

SSEの世代交代モデルを改良したcSSEについて

丸山 崇[†] 北 栄輔[†]

組合せ最適化問題を効率良く解く進化的計算手法の1つに、確率的スキーマ貪欲法 (Stochastic Schemata Exploiter: SSE) がある。SSEは、母集団に多様性を持たせることで、優れた収束特性を維持したまま大域的探索能力が向上する可能性がある。そこで、本論文ではSSEの世代交代モデルを改良した Cross generational elitist selection SSE (cSSE) を提案する。さらに、0/1 組合せ最適化問題において、SSE, cSSE を Minimal Generation Gap (MGG) に基づく GA, 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。その結果、cSSE は優れた収束特性と大域的探索能力を有していることが分かった。

A Study on cSSE that Improves Generation Alternation Model of SSE

TAKASHI MARUYAMA[†] and EISUKE KITA[†]

The Stochastic Schemata Exploiter (SSE) is one of the evolutionary optimization algorithms for solving the combinatorial optimization problems. The SSE can improve the global search ability by maintaining the diversity of the population. In this paper, we present the Cross generational elitist selection SSE (cSSE) algorithms which improves the generation alternation model of the SSE. The SSE and the cSSE are compared with the GA 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. As a result, we indicate that cSSE has an excellent convergence property and the global search ability.

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、「良い解どうしは似通った構造を持つ」という大谷構造 (big valley structure) の特徴を持っている¹⁾。このような概念は、Proximate Optimality Principle (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、複数の最適化手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。このような問題において、効率良く近似解を得る手法として進化的計算手法が研究されており、その1つに、相澤ら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、適合度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であり、POPの方針を持つ

探索手法である。よって、SSEは大谷構造の特徴を持つ組合せ最適化問題に適した解探索手法であると考えられる。

SSEのアルゴリズムでは、評価の良いスキーマを母集団中に早く行きわたらせることになる。これが、早い初期収束特性につながる一方で、個体の多様性が早く失われるので、大域的探索性能を低下させる。したがって、SSEにおいて、母集団に多様性を持たせることで、大域的探索能力が向上する可能性がある。そこで、以前の研究において Extended SSE (ESSE) を提案した^{5),6)}。この方法では、次世代の個体を生成するためのスキーマ群から同一、または類似のスキーマを削除することで、母集団のスキーマに多様性を持たせ大域的探索能力の向上を実現した。

これに対して、本研究では、SSEの世代交代モデルを改良した Cross generational elitist selection SSE (cSSE) を提案する。cSSEは、母集団から同一の個体を排除し個体の多様性を保ちながら世代交代を行うことで、SSEの優れた収束特性を維持したまま大域的探索能力の向上を実現する。個体部分集合からスキーマを抽出し、それをソートする際にスキーマの多様性を持たせる ESSE に対して、cSSEは、生成され

[†] 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

た個体と母集団の個体を比較し、同一の個体を排除することで母集団の個体の多様性を維持する。また、適合度値の降順に従ってソートすることで、今までに探索された真に良い個体のみから次世代の母集団を生成する。さらに本研究では、0/1 組合せ最適化問題において、SSE, cSSE を実問題に有効とされる Minimal Generation Gap (MGG) に基づく GA, SSE と類似の設計方針を持つ Bayesian Optimization Algorithm (BOA) と性能比較を行い、cSSE の有効性を示す。

本論文の構成は以下のようになっている。2 章において、研究背景と、性能比較で用いる GA (MGG), BOA, SSE, cSSE の概要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 cSSE の概要とアルゴリズムの詳細について述べる。さらに、4 章において、0/1 組合せ最適化問題を用いて GA (MGG), BOA, SSE, cSSE の探索性能を比較する。最後に、5 章において、本論文全体のまとめを述べる。

2. 研究の背景

本章では、提案する Cross generational elitist selection SSE (cSSE) と、その基本となる SSE, さらに比較対象となる GA (MGG), BOA について簡単に述べる。

2.1 GA (MGG)

Holland によって考案された遺伝的アルゴリズム (Genetic Algorithm: GA)⁷⁾ 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。GA には、さまざまな拡張手法が提案されており、その中の 1 つに佐藤ら⁸⁾ が提案した Minimal Generation Gap (MGG) がある。MGG は、GA の世代交代モデルを改良することで、初期収束 (early convergence) と進化的停滞 (evolutionary stagnation) という問題^{9),10)} を解決したアルゴリズムである。MGG は、レンズ設計、タンパク質の構造解析、またその並列処理化など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¹¹⁾⁻¹³⁾。

2.2 BOA

優良解集団の情報から確率モデルによって良い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。EDA では、交叉、突然変異という操作は行われずに、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 の設計方針は、後述すると SSE と類似している。そして、EDA の

1 つに Bayesian Optimization Algorithm (BOA)¹⁴⁾ がある。BOA は、ベイジアンネットワークを用いて個体分布を評価し、優良解を探索するアルゴリズムである。

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、局所的探索処理を重視しており、優れた初期収束特性を持つ。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。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。そして、良い個体部分集合から抽出したスキーマから次世代の個体を生成していく。良いスキーマが早く母集団中に広がるので、その結果、非常に優れた初期収束特性を示す。その一方で、個体の多様性が早く失われることになる。

2.4 cSSE

SSE は、GA と同様に複数の個体を用いた多点探索であり、解が収束するにつれて同じ個体が母集団中に多く分布する。しかし、SSE は、母集団に多様性を持たせることで、優れた収束特性を維持したまま大域的探索能力が向上する可能性がある。そこで、本研究では、SSE の世代交代モデルを改良した cSSE を提案する。cSSE は、母集団から同一の個体を排除し多様性を保ちながら世代交代を行う。

3. SSE と cSSE のアルゴリズム

3.1 SSE とその問題点

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。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 (番号付け) を c_1, c_2, \dots, 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ックスを $L(S)$ 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。

SSE は、以上の半順序関係を用いることで、 $S = \{c_1\}$ から順に個体部分集合を生成していく (個体部

分集合の生成手順は 3.3 節のアクティブリストの生成において述べる)。

具体的には、SSE は、以下に示す手順 1, 2 を繰り返すことでより良い解を探索する。

- 手順 1 (個体部分集合の生成とスキーマの抽出)
母集団の個体を適合度の降順にソートしてインデックスを生成する。インデックスから半順序関係に基づき平均評価値の高い個体部分集合を生成する。各個体部分集合から共通のスキーマを抽出する。

