

負の慣性項を有する粒子群最適化の基本性能について

渡邊 恭成¹ 中野 秀洋¹ 宮内 新¹

概要：近年、複雑な問題を効率よく解くための手法として粒子群最適化 (PSO) が注目されている。PSO は生物の社会的行動を模倣した最適化手法であり、群れで情報を共有しながら探索を行う。PSO は位置の更新の際に群れの最良の位置情報、各粒子の最良の位置情報、現在の速度に対する慣性の3つの項を用いる。これら3つの項は粒子の動きを決定し、解探索性能に大きな影響を与える。3つの項の適切な調整を行うための従来研究は多く存在するが、いずれもこれらの項のパラメータが正の値をもつことを前提としている。そこで本稿では、慣性項のパラメータを負の値に設定した際の解探索性能について検証を行う。

1. はじめに

近年、システムの複雑化、大規模化に伴い、これらに対する最適化問題において現実的な時間内に最適解を求めることが困難となってきている。そこで、最適解を得られることは保証されないが現実的な時間内で良好な解を求めることができるメタヒューリスティクスが注目されている。代表的なメタヒューリスティクスとしては、遺伝的アルゴリズム (GA: Genetic Algorithm)、差分進化法 (DE: Differential Evolution[10])、免疫アルゴリズム (IA: Immune Algorithm)、粒子群最適化 (PSO: Particle Swarm Optimization[1])、人工蜂コロニー最適化 (ABC: Artificial Bee Colony[11]) などが挙げられる。

そのメタヒューリスティクスの中でも注目され活発に研究されている手法の一つが PSO である [1]-[9]。PSO は設計が容易であり、複雑な計算を必要としないため多くの最適化問題へ適用することができる。しかし、PSO は問題空間の高次元化に伴い解探索性能が極端に劣化することが指摘されている。そこで本稿では、複雑な改良を行わずに解探索性能を向上させる簡素な手法について検討する。

2. 粒子群最適化 (PSO)

PSO は粒子が互いの情報を共有しながら多次元空間内を探索する最適化アルゴリズムである [1]。粒子全体の過去の探索結果の中で最も良い評価値を持つ位置情報を $gbest$ と呼び、各粒子の過去の探索結果の中で最も良い評価値を持つ位置情報を $pbest$ と呼ぶ。

各粒子は現在の速度 v_i^t と群れの最良位置である $gbest^t$ 、

各粒子の最良位置である $pbest_i^t$ を用いて速度を決定し位置を更新する。各粒子の速度更新式と位置更新式は式 (1), (2) で表される。

$$v_{ij}^{t+1} = W \cdot v_{ij}^t + C_1 \cdot R_1 \cdot (pbest_{ij}^t - x_{ij}^t) + C_2 \cdot R_2 \cdot (gbest_j^t - x_{ij}^t) \quad (1)$$

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

ただし、 i は粒子番号、 j は次元成分番号、 t は探索更新回数、 R_1, R_2 は 0 から 1 の範囲の一樣乱数である。また、 W は慣性定数、 C_1, C_2 はそれぞれ学習係数と呼ばれる。右辺第一項は慣性項と呼ばれ、粒子の現在の方向に継続して向かう力を決定する。本稿では特にこの慣性項に注目する。また、第二項は $pbest$ へと向かう力を、第三項は $gbest$ へと向かう力をそれぞれ決定する。粒子の動きは慣性定数 W 、学習係数 C_1, C_2 に応じて決定され、これらの設定値が解探索性能に大きな影響を与える。

3. 負の慣性項を有する PSO の基本性能

前節で述べたように、PSO の粒子の動きは3つの項で決定される。これら3つの項に関する研究は多く、代表的なものとして安定性解析に基づきパラメータを設定する手法 [2]、イテレーションの増加に伴い W の値を線形に減少させる手法 [3]、粒子の分布情報に基づいて C_1, C_2 を線形に変化させる手法 [4]、 W の値を乱数を用いて変化させる手法 [5] などが挙げられる。

