ALife による分散最適化

准員 林 大作 (山口大学)

正員 佐藤 泰司 (山口大学)

正員 沖田 豪 (広島市立大学)

Distributed Optimization by Using Artificial Life

Daisaku Hayashi, Taiji Satoh (Yamaguchi University)

Tsuyoshi Okita (Hiroshima City University)

This paper presents a distributed algorithm for minimizing a nonconvex multimodal function. The local search methods based on the gradient information have efficient convergence property, but these methods have a tendency to get stuck at a local minima. Therefore a global optimization method with good convergence property is desired.

Although many global optimization methods such as Simulated Annealing and Tree Annealing have been proposed, these methods require much computational cost. In recent years, new distributed algorithms based on Artificial Life (ALife) is studied and its potential power is reported. In this paper, therefore, the frame work of ALife is employed into a function minimization. Since the proposed method utilizes no gradient information, it can be applied to very wide class of problems.

In this paper, firstly, ALife in the two-dimensional discrete system is realized. Then it is extended to the multi-dimensional continuous system. Next, ALife is applied to an optimization problem of function minimization. The effectiveness of the proposed method is demonstrated through some numerical tests on multimodal test functions. The numerical tests also show that the proposed method is superior to the Simplex method and Tree Annealing method.

キーワード : ALife, 大域的最適化, 分散最適化, *Emergent Colonization*

1. 緒言

非線形最適化⁽¹⁾は制御の様々な分野において重要な役割を果たしており、例えばシステム同定の際のパラメータ決定などに使用されている。このような多変数関数の最小値を求める最適化手法は、実用上の問題を解くために応用され、さらに非線形問題解決のための数値解析の手段を提供する。

微分可能な凸関数の最適解を求める方法としては勾配情報を利用した準ニュートン法⁽²⁾などの方法が提案されている。これらの方法は収束は非常に速いが、非凸関数に対しては大域的最適解ではなく局所解に陥る可能性がある。すなわち、非凸性が強い多峰性関数に対しては勾配を基にした最適化手法では限界がある。

多峰性を持つ非凸関数の大域的最適解を求めるための勾配を利用しない最適化手法も様々なものが提案されている。良く知られているものでは *Simulated Annealing* 法⁽³⁾ (SA 法) がある。SA 法では確率的な動作を内在させることにより、最適解を発見する可能性を高くしている。しかしながら、この方法は膨大な計算時間が必要となる。従って、効率の良い大域的最適化手法が望まれている。

従来の最適化手法は基本的には集中最適化の考えを用い

ている。最近の研究⁽⁴⁾⁽⁵⁾において、最適化手法として分散最適化による手法が提案されている。これを基にして、本研究においては非凸関数の最適化に対して分散最適化の考えを導入する⁽⁶⁾。この分散最適化手法による大域的最適化の実現のために、*Artificial Life* (人工生命 : *ALife*)⁽⁵⁾を計算機上で実現する。文献⁽⁵⁾では 2 次元の離散値系で *ALife* を実現しているが、本研究ではこれを多次元・連続系に拡張を行った。そのための基礎として、まず最も基本的な 2 次元の離散値系において *Emergent Colonization* (自然発生の群棲 : *E.C.*) が起こることを確認する。その後多次元・連続系への拡張を行ない、*E.C.* を用いた大域的最適化手法について述べる。ここで提案する *ALife* による最適解探索手法は、良好な解を高速に発見することが可能であることが数値計算より確かめられた。

2. 離散値系における定式化

2.1 *Artificial World* の定義

E.C. が実際に計算機上で発生することを文献⁽⁵⁾に従って最も基本的な 2 次元の離散値系で確かめる。

2 次元の離散値系における *Artificial World* (人工世界 :

AWorld) は、最適化への適用を考えるために $N \times N$ 個の離散的なセルを持つ正方形のデカルト平面とする。ここで、 N は正方形の辺の長さであり、8 近傍(そのセルに隣接する 8 つのセル)で近傍を定義する。AWorld 中のセルは Resource (“エネルギー資源”あるいは“食物”)を含むことができ、Artificial Organism (人工生物:Artorg)と呼ばれる幾種類かの“種族”の個体が住むこともできる。また一つのセルは高々一単位の Resource と一単位の Artorg を含むことができる。但し、Artorg が運んでいる Resource は一単位の Resource とは考えない。

