

## 停滞回避 PSO の提案と評価

小林史典<sup>†1</sup> 金子正人<sup>†2</sup> 岩井俊哉<sup>†2</sup>

**概要** : Particle Swarm Optimization (PSO) は、関数の最小値問題のような連続値最適化問題に適用できるメタヒューリスティクスの一つである。一般に PSO だけでなくメタヒューリスティクス解法では、まだ訪れていない探索空間を探索する大域探索能力と既に訪れた探索空間をさらに探索する局所探索能力が必要である。PSO は局所探索能力が強く、局所解に陥りやすい。そのため PSO に大域探索能力を取り込んだ様々な改良版が考案されている。しかしながら、改良版 PSO は関数のタイプにより探索性能が異なる。従って、関数のタイプに依らず探索性能が高い汎用的な PSO を改良することが重要である。Dynamical Multi Swarm-PSO (DMS-PSO) を基に、我々は大域および局所的な探索能力を適応的に取り込んだ停滞回避 PSO を提案する。停滞回避 PSO では、探索の停滞時に大域探索粒子により初期収束を防ぎ局所探索粒子により停滞位置の局所探索を行う。ベンチマーク関数の最小値問題で提案手法と既存手法との性能比較を行い、提案手法が関数のタイプに依らず探索性能が高いことを示す。

**キーワード** : 粒子群最適化法, 連続値最適化問題, 初期収束, 局所探索, 大域探索

## Proposal and Evaluation of Stagnation Avoidance PSO

FUMINORI KOBAYASHI<sup>†1</sup> MASATO KANEKO<sup>†2</sup> TOSHIYA IWAI<sup>†2</sup>

**Abstract** : Particle swarm optimization (PSO) is one of meta-heuristics that can be applied to such continuous optimization problems as minimization problems of function. Generally, not only PSO but also meta-heuristics require both the global search ability, which make particles explore the never visited regions of the search space, and the local search ability, which make particles explore the already visited regions of the search space. PSO often causes particles to trap into local optimum, because it has high local search ability. Thus, various PSO variants are proposed with the aim to incorporate the global search ability. However, the search efficiency of the variant PSO changes according to the difference in the type of functions. Therefore, it is important to devise such a variant PSO that perform well for all types of functions. Based on Dynamical Multi Swarm PSO (DMS-PSO), we propose Stagnation Avoidance PSO (SA-PSO) that adaptively incorporate both local and global search abilities. When searching processes stagnate, SA-PSO enable global search particles to prevent stagnation convergence into a local minimum and local search particles to exploit around regions where particles stagnate. By comparison with SA-PSO and various PSO variants, it is found that SA-PSO show high search efficiency for minimization problems of benchmark function irrespective of types of function.

**Keywords** : particle swarm optimization (PSO), continuous optimization problem, stagnation convergence, local search, global search

### 1. はじめに

Particle Swarm Optimization (以下、PSO と略記) [1][2] とは、J. Kennedy と R. C. Eberhart らによって提案された群知能の一種であり、鳥や魚の群れの振る舞いを模倣したメタヒューリスティクスである。PSO では多次元の関数空間上を位置ベクトルと速度ベクトルを持つ多数の粒子が、情報を共有して関数の最適値の探索を行う。PSO では、まだ訪れていない探索空間を探索する大域探索能力と、既に訪れた有望な探索空間をさらに探索する局所探索能力が必要である。しかし、この二つの能力は相反する性質を持つため、バランスよく二つの能力を実装することは難しい。PSO は局所探索能力が強く、多峰性関数の関数最適化問題では

局所解に陥りやすい欠点を持っている。そのため関数空間を隈なく探索する大域探索能力を取り込んだ CLPSO[3], DMS-PSO[4], StPSO[5]など様々な改良版が考案されている。これらの改良版 PSO は PSO より探索性能が高いが、関数のタイプにより探索性能が異なる。例えば CLPSO は単峰性関数に比べ多峰性関数で探索性能が高く、逆に DMS-PSO は多峰性関数に比べ単峰性関数で探索性能が高い。そこで、関数のタイプに依らず探索性能が高い汎用的な PSO の改良が必要である。その一つの試みとして、探索の状態に適応してパラメータを自律的に調節する APSO[6]などの手法が提案されている。本研究の提案手法も探索の状態を推定するという点では APSO と類似しているが、探索状態の推定によりトポロジーを変更し大域および局所探索能力を適応的に制御する点が APSO と異なる。

