

修正 PSO によるトラス構造物の最小重量設計

○菅谷 明誉*¹
曾我部 博之*²

キーワード：粒子群最適化 高次元問題 多峰性関数 構造最適化 最小重量設計

1. はじめに

粒子群最適化 (PSO: Particle Swarm Optimization) ¹⁾は、鳥などの群れをつくり生活をする生物の捕食活動から発想を得て創られた最適化手法である。アルゴリズムが単純で、その扱い易さから近年では様々な工学分野に応用され始めており注目を集めている。しかし、目的関数が多峰性あるいは高次元となる問題では、その探索性能が低下するという問題が指摘されている。本研究では、この問題点を修正した PSO (mPSO: modified PSO) ²⁾の性能を数値実験により検証し、さらにトラス構造物を対象とした高次元最適化問題に適応を試みる。

2. 粒子群最適化

2.1 計算の流れ

PSOは多数の粒子が集まって最適解を探索する手法で、それぞれの粒子 i は、速度ベクトル v_j^i と位置ベクトル x_j^i 、評価値 E^i の3つの情報を持っている。ここでの位置ベクトルは、最適化問題における設計変数に相当する。

図1はmPSOの計算手順をフローチャートで示したものである。計算の流れは従来のPSOと同じであるが、初期値の設定方法と漸化式における乱数の取り扱い方(図中の*)が従来のPSOと異なっている。

まず始めに、後述する速度ベクトルと位置ベクトルの漸化式で使われるパラメータを設定する。

2.2 初期値の与え方

次に、各粒子における速度ベクトルと位置ベクトルの初期値を与える。初期速度は次式のように与える。

$$v_j^i(0) = \frac{2(c+1)}{w}(-1)^i \quad (1)$$

ここで、 $v_j^i(0)$ は粒子 $i(i=1\sim m)$ における初期速度ベクトルの $j(j=1\sim n)$ 成分である。 w, c は速度ベクトルの漸化式における重みパラメータである。従来のPSOでは初期速度をランダムに与え、初期位置も実設計空間にランダムに与えている。しかし初期速度を小さく与えてしまうと、早い段階で粒子の位置ベクトルが原点付近に集まり、最適解に近づかずに探索を終了してしまうことがある。そこでmPSOでは初期速度を式(1)のように与え、解の探索が初期段階で広範囲に行われるようにした。なお、設計変数は区間 $[-1, 1]$ の標準化設計変数に変換している

ことから、位置ベクトルの初期値 $x_j^i(0)$ は、区間 $[-1, 1]$ の一様乱数で与えている。

2.3 漸化式について

粒子の速度ベクトルと位置ベクトルは、次の2つの漸化式を用いて更新を行う。

$$v_j^i(t+1) = wv_j^i(t) + cr^i(t)\{p_j^i(t) - x_j^i(t)\} + cr^i(t)\{g_j(t) - x_j^i(t)\} \quad (2)$$

$$x_j^i(t+1) = x_j^i(t) + v_j^i(t+1) \quad (3)$$

ここで、 $p_j^i(t)$ はステップ t までの粒子 i 自身の最良解を表し、 $g_j(t)$ は群れの中での最良解を表す。最良解とは評価値の最も良い粒子に対する位置ベクトルのことである。さらに、 $r^i(t)$ は $[-1, 1]$ の一様乱数である。式(2),(3)を最大ステップ数まで繰り返したときの $g_j(t)$ が、最適解となる。

従来のPSOと異なっている点は、速度ベクトルの漸化式における乱数の取り扱い方にある。従来のPSOでは、漸化式の2項目と3項目で異なった乱数を取り、さらにベクトルの成分毎にも異なった乱数を採用している。すなわち、同じ粒子で複数の乱数が与えられることになるが、mPSOでは1つの粒子内で乱数は一つだけになる。

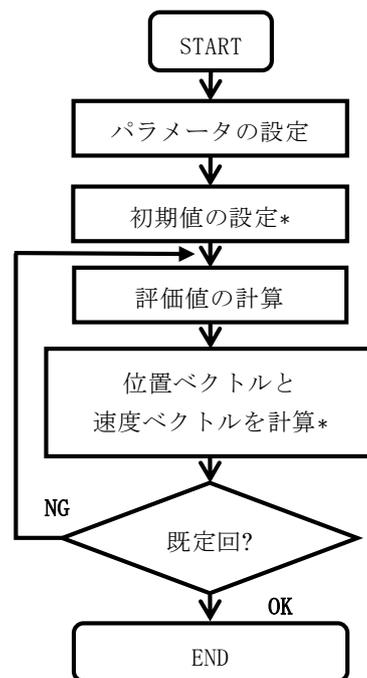


図1 mPSOのフローチャート

3 数値計算例

3.1 ベンチマーク問題

mPSO の探索性能を以下に示すベンチマーク問題で検討する. 比較の対象は, 従来の PSO 及び PSO と同じ群知能の一つである ABC (Artificial Bee Colony) ³⁾ とする. また, mPSO の初期速度が探索性能に及ぼす影響についても考察を行う.

