

動的パラメータ変更を導入したコーシー適応型粒子群最適化法に関して

吉丸 直人¹ 中原 啓貴¹ 大島 賢一¹ 井上 拓弥¹

概要: LSI の微細化に伴う製造ばらつきを解決するため, 出荷時に微調整が行われており, 調整値の探索は組合せ探索問題に帰着できる. 通常, LSI はカタログスペックを満たせば良いので, 必ずしもチップ毎の最大の性能を引き出す調整値を求める必要はない. また, 生産効率を上げるためには調整値を短時間で見つける必要がある. 従って, LSI の調整値探索は時間制約付き準最適解を求める問題である. 近年, 群知能を用いた最適化アルゴリズムの一種として粒子群最適化 (PSO: Particle Swarm Optimization) が提案されており, 局所解からの脱出を企図したコーシー適応型 PSO (CMPSO: Cauchy Mutation PSO) が多用されている. 本論文では, 探索中にパラメータを動的に変更して高速に準最適解を求める改良 CMPSO (ICMPSO: Improved CMPSO) を提案する. ICMPSO は, 探索初期に広域探索を行うパラメータを用いて大まかな最良解を求め, 探索後期に狭域探索を行うパラメータを用いて準最適解を求める. 既存の多峰性関数を用いてパラメータに対する提案アルゴリズムの特徴を解析し, 高速に準最適解を求めるためのパラメータ変更法を提案する. 既存の PSO との比較を行い, 提案する ICMPSO が最も高速に準最適解を発見できることを示す.

1. はじめに

1.1 時間制約付き準最適解探索問題

LSI の微細化に伴う製造ばらつきにより, 出荷時に微調整が行われており, 調整値の探索は組合せ探索問題に帰着できる. 通常, LSI はカタログスペックを満たせば良いので, 必ずしもチップ毎の最大の性能を引き出す調整値を求める必要はない. また, 生産効率を上げるためには調整値を短時間で見つける必要がある. 従って, 時間制約付き準最適解を求める問題に帰着できる. 我々はこの問題の一種である光配線用 VCSEL ドライバのエンファシス値 [2] を自動的に求める問題に取り組んでいる. 本論文では時間制約付き準最適解を求める発見的アルゴリズムの改良法を提案する.

1.2 粒子群最適化法

近年, 群知能を用いた最適化アルゴリズムの一種として粒子群最適化 (PSO: Particle Swarm Optimization) [4] が提案されている. 群知能は, 単純な動きをする生物が集団として複雑かつ合理的な動きを行い, 群れ全体として最適解を探索することで高度な知能を発揮する. PSO の問題点は多峰性関数最適化における局所解への初期収束である.

多くの最適化問題は多峰性関数の最適化に帰着でき, PSO をそのまま適用したのでは良い解を得ることは困難であることが報告されている [6]. そこで, 局所解からの脱出を企図したコーシー適応型 PSO (CMPSO: Cauchy Mutation PSO) [5] が提案されている. CMPSO では粒子の移動にコーシー分布に基づくランダム性を導入している. すなわち, 高い確率で狭領域を探索しながらごく稀に広領域を探索するロングジャンプを導入することで局所解からの脱出を実現している.

1.3 提案手法

LSI の微調整等では準最適解な調整値を早く見つけることで生産性を上げることが可能である. CMPSO ではコーシー分布の広がり γ を決めなければならない. 一般的に γ を大きくするとロングジャンプが頻繁に発生し, 広域探索が行われる. 一方, γ を小さくすると狭域を探索する. 既存の CMPSO は γ を初期に与えるのみで, 探索中の変更は行われていなかった. そこで, 本論文では動的にパラメータを変更することで準最適解を高速に求める改良 CMPSO (ICMPSO: Improved CMPSO) を提案する. PSO の欠点は早い段階で局所解へ陥ることであるから, γ を大きめにとって広域を探索しておき, ある程度最良解を得られれば γ を小さくして狭域を集中的に探索することで, 高

¹ 鹿児島大学
Korimoto 1-21-40, Kagoshima, 890-0065, Japan

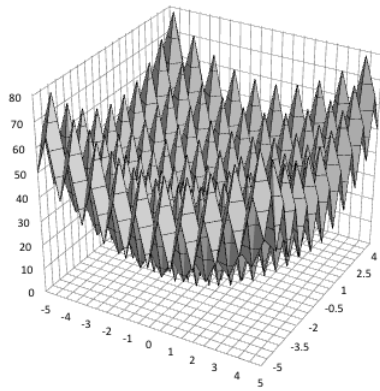


図 1 Rastrigin 関数の概形.

