論文

突然変異を適用した粒子群最適化の探索性能に関する研究

A Study on Search Performance of Particle Swarm Optimization with Mutation Operators

印南 信男（近畿職業能力開発大学校）
Michio Innami

Particle Swarm Optimization (PSO) is one of the metaheuristic techniques. It is widely used in various fields. It is excellent in convergence of solutions. However it has a tendency to easily drop into local optima and is difficult to get out of them. Mutation operators used in Genetic Algorithm (GA) are effective in avoiding local optima and in maintaining diversity. Recently some researchers try to incorporate mutation operators to PSO to improve its search performance. In this paper, the influence that the amount of displacements caused by mutation operators gives the search performance is investigated. Five test functions are taken up to evaluate the performance.
Keyword: Algorithm, Metaheurisitics, Particle Swarm Optimization, Mutation

1. はじめに

組合せ最適化問題などに用いられるメタヒューリスティクスには，多くの手法が提案されている．その中のひとつである粒子群最適化 (Particle Swarm Optimization : PSO) [1-3] はアルゴリズムが比較的単純で，解の収束性に優れるという特徴を有しており，さまざまな分野への適用例が数多く報告されている．その一方で各粒子は最良解についての情報を集団内で共有しており，解の多様性が維持されづらい．そのため局所解に陥ると，そこからの脱出は容易ではない．

遺伝的アルゴリズム (Genetic Algorithm : GA) は突然変異オペレータによって，局所解に陥りづらくなる仕組みを有している．この操作を PSO に適用する試みが近年行われている．Wang ら [4] は突然変異による変位量がコーシー分布に従うモデルを提案した．Higashi ら [5] と Andrews [6] はいずれもガウス分布に従うモデルを適用した．Pant ら [7] はベータ分布によるモデルを採用した．Jančauskas [8] は設計変数を一様分布乱数によって得られた値に置き換えることによって，良好な結果が得られたと報告している．これらはいずれも特定の関数の確率密度に従って突然変異による変位量が決定されている．その一方で，変位量を一定とした突然変異について論じられた研究は，筆者の知る限り見当たらない．

そこで本研究では PSO に一定値の変位量を与える突然変異を作用させ，その効果を調べることとする．評価関数に対し，数種類の変位量による試行を行い，それらの結果を比較する．

2. 粒子群最適化 (PSO)

PSO は，鳥や魚などの群れの行動をモデルとして提案された群知能の一種である．集団を構成する各粒子(個体)の解空間内における座標は設計変数として用いられ，各イテレーションで以下のように更新される．

$$\boldsymbol{v}_i^{k+1} = w\boldsymbol{v}_i^k + c_1 r_1 (\boldsymbol{pbest}_i - \boldsymbol{x}_i^k) + c_2 r_2 (\boldsymbol{gbest} - \boldsymbol{x}_i^k) \tag{1}$$

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k + \boldsymbol{v}_i^{k+1} \tag{2}$$

ここで，$\boldsymbol{x}_i^k, \boldsymbol{v}_i^k$ はそれぞれ k 回目のイテレーションにおける粒子 i の座標ベクトル、速度ベクトルである．$\boldsymbol{pbest}_i$ は粒子 i の移動履歴中において最良解が得られた地点の座標ベクトル，$\boldsymbol{gbest}$ は全粒子の移動履歴中において最良解が得られた地点の座標ベクトルである．w は直前の速度成分に対する重みを表す定数，c_1, c_2 は粒子がそれぞれ $\boldsymbol{pbest}_i$，$\boldsymbol{gbest}$ に向かう成分に対する重みを表す定数である．r_1, r_2 は 0 から 1 までの範囲の一様乱数である。探索はイテレーションがあらかじめ設定された回数に達するまで繰り返される．