2.2 Artorg の定義

Artorg は 4 種類 (White, Red, Green, Blue) 存在するとし、それらの間で環状食物連鎖の様なものを持ち、自分の要求する Resource を得たときに代謝を実行する。ここで要求される Resource は、Artorg の種類により決定される。Artorg は代謝を実行したときに内部エネルギーを規定量増加させる。そして代謝を実行した後に自分の近傍のランダムな場所に Waste (廃物) を生成する。近傍が一杯の場合は、自分から最も近いセルの内から、ランダムに生成場所を決定する。この Waste が他の Artorg に対する Resource となる。

この様子を図 1 を用いて説明する。まず、White Artorg が自分の要求する Resource(B) を得ることによって代謝を行い、Waste(W) = Resource(W) を生成する。この Resource(W) が Red Artorg が要求する Resource となる。このような食物連鎖を 4 種類の Artorg 間で行う。

図 1 Artorg 間の食物連鎖
Fig. 1. A Circular food chain of Artorgs

Artorg は知覚システムを持っており、自分から一番近くにある Resource あるいは他の Artorg を発見可能で、発見した Resource に向かって進む。

各々の Artorg は内部エネルギーを持ち、そのエネルギーがある閾値以下になるとその Artorg は死亡する。Artorg

はそれぞれの生存期間に比例したエネルギーを消費するため、年齢の高くなった Artorg は生存し難くなり寿命が暗黙的に決まる。

2.3 資源交換

Artorg は Resource の交換を他の Artorg と行なうことが可能であり、そのための交換戦略を持っている。Artorg の交換戦略は 16 ビットのテーブルで構成されている(表 1)。この戦略に基づき自分の近傍に存在する他の Artorg と Resource の交換を行なう。テーブルの各行は Artorg が現在運んでいる Resource の種類を表し、各列は交換相手が提供する Resource の種類を表す。このテーブルのうちで“1”に設定されたビットは Artorg がその交換を行なうことを示し、“0”のビットはその交換を拒絶することを示す。

例えば、表 1 に与えられた交換戦略では、Red Resource (表 1 の第 2 行に対応) を運んでいる Artorg は運んでいる Resource を White あるいは Blue Resource (第 1,4 列) と交換を行うが、Red あるいは Green Resource (第 2,3 列) を提示した Artorg とは交換を拒否する。また交換が実際に起こるためには交換を行なう両者が同意しなければならない。つまり、交換の際に与える Resource と受け取る Resource に対応する交換戦略のビットが交換を行なう両者において“1”を持っていなければならない。

表 1 交換戦略の一例
Table 1. An Example of Trading Strategy

	White	Red	Green	Blue
White	0	1	1	0
Red	1	0	0	1
Green	0	1	1	1
Blue	1	0	0	1

明らかに、ある一つの種の Artorg に対して、潜在的に良い交換戦略および悪い交換戦略がある。一般には、良い交換戦略を持った Artorg は運んでいるものが何であろうと、それを自分が代謝できる種類の Resource と交換する。良い交換戦略を持つ Artorg は、良い戦略を持たない競争者 (Artorg) に対して進化論的に有利となることが推測できる。

2.4 生殖作用

Artorg は子孫を残すことができる。生殖を行なうためにはある一定の年齢に達する必要がある、かつ、ある閾値以上の内部エネルギーを持つ必要がある。生殖を行なうか否かは確率(パラメータとして与えられる)に基づいて決定され、自分から最も近くにいる同種の Artorg を相手に選び、自分の近傍内のランダムな地点に新しい Artorg を生成する。生殖による新しい Artorg の交換戦略は Genetic Algorithm⁽⁷⁾ (GA) によってその両親の戦略から決定される。ここで使用する GA は一点交叉と突然変異だけを用い

た Simple GA である。

実際にどのように染色体が生成されるのかを説明する。図 2 は Artorg の染色体の一例である。この染色体は表 1 の交換戦略と対応しており、表 1 の交換戦略を行順に並べたものが図 2 となる。図 3 は一点交叉による新しい染色体の生成を示す。ここでは A と B の Artorg から生殖が行われるとする。まず、ランダムな位置でそれぞれの染色体をカットする。その時にそれぞれの染色体は A', A'' と B', B'' に分けられる。この組合せにより 2 種類の新しい染色体が生成される。この 2 種類の新しい染色体から 1 つをそれぞれ 50 パーセントの確率で選択し、選択された染色体に対して各遺伝子(ビット)毎に 0.5 パーセントの確率で突然変異操作を行う。従って、突然変異が起こらない確率は $(1 - 0.005)^{16} \approx 0.995^{16} \approx 0.923$ となる。これらの操作によって得られた染色体を、生殖により生まれてくる新しい Artorg の染色体とする。