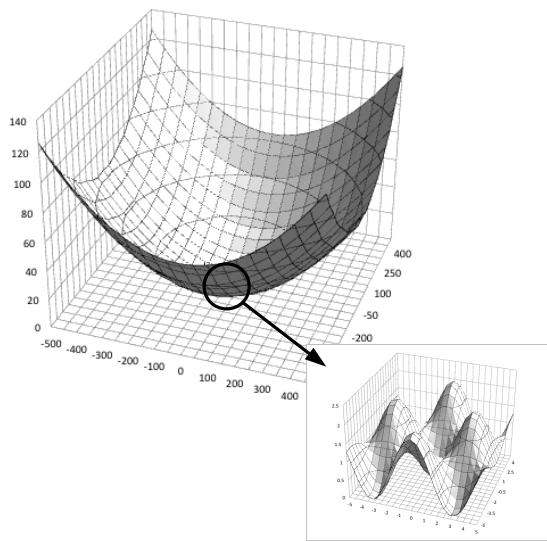


図 2 Griewank 関数の概形.

速に準最適解を求めることができる。本論文では、一般的な多峰性関数である Rastrigin 関数, Alpine 関数, Griewank 関数を用いて γ に対する CMPSO の傾向を明らかにし, γ を変更する指針を提案する。 γ を動的に変更する ICMPSO を実装し, 既存の PSO と比較を行い探索時間と解の精度を明らかにする。

1.4 本論文の構成

第 2 章では時間制約付き準最適解探索問題を述べ, 第 3 章ではコーシー適応型粒子群最適化法 (CMPSO) について述べ, 第 4 章では動的パラメータ変更を導入した ICMPSO について述べ, 第 5 章では実験結果を示し, 第 6 章で本論文のまとめを行う。

2. 時間制約付き準最適解探索問題

2.1 多峰性関数

多峰性関数とは準最適解となる凸が多数存在する関数であり, 一般的な PSO では局所解に陥るため難しいとされている関数である。現実存在する多峰性関数最適化問題と

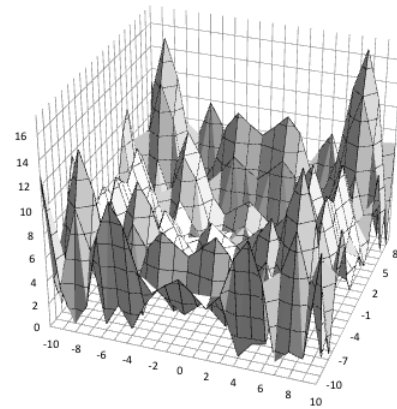


図 3 Alpine 関数の概形.

して, 建築材料のひずみエネルギー最小化, ダム構造や航空機の翼の重量最小化 [1], タンパク質の構造分析 [3] 等が挙げられる。本論文では一般的に知られた多峰性関数を用いて評価を行う。これらのベンチマーク関数は予め最適値が知られており, その特徴も明らかになっているためアルゴリズムの特徴を把握するのに適している。

2.1.1 Rastrigin 関数

図 1 に **Rastrigin 関数**の概形を示す。Rastrigin 関数は最適解の周辺に格子状に準最適解 (最適値に近い値を持つ局所的最適解) を持つ多峰性関数であり, 設計変数間に依存関係はない。Rastrigin 関数は以下の式で与えられる。

$$f_R = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

最適解は $f_R(x_1, x_2, \dots, x_n) = (0, 0, \dots, 0) = 0$ である。

2.1.2 Griewank 関数

図 2 に **Griewank 関数**の概形を示す。Griewank 関数は設計変数間に依存関係を有する多峰性関数である。大域的には単峰性関数のような性質を持つため, 準最適解は比較的容易に求まる。しかし, 局所的には多数の局所的最適解が存在するため, 最適解を発見するのは困難である。Griewank 関数は以下の式で与えられる。

$$f_G = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

最適解は $f_G(x_1, x_2, \dots, x_n) = (0, 0, \dots, 0) = 0$ である。図 2 が示すように, Griewank 関数は大域的には凸関数のように見えるが, 拡大すると多峰性関数となっていることがわかる。