- 手順 2 (個体の生成と世代交代)

手順 1 で得られたスキーマからランダムに個体を生成し、突然変異を施す。生成された個体から次世代の母集団を作る。

SSE は、適合度の高い個体を要素に持つ個体部分集合を生成し、それらの共通のスキーマから次世代の個体を生成する。SSE は、良い個体に似通った構造を持つ新たな個体を生成し、それらの個体を母集団に急速に分散させることで優れた収束特性を実現している。しかし、それゆえに、母集団の多様性が急速に低下し、GA に比べて大域的探索性能が低い。そこで、本研究では、提案する cSSE において母集団の多様化を実現し、大域的探索性能の向上を目指す。

3.2 cSSE の概要

SSE は、個体部分集合からスキーマを抽出して個体を生成し、毎世代において新しく生成した個体を母集団のすべての個体と入れ替える。SSE は、解が収束するにつれて同じ個体が母集団中に多く分布するようになる。ここで、個体部分集合の各個体における共通のスキーマを抽出する際に、スキーマを抽出するための元となる同一の個体が複数存在する必要はない。また、個体部分集合は、平均評価値の半順序関係から生成され、確率的な個体の選択による生成ではないので、母集団中における同じ個体の総数による重みが与える影響は少ないと思われる。よって、SSE において、ある個体が次世代の個体群に与える影響は、同じ個体が母集団中に分布する総数にあまり依存しないと考えられる。

以上より、母集団から同一の個体を排除して多様性を維持すれば、SSE の優れた収束特性を維持したまま大域的探索能力を向上させることができると思われる。そこで、本研究では、SSE の母集団に個体の多様性維持を導入した cSSE を提案する。なお、母集団中における個体の多様性維持は、次世代の母集団を形成する個体の選択であり、世代交代モデルの改良に対応する。本研究では、SSE の世代交代モデルを改良するために

世代間エリート選択 (cross generational elitist selection) を用いる。世代間エリート選択は、Eshelman が GA の拡張手法として提案した CHC¹⁵⁾ の世代交代モデルとして用いられており、母集団の M 個体と新しく生成した子個体集団の M 個体の合計 $2M$ 個体を適合度によりソートし、上位 M 個体を次世代の母集団とする世代交代モデルである。さらに、本研究で提案する cSSE では、ソートする過程で同一の個体を排除する操作を加えることで母集団の多様性を維持する。

生成した子個体を無条件に次世代の母集団とする SSE に対して、cSSE は、多様性を導入した世代間エリート選択を用いることで、今までに探索された真に良い個体から次世代の母集団を生成する。よって、SSE に比べて cSSE は、適合度の良い個体と似通った構造を持つ個体をより集中的に探索するアルゴリズムを目指している。

3.3 cSSE のアルゴリズム

cSSE のアルゴリズムを以下に述べる。

- (1) 初期母集団の生成
初期個体 M 個をランダムに生成する。
- (2) 個体の評価計算
各個体の適合度を評価する。
- (3) 初期個体リストの作成
個体を適合度の降順に並べて個体リストに挿入する。
- (4) アクティブリストの生成
インデックスから半順序関係に基づき平均評価値の高い個体部分集合を M 個作り、それらを登録したアクティブリストを生成する (詳細については後述する)。
- (5) 子個体の生成と世代交代
 $i = 1$ とする。
 - (a) 子個体の生成
アクティブリストの i 番目の個体部分集合から共通のスキーマを抽出し、'*'の部分にランダムに '0' か '1' を入れ個体を生成する。さらに、突然変異を施す。
 - (b) 個体の評価計算
生成した子個体の適合度を評価する。
 - (c) 世代交代
子個体と同じ設計変数を持つ個体が個体リストになれば、子個体を適合度の降順に従い個体リストに挿入する。リストからあふれた個体は捨てる。
 - (d) $i < M$ ならば $i = i + 1$ として、(a) か

表 1 個体部分集合リストの例
Table 1 Example of subset list.

Rank	Subset
1	{ c_1 }
2	{ c_1, c_2 }
3	{ c_2 }
4	{ c_1, c_2, c_3 }
5	{ c_1, c_3 }
⋮	⋮

ら (c) を繰り返す . そうでなければ終了 .

(6) 終了条件

停止条件が満たされるまで , (4) から (5) を繰り返す .

個体リストに格納されている個体の並びがインデックスに対応する . また , 個体リストは母集団における個体の保持とソートを行うリストであり , アクティブリストは個体部分集合のソートを行うリストである . アクティブリストの生成手順を次に述べる .

アクティブリストの生成

アクティブリストを生成するためには , 最初に 3.1 節で示した半順序関係に従って個体部分集合を生成する必要がある . まず , 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 (番号付け) を c_1, c_2, \dots, c_M とおく . 平均評価値の最も高い個体部分集合は , $\{c_1\}$ である . 半順序関係によって , $\{c_1\}$ から次に平均評価値の高い $\{c_1, c_2\}$ と $\{c_2\}$ が生成できる . さらに , $\{c_1, c_2\}$ からは $\{c_1, c_2, c_3\}$ と $\{c_1, c_3\}$ が生成でき , $\{c_2\}$ からは $\{c_2, c_3\}$ と $\{c_3\}$ が生成できる . このようにして生成した個体部分集合を平均評価値の降順にソートしてできたリストの例が表 1 である .

その処理を箇条書きにすると次のようになる .

- (1) リストの先頭の要素に $\{c_1\}$ を格納する . $i = 1$ とする .
- (2) リストの i 番目の要素において , 格納されている個体部分集合から , 半順序関係に従って 2 つの新しい個体部分集合を生成する .
- (3) 生成した 2 つの個体部分集合を , $i + 1$ 番目以降の要素に平均評価値の降順に従って格納する . リストからあふれた要素は捨てる .
- (4) $i < M - 1$ ならば $i = i + 1$ として , (2) から (3) を繰り返す . そうでなければ終了 .

4. 性能比較

本章では , 0/1 組合せ最適化問題を用いて GA (MGG) , BOA , SSE , cSSE の探索性能を比較する .

