

注視領域の適応的決定機構を有する遺伝的アルゴリズム

坂 無 英 徳[†] 鈴木 恵 二[†] 嘉 数 侑 昇[†]

自然界における生物の適応機構をモデルとして開発された遺伝的アルゴリズム (Genetic Algorithm; GA) は, 探索性能のロバスト性において, 現在最も強力な手法の1つである。しかし, けっして万能ではなく, GA 困難と呼ばれる問題クラスが存在が知られており, これを克服することが GA 研究の主要なテーマとなっている。一方, 一般の探索行為を考えると, 探索空間を広範に見渡す大域的な注視と, 有望な領域内を詳細に調査するための局所的な注視の両者をバランスよく実行することが重要である。しかしながら, この探索戦略を計算機上に実装するためには, 計算時間や記憶量に関する様々な問題がともなう。そこで本研究では, このような注視領域の適応的な制御・決定機構と GA とを互いに補完し, GA の探索性能向上を目指した枠組みを構築する。具体的には, 注視すべき領域そのものの探索と, 領域内の詳細な探索を担う2つの GA を並列に動作させる。このときに重要なのは, 並列的に動作する両者を有機的に相互作用させる機構であり, これが適切に機能することによって, 大域的・局所的注視のバランスが適応的に変化する。論文中で提案する手法である分散型フィルタリング GA (Distributed Filtering-GA; DFGA) では, (1) 注視すべき領域の位置, (2) 領域内の探索の完了度, (3) そこに含まれる探索点の評価値などを圧縮保存可能な機構を実現し, これを介して2つの GA の円滑な情報交換を可能にしている。以下では, このような問題背景に関する詳しい議論を行った後, GA において注視領域の制御機構がどのように実現されているかを概観する。次いで, 本枠組みの基本概念, および実際の多点探索手法に実装するためのアプローチについて検討し, DFGA を提案する。計算機実験においては, GA 困難クラスに属する3タイプの最適化問題に提案手法を適用し, その有効性の確認, さらには本方法論の妥当性について考察する。

Adaptive Attention Control in Genetic Algorithm

HIDENORI SAKANASHI,[†] KEIJI SUZUKI[†] and YUKINORI KAKAZU[†]

Genetic Algorithm (GA) is inspired from the adaptive behaviors of organisms in the natural world, and is one of the most powerful and robust search procedures. It is well known, however, that there is a class of problems, called GA-hard problems, and the canonical GA can hardly solve the problems in the class. To overcome this difficulty is one of the major subjects in the research field of GA. On the other hand, considering about efficient and general search method, it is important to properly change the balance of executing phases for searching with globally wide attention and locally precise attention. However, there are many problems for implementing this search strategy in the computer, concerning costs for memory, calculation time, and so on. In this paper, therefore, we attempt to resolve their difficulties using their advantageous features, and to construct the advanced framework of GA. In concrete, the framework adopts two GAs, for searching the regions which is expected to contains good search points, and for exploring in the regions precisely. The important thing of the framework is the mechanism to interact the both attentions in the organic way, and the global and local attentions can be executed adaptively when it works properly. In reality, the Distributed Filtering GA (DFGA), which is proposed in this paper, can smoothly interact two GAs through the mechanism for storing information about the position of the targeting regions, the degree of focusing attentions, and the degree of their hopefulness. In this paper, to realize the adaptive attention control, we argue the methodology for introducing the organized social dynamics into the multi-points search procedure, and propose the advanced framework named DFGA. Some results of computer simulations, where DFGA is applied to some GA-hard problems, exhibit characteristics of search dynamics of DFGA, demonstrate its efficiency, and reveal some problems to be overcome in future.

1. はじめに

我々が何らかの物を探そうとするとき, まず搜索範

囲全体を大雑把に見渡し, 有望と思われるところから順に詳細に調べていく。そしてこのように, 大域的な注視領域の決定と, 局所的に詳細な調査を適切に繰り返すことは, 効率的に探索を行うための単純でありながら強力な戦略の1つである。しかし搜索範囲全体を見渡すために計算機内部でデータ構造全体を把握する

[†] 北海道大学工学部

Faculty of Engineering, Hokkaido University

ことは、計算時間的にも、記憶量的にも困難である。つまり、この戦略をそのまま一般の工学的な探索問題へと持ち込むためには、現実的には様々な創意工夫を必要とする。

大域注視を行うためには多点探索が有効であるが、探索者間で円滑な協調が行われたとき、より効率的な探索が可能となる。さらには、協調を適切に行うための優れた戦略を持つ管理者の存在によって、より簡単に最適解を発見できると期待される。換言すれば、

- 大域注視に基づく大域的探索を担う管理者と、局所注視により局所的探索を行う探索者として役割を分担、単純化し、
- 大域注視が局所注視に対して制御/制約を施し、逆に局所注視が大域注視に影響を与え、修正することによって、適応的な注視領域制御を実現することができる。

本研究の目的は、このような管理者と探索者による社会的な組織化ダイナミクスを多点探索に導入し、大雑把な大域注視と、詳細な局所注視のバランスを、組織全体として自律的かつ適応的に調整するための方法論を構築することである。そしてこれを遺伝的アルゴリズム (Genetic Algorithm; GA) の枠組みにおいて実現することを試みる。以降ではまず、従来法 GA で注視領域の制御機構が、どのように実現されているかを概観する。次いで、本方法論の基本概念、および GA に実装するためのアプローチに関して詳しい議論を行う。その後、分散型フィルタリング GA (Distributed Filtering-GA; DFGA) を提案し、その詳細を説明する。計算機実験においては、いくつかの特徴的な最適化問題に提案手法を適用し、その有効性の確認、さらには本方法論の妥当性について考察する。