3. PSO の突然変異オペレータ

PSO においても，局所解から脱出するためには突然変異オペレータが有効であると考えられる．

突然変異の操作を含む PSO のアルゴリズムは以下の

とおりである．

1. 各粒子の初期座標として，解空間内における座標の初期値をランダムな値として生成する．またランダムな値の初期速度も生成する．
2. 各粒子について評価関数の値を求める．
3. 粒子 i について評価関数の値が $\boldsymbol{pbest}_i$ における値よりも小さい(より最適解に近い)場合には，その座標を新たに $\boldsymbol{pbest}_i$ とする．これをすべての粒子について行う．
4. 各粒子について評価関数の値が $\boldsymbol{gbest}$ における値よりも小さい(より最適解に近い)場合には，その粒子の座標を新たに $\boldsymbol{gbest}$ の値とする．
5. 式(1)，(2)によって各粒子の位置と速度を更新する．
6. 各粒子の各座標成分について，あらかじめ定められている突然変異率 p_m のもとで突然変異を適用し，その値を変化させる．
7. イテレーションが一定回数に達した場合は終了し，そうでなければ 2 に戻る．

突然変異による設計変数の変位量を適切な値に設定すれば，探索性能の向上により寄与するのではないかと考えられる．そこで，本研究ではその変位量を以下の式で定義した．

$$m = \pm w_\mathrm{m}(x_\mathrm{max} - x_\mathrm{min}) \tag{3}$$

ここで， w_m は突然変異による値の変位量の度合いを示す係数であり， $x_\mathrm{min}, x_\mathrm{max}$ はそれぞれ解空間内で設計変数がとり得る最小値，最大値である．式(3)における複号の一方がそれぞれ 0.5 の確率で選択されるものとする．突然変異によって設計変数が x_min 以下もしくは x_max 以上の値となった場合には，あらためて式(3)の複号の他方を採用する．その場合にも定義された領域内から外れるときには，それぞれ $x_\mathrm{min}, x_\mathrm{max}$ に置き換えることとする． $w_\mathrm{m} = 0.1, 0.3, 0.5, 0.7$ について探索を行う．

比較のために，突然変異によって設計変数を $[x_\mathrm{min}, x_\mathrm{max}]$ の区間の一様乱数で発生した値に置き換えた場合と，変化量がガウス分布に従う場合についてもとりあげた．ガウス分布については，参考文献[6]で提案されているとおりに分布が平均値 0，標準偏差 $0.1(x_\mathrm{max} - x_\mathrm{min})$ に従う乱数値を変位量とする．

ここでは突然変異率を $p_\mathrm{m} = 0.005$ とする．

4. 評価関数

探索性能を検証するため，以下に示す評価関数について大局的最小値を求める．座標の次元数 n の値は，突然変異を適用しない試行においてイテレーション回数 100 における評価関数の平均値がおおむね 10 を下回らない程度に設定した．

(a) Sphere 関数

$$f_1(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2 \tag{4}$$

$$-5.12 \le x_i \le 5.12 \ (i = 1, 2, \cdots, n)$$
$$\min(f_1(\boldsymbol{x})) = f_1(0, \cdots, 0) = 0$$

単峰性関数であり，変数間に依存関係は存在しない．n が大きな値になると急激に収束性が悪くなる． $n = 100$ とする．

(b) Rosenbrock 関数

$$f_2(\boldsymbol{x}) = \sum_{i=1}^{n-1} \{100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2\} \tag{5}$$

$$-2.048 \le x_i \le 2.048 \ (i = 1, 2, \cdots, n)$$
$$\min(f_2(\boldsymbol{x})) = f_2(1, \cdots, 1) = 0$$

単峰性関数であり，変数間に依存関係が存在する． $n = 30$ とする．

(c) Bohachevsky 関数

$$f_3(\boldsymbol{x}) = \sum_{i=1}^{n-1} \{x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7\} \tag{6}$$

$$-5.12 \le x_i \le 5.12 \ (i = 1, 2, \cdots, n)$$
$$\min(f_3(\boldsymbol{x})) = f_3(0, \cdots, 0) = 0$$

多峰性関数であり，変数間に依存関係は存在しない． $n = 30$ とする．

(d) Rastrigin 関数

$$f_4(\boldsymbol{x}) = 10n + \sum_{i=1}^{n} \{x_i^2 - 10\cos(2\pi x_i)\} \tag{7}$$

$$-5.12 \le x_i \le 5.12 \ (i = 1, 2, \cdots, n)$$
$$\min(f_4(\boldsymbol{x})) = f_4(0, \cdots, 0) = 0$$