① Parabolic 関数

$$\begin{aligned} \min. \quad & f_1(x) = \sum_{j=1}^n x_j^2 \\ \text{subject to} \quad & -5.0 \leq x_j \leq 5.0 \\ \text{大域的最適解} \quad & \mathbf{x}^* = (0, \dots, 0), \quad f_1(\mathbf{x}^*) = 0 \end{aligned}$$

② Rastrigin 関数

$$\begin{aligned} \min. \quad & f_2(x) = 10n + \sum_{j=1}^n (x_j^2 - 10\cos(2\pi x_j)) \\ \text{subject to} \quad & -5.12 \leq x_j \leq 5.12 \\ \text{大域的最適解} \quad & \mathbf{x}^* = (0, \dots, 0), \quad f_2(\mathbf{x}^*) = 0 \end{aligned}$$

③ Rosenbrock 関数

$$\begin{aligned} \min. \quad & f_3(x) = \sum_{j=1}^{n-1} \{100(x_j^2 - x_{j+1})^2 + (1 - x_j)^2\} \\ \text{subject to} \quad & -2.048 \leq x_j \leq 2.048 \\ \text{大域的最適解} \quad & \mathbf{x}^* = (1, \dots, 1), \quad f_3(\mathbf{x}^*) = 0 \end{aligned}$$

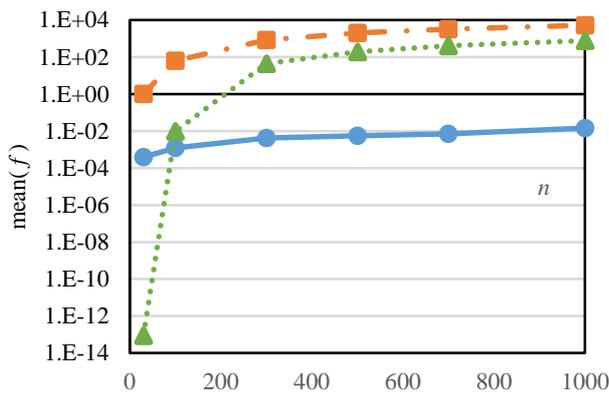
④ 2ⁿminima 関数

$$\begin{aligned} \min. \quad & f_4(x) = \sum_{j=1}^n (x_j^4 - 16x_j^2 + 5x_j) \\ \text{subject to} \quad & -5.0 \leq x_j \leq 5.0 \\ \text{大域的最適解} \quad & \mathbf{x}^* = (-2.9, \dots, -2.9), \quad f_4(\mathbf{x}^*) \cong -78n \end{aligned}$$

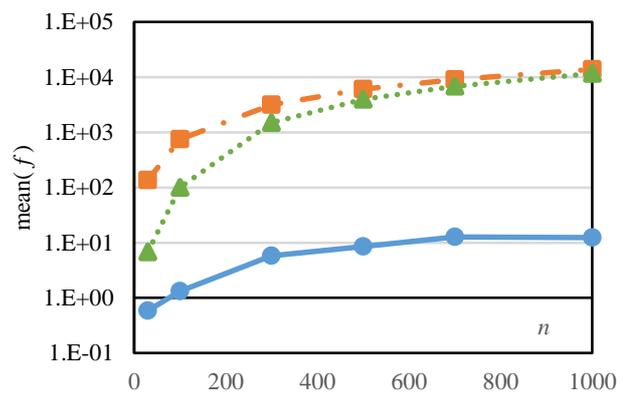
3.2 数値実験の結果

図 2 は, 次元 n 毎に 100 回の試行計算を行い, そこの評価値の平均 $\text{mean}(f)$ を示したもので, 従来の PSO ($c_1 = c_2 = 1.5, w = 0.7$) と mPSO ($c = 1.0, w = 0.7$) 及び ABC を比較したものである. PSO の粒子数は $m=20$, ABC のコロニーサイズは $N=20$ とし, 最大ステップ数を $t_{\max}=1000$ とした. 単峰性関数である Parabolic 関数の低次元では, ABC の探索性能が優れているが, 高次元になるにつれて, mPSO の探索性能が向上している. その他のベンチマーク関数では, mPSO の探索性能が全ての次元において優れているのがわかる.

