進化戦略における選択操作に関する一考察

張　明　　今井　智彦　　杉山　正晴

A Two-Step Selection Scheme for Evolutionary Optimization

Ming CHANG　　Tomohiko IMAI　　Masaharu SUGIYAMA

あらまし　進化計算における選択操作を個体の繁殖力と生存力の視点から見ると，遺伝的アルゴリズムは繁殖力に，進化戦略および進化的プログラミングは生存力にそれぞれ重点を置いていることが分かる．本稿では，まず準種モデルを用いて，選択操作の違いが進化過程に及ぼす影響を示す．次に進化戦略に繁殖力と生存力の両方に選択圧をかける選択操作を提案し，実関数最適化問題および制約付最適化問題においてその性能を検証した．

キーワード　進化計算，選択操作，準種モデル，進化戦略

１．はじめに

現実の複雑なシステムを対象とした最適化問題を扱う場合は，最適性の保証がなくとも実用的な実行可能解を求めるための系統的な計算手法が必要となる．そのため，1950年代や1960年代の計算機科学者達は，最適化問題を解くための道具として進化を利用することができると考え，進化システムの研究を始めた．たとえば，生物の進化過程をモデル化する組合せ最適化問題のための遺伝的アルゴリズム(Genetic Algorithms: GA)，実数値変数を持つ関数の最適化に進化戦略(Evolution Strategies: ES)，有限状態機械(Finite State Machine)を進化させるための進化的プログラミング(Evolutionary Programming: EP)など，様々なアルゴリズムが提案された．その後，計算機技術の飛躍的な発展により，それらに関連する研究領域ならびに関連研究領域に研究者が集まり，進化計算(Evolutionary Computation: EC)と呼ばれる分野が形成された．現在では，進化的アルゴリズム(Evolutionary Algorithms: EA)間の境界が曖昧になりつつあるが，以下では，まずESを中心にEP及びGAについて述べ，ECにおける選択操作を繁殖力と生存力の視点から考察する．次に準種モデルに用いて繁殖力選択と生存力選択の違いがエラー閾値に与える影響を示す．ESに繁殖力と生存力の両方に選択圧をかける選択操作を提案し，計算機実験を通じてその有効性を検証する．

２．進化計算手法の概略

進化戦略(ES)[1]は，1964年ドイツのTechnical University of Berlinでノズルの設計に際して開発された．翌年には1個体の親が1個体の子を産み，親子の中の適応度の高い個体が次世代の親となる(1+1)-ESが発表され，その後，m ($m>1$)個体の親から1個体の子を生成し，最劣個体と置き換える($m+1$)-ESと，1個体の親から l 個体の子を生成する(1, l)-ESを経て，並列計算と自己適応を可能にした(m, l)-ESに到達した($m<l$ とする)．(m, l)-ESにおいては，m 個体からなる親集団から l 個体の突然変異した子を生成し，その中の適応度の最も高い m 個体を次世代の親とする．

n 次元の実関数最適化問題を対象とする場合，n 次元実数値ベクトル x_i と h_i を，それぞれ変数と変数の変異幅を制御する戦略パラメータとし，CES(Classical Evolution Strategy)[2]の計算手続きを以下のように定義する：

$$h'_k(j) = h_i(j)\exp(t'N(0,1) + tN_j(0,1))$$
$$x'_k(j) = x_i(j) + h'_k(j)N_j(0,1)$$

ただし $t = 1/\sqrt{2\sqrt{n}}$，$t' = 1/\sqrt{2n}$．$x_i(j)$ と $h_i(j)$ は i 番目親個体の変数ベクトルと戦略パラメータベクトルの第 j 成分，$x'_k(j)$ と $h'_k(j)$ はその親個体から生成される k 番目子個体の相応成分をそれぞれ示す．$N_j(0,1)$ は j ごとに得られるそれぞれ独立な平均0，標準偏差1の標準正規分布に従う乱数値である．また，本稿では変数ベクトルと戦略パラメータベクトルをまとめてMulti-Parent Discrete Recombination[2]を適用したものを以下のように定義し，D-CES[3]と呼ぶ．

$$h'_k(j) = h_{c_j}(j)\exp(t'N(0,1) + tN_j(0,1))$$
$$x'_k(j) = x_{c_j}(j) + h'_k(j)N_j(0,1)$$

c_j は j ごとに得られるそれぞれ独立な $\{1,2,....,m\}$ に一様分布する整数乱数である．

進化的プログラミング(EP)は，1960年代前半にL.J. Fogelが有限状態機械を進化させる枠組として提案したものを，その息子D.B.Fogelが1990年代に“meta-EP”[4]として実関数最適化問題向けに再定式化したものである．

