

セル構造を有する分散遺伝的アルゴリズムの提案

中島 智晴[†] 有山 貴信[†] 石 淵 久 生[†]

遺伝的アルゴリズムにおいて個体集団をいくつかの部分集団に分割する分散化の研究が活発である。分散化の方法として代表的なものに、Coarse-grained 遺伝的アルゴリズムに分類される島モデル遺伝的アルゴリズムと Fine-grained 遺伝的アルゴリズムに分類されるセルラー遺伝的アルゴリズムがある。本論文では、これら 2 つの遺伝的アルゴリズムを融合したセルラー構造を有する遺伝的アルゴリズムを提案する。提案手法では、各部分個体群がセルラー遺伝的アルゴリズムを実行し、部分個体群間の境界に位置する個体のみが、他の部分個体群の個体と相互作用することができる。数値実験では、関数最適化問題や巡回セールスマン問題に提案手法を適用し、他の手法と比較する。比較の結果や提案手法の有効性を示す。また、提案手法に対していくつかの考察を行い、なぜ提案手法が有効であるかの議論も行う。

A Distributed Genetic Algorithm That Has Cellular Structures

TOMO HARU NAKASHIMA,[†] TAKANOBU ARIYAMA[†]
and HISAO ISHIBUCHI[†]

Various types of genetic algorithms have been proposed in the literature. One of the active topics is on the distribution of genetic algorithms. The distributed genetic algorithms generally fall into two categories: Coarse-grained genetic algorithms and fine-grained genetic algorithms. In this paper, we propose a method, which is a hybrid of the two distributed genetic algorithms. In the proposed method, there are a number of subpopulations that evolve independently from each other. Each subpopulation has a cellular structure as in cellular genetic algorithms. Only those individuals on boundary cells in a subpopulation can communicate with individuals in other subpopulations. In computer simulations on function optimization problems and traveling salesman problems, we compare the performance of the proposed method with that of other genetic algorithms. From simulation results, the effectiveness of the proposed method is shown.

1. はじめに

遺伝的アルゴリズムや遺伝的プログラミング、進化プログラミングなどに代表される進化的手法は、生物進化の過程からヒントを得た解探索手法として計算機科学の分野を中心に広範囲な研究が行われている。特に、遺伝的アルゴリズム^{(6),(7)}においては、探索性能を向上させるための様々なヴァリエーションが研究されている。その中でも活発な研究が行われているのは、遺伝的アルゴリズムの分散化である⁽³⁾。すなわち、個体群を複数の部分個体群に分割し、それぞれの部分個体群において遺伝的アルゴリズムを実行させる手法である。遺伝的アルゴリズムの分散化の利点は大きく 2 つある。1 つは、各部分個体群を並列に実行することで、遺伝的アルゴリズムの総計算時間が短縮できると

いう高速処理の可能性である。もう 1 つは、遺伝的アルゴリズムにおける選択や交叉などの遺伝的操作を局所的に行うことによる、解探索性の向上の可能性である。1 つ目の利点は、個体群を分散化させることにより、特に並列計算機を用いてアルゴリズムを実行することが容易となるというハードウェア的な面からの利点であり、2 つ目の利点は、遺伝的操作というソフトウェア（もしくはアルゴリズム）的な面からの利点である。本論文では、2 つ目の利点であるソフトウェア的な面からの利点について注目することにする。したがって、遺伝的アルゴリズムを分散化し、並列計算機を駆使して遺伝的アルゴリズムの探索性能を向上させるという目的は、本論文では考慮されていない。

分散化された遺伝的アルゴリズムは、その分散の度合い（粒度）によって大きく 2 つに分類することができる。1 つは Coarse-grained 遺伝的アルゴリズム^{(2),(13)}であり、もう 1 つは Fine-grained 遺伝的アルゴリズム^{(8),(11)}である。Coarse-grained 遺伝的アルゴリズム

[†] 大阪府立大学工学部
College of Engineering, Osaka Prefecture University

では、個体群はいくつかの部分個体群に分割されており、各部分個体群では独立に遺伝的アルゴリズムが実行される。部分個体群は一種の島と見なすことができるので、Coarse-grained 遺伝的アルゴリズムは島モデル遺伝的アルゴリズムとも呼ばれる。一方、典型的な Fine-grained 遺伝的アルゴリズムであるセルラー遺伝的アルゴリズムでは、セル構造をなす格子上の空間に個体が配置されている。遺伝的操作の及ぶ範囲は各個体ごとに決定された近傍構造により限定される。これらの分散化遺伝的アルゴリズムの長所として、個体間の多様性が維持しやすく、局所最適解に陥るような初期収束を招きにくい点があげられる。

本論文では、島モデル遺伝的アルゴリズムとセルラー遺伝的アルゴリズムの考えを融合した、セル構造を有する分散遺伝的アルゴリズムを提案する。提案手法では、個体群はいくつかの部分個体群に分割されており、各部分個体群はセル構造を持つ。部分個体群は独立に遺伝的アルゴリズムを実行するが、部分個体群間の境界に配置されている個体のみ、隣り合う部分個体群と相互作用することが許されている。

数値実験では、8つの関数最適化問題と2つの巡回セールスマン問題に対して提案手法の性能を調べる。提案手法は、Simple genetic algorithm や標準のセルラー遺伝的アルゴリズムの性能と比較される。また、部分個体群のトポロジを変化させた場合に、提案手法の性能に与える影響も調べられる。これらの数値実験によって得られた結果から、特別な近傍サイズを使用しているときに提案手法の性能が特に良いことを示し、さらなる手法の解析を進める。

2. 分散遺伝的アルゴリズム

本章では、遺伝的アルゴリズムにおける遺伝的操作を局所化させる分散遺伝的アルゴリズムの説明を行う。

2.1 島モデル遺伝的アルゴリズム

島モデル遺伝的アルゴリズムでは、個体群を複数の部分個体群(島)に分割し、各島で遺伝的アルゴリズムを実行させる^{2),13)}(図1)。

各島は、ときどき相互作用しあい、他の島からの個体を受け入れたり、他の島へ個体を送り出したりする。この操作を移住と呼ぶ。移住操作を発生させるタイミングによって、同期型と非同期型とに分類することができる。同期型では、ある決められた世代(移住間隔と呼ぶ)ごとに移住操作が発生する。島における個体群の大きさに対する移住される個体の数を移住率と呼ぶ。移住操作では、各島から移住される個体がある選択確率に従って選択される。移住先の島はランダムに

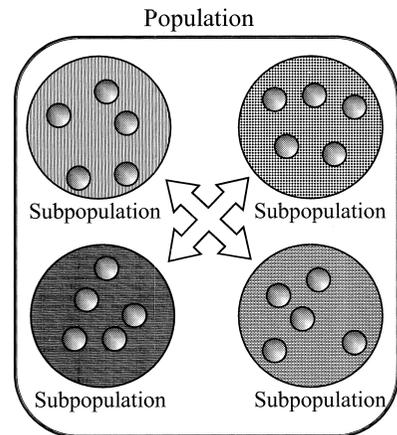


図1 島モデル遺伝的アルゴリズム
Fig. 1 An island genetic algorithm.

