確率的スキーマ貪欲法の検討と拡張，性能比較について

丸 山 崇† 北 栄 輔†

組合せ最適化問題に対する進化的計算手法の 1 つに確率的スキーマ貪欲法（Stochastic Schemata Exploiter: SSE）がある．本論文では，スキーマの関係に着目して SSE を改良した拡張型確率的スキーマ貪欲法（Extended Stochastic Schemata Exploiter: ESSE）を提案する．また，0/1 組合せ最適化問題において，SSE と ESSE を Minimal Generation Gap（MGG）や Bayesian Optimization Algorithm（BOA）と性能比較を行い，それらの探索性能を検討する．

Investigation and Extension of Stochastic Schemata Exploiter

Takashi Maruyama† and Eisuke Kita†

Stochastic Schemata Exploiter (SSE) is one of the evolutionary optimization algorithms for solving the combinatorial optimization problems. Next, we present the Extended SSE (ESSE) algorithms which are composed of the original SSE and new ESSE operations. The SSE and the ESSE are compared with the Minimal Generation Gap (MGG) and the Bayesian Optimization Algorithm (BOA) in 0/1 combinatorial optimization problem in order to discuss their convergence property.

1. はじめに

工学のさまざまな分野において，大規模組合せ最適化問題をできるだけ短い計算時間で解くことが求められている．また，多くの組合せ最適化問題が大谷構造（big valley structure）と呼ばれる特徴を持っている[1]．大谷構造の特徴を持った問題は，「良い解どうしは似通った構造を持っている」といい換えることができ，このような特徴を Proximate Optimality Principle（POP）という[2]．このような問題を効率的に解く方法として進化的計算手法がある．その 1 つに，相澤が提案した確率的スキーマ貪欲法（Stochastic Schemata Exploiter: SSE）[3],[4] というアルゴリズムがある．SSE は，評価値の良い個体を基にして，そこから似通った構造を持つ個体を集中的に探索するアルゴリズムなので，大谷構造の特徴を持つ多くの実問題に適したアルゴリズムであると考えられる．そこで，Minimal Generation Gap（MGG）や Bayesian Optimization Algorithm（BOA）との性能比較を通して，SSE アルゴリズムの有効性を検証する．

さらに，本研究では，SSE のスキーマ処理を拡張した拡張型確率的スキーマ貪欲法（Extended Stochastic Schemata Exploiter: ESSE）を提案し，SSE との性能比較を通して ESSE の有効性を示す．性能比較には，各探索手法の探索能力が最大限に引き出された状態で性能を比較するため，コード化，交叉の設計が容易な 0/1 組合せ最適化問題を用いた．

本論文の構成は以下のようになっている．2 章では，研究背景と，SGA，MGG，BOA，SSE，ESSE について簡単に述べる．3 章では，SSE のアルゴリズム，本研究で提案する ESSE の概要とアルゴリズムの詳細について述べる．4 章では，0/1 組合せ最適化問題を用いて SGA，MGG，SSE，ESSE の性能を比較し，5 章では，BOA，SSE，ESSE を比較する．最後に，6 章において，本論文全体のまとめを述べる．

2. 研究の背景

工学で現れる多くの組合せ最適化問題の解空間は，局所解がすり鉢上に分布する大谷構造（big valley structure）と呼ばれる地形の特徴を持つ[1]．大谷構造の特徴を持つ問題は，「良い解どうしは似通った構造を持つ」といえる．この概念は，POP と呼ばれる[2]．問題が POP の特徴を持つとき，良い解に似通った構造を持つ解の中に，より良い解が見つかる可能性が高いと考えられ，複数の最適化法を組み合わせるメタ戦略の多くは，この概念に基づいて設計されている．ところで，相澤が確率的スキーマ貪欲法（Stochastic Schemata

† 名古屋大学大学院情報科学研究科複雑系科学専攻
Graduate School of Information Science, Nagoya University

Exploiter: SSE）[3),4)] というアルゴリズムを提案している．SSE は探索アルゴリズムに「スキーマ貪欲な方針」を用いている．「スキーマ貪欲な方針」とは，アルゴリズムの構成要素に "既存のスキーマの中で，サンプル平均の良いスキーマをさらにサンプルする" 機構を含むことを指す．SSE は，評価値の良い個体と似通った構造を持つ個体を集中的に探索するアルゴリズムであり，POP に適したアルゴリズムであるので，大谷構造の特徴を持つ組合せ最適化問題において，SSE のスキーマ貪欲な方針による解探索は有効であると考えられる．そこで，SSE を拡張した拡張型確率的スキーマ貪欲法（Extended Stochastic Schemata Exploiter: ESSE）を提案する．また，SSE と ESSE を実問題に有効とされる Minimal Generation Gap（MGG）や SSE と類似の設計方針を持つ Bayesian Optimization Algorithm（BOA）と比較する．

2.1　SGA と MGG

Holland によって考案された遺伝的アルゴリズム（Genetic Algorithm: GA）[5)] は，自然界の進化のメカニズムを模倣したアルゴリズムであり，最適化，適応，学習など多方面に応用されている．基本的な GA として，単純 GA（Simple Genetic Algorithm: SGA）[6)] があり，そこからさまざまな拡張手法が提案されている．SGA には，初期収束（early convergence）と進化的停滞（evolutionary stagnation）という問題がある[7),8)]．佐藤らが提案した Minimal Generation Gap（MGG）[9)] はこの問題を解決したアルゴリズムである．MGG は，さまざまな問題への適用研究がされており，実問題に対して有用と考えられている[10)〜12)]．

2.2　BOA

SGA のアルゴリズムは，主に選択，交叉，突然変異からなる．交叉，突然変異を用いて 2 つの親個体から子個体を生成するかわりに，優良解集団の情報から新たな解を生成する手法が提案されており，分布推定アルゴリズム（Estimation of Distribution Algorithm: EDA）と呼ばれる．このアルゴリズムでは，交叉，突然変異という操作は行われないで，集団における個体分布状況を確率モデルによって推定し，推定された確率モデルに基づいて，次世代の子個体を生成する．このように EDA は SSE と類似の設計思想を持っている．Bayesian Optimization Algorithm（BOA）[13)] は，EDA の 1 つである．

2.3　SSE

SSE は，GA と同様に個体集団を用いた多点探索であるが，以下に述べるスキーマ貪欲な方針を用いているのが特徴である．

SSE では，集団中のスキーマ H を集団中で H を含む個体の部分集合と対応付け，良いスキーマの選出問題を優れた個体部分集合の選出問題に置き換えている．SSE では，観測された良い個体部分集合に含まれるスキーマを選ぶことで，サンプル平均が良いスキーマだけが選択される．この操作を繰り返すことでスキーマ貪欲な方針（既存のスキーマの中で，サンプル平均の良いスキーマをさらにサンプルすること）を実現する．

また，SSE は，必要な制御パラメータは個体数と突然変異率だけという特徴を持つ．

2.4　ESSE

本研究では，スキーマの関係に着目して SSE を改良した拡張型確率的スキーマ貪欲法（Extended Stochastic Schemata Exploiter: ESSE）を提案する．SSE は，スキーマ貪欲な方針を実現するために個体部分集合の選出問題を用いている．その際，個体部分集合の平均評価値の半順序関係を用いて，新たな個体部分集合の生成と選出のためのソーティングを行っている．SSE ではつねにソーティングするだけであるが，ESSE ではソーティングの過程で，個体部分集合から抽出されるスキーマを比較し，両者が同一の場合，包含関係にある場合，部分的に一致する場合において個体部分集合のマージなどの操作を行うことで SSE の性能を改善することを目指している．

3.　SSE と ESSE のアルゴリズム

3.1　SSE アルゴリズム

SSE では，良いスキーマの選出問題を優れた個体部分集合の選出問題に置き換えている．個体部分集合の生成には，以下に示す個体部分集合の平均評価値の半順序関係を用いる．

個体集団 P_t において，M 個の個体をその適合度の降順に並べたインデックスを $c_1, c_2, \cdots, c_M$ とおく．P_t の任意の個体部分集合 S $(\neq \phi)$ について，その中に含まれる最大のインデクスを $L(S)$ で表す．$(S - c_k)$ は，集合 S から要素 c_k を除いた集合とする．また，和集合を $\cup$ で表す．$L(S) < M$ のとき，P_t の個体部分集合の間に以下の半順序関係が存在する．

