離散構造制約条件付き最適化問題に対する PSO を用いた進化計算

小 熊 祐 司*・古 澤 敦 郎**・相 吉 英太郎**

Evolutional Computing Using PSO for Optimization Problems with Disconnected Constraints

Yuji KOGUMA*, Atsuro FURUSAWA** and Eitaro AIYOSHI**

It is known that the evolutionary computing methods hybridized with local search methods, which are called Memetic Algorithm, are efficient as global optimization methods. The memetic algorithms are divided into two classes of hybridization strategies, which are called Baldwinian and Lamarckian types. This paper is concerned with computational considerations for these two types of hybridization strategies in a case when Particle Swarm Optimization is used as evolutionary algorithm from the standpoint of global optimization methods. Especially, applications to optimization problems with constraints described by disconnected plural sets, which are difficult to solve by usual methods, are considered.

Key Words: evolutionary computing, Memetic algorithm, Baldwinian strategy, Lamarckian strategy, Particle Swarm Optimization

1. はじめに

組合せ最適化問題における大域的最適解探索の計算特性の向上を実現するために、遺伝アルゴリズム^{1),2)}を中心とする進化型アルゴリズムに局所的探索手法を融合させたハイブリッド手法が提案され、Memetic アルゴリズム^{3),4)}と称されるに至り、またそのアルゴリズムの設計指針の検討もなされている⁵⁾。この用語の語源である“meme”とは、遺伝子をもつ生命体が出生後に、経験や学習によって後天的に獲得する情報ないしはその担い手のことを意味する造語であり、生命体の中でその先天的な状態が本質的には変化することがないとする gene (遺伝子) と対立する用語である。Memetic アルゴリズムにおいては、遺伝アルゴリズムが世代交代での gene の状態遷移則を規定するアルゴリズムに対応し、局所的探索アルゴリズムが経験や学習による meme の状態遷移則に対応する。

ところで、これらの融合のさせ方、すなわち Memetic アルゴリズムの構成の仕方は、ラマルク型進化方略 (Lamarckian evolutionary strategy) とボールドウィン型進化方略 (Baldwinian evolutionary strategy) に大別されている⁶⁾。前者は、「meme により後天的に獲得された状態の影響により、先天

的に与えられた遺伝子状態に変化が生じ、世代交代によってそれが次世代への遺伝情報として引き継がれる」とする仮説で、進化論的には受け入れがたい説であるのに対して、後者は、「先天的な遺伝子状態の環境適応度は、meme によって後天的に獲得される状態によって評価されるものの、その後天的獲得状態が遺伝子状態に影響を与えたり、次世代に引き継がれたりすることはなく、次世代に進化的に引き継がれるのは、あくまでも元の遺伝子状態である」とする仮説で、進化論的には常識的な説である。

本論文では、連続変数最適化問題を解法の対象としつつも、その制約領域が互いに分離した多くの部分集合からなる、いわゆる離散構造を有する制約条件のある最適化問題を想定し、この種の特異的な制約条件の扱いに、前述した Memetic アルゴリズムの進化方略の概念を拡張して適用することを考える。具体的には、連続変数の更新に用いる Particle Swarm Optimization^{7),8)} (以下 PSO と記す) を、従来の Memetic アルゴリズムにおいて遺伝アルゴリズムが担っている進化型アルゴリズムとして位置づけ、これに融合させる局所的探索手法として、離散構造制約条件の種類に応じた各種の許容化アルゴリズムを対応づけるものとする。本論文では、連続変数の更新を進化型アルゴリズムにおける世代交代とみなし、制約条件に対する許容化操作を局所的探索とみなした場合の、ラマルク型およびボールドウィン型進化方略の構造的な違いを、まず 2 章で明確にする。これを踏まえたうえで、3 章で示す各種の離散構造制約条件付き連続変数最適化問題に対し、それら制約条件の種類に応じた各種の PSO ベースの Memetic アルゴリズムを 4 章で示す。そして 5 章において、両進化方略の計算特性の違いや有効性を確認する数値実験結果を示す。

* 慶應義塾大学大学院理工学研究科 横浜市港北区日吉 3-14-1

** 慶應義塾大学理工学部 横浜市港北区日吉 3-14-1

* Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kouhoku-ku, Yokohama

** Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kouhoku-ku, Yokohama

(Received April 2, 2009)

(Revised August 18, 2009)

2. 制約条件付き最適化手法としての進化方略

Memetic アルゴリズムにおけるラマルク型およびボールドウィン型進化方略を制約条件付き最適化手法の解法方略としてみなし、そのアルゴリズムの構造的差異を明確にするために、まず、 $E(\mathbf{x})$ を目的関数とする制約条件付き最適化問題

$$\min_{\mathbf{x}} E(\mathbf{x}) \quad (1a)$$

$$\text{subj. to } \mathbf{x} \in X \quad (1b)$$

$$p_n \leq x_n \leq q_n, \quad n = 1, \dots, N \quad (1c)$$

を考える。ここで、集合 X は N 次元実空間 R^N の部分集合とする。また制約条件 (1c) 式は、探索領域・解候補の有限性を仮定するために課した条件であり、制約条件 (1c) 式の内部に大域的最適解が存在する仮定の下で、最適化問題 (1) 式は、制約条件 (1c) 式のない無制約問題と本質的に等価であることに注意されたい。「はじめに」で記したように、ラマルク型およびボールドウィン型進化方略の構造的差異は、前者では meme による局所的探索の結果獲得された状態が次世代へ遺伝情報として引き継がれるのに対して、後者では meme による局所的探索の結果獲得された状態は、原則として次世代に遺伝情報として引き継がれることはない。そこで、meme による局所的探索を、 N 次元実空間 R^N からその部分集合 X への限定操作、すなわち N 次元実空間 R^N 内にとられた内部状態 $\mathbf{u}$ を部分集合である制約集合 X に属する出力状態 $\mathbf{x}$ へ対応づける「許容化操作」とみなすすると、ラマルク型進化方略は、この許容化操作を施した結果得られる出力状態 $\mathbf{x} \in X$ を次世代に遺伝情報として進化させる手法であるのに対して、ボールドウィン型進化方略は、許容化操作を施す前の内部状態 $\mathbf{u} \in R^N$ を次世代に遺伝情報として進化させる手法といえる。後者では、遺伝情報である内部状態 $\mathbf{u} \in R^N$ に対するある種の評価値である適応度は、meme によって許容化された出力状態 $\mathbf{x} \in X$ に対する適応度が用いられることに注意されたい。すなわち、先天的な内部状態 $\mathbf{u} \in R^N$ の評価は、出生後の経験や学習によって meme が獲得する後天的な出力状態 $\mathbf{x} \in X$ で決まるという性質である。

