関数最適化問題を解く熱力学的遺伝アルゴリズムの有効性の検証
～遺伝的アルゴリズムにおける多様性の測定と淘汰戦略への応用手法の研究～

Evaluation of Thermo-dynamical Genetic Algorithm to Solve the Function Optimization Problem
—Diversity Measurement and its Application to Selection Strategies in Genetic Algorithms—

髙橋 良英（京都情報大学院大学）
Ryouei Takahashi（The Kyoto College of Graduate Studies for Informatics）

Abstract

In this paper, it is experimentally verified that TDGA (Thermo Genetic Algorithms) is effective in solving a function optimization problem using Genetic Algorithms, because of their sustainability of population diversity and efficiency of searching for solutions. In our investigation, we measure the diversity of the population with the entropy defined in TDGA. We experimentally and quantitatively verify the hypothesis that we can improve the ratio of searching for the optimum solution and the accuracy of the solution by maintaining the diversity. TDGA is a selection strategy based on the minimum free energy principle in thermos-dynamics. In TDGA, we select individuals in the next population from those candidates generated temporarily for the next generation to satisfy the above principle. In applying the principle to GAs, we select them to make the average energy E a minimum and the entropy H a maximum. In this investigation, TDGA and immune-GA are compared with simple GA and GA with scaling windows from the viewpoint of sustainability of diversity of population. The effectiveness of these four GA models is verified quantitatively by diversity measurement. Furthermore, ten test functions are investigated in this report. They are well-known De Jong's five functions: F1, F2, F3, F4, and F5, and five well-known multimodal functions: Schaffer's F6 function, Rastringin's function, Schwefel's function, Griewank's function, and Ackley's function.

Keywords: Genetic Algorithm; Diversity Measurement; TDGA; Immune-GA; BLX-α; Function Optimization Problem

1. はじめに

生命の進化を模した確率的な解探索モデルである遺伝的アルゴリズム（Genetic Algorithm: GA）[1] は，組み合わせ最適化問題 [2] の解法として有効であることは良く知られている。遺伝的アルゴリズムの課題の１つは「比較的初期の世代で局所的に最適な解に陥ってしまい，そこから脱出できず大域的に最適な解を探索できなくなってしまう」という初期収束の問題を解決することである [3]。初期収束が起こる原因は集団の多様性が確保されていないことである [4]。すなわち，様々な遺伝子構造を持った染色体が集団を構成していないことである。このような状態では，選択される親の遺伝子構造パターンに限りがあることから，二点交叉（等）による単純な遺伝子交叉では，生成できる遺伝子構造パターンに限りが生じる。このため，解の改善が見られず，局所最適解に陥ってしまうことが多い。上記の問題を解決するためには，集団の多様性を確保しながらかつ解探索効率を落とさない解探索法が必要となる。

本検討では，関数最小値探索問題を遺伝的アルゴリズムで解く場合に [5], [6]，集団の多様性を確保しながら効率的探索を実現する方法として，TDGA (Thermo Dynamical Genetic Algorithms) による手法が有効であることを，immune-GA (immune Genetic Algorithms) による手法と比較しながら，

実験により検証する。本実験では，集団の多様性をTDGAで定義されたエントロピーで定量的に測定する。「集団の多様性を確保することにより，最適解探索率や探索解の精度を向上できる」という仮説を，実験により，定量的に検証する [7]。TDGAは熱力学における自由エネルギー最小化の原理に基づく個体選択法である [8]。そこでは，GAで生成した個体群の中から自由エネルギー F が最小となるように個体群を選択する。それは，エントロピー H が最大となるように, かつ平均エネルギー＜E＞が最小（目的関数値が最適）となるように個体群を選択することである。immune-GAはJerneの免疫システムの恒常性に関するイディオタイプネットワーク仮説に基づく最適化モデルである [9], [10]。そこでは，GAで生成した個体と抑制抗体間の染色体構造の類似度を調べ，類似度が高い場合に，その個体の生成を抑止する。抑制抗体群は，互いに類似度の異なる適応度の高い（目的関数値が最適値に近い），多様な染色体構造を持った個体群であり，GAがこれまで生成した個体群の中から逐次選択更新される。

一方，n 変数の関数最小値探索問題の場合，これまで, 遺伝子交叉演算に工夫をこらしたBLX (Blend Crossover)-α[11], [12] やSPX (Simplex Crossover) [13] が開発されている [14]。BLX-α やSPX法は，親として選択した β 個（BLX-α では β = 2，SPXでは β = n+1）の個体をn次元実数空間内の点（n 次元ベクトル）に変換し，変換した β 個の点を含む直方体内や（n+1）角形内で，ランダムに解候補となる個体を選択・生成する。この個体生成法は解の探索空間を拡げるので，生成個体の染色体構造の多様性を確保できると考えられる。その意味で，これ等の方法は遺伝子二点交叉等のGAによる広域的な探索法を補う交叉法として有効であると期待される。実際，広域探索に遺伝子二点交叉，局所探索にSPX法を応用したBee System [15] 等が提案され，その有効性が実験で検証されている。我々もBLX-α やSPX法が，単純GAやスケーリングウィンドウ技法を有するGA [16] において，更に精度の高い解が得られることを実験で検証できた。そこでは，ルーレット方式 [3] で選択した二個体に対して二点交叉法で生成した個体にBLX-α やSPX法を適用して変換した実数値の近傍を集中的に解探索させている。しかし，これ等の交叉演算のみでは集団の多様性を充分確保できないことを実験で確認した。

本検討では，TDGAによる手法をimmune-GAによる手法や単純GA手法やスケーリングウィンドウ技法と集団の多様性制御の観点から，比較評価した。そして，それ等の有効性を，多様性を測定することにより，定量的に評価している。スケーリングウィンドウ技法は個体の適応度をその個体と何世代か前の最悪個体（最大関数値を持つ個体）の関数値の差として計算する方法である。当技法では各世代の終了時に個体の適応度を再計算し，集団の多様性を確保している。尚，本検討のTDGAやimmune-GAは，スケーリングウィンドウ技法を有するGA上で動作する。

上記の4つの世代交代モデルではGAの二点交叉で個体を生成するが，生成した個体群の多様性が閾値以下になった時に，BLX-α を動作させ解探索の局所化･集中化を実現する方式も併せて検討した。BLX-α は親として選択した二個体の表現型を n 次元実数空間内の二点（n 次元ベクトル）に変換し，変換した2個の点を含む直方体内（探索空間の拡張率 α=0.5）で，ランダムに解候補となる点を選択･生成することで，生成する個体の染色体構造の多様性を確保する方法である。BLX-α による探索で多様性が閾値以上に向上した場合は，二点交叉法によるGAの広域探索法に戻す方式を検討した。

本実験でのGAのベンチマークテストで使用される関数はDe Jongの標準関数F1からF5と，良く知られた5つの多峰性関数F6からF10（Schaffer's F6 function, Rastringin's function, Schwefel's function, Griewank's function, Ackey's function）である [17]，[18]。

2. 前提条件

2-1 関数最適化問題

y を実数空間Rに値を取る従属変数, $x_i(i=1, 2,\cdots, n)$ を実数Rに値を取る n 個の独立変数とする関数 f は $y=f(x_1,..., x_n)$ と表現され，n 次元の実数の直積空間 R^n から実数空間Rへの写像を表す。関数 $y=f(x_1,..., x_n)$ の最適化問題とは，関数 f の従属変数 y の最適値 $y_{opt}=f_{opt}$（最小値 f_{min} または最大値 f_{max}）を実数空間R内で求めること，そして関数 f が最適値 f_{opt} をとった時の n 個の独立変数 x_i（i=1,2,・・・,n）の値 x_{i0} を n 個の実数の直積空間 R^n におけるある領域 R_0^n（定義域）内で求める問題である。De Jongは「遺伝的アルゴリズム（GA）のあるクラスに挙動に関する研究」というタイトルで博士論文を書いた。そこで

は，関数最小値探索問題をGAで解く手法を検討しているが，その中で使用されたテスト関数がDe Jongの標準関数F1からF5である。関数fの最大値探索問題に対する一般的解法が見つかれば，関数$y=f(x_1,\cdots,x_n)$の最小値探索問題は，関数fに－1を掛けた関数$z=g(x_1,\cdots,x_n)=-f(x_1,\cdots,x_n)$の最大値探索問題に問題を帰着できるので，その解を求めることができる。

2-2　関数最小値探索問題を遺伝的アルゴリズムで解く場合の個体の遺伝子型と表現型に関する前提

遺伝的アルゴリズムで関数$y=f(x_1,\cdots,x_n)$の最小値を求める時，解候補$(x_1,\cdots,x_n)$を個体Xとして設計する。個体Xは染色体と適応度で特徴づけられる。染色体は0か1かの値をとる二進固定小数点データ列として設計する。この二進固定小数点データ列は関数fのn個の独立変数x_iの値を表現する$(i=1,2,\cdots,n)$。複数変数の場合，二進固定小数点データ列は添え字の順番に変数分並べられる。この二進固定小数点データ列を遺伝子型と呼び，それが表現する各変数x_iの値を表現型と呼ぶ。二進固定小数点データの各1ビットを1つの遺伝子と解釈する。遺伝子列長Lは染色体長(chrom_size)と呼ばれ，定義域内における解の探索精度から決まる。x_iの定義域が$-a$から$+a$で，各変数の探索の精度をl個の遺伝子列で計算できる範囲とすれば，解探索の精度は$a/2^{l-1}$となる。すなわち，2^l個の離散個のデータ$x_i=-a+k\times a/2^{l-1}$ $(k=0,1,\cdots,2^l-1)$の中から関数最小値を満たすx_iの値を探索する。そして，染色体長Lは$L=n\times l$と計算できる。例えばDe Jongの標準関数F2の独立変数x_1, x_2の定義域は$-2.048\leqq x_1, x_2<2.048$である。ここで各変数の遺伝子列長$l$を$l=12$と設計すれば，解探索の精度は$2.048/2^{12-1}=1/1000$となる。そして各独立変数$x_i$に関して，$4096=2^{12}$個の離散個の値$x_i=-2.048+k\times 1/1000$ $(i=1,2;\ k=0,1,2,\cdots,4095)$の中から解を探索する。また$l=13$と設計すれば，解探索の精度は$I=12$の2倍となり1/2000となる。この時，各独立変数x_iの解は$8192=2^{13}$個の離散値$x_i=-2.048+k\times 1/2000$ $(i=1,2;\ k=0,1,2,\cdots,8191)$の中から探索する。

尚，BLX-αやSPXによる交叉演算では独立変数x_iの解の探索空間を二進固定小数点で表現される離散空間から連続値を取る実数空間に拡張して求めるが，求めた実数解x_{i0}は再び精度lの二進固定小数点データで近似される。そして，その近似値として求めた固定小数点データを交叉演算の結果生成した個体と解釈する。また，定義域を超えて解が求まった場合は，「x_{i0}をaで割った剰余$-a$」を解とする。ここで$-a$としたのは，実数空間を定義域の周期$[-a, a)$で分割した時の開始点だからである。例えば，精度$l=12$で求めたx_{i0}が$x_{i0}=3.126$で$a=2.048$の場合，x_{i0}は$x_{i0}=3.126\ \%\ 2.048-2.048=1.078-2.048=-0.97$と再計算される。