いずれの手法も3つの項を調整することで解探索性能の向上を図っているが、パラメータが正の値をもつことを前提としている。そこで本稿では、慣性定数を負の値に設定

¹ 東京都市大学
Tokyo City University

表 1 ベンチマーク関数
Table 1 Benchmark functions

関数	探索領域	最適値	関数名
$f_1(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	[-5.12, 5.12]	0	Rastrigin
$f_2(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	[-512, 512]	$-418.98288727 \times D$	Schwefel
$f_3(x) = 20 - 20 \exp\left(-0.2\sqrt{1/n \sum_{i=1}^n x_i^2}\right) + e - \exp(1/\sum_{i=1}^n \cos(2\pi x_i))$	[-32, 32]	0	Ackley
$f_4(x) = 1 + \sum_{i=1}^n x_i^2/4000 - \prod_{i=1}^n (\cos(x_i/\sqrt{i}))$	[-512, 512]	0	Griewank

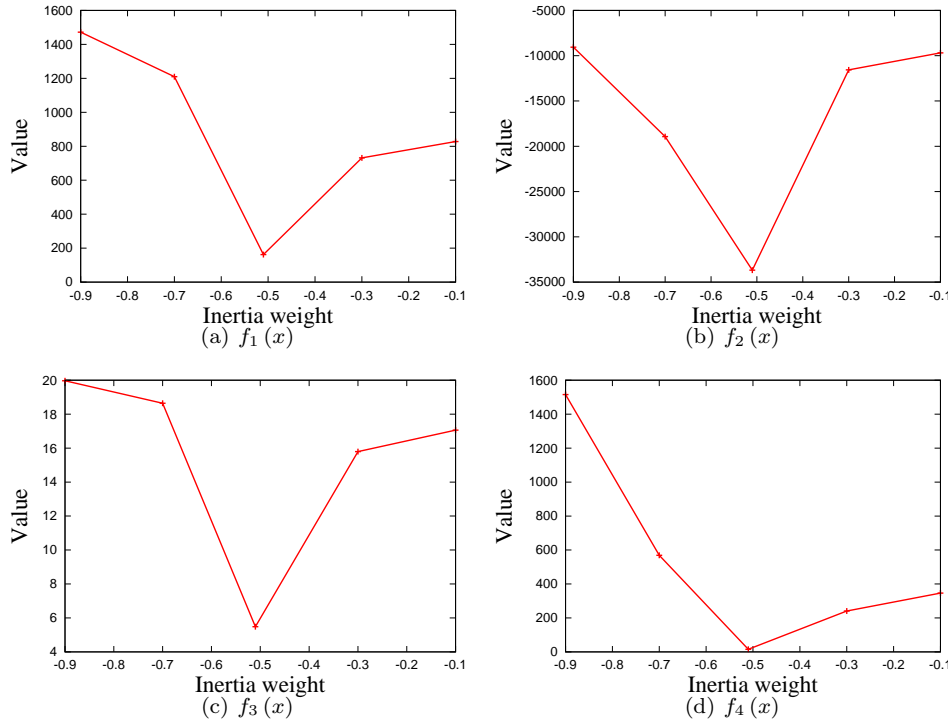


図 1 慣性定数 W に対する解探索性能
Fig. 1 Solving performances for inertia weight W

表 2 負の慣性項を有する PSO の基本性能

Table 2 Basic performances of a particle swarm optimizer with a negative inertia term