なお、以下では許容化操作を、写像または作用素として記号 $\mathcal{A}$ で表わし、内部状態 $\mathbf{u}$ に許容化操作 $\mathcal{A}$ を施して出力状態 $\mathbf{x}$ を得ることを $\mathbf{x} = \mathcal{A}(\mathbf{u})$ と記す。

2.1 ラマルク型進化方略による制約条件付き最適化の手順

ラマルク型進化方略での内部状態や出力状態を、ボールドウィン型進化方略でのこれらと区別して、おのおの $\mathbf{u}^L, \mathbf{x}^L$ と記す。問題 (1) に対するアルゴリズムとして、これをラマルク型進化方略により解く手順をまとめると **List 1** のようになる。

Step 1 :

初期の出力状態として $\mathbf{x}^L(0) \in X$ を設定する。ただし、この実行が困難ならば、内部状態 $\mathbf{u}^L(0) \in R^N$ をとり、こ

れから meme による「許容化操作 $\mathcal{A}$ 」によって、出力状態 $\mathbf{x}^L(0) = \mathcal{A}(\mathbf{u}^L(0)) \in X$ を生成する。また、 $k = 0$ とおく。

Step 2 :

出力状態 $\mathbf{x}^L(k) \in X$ に対する適応度を、 $E^L(k) = E(\mathbf{x}^L(k))$ とする。

Step 3 :

進化の状態遷移則により、出力状態 $\mathbf{x}^L(k) \in X$ を次世代に進化させ、内部状態として $\mathbf{u}^L(k+1) \notin X$ とおく。

Step 4 :

meme による許容化操作 $\mathcal{A}$ を実行し、 $\mathbf{u}^L(k+1) \notin X$ から許容な状態 $\mathbf{x}^L(k+1) \in X$ を生成する。

Step 5 :

終了判定条件を満たせば、進化計算を停止し、そうでなければ、 $k = k + 1$ として **Step 2** へ戻る。

List 1 The algorithm of optimization by Lamarchian strategy

2.2 ボールドウィン型進化方略による制約条件付き最適化の手順

ボールドウィン型進化方略においては、許容化操作 $\mathcal{A}$ を施す前の内部状態を $\mathbf{u}^B$ とすると、許容化操作 $\mathcal{A}$ によって獲得した出力状態 $\mathbf{x}^B = \mathcal{A}(\mathbf{u}^B)$ は、内部状態 $\mathbf{u}^B \in R^N$ に対する適応度の評価のみに使用し、直接には進化のための遺伝情報として用いることはない。このことは、内部状態 $\mathbf{u}^B \in R^N$ に関する適応度 $E(\mathbf{u}^B)$ の代わりに、 $\mathbf{u}^B$ に写像または作用素 $\mathcal{A}$ を介在させた適応度 $E(\mathcal{A}(\mathbf{u}^B))$ の最適化を考えていることになり、 $\mathcal{A}$ を考慮した内部状態 $\mathbf{u}^B$ に関する最適化問題

$$\min_{\mathbf{u}^B} E(\mathcal{A}(\mathbf{u}^B)) \quad (2a)$$

$$\text{subj. to } \mathbf{u}^B \in R^N \quad (2b)$$

を内部状態に関して解いていることになる。したがって、ボールドウィン型進化方略とはすなわち、目的関数 $E(\cdot)$ を、新しい目的関数 $E(\mathcal{A}(\cdot)) = \Phi(\cdot)$ に変換する方略であると考えられることができる。問題 (2) をボールドウィン型進化方略により解く手順をまとめると、**List 2** のようになる。

Step 1 :

初期の内部状態として $\mathbf{u}^B(0) \in R^N$ を設定し、 $k = 0$ とおく。

Step 2 :

meme による許容化操作 $\mathcal{A}$ を実行し、内部状態 $\mathbf{u}^B(k) \notin X$ から許容な出力状態 $\mathbf{x}^B(k) = \mathcal{A}(\mathbf{u}^B(k)) \in X$ を生成する。

Step 3 :

出力状態 $\mathbf{x}^B(k) = \mathcal{A}(\mathbf{u}^B(k)) \in X$ に対する適応度を $E^B(k) = E(\mathbf{x}^B(k)) = E(\mathcal{A}(\mathbf{u}^B(k)))$ とする。

Step 4 :

進化の状態遷移則により、内部状態 $\mathbf{u}^B(k) \notin X$ を次世代に進化させ、 $\mathbf{u}^B(k+1) \notin X$ とおく。

Step 5 :

終了判定条件を満たせば、進化計算を停止し、そうでなければ、 $k = k + 1$ として **Step 2** へ戻る。

List 2 The algorithm of optimization by Baldwinian strategy

以上において、ラマルク型およびボールドウィン型進化方略の最適化手法としての手順を概念的にまとめたが、両者のアルゴリズムにおける、meme による後天的局所的探索に相当する、内部状態 $\mathbf{u}$ から出力状態 $\mathbf{x}$ への写像（許容化操作） $\mathcal{A}$ の位置の違いに注意してほしい。次章では、制約集合が互いに分離した多くの部分集合からなる、いわゆる離散構造制約条件付き最適化問題をいくつか例示する。

3. 離散構造制約条件付き最適化問題の定式化

本論文で考察する離散構造制約条件付き最適化問題を、

$$\min_{\mathbf{x}} E(\mathbf{x}) \quad (3a)$$

$$\text{subj. to } \mathbf{x} \in X = \bigcup_{m=1}^M X^m \quad (3b)$$

$$p_n \leq x_n \leq q_n, \quad n = 1, \dots, N \quad (3c)$$

と一般的に定式化する。ただし、 $\mathbf{x} \in R^N$ であり、部分制約集合 $X^m (m = 1, \dots, M)$ は、

$$X^{m_1} \cap X^{m_2} = \emptyset \quad \forall m_1, m_2 \in \{1, \dots, M\}, m_1 \neq m_2 \quad (4)$$

すなわち、制約集合 X は互いに分離した M 個の制約集合 $X^m, m = 1, \dots, M$ からなっている、いわば離散構造を有する制約集合とする。制約集合 X の例としては以下のようなものがある。