2-3　適用度設計

適応度はルーレット方式等で親となる個体を現世代の集団の中から確率的に選択する時に使用される。

その値が高くなればなる程，親として選択される確率は大きくなる。

2-3-1　スケーリングウィンドウを持たないGAの場合の適応度設計

GAがスケーリングウィンドウを持たない場合，適応度は個体の染色体構造に合わせ固定値を持つように一般的には設計される。すなわち個体Aと個体Bが同じ染色体構造を持つならば，世代を越えて，同じ適応度となるように設計する。関数最小値探索問題の場合，関数最小値探索問題の場合，本検討では，適応度Fを$F=1/(f+\alpha_1)$で計算する。α_1は$f+\alpha_1$が正（>0）となるような調整係数であり，プログラム定数として与える。例えば$f\geqq 0$なら$\alpha_1=1$として設計した。

2-3-2　スケーリングウィンドウ技法による適応度設計

ルーレット方式で適応度の大きさに応じて個体を選択する場合，個体Xの適応度$F(X)$は正でなくてはならない。本検討の事例でも見られるように一般にテスト関数fは正の値も負の値もとる。このため，関数fの最小値探索問題の解を求めるのに，関数fに－1を掛けて$g=-f(x_1,\cdots,x_n)$の最大値探索問題に変更して，その問題の解を求めようとしても，一般に「$-f$」は負の値をとるので，「$-f$」を個体の適応度とすることはできない。従ってW世代前の個体のうち「$-f$」が最悪値(最小値)となる個体$X_0=(x_{10},\cdots,x_{n0})$の関数値「$-f_w(x_{10},\cdots,x_{n0})$」と現世代の個体$X=(x_1,\cdots,x_n)$の関数値「$-f$」との差「$-f(x_1,\cdots,x_n)+f_w(x_{10},\cdots,x_{n0})$」を個体

Xの適応度$F(X)$とするスケーリングウィンドウ技法を検討することとした。そうすれば個体Xの適応度$F(X)$は正となる。尚，この定義によりスケーリングウィンドウ技法では同じ遺伝子構造でも世代が異なれば適応度が異なることがわかる。また淘汰圧が関数値の大きな個体にかかることを回避でき，集団の多様性を確保できる。

2-4　遺伝子交叉演算の拡張

二点遺伝子交叉等で生成した個体の中から多様性を考慮しつつ次世代の個体を選択するTDGAやimmune-GAによる解の探索手法は，遺伝的アルゴリズム(GA)の積み木仮説に基づく解探索手法を前提としている。Lを染色体長とすると，TDGAやimmune-GAは，2^L個の離散個の探索空間の中で，解を確率的に探索する。関数最小値探索問題に遺伝的アルゴリズムを適用して解を探索する場合に，関数の連続性を考慮して,関数の最小値を探索すれば,更に効率的な解探索が可能となる。本検討では，TDGAやimmune-GAによる解の探索法を補う手法として，ルーレット方式等で選択した二個体に対して，関数の連続性を考慮して，その二個体の近傍を解探索空間として解探索の局所化を図る遺伝子交叉演算オペレータとしてBLK-αを検討した。

BLX-αは親として選択した二個体の表現型をn次元実数空間内の二点に変換し，変換した二点を含む直方体内で，ランダムに解候補となる点を選択・生成する。αは各変数の探索空間の拡張率であり，本検討ではα=0.5とした。この探索空間の拡張により，生成する個体の染色体構造の多様性を確保できる。BLX-αによる方法の詳細は,以下の通りである。

関数$y=f(x_1,\cdot\cdot,x_i,\cdot\cdot,x_n)$の最小値探索問題において，親となる二個体として$X_1=(x_{11},\cdot\cdot,x_{1i},\cdot\cdot,x_{1n})$と$X_2=(x_{21},\cdot\cdot,x_{2i},\cdot\cdot,x_{2n})$が選択したとしよう。この時，解$(x_1,\cdot\cdot,x_i,\cdot\cdot,x_n)$を，以下で定義される閉区間$(S_1,\cdot\cdot,S_i,\cdot\cdot,S_n)$の中で，すなわち解$x_i \in Si$として,ランダムに探索する方法がBLX-$\alpha$である。

$$S_i=[\min(x_{1i},x_{2i})-\alpha\times|x_{1i}-x_{2i}|,\ \max(x_{1i},x_{2i})+\alpha\times|x_{1i}-x_{2i}|],\quad (i=1,2,\cdot\cdot\cdot,n) \qquad (1)$$

本検討では，集団の多様性が閾値β以下になった時に大域的探索法である二点交叉法から局所探索法であるBLK-αに変更する遺伝子交叉オペレータ交代法(Changing Crossover Operators)[19]を検討した。ここでβはプログラム定数であり，本実験では$\beta=(0.5\sim0.9)\times L$とした。本方式では，BLK-$\alpha$による遺伝子交叉により集団の多様性が閾値$\beta$以上に回復した場合,二点交叉法に戻している。本遺伝子交叉オペレータ交代法は，二点交叉法により大域的探索を実現し，BLK-αによりある点の近傍において集中的・局所的探索を実現している。

2-5　スケーリングウィンドウ技法

2-5-1　機能

スケーリングウィンドウ技法は同じ染色体構造を持った個体でも，その適応度を世代と共に変化させることによって，集団の多様性を確保する方式である。当技法は，ルーレット方式で適応度に応じて確率的に親となる個体を選択する際に,様々な遺伝子構造を持った個体が選択される確率を高くし，結果として生成される子供の遺伝子構造パターンを多くすることがそのねらいである。これによって局所最適解に陥ることを回避する。

2-5-2　スケーリングウィンドウ技法

スケーリングウィンドウ技法は，関数最小値探索問題の場合，W世代前の個体のうち関数fの値が最悪値(最大値)となる個体X_0の関数値$f_W(X_0)=f(X_0)$と現世代の個体Xの関数値$f(X)$との差を個体Xの適応度$F(X)$とする方法であり，以下の式で定式化される。

$$F(X)=f_W(X_0)-f(X) \qquad (2)$$

従って，スケーリングウィンドウ技法では，同じ構造を持った染色体でも，世代が異なると，異なった適応度となる。このスケーリングウィンドウ技法により「適応度$F(X)$を，染色体の構造のみによって決まる適応度固定法，例えば（関数値$f(X)+\alpha_1$）の逆数$F(X)=1/(f(X)+\alpha_1)$とする方法，を用いると関数値の低いある特定の個体のみが親として選択される傾向がある」という淘汰圧の課題を解決でき，GAで生成する個体群の多様性を確保できる。個体の適用度の更新は，世代交代の度に行う。スケーリングウィンドウサイズWはGAの起動パラメータで与える。

2-6　HeSEAにおける個体選択法HpSの応用

2-6-1　機能

HpS手法（Heterogeneous paring selection method）は，親Aと遺伝子交叉させる親Bを，その遺伝子構造の親Aとの類似度S_{AB}が集団Pの親A

との平均の類似度 S_A より低い個体群の中から確率的に選択することで，二点交叉等の遺伝子交叉によって生成される個体群の多様性を間接的に確保する方式である。

2-6-2　背景

HpS 手法は，遺伝子交叉の際，親 A と親 B を現世代の個体群の中から適応度の大きさに応じて確率的に選択するルーレット方式による親の選択法を修正する方法として提案された。HpS 手法は，親 A とペアとなる親 B を，親 A と染色体の構造が似ていない個体群の中から選択することで遺伝子交叉によって生成される個体群の多様性を確保することをねらいとしている。HpS 手法はもともと巡回セールスマン問題 (Traveling Salesman Problem: TSP) を解く HeSEA (Heterogeneous Evolutionary Selection Evolutionary Algorithm) [4] で開発された。本検討の HpS 手法は，それを関数最大値探索向けに修正した手法である。

2-6-3　類似度の定義

(A) 個体 A と個体 X の類似度 S_{AX}

個体 A と個体 X の類似度 S_{AX} を，個体 A と個体 X で同じ j 番目の遺伝子座にある遺伝子の値 A_j と X_j が一致する（$A_j = X_j$）となる遺伝子数 α_{AX} の染色体長 L に対する割合 $r = \alpha_{AX}/L$ として定義する (j=1, 2,･･･, L)。例えば A=(0,0,1,0,1,0,0,1), X=(1,0,0,1,0,0,0,0) とすると，染色体長 L は 8 となる。2 番目，6 番目，7 番目の遺伝子座の対立遺伝子の値は A と X で一致しているので α_{AX} = 3 となる。この時 r は r= 3/8=0.375 と計算される。S_{AX}=S_{XA} である。

(B) 集団 P の個体 A との類似度 S_A

集団 P が N 個体から成る時，集団 P の個体 A との類似度 S_A を，個体 A を除いた集団 P の全ての個体 I_j の個体 A との類似度 S_{AI_j} の平均値とする。すなわち $S_A = \sum_{j=1, A \neq I_j}^{N} S_{AI_j} / (N-1)$ として定義する。

2-6-4　HpS のアルゴリズム

(A) ステップ 1・・親 A を，ルーレット方式で確率的に選択する。

(B) ステップ 2・・親 A の交叉対象となる個体 B を，その類似度 S_{AB} が集団 P の個体 A との類似度 S_A より小さな個体，すなわち，$S_{AB} \leqq S_A$ を満たす個体 B をランダムに選択する。

3. 最小値探索問題を解く GA の基本的構造

2 節で述べた $y=f(x_1,・・・,x_n)$ の最適化問題（最小値探索問題）を解く基本 GA の構造は以下の通りである。TDGA と immune-GA はスケーリングウィンドウ技法を具備した本 GA 上で構築している。

3-1　初期化処理

集団は個体群から成る。各個体は染色体と適応度で特徴づけられる。

(1) 個体（染色体）群を集団サイズ分初期生成する。

一様乱数を用いて染色体を構成する遺伝子の値をランダムに生成する。遺伝子の値は 0 か 1 である。0 と 1 は対立遺伝子である。

(2) 生成した個体の適応度 F の計算

（ｉ）個体の遺伝子型から表現型への変換と表現型が表す関数値を求める。

染色体を構成する遺伝子列に解釈を与える。関数 $y=f$（$x_1,x_2,$・・・,x_n）の最小値探索問題の場合は独立変数 x_i の二進固定小数点データと解釈し，各個体を表現する染色体の具体的な遺伝子列の値 c_j (j=1,2,⋯,L) から関数 f の値を計算して求める。ここで L は 2.2 節で定義した遺伝子列長である。これ以降，個体 X =（$x_1,x_2,$・・・,x_n）とその染色体 $\bar{x} = (c_1, c_2, \cdots, c_L)$ は必要のない限り区別をつけないこととする。

