日本建築学会・情報システム技術委員会
第 40 回情報・システム・利用・技術シンポジウム 2017

量子的振る舞いを用いた発見的最適化手法の解探索能力

－探索範囲に関係する無次元化パラメータ導入による解探索性能の向上－

○小田　佳明*1　本間　俊雄*2
横須賀　洋平*3

キーワード：最適化　発見的手法　量子的振る舞い　無次元化パラメータ　解探索性能の比較

1. はじめに

群知能(SI)に分類される粒子群最適化(PSO)[1]に、粒子(個体)に量子的振る舞いを取り入れた QPSO (Quantum behaved PSO)が提案され、ベンチマーク問題を通して近似最適解の探索能力向上が示されている[2]。著者らは他の発見的最適化手法である人工蜂コロニー(ABC) [3]と疑似焼きなまし法(SA) [4]及びホタルアルゴリズム(FA)[5]に、この量子的振る舞いを応用した QABC, QSA, QFA を構築してオリジナルアルゴリズムよりも近似最適解精度が向上することを示してきた[6]。

一般に発見的最適化手法は、解空間が複雑な問題や設計変数が複数存在する高次元の最適化問題において解探索能力が低下することが知られている。解空間や設計変数の変化に伴う計算パラメータの調整は直接計算コストの増加を意味する。本研究では量子的振る舞いスキームの解更新時に探索範囲を決定する計算パラメータの無次元化の導入により、問題に依存しない効率的な解探索スキームを示す。

本稿では特性が異なる 2 種類の高次元ベンチマーク問題を用いて、探索範囲に関係する無次元化パラメータの有無による量子的振る舞いアルゴリズムの比較を示し、解探索性能向上の検討を行う。なお極値解の確認には山登り法を採用する。

2. 最適化手法と量子的振る舞い

2.1 無次元化した量子的振る舞いスキーム

各探索個体が設計変数空間の特定の探索個体位置 Q_b に収束するモデルとして次式が与えられる[7]。

$$Q = \frac{(\varphi_1 Q_i + \varphi_2 Q_b)}{(\varphi_1 + \varphi_2)} \tag{1}$$

ここで、φ_1, φ_2: $[0,1]$ の乱数、Q_i: 個体 i の位置を表す。Q は変数の解更新に伴う探索幅の基準点である。設計変数の上限値と下限値をそれぞれ $X_{\max}, X_{\min}$(側面制約)としたとき以下の無次元化パラメータ h を設定する。

$$h = \left| \frac{Q_i - Q}{X_{\max} - X_{\min}} \right| \tag{2}$$

Q 周りの設計変数の探索に伴う探索幅 L は h を用いて次式で与える。

$$L = \begin{cases} \left(\frac{1}{g}\right)|Q_i - Q| & if \quad (\sqrt{h} \le g) \\ \left(\frac{1}{\sqrt{h}}\right)|Q_i - Q| & if \quad (g \le \sqrt{h} \le 0.996) \\ \left(\frac{1}{0.996}\right)|Q_i - Q| & otherwise \end{cases} \tag{3}$$

ここで、g は $g \geqq \ln\sqrt{2}$ を満たし、個体 Q_i の解更新をする上で、より精度の高い解に収束させる探索幅 L の調整を行う係数である。

更新個体位置は、基準点 Q から乱数で移動方向を決定し、探索幅 L を用いて次式で設定する。

$$\mathbf{X}_i^s = \begin{cases} Q - L(\ln(\frac{1}{u})) & if \quad i(r \ge 0.5) \\ Q + L(\ln(\frac{1}{u})) & otherwise \end{cases} \tag{4}$$

ここで、u, r: $[0,1]$ の乱数、s: 反復回数である。

Q_b の選択方法は各オリジナルアルゴリズムの解更新の手順に応じて決定し、式(4)の適用により各解法に対して量子的振る舞いの導入が可能となる。

以下に各解法の量子的振る舞いスキームを説明する。

2.2 量子的振る舞いを導入した PSO(QPSO)

