

推定収束点を用いた対話型進化計算高速化の可能性

Accelerating Interactive Evolutionary Computation Using an Estimated Convergence Point

余 俊

Jun YU

九州大学大学院芸術工学府

Graduate School of Design, Kyushu University

○ 高木 英行

Hideyuki TAKAGI

九州大学大学院芸術工学研究院

Faculty of Design, Kyushu University

Abstract

We propose to use the estimated convergence point of population as an elite individual, replace the worst individual with the estimated convergence point of interactive evolutionary computation (IEC), and accelerate IEC convergence. We confirm its acceleration effect through an IEC simulation experiment as the preliminary experiment of a real human subjective test using IEC users and show the possibility of the acceleration effect. Unlike normal EC, the number of IEC evaluations grades is limited and small differences in fitness values cannot be distinguished, which results the estimation error of convergence point. Nevertheless, experimental results showed the effect of the proposed method.

1 はじめに

対話型進化計算は強力なコンピュータの最適化に能力と人間の評価能力を組み合わせた手法であり、性能計測が困難であるが主観的评价ならば可能であるタスク、例えば補聴器の聴きやすさを最適にしたり好みにあったレイアウトデザインをするなどの場合に用いられる。通常の進化計算は評価関数あるいは性能計測で各個体の性能を絶対値で表現するが、対話型進化計算では、人間の評価能力を用いて相対的に個体評価を行うことが多い。一番の問題点は人間の評価に時間がかかり疲労問題が避けて通れない点である。この問題を克服するために、色々な疲労軽減手法が提案されてきた [6, 8]。

本研究の最終的な目的は、推定収束点を利用することで対話型進化計算が高速化され疲労軽減につながるかどうか、効果があるとするならばどの程度の効果があるかを明らかにすることである。しかし人間のユーザを用いた対話型進化計算実験を行う前に、その有効性の可能性を確認しておく必要がある。そこで本論文では、推定収束点を対話型進化計算に利用することで高速化が期待できるかどうかを、疑似ユーザを用いたシミュレーションで明らかにすることを目的とする。

以下、第2節で進化計算の収束点を推定する手法を簡単に紹介した後、第3節で推定収束点をどのように高速化に用いるかを述べる。その後、第4節で対話型進化計算のシミュレーションを通じて推定収束点を対話型進化計算に用いることで高速化が期待できることを示し、第5節と第6節で考察と結論を述べる。

2 個体群の収束点推定法

親個体から子個体への方向ベクトルをこれまで同様移動ベクトルと呼ぶことにする [1, 5, 9, 11, 12]。さらに拡大解釈して本論文では、fitness 値の小さい個体から大きい

個体への方向ベクトルも移動ベクトルと呼ぶことにする。

個体群が収束する収束点は、移動ベクトル群が最も近づく点として推定できる。この推定収束点は数学的に計算可能である [1, 2]。最初に記号を定義し、この推定収束点の計算方法を述べる。図1の a_i と c_i を i 番目の親個体とその子個体とする ($a_i, c_i \in R^d$)。 i 番目の方向ベクトルを $b_i = c_i - a_i$ と定義する。 b_i の単位ベクトルは $b_{0i} = b_i / \|b_i\|$ 、すなわち、 $b_{0i}^T b_{0i} = 1$ と定義される。

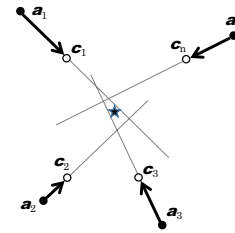


図1: ベクトル $b_i (= c_i - a_i)$ は d 次元探索空間上の親個体 a_i とその子個体 c_i から求められる。* はこれらの移動ベクトルの収束点である。