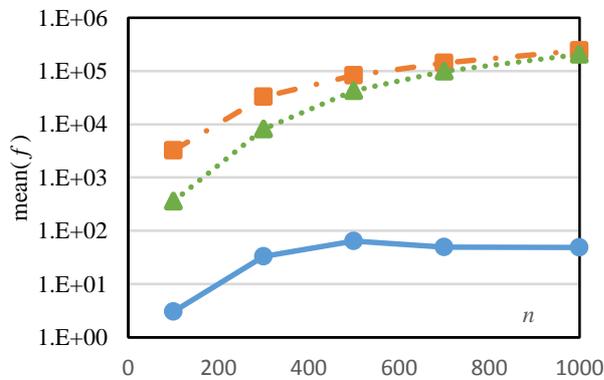
図 3 は, 次元数を $n=30$ として, mPSO の初期速度が探索性能に及ぼす影響について調べたものである. 最適解が原点 $\mathbf{x}^*=(0, \dots, 0)$ となる Rastrigin 関数や Parabolic 関数では, 初期速度を乱数と限界値のどちらで与えても探索



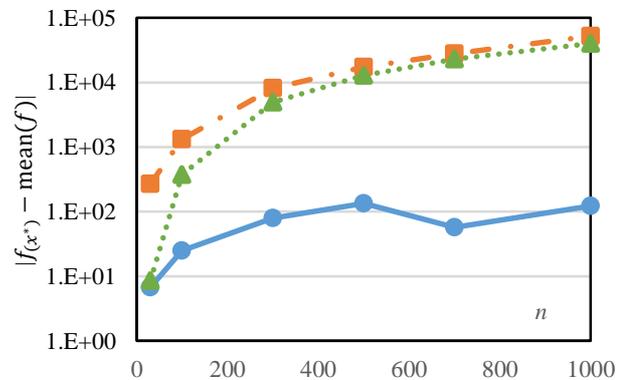
(a) Parabolic 関数



(b) Rastrigin 関数



(c) Rosenbrock 関数



(d) 2ⁿminima 関数

—●— mPSO —■— PSO —▲— ABC

図 2 探索性能の比較

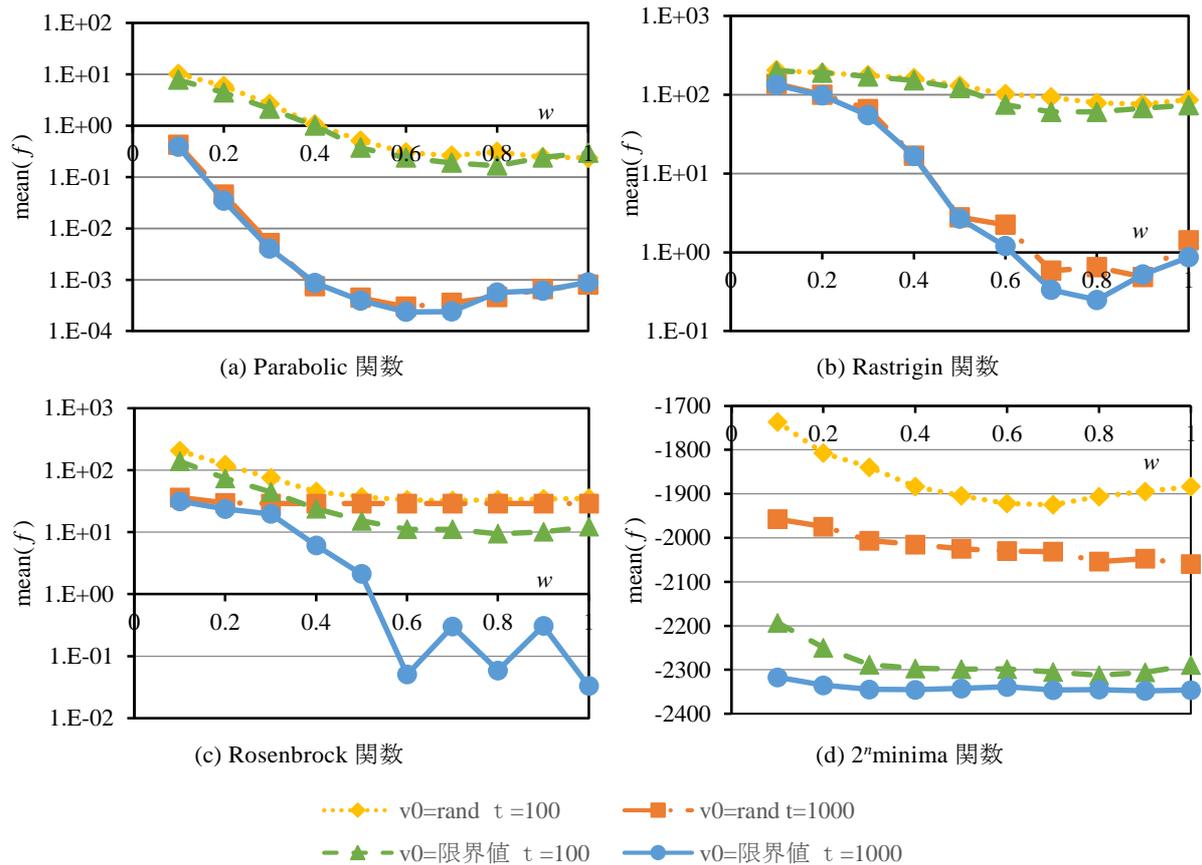


図3 初期値の影響

性能に大きな変化がみられない。最適解が原点以外である Rosenbrock 関数や 2^m minima 関数では、初期速度を限界値で与えることによって、探索性能が $t_{\max}=100$ と 1000 で向上しているのがわかる。さらに、 $t_{\max}=100$ と 1000 のときを比較すると初期速度を限界値で与えた方が $\text{mean}(f)$ の差が小さい、すなわち、収束が早くなっていることがわかる。注目すべき点は、初期速度を限界値で与えた $t_{\max}=100$ のほうが初期速度を乱数で与えた $t_{\max}=1000$ より探索性能が良くなっていることである。

実設計では、最適解が原点とは限らないこと、さらに高次元問題での計算時間短縮の観点から、式(1)で与えられる初期速度が有効であると考えられる。