適応度計算法はスケーリングウィンドウ技法の場合とそうでない場合で異なる。

（ii）適応度 F の計算

<u>スケーリングウィンドウ技法の場合</u>

(a) 生成した個体の中で最悪関数値（関数 f の最大値 $f_{max}(0)=f(X_{max}(0))$）を値に持つ個体 $X_{\max}(0) = (x_{\max.1}(0), x_{\max.2}(0), \cdots, x_{\max.n}(0))$ を探索する。ここで $f_{max}(0)$ 中の 0 や $X_{max}(0)$ 中の 0 は現世代 t が初期世代を示す（t=0）。関数最悪値管理テーブル T にスケーリングウィンドウサイズ W 世代数分，最悪値 $f_{max}(0)$ とその値を実現する個体 $X_{max}(0)$ の染色体をコピーする。

(b) W 世代前の世代の中で探索した解の中での最悪値 $f_w(X_{max}) = f_{max}(0 - W)=f(X_{max}(0 - W))$ と現世代の個体 X_i の関数値 $f(X_i)$ との差 $f_w(X_{max}) - C \times f(X_i)$ を個体 X_i の適応度 $F(X_i)$ として計算する。C は，適応度 $F(X_i)$ ＞ 0 となるための調整係数でありプログラム定数として与える。調整係数 C はテスト関数により異なる。

スケーリングウィンドウを使わない場合

個体 X の適応度 $F(X)$ は関数値 $f(X)$ によって決まり，世代を越えて固定となる。

次式で $F(X)$ を定義する。

$$F(X)= 1/（f(X) + \alpha_1）>0$$

ここで，α_1 は調整係数で固定値である。

(3) 世代毎の統計情報出力

生成した個体群の関数値の平均，現世代で探索した個体のうち最良適応度を持つ個体の関数値（世代内最小関数値），これまでの世代で探索した個体の内，最良適応度を持つ個体の関数値（世代間最小関数値），計算時間，集団の多様性を計算して出力する。

3-2　終了条件判定

以下のいずれかの条件を満たした時に，ジョブを終了する。

（ｉ）最大観測世代数（M）に達した場合。この数値 M はプログラム定数であり，実験では 10,000 とした。

（ii）探索解の改良がある世代数（G）間，見られない場合。この数値 G はプログラム定数であり，実験では 200 とした。

（iii）集団の多様性（D）が数世代に亘って変化しない場合。この数値 D はプログラム定数であり，実験では 5 とした。

3-3　次世代の個体の生成処理

集団サイズ分以下を繰り返す。

(1) 選択

two-point crossover または BLX-α の場合

親となる二個体 A と B をルーレット方式または HpS 法で現世代の個体の中から選択する。ルーレット方式では親 A と親 B を両方とも適応度の大きさに応じて確率的に選択する。HpS 法 [4] では親 A をルーレット方式で選択する。親 B は，親 A との類似度 S_{AB} が集団 P の親 A との類似度 S_A より小さな個体の中からランダムに選択する。類似度については 2-6-3 節で定義している。

(2) 遺伝子交叉

選択した親に対して確率 Pc で遺伝子交叉を行ない，子供を二個体生成する。遺伝子交叉オペレータとして以下の二種類のオペレータを集団の多様性を観測して自動的に切り替える。

（ｉ）広域探索法：k 点交叉，SPX

k 点交叉による方法。本実験では k=2 である。または SPX の遺伝子交叉演算である。SPX では関数が n 個の独立変数を持つ時，①にて親として選択した (n+1) 個の点の中から (δ+1) 個の点をランダムに選択可能とし，選択した (δ+1) 個の点を含む拡大率 R の拡張 (δ+1) 角形の領域内で次世代の個体をランダムに選択する。この δ はプログラム起動パラメータとして与える。δ は，2 ～ n の値をとり得る。実験では δ=2 または n とした。

（ii）局所探索法：BLK-α

集団の多様性が閾値 β 以下になった時，k 点交叉または SPX から BLK-α の遺伝子交叉演算に交替する。BLK-α による遺伝子交叉演算の結果，集団の多様性が閾値 β 以上に回復したら二点交叉または SPX の遺伝子交叉演算に戻す。

(3) 突然変異

生成した二個体のある遺伝子の on/off の値を確率 P_m で反転させる。反転する遺伝子座はランダムに決定する。

(4) 淘汰

生成した子二個体と親の二個体，合わせて四個体の中から次世代の個体を 1 つ選択する。先ず，四個体の関数値を計算する。以下の二つの場合に分けて個体を選択する。

親と子のトーナメントを行う場合

上記四個体の中で最も関数値の小さな個体を次世代の個体として選択する。

親と子のトーナメントを行なわない場合

生成した二個体のうち関数値の小さな個体を次世代の個体として選択する。

3-4　世代交代処理（t − 1 世代→ t 世代）

(1) 3-3 で成した次世代（t 世代）候補個体で現世代（t − 1 世代）の個体を generation-gap 分入れ替える。その際，古い個体の方から先に，新しい個体に入れ替える。

(2) 3-3 で生成した個体の適応度を求める。

（ｉ）各個体を実現する染色体の具体的な遺伝子列の値から関数 f の値を計算して求める。

スケーリングウィンドウ技法の場合

（ii）最悪値管理テーブル T において，k − 1 世代前の最悪個体とその関数値を k 世代前の最悪個体とその関数値で入れ替える（k= W, W − 1, …, 2, 1)。現世代において最悪関数値（関数 f の最大値 $f_{max}(0)$）を持つ個体 β を探索する。最悪値管理テーブル T において現世代 (k=0)

の最悪個体 $T(0)$ を個体 β で入れ替える。

（iii）W をスケーリングウィンドウサイズとする。この時，W 世代前の世代の中で探索した解の中での最悪値 $f_{max}(t\text{-}W)$ と個体 X_i の関数値 $f(X_i)$ との差 $f_{max}(t - W) - C \times f(X_i)$ を個体 X_i の適応度 $F(X_i)$ として計算する。この時，全ての個体 X_i について適応度 $F(X_i) > 0$ となるように調整係数 C の値を決める。調整係数 C はプログラム定数として与える。

適応度固定方式の場合

個体 X_i の適応度 $F(X_i)$ は世代をこえて固定値であり，$F(X_i)=1/(f(X_i) + \alpha_1)$ で計算する。$F(X_i) > 0$ となるように調整係数 α_1 の値を決める。調整係数 α_1 はプログラム定数として与える。

(3) 集団の評価と統計情報出力

生成した個体群の関数値の平均，現世代で探索した個体のうち最良適応度を持つ個体の関数値（世代内最小関数値），これまでの世代で探索した個体の内，最良適応度を持つ個体の関数値（世代間最小関数値），計算時間，集団の多様性を計算して出力する。

(4) 3-2 の処理に戻る。

4. 集団 P の多様性を測る尺度 H

本研究における集団 P の多様性を測る尺度 H は TDGA のエントロピーに由来する。TDGA を利用して組み合わせ最適化問題を解く時，エントロピー H は問題対応に異なった解釈を与えなくてはならない。これまで，巡回セールスマン問題 (TSP) を解く GA においてエントロピー H の解釈事例が紹介されている。この事例を参考にして，関数最小値探索問題を GA で解く場合，集団 P のエントロピー H を，以下の式で定義した。この式は，染色体を構成する第 j 番目の遺伝子座にある遺伝子の値の確率分布から計測されるシャノンのエントロピー情報量 H_j の総和としてエントロピー H を定義することを示している。

$$H = \sum_{j=1}^{chrom_size} H_j \qquad (3)$$

$$H_j = -\sum_{i=0}^{1} p_{ij} \times \log_2 p_{ij} \qquad (4)$$

$$p_{0j} = \frac{N_{0j}}{N}, p_{1j} = \frac{N_{1j}}{N} \qquad (5)$$

上記にて，$chrom_size$ は染色体長 L のことであり。N は集団 P の集団サイズ（個体数）であり，N_0^j は集団 P の中で第 j 番目の遺伝子の値が 0 となる個体の数であり，N_1^j は集団 P の中で第 j 番目の遺伝子の値が 1 となる個体の数である。従って，$N_0^j + N_1^j = N$ となる。p_0^j は集団 P の中で第 j 番目の遺伝子の値が 0 となる確率であり，p_1^j は集団 P の中で第 j 番目の遺伝子の値が 1 となる確率である。従って，$p_0^j + p_1^j = 1$ となる。エントロピー H_j は，確率分布 $\{p_0^j, p_1^j\}$ から決定されるシャノンのエントロピーである。

5. 力学的遺伝アルゴリズム（TDGA）による関数最小値探索問題の解法の性能改善

5-1 ねらい

遺伝的アルゴリズム（GA）の早期収束の問題を解決するためには，染色体構造の異なる個体がバランス良く集団内に混在している必要がある。集団の多様性を維持しながら最適解探索を効率的に続行する有効な手法として，熱力学的遺伝アルゴリズム（TDGA）が開発された。TDGA は，各世代で GA が一括生成した個体群の中から，自由エネルギー最小化の原理を満たすように個体を逐次取捨選択する手法である。

5-2 熱力学的遺伝アルゴリズム（TDGA）の考え方

5-2-1 自由エネルギー F の最小化

TDGA は自由エネルギー最小化の原理に基づく個体選択方式である。そこでは，集団の多様性をエントロピー H として明確に測定し，自由エネルギー F を最小とするように個体を選択する。自由エネルギー F は，集団の平均エネルギー $\langle E \rangle$ を最小化しかつエントロピー H を最大化するための尺度であり以下の式で定義される。エントロピー H は集団の多様性を測る尺度である。

$$F = \langle \mathrm{E} \rangle - T \times H \qquad (6)$$

T はシミュレーテッドアニーリング手法における温度のことであり，調整係数である。

5-2-2 温度パラメータ T の調整法

温度パラメータ T はシミュレーテッドアニーリング機能における温度のことであり，集団 P が熱平衡状態に達する度に，高温から低温に徐々に温度を下げて，解探索を収斂させる働きを持つ。その際，温度の減少率を r とする。すなわち，t 世代目の温