$x \in R^d$ を n 本の延長方向線分 $a_i + t_i b_i$ ($t_i \in R$) に最も近い点とする。最近傍ということは、 x から n 本の延長方向線分への距離の和、すなわち、式 (1) の $J(x, \{t_i\})$ 、が最小になる。

$$J(x, \{t_i\}) = \sum_{i=1}^n \|a_i + t_i b_i - x\|^2 \quad (1)$$

推定収束点 x から延長方向線分への最小線分は、延長方向線分と直交するので、直交条件の式 (2) を式 (1) に代入して t_i を消去する。

$$b_i^T (a_i + t_i b_i - x) = 0 \quad (2)$$

式 (1) の距離和を最小にする \hat{x} は、 x の各要素を偏微分して 0 と置くことで求められる。最後に、推定収束点

\hat{x} は式 (3) で与えられる．ここで， I_d は単位行列である．

$$\hat{x} = \left\{ \sum_{i=1}^n (I_d - b_{oi} b_{oi}^T) \right\}^{-1} \left\{ \sum_{i=1}^n (I_d - b_{oi} b_{oi}^T) a_i \right\} \quad (3)$$

3 推定収束点を利用した対話型進化計算の高速化

前節で述べた推定法で個体群の収束点が推定でき，その推定点をエリート個体として用いることで進化計算の収束を高速化させることが可能である [5, 9, 11, 12]．しかし，この考えを対話型進化計算に適用しようとするといくつかの問題が生じる．これらの問題は，fitness 数値で僅かな差も区別できる通常の進化計算と異なり，対話型進化計算ユーザは僅かな違いの類似個体を区別できないため，これらの個体に同じ評価点を与えざるを得ない点に起因する．

第 1 の問題点は，評価差が粗くならざるを得ないので推定収束点計算の誤差が大きくなる点である．第 2 の問題点は同じ評価点の個体間では収束点推定のための移動ベクトルが構成できないことである．第 3 の問題点は，最悪個体が複数あるので，エリート個体をどの最悪個体と置換すべきかが分からない点である．

第 1 の問題点に対しては，収束推定精度を向上させる手法を開発し組み合わせることで補う [10]．第 2 の問題点に対しては，一つの子個体への移動ベクトルだけでなく，評価点の異なるすべての個体間で移動ベクトルを計算して収束点推定のための情報を増やすことで補う．第 3 の問題点に対しては，最悪個体群のどれか一個体をランダムに選択し，エリート個体とする推定収束点と入れ替えることで補う．

本論文では，推定収束点をエリート個体として利用することで対話型進化計算の高速化に寄与できるかどうかを確認する．対話型進化計算の収束が早くなれば，対話型進化計算の実用上の課題であるユーザの疲労軽減につながる．推定収束点の推定精度は移動ベクトルに影響される．対話型進化計算では個体の比較評価時に評価の差がほとんどない場合は同じ評価値を与えるので，親子個体の評価値が同じになることもよく起きる．この場合は収束点推定に使う移動ベクトルが生成できないので，収束点の推定精度を上げるためには移動ベクトル数を増やす必要がある．例えば，親子個体に関係なく評価値の低い個体から高い個体への方向ベクトルを移動ベクトルとすることが考えられる．

この研究の第一歩として，本提案手法の可能性を探る．したがってこれまで推定収束点をエリート個体とすること

で進化計算の高速化に利用するこれまでの研究 [5, 9, 11, 12] を，対話型進化計算に拡張できるかどうかをシミュレーションで確認することにとどめ，実際の人間のユーザを用いた評価実験は本論文の実験で応用可能性が確認されてから行うこととする．

最小化問題の評価関数を用いて対話型進化計算シミュレーションを行う．環境適応の語源からは，厳格に言えば fitness 値が大きいほど良い個体であることを意味するが，fitness 値と評価関数値の大小関係での混乱を避けるため，最小化問題を扱う場合は fitness 値 (= 評価関数値) が小さい程良い個体である，として以下記述する．

$f_{EC}(x_i)$ を i 番目個体の fitness 値， F_{max} と F_{min} を通常進化計算での個体群の最悪・最良 fitness 値とする．式 (4) によって進化計算 fitness 値を対話型進化計算評価値に変換する [3]．

$$f_{IEC}(x_i) = \text{ROUND} \left(\frac{F_{max} - f_{EC}(x_i)}{F_{max} - F_{min}} * s \right) \quad (4)$$