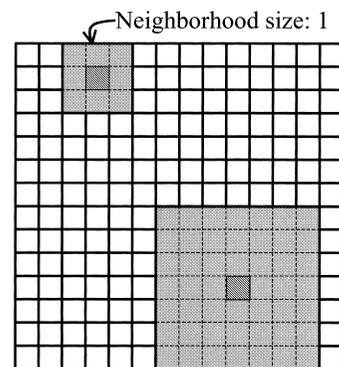


図2 セルラー遺伝的アルゴリズム
Fig. 2 A cellular genetic algorithm.

決定され、ランダムに選択された移住先の島内の個体と置き換えられる。

非同期型の移住方法には様々なものが考えられる。たとえば、各島ごとに移住間隔をランダムに与えたり、解の収束などの基準を設けて、各島がその基準を満たし次第移住を行ったりすることが考えられる。

2.2 セルラー遺伝的アルゴリズム

Fine-grained 遺伝的アルゴリズムの代表例であるセルラー遺伝的アルゴリズムでは、各個体は格子状に並べられたセル上に配置されている(図2)。1つのセルに1つの個体が配置される場合が多いが、ASPARAGOS^{9),10)}のようにやや特殊な場合もある。

交叉操作における親を選択する際、対象となる個体の近傍サイズ以内に配置されている個体のみが選択操作の対象となる。図2には、近傍サイズが1の場合の選択操作の対象領域と近傍サイズが3の場合の選択操作の対象領域が示されている。

セルラー遺伝的アルゴリズムにおけるセル構造は、上下左右がそれぞれつながっている、いわゆるトーラス状である場合もあるが、本論文で用いるセル構造は、トーラス状であることを仮定しないことにする。

3. 提案手法

本章では、セル構造を有する分散遺伝的アルゴリズムを提案する。提案手法では、個体群がいくつかの部分個体群に分割されており、各々の部分個体群にはセル構造が組み込まれている。また、部分個体群には互いに隣接関係が仮定されている。

図3は提案手法の一例である。図3では、400個の個体からなる個体群が4つの部分個体群に分割されており、各部分個体群は 10×10 のセル構造を有している。このように、各部分個体群において独立してセルラー遺伝的アルゴリズムを実行することが提案手法の特徴である。提案手法のもう1つの特徴は、部分個体群間の境界のセルに位置している個体（図3では塗りつぶされた領域）のみが、隣接する部分個体群と相互作用することが可能な点である。本論文では、境界のセルに位置している個体の近傍サイズをつねに1に設定する。たとえば、図3において、左上の部分個体群内の境界のセルに位置している個体Aの近傍サイズは1であり、太線で囲まれた、左下の部分個体群内のセルを含む近傍から交叉の親となる個体を選択することができる。境界以外のセルでは、他の部分個体群内の個体を選択操作の対象とすることはできないものとする。このことにより、境界のセルに位置している個体は、隣接している部分個体群の境界に位置しているセルの個体と局所的に相互作用することが可能となる。

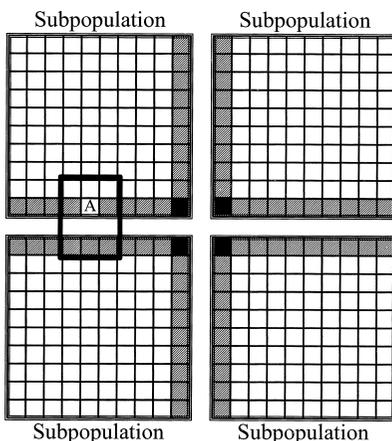


図3 提案手法（ 2×2 のトポロジー）
Fig.3 The proposed method (2×2).

以下に、提案手法のアルゴリズムを記述する。

[提案手法のアルゴリズム]

- (1) 個体群の初期化。
- (2) 各部分個体群に対して、以下を実行する。
 - (a) 各個体に対して、(i) もし、その個体が部分個体群間の境界に位置していれば、無条件に近傍サイズを1とする。(ii) もし、そうでないならば近傍サイズを N とする。
 - (b) 各個体に対して、自分と近傍サイズ内に位置する個体の中からルーレット選択により親となる個体を選択する。このとき、境界に位置している個体については、隣接する部分個体群中の個体も近傍に含めることができる。部分個体群間の境界に位置していない個体に対しては、自分の属する部分個体群中の個体のみを近傍サイズ内の個体とする。
 - (c) 親個体から交叉操作により子個体を生成する。
 - (d) 生成された子個体に突然変異操作を適用する。(a)~(d)の操作は各個体に対して同時に行うものとする。
 - (e) 前世代におけるエリート個体を保存する。
- (3) 終了判定。終了条件が満たされていないければ(2)へ戻る。

エリート個体の保存の方法としては、各部分個体群のエリートをあらかじめ決められた位置に配置する（たとえば、図3では中央付近の黒塗り）方法や、ランダムに各部分個体群に配置する方法がある。4章の数値実験では、これら2つの手法について性能を調査する。

提案手法のねらいは、セルラー遺伝的アルゴリズムや島モデル遺伝的アルゴリズムの両方の考え方を簡単な方法で融合させることにより、個体の進化を部分個体群ごとで行い、全体として解の多様性を維持し、より良い個体の探索効率を向上させるというものである。すなわち、本論文の主眼は融合させる従来手法の長所を併せ持つ枠組みを開発することにある。

以下に、提案手法と従来手法であるセルラー遺伝的アルゴリズム、島モデル遺伝的アルゴリズムとの違いを述べる。

[提案手法とセルラー遺伝的アルゴリズムとの違い]

- (1) 提案手法では個体群がいくつかの部分個体群に分割されているが、セルラー遺伝的アルゴリズム

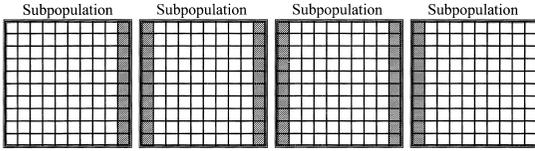


図4 提案手法(1×4のトポロジー)

Fig. 4 The proposed method with a different topology (1×4).