[1] 部分制約集合 X^m が唯一点からなる集合 $X^m = \{\mathbf{x}^m\}$ であって、 $\mathbf{x}^m$ は関数 $E(\mathbf{x})$ の孤立局所的最適解であり、関数 $E(\mathbf{x})$ が連続微分可能な場合には、 $\nabla E(\mathbf{x}^m) = \mathbf{0}$ が成り立つ場合。すなわち、問題 (3) が、勾配がゼロとなる M 個の孤立局所的最適解の中で、関数 $E(\mathbf{x})$ を最小化する局所的最適解を求める、いわゆる大域的最適化問題となる場合

[2] 部分制約集合 X^m が唯一点からなる集合 $X^m = \{\mathbf{x}^m\}$ であって、その成分 x_n^m がある範囲の整数値をとる場合。すなわち、問題 (3) が、 R^N 空間内のある範囲の格子点の中で、関数 $E(\mathbf{x})$ を最小化する格子点を求める整数計画問題となる場合。この場合には、記述を簡単化すると、集合 X は、

$$X = \{\mathbf{x} | x_n \in \mathbb{Z}, \quad n = 1, \dots, N\} \quad (5)$$

と書ける。

[3] [2] において、整数値がとくに 0 または 1 に限られる、すなわち制約集合 X が、

$$X = \{\mathbf{x} | x_n \in \{0, 1\}, \quad n = 1, \dots, N\} \quad (6)$$

Fig. 1 Disconnected feasible domain (case [4])

である場合 (0-1 計画問題)

[4] 部分制約集合 X^m が唯一点ではなく、一定の広さをもつ連結的な集合で、たとえば [2] の格子点 $\mathbf{x}^m$ を中心として、

$$X^m = \{\mathbf{x} | x_n^m - r \leq x_n \leq x_n^m + r, n = 1, \dots, N\} \quad (7)$$

$$m = 1, \dots, M$$

の場合 (**Fig. 1**). ただし、 $r < 1/2$ である。したがって、制約集合 X は、上下限制約集合が等間隔で規則正しく並んでいる集合である。

[5] [4] において $M = 1$ とした特殊な場合で、制約集合 X が上下限制約集合

$$X = \{\mathbf{x} | p_n \leq x_n \leq q_n, n = 1, \dots, N\} \quad (8)$$

の場合。この場合は離散構造はなくなり、問題 (3) は、いわゆる上下限制約条件付き最適化問題となる。

4. 離散構造制約条件付き最適化問題の進化型解法

一般的にいつて、問題 (3) を直接解くことは、その制約条件の特殊性からいつて容易ではない。そこで、許容化操作を表わすある写像 $\mathcal{A}: R^N \mapsto X$ が存在して、任意の出力状態 $\mathbf{x} \in X$ に対し、

$$\mathbf{x} = \mathcal{A}(\mathbf{u}) \quad (9)$$

となる内部状態 $\mathbf{u} \in R^N$ が存在すると同時に、逆にすべての $\mathbf{u} \in R^N$ に対して (9) 式のような $\mathbf{x} \in X$ が存在すると仮定し、このような写像 $\mathcal{A}$ を [1]~[5] ごとに定義して用いることにする。なお、 $\mathcal{A}$ は必ずしも写像でなくてもよく、作用素やアルゴリズムの類であつてよい。

たとえば、[1] の例の場合は、内部状態 $\mathbf{u}$ を初期点として実行する最急降下法などの局所的最適化アルゴリズムが許容化操作 $\mathcal{A}$ に相当し、その収束点が出力状態 $\mathbf{x}$ である。また [2] の場合は、許容化操作 $\mathcal{A}$ として、内部状態の各成分 u_n 以下の最大の整数をとる床関数

$$x_n = \mathcal{A}_n(u_n) = \lfloor u_n \rfloor, \quad n = 1, \dots, N \quad (10)$$

Fig. 2 Function curve of eq.(11) ($r = 0.3$)

を考えることができる。さらに、[3]の場合には、許容化操作 $\mathcal{A}$ として、内部状態の各成分 u_n を一度閉区間 $[0, 1]$ に閉じ込めた後に四捨五入するなどのアルゴリズムを考えることができる。閉区間への閉じ込めに関しては後述する。[4]の場合には、許容化操作 $\mathcal{A}$ として、床関数を用いた合成関数

$$x_n = \mathcal{A}_n(u_n) = \begin{cases} \lfloor u_n + 0.5 \rfloor + \frac{r(u_n - \lfloor u_n + 0.5 \rfloor)}{0.5 - r}, & \text{C1} \\ \lfloor u_n + 0.5 \rfloor + \frac{r(u_n + r - \lfloor u_n + 0.5 \rfloor)}{0.5 - r}, & \text{C2} \\ u_n, & \text{else} \end{cases} \quad (11)$$

$$\text{C1: } u_n > \lfloor u_n + 0.5 \rfloor + r$$

$$\text{C2: } u_n < \lfloor u_n + 0.5 \rfloor - r$$

を考えることができる。たとえば、 $r = 0.3$ の場合の写像を図示すると Fig. 2 のようになる。[5] の場合は、許容化操作 $\mathcal{A}$ として、上限を超えた分だけ下限から戻す、いわゆるトラス型関数⁹⁾

$$x_n = \mathcal{A}_n(u_n) = \begin{cases} p_n + [(u_n - q_n) \bmod (q_n - p_n)], & u_n > q_n \\ q_n - [(p_n - u_n) \bmod (q_n - p_n)], & u_n < p_n \\ u_n, & \text{else} \end{cases} \quad (12)$$

があげられ、また、これとは別に、上下限制約上での反射により上下限制約条件内に閉じ込める方法も考えることができる。この場合は2種類のモデルを考えることができ、1つは単純にトラス型関数において上限を超えた分だけ「下限から戻す」という部分を、「上限から戻す」としたもので、許容化操作 $\mathcal{A}$ の表式は、

$$x_n = \mathcal{A}_n(u_n) = \begin{cases} q_n - [(u_n - q_n) \bmod (q_n - p_n)], & u_n > q_n \\ p_n + [(p_n - u_n) \bmod (q_n - p_n)], & u_n < p_n \\ u_n, & \text{else} \end{cases} \quad (13)$$