ここで ROUND() は四捨五入関数， $f_{IEC}(x_i)$ は i 番目個体の進化計算評価値 $f_{EC}(x_i)$ を対話型進化計算評価値に変換した値， s は対話型進化計算の評価段階数 (本論文での実験では $s = 5$ 段階評価を採用) である．

4 対話型進化計算シミュレーションによる収束高速化の評価実験

提案手法の有効性を評価するため， n 個の d 次元ガウス関数を組み合わせた式 (5) の混合ガウス関数を対話型進化計算シミュレーションに用いる [3]．混合ガウス関数を用いる根拠は，人間は細かい差を区別して評価できないため心の評価系 (fitness 景観) はなだらかであろうとする仮説と，実問題では多峰性があり人間は多峰性の区別ができることに基づく [4]．

$$f(x) = - \sum_{j=1}^n \left\{ a_j \exp \left(- \sum_{i=1}^d \frac{(x_{ij} - \mu_{ij})^2}{2\sigma_{ij}} \right) \right\} \quad (5)$$

進化計算アルゴリズムには差分進化を用い，表 1 にその実験パラメータ設定を示す．評価回数を考慮した公平な実験条件にするために，世代数での収束評価ではなく評価回数軸での収束評価を行う．現実的に対話型進化計算ユーザが長時間評価を続けることは困難なため，人間が評価できそうな回数を考慮してシミュレーション実験での最大評価回数を設定する．三つの異なる次元タスクそれぞれで 30 試行の実験を行い，最大評価回数 MAX_{NFC} 時点で推定収束点をエリート個体として用いた対話型差分進化と用いなかった対話型差分進化の平均評価値間に有意な差があるのかどうかを Wilcoxon の符号検定で確認する．

表 1: 差分進化のパラメータ設定

個体数	20
scale factor F	0.8
交差率	0.9
演算アルゴリズム	DE/rand/1/bin
最大評価回数 MAX_{NFC}	500
タスクの次元数 D	2, 5, 10 次元
実験試行回数	30
全次元変数の探索範囲	$[-6, 6]$

表 2: 30 試行の平均最良 fitness 値間の Wilcoxon の符号検定結果. A と B は各々, 推定収束点対話型進化計算にエリート個体として用いた提案手法と用いない従来型対話型進化計算を意味し, $A \gg B$ は提案手法 A が従来手法 B よりも危険率 1% で有意に収束が早い, $A \approx B$ は両者間に有意差なし, を意味する.

混合ガウス関数の構成ガウス関数数	2-D	5-D	10-D
$n = 1$	$A \gg B$	$A \gg B$	$A \gg B$
$n = 2$	$A \gg B$	$A \gg B$	$A \gg B$
$n = 3$	$A \approx B$	$A \gg B$	$A \gg B$
$n = 4$	$A \approx B$	$A \gg B$	$A \gg B$

検定結果を表 2 に示す. また, 10 次元混合ガウス関数をタスクにした場合の提案対話型進化計算と従来型対話型進化計算の平均収束特性を図 2 に示す.

5 考察

まず提案手法の効果について述べる. 騙し問題のように個体群が大局的最適解に向かわない場合は推定収束点在实际の最適点近傍になるとは限らないが, 個体群が大局的最適解に向かうのであれば, 推定収束点は实际の最適点に近くなり, 次世代での有力な個体候補になり得る. 言い換えると, 推定収束点を利用する提案手法は, 進化計算の探索ができない時に助ける手法ではなく, 最適化がうまく進む時に加速する手法であると言える. また, 騙し問題のように推定収束点在实际の大局的最適点と離れた場合にエリート個体として提案手法を用いた場合は, 多くある個体群の 1 個体の効果が期待できないだけで全体に与える影響は小さいが, 探索が上手くいく場合はそのエリート個体が強力な探索点になる. すなわち, 最悪個体をエリート個体と置換する手法は, *Low Risk, High Return* 手法と言える [7].