関数	次元数		$W = -0.51$	$W = 0.9$
$f_1(x)$	10	平均値	3.1×10^{-1}	9.0×10^{-1}
		最良値	0.0	0.0
	100	平均値	1.6×10^2	6.6×10^2
		最良値	9.3×10^1	5.0×10^2
$f_2(x)$	10	平均値	-3.6×10^3	-3.6×10^3
		最良値	-4.0×10^3	-4.0×10^3
	100	平均値	-3.4×10^4	-2.7×10^4
		最良値	-3.6×10^4	-3.4×10^4
$f_3(x)$	10	平均値	3.6×10^{-15}	3.6×10^{-15}
		最良値	3.6×10^{-15}	3.6×10^{-15}
	100	平均値	5.5×10^0	2.0×10^1
		最良値	3.1×10^0	1.8×10^1
$f_4(x)$	10	平均値	5.6×10^{-2}	4.9×10^{-2}
		最良値	9.9×10^{-3}	5.6×10^{-2}
	100	平均値	1.5×10^1	3.0×10^2
		最良値	1.3×10^0	6.6×10^1

し、粒子を振動させながら探索させた際の解探索性能について検証を行う。

粒子数 $N = 100$, 次元数 $D = 10, 100$, イテレーション $T_{max} = 5000$, 学習係数 $C_1 = C_2 = 1.0$, 慣性定数 $W = -0.51, 0.9$, 実行回数を 100 回とし、表 1 の関数を用いて比較実験を行った。実験結果を表 2 に示す。なお、表における平均値は 100 回の探索終了時の評価値を平均した値であり、最良値は 100 回の探索終了時の評価値で最も良い値である。また $W = -0.51, W = 0.9$ の結果のうち、良い方を太字で示している。多くの関数において、慣性定数を負の値に設定した方が良い結果となっていることが確認できる。特に高次元問題ではその差が顕著に表れていることが確認できる。そこで、次節では適切な慣性定数の値と、高次元問題における解探索性能向上の要因を検証する。

4. 負の慣性項を有する PSO の性能解析

4.1 慣性定数 W の設定値

本節では慣性項を変化させた際の解探索性能の違いを

表 3 慣性定数 W に対する解探索性能
Table 3 Solving performances for inertia weight W

関数		-0.48	-0.49	-0.50	-0.51	-0.52
$f_1(x)$	平均値	4.1×10^2	3.6×10^2	2.6×10^2	1.6×10^2	3.1×10^2
	最良値	3.3×10^2	2.6×10^2	1.8×10^2	9.3×10^1	1.4×10^2
$f_2(x)$	平均値	-1.9×10^4	-2.0×10^4	-2.4×10^4	-3.4×10^4	-3.0×10^4
	最良値	-2.3×10^4	-2.5×10^4	-3.2×10^4	-3.6×10^4	-3.5×10^4
$f_3(x)$	平均値	1.4×10^1	1.3×10^1	1.0×10^1	5.5×10^0	1.3×10^1
	最良値	1.3×10^1	1.2×10^1	7.5×10^0	3.1×10^0	1.0×10^1
$f_4(x)$	平均値	1.0×10^2	8.3×10^1	5.4×10^1	1.5×10^1	2.4×10^1
	最良値	6.7×10^1	5.6×10^1	2.3×10^1	1.3×10^0	4.2×10^0

検証する。粒子数 $N = 100$, 次元数 $D = 100$, イテレーション $T_{max} = 5000$, 学習係数 $C_1 = C_2 = 1.0$, 実行回数を 100 回とし, 前節で使用した慣性定数 $W = -0.51$ に加え, $W = -0.1, -0.3, -0.7, -0.9$ の場合について調べる。実験結果を図 1 に示す。横軸は慣性定数, 縦軸は 100 回の探索終了時の評価値を平均した値である。全ての関数において慣性定数 $W = -0.51$ のときに最も良い結果となっている。

そこで, $W = -0.51$ 付近の値についてさらに実験を行った。慣性定数を $W = -0.48, -0.49, -0.50, -0.51, -0.52$ とする。また, その他のパラメータ値に関してはこれまでと同じ値を使用する。実験結果を表 3 に示す。全ての関数において慣性定数 $W = -0.51$ のときが平均値, 最良値ともに良い結果となっている。また, その他の慣性定数の結果に注目すると $W = -0.51$ との差は大きく, わずかな値の違いが解探索性能に大きな影響を与えることが見て取れる。よって慣性定数を負の値に設定した場合, 精密なパラメータの調整が必要となる可能性がある。