- S の平均評価値は $S \cup c_{(L(S)+1)}$ の平均評価値よりも良い．
- S の平均評価値は $(S - c_{L(S)}) \cup c_{(L(S)+1)}$ の平均評価値よりも良い．

以上の半順序関係を用いることで，個体部分集合を生成していく．

SSE アルゴリズムは以下のようになる．

(1)　初期集団の生成

初期の個体 M 個をランダムに生成する．

(2) インデックスの作成
適合度の降順に個体を並べたインデックスを作る．

(3) アクティブリストの生成
インデックスから半順序関係を用いて個体部分集合を M 個作る．

(4) スキーマの抽出
(3) で得られたアクティブリストの M 個の個体部分集合からそれぞれ共通スキーマを抽出する．

(5) 子個体の生成
(4) で得られた M 個のスキーマからそれぞれランダムに個体を生成し，突然変異操作を適用する．

(6) 世代交代
(5) で生成された個体群を次世代の個体とする．

(7) 終了条件
停止条件が満たされるまで，(2) から (6) を繰り返す．

ここで，インデックスは全個体が適合度の降順に格納されたリストであり，アクティブリストは個体部分集合が平均評価値の降順に格納されたリストである．

3.2 ESSE の概要

SSE では，“新しい個体部分集合” と “すでにアクティブリストに挿入されている個体部分集合” の平均評価値を比較し，“新しい個体部分集合” をアクティブリストに挿入する．ESSE では，個体部分集合から抽出されたスキーマどうしの比較を行うことで，“新しい個体部分集合” をアクティブリストに挿入する．ここで，2 つのスキーマを比較したとき以下の 4 つの場合が存在する．

- まったく同じとなる場合（case1）
- 一方がもう一方に含まれる包含関係になる場合（case2）
- 部分的に一致する場合（case3）
- まったく異なった場合（case4）

まったく異なった場合は 2 つのスキーマから共通のスキーマを抽出することはできないので，case1〜case3 の場合について考える．

以下，2 つのスキーマを A，B とする．A の個体部分集合と B の個体部分集合の和集合で作られた個体部分集合から抽出されるスキーマを C とする．また，スキーマの評価値を f で表す．以下に case1〜3 に対応した処理 1〜3 を述べる．

- 処理 1（A と B が同一のとき）
  A と B が同一であるとき，B の個体部分集合の要素を A の個体部分集合の要素に加え，A の個体部分集合をアクティブリストに挿入する．
  処理 1 は個体部分集合を再構築して，スキーマをより正確に評価し直すことを目的とする．これにより，処理 1 は集団中における同じスキーマの存在を防ぎ，多様性の維持が実現できる．
- 処理 2（A が B を含むとき）
  A と B に包含関係があるとき，2 つのスキーマの評価値の関係が $f(A) \geq f(B)$ ならば，A の個体部分集合をアクティブリストに挿入し，A の個体部分集合の要素を B の個体部分集合の要素に加え，B の個体部分集合をアクティブリストに挿入する．$f(A) < f(B)$ ならば，A の個体部分集合の要素を B の個体部分集合の要素に加え，B の個体部分集合をアクティブリストに挿入する．
  B が A よりも評価値が良いということは，A が B から成長して大きくなる際に，評価値を下げたことを意味する．処理 2 は，スキーマが成長し大きくなる課程で，成長により平均評価値が低下したスキーマを排除することで，解の精度を低下させるような探索方向をとらないようにする．
- 処理 3（A と B に共通部分があるとき）
  A と B に共通部分があるとき，A と B のうち評価値の高いスキーマの個体部分集合をアクティブリストに挿入し，A，B の共通のスキーマである C の個体部分集合をアクティブリストに挿入する．
  処理 3 は 2 つのスキーマの共通スキーマが生成されるので，それらのスキーマの隣接解を調べる近傍探索の効果が期待できる．

ESSE は，これら 3 つの処理の 1 つまたは複数を SSE に組み合わせたアルゴリズムである．

3.3 ESSE のアルゴリズム

前節で述べた処理 1〜3 を SSE のアクティブリスト生成に組み込むために，3.1 節で述べた SSE アルゴリズムの (2)〜(3) を改良した．ESSE アルゴリズムを以下に示す 3.3.1 項の「アクティブリスト生成」，3.3.2 項の「スキーマ処理」で述べる．なお，「アクティブリスト生成」の中で「スキーマ処理」が呼び出され，処理 1〜3 を実行する．

3.3.1 アクティブリスト生成

(1) 長さ M の参照用リスト original list を作り，M 個の個体を適合度の降順に並べてリストに挿入する．

(2) 長さ M の処理用リスト active list を作り，

その第 1 要素に original list の第 1 要素である $\{c_1\}$ を，それ以外の要素に ϕ をおく．$start\_index = 1$ とする．

(3) $i = 1$ とする（ステップ 1）．

(4) ステップ i において，active list の第 i 要素 S_i の要素数が 1 以上である場合は (a) へ，0 である場合は (b) へ．

 (a) active list の第 i 要素 S_i を S'_i にコピーする．(5) へ．

 (b) $start\_index = M$ ならば終了．そうでなければ，$start\_index = start\_index + 1$ とし，$S'_i = \{c_{start\_index}\}$ とする．(5) へ．

(5) $L(S'_i) < M$ ならば，半順序関係に従う 2 つの個体部分集合 $S'_i \cup c_{(L(S'_i)+1)}$ および $(S'_i - c_{L(S'_i)}) \cup c_{(L(S'_i)+1)}$ を生成する．ここで c_k は original list の第 k 要素である．

(6) $S'_i \cup c_{(L(S'_i)+1)}$ および $(S'_i - c_{L(S'_i)}) \cup c_{(L(S'_i)+1)}$ を，active list の第 0 番目の要素から順に全要素と比較し，スキーマの関係に従い case1〜4 の処理（3.3.2 項）を実行する．

(7) $i < M$ ならば $i = i + 1$ として，(4) から (6) を繰り返す．$i = M$ ならば終了する．

(4) はアクティブリストの要素数の判定を行い，要素数が 0 であった場合は半順序関係を生成するインデックスを増加させていく．これにより，要素数が 0 になる個体部分集合の生成を抑えることができる．

3.3.2 スキーマ処理

$S'_i \cup c_{(L(S'_i)+1)}$，$(S'_i - c_{L(S'_i)}) \cup c_{(L(S'_i)+1)}$ を S'' とおき，以下のアルゴリズムをそれぞれ実行する．

(1) $j = 1$ とする．

(2) S'' の個体部分集合から抽出したスキーマと active list の第 j 番目の個体部分集合から抽出したスキーマの関係を調べ，その関係に対応する処理を実行する．

- case1（A と B が同一である場合）
  C を active list の第 j 番目の要素にする．そして終了する．
- case2（A が B を含んでいる場合）
  - $f(A) \geq f(B)$
    A を active list の第 j 番目の要素にする．C を S'' とする．(3) へ．
  - $f(A) < f(B)$
    C を active list の第 j 番目の要素にする．そして終了する．
- case3（A と B が部分的に一致している場合）
  - $f(A) \geq f(B)$
    A を active list の第 j 番目の要素にする．C を S'' とする．(3) へ．
  - $f(A) < f(B)$
    B を active list の第 j 番目の要素にする．C を S'' とする．(3) へ．
- case4（A と B がまったく違う場合）（SSE と同じ処理）
  - $f(A) \geq f(B)$
    A を active list の第 j 番目の要素にする．B を S'' とする．(3) へ．
  - $f(A) < f(B)$
    B を active list の第 j 番目の要素にする．A を S'' とする．(3) へ．

(3) $j < M$ ならば $j = j + 1$ として，(2) を繰り返す．そうでなければ終了する．

4. 性能比較 1

本章では，SGA，MGG と SSE，ESSE の性能比較を行う．

4.1 テスト問題

GA は形質遺伝に優れたコード化，交叉の設計をしないと，探索能力を最大限に引き出すことができないことが指摘されている[14]．そこで，コード化，交叉の設計が容易な 0/1 組合せ最適化問題を用いて性能比較を行う．ここでは，以下に述べる騙し問題[15]，ナップザック問題[16]，グラフ分割問題[17] を用いて性能比較をする．

- 騙し問題
  GA および他の多くの最適化アルゴリズムが苦手とするのは，騙し問題といわれるタイプの非線形問題である．実際の応用問題は騙し問題の性質を持っている場合が多い[18]．ここでは，**表 1** に示す 4 ビットの騙し関数[15] を部分解とし，その和として定義される騙し問題を考える．騙し問題の定義式を式 (1) に示す．

$$f_{deception} = \sum_{i=1}^{n} f_d(x_i) \qquad (1)$$
$$x_i \in 0000, 0001, \cdots, 1111$$

  ここで，設計変数 x_i は 4 つの 0/1 の組合せである．n は部分解の個数であり，ここでは，n を 10 とする．
- ナップザック問題
  n 個の荷物から制限重量 b の範囲で価値を最大

表 1 騙し問題の部分解
Table 1 Part solutions of deceptive problem.