その後多くのベンチマーク問題だけでなく実問題でもその優れた探索能力が検証されている．EPはESとほぼ同じ手続きによって構成されるが，EPには①種の進化をモデル化しているため組替え演算を適用しない，②確率的選択過程を用いる，などの特徴がある．

遺伝的アルゴリズム(GA)は，集団(個体群)の進化過程をモデル化したアルゴリズムであり，離散システムの最適化のみならず，様々な最適化に応用可能な手法である．GAでの多くの概念はJ.H. Holland[5]によってはじめて記述されたが，Holland自身はGAを最適化アルゴリズムより，むしろ適応アルゴリズムとして捉えている．GAの最適化問題への適用は，Hollandに学んだK.A. De Jongによって大きく発展された．GA研究の普及に大きく貢献したのは1989年のD.E. Goldbergの著書であり，その中には理論と共に数多くの例題が解説されている．その後のGAの拡張として，不定長な遺伝子表現の木構造を用いる計算機のプログラムを進化させる遺伝的プログラミング(Genetic Programming)が1992年にJ.R. Koza[6]によって提案されている．

これまでにEAが様々な問題，分野に適用されてきた．例えば，最適化，自動プログラミング，機械学習，経済学，生態系，集団遺伝学，社会システムなどがある．本稿では実関数最適化問題に焦点を当てている．

3．進化計算における選択操作

EAの設計においては，突然変異と選択が必要最小限の操作である．特に選択操作は中心的な役割を果たす．これは突然変異操作が問題解のコーディングの仕方に依存するのに対して，選択操作はある程度の一般性を備えているためであり，ランダムに突然変異操作を施した個体を評価，選択することで，問題空間における探索に方向性を与えることが可能となるためである．選択なしの進化計算はただのランダム探索となる．これまでに，GAでは多様な選択操作が提案されてきたのに対して，ES及びEPでは，決定論的な選択操作が主流である．たとえば，ESとEPに多用されている $(m,^{+}l)$ -Selection[7]では，親個体が同じ繁殖力をもつと想定し，1親個体あたり平均にして l/m 個の子個体を産み，子集団((m,l) の場合)あるいは子集団と親集団($(m+l)$ の場合)から，適応度の最も高い m 個体が次世代の親個体として選ばれる．しかし，自然界の生物を考えると，個体が同じ繁殖力を持つという仮定は明らかに非現実的である．また孔雀の尾羽のような，生存力と繁殖力に関しては対立している形質が進化してくることもありえる[8]．進化計算での選択操作[9,10,11]を生存力と繁殖力の視点から見ると，ESとEPは生存力に，GAは繁殖力に，それぞれ重点をおいていることが分かる．このような違いには二つの原因が考えられる．一つは，それぞれの進化的アルゴリズムは生物進化の異なる側面を強調しているためである．たとえば，GAは個体群の進化，ESは人工育種，EPは種の進化，をそれぞれモデルにしている．もう一つはESとEPでは，変数とその変異幅を制御する戦略パラメータの両方を個体表現に用い，戦略パラメータも突然変異操作の対象となるため，その効果を適切に評価するには1親個体あたり平均にして親個体より適応度の高い子個体をすくなくとも一つ生成する必要がある[12]．$(m,^{+}l)$ -Selectionにおいては，それが親個体あたり平均にして l/m 個の子個体を産むということによって保証されている．これがESでの選択操作は生存力にのみ注目していることの原因である．本稿では，繁殖力と生存力の両方に選択圧をかける選択操作をESに導入することを提案し，この新しい選択操作の性能を計算機実験で検証する．

4．準種モデル(The Quasi-Species Model)

