

複数解探索を考慮した分散型 Bat Algorithm

○岩瀬拓哉 高野諒 上野史 梅内祐太 石井晴之 佐藤寛之 高玉圭樹（電気通信大学）

概要 近年、最適解問題に対する大域的探索手法として Bat Algorithm が提案され様々な研究がなされている。しかし、Bat Algorithm は解を更新する際に過去に探索した最良解を参照する探索特性をもつため、最終的に最適解付近に収束してしまい複数の局所解を同時に探索することが困難である。そこで、複数の局所解の同時探索を可能にするため 2 つの修正を加えた分散型 Bat Algorithm を提案する。具体的な修正点は以下の 2 点である。(1) 各個体間の距離を考慮した速度の更新;(2) 新たな解の生成時に各個体の最良解 (パーソナルベスト) に基づく新たな解の生成。複数の局所解をもつ多峰性ベンチマーク問題を用いたシミュレーション実験により提案手法の複数の局所解の網羅的な探索性能の有効性を検証する。その結果、上記の修正を加えることによって従来の BA と比較して分散 BA がより多くの局所解を探索しており、修正が複数の局所解の網羅的な探索に対して有効であることを示した。

キーワード: Meta-Heuristics, Bat Algorithm, Optimization Problem

1 はじめに

Bat Algorithm(BA) はコウモリが獲物や障害物の距離を測定するために用いるエコーロケーションという行動に着想を得て設計された多点探索型アルゴリズムである。¹⁾ BA は他の多点探索アルゴリズムである魚や鳥の群れをモデル化した粒子群最適化 (Particle Swarm Optimization:PSO)²⁾ や光強度によって他の個体を引き寄せるホタルの特性に着想を得た Firefly Algorithm(FA)³⁾ や様々な問題に対応させた (FA Searching Multiple Solutions:MSFA)⁴⁾ などと同様に多峰性を持つ高次元の最適化問題において優れた探索性能を持つことが知られている。BA は解の探索において各個体が過去に探索した最優良解を参照し探索するため、最終的に全個体が最適解付近に収束してしまう。しかし、現実世界の問題への応用を考えたとき、災害時の要救助者の探索において被災者の数や、その状態による優先度を考慮するように、最適解のみではなくその他の優良な局所解の探索が求められる場合がある。他の多点探索アルゴリズムに基づく複数の局所解の探索を考慮した手法として、人工蜂コロニーアルゴリズム (Artificial Bee Colony Algorithm:ABC) に基づいた ABC-lis⁵⁾ などが提案されているが、BA に基づいた複数の局所解の探索を考慮した修正については多くの研究がなされていない。そこで本研究では、従来の BA に対して複数の局所解の網羅的な探索性能向上のために 2 つの修正を加えた分散型 Bat Algorithm(分散 BA) を提案する。具体的な修正点は以下の 2 つである:(1) 各世代において各個体間の距離を考慮した速度の更新;(2) 新たな解の生成時に各個体の最良解 (パーソナルベスト) を参照した解の生成。提案手法である分散 BA の複数の局所解の網羅的な探索性能を検証を目的として、複数の局所解を持つ多峰性ベンチマーク関数を用いたシミュレーション実験により、探索した局所解の個数、最適解付近の探索性能の 2 つの観点に関して、従来手法である BA と比較する。

本論文の構成は次の通りである。まず、2 章において BA の概要とそのアルゴリズムの詳細を説明し、3 章において、複数の局所解の網羅的な探索性能のための修正を加えた分散 BA とその具体的な修正点について説明する。さらに、4 章において多峰性ベンチマーク関数を用いたシミュレーション実験の内容とその結果について記述し、5 章で 4 章で得られた結果について考

察する。そして、最後に 6 章において本論文の結論を述べる。

2 Bat Algorithm

2.1 全体の概要

BA は群知能アルゴリズムの一つで、複数の局所解を持つ目的関数から最適解を見つけることが可能な大域的探索に適したアルゴリズムである。BA は反射波を受信して対象物までの方向や距離を知るコウモリの特性 (エコーロケーション) を利用して周囲の状況を知ることができる。BA において、コウモリは自らの発する超音波の周波数を持ち、その周波数を調整するためのパラメータとしてラウドネス A^0 を用いる。ラウドネス A^0 は、コウモリが対象物に近づくとき値が減少し、移動距離も比例して短くなる。コウモリの行動は以下 3 つの特徴で構成される。