本研究では Dynamical Multi Swarm-PSO (以下、DMS-PSO と略記) を基に、停滞回避アルゴリズムにより

<sup>†1</sup> 日本大学大学院工学研究科  
Graduate School of Engineering, Nihon University

<sup>†2</sup> 日本大学工学部  
College of Engineering, Nihon University

大域および局所探索能力を適応的に取り込んだ停滞回避 PSO を提案する。さらに、提案手法の探索性能を検証するために、ベンチマーク関数の最小値問題で4つの既存手法との性能比較を行う。

## 2. 停滞回避 PSO の提案

まず2.1節では、提案手法に用いた PSO の改良版である DMS-PSO とその一部改良版である MDMS-PSO のアルゴリズムを説明する。次に2.2節で提案手法である停滞回避 PSO の概要を説明し、2.3節で提案手法が用いている停滞回避アルゴリズムを詳しく述べる。基となる改良版 PSO の違いから提案手法には二種類あり、停滞回避 DMS-PSO (以下、英文表記では SA-DMS-PSO と略記) と停滞回避 MDMS-PSO (以下、英文表記では SA-MDMS-PSO と略記) と呼ぶ。本稿では、関数の最小値問題を例に説明する。

### 2.1 DMS-PSO

DMS-PSO とは、Liang らによって提案された PSO の改良版の一つである。DMS-PSO は、全粒子を複数のグループに分割して、グループ内で情報を共有しながら探索を行うことで大域探索能力を取り込んだ手法である。世代数を  $t$  として、 $i$  番目粒子の速度ベクトル  $v_i$ 、位置ベクトル  $x_i$  の更新式を式(1)、(2)に示す。

$$v_i^{(t+1)} = wv_i^{(t)} + c_1r_1(p_i^{(t)} - x_i^{(t)}) + c_2r_2(l_j^{(t)} - x_i^{(t)}) \quad (1)$$

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \quad (2)$$

ここで、 $p_i^{(t)}$  は  $i$  番目粒子の初期世代から現世代  $t$  までの軌道上の最適な位置ベクトル (以下、pbest と略記) であり、 $l_j^{(t)}$  は現世代  $t$  までにグループ  $j$  内で見つけた最適な位置ベクトル (以下、lbest と略記) である。 $w$  は1ステップ前の速度を抑制する慣性重み定数で1より小さい値を取る。 $c_1$ 、 $c_2$  は重み定数と呼ばれるものであり、それぞれ粒子が pbest や lbest へ向かおうとする力の度合いを表す。 $r_1$ 、 $r_2$  は [0, 1] の値を取る一様乱数である。DMS-PSO では、全粒子を同数のメンバーからなるグループに分ける。このとき1つの粒子が複数のグループに属すことはない。また  $R$  世代ごとにグループの組換え (再グループ化) を行う。この再グループ化の際、各粒子の位置・速度および pbest は保持されるが、lbest は新たなグループで再計算される。このように DMS-PSO はグループごとに情報を共有することで大域探索を実現し、再グループ化によるグループ間での情報交換を通して全粒子間でゆっくりと情報を伝達させる。従って DMS-PSO は、PSO と比べ局所探索能力が劣るが、大域探索能力に優れている。PSO の強すぎる局所探索能力を考えると、このことは PSO より DMS-PSO の方が二つの探索能力のバランスが良いことを意味する。

次に、我々が DMS-PSO を一部改良した Modified DMS-PSO (以下、MDMS-PSO と略記) を説明する。

MDMS-PSO は、DMS-PSO の速度更新式(1)中の lbest をグループメンバーの現在の最良位置に置き換えたアルゴリズムである。従って、DMS-PSO における lbest での関数値と比較して MDMS-PSO における lbest での関数値は大きな値となる可能性が高く、一般的に MDMS-PSO は DMS-PSO よりも大域探索能力が高いが、局所探索能力は低い。

### 2.2 提案手法の概要

本研究では、2.1節で説明した DMS-PSO、MDMS-PSO のアルゴリズムが持つグループ化という性質を拡張して探索の停滞時に大域探索粒子と局所探索粒子という二つのグループ群を導入する。DMS-PSO を基にした理由は、PSO、StPSO や CLPSO に比べ DMS-PSO は探索序盤で大域探索能力が高く終盤で局所探索能力が高い手法であり、本研究で提案する停滞回避アルゴリズムを組み込むことで探索性能が高まると考えたためである。停滞回避アルゴリズムとは探索の停滞時に行う操作であり、探索が停滞したと判定したときに全粒子を大域探索粒子と局所探索粒子に二分し、局所収束を防ぐため大域探索粒子を再配置する。停滞回避アルゴリズムと停滞判定については2.3節で述べる。