$f_d(1111) = 30$	$f_d(0000) = 28$	$f_d(0001) = 26$
$f_d(0010) = 24$	$f_d(0100) = 22$	$f_d(1000) = 20$
$f_d(0011) = 18$	$f_d(0101) = 16$	$f_d(0110) = 14$
$f_d(1001) = 12$	$f_d(1010) = 10$	$fd(1100) = 8$
$f_d(1110) = 6$	$f_d(1101) = 4$	$f_d(1011) = 2$
$f_d(0111) = 0$		

にする荷物群を選ぶナップザック問題を考える．目的関数と制約条件は次式で与えられる．

$$\max_{\{x_i\}} \sum_{i=1}^{n} c_i x_i$$
$$\text{subject to} \sum_{i=1}^{n} a_i x_i \leq b \qquad (2)$$
$$x_i \in 0, 1 \qquad (i = 1, \cdots, n)$$

ここで，荷物 i の重さ a_i と価格 c_i は 1 から 100 までの一様乱数により定める．また，制限重量 b は 10000，荷物数 n（次元数）は 400 とする．

- グラフ分割問題

グラフ分割問題は，頂点と枝が定義されているとき，またがる枝の本数を最小にしながら，頂点を等分割する問題である．頂点は 2 つの集合のどちらかに含まれるので，0/1 の組合せとして表現できる．本研究では，Johnson らの研究で用いられた G124.08 [17] を用いる．G124.08 は頂点数が 124 であり，任意の頂点間に確率 0.8 で枝を与えたランダムグラフである．2 つの集合を L，R，各集合の頂点数を $|L|$，$|R|$ とし，またがる枝の数を $c(L, R)$ と表す．グラフ分割問題は以下の式 (3) で与えられる．

$$f_{graph}(L, R) = -c(L, R) - \alpha(|\,L\,| - |\,R\,|)^2 \qquad (3)$$

ここで，制約条件の違反を表す正の定数 α を 0.1 とする．

4.2 実験結果

SGA，MGG の交叉には 2 点交叉を用い，交叉率は 1 とする．突然変異率の値は，値を変化させて実験し，各探索法でそれぞれ最適と思われる値を用いる．MGG は，各 1 世代で生成する子個体数を母集団の個体数と同じとする．

ESSE は，処理 1〜3 の組合せで，7 通りある．以下，その組合せを c1〜c7 と表記する（**表 2**）．

また，各アルゴリズムともに，評価回数は 1 世代あたり母集団の個体数だけ必要となる．各アルゴリズムともに 1 世代あたりの適合度評価による計算負荷は同じであるので，実行世代数で評価することが可能となる．実行世代数は，騙し問題では 40000，ナップザック問題では 10000，グラフ分割問題では 7500 とし，各アルゴリズムを異なる初期個体集団から 50 回実行する．各アルゴリズムにおいて，個体数は 10，50，100 の 3 種類で実行する．各アルゴリズムによる最良個体の最終到達解の平均値を**表 3** に，最良個体の最終到達解の標準偏差値を**表 4** に示す．

表 2 処理 1〜3 の組合せ
Table 2 Combination of process.

	処理 1	処理 2	処理 3
c1	有		
c2	有	有	
c3		有	
c4		有	有
c5			有
c6	有		有
c7	有	有	有

4.2.1 ESSE 処理の検討

表 3，表 4 より，すべての問題において，個体数によらず，処理 1 を含んだ c1，c2，c6 が良い解探索性能を示している．また，処理 1〜3 のいずれか 1 つだけを含む c1，c3，c5 に注目すると，処理 1 のみから構成される c1 が各個体数において最終到達解の平均値，最終到達解の標準偏差値のどちらも良い性能を示している．

以上のことからすべての問題において，処理 1 のみから構成されている c1 が平均的に良い性能を示しているといえる．しかし，c2 は個体数 10 で行った騙し問題と，個体数 100 で行ったナップザック問題，グラフ分割問題において最も良い到達解を得ている．また，c6 は個体数 10，50 で行ったグラフ分割問題において最も良い到達解を得ている．このことから，処理 1 とともに処理 2，処理 3 を適切に利用することで解探索性能が向上する可能性があるといえる．

4.2.2 SSE，ESSE の収束特性と解探索性能

前項の考察で 7 つの ESSE のうちで最良性能の出た c1 と SGA，MGG，SSE を比較することにする．

以下の図 1〜18 において，横軸に世代数をとり，縦軸に最良個体の到達解の平均値，もしくは，最良個体の到達解の標準偏差値をとる．また，アルゴリズム名の後ろの数値は集団を構成する個体数を示している．

4.2.2.1 騙し問題

MGG による最良個体の到達解の平均値のグラフを**図 1** に，標準偏差値のグラフを**図 2** に示す．図 1 より，個体数を 10 から 50，100 へと大きくすると到達解の精度が向上するが，収束速度が低下することが分かる．また，図 2 から，個体数を大きくすると標準

表 3　最良個体の最終到達解の平均値
Table 3　Average values of final solutions.

	騙し問題	ナップザック問題	グラフ分割問題
	個体数 10 個体数 50 個体数 100	個体数 10 個体数 50 個体数 100	個体数 10 個体数 50 個体数 100
SGA	296.52 300.00 300.00	15923.3 15996.4 15962.2	−186.33 −185.82 −185.14
MGG	296.04 300.00 300.00	16022.0 16140.4 16153.6	−188.56 −186.62 −184.18
SSE	297.44 300.00 300.00	16081.8 16140.8 16146.7	−188.16 −185.24 −183.91
c1	298.24 300.00 300.00	16107.1 16150.2 16153.5	−186.24 −183.30 −182.78
c2	298.92 300.00 300.00	16094.2 16149.5 16154.0	−186.30 −183.22 −182.75
c3	297.32 300.00 300.00	16066.3 16135.7 16134.3	−187.52 −184.60 −183.36
c4	297.84 299.56 299.12	16074.4 16121.3 16128.5	−187.71 −184.02 −183.21
c5	298.12 300.00 300.00	16090.9 16142.9 16147.7	−188.00 −185.06 −184.69
c6	297.64 300.00 300.00	16103.5 16148.1 16153.7	−185.98 −183.02 −183.42
c7	291.40 290.52 290.80	15738.6 15644.9 15663.4	−188.27 −190.10 −189.63

表 4　最良個体の最終到達解の標準偏差値
Table 4　Standard deviation of final solutions.

	騙し問題	ナップザック問題	グラフ分割問題
	個体数 10 個体数 50 個体数 100	個体数 10 個体数 50 個体数 100	個体数 10 個体数 50 個体数 100
SGA	2.58 0.00 0.00	46.8 35.7 34.7	3.98 5.04 4.94
MGG	2.38 0.00 0.00	29.5 7.3 1.7	4.29 4.65 3.34
SSE	3.12 0.00 0.00	17.6 5.6 4.4	5.70 4.43 3.00
c1	2.81 0.00 0.00	13.5 2.7 1.4	4.88 3.59 3.11
c2	2.30 0.00 0.00	14.6 3.0 1.0	5.29 3.47 2.95
c3	2.73 0.00 0.00	17.5 7.8 7.6	4.97 3.66 3.73
c4	2.11 0.82 1.21	19.8 10.6 7.0	4.64 3.33 3.34
c5	2.81 0.00 0.00	17.8 4.8 3.7	5.67 4.09 4.05
c6	3.11 0.00 0.00	15.0 4.6 1.6	4.90 4.25 3.69
c7	2.00 2.18 2.33	82.4 79.1 82.7	5.24 6.03 6.04

偏差値がより小さい値へ収束し，初期集団のとり方に依存しにくい安定した解探索を実現していることが分かる．

SSE，c1 による最良個体の到達解の平均値のグラフを**図 3** に，標準偏差値のグラフを**図 4** に示す．図 3 から，個体数を大きくすると SSE，c1 ともに収束速度と到達解の精度が向上することが確認できる．また，個体数が 50，100 のとき，c1 は SSE に比べて収束速度が優れていることが確認できる．また，図 4 から，SSE，c1 は個体数を大きくすると標準偏差値がより速く小さい値へ収束し，初期集団のとり方に依存しにくい安定した解探索を実現していることが分かる．

