F-029

擬似的な勾配情報の活用による粒子群最適化の効率化
Improvement of Particle Swarm Optimization Using Pseudo-Gradient Information

内海 翔太† 亀山 啓輔‡
Shota Utsumi Keisuke Kameyama

1. はじめに

最適化問題とは，D次元上の空間Iで定義されるスカラー関数$f(\mathbf{x})$において，その関数が最大もしくは最小となる$\mathbf{x}$の値を求める問題である．最適化問題は幅広い分野で現れる問題であり，スケジューリング，省エネルギーのための資源の運用，画像の学習による認識，ニューラルネットワークの学習，ナビゲーションの経路探索など，最適化問題の重要性は高いものとなっている．最適化問題を解く手法には様々なものがあるが，近年，粒子群最適化が注目を集めている．粒子群最適化は不連続・非線形な目的関数を持つ最適化問題に適用でき，また収束性が高いという利点を持つため，様々な分野で用いられている．しかし，高次元での探索効率の低下や局所最適解への収束が起きることが欠点として挙げられている．また，探索の初期などでは，粒子の速度が大きいため，最適解の近傍を通過してもすぐに最適解に収束することができないことがある．そのため，本研究では勾配法と組み合わせることにより収束性を高め，より効率的な最適解への到達を目指す．

2. 粒子群最適化

粒子群最適化(Particle Swarm Optimization:PSO)とは，1995 年に Kennedy と Eberhart によって考案された非線形最適化問題を解くための手法である[1]．この手法は鳥や魚の群れの動きから考案されたものであり，複数の生物の個体に相当する「粒子」同士が情報交換を行うことで協調しながら空間 I を移動し最適解を探索する．

目的関数$f(\mathbf{x})$を最適化するとき，まず解空間上に複数の動作点(粒子)がランダムな位置と速度で置かれる．世代tにおけるi番目の粒子の速度ベクトル$\mathbf{v}_i(t)$と位置ベクトル$\mathbf{x}_i(t)$は以下の式(2.1)，式(2.2)により更新される．

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + \phi_1 r_1\left(\mathbf{p}_{best_i}(t) - \mathbf{x}_i(t)\right) + \phi_2 r_2\left(\mathbf{g}_{best_i}(t) - \mathbf{x}_i(t)\right) \quad (2.1)$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t) \quad (2.2)$$

このとき，ϕ_1，ϕ_2は定数，r_1，r_2は[0,1]の一様乱数となっている．また，$\mathbf{p}_{best_i}(t)$とは，時刻tまでの粒子iの時刻tまでに粒子が到達した最も評価の高かった位置，$\mathbf{g}_{best_i}(t)$とは時刻tまでの全ての粒子の到達した最も評価の高かった位置となっている．wはShiとEberhartによって提案された慣性係数である[2]．この係数を用いることにより，粒子の速度を減衰させ収束性を高めることができる．この計算をあらかじめ決めた世代の上限まで繰り返し実行する．

3. 勾配情報を利用した粒子群最適化

本節では，勾配法を組み合わせた粒子群最適化法を提案する．前節で述べたとおり，従来の PSO では，図 3.1 のように，特に速度が大きい場合に最適解の近くを通過することがあっても収束に至らないことがある．一方最急降下法などの勾配法では，動作点が最適解の十分な近傍に至った後には図 3.2 のように短時間で局所最適解に到達する．しかし勾配法は極値を複数持つ目的関数では，準最適解に収束する可能性がある．そこで，PSO と勾配法を組み合わせ，図 3.3 のように最適解の近くを通過する際に最適解に向かうようにし，PSO の欠点を補う新たな手法の提案を行う．

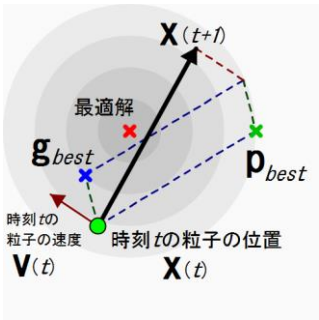

図 3.1 従来の PSO における最適解近傍にある粒子において，特に速度が大きい場合の動き

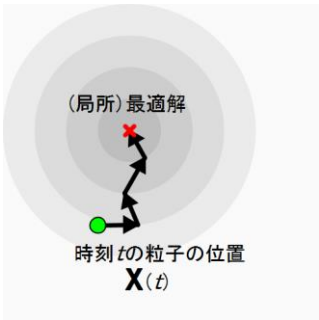

図 3.2 最急降下法における，最適解近傍にある粒子の動き

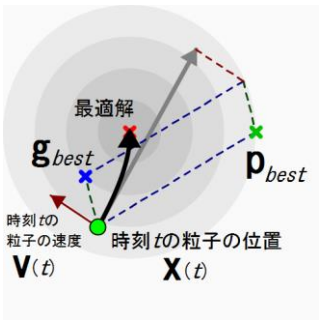

図 3.3 提案手法の動作の目標

†筑波大学 大学院システム情報工学研究科 コンピュータサイエンス専攻 Department of Computer Science, Graduate school of SIE, University of Tsukuba
‡筑波大学システム情報系 Faculty of Engineering, Information and Systems, University of Tsukuba