図 2 染色体の一例
Fig. 2. An Example of Chromosome

図 3 一点交叉による染色体の生成例
Fig. 3. An Example of Crossover

3. 離散値系でのシミュレーション

3.1 初期設定

AWorld の初期状態は以下のように設定する。

- 各々同数の Artorg をランダムに配置する。
- 各 Artorg の内部エネルギーにデフォルトの値を与える。
- Artorg の総数と同数の Resource を Artorg の存在しない場所にランダムに配置する。

3.2 Algorithm

Artorg のライフサイクルは以下の Algorithm で示される。

- Step 0: 初期設定に基づき初期化を行う。
- Step 1: 自分から最も近くにある Resource を探索する。
- Step 2: 移動を行なう。
- Step 2.1: 発見した Resource に向かって移動する。同距離に複数発見した場合は、ランダムにそれらの内の一つの Resource に向かって移動する。
- Step 2.2: 近傍に Resource を発見できない場合、近傍内をランダムに移動する。
- Step 3: 代謝を実行する。
- Step 4: 交換を実行し、可能ならば交換により得た Resource によって代謝を実行する。
- Step 5: 生殖を実行する。
- Step 6: 内部エネルギーを減少させ、生存不可能ならばその Artorg は消滅する。
- Step 7: Step 1 に戻る。

3.3 Emergent Colonization

$N = 25$, 初期 Artorg 数 20 として、前節で定義した Artorg を計算機上の AWorld に放す事によって E.C. が起きた。つまり、最初はランダムに分散させた Artorg がある程度の時間を経ることによって AWorld 上のある地点に群棲を生成した(図 4, 5 参照)。

ここで、○, △, ◇, □ はそれぞれ White, Red, Blue, Green の Artorg を表す。図 4 は初期状態であり、前節のアルゴリズム 1 回を 1 サイクルとしたときに、図 5 は 1000 サイクル後の様子である。この群棲が起こる地点は全く予測できない位置であり、その地点に集まるように Artorg に指定した訳ではない。

図 4 初期状態 (Artorg 数 = 20)
Fig. 4. Initial State (Number of Artorgs = 20)

図 5 Emergent Colonization (Artorg 数 = 57)
Fig. 5. Emergent Colonization (Number of Artorgs = 57)

同条件で乱数系列を変える (50 回) ことによって、どの程度 E.C. が発生するかを調べた。E.C. の発生の有無については、人間の視覚によって判定を行なった。その結果を表 2 に示す。結果より、E.C. がかなりの確率で発生することが確認された。

表 2 Emergent Colonization の発生
Table 2. Appear of Emergent Colonization

	Prob.
E.C. 発生	0.82
E.C. 未発生	0.12
Artorg 死滅	0.06

このような群棲が起こった理由は、環状食物連鎖を持つ生物がそのうちの一種が欠けただけで全体の均衡が崩れ、結果として全種の滅亡につながることから、ある程度集まっているほうが生態系としてバランスが良くなるためであると考えられる。

4. 連続系・多次元への拡張

4.1 AWorld の再定義

前節までにおける AWorld は $N \times N$ 個の離散的な点を持つ正方形のデカルト平面から構成されていたため、その応用範囲は限られたものとなる。ここでは、今までの離散点で定義された空間の連続空間への拡張と、多次元への拡張を考える。そのため、本節より後は AWorld を $\mathbf{x}_{min}, \mathbf{x}_{max} \in R^n$ (n は変数の次元) を上下限とする空間と定義する。

4.2 Artorg の再定義

連続系に対しても Artorg は 4 種類存在するものとし、初期状態としては Artorg, Resource 共にランダムに分布している。交換戦略等の大部分は基本的には前節までの定義