SGA，MGG，SSE，c1 による最良個体の到達解の平均値のグラフを**図 5** に，標準偏差値のグラフを**図 6** に示す．各アルゴリズムの個体数は，10，50，100 を比較して最も良い性能を示した個体数を用いている．図 5 から，SGA，MGG に比べ SSE，c1 は収束速度が速いことが確認できる．特に，c1 は最も良い収束速度を示していることが分かる．また，図 6 から，MGG は他の手法に比べて標準偏差値の値の収束が遅いことが分かる．このことは，他の手法に比べて，MGG は初期集団のとり方により到達解にばらつきが生じることを示している．

4.2.2.2　ナップザック問題

MGG による最良個体の到達解の平均値のグラフを**図 7** に，標準偏差値のグラフを**図 8** に示す．図 7 から，騙し問題と同様に個体数を大きくした場合に収束速度が低下していることが分かる．図 8 から，個体数を大きくすると標準偏差値がより小さい値へ収束し，より安定した解探索を実現していることが確認できる．

SSE，c1 による最良個体の到達解の平均値のグラフを**図 9** に，標準偏差値のグラフを**図 10** に示す．図 9 から，個体数を大きくすると SSE，c1 ともに収束速度と到達解の精度が向上することが確認できる．さらに，各個体数において，c1 が SSE に比べ収束速度，到達解ともに優れていることが確認できる．特に，個体数

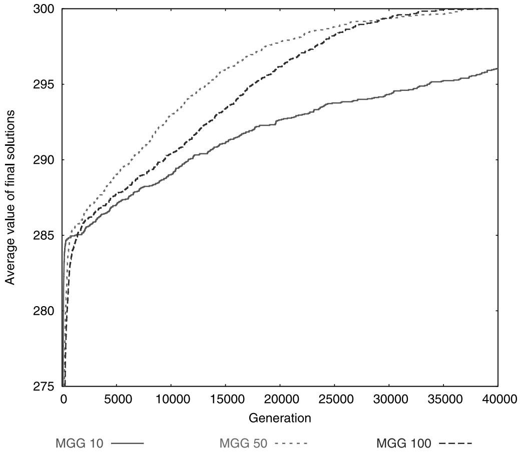

図 1 騙し問題における MGG の最良個体の到達解の平均値
Fig. 1 Average values of solutions of MGG on deception problem.

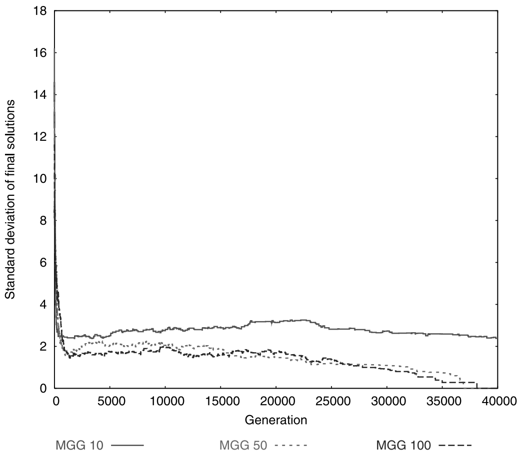

図 2 騙し問題における MGG の最良個体の到達解の標準偏差値
Fig. 2 Standard deviation of solutions of MGG on deception problem.

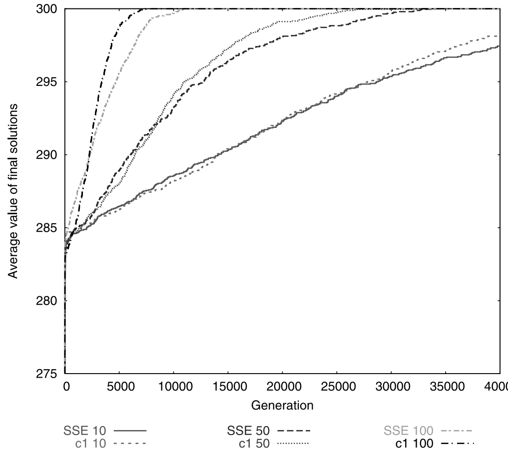

図 3 騙し問題における SSE，c1 の最良個体の到達解の平均値
Fig. 3 Average values of solutions of SSE and c1 on deception problem.

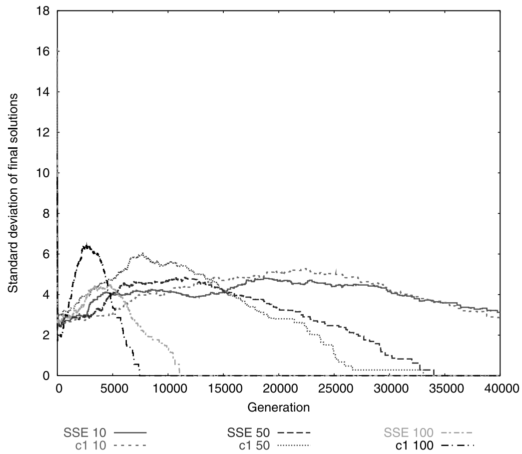

図 4 騙し問題における SSE，c1 の最良個体の到達解の標準偏差値
Fig. 4 Standard deviation of solutions of SSE and c1 on deception problem.

50 の c1 の方が個体数 100 の SSE より良い性能を示していることが分かる．また，図 10 から，個体数を大きくすると標準偏差値がより小さい値へ収束し，より安定した解探索を実現していることが分かる．さらに，各個体数において，c1 は SSE に比べより小さい値へ収束していることが確認でき，個体数 50 の c1 の方が個体数 100 の SSE より良い性能を示していることが分かる．

SGA，MGG，SSE，c1 による最良個体の到達解の平均値のグラフを**図 11** に，標準偏差値のグラフを**図 12** に示す．各アルゴリズムの個体数は，10，50，100 を比較して最も良い性能を示した個体数を用いている．図 11 から，騙し問題と同様に SGA，MGG に比べ SSE，c1 は収束速度が速いことが確認できる．さらに，c1 は MGG よりも良い収束特性で MGG と同等の性能の到達解を実現していることが分かる（表 3）．また，図 12 から，SGA，MGG に比べて SSE，c1 は標準偏差値がより速く，より小さい値へ収束しており，SSE，c1 は安定した解探索を実現していることが確認できる．特に，c1 は他の方法に比べて解探索が最も良く安定していることが分かる（表 4）．

4.2.2.3 グラフ分割問題

MGG による最良個体の到達解の平均値のグラフを**図 13** に，標準偏差値のグラフを**図 14** に示す．図 13 から，騙し問題，ナップザック問題と同様に個体数を大きくした場合に収束速度が低下することが確認できる．また，図 14 から，個体数が 50 のときは初期世代から，個体数が 100 のときは約 500 世代から標準偏

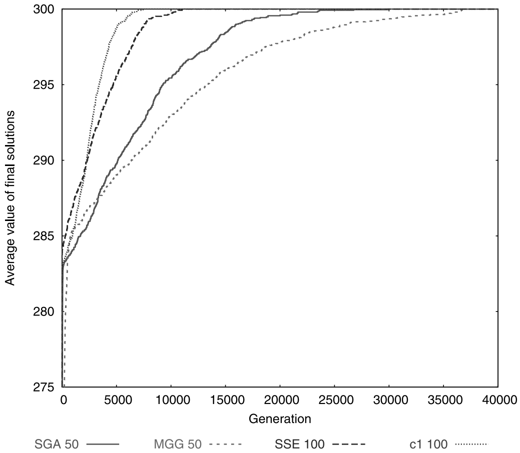

図 5 騙し問題における SGA，MGG，SSE，c1 の最良個体の到達解の平均値

Fig. 5 Average values of solutions of SGA, MGG, SSE and c1 on deception problem.

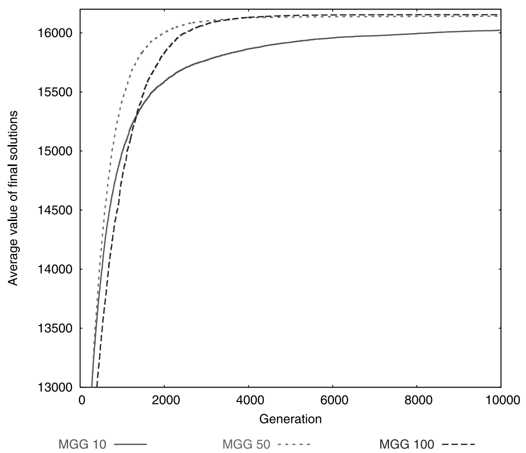

図 7 ナップザック問題における MGG の最良個体の到達解の平均値

Fig. 7 Average values of solutions of MGG on knapsack problem.