準種モデル[13]は生命起源にかかわるRNA分子集団の進化を記述するために提案された．それが分子進化を考える場合の重要な概念の一つ：エラー閾値(error threshold)をもたらした．エラー閾値はある長さの遺伝情報を集団の中に保持されるための突然変異率の上限を示すものである．突然変異率がエラー閾値を超えると，分子集団のなかの遺伝情報は蓄積された突然変異に由来するエラーによって破壊されてしまう．Maynard Smithはマスター種及び変種の2種からなる準種モデルを以下のように定式化している[14]．

$$\begin{cases} \dot{x}_m = x_m(A_m Q - E) \\ \dot{x}_j = x_j(A_j - E) + x_m A_m(1-Q) \end{cases}$$

ただし $Q=(1-p)^L$．L と p はRNA分子の長さと塩基あたりの突然変異率を表す．E は死亡率をいい，$x_m + x_j = 1$ となるように $E=(A_m x_m + A_j x_j)/(x_m + x_j)$ と設定される．x_m, x_j 及び A_m，A_j はそれぞれマスター種と変種に属する分子の集団の大きさと再生率である．またマスター種は変種より再生率が高い($A_m > A_j$)とする．そのためマスター種に準種集団が獲得した有用な遺伝情報が含まれていることになる．このモデルではマスター種と変種に同じ死亡率 E が仮定されているため，繁殖率(A_m, A_j)のみに選択をかけることになる．準種集団が平衡状態にある場合，つまり，マスター種と変種の個体数の変動がなくなるときに($\dot{x}_m = 0$，$\dot{x}_j = 0$)，上の式は次のようになり，

$$\begin{cases} A_m Q = E \\ x_j(E - A_j) = x_m A_m(1-Q) \end{cases}$$

この時，有用な遺伝情報が集団内に存在することは $x_m > 0$ に相当するため，次の不等式が得られる．

$$Q > A_j / A_m$$

$p \ll 1$ の場合，$\ln(1-p) \approx -p$ となるので，上の不等式は

次のように書き換えられる．

$$L < \frac{\ln(A_m / A_j)}{p}$$

この不等式は，ある突然変異率のもとで得られる遺伝情報の長さの上限，あるいはある長さの遺伝情報を得るための突然変異率の上限，つまりエラー閾値を示している．Maynard Smithが指摘したように，この結果には次のような「鶏・卵」ジレンマを抱えている：長い遺伝情報を得るためにエラー校正ための酵素が必要となるが，そのような酵素は長い遺伝情報のもとではじめて可能となる．そのジレンマの解決策の一つとしてハイパーサイクル(Hypercycle)モデル[15]が提案されたが，Schwefel[16]はこの問題を少し違う視点で捉えた．Schwefelはマスター種と変種に同じ再生率 A，異なる死亡率を割り当て，モデルの再定義した：

$$\begin{cases} \dot{x}_m = x_m(AQ - E_m) \\ \dot{x}_j = x_j(A - E_j) + x_m A(1-Q) \end{cases}$$

この場合，遺伝情報の長さと突然変異率の関係を表す不等式は次のようになる．

$$L < \frac{\ln(E_j / E_m)}{p}$$

Schwefelは生命起源の初期においては，死亡率，つまり生存力(RNA分子の構造的な安定性など)にかける選択圧は繁殖力にかける選択圧よりも大きいこと，言い換えると $E_j / E_m > A_m / A_j$ が大いにあるので，同じ突然変異率でも，Maynard Smithのモデルより長い遺伝情報が得られることを指摘した．

このように，Maynard SmithとSchwefelのモデルはそれぞれ繁殖力と生存力に選択圧をかけているが，繁殖力と生存力の両方に選択圧をかけるように，モデルを次のように定式すると，

$$\begin{cases} \dot{x}_m = x_m(A_m Q - E_m) \\ \dot{x}_j = x_j(A_j - E_j) + x_m A_m(1-Q) \end{cases}$$

遺伝情報の長さと突然変異率の関係は

$$L < \frac{\ln(A_m / A_j) + \ln(E_j / E_m)}{p}$$

のようになる．この場合に得られる遺伝情報の長さの上限は，Maynard Smith と Schwefel のモデルでそれぞれ得られた遺伝情報の長さの上限の和となる．

