B-101

多数目的最適化問題に対する差分進化の適用
Differential Evolution for Many-Hard-Objective Optimization

今村 晃啓† 田川 聖治‡
Akihiro Imamura Kiyoharu Tagawa

1．はじめに

本稿では、多くの目的関数がゴールポイントによって制限された多数目的最適問題である Many-Hard-objective Optimization (MHOP) について述べる。また、MHOP の多様なパレート実行可能解を探索するために、Differential Evolution (DE) [1]に基づく、新たな最適化アルゴリズム Differential Evolution for Many-Hard-objective Optimization (DEMHO) を提案する。DEMHO は非劣解集合の評価を Pairwise Exclusive Hypervolume (PEH) [2]とその高速計算アルゴリズムにより行う。また、MHOP の実行不可能解に対しては二段階選択を適用する。最後に、計算実験と統計的検定により DEMHO の有効性を確認する。

2．多数目的最適化問題

多目的最適化問題は「複数個の互いに競合する目的関数を与えられた制約の中で何らかの意味で最小化（最大化）する問題」と定義される。特に、多目的最適化問題において目的の数が４以上の場合、多数目的最適化問題と呼ばれる。多目的最適化問題では、目的関数同士が互いに競合し合って、すべての目的関数の値を最小にするという意味で最良な解を求めることができない。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。この様な解をパレート最適解 (Pareto-optimal solution) と呼ぶ。

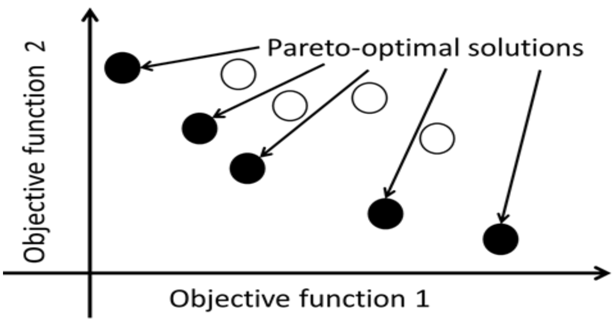

図１　パレート最適解

図１に２目的の場合の非劣解の概念図を示す。丸印が目的関数空間における解を表し、黒丸が非劣解である。非劣解は良いとこ[3]をもち互いに勝負がつかない解であるため、多数目的最適化問題では数多く存在する。また、解空間全体における非劣解がパレート最適解である。

本論文では、目的の数が４以上の多数目的最適化問題の中の Many-Hard-objective Optimization Problem (MHOP) を扱う。MHOP はすべての目的関数 $f_m(\vec{x})$ が上限値 $g_m \in \Re$ (m＝1, …, M) によって制約されている問題[2]である。MHOP は以下の記述することができる。

$$\left[\begin{array}{ll} \text{Minimize} & \vec{f}(\vec{x}) = (f_1(\vec{x}), \cdots, f_M(\vec{x})) \\ \text{Subject to} & \forall m \in \{1, \cdots, M\}\ f_m(\vec{x}) \le g_m \end{array}\right. \tag{1}$$

3. PEH

多数目的最適化問題の分野において、Hypervolume は非劣解集合の評価指標として広く用いられている。また、Exclusive Hypervolume (EH) [3]とは、図２に示すようにある参照点 (Reference point) の Hypervolume に対する個々の非劣解の貢献度を表すものであり、２つの Hypervolume の差として定義される。EH は多数目的最適化のアルゴリズムにおける非劣解の有効な評価指標であると考えられ、その高速計算アルゴリズムが研究されていが、EH の計算量は指数時間となる。そこで図３に示す EH の近似値である Pairwise Exclusive Hypervolume (PEH) [2]を使用する。PEH は多項式時間で計算できる。

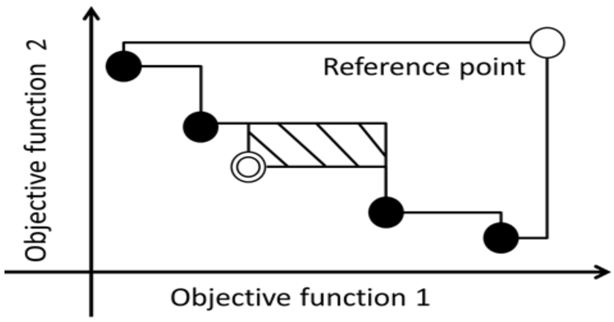

図２　Exclusive Hypervolume (EH)