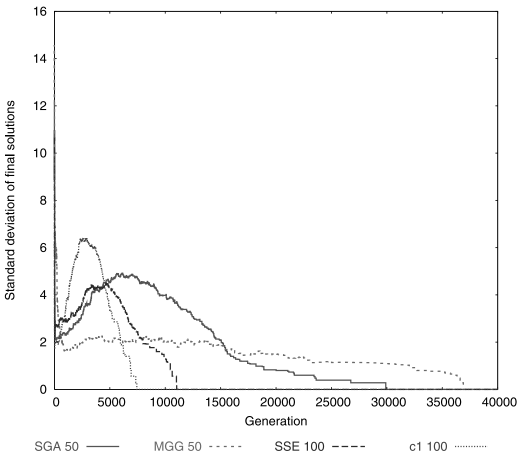

図 6 騙し問題における SGA，MGG，SSE，c1 の最良個体の到達解の標準偏差値

Fig. 6 Standard deviation of solutions of SGA, MGG, SSE and c1 on deception problem.

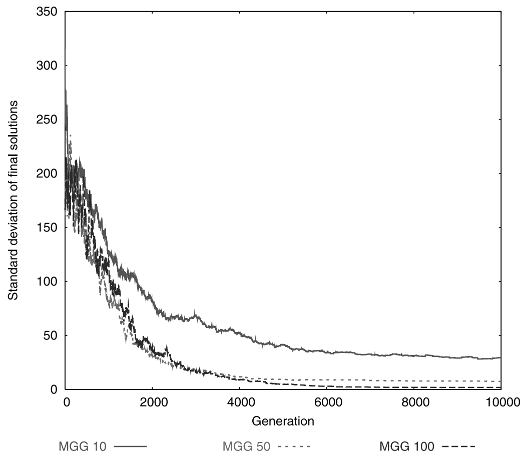

図 8 ナップザック問題における MGG の最良個体の到達解の標準偏差値

Fig. 8 Standard deviation of solutions of MGG on knapsack problem.

差値の値が増加していることが分かる．このことは，初期集団のとり方により到達解にばらつきが生じることを示している．

SSE，c1 による最良個体の到達解の平均値のグラフを**図 15** に，標準偏差値のグラフを**図 16** に示す．図 15 から，ナップザック問題と同様に個体数を大きくすると SSE，c1 ともに収束速度と到達解の精度が向上することが確認できる．さらに，各個体数において，c1 が SSE に比べ収束速度，到達解ともに優れていることが確認できる．特に，個体数 50 の c1 は個体数 100 の SSE より良い性能を示している．また，図 16 から，ナップザック問題と同様に個体数を大きくすると標準偏差値がより小さい値へ収束し，より安定した解探索を実現していることが分かる．

SGA，MGG，SSE，c1 による最良個体の到達解の平均値のグラフを**図 17** に，標準偏差値のグラフを**図 18** に示す．各アルゴリズムの個体数は，10，50，100 を比較して最も良い性能を示した個体数を用いている．図 17 から，騙し問題，ナップザック問題と同様に MGG に比べ SSE，c1 は収束速度が速いことが確認できる．特に，c1 は最も良い到達解を実現している（表 3）．また，図 18 から，騙し問題，ナップザック問題と同様に，SGA，MGG に比べて SSE，c1 は標準偏差値がより速く，より小さい値へ収束しており，

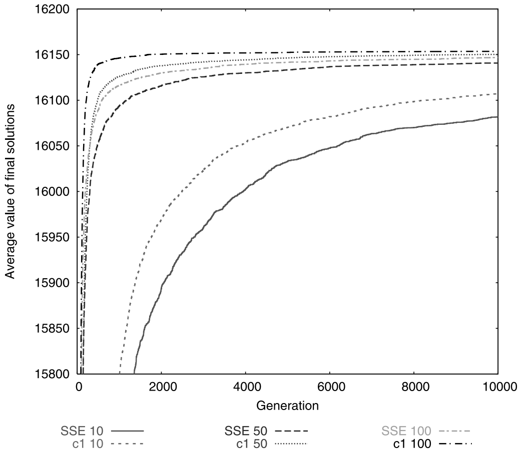

図 9 ナップザック問題における SSE，c1 の最良個体の到達解の平均値

Fig. 9 Average values of solutions of SSE and c1 on knapsack problem.

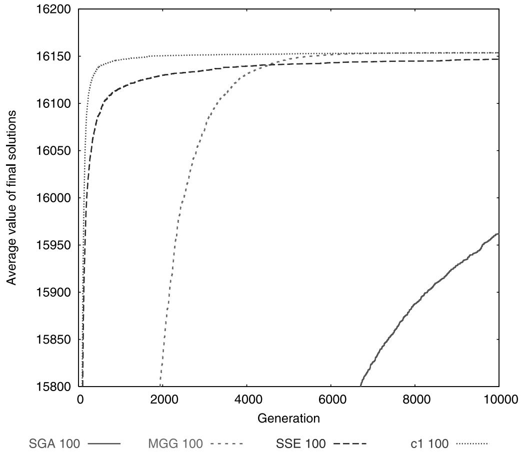

図 11 ナップザック問題における SGA，MGG，SSE，c1 の最良個体の到達解の平均値

Fig. 11 Average values of solutions of SGA, MGG, SSE and c1 on knapsack problem.

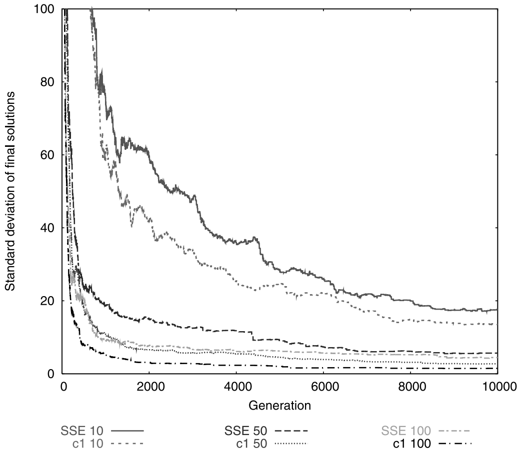

図 10 ナップザック問題における MGG の最良個体の到達解の標準偏差値

Fig. 10 Standard deviation of solutions of SSE and c1 on knapsack problem.

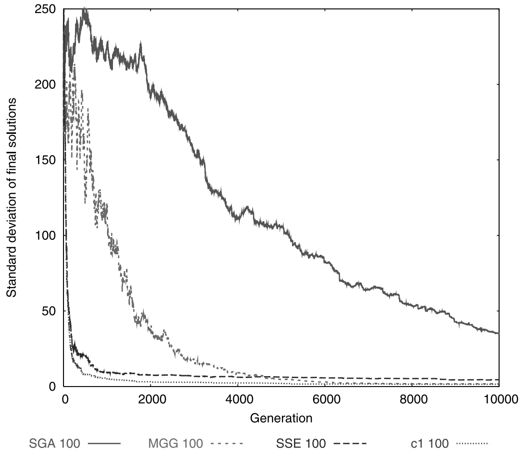

図 12 ナップザック問題における SGA，MGG，SSE，c1 の最良個体の到達解の標準偏差値

Fig. 12 Standard deviation of solutions of SGA, MGG, SSE and c1 knapsack problem.