ムでは分割されていない。

- (2) エリート個体の数が、提案手法では部分個体数だけあるのに対し、標準的なセルラー遺伝的アルゴリズムでは1つである。

[提案手法と島モデル遺伝的アルゴリズムとの違い]

- (1) 提案手法では各部分個体群においてセル構造が仮定されているが、島モデル遺伝的アルゴリズムではセル構造は仮定されていない。
- (2) 部分個体群間の相互作用を行うために、島モデル遺伝的アルゴリズムでは移住率や移住間隔、同期移住か非同期移住かなど、様々な設定をしなければならないが、提案手法では、部分個体群間の隣接関係を定めるだけでよい。

島モデル遺伝的アルゴリズムにおける移住間隔と移住率は、その可能な組合せが膨大であり、より良い性能を求めるためには十分な予備実験が必要である。一方、提案手法では、近傍サイズのみを決定すればよいので、予備実験の量は少なくすむことが予想される。また、4章の数値実験で示されるように、提案手法の性能は近傍サイズの設定で大きく変わることがない。したがって、実際の実装においても、提案手法は島モデル遺伝的アルゴリズムよりも有利であるといえる。なお、提案手法における部分個体群間の隣接関係として、図3のような2×2のほかにも、図4のように1×4のトポロジーを考えることもできる。後者については、4.4節で考えることにする。

4. 数値実験

本章では、様々なテスト問題に対する提案手法の性能を調査する。テスト問題として、本論文では、関数最適化問題と巡回セールスマン問題を扱った。以下の節で、各問題に対する数値実験の結果を述べる。

4.1 関数最適化問題

提案手法の有効性を調べるために、関数最適化問題として、De Jong のテスト問題⁴⁾、Rastrigin 関数、Schwefel 関数、Griewangk 関数を取り扱った。これらの関数を以下に示す。また、De Jong のテスト関数を

F1~F5, Rastrigin 関数を F6, Schwefel 関数を F7, Griewangk 関数を F8 と表記することにする。

$$F1 : f_1(x_i | i = 1, 2, 3) = \sum_{i=1}^3 x_i^2, \quad x_i \in [-5.12, 5.11] \quad (1)$$

$$F2 : f_2(x_i | i = 1, 2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2, \quad x_i \in [-2.048, 2.047] \quad (2)$$

$$F3 : f_3(x_i | 1 \leq i \leq 5) = \sum_{i=1}^5 \text{Integer}(x_i), \quad x_i \in [-5.12, 5.11] \quad (3)$$

$$F4 : f_4(x_i | 1 \leq i \leq 30) = \left[\sum_{i=1}^{30} i(x_i)^4 \right] + \text{Gauss}(0, 1), \quad x_i \in [-1.28, 1.27] \quad (4)$$

$$F5 : f_5(x_i | i = 1, 2) = \frac{1}{0.002 + \left[\sum_{j=1}^{25} \frac{1}{j + \left[\sum_{i=1}^2 \frac{1}{(x_i - a_{ij})^6} \right]} \right]}, \quad x_i \in [-65.536, 65.535] \quad (5)$$

$$F6 : f_6(x_i | 1 \leq i \leq 20) = 20 \times 10 + \left[\sum_{i=1}^{20} x_i^2 - 10 \cos(2\pi x_i) \right], \quad x_i \in [5.12, 5.11] \quad (6)$$

$$F7 : f_7(x_i | i \leq i \leq 10) = 418.9829 \times 10 + \left[\sum_{i=1}^{10} \left(-x_i \sin \left(\sqrt{|x_i|} \right) \right) \right], \quad x_i \in [-512, 511] \quad (7)$$

$$F8 : f_8(x_i | 1 \leq i \leq 10) = \left[\sum_{i=1}^{10} \frac{x_i^2}{4000} \right] - \left[\prod_{i=1}^{10} \cos \left(\frac{x_i}{\sqrt{i}} \right) \right] + 1, \quad x_i \in [-512, 511] \quad (8)$$

式(5)において、 a_{ij} は以下のように定義される行列 A の要素である。

$$A = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 \\ -32 & \cdots & 0 & 16 & 32 \\ -16 & \cdots & 32 & 32 & 32 \end{pmatrix} \quad (9)$$

これらの関数はすべて、最小化問題である。表1に各

表 1 関数最適化問題
Table 1 Function optimization problems.

関数	入力変数	ビット数	最適値
F1	3	10	$f_1(x_i = 0.0) = 0.0$
F2	2	12	$f_2(x_i = 1.0) = 0.0$
F3	5	10	$f_3(x_i = [-5.12, -5.00]) = -30$
F4	30	8	$f_4(x_i = 0.0) = 0.0$
F5	2	17	$f_5(x_i = -32.0) = 0.998$
F6	20	10	$f_6(x_i = 0.0) = 0.0$
F7	10	10	$f_7(x_i = 420.9687) = 0.0$
F8	10	10	$f_8(x_i = 0.0) = 0.0$

関数の入力変数の数と各変数を 2 進コーディングする際のビット数、コーディングにおける大域的最適値を示す。

これらの関数最適化問題に対して、以下の 4 つのアルゴリズムの性能を調べた。

手法 1 セル構造を持たず、個体群の分割も行われていない、いわゆる Simple genetic algorithm。ただし、各世代で最も適応度の高い個体を次世代に残すエリート保存戦略が用いられている。

手法 2 各個体が 20×20 のセル構造上に配置されているが、個体群の分割は行われていない、標準のセルラー遺伝的アルゴリズム。手法 1 と同様にエリート保存戦略が用いられる。この場合、各世代で最も適応度の高い個体が次世代のランダムに選択されたセルに配置される。

手法 3 セルラー構造と個体群分割を行った提案手法。各部分個体群は 10×10 のセル構造を保持している。また、部分個体群間の境界に位置している個体のみ、隣接する部分個体群との相互作用を行う。本実験では、図 3 のように 2×2 に部分個体群を配置した。また、各部分個体群において、世代ごとで最も適応度の高い個体を次世代の決められた場所(図 3 の黒塗り部分)に配置するエリート保存戦略を使用した。これは、エリート個体を隣接する 3 つの部分個体群と相互作用を行うことのできる場所に配置することで、個体群全体に良好な解が広がることを意図したものである。

手法 4 セル構造と個体群分割を行った提案手法。手法 3 との違いは、エリート保存戦略において、各部分個体群のエリートを、次世代における部分個体群内でランダムに選択されたセルに配置する点である。