表 2 騙し問題の部分解
Table 2 Part solutions of deceptive problem.

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

4.1 テスト問題

本節では , 0/1 組合せ最適化問題として用いる騙し問題¹⁶⁾ , HIFF 問題¹⁷⁾ , H-Trap 問題¹⁸⁾ , ナップザック問題¹⁹⁾ , グラフ分割問題²⁰⁾ の詳細を述べる .

(1) 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²¹⁾ . ここでは , 表 2 に示す 4 ビットの騙し関数¹⁶⁾ 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 .

$$f_{deception} = \sum_{i=1}^n f_d(x_i) \quad (1)$$

$$x_i \in 0000, 0001, \dots, 1111$$

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 20 とする . 次元数 80 の 0/1 組合せ最適化問題である .

(2) HIFF 問題

HIFF (The Hierarchical-if-and-only-if) 問題は , 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 , 子ノードのシンボルが両方とも '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, \dots, b_k) を持つ親ノードを B , ノード B 以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 .

$$t(b_{left}, b_{right}) = \begin{cases} 0 & \text{if } b_{left} = 0 \text{ and } b_{right} = 0 \\ 1 & \text{if } b_{left} = 1 \text{ and } b_{right} = 1 \\ null & \text{otherwise} \end{cases} \quad (2)$$

$$f(x) = \begin{cases} 1 & \text{if } x = 1 \text{ or } x = 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$T(B) = \begin{cases} B & \text{if } |B| = 1 \\ t(T(b_1), \dots, T(b_k)) & \text{otherwise} \end{cases} \quad (4)$$

$$F(B) = \begin{cases} f(B) & \text{if } |B| = 1 \\ |B|f(T(B)) + \sum_{i=1}^k F(b_i) & \text{otherwise} \end{cases} \quad (5)$$

式 (2) は親ノードがどの状態をとるか示す関数であり、 b_{left} , b_{right} は左右の子ノードのシンボルを表す。式 (3) は各ノードの部分適合度関数であり、 x はノードのシンボルを表す。式 (4) は子ノードを展開する関数であり、式 (5) 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 (最適解 80, 準最適解 64^{17}) を部分分解として 10 個足し合わせた関数を適合度関数とする。次元数 160 の 0/1 組合せ最適化問題である。

(3) 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 (6) で与えられる。

$$f_{trap}(u) = \begin{cases} f_{max} & u = k \\ f_{min} - u \cdot \frac{f_{min}}{k-1} & u \neq k \end{cases} \quad (6)$$

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)⁸⁾ を部分分解として 10 個足し合わせた関数を適合度関数とする。次元数 90 の 0/1 組合せ最適化問題である。

(4) ナップザック問題

ナップザック問題は、 n 個の荷物から制限重量 b の範囲で価値を最大にする荷物群を選ぶ問題である。目的関数と制約条件は次式で与えられる。

$$\begin{aligned} \max_{\{x_i\}} & \sum_{i=1}^n c_i x_i \\ \text{subject to} & \sum_{i=1}^n a_i x_i \leq b \\ & x_i \in \{0, 1\} \quad (i = 1, \dots, n) \end{aligned} \quad (7)$$

本研究では、制限重量 b を平均重量の総和の 1/2 とする¹⁹⁾。荷物数 n は 400、荷物 i の重さ a_i と価格 c_i は 1 から 100 までの一様乱数とし、制限重量 b は

表 3 SSE, cSSE-D, cSSE-C, cSSE の比較
Table 3 Comparison of SSE, cSSE-D, cSSE-C, cSSE.

	SSE	cSSE-D	cSSE-C	cSSE
騙し問題	586.9	588.7	581.4	594.6
HIFF 問題	743.3	744.0	712.1	745.6
H-Trap 問題	171.10	171.10	171.12	171.07
ナップザック問題	16148.3	16154.3	16136.0	16154.8
グラフ分割問題	-184.42	-183.41	-185.20	-183.93

10,000 とする。次元数 400 の 0/1 組合せ最適化問題である。

(5) 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 G124.08²⁰⁾ 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 (8) で与えられる。

$$f_{graph}(L, R) = -c(L, R) - \alpha(|L| - |R|)^2 \quad (8)$$

ここで、制約条件の違反を表す正の定数 α を 0.1 とする。次元数 124 の 0/1 組合せ最適化問題である。

4.2 多様性維持と世代間エリート選択の探索効果の比較

cSSE は、多様性維持と世代間エリート選択という 2 つの操作を用いている。本節では、多様性維持と世代間エリート選択のそれぞれがどのような効果を与えているかを検討する。そこで、SSE、多様性維持のみを用いた cSSE (これを cSSE-D とする)、世代間エリート選択のみを用いた cSSE (これを cSSE-C とする)、通常の cSSE の実験結果を比べる。

各問題において実行世代数は、騙し問題は 40,000、HIFF 問題は 7,500、H-Trap 問題は 1,500、ナップザック問題は 15,000、グラフ分割問題は 15,000 とする。突然変異率は、値を変化させて実験し、各アルゴリズムでそれぞれ最適と思われる値を用いる。各アルゴリズムにおける個体数を 100 とし、異なる初期個体集団から 50 回実行し、それらの平均値を示す。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 に示す。

問題ごとに少し傾向は異なるが、グラフ分割問題を除く 4 つの問題において、cSSE が最も良い解を探索していることが分かる。続いて、cSSE-D、SSE、cSSE-

C の順で良い結果を示している．一方，グラフ分割問題では，最も良い結果を示したのは cSSE-D であり，続いて cSSE，SSE，cSSE-C となっている．

SSE は，もともと良い解の近くにあるさらに良い解を探索するアルゴリズムなので，局所解に陥りやすい傾向がある．cSSE-D と cSSE は SSE における解の多様性を改善したアルゴリズムであるので，SSE よりも大域的な探索性能が改善されている．したがって，これらのアルゴリズムはすべての問題において SSE よりも良い性能を示したと考えられる．一方，cSSE-C は，SSE に世代間エリート選択を組み合わせた結果，SSE よりもさらに局所解付近を探索する特性を強化している．そのために，かえって局所解に陥りやすくなり，すべての問題で最も悪い結果となったと考えられる．最後に，cSSE-D と cSSE を比較すると，cSSE は，グラフ問題を除く 4 つの問題において，cSSE-D よりも良い解を探索していることが分かる．cSSE は，多様性維持と世代間エリート選択を同時に組み合わせることで，多様性を維持しながら良い個体と似通った構造を持つ個体をより集中的に探索する．それにより，cSSE-D と比べて cSSE は，より強い POP の方針を持っており，0/1 組合せ最適化問題に良く適合し，良い探索性能を実現することができた．しかし，グラフ分割問題では，cSSE よりも cSSE-D の方が若干良い最終到達解を示しており，cSSE の探索性能は，問題によっては，cSSE-D よりも若干低下する場合があることが分かる．