- 各コウモリは、自身が発する周波数 f_i の反響によって対象物との距離を知る。
- コウモリは位置 x_i において速度 v_i で、対象物に近い他のコウモリの方へランダムに移動する。
- コウモリが対象物に近づくにつれ、ラウドネス A^0 を減少させる。

2.2 アルゴリズムについて

BA において、各コウモリの速度 v_i と位置 x_i 、周波数 f_i は以下の式で定義し、更新される。

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (1)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i \quad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

コウモリの周波数 f_i は、コウモリ自身の位置と対象物までの距離を f_{max} で表し、 f_{min} は 0 として設定した。 β は 0 から 1 までの一様乱数を表し、式 (2) は対象物まで移動する度合いを示す。初期個体生成時、全てのコウモリの初期位置 x_i^{t-1} から最良解 x_* までの距離を算出する。各コウモリは最良解 x_* に向かって速度 v_i で x_i まで移動する。また、現在最も優良な解 x_{old} の周辺に新しい解 x_{new} を生成する。生成式は次の通りである。

$$x_{new} = x_{old} + \epsilon A^t \quad (4)$$

ϵ は $[-1, 1]$ 区間でランダムな値が割り当てられ、 A^t 全てのコウモリの平均ラウドネス値を時間ステップごと

に表す。初期生成されたコウモリはラウドネス A_i を発して対象物を探し始め、その反射波をパルスレート r_i で表される。ラウドネスとパルスレートは以下に基づいて更新される。

$$A_i^{t+1} = \alpha A_i^t \quad (5)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (6)$$

ラウドネス A_i とパルスレート r_i は世代数 t 毎に更新される。ラウドネスはコウモリが対象物を発見し、近づくにつれてと徐々に数値を減少させ、その間にパルスレート r_i は増加する機能を持つ。初期のパルスレート r_i^0 は 0 周辺、あるいは $r_i^0 \in [0, 1]$ 区間の値を持つ。式 (4) で生成した新しい解が更新されたとき、式 (5), (6) によって A_i と r_i も更新される。 α と γ は減衰係数を表し、シミュレーション上では $\alpha = \gamma = 0.9$ として使用した。

従来 BA アルゴリズムの疑似コードを Algorithm 1 に記す。Algorithm 1 において、*Max number of iterations* は最大の学習回数、*rand* は生成された乱数を表す。BA では、まず各コウモリに対して (1) 周波数を設定する。そして、(2) 探索速度 v_i^t を設定し、(3) 今までの解からゴールに近づくように、新たな解を作る (6, 7, 8 行目)。次に、乱数を生成し、それが r_i より大きな値であれば (9 行目)、現状の最優良解の付近に新たな解を作る (10 行目)。そして、ランダムに新たな解を生成し (12 行目)、その後乱数がラウドネス A_i^t よりも小さく、解の評価 $f(x_i)$ が現状の最優良解の評価 $f(x_*)$ よりも小さいとき (13 行目)、解を更新し、パルスレート r_i^{t+1} とラウドネス A_i^{t+1} の値を更新する (14, 15, 16 行目)。最後に新たに生成された個体を評価して、最優良解 x_* を更新する (18 行目)。

Algorithm 1: 従来 BA

```

評価関数の設定  $f(x), x = (x_1, x_2, \dots, x_d)^T$ 
個体の初期化  $x_i (i = 1, 2, \dots, n)$  and  $v_i$ 
周波数の定義  $f_i$  at  $x_i$ 
パルスレート  $r_i$  とラウドネス  $A_i$  の初期化
while ( $t < \text{Max number of iterations}$ )
  周波数の調整による新しい解の生成と
  解の位置と速度の更新 [eqs. (1) to (3)]
  if ( $\text{rand} > r_i$ )
    最良解周辺の解を選択
    最良解周辺に新しい解を生成 [eq. (4)]
  end if
  ランダムで新しい解を生成
  if ( $\text{rand} < A_i \& f(x_i) < f(x_*)$ )
    新しい解を更新
    パルスレート  $r_i$  の増加とラウドネス  $A_i$  の減少
    [eqs. to (5)(6)]
  end if
  個体の評価と最適解の更新  $x_*$ 
end while