2.1.3 Alpine 関数

図 3 に **Alpine 関数**の概形を示す。Rastrigin 関数, Griewank 関数が関数の凸が鋭い強い多峰性関数であるのに対し, Alpine 関数は凸が緩やかな弱い多峰性関数である。また, 設計変数間に依存関係はない。Alpine 関数は以下の式で与えられる。

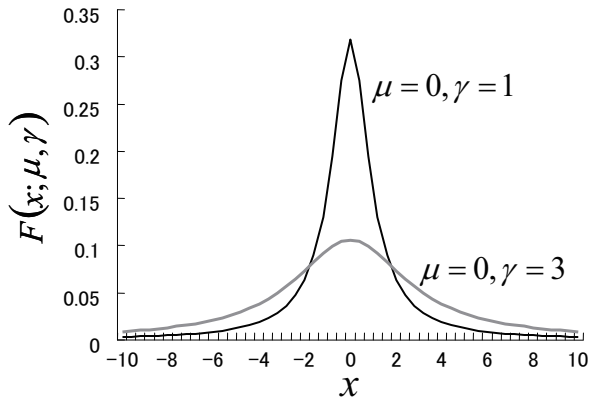


図 4 コーシー分布の例.

$$f_A = \sum_{i=1}^n |x_i \sin x_i + 0.1 x_i|$$

最適解は $f_A(x_1, x_2, \dots, x_n) = (0, 0, \dots, 0) = 0$ である.

3. コーシー適応型粒子群最適化法 (CMPSO)

3.1 粒子群最適化法

粒子群最適化法 (PSO: Particle Swarm Optimization) は, J.Knedy と R.Eberhart によって開発された社会的行動を基にしたメタヒューリスティクス (発見的手法) である. その基本的な概念は, 鳥などの群れで餌を探す過程において情報を群れ全体で共有しているという現象に基づく. すなわち, 群れを構成する個体が独立に行動するのではなく各個体の個別情報と群れ全体の共通情報を組み合わせる規則に則って行動するところにある. その概念に基づき, 粒子群最適化法は位置や速度の情報を伴う微粒子 (Particle) が, 人工生命の研究における群れ (Swarm) を構成し探索を行う. 粒子群最適化法の特徴は, 多点探索をおこなう発見的最適化手法であり, シンプルなアルゴリズム構成で構築できる. また, 各点の情報と多点間で共有する最良解の情報に基づいて探索を行い, 最適解もしくは準最適解への到達を目指す.

n 次元多峰性関数 $f(\vec{X})$ の最適化を考える. PSO は i 番目の粒子 ($1 \leq i \leq N$, N は粒子数) 毎に座標 $\vec{X}_i = (x_1, x_2, \dots, x_n)$ と速度 $\vec{V}_i = (v_1, v_2, \dots, v_n)$ の組を持つ. また, 各粒子は以前の最良値 $p_i = f(\vec{X}_i')$ とその座標 $\vec{X}_i' = (x_{p1}, x_{p2}, \dots, x_{pn})$ を記憶しており, 粒子全体の最良値 $p_b = f(\vec{X}_b)$ とその座標 $\vec{X}_b = (x_{b1}, x_{b2}, \dots, x_{bn})$ を知ることができる. 各粒子 i は次の速度ベクトルの各成分 $v_{ij}^{(t+1)}$ を以下の式で更新する.

$$v_{ij}^{(t+1)} = w \cdot v_{ij}^{(t)} + c_1 \cdot \text{rand}() \cdot (x_{pj} - x_{ij}^{(t)}) + c_2 \cdot \text{rand}() \cdot (p_{bj} - x_{ij}^{(t)}) \quad (1)$$

ここで, w は慣性重量 [7], c_1, c_2 は加速定数であり, $\text{rand}()$ は $[0, 1]$ をとる一様乱数である. PSO は速度ベクトルを用いて次の位置ベクトルの各成分 x_{ij} を以下の式で更新する.