4.3 GA (MGG), BOA, SSE, cSSE の探索性能の比較

GA (MGG) では，交叉に 2 点交叉を用い，生成する子個体数を母集団の個体数とする．BOA において，個体生成における親個体数を母集団の 50%，生成する子個体数を母集団の 50% とする．また，GA (MGG)，SSE，cSSE において，突然変異率は，値を変化させて実験し，各アルゴリズムでそれぞれ最適と思われる値を用いる．

各問題において，各アルゴリズムを異なる初期個体集団から 50 回実行し，平均値をとることで各アルゴリズムの性能を評価する．

個体数は，探索性能のポテンシャルを上げるパラメータである．よって，いくつかの個体数において実験することで，それぞれの探索手法の探索性能や探索特性を検討する．ここで，GA，SSE，cSSE は，1 世代に母集団の個体数分に相当する M 個の子個体を生成し，BOA は，1 世代に母集団の 50% に相当する $M/2$ 個の子個体数を生成する．GA，SSE，cSSE は，個体数

表 4 最良個体の最終到達解の平均値
Table 4 Average values of final solutions.

	GA(MGG)	BOA	SSE	cSSE
	個体数 10	個体数 20	個体数 10	個体数 10
	個体数 50	個体数 100	個体数 50	個体数 50
	個体数 100	個体数 200	個体数 100	個体数 100
	個体数 250	個体数 500	個体数 250	個体数 250
騙し問題	571.2	533.4	570.0	573.0
	574.9	560.4	575.2	588.3
	576.1	560.0	586.9	594.6
	575.8	560.0	590.7	597.1
HIFF 問題	659.8	447.4	671.9	675.5
	723.6	623.7	727.2	734.4
	742.8	649.0	743.3	745.6
	764.1	673.2	753.2	764.4
H-Trap 問題	171.21	157.81	171.14	171.19
	171.03	171.00	171.14	171.12
	171.01	171.00	171.10	171.07
	171.01	171.00	171.10	171.09
ナップザック問題	16,060.7	12,754.9	16,104.4	16,141.0
	16,141.9	15,829.3	16,142.8	16,154.1
	16,153.8	16,111.9	16,148.3	16,154.8
	16,155.0	16,153.9	16,149.5	16,155.0
グラフ分割問題	-187.14	-236.13	-187.85	-185.42
	-184.56	-200.79	-185.50	-183.93
	-183.40	-195.70	-184.42	-183.56
	-182.72	-191.33	-183.34	-182.90

を 10, 50, 100, 250 ととり，BOA は，個体数を 20, 100, 200, 500 ととる．よって，各探索手法ともに，1 世代あたりに必要な評価回数は同じとなる．各探索手法ともに 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．

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 4 に，最良個体の最終到達解の標準偏差値を表 5 に示す．また，最良個体の到達解の平均値と標準偏差値のグラフを図 1 ~ 図 10 に示す．各グラフにおいて，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．グラフに用いた各アルゴリズムにおける母集団の個体数は，GA (MGG) は 250，BOA は 500，SSE は 250，cSSE は 250 を用いている．

4.3.1 騙し問題

騙し問題におけ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．

表 4 より，すべての個体数の場合において，cSSE が最も良い最終到達解を示していることが分かる．また，図 1 より，GA (MGG) に比べ，BOA，SSE，cSSE は収束速度が速く，また，cSSE が最も良い収束特性を示していることが分かる．表 4，図 1 より，BOA は，準最適解に収束してから解の改善がおきないこと

表 5 最良個体の最終到達解の標準偏差値
Table 5 Standard deviation of final solutions.

	GA(MGG)	BOA	SSE	cSSE
	個体数 10 個体数 50 個体数 100 個体数 250	個体数 20 個体数 100 個体数 200 個体数 500	個体数 10 個体数 50 個体数 100 個体数 250	個体数 10 個体数 50 個体数 100 個体数 250
騙し問題	3.4 4.0 2.9 2.1	12.5 1.0 0.2 0.0	3.6 7.0 9.0 4.0	3.6 3.7 2.0 2.0
HIFF 問題	34.7 30.1 23.0 19.0	24.2 32.7 32.9 32.7	35.5 28.0 26.4 21.9	28.3 19.7 22.4 36.3
H-Trap 問題	0.38 0.17 0.12 0.12	3.17 0.00 0.00 0.00	0.32 0.32 0.29 0.34	0.37 0.40 0.30 0.32
ナップザック問題	23.3 6.3 1.4 0.0	289.5 68.5 24.5 1.4	13.0 5.5 3.1 2.9	8.8 1.3 0.5 0.0
グラフ分割問題	4.97 3.57 3.35 2.67	6.59 6.01 5.15 4.45	4.86 5.51 3.81 4.03	4.28 4.08 3.96 3.26

が分かる。

表 5 より、各アルゴリズムともに個体数を大きくすると、より小さい標準偏差値を示しており、より安定した解探索を実現することが分かる。特に、SSE に比べ cSSE は、すべての個体数においてより小さい標準偏差値を示していることが分かる。また、BOA は個体数を大きくすると、初期集団のとり方によらずに、準最適解に収束することが分かる。

4.3.2 HIFF 問題

HIFF 問題におけ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。

表 4 より、他の手法に比べ cSSE は、すべての個体数において良い最終到達解を示していることが分かる。図 3 より、GA (MGG) に比べ、BOA, SSE, cSSE は収束速度が速いことが分かる。また、BOA は、いったん収束してから解の改善がおきないことが分かる。さらに、すべての個体数において、他の手法に比べ BOA は最終到達解が良くないことが分かる。