```

3 分散型 Bat Algorithm

従来の BA は常に過去に探索した最良解を参照しながら探索するため、最終的にすべての個体が最適解の

方へ向かっていくアルゴリズムである。そのため、複数の局所解を網羅的に探索するためには個体間にある程度の距離を保たせ、局所解付近に留まるよう修正を加えなければならない。そこで本研究では、複数の異なる局所解探索のために従来 BA に各個体間の距離に応じて各個体が反発して移動する機構を導入した分散型 BA を提案する。まず、全ての個体が他の個体との距離を参照するために従来の更新式の (2) を式 (7), (8) のように変更した。

$$r_{ij} = |x_i^t - x_j^t| \quad (7)$$

$$v_i^t = v_i^{t-1} + \frac{1}{\exp(r_{ij} * f_i)} \quad (8)$$

式 (7) において $|x_i^t - x_j^t|$ は世代 t における個体 i と個体 j の距離を表す。式 (8) は速度の更新を表す。式 (8) の右辺の第 2 項目によって個体間の距離が近いほど更新する速度を増加させ、距離が遠いほど減速させる。また選択される個体 j はランダムで選択される。さらに、新たに生成される個体が局所解周辺に集中して生成されることを防ぐために、新たな解の生成式である式 (4) を式 (9) のように変更した。

$$x_{new} = x_{i*} + \epsilon A^t \quad (9)$$

x_{i*} は個体 i が過去に探索した最良解を表す。その為、式 (9) によって新たに生成される解は、各個体の最良解の付近に生成される。目的関数を $f(x)$ とした最小化問題における分散型 BA のアルゴリズムの流れを下記に示した。

- **ステップ 1 初期個体の生成**

個体数を n として位置 x_i^0 ($i = 0, 1, 2, \dots, n$) と速度 v_i^0 ($i = 0, 1, 2, \dots, n$) をランダムに生成する。また初期値として各個体にラウドネス A_0 とパルスレート r_i 、周波数 f_i を設定する。

- **ステップ 2 新しい解の生成**

式 (7), 式 (8) により計算された速度に基づき式 (3) によって新しい解 x_i^t を生成する。

- **ステップ 3 生成した解近辺を局所探索**

$\text{rand} > r_i$ を満たした場合、生成した解 x_i^t の周辺に新しく解 x_{area}^t を生成する。

- **ステップ 4 ランダムで新しい解を一つ生成**

イタレーション毎に新しく解 x_{rand}^t を一つ生成する。

- **ステップ 5 解の評価と更新**

$\text{rand} < A_i$ を満たす場合に、解 x_i と x_{area} , x_{rand} の 3 つの中から評価値を良い解を選択する。解の更新をしたら、式 (5)(6) により A_i と r_i も更新する。

- **ステップ 6 ループ**

ステップ 2 へ戻る

分散型 BA のアルゴリズムの疑似コードを Algorithm 2 に示す。

Algorithm 2 において、Algorithm 1 と同様に、*Max number of iterations* は最大の学習回数、

$rand$ は生成された乱数を表す。分散型 BA では、まず各コウモリに対して (1) 周波数を設定する。そして、(2) 探索速度 v_i^t を設定し、(3) 今までの解からゴールに近づくように、新たな解を作る (6, 7, 8, 9 行目)。その際、BA と異なり、 $v_i^t = v_i^{t-1} + \frac{1}{\exp(r_{ij} * f_i)} (r_{ij} = |x_i^t - x_j^t|)$ を用いる。次に、乱数を生成し、それが r_i より大きな値であれば (11 行目)、全体の最優良解付近に解を生成する BA と異なり、各コウモリの最優良解付近に新たな解を作る (12 行目)。そして、BA と同様にして、ランダムに新たな解を生成し (13 行目)、乱数がラウドネス A_i よりも小さく、解の評価 $f(x_i)$ が現状の最優良解の評価 $f(x_*)$ よりも小さいとき (17 行目)、解を更新し、パルスレート r_i^{t+1} とラウドネス A_i^{t+1} の値を更新する (18, 19, 20 行目)。最後に新たに生成された個体を評価して、各コウモリの最優良解 x_{i*} と、最優良解 x_* を更新する (22 行目)。