目的関数値の最小化を対象とした計算手順は以下の通りである。

1) 群れの初期化 ： 探索点 $\mathbf{X}_i^1 (i = 1,2,...,n)$ を設計変数空間にランダムに配置する。n は個体数である。

$${}_p\mathbf{X}_i^1 = \mathbf{X}_i^1 \tag{5}$$

$${}_g\mathbf{X}_i^1 = {}_p\mathbf{X}_{ig}^1 \tag{6}$$

ここで、 $ig = \arg\min f({}_p\mathbf{X}_{ig}^1)$ である。

2) 位置の更新 ： 式(4)を用いて位置を $\mathbf{X}_i^{k+1}$ に更新する。ただし $\mathbf{X}_i^{k+1} = \mathbf{X}_i^s$ である。Q_b は群全体の中で目的関数を最小とする位置 ${}_g\mathbf{X}^k$ とする。

3) *pbest* の更新 ： 目的関数値より $f(\mathbf{X}_i^{k+1}) \le f({}_p\mathbf{X}_i^k)$ のとき ${}_p\mathbf{X}_i^{k+1} = \mathbf{X}_i^k$ 、$f(\mathbf{X}_i^{k+1}) > f({}_p\mathbf{X}_i^k)$ のとき ${}_p\mathbf{X}_i^{k+1} = {}_p\mathbf{X}_i^k$ とする。

4) *gbest* の更新 ：最も評価値の高い *pbest* から *gbest* を更新する。

$${}_g\mathbf{X}^{k+1} = {}_p\mathbf{X}_{ig}^{k+1} \tag{7}$$

以上 2)~5)の操作を指定した反復回数繰り返す。

2.3　量子的振る舞いを導入したABC（QABC）

目的関数値の最小化を対象とした計算手順は以下の通りである。

1）　<u>初期食糧源決定</u> ： *employed bee* の数だけ探索点 $\mathbf{X}_i^1 (i=1,2,...,n)$ を設計変数空間にランダムに配置する。*employed bee* を各一匹 ${}_{eb}\mathbf{X}_i^1$ を割り当てる。目的関数値 $f(\mathbf{X}_i^1)$ を計算する。ここで n は個体数である。

2）　<u>*employed bee* の探索</u> ： 食糧源の近傍で新たな食糧源を探索するために式(4)より位置を ${}_{eb}\mathbf{X}_{iw}^k$ に更新する。ただし ${}_{eb}\mathbf{X}_{iw}^k = \mathbf{X}_i^s$， ${}_{eb}\mathbf{X}_{iz}^k = \mathbf{X}_{iz}^{k-1}$ である。Q_b は近傍の中で目的関数を最小とする位置である。ここで w はランダムに選択された一つの設計変数であり、z は w 以外の設計変数である。目的関数値 $f({}_{eb}\mathbf{X}_i^k)$ を算出し、 $f({}_{eb}\mathbf{X}_i^k) \le f(\mathbf{X}_i^k)$ のとき $\mathbf{X}_i^{k+1} = {}_{eb}\mathbf{X}_i^k$ 、 $f({}_{eb}\mathbf{X}_i^k) > f(\mathbf{X}_i^k)$ のとき $\mathbf{X}_i^{k+1} = \mathbf{X}_i^k$ とする。

3）　<u>*onlooker bee* の探索</u> ： 新たな食糧源の評価値 fit_i^k を式(8)、相対価値確率 pl_i^k を式(9)により算出し、ルーレット選択により食糧源を選択して *onlooker bee* ${}_{ob}\mathbf{X}_i^k$ とする。

$$fit_i^k = \begin{cases} \dfrac{1}{1+f({}_{eb}\mathbf{X}_i^k)} & if \quad f({}_{eb}X_i^k) \ge 0 \\ 1+abs(f({}_{eb}\mathbf{X}_i^k)) & otherwise \end{cases} \tag{8}$$

$$pl_i^k = \frac{fit_i^k}{\sum_{j=1}^{N} fit_j^k} \tag{9}$$

ここで N は食糧源の数である。ルーレット選択後、式(4)を用いて解の比較を行う。

4）　<u>最良食糧源の記憶</u> ： 最も評価の高い食糧源は最良食糧源として保存する。 $f(\mathbf{X}_i^{k+1}) \le f({}_{best}\mathbf{X}_i)$ のとき ${}_{best}\mathbf{X}_i = \mathbf{X}_i^{k+1}$ とする。ただし初期値は ${}_{best}\mathbf{X}_i = \mathbf{X}_i^1$ である。