表 5 より、BOA, cSSE に比べ GA (MGG), SSE は、個体数を大きくするとより小さい標準偏差値を示しており、より安定した解探索を実現することが分かる。

4.3.3 H-Trap 問題

H-Trap 問題における最良個体の到達解の平均値の

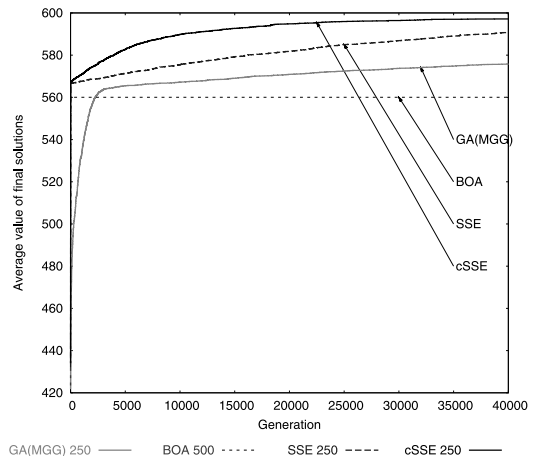


図 1 騙し問題における最良個体の到達解の平均値

Fig. 1 Average values of solutions on deception problem.

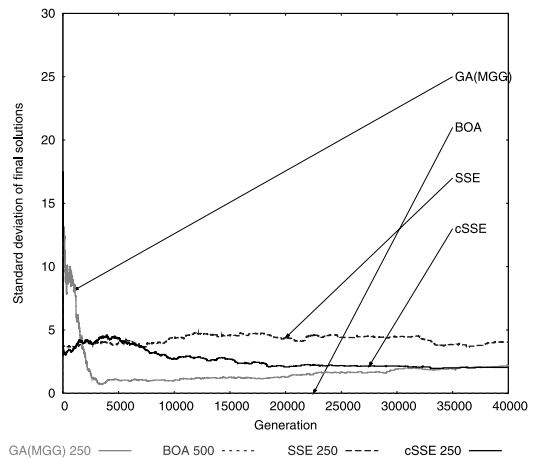


図 2 騙し問題における最良個体の到達解の標準偏差値

Fig. 2 Standard deviation of solutions on deception problem.

グラフを図 5 に、標準偏差値のグラフを図 6 に示す。

表 4 から、BOA の個体数 20 を除いて各アルゴリズムともに、準最適解付近に収束してから解の改善がおきず、同程度の最終到達解を示していることが分かる。また、図 5 より、GA (MGG) に比べ、BOA, SSE, cSSE は収束速度が速いことが分かる。

表 5 より、BOA は個体数を大きくすると、初期集団のとり方によらずに、準最適解に収束することが分かる。また、図 6 より、GA (MGG) に比べ BOA, SSE, cSSE は、標準偏差値がより速く、より小さい値へ収束しており、初期集団のとり方によらない安定した探索性能を実現していることが分かる。

4.3.4 ナップザック問題

ナップザック問題における最良個体の到達解の平均

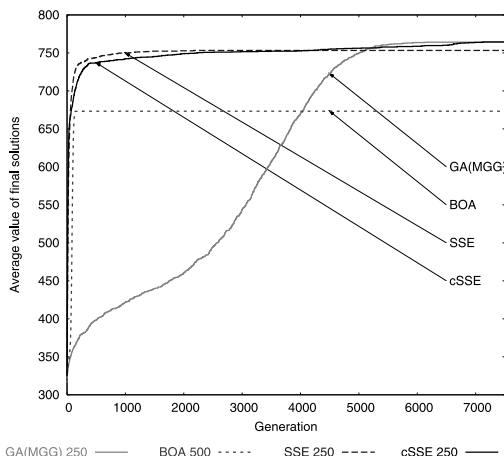


図 3 HIFF 問題における最良個体の到達解の平均値

Fig. 3 Average values of solutions on HIFF problem.

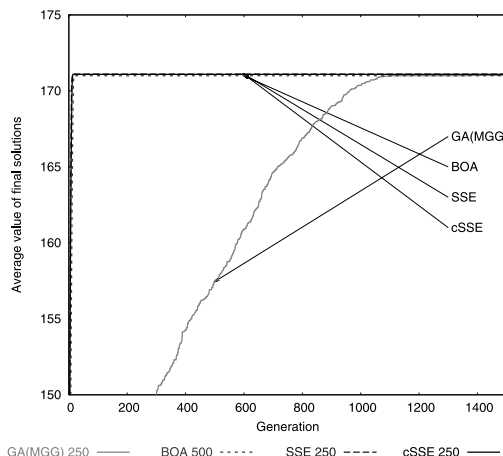


図 5 H-Trap 問題における最良個体の到達解の平均値

Fig. 5 Average values of solutions on H-Trap problem.

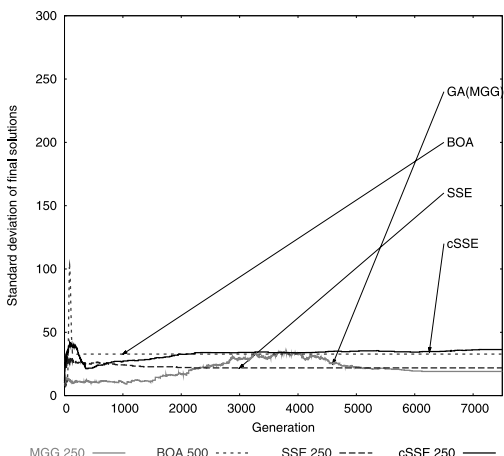


図 4 HIFF 問題における最良個体の到達解の標準偏差値

Fig. 4 Standard deviation of solutions on HIFF problem.

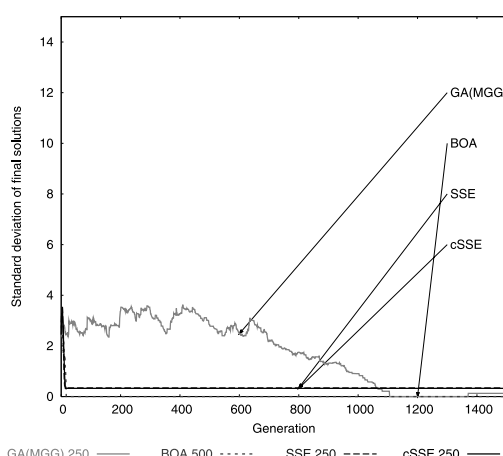


図 6 H-Trap 問題における最良個体の到達解の標準偏差値

Fig. 6 Standard deviation of solutions on H-Trap problem.