SSE，c1 は初期集団のとり方によらず安定した探索性能を実現していることが分かる．

5. 性能比較 2

本章では，BOA と SSE，ESSE（c1）の性能比較を行う．

5.1 テスト問題

HIFF 問題，H-Trap 問題は，GA のスキーマ処理と合成を調べる階層型問題である．H-Trap 問題は，HIFF 問題に騙し性のある関数を加えることで，問題をより難しくしている．ここでは，以下に述べる HIFF 問題[19]，H-Trap 問題[20] の最大化問題を用いて性能比較をする．

- HIFF 問題

HIFF（The Hierarchical-if-and-only-if）問題とは，0 と 1 の間に IFF（If and Only If）の関係があり，二分木の階層構造を持つ問題である．各ノードは，子ノードが両方とも ‘0’ であれば ‘0’，両方とも ‘1’ であれば ‘1’，それ以外の場合は ‘-(null)’ のシンボルをとり，シンボルが ‘0’ もしくは ‘1’ のときに適合度に部分値を加える．0/1 のストリングが木の葉として入力され，各ノードの部分値を足した値が適合度となる．子ノード $(b_1, \cdots, b_k)$ を持つ親ノードを B，ノード B 以

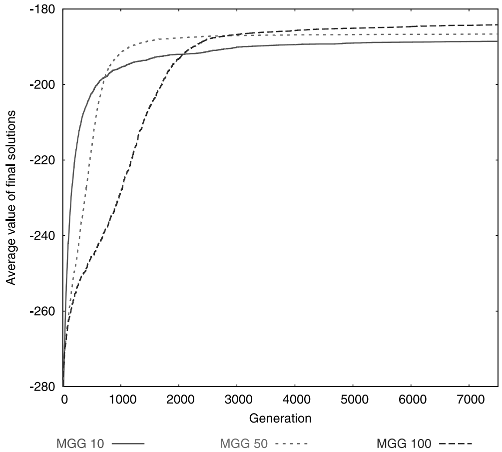

図 13 グラフ分割問題における MGG の最良個体の到達解の平均値

Fig. 13 Average values of solutions of MGG on graph partitioning problem.

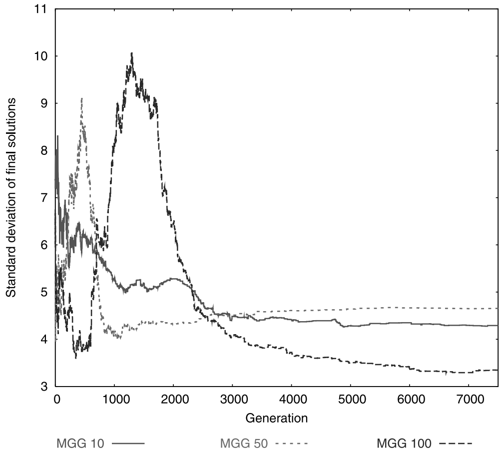

図 14 グラフ分割問題における MGG の最良個体の到達解の標準偏差値

Fig. 14 Standard deviation of solutions of MGG on graph partitioning problem.

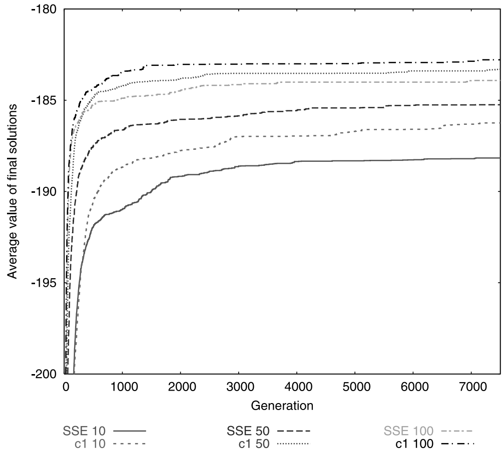

図 15 グラフ分割問題における SSE，c1 の最良個体の到達解の平均値

Fig. 15 Average values of solutions of SSE and c1 on graph partitioning problem.

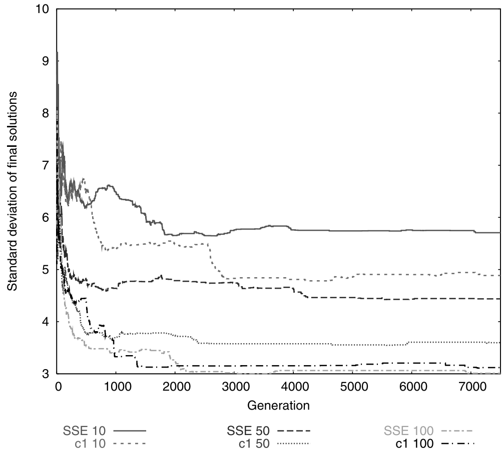

図 16 グラフ分割問題における SSE，c1 の最良個体の到達解の標準偏差値

Fig. 16 Standard deviation of solutions of SSE and c1 on graph partitioning problem.

下に存在する葉の総数を $|B|$ と表すと，HIFF 問題は以下の式で定義される．

$$t(a,b) = \begin{cases} 0 & \text{if a=0 and b=0} \\ 1 & \text{if a=1 and b=1} \\ null & \text{otherwise} \end{cases} \quad (4)$$

$$f(a) = \begin{cases} 1 & \text{if a=1 or a=0} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$T(B) = \begin{cases} B & \text{if } |B| = 1 \\ t(T(b_1), \cdots, T(b_k)) & \text{otherwise} \end{cases} \quad (6)$$

$$F(B) = \begin{cases} f(B) & \text{if } |B| = 1 \\ |B| f(T(B)) + \sum_{i=1}^{k} F(b_i) & \text{otherwise} \end{cases} \quad (7)$$

式 (4) は親ノードがどの状態をとるか返す関数であり，式 (5) は各ノードの部分適合度関数である．式 (6) は子ノードを展開する関数であり，式 (7) は HIFF 問題の適合度関数である．本研究では，$k = 2$ とした 16 ビットの HIFF 問題[19] を部分解として 10 個足し合わせた関数を適合度関

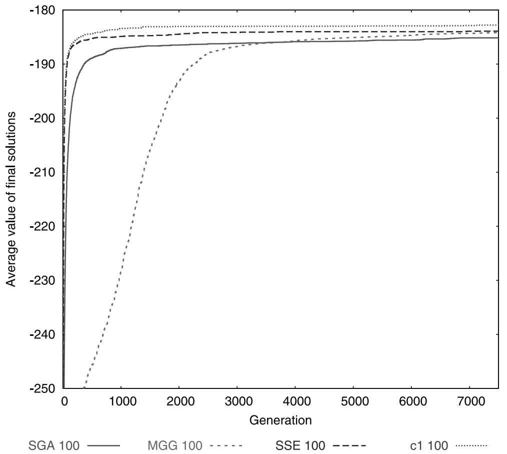

図 17 グラフ分割問題における SGA，MGG，SSE，c1 の最良個体の到達解の平均値
Fig. 17 Average values of solutions of SGA, MGG, SSE and c1 on graph partitioning problem.

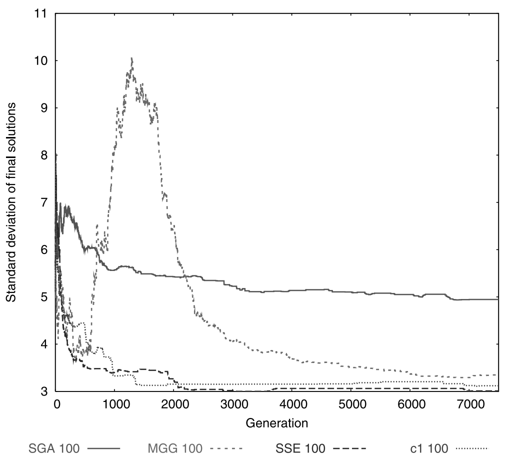

図 18 グラフ分割問題における SGA，MGG，SSE，c1 の最良個体の到達解の標準偏差値
Fig. 18 Standard deviation of solutions of SGA, MGG, SSE and c1 on graph partitioning problem.

数とする．

- H-Trap 問題

H-Trap 問題は，HIFF 問題の枝の数を $k \geq 3$ とし，さらに，葉以外の各ノードの部分適合度関数に騙し性のある関数を用いた問題である．u を子ノードにあるシンボル‘1’の数とし，k を子ノードの総数とすると，H-Trap 問題で用いる部分適合度関数は以下の式 (8) で与えられる．

$$f_{trap}(u) = \begin{cases} f_{max} & u = k \\ f_{min} - u \cdot \dfrac{f_{min}}{k-1} & u \neq k \end{cases} \tag{8}$$

表 5 最良個体の最終到達解の平均値
Table 5 Average values of final solutions.

		HIFF 問題	H-Trap 問題
	（個体数 20）	445.60	157.448
BOA	（個体数 100）	619.00	171.000
	（個体数 200）	644.56	171.000
	（個体数 10）	647.12	171.144
SSE	（個体数 50）	695.92	171.090
	（個体数 100）	729.92	171.054
	（個体数 10）	470.96	171.090
c1	（個体数 50）	592.32	171.090
	（個体数 100）	617.68	171.036

表 6 最良個体の最終到達解の標準偏差値
Table 6 Standard deviation of final solutions.