Algorithm 2: 分散型 BA

```

評価関数の設定  $f(x), x = (x_1, x_2, \dots, x_d)^T$ 
個体の初期化  $x_i (i = 1, 2, \dots, n)$  and  $v_i$ 
周波数の定義  $f_i$  at  $x_i$ 
パルスレート  $r_i$  とラウドネス  $A_i$  の初期化
while ( $t < \text{Max number of iterations}$ )
  周波数の調整による新しい解の生成と
  解の位置と速度の更新
  [eqs. to (1) (3) (7) and (8)]
  for  $i = 1$  to  $n$ 
    if ( $rand > r_i$ )
      パーソナルベスト周辺に新しい解  $x_{area}$  を生成
      [eq. (4)]
    end if
    ランダムに新たな解  $x_{rand}$  を生成
  end for
  for  $i = 1$  to  $n$ 
    if ( $rand < A_i \& f(x_i) < f(x_*)$ )
      新しい解の更新
      パルスレート  $r_i$  の増加と
      ラウドネス  $A_i$  の減少 [eqs. to (5)(6)]
    end if
    個体の評価と  $x_{i*}$  及び  $x_*$  の更新
  end for
end while

```

4 実験

提案手法である分散型 BA を複数の局所解を持つベンチマーク問題を用いて、従来の BA と比較とすることで複数の局所解の網羅的な探索性能を検証する。

4.1 問題設定

目的関数として、複数の局所解を持つ多峰性ベンチマーク問題の一つである Griewank 関数⁶⁾を用いる。式 (10) は実験で用いた目的関数を表す。

$$f(x_1, x_2) = \frac{x_1^2 + x_2^2}{4000} - \cos(x_1) \cos\left(\frac{x_2}{\sqrt{2}}\right) + 1 \quad (10)$$

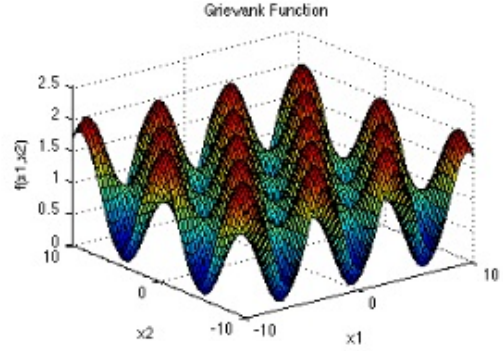


Fig. 1: Griewank 関数の概形

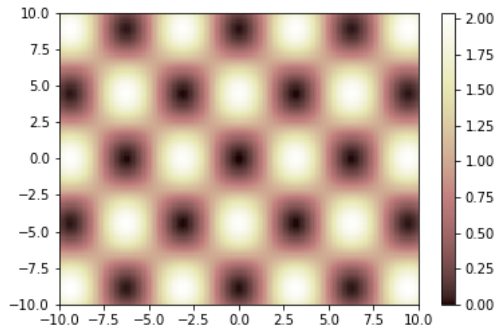


Fig. 2: 評価関数における色濃度別の局所解の位置

また、その関数概形を Fig. 1, 上から見た図を Fig. 2 に示す。Fig. 2 において、横軸は変数 x_1 , 縦軸は変数 x_2 を表し、右側の棒状のグラフは関数の評価値の大小と色の濃度の対応を示す。色が濃いほど評価値が低く、色が薄いほど評価値が高いことを表す。つまり、関数最小化問題であるこの実験において、各個体が Fig. 2 の色の濃い部分に位置していることが複数の局所解を網羅的に探索する能力を持つことを示す。Fig. 2 より、式??において $(x_1, x_2, [-10, 10])$ の範囲では 17 個の局所解が存在する。