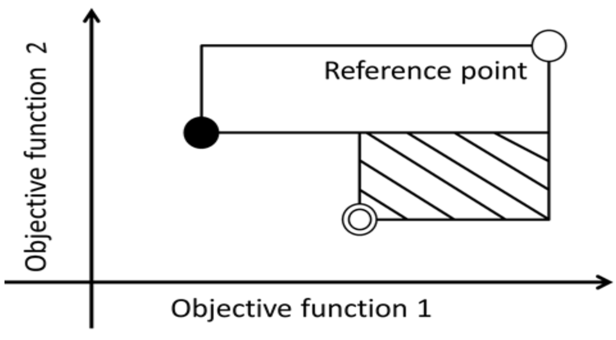

図３　Pairwise Exclusive Hypervolume (PEH)

†近畿大学総合理工学研究科，Graduate School of Science and Engineering Research，Kinki University

‡近畿大学理工学部，School of Science and Engineering，Kinki University

４．DEMHO のアルゴリズム

4.1 DEMHO

DEMHO は DEMO[4]を拡張したものであり、（１）式の解 $\vec{x}_i$（ｉ＝１，…，N_p）の解集団 P をもっている。集団 P から新しい個体をつくるために、DEMHO は DE の戦略から最も基本的な「DE/rand/1/exp」[1]を採用する。

この戦略では、集団 P からターゲットベクトルと呼ばれる個体 $\vec{x}_i$ を順番に指定する。次に、集団 P から $\vec{x}_i$ とは別の３つの異なる個体 $\vec{x}_{i1}$, $\vec{x}_{i2}$, $\vec{x}_{i3}$ をランダムに選択し、以下のように変異ベクトル $\vec{v}$ を生成する。

$$\bar{v} = \bar{x}_{i1} + S_F(\bar{x}_{i2} - \bar{x}_{i3}) \tag{2}$$

ただし、スケール係数 S_Fは制御パラメータである。

次に、ターゲットベクトル $\vec{x}_i$ と変異ベクトル $\bar{v}$ との指数交叉により、新たな個体の候補であるトライアルベクトル $\bar{u}$ を生成する。指数交叉では交叉率 C_R に基づき $\vec{x}_i$ と $\bar{v}$ から要素を確率的に選択して $\bar{u}$ の要素とする。

最大世代数を N_G とし、現在の世代数をｔとして、提案する DEMHO の手順を以下に示す。

[DEMHO]

手順 1：初期集団 P として $\bar{x}_i \in \chi$ を N_P 個生成する。すべての $\bar{x}_i \in P$ に対して目的関数ベクトル $\vec{f}(\vec{x})$ の値を計算する。ｔ＝０と初期化する。

手順 2：ｔ＝N_G になれば、得られた実行可能な非劣解集合を出力して終了する。

手順 3：ターゲットベクトル $\bar{x}_i \in P$ に対して、手順 3.1 から 3.3 を実行する。

手順 3.1：トライアルベクトル $\bar{u} \in \chi$ を生成する。

手順 3.2：トライアルベクトル $\bar{u}$ に対して目的関数ベクトル $\bar{f}(\bar{u})$ を計算する。

手順 3.3：$\bar{u}$ が $\vec{x}_i$ を優越していたら $\bar{u}$ を $\vec{x}_i$ に代入する。$\vec{x}_i$ が $\bar{u}$ を優越していたら $\bar{u}$ は捨てる。その他の場合は $\bar{u}$ をそのまま $\vec{x}_i$ を含む集団 P に追加する。

手順 4：集団 P のサイズが N_P を超えたら、集団 P に対して選択操作を適用して良い N_P 個の解を選ぶ。

手順 5：ｔ＝ｔ＋１にして、手順２に戻る。

以上のような手順が DEMHO の流れである。手順４では N_P を超える集団サイズになるとき選択操作を行う。

4.2 従来の選択操作

上記で述べた DEMHO の手順４の伝統的な２つの選択操作について紹介する。１つ目の選択操作は実行可能解と実行不可能解の区別をしないものである。個体を優越関係[6]によってランク付けして、ランクの小さい個体から選択する。同一ランクの個体の優劣は PEH により決定する。MHOP では各目的関数を最小化することにより、その目的関数に対する制約条件を満たすことができる。このため、実行可能解と実行不可能解を区別しなくても、目的関数を最小化することで実行可能解が得られることが期待できる。選択操作１の手順を以下に示す。