		HIFF 問題	H-Trap 問題
	（個体数 20）	27.14	3.260
BOA	（個体数 100）	38.92	0.000
	（個体数 200）	29.95	0.000
	（個体数 10）	38.50	0.329
SSE	（個体数 50）	35.53	0.270
	（個体数 100）	30.21	0.213
	（個体数 10）	27.42	0.324
c1	（個体数 50）	39.96	0.270
	（個体数 100）	32.85	0.176

本研究では，$k = 3$，ルートのノードでは $f_{min} = 0.9$，$f_{max} = 1$，それ以外のノードでは $f_{min} = f_{max} = 1$ とした 9 ビットの H-Trap 問題（最適解 18.0，準最適解 17.1）[20] を部分解として 10 個足し合わせた関数を適合度関数とする．

5.2 実験結果

BOA のパラメータにおいて，親個体を個体集団の 50%，子個体を個体集団の 50%とする．各アルゴリズムともに 1 世代あたりの適合度評価による計算負荷を同じとすることで，実行世代数で評価することが可能となる．そこで，BOA では個体数を 20，100，200 ととり，SSE，c1 においては個体数を 10，50，100 ととる．また，実行世代数は，HIFF 問題では 1500，H-Trap 問題では 500 とし，各アルゴリズムを異なる初期個体集団から 50 回実行する．

各アルゴリズムによる最良個体の最終到達解の平均値を**表 5** に，最良個体の最終到達解の標準偏差値を**表 6** に示す．

また，以下の図 19〜22 において，横軸に世代数をとり，縦軸に最良個体の到達解の平均値，もしくは，最良個体の到達解の標準偏差値をとる．また，アルゴリズム名の後ろの数値は集団を構成する個体数を示している．

5.2.1 HIFF 問題

BOA，SSE，c1 による最良個体の到達解の平均値

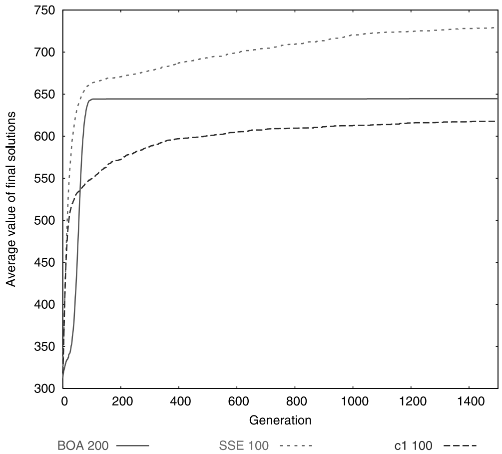

図 19　HIFF 問題における BOA，SSE，c1 の最良個体の到達解の平均値

Fig. 19　Average values of solutions of BOA, SSE and c1 on HIFF problem.

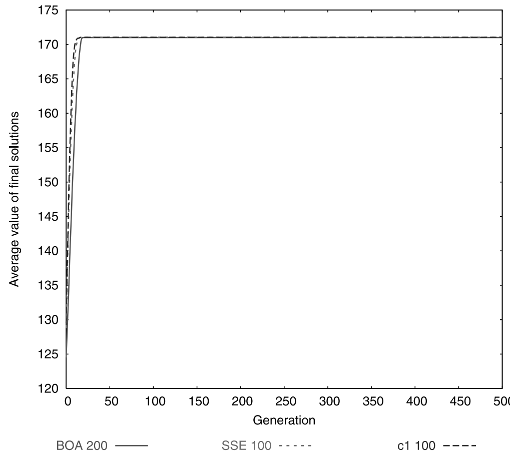

図 21　H-Trap 問題における BOA，SSE，c1 の最良個体の到達解の平均値

Fig. 21　Average values of solutions of BOA, SSE and c1 on H-Trap problem.

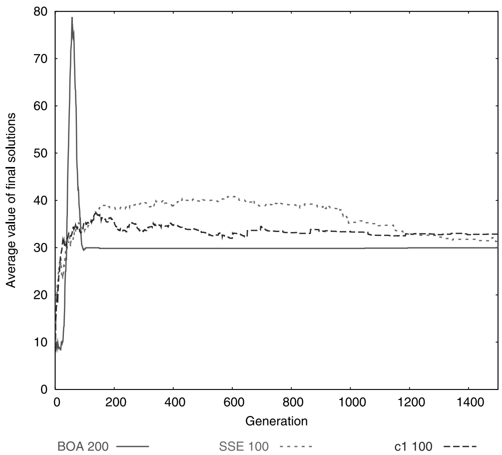

図 20　HIFF 問題における BOA，SSE，c1 の最良個体の到達解の標準偏差値

Fig. 20　Standard deviation of solutions of BOA, SSE and c1 on HIFF problem.

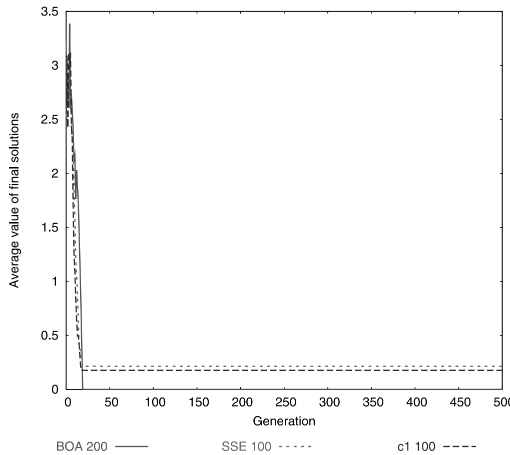

図 22　H-Trap 問題における BOA，SSE，c1 の最良個体の到達解の標準偏差値

Fig. 22　Standard deviation of solutions of BOA, SSE and c1 on H-Trap problem.

のグラフを**図 19** に，標準偏差値のグラフを**図 20** に示す．各アルゴリズムの個体数は，最も良い値を示した個体数として，BOA は 200，SSE，c1 は 100 を用いている．図 19 から，速やかに収束してから解の改善が起きない BOA に対して，SSE は収束速度，到達解ともに良い性能を示している．さらに，表 5 から，個体数が 10 のときの SSE は，個体数が 200 の BOA，100 の c1 よりも良い到達解を示していることが分かる．また，各個体数において，c1 は SSE よりも解探索性能が低下していることが分かる．HIFF は 0 と 1 に IFF の関係があり，多くの遺伝子座で 0 と 1 が対になる個体どうしは近い適合度を持つ．このような 2 つの個体からは次世代の個体を特徴付けるスキーマが抽出できない．さらに，c1 の多様性維持は，次世代の個体集団を形成するスキーマ群が 0 もしくは 1 のどちらかに偏ることを防ぐ．以上より，c1 は SSE と比べて，0 もしくは 1 のどちらかが多数を占める局所解に速く収束できない．また，表 6，図 20 から，各探索手法において，初期集団のとり方による到達解のばらつきが同程度であることが分かる．

5.2.2　H-Trap 問題

BOA，SSE，c1 による最良個体の到達解の平均値のグラフを**図 21** に，標準偏差値のグラフを**図 22** に示す．各アルゴリズムの個体数は，BOA は 200，SSE， c1 は 100 を用いている．表 5，図 21 から，各探索手法ともに，収束速度，到達解ともに同程度であることが分かる．ただし，BOA は個体数が 20 の場合において，最終到達解の値が小さいことが分かる．また，表 6，図 22 から，各探索手法において，初期集団のとり方による到達解のばらつきが同程度であることが分かる．

6. ま と め

本論文では，スキーマの関係に着目して SSE を改良した ESSE を提案した．そして，0/1 組合せ最適化問題において，SGA，MGG，BOA，SSE，ESSE の性能比較を行った．

ESSE ではソーティングの過程で，個体部分集合から抽出されるスキーマを比較し，両者が同一の場合，包含関係にある場合，部分的に一致する場合にそれぞれに対応した ESSE 処理 1〜3 を行うことで SSE の性能を改善するアルゴリズムであり，7 通りの組合せが存在する．

実験の結果，ESSE の処理 1〜3 の適切な組合せは適用する問題と個体数の設定に依存しているが，その中で，処理 1 だけを含んだ c1 の ESSE がすべての問題において，収束速度と最終到達解の性能の両方において良い性能を示すことを確認した．