4.2 評価基準

本実験において、複数の局所解の網羅的な探索性能を 17 個の局所解をどれだけ探索できたかによって評価する。各局所解の判断基準として、目的関数のグラフが山から谷の形となる境目の評価値が 1 となることを利用して、各個体の評価値の値が 0 ~ 1 (Fig. 2 において色の濃い部分) であればその個体は近傍の局所解を探索していると判定する。各シードにおいて最終世代の解の分布から探索率 (探索できた局所解の個数/すべての局所解の数 (17)) を計算し、10 シード分の探索率の平均により平均探索率として評価する。さらに、探索によって更新された最良解 (x_*) の比較を行うことで最適解に対する探索性能を評価する。その際、10 シードの最終世代の最良解の評価値の平均を比較する。この問題は最小化問題であるため、最終的な最良解の値が小さいほど最適解に対する探索性能が高いことを示す。

4.3 パラメータの設定

局所解の個数 17 個より多い $n = 20$ を個体数とし、各個体のパラメータであるラウドネス $A^0 = 1$ 、周波数 $f_{max} = 1, f_{min} = 0$ 、パルスレート $r^0 \in [0, 1]$ と設定した。また α と γ と従来と同様、 $\alpha = \gamma = 0.9$ と設定し、実験を行った。世代数については複数の世代数で実験した結果十分な収束が確認できた $N = 1000$ とした。また、上記のパラメータは従来 BA と分散型 BA で共通である。

4.4 実験結果

Griewank 関数を目的関数として用い、従来手法と分散型 BA のあるシード ($s = 1$) における結果を Fig. 3 から Fig. 7、各アルゴリズムのシードを 10 回実行し、得られた実験結果を Table 1, 2 に記す。 s はシードを表す。

4.4.1 解探索分布

Griewank 関数を目的関数として用いた従来手法と分散型 BA の各アルゴリズムを 10 回実行し、得られた実験結果を以下に記す。 Fig. 3, 4 は時間推移とともに更新された評価値 x_{i*} である。従来 BA において、Fig. 3 と Fig. 5(a) より 1000 世代目に着目すると、多くの個体が一つの局所解に多く密集しているが、各個体の評価値をみるとばらつきがあることが分かる。それに対し、分散型 BA は Fig. 4 と Fig. 5(b) より評価関数内で個体がまばらに複数の局所解に位置し、評価値は従来 BA と比べて数値が高いものの、多くの個体がほぼ同じ評価値で一定となっている。このことから分散型 BA は局所領域内に入るとその場所に留まりやすく、個体間の距離を保ちながら大域的に複数の局所解の探索していると考えられる。また、従来 BA では 200 世代で解探索が収束してしましたが最良値は低く、分散型 BA では 500 世代目まで探索を行っていたにもかかわらず従来 BA より評価値は高かった。また各手法のアルゴリズムを 10 回実行し、1000 世代目における評価値の平均と標準偏差を Table 1 に示す。 Table 1 から、分散型 BA より従来 BA の方が最適解を探索する性能が高かった。また Table 2 では、すべての局所解 17 個のうち、探索した数の平均とその標準分散、その割合 (探索率=探索した解の数/全局所解の数) を示す。 Table 2 より、従来 BA より分散型 BA の方が探索した局所解の数が 2 倍近く多かった。

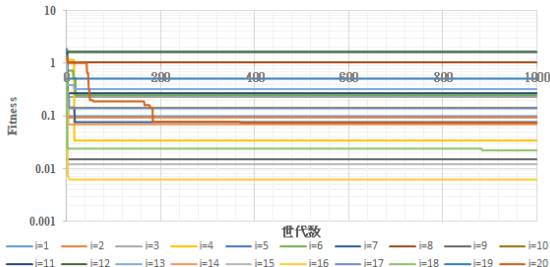


Fig. 3: 従来 BA の評価値 x_{i*}

4.4.2 最良解

Fig. 6 は時間推移とともに更新された全個体の中の最良な評価値 x_* である。縦軸は最良な個体の評価値、横軸は世代数を表し、グラフ上の青い曲線は分散型 BA

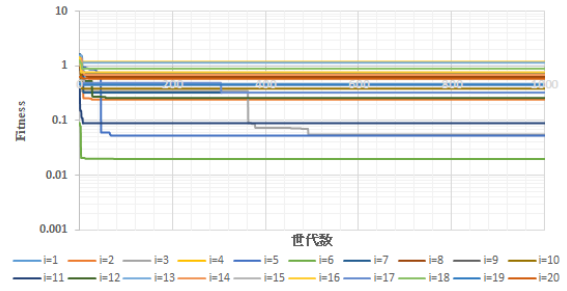
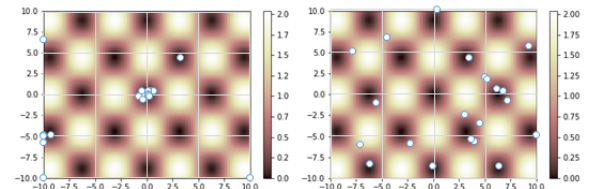