5）　<u>*scout bee* の探索</u> ： 食糧源が最初に指定した *limit* 回更新されなければ食糧源の一つを初期化する。

以上2)~5)の操作を指定した反復回数繰り返す。

2.4　量子的振る舞いを導入したSA（QSA）

目的関数値の最小化を対象とした計算手順は以下の通りである。

1）　<u>初期探索位置決定</u> ： 温度パラメータ T, スケーリングパラメータ s, 解の探索範囲 d を与える。設計変数空間内に探索個体 $\mathbf{x}_1$ をランダムに配置する。

2）　<u>位置の更新</u> ： 現在の解 $\mathbf{x}_k$ の近傍解を指定された数 i だけ発生し、各々目的関数値を算出する。その中で最小 $f(\mathbf{X}_k^i)$ の解を Q_b として、式(4)を用いた位置の更新を行う。ここで $\mathbf{X}_k^j = \mathbf{X}_i^s$ である。

$f(\mathbf{X}_k^j) \le f(\mathbf{X}_k)$ のとき $\mathbf{x}_{k+1} = \mathbf{x}_k^j$ として解を更新する。$f(\mathbf{X}_k^j) \ge f(\mathbf{X}_k)$ のとき次式で示す受理確率 q が乱数 $[0,1]$ 以上のとき $\mathbf{x}_{k+1} = \mathbf{x}_k^j$ として解を更新する。

$$q = \exp\left(-\frac{\left| f(\mathbf{X}_k^j) - f(\mathbf{X}_k) \right|}{Ts} \right) \tag{10}$$

3）　<u>パラメータの更新</u> ： パラメータ T, d に対し、あらかじめ設定した低減係数を乗じて更新する。

以上2)~3)の操作を指定した反復回数繰り返す。

2.5　量子的振る舞いを導入したFA（QFA）

目的関数値の最小化を対象とした計算手順は以下の通りである。なおクラスタ化はk-means法を用いる。

1）　<u>初期探索位置決定</u> ： パラメータ $\alpha \in [0,1]$, $\beta = 1.0$, $\gamma = [0,\infty]$ を与える。設計変数空間内に探索個体 $\mathbf{X}_i^1 (i=1,2,...,n)$ をランダムに配置する。ここで、n は個体数である。

2）　<u>目的関数値算出</u> ： 反復回数 k 回目の探索における i 番目の目的関数値 $f(\mathbf{X}_i^k)$ を計算する。

3）　<u>評価値計算</u> ： クラスタリングを適用した後、クラスタ内の無次元化した個体間距離 r_{ij} を式(11)により求め、評価値 I_{ij} を式(12)で計算する。ここで S は次元数である。

$$r_{ij} = \left\| \frac{\mathbf{X}_i^k}{\|\mathbf{X}_i\|} - \frac{\mathbf{X}_j^k}{\|\mathbf{X}_j\|} \right\| = \frac{1}{\sqrt{S}} \sqrt{\sum_{l=1}^{S} \left(\frac{X_{il} - X_{jl}}{X_{\max} - X_{\min}} \right)^2} \tag{11}$$

$$I_{ij} = I_0 e^{-\gamma r_{ij}} \tag{12}$$

$$I_0 = \begin{cases} \dfrac{1}{f(\mathbf{X}_i^k)} & if \quad f(\mathbf{X}_i^k) \ge 0 \\ \left| f(\mathbf{X}_i^k) \right| & otherwise \end{cases} \tag{13}$$