値のグラフを図 7 に、標準偏差値のグラフを図 8 に示す。

表 4 より、他の手法に比べ cSSE は、すべての個体数において良い最終到達解を示していることが分かる。特に、個体数が 50 の cSSE は、個体数が 100 の GA (MGG)、500 の BOA、250 の SSE よりも良い最終到達解を示していることが分かる。また、図 7 から、GA (MGG) に比べ、BOA、SSE、cSSE は収束速度が速いことが分かる。BOA は、いったん収束してから解の改善がおきないが、個体数 500 では、SSE よりも GA (MGG)、cSSE に近い最終到達解を示している。ただし、小さい個体数において、他の手法に比べ BOA は最終到達解が良くないことが分かる。

表 5 より、各アルゴリズムともに個体数を大きくすると、より小さい標準偏差値を示しており、より安定

した解探索を実現していることが分かる。特に、他の手法に比べ cSSE は、すべての個体数において小さい標準偏差値を示していることが分かる。また、図 8 より、GA (MGG) に比べ BOA、SSE、cSSE は、標準偏差値がより速く、より小さい値へ収束しており、初期集団のとり方によらない安定した探索性能を実現していることが分かる。

4.3.5 グラフ分割問題

グラフ分割問題におけ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。

表 4 より、各アルゴリズムともに、個体数を大きくすると最終到達解が向上していることが分かる。また、図 9 から、GA (MGG) に比べ、BOA、SSE、cSSE

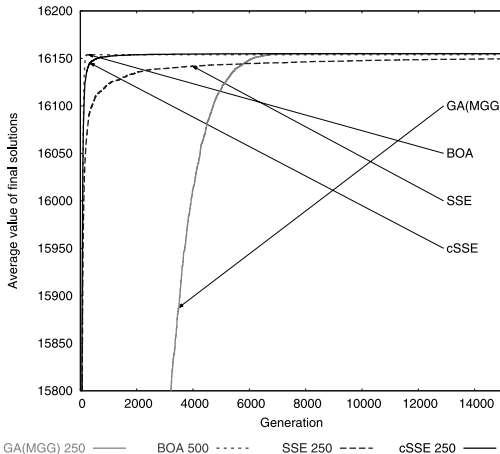


図 7 ナップザック問題における最良個体の到達解の平均値
Fig. 7 Average values of solutions on knapsack problem.

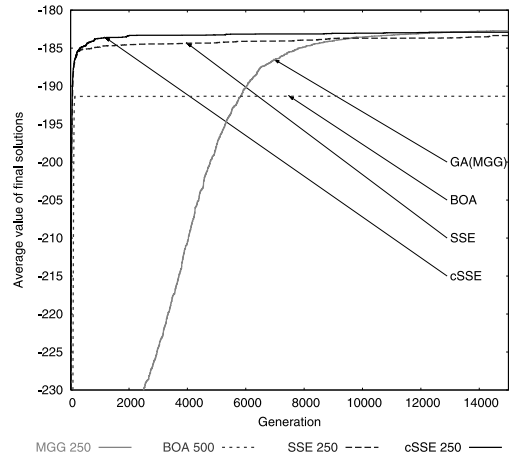


図 9 グラフ分割問題における最良個体の到達解の平均値
Fig.9 Average values of solutions on graph partitioning problem.

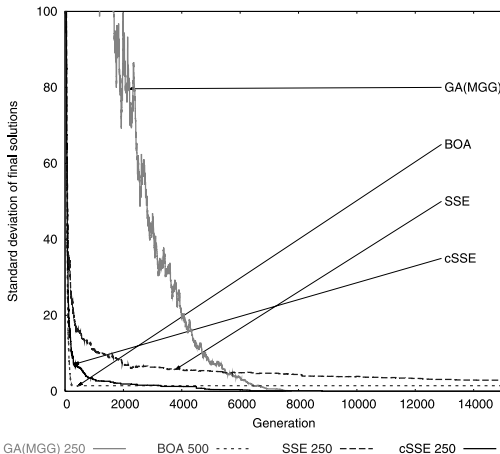


図 8 ナップザック問題における最良個体の到達解の標準偏差値
Fig.8 Standard deviation of solutions on knapsack problem.

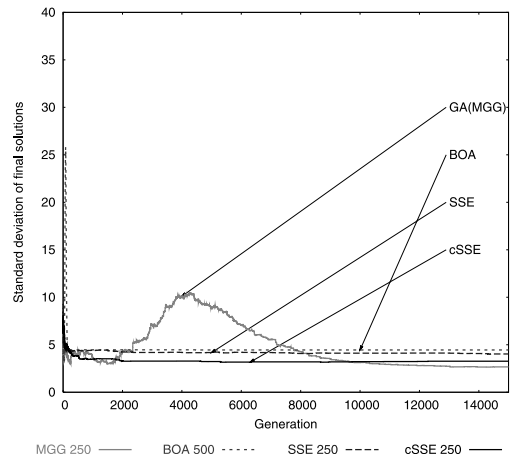


図 10 グラフ分割問題における最良個体の到達解の標準偏差値
Fig.10 Standard deviation of solutions on graph partitioning problem.

は収束速度が速いことが分かる。SSE に比べ cSSE は、GA (MGG) と同程度の最終到達解を示していることが分かる。また、BOA は、いったん収束してから解の改善がおきないことが分かる。さらに、すべての個体数において、他の手法に比べ BOA は最終到達解が良くないことが分かる。

表 5 より、各アルゴリズムともに個体数を大きくすると、より小さい標準偏差値を示しており、より安定した解探索を実現していることが分かる。また、図 10 より、GA (MGG) に比べ BOA , SSE , cSSE は、標準偏差値がより速く、より小さい値へ収束しており、初期集団のとり方によらない安定した探索性能を実現していることが分かる。

4.4 GA (MGG) , BOA , SSE , cSSE の計算時間の比較

本節では、GA (MGG) , BOA , SSE , cSSE の計算時間を比較する。GA (MGG) は 250 , BOA は 500 , SSE は 250 , cSSE は 250 の個体数を用いて解析を行う。各アルゴリズムともに、1 世代あたりの適合度計算回数は 250 となる。各アルゴリズムにおける 1 世代あたりの計算時間を表 6 に示す。表 6 より、SSE , cSSE は、GA (MGG) , BOA に比べ計算時間が長いことが分かる。SSE , cSSE は、スキーマ抽出と個体のソートの操作が必要であり、そのため GA (MGG) , BOA に比べ 1 世代あたりの計算時間が長くなった。

しかし、グラフ分割問題では、他の問題と比較して計算時間の差が小さくなっていることが確認できる。

表 6 1 世代あたりの計算時間 (秒)

Table 6 Calculation time (second) of the one generation.