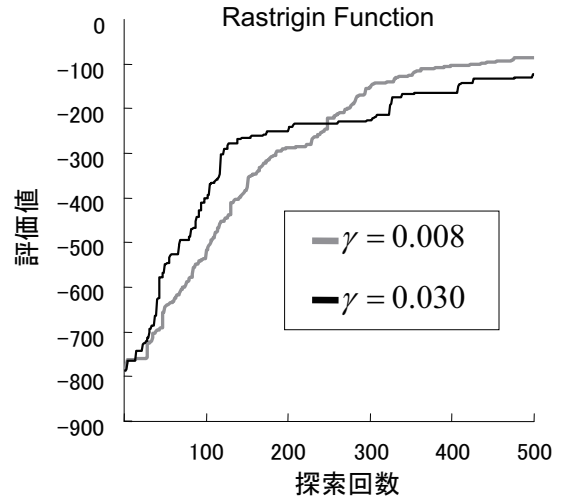


図 5 尺度母数 γ による CMPSO の影響 (Rastrigin 関数).

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t+1)} \quad (2)$$

3.2 コーシー適応型粒子群最適化法

粒子群最適化法は低次元多峰性関数の最適解を比較的高速に求めることができる. しかし, 高次元では局所解に陥るため探索能力を十分に発揮できないことが知られている. その原因は, 各粒子は p_{bj} と x_{pj} の重みつき平均の周りを減衰振動し新たな p_{bj} や x_{pj} が見つからないときに速度を失い, 局所解に収束してしまうからである. **コーシー適応型粒子群最適化法 (CMPSO: Cauchy Mutation PSO)** は局所解への脱出を目的としてコーシー分布に基づく乱数を用いて次のステップの移動量を求める. コーシー分布は

$$f(x; \mu, \gamma) = \frac{1}{\pi \gamma (1 + (\frac{x-\mu}{\gamma})^2)}$$

で与えられる. μ は分布の最頻値を与える位置母数, γ は半値半幅を与える尺度母数である. コーシー分布は期待値や分散が定義されない確率分布である. 図 4 にコーシー分布の例を示す. コーシー分布は最頻値 μ 付近が鋭く, 両裾に値を持っている特徴がある. γ が小さいときは分布の中心に高い確率で次状態に移移する集中性を持つため狭域探索を行う. 一方, γ が大きいときは両裾の広さからロングジャンプを可能するため, 広域探索も可能である.

コーシー分布の逆累積分布関数, 即ち確率値は以下の式で与えられる.

$$F^{-1}(q; \mu, \gamma) = \mu + \gamma \cdot \tan(\pi(q - 0.5)) \quad (3)$$

ここで, q は一様な乱数である. コーシー適応型粒子群最適化法 (CMPSO) は以下の式を用いて位置ベクトルを更新する.

$$x_{ij}^{(t+1)} = F^{-1}(q; x_{ij}^{(t)}, \gamma) \quad (4)$$

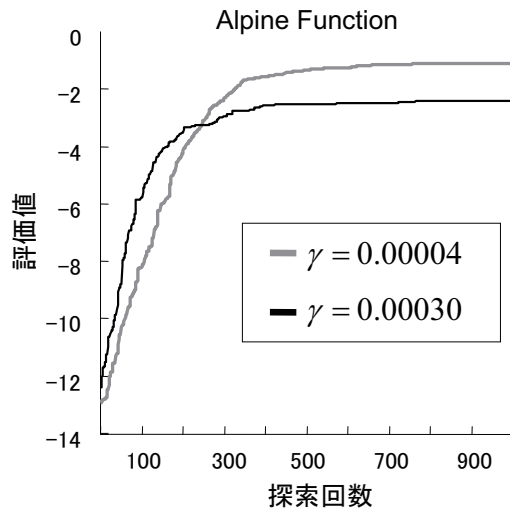


図 6 尺度母数 γ による CMPSO の影響 (Alpine 関数).