2. GA における注視領域の制御

GA は自然界、あるいは人工システムにおいて見られる適応過程をモデル化するために提案され¹⁴⁾、その後、様々な最適値探索問題に適用されたことで優れた探索能力を示してきた手法である⁵⁾。近年では人工生命研究などで、自律エージェントの進化モデルにおける適応的行動戦略獲得のための中枢機構として用いられるケースも増加している²⁾が、本論文では狭義の探索手法としての GA を対象とする。

最も一般的なタイプの単純 GA では、探索空間内の探索点 X_i を符号化して文字列 S_i で表現し、それを集団として取り扱う。本論文では、これらの関係を以下のように定義、記述する。

$$Pop \in [[S_i]]^N, S_i \in \{s_{ij}\}^l, s_{ij} \in \{0, 1\}, \quad (1)$$

$$S_i = D^{-1}(X_i),$$

$$F(X_i) = F(D(S_i)) = G(S_i) \quad (2)$$

ここで、 Pop は集団、 N は集団の構成メンバ数、 l は文字列長、 D は遺伝子型から表現型への復号化関数、 F と G は表現型と遺伝子型における目的関数である。

単純 GA の動作は、初期集団をランダムに生成し、評価、および選択、交叉、突然変異などの遺伝的操作を反復することで進行する。ここでランダムに生成される初期集団は、大域注視を実現していると思えることができる。その後、選択によって集団内の各メンバ相互の類似度が高まり、注視領域が限定されていく。また交叉によってメンバ間で情報交換を行い、突然変異が集団に新たな要素を導入することで、絞りつつある注視領域は移動、あるいは拡大する。このダイナミクスはスキーマ定理 (Schema Theorem)¹⁰⁾として知られているが、一般的に不可逆で一度縮小した注視領域が再び拡大することはほとんどない。つまりサンプリング誤差のために局所解へ収束してしまうと、最適解を発見することはほぼ不可能となる。

このような注視領域制御のための単純 GA の不完全な戦略を改善するために、これまでに数多くの拡張型 GA が提案されてきた (表 1)。

タイプ I: 特殊な遺伝的操作により保持される集団中の豊富な多様性により大域注視を維持。

タイプ II: 注視領域が収束したときに、適切に集団を再初期化して、大域注視を回復。

タイプ III: 遺伝子座の順序も同時に最適化することで、注視領域の最適解近傍への確実な収束を促進。

タイプ IV: 擬似的に生成する小集団が異なる領域に対して局所注視を実行。

タイプ V: 拡張コーディングにより、各メンバが複数の局所注視を実行可能。

表 1 拡張 GA

Table 1 Examples of extended GA.

Type I	Incest Prevention ⁷⁾ Adaptive Mutation ²⁰⁾
Type II	Sequential Niching ¹⁾ Selectively Destructive Re-start ¹⁵⁾
Type III	Messy GA ¹²⁾ SYMBIOT ¹⁷⁾
Type IV	Eco-Framework ⁴⁾ Decentralizing Selection ⁶⁾ Niche Model ⁸⁾
Type V	Structured GA ³⁾ Diploidy Model ⁹⁾ Neutral Mutation Model ¹⁶⁾

これらのうち、あるものは非常に強力であり、またあるものは興味深いコンセプトを採用している。しかしその多くは、計算時間や記憶量に関して莫大な計算コストを要し、パラメータ設定に対して鋭敏すぎ、対象問題領域に関する先験的知識を必要とするなどのいくつかの問題点を抱えている。これに対し次章で提案するアプローチは、対象問題やパラメータ設定に依存しないロバストな探索戦略の実現を目指している。

3. 組織的ダイナミクスによる注視領域制御

探索空間に関する先験的知識を用いずに探索するということは、濃霧の中で登山をするようなものである。このとき単独の探索者だけでは、自分が現在いる場所の状況しか観測できず、地勢的に遠く離れた場所の情報を把握することはできない。そこで探索者は、自分の探索履歴を長期的に記憶し、参照しながら次に調査すべき地点を決定しなければならない。しかし、効率的に探索を進めるための履歴情報の保存量は、探索空間上に評価関数が生成する曲面形状の複雑性に依存する。完全なランダム探索によるアプローチ以外の手法では、これが不足すると最適解の発見はほとんど不可能になり、過剰の保存は探索効率の悪化を招く。これに対して、複数の探索者が情報を交換しつつ、協調あるいは競争しながら探索できれば、増大する履歴情報を分担して保存することが可能になる。このとき、探索者間での相互作用が適切ならば、群内で自律的に保存すべき履歴情報の取捨選択を行うことが可能になるものと期待できる。また、全体の情報を総合しつつ各探索者を適切に統制できる管理者により、探索者の行動決定の精度が向上し、目的遂行はより簡単になるであろう。このとき各探索者は、互いに情報交換し、それぞれの現状と管理者の指示に基づいて決定した行動の優劣を競うことになる。

ここでさらに探索者だけでなく、複数の管理者に協調と競争を行わせることを考える。これにより管理者群は、情報交換と探索者に対する指示の適切さに基づく競争を経て、適切な統制方法を獲得していく。またこのとき、管理者と探索者が互いの組合せを自由に変更できれば、全体としてより効率的に探索することが可能になるものと期待される。

本研究では上記の社会的な組織化ダイナミクスを現実のものとするために、GAを管理者および探索者による2種の集団により構成する(図1)。ただし、探索点ではなく、探索領域を取り扱う必要がある管理者の表現方法として、スキーマ表現を導入する。スキーマとは遺伝子型空間における距離の近い探索点による

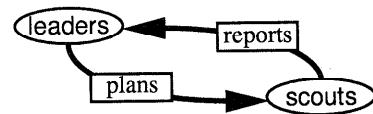


図1 提案手法の基本概念
Fig. 1 Basic concept.