[選択操作１]

手順 1：集団内の個体を優越関係により分類をすることで、ランク付けを行う。

手順 2：ランクの昇順に個体を選択する。

手順 3：幾つかの個体を同じランクの集合から選択する必要があるとき、同一ランクの集合から個体を 3.1 から 3.3 の手順によって選択する。

手順 3.1：同一ランクの個体集合の参照点を求める。

手順 3.2：各個体について PEH を計算する。

手順 3.3：同一ランクの個体を PEH の降順に選択する。

２つ目の選択操作では、実行可能解と実行不可能解を区別する。実行可能解は実行不可能解よりも優先して選択され、実行不可能解は 違反量に基づいて評価される。選択操作２は、従来の制約条件付き多目的最適化問題に対する進化的アルゴリズムにおいて広く使用されている手法である。選択操作２の手順を以下に示す。

[選択操作２]

手順 1：実行可能解の集合を求める。

手順 2：集団内の実行可能解が N_P より多いとき、選択操作１を使い良い N_P 個の実行可能解を選ぶ。

手順 3：集団内の実行可能解が N_P より少ないとき、3.1 から 3.3 の手順で良い N_P 個の解を選択する。

手順 3.1：全ての実行可能解を選択する。

手順 3.2：実行不可能解に対して違反量 $n(\bar{f})$ を計算する。

$$n(\bar{f}) = \max(f_m - g_m) \tag{3}$$

手順 3.3：実行可能解を違反量の降順に選択する。

4.3 提案する選択操作

提案する選択操作３でも、実行可能解は実行不可能解に優先して選択する。また、実行不可能解は２つの方法で選択する。実行可能解の数が N_s（$N_P > N_S \geq 0$）より大きいとき、実行不可能解の選択には選択操作２を使用する。そうでない場合、実行不可能解の選択には選択操作１を使用し、実行可能解と同様に、ランクと PEH に基づき実行不可能解を評価して選択する。選択操作３の手順を以下に示す。

[選択操作３]

手順 1：実行可能解の集合を求める。

手順 2：集団の実行可能解の数が N_s 以上のとき、選択操作２を使い良い N_P 個の解を選択する。

手順 3：集団の実行可能解が N_s より少ないとき、3.1 から 3.4 の手順で良い N_P 個の解を選択する。

手順 3.1：全ての実行可能解を選択する。

手順 3.2：実行不可能解をランク付けする。

手順 3.3：ランクの昇順に実行不可能解を選択する。

手順 3.4：同一ランクの実行不可能解の集合から PEH の降順に解を選択する。

上記のように提案する選択操作３は、従来の選択操作を２段階に分けることによって、実行可能解の最適性と多様性を向上させることを狙った手法である。

5. PEH の評価

5.1 テスト問題

テスト問題は DTLZ1、DTLZ2、DTLZ3、DTLZ4[5]を使用する。また、解の質の評価に Convergence Measure(CM)を使用する。CM は各解とパレート最適解との距離である。上記のテスト問題に対する CM は以下のようになる。

・DTLZ1

$$CM(\bar{f}) = |\sum_{m=1}^{M} f_m - 0.5| \quad (4)$$

・DTLZ2, DTLZ3, DTLZ4

$$CM(\bar{f}) = |\sum_{m=1}^{M} f_m^2 - 1| \quad (5)$$

PEH の適正を評価のために Crowding-Distance (CD) [6]やε-DOM[7]と比較する。CD は各非劣解の良さを隣接する解とのマンハッタン距離の総和を使って評価する。ε-DOM は各非劣解の質を他の解に優越する程度によって評価する。上記の３つの評価指数を使って、目的数 M のテスト問題に DEMHO を 50 回ずつ適用する。DEMHO プログラムは Java 言語で実装し、数値実験には (CPU: Intel® Core ^{TM} i7@3.33[GHz]; memory: 2[GB]; OS: Microsoft Windows XP) 市販のパソコンを使用する。