5．実関数最適化問題

実関数最適化問題とは，n 次元の実行可能領域 $S(S \subseteq \Re^n)$ を持つ有界な実数値目的関数 $f(x)$ において，$f(x_0)$ が最小となる変数ベクトル $x_0 \in S$ を見つける問題である．ESは実数値を個体表現にしているため，実関数最適化問題に特に適している．すでに述べたように，CESでは，個体の繁殖力に選択をかけていない．本稿では従来の生存力に対する (m, l) -selectionに加えて，個体の繁殖力には以下のような線形ランキング選択をかける．適応度順にランクされた親集団の i 番目個体の親になる確率 p_i を i の線形関数として

$$p_i = \frac{1}{m}\left(a^+ - \left(a^+ - a^-\right)\cdot\frac{i-1}{m-1}\right)$$

のように定義する[7]．最良個体と最劣個体の繁殖力はそれぞれ a^+，a^- とし，制約条件 $\sum_{i=1}^{m} p_i = 1$ から，$a^+ + a^- = 2$ が得られるため，$a^+ (0 < a^+ < 2)$ によって繁殖力にかける選択圧力を調節することができる．計算機実験は11個のテスト関数において実施した[17,18]．ここでは，下記の関数に関する実験結果を示す．

$$f_1 = \sum_{i=1}^{n} x_i^2,\ -100 \le x_i \le 100$$

$$f_5 = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2],\ -30 \le x_i \le 30$$

$$f_8 = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i) + 20 + e,\ -32 \le x_i \le 32$$

$$f_9 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1, -600 \le x_i \le 600$$

すべての実験において，$m = 30, l = 200$，関数次元数 $n = 30$ とし，a^+ を0.0から2.0まで0.1刻みで増やし，実験結果に与える影響について調べた．各問題の100試行に関する結果の第1，2，3四分位点及び最小値，最大値を図に示した．さらに $a^+ = 1$ の時，$a^+ + a^- = 2$ から $a^+ = a^- = 1$ が得られるため，最良個体と最劣個体に同じ繁殖力を割り当てることになり，繁殖力に選択圧をかけないことに相当する．また $0 < a^+ < 1$ の場合では，個体の繁殖力に対する選択圧と生存力に対する選択圧が相反になり，結果的に個体に対する選択圧力を弱めることになる．同様な理由から，$1 < a^+ < 2$ の場合は，選択圧力を高めることになる．

図1，図2には f_1 に関するCESとD-CESの実験結果を示している．CESに関しては，a^+ の増加に伴い，アルゴリズムの性能は緩やかに向上している．D-CESに関しては a^+ の値が1.0から1.1に変わった時，性能の飛躍的な向上が見られるが，0.0～1.0及び1.1～2.0の間の変動は大きくない．この場合，一般的に選択圧の増大が性能の向上につながる．

図3，図4は f_5 に関するCESとD-CESの実験結果を示している．f_1 の場合と同じ傾向を示しているが，D-CESの場合，性能の向上がもっとも著しいのは $a^+ = 0.5 \sim 0.8$ の間に見られる．注意しなければならないのはこの時，$a^+ < 1$ になっていることである．

図5，図6は f_8 に関する実験結果を示している．CESの場合では，a^+ の値に関わらず，ほぼ同様な性能を示し

ているのに対して，D-CESでは，$a^+ = 1.1$のところで大きな性能向上が見られる．しかし，それ以外のところではCESと同様に性能上の変化はあまりない．

図7，図8のf_9に関する結果では，CESとD-CESが同じような傾向を示す．まず，$0 < a^+ < 1$の間，a^+の増大に伴い，性能の向上が見られるが，$1.1 < a^+ < 2$の間では，性能上の変化が見られない．特に，$1 < a^+ < 1.1$のところでは性能上の大きな劣化が見られる．

以上の実験では生存力と繁殖力に対する選択は同じ目的関数を通して行われたため，結果的に選択圧の変化になるが，次の章ではそれぞれ制約条件と目的関数によって評価し，制約付実関数最適化問題へ適用した結果を示す．