と同様であるが、近傍についての定義を以下のように変更する。つまり、連続系においては、ある点を中心とするユークリッド距離以内の空間をその点に対する近傍と呼ぶ。従って、ある点 $\mathbf{x}_s$ の近傍 C は次式で定義される。

$$C = \{\mathbf{x} \in R^n \mid \|\mathbf{x} - \mathbf{x}_s\|_2 \leq D\}$$
 (1)

但し、 D は Artorg の 1 サイクルでの移動可能範囲である。このように定義された近傍を、Artorg の移動・資源交換・生殖で用いる。また、Artorg の 1 サイクルでの移動可能な範囲はこの近傍で定義された空間内とし、Resource が近傍内に存在しない場合は Artorg は近傍内をランダムに移動するものとする。

5. 最適化問題への適用

以下では、 $f : R^n \rightarrow R^1$ に対する非線形最適化問題

$$\min_{\mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max}} f(\mathbf{x})$$
 (2)

を考え、前節で連続系に拡張した E.C. を最適解の探索に導入する。そのために、AWorld を解空間と考え、最適解あるいは最適解に近い良好な解の探索の実現を考える。

5.1 知覚システムの改良

ミクロな相互作用の結果としてシステム全体のマクロな E.C. が生成することを考えると、これを関数の最適化に用いるためには関数値の小さな領域で E.C. を発生させる必要がある。そのために、解空間における関数値を Artorg の知覚システムに導入する。つまり、Artorg の近傍の Resource に対して、現在の地点とその地点との関数値の差をそこに移動するためのコストと考える。そのために Artorg が自分の近傍にあるどの Resource に移動するかを決定する基準として移動距離と目的関数値を考慮した次の評価関数 $F(\mathbf{x}_s, \mathbf{x}_r)$ を定義する。

$$F(\mathbf{x}_s, \mathbf{x}_r) = \|\mathbf{x}_r - \mathbf{x}_s\|_2 + W \cdot (f(\mathbf{x}_r) - f(\mathbf{x}_s))$$
 (3)

ここで、 $\mathbf{x}_s$ は Artorg の現在の座標であり、 $\mathbf{x}_r$ は対象とする Resource の座標である。 W は移動距離と目的関数値のバランスをとる正の重み定数であり、 $f(\mathbf{x})$ は $\mathbf{x}$ における関数値である。Artorg は (3) 式で定義された評価関数が最小であるような Resource に移動する。この式の第一項はその Resource が自分からどの程度離れているかを表しており、第二項はその Resource がどの程度自分のいる地点から $f(\mathbf{x})$ が大きくなっているかを表している。従って、この評価関数を小さくする方向に Artorg が移動することにより、 $f(\mathbf{x})$ が小さい解空間に E.C. が発生しやすくなると考えられる。

6. 連続・多次元空間でのシミュレーション

本手法の有効性を確かめるために Banana Function と Camel Function に対して最適解探索を行った。この二つ

の関数は共に2次元の関数でテスト関数として良く用いられるものである。更に、*Camel Function* を4次元に拡張した関数 (*Double Camel Function*) に対しても最適解の探索を行なった。

6.1 テスト関数

6.1.1 Rosenbrock の *Banana Function*

Banana Function ⁽¹⁾ は次式で表される。

$$f_B(x) = 100.0 \times (x_2 - x_1^2)^2 + (1.0 - x_1)^2 \tag{4}$$

この関数の局所解は1つで(すなわち大域的最小解)、
 $f_B(1.0000, 1.0000) = 0.0000$ である。

6.1.2 *Camel Function*

Camel Function ⁽⁸⁾ は次式で表される。

$$f_C(x) = (4.0 - 2.1 \cdot x_1^2 + \frac{x_1^4}{3}) \cdot x_1^2 + x_1 \cdot x_2 + (-4.0 + 4.0 \cdot x_2^2) \cdot x_2^2 \tag{5}$$

この関数の局所解は6個で、そのうちの2つが大域的
最小解であり、
 $f_C(\pm 0.0898, \mp 0.7126) = -1.0316$ である。

6.1.3 *Double Camel Function*

Double Camel Function を次式で定義する。

$$f_D(x, y) = f_C(x) + f_C(y) \tag{6}$$

この関数の局所解は36個で、そのうちの4つが大域的
最小解であり、
 $(x_1, x_2) = (\pm 0.0898, \mp 0.7126)$ and
 $(y_1, y_2) = (\pm 0.0898, \mp 0.7126)$ の時、
 $f_D(x, y) = -2.0632$ である。