図1に提案手法である停滞回避 PSO のフローチャートを示す。停滞回避 DMS-PSO と停滞回避 MDMS-PSO の違いはフローチャート中の「全粒子の位置、速度の更新」で使用する更新式の違いにあるため、両手法を図1のフローチャートで説明できる。フローチャート中の gr\_mode とは探索の状態を定める変数であり、gr\_mode=0 では全粒子を DMS-PSO あるいは MDMS-PSO の粒子 (以下、両手法ともに DMS 粒子と略称) として扱い、gr\_mode=1 では全粒子を大域探索粒子と局所探索粒子に二分する。(1) 最初に gr\_mode=0 として全ての粒子を DMS 粒子として探索空間内にランダムに配置し、各粒子の pbest を現在位置とし、グループ化を行う。(2) 次に、更新則に従い粒子の位置と速度を更新し、 $R$  世代毎に再グループ化を行う。さらに更新時に停滞判定を行う。(3) もし停滞条件が満たされた場合、全粒子は局所解あるいは最適解へ落ち込んでいる可能性があるため、gr\_mode=1 として全粒子を大域探索粒子と局所探索粒子に二分し、大域探索粒子の再配置処理を行う。すなわち局所解に落ち込んだ場合の対策として、大域探索粒子を再配置することにより初期収束を回避して大域探索を進める。一方、最適解付近に停滞した場合の対策として、局所探索粒子による最適解のさらなる探索を進める。(4) 停滞回避アルゴリズムの適用後、更新則に従い粒子の位置と速度を更新し、 $R$  世代ごとに再グループ化を行う。ただし、大域探索粒子間・局所探索粒子間で別々に再グループ化を行う。(5) 探索が進み大域探索粒子が gbest を更新した場合、局所探索粒子は更新された gbest と異なる局所解の探索を行っている可能性が高い。そこで、更新された gbest の情報を局所探索粒子にも共有させるために、gr\_mode=0 として全粒子を DMS 粒子とする。

このように停滞判定と *gbest* の更新に従いトポロジーを遷移させて終了条件が満たされるまで探索を行う。

### 2.3 停滞回避アルゴリズム

2.2 節で述べたように、探索が停滞したときに全粒子を大域探索粒子と局所探索粒子に二分し、大域探索粒子の再配置処理を行う。この操作を停滞回避アルゴリズムと呼ぶ。まず探索の停滞判定条件について説明する。次に述べるように、粒子の速度 ( $V_{av}$ ) と最良個体の位置 (*gbest*) の二つの観点から探索が停滞したかどうかを判定する。

(I) **速度による判定**: 次式で定義する速さの平均  $V_{av}$  を求める。この速さの平均は安田ら[7]が着目し探索の状態との関係性を詳しく調べ、彼らの研究で探索の状態の推定に用いているものと同一である。

$$V_{av} = \frac{1}{P_g} \sum_{i=1}^{P_g} \frac{1}{D} \sum_{d=1}^D v_{id} \quad (3)$$

ここで、 $P_g$  は *gr\_mode*=0 のとき DMS 粒子数を表し、*gr\_mode*=1 のとき大域探索粒子数を表す。 $D$  は関数空間の次元数であり、 $v_{id}$  は  $i$  番目粒子の速度の  $d$  次元成分である。カウンタ変数 *count\_V* を設け、*count\_V*  $\geq 3000$  のとき速度の観点から停滞と判定する。 $L$  を関数空間の一次元方向の長さとして、 $V_{av}/L$  の値が小さくなるにつれ停滞の度合いが大きいと見え、 $V_{av}/L$  の値を *count\_V* への加算値の大きさに関係づけ探索過程で *count\_V* を加算する。

(II) **最良個体の位置(*gbest*)による判定**: 世代毎に *gbest* を計測し、以下の条件でカウンタ変数 *count\_g* を加算し、*count\_g*  $\geq 200$  のとき最良個体の位置の観点から停滞と判定する。

(i) 各世代で、大域探索粒子により *gbest* が更新され、*gbest* の関数値の減少分が  $1.0 \times 10^{-8}$  以下のとき、*count\_g* に 1 を加算する。(ii) 大域探索粒子により *gbest* が更新され、*gbest* の関数値の減少分が  $1.0 \times 10^{-8}$  より大きいとき、*count\_g*