3.1 擬似的な勾配情報としての平均変化量

本研究では PSO と勾配法を組み合わせるが，勾配法にない PSO の長所である，微分不可能な目的関数でも最適化できるという特性を保つため，目的関数 $f(\mathbf{x})$ の勾配を，微分を用いてそのまま計算するのではなく，式(3.1)で表すように，ある座標 $\mathbf{x}_i$ の評価値 $f(\mathbf{x}_i)$から近傍$\hat{\mathbf{x}}_i$ の評価値 $f(\hat{\mathbf{x}}_i)$の差を座標のベクトルの各次元の長さで除算する，平均変化量$\nabla\hat{f}\left(\mathbf{x}_i(t),\mathbf{n}_i(t)\right)$を用いる．

$$\nabla\hat{f}\left(\mathbf{x}_i(t),\hat{\mathbf{x}}_i(t)\right)=\begin{pmatrix}\frac{f(\mathbf{x}_i(t))-f(\hat{\mathbf{x}}_i(t))}{x_{i1}(t)-\hat{x}_{i1}(t)}\\ \vdots \\ \frac{f(\mathbf{x}_i(t))-f(\hat{\mathbf{x}}_i(t))}{x_{iD}(t)-\hat{x}_{iD}(t)}\end{pmatrix} \quad (3.1)$$

本研究で平均変化量に用いる座標$\hat{\mathbf{x}}_i$ として，評価値 $f(\hat{\mathbf{x}}_i)$ の計算量を節約するために，粒子群最適化にも使われる座標を利用する．

まず，平均変化量は $\mathbf{x}_i$ と$\hat{\mathbf{x}}_i$ が近いほど勾配の推定精度が高くなるため，以下のような候補が挙げられる．

- nearest 法

最近傍の粒子
粒子 $\mathbf{x}_i$ の現在の座標を平均変化量に用いる．

- history 法

粒子 $\mathbf{x}_i$ の座標の N_b 個の履歴$B=\left\{\mathbf{x}_i^{t-N_b},\mathbf{x}_i^{t-N_b+1},\dots,\mathbf{x}_i^{t-1}\right\}$を用いる．

- pbest 法

粒子 $\mathbf{x}_i$ の今までで最も評価値が高かった座標 $\mathbf{p}_{besti}$ を用いる．

- gbest 法

粒子全体の今までで最も評価値が高かった座標 $\mathbf{g}_{besti}$ を用いる．

本研究ではどの手法が最も優れているかを実験により検証する．

3.2 gradientPSO

提案手法である，平均変化量を組み合わせた PSO を本研究では gradientPSO(以下 gPSO) と呼称する．gPSO の速度更新式は以下の式(3.2) となる．

$$\begin{aligned}\mathbf{v}_i(t+1)&=w\mathbf{v}_i(t)\\ &+\alpha\left(\phi_1 r_1\left(\mathbf{p}_{best_i}(t)-\mathbf{x}_i(t)\right)+\phi_2 r_2\left(\mathbf{g}_{best_i}(t)-\mathbf{x}_i(t)\right)\right)\\ &-(1-\alpha)\left(\phi_3\nabla\hat{f}\left(\mathbf{x}_i(t),\mathbf{n}_i(t)\right)\right)\end{aligned} \quad (3.2)$$

PSO と勾配法のバランスを調節する項 α と，評価値の擬似的な勾配を示す$\nabla\hat{f}\left(\mathbf{x}_i(t),\mathbf{n}_i(t)\right)$，偏微分の係数である$\phi_3$ を追加した．

PSO の探索能力を活かすため，また最適解の付近では PSO の問題点であった速い速度を弱めるために，α と β は，PSO と勾配法を切り替えるための係数となるものが入る．本研究で用いた係数の決定法は以下のとおりである．

- gPSO-sigmoid 法

評価値の値によって変化するシグモイド関数

$$\alpha=\varsigma\left(f\left(\mathbf{x}_i(t)\right)\right)=\frac{1}{1+exp\left(-cf\left(\mathbf{x}_i(t)\right)+h\right)} \quad (3.3)$$

この方式は評価値が低い時，PSO による広域探索を行い，評価値が高い時に勾配法による探索で早期の収束を行うものである．それにより，大域的探索では PSO の探索能力が生かされ，最適解付近での局所的探索では，勾配法の利点である解へ高速に収束する利点を用いることができる．目的関数により評価値の幅に差が出るため，係数 c，h による調節を行うこともある．