Fig. 4: 分散型 BA の評価値 x_{i*}



(a) 従来 BA (b) 分散型 BA

Fig. 5: 各手法における解の分布 ($t = 1000$)

を、橙色は従来 BA を示す。このグラフから、従来 BA のほうが早い段階で評価値が大きく下がって最適解を探索しており、それに対して分散型 BA は評価値が大きいまま、収束していることが分かる。これは、Fig. 3, 4 の結果を裏付けていることが分かる。1000 世代目における評価値の平均と標準偏差を Table 2 に示す。また Table 2 から、分散型 BA より従来 BA の方が最適解を探索する性能が高かった。また標準偏差から従来 BA より分散型 BA はばらつきが大きく、分散していることが分かった。

5 考察

実験結果 Fig. 5 から、従来 BA より提案した分散型 BA の方が個体が複数の局所解を探索しており、一つの局所解に集中することなく散らばっていることが分かる。これは、従来 BA では各個体が最良解 x_* に向かって移動していたものを、分散型 BA では 2 個体間に距離を持たせることで一つの局所解に陥らず、複数解の探索性能が向上したと考えられる。また新しい解の生成時、最良解 x_* 付近ではなく、各個体の最優良解である x_{i*} 付近に作ることで大域的かつ局所的な探索をすることが可能となったといえる。また、分散型 BA について複数解探索の性能の有効性を確かめるため、評価尺度を世代数、個体数、シードの値として設定した 3 つの観点から分析を行った。

5.1 世代数の影響

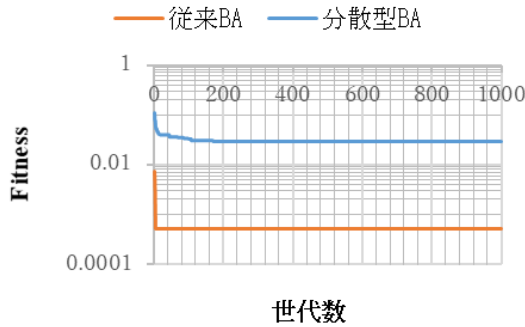
分散型 BA における性能の有効性を確かめるため、評価尺度を世代数として算出した結果を Fig. 7 に表す。 Fig. 7 は $s = 1$ における 200 世代と 1000 世代に着目した解探索数の平均値を表す。 Fig. 8 は縦軸を探索し

Table 1: 各手法の解の探索数

	平均値	平均探索率	標準偏差
従来 BA	6.4	37.65%	1.505545
分散型 BA	12	70.59%	0.942809

Table 2: 1000 世代目における各手法の評価値 x_*

	平均値	標準偏差
従来 BA	0.000532	0.000851
分散型 BA	0.029676	0.014348

Fig. 6: 従来 BA と分散型 BA の評価値 x_*

た局所解として、世代数が $t = 200$ と $t = 1000$ とでは新たに発見した解が 1 つであり、大きな差は見られなかった。

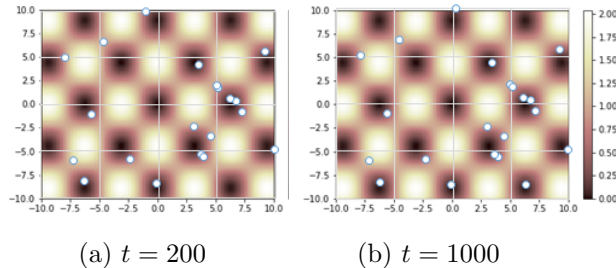


Fig. 7: 提案手法における解の分布

5.2 探索点 (個体数) による効果

Fig. 9 の縦軸を探索した解の数、横軸を個体数 (探索率=探索した局所解/全局所解の数) とし、Fig. 10 (a)(b) は 1000 世代目における個体の位置を示す。個体数の数に比例して探索性能は非常に高くなったが、同時に同じ局所解に陥る解も増加した。複数解探索において個体数の変化による影響は大きかったが、個体数の変化による個体全体の分散に影響は見られなかった。また Fig. 12 と Fig. 13 は探索した個体数別の最良解の時間推移である。このことから、個体数に応じて解探索の精度が変わるということがいえる。上述した個体数に比例して探索性能に影響を与えるということを裏付ける結果が得られた。