となる。この場合、結果的にはどのような内部状態 u に対しても上下限制約すなわち X に閉じ込めることが可能であるが、制約条件を大きく逸脱した場合を想定していない。そこでたとえば、「上限で反射させたことにより下限を超えてしまった場合、そこで再び反射させる」性質をもつもう1つの反射型関数が考えられ、この場合の許容化操作 $\mathcal{A}$ の表式は、

$$x_n = \mathcal{A}_n(u_n) = \begin{cases} q_n - [(u_n - q_n) \bmod (q_n - p_n)], & \text{C1} \\ p_n + [(u_n - q_n) \bmod (q_n - p_n)], & \text{C2} \\ p_n + [(p_n - u_n) \bmod (q_n - p_n)], & \text{C3} \\ q_n - [(p_n - u_n) \bmod (q_n - p_n)], & \text{C4} \\ u_n, & \text{otherwise} \end{cases} \quad (14)$$

$$\text{C1: } u_n > q_n, [(u_n - q_n) \bmod 2(q_n - p_n)] < (q_n - p_n)$$

$$\text{C2: } u_n > q_n, [(u_n - q_n) \bmod 2(q_n - p_n)] \geq (q_n - p_n)$$

$$\text{C3: } u_n < p_n, [(p_n - u_n) \bmod 2(q_n - p_n)] < (q_n - p_n)$$

$$\text{C4: } u_n < p_n, [(p_n - u_n) \bmod 2(q_n - p_n)] \geq (q_n - p_n)$$

のようになる。[3] で必要となる閉区間 $[0, 1]$ への閉じ込めは、これらのうち1つを用いれば可能となる。(12) 式から (14) 式の写像を図示すると、Fig. 3~5 のようになる。

以上のような許容化操作 $\mathcal{A}$ を内包させた進化の状態遷移則として PSO の更新式を用いた場合について以下で述べる。

PSO は複数の探索点の目的関数値を考慮した探索の集中化と、できるだけ多くの解候補を探索する多様化の概念によって特徴付けられた発見的手法であり、各探索点の探索履歴における p -best と称する最良点と、探索点群全体での探索履歴における g -best と称する最良点を用いて、目的関数値のより良い有望領域に探索を集中させつつ、その領域内を多様に探索する動特性を有している。ラマルク型およびボールドウィン型の2種類の進化方略において、許容化操作 $\mathcal{A}$ を PSO に内包させた場合は、それらのアルゴリズムの構造上、 p -best や g -best の評価が出力状態 x の空間でなされるか、または内部状態 u の空間でなされるかという違いおよび、許容化操作の位置の違いが生じる。

4.1 ラマルク型進化方略に基づく PSO

問題 (4) のような特殊な構造の制約条件に対して、本質的には制約条件が無い最適化問題にしか適用できない無制約最適手法を用いる場合、もっとも原始的な方略として、探索点が制約条件を満たすことを前提とし、いったん制約条件を無視した内部状態 u に関する緩和問題

$$\min_u E(u) \quad (15)$$

を想定し、最適化アルゴリズムの更新則を適用した結果、その更新点 u^L が集合 X に属さず制約条件を侵害する場合は、許容化操作 $\mathcal{A}$ によって制約条件を満たすいわゆる許容点 $x^L \in X$ に引き戻す方法が考えられる。ここでの最適化アルゴリズム

Fig. 3 Image of toroidalization model (eq.(12))

Fig. 4 Image of reflection model I (eq.(13))

Fig. 5 Image of reflection model II (eq.(14))

として PSO を用いると、許容化操作 $\mathcal{A}$ を内在させたラマルク型方略の手順は **List 3** のようになる。

Step 0 :

出力状態とその変動量 $\mathbf{x}^p, \mathbf{v}^p$ の初期値 $\mathbf{x}^p(0), \mathbf{v}^p(0)$ を設定する ($p = 1, \dots, P$)。ただし $\mathbf{x}^p(0) \in X$ を満たすとする。これを実行するのが困難ならば内部状態 $\mathbf{u}^p(0)$ ($p = 1, \dots, P$) をとり、これを許容化操作 $\mathcal{A}$ を用いて X の内部に引き戻す。また、 $k = 0$ とする。

Step 1 :

目的関数値 $E(\mathbf{x}^p(k))$ を計算して、出力状態に関する p -best および g -best を、

$$\mathbf{x}^{p\text{-best}}(k) = \arg \min_{\mathbf{x}^p(i)} \{E(\mathbf{x}^p(i)) \mid i = 1, \dots, k\} \quad (16)$$

$$p = 1, \dots, P$$

$$\mathbf{x}^{g\text{-best}}(k) = \arg \min_{\mathbf{x}^{q\text{-best}}(k)} \{E(\mathbf{x}^{q\text{-best}}(k)) \mid q = 1, \dots, P\} \quad (17)$$

でもって設定し、終了条件を満たしていれば終了する。そうでなければ **Step 2** へ。

Step 2 :

PSO の更新式

$$\begin{aligned} \mathbf{v}_n^p(k+1) = & \lambda \mathbf{v}_n^p(k) + c_1 r_{1n}^p(k) (\mathbf{x}_n^{p\text{-best}}(k) - \mathbf{x}_n(k)) \\ & + c_2 r_{2n}^p(k) (\mathbf{x}_n^{g\text{-best}}(k) - \mathbf{x}_n(k)) \end{aligned} \quad (18a)$$

$$\mathbf{u}^p(k+1) = \mathbf{x}^p(k) + \mathbf{v}^p(k+1) \quad (18b)$$

where $\lambda, c_1, c_2 \geq 0, r_{1n}^p(k), r_{2n}^p(k) \leftarrow U[0, 1]$

$n = 1, \dots, N, p = 1, \dots, P, k = 0, 1, 2, \dots$

により内部状態を更新する。

Step 3 :

許容化操作 $\mathcal{A}$ により、

$$\mathbf{x}^p(k+1) = \mathcal{A}(\mathbf{u}^p(k+1)), p = 1, \dots, P \quad (19)$$

として内部状態 $\mathbf{u}^p(k+1)$ から出力状態 $\mathbf{x}^p(k+1)$ を求め、さらに、

$$\mathbf{v}^p(k+1) := \mathbf{x}^p(k+1) - \mathbf{x}^p(k), p = 1, \dots, P \quad (20)$$

として変動量 $\mathbf{v}^p(k+1)$ を実質的な移動分から再定義する。

Step 4 :