4 立体トラスの最小重量設計

4.1 解析モデル

最適化の対象は、図4に示すような某体育館の屋根トラスとする。応力分布の対称性や境界条件を考慮し、図5に示すような立体トラス（設計変数 324）で最適化を実施する。

使用する部材は全て ss400 の鋼管である。荷重は、上弦材節点に鉛直荷重として 1.13 KN/m^2 が作用し、さらに各部材の自重も加わるものとする。

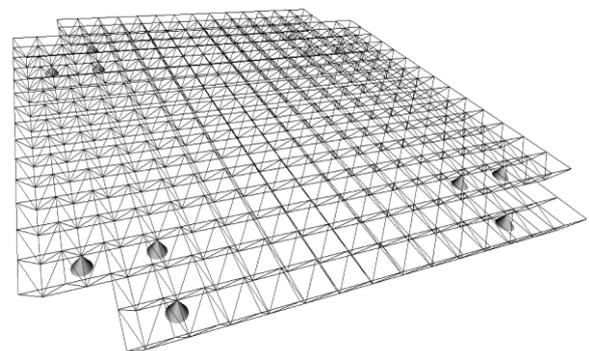


図4 全体パース

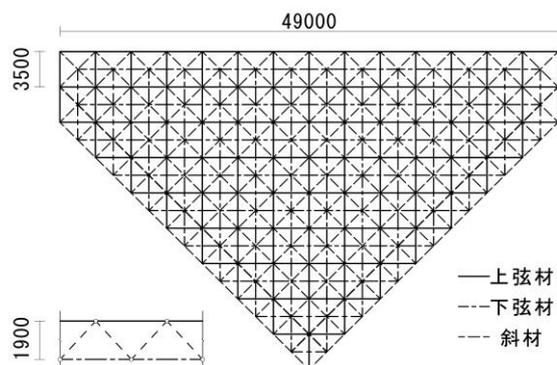


図5 解析モデル

4.2 最適化問題の定式化

立体トラス構造物の最小重量設計は、次式のように定式化される。

$$\min. \quad f_{(x)} = \frac{1}{w_0} \sum_{i=1}^n \rho x_i l_i \quad (4)$$

$$\text{subject to} \quad 100 \leq x_i \leq 3500$$

$$\begin{cases} \sigma_i \leq f_t \\ \sigma_i \leq f_c \end{cases} \quad (5)$$

ここで w_0 : 某体育館の屋根トラスの重量, ρ : 鋼材の密度, x_i : 部材 i の断面積, l_i : 部材 i の部材長, σ_i : 部材 i の応力度, f_t : 引張の許容応力度, f_c : 圧縮の許容応力度である。なお, 本研究では断面積 x_i の定義域を $[-1, 1]$ の標準化設計変数に変換している。