4）　<u>探索個体の移動</u> ： 式(14)により探索点 j の誘引度を算出し、式(4)を参考に式(15)を用いて探索個体を移動させる。ここで $\theta_i^{k+1} = \mathbf{X}_i^s$ である。

$$\beta_j = \beta e^{-\gamma r_{ij}^2} \tag{14}$$

$$\theta_i^{k+1} = Q \pm \beta_j (L(\ln(\frac{1}{u}))) \pm \alpha (X_{\max} - X_{\min}) \varepsilon \tag{15}$$

Q_b は個体間距離に基づく評価値が最も高い個体を選択し、乱数 r により移動の方向を決定する。

5）　<u>探索位置の比較</u> ： 目的関数値を比較し、$f(\theta_i^{k+1}) = f(\mathbf{X}_i^k)$ のとき $\mathbf{X}_i^{k+1} = \theta_i^{k+1}$、$f(\theta_i^{k+1}) \ne f(\mathbf{X}_i^k)$ のとき $\mathbf{X}_i^{k+1} = \mathbf{X}_i^k$ とする。

以上2)~5)の操作を指定した反復回数繰り返す。

2.6　局所探索（山登り法）

獲得解が近似的極値解であることを示すため、獲得解を初期値とした局所探索（山登り法）を適用する。計算手順は以下に示す通りである。

1）　<u>初期値</u> ： FAで得られた解 $\mathbf{X}_i$ を初期位置 ${}^0\mathbf{X}_i = \mathbf{X}_i$ とし、側面制約を基準に乱数発生範囲r%を設定する。

2）　<u>近傍解集合作成</u> ： 探索位置 ${}^k\mathbf{X}_i (k \ge 0)$ を中心とした r%範囲に標準偏差 σ の正規乱数を用いた近傍解 ${}^{k+1}\theta_{ij} (j=1,...,M)$ を配置する。ここでは、近傍解集合内の許容解の割合に応じ、以下の式でr%を狭める。

$$m_{k+1} = \begin{cases} m_k + 1 & if \quad M_i \le 0.2M \\ 0 & otherwise \end{cases} \tag{16}$$

$$r = \begin{cases} 0.8r & if \quad m_{k+1} \le 200 \\ r & otherwise \end{cases} \tag{17}$$

ここで、M: 近傍解集合数, M_I: 近傍解集合内の許容解の数

である。

3) <u>目的関数値算出</u> ： 反復回数k回目の目的関数値 $f({}^{k}\mathbf{X}_i), f({}^{k+1}\theta_{ij})$ を算出する。

4) <u>探索点位置移動</u> ： 近傍解集合内で目的関数値 $f({}^{k+1}\theta_{ij})$ の評価が最も良い近傍個体$j=g$を決定する。

5) <u>探索点位置比較</u> ： 近傍個体gの目的関数値と比較し、$f({}^{k+1}\theta_{ig}) \le f({}^{k}\mathbf{X}_i)$ のとき ${}^{k+1}\mathbf{X}_i = {}^{k+1}\theta_{ig}$ 、 $f({}^{k+1}\theta_{ig}) > f({}^{k}\mathbf{X}_i)$ のとき ${}^{k+1}\mathbf{X}_i = {}^{k}\mathbf{X}_i$ とする。

以上 2)~5)を指定した反復回数もしくは、収束条件を満たすまで繰り返し、極値解を同定する。

3. ベンチマーク問題による性能比較

以下に示すベンチマーク問題(n 次元関数最小値探索問題)により、既往のアルゴリズムとの性能比較を示し提示したスキームの有効性を検討する。

【Rastrigin 関数】

$$f_1(\mathbf{X}) = \sum_{j=1}^{n} \left\{ x_j^2 - 10\cos(2\pi x_j) + 10 \right\} \qquad (-5.12 \le x_j \le 5.12) \tag{18}$$

【Rosenbrock 関数】

$$f_2(\mathbf{X}) = \sum_{j=1}^{n-1} \left\{ 100(x_{j+1} - x_j^2)^2 + (x_j - 1)^2 \right\} \quad (-100 \le x_j \le 100) \tag{19}$$