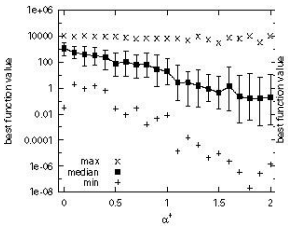

図1：CES on f_1

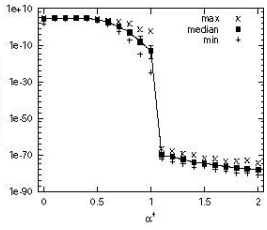

図2：D-CES on f_1

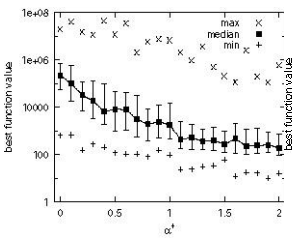

図3：CES on f_5

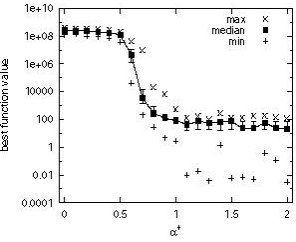

図4：D-CES on f_5

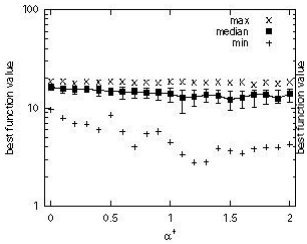

図5：CES on f_8

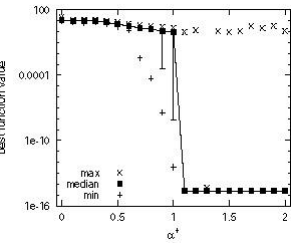

図6：D-CES on f_8

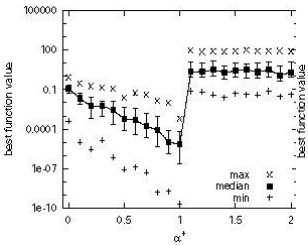

図7：CES on f_9

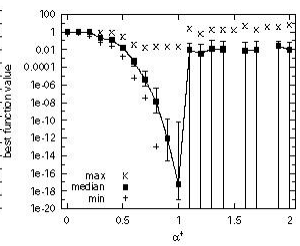

図8：D-CES on f_9

6．制約付実関数最適化問題

一般的に，制約付実関数最適化問題は以下のように定式化できる：

$$\begin{aligned}&\text{Minimize } f(x),\ x = (x_1, x_2, ..., x_n) \in \Re^n\\ &\text{subject to}\quad \underline{x}_i \le x_i \le \overline{x}_i \quad i = 1,2,...,n\\ &\qquad\qquad g_j(x) \le 0 \qquad j = 1,2,...,q\\ &\qquad\qquad h_j(x) = 0 \qquad j = q+1,...,m.\end{aligned}$$

さらに，計算機上で処理する場合には，$h_j(x) = 0$を下記のような不等式制約に書き換えられる．

$$g_j(x) \equiv |\, h_j(x) - d \le 0\,|, \quad j = q+1,...,m.$$

dは小さな値であり，誤差の許容範囲を表す．

EAでは，制約条件を扱うには“Penalty Function”[19]という手法があり，それは制約条件を満たさない解にはペナルティを課する．本稿では下記のような“Quadratic Loss Function”というペナルティ関数を用いる．

$$f(x) = \sum_{j=1}^{m} \max\{0, g_j(x)\}^2$$

従来手法の多くはペナルティ関数値と目的関数値を一つの適応度値にまとめるのに対して，本稿では，ペナルティ関数値を解の生存力，目的関数値を解の繁殖力とみなし，それぞれに対して選択圧力をかけることによって問題の解決を試みる．計算手続きは他の手法[20,21]と比較するため，文献[20]に従い，次のように定義した．

$$h'_k(j) = \frac{1}{2}(h_i(j) + h_{h_j}(j))\exp(t'N(0,1) + tN_j(0,1))$$

$$x'_k(j) = x_i(j) + h'_k(j)N_j(0,1)$$

提案したアルゴリズムと文献[20]のアルゴリズムと違いは，①選択操作が異なること，②戦略パラメータに上限を設けないこと，③問題空間を定義する$\underline{x}_i \le x_i \le \overline{x}_i$を，$x_i \le \overline{x}_i$と$-x_i \le -\underline{x}_i$のように整理し，他の制約条件とまとめて扱うことにある．

繁殖力選択及び生存力選択をともに“Truncation Selection”として実装した．まず，lの子個体からペナルティ関数値に従い，上位のz個体を選び出し，目的関数によって評価し，その中から上位のm個体を次世代の親個体とする．zは繁殖力と生存力にそれぞれかける選択圧間のバランスを表し，その値がアルゴリズムの性能に強く影響すると予想される．