$k \leftarrow k+1$ として **Step 1** へ戻り、 p -best および g -best の再設定を行なう。

List 3 The algorithm of Lamarchian PSO

4.2 ボールドウィン型進化方略に基づく PSO

許容化操作 $\mathcal{A}$ を考慮した内部状態に関する最適化問題

$$\min_{\mathbf{u}} E(\mathcal{A}(\mathbf{u})) = \Phi(\mathbf{u}) \quad (21)$$

に PSO を用いると、その計算手順は **List 4** のようになる。

Step 0 :

内部状態量とその変動量 $\mathbf{u}^p, \mathbf{v}^p$ の初期値 $\mathbf{u}^p(0), \mathbf{v}^p(0)$ を設定する ($p = 1, \dots, P$)。また、 $k = 0$ とする。

Step 1 :

内部状態 $\mathbf{u}^p(k)$ に対して許容化操作 $\mathcal{A}$ を用いて出力状態 $\mathbf{x}^p(k) = \mathcal{A}(\mathbf{u}^p(k))$ ($p = 1, \dots, P$) を計算し、内部状態に関する p -best および g -best を、

$$\mathbf{u}^{p\text{-best}}(k) = \arg \min_{\mathbf{u}^p(i)} \{E(\mathcal{A}(\mathbf{u}^p(i))) \mid i = 1, \dots, k\} \quad (22)$$

$$p = 1, \dots, P$$

$$\mathbf{u}^{g\text{-best}}(k) = \arg \min_{\mathbf{u}^{q\text{-best}}(k)} \{E(\mathcal{A}(\mathbf{u}^{q\text{-best}}(k))) \mid q = 1, \dots, P\} \quad (23)$$

でもって設定し、設定した終了条件を満たしていれば終了する。そうでなければつぎの **Step 2** へ。

Step 2 :

PSO の更新式

$$\begin{aligned} \mathbf{v}_n^p(k+1) = & \lambda \mathbf{v}_n^p(k) + c_1 r_{1n}^p(k) (\mathbf{u}_n^{p\text{-best}}(k) - \mathbf{u}_n(k)) \\ & + c_2 r_{2n}^p(k) (\mathbf{u}_n^{g\text{-best}}(k) - \mathbf{u}_n(k)) \end{aligned} \quad (24a)$$

$$\mathbf{u}^p(k+1) = \mathbf{u}^p(k) + \mathbf{v}^p(k+1) \quad (24b)$$

where $\lambda, c_1, c_2 \geq 0, r_{1n}^p(k), r_{2n}^p(k) \leftarrow U[0, 1]$

$n = 1, \dots, N, p = 1, \dots, P, k = 0, 1, 2, \dots$

により内部状態 $\mathbf{u}^p(k)$ を更新する.

Step 3 :

$k \leftarrow k+1$ として Step 1 へ戻る.

List 4 The algorithm of Baldwinian PSO

ところで, PSO は目的関数の勾配情報を必要としない手法であるが, 制約条件が3章における [1] の場合は, 許容化操作を表わす写像 $\mathcal{A}$ として最急降下法を用いるため, 関数 E の連続微分可能性を前提とし, 適用可能な問題のクラスは狭まるという点に注意されたい.

また, PSO の更新式にはさまざまな改良モデルがあるが, ここでは説明簡略化のため, もっとも単純な PSO (Simple-PSO) を進化型アルゴリズムとして用いた場合を記すことにした. 次章の数値実験においても, 問題ごとのモデルの得手不得手やパラメータ依存性といった影響を極力小さくしたうえで両進化方略の性質を比較検討するため, Simple-PSO を採用した.

5. 数値実験

Simple-PSO ($\lambda = 0.729, c_1 = c_2 = 1.494$)¹⁰⁾ を進化計算に用い^(注1), 許容化操作の組み入れに関してラマルク型とボールドウィン型の比較を行なった結果を示す. PSO の探索点 P および最大更新回数 $k_{\max}$ は, 20 次元程度の変数の問題に対して, PSO が十分収束するような経験値を採用し, それぞれ $P = 20, k_{\max} = 5000$ とした. 離散構造の制約条件には, 3章で言及した例のうち, 離散制約が [1] の孤立局所解集合の場合, [2] の整数格子点の場合, [4] の上下限制約集合が等間隔で規則正しく並んでいる場合, [5] の離散制約ではない上下限制約の場合, を取りあげる. なお, [3] は [2] の特殊な場合であり, [3] と比べると容易なため, 数値実験では省略する. 目的関数については, 2^N minima Function (Prob. 1), Rastrigin's Function (Prob. 2), Schwefel's Function (Prob.3) を用い, 探索領域と候補を有限に限定するための上下限制約については, [2] の場合は (A.1b) 式, (A.2c) 式, (A.3b) 式で示した上限値に 1.0 を加算した. この上下限制約に対する許容化操作は, ラマルク型やボールドウィン型進化方略にかかわらず, 各場合において, 対応する許容化操作 $\mathcal{A}$ を適用する直前に, トーラス型変換 ((12) 式: "Toroidalization" と記す), 単純反射変換 ((13) 式: "Reflection I" と記す), 往復反射変換 ((14) 式: "Reflection II" と記す) の 3 通りを用い, それらの違い

Table 1 Paramater settings of steepest descent method for local search

Problem	ΔT	$k_{\max}$	η
Prob. 1	0.02	25	2.63555
Prob. 2	0.002	25	1.16547
Prob. 3	5.0	25	2.12542

Table 2 Simulation results of [1] (Lamarchian)

Problem	N	Lock up	C.R	Average	Worst	Best
Prob. 1	20	Toroidalization	96	-1565.23	-1510.10	-1566.65
		Reflection I	42	-1546.86	-1481.83	-1566.65
		Reflection II	42	-1546.86	-1481.83	-1566.65
Prob. 2	20	Toroidalization	59	0.51	2.98	0.00
		Reflection I	36	0.85	4.97	0.00
		Reflection II	36	0.85	4.97	0.00
Prob. 3	20	Toroidalization	0	-6580.70	-6009.30	-7274.22
		Reflection I	0	-7513.87	-6839.96	-8142.78
		Reflection II	0	-7513.87	-6839.96	-8142.78

も確認した. また, [5] に関しては, この上下限制約への閉じ込めがただちに対応する許容化操作 $\mathcal{A}$ となる.