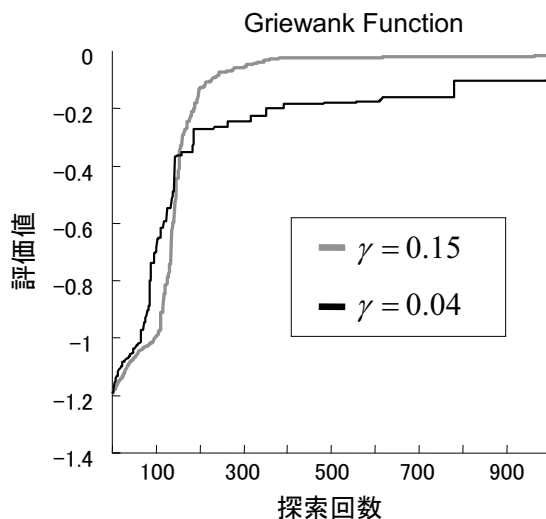


図 7 尺度母数 γ による CMPSO の影響 (Griewank 関数).

4. 動的パラメータ変更を導入した CMPSO

4.1 パラメータと最良解の関係

提案手法は CMPSO で探索中に尺度母数 γ を変更する。まず、ベンチマーク関数を用いて γ に対する CMPSO の影響を明らかにする。図 5 に γ による Rastrigin 関数を用いた CMPSO の影響を、図 6 に γ による Alpine 関数を用いた CMPSO の影響を、図 7 に γ による Griewank 関数を用いた CMPSO の影響をそれぞれ示す。ここで、各図の横軸は探索回数を示し、縦軸は評価値を表し値が小さい程解が優れていることを表す。

図 5, 6, 7 が示すように、 γ が大きいときは広域探索するので準最適解に比較的早く近づくことができるが、狭域探索が行われないため集中性を失い、準最適解を得ることができない。一方、 γ が小さいときは狭域探索するので準最適解を得ることができるが、広域探索が行われないため多様性を失い、準最適解を得るには遅くなってしまう。

実験的な知見により、早い時間に準最適解を得るには探索初期の段階で γ を大きく設定して準最適解に近づき、途中で γ を小さく設定して集中的に準最適解を探索するのがよいと予想される。本論文では、この方針に基づいた新しいアルゴリズムを提案する。

4.2 動的パラメータ変更を導入した CMPSO

尺度母数 γ を探索中に更新する改良 CMPSO (ICMPSO: Improved CMPSO) のアルゴリズムを示す。

アルゴリズム 4.1 アルゴリズムで求める目標値を α とし、パラメータを変更する目標値を β とする。ただし、 $\alpha > \beta$ である。

1. 各粒子に対して速度と位置をランダムにセットする。
2. 各粒子 i に対して評価関数 $f(\vec{X}_i)$ を求める。
3. 各粒子に対して、 $f(\vec{X}_i)$ が自身の最良値 $f(\vec{X}_{pi})$ よりも大きい場合、 $f(\vec{X}_{pi})$ を更新する。
4. 各粒子に対して、 $f(\vec{X}_{pi})$ が粒子全体の最良値 $f(\vec{X}_b)$ よりも大きい場合、 $f(\vec{X}_b)$ を更新する。
5. 粒子全体の最良値 $f(\vec{X}_b)$ が設定した値 α よりも大きい場合、9. へ。
6. 粒子全体の最良値 $f(\vec{X}_b)$ が設定した値 β よりも大きい場合、尺度母数 γ を変更する。
7. 各粒子に対して、以下を行う。
 - 7.1. 式 (1), (2) に基づき粒子 t を更新する。
 - 7.2. 式 (4) に基づき粒子 t' を更新する。
 - 7.3. 粒子 t と t' を比較し、評価関数 f が小さい方を次の粒子として選択する。
8. 2. へ戻る。
9. 停止。

アルゴリズム 4.1 において、手順 7.2 は γ の範囲で探索する。これは局所探索と同じ考え方であるが、コーシー分布に基づく探索であるため、ごく稀に広域探索を行うことが可能である。従って、提案アルゴリズムは局所解に陥っても脱出が可能である。