集合を示す概念である。通常、GA集団の構成メンバーが2進表現される場合、スキーマ H は $\{0, 1, \#\}^l$ で表現される。ここで $\#$ は0と1の両方に対応する記号であり、スキーマ中の $\#$ 以外の要素数はオーダと呼ばれる。すなわちスキーマを管理者と見なすとき、オーダ次数が小さいということは、広い探索領域を大域注視していることを意味し、逆に大きい場合は範囲を絞った注視を行っていることを示す。そしてオーダが最大のスキーマは、探索者としての文字列と同様に、探索点を局所注視することとなる。

探索者は管理者によって計画され、有望であると判断された領域内で局所探索を行う。管理者は、探索者から報告される結果に基づき、有望な領域を求めて探索する。すなわち、管理者が決定する計画は探索者を制御し、探索者が提出する報告は計画がどの程度適切であったかの指標となる。しかしこのとき、探索者と管理者のすべての組合せを考慮して相互作用を行うことは、各集団サイズが大きくなるにつれて、莫大な計算負荷を招く。そこで、管理者は探索者の一部に、探索者は管理者の一部にしか干渉できないように、集団間の相互作用を制限する必要がある。ところがこの方法では、探索者と管理者の相互作用対象をどのように決定するのかという問題が生じる。この新たな問題に対して、本研究では現実的な解決方法として、前章で分類したタイプIVの拡張GAのうち、Decentralizing Selection⁶⁾の枠組みを導入し、それぞれの集団を同じ構造を持つ2つの格子平面上に配置する。このとき、選択の際の適応度計算、交叉におけるペア決定などの集団内における相互作用は、格子空間における各メンバーの9近傍内で局所的に行われる(図2(a))。たとえば、各メンバー S の適応度 f は、評価関数を G 、格子空間上において S を含む S の9近傍を Γ とすると、次式のように計算される。

$$f = \left\{ \frac{S - G_{\min}}{\sum_{S_k \in \Gamma} \{G(S_k) - G_{\min}\}} \right\}^{P_{prs}} \quad (3)$$

ここで G_{\min} は Γ における最小の評価値、 P_{prs} は淘汰圧を決定するパラメータである。後述の実験では $P_{prs} = 1$ としているが、これが大きくなるにつれて

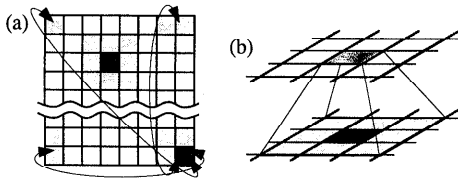


図2 集団内および集団間における構成要素の相互作用
Fig. 2 Interactions on a plane and between two planes.

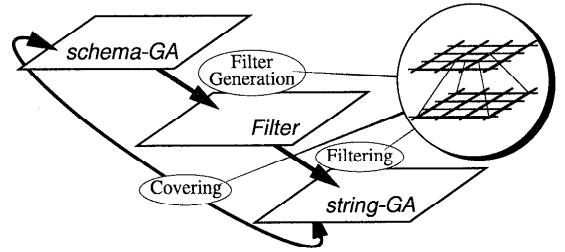


図4 DFGAにおける構成要素間の相互作用
Fig. 4 Interaction between components of DFGA.

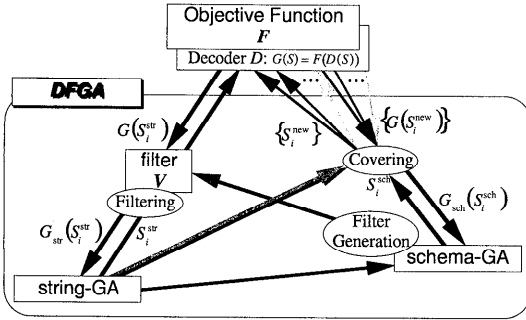


図3 分散型フィルタリング GA の構成
Fig. 3 Architecture of Distributed Filtering-GA.

淘汰圧が高くなり、進化速度が向上する半面、集団が収束状態に陥りやすくなる。

また、探索者と管理者の相互作用、すなわち集団間の相互作用は、ある平面のメンバに対して、他の平面上のあるメンバを中心とした9近傍内から対象が選択される(図2(b))。なお、他平面上で中心となる対象メンバは、それぞれが独自に指定し、変更可能とするが、この手続きの詳細は次章で述べる。

4. 分散型フィルタリング GA

本章では、これまでに議論を行ってきた方法論を実現するために、拡張型 GA を提案する。以降では提案手法を分散型フィルタリング GA (Distributed Filtering-GA; DFGA) と呼ぶ。