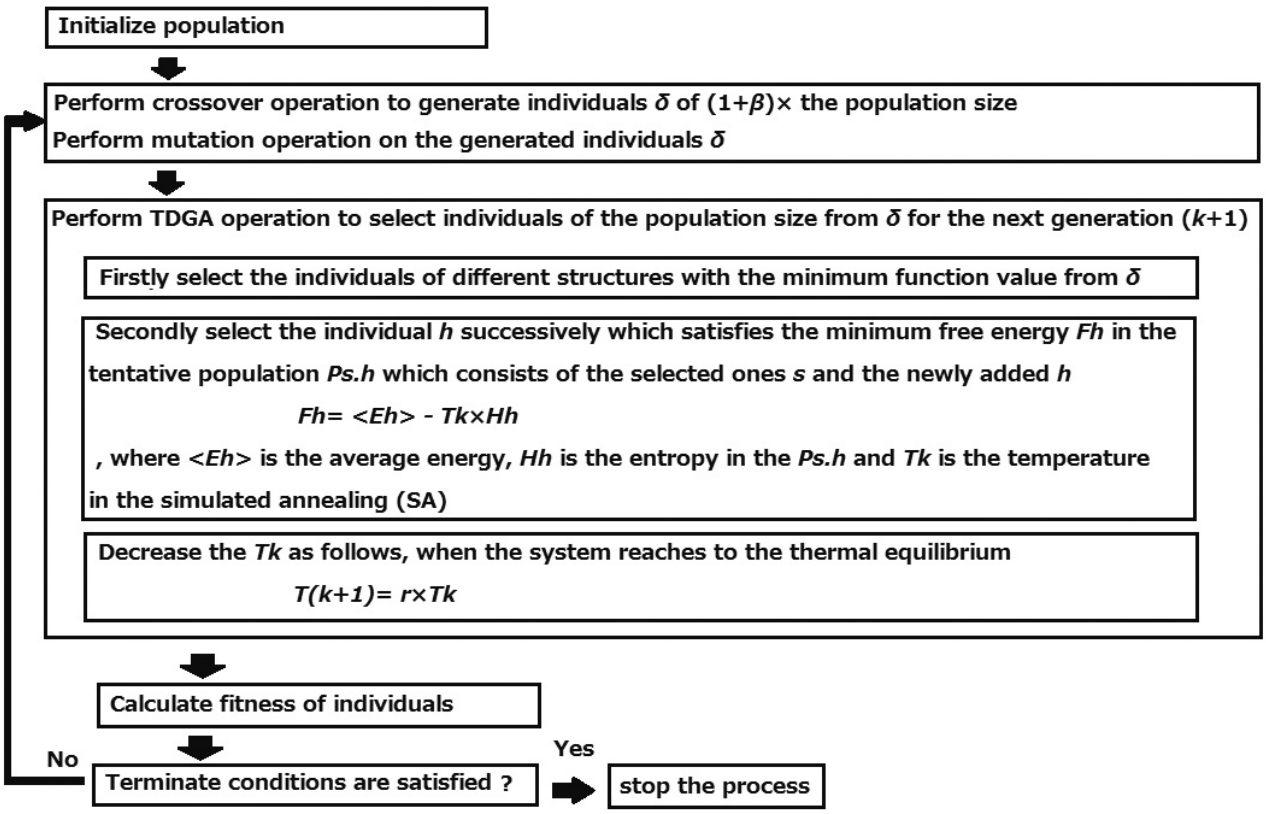

Fig.1. Implementation of TDGA to solve the function optimization problem.

度 T を $T = T_t$，平均エネルギー＜E＞を＜E＞＝E_t，エントロピー H を $H=H_t$，これまでの最小エネルギーを E_{min}，最小エントロピーを H_{min} と表した時，$E_t \geqq E_{min}$, $H_t \geqq H_{min}$ を満たしたならばシステムは熱平衡状態に達したとして，t+1 時の温度 T を $T = T_{t+1} = r \times T_t$ と減少させる。本実験では，温度の減少率 r = 0.9 ～ 0.999 とした。

5-3　TDGA の遺伝的アルゴリズムの手続きの中での位置づけ

5-3-1　TDGA のアルゴリズムを組み込む対象となる遺伝的アルゴリズム

本検討では，スケーリングウィンドウ方式（Grefenstette のメタ GA で検討された GA パラメータ）で適応度を動的に管理する GA 上で TDGA を実現している。

5-3-2　TDGA のアルゴリズム（図 1（Fig.1）参照のこと）

TDGA では，遺伝的アルゴリズム（GA）で生成した次世代の個体群候補 X の中から関数値が小さくかつ集団の多様性を最大とするような個体群 Y を次世代の個体群として選択する。その選択率を α=｜Y｜/｜X｜とする。α は GA の起動パラメータであり，実験では 1/2 または 1/3 とした。そこでは，先ず，$Y=\phi$（空集合）として，X 中で適合度の最も高い（関数最小化問題の場合は関数値の最も小さな）個体を Y に追加することから始める。次に，追加することにより自由エネルギー F が最小となるような個体を X から選択し，逐次 Y に追加する。

＜注意＞

①図 1 にて $TDCONST=(1+\beta)=1/\alpha$ である。

②オリジナルの TDGA では，各世代の開始時，現世代の個体群の中から重複個体を取り除き，取り除いた個体群を対象として TDGA を適用している。しかし，我々の実験では，この関数最小値探索問題の場合，局所解への収束速度が速くなりすぎた。このため，本検討では，重複個体の取り除きを行わないこととした。従って，TDGA で生成する個体の数（集団サイズ× TDCONST）は世代を越えて一定とした。

③ TDGA の最初のステップにて適応度の最も高い個体を次世代の個体として選択するが，その際，その最大適応度個体と同じ適応度を持つ個体で異なる染色体構造を持つ個体を次世代の個体として選択する。

④本検討では，以下の温度制御機能を，オリジナルの TDGA 機能に追加している。

「集団が以下の条件を満たした時，多様性が損なわれたとして，温度を $T(t+1) = r_1 \times T(t)$ と上げる。その時の温度係数が r_1 である。本実験では r_1=1.001 とした。

($new_entropy\ H_t < minimum_entropy$)

new_entropy：新しい世代 t のエントロピー H_t，*minimum_entropy*：これまでの世代で求めたエントロピーの最小値 H_{min}」

5-4　関数最小値探索問題への TDGA のカスタマイズ

TDGA で定義された自由エネルギー F を求めるために必要な平均エネルギー＜E＞とエントロピー H の解釈は組み合わせ最適化問題によって異なる。関数最小値探索問題はそれぞれを以下のように解釈する。

5-4-1　平均エネルギー＜E＞

本検討では，標準関数 f の最小値探索問題を，関数 f に－1を掛け，関数 $g=\beta - f$ の最大値探索問題として問題を変換して解く。β は定数である。関数 g の値が大きければ大きいほど適応度は高く設定しなければならない。一方，自由エネルギー最小化の原理で解を求める場合，平均エネルギー＜E＞は，適応度が高くなればなる程小さくなるように設計されなければならない。従って関数値 $f(\bar{x})$ を各個体のエネルギーとして定義することとした。

そして集団 P を構成する個体 の関数値 $f(\bar{x})$ の平均値を平均エネルギー＜E＞と定義し，それは以下の式で定式化される。ここで, i=1, 2,⋯,N であり，N は集団サイズである。

$$< E > = (f(\bar{x}_1) + f(\bar{x}_2) \ldots + f(\bar{x}_N))/N \qquad (7)$$

5-4-2　エントロピー H

関数最小値探索問題を GA で解く場合のエントロピー H の解釈は第 4 節で述べたので割愛する。

6. 免疫遺伝アルゴリズム（immune-GA）による関数最小値探索問題を解く GA の性能改善

6-1　ねらい

遺伝的アルゴリズム（GA）で生成した個体を，次世代の個体として選択するか否かを決めることは集団の収束性と多様性に大きな影響を与える。単に，二個体のうち優秀な個体のみ残す（間引き），親の方が子より適応度が高ければ，子を捨てて親を残す等の淘汰処理では，集団の多様性を考慮していないため，局所最適解に陥ることがある。集団の多様性の維持を考慮して，次世代の個体選択に工夫をこらす方法が免疫遺伝アルゴリズム（immune-GA）である。Immune-GA は Jerne の提案したイディオタイプネットワークによる免疫システムの恒常性に関する仮説に基づく最適化モデルである。

6-2　人工的免疫システム（Artificial Immune System）

人工的免疫システムは Jerne の提案したイディオタイプネットワークに基づく情報処理システムのことである。イディオタイプネットワークとは「免疫システムでは抗原に対する免疫抗体が生成され，抗原をその抗体が排除している。免疫抗体は他の免疫抗体の抗原として働く。これによってある特定の抗体が抗体集団を占めることを回避し，様々な抗原に対して柔軟に対処可能な免疫システムを実現している。」という免疫システムの恒常性に関する仮説である。人工的免疫システムは以下の機能から成る。

①抗原の認識機能，②抗体の再構成機能，③抗体の抑制機能，④排除に利用した抗体（抑制抗体）の記憶機能，⑤抗体による抗原の排除機能

6-3　人工的免疫システムの関数最小値探索問題を解く遺伝的アルゴリズム（GA）の中でのカスタマイズ

本研究では人工的免疫システムの機能（上述①～⑤）を以下のように関数最適値探索問題を解く GA の中で実現する。当 GA は TSP を解く immune-GA[7] の改良版である。そこでは，①抗原を関数最適値探索問題, 抗体をその最適解候補の独立変数（個体）と解釈する。② GA の選択・遺伝子交叉・突然変異機能により抗体を再構成する。本実験では GA の遺伝子交叉演算オペレータとしてテスト関数 F1 ～ F10 の最小値探索問題を解く遺伝子交叉演算オペレータ OPs を使用する。当オペレータは Two-point Crossover または BLX-α で二個体生成する。このオペレータ自身は関数に依存しない。関数に依存するのは適合度計算に関わる部分のみである。③集団の多様性を確保するため，遺伝子交叉オペレータで生成した抗体を排除する機能，④排除に使用した抑制抗体群（サプレッサ細胞）の更新管理を行う

機能。⑤最適解を探索するまで②③④を繰り返す。本人工的免疫システムをImmune Genetic Algorithm（Immune-GA）と呼ぶ。immune-GAでは，次世代の個体を一つ一つ追加する際に，集団の多様性をチェックし，それを維持するように，適応度の高い個体を追加・選択する。

6-4　抗体 v と抗体 w の類似度（affinity）$\alpha_{v,w}$ の定義

生成した抗体のうち，これまで生成した抗体との類似度が高い抗体を排除するのがimmune-GAである。

抗体 v と抗体 w の類似度（affinity）$\alpha_{v,w}$ は同じ遺伝子座にある抗体 v と抗体 w の遺伝子の値が一致している遺伝子数 η の全遺伝子数（染色体長）L に対する割合 η/L として定義する。例えば v の遺伝子型が (00010111)，w の遺伝子型が (10101100) ならば，染色体長 L=8 であり，v と w で遺伝子の値が一致している遺伝子座は 2 番目の '0' と 6 番目の '1' の 2 つであり η = 2 となる。この時，類似度 $\alpha_{v,w}$ = 2/8=0.25 となる。この定義は TSP を解く HpS[4] と同じである。

6-6　免疫 operation の詳細

6-6-1　初期サプレッサ細胞の生成

初期抗体を GA の染色体として生成後に以下を行う。

①抗体 v の抗原に対する関数値 $g(v)$ ($g(v) + \beta$ (β>0) が GA の適応度 *fitness* 相当となる）を計算する。

②サプレッサ細胞を形成する。

- 抗体 v と抗体 w の類似度（affinity）$\alpha_{v,w}$ を計算する
- 抗体 v と類似する抗体の集団に占める割合 C_v を計算する。v と w との類似度 $\alpha_{v,w}$ が閾値 *TAC1* 以上の個体 w の全個体中に占める割合（数の比）を C_v とする。
- *Cv* がある閾値 *TC* 以上になれば，それを記憶し，サプレッサ細胞とする。
- 尚，サプレッサ細胞のサイズは最大人口サイズとする。
- 同一構造を持った抗体がすでにサプレッサ細胞にある時，その抗体をサスプレッサ細胞に追加しない。