各問題に対して手法 1~4 を適用し、それらの最適値への探索性能を調べる。実験において使用したパラメータ設定は以下のとおりである。

個体数 400 (20×20)
 交叉操作 一点交叉
 突然変異操作 ビット反転
 交叉確率 1.0
 突然変異確率 0.05
 近傍サイズ 1, 2, 3, 4, 5, 最大半径
 アルゴリズム終了条件 500 世代
 交叉確率については、予備数値実験により、0.9 や 1.0 の場合に良好な成果が得られることを確認している。したがって、本論文では、交叉確率を 1.0 とした。

ただし、手法 3 と手法 4 では図 3 に示されているように、 10×10 のセル構造をしている部分個体群が 4 つ集まって構成されている。また、近傍サイズはセル構造を持たない手法 1 に対しては適用されない。最大半径とは、手法 2 では 19 のことであり、手法 3 と手法 4 においては 9 のことを指している。どの手法においても、適応度として、個体が表現している情報をデコード化して得られる入力ベクトルに対する関数の値の逆数が用いられた。したがって、遺伝的アルゴリズムの中では最大化が行われていることになる。

問題 F1~F8 に対して、手法 1~手法 4 を 10 回適用し、その実行終了時におけるエリート個体の平均、Max, Min を調べた。数値実験結果を図 5~図 8 に示す。図 5~図 8 は関数 F4, F6, F7, F8 における手法 1~手法 4 の性能を近傍サイズごとに並べたものである。他の関数(F1, F2, F3, F5)では、手法 1~4 のほとんどすべての手法において最適解を獲得することができていたため、実験結果の報告は割愛し、より手法間の性能の差が顕著である関数 F4, F6, F7, F8 に対する数値実験結果のみを示すことにした。

図 5~図 8 より、手法 1 よりも手法 2 が、手法 2 よりも手法 3 と手法 4 がより最適値に近い解を見つけ出すことに成功していることが分かる。また、手法 2 では近傍サイズごとの性能のばらつきが大きいものに対して、手法 3 と手法 4 では、近傍サイズにほとんど依存することなく良い手法が得られていることが分かる。したがって、提案手法の性能は近傍サイズの設定に敏感でないと結論付けることができる。

さらに、手法 3 と手法 4 を比較すると、手法 4 の方が良好な解が得られていることが分かる。したがって、以降の数値実験では、エリート個体が各部分個体群でランダムに選択されたセルに配置される手法 4 のみを提案手法として扱い、その性能を調べることにする。

4.2 巡回セールスマン問題

次に、順序組合せ最適化問題である巡回セールスマン問題に対して提案手法の有効性を検討した。本論文

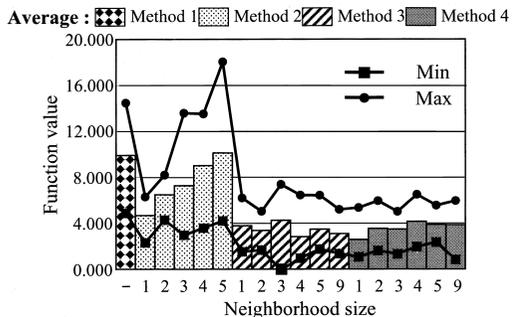


図 5 関数 F4 に対する数値実験結果
Fig. 5 Simulation results on F4.

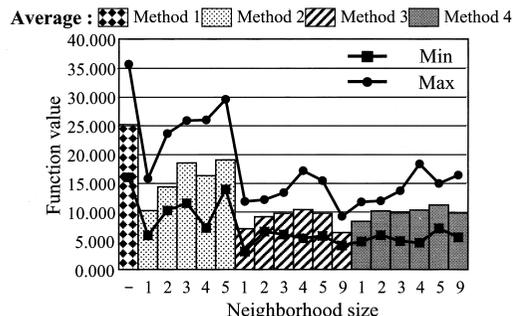


図 6 関数 F6 に対する数値実験結果
Fig. 6 Simulation results on F6.

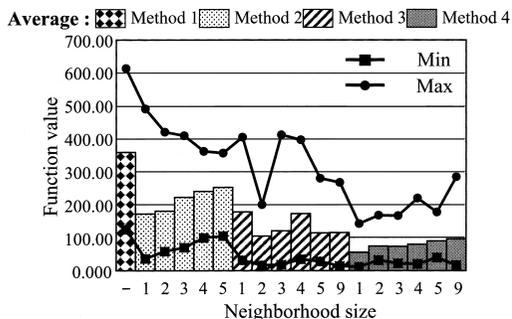


図 7 関数 F7 に対する数値実験結果
Fig. 7 Simulation results on F7.

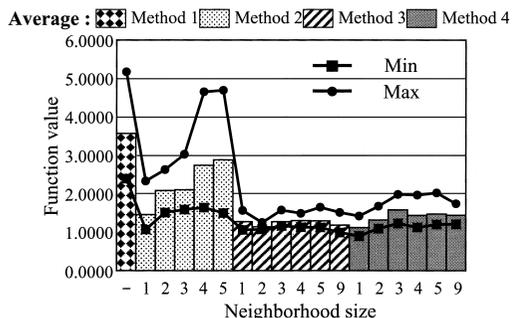


図 8 関数 F8 に対する数値実験結果
Fig. 8 Simulation results on F8.

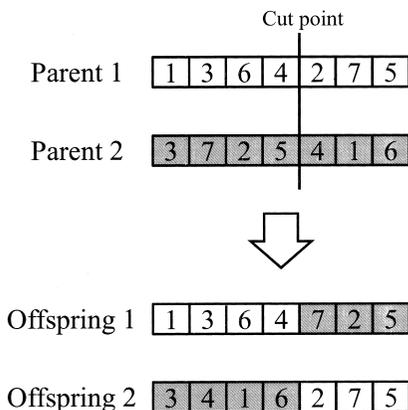


図 9 一点順序交叉
Fig. 9 One-point order crossover.

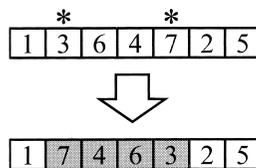


図 10 逆位操作
Fig. 10 Inversion operation.