ところで, 本論文では, [1] の場合の局所的探索手法として最急降下法を用いるが, このとき, アルゴリズムには勾配計算と目的関数値計算が混在するため, その計算量の比較が困難となる. そこで, 擬似乱数により探索領域内の座標を多数生成し, その座標すべてにおける勾配および目的関数値の計算時間を計測し, その比 η を求め, 目的関数値計算回数 C_V , 勾配計算回数 C_G および計算時間比 η より計算される

$$L = C_V + \eta C_G \quad (25)$$

をもって計算量とした. 本章で行なう実験では, L が 100,000 を超えたところでその計算を打ち切った. Table 1 に局所的探索で用いる最急降下法の離散化幅 ΔT および (25) 式における η を示す.

各離散制約条件について, ベンチマーク問題, 制約条件内への閉じ込め手法の各組み合わせに対し, それぞれ 100 回の数値実験を行なった結果を Table 2~8 に示す. ここで, N , C.R., Average, Worst, Best は, それぞれ

- N : ベンチマーク問題の変数次元
- C.R.: 大域的最適解への収束率
- Average: 探索で得られた解の目的関数値の平均値
- Worst: 探索で得られた解の目的関数値の最大値
- Best: 探索で得られた解の目的関数値の最小値

を示している.

6. 検 討

6.1 [1] 離散制約が孤立局所最適解の場合

Table 2 がラマルク型, Table 3 がボールドウィン型の進化方略の結果である. おおむね, ボールドウィン型進化方略のほうがラマルク型進化方略に比べ高い収束率を示している. ラマルク型進化方略の場合, 進化の結果として探索点が特定の局所最適解に集中することで, PSO の大域的探索能力の核である多様性が失われてしまうため, ボールドウィン型進

(注1) パラメータが問題に依存しない, より探索能力の高い改良 PSO モデルとして IWA-PSO¹⁰⁾ があげられるが, IWA-PSO は探索ステップ数 $k_{\max}$ をあらかじめ定める必要がある. 後述するように, 本論文の数値実験では探索状況に応じて計算を終了するステップ数が異なってくるため, 推奨値パラメータによる探索を行なった.

Table 3 Simulation results of [1] (Baldwinian)

Problem	N	Lock up	C.R	Average	Worst	Best
Prob. 1	20	Toroidalization	98	-1566.08	-1538.37	-1566.65
		Reflection I	98	-1566.08	-1538.37	-1566.65
		Reflection II	98	-1566.08	-1538.37	-1566.65
Prob. 2	20	Toroidalization	52	0.64	4.97	0.00
		Reflection I	45	0.99	4.97	0.00
		Reflection II	45	0.99	4.97	0.00
Prob. 3	20	Toroidalization	0	-6524.09	-5989.55	-7057.08
		Reflection I	0	-7421.36	-6838.44	-8142.78
		Reflection II	0	-7423.72	-6838.44	-8142.78

Table 4 Simulation results of [2] (Lamarchian)

Problem	N	Lock up	C.R	Average	Worst	Best
Prob. 1	20	Toroidalization	6	-1481.28	-1362.00	-1560.00
		Reflection I	4	-1437.80	-1268.00	-1560.00
		Reflection II	4	-1438.38	-1268.00	-1560.00
Prob. 2	20	Toroidalization	0	5.01	11.00	1.00
		Reflection I	0	16.38	42.00	2.00
		Reflection II	0	16.38	42.00	2.00
Prob. 3	20	Toroidalization	0	-4999.49	-3817.42	-5787.55
		Reflection I	0	-6517.29	-4921.66	-7676.38
		Reflection II	0	-6513.34	-4921.66	-7676.38

Table 5 Simulation results of [2] (Baldwinian)

Problem	N	Lock up	C.R	Average	Worst	Best
Prob. 1	20	Toroidalization	4	-1485.76	-1292.00	-1560.00
		Reflection I	0	-1363.50	-1230.00	-1470.00
		Reflection II	0	-1363.50	-1230.00	-1470.00
Prob. 2	20	Toroidalization	88	0.17	5.00	0.00
		Reflection I	85	0.62	25.00	0.00
		Reflection II	85	0.62	25.00	0.00
Prob. 3	20	Toroidalization	0	-4969.93	-4226.50	-5879.33
		Reflection I	0	-6474.62	-5196.73	-7530.79
		Reflection II	0	-6476.81	-5196.73	-7530.79

Table 6 Simulation results of [4] (Lamarchian $r = 0.3$)

Problem	N	Lock up	C.R	Average	Worst	Best
Prob. 1	20	Toroidalization	27	-1531.87	-1425.28	-1566.65
		Reflection I	0	-1378.06	-1227.37	-1510.10
		Reflection II	0	-1378.06	-1227.37	-1510.10
Prob. 2	20	Toroidalization	0	14.52	25.87	4.97
		Reflection I	0	21.20	46.76	5.97
		Reflection II	0	21.20	46.76	5.97
Prob. 3	20	Toroidalization	0	-4992.70	-3750.48	-6110.82
		Reflection I	0	-6496.20	-5316.87	-7905.88
		Reflection II	0	-6496.20	-5316.87	-7905.88

Table 7 Simulation results of [4] (Baldwinian $r = 0.3$)

Problem	N	Lock up	C.R	Average	Worst	Best
Prob. 1	20	Toroidalization	0	-1482.59	-1348.75	-1556.47
		Reflection I	0	-1381.92	-1250.40	-1509.77
		Reflection II	0	-1381.92	-1250.40	-1509.77
Prob. 2	20	Toroidalization	0	16.03	33.84	5.97
		Reflection I	0	25.44	55.74	7.96
		Reflection II	0	25.30	55.74	7.96
Prob. 3	20	Toroidalization	0	-4943.60	-3914.55	-5907.92
		Reflection I	0	-6475.09	-5316.86	-7668.99
		Reflection II	0	-6475.09	-5316.86	-7668.99

化方略に比べ探索能力が劣るものと考えられる。

また、ラマルク型進化方略では進化の結果により探索点が局所的最適解の近傍に存在しやすくなること、また多様性に劣ることから、ローカルサーチを完了するのに必要な局所的探索ステップ数は更新回数を追うごとに減少していく傾向が見られる。一方ボールドウィン型進化方略では多様性は失われず、また進化により局所的最適解付近に探索点が移動する

Table 8 Simulation results of [5] (Lamarchian)

