数の更新の量を強調するような重み付けの方法が求められる。更新量に対する重み付けのアルゴリズムを以下に示す (イメージ図: 図 2)。

1. 更新の量を算出するため、基準となる個体を決める (以後、基準となる個体を  $I_b$ 、他の個体を  $I_t$  とする)。
2.  $I_b$  は式 (2) で正規化を行い、 $I_t$  は以下の式 (4) で正規化、重み付けをする。

$$\bar{f}_i^t = \left( \frac{f_i^t - z_i}{\bar{z}_i^{nad} - z_i} - \bar{f}_i^b \right) * \frac{UW_{max}}{UW_i} + \bar{f}_i^b \quad (4)$$

$\bar{f}_i^b$  は式 (2) による正規化後の  $I_b$  の評価値であり、 $f_i^t$  は  $I_t$  の元の評価値を表している。また、 $UW_{max}$  は  $\max\{UW_i : i = 1, \dots, M\}$  であり、 $M$  は目的関数の数である。その他は式 (2) と同じである。

この重み付けは、 $I_t$  の  $I_b$  との評価値の差分に対して、各目的関数の更新量に対する重みを考慮した仮想的な評価値で評価することを表しており、探索の困難な目的関数に対しても、その更新量が大きく評価されることで、他の目的関数と同等の探索し易さとなることが期待できる。

##### 3.2.2 局所的な重み付け

重み付けのアルゴリズムは 3.2.1 で述べたが、更新幅はあくまで他の目的関数を悪化させずに更新できる幅を表しているた

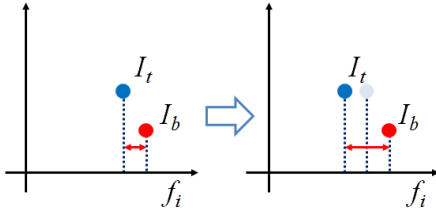


図 2: 重み付けイメージ図

め、評価値空間全体に適用できる指標ではない（重みの付いた目的関数に対して、他の目的関数を悪化させることで良い値となったものが過剰に強調される）。そこで、他の目的関数を悪化させていない領域においては更新幅に基づく重み（式 (4) の  $\frac{UW_{max}}{UW_i}$ ）が強くなり、他の目的関数を悪化させている領域についてはその重みを弱くし、式 (2) による空間的な正規化を行うといった、局所的な重み付けを行う。その際に用いる係数  $\alpha_i$  を式 (5) に示す。

$$\alpha_i = \max(\bar{f}_{n,k} : n = 1, \dots, N; k \neq i, k = 1, \dots, M) \quad (5)$$

$\bar{f}_{n,k}$  は、 $n$  番目の個体の目的関数  $k$  の評価値を表す。この  $\alpha_i$  が大きいほど、目的とする目的関数  $i$  以外の目的関数の値が悪い（最小化問題を前提とした場合）、すなわち他の目的関数を悪化させている領域であるとし、 $\alpha_i$  が大きいほど重みを小さくする（式 (6)）。

$$W_i^* = \frac{UW_{max}}{UW_i} * (1 - \alpha_i)^n \quad (6)$$

$n$  は重みの影響範囲の調整係数であり、 $1 - \alpha_i \leq 1$  であることから、 $n$  が大きいほど式 (6) の右辺は小さくなり、更新幅による重みの影響領域が狭くなることを表している。

3.2.1 で述べた更新幅正規化（式 (4)）と、ここで述べた局所的な重み付け（式 (6)）を組み合わせることにより（式 (7)）、最大値/最小値の正規化における目的関数の偏りを軽減させた探索が期待される。

$$\bar{f}_i^t = \left( \frac{f_i^t - z_i}{z_i^{nad} - z_i} - \bar{f}_i^b \right) * W_i^* + \bar{f}_i^b \quad (7)$$

## 4. 実験

3 章で述べた更新幅による局所的な重み付け（式 (7)）を MOEA/D に適用し、この重み付け方法の効果を確認する。

### 4.1 実験方法

MOEA/D[Zhang 07] では、各重みベクトルに 1 個体が属し、ベクトルごとに個体を更新していくことにより探索を進めていく。よって MOEA/D では、各ベクトルに属する個体を、生成された子個体により更新するかしないかという 2 個体間比較しか行われない。そこで、ベクトルに属する個体を  $I_b$ 、新たに生成された子個体を  $I_t$  とすることにより、式 (7) を適用する（ $I_b$ ,  $I_t$  は 3.2.1 を参照）。

