

群知能に基づく動的な最大電力点追従について

On swarm intelligence based dynamic maximum power tracking

岩橋祐典[†]
Yusuke Iwahashi安川新[†]
Shin Yasukawa斎藤利通[†]
Toshimichi Saito

1. はじめに

粒子群最適化(PSO)[1]は生物の行動を参考にした最適化手法の一つであり複数の粒子が相互作用しながら探索を行う。

アルゴリズムが簡素であることから様々な問題で応用されている。その中の一つとして太陽電池の最大電力点追従(MPPT)がある。

本論文では PSO の MPPT への応用[2]の基礎として人工的な目的関数の最大値を求める問題を2つの手法を用いて考える。一つ目の手法は複数粒子を用いた山登り法である。二つ目の手法は PSO である。基本的な例題を用いた数値実験を行い、両手法の有効性を比較検討する。

2. 本論

PSO と山登り法を用いた数値実験では以下の正定関数の最大値を求めることを目的とする

$$f(x) \geq 0, x_{min} \leq x \leq x_{max} \quad (1)$$

今回用いた正定関数は太陽電池の MPPT において $x \rightarrow v[V]$, $f(x) \rightarrow P[W]$ に対応している関数とする。図1に(1)式の正定関数の例を示す。

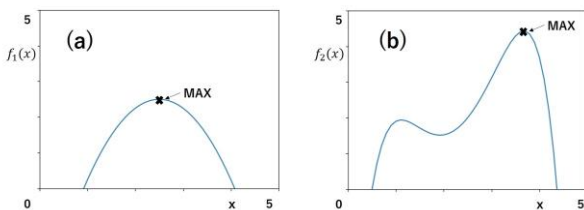


図1 目的関数の例(a)単峰性関数, (b)多峰性関数

2.1 山登り法のアルゴリズム

複数の粒子を用いそれぞれに山登り法を用いて最大値の探索を行う。

探索では以下のパラメータを使用した。

ステップ数 $t = 0 \sim T_{max}$

x_i : i 番目の粒子の位置

粒子数 $i = 1 \sim M$

山登り法は以下で定義する。

STEP1 (初期化): ステップ数を $t = 0$ とし, M 個の粒子を $0 \leq x_i \leq x_{max}$ の範囲で一様乱数を用いランダムに配置する。

STEP2 (粒子の位置更新): 各粒子の位置を次式に従って更新する。

$$\begin{aligned} x_i &\leftarrow x_i + dx & x_i &\leftarrow x_i - dx \\ \text{if } f(x_i) < f(x_i + dx) & \text{if } f(x_i) < f(x_i - dx) \end{aligned} \quad (2.1)$$

STEP3 (終了条件): $t \leftarrow t + 1$ に更新し, STEP2 へもどり, $t = T_{max}$ になるまで繰り返す。

2.2 PSO アルゴリズム

数値実験で用いる PSO の各パラメータは以下のように設定した。

$x_j(t)$: 時刻 t における j 番目の粒子の位置

$v_j(t)$: 時刻 t における j 番目の粒子の速度

粒子数 $j = 1 \sim N$

STEP1(初期化): ステップ数 $t = 0$ とする。粒子の位置と速度を以下の範囲でランダムに与える。

$$\begin{aligned} 0 &\leq x_j \leq x_{max} \\ -v_{max} &\leq v_j \leq v_{max} \end{aligned} \quad (3.1)$$

STEP2 位置、速度の更新: 各粒子の位置と速度の更新を次式によって行う。

$$\begin{aligned} x_j &= x_j + v_j \\ v_j &= wv_j + cp(x_p - x_j) + cp(x_L - x_j) \end{aligned} \quad (3.2)$$

ただし x_p : パーソナルベスト位置. x_L : ローカルベスト位置. ローカルベストからなるリング構造を用いる. w : 慣性係数, c : 加速度係数, p : $[0,1]$ 上の一様分布に従う乱数である。

STEP3 各粒子のパーソナルベスト, ローカルベストの更新を以下の式に従って行う。

$$\begin{aligned} x_p &= x_j, \text{ if } F(x_j) > F(x_p) \\ x_p &= x_p, F_p = F(x_p), \text{ else} \end{aligned} \quad (3.3.1)$$

$$\begin{aligned} x_L &= x_j \text{ for } F_{x_k} = \max_j F_{x_j} \\ \text{ただし } j &\in \{i, i-1, i+1\} \end{aligned} \quad (3.3.2)$$

STEP4 (終了判定): ステップ数を $t \leftarrow t + 1$ に更新し, STEP2 に戻り $t = T_{max}$ となるまで繰り返す。

3. 数値実験

複数粒子を用いた山登り法と PSO の比較を行うために数値実験を行った。数値実験での目的関数を図1(a)の単峰性関数 $f_1(x)$, 図1(b)多峰性関数 $f_2(x)$ とし二つの手法で最大値探索を行った。

山登り法のパラメータの値を次の様に設定した。

粒子数: $M = 5$

探索範囲の最大値: $x_{max} = 5$

粒子の速度の最大値: $v_{max} = 1$

探索時間の最大値: $T_{max} = 10$

粒子の刻み幅: $dx = 0.4$

次に, PSO のパラメータの値を次の様に設定した

粒子数: $N = 5$

探索範囲の最大値: $x_{max} = 5$

ステップ数の最大値: $T_{max} = 10$

慣性係数: $w = 0.5$
 加速度係数: $c = 1.0$.