計算機実験においてはzの値を35から110まで5刻みで増加させ，それが実験結果に与える影響について調べた．提案手法を文献[20]の中の13個のベンチマーク問題に適用した[22]．各問題に対して最適なzの値は異なるが，13個中11個の問題に対して$z = 55$の時，問題g06とg10においては$z = 35$の時，相対的に良い解を見つけた．それらの結果を表1に，また他手法[20,21]との比較を表2に示す．

表2から分かるように，“Best Result”項目に関して，

表1　13個のベンチマーク問題に関する実験結果.

fcn	optimal	fea	best	median	worst	mean	st. dev.
g01	-15.000	14	-15.00000	-15.00000	-13.00000	-14.857088	0.515064
g02	-0.803619	30	-0.801925	-0.743880	-0.644350	-0.745795	0.039788
g03	-1.000	30	-1.000478	-0.999783	-0.969545	-0.997956	0.006124
g04	-30665.539	30	-30665.538584	-30665.299304	-30640.413317	-30662.783855	5.902188
g05	5126.498	21	5126.496721	5127.212107	5134.712328	5128.221492	2.358852
g06	-6961.814	2	-5669.343588	-5358.511179	-5047.678770	-5358.511179	310.832409
g06*	-6961.814	28	-6961.805956	-6960.929617	-3772.049822	-6741.525009	740.525707
g07	24.306	30	24.330104	24.531033	26.139391	24.663972	0.421972
g08	-0.095825	30	-0.095825	-0.095825	-0.095825	-0.095825	0.000000
g09	680.630	30	680.631577	680.718828	681.649279	680.838015	0.256505
g10	7049.331	0	-	-	-	-	-
g10*	7049.331	30	7069.294325	7251.500158	8067.262840	7334.744715	273.519647
g11	0.750	30	0.749900	0.749903	0.751233	0.750002	0.000266
g12	-1.000000	30	-1.000000	-1.000000	-0.994375	-0.999062	0.002096
g13	0.053950	30	0.053943	0.055190	0.448195	0.086380	0.096520

表2　提案手法(V&F: Viability and Fertlityの略語)と文献[20,21]（それぞれR&Y，K&Mで表している）の比較.

	Best Result			Mean Result		
fcn	V&F	R&Y	K&M	V&F	R&Y	K&M
g01	-15.000	-15.000	-14.7864	-14.857	-15.000	-14.7082
g02	-0.801925	-0.803515	-0.79953	-0.745795	-0.781975	-0.79671
g03	-1.000	-1.000	-0.9997	-0.9979	-1.000	-0.9989
g04	-30665.538584	-30665.539	-30664.5	-30662.783855	-30665.539	-30655.3
g05	5126.4967	5126.497	-	5128.221	5128.881	-
g06	-6961.806	-6961.814	-6952.1	-6741.525	-6875.940	-6342.6
g07	24.330	24.307	24.620	24.664	24.374	24.826
g08	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	-0.0891568
g09	680.631	680.630	680.91	680.838	680.656	681.16
g10	7069.294	7054.316	7149.9	7334.745	7559.192	8163.6
g11	0.7499	0.750	0.75	0.750	0.750	0.75
g12	-1.00	-1.00	-0.999999857	-0.999062	-1.0	-0.999134613
g13	0.053943	0.053957	0.054	0.086380	0.057006	0.064

提案手法は手法[21]より良い，またg02とg10を除いて手法[20]と同じ最適解を見つけた．g08においては三つの手法がともに最適解を見つけた．提案手法がg05，g11とg13において最適解より良い解を見つけた原因は等式制約を不等式制約条件に書き換えたことにある．“ Mean Result ”に関して提案手法の性能はg03，g12とg13以外では手法[20]と手法[21]の間に位置する．

7．おわりに

生物の集団遺伝・進化過程を模倣した最適解を探索する計算モデルを総じて進化的アルゴリズム（EA）と呼び，これまでに様々のアルゴリズムが提案されているが，これらのアルゴリズムに共通するのは，①問題の解を個体（計算機内のデータ）として，その集合（集団）を保持すること，②突然変異，組替えなどの遺伝演算子を適用し，新しい個体を生成すること，③選択によって，良い解を表す個体が保存されること，である．その中，特に選択操作は中心的な役割を果たす．これは突然変異操作や問題解のコーディングの仕方は具体的な問題に依存するのに対して，選択操作はある程度の一般性を備えているためである．