	GA(MGG)	BOA	SSE	cSSE
騙し問題	7.32e-03	5.07e-03	7.16e-02	6.54e-02
HIFF 問題	1.31e-02	1.02e-02	1.07e-01	9.77e-02
H-Trap 問題	4.92e-03	4.17e-03	6.69e-02	8.23e-02
ナップザック問題	1.46e-02	1.71e-02	1.96e-01	1.82e-01
グラフ分割問題	4.71e-02	3.18e-02	1.19e-01	1.01e-01

表 7 適合度関数の計算時間 (秒)

Table 7 Calculation time (second) of fitness function.

騙し問題	1.15e-05
HIFF 問題	2.18e-05
H-Trap 問題	7.35e-06
ナップザック問題	1.55e-05
グラフ分割問題	9.71e-05

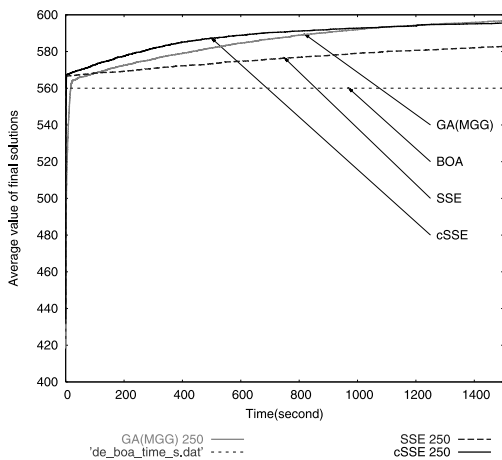


図 11 騙し問題における最良個体の到達解の平均値 (横軸: 計算時間)

Fig. 11 Average values of solutions on deception problem (x-axis: time).

そこで、各問題で適合度関数を 1 回評価するにかかる計算時間を比較すると表 7 のようになる。この表より、他の問題に比べてグラフ分割問題は適合度関数の計算時間が長いことが分かる。適合度関数の計算時間が多くなると、全計算時間におけるスキーマ抽出と個体のソートの操作の計算時間が占める割合が小さくなるので、SSE、cSSE と GA (MGG)、BOA の計算時間の差が小さくなったと考えられる。

次に、各アルゴリズムの到達解と計算時間のグラフを図 11 ~ 図 15 に示す。この節以前の解析例では横軸に世代数をとっていたが、図 11 ~ 図 15 では横軸に計算時間 (CPU 時間) をとり、縦軸に最良個体の到達解の平均値をとる。

図 11 から、騙し問題では GA (MGG) に比べ

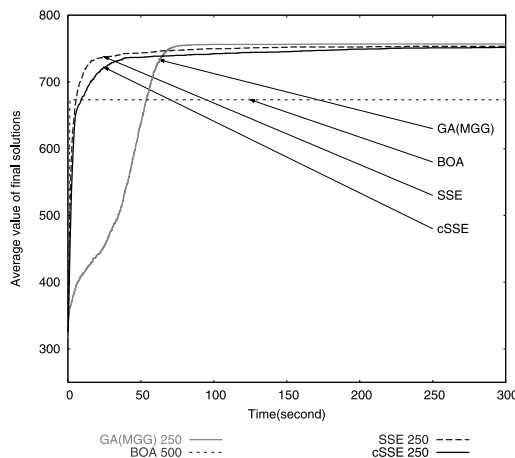


図 12 HIFF 問題における最良個体の到達解の平均値 (横軸: 計算時間)

Fig. 12 Average values of solutions on HIFF problem (x-axis: time).

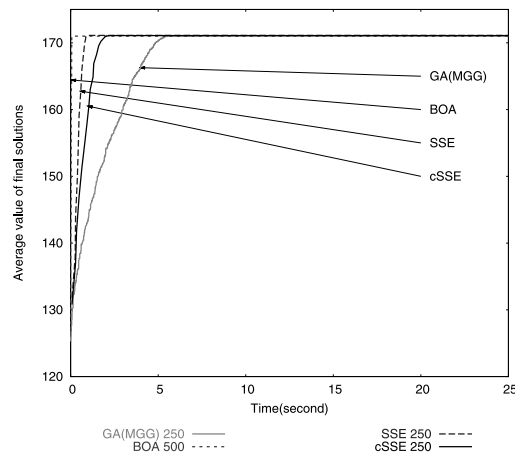


図 13 H-Trap 問題における最良個体の到達解の平均値 (横軸: 計算時間)

Fig. 13 Average values of solutions on H-Trap problem (x-axis: time).

て SSE は収束に長い時間がかかるが、cSSE は GA (MGG) よりも到達解が若干早く収束していることが分かる。

図 13、図 14 から、H-Trap 問題、ナップザック問題において、SSE、cSSE とともに GA (MGG) に比べて短い時間で到達解が収束していることが分かる。

図 12、図 15 から、HIFF 問題、グラフ分割において、BOA と同様に SSE、cSSE は、GA (MGG) に比べて短い計算時間で到達解が収束していることが分かる。

これらの解析結果より、SSE、cSSE は、1 世代あたりの計算時間は他のアルゴリズムよりも若干長いにも

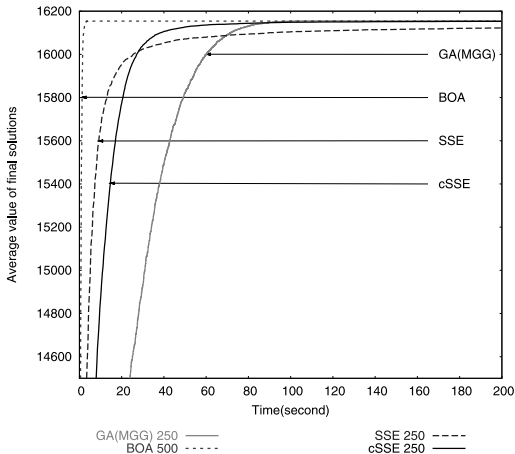


図 14 ナップザック問題における最良個体の到達解の平均値(横軸: 計算時間)

Fig. 14 Average values of solutions on knapsack problem (x-axis: time).