＜留意点＞

初期抗体を GA で生成する方法としては，染色体の各遺伝子の 0/1 の値を一様乱数によりランダムに決定する方法と他のジョブで生成した染色体ファイルを引き継ぐ方法がある。前者は遺伝子の値として 0 か 1 の値が等確率で決定される。後者の場合は解の収束がある程度進んだ染色体集合を引き継ぐことを仮定しているため，染色体構造の分布に偏りが見られる。このため，本手続き 6-6-1 でサプレッサ細胞が生成される確率が高いのは後者の場合であり，前者の場合はその確率が低い。

6-6-2　immune operation の実行

GA の遺伝子交叉操作により個体を生成し，生成した個体に突然変異操作を施す度に，生成した個体を次世代の個体（抗体）候補として残すか否かを，以下の immune-operation により判断する。これにより，集団の多様性を確保しつつかつ，最適解への収束速度を向上させる。

(a) 抗体 v の抗原に対する効用を関数値 g（v）（GA の適応度（*fitness*）相当値）として計算する。

(b) サプレッサ細胞の更新処理・・関数値の高い構造の異なる抗体をサプレッサ細胞として記憶する。

- サプレッサ細胞 w より適応度 (関数値 $g(v)$) が高い，または適応度が同じでも類似度のみが異なる抗体 v は，サプレッサ細胞に追加更新する。
- 過去のサプレッサ細胞に今期のサプレッサ細胞を追加することにより人口サイズ以上のサプレッサ細胞になった場合は，適応度の低い過去のサプレッサ細胞を削除する。

(c) 抑制処理（類似抗体の排除）・・多様性を確保するために，既に生成された抗体と類似する抗体の生成を抑制する。

以下の 3 通りの場合を考える。

(場合 1) 個体（抗体）v がサプレッサ細胞群 S に属するあるサプレッサ細胞 w より適応度が低い（関数値 $g(v)$ が大きい）場合，以下の 2 つの場合を考える。

(場合 1-1) あるサプレッサ細胞 w との類似度 $\alpha_{v,w}$ が *TC1* 以上の個体（抗体）v の生成を抑止する。

(場合 1-2) 全てのサプレッサ細胞 w との類似度 $\alpha_{v,w}$ が *TC1* 以下の個体（抗体）v は次世代の個体候補として残す。

(場合 2) 生成された抗体 v があるサプレッサ細胞 w と同一構造個体の場合で，かつ次世代の個体群候補の 1 つとしてまだ暫定プール中に記憶されていない場合，次世代の個体群候補として残す。

(場合 3) 生成された抗体 v があるサプレッサ細胞 w

と同一構造個体の場合で，かつ次世代の個体群候補の１つとして既に暫定プール中に記憶されていた場合，以下の３つの場合を考える。尚，以下に述べる閾値 *THRESHOLD* は GA 起動パラメータとして与える。

(場合 3-1) サプレッサ細胞 S の平均関数値が閾値 *THRESHOLD* ×（これまでの最良解の関数値 (*best-function value*)）以下の場合は，次世代の個体群候補として残す。世代が早期の段階では平均適応度が低いので，重複して生成された個体でも残存確率を高くし，解への収束を早める。世代が早期の段階は収束性重視となる。

(場合 3-2) サプレッサ細胞 S の平均関数値が閾値 *THRESHOLD* ×（これまでの最良解の関数値 (*best-function value*)）以上の場合は，その生成を抑止し，次世代の個体群候補から削除する。世代が後期の段階は平均適応度が高いので，重複した個体の残存確率を低くし，集団の多様性を重視する。世代の後期の段階は多様性重視となる。

＜注意事項＞

6-4 節で述べたように，関数の平均値（Avg(g)）やこれまでの最良解の関数値 (*best-function value so far*）は一般に負の値をとるため，Avg(g) が *best-function value* の閾値 *TH* 倍以上か否かで集団の収束が始まったことを判断するためには *TH* の値を１以上にする必要がある。

(d) 淘汰 (selection)・・適応度が低く，かつ互いに類似度の高い抗体が生成されることを回避して，解探索速度を向上させる。

当プロセスではプロセス (c) で生成を抑制されなかった抗体に対して，次期世代の生存期待値 e_v を計算し，e_v が閾値（サプレッサ細胞 S の平均関数値 Avg（S）の *TC2* 倍）以下となれば，その抗体の生成を抑止し，次世代の個体群候補から削除する。抗体 v の生存の期待値 e_v は次のように設計する。

抗体 v との類似度が大きい個体が多ければ多い程，抗体 v の生存の期待値 e_v は小さくなるように設計する。また，抗体 v の関数値 $g(v)$ が高い程抗体 v の生存の期待値 e_v は大きくなるように設計する。また，関数値 $g(v)$ が正の値をとる場合は $g(v) - \delta$ が負の値を取るように調整係数（$\delta > 0$）を選択する。この調整係数 δ はプログラム定数として与える。そして $g(v) - \delta$ を改めて関数 $g(v)$ とおき，関数値 $g(v)$ が負となるようになるようにテスト関数の適応度設計等に工夫がされていることを仮定する。これ等のことを考慮して，抗体 v の生存の期待値 e_v を以下の式で定義する。

$$e_v = g(v) \times \prod_{i=1}^{S_0} (1/\alpha_{v,w_i}) \qquad (8)$$

上式は，e_v を抗体 v との類似度 $\alpha_{v,wi}$ が閾値 *TAC2* 以上のサプレッサ細胞 w_i について，その類似度 $\alpha_{v,wi}$ を関数値 $g(v)$ に掛け合わせた数値として定義することを示している。S_0 は抗体 v との類似度 $\alpha_{v,wi}$ が閾値 *TAC2* 以上のサプレッサ細胞 w_i の数である。

＜注意事項＞

$g(v)$ が負のため，e_v とサプレッサセルの平均 Avg（S）は負となる。$\alpha_{v,w}$ は 0 から 1 の値をとる。従って，e_v が Avg（S）の *TC2* 倍より低い時に抗体 v の生成を抑制するためには，v と w の類似度の逆数 $1/\alpha_{v,wi}$ を $f(v)$ に掛け合わせてそれを e_v の定義としなくてはならないことに注意する。同様に *TC2* は 1 以上でなくてはならない。

7. 実験とその結果

7-1 テスト関数

テスト関数は F1 から F10 の 10 個の関数である。詳細は 7-6 と付録を参照のこと。

7-2 実験環境

数値実験は，主に，Dell Precision T3500, Intel(R) Xeron (R) CPU W3680, 3.33 GHz, 6GB RAM, Windows 10 上で試験を行なった。システム開発言語は C 言語である。C 言語の処理系は Microsoft Visual C++ 2010 デベロプメントシステムである。

7-3 評価対象 GA

以下の 4 つの GA モデルを，C プログラムで実現した。そして，最適解の探索効率と解の探索精度と多様性維持機能を比較評価した。

(1) Non-Scaling GA(Non-SW)・・スケーリングウィンドウ技法を持たない単純 GA

(2) Scaling Window GA(SW-GA)・・スケーリングウィンドウ技法を有する GA

(3) Immuune-GA with scaling window・・スケーリングウィンドウ技法を有する Immune-GA

(4) TDGA with scaling window・・スケーリングウィンドウを有する TDGA

Table 1. Results of evaluation of TDGA, immune-GA, Scaling Window GA and non-Scaling GA through ten test functions.

Test Function	GA models	Ratio of optimum trials	Relative error	Best minimum value	Ave. gen.	Ave. initial div.	Ave. best div.	Ave. final div.
F1	Non-SW*	15/15	0	0	13	29.93	27.24	0
	SW-GA**	15/15	0	0	15	29.93	26.87	0
	Immune-GA	15/15	0	0	16	29.93	26.54	0
	TDGA	15/15	0	0	13	29.79	27.83	23.93
F2	Non-SW	13/15	0.000000033334167	0	23	25.93	9.62	0
	SW-GA	12/15	0.000009016607500	0	22	25.93	10.79	0
	Immune-GA	13/15	0.000002087760417	0	27	25.93	12.26	0
	TDGA	13/15	0.000273394880000	0	40	25.79	17.9	9.42
F3	Non-SW	15/15	0	- 30	19	49.88	26.77	4.18
	SW-GA	15/15	0	- 30	18	49.88	25.38	1.18
	Immune-GA	15/15	0	- 30	21	49.88	28.79	3.94
	TDGA	15/15	0	- 30	22	49.62	19.97	2.97
F4	Non-SW	2/15	0.020768388371756	0.007090033793755	608	239.42	127.85	125.22
	SW-GA	2/15	0.014790146973541	0.006202866585284	532	239.42	125	126.56
	Immune-GA	2/15	0.019095658949756	0.007374419566332	600	239.42	126.57	126.22
	TDGA	4/15	0.013520268602822	0.006378799132664	666	238.24	220.02	181.75
F5	Non-SW	15/15	0.000000000000575	0.998003837795203	17	33.92	13.73	0
	SW-GA	15/15	0.000000000000575	0.998003837795203	17	33.92	13.19	0
	Immune-GA	15/15	0.000000000942441	0.998003837794628	13	33.92	18.2	0.08
	TDGA	15/15	0.000000000000575	0.998003837795203	77	33.73	17.21	4.87
F6	Non-SW	15/15	0	0	7	15.96	12.86	0
	SW-GA	15/15	0	0	7	15.96	11.94	0
	Immune-GA	15/15	0	0	7	15.96	13.84	0
	TDGA	15/15	0	0	6	15.88	15.31	4.86
F7	Non-SW	0/15	11.75761412915560	5.969921589175300	156	279.34	18.1	0
	SW-GA	0/15	9.967152137000000	4.975318872000000	149	279.34	20.1	0
	Immune-GA	0/15	8.159139807873520	3.980137279018880	165	279.34	27.46	0.48
	TDGA	12/15	1.525963631000000	0.000077496800000	359	277.95	85.52	13.72
F8	Non-SW	15/15	0.006237505352	-4189.822635218985	43	99.77	18.47	0
	SW-GA	15/15	0.006237505352	-4189.822635218985	45	99.77	16.18	0
	Immune-GA	15/15	0.006237505352	-4189.822635218985	45	99.77	19.79	0
	TDGA	15/15	0.006237505352	-4189.822635218985	132	99.31	64.44	49.23
F9	Non-SW	7/15	0.115743947894937	0.032507253194352	61	109.73	26.72	0
	SW-GA	6/15	0.111482000275566	0.032507253194352	57	109.73	31.64	0
	Immune-GA	8/15	0.143519178881590	0.046797078163841	67	109.73	21.44	0
	TDGA	14/15	0.063024064593111	0.032507253194352	99	109.2	63.27	12.48
F10	Non-SW	13/15	0.163432156222627	0	27	59.86	36.13	0
	SW-GA	14/15	0.081716078111313	0	25	59.86	37.41	0
	Immune-GA	15/15	0	0	30	59.86	43.54	0
	TDGA	15/15	0	0	32	59.59	50.99	1.9