では、インターネット上で利用可能である Reinelt の TSP library¹²⁾ から Eil51 と KroA100 の 2 つの巡回セールスマン問題を考えることにした。Eil51 は 51 都市問題，KroA100 は 100 都市問題であり，すでに最適値が求められている。

各問題に対して，4.1 節で用いられた手法を適用した。ただし，4.1 節の数値実験により，提案手法ではエリート個体をランダムに配置した場合（手法 4）の方が，あらかじめ決められた場所に置く場合（手法 3）よりも良い性能を示していたため，本節では手法 3 を適用せず，手法 1，手法 2，手法 4 で性能を比較することにした。各手法の実装において，都市の訪問順をそのまま遺伝的アルゴリズムにおける個体にコード化した。実験条件に関する 4.1 節からのその他の変更点としては，交叉操作として一点順序交叉（図 9），突然変異操作として逆位（図 10）を用いた。個体の適応度の導出法として，以下の式が用いられた。

$$fitness_i(x) = \frac{Opt}{Distance(x)}, \quad i = 1, \dots, 400 \quad (10)$$

ここで， $fitness_i$ は i 番目の個体の適応度， x は都市の訪問順序を表すベクトル， Opt は各問題における最適（すなわち最短）距離， $Distance(x)$ はベクトル x に従って都市を巡回した場合の総巡回経路長である。

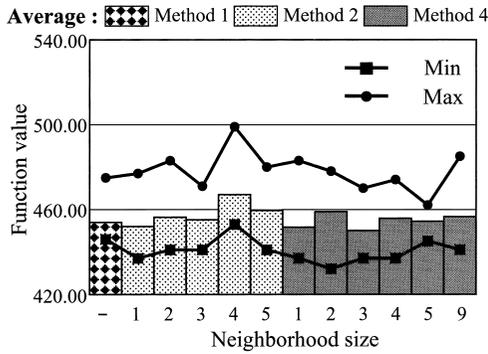


図 11 Eil51 に対する数値実験結果
Fig. 11 Simulation results on Eil51.

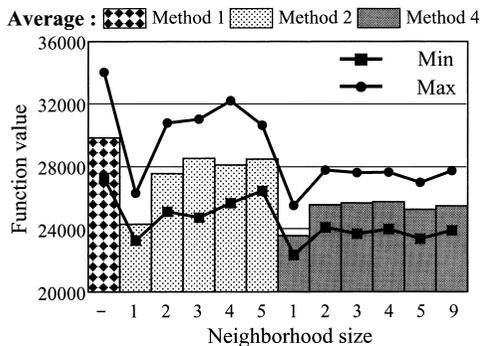


図 12 KroA100 に対する数値実験結果
Fig. 12 Simulation results on KroA100.

なお、交叉操作や突然変異操作としては、他の優れた操作も文献では報告されているが、各操作における性能をすべて調査することは本論文の趣旨からは外れている。ただし、他の交叉操作や突然変異操作を用いても同程度の性能が得られることは予備実験により確認済みである。

提案手法については、4.1 節と同様、図 3 で表されている構造を用いた。交叉確率を 1.0、突然変異確率を 0.1、終了基準を 1000 世代として手法 1、手法 2、手法 4 をそれぞれ 10 回実行し、各手法で得られた最終世代のエリートの性能を平均、Min、Max に分けて調べた。各問題に対して、手法 1、手法 2、手法 4 を 10 回実行させた後の数値実験結果を図 11 と図 12 に示す。

図 11 と図 12 から、51 都市問題 Eil51 では各手法間の差は大きくないが、100 都市問題 KroA100 では手法 4 によって得られた解が、他手法によって得られた解よりも良好であることが分かる。おそらく、51 都市問題 Eil51 は 100 都市問題 KroA100 ほど複雑な問題ではないので、手法間の性能において顕著な差は表

れなかったものと思われる。

以上のように、Simple genetic algorithm や標準のセルラー遺伝的アルゴリズムと比較した場合に、提案手法により良好な解が得られることが示された。しかし、依然として以下の問題が未解決のままである。

- (1) 島モデル遺伝的アルゴリズムと提案手法を比較した場合ではどうか。
- (2) 提案手法において、部分個体群の隣接状態(すなわち、トポロジー)を変化させた場合に性能に与える影響はどうか。
- (3) 提案手法がなぜ他の手法と比較して良好な結果を得ることができたのか。

次節以降では、これらの問題に対する考察を行うことにする。

4.3 島モデル遺伝的アルゴリズムとの比較

提案手法は、セルラー遺伝的アルゴリズムと島モデル遺伝的アルゴリズムの融合版である。4.1 節、4.2 節では、提案手法とセルラー遺伝的アルゴリズムとの比較が行われていたが、島モデル遺伝的アルゴリズムとの比較は行われていなかった。本節では、提案手法と島モデル遺伝的アルゴリズムの性能を比較することにする。島モデル遺伝的アルゴリズムとして、特に同期型と非同期型の両方の性能を調べ、提案手法と比較することにする。

まず、比較手法として用いた同期型と非同期型の島モデル遺伝的アルゴリズムの概要を述べる。同期型では、移住操作はある移住間隔ごとに行われるとする。同期間隔を 5, 10, 20, 50, 100, 300 世代とした。移住元から移住先の決定はランダムに行われるものとした。移住する個体の選択は、適応度を考慮したルーレット選択により移住元となる島の 50% の個体が選択された。また、移住先で新しい個体と置き換えられる個体として、移民となる個体の数と同じ数の個体がランダムに一樣選択で選択された。

非同期型では、各島内での解が収束したと判断された時点で移住を行うことにした。本論文では、ある一定世代以上の間エリート個体の性能が改善されなければ収束したと判断することにした。この収束判断世代として、5, 10, 20, 50, 100, 300 世代のそれぞれについて性能を調べた。この収束判断に用いられる世代数を、非同期型における移住間隔と呼ぶことにする。また、非同期型では、移住の方法は同期型よりも少し複雑になる。まず、最初に収束したと判断される島内で、適応度を考慮したルーレット選択により、島内の 50% の個体が選択される。選択された個体は、“待機所”と呼ばれるプールに自分をコピーする。このとき、

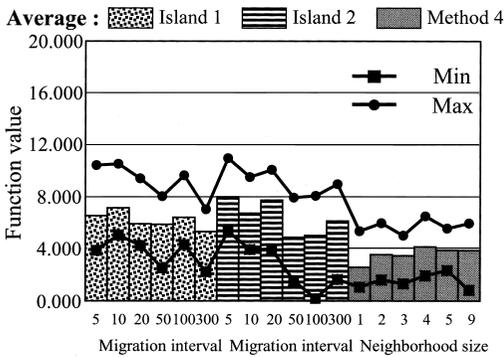


図 13 島モデル遺伝的アルゴリズムとの比較 (関数 F4)

Fig. 13 Comparison with island genetic algorithms on F4.