山登り法を用いて数値実験を行った結果を図 2 に示す。

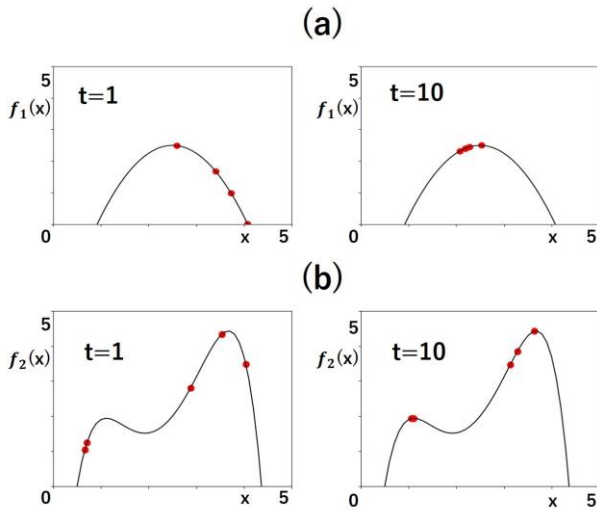


図 2 山登り法の数値実験結果
 (a) 単峰性関数, (b) 多峰性関数

図 2 の結果から、単峰性関数に対して山登り法を用いて最大点探索を行った場合 3 個の粒子が最大点に到達できているのに対し、多峰性関数で最大点に到達できた粒子は 1 個のみであることが確認できた。また多峰性関数では 2 個の粒子が準最大点でトラップしていることが確認できた。

次に PSO を用いて数値実験を行った結果を図 3 に示す。

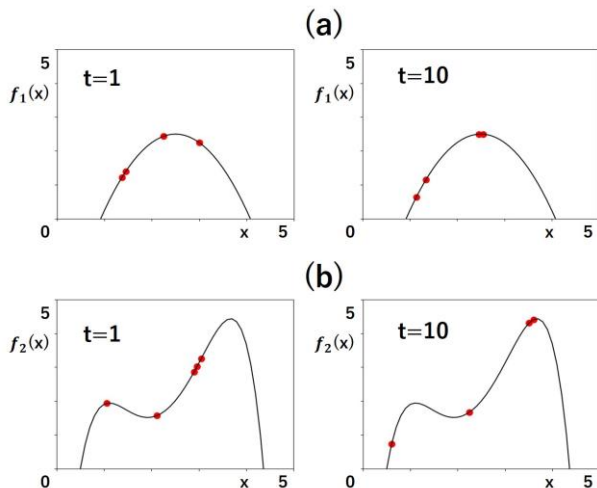


図 3 PSO の数値実験結果
 (a) 単峰性関数, (b) 多峰性関数

図 3 の結果から、単峰性関数に対して PSO を用いて最大点探索を行った場合 2 個の粒子が最大点に到達していることが確認できた。多峰性関数で探索を行った場合には 2 個の粒子が最大点を発見していることが確認できた。

以上の結果から PSO と山登り法を比較すると、単峰性関数においては複数粒子を用いた山登り法の方が最大点に到

達している粒子の数が多くことが前述の結果から確認できた。一方多峰性関数では PSO の方が最大点に到達している粒子の個数が多いことが確認できた。単峰性関数では複数粒子を用いた山登り法の方が有効である。多峰性関数では勾配情報のみで位置更新を行う山登り法は準最大点でトラップしてしまうため、リング構造の PSO の方が有効である。

4. むすび

単峰性関数、多峰性関数の最大点探索を複数粒子を用いた山登り法、リング構造の PSO で行った。山登り法で単峰性関数の探索を行うと最大点に到達している粒子は 3 個であり多峰性関数では 1 個であった。PSO を用いると単峰性関数で最大点に到達している粒子は 2 個であった。多峰性関数では 2 個であった。この結果を比較すると単峰性関数では勾配情報を用いた山登り法が有効である。一方多峰性関数では勾配情報を用いると準最大点でトラップしてしまうためリング構造の PSO の方が有効であることが確認できた。今後の課題としては MPPT の応用として目的関数が動的な関数での数値実験を行うなどが挙げられる。

参考文献

- [1] A. P. Engelbrecht, "Fundamentals of computational swarm intelligence", Wiley, 2005
- [2] S. Yasukawa, T. Saito, A multi-objective problem in a PSO-based control system for maximum power point tracking, Proc. IEEE/CEC, pp. 2629~2634, 2019