7-4　GA 起動パラメータの組み合わせ

上記の 4 つの GA モデルの各々について，以下の GA 起動パラメータの組み合わせ毎に，一様乱数の初期値 = 1 ～ 15 とした 15 回の独立な試験を行なった。一様乱数の初期値が異なれば，最適解候補となる各独立変数の初期値が異なり，試験の独立性を保証できる。

(1) 集団サイズ Table1 は⑤の場合のまとめである。
　① 30　② 60　③ 120　④ 240　⑤ 300

(2) Scaling windowsize
　Table1 は 7 の場合のまとめである。
　　2 か 3 か 7

(3) 親選択法　Table1 は②の場合のまとめである。
　① Roulette 方式　② HpS 方式

(4) 交叉演算法
① Two-point-crossover 法
② Two-point-crossover + BLX- α 法

Two-point-crossover（大域探索）から BLX-α（局所探索）への遺伝子交叉オペレータの交替は集団の多様性（TDGA のエントロピー H）が閾値 $\beta=L \times 0.9$，または $\beta = L \times 0.5$ 以下になった時とした。集団の多様性 H が閾値 β 以上に回復したら，再び BLX-α から Two-point-crossover に遺伝子交叉オペレータを交替する。ここで L は染色体長である。尚，α=0.5 である。

7-5　評価項目

(1) 最適解探索率 (*Ratio of optimum trials* (R_{opt}))：15 回の独立な試験の中で最適解（関数最小値）を探索できた回数 M の割合のこと。$R_{opt} = M/15$ で計算する。最適解（関数最小値）はテスト関数の形がわかり易い場合はすぐにその値がわかる。しかし，遺伝アルゴリズムで探索可能な解は，染色体長を L とすれば，2^L 個の離散値であり，探索の範囲が限られる。従って染色体長 L が短い場合は，探索の範囲に，理論的な最適解が含まれない場合がある。その場合は，染色体長 L の精度で見つけ得る解の中で最も関数の値が小さな値を最適解と呼ぶ。最適解とは，関数の従属変数の値と，それを実現する独立変数の値のことをいう。

(2) 相対誤差 (*Relative Error*(*ER*))：15 回の独立な試験で探索した関数最小値の平均値 − 最適最小値。最適最小値は 7-6 節で説明されるが，それは理論的または実験的に計算して求められた最小値である。

(3) 最良関数最小値 (*Best minimum value*)：最良モデルが見つけた関数の最小値。最良モデルとは，15 回の独立な各々の試験の中で見つけた関数の最小値の中で最も小さな最小値を見つけたモデルのこと。

(4) 平均世代番号 (*Average number of generations*(*Ave. gen.*))：15 回の独立な各々の試験において，関数の最小値を見つけるのに要した世代番号の平均値

(5) 平均計算時間 (単位：sec.)：15 回の独立な各々の試験において，関数の最小値を見つけるのに要したコンピュータ計算時間の平均値

(6) 多様性の平均初期値 (*Average initial diversity* (*Ave. initial div.*))：15 回の独立な各々の試験において，初期世代に生成した個体群の多様性の平均値

(7) 多様性の平均最良値 (*Average best diversity* (*Ave. best div.*: $=Div._{opt}$))：15 回の独立な各々の試験において，関数の最小値を見つけた世代開始時の個体群の多様性

(8) 多様性の平均最終値 (*Average final diversity* (*Ave. final div.*))：15 回の独立な各々の試験において，ジョブを停止した世代終了時の個体群の多様性

7-6　実験結果

実験結果を表 1 にまとめる。

表では，Two point Crossover + BLX-α（α=0.5）の場合の実験結果をまとめた。染色体長 L は多様性の平均初期値を四捨五入して得られることに注意せよ。何故なら染色体を構成する遺伝子の on/off は一様乱数で決定されるからである。例えばテスト関数 F1 の多様性の平均初期値は 29.93 であるが，小数点以下を四捨五入して 30 となる。この値は F1 の個体の染色体長 L=10 ビット × 3 に一致している。

以下に実験結果を要約する。

尚，テスト関数 f の特徴や観測精度の説明ならびにそのグラフを付録として追加した。本検討では，関数最小値探索問題は関数に − 1 を掛けて関数最大値探索問題に変換して問題を解いている。このため，各テスト関数 f に − 1 を掛けた関数のグラフ g= − f を以下に描いている。

(1) De Jong's test function F1: Parabola (sphere function) (unimodal: single peak) (see Fig. A.1)

＜定義式＞ $z = f(x_1, x_2, x_3) = \sum_{i=1}^{3} x_i^2$

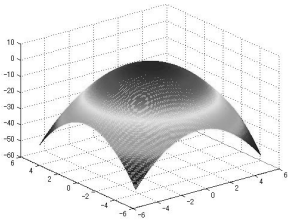

Fig. A.1　Inverted De Jong test function F1: parabola.

＜定義域＞－ 5.12 ≦ x_i ＜ 5.12 (i=1, 2, 3)
＜染色体長 L ＞ 30 (bit)=10 (bit) × 3 (変数)
＜最適最小値＞ x_i=0 (i=1, 2, 3) の時，F1 は最小値 0 をとる。
＜実験結果＞ 表 1 から 4 つの GA モデルの各々について最適解探索率 R_{opt} が 100% (=15/15) であることがわかる。
＜説明＞ De-Jong の標準関数 F1: parabola は 3 個の独立変数 x_1，x_2，x_3 の二乗和を求める関数である。式の形から，x_1 = 0，x_2 = 0，x_3 = 0 の値をとった時に最小値 0 をとることがすぐわかる。関数 F1 の最小値探索問題は GA の解探索性能を評価する上で最も入門的な問題である。尚，各独立変数の値の観測間隔は 5.12 × 2 /2^{10} =0.01 である。

(2) De Jong's test function F2: Rosenbrock's saddle (unimodal): (see Fig. A.2)

＜定義式＞ $z = f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$

＜定義域＞－ 2.048 ≦ x_i ＜ 2.048　(i=1, 2)
＜染色体長＞ 26 (bit) =13 (bit) × 2 (変数)
＜最適最小値＞ x_1=1, x_2=1 の時 F2 は最小値 0 をとる。
＜実験結果＞表 1 から 4 つの GA モデルの各々について最適解探索率 R_{opt} が 80% (=12/15) 以上であることがわかる。
＜説明＞ De-Jong の標準関数 F2: Rosenbrock's saddle は 2 個の独立変数 x_1 と x_2 があり x_1 の二乗と x_2 の差の二乗を 100 倍した数値と（1 － x_1）の二乗を和した数値を求める関数である。式の形から x_1=1, x_2=1 の時に最小値が 0 であることはすぐわかるが，2 変数の関係で F2 の最小値が求まるので GA にとっては問題を解くことがやや難しい。各独立変数の値の観測間隔は 2.048 × 2 /2^{13} =0.005 である。De Jong の論文では，解の探索精度を，0.001 刻みで，24=12 ビット × 2 変数を染色体長としている。本検討では解の精度を 2 倍に向上させた。

(3) De Jong's test function F3: Step function (unimodal): (see Fig. A.3)

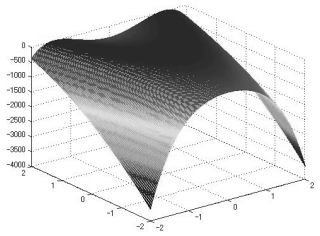

Fig. A.2　Inverted De Jong Test Function F2: Rosenbrock's saddle.

＜定義式＞ $z = f(x_1, x_2, x_3, x_4, x_5) = \sum_{i=1}^{5} \lfloor x_i \rfloor$

＜定義域＞－ 5.12 ≦ x_i ＜ 5.12 (i=1, 2, 3, 4, 5)

＜染色体長＞ 50 (bit) =10 (bit) × 5 (変数)
＜最適最小値＞閉区間 [-5.12,-5.01] に属する数値のうち上記精度のもと GA で探索可能な 12 個の離散値 {-5.12, -5.11, -5.10,...,-5.01} のいずれかの値を各 x_i (i=1, 2, 3, 4, 5) が取った時に，F3 は最小値 -30 をとる。
＜実験結果＞表 1 から 4 つの GA モデルの各々について最適解探索率 R_{opt} が 100% (=15/15) であることがわかる。
＜説明＞ De Jong の標準関数 F3: Step Function は，x_i を越えない最大の整数 $\lfloor x_i \rfloor$ = floor(x_i) を 5 つ足した整数である。ここで $\lfloor\ \rfloor$ はガウス記号である。例えば，$\lfloor$-5.12$\rfloor$ = － 6,$\lfloor$5.12$\rfloor$ = 5,$\lfloor$5$\rfloor$ = 5 となる。従って F3 の最小値は $\lfloor x_i \rfloor$= － 6 をとった時でその値は－ 30 である。他の関数にない特徴は，最適解を持つ点が唯一つではなく，複数の点（先の議論から 12^5 個となる）で最適値を取ることである。点は 5 つの独立変数をその要素とするベクトルである。各独立変数の値の観測間隔は 5.12 × 2 /2^{10} =0.01 である。

(4) De Jong's test function F4: Quadratic with noise (unimodal): (see Fig. A.4)

＜定義式＞ $z = f(x_1, x_2, \cdots, x_{30}) = \sum_{i=1}^{30} i x_i^4 + GAUSS(0,1)$

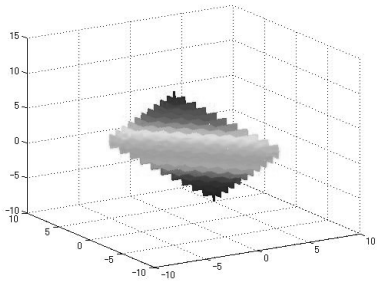

Fig. A. 3.　Inverted, two-dimensional version of De Jong Test Function F3: Step function.

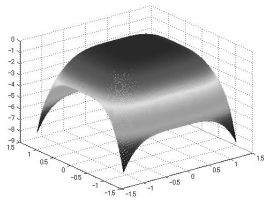

Fig. A. 4　Inverted, two-dimensional version of De Jong Test Function F4: Quadratic with Noise.

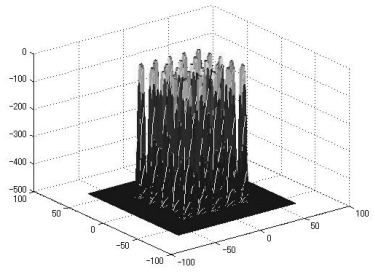

Fig. A. 5　Inverted, two-dimensional version of De Jong Test Function F5: Shekel's foxholes.