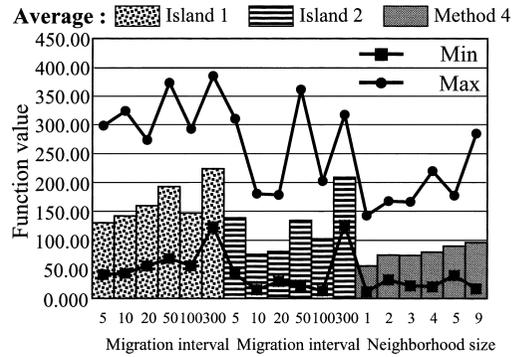


図 15 島モデル遺伝的アルゴリズムとの比較 (関数 F7)

Fig. 15 Comparison with island genetic algorithms on F7.

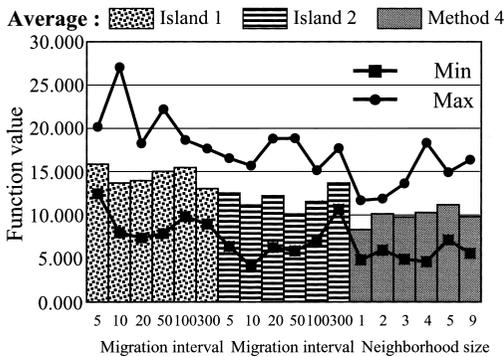


図 14 島モデル遺伝的アルゴリズムとの比較 (関数 F6)

Fig. 14 Comparison with island genetic algorithms on F6.

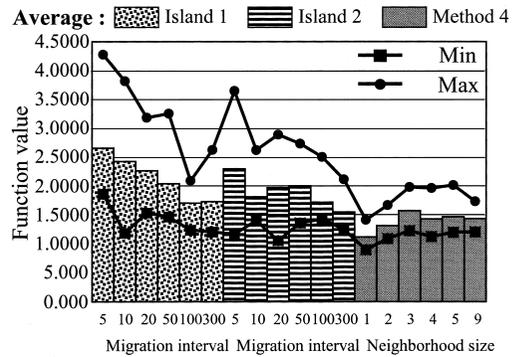


図 16 島モデル遺伝的アルゴリズムとの比較 (関数 F8)

Fig. 16 Comparison with island genetic algorithms on F8.

待機所に個体をコピーするだけであり、島内の個体を消去するものではないとする。また、ランダムに選択された島から一様選択により 50%の個体を島に取り入れ、一様選択により選ばれた島内の個体と入れ替えられる。待機所にいる個体の移住先は次に収束したと判断される島となる。移住先の島では、ルーレット選択により 50%の個体を待機所にコピーした後、一様選択により移住してくる個体の数だけの個体を選択され、新しい個体と入れ替えられる。

なお、移住間隔として、提案手法に近い形式である移住間隔 1 の場合の性能調査も必要であるが、予備実験を行ったところ、移住間隔が 1 の場合、他のどの移住間隔を用いた場合よりも性能が劣っていたため、本論文ではその性能を示さないでおくことにする。

上で述べた島モデル遺伝的アルゴリズムと提案手法の性能を比較することにする。実験のための問題として、4.1 節で用いた F1~F8 の関数に対して同期型と非同期型の島モデル遺伝的アルゴリズムをそれぞれ 10 回実行した。遺伝的操作や、遺伝的操作の確率、終了条件などは 4.1 節と同じとした。数値実験結果を

図 13~図 16 に示す。図 13~図 16 に示されている Method 4 とは、4 章での数値実験で用いられている手法 4 (提案手法) である。これらの図より、島モデル遺伝的アルゴリズムの性能は、セルラー遺伝的アルゴリズムよりも良いことが分かる (図 5~図 8 との比較より)、さらに、提案手法と島モデル遺伝的アルゴリズムの性能を比較すると、提案手法により、島モデル遺伝的アルゴリズムよりも良好な解を得ていることが分かる。

以上の結果と 4.1 節の結果より、セルラー遺伝的アルゴリズムと島モデル遺伝的アルゴリズムの融合である提案手法の性能は、元となる手法よりも良い性能が得られることが示された。

4.4 部分個体群のトポロジーについて

本節では、部分個体群間の隣接関係 (トポロジー) について議論する。これまでの数値実験では、図 3 のように、10x10 のセル構造を持つ 4 つの部分個体群が 2x2 で配置されていた。もし、この部分個体群の隣接関係を変更してもこれまでと同程度の性能が得られるのであれば、提案手法の有効性という点で、より

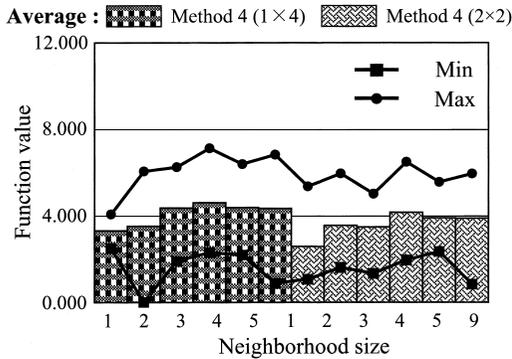


図 17 数値実験結果 (F4, トポロジー変更後)

Fig. 17 Simulation results on F4 (Topology changed).

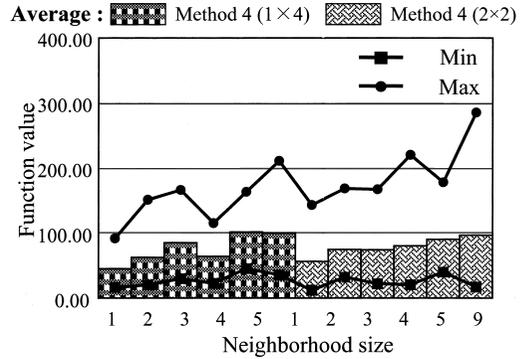


図 19 数値実験結果 (F7, トポロジー変更後)

Fig. 19 Simulation results on F7 (Topology changed).

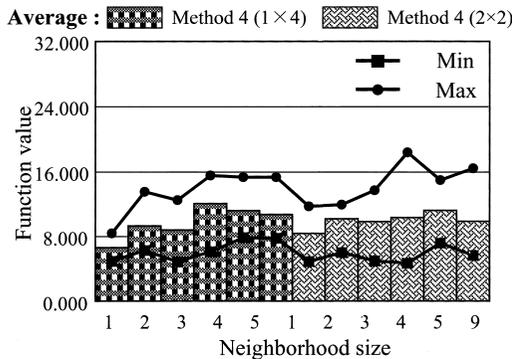


図 18 数値実験結果 (F6, トポロジー変更後)

Fig. 18 Simulation results on F6 (Topology changed).