- gPSO-sin 法

時間変化する sin 関数

$$\alpha=0.5\sin\left(\frac{2\pi t}{a}\right)+0.5 \quad (3.4)$$

gPSO-sigmoid 法の考えられる欠点として，勾配法で収束した位置が準最適解であるとき，gbest や pbest の位置や評価値にかかわらず，局所解から抜け出せなくなる可能性があることである．そのため，gPSO-sin 法では周期的に PSO と勾配法の係数を変えることにより，勾配法で局所解に陥ってしまった粒子が再び広域探索に切り替わり gbest や pbest へ向かうことができるようになることを期待したものである．

4. 実験

4.1 実験の目的

前章で提案した gPSO を従来の PSO とともに最適化問題に適用し，その性能を比較する．まず前章で提案された平均変化量の推定法と，粒子の速度修正法の中で，どの手法が優れているのかをベンチマーク関数の最適化問題により評価する．そして，最も評価の高かった手法に対して実問題に近い問題として，文字認識を行う層状ニューラルネットワークのパラメータ最適化を行い，従来の PSO との性能比較を行う．

4.2 実験環境

実験を行うに当たり，全ての実験において慣性項wを 0.9 とした．慣性項が 1 より小さい場合粒子全体がより収束しやすくなる．

また，実験を行うために使用した計算機環境を以下の表 4.1 に示す．

表 4.1　実験に使用した計算機の環境

CPU	Intel Core i5-3570 CPU @ 3.40GHz 3.80GHz
メモリ	8.0GB
OS	Windows 7 Home Premium Service Pack 1
開発環境	Microsoft Visual Studio Express 2012
使用ライブラリ	OpenCV 2.45

4.3 実験 1:ベンチマーク最適化問題による評価

表 4.2 にある 9 つのベンチマーク関数を用いて，PSO と gPSO の性能の比較をし，最良の gPSO 適用条件について考察する．

実験 1 での性能の評価方法は，gbest と最適解との空間上の距離が 10^{-5} 以下になるまでの世代数を，表 4.1 にあるそれぞれのベンチマーク関数で 20 回実験を行い，平均を取る．ただし，次元 D を 2 とし，世代数の上限を 2000 世

表 4.2　実験に用いた関数

名前	評価関数
Sphere	$f(\mathbf{x}) = \sum_{i=1}^{D} x_i^2$
Rosenbrock	$f(\mathbf{x}) = \sum_{i=1}^{D-1} \left(100\left(x_i^2 - x_{i+1}\right)^2\right) + (1 - x_i)^2$
3rd De Jong	$f(\mathbf{x}) = \sum_{i=1}^{D} \lvert x_i \rvert$
4th De Jong	$f(\mathbf{x}) = \sum_{i=1}^{D} i x_i^4$
Rastrigin	$f(\mathbf{x}) = \sum_{i=1}^{D} \left(x_i^2 - 10cos(2\pi x_i) + 10\right)$
Ackley	$f(\mathbf{x}) = \sum_{i=1}^{D-1} \left(20 + e - 20e^{-0.2\sqrt{0.5\left(x_i^2 + x_{i+1}^2\right)}} - e^{0.5\left(cos(2\pi x_i) + cos(2\pi x_{i+1})\right)}\right)$
Stretched V sine wave	$f(\mathbf{x}) = \sum_{i=1}^{D-1} \left(x_i^2 + x_{i+1}^2\right)^{0.25} \left(1 + sin^2\left(50\left(x_i^2 + x_{i+1}^2\right)^{0.1}\right)\right)$
Griewank	$f(\mathbf{x}) = 1 + \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} cos\left(\frac{x_i}{\sqrt{i}}\right)$
Ridge	$f(\mathbf{x}) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j\right)^2$

代とする．そして，従来の PSO からどれだけ改善されているかの指標である，改善率を以下の式(4.1)により計算する．すべての関数での改善率の平均が，その gPSO の性能評価の指標となる．

$$\text{改善率} = \frac{\textit{従来の PSO の解の到達までの世代数}}{\textit{gPSO の解の到達までの世代数}} \qquad (4.1)$$

これらの関数の最小値は，Rosenbrock 関数が 1.0，それ以外の関数は 0 となっている．

実験 1 で比較する方式やパラメータは以下のものである．

- 平均変化量に用いる座標の選択方式
  - nearest 方式
  - history 方式
  - pbest 方式
  - gbest 方式
- PSO と勾配法の比率を調整する係数
  - sigmoid 方式，sin 方式に共通
    - ✧ PSO と勾配法の切り替えをする係数 α
    - ✧ 勾配法の定数係数ϕ_3
  - sin 方式での周期 a