幸いなことに, 人間は細かな差の評価の区別が付かないため対話型進化計算の fitness 景観 (fitness 景観の評価

軸は心理尺度) はなだらかで単純な景観になりがちであり, 進化計算のベンチマーク関数のような探索が困難な fitness 景観を考える必要はない [4] ので, 提案手法は多くの場合収束の加速化が期待できる. エリート個体 (推定収束点) を評価する手間が 1 回増えるが, 評価実験結果を見て判るように, *cost-performance* の観点からそれ以上の収束高速化が期待できる

検定結果から, 多くの場合で提案手法が収束高速化に寄与していることが判る. 収束点推定アルゴリズムは基本的に単峰性の場合に適用する推定方法であるため, 多峰性問題の探索初期世代のように個体群サブグループが異なる局所最適解を目指す場合, 全移動ベクトルを使って収束点を推定すると大きな誤差が生じることになる. しかし世代に連れて探索が進むと大局的最適解に向かう個体数が増えるので段々単峰性問題のようになり推定収束点が实际の大局的最適解に近づく. この理論的な振る舞いは図 2 の平均最良 fitness 曲線からも確認できる. 初期世代から提案手法が有効に働くようにするためには, 局所最適解領域を分離する手法を組み合わせることが必要であり, 現在筆者等は本論文での取り組みと並行してこの分離手法にも取り組んでいる [9, 10, 11, 12].

次に, 対話型進化計算シミュレーションのパラメータ設定について考察する. 人間の能力には限りがあるので, 個体数や評価回数を大きくすることは現実的ではない. しかし個体数が多いほど収束点推定のための移動ベクトル数を増やすことができるため有利である. そのため, どのようなパラメータ設定にすることが良いのかという点に注意を払う必要がある. 特に, タスクの次元数と移動ベクトル数との関係, 限られた個体数の範囲でいかに妥当な移動ベクトルを得るか, 等の点から, このパラメータ設定問題を考慮すべきである.

6 結論

本論文では, 個体群の推定収束点を対話型進化計算のエリート個体として利用することで収束の高速化に役立たせることができることをシミュレーション実験で確認した. この効果は, エリート個体を余計に 1 回評価するという手間以上の効果であることを確認しており, 人間のユーザを用いた実際の対話型進化計算に利用できる可能性が示された.

今後は, 実際の対話型進化計算タスクと人間のユーザを用いた実験を行い, 効果を分析する予定である. また考察で述べたパラメータ設定に関する研究も必要である.

謝辞

本研究はJSPS 科学研究費 (課題番号 JP15K00340) の助成を受けたものである .