を 0 にリセットする。

上記の二つの停滞判定が両立したとき、再配置処理を行う。再配置処理は次の (1) ~ (4) からなる。(1) 全粒子を大域探索粒子と局所探索粒子に二分し、二つのグループ群を作成する。(2) 大域探索粒子の位置と速度をランダムに再配置する。(3) 大域探索粒子の位置と速度を再配置しても、*pbest* を保持している大域探索粒子は再配置後に停滞位置に再び戻るので、*pbest* を次元成分ごとに確率  $P$  で再配置し、確率  $1-P$  で *pbest* の当該成分を保持させる。(4) さらに、*pbest* の再配置で *gbest* の情報を失った場合、その後の探索性能が低下する可能性がある。そこで  $m$  個の粒子をランダムに選び、その粒子の *pbest* に *gbest* を格納することで *gbest* を保存した。

## 3. 数値実験

### 3.1 数値実験の内容と方法

まず、提案手法である停滞回避 DMS-PSO、停滞回避 MDMS-PSO と提案手法の基になっている DMS-PSO で、大域探索能力を比較する。そのため、関数の最小値問題で全粒子・全次元成分の平均位置  $\mu$  と標準偏差  $\sigma$  を求め、 $\mu - \sigma$ 、 $\mu$ 、 $\mu + \sigma$  の世代推移を比較する。次に、提案手法である停滞回避 DMS-PSO、停滞回避 MDMS-PSO と既存手法 PSO, DMS-PSO, StPSO, CLPSO との探索性能を比較するため、ベンチマーク関数の最小値問題をそれぞれのアルゴリズムで 100 回数値実験し、そこで得た *gbest* の関数値の加算平均 (以下、 $\langle f(\text{gbest}) \rangle$  と記述) の世代推移を測定した。全手法で用いるパラメータの値を表 1 に示した。表 1 の「1 次元成分の最大速度」とは、粒子が持つ各次元成分の速度の大きさの最大値である。つまり、速度更新で求めた次世代の速度成分の大きさが最大速度を超えたとき、それを最大速度の値とする。また、慣性重み  $w$  の値を大きく設定すると大域探索能力が強くなり、小さく設定すると局所探索能力が強くなるということが知られている。本稿では表 1 のように停滞回避 DMS-PSO では  $w$  を世代に依らず一定とし、停滞回避 MDMS-PSO では世代とともに  $w$  の値を線形減衰させた[8]。提案手法で必要となるパラメータについて説明する。様々な値の変更確率  $P$  で数値実験を行った結果、関数の次元数に関わらず大域探索粒子の *pbest* の 2 成分程度の値を再配置することで探索性能が向上することが判明し

表 1 全手法で用いるパラメータ値

Table 1 Parameter values used in all methods

パラメータ名	値
粒子数 $N$	90
次元数 $D$	10      30
1 次元成分の最大速度	0.2L, 0.6L
慣性重み $w$	PSO, DMS-PSO, 停滞回避 DMS-PSO : 0.729
	停滞回避 MDMS-PSO : 0.9-(0.7 × 現世代数/最大世代数)
	StPSO, CLPSO : 0.9-(0.5 × 現世代数/最大世代数)

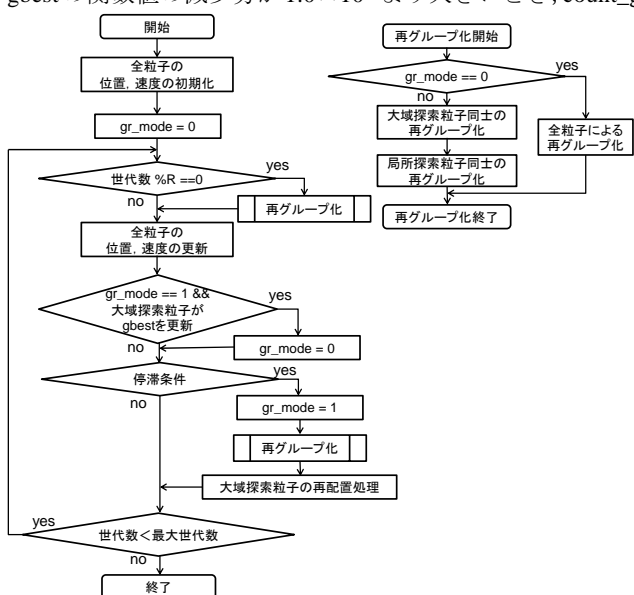


図 1 停滞回避 PSO のフローチャート

Figure 1 Flowchart for SA-PSO

表2 ベンチマーク関数の性質  
 Table 2 Properties of benchmark functions

関数名	形状	変数間の 依存関係	解
①Sphere	単峰性	なし	$\forall d$ で $x_d=0$
②Rosenbrock	単峰性	あり	$\forall d$ で $x_d=1$
③Rastrigin	多峰性	なし	$\forall d$ で $x_d=0$
④Schwefel	多峰性	なし	$\forall d$ で $x_d=420.968..$
⑤Griewank	多峰性	あり	$\forall d$ で $x_d=0$