多峰性関数であり，変数間に依存関係は存在しない．領域内には規則的に多数の局所解が存在する． $n = 30$ とする．

(e) Schwefel 関数

$$f_5(\boldsymbol{x}) = \sum_{i=1}^{n} \left(-x_i \sin\sqrt{|x_i|}\right) + 418.9828873n \tag{8}$$

$$-512 \le x_i \le 512 \ (i = 1, 2, \cdots, n)$$
$$\min(f_5(\boldsymbol{x})) = f_5(420.9687, \cdots, 420.9687) = 0$$

多峰性関数であり，変数間に依存関係は存在しない．

解空間の領域の境界付近に最適解が存在し，その周囲に局所解はない． $n = 10$ とする．

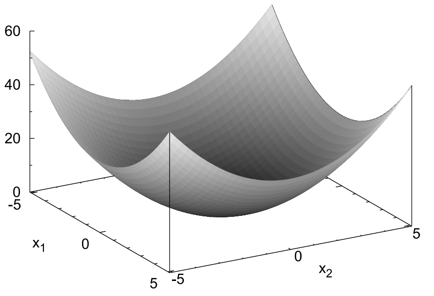


図 1　Sphere 関数

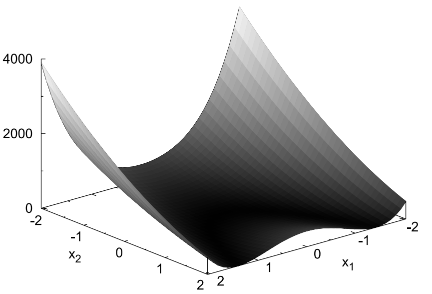


図 2　Rosenbrock 関数

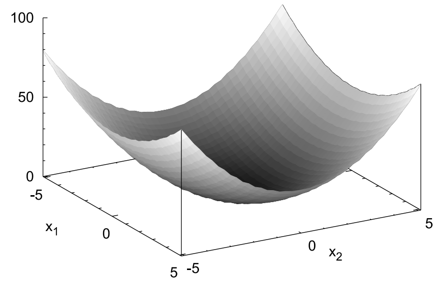


図 3　Bohachevsky 関数

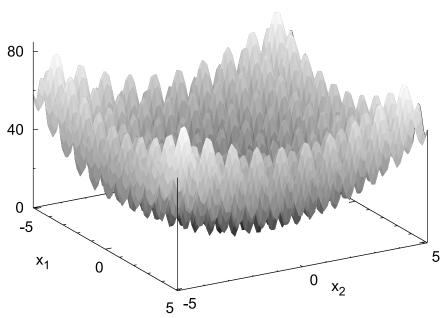


図 4　Rastrigin 関数

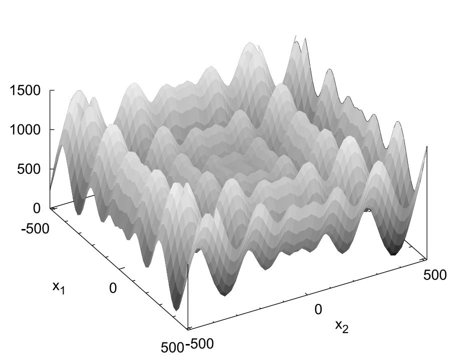


図 5　Schwefel 関数

それぞれの関数について，$n=2$ の場合の形状を図 1～5 に示す.

5.　試行結果

式(1)におけるパラメータを $w=0.729844$, $c_1=c_2=1.496180$ とした。これらは参考文献[9]で推奨されている値である．粒子数を 50 とした．それぞれの評価関数について, 各設定に関して 30 回ずつの試行を行った．計算はイテレーションが 10000 回に達するまで続けた．探索によって得られた関数値について，全試行の平均とイテレーション回数との関係を図 6 に示す.

いずれも突然変異を適用した探索ではおおむね良い成績が得られている.

Sphere 関数の場合には，突然変異による変位量は小さい値の方が好成績を得られている．単峰性関数であり設計変数間に依存性も存在しないため，大きな変化を与えてしまうと大局的最適解に近づいていた粒子がそこから離れてしまう可能性が考えられる.

Rosenbrock 関数では，突然変異を適用したいずれの場合でもあまり改善が見られなかった．本関数は大局的最小解が狭い溝状の領域内に存在しており，突然変異によってもこの部分に設計変数が存在できる確率がかなり小さいためであると考えられる.