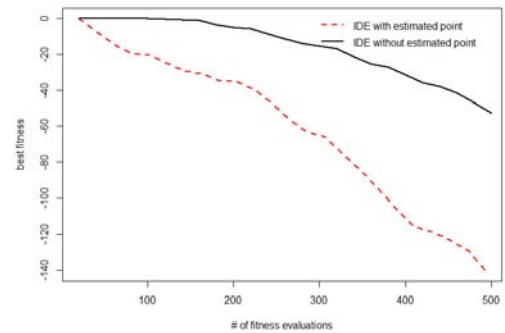
参考文献

- [1] 村田昇, 西井龍映, 高木英行, 裴岩「世代間移動ベクトル群の収束点推定法」2014 進化計算シンポジウム, pp.210-215, 廿日市市, 広島県 . (2014 年 12 月 20-21 日).
- [2] Murata, N., Nishii, R., Takagi, H., and Pei Y., “Analytical Estimation of the Convergence Point of Populations,” IEEE Congress on Evolutionary Computation (CEC2015), pp.2619-2624, Sendai, Japan (May 25-28, 2015).
- [3] Nakano, Y. and Takagi, H., “Influence of Quantization Noise in Fitness on the Performance of Interactive PSO,” IEEE Congress on Evolutionary Computation (CEC2009), Trondheim, Norway, pp.2146-2422 (May 18-21, 2009).
- [4] 裴岩, 高木英行「3 点および 4 点比較ベースの対話型差分進化と差分進化」進化計算学会論文誌, vol.3, no.3, pp.98-108 (2012).
- [5] 裴岩, 余俊, 高木英行「推定収束点を用いた進化計算高速化の評価」進化計算シンポジウム 2015, pp.292-297, 西尾 (2015 年 12 月 19-20 日).
- [6] Takagi, Hideyuki, “Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation,” Proceedings of the IEEE, vol. 89, no. 9, pp.1275-1296 (2001).
- [7] 高木英行, 印具毅雄, 大西圭 「単峰性関数当てはめによる GA 収束高速化」知能と情報 (日本知能情報ファジィ学会誌), vol.15, no.2, pp.219-229 (2003).
- [8] Takagi, Hideyuki, “New types of IEC Applications and Latest Research on Reducing User Fatigue” 8th Int. Conf. on Adaptive Computing in Design and Manufacture (ACDM2008), Clifton, Bristol, UK (April 29 - May 1, 2008).
- [9] 余俊, 高木英行「多峰性最適化問題での局所最適解推定高度化のための補正法 - 局所最適解が 2 個の場合 -」第 9 回進化計算研究会, pp.92-97, 神戸 (2015 年 9 月 7-8 日).
- [10] Yu, J. and Takagi, H. “Clustering of Moving Vectors for Evolutionary Computation,” 7th Int. Conf. on Soft Computing and Pattern Recognition (SoCPaR2015), pp.169-174, Fukuoka, Japan (Nov. 13-15, 2015).
- [11] Yu, J., Pei Y. and Takagi, H., “Accelerating Evolutionary Computation Using Estimated Convergence Point,” IEEE Congress on Evolutionary Computation (CEC2016), pp.1438-1444, Vancouver, Canada (July 24-29, 2016).
- [12] 余俊, 高木英行「個体間距離順位とフィットネス順位に基づく局所解領域の推定」第 12 回進化計算学会研究会プログラム, 福岡, pp.203-206 (2017 年 3 月 13-14 日)

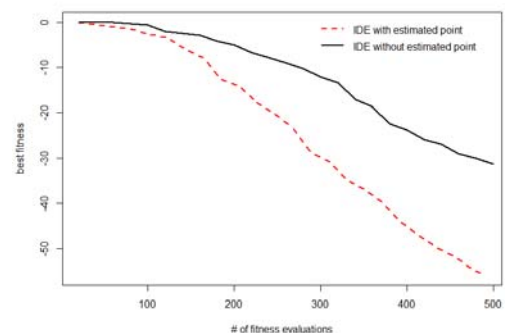
連絡先

高木 英行 (九州大学大学院芸術工学研究院)

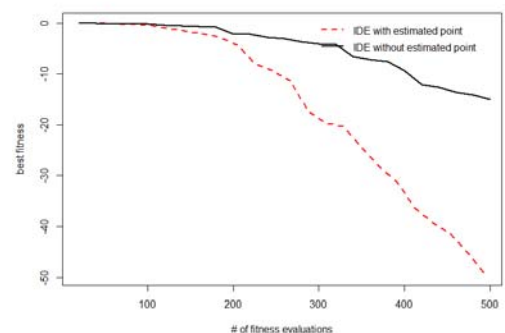
URL: <http://www.design.kyushu-u.ac.jp/~takagi>



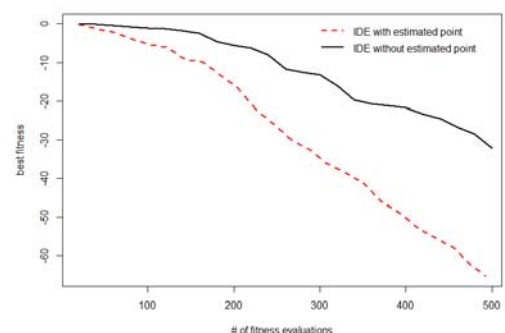
(a) 構成ガウス関数が 1 個の場合 ($n = 1$)



(b) 構成ガウス関数が 2 個の場合 ($n = 2$)



(c) 構成ガウス関数が 3 個の場合 ($n = 3$)



(d) 構成ガウス関数が 4 個の場合 ($n = 4$)

図 2: 10 次元混合ガウス関数を用いたシミュレーションの通常対話型差分進化 (IDE) と提案手法を組み込んだ対話型差分進化の最良 fitness 収束特性の 30 試行平均 .