許容応力度は以下のように設定する⁴⁾。

$$f_t = 156 \text{ N/mm}^2 \quad (6)$$

$$f_c = \begin{cases} \left\{ \frac{1-0.4\left(\frac{\lambda}{\Lambda}\right)^2}{\frac{3}{2} + \frac{2}{3}\left(\frac{\lambda}{\Lambda}\right)^2} \right\} F & \lambda \leq \Lambda \\ \frac{0.227F}{\left(\frac{\lambda}{\Lambda}\right)^2} & \lambda > \Lambda \end{cases} \quad (7)$$

ここで, $\Lambda = \sqrt{\frac{\pi^2 E}{0.6F}}$, $\lambda = \frac{l_k}{i}$, Λ : 限界細長比, λ : 細長比, l_k : 座屈長さ, i : 断面二次半径, E : ヤング係数, F : F 値である。解析は短期設計で行うため, 上式で求めた許容応力度を 1.5 倍する。断面二次半径を求めるには, 鋼管の外径(D)と板厚(t)を求める必要がある。ここでは, 体育館で使用されている鋼管の断面と外径, 板厚の関係を調べ, 以下の近似式を用いている。

$$D_i = \sqrt{\frac{4x_i}{0.121}} \quad (8)$$

$$t_i = \frac{D_i - \sqrt{D_i^2 - \frac{4x_i}{\pi}}}{2} \quad (9)$$

4.3 評価値

各粒子の評価値 E は, 目的関数とペナルティ関数の和として, 次のように与えた。

$$E = f_{(x)} + c\gamma \quad (10)$$

ここで, c : 定数, γ : 許容応力度を超過した部材数である。

4.4 解析結果

mPSO のパラメータを $c=1.0$, $w=0.7$, 従来の PSO のパラメータを $c_1 = c_2 = 1.5$, $w = 0.7$ とし, 双方ともに粒子数 $m=20$, 最大ステップ数 $t_{max} = 1000$ とした。表 1 は 20 回の試行計算を行い, 評価値に関する統計量を示したものである。mPSO を用いることによって, 実在する屋根トラスの重量を 2 割程度削減することができた。従来の PSO では, あまり最適化が進まず, わずかな軽量化で探索が終了している。また, mPSO の変動係数が小さくな

表 1 評価値の結果

	mPSO	PSO
平均	0.869	1.14
最良値	0.785	0.982
最悪値	1.02	1.34
標準偏差	0.06	0.11
変動係数	0.069	0.097

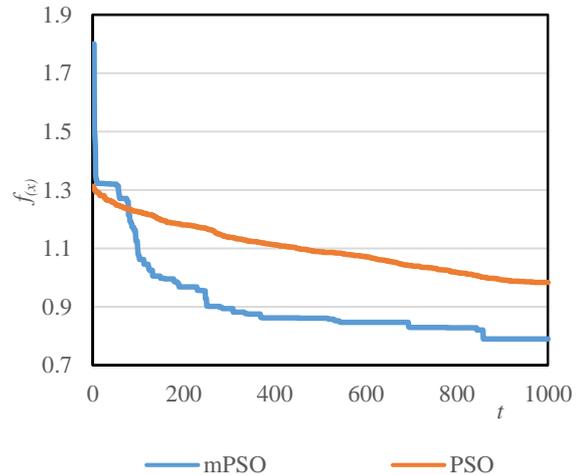


図 6 収束状況

っていることから, ばらつきの小さい安定した最適化手法であることがわかる。

5 おわりに

本研究では, 従来の PSO を僅かに修正した mPSO を提案し, 目的関数が多峰性あるいは高次元となる最適化問題において探索性能が大幅に改善されることを数値実験により示した。さらに, 実在する屋根トラスの最小重量設計問題に mPSO を適用し, その有効性を示すことができた。

【参考文献】

- 1) J.Kennedy and R.C.Eberhart : Particle Swarm Optimization IEEE Int. Conf. on Neural Networks, pp.1942-1948, 1995
- 2) 曾我部博之 : PSO における探索性能の向上に関する検討, 計算工学講演会論文集, Vol.19, 2014.6
- 3) D.Karaboga and B.Basturk : On the performance of artificial bee colony (ABC) algorithm, Applied Soft Computing, vol.8, pp.687-697, 2007
- 4) 日本建築学会 : 鋼構造設計規準-許容応力度設計法, 丸善, 2005

*1 愛知工業大学大学院 工学研究科 博士前期課程

*2 愛知工業大学工学部建築学科 教授 工博