注意：ガウス分布に従って発生する雑音が負の場合，符号を変えて正とする。

＜定義域＞ $-1.28 \leqq x_i < 1.28$ (i=1, 2,..., 30)

＜染色体長＞ 240 (bit) =8 (bit) × 30 (変数)

＜最適最小値＞全ての x_i (i=1, 2,..., 30) が $x_i = 0$ で，かつ雑音 =0.0 の時に，F4 は最小値 0 をとる。

＜実験結果＞

F4 による評価では，探索した関数値が 0.01 以下の時に最適解を探索したモデルであると判断した。表 1 から 4 つの GA モデルの各々について最適解探索率 R_{opt} が 27% (=4/15) 以下であることがわかる。しかしながら 4 つのモデルの中で TDGA モデルが解の相対誤差 ER の値が最も小さくその値は 0.01352 であった。この理由は TDGA が最適解探索時に集団の多様性 $Div._{opt}$ を 220 以上確保できていたのに対し，他のモデルは最適解探索時に集団の多様性 $Div.opt$ を 130 以下しか確保できていなかったからである。この結果は集団の多様性を確保することが解の探索精度の向上をもたらすことを示している。更に，ジョブの停止条件を 200 世代から 10000 世代に延長し，10000 世代観測すると，4 つの GA モデルの各々について最適解探索率 R_{opt} が 87% (=13/15) 以上になることを確認した。最適解探索率を向上できた理由は，各世代の終了時の集団の多様性を 50 以上確保できていたからと考えられる。尚，最適解探索効率が悪い理由は雑音が評価関数に含まれていたためである。

＜説明＞ De-Jong の標準関数 F4: Quadratic with Noise は 30 個の「独立変数の 4 乗に変数番号を掛けた数値」の総和で表現される信号と，平均値 0，分散 1 のガウス分布に従って $-\infty$ から $+\infty$ の実数値を確率的に取る雑音 (乱数) の和である。尚 , ガウス分布に従って発生する乱数が負の場合，符号を変えて正とする。関数の形から全ての独立変数 x_i が 0 をとった時最小値 0 となることはすぐわかるが，雑音がある分問題が難しくなっている。各独立変数の値の観測間隔は $1.28 \times 2/2^8 = 0.01$ である。

＜ガウス分布に従って発生する乱数の発生法＞

以下の 2 つから選択可能とした。

・中心極限定理による方法

・Box Müller 法 (cos 型または sin 型)

(5) De Jong's test function F5: Shekel's foxholes (multimodal: multiple peaks): (see Fig. A.5)

< 定義式 >

$$z = f(x,y) = [\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + (x - a[j])^6 + (y - b[j])^6}]^{-1}$$

＜定義域＞ $-65.536 \leqq x, y < 65.536$

＜染色体長＞ 34 (bit) =17 (bit) × 2 (変数)

＜最適最小値＞我々の実験では (x, y)= (− 31.9785, − 31.9785) の時に最小値 0.998003837794628 を探索した。

＜実験結果＞ F5 による評価では，探索した関数値が 0.998004 以下の時に最適解を探索したモデルであると判断した。表 1 から 4 つの GA モデルの各々について最適解探索率 R_{opt} が 100% (=15/15) であることがわかる。

＜説明＞ De Jong の標準関数 F5: Shekel's foxholes は，25 個の二次元の特異点 $a[j] \times b[j]$ からの点 (x,y) の 6 乗誤差に点番号を足した数値の逆数の総和に 1/500 を和した数値の逆数をその値とする関数である。F5 式は 25 個の特異点の値をとった時に極小点となる。多くの極小点をその値としてとることから，当関数は多峰性関数と呼ばれている。極値を複数持つので，その近傍から最小値を探索することは難しい。各独立変数の値の観測間隔は $65.536 \times 2/2^{17} = 0.001$ である。

＜点 a[j] × b[j](j=1,2,･･･,25) ＞は特異点であり，fox holes と呼ぶ。その値は De Jong の博士論文のデータと同じであり，以下の値としている。

a[25]={-32.0,-16.0,0.0,16.0,32.0,-32.0,-16.0,0.0,16.0,32.0,-32.0,-16.0,0.0,16.0,32.0,-32.0,-16.0,0.0,16.0,32.0,-32.0,-16.0,0.0,16.0,32.0}

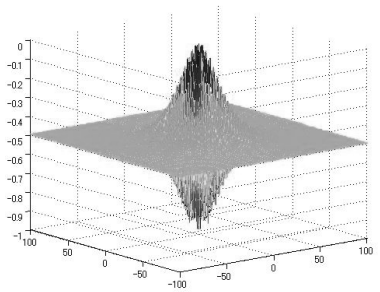

Fig. A. 6　Inverted, Schaffer's F6 function.

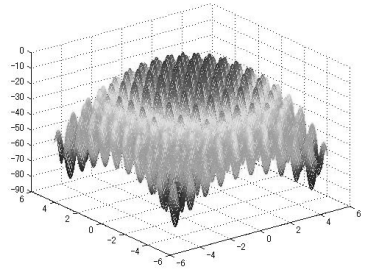

Fig. A. 7　Inverted, two-dimensional version of Rastringin's function.

b[25]={-32.0,-32.0,-32.0,-32.0,-32.0,-16.0,-16.0,-16.0,-16.0,-16.0, 0.0,0.0,0.0,0.0,0.0,16.0,16.0,16.0,16.0,16.0,32.0,32.0,32.0,32.0,32.0}

(6) Schaffer's test function F6 (multimodal): (see Fig. A.6)

＜定義式＞ $z = f(x,y) = 0.5 + \dfrac{\sin^2(\sqrt{x^2+y^2}) - 0.5}{(1+0.001\times(x^2+y^2))^2}$

＜定義域＞－ 100 ≦ x , y ＜ 100

＜染色体長＞ 16 (bit) =8 (bit) × 2 (変数)

＜最適最小値＞ x=y=0 の時，F6 は最小値 0 をとる。

＜実験結果＞表 1 から 4 つの GA モデルの各々について最適解探索率 R_{opt} が 100% (=15/15) であることがわかる。

＜説明＞ α が実数の時，一般に sin(α) 関数は -1 から 1 の値をとるので，1 ≧ $\sin^2(\alpha)$ ≧ 0 となる。従って Schaffer's　F6 関数の場合，1 ≧ z=$f(x,y)$ ≧ 0 となっている。このことから，F8 が 0 となるのは x=0,y=0 の場合であることがすぐわかる。各独立変数の値の観測間隔は 100 × 2 /2^8=0.78125 である。

(7) Rastringin's test function F7 (multimodal): (see Fig. A.7)

＜定義式＞

$$z = f(x_1, x_2, \cdots, x_{20}) = 10\times 20 + \sum_{i=1}^{20}(x_i^2 - 10\times\cos(2\pi x_i))$$

＜定義域＞ － 5.12 ≦ x_i ＜ 5.12 (i=1, 2,…., 20)

＜染色体長＞ 280 (bit) =14 (bit) × 20 (変数)

＜最適最小値＞ 全ての x_i (i=1, 2,..., 20) が x_i = 0 の時，F7 は最小値 0 をとる。

＜実験結果＞ F7 による評価では，探索した関数値が 2 以下の時に最適解を探索したモデルであると判断した。表 1 から TDGA モデルのみが 2 以下の関数値を探索していることがわかる。そしてその最適解探索率 R_{opt} は 80% (=12/15) であった。一方，表 1 から TDGA のみが最適解探索時に集団の多様性 Div_{opt} を 85 以上確保できていたのに対し，他のモデルは最適解探索時に集団の多様性 $Div._{opt}$ を最大 28 しか確保できていなかったことがわかる。この結果は，集団の多様性を確保することが最適解探索率 R_{opt} の向上をもたらすことを示している。

＜説明＞一般に α が実数の時，（1 － cosα）≧ 0 なので，関数 F7 の従属変数 z ≧ 0 となる。z=0 となるのは x_i ^{2}=0，かつ 1 － cos x_i =0 の時であり，両方を満たすのは x_i =0 の時であることがわかる。各独立変数の値の観測間隔は 5.12 × 2 /2^{14}=0.000625 である。

(8) Schwefel's test function F8 (multimodal): (see Fig. A.8)

＜定義式＞ $z = f(x_1, x_2, \cdots, x_{10}) = \sum_{i=1}^{10} - x_i \times \sin(\sqrt{|x_i|})$

＜定義域＞ － 500 ≦ x_i ＜ 500 (i=1, 2, 3..., 10)

＜染色体長＞100 (bit) =10 (bit) × 10 (変数)

＜最適最小値＞全ての x_i(i=1, 2,...,10) が x_i=420.968746 の時，F8 は最小値 － 4189.828873 をとる。

＜実験結果＞表 1 は，染色体長が 100 ビットの精度のもと，最小値 － 4189.822635 を探索できたことを示している。そして，4 つの GA モデルの各々について最適解探索率 R_{opt} が 100% (=15/15) であることがわかる。

染色体長が 100 ビットの精度のもとでは，GA で

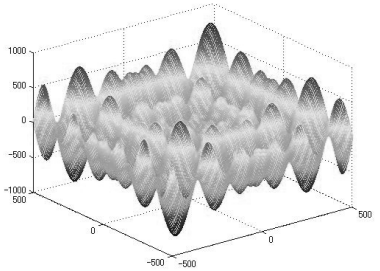

Fig. A. 8　Inverted, two-dimensional version of Schwefel's function.

探索可能な最小値は全てのx_iについてx_i=420.8984375の時でその時のF8の値は－4189.822635であることを実験で検証できる。従って，F8による評価では，探索した関数値が－4189.822635の時に最適解を探索したモデルであると判断した。

全てのx_i (i=1, 2,...,10) についてx_i=420.968746の時，F8は最小値－4189.8288727243371をとることはよく知られている。染色体長が100ビットの精度のもとでは，全てのx_iがx_i=420.8984375の時にGAは－4189.8226352189849の極小値を探索できることを実験で検証した。この数値の前後で観測可能なx_iは419.921875と421.875であるが，その時の関数値は－4188.4466107056924と－4188.7922142631587であり，－4189.822635218984900より大きい。そして，x_i=420.968746は観測可能点x_i=420.8984375とx_i=421.875の区間に入っている。従って，関数の連続性より，極小値－4189.8226352189849を関数F8の最小値であると判断した。従って，F8による評価では，探索した関数値が－4189.8226352189849の時に最適解を探索したモデルであると判断した。表1は，4つのGAモデルの各々が最小値－4189.8226352189849を探索できたこと，最適解探索率R_{opt}が100% (=15/15) であることを示している。

＜説明＞一般にαが実数の時, $-1 \leqq \sin\alpha \leqq 1$なこと，10個の変数$x_i$について｜$x_i$｜≦500（$i$=1,2・・・,10）なことから，Schwefel's functionの従属変数zの値域は，$-5000<z<5000$となる。従って，従属変数zは正負のいずれの値もとる。周期関数sinを含んでいるので，F8は複数の局所最小解を持つ。各独立変数の値の観測間隔は$500 \times 2/2^{10} = 0.9765625$である。

(9) Griewank's test function F9 (multimodal): (see Fig. A.9)

＜定義式＞

$$z = f(x_1, x_2, \cdots, x_{10}) = \frac{1}{4000}\sum_{i=1}^{10} x_i^2 - \prod_{i=1}^{10}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