Problem	N	Lock up	C.R	Average	Worst	Best
Prob. 1	20	Toroidalization	5	-1493.70	-1397.01	-1566.65
		Reflection I	0	-1370.15	-1170.82	-1538.37
		Reflection II	0	-1370.15	-1170.82	-1538.37
Prob. 2	20	Toroidalization	0	24.19	46.76	9.95
		Reflection I	0	53.50	111.44	22.88
		Reflection II	0	53.50	111.44	22.88
Prob. 3	20	Toroidalization	0	-4909.19	-3837.47	-6004.24
		Reflection I	0	-6595.74	-5594.69	-7905.90
		Reflection II	0	-6595.94	-5594.69	-7905.90

Fig. 6 Local search steps of each evolutionary strategy (Prob.1 Toroidalization)

ことはないで、局所的探索ステップ数は更新回数が増えていってもさほど変わらないといった傾向を示した (Fig. 6)。

ところで、[1] で計算している最適解とは、[5] のそれに一致しており、勾配計算が可能であるという仮定を認めるならば、Table 8 の結果と比較が可能である。Prob. 1~Prob. 3 のいずれにおいても、単純な PSO のみにより探索を行なうよりも、局所的探索を、とくにボールドウィン型進化方略により行なうことで結果が大幅に改善されていることがわかる。Fig. 7 は、適当な多峰性関数 $E(x)$ と、それを解くためにボールドウィン型進化方略を適用したときの実質的な目的関数 $\Phi(u)$ の形状である。Prob. 2 は、いわゆる POP¹¹⁾ の成立する目的関数であるが、各局所的最適解の合間に大きい数値的山を有しているために、この山を越えて目的関数値を改善することの難しい問題である。しかし、Fig. 7 より、各局所的最適解の間にどれだけ大きな数値的山があろうとも、ボールドウィン型進化方略を適用すれば、それらを取り払うことができる。また、ラマルク型進化方略の場合においても、局所的探索が実行されることで探索点は谷底に推移するため、数値的山の急峻さはほとんど無関係となる。このことを示すために、Prob. 2 において、(A. 2b) 式に示す A の値を、2.0, 10.0 (Original), 20.0, 50.0 と変化させて、Simple-PSO と両進化方略に基づく PSO をこれらに適用する数値実験を行なった。その結果を Table 9 に示す。

各 A に対する Prob. 2 の概形は Fig. 8 に示したとおりであり、 A の値が大きくなるにつれて各局所的最適解の間の数値的山が急峻になるが、局所的最適解における目的関数値自体はほとんど変化しない。Table 9 をみると、Simple-PSO

Fig. 7 Original objective function ($E(x)$) and virtual objective function ($\Phi(u)$)

Table 9 Simulation results of [1] (Prob. 2 for several A)

Method	Lock up	A							
		2		10		20		50	
PSO	Toroidalization	0	9.16	0	24.19	0	29.42	0	36.93
	Reflection I	0	12.51	0	53.50	0	94.13	0	138.75
	Reflection II	0	12.51	0	53.50	0	94.08	0	138.21
Lamarchian PSO	Toroidalization	47	0.71	59	0.51	10	1.93	9	2.18
	Reflection I	40	0.71	36	0.85	10	1.88	10	1.92
	Reflection II	40	0.71	36	0.85	10	1.88	10	1.92
Baldwinian PSO	Toroidalization	53	0.73	52	0.64	59	0.63	38	0.85
	Reflection I	46	0.77	45	0.99	41	1.44	41	1.25
	Reflection II	46	0.77	45	0.99	41	1.44	41	1.25

Fig. 8 Shapes of Prob. 2(Rastrigin Function) for several A

を用いた場合では A の値が大きくなるにつれ結果が悪化していくが、両進化方略に基づく PSO を用いて解いた場合、上で考察したように、 A の値による結果の差は Simple-PSO のそれに比べ小さく、また、やはりボールドウィン型進化方略のほうが優れた結果を示している。

6.2 [2] 離散制約が整数格子点の場合

Table 4 がラマルク型、Table 5 がボールドウィン型の進化方略の結果である。[1] の場合と同じく、ラマルク型進化方略の場合、進化の結果として探索点が特定の整数格子点に重なってしまうことで多様性が削がれ、ボールドウィン型進化方略に劣ったものと考えられる。

6.3 [4] 上下限制約集合が等間隔で規則正しく並んでいる場合

Table 5 がラマルク型、Table 6 がボールドウィン型の進化方略の結果である。また、ラマルク型進化方略の場合における探索軌道図を Fig. 9 に示す^(注2)。許容化操作 (11) を施す

(注2) 図の可読性を考慮して、Fig. 9(a) に関しては $k = 500$ までのデータをプロットしている。

ことで、整数格子点を中心として一定の広がりをもつ空間内のみに限定して探索を行なっているようすがわかる。この離散制約条件の場合、許容化操作 $\mathcal{A}$ により局所的最適解や整数格子点のような孤立した点に収束することはないので、[1][2] に比べると、両進化方略の優劣が際立ってはいない。

(a) k v.s. x_1^1

(b) x_1^1 v.s. x_2^1

Fig. 9 Searching trajectory for Prob. 1 (case 2, Lamarchian, Toroidalization)

6.4 [5] 上下限制約の場合

この場合は、“Toroidalization”、“Reflection I”、“Reflection II” のみをそれぞれ許容化操作として用いるが、上記の例とは逆に、ボールドウィン型進化方略ではまったく大域的最適解に収束しないため、ラマルク型進化方略のみの結果を Table 8 に示す。制約集合が離散的でない上下限制約の場合に対して、ボールドウィン型進化方略が不向きな理由は、 $\mathcal{A}$ による許容化操作を施すことで同じ目的関数値を取る局所的最適解を、 R^N 内で無限に増やしていることになり、このことが PSO の収束性を悪化させているためと考えられる。

上下限制約への引き戻し方法についてみると、Prob. 1 および Prob. 3 の関数で差が顕著であり、前者では “Reflection I”、“Reflection II” が弱く “Toroidalization” が強い。一方、後者では “Reflection I”、“Reflection II” が強く “Toroidalization” が弱い。Prob. 1 は局所的最適解が少なく、それぞれの谷がきわめて広い。ゆえに局所的最適解の谷からの脱出如