5.2 結果と考察

DEMHO の実行時間を表１に示す。表１からわかるように、CD が最も実行時間が短いが、どの評価指標とも大きな差は見られないので、CD が一番良いとは判断はできない。表２に DEMHO で得られた各テスト問題の解に対する CM を示す。表２から、どのテスト問題でも良い結果が得られているのは PEH である。PEH と各評価指数とのウィルコクソンの順位検定で CM を比較した結果を表３と表４に示す。△（▽）は危険率 0.01 で PEH が良い（悪い）、▲（▼）は危険率 0.05PEH で良い（悪い）を意味する。有意な差がないときは“—”で表している。表３から、明らかに PEH は CD よりも良く、表４から、PEH はε-DOM よりも良いことがわかる。以上の結果より、評価指数 PEH は優れていることがわかる。

表１　DEMHO の計算時間

criteria	M	DTLZ1	DTLZ2	DTLZ3	DTLZ4
CD	4	404.3	407.1	2170.9	451.8
	6	536.8	523.1	2761.5	574.7
	8	666.9	645.3	3461.0	665.9
ε-DOM	4	575.6	598.1	3109.0	625.9
	6	858.8	888.1	4888.3	899.1
	8	1120.3	1173.1	6550.6	1111.6
PEH	4	538.1	557.8	2920.2	586.8
	6	748.5	750.0	4269.7	774.6
	8	944.0	954.0	5401.2	924.7

表２　Convergence Measure (CM)

criteria	M	DTLZ1	DTLZ2	DTLZ3	DTLZ4
CD	4	0.3625	1.30E-07	0.3559	1.38E-07
	6	293.33	0.3448	1.16E+05	0.2564
	8	395.13	0.9373	609000.0	0.6372
ε-DOM	4	1.4020	1.67E-08	0.7194	1.05E-08
	6	1.6572	3.55E-07	4.8763	1.69E-08
	8	2.6893	1.41E-06	7.0184	8.36E-08
PEH	4	1.0982	5.56E-09	0.1962	2.16E-09
	6	2.3477	4.E-08	3.3888	6.21E-09
	8	2.6445	1.E-07	6.6024	3.46E-08

表３　Convergence Measure (CM) の統計的検定

criteria	M	DTLZ1	DTLZ2	DTLZ3	DTLZ4
CD	4	—	△	△	△
	6	△	△	△	△
	8	△	△	△	△
ε-DOM	4	▲	△	△	△
	6	—	△	—	△
	8	—	△	—	—

表４　Hypervolume の統計的検定

criteria	M	DTLZ1	DTLZ2	DTLZ3	DTLZ4
ε-DOM	4	△	△	△	△
	6	△	△	△	△
	8	△	△	△	△

6. DEMHO の評価

３つの異なるゴールポイントを設定したテスト問題 DTLZ1、DTLZ2、DTLZ3、DTLZ4 に対して PEH を使用した DEMHO を 50 回ずつ適用した。ここで、提案した選択操作３と従来法の選択操作１と選択操作２を比較する。目的数は M=8 に固定して、選択操作３に対しては $N_s = 0.2 \times N_P$ で数値実験を行った。実験の結果を表５から表８に示す。得られた解の個数を表５に示す。表５からわかるように、実行可能領域が狭くなると選択操作１は良い結果が得られない。また、選択操作２のでは DTLZ1、DTLZ3 に対して良い結果が得られない。このことから、選択操作３が最も優れている。表６に各テスト問題の解に対する CM を示す。表７に選択操作３と他の選択操作を CM についてウィルコクソンの順位検定で比較した結果を示す。表６から選択操作１は DTLZ1 に対して優れていることがわかるが、表５から得られた解の個数が少ない。表７からも良い結果が確認できなかった。表８に選択操作３と他の選択操作を Hypervolume についてウィルコクソンの順位検定で比較した結果を示す。表８から実行可能領域が狭くなると選択操作３は選択操作１よりも優れていることがわかる。DTLZ1、DTLZ3 についても選択操作３は選択操作２よりも良い結果が得られていることがわかる。以上の結果より、選択操作３は従来の選択操作より優れていることがわかる。

