

高次元関数最適化のための Particle Swarm に関する一考察

是永 剛志 畠中 利治

大阪大学大学院情報科学研究科情報数理学専攻

Particle Swarm Optimization(PSO) は鳥や魚の群の振る舞いに範を得た最適化法で、複数の粒子 (particle) による多点の確率的探索アルゴリズムである。探索性能を改善することを目的に particle 動作モデルの係数の決定方式などが研究されており、さまざまな最適化問題に適用され、優れた結果が報告されている。しかしながら、高次元の関数最適化に適用した際には、粒子の速度ベクトルが縮退し、十分な探索を行えないことがある。そこで、本研究では、高次元空間における探索能力の向上を目的に、座標変換を導入した新たな PSO の計算モデルを提案し、いくつかのベンチマーク関数へ適用し提案手法の有用性を示す。

Improvement of Particle Swarm for High-Dimensional Function Optimization

Takeshi Korenaga and Toshiharu Hatanaka

Department of Information and Physical Sciences,

Graduate School of Information Science and Technology, Osaka University

Particle Swarm Optimization (PSO) is a population-based probabilistic search methodology inspired by social behavior, such as flocks of birds and schools of fish. In many studies, PSO has been successful in a variety of optimization problems. However, due to degeneracy that is early convergence in the certain dimensions, the PSO algorithm does not work well for the high-dimensional function. The purpose of this paper is improvement of PSO performance in case of high-dimensional function optimization problems. A novel PSO model is proposed in this paper, by introducing the coordinate conversion. The numerical simulation results show the proposed method is effective in optimizing high-dimensional functions.

1 はじめに

Particle Swarm Optimization(PSO) は、1995 年に Kennedy と Eberhart によって紹介された、鳥や魚の群れの振る舞いを関数最適化に応用した手法である [1]。解の候補である粒子 (particle) が群れ (swarm) を形成するように相互作用することによって、解の探索能力を高めている。PSO は比較的実装が容易でありながら、様々な最適化問題においてその有用性が報告されている。

本研究の目的は、PSO の高次元関数に対する探索能力の向上にある。PSO では、目的関数の次元数が大きくなると、最適解や局所解以外の場所で、解ベクトルの中のいくつかの変数に関する探索が停滞してしまい、良い解が得られないということがある。このような停滞を防いで探索効率を向上させるために、速度ベクトルの更新に座標変換を導入した新たな PSO のモデルを提案する。また、いくつかのベンチマーク関数に提案手法を適用し、その有用性を示す。

2 Particle Swarm Optimization

PSO は集団ベースの最適化法である。位置および速度を持った粒子の集団が最適解を探して、問題空

間上を飛び回る。

各粒子はそれぞれ、“ p ”(パーソナルベスト) および “ g ”(グローバルベスト) の 2 つの位置情報を記憶する。 p はそれまでに見つけた自己最良の解である。 g は粒子同士のつながりを表わすトポロジーによって定義された近傍に位置する粒子の p の中で最良のものである。これらの解の記憶を参照して、速度更新式

$$v_d^{(i)} = \chi(v_d^{(i)} + \phi_{1,d}r_{1,d}(p_d^{(i)} - x_d^{(i)}) + \phi_{1,d}r_{2,d}(g_d^{(i)} - x_d^{(i)})) \quad (1)$$

および、位置更新式

$$x_{id} = x_{id} + v_{id}. \quad (2)$$

によって、粒子の速度と位置を更新する。ここで、 i , ($i = 1, 2, \dots, N$) は粒子の番号を、 d , ($d = 1, 2, \dots, D$) は次元の指標を表わす。 N と D はそれぞれ集団を構成する粒子の個数と、粒子の存在する空間の次元を表わす。また、制限係数 χ は速度の爆発を防ぎ、大域的探索と局所探索の間のバランスを取る定数で、 $r_{1,d}, r_{2,d}$ は $[0, 1]$ を範囲とする一様乱数である。 $\chi = 0.7298, \phi_{1,d} = \phi_{2,d} = 2.05$ の値が推奨されている [2]。

このように PSO アルゴリズムは現在の位置の評価、記憶の更新、速度と位置の更新を繰り返すことで進行する。

3 高次元関数における初期収束

アルゴリズムを単純化 (eg, 単一粒子, 次元数 1, 確率的要素の排除) することによって, PSO の粒子の軌跡に対する様々な理論的解析が試みられた [2] [3] [4]. それらの研究によると, 各粒子は p と g の重み付き平均の周りを減衰振動し, 新たな p や g が見つからないと, やがて速度を失い収束してしまう. ここで問題になるのが, 粒子が収束する p と g の重み付き平均は, 大域的最適解や局所最適解だとは限らないことである. 特に対象とする関数が高次元である場合, 集団全体で同一座標軸方向の速度が失われやすく, 探索性能が大きく損なわれる. 例として, PSO を単峰性の Sphere 関数に適用した結果を紹介する. Sphere 関数は

$$f = \sum_{d=1}^D x_d^2, \quad (3)$$

で定義される. 集団のサイズを $N = 20$ とし全ての粒子間につながりが存在する Star トポロジーを用いた. 20 回の実験を行い, ステップ数に対する最良解の評価値の平均を図 1, 2 に示す. この結果から $D = 10$ の場合に比べ, 高次元の $D = 100$ の場合には評価値の改善に滞りが見られることがわかる.