4.1 DFGA の基本構成

DFGA は3つの基本要素と、それらの間の3つの相互作用から構成される(図3)。基本要素はそれぞれが管理者と探索者の集団、そして管理者が立案する計画に相当する Schema-GA と String-GA フィルタである。大域注視を担う Schema-GA と局所注視を行う String-GA の集団、 Pop^{sch} と Pop^{str} は、次式のように定義される。

$$Pop^{sch} = \left[[S_i^{sch}] \right]^{N^{sch}},$$

$$S_i^{sch} \in \{s_{ij}^{sch}\}^l, s_{ij}^{sch} \in \{0, 1, \#\}. \quad (4)$$

$$Pop^{str} = \left[[S_i^{str}] \right]^{N^{str}},$$

$$S_i^{str} \in \{s_{ij}^{str}\}^l, s_{ij}^{str} \in \{0, 1, \#\}. \quad (5)$$

ここで Pop^{sch} と Pop^{str} は、それぞれスキーマとストリングによって構成され、 N^{sch} と N^{str} は両者の集団サイズである。なお、両 GA において、選択と交叉は各メンバの8近傍内で実行され、突然変異は単純 GA とまったく同様に実施される。

フィルタ V は、 Pop^{str} のメンバが $\{0, 1\}$ の2値で符号化されていることに対応して、2つの実数値配列によって構成され、以下のように定義される。

$$V = \left[[V^k] \right]^{N^{flt}},$$

$$V^k = \{V_i^k\}^2, V_i^k \in \{v_{ij}^k\}^l, v_{ij}^k \in R. \quad (6)$$

ここで N^{flt} はフィルタの数である。なおフィルタは、すでに述べた2つの集団と同一の位相構造を持つ格子平面上に配列し、 $N^{flt} = N^{sch} = N^{str}$ とする。すなわち、DFGA は同一の位相構造を持つ3つの格子平面によって構成される(図4)。

4.2 DFGA 構成要素間の相互作用

DFGA におけるメンバ間の相互作用方式の基本は、格子平面の位相構造に基づく局所作用である。そこで構成要素間の相互作用に関する詳しい説明を行う前に、平面上に配置されるメンバの関係の表現方法を定義する。あるメンバ M_i と同一平面上で、その8近傍に属する全メンバを $ngbr(M_i)$ と表す。また M_i が指定し、他平面上での相互作用の中心となるメンバを $crspnd(M_i, char)$ と表現する。 $char$ には文字列 $\{sch, str, flt\}$ が代入され、それぞれ対象メンバが Schema-GA, String-GA フィルタに属することを示す。

4.2.1 フィルタ生成 (計画立案)

フィルタの目的は、大域注視を担う Schema-GA と、局所注視を行う String-GA 間の適切な情報交換と、円滑な相互作用を支援することにある。これは毎世代、Schema-GA の集団 Pop^{sch} から、次式に基づいて生

成される.

$$v_{ij}^k = \frac{1}{|E_1|} \sum_{S_i^{\text{sch}} \in E_1} \left\{ \frac{G_{\text{sch}}(S_i^{\text{sch}})}{o(S_i^{\text{sch}})} \times \text{comp}(i, s_{ij}^{\text{sch}}) \right\}, \quad (7)$$

$$E_1 = \left\{ \text{crspnd}(V^k, \text{sch}) \right\} \cup \text{ngbr}(\text{crspnd}(V^k, \text{sch})), \quad (8)$$

$$\text{comp}(a, b) = \begin{cases} 1; & \text{if } a = b, \\ 0; & \text{otherwise.} \end{cases} \quad (9)$$

ここで E_1 は Pop^{sch} 内の小集団, G_{sch} は後述する Schema-GA の評価関数である. この式は, 評価値が高いほど, また文字列中に#の数が多い S_i^{sch} ほど, フィルタ生成に大きな影響を与えることを示す. さらに, 文字列中の特定の遺伝子座に#以外の特定の値を持つメンバが E_1 中に多いほど, フィルタの対応する配列要素に大きな値が格納される. すなわちフィルタは, 小集団 E_1 が大域注視を行うべきだと特定した探索領域の位置や, #の数で特定される領域の広さ, 注視強度すなわち E_1 の収束度合などに関する情報を, 圧縮して保持することが可能となる.

4.2.2 フィルタリング (計画実行)

この操作は, 大域注視を実現するための機構に相当し, 次式に示す String-GA の評価関数によって実現される.

$$G_{\text{str}}(S_i^{\text{str}}) = G(S_i^{\text{str}}) - \frac{1}{|E_2|} \sum_{V^k \in E_2} \sum_{j=0}^l v_{s_{ij}^{\text{str}}}^k, \quad (10)$$

$$E_2 = \left\{ \text{crspnd}(S_i^{\text{str}}, \text{flt}) \right\} \cup \text{ngbr}(\text{crspnd}(S_i^{\text{str}}, \text{flt})). \quad (11)$$

ここで E_2 はフィルタの小集団, G は最大化の対象となる目的関数で, 第2項目は探索者 S_i^{str} が, 管理者に相当する Schema-GA のメンバ群によって立案された計画を, フィルタから引き出す操作に対応する. 第2項目の値が大きいうことは, Schema-GA が, 強い大域注視を行っている領域中に探索者 S_i^{str} が存在している可能性が高いことを意味する. すなわち, Pop^{sch} (正確には E_2) 中に S_i^{str} が示す探索点を含む領域を示すメンバが多く存在, あるいはそのような領域が Schema-GA において高く評価されていることを示す. そしてこのような領域内の探索点は, 次項で述べるカバーリング操作によりサンプルすることが可能である. つまりそのような領域内を, あらためて探

索する必要性は低いため, 第2項目の値を引算することにより, String-GA はそのような領域を避けながら探索を進めることが可能となる.

