

領域適応型Differential Evolutionの提案

著者	小貫 涼介, 北山 哲士, 山崎 光悦, 荒川 雅生
著者別表示	Onuki Ryosuke, Kitayama Satoshi, Yamazaki Koetsu, Arakawa Masao
雑誌名	日本機械学会論文集. C編 = Transactions of the Japan society of mechanical engineers. C
巻	79
号	798
ページ	429-441
発行年	2013-01-01
URL	http://doi.org/10.24517/00008801

doi: 10.1299/kikaic.79.429



領域適応型 Differential Evolution の提案 *

小貫涼介^{*1}, 北山哲士^{*2}, 山崎光悦^{*3}, 荒川雅生^{*4}

Proposal of Adaptive Range Differential Evolution

Ryosuke ONUKI^{*1}, Satoshi KITAYAMA, Koetsu YAMAZAKI and Masao ARAKAWA

^{*1}Graduate School of Natural & Science Technology, Kanazawa University,
Kakuma-machi, Kanazawa, 920-1192 Japan

This paper proposes a new method called Adaptive Range Differential Evolution (ARDE). The basic idea can be obtained from the Adaptive Range Particle Swarm Optimization (ARPSO). In the ARDE, the active search domain range consists of the mean and the standard deviation of each design variable. Also the best position is included in this search range. Therefore, the active search domain range is newly defined by utilizing the mean and the standard deviation of each design variable, and the best position. The detailed procedure of the new active search domain range is described in this paper. To keep the best position for the new active search domain range will lead to the wide search range. Of course, the newly active search domain range will shrink through the search iteration. As the result, a highly accurate global minimum can be found. The effectiveness and validity of the ARDE are examined through typical benchmark problems. It could be found through benchmark problems that the ARDE can find a global minimum with the small number of function evaluations in comparison with basic Differential Evolution.

Key Words : Global Optimization, Adaptive Range Differential Evolution, Optimal Design, System Engineering, Engineering Optimization

1. 結 言

Differential Evolution (DE)は連続型多峰性関数の大域的最適解を見つけるために、1995年に Storn と Price によって提案された関数の感度 (勾配) を利用せず大域的最適解を求める多点同時探索法である^(1,2)。同年に Particle Swarm Optimization (PSO)も提案されており⁽³⁾、遺伝的アルゴリズム(GA)や Ant Colony Optimization (ACO)と同様、進化計算と称される最適化手法に分類される。PSOは p-best と g-best の線形結合で新しい探索点を生成するのに対し、DEは2章で述べるようにランダムに3つの探索点を選び、突然変異や交叉といった操作を通じて、新しい探索点を生成する。PSOは勾配法との類似構造を持っていたり、次の探索点を生成する近傍が明確であること⁽⁴⁾、また力学系と見なすことで、速度の更新式のパラメータ設定範囲を絞り込むなどの研究が盛んに行われている⁽⁵⁾。一方でDEは、アルゴリズム内に用いられるパラメータが探索性能に与える影響についての研究が行われているが^(6,7)、最適化手法としての位置づけが必ずしも明確になっておらず、PSOほど活発に研究がされているとは言い難い。筆者らはDEの探索性能の高さに注目し、最適化手法としての位置づけ、特に勾配法やPSOなどとの類似性について研究を行ったり⁽⁸⁾、離散変数問題へ適用するなどの研究を行っている⁽⁹⁾。また、スケジューリング問題へ適用する研究や^(10,11)、安定性解析を行うことで収束性を論じる研究もある⁽¹²⁾。

一般に進化計算と称される多点同時探索型最適化手法では、はじめに探索領域を固定して大域的最適解の探索

* 原稿受付

^{*1} 学生員, 金沢大学大学院自然科学研究科 (〒920-1192 石川県金沢市角間町).

^{*2} 正員, 金沢大学理工研究域.

^{*3} 正員, フェロー, 金沢大学理工研究域.

^{*4} 正員, 香川大学工学部 (〒761-0396 高松市林町 2217-20).