6.2 シミュレーション結果

離散値系と同様に4種類の *Artorg* を用いてシミュレ
ーションを行なった。そのシミュレーション結果を表3に
示す。これは最終的に生存している *Artorg* の存在する
場所の関数値の最小値である。ここで、パラメータの
値としては $D = 1.0, W = 0.2$ を与え、初期 *Artorg* 数は
160匹とした。また、探索空間として f_B, f_C に対しては
 $x_{min} = (-1.5, -1.5), x_{max} = (1.5, 1.5)$ とし、 f_D については
上記の条件に $y_{min} = (-1.5, -1.5), y_{max} = (1.5, 1.5)$ を加
えた。

ここでは、全ての目的関数に対して D, W の値を一定と
したが、実際にはそれぞれの目的関数に対して適切な値を
与えるほうが良い結果が得られる。 D, W の値について簡
単に解説すると、 D を大きくすると *Artorg* の移動可能範
囲が広がるため、*E.C.* の発生に時間がかかる、あるいは
発生しなくなる等の傾向がある。逆に D を小さくすると
Artorg の移動可能範囲が狭まり、他の *Artorg* との交流機

会が減少するために、全体として総個体数の減少、さら
には全 *Artorg* の死滅等の傾向がでてくる。 W の値は目的
関数によって変化させるのが望ましいと考えられる。具体
的には、目的関数の関数値の起伏が激しい場合は、 W を小
さく、逆に目的関数の関数値の起伏が滑らかな場合は、 W
を大きくしたほうがよいようである。

表3 シミュレーション結果
Table 3. Simulation Result

	Best Sol.	Optimal
Banana	0.0009	0.0000
Camel	-1.0312	-1.0316
Double Camel	-2.0421	-2.0632

E.C. が望ましい位置に発生していることを確かめるた
めに、図6に *Banana Function* の等高線を示し、*Artorg*
の初期分布を図7に示し、*E.C.* の様子を図8に示す。図8
の状態は2000サイクル後の状態で約230匹程度の *Artorg*
が生息している。図6と図8より最適解の周辺に *E.C.* が
発生していることが分かる。

図6 *Banana Function* の等高線
Fig. 6. Contour Line for *Banana Function*

図7 *Artorg* の初期分布 (*Artorg* 数 = 160)
Fig. 7. Initial distribution of *Artorg*
(Number of *Artorgs* = 160)

図 8 Banana Function における E.C. (Artorg 数 = 232)
Fig. 8. E.C. for Banana Function
(Number of Artorgs = 232)

また、有望と思われる関数値の小さい解空間に E.C. が実際に発生していることを確かめるために、時刻 t における各 Artorg の位置の関数値と大域的最小解の差の 2 乗平均を次式で定義する。

$$m(x,t) = \frac{1}{n} \sum_{i=1}^n (f(x_i(t)) - f(x_{opt}))^2 \tag{7}$$

ここで、 n は Artorg の総個体数、 $f(x_{opt})$ は目的関数 $f(x)$ の大域的最小解である。図 9 に Banana Function におけるシミュレーション時の $m(x,t)$ の変化を示す。この図より関数値の小さい空間に Artorg が E.C. を生成していることがわかる。

図 9 大域的最適関数値からの二乗平均値
Fig. 9. Mean Square Value from Global Optimum

6.3 他手法との比較

本手法の有効性を検証するために他手法との比較を行った。比較の対象として Simplex 法⁽⁹⁾と Tree Annealing 法⁽¹⁰⁾を用いた。Simplex 法は探索空間上にシンプレックスと呼ばれる幾何学的図形の頂点に点を配置し、それらの点における目的関数の値を比較することにより、新しい点を次々と生成する方法であり、非常に高速に解を発見することが可能であるが、多峰性関数に対して容易に局所解に陥る傾向がある。一方、Tree Annealing 法は多くの計算時間

が必要ではあるが、多峰性関数に対しても大域的最適解を発見することが可能である。比較は Camel Function と Double Camel Function を用いて行なった。解の精度を同程度に設定したときの計算時間を表 4 に示す。ここでは、Simplex 法を用いて Camel Function の解を求めたときの計算時間を基準とした比で表わしている。