4.2.3 カバーリング (結果報告)

カバーリング $\text{cover}(H, S)$ は String-GA メンバ S が示す探索点を, Schema-GA メンバ H が特定する領域内へと写像し, 局所注視を促進させるための操作で, 次式に示す Schema-GA の評価関数 G_{sch} に内蔵されている.

$$G_{\text{sch}}(S_i^{\text{sch}}) = \max_{S_k^{\text{str}} \in E_3} \left\{ G(\text{cover}(S_i^{\text{sch}}, S_k^{\text{sch}})) \right\}, \quad (12)$$

$$E_3 = \left\{ \text{crspnd}(S_i^{\text{sch}}, \text{str}) \right\} \cup \text{ngbr}(\text{crspnd}(S_i^{\text{sch}}, \text{str})), \quad (13)$$

$$S^{\text{new}} = \text{cover}(H, S), \quad (14)$$

$$s_j^{\text{new}} = \begin{cases} h_j; & \text{if } h_j \neq \#, \\ s_j; & \text{otherwise.} \end{cases} \quad (15)$$

$$H = \{h_j\}^l, \quad h = \{0, 1, \#\}, \\ S = \{s_j\}^l, \quad s = 0, 1, \quad (16)$$

ここで E_3 は Pop^{str} 内の小集団である. Pop^{sch} のメンバは探索領域を特定しているが, その有望度を測定するために領域内の全探索点をサンプルするのは非効率的である. そこで Pop^{str} から関連性の強いメンバが示す探索点をカバーリングにより写像し, 代表サンプル点として評価し, その最大値を S_i^{sch} が局所注視を行うべきだとした領域の有望度とする.

4.2.4 集団間作用対象の変更

DFGA における3つの構造間の相互作用対象は, 初期状態では, それぞれの構成要素が配置される各平面の同一位相の格子に属するメンバに設定される. その後, この作用関係は, 各メンバの利己的な基準によって変更されていく. ここでは, その変更手続について説明する.

これまでに行った説明からも明らかなように, 各構造は, 一方方向のサイクリックな相互依存関係を持つ. また本研究では, Schema-GA とフィルタの関係 $\text{crspnd}(V, \text{sch})$ は固定のため, $\text{crspnd}(S^{\text{str}}, \text{flt})$ を $\text{crspnd}(S^{\text{str}}, \text{sch})$ としても差し支えない. よって, ここで考慮するのは, $\text{crspnd}(S^{\text{sch}}, \text{str})$ と $\text{crspnd}(S^{\text{str}}, \text{sch})$, つまり Schema-GA と String-GA の相互関係だけでよい.

管理者は, 自分が立案した計画を基に, より優れた報告を返す探索者と結びつくことで, 自らも高い評価値を受け取ることができる. そこで E_3 中で式 (12) の最大値の生成に直接貢献した S_k^{str} へと $\text{crspnd}(S^{\text{sch}}, \text{str})$

を変更する。探索者の場合は、自分をより高く評価してくれる管理者と契約を結ぶことで、他の探索者との競争が有利になる。よって式 (10) において最小の $\sum_{j=0}^l v_{s_{ij}^{str}}^k$ を示したフィルタ V^k を生成した Schema-GA のメンバ S_k^{sch} へと $\text{crspnd}(S^{str}, \text{sch})$ を変更する。

しかしあまりに頻繁な相互作用対象の変更は、各集団内での競争を不安定化し、メンバの進化を妨げる恐れがある。そこで、次式を満たす管理者 S^{sch} と探索者 S^{str} だけが、実際に相互作用対象を変更可能とする。

$$\text{crspnd}(\text{crspnd}(S^{sch}, \text{str}), \text{sch}) = S^{sch}, \quad (17)$$

$$\text{crspnd}(\text{crspnd}(S^{str}, \text{sch}), \text{str}) = S^{str}. \quad (18)$$

これによって、強く希望し合う管理者と探索者のペアだけの変更できるようになる。

4.3 計算コスト

集団サイズが N_S の単純 GA において 1 回の評価を行うためには、全メンバを最低 1 度ずつ走査しなければならず、このコストを N_S とする。選択、交叉、突然変異もそれぞれ同様に N_S のコストが必要であるため、合計コストは $4 \times N_S$ となる。

これに対し、我々が以前に提案した拡張型 GA である Filtering-GA (fGA) と、単純 GA とのコスト比を計算すると次式のようになる¹⁸⁾。

$$\frac{\text{cost}(\text{fGA})}{\text{cost}(\text{canonical-GA})} \approx \frac{1}{4} + \frac{9}{4\sqrt{N_S}}. \quad (19)$$

なお、DFGA は fGA をベースに開発されており、基本的なアーキテクチャや採用オペレータの機構そのものは同一であるが、DFGA への decentralizing selection⁶⁾ の枠組みの導入にともない、

- オペレータが集団中、集団間で局所的に動作すること (3 章, 4.1, 4.2 節) や、
- さらにその動作におけるメンバ間の相互作用関係が適応的に変化すること (4.2.4 項)、さらに
- 単純 GA には存在しないパラメータを排除したこと (5 章の表 2 参照のこと)

などが主な相違点である。fGA の詳細に関しては文献 21) を参照されたい。

さらに fGA と同じ集団サイズの DFGA を考えると、選択、交叉、突然変異、String-GA の評価に必要なコストは単純 GA の場合と同様で $N_F (= \sqrt{N_S})$ である。しかしカバーリング、フィルタリング、フィルタ生成は局所的にしか行われなため、それぞれのコストはともに $9 \times N_F$ となる。よって、単純 GA との合計コスト比は

$$\frac{\text{cost}(\text{DFGA})}{\text{cost}(\text{canonical-GA})} \approx \frac{17}{2\sqrt{N_S}}, \quad (20)$$