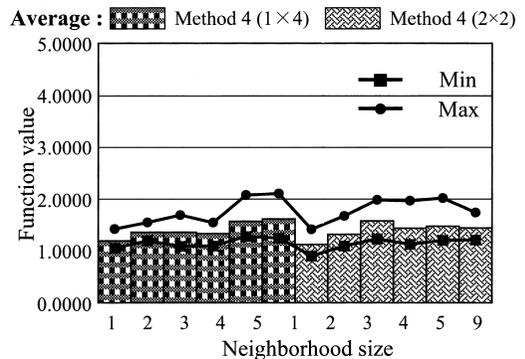


図 20 数値実験結果 (F8, トポロジー変更後)

Fig. 20 Simulation results on F8 (Topology changed).

説得力を増すはずである。

そこで、本節では、図 4 のように、部分個体群を 1×4 に配置し、その性能を調べることにする。図 4 で、塗りつぶされているセルが、部分個体群間で境界に位置しているセルを表し、これらのセルのみが、隣接している部分個体群と相互作用する。

トポロジーの変更以外には手法 4 とまったく同じ実験条件で、関数 F4, F6, F7, F8 に対する性能を調べた。各関数に対して 10 回の数値実験を行った結果を図 17~図 20 に示す。

図 17~図 20 より、トポロジーを変更した後の提案手法の性能は、変更前(図 3)のときと同程度の性能であることが分かる。このことは、提案手法は、トポロジーを 1×4 に変更した後でも Simple genetic algorithm や標準のセルラー遺伝的アルゴリズム、島モデル遺伝的アルゴリズムよりも良好な解が得られていることを意味している。

以上の結果からは、一般的に提案手法の性能がトポロジーとは独立であると結論付けることはできないが、少なくとも、1 つだけのトポロジーの場合にのみ提案

手法が有効であるとはいえないことが、本節の数値実験により示された。提案手法の性能がトポロジーに独立かどうかの一般的な判断を下すためには、より解析的な考え方をういて議論を行っていかねばならないが、それは今後の研究課題である。

4.5 近傍サイズが 1 の場合の解析

図 5~図 8 において、手法 4 の性能のみに注目すると、近傍サイズが 1 のときが一番良い性能であることが分かる。そこで、本節では、提案手法において近傍サイズが 1 のときに焦点を当てて議論を行うことにする。

提案手法では、部分個体群における境界のセルに配置されている個体の近傍サイズは 1 に固定されている。したがって、境界以外のセルに配置されている個体の近傍サイズが 1 であるときは、すべてのセルに配置されている個体の近傍サイズは 1 である。このとき、提案手法の振舞いは、標準のセルラー遺伝的アルゴリズムに非常に近くなる。唯一の違いは、提案手法では、エリート個体が各部分個体群ごとに選択され、部分個体群における次の世代に引き継がれるのに対し、セル

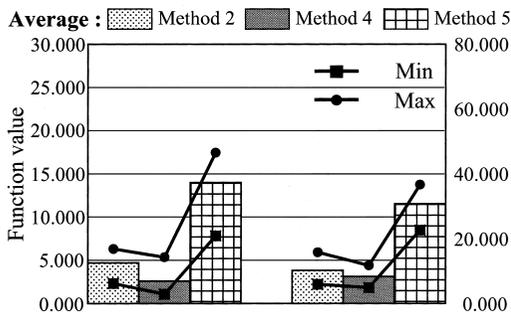


図 21 手法 2, 手法 4, 手法 5 の性能 (F4, F6)

Fig. 21 Performance of Method 2, Method 4, and Method 5 on F4 and F6.

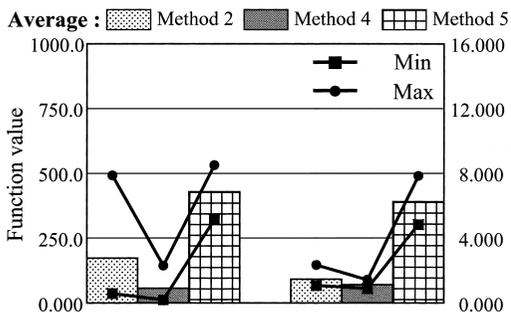


図 22 手法 2, 手法 4, 手法 5 の性能 (F7, F8)

Fig. 22 Performance of Method 2, Method 4, and Method 5 on F7 and F8.

ラー遺伝的アルゴリズムでは、個体群全体に対してただ 1 つのエリート個体が選択され、次世代に引き継がれる、という点である。

そこで、この違いが提案手法の有効性の原因となっているのかどうかを調べるために、以下の手法の性能を調べた。この手法では、個体群は分割されず、 20×20 のセル構造を持つ標準のセルラー遺伝的アルゴリズムが実行される。標準のセルラー遺伝的アルゴリズムとの違いは、エリート個体が個体群全体から適応度の高い順に 4 つ選択され、次世代の個体群生成後、それらがランダムな位置に配置される点にある。この手法を手法 5 と呼ぶことにする。

手法 5 を関数 F4, F6, F7, F8 の最適化問題に適用し、性能を調べた。10 回の数値実験を行った結果、図 21 と図 22 の結果が得られた。参考のために、図 21 と図 22 には手法 2 (エリート個体数が 1 の標準セルラー遺伝的アルゴリズム) と手法 4 (提案手法) の性能も比較のために再掲している。

図 21 と図 22 より、手法 5 の性能は著しく他の手法と比較して悪いことが分かる。すなわち、提案手法に比べて良好な解が得られなかったのはもちろん、エリート数が異なるだけの手法 2 と比べても、良好な解

が得られていないことが分かる。

この理由として、解の多様性の破壊が理由としてあげられる。手法 5 では、セル構造によって達成されている解の多様性が、エリート数を多くし、次の世代にも生き残ることで、たまたまアルゴリズム実行の初期段階で適応度の高かった個体が急速に広まり、いわゆる初期収束という現象を引き起こしやすい状況を作り出していると考えられる。

一方で、提案手法では、各部分個体群に対してエリート個体が 1 つ選択されるので、本実験における実装では、全体として手法 5 と同じく 4 つのエリートが選択されていることになる。しかし、提案手法におけるエリート個体は、各部分個体群内という局所的な意味合いを持つ。さらに、各部分個体群で選択されたエリート個体は、次世代において同じ部分個体群の中でしか保存されない。個体群全体に広まるためには、境界のセルが異なる部分個体群へエリート個体の情報を伝える必要があり、これがエリート個体の急速な広がり抑制していると考えられる。結局、提案手法は解の多様性を維持しつつ、局所的な意味合いを持つ複数のエリート保存戦略を利用することで高い探索能力を得ていることが分かる。