4.3.1 結果

従来の PSO の評価値と提案する gPSO を用いた場合の評価の結果を以下の表 4.3 に示す．表の中に書いてあるパラメータは，その方式で最も良かった値を入が入っている．平均世代数と改善率は，そのパラメータ上で計算を行った

表 4.3　それぞれの方式の性能評価(世代数)

α-β	平均変化量	ϕ_3	N_b	a	平均世代数	改善率
従来 PSO					156.91	1.0
sigmoid	pbest	0.05			98.38	**1.5984037**
	history	0.02	1		104.62	1.4092669
	nearest				到達できず	到達できず
	gbest	0.01			194.96	0.4728981
sin	pbest	0.01		10	**89.32**	1.0395643
	history	0.04	1	10	139.34	0.6796525
	nearest				到達できず	到達できず
	gbest				到達できず	到達できず

ときの関数ごとの平均とした．

表 4.3 から，最も平均世代数が低いのは gPSO-sin/pbest の場合であるが，最も改善率が高いのは gPSO-sigmoid/pbest であることがわかる．これは，gPSO-sin/pbest は Griewank 関数のみ性能が著しく高かったためであり，Griewank 関数以外では性能は従来 PSO より劣るものとなっている．以上より，最も性能が良いのは gPSO-sigmoid/pbest で，ϕ_3=0.05 のときとわかった．

4.4 実験 2:手書き文字の認識の検証

より実問題に近い問題での評価のために，MNIST DATABASE による手書き数字の認識を行う 2 層パーセプトロンの結合荷重，閾値の学習を PSO により行う．従来の PSO と実験 1 で最も良かった方式とパラメータの gPSO との性能の比較をした．

4.4.1 実験に用いたデータ

実験 2 で用いたデータは MNIST DATABASE[3]である．これは 28×28 ピクセルの手書きの数字の画像データであり，クラス数は 10，学習用データが 60000，テストデータが 10000 あるものである．

4.4.2 実験手順

手書き文字の認識には，シグモイド関数を出力関数とする 2 層のパーセプトロンで行い，各素子の結合荷重と閾値を PSO により最適化する． gPSO の方式は gPSO-sigmoid/pbest，ϕ_3=0.05 を使用した．粒子数は 500 である．また，隠れ層は 100，出力層は 10 であり，テストの際には最も値が大きい出力層の素子の番号を認識クラスとした．PSO の探索空間の次元数は 79510 次元となった．

学習に当たり，学習データを 1000 個取り出して学習を行った．評価値は学習データの誤答率である．2000 世代経過した時点でテストデータでの評価を行い，正答率を最終的な評価とする．これを 5 回繰り返し，平均正解率を求めた．

4.4.3 実験結果

実験の結果を表 4.4 に示す．

表 4.4　gPSO と従来の PSO のニューラルネットワークにおける文字認識での正解率

	PSO	gPSO
1 回目	0.2616	0.2606
2 回目	0.2692	0.3399
3 回目	0.3000	0.3720
4 回目	0.3045	0.3765
5 回目	0.2987	0.4040
平均	0.2868	0.3506

表 4.4 から，実問題に近い問題でも gPSO は従来の PSO より高い性能を示すことが確認された．性能の差が実験 1 よりも大きくないのは，問題の複雑さと世代数の低さに原因が有ると考えられる．

5. まとめと今後の課題

PSO は最適化手法として勾配法と比較すると優れた手法であるが，粒子の速度が大きい場合最適解のそばを通過してもそのまま通過してしまい，解の早期の発見につながらない事があった．そのため，PSO の効率化として収束性の向上を目指した．PSO と収束性の強い勾配法と組み合わせるにあたり，微分不可能な目的関数への適用は PSO の利点の一つであるため，目的関数の微分ではなく平均変化量による擬似的な勾配法を PSO と組み合わせた．この提案手法を gPSO とし，適切な適用条件を示した．そしてベンチマーク関数とニューラルネットワークのパラメータの最適化問題により，従来の PSO を上回る解の探査能力を持つことが確認できた．

今後の課題として，MNIST のホームページに示されている条件で，バックプロパゲーションなど他の手法を上回ることができるのかの検証を行う．今回の実験は小規模なものであったため，より粒子数と実験時間を増やした条件で実験を行う．

参考文献

[1] Kennedy, Eberhart, “Particle swarm optimization”, Proceedings of IEEE International Conf. on Neural Networks, pp.1942-1948 (1995).
[2] Y.Shi, R.C. Eberhart, “A modified particle swarm optimizer”, Proceedings of IEEE International Conf. on Evolutionary Computation, pp.69-73 (1998).
[3] LeCun, Cortes, and J.C. Burges. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/.