

人工知能 課題番号 25 「多目的 PCO の実装」

工学部電子情報工学科 03-175001 浅井明里

2018 年 1 月 11 日

## 1 本課題について

本課題では、個体群に基づく最適化アルゴリズムの一つである粒子群最適化アルゴリズム (Particle Swarm Optimizers、以下 PCO) を python で実装し、それを多目的関数最適化を可能にするよう拡張、また結果を 3 次元プロットにより可視化した。

## 2 PSO アルゴリズム及び MOPSO アルゴリズムについて

PSO は探索の対象となる目的関数が与えられたとき、複数の粒子が互いに情報を共有しながら最適解を求めて探索空間内を動き回ることにより最適化を行うアルゴリズムである。柔軟な並列処理が可能であること、勾配情報を用いない探索が可能であること、また非線形システムにも対応が可能など様々な利点があり、画像処理や組合せ最適化、ゲーム戦略など、様々な領域に応用が可能である。

まず、PSO の基本アルゴリズムについて説明を行う。 $M$  次元の探索空間  $S_0$  で、正定関数  $F$  の最小値を探索することとする。

$$F_x \leq C_1, \mathbf{x} = (x_1, \dots, x_M) \in S_0$$

これはある閾値  $C_1$  以下となる  $\mathbf{x}$  を探索する。 $C_1 = 0$  が厳密値の探索であり、 $0 < C_1$  が近似解の探索に対応する。PSO では  $N$  個の粒子が探索空間  $S_0$  を動き回って解を探索する。各粒子は

$$\text{位置 : } \mathbf{x}^i = (x_1^i, \dots, x_M^i), \text{ 速度 : } \mathbf{v}^i = (v_1^i, \dots, v_M^i)$$

により特徴付けられる。ここで  $i$  はそれぞれの粒子につけられた ID (インデックス) である。各粒子は  $S_0$  内に与えられた初期位置から与えられた初期速度で動作を開始する。この動作においては、以下の二つの量が重要となる。

### パーソナルベスト

$i$  番目の粒子の現在までの最良値を与える粒子位置。  $p^i = (p_1^i, \dots, p_N^i)$

また、全ての粒子のパーソナルベストの最良値をグローバルベスト  $g = (g_1, \dots, g_N)$  と呼ぶ。

### ローカルベスト

$i$  番目の粒子の近傍のパーソナルベストの最良値。  $l^i = (l_1^i, \dots, l_N^i)$

ここで近傍は粒子の結合形態に依存して決まる。全結合を仮定するとき、ローカルベストとグローバルベストは一致する。各粒子の相互作用は目的関数に基づくローカルベストを介してなされ、その相互作用によって、粒子群が形成される。

今回の実装にあたっては、粒子の更新式は以下のものを用いた。

$$v_j^i(n+1) = wv_j^i(n) + \alpha_1(p_j^i - x_j^i(n)) + \alpha_2(l_j^i - x_j^i(n))$$

$$x_j^i(n+1) = x_j^i(n) + v_j^i(n)$$

ここで  $w, \alpha_1, \alpha_2$  はパラメータである。パーソナルベストは各時刻の目的関数の値によって更新される。

$$p_i = x^i(n) \text{ if } Fx^i(n) < F(p^i)$$

またこのパーソナルベストの更新と連動して、ローカルベストとグローバルベストを更新する。

### 3 Multi-Objective PSO の実装

今回はこの PSO アルゴリズムを多目的関数に対応できるよう拡張した Multi-Objective Particle Swarm Optimization を実装した。Kumar and Minz(2014) は Multi-Objective Particle Swarm Optimization を実装する方法をいくつかあげているが、今回はそのうち最も広く用いられていると言われている、重み付けされた目的関数を複合し、一つの目的関数に置き換える方法を採用した。

目的関数の重み付け和は以下の式で表される。

$$F(x) = \sum_{i=1}^k w_i f_i(x)$$

ここで  $i = 1, 2, 3, \dots, k$  は  $k$  個の目的関数に与えられたインデクスであり、重みは全体で 1 になる非負の数である。

$$\sum_{i=1}^k w_i = 1$$

この関数の重み付けを動的に変更するアルゴリズムに Bang-Bang Weighted Aggregation(BWA) と Dynamic Weighted Aggregation(DWA) があり、今回は DWA を利用した。DWA は二つの目的関数に対する重み付けを以下の式に従って動的に変更する。

$$w_1(t) = \left| \sin \left( \frac{2\pi t}{f_w} \right) \right|$$
$$w_2(t) = 1 - w_1(t)$$