式(18)は設計変数間が独立した深い局所的な谷を有する多峰性関数である。式(19)は設計変数間に従属性を有する単峰性関数である。共に最小値は 0 である。表 1 に各探索手法の計算パラメータ、表 2 に山登り法のパラメータ、図 1 に式(18)の例として 3 次元グラフ(n =2)を示す。各解法の 50 回の試行で得られた最良値 Best, 平均値 Ave., 最悪値 Worst を表 3, 4 に示す。図 2, 3 はそれぞれ式(18), (19)の 3, 5, 20 次元に対応する改良型 QPSO, QFA と PSO, FA の目的関数値の Ave.の収束状況である(縦軸: 目的関数値, 横軸: ステップ数)。また反復回数 10000 回に設定し各試行で評価値の更新が 200 回行われる間に評価値の差が 1.0×10^{-8} 以内となる場合を探索の終了条件とした(絶対評価)。ただし一般の最適化問題では相対評価を用いる必要がある。

4. 考察

表 3 より式(18)を用いた最適解探索能力に対して無次元化を取り入れた改良型 QPSO, QABC, QFA はそれぞれ旧量子的振る舞い QPSO, QABC, QFA より全次元で解精度が向上した。表 4 より式(19)に対しても、改良型 QPSO, QABC, QFA は全次元を通して優位性を保つ。QSA については同等の精度となっている。既往のオリジナルアルゴリズム PSO, ABC, SA, FA との比較で、各解法に対し式(18)の高い優位性がある。特に QFA は 2, 3, 5 次元で FA で得られた評価平均値の 1/10 以下で収束解が得られている [6]。

なお QFA は局所最適解を含む極値解を獲得する手法である。ここで 20 次元で得られた解を初期解と設定し、山登り法の適用によって改良型 QPSO, QABC, QFA, QSA の収束値の極値性を検討した。単峰性関数の式(19)では各解

表 1. 計算パラメータ

	PSO	SA
個体数	60	60
反復回数	1000	1000
T		1.0
s		1.0
冷却率		0.96
近傍数		1
C_1	1.2	
C_2	1.2	
g	0.8	0.8
	ABC	**FA**
個体数	60	60
反復回数	1000	1000
α		0.01
β		1.0
γ		0.01
クラスタ		10
employed	60	
onlooker	60	
limit	100	
g	0.8	0.8

表 2. 山登り法パラメータ

		PSO
近傍数		200
反復回数		10000
σ		0.3
r	上限値	0.01
	下限値	0.0001

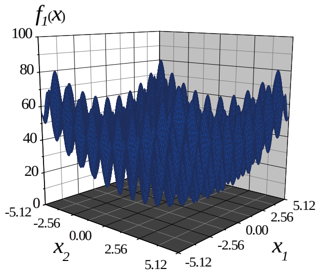


図 1. 多峰性関数の形状

Rastrigin 関数 (n =2))

表 3. 各解法による解の収束状況 (Rastrigin 関数式 (18))