上記の結果のように高次元関数に PSO を適用す

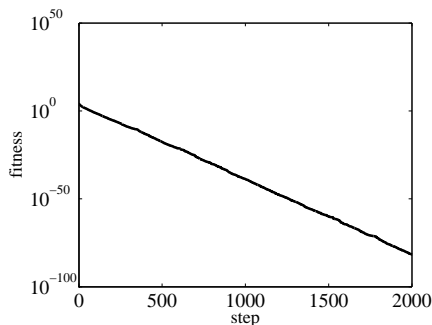


図 1 評価値 $D = 10$

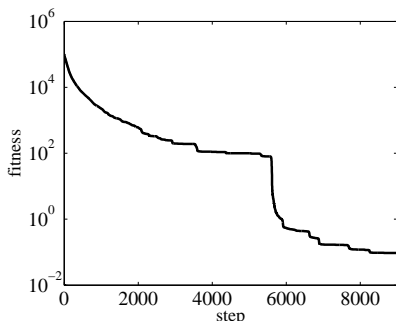


図 2 評価値 $D = 100$

ると, 局所解に捕われることなく, 各粒子の同一座標軸方向の速度を極めて小さくしてしまふことで, 探索効率が劣化することがしばしば起こる. これは, (1) 式において, d 番目の速度成分 $v_{i,d}$ が位置や記憶の d 番目の成分のみで決定され, 一度ある次元の位置や記憶の多様性が集団から失われると, その方向の探索の再開は困難であることが原因だと考えられる.

次節では, 他の次元の情報を利用する新たな速度更新法を提案する.

4 Roatated particle swarm

従来速度更新法を再考するために, (1), (2) 式をベクトルと行列で以下のように書き換える.

$$\mathbf{v}_i = \chi(\mathbf{v}_i + \Phi_1(\mathbf{p}_i - \mathbf{x}_i) + \Phi_2(\mathbf{g}_i - \mathbf{x}_i)), \quad (4)$$

$$\Phi_1 = \text{diag}(\phi_{1,1}r_{1,1}, \phi_{1,2}r_{1,2}, \dots, \phi_{1,D}r_{1,D}), \quad (5)$$

$$\Phi_2 = \text{diag}(\phi_{2,1}r_{2,1}, \phi_{2,2}r_{2,2}, \dots, \phi_{2,D}r_{2,D}). \quad (6)$$

図 3 は 2 次元空間における (4) 式の第 2 項である $\Phi_1(\mathbf{p}_i - \mathbf{x}_i)$ の空間分布を图示したものである.

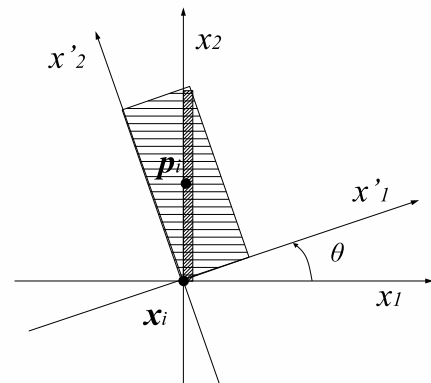


図 3 二次元空間における $\Phi_1(\mathbf{p}_i - \mathbf{x}_i)$ の空間分布.

座標系 x_1-x_2 の様に \mathbf{x}_i と \mathbf{p}_i が座標軸に平行に並ぶならば, $\Phi_1(\mathbf{p}_i - \mathbf{x}_i)$ の分布の自由度は 1 である. 一方, 座標系 $x'_1-x'_2$ の場合, 分布の自由度は 2 である. $\Phi_2(\mathbf{g}_i - \mathbf{x}_i)$ についても同様である.

本研究では, 速度の各成分の更新に他の次元の情報を利用するために, 座標軸 x_1, x_2 を回転させ座標系 $x'_1-x'_2$ に変換するような座標変換を速度の更新に導入する.