4.2 平均速度の分布

慣性定数を負の値に設定することで高次元問題における解探索性能が向上することを 3 節で確認した。そこで, 慣性定数を 4.1 節の実験で最も良い結果となった $W = -0.51$ とした際の粒子の速度を観測し, 解探索性能向上の要因を探る。

図 2 はイテレーション $T_{max} = 1000$ における, 代表的な 3 つの粒子の各次元成分の平均速度を求め, 降順にソートした結果を示している。 $W = -0.51$ のときは大きな速度を持つ次元成分が少ないことが見て取れる。高い探索性能を持つ代表的なメタヒューリスティクスとして DE[10], ABC[11] などが挙げられるが, いずれの手法も一部の次元の位置を更新しながら解探索を行う傾向がある。しかし PSO は基本的に全ての次元の位置を一斉に更新するため, 問題が高次元になるほど局所解に早期に収束し, 解探索性能が極端に劣化することが考えられる。一方, 図 2 を見てもわかるように, $W = -0.51$ とした場合は大きく位置を更新する次元成分がさほど多くない。これによって一部の次

元を集中的に探索することで, 高次元問題においても比較的良い解を求めることができたのではないかと考えられる。

5. 異なる慣性定数の粒子が混在した PSO

慣性定数 $W = -0.51$ とした場合は高次元問題において有効であるが, 速度が早期に減衰しやすいため大域的な探索が行われていない可能性がある。そこで $W = -0.51$ とする粒子と, $W = 0.9$ とする粒子を混在させ, 探索を行った際の解探索性能について検証をする。

総粒子数 $N = 100$, 次元数 $D = 100$, 学習係数 $C_1 = C_2 = 1.0$, 慣性定数 $W = -0.51, 0.9$, 実行回数を 100 回とする。また, 解の収束値を確認するためにイテレーション $T_{max} = 30000$ とし, 負の慣性項を有する粒子の割合を変化させて実験を行った。

実験結果を表 4 に示し, 最も良い結果を太字で表している。負の慣性項をもつ粒子だけでなく, 正の慣性項をもつ粒子を同時に使用することで解探索性能が向上し, $f_4(x)$ では最適解を求めることができた。負の慣性項を有する粒子の数に関しては, 粒子全体の 3 割程度に設定した場合が全体的に最も良い結果となった。 $f_1(x), f_2(x)$ では大きな差は見られないが, $f_3(x), f_4(x)$ は負の慣性項のみの PSO より解探索性能が大きく向上していることが確認できる。 $f_3(x)$ は局所解が多く存在する関数のため, 速度の減衰が大きい $W = -0.51$ だけでは良い解を求めることが困難である。また, $f_4(x)$ は設計変数間に依存関係をもつため, 大きく位置の更新を行う次元成分が少ない $W = -0.51$ の場合は良い解を求めることが困難である。

次に解への収束速度について調べる。図 3 に実験結果を示す。この図の横軸はイテレーション, 縦軸は評価値, 凡例は負の慣性項をもつ粒子の割合を表している。 $f_1(x)$ を除いて, 異なる慣性項の粒子が混在した PSO の収束速度が向上していることがわかる。解精度, 収束速度の観点から, 異なる慣性項の粒子が混在した PSO は負の慣性項のみの PSO と比較し, 多くの問題において有効であると言える。