以上のように、エリートの取扱いが重要となるのは、提案手法だけでなく、セルラー遺伝的アルゴリズムや島モデル遺伝的アルゴリズム、Minimal Generation Gap モデル¹⁴⁾ などにもいえる。

5. おわりに

本研究では、セルラー遺伝的アルゴリズムと島モデル遺伝的アルゴリズムを融合させた、セル構造を有する分散遺伝的アルゴリズムを提案した。提案手法では、個体群はいくつかの部分個体群に分割され、各部分個体群にはセル構造が与えられ、それぞれが独立にセルラー遺伝的アルゴリズムを実行する。部分個体群間の境界に位置しているセルのみが他の部分個体群との相互作用をすることができる。

関数最適化問題や順序組合せ最適化問題を用いた数値実験により、提案手法の性能は、Simple genetic algorithm や標準のセルラー遺伝的アルゴリズムよりも良好な解を探索することが可能であることが示された。また、同期型、非同期型の島モデル遺伝的アルゴリズムとの比較も行った結果、提案手法は、基本となる 2 つのアルゴリズム (セルラー遺伝的アルゴリズムと島モデル遺伝的アルゴリズム) よりも性能が良いことが示された。

また、追加実験を行い、提案手法における部分個体

群の隣接関係(トポロジー)を変更しても,提案手法の性能に大きな影響を与えないことが示された.さらに,近傍サイズが1の場合には,標準のセルラー遺伝的アルゴリズムとよく似た振舞いであるにもかかわらず,最も良い性能を示すことに着目し,近傍サイズが1であり,かつ,エリート個体の選択方法を変更した手法の性能調査も行った.実験の結果,個体群を分割することが,解の多様性を維持するために必要であり,また,解の多様性を維持することにより探索性能が向上することが示された.

遺伝的アルゴリズムを分散化する際,分散化の長所を最も生かすことのできる方法の1つは,並列計算機を用いたアルゴリズムの高速化であろう.しかし,本研究では,アルゴリズムを分散化すること自体が探索性能に与える影響に焦点を当てて研究を行った.したがって,本論文の目的は,並列計算を行って高速処理を行うことではなく,探索に優れたアルゴリズムの分散化の手法を提案することにあった.

提案手法に関する残された課題として,提案手法の有効性の解析を理論的に行うことが考えられる.セルラー遺伝的アルゴリズムに対する理論的解析としては,De Jongら⁵⁾やBaluja¹⁾の優れた研究があり,これらの研究で用いられている解析方法が提案手法にも適用できると思われる.また,部分個体群ごとに,近傍サイズや遺伝的操作の確率などを様々な値に設定し,部分個体群間に多様性を持たせた場合の探索性能の調査も興味深い研究と思われる.

参 考 文 献

- 1) Baluja, S.: Structure and Performance of Fine-Grain Parallelism in Genetic Search, *Proc. 5th International Conference on Genetic Algorithms*, pp.155-162 (1993).
- 2) Belding, T.C.: The Distributed Genetic Algorithm Revisited, *Proc. 6th International Conference on Genetic Algorithms*, pp.114-121 (1995).
- 3) Cantú-Paz, E.: A Survey of Parallel Genetic Algorithms, *IlliGAL Report*, No.97003 (1997).
- 4) DeJong, K.A.: Analysis of the Behavior of a Class of Genetic Adaptive Systems, Ph.D. Dissertation, University of Michigan (1975).
- 5) De Jong, K. and Sarma, J.: On Decentralizing Selection Algorithms, *Proc. 6th International Conference on Genetic Algorithms*, pp.17-23 (1995).
- 6) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, Massachusetts (1989).
- 7) Holland, J.: *Adaptation in Natural and Artificial Systems*, Ann Arbor, the University of Michigan Press (1975).
- 8) Manderick B. and Spiessens P.: Fine-Grained Parallel Genetic Algorithms, *Proc. 3rd International Conference on Genetic Algorithms*, pp.428-433 (1989).
- 9) Gorges-Schleuter, M.: ASPARAGOS An asynchronous parallel genetic optimization strategy, *Proc. 3rd International Conference on Genetic Algorithms*, pp.422-427.
- 10) Mülenbein, H.: Parallel Genetic Algorithms, Population Genetics and Combinatorial Optimization, *Proc. 3rd International Conference on Genetic Algorithms*, pp.416-421 (1989).
- 11) Mühlenbein, H., Schomisch, M. and Born, J.: The Parallel Genetic Algorithm as Function Optimizer, *Proc. 4th International Conference on Genetic Algorithms*, pp.271-278 (1991).
- 12) Reinelt, G.: *TSPLIB*, available from the Internet. <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>
- 13) Tanese, R.: Distributed Genetic Algorithms, *Proc. 3rd International Conference on Genetic Algorithms*, pp.434-439 (1989).
- 14) Satoh, H, Yamamura, M. and Kobayashi, S.: Minimal Generation Gap Model for GAs Considering Both Exploration and Exploitation *Proc. IIZUKA '96*, pp.494-497 (1996).

(平成 14 年 2 月 5 日受付)

(平成 14 年 5 月 3 日再受付)

(平成 14 年 7 月 6 日採録)



中島 智晴

1971 年生. 2000 年大阪府立大学大学院工学研究科電気・情報系専攻博士後期課程修了. 同年大阪府立大学大学院工学研究科助手. 2001 年同講師. 博士(工学). 1998 年南オーストラリア大学客員研究員. データマイニング, 遺伝的アルゴリズム, 機械学習, ゲーム理論等の研究に従事. IEEE, 計測自動制御学会, 日本ファジィ学会, 日本経営工学会各会員.



有山 貴信

1978年生。現在、大阪府立大学工学部在学中。遺伝的アルゴリズム、学習エージェントの研究に従事。日本ファジィ学会、計測自動制御学会、システム制御情報学会各会員。



石淵 久生(正会員)

1963年生。1987年京都大学大学院工学研究科修士課程修了。同年大阪府立大学工学部助手。1999年同教授。2000年同大学院工学研究科教授。博士(工学)。1994年~1995年および1997年~1998年トロント大学客員研究員。1996年度日本経営工学会論文奨励賞受賞。データマイニング、遺伝的アルゴリズムの応用、マルチエージェント等に関する研究に従事。IEEE, IFSA, INNS, 日本ファジィ学会、日本経営工学会各会員。