そこで, A を座標軸を回転させる $D \times D$ 行列とし, (5) 式を

$$\mathbf{v}_i = \chi(\mathbf{v}_i + A^{-1}\Phi_1A(\mathbf{p}_i - \mathbf{x}_i) + A^{-1}\Phi_2A(\mathbf{g}_i - \mathbf{x}_i)), \quad (7)$$

に置き換える. $\Phi_1(\mathbf{p}_i - \mathbf{x}_i)$, $\Phi_2(\mathbf{g}_i - \mathbf{x}_i)$ から値がサンプルされる前に座標軸を回転し, サンプル後は逆行列で元に戻す. 以下では, 速度更新式 (7) を用いた PSO を Roatated Particle Swarm(RPS) と呼ぶこととする.

5 数値実験

RPS および canonical な PSO [2] を Sphere 関数および以下の 4 種のテスト関数に適用した.

$$\text{DejongF4} : f = \sum_{d=1}^D ix_d^4 \quad (8)$$

Rosenbrock :

$$f = \sum_{d=1}^{D-1} 100(x_{d+1} - x_d^2)^2 + (x_d - 1)^2 \quad (9)$$

Griewank :

$$f = \frac{1}{4000} \sum_{d=1}^D x_d^2 - \prod_{i=1}^D \cos(x_d/\sqrt{d}) + 1 \quad (10)$$

Rastrigin :

$$f = \sum_{d=1}^D x_d^2 + 10 - 10 \cos(2\pi x_d) \quad (11)$$

Sphere 関数, DejongF4 関数, Rastrigin 関数は単峰性関数で, それ以外は多峰性関数である. 各関数は原点もしくは, その付近に最適解が存在する. 初期集団は表 1 のように, 原点に対し非対称に配置した [6].

表 1 探索領域と初期集団の範囲.

	探索領域	初期配置
Sphere	$[-50, 50]^D$	$[25, 40]^D$
DejongF4	$[-20, 20]^D$	$[10, 16]^D$
Rosenbrock	$[-100, 100]^D$	$[50, 80]^D$
Griewank	$[-600, 600]^D$	$[300, 500]^D$
Rastrigin	$[-5.12, 5.12]^D$	$[1, 4.5]^D$

集団のサイズは $N = 20$ とした. トポロジーは Star トポロジーおよび Von Neuman トポロジーを採用した. 次元数 $D = 30, 400$ のそれぞれに対して, 20 回の実験で得られた各ステップにおける最良の解の評価値の平均値を調べた. 各関数に対して 20 回の実験を行い, ステップ数に対する最良解の評価値の平均を図 4-13 に示す.