algorithm	Dim.	2	3	5	20	50
QPSO (改良型)	Best	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	7.96×10^{0}	7.66×10^{1}
	Ave.	0.00×10^{0}	0.00×10^{0}	3.98×10^{-2}	1.64×10^{1}	1.02×10^{2}
	Worst	0.00×10^{0}	0.00×10^{0}	9.95×10^{-1}	2.89×10^{1}	1.63×10^{2}
QPSO	Best	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	8.95×10^{0}	7.66×10^{1}
	Ave.	0.00×10^{0}	0.00×10^{0}	1.39×10^{-1}	2.05×10^{1}	1.09×10^{2}
	Worst	0.00×10^{0}	0.00×10^{0}	9.95×10^{-1}	3.78×10^{1}	1.65×10^{2}
QABC (改良型)	Best	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	3.60×10^{-1}
	Ave.	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	1.48×10^{0}
	Worst	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	2.05×10^{0}
QABC	Best	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	1.44×10^{0}
	Ave.	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	5.06×10^{0}
	Worst	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}	7.38×10^{0}
QSA (改良型)	Best	7.74×10^{-5}	3.42×10^{-3}	1.76×10^{-1}	1.19×10^{2}	5.22×10^{2}
	Ave.	1.77×10^{-1}	1.16×10^{0}	4.10×10^{0}	1.51×10^{2}	6.40×10^{2}
	Worst	2.01×10^{0}	3.04×10^{0}	3.93×10^{1}	1.94×10^{2}	7.02×10^{2}
QSA	Best	1.81×10^{-4}	1.50×10^{-2}	1.40×10^{0}	1.13×10^{2}	5.46×10^{2}
	Ave.	2.67×10^{-1}	1.08×10^{0}	4.20×10^{0}	1.45×10^{2}	6.33×10^{2}
	Worst	1.99×10^{0}	3.07×10^{0}	8.48×10^{0}	1.87×10^{2}	7.32×10^{2}
QFA (c=10) (改良型)	Best	1.88×10^{-7}	3.31×10^{-4}	7.32×10^{-2}	1.45×10^{1}	9.99×10^{1}
	Ave.	2.06×10^{-2}	1.84×10^{-1}	9.56×10^{-1}	2.07×10^{1}	1.12×10^{2}
	Worst	7.93×10^{-1}	9.87×10^{-1}	2.06×10^{0}	3.89×10^{1}	1.38×10^{2}
QFA (c=10)	Best	4.63×10^{-7}	4.15×10^{-4}	8.96×10^{-2}	1.92×10^{1}	1.05×10^{2}
	Ave.	3.99×10^{-2}	2.03×10^{-1}	1.00×10^{0}	2.99×10^{1}	1.34×10^{2}
	Worst	9.95×10^{-1}	1.00×10^{0}	2.12×10^{0}	4.03×10^{1}	1.59×10^{2}

法で大幅な解の更新が行われ、個体が極値に達していないと判断する。多峰性関数の式(18)では、改良型 QPSO, QABC, QSA は目的関数の最小値が極値を捉えていることを確認した。一方で改良型 QFA は全探索個体の設計変数がそれぞれ1.0×10^{-2}以下の微小変化で更新が行われ、他の解法と比較して、収束解の変化が極めて小さい。これらの数値結果より改良型 QFA で獲得した収束解が局所解であると判断している。

改良型 QPSO, QABC, QFA, QSA とオリジナルアルゴリズム PSO, ABC, SA, FA による反復回数 1000 回の解析時間比較では、オリジナルアルゴリズムの解析時間をそれぞれ 1.0 とするとおおよそ改良型 QPSO は式(18), 式(19)で全次元で 1.2、改良型 QABC は式(18)では全次元で 1.0、式(19)では 5 次元まで 1.0、より高次元になると 1.1 になる。改良型 QSA は式(18), 式(19)で全次元で 1.1、改良型 QFA は式(18), 式(19)で全次元で 1.0 であった。