通常、 α は問題毎に予め与えられている。 β は問題に応じて経験的に指定する必要がある。また、尺度母数 γ の変更量も経験的に指定する必要がある。次の章でベンチマーク関数を用いて傾向を明らかにする。

5. 実験結果

5.1 実験環境

提案する ICMPSO を C 言語で実装し、既存の PSO との比較を行った。実装に用いた PC は Core i3 2.4GHz (1 次キャッシュ: 512kB (データ), 512kB (命令), 2 次キャッシュ: 4MB), メインメモリ 2GB, OS: Windows7 Home Edition である。コンパイラは gcc を使い、最適化オプション-O3 を与えて実効コードを生成した。

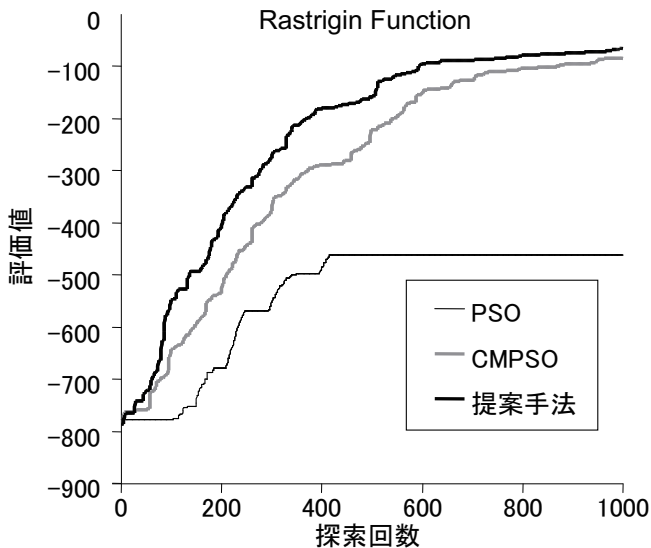


図 8 Rastrigin 関数を用いた比較.

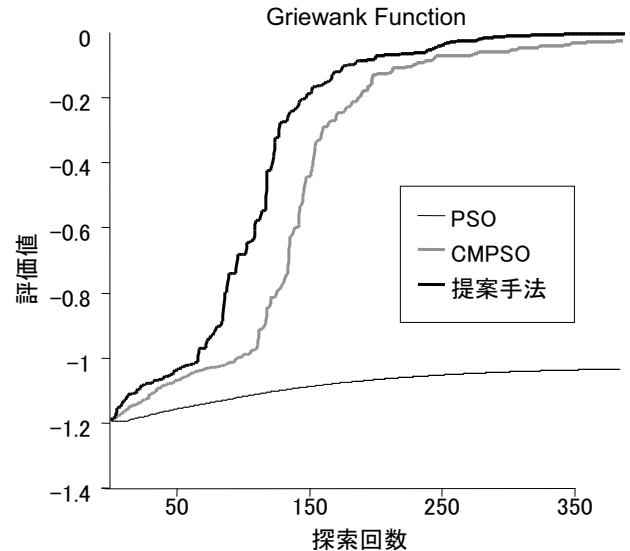


図 10 Griewank 関数を用いた比較.

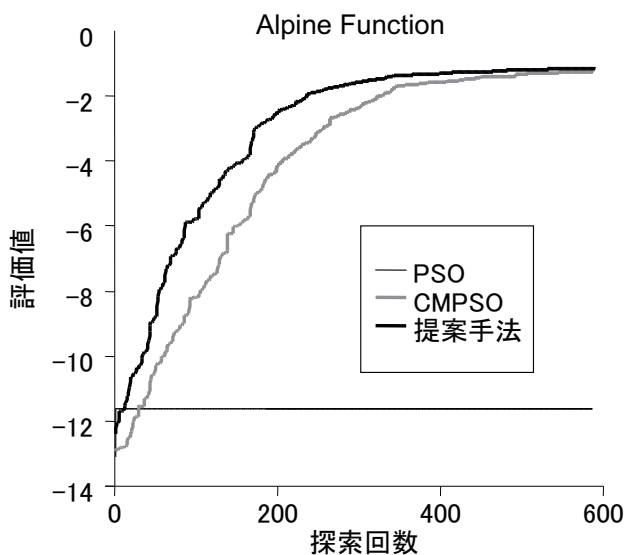


図 9 Alpine 関数を用いた比較.