E-mail: fkm1.02@stu.kanazawa-u.ac.jp

を行うため、初期探索領域の設定は極めて重要である。筆者らは探索点のばらつき状況に応じて探索領域を更新する領域適応型 PSO(Adaptive Range PSO: ARPSO)を提案し、数値計算を通じ、その有効性を検討した⁽¹³⁾。ARPSOでは、探索点をもつ各設計変数の平均値と標準偏差、g-best を利用し、正規分布を基調とした有効探索領域と称される探索領域を生成する。探索点がばらついていれば有効探索領域は拡大し、一方で探索点の集中化が起きれば有効探索領域は縮小する。これらを繰り返すことで、最終的に精度の高い大域的最適解を得る方法であった。

この手法の中では、最良値の保存という操作を行うことで常に g-best を有効探索領域内に残す操作を行う。そのため、正規分布の形を強制的に変化させながら探索を続けるため、どこか不自然さを含むものであった。最良点 (g-best) が重要な意味を持つのであれば、強制的に正規分布の形を変形するのではなく、有効探索領域に用いる正規分布の形を保持したまま有効探索領域を生成するほうが望ましい。また、PSO では探索点が常に g-best へ向かうという性質があり、探索点が g-best へ向かう過程で目的関数値を改善する点が g-best に置き換わるため、内挿領域を集中探索している方法と捉えることができる。一方、DE では、3つの探索点をランダムに選択するが、2つの探索点が探索方向ベクトルを生成するため⁽⁸⁾、PSO のように常に内挿領域に探索方向ベクトルが向くのではなく、場合によっては外挿領域へ向かうこともあり、その結果、未知探索領域の探索を行うことになるので大域的最適解も見つけることが期待できる。このようなことを考えれば、ARPSO のように、ある意味で強引に正規分布の形を崩すのではなく、正規分布の形を維持するような有効探索領域の設定を行うことが自然であろう。そのためには、最良点の各設計変数を中心とした有効探索領域と、探索点の各設計変数の平均値を中心とした有効探索領域をそれぞれ設定し、それらを合成した領域を有効探索領域とすればよい。また ARPSO で提案された探索領域では、領域を強制的に縮小させるパラメータが導入されている。一般に最適化問題の大域的最適解は未知であるため、有効探索領域を強制的に縮小すると、場合によっては初期収束により局所的最適解の発見に留まることも考えられる。

上述のような ARPSO の不自然さやパラメータ設定の問題を踏まえ、本論文では、探索領域を探索状況に応じて適応的に変化させる領域適応型 Differential Evolution (Adaptive Range Differential Evolution: ARDE)を提案する。ARDE で導入する有効探索領域は、3章で述べるように、各設計変数の平均と標準偏差、そして最良点を利用して決定される。これにより、ARPSO で導入していた最良点の保存といった不自然さが解消される。単に領域の取り方を変更しただけであり、他は ARPSO と同じではないかと思われるかもしれないが、PSO の基本的思想は、既に述べたように力学系に基づく内挿領域の集中探索であり、一方で DE では、ランダム性を基調とした外挿領域の探索も可能な多点同時探索法である。これら両者を結びつけ、さらに ARPSO で不自然と考えられている正規分布の強制変化をなくし、またパラメータによる強制的な有効探索領域の縮小をなくし、外挿領域への探索を期待した方法が ARDE である。代表的な計算例を通じて、ARDE の有効性を検討する。

2. Differential Evolution

DE では様々なモデルが提唱されているが、本論文では最も基本的な DE/1/rand/bin を対象とする。DE では、DE/X/Y/Z と表記し、X は基底ベクトルの数、Y は親となる探索点の選択方法、Z は交叉方法をそれぞれ意味している。本論文で対象とする DE/1/rand/bin とは、基底ベクトルの数が 1 で、親となる探索点をランダムに選択し、交叉方法は一様 (Binomial) 交叉を用いるということを意味している。

2・1 基本アルゴリズム

k 回目の探索における d 番目の探索点を \mathbf{x}_d^k と表記し、 n を設計変