何に最適化性能がかかっている。“Reflection I”, “Reflection II”を用いた場合、上下限制約を侵害した場合、侵害前に位置していた谷と同じ谷に陥りやすい。しかし“Toroidalization”を用いた場合は、異なる局所的最適解の谷へ移動することができる。ゆえに Prob. 1 では“Toroidalization”のほうが良い結果を残したと考えられる。また Prob. 3 は、上下限制約の両端付近に良い解を保持しており、“Toroidalization”では、そこに探索点が向かった結果、トラスの作用によりその反対側に探索点が移動してしまうことが起こりえてしまい、探索過程で得た良い解付近の領域を探索するという、PSO に意図された挙動を実現するのが難しくなるため、“Reflection I”, “Reflection II”の2つの反射型の閉じ込め関数のほうが良い結果を残したものと推察される。なお、“Reflection I”, “Reflection II”の2つに関しては、有意な差は認められなかった。

7. おわりに

本論文では、組合せ最適化問題の解法である遺伝アルゴリズムの改良手法にあたる、Memetic アルゴリズムの考え方を、変数が連続である場合の離散構造制約条件付き最適化問題に適用することを試みた。その過程で、Memetic アルゴリズムの2つの進化方略であるラマルク型進化方略およびボールドウィン型進化方略の適用則を数学的に明示し、PSO の更新式の中に組み入れた。

また、これらの応用例として、離散制約条件が、[1] 孤立局所的最適解集合の場合、[2] 整数格子点の場合、[4] 上下限制約集合が等間隔で規則正しく並んでいる場合、[5] 上下限制約の場合について最適化計算の数値実験を行ない、両進化方略の特性を検討した。また、とくに [1] の結果から、アルゴリズムの1回の反復ごとに探索点をボールドウィン型進化方略により局所的最適解に落とし込む方策が、各局所最適解の間にある数値的山を実質的に取り払い、大域的最適化計算の性能を向上する効果があることを確認した。さらに、上下限制約領域内への許容化操作として3つのモデルをとりあげ、問題により有効なモデルが異なることを確認した。

参考文献

- 1) 北野宏明編：遺伝アルゴリズム，産業図書（1993）
- 2) 三宮，玉置，喜多，岩本：遺伝アルゴリズムと最適化，朝倉書店（1998）
- 3) P. Moscato: Memetic Algorithms: A Short Introduction. In D. Corne, M. Dorigo & F. Glover (eds), New Ideas in Optimization, McGraw Hill, 219/234 (1999)
- 4) N. Krasnogor: Toward Robust Memetic Algorithms, W.E. Hart, N. Krasnogor, J.E. Smith (eds), Recent Advances in Memetic Algorithms, Springer, 185/207 (2004)
- 5) 片山，石淵：Memetic アルゴリズム，計測と制御，**47-6**, 487/492 (2008)
- 6) T. Sasaki and M. Tokoro: Comparison between Lamarckian and Darwinian Evolution on a Model Using Neural Networks and GA's, Knowledge & Information Systems, **2-2**, 201/222 (2000)
- 7) J. Kennedy and R.C. Eberhart: Particle Swarm Optimization, Proc. of ICNN, **4**, 1942/1948 (1995)
- 8) J. Kennedy and R.C. Eberhart: Swarm Intelligence, Morgan Kaufmann Publ. (2001)
- 9) T. Okamoto and E. Aiyoshi: The Improved Draining Method and Its Application to Proper Benchmark Problems, Proc. of SICE-ICCAS 2006, 2190/2195 (2006)
- 10) R.C. Eberhart and Y. Shi: Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization, Proc. of the Congress on Evolutionary Computation (CEC2000), 84/88 (2000)
- 11) 相吉，安田編：メタヒューリスティクスと応用，電気学会（2007）
- 12) L.A. Rastrigin: Systems of Extremal Control, Nauka, USSR (1974)
- 13) H.P. Schwefel: Numerische Optimierung von Computer-Modellen Mittels der Evolutionsstrategie, Birkhauser Stuttgart (1977)

《付 録》

A. ベンチマーク問題

Prob. 1 2^N minima Function

$$\min_x E(x) = \sum_{n=1}^N (x_n^4 - 16x_n^2 + 5x_n) \quad (\text{A. 1a})$$

$$\text{subj. to } -5.0 \leq x_n \leq 5.0 \quad n = 1, 2, \dots, N \quad (\text{A. 1b})$$

$$x^* = -2.9035, \quad E(x^*) = -78.3323N$$

Prob. 2 Rastrigin's Function ¹²⁾

$$\min_x E(x) = \sum_{n=1}^N (x_n^2 - A \cos(2\pi x_n) + A) \quad (\text{A. 2a})$$

$$\text{where } A = 10.0 \quad (\text{A. 2b})$$

$$\text{subj. to } -5.0 \leq x_n \leq 5.0 \quad n = 1, 2, \dots, N \quad (\text{A. 2c})$$

$$x^* = \mathbf{0.0}, \quad E(x^*) = 0.0$$

Prob. 3 Schwefel's Function ¹³⁾

$$\min_x E(x) = \sum_{n=1}^N \left(-x_n \sin \left(\sqrt{|x_n|} \right) \right) \quad (\text{A. 3a})$$

$$\text{subj. to } -500.0 \leq x_n \leq 500.0 \quad n = 1, 2, \dots, N \quad (\text{A. 3b})$$

$$x^* = 420.969, \quad E(x^*) = -418.983N$$

[著 者 紹 介]

小 熊 祐 司

2007 年慶應義塾大学理工学部物理情報工学科卒業。2009 年同大学院理工学研究科基礎理工学専攻修士課程修了。同年 (株) テプコシステムズ入社。大学院在学中は、PSO の改良など大域的最適化手法に関する研究に従事。電気学会の会員。

古 澤 敦 郎

2009 年慶應義塾大学理工学部物理情報工学科卒業。同年同大学院理工学研究科基礎理工学専攻修士課程進学。在学中は、最適化手法としての進化方略の計算特性に関する研究に従事。

相 吉 英太郎 (正会員)

1975 年慶應義塾大学工学部計測工学科卒業。80 年同大学大学院工学研究科計測工学専攻博士課程修了。助手・専任講師・同理工学部助教授を経て、96 年より同大学理工学部物理情報工学科教授。工学博士。最適化の理論と計算手法に関する基礎研究に従事。電気学会・システム制御情報学会などの会員。2006 年度から 2 年間、計測自動制御学会理事。