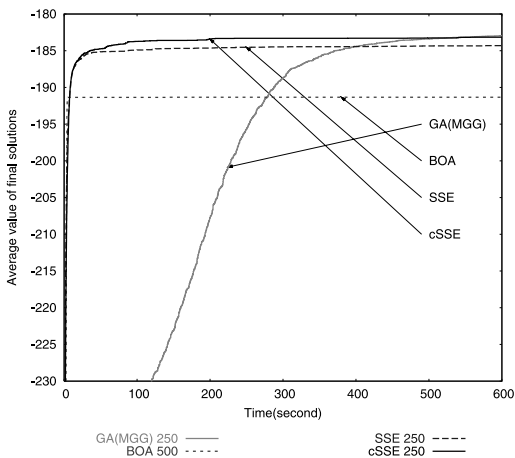


図 15 グラフ分割問題における最良個体の到達解の平均値(横軸: 計算時間)

Fig. 15 Average values of solutions on graph partitioning problem (x-axis: time).

かわらず、収束速度が速いという特性から、結果的に GA (MGG) よりも短い時間で解探索ができていることが分かる。

5. まとめ

本研究では、SSE の母集団に多様性を持たせることで大域的探索能力が向上する可能性があることから、SSE の世代交代モデルを改良した cSSE を提案した。そして、0/1 組合せ最適化問題において、SSE、cSSE を実問題に有効とされる MGG に基づく GA、SSE と類似の設計方針を持つ BOA と性能比較を行った。性能比較の結果以下のことが分かった。

GA (MGG) は、世代間における母集団の差異を最小にする探索手法であり、他の探索手法と比べ収束速度は遅いが、大きい個体数において良い最終到達解を示していることが分かった。BOA は、優れた初期収束特性と初期集団のとり方によらない安定した解探索性能を有していることが分かった。SSE は BOA と同程度の収束速度と安定した解探索で、BOA よりも良い最終到達解を示していることが分かった。

次に、本研究で提案した cSSE の探索性能について述べる。実験の結果から、cSSE は、グラフ分割問題では SSE に比べ若干の収束速度の低下が見られたが、騙し問題、ナップザック問題では SSE よりも良い収束速度を示していることが分かった。また、すべての問題において、cSSE は GA (MGG) と同程度の最終到達解を示していることが分かった。特に、騙し問題、HIFF 問題、ナップザック問題では、すべての個体数において最も良い最終到達解を示していることが分かった。cSSE は、BOA、SSE と同程度の優れた収束速度と初期集団のとり方によらない安定した解探索を示していることが分かった。

以上より、cSSE は、SSE の優れた初期収束特性を維持したまま、GA (MGG) と同程度の最終到達解を安定した探索で実現していることが分かった。

最後に、実問題への対応を含めて、本研究で提案した cSSE の特徴について述べる。生成した子個体を無条件に次世代の母集団とする SSE に対して、cSSE は、多様性を導入した世代間エリート選択を用いることで、今までに探索された真に良い個体から次世代の母集団を生成する。よって、SSE に比べて cSSE は、より集中的に適合度の良い個体と似通った構造を持つ個体を探索するアルゴリズムであり、より強い POP の方針を持っているといえる。cSSE は、BOA、SSE と同じく高速で安定した収束特性と、GA (MGG) と同程度の大域的探索能力を有していることが分かった。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組合せ最適化問題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、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, Vol.J78-D-II, No.1, pp.94-104 (1995).
- 5) 丸山 崇, 北 英輔: スキーマどん欲法の検討と拡張について, 情報処理学会数理モデル化と問題解決, Vol.2004, No.042, pp.5-8 (2004).
- 6) 丸山 崇, 北 栄輔: 確率的スキーマ貪欲法の検討と拡張, 性能比較について, 情報処理学会論文誌: 数理モデル化と応用, Vol.47, No.TOM15, pp.16-30 (2006).
- 7) Holland, J.H.: *Adaptation in Natural and Artificial Systems*, The University of Michigan Press (1975).
- 8) 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.734-744 (1997).
- 9) Baker, J.E.: Reducing bias and inefficiency in the selection algorithm, *Proc. 2nd International Conference on Genetic Algorithms*, pp.14-21 (1987).
- 10) 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).
- 11) 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).
- 12) 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).
- 13) 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).
- 14) Pelikan, M., Goldberg, D.E. and Cantu-Paz, E.: Boa: The bayesian optimization algorithm, *Proc. Genetic and Evolutionary Computation Conference 1999 (GECCO-1999)*, San Francisco, 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).
- 15) Eshelman, L.J.: The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination, *Foundations of Genetic Algorithms 1991 (FOGA 1)*, Rawlins, G.J.E. (Ed.), pp.265-283, Morgan Kaufmann (1991).
- 16) 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).
- 17) 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 Zalzal, A. (Eds.), pp.1406-1413, IEEE (1999).
- 18) Pelikan, M. and Goldberg, D.E.: Escaping hierarchical traps with competent genetic algorithms, *Proc. Genetic and Evolutionary Computation Conference 2001 (GECCO-2001)*, San Francisco, CA, Spector, L., et al. (Eds.), pp.511-518, Morgan Kaufmann (2001).
- 19) Simoes, A.B. and Costa, E.: An evolutionary approach to the zero/one knapsack problem: Testing ideas from biology, *Proc. 5th International Conference on Neural Networks and Genetic Algorithms (ICANNGA '01)*, Kurkova, V., Steele, N., Neruda, R. and Karny, M. (Eds.), pp.236-239, Springer Verlag, Wien (2001).
- 20) 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).
- 21) 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).

(平成 17 年 12 月 6 日受付)

(平成 18 年 8 月 9 日再受付)

(平成 18 年 10 月 2 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, 日本計算工学会各会員．