5.2 既存の PSO との比較

2 章で述べた関数を用いてオリジナルの PSO (PSO), コーシー適応型 PSO (CMPSO) と提案手法 (提案手法) の比較を行った. 図 8 に Rastrigin 関数を用いた比較結果を, 図 9 に Alpine 関数を用いた比較結果を, 図 10 に Griewank 関数を用いた比較結果をそれぞれ示す. 各図の横軸は探索回数を示し, 縦軸は評価値を表し値が大きい程解が優れていることを表す.

各 PSO のパラメータは実験的に求めた値を設定した. ICMPSO は途中で γ の値を変更するが, 初期の段階で広域探索を行い, 途中から狭域探索に切り替えた方が局所解に陥ること無く早く最良解に辿り着いた. これは予備実験に基づく知見と同じ傾向である. 比較実験では, Rastrigin 関数に関しては $\gamma = 0.03$ を探索回数 160 で 0.008 に, Alpine 関数に関しては $\gamma = 0.0003$ を探索回数 80 で 0.00004 に, Griewank 関数に関しては $\gamma = 0.15$ を探索回数 75 で 0.1

に, そして探索回数 150 で 0.04 に変更した.

図 8, 9, 10 より, ICMPSO が最も早く最良解にたどり着いていることがわかる. 特に, 従来の PSO では早い段階で局所解に陥りやすい Alpine 関数, Rastrigin 関数でも局所解に陥ること無く探索ができていたことがわかる. また, コーシー適応型 PSO と比較すると, 提案手法は途中で γ を小さく取り局所探索に切り替えているため, 早い探索回数で準最適解に達していることがわかる. 従って, ICMPSO は時間制約付き準最適解を求める問題に対して最も効果的なアルゴリズムであると言える.

6. まとめ

本論文では, 時間制約付き準最適解を求める問題に対して CMPSO に探索中にパラメータを動的に変更する方法を加えた ICMPSO を提案した. 既存の多峰性関数を用いて CMPSO の特徴を解析した結果, 探索の初期では広域探索を行うパラメータに設定して大まかな最良解を求め, 探索後期になると狭域探索を行うパラメータに設定して準最適解を求めると比較的早く準最適解に達することを発見した. 既存の多峰性関数を用いて既存の PSO との比較を行い, 提案する ICMPSO が最も早く準最適解に達することを示した.

今後の課題は提案アルゴリズムを実際の最適化問題に適用することである. 具体的には光ドライバ VCSEL の調整機器に組み込み, その有用性を明らかにする予定である.

謝辞 本研究は, 一部, 日本学術振興会・科学研究費補助金 (若手 (B)) による.

参考文献

- [1] 相吉英太郎, 安田恵一郎, “メタヒューリスティクスと応用,” オーム社, 2007.
- [2] N. Chujo, T. Takai, T. Sugawara, Y. Matsuoka,

D. Kawamura, K. Adachi, T. Kawamata, T. Ohno, and K. Ohhata, “A 25Gb/s 65-nm CMOS Low-power laser diode driver with mutually coupled peaking inductors for optical interconnects,” *IEEE Trans. on Circuits and Systems-I: Regular Papers*, Vol. 58, No. 9, 2011, pp. 2061-2068.

- [3] 金久実, “ゲノム情報への招待,” 共立出版, 1996.
- [4] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *IEEE Int’l Conf. on Neural Networks*, 1995.
- [5] C. Li, Y. Liu, A. Zhou, L. Kang, and H. Wang, “A fast particle swarm optimization Algorithm with Cauchy mutation and natural selection strategy,” *Advances in Computation and Inteligence*, Vol. 14683, pp. 334-343, 2007.
- [6] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, “Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients,” *IEEE Trans. on Evolutionary Computation*, vol. 8, No. 3, pp. 240–255, 2004.
- [7] Y. Shi and R. Eberhart, “A modified partilce swarm optimizer,” *IEEE Congress on Evolutionary Computation (CEC 1998)*, pp.69-73, 1998.