5.3 シードによる影響

Fig. 11(a)(b) はあるシードの個体の位置を表し、Fig. 14 はそのシードにおける各個体の評価値を示す。Fig. 4 と比較すると、Fig. 14 のほうが早い段階で収束していることが分かるが、グラフよりあまり大きな変化はないと捉えることができる。しかし、 $s = 1$ では同じ局所解に複数の個体が位置し、 $s = 3$ では個体間の距離が遠くなるよう動作しているとグラフより読み取ることができる。またグラフより局所解に陥る個体の場所は、各個体の持つ評価値に依存し一様に決まらな

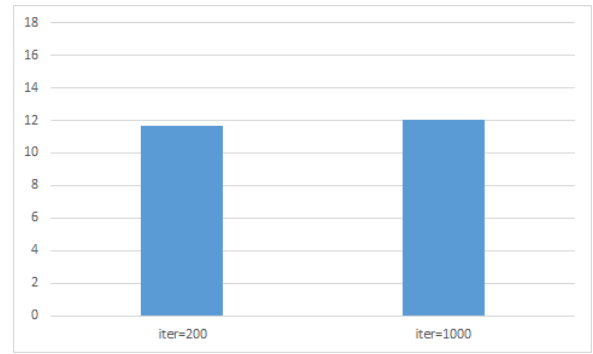


Fig. 8: 探索した局所解数の平均値 (世代数別)

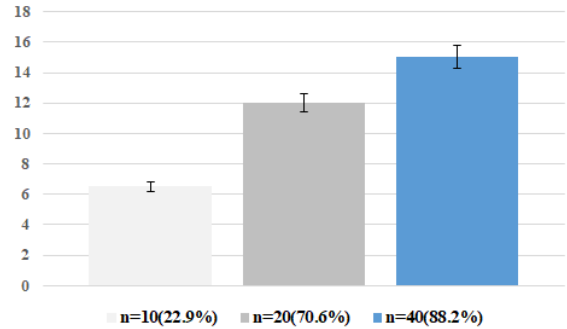
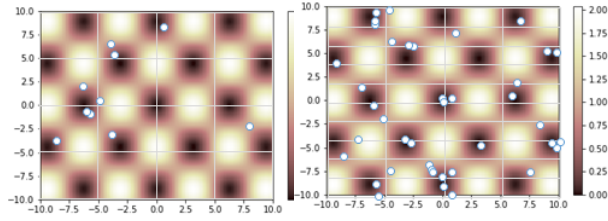


Fig. 9: 探索した局所解数の平均値 (個体数別)