なお, A はランダムに回転の基準となる座標軸の組を選び, それらの軸の組で定義される平面上で, そ

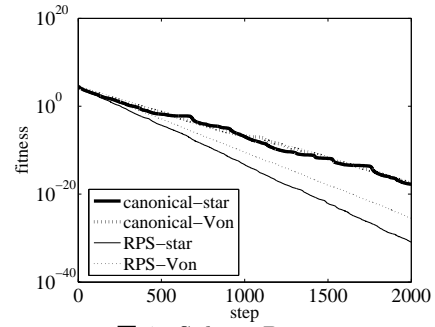


図 4 Sphere $D = 30$

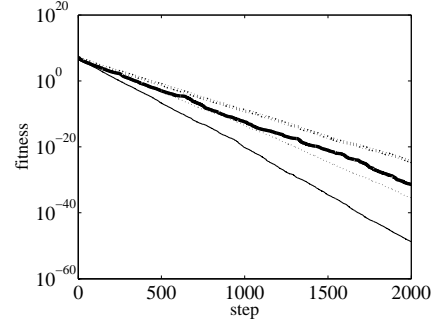


図 5 DejongF4 $D = 30$

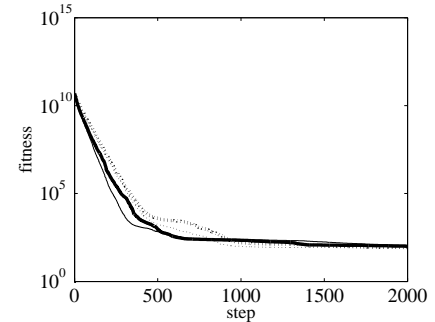


図 6 Rosenbrock $D = 30$

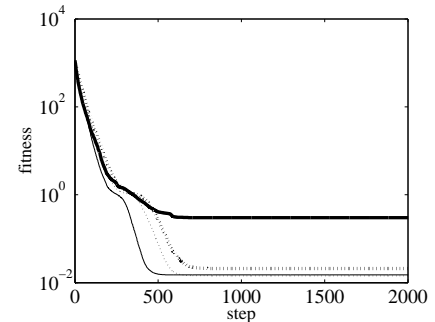


図 7 Rosenbrock $D = 30$

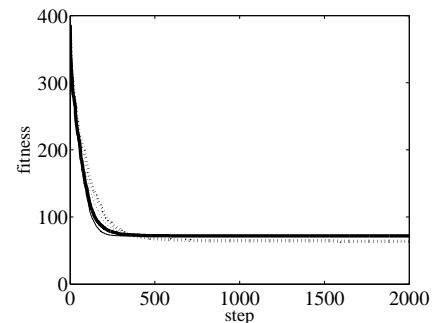


図 8 Rastrigin $D = 30$

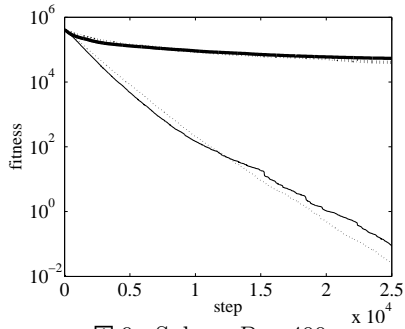


図9 Sphere $D = 400$

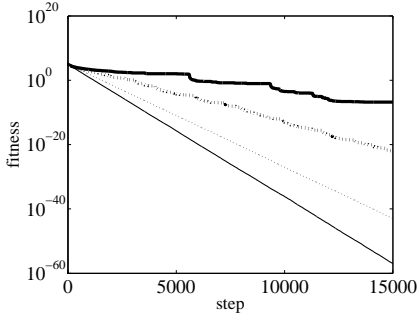


図10 DejongF4 $D = 400$

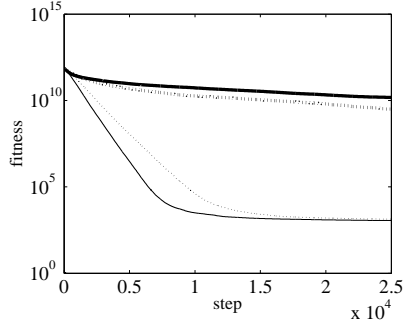


図11 Rosenbrock $D = 400$

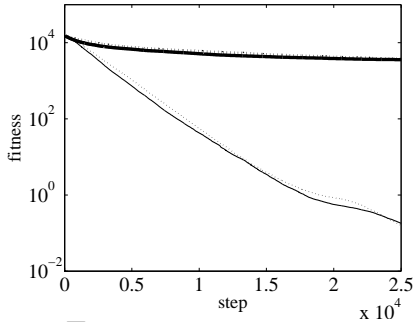


図12 Rosenbrock $D = 400$

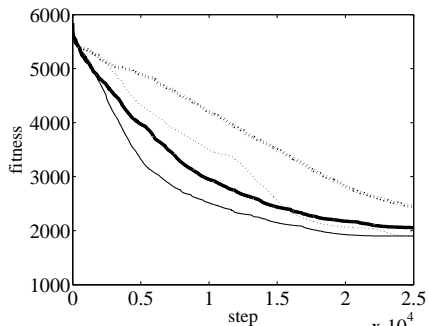


図13 Rastrigin $D = 400$

それぞれ θ だけ回転するように設定した．この実験では， $\theta = \pi/10$ とした．回転を作用させる平面は，問題の次元 D に対して，80% の確率で座標軸を選び組み合わせるものとした．

Sphere, DejongF4, Rosenbrock や Griewank に対して，いずれのトポロジーにおいても，RPS が canonical な PSO と比較して優れた結果を残している．特に $D = 400$ の場合において，図 9–13 に示すように，canonical な PSO の評価値がほぼ停滞しているのに対し，RPS では適合度が急速に改善されていくことがわかる．

6 おわりに

本研究では，高次元関数に対する探索性能の向上のために PSO に座標変換を導入した Roatated Particle Swarm(RPS) を提案した．RPS は座標軸を回転させることによって，大域的最適解や局所最適解以外での探索の停滞を防ぐ．数値実験の結果，高次元関数に対する有用性が示された．

参考文献

- [1] J. Kennedy and R.C. Eberhart, “Particle Swarm Optimization,” *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.
- [2] M. Clerc and J. Kennedy, “The particle swarm: Explosion, stability, and convergence in a multidimensional complex space,” *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, pp. 58–73, 2002.
- [3] E. Ozcan and C. K. Mohan, “Analysis of a Simple Particle Swarm Optimization System,” *Intelligent Engineering Systems Through Artificial Neural Networks*, Vol. 8, pp. 253–258, 1998.
- [4] E. Ozcan and C. K. Mohan, “Particle swarm optimization: Surfing the waves,” *Proceedings of 1999 IEEE Congress on Evolutionary Computation*, Vol. 3, pp. 1939–1944, 1999.
- [5] J. Kennedy and R. Mendes, “Population structure and particle swarm performance,” *Proceedings of 2002 IEEE Congress on Evolutionary Computation*, Vol. 2, pp. 1671–1676, 2002.
- [6] P. J. Angeline, “Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences,” *Evolutionary Programming*, Vol. 1447, pp. 84–89, 1998.