表 4 より、本手法では Tree Annealing 法と比較して倍以上の高速化が実現できたことがわかる。Simplex 法と比較すると計算時間は大きい、Simplex 法は多峰性関数に対しては局所解に陥りやすい。これに対して、本手法は大域的探索を行なうので局所解に陥りにくい。また、問題の規模の増加に伴う計算量の増加割合も本手法が最も小さく、並列機構の導入による、より一層の高速化も可能である。

表 4 計算時間の比較
Table 4. Comparison of CPU Time

Method	CPU Time (Camel)	CPU Time (Double)
Simplex	1.00	4.15
Tree Annealing	43.53	102.94
ALife	21.76	33.53

7. 結言

勾配情報を基にした最適化手法では初期点を適切に設定しなければ多数の局所解を持つ関数に対しては大域的最適解を得られない可能性が高いが、本手法においては多峰性関数に対して、大域的最適解に近い良好な解の探索が実現できた。また、実際の計算時間も Tree Annealing 法等の勾配を用いない大域的最適化手法と比較してかなり小さく、求める関数の次元が大きくなっても計算時間が急激に増加することもない。

本手法は関数値のみを用いており、凸性・微分可能性・連続性等の仮定を必要としないため、より多くの問題に適用が可能と考えられる。今後の課題としては本手法と他の最適化手法の結合により、より精度の高い解の探索を行うことと、最適制御等における実際の最適化問題に対してその有効性を検証することである。

(平成 7 年 7 月 7 日受付, 同 7 年 11 月 2 日再受付)

文 献

[1] J. コワリック, M. R. オスボーン, 非線形最適化問題, 培風館, 1970.
[2] A. L. Peressini, F. E. Sullivan and J. J. Uhl, Jr, "The Mathematics of Nonlinear Programming", Springer-Verlag, 1993.

- [3] P. J. M. van Laarhoven and E. H. L. Aarts, "Simulated Annealing : Theory and Applications", Kluwer Academic Publishers, 1987.
- [4] A. Colomi, M. Dorigo, and V. Maniezzo, "Distributed Optimization by Ant Colonies", in *Toward A Practice of Autonomous Systems*, MIT Press, 1992.
- [5] A. M. Assad and N. H. Packard, "Emergent Colonization in an Artificial Ecology", in *Toward A Practice of Autonomous Systems*, MIT Press, 1992.
- [6] 林, 佐藤, 沖田, "Artificial Life を用いた大域的最適化", 電気・情報関連学会中国支部第 45 回連合大会講演論文集, pp. 164-165, 1994.
- [7] 北野 宏明 編, 遺伝的アルゴリズム, 産業図書, 1993.
- [8] A. V. Levy and A. Montalvo, "The Tunneling Algorithm for The Global Minimization of functions", *SIAM J. Sci. Stat. Comput.*, Vol. 6, No 1, 1985.
- [9] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization", *Computer J.*, Vol. 7, 1965.
- [10] G. L. Bilbro and W. E. Snyder, "Optimization of Function with Many Minima", *IEEE Trans. Syst., Man & Cybern.*, Vol. 21, 1991.

林 大 作

(準員) 昭和 47 年 3 月 6 日生。平成 6 年山口大学工学部電子工学科卒業。同年 4 月山口大学大学院工学研究科電気電子工学専攻博士前期課程進学, 現在に至る。主として, 非線形最適化の研究に従事。

佐 藤 泰 司

(正員) 昭和 32 年 12 月 14 日生。昭和 60 年 3 月広島大学大学院工学研究科システム工学専攻博士課程修了。同年 4 月広島大学工学部助手。平成 3 年 4 月山口大学工学部電気電子工学科助教授, 現在に至る。工学博士。主として, 非線形最適化および組合せ計画に関する研究に従事。日本 OR 学会, 計測自動制御学会, IEEE 会員。

沖 田 豪

(正員) 昭和 12 年 8 月 8 日生。昭和 39 年 3 月東北大学大学院修士課程修了。同年 4 月広島大学工学部助手。同講師, 同助教授を経て, 53 年 4 月山口大学工学部教授。平成 7 年 4 月広島市立大学情報科学部教授, 現在に至る。工学博士。主として, システム同定の研究に従事。計測自動制御学会, システム制御情報学会会員。