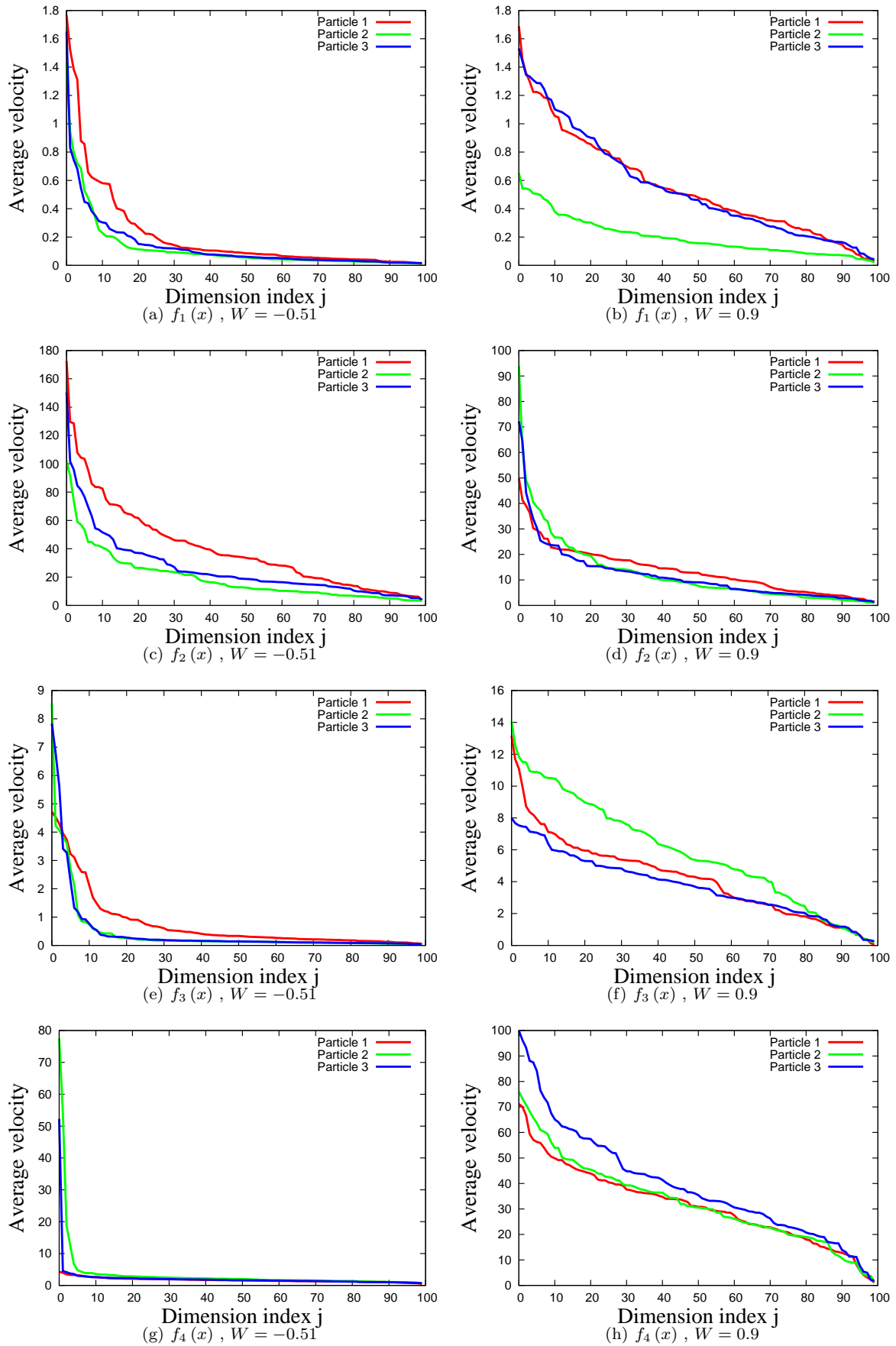


図 2 各関数における次元成分ごとの平均速度の分布

Fig. 2 Distribution of average velocity for each dimension component in each function

表 4 負の慣性項をもつ粒子の割合に対する解探索性能

Table 4 Solving performances for mixing ratio of the particles with negative inertia term