### 4.2 ベンチマーク関数実験

従来の正規化方法による MOEA/D と、3.2 で示した更新幅による局所的な重み付けを用いた MOEA/D を、ベンチマ

ーク関数問題に適用して性能の比較を行う。ここで用いるベンチマーク関数は、DTLZ2 に対し、目的関数ごとに探索の難易度を変更し、収束性にかかわる設計変数を目的関数ごとに分割したものである。

$$\left. \begin{aligned} & \text{Minimize } f_1(x) = (5 + g(\mathbf{x}_{M1}/10))\cos(x_1\pi/2) \\ & \text{Minimize } f_2(x) = (1 + g(\mathbf{x}_{M2}))\sin(x_1\pi/2) \\ & 0 \leq x_i \leq 1, \text{ for } i = 1, 2, \dots, n, n = 2n_c + 1 \\ & \text{where } g(\mathbf{x}_{M1}) = \sum_{i=2}^{n_c+1} (x_i - 0.5)^2 \\ & \text{where } g(\mathbf{x}_{M2}) = \sum_{i=n_c+2}^n (x_i - 0.5)^2 \end{aligned} \right\} \quad (8)$$

$$\left. \begin{aligned} & \text{Minimize } f_1(x) = (5 + g(\mathbf{x}_{M1}/100))\cos(x_1\pi/2) \\ & \text{Minimize } f_2(x) = (1 + g(\mathbf{x}_{M2}))\sin(x_1\pi/2) \\ & 0 \leq x_i \leq 1, \text{ for } i = 1, 2, \dots, n, n = 2n_c + 1 \\ & \text{where } g(\mathbf{x}_{M1}) = \sum_{i=2}^{n_c+1} (x_i - 0.5)^2 \\ & \text{where } g(\mathbf{x}_{M2}) = \sum_{i=n_c+2}^n (x_i - 0.5)^2 \end{aligned} \right\} \quad (9)$$

$$\left. \begin{aligned} & \text{Minimize } f_1(x) = (5 + g(\mathbf{x}_{M1}/100))\cos(x_1\pi/2) \\ & \text{Minimize } f_2(x) = (1 + g(\mathbf{x}_{M2}))\sin(x_1\pi/2) \\ & 0 \leq x_i \leq 1, \text{ for } i = 1, 2, \dots, n, n = 2n_c + 1 \\ & \text{where } g(\mathbf{x}_{M1}) = \\ & |n_c| + \sum_{i=2}^{n_c+1} \{(x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\} \\ & \text{where } g(\mathbf{x}_{M2}) = \sum_{i=n_c+2}^n (x_i - 0.5)^2 \end{aligned} \right\} \quad (10)$$

式 (8) は、オーダーを揃えつつ、 $f_1$  の更新の量を 10 分の 1 にしたものである。目的関数間において、値そのもののオーダーや、評価値の取り得る値の幅はほぼ変わらないが、更新の量に大きな差があることにより、目的関数間の更新のし易さの偏りを強調したものである。また式 (9) は、式 (8) に対して、目的関数間の偏りをより顕著にしたもの（10 分の 1 → 100 分の 1）、式 (10) は  $f_1$  について多峰性にしたものである。これらのベンチマーク問題に対して、2 目的、分割数 100、近傍個体数 10、評価回数 10,000 とし MOEA/D を適用する。また、スカラー化関数には PBI[Zhang 07] を用いる。評価指標には、最終個体の  $g(\mathbf{x}_{M1})$ ,  $g(\mathbf{x}_{M2})$ （式 (8)）の平均の 10 試行平均値を用いる。それぞれ値が 0 に近いほど、 $f_1$ ,  $f_2$  に対する収束性が高いことを示す。

### 4.3 実験結果

実験の結果を表 1 に示す。表 1 より、 $g(\mathbf{x}_{M1})$ ,  $g(\mathbf{x}_{M2})$  の間に大きく差があり、すべての関数や条件において  $f_2$  に対して  $f_1$  の探索が極端に進んでいないことがわかる。また重み有り（式 (7) による重み付け）と重み無し（式 (2) による正規化）を比較すると、更新幅による局所的な重み付けを行ったものが、ベンチマーク関数 1, 2 においては、 $g(\mathbf{x}_{M1})$  の値が小さくなっていることがわかる。この結果から、重み付けを行うことにより、探索の難しい  $f_1$  の探索が改善されたことが確認できる。

一方ベンチマーク関数 3 においては、重みを付けることによって、探索の精度が落ちてしまっていることが確認できる。今回用いた重み付け方法では、各目的関数について、更新さ



表 1: 実験結果

		$g(\mathbf{x}_{M1})$	$g(\mathbf{x}_{M2})$
ベンチマーク関数 1	重み無し (式 (2))	1.120	0.105
	式 (8)	0.896	0.048
ベンチマーク関数 2	重み無し (式 (2))	2.875	0.089
	式 (9)	2.547	0.020
ベンチマーク関数 3	重み無し (式 (2))	14.01	0.174
	式 (10)	14.67	0.178

れた量を更新幅に基づく重みにより強調するというものであった。それにより、評価値が悪化することについても敏感になってしまい、局所解にはまりやすくなり、多峰性の目的関数においては十分な性能が発揮できなかった可能性がある。しかし、実問題を考えると、多くの問題で多峰性の目的関数を持つと考えられる。そのため、多峰性の問題に対しても本重み付けが有効に働くよう、何らかの形で局所解から脱却する機構を、工夫として取り入れる必要があると考えられる。