た。そこで、変更確率  $P$  の値を  $2/D$  とした。

### 3.2 ベンチマーク関数

本研究では、表2の①～⑤の5つのベンチマーク関数で数値実験を行った。各関数の形状・変数間の依存関係の有無および解を表2にまとめ、関数式を式(4)～(8)に示す。なお、全ての関数の最小値は0である。

#### ①Sphere 関数

$$f(x) = \sum_{d=1}^D x_d^2 \quad (-5.12 \leq x_d \leq 5.12) \quad (4)$$

#### ②Rosenbrock 関数

$$f(x) = \sum_{d=1}^{D-1} (100(x_{d+1} - x_d^2)^2 + (1 - x_d)^2) \quad (-2.048 \leq x_d \leq 2.048) \quad (5)$$

#### ③Rastrigin 関数

$$f(x) = \sum_{d=1}^D (x_d^2 - 10 \cos(2\pi x_d) + 10) \quad (-5.12 \leq x_d \leq 5.12) \quad (6)$$

#### ④Schwefel 関数

$$f(x) = 418.9829 \times D + \sum_{d=1}^D -x_d \sin(\sqrt{|x_d|}) \quad (-512 \leq x_d \leq 512) \quad (7)$$

#### ⑤Griewank 関数

$$f(x) = 1 + \sum_{d=1}^D \frac{x_d^2}{4000} - \prod_{d=1}^D \left( \cos\left(\frac{x_d}{\sqrt{d}}\right) \right) \quad (-512 \leq x_d \leq 512) \quad (8)$$

## 4. 数値実験結果と考察

### 4.1 提案手法と DMS-PSO との大域探索能力比較

提案手法と DMS-PSO との大域探索能力の比較を行うため、 $D=10$  の Griewank 関数の最小値を探索し、各世代での全粒子・全成分の平均位置  $\mu$  と標準偏差  $\sigma$  を求め、 $\mu - \sigma$ 、 $\mu$ 、 $\mu + \sigma$  の世代推移を図2-4に示す。まず、DMS-PSOで求めた  $\mu - \sigma$ 、 $\mu$ 、 $\mu + \sigma$  の世代推移を図2に示す。図より  $\mu$  は約 1.07 の値を取り世代とともに変化がなく、 $\sigma$  にも変化がないことから、DMS-PSO では初期収束が起こっていることが分かる。次に、図3、4にそれぞれ停滞回避 DMS-PSO と停滞回避 MDMS-PSO で求めた  $\mu - \sigma$ 、 $\mu$ 、 $\mu + \sigma$  の世代推移を示す。図3、4では  $\mu$  の微小な変動とその

周りでの  $\sigma$  の変化が見られる。このことから提案手法では再配置処理が行われた後、局所解から抜け出し、大域探索が継続されていると推測できる。また図3、4の提案手法の停滞回避 DMS-PSO と停滞回避 MDMS-PSO を比べたとき、探索の序盤における  $\sigma$  の大きさが異なることが分かる。これから探索の序盤において停滞回避 MDMS-PSO は停滞回

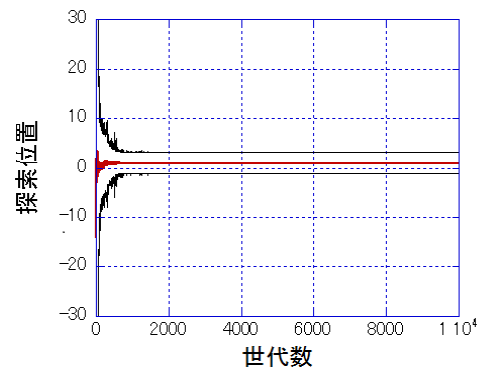


図2 粒子の平均位置と標準偏差の推移  
 (Griewank 関数,  $D=10$ , DMS-PSO)

Figure 2 Time course of both  $\mu$  and its  $\sigma$   
 (Griewank Function,  $D=10$ , DMS-PSO)