となる。

非常に大雑把な概算ではあるが、これらの式からは以下の事実が導かれる。(1) fGA は、単純 GA が大サイズの集団を必要とする問題、 N_S が大きくなるほど、少ないコストで探索を行うことが可能。(2) DFGA は小規模な問題においても単純 GA 程度で、大規模な問題では fGA よりも少ない計算量で探索が可能。なお、これらの結果については、計算機実験において検証する。

5. 計算機実験

本章では、提案手法の有効性を検証し、その挙動の観測結果から特性を議論するため、計算機実験を行う。比較対象としては単純 GA と、前章の最後に触れた Filtering-GA (fGA) を用いた。

最適化対象として、以下に示す 3 つの問題を用意する。まず、第 1 問 G_1 は文献 19) で定義された騙し関数である。

$$G_1(S) = \begin{cases} 360 \frac{D_C(S)}{l/2}; & \text{if } \sum_{j=1}^l = 0, \\ 300 \frac{D_C(S)}{l}; & \text{otherwise,} \end{cases} \quad (21)$$

ここで $D_C(S) = \sum_{j=1}^l S_j$ である。多くの騙し関数では、遺伝子型空間で最適解からのハミング距離が最大となる地点に騙し解が存在する。そのような問題では、集団内のメンバを評価する際、単純にビット列をすべて反転した解を生成し、それらも同時に評価することで簡単に発見できてしまう¹³⁾。しかしながら G_1 に対して、そのような単純な戦略は通用しない。

第 2 問では時間変化をとまなう問題に対する提案手法の追従性を検証する。

$$G_2(S) = \begin{cases} \sum_{j=1}^l v_j \cdot w_j; & \sum_{j=1}^l w_j \geq W_{\text{limit}} \\ 0; & \text{otherwise,} \end{cases} \quad (22)$$

ここで、 v_j と w_j の与え方、および W_{limit} の変化のさせ方は文献 3) と 9) に従う。前者の場合は 2 値、後者の場合は 3 値の W_{limit} が 15 世代ごとに周期的に変化し、文字列長はそれぞれ 17 と 7 である。

第 3 問は超多峰性関数である。

$$G_3(S) = \sum_{k=1}^5 G_{\text{MDF}}(S^k), \quad (23)$$

表2 標準パラメータ設定
Table 2 Standard setting of all parameters.

	Canonical GA	fGA		DFGA	
		String-GA	Schema-GA	String-GA	Schema-GA
population size	1000	30	30	5 × 6	5 × 6
string length	30	30	30	30	30
maximum generation	1000	1000	1000	1000	1000
elitist strategy	on	off	off	on	on
crossover type	uniform	uniform	uniform	uniform	uniform
crossover rate	0.8	0.8	0.6	1.0	1.0
mutation rate	0.03	0.05	0.03	0.05	0.03
P_h	—		10		—
P_{ext}	—		0.8		—

$$G_{MDF}(S_i^k) = \begin{cases} \frac{6}{l'} \left| D_C(S_i^k) - \frac{l'}{2} \right| - 2; & \text{if } \left| D_C(S_i^k) - \frac{l'}{2} \right| \geq \frac{l'}{3}, \\ 0.8 \times \left| \frac{3}{l'} D_C(S_i^k) - \frac{l'}{2} \right| - 1; & \text{otherwise,} \end{cases} \quad (24)$$

ここで S^k は文字列 S を単純に 5 等分した k 番目の部分文字列である。この問題は文献 11) で定義される関数で、32 の最適解を含む 500 万以上の局所解を持つ。無数にある準最適解からどれだけの最適解を発見できるかが、探索手法の性能を示す指標となる。

実験に用いたすべての標準パラメータ設定を表 2 に示す。この設定では、fGA と DFGA における String-GA が未探索領域を求めべく広範な探索を行い、Schema-GA は有望な領域において素早く局所解を発見する必要があるために、交叉率や突然変異率に関して単純 GA と異なる設定をしている。また、DFGA は Decentralizing Selection⁶⁾ の枠組みを導入しているため、小さな集団の過剰な収束を考慮せずにエリート戦略¹⁰⁾ を利用することができ、同様の理由によって、過剰な Mixing²²⁾ も抑制されると期待できるために、高い交叉率を採用している。なお、いずれの手法のパラメータも厳密に調整を施したのではなく、以上の直感的な予測に基づいて設定したものであり、後述の実験結果はこれらの手法の限界性能を示すものではないことをあらかじめ言及しておく。

6. 結果および検証, 考察

第 1 の問題である騙し関数に対し、各手法を 10 回適用し、その結果を平均したものを図 5 に示す。グラフ中、横軸は評価回数、縦軸は各世代までに発見された最良値である。3 手法のうち、DFGA が圧倒的に速

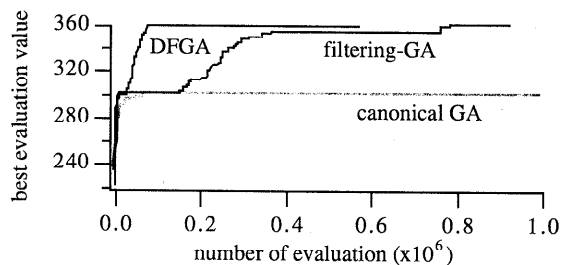


図5 騙し関数における探索結果
Fig. 5 Results simulations on deceptive problem.