表 4. 各解法による解の収束状況（Rosenbrock 関数式（19））

algorithm	Dim.	2	3	5	20	50
QPSO（改良型）	Best	0.00×10^{0}	2.27×10^{-7}	2.28×10^{-4}	1.24×10^{-2}	2.61×10^{0}
	Ave.	1.61×10^{-29}	2.29×10^{0}	6.99×10^{0}	2.80×10^{1}	1.07×10^{2}
	Worst	6.02×10^{-28}	2.67×10^{1}	4.26×10^{1}	2.76×10^{2}	5.82×10^{2}
QPSO	Best	0.00×10^{0}	2.94×10^{-7}	1.82×10^{-3}	1.46×10^{-2}	2.18×10^{0}
	Ave.	2.88×10^{-23}	6.85×10^{0}	1.18×10^{1}	3.77×10^{1}	1.25×10^{2}
	Worst	2.88×10^{-21}	5.99×10^{1}	2.69×10^{2}	2.17×10^{2}	6.24×10^{2}
QABC（改良型）	Best	3.16×10^{-9}	3.46×10^{-7}	1.25×10^{-4}	1.14×10^{-3}	3.04×10^{0}
	Ave.	2.35×10^{-2}	7.23×10^{-1}	1.30×10^{0}	3.76×10^{0}	6.11×10^{1}
	Worst	4.12×10^{0}	1.82×10^{1}	9.29×10^{0}	4.42×10^{1}	1.28×10^{2}
QABC	Best	5.84×10^{-9}	6.04×10^{-7}	1.40×10^{-4}	4.62×10^{-3}	3.85×10^{0}
	Ave.	5.31×10^{-2}	8.93×10^{-1}	1.75×10^{0}	4.84×10^{0}	7.47×10^{1}
	Worst	1.78×10^{0}	1.19×10^{1}	8.34×10^{0}	4.64×10^{1}	1.87×10^{2}
QSA（改良型）	Best	1.59×10^{-3}	4.25×10^{-2}	4.99×10^{0}	4.02×10^{8}	2.01×10^{10}
	Ave.	4.34×10^{1}	1.29×10^{3}	8.73×10^{3}	1.53×10^{9}	3.59×10^{10}
	Worst	1.21×10^{2}	9.04×10^{3}	2.59×10^{5}	3.79×10^{9}	4.89×10^{10}
QSA	Best	1.99×10^{-3}	2.92×10^{-1}	2.88×10^{1}	4.11×10^{8}	2.13×10^{10}
	Ave.	4.10×10^{1}	2.01×10^{3}	6.98×10^{3}	1.56×10^{9}	3.72×10^{10}
	Worst	1.21×10^{2}	1.01×10^{4}	6.90×10^{4}	4.39×10^{9}	4.97×10^{10}
QFA (c=10)（改良型）	Best	7.49×10^{-6}	1.89×10^{-3}	6.28×10^{-1}	2.89×10^{2}	1.99×10^{4}
	Ave.	1.08×10^{-3}	5.22×10^{-2}	2.72×10^{0}	7.44×10^{2}	3.50×10^{4}
	Worst	9.61×10^{-3}	1.34×10^{-1}	5.71×10^{0}	1.93×10^{3}	3.99×10^{4}
QFA (c=10)	Best	8.40×10^{-6}	2.06×10^{-3}	9.45×10^{-1}	4.05×10^{2}	2.65×10^{4}
	Ave.	1.24×10^{-3}	7.07×10^{-2}	3.66×10^{0}	8.67×10^{2}	3.88×10^{4}
	Worst	1.04×10^{-2}	1.99×10^{-1}	6.21×10^{0}	2.09×10^{3}	4.77×10^{4}

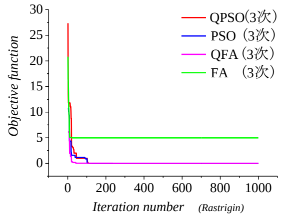
図 2 に示す式(18)の収束状況(代表的な一例)の比較より改良型 QPSO が PSO より早い段階で収束解が得られている。反復回数 10000 回の計算終了条件では 20 次元で PSO が平均 1441 回の反復探索で終了するのに対し、改良型 QPSO では平均 489 回の約 1/3 コストで終了した。同じく図 3 に示す式(19)の 20 次元では FA が平均 887 回の反復探索で終了するのに対し、改良型 QFA では平均 572 回の約 3/5 のコストで終了した。量子的振る舞いスキームによる獲得解はオリジナルアルゴリズムより早く収束する特性を持つ。改良型の提示スキームは旧量子的振る舞いスキームと同様に早い収束性を有し、目的関数の収束状況もよい(図 4. 代表的な一例)。改良型の収束解の精度は旧スキームと比較しても確実に向上している。

5. まとめ

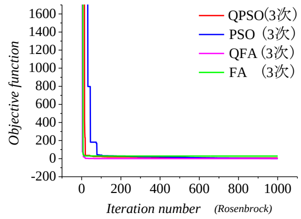
本稿では探索範囲に関係する無次元化パラメータを導入した改良型 QPSO, QABC, QSA, QFA を示した。高次元最小値探索問題の計算を用いて、旧 QPSO, QABC, QSA, QFA との比較により解探索性能を調べた。無次元化パラメータを導入した量子的振る舞いスキームの導入により最小値を探索するベンチマーク問題式(18)，(19)に対して、全ての解法で近似解の精度を向上させることができた。また改良型 QPSO, QABC, QSA, QFA と PSO, ABC, SA, FA の解の探索で提示スキームの解収束性の速さから、より小さなコストで精度が高い解の獲得可能性を確認した。