また、 $g(\mathbf{x}_{M2})$  について、重み無しのものよりも重み有りの方が低い値となっていることがわかる。本稿で用いた重み付けでは、探索のし易い  $f_2$  を多少悪化させてでも、更新幅の小さい  $f_1$  を強調して探索するものであるため、 $g(\mathbf{x}_{M2})$  については、悪化することはあっても改善されることは考えにくい。そこでなぜこのような結果が得られたのかについて、次節で考察を行う。

#### 4.4 結果考察

4.3 の結果より、重み付けを行うことにより、更新幅の小さい目的関数でも探索が進むことが確認できた。しかし同時に、目的としていない目的関数  $f_2$  についても値の改善が見られた。その理由について考察する。重みを付けることによって得られた結果であるため、重みが強くかかる  $x_1$  が小さい個体に注目する。 $x_1$  が小さい個体に重みが強くかかる理由は、これらの個体が存在する領域は式 (6) の  $\alpha_i$  の値が小さく、結果  $W_i^*$  が大きい領域であるためである。今回の実験では、 $f_2$  の更新幅の方が大きいことから  $W_2^* = 1$  となるため、 $f_1$  方向にのみ強い重みがかかる。そのため、図 3 に示すような重み付けにより、個体が評価されることとなる。

PBI の計算式は  $\text{minimize}(d_1 + \theta d_2)$  であるため、参照点に近く、かつ重みベクトルに近い個体が優先される。ここで、2 つの個体が図 3 の左図のような位置関係にあった場合、重みがかかることにより、 $d_1, d_2$  共に改善されているが、右図においては、 $d_1$  は改善されているものの、 $d_2$  は悪化していることがわかる。これは、重みベクトルの傾きにより、ベクトルの下側、すなわち  $f_2$  の値の低い領域にある個体の方が、重みの影響を良い方向に受けやすいことを示している。このことから、上述の予想外の結果は、更新幅の小さい  $f_1$  の更新量を強調することにより、副産物的に  $f_2$  の小さい個体が優先された結果であると考えられる。

#### 5. おわりに

本稿では、多目的最適化問題における、従来の最大値/最小値に基づいた正規化方法の問題点に注目した。更新幅という指標を定義し、その問題点を持つ問題に対して、更新幅を用いた局所的な重み付け方法を提案した。

提案した、更新幅による局所的な重み付け方法を MOEA/D

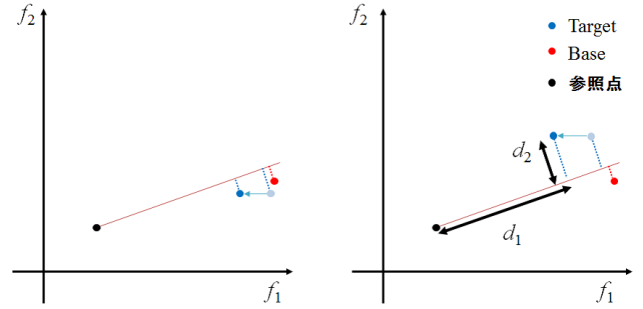


図 3: 重み付けの影響

に適用し、従来のベンチマーク関数を、実問題を想定して修正を行った関数に対して適用することにより、その有用性の検証を行った。その結果、更新幅と局所性を組み合わせた更新幅による局所的な重み付けにより、従来の最大値/最小値を用いた正規化では探索が進まなかった目的関数において、探索の精度が大きく向上することを確認した。しかし、多峰性を持たせた関数においては、性能が悪化した。また、今回の実験では、目的としていない目的関数についても同時に精度が向上したため、その原因について考察を行った。

今後は、局所解の脱却の機構の検討と、本重み付け方法の他の手法 [岸上 15] への適用、実問題での検討を行っていく予定である。

#### 6. 謝辞

本研究は、文部科学省科学研究費（基盤研究 (C)、No.15K00336）の補助を得て遂行された。

#### 参考文献

- [Deb 02a] Deb, K.: *A Fast and Elitist Multiobjective Genetic Algorithm : NSGA-II* (2002)
- [Deb 02b] Deb, K., Thiele, L., Laumanns, M., and Zitzler, E.: Scalable multi-objective optimization test problems, *Proceedings of the Congress on Evolutionary Computation (CEC-2002)*, pp. 825–830 (2002)
- [Deb 14] Deb, K. and Jain, H.: An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints, *IEEE Transactions on Evolutionary Computation*, Vol. 18, No. 4, pp. 577–601 (2014)
- [Zhang 07] Zhang, Q. and Li, H.: MOEA/D: A multi-objective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation*, Vol. 11, No. 6, pp. 712–731 (2007)
- [岸上 15] 岸上利裕, 吉川大弘: 多目的問題における評価値の正規化に対する問題点の検討, 第 9 回進化計算学会研究会, pp. 73–75 (2015)