く最適解を発見できた。探索が局所注視主導で行われるならば、単純 GA の結果と同様に局所解への収束は避けられない。逆に大域注視を重視しすぎると、局所解さえも発見できない。しかしグラフより、DFGA は局所解をいち早く発見した後に、最適解へと探索の方向を変化させており、本手法が局所注視と大域注視をバランスよく実行していることが分かる。

第 2 の問題である動的ナップサック問題に対し、DFGA を 10 回適用した結果の平均を図 6 と図 7 に示す。グラフの横軸は図 5 と同様に評価回数であるが、縦軸は各世代での最良値を示す。またグラフ中の水平線は、各 W_{limit} 値における最適値である。残念ながら DFGA は、どちらのタイプにおいても、完全に追従できたとはいえない。これらの結果は、長期的な探索履歴の保存機構を持たず、小さな集団で適応度の山を効率的に登ることを目指した本アプローチ固有の問題点を露呈している。これを解消するために、今後は集団間相互作用における作用対象の変更手を拡張し、Schema-GA とフィルタ間、フィルタと String-GA 間の関係も適応的に変化しうる機構を用意する必要があるものと思われる。

本論文中で図表としては示さないが、第 3 の問題である超多峰性関数において、DFGA は 10000 世代で

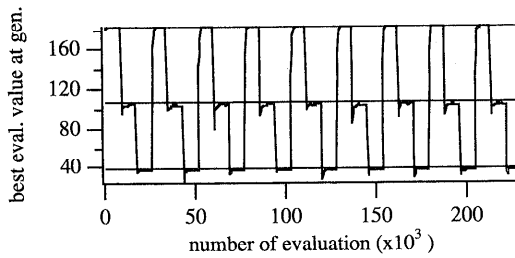


図6 動的ナップサック問題(2状態)におけるDFGAの探索結果
Fig. 6 Result of DFGA on periodically unstational knapsack problem (2 state).

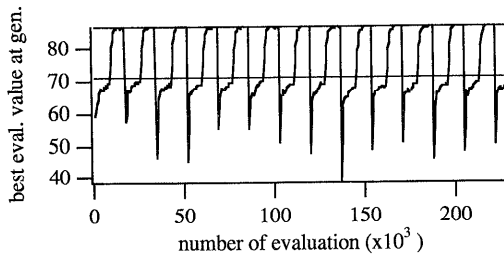


図7 動的ナップサック問題(3状態)におけるDFGAの探索結果
Fig. 7 Result of DFGA on periodically unstational knapsack problem (3 state).

30個の最適解を発見することができた。全最適解の発見には至らなかったが、この結果は探索空間全体の約0.531%の探索点をサンプルするだけで、90%以上の最適解を発見することができた計算になる。DFGAは探索履歴を積極的に保持する機構を持たないため、適切な局所注視と大域注視の制御が実現されているとしても、少数の最適解の間で振動してしまう可能性を否定できない。しかしながら、これだけ多数の最適解を発見できたという事実は、集団の多様性と、その変化に遅れて追従する集団間相互作用によって、探索の方向性を保持し、逆行を防ぐ何らかの機能が発現されていたことを示している。

本章で採用した3つの問題は、いずれもGA困難と呼ばれる問題クラスに属し、このクラスを克服することは、GA研究における1つの主要なテーマとなっている。表1に示した拡張GAの大半も、このような目的で開発されたものであり、DFGAがすべての問題において、これらを上回る探索性能を示すことはできなかった。しかしながら本手法のように、まったく同じパラメータ設定で様々な問題タイプに適用した報告はなされていない。すなわち、問題タイプやパラメータ設定に関するロバスト性という点に着目すると、本手法は非常に有効であり、強力なアプローチであると

いえる。

最後に、各手法の計算速度を比較したところ、単純GAとfGA、DFGAの計算時間コスト比は、およそ100:50:40であった。fGAと単純GAのコスト比は式(19)によく合致しているが、DFGAに関しては理論値の倍に相当する。fGAに比べて、DFGAでは処理の分散化が著しく進んでいるため、冗長な計算による浪費時間の増幅度も大きくなったことが原因と思われる。並列GAへの実装による改善が必要であろう。

7. おわりに

本研究では、探索を効率的に行うための単純かつ強力な戦略を実現するため、多点探索における社会的な組織化ダイナミクスを導入し、大雑把な大域注視と、詳細な局所注視のバランスを、組織全体として自律的かつ適応的に調整するための方法論を構築した。これにより大域注視による局所注視の制御、また逆に局所注視の影響による大域注視の変化が実現され、2つの注視機能が有機的に融合された。

論文中では、このような問題背景と本方法論の基本概念に関する詳しい議論を行った後、実際にGAに実装した手法である分散型フィルタリングGA(Distributed Filtering-GA; DFGA)を提案した。計算機実験においては、GA困難と呼ばれるクラスに属する3つの典型的な問題に適用した。その結果、

- 探索手法のDFGAの有効性を確認し、
- 社会性を模した組織化ダイナミクスによって、注視領域を適応的に制御できていることを明らかにし、また
- 探索履歴の保存機構をまったく持たないにもかかわらず、短/中期の記憶機能が発現していると思われる現象を観測した。

また、時間変化をとまなう問題に対する追従性を高めるために、より柔軟かつ多様な組織的ダイナミクスの導入が今後の課題である。

謝辞 本研究の一部は文部省科学研究費補助金(特別研究員奨励費)の補助を受けて実施されたものである。

参考文献

- 1) Beasley, D., Bull, D.R. and Martin, R.R.: A Sequential Niche Technique for Multimodal Function Optimization, *Evolutionary Computation*, Vol.1, No.2, pp.101-125 (1993).
- 2) Brooks, R.A. and Maes, P. (Eds.): *Artificial Life IV: Proc. 4th Intl. Workshop on the Synthesis and Simulation of Living Systems*, p.444,