＜定義域＞ $-600 \leqq x_i < 600$ (i=1, 2, 3..., 10)

＜染色体長＞ 110 (bit) =11 (bit) × 10 (変数)

＜最適最小値＞全てのx_i (i=1, 2,..., 10) についてx_i=0の時，F9は最小値0をとる。

＜実験結果＞ F9による評価では，探索した関数値が0.1以下の時に最適解を探索したモデルであると判断した。表1から4つのGAモデルのうちTDGAのみが最適解探索率R_{opt}が93% (=14/15) であること，他の3つのモデルは53% (=8/15) 以下であることがわかった。一方，表1からTDGAのみが最適解探索時に集団の多様性$Div._{opt}$を60以上確保できていたのに対し，他のモデルは最適解探索時に集団の多様性$Div._{opt}$を最大32しか確保できていなかったことがわかる。この結果は，集団の多様性を確保することが最適解探索率R_{opt}の向上をもたらすことを示している。

＜説明＞ $x_i^2 \geqq 0, 1 - \prod_{i=1}^{10}\cos\left(\frac{x_i}{\sqrt{i}}\right) \geqq 0$, ≧0なので，F9は正の値を取る。従って，F9が0となるのは両不等式が0となる時であり，それを満たすのはx_i = 0の時であることがわかる。各独立変数x_iの値の観測間隔は$600 \times 2/2^{11} = 0.5859375$である。

(10) Ackley's test function F10 (multimodal): (see Fig. A.10)

＜定義式＞$z = f(x_1, x_2, \cdots, x_{10}) =$

$$-20 \times \exp\left(-0.2 \times \sqrt{\frac{1}{10}\sum_{i=1}^{10} x_i^2}\right) - \exp\left(\frac{1}{10}\sum_{i=1}^{10}\cos(2\pi x_i)\right) + 20 + e$$

＜定義域＞ $-32 \leqq x_i < 32$ (i=1, 2, 3..., 10)

＜染色体長＞ 60 (bit) =6 (bit) × 10 (変数)

＜最適最小値＞全てのx_i (i=1, 2,...,10) がx_i=0の時，F10は最小値0をとる。

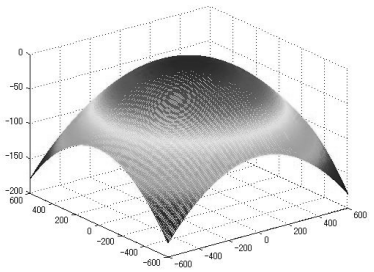

Fig. A. 9 Inverted, two-dimensional version of Griewank's function.

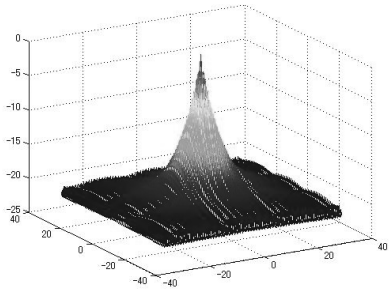

Fig. A.10 Inverted, two-dimensional version of Ackley's function.

＜実験結果＞表 1 から 4 つの GA モデル全体で最適解探索率 R_{opt} が 95% (=(13+14+15+15)/(15 × 4)) であることがわかる。
＜説明＞ F10 は，指数関数 $y=e^x=\exp(x)$ の性質を利用して GA がその最小値を実験的に導けるかという問題である。理論的に関数 F10 の最小値が 0 となる理由は，以下である。先ず，e^x は単調増加関数であり, $1=e^0$ なことから e^x は x が負の時 1 以下となる。

$$1 \geq \exp(-0.2 \times \sqrt{\frac{1}{10}\sum_{i=1}^{10} x_i^{\,2}})$$

また，$e=e^1$ = 2.718・・，$-1 \leqq \cos x \leqq 1$ なことから，以下の不等式が成立することがわかる。

$$e \geq \exp(\frac{1}{10}\sum_{i=1}^{10}\cos(2\pi x_i))$$

従って，F10 は正の値をとることがわかる。F10 が最小値 0 となるのは，両不等式にて等号が成立する場合であり，それは x_i が 0 の場合であることがわかる。各独立変数 x_i の値の観測間隔は 32 × 2 /2^6=1 である。

7-7　実験結果のまとめ

10 個のテスト関数を用いた実験結果を最適解探索効率 (R_{opt}) の観点から以下にまとめる。相対誤差 (ER) の観点からの実験結果のまとめについては割愛する。

<u>最適解探索効率 (R_{opt}) の観点からの 8 つの GA モデルの評価</u>

(1) Two-point crossover 法：
0.46 (=278/(15 × 10 × 4))
内訳：
(a) Non-Scaling GA: 0.37 (=56/150)
(b) Scaling Window GA: 0.34 (=51/150)
(c) Immune-GA with scaling window: 0.4 (=60/150)
(d) TDGA with scaling window: 0.74 (=111/150)
各々の関数に関する上記の内訳は割愛する。

(2) Two-point crossover + BLX- α (α =0.5) 法：
0.78 (=466/(15 × 10 × 4))
内訳：以下は表 1 の要約である。
(a) Non-Scaling GA: 0.73 (=110/150)
(b) Scaling Window GA: 0.73 (=109/150)
(c) Immune-GA with scaling window: 0.75 (=113 /150)
(d) TDGA with scaling window: 0.89 (=134/150)

上記の実験結果は，GA で関数最小値探索問題を解く時，BLX-α と二点交叉（two-point crossover）が協調しながら交叉演算を行なう場合の TDGA が最も効率的な解探索を実現できること，実験では最適解探索率 R_{opt} が 0.89% (=134/150)) であったことを示している。そして，10 個の良く知られたテスト関数を用いた実験により，集団の多様性を確保することが結果として最適解探索率 R_{opt} を向上させることに繋がることを示した。

8. 結論と今後の課題

関数最小値探索問題を遺伝的アルゴリズムで解く場合に，集団の多様性を TDGA におけるエントロピーで定量的に測定し，「集団の多様性を確保することにより，最適解探索率や探索解の精度を向上できる」という仮説を，De Jong の 5 つの標準関数と 5 つの多峰性関数を含む良く知られた 10 個の標準テスト関数を用いた実験により，定量的に検証した。集団の多様性を確保しながら効率的探索を実現する方法として，TDGA による手法が有効であること，そして広域探索を GA の二点交叉で実現し，局所探索を BLX-α で実現する手法が有効であることを実験により検証した。今後，（1）離散空間で解を近似する際の解の精度の向上策，（2）拡張遺伝子交叉オペレータ交替法の有効性の検証，（3）広域探索を SPX，局所探索を BLX- α で実現する場合の仮説の検証を今後の課題とする。

後書き

本論文は，以下の論文 [19] の邦訳である。

謝辞

本研究は JSPS 科研費　JP15K00347 の助成を受けたものです。ここに感謝致します。

【参照文献】

[1] J. H. Holland, “Adaptation in Natural and Artificial Systems,” MIT Press, 1992.
[2] E. Aarts and J. K. Lenstra, Local Search in Combinatorial Optimization, Princeton University Press, 2003.
[3] D. E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning,” Addison-Wesley Publishing Company, Inc., 1989.

[4] M.-K. Tsai, J.-M. Yang, Y.-F. Tsai, and C.-Y. Kao, "An Evolutionary Algorithm for Large Traveling Salesman Problems," IEEE Transactions on Systems, Man, and Cybernetics- Part B. Cybernetics., Vol. 34, No.4, pp. 1718 – 1729 (2004)

[5] K. A. De Jong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," Doctoral Dissertation, University of Michigan, January 1975.

[6] R. L. Haupt and S. E. Haupt, "Practical Genetic Algorithms," John Wiley & Sons, Inc., 2004.

[7] R. Takahashi, "Quantitative Evaluation of Iterative Extended Changing Crossover Operators to Solve the Traveling Salesman Problem," in Proceedings of 2014 10th International Conference on Natural Computation (ICNC), pp. 235-244, IEEE, 2014.

[8] K. Maekawa, N. Mori, H. Tamaki, H. Kita and Y. Nishikawa, "A Genetic Solution for the Traveling Salesman Problem by means of a Thermodynamical Selection Rule", Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96), pp. 529-534, 1996.

[9] K. Mori, M. Tsukiyama, and T. Fukuda, "Application of an Immune Algorithm to Multi-Optimization Problem," T. IEE Japan, Vol.117-C, No.5, pp.593-598, 1997 (in Japanese).

[10] J. Lu and M. Xie: "Immune-Genetic Algorithm for Traveling Salesman Problem," TRAVELING SALESMAN PROBLEM, THEORY AND APPLICATIONS, Edited by Donald Davendra, Published by InTech, pp. 81-96, 2010.

[11] L. J. Eshelman and J. D. Schaffer, "Real Coded Genetic Algorithms and Interval-Schemata," Foundations of Genetic Algorithms 2, pp. 187-202, Morgan Kaufmann, 1993.

[12] M. Takahashi and H. Kita, "A Crossover Operator Using Independent Component Analysis for Real-Coded Genetic Algorithms," Proceedings of the 2001 IEEE Congress on Evolutionary Computation Seoul, pp.643-649, 2001.

[13] S. Tsutsui, M. Yamamura, and T. Higuchi, "Multi-parent Recombination with Simplex Crossover in Real Coded Genetic Algorithms," Proceeding of Genetic and Evolutionary Computation Conference, pp.657-664, ACM, 1999.

[14] H. Kita, "Up-to-date Genetic algorithm to solve the combinatorial optimization problem," Tutorial on Information Processing based on Probabilistic Algorithm, Workshop on SMAPIP, 2003 (in Japanese).

[15] T. Sato and M. Hagiwara, "Bee System: Finding Solution by a Concentrated Search," T. IEE Japan, Vol.118-C, No.5, pp.721-726, 1998 (in Japanese).

[16] J. J. Grefenstette, "Optimization of Control Parameters for Genetic Algorithms," IEEE Trans. on Systems, Man, and Cybernetics, vol.SMC-16, no.1, pp.122-128, 1986.

[17] J. M. Dieterich and B. Harkte, "Empirical Review of Standard Benchmark Functions Using Evolutionary Global Optimization," Applied Mathematics, Vol. 3, pp. 1552-1564, 2012.

[18] X.-S. Yang, "Test Problems in Optimization, in: Engineering Optimization: An Introduction with Metaheuristic Applications (Eds. X.-S. Yang)", John Wiley & Suns, 2010.

[19] R. Takahashi, "Empirical Evaluation of Changing Crossover Operators to Solve Function Optimization Problems", Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence(IEEE SSCI 2016), pp.1-10, DOI: 10.1109/SSCI.2016.7850141,2016.

◆著者紹介

髙橋　良英　Ryouei Takahashi

京都情報大学院大学教授。
博士（工学）（早稲田大学）。
早稲田大学大学院理工学研究科修士課程修了（数学専攻）。
早稲田大学理工学部数学科卒業（理学士）。
元八戸工業大学大学院教授。
元 NTT 情報流通プラットフォーム研究所勤務。
元 NTT 横須賀電気通信研究所勤務。