表５　得られた解の数

selection	Gm	DTLZ1	DTLZ2	DTLZ3	DTLZ4
method1	0.6	32.1	8.7	8.1	4.5
	0.8	44.3	67.9	58.1	62.5
	1.5	72.3	100.0	120.7	100.0
method 2	0.6	6.0	100.0	12.0	100.0
	0.8	18.0	100.0	32.0	100.0
	1.5	56.0	100.0	88.0	100.0
method 3	0.6	72.0	100.0	84.0	100.0
	0.8	88.0	84.2	88.0	100.0
	1.5	92.0	100.0	176.0	89.5

表６　Convergence Measure(CM)

selection	Gm	DTLZ1	DTLZ2	DTLZ3	DTLZ4
method1	0.6	0.9692	1.36E-07	0.0059	2.00E-08
	0.8	1.3942	1.26E-07	0.0678	2.95E-08
	1.5	1.7356	1.29.E-07	1.8853	3.46.E-08
method 2	0.6	2.8905	2.52.E-08	0.3481	1.37.E-08
	0.8	3.9374	1.23.E-07	2.1117	1.94.E-08
	1.5	5.9576	1.45.E-07	6.5875	2.79.E-08
method 3	0.6	1.2762	2.60.E-08	0.0250	1.87.E-08
	0.8	1.6745	1.60.E-08	0.0677	2.28.E-08
	1.5	2.0206	1.44.E-07	2.3259	3.21E-08

表７　Convergence Measure(CM)の統計的検定

selection	Gm	DTLZ1	DTLZ2	DTLZ3	DTLZ4
method1	0.6	―	△	△	△
	0.8	―	△	△	△
	1.5	―	―	―	―
method 2	0.6	▲	―	―	―
	0.8	△	―	△	―
	1.5	△	―	△	―

表８　Hypervolume の統計的検定

selection	gm	DTLZ1	DTLZ2	DTLZ3	DTLZ4
method1	0.6	―	△	▲	―
	0.8	―	―	―	▲
	1.5	―	―	―	―
method 2	0.6	▲	▼	―	△
	0.8	△	―	△	―
	1.5	△	―	△	―

7. おわりに

MHOP の多様なパレート実行可能解を入手するために、新しいアルゴリズム DEMHO を提案した。DEMHO は PEH とその高速計算アルゴリズムを使用し非劣解を評価した。また、MHOP の実行不可能解は２段階の選択操作を使って選択した。テスト問題を使った数値実験により、PEH を CD や ε-DOM と比較し、PEH が他の２つに勝ることを示した。また、新しく提案した選択操作についても、従来の選択操作と比較し、その有用性を確認した。

今後の課題は、現実的な MHOP に対して DEMHO を適用することである。

参考文献

[1] R.storn and K.Price : Differential evolution – a simple and efficient heuristic for global optimization over continuous space, Journal of Global Optimization, Vol. 11, No. 4, pp. 341-359 (1997)

[2] K. Tagawa, Y. Sasaki, and H. Nakamura : Indicator-based differential evolution using exclusive hypervolume approximation and parallelization for multi-core processors, In Proceedings of Genetic and Evolutionary Computation Conference (GECCO2011), pp. 657–664 (2011)

[3] L. Bradstreet, L. While, and L. Barone : A fast incremental hypervolume algorithm, IEEE Trans. On Evolutionary Computation, Vol. 12, No. 6, pp. 714–723 (2008)

[4] T. Robic and B. Filipic. Demo: differential evolution for multi-objective optimization, In Proceedings of the 3rd International Conference on Evolutionary Multi-criterion Optimization(EMO'05), pp. 520–533 (2005)

[5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler : Scalable test problems for evolutionary multi-objective optimization, In TIK-Technical Report No. 112, pp. 1–27 (2001)

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan : A fast and elitist multi-objective genetic algorithm: NSGA-Ⅱ. IEEE Trans. on Evolutionary Computation, Vol. 6, No. 2, pp. 182–197 (2002)

[7] M. Koppen and K. Yoshida : Substitute distance assignments in NSGA-Ⅱ for handling many-objective optimization problems, In Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization (EMO'07), pp. 727–741 (2007)