Bohachevsky 関数，Rastrigin 関数については，変位量が小さい方が良い結果を得られている．一方 Schwefel 関数では変位量を大きく設定した方が良い結果となった．これらはいずれも多峰性である．前者の 2 関数は局所解が多く存在し, 互いの位置が比較的接近しているのに対し, Schwefel 関数はその間隔が大きいことが理由として考えられる.

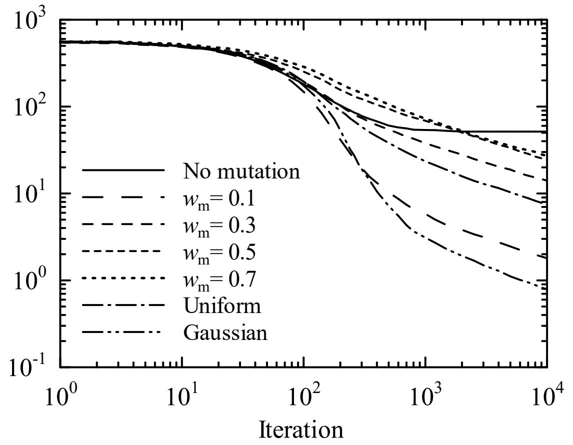
なお，一様分布による突然変異は変位量の分布が平均的であるため, 探索性能も中間的な結果となっている. ガウス分布による場合は，変位量が $\pm 0.1(x_{\max}-x_{\min})$ の間の値をとる確率が約 68%となるため，変位量が小さな場合にほぼ準じる結果となったと考えられる.

今回行った試行では，いずれも突然変異なしの場合はイテレーションが 100～1000 程度で解がほぼ収束しそれ以上探索を続けてもほとんど改善は見られないが，突然変異を適用した場合は多くの試行で 10000 回目まで改善が認められた.

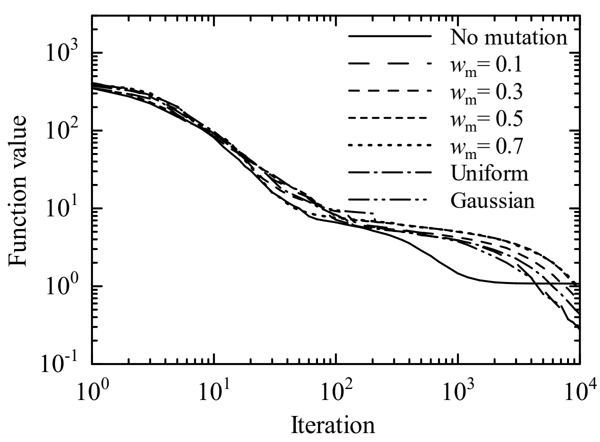
6.　まとめ

本研究では，PSO に突然変異オペレータを組込み，突然変異による設計変数の変位量が探索性能に及ぼす影響について検証した．本報でとりあげた評価関数に対する試行から以下の結論が得られた.

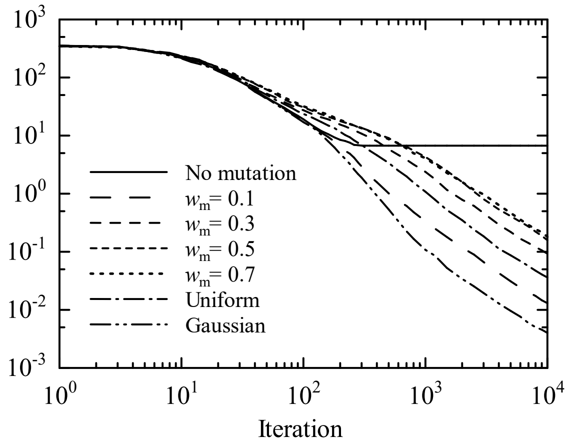
1. 単峰性関数で，変数間に依存性がない場合は，突然変異による変位量は小さい方が好成績を得られた.
2. 多峰性関数では，領域内で局所解が存在する位置ど