ESSE 処理 1 は，個体部分集合から抽出されたスキーマのうち重複したスキーマを排除することで探索性能を下げることなく，多様性を維持する．ESSE 処理 2 は，スキーマが成長し大きくなる過程で，成長により平均評価値が下がったスキーマを排除することで，解性能を低下させるような探索方向を排除する．ESSE 処理 3 は，2 つのスキーマのうちの平均評価値が高い方と 2 つのスキーマの共通スキーマを用いて探索することで，近傍探索の効果が期待できる．つまり，ESSE 処理 1 は，SSE の収束特性を下げることなく探索性能を改善しているが，局所解に収束した後の解の改善はあまりみられない．ESSE 処理 2，3 は，局所解から脱出できる可能性があるが，そのために SSE の収束特性を犠牲にしている．これらのことから，問題によっては ESSE 処理 2，3 が良い結果を示すこともあるが，全体として収束特性と探索性能の両者をみると ESSE 処理 1 だけを有する c1 が良い性能を示したと想像される．

SSE，c1 の ESSE と MGG の性能比較を行った結果以下のことが分かった．SSE，c1 の ESSE は，個体数を大きくすると収束速度と到達解の精度が同時に向上する．これに対して，MGG では，個体数を増やすと到達解の精度は向上する一方で，収束速度が低下する場合がみられる．つまり，MGG では到達解の精度と収束速度の両方にとって適切な個体数の設定が難しい．SSE，c1 の ESSE では収束速度を犠牲にしない範囲で十分多くの個体数をとればよいので，MGG に比べて個体数の設定が容易であるといえる．また，c1 の ESSE は，多様性を維持することで優れた収束特性と安定した解探索能力を失わずに，MGG と同等の大域的探索が実現できることが分かった．

次に，SSE，c1 と BOA の性能比較を行った結果以下のことが分かった．HIFF 問題において，SSE は，BOA より収束速度，到達解ともに良い性能を示しており，個体数が小さい場合においても良い到達解を示している．また，IFF の関係がある問題では，c1 のスキーマの多様性維持が収束速度を低下させることが分かった．また，BOA は，個体集団のサイズが小さいと個体分布状況の確率モデルの推定値が十分に高まらず，SSE に比べて良い解探索ができない．さらに，BOA は，多様性維持や突然変異がないので，収束してしまうと解の改善が起こらないといえる．H-Trap 問題において，BOA，SSE，c1 は同程度の解探索性能を示していることが分かった．また，HIFF 問題，H-Trap 問題において，標準偏差値の結果から，BOA，SSE，c1 ともに，解探索の安定性は同程度であることが分かった．

最後に，実問題への対応を含めて，少し ESSE の特徴について述べておく．ここで示した解析例より，ESSE は SSE と同じく高速で安定した収束特性を有していることが分かる．しかし，反面解こうとする問題によっては大域的最適解に到達することが容易ではないことや，大域的最適解を得るためにかなり多数の個体を必要とする場合があることが予想される．このような特徴から，必ずしも最適解でなくても，準最適解を得ることができれば実用的には十分であるような問題の解析に対して特に有効なアルゴリズムであると思われる．

謝辞 本研究を遂行するにあたり，名古屋大学 21 世紀 COE プログラム「計算科学フロンティア」から援助をいただいた．ここに記して謝意を表する．

参 考 文 献

1) 吉澤大樹，坂野 鋭，橋本周司：最適化のための粗視化ニュートン法，情報処理学会数理モデル化と問題解決，Vol.2003, No.020, pp.21–24 (2003).
2) 柳浦睦憲，茨木俊秀：組合せ最適化—メタ戦略を中心として，朝倉書店 (2001).

3) 相澤彰子：スキーマ処理に基づく集団型探索アルゴリズム，情報処理学会研究報告「人工知能」，Vol.1994, No.093, pp.1–8 (1994).
4) 相澤彰子：スキーマ処理に基づく集団型探索アルゴリズムの構成，電子情報通信学会論文誌 D-II, Vol.J78-D-II, No.1, pp.94–104 (1995).
5) Holland, J.H.: *Adaptation in Natural and Artificial Systems*, The University of Michigan Press (1975).
6) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley (1989).
7) Baker, J.E.: Reducing bias and inefficiency in the selection algorithm, *Proc. 2nd International Conference on Genetic Algorithms*, pp.14–21 (1987).
8) Whitley, D.: The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best, *Proc. 3rd International Conference of Genetic Algorithms*, pp.116–121 (1989).
9) 佐藤 浩，小野 功，小林重信：遺伝的アルゴリズムにおける世代交代モデルの提案と評価，人工知能学会誌，Vol.12, No.5, pp.734–743 (1996).
10) Ono, I., Kobayashi, S. and Yoshida, K.: Global and multi-objective optimization for lens design by real-coded genetic algorithms, *International Optical Design Conference*, Vol.3482, pp.110–121 (1998).
11) Takahashi, O., Kita, H. and Kobayashi, S.: Protein folding by a hierarchical genetic algorithm, *The 4th International Symposium on Artificial Life and Robotics*, pp.334–339 (1999).
12) 小野 功，今出広明，中田秀基，小野典彦，松岡聡，関口智嗣，楯 真一：蛋白質立体構造の進化的解析のための Ninf 版並列 MGG とその性能評価，情報処理学会研究報告「ハイパフォーマンスコンピューティング」，Vol.2003, No.93, pp.149–154 (2003).
13) Pelikan, M., Goldberg, D.E. and Cantu-Paz, E.: Boa: The bayesian optimization algorithm, *Proc. Genetic and Evolutionary Computation Conference 1999* (*GECCO-1999*), San Fransisco, CA, Banzhaf, W., Daida, J., Elben, A.E., Garzon, M.H., Honavar, V., Jakiela, M. and Smith, R.E. (Eds), pp.525–532, Morgan Kaufmann (1999).
14) 山村雅幸，小林重信：遺伝的アルゴリズムの工学的応用，人工知能学会誌，Vol.9, No.4, pp.506–511 (1994).
15) Whitley, L.D.: Fundamental principles of deception in genetic search, *Foundations of Genetic Algorithms 1991* (*FOGA 1*), Rawlins, G.J.E. (Ed), pp.221–241, Morgan Kaufmann (1991).
16) Jaszkiewicz, A.: On the performance of multiple objective genetic local search on the 0/1 knapsack problem—A comparative experiment, *IEEE Trans. Evolutionary Computation*, Vol.6, pp.402–412 (2002).
17) Johnson, D.S., Aragon, C.R., McGeoch, L.A. and Schevon, C.: Optimization by simulated annealing: An experimental evaluation: Part i, graph partitioning, *Operations Research*, Vol.37, pp.865–892 (1989).
18) 高橋 治，木村周平，小林重信：交叉的突然変異による適応的近傍探索—騙しのある多峰性関数の最適化，人工知能学会誌，Vol.16, No.2, pp.175–184 (2001).
19) Watson, R.A. and Pollack, J.B.: Hierarchically-consistent test problems for genetic algorithms, *Proc. 1999 Congress on Evolutionary Computation* (*CEC 99*), Angeline, P.J., Michalewicz, Z., Schoenauer, M., Yao, X. and Zalzala, A. (Eds), pp.1406–1413, IEEE (1999).
20) Pelikan, M. and Goldberg, D.E.: Escaping hierarchical traps with competent genetic algorithms, *Proc. Genetic and Evolutionary Computation Conference 2001* (*GECCO-2001*), San Fransisco, CA, Spector, L., Goodman, E.D., Wu, A., Langdon, W.B., Voigt, H.M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M.H. and Burke, E. (Eds), pp.511–518, Morgan Kaufmann (2001).

(平成 17 年 4 月 12 日受付)
(平成 17 年 6 月 6 日再受付)
(平成 17 年 7 月 7 日再々受付)
(平成 17 年 7 月 17 日採録)

丸山 崇（学生会員）

1978 年生．名古屋大学大学院情報科学研究科博士課程後期課程在学中．遺伝的アルゴリズム等の進化的計算手法の性能向上と実問題への応用について研究している．

北　　栄輔（正会員）

1964 年生．1991 年名古屋大学大学院工学研究科博士課程後期課程修了．博士（工学）．1999 年より名古屋大学助教授，現在に至る．数値解析法（BEM，Trefftz 法），セルオートマトン（Cellular Automata）等の研究に従事．著書に，『偏微分方程式の数値解法』，『計算のための線形代数』，『Trefftz 法入門』等．IEEE，ISBE，応用数理学会，日本機械学会，シミュレーション学会，日本計算工学会等各会員．