改良型 QPSO、QABC は低次元だけでなく高次元の大域最適解の探索精度が向上し、複雑な構造モデルに対して近似的大域的最適解の獲得が期待できる。一方で改良型 QFA では局所解が得られるので、様々な制約条件を満たした構造モデルの極値解が得られることが予想される。

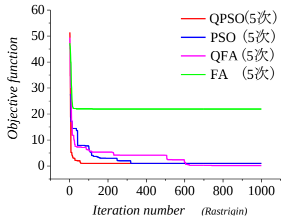
今後、提示アルゴリズムのさらなる改良を含め、実構造の最適化問題への適用例を示していきたい。



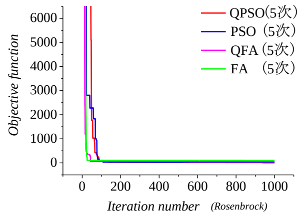
a. 3 次元(Rastrigin 関数)



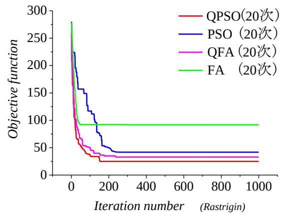
a. 3 次元(Rosenbrock 関数)



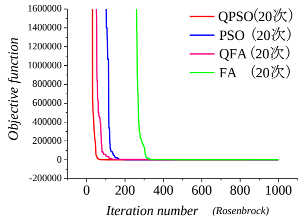
b. 5 次元(Rastrigin 関数)



b. 5 次元(Rosenbrock 関数)

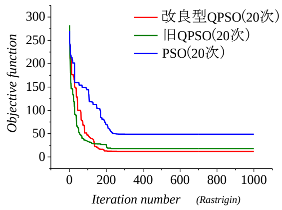


c. 20 次元(Rastrigin 関数)

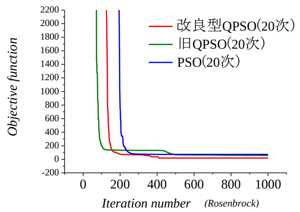


c. 20 次元(Rosenbrock 関数)

図 2. 目的関数収束値(Ave.)　**図 3. 目的関数収束値(Ave.)**



a. 20 次元(Rastrigin 関数)



b. 20 次元(Rosenbrock 関数)

図 4. 改良型 QPSO, 旧 QPSO, PSO 目的関数収束値(Ave.)

［参考文献］

1) J. Kennedy and R C. Eberhart : Particle Swarm Optimization. In: Proc. IEEE Int. Conf. Neural Networks, 1942–1948, 1995
2)J. Sun, B. Feng and W. Xu, : Particle Swarm Optimization with Particles Having Quantum Behavior, Proc. Congress on Evalutionary Computation, 2004
3)D.Karaboga and B. Basturk : A powerful and efficient algorithm for numerical function optimization : artificial bee colony (ABC) algorithm, Joumal of Glob Optimization 39, 459-471, 2007
4) Kirkpatrick, S., Gelett, Jr. C. D., and Vecchi, M. P. : Optimization by simulated annealing, Science, Vol. 220, No.4598, 671-680, 1983
5)Xin-She Yang, Nature-Inspired Metaheuristie Algorithm Second Eddition, Luniver Press, 81-96, 2008
6)小田佳明，本間俊雄，横須賀洋平: 量子的振る舞いを導入した発見的最適化手法による解探索性能，日本建築学会九州支部研究報告，**56**, 241-244, 2017
7)M.Clerc and J.Kennedy, The Particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space. IEEE T. Evolutionary Computation, Vol.6, 58-73, 2002

＊1　鹿児島大学大学院理工学研究科建築学専攻　大学院生
＊2　鹿児島大学大学院理工学研究科建築学専攻　教授・工博
＊3　鹿児島大学大学院理工学研究科建築学専攻　助教・博士(情報科学)