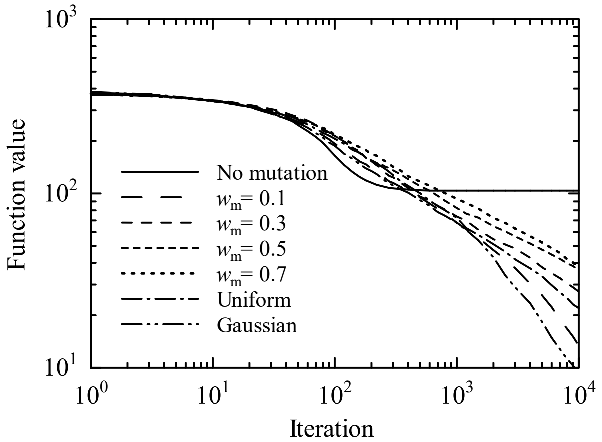
(a) Sphere 関数 (n = 100)



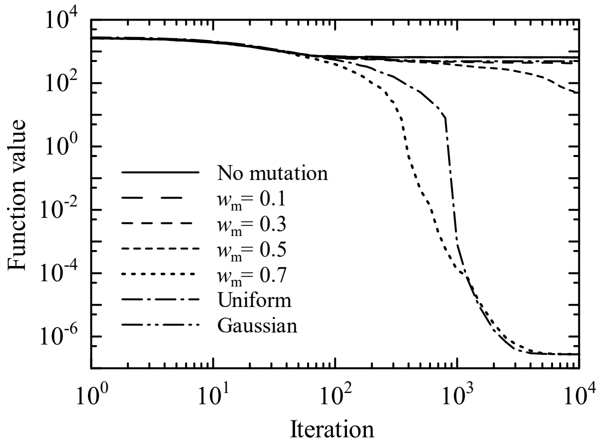
(b) Rosenbrock 関数 (n = 30)



(c) Bohachevsky 関数 (n = 30)



(d) Rastrigin 関数 (n = 30)



(e) Schwefel 関数 (n = 10)

図 6　試行で得られた関数の平均値の推移

うしの距離が比較的小さい場合には変位量が小さいほど優れた結果が得られ，逆に間隔が大きい場合は，変位量が大きいほど良い結果が得られた．
3. 関数の形状によっては突然変異の効果があまり現れない場合が存在する．
4. 突然変異を用いた探索ではイテレーション回数が大きくなっても，解が改善し続ける傾向がある．

今後は突然変異率をさまざまな値に設定して，その影響についても検討したいと考えている．

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, Piscataway, NJ (1995).

[2] J. Kennedy, and R. C. Eberhart: “A New Optimizer Using Particle Swarm Theory”, *Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (*MHS’95*), pp. 39-43, Nagoya, Japan, Piscataway, NJ (1995).

[3] Y. Shi, and R. C. Eberhart: “A Modified Particle Swarm Optimizer”, *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 69-73, Anchorage, Alaska, Piscataway, NJ (1998).

[4] H. Wang, Y. Liu, C. H. LI, and S. Y. Zeng: “A Hybrid Particle Swarm Algorithm with Cauchy Mutation”, *Proceedings of IEEE Swarm Intelligence Symposium*, pp. 356-360 (2007).

[5] N. Higashi, H. and H. Iba: “Particle Swarm Optimization with Gaussian Mutation”, *Proceedings of IEEE Swarm Intelligence Symposium*, pp. 72-79, Indianapolis (2003).

[6] S. P. Andrews: “An Investigation into Mutation Operators for Particle Swarm Optimization”, *IEEE Congress on Evolutionary Computation*, pp. 1044-1051 (2006).

[7] M. Pant, R Thangaraj, and A. Abraham: “Particle Swarm Optimization Using Adaptive Mutation”, *Database and Expert Systems Application, 2008* (*DEXA'08*). *19th International Workshop on. IEEE,* pp. 519-523 (2008).

[8] V. Jančauskas: “Empirical Study of Particle Swarm Optimization Mutation Operators”, *Baltic Journal of Modern Computing*, Vol. 2, No. 4, pp. 199-214 (2014).

[9] F. Van Den Bergh: “An Analysis of Particle Swarm Optimizers”, *PhD Thesis. University of Pretoria* (2006).

（原稿受付 2015/11/30，受理 2016/3/2）

*印南信男，
近畿職業能力開発大学校，〒596-0103　大阪府岸和田市稲葉町 1778　email:Innami.Michio@jeed.or.jp
Michio Innami, Kinki Polytechnic College, 1778 Inabacho Kishiwada, Osaka 596-0103