関数		0%	30%	50%	70%	100%
$f_1(x)$	平均値	5.0×10^2	1.3×10^2	1.2×10^2	1.3×10^2	1.4×10^2
	最良値	2.3×10^2	5.3×10^1	6.0×10^1	7.5×10^1	5.0×10^1
$f_2(x)$	平均値	-2.7×10^4	-3.6×10^4	-3.6×10^4	-3.6×10^4	-3.6×10^4
	最良値	-3.4×10^4	-3.7×10^4	-3.7×10^4	-3.8×10^4	-3.8×10^4
$f_3(x)$	平均値	1.9×10^1	4.1×10^{-8}	8.1×10^{-7}	8.3×10^{-6}	3.1×10^{-1}
	最良値	1.4×10^1	3.1×10^{-13}	6.4×10^{-13}	7.7×10^{-10}	2.4×10^{-5}
$f_4(x)$	平均値	3.0×10^2	5.7×10^{-2}	7.0×10^{-2}	8.6×10^{-2}	4.9×10^0
	最良値	6.6×10^1	0.0	0.0	0.0	2.0×10^{-4}

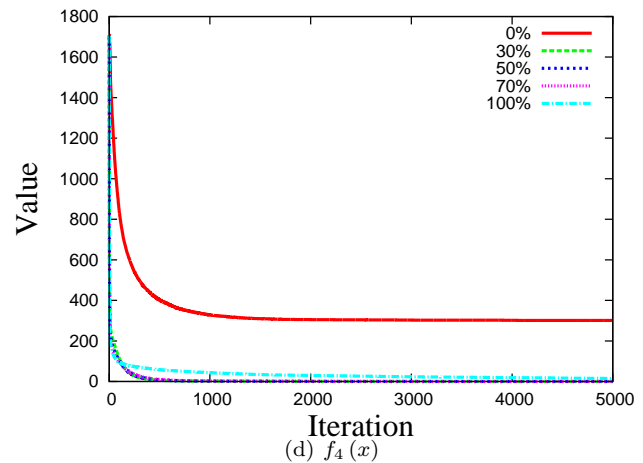
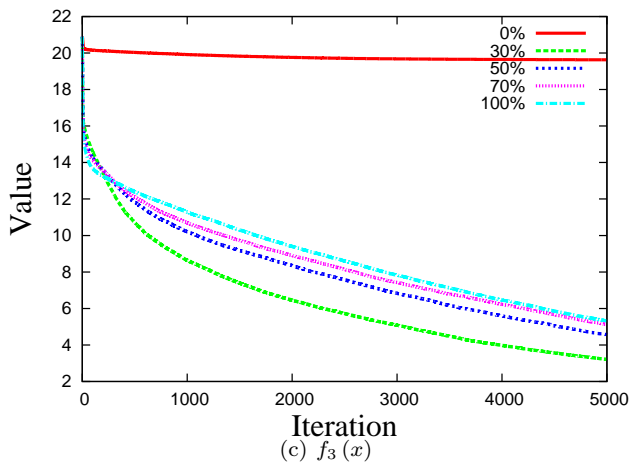
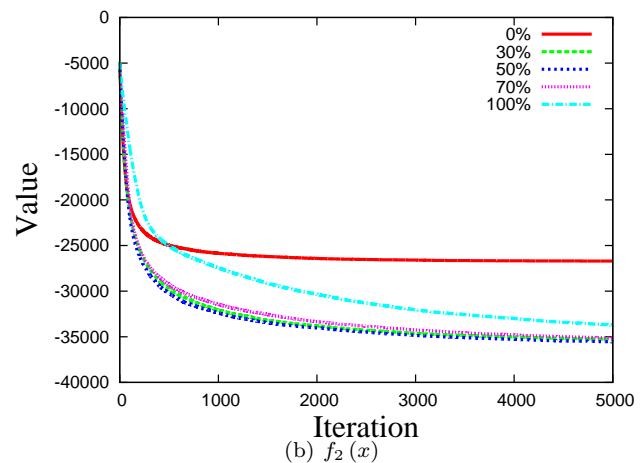
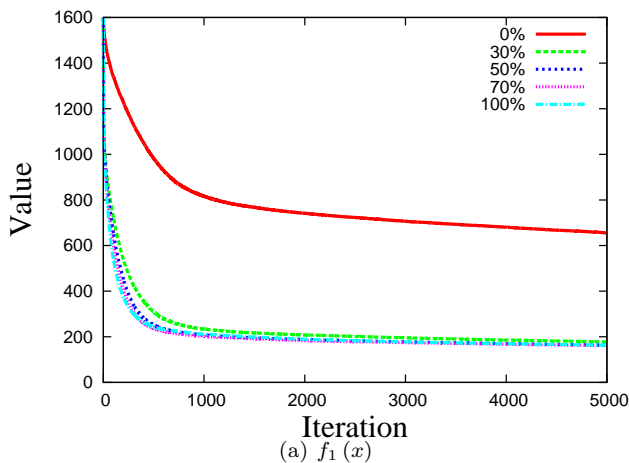