- MIT Press (1995).
- 3) Dasgupta, D. and McGregor, D.R.: Non-stationary Function Optimization Using the Structured Genetic Algorithm, *Parallel Problem Solving from Nature*, 2, pp.145-154, North-Holland (1992).
 - 4) Davidor, Y., Yamada, T. and Nakano, R.: The Ecological Framework II: Improving GA Performance at Virtually Zero Cost, *Proc. ICGA 5*, pp.171-176, Morgan Kaufmann (1993).
 - 5) Davis, L. (Ed.): *Handbook of Genetic Algorithms*, p.385, Van Nostrand Reinhold (1991).
 - 6) DeJong, K. and Sarma, J.: On Decentralizing Selection Algorithms, *Proc. ICGA 6*, pp.17-23, Morgan Kaufmann (1995).
 - 7) Eshelman, L.J.: The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination, *Foundations of Genetic Algorithms*, pp.265-283, Morgan Kaufmann (1991).
 - 8) Goldberg, D.E. and Richardson, J.: Genetic Algorithms with Sharing for Multimodal Function Optimization, *Proc. ICGA 2*, pp.41-49, Lawrence Erlbaum Associates (1987).
 - 9) Goldberg, D.E. and Smith, R.E.: Nonstationary Function Optimization Using Genetic Algorithms with Dominance and Diploidy, *Proc. ICGA 2*, pp.59-68, Lawrence Erlbaum Associates (1987).
 - 10) Goldberg, D.E.: Genetic Algorithms in Search, *Optimization and Machine Learning*, p.412, Addison-Wesley (1989).
 - 11) Goldberg, D.E., Deb, K. and Horn, J.: Massive Multimodality, Deception, and Genetic Algorithms, *Parallel Problem Solving from Nature*, 2, pp.37-45, North-Holland (1992).
 - 12) Goldberg, D.E., Deb, K., Kargupta, H. and Harik, G.: Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms, *Proc. ICGA 5*, pp.56-64, Morgan Kaufmann (1993).
 - 13) Grefenstette, J.J.: Deception Considered Harmful, *Foundations of Genetic Algorithms 2*, pp.75-91, Morgan Kaufmann (1993).
 - 14) Holland, J.H.: *Adaptation in Natural and Artificial Systems*, p.183, University of Michigan Press (1975).
 - 15) Maresky, J., Davidor, Y, Gitler, D., Aharoni, G. and Barak, A.: Selectively Destructive Re-start, *Proc. ICGA 5*, pp.144-150, Morgan Kaufmann (1993).
 - 16) Ohkura, K. and Ueda, K.: A Genetic Algorithm with Neutral Mutations for Massively Multimodal Function Optimization, *Proc. 1995 Intl. Conf. on Evolutionary Computation*, pp.361-366, IEEE (1995).
 - 17) Paredis, J.: The Symbiotic Evolution of Solutions and their Representation, *Proc. ICGA 6*, pp.359-365, Morgan Kaufmann (1995).
 - 18) Sakanashi, H., Suzuki, K. and Kakazu, Y.: Controlling Dynamics of GA through Filtered Evaluation Function, *Parallel Problem Solving from Nature, III*, Springer (1994).
 - 19) Suzuki, K., Sakanashi, H. and Kakazu, Y.: Iterative Schema Extracting Operation for Genetic Algorithms, *Proc. Australian and New Zealand Conference on Intelligent Information Systems*, pp.517-520 (1993).
 - 20) Whitley, D. and Hanson, T.: Optimizing Neural Networks Using Faster, More Accurate Genetic Search, *Proc. ICGA 3*, pp.391-396, Morgan Kaufmann (1989).
 - 21) 坂無, 鈴木, 嘉数: フィルタリング GA による他方性関数の最適化, *情報処理学会論文誌*, Vol.36, No.8, pp.1800-1808 (1995).
 - 22) Goldberg, D.E., Deb, K. and Thierens, D.: Toward a Better Understanding of Mixing in Genetic Algorithms, *計測と制御*, Vol.32, No.1, pp.10-16 (1993).

(平成 8 年 7 月 1 日受付)

(平成 9 年 5 月 8 日採録)



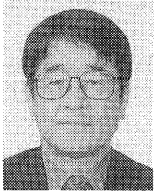
坂無 英徳 (正会員)

昭和 44 年生。平成 4 年北海道大学工学部精密工学科卒業。平成 8 年同大学大学院博士後期課程修了。工学博士。現在、日本学術振興会平成 8 年度特別研究員。進化的計算理論、進化型ハードウェア、自律エージェント研究などに従事。



鈴木 恵二 (正会員)

昭和 40 年生。昭和 63 年北海道大学工学部精密工学科卒業。平成 5 年同大学大学院博士課程修了。工学博士。同年同大学工学部助手、平成 8 年同大学大学院複雑系工学講座助教授。現在に至る。遺伝的アルゴリズム、マルチエージェントシステムなどに興味を持つ。日本機械学会、人工知能学会各会員。



嘉数 侑昇（正会員）

昭和16年生。昭和48年北海道大学大学院工学研究科精密工学専攻博士課程修了。工学博士。北海道大学大学院工学研究科，システム情報工学専攻，複雑系工学講座教授。機械学習，自然知能工学，ロボティクス，知識工学などの研究に従事。日本機械学会，精密工学会，計測自動制御学会，日本ロボット学会，人工知能学会各会員。