### 4 実装及び結果

Multi-Objective Particle Swarm Optimization については mopso.py というファイルにて、MOPSO クラスが定義されており、また同ファイル内の visualize\_result() 関数によって csv ファイルに保存された粒子の位置が三次元上にプロットできる。

MOPSO クラスについては  $x$  及び  $y$  の最大最小値などを渡して初期化したのち、任意の関数を定義しこの関数オブジェクトを set\_objective\_functions 関数に渡すことで、ある MOPSO インスタンスの目的関数に指定できる。実際に粒子の位置及び速度を更新するのは update\_particles 関数であり、これを指定されたステップ数実行し終わることを終了条件としている。

今回は以下の  $f_1, f_2$  を目的関数として MOPSO によりより良い解の探索を行った。

$$f_1 = x^2 + y^2$$
$$f_2 = \sin(x) + \cos(y)$$

この結果をプロットしたものが次の図 1 であり、実際に  $f_1$  及び  $f_2$  が小さくなるような  $(x, y)$  周辺に粒子が集まっている様子が確認できる。

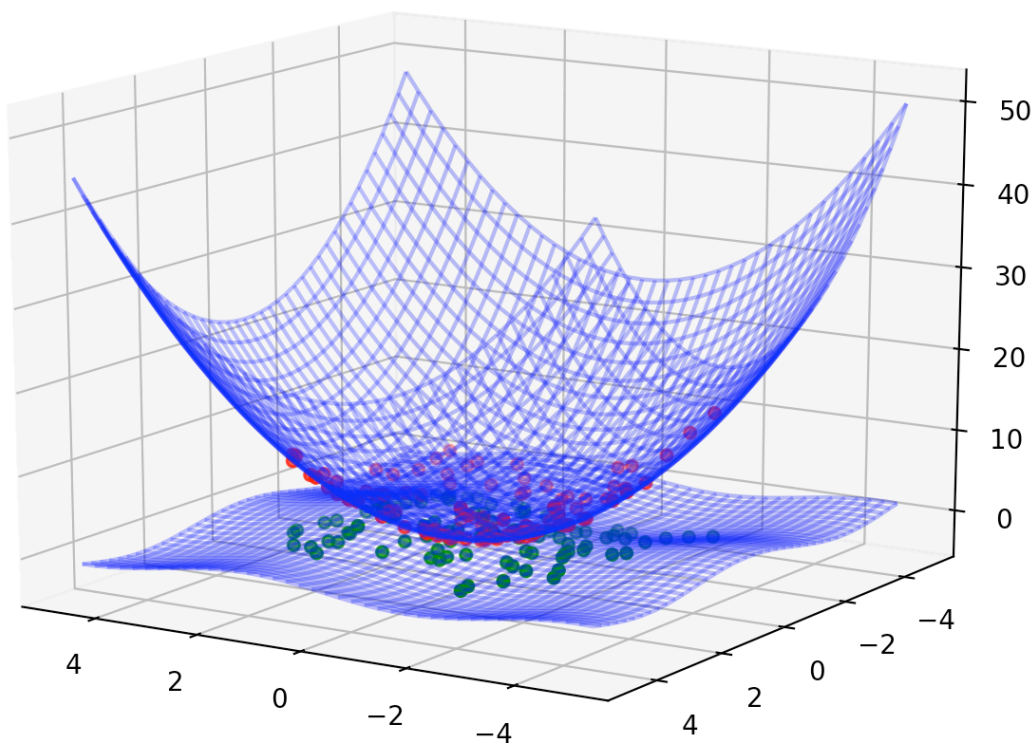


図1 MOPSO アルゴリズム結果の三次元プロット

## 参考文献

- [1] Vipin Kumar and Sonajharia Minz, "Multi-Objective Particle Swarm Optimization: An Introduction", Smart Computing Review, vol.4, no.5, p.335-353. 2014.
- [2] 斎藤利道『粒子群最適化と非線形システム』電子情報通信学会 基礎・境界ソサイエティ Fundamentals Review 5 巻 2 号. p.155-161. 2011.