図 3 負の慣性項をもつ粒子の割合に対する収束速度

Fig. 3 Convergence speed for mixing ratio of the particles with negative inertial term

6. おわりに

PSOのパラメータの一つである慣性項に注目し、慣性定数を負の値に設定した際の解探索性能について検証を行った。負の慣性項は精密なパラメータ設定が必要ではあるが、アルゴリズムの複雑な改良を必要とせず、高次元問題における高い解探索性能を実験により確認した。また、異なる

慣性項を混在させた PSO の提案を行い、解探索性能と収束速度の向上を確認した。今後の課題としては、問題に応じた適切なパラメータ調整、より複雑な問題での実験などが挙げられる。

参考文献

- [1] J. Kennedy, and R. Eberhart,: Particle Swarm Optimization, *Proc of IEEE International Conference on Neural Network*, pp. 1942–1948 (1995).
- [2] Y. Shi, : Particle Swarm Optimization , *IEEE Connections*, (2004).
- [3] J. Kennedy, and R. Eberhart, : Swarm Interlligence ,*Morgan Kaufmann* (2001).
- [4] Z. Zhan, J. Zhang, Y. Li, H. S. Chung, : Adaptive Particle Swarm Optimization, *IEEE Transactions on Systems, Man, and Cybernetics.Part B: Cybernetics*. Vol. 39, No. 6, pp. 1362–1381 (2009)
- [5] R. Eberhart, and Y. Shi,: Tracking and optimizing dynamic systems with particle swarms, *Proc of IEEE Congr. Evolutionary Computation 2001*, pp. 94–97 (2001).
- [6] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson,: Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleratioin Coefficients, *IEEE Transaction on Evolutionary Computation*, Vol. 8, NO. 3, pp. 240–256 (2004).
- [7] N. Iwasaki, and K. Yasuda,: Adaptive Particle Swarm Optimization via Velocity Feedback, *電学論 C*, 125 巻, 6 号, pp. 987–988 (2005).
- [8] 石亀 篤司, 安田 恵一郎.: 群れの知能:Particle Swarm Optimization, 知能と情報 (日本知能情報フアジイ学会誌), Vol. 20, NO. 6, pp. 829–839 (2008).
- [9] 斉藤 利通,: 粒子群最適化と非線形システム, *IEICE Fundamentals Review*, Vol. 5, NO. 2, pp. 155–161 (2011).
- [10] R. Storn, and K. Price: Differential Evolution - A simple and effient adaptive scheme for global optimization over continuous spaces, *Technical Report TR-95-012, International Computer Science Institute*, (1995).
- [11] D. Karaboga,: An idea based on honey bee swarm for numerical optimization, *Technical Report-TR06, Erciyes University, Engineering Faculty Computer Engineering Department*, (2005).