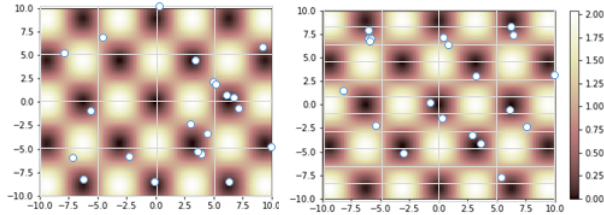
いえる。

6 終わりに

本論文では、従来の BA における最良解を参照することによる全個体の最適解への収束を防ぎ、複数の局所解を網羅的に探索する 2 つの修正を加えた分散型 BA を提案した。具体的に修正した点は以下の 2 つである。(1) 各世代において各個体間の距離を考慮した速度の更新; (2) 新たな解の生成時に各個体の最良解 (パーソナルベスト) を参照した解の生成。提案手法である分散 BA の複数の局所解の網羅的な探索性能を検証を目的として、複数の局所解を持つ多峰性ベンチマーク関数を用いたシミュレーション実験により、探索した局所解の個数、最適解付近の探索性能の 2 つの観点に関して、従来手法である BA と比較した。その結果、分散型 BA の最適解付近の探索性能は従来の BA に比べて低下したが、従来手法に比べて、分散型 BA は複数の局所解の網羅的な探索性能が向上した。その理由として、(1) の修正により各個体同士が離れやすくなり、さらに (2) の修正により各個体が探索した各個体の最良解の付近の局所的な探索が有効に働いたと考えられる。しかし、分散型 BA は多くの局所解を探索できているものの、すべての局所解を探索することはできなかった。また、分散型 BA は早い段階で探索が収束してしまったため、探索を制御するパラメータであるラウドネスとパルスレート調整も検討すべきである。そのため、今後の課題として、すべての局所解を探索可能な機構の導入、最適なパラメータの調査の 2 つを考慮した複数の局所解の網羅的な探索性能向上を目指す。



(a) $n = 10$ (b) $n = 40$
Fig. 10: 分散型 BA の解分布



(c) $s = 1$ (d) $s = 3$
Fig. 11: 分散型 BA の解分布

参考文献

- 1) Yang, X. S. "A Metaheuristic Bat-Inspired Algorithm", in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds J.R. Gonzalez et al.), Studies in Computational Intelligence, Springer Berlin, 284, Springer, 65-74 (2010)
- 2) Eberhart, R. C., and Kennedy, J. : "A New Optimizer Using Particle Swarm Theory", Proc. Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan), IEEE Service center, Piscataway, NJ, 39-43 (1995)
- 3) Yang, X. S. "Firefly Algorithms for Multimodal Optimization", in: Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences, Vol. 5792, pp. 169-178 (2009)
- 4) 上村 昌史 "複数解探索を行う Firefly Algorithm", 電子情報通信学会技術研究報告, 信学技報 113(271), 1-4, NLP2013-69(2013-10), Vol. 5792, pp. 169-178 (2009)
- 5) R., Takano, T. Harada, and K., Takadama. "Artificial bee colony algorithm based on local information sharing in dynamic environment.", Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1. Springer, Cham, (2015).
- 6) Surjanovic, S. and Bingham, D. (2013). Virtual Library of Simulation Experiments: Test Functions and Datasets, Retrieved October 9, (2017)

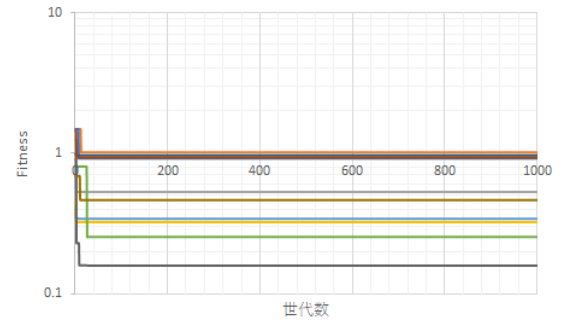


Fig. 12: 分散型 BA の評価値 ($n = 10$)

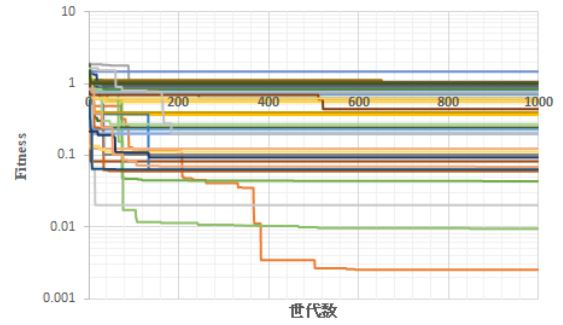


Fig. 13: 分散型 BA の評価値 ($n = 40$)

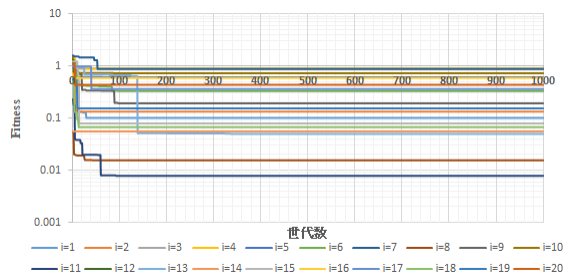


Fig. 14: 分散型 BA の評価値 x_{i*} ($s=3$)