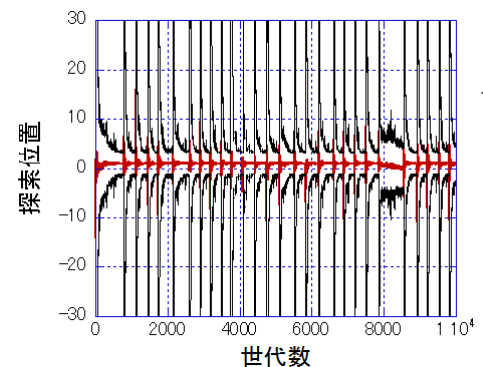


図3 粒子の平均位置と標準偏差の推移  
 (Griewank 関数,  $D=10$ , 停滞回避 DMS-PSO)

Figure 3 Time course of both  $\mu$  and its  $\sigma$   
 (Griewank Function,  $D=10$ , SA-DMS-PSO)

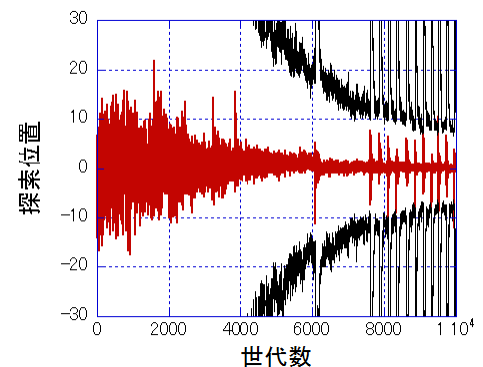


図4 粒子の平均位置と標準偏差の推移  
 (Griewank 関数,  $D=10$ , 停滞回避 MDMS-PSO)

Figure 4 Time course of both  $\mu$  and its  $\sigma$   
 (Griewank Function,  $D=10$ , SA-MDMS-PSO)

避 DMS-PSO よりも大域探索能力に優れていることが分かる。

#### 4.2 既存手法との探索性能比較

5つのベンチマーク関数での数値実験から得た  $\langle f(\text{gbest}) \rangle$  の世代推移を図5~9に示す。本研究では  $D=10$ ,  $30$  での数値実験を行ったが, Griewank 関数では  $D=10$  の結果を, 他の関数では  $D=30$  の結果を示す。多くの関数では次元数が大きいほど問題が難しくなる傾向があるが, Griewank 関数では次元数が小さいほど問題が難しくなるので Griewank 関数のみ  $D=10$  の結果を示した。

Sphere 関数での  $\langle f(\text{gbest}) \rangle$  の世代推移を図5に示す。15000世代までに全ての手法で最適解に達しているように見えるが, StPSO, CLPSO では最適解に達していない。停滞回避 MDMS-PSO では最適解に達しているが, 達するまでに最も長い世代を要した。このように StPSO, CLPSO および停滞回避 MDMS-PSO で探索性能が悪かったのは, これらの手法では  $w$  の値を線形減衰させており, 序盤で探索の多様性が高すぎたためと考えられる。

Rosenbrock 関数での  $\langle f(\text{gbest}) \rangle$  の世代推移を図6に示す。図6では, DMS-PSO の探索性能が最も良く, 次に停滞回避 DMS-PSO が良い。Rosenbrock 関数は単峰性関数であるため, 強い局所探索能力が必要であると考えられる。一方, この関数は変数間に強い依存関係を持っており, 最小値付近には関数値の小さい領域が双曲線状に大きく広がっているため, 探索の序盤では大域探索能力も必要であると考えられる。そのため強い局所探索能力を持っているが大域的な探索能力に欠ける PSO の解探索性能が低かったと考えられる。一方, 二つの能力をある程度バランスよく持っている DMS-PSO の探索性能が良かったと考えられる。さらに, DMS-PSO に停滞回避アルゴリズムを追加した停滞回避 DMS-PSO では, 停滞回避アルゴリズムによる再配置処理が無駄になり, 結果として DMS-PSO より探索性能が悪かったと考えられる。

Rastrigin 関数での  $\langle f(\text{gbest}) \rangle$  の世代推移を図7に示す。図7では, 提案手法の二手法と CLPSO で最適解に到達している。また, 図8に Schwefel 関数での  $\langle f(\text{gbest}) \rangle$  の世代推移を示す。図8でも提案手法の二手法と CLPSO の解探索性能が良いことが分かる。この二つの関数のように多峰性で変数間に依存関係がない関数において提案手法と CLPSO が高い探索性能を示していることが分かる。それに比べ, 提案手法の基となっている DMS-PSO は局所解から抜け出すことができていない。従って, 初期収束を回避する停滞回避アルゴリズムが多峰性関数で有効に働いていると推測する。以上より, 提案手法と CLPSO は変数間に依存関係のない多峰性関数において優れた大域探索能力を持つことが分かる。