本稿では，まず代表的なEAについて簡単な紹介をした．それから，EAにおける選択操作を繁殖力と生存力の視点から考察し，それぞれの特徴を挙げた．次に準種モデルに用いて繁殖力選択と生存力選択の違いがエラー閾値に与える影響を示した．進化戦略（Evolution Strategies: ES）における従来の選択操作においては，個体の生存力に選

択をかけることによって探索に方向性を与えるのが一般的であるが，本稿では，繁殖力にも選択をかける選択操作を提案し，ESの性能が改善されることを実関数最適化問題及び制約付実関数最適化問題を通して示した．計算機実験における提案した選択操作の実装はES及びTruncation Selectionによって行われたが，ほかのEAや既存の選択操作によっても実現できる．

文　献

[1] Schwefel H.-P., *Evolution and Optimun Seeking*, John Wiley & Son, 1995.

[2] Back T. and Schwefel H.-P.，“An overview of Evolutionary Algorithms for parameter optimization”, *Evolutionary Computation* 1(1), pp.1-23, 1993.

[3] Chang M., Ohkura K. and Ueda K., “Some Experimental Observation of $(m/m, l)$-Evolution Strategies”, *Proc. of the 2001 Congress on Evolutionary Computation*, pp.663-670, IEEE Press, 2001.

[4] Fogel D.B., *Evolutionary Computation Toward a New Philosophy of Machine Intelligence*, IEEE Press, 1995.

[5] Holland J.H., *Adaptation in Natural and Artificial System*, MIT Press, 1992.

[6] Koza J.R. , *Genetic Programming*, MIT Press, 1992.

[7] Back T., *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, 1996.

[8] Sober E., *Philosophy of Biology*, Westview Press, 2000.

[9] Goldberg D.E. and Deb K. “A Comparative analysis of selection schemes used in genetic algorithms”, *Foundations of genetic algorithms*, pp.69-93, 1991.

[10] Blickle T. and Thiele P.“A comparison of selection schemes used in evolutionary algorithms”, *Evolutionary Computation* 4(4), pp.361-394, 1997.

[11] Chakraboty U.K. and Deb K.,“Analysis of Selection Algorithms: A Markov Chain Approach”, *Evolutionary Computation* 4(2), pp.133-167, 1997.

[12] Beyer H.-G., *The Theory of Evolution Strategies*, Springer, 2001.

[13] Eigen M., “Selforganization of matter and the evolution of biological macromolecules”, *Naturwissenschafter* 58, pp.465-532, 1971.

[14] Maynard Smith J., “Models of Evolution”, *Proc. R. Soc. Lond*. B 219, pp.315-325, 1983.

[15] Eigen M. and Schuster P.“The hypercycle. A principle of natural self-organization. Part A: emergence of the hypercycle”, *Naturwissenschafter* 64, pp.541-565, 1977.

[16] Schwefel H.-P., “Deep insight from simple models of evolution”, *BioSystems* 64, pp.189-198, 2002.

[17] Chang M., Ohkura K., Ueda K. and Sugiyama M., “Some Experimental Observation of (m, l)-ES with Linear Ranking Selection on Fertility”, *Proc. of the 6th International Conference on Complex System*, pp.339-346, 2002.

[18] Chang M., Sugiyama, M., Ohkura K. and Ueda K., “(m, l)-Linear Ranking Selection in Evolution Strategies”, *Proc. of the 4th Asia Pacific Conference on Simulated Evolution and Learning*, pp.236-240, 2002.

[19] Smith A.E. and Coit D.W., “Penalty Functions”, *Handbook on Evolutionary Computation*, pp.C5.2:1-6, Oxford University Press, 1997.

[20] Runarsson T.P. and Yao X., “Stochastic Ranking for Constrained Evolutionary Optimization”, *IEEE Transaction on Evolutionary Computation*, 4(3), pp.284-294, 2000.

[21] Koziel S. and Michalewize Z., “Evolution Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization”, *Evolutionary Computation,* 7(1) pp.19-44, 1999.

[22] Chang M., Ohkura K., Ueda K. and Sugiyama M., “A Two-Step Selection Scheme for Constrained Evolutionary Optimization”, Submitted to *IEEE International Conference on Neural Network & Signal Processing.*