Griewank 関数での  $\langle f(\text{gbest}) \rangle$  の世代推移を図9に示す。図9では CLPSO の探索性能が最も良いが, 次に停滞回避

MDMS-PSO が良い。Griewank 関数のような変数間に依存関係のある多峰性関数では, 停滞回避 DMS-PSO よりも停滞回避 MDMS-PSO の方が高い探索性能を発揮していることが分かる。また, 変数間に依存関係のある多峰性関数でも, 提案手法と CLPSO の探索性能が高いことが分かる。

手法ごとに特徴をまとめると, PSO は多くの関数で初期収束している。StPSO は探索の序盤から中盤の間で, 探索が停滞してしまう傾向が見られる。CLPSO は多峰性関数では探索性能が高いが, 単峰性関数である Rosenbrock 関数では探索の中盤で, 探索が停滞している。このことから CLPSO は大域探索能力が高いが, 局所探索能力は低いと推測できる。提案手法は, 単峰性・多峰性および変数間の依存関係の有無によらず, 全ての関数において最適解近くを探索できていることから, 大域探索能力と局所探索能力の双方の能力をうまく発揮できていることが分かる。

#### 5. まとめと今後の課題

本研究では, 既存手法である DMS-PSO を基に停滞回避アルゴリズムを追加し, 大域および局所探索能力を適応的に取り込んだ停滞回避 PSO を提案した。数値実験による結果が示す通り, 本稿で挙げた4つの既存手法と比較しても提案手法では多くの関数で良好な解探索性能を得ていることから, 単峰性・多峰性および変数間の依存関係の有無によらず様々なタイプの関数に適用できる汎用的な PSO であることが分かった。その原因として, 提案手法では大域探索能力と局所探索能力を発揮できる二種類の粒子を共存させることにより, 双方の能力をアルゴリズム上で並列に行えるように実装している点が挙げられる。

今後の課題として, 以下の二つが挙げられる。まず, 提案手法はパラメータの種類が多いため, パラメータの値の決定が難しい。特に停滞判定条件 (I) のカウンタ変数  $\text{count}_V$  の加算条件や, 再配置処理での  $\text{pbest}$  の変更確率  $P$  の値は, 関数のタイプによって適切な値が変動する。そこで, 様々な関数でより高い探索能力を発揮できるパラメータ値の汎用的な選択方法を検討する。また, APSO など, 今回の数値実験で比較した既存手法以外の手法との性能比較を行うことである。

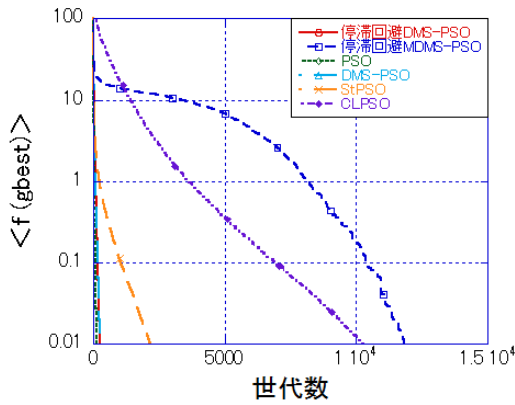


図5  $\langle f(\text{gbest}) \rangle$ の推移  
(Sphere関数,  $D=30$ )

Figure 5 Time course of  $\langle f(\text{gbest}) \rangle$   
(Sphere function,  $D=30$ )

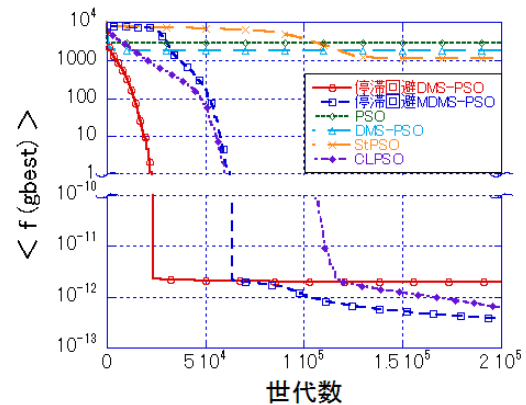


図8 gbest関数値の加算平均の推移  
(Schwefel関数,  $D=30$ )

Figure 8 Time course of  $\langle f(\text{gbest}) \rangle$   
(Schwefel function,  $D=30$ )

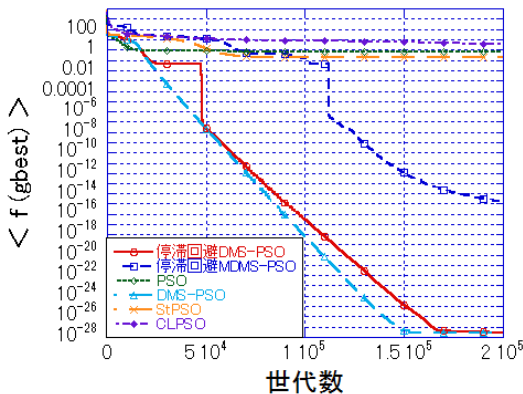


図6 gbest関数値の加算平均の推移  
(Rosenbrock関数,  $D=30$ )

Figure 6 Time course of  $\langle f(\text{gbest}) \rangle$   
(Rosenbrock function,  $D=30$ )

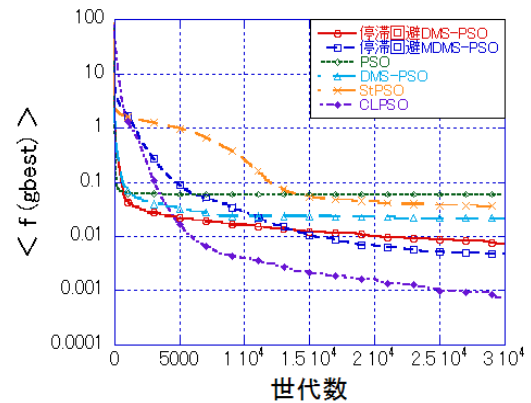


図9 gbest関数値の加算平均の推移  
(Griewank関数,  $D=10$ )

Figure 9 Time course of  $\langle f(\text{gbest}) \rangle$   
(Griewank function,  $D=10$ )

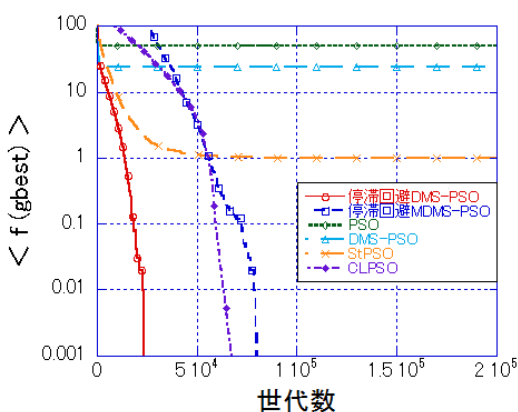


図7 gbest関数値の加算平均の推移  
(Rastrigin関数,  $D=30$ )

Figure 7 Time course of  $\langle f(\text{gbest}) \rangle$   
(Rastrigin function,  $D=30$ )

## 参考文献

- 1) Kennedy, J. and Eberhart, R. C.: Particle swarm optimization, Proc. of IEEE the International Conference on Neural Networks, Vol. 4, pp.1942-1948 (1995).
- 2) Eberhart, R. C. and Kennedy, J.: A new optimizer using particle swarm theory, Proc. of the Sixth Int. Symposium on Micromachine and Human Science, Nagoya, Japan, pp.39-43 (1995).
- 3) Liang, J. J., Qin A. K., Suganthan P. N., and Baskar S.: Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions, IEEE Transactions on Evolutionary Computation, Vol.10, No.3, pp.281-295 (2006).
- 4) Liang, J. J. and Suganthan, P. N.: Dynamical multiswarm particle swarm optimizer, Proc. of the IEEE swarm intelligence symposium (SIS), pp.124-129 (2005).
- 5) Sevkli, A. Z. and Sevilgen F. E.: StPSO:Strngthened particle swarm optimization, Turkish Journal Of Electrical Engineering & Computer Sciences, Vol.18, No.3, pp.1095-1114 (2010).
- 6) Zhan Z. H., Zhang J., Li Y., Chung H.S.: Adaptive particle swarm optimization, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol.39, No.6, pp.1362-1381 (2009).
- 7) Yasuda K. and Iwasaki N.: "Adaptive Particle Swarm Optimization using Velocity Information of Swarm", Proc. of IEEE the International Conference on Systems, Man & Cybernetics, pp.3475-3481 (2004).
- 8) Shi, Y. and Eberhart, R. C.: A modified particle swarm optimizer, Proc. of the IEEE international conference on evolutionary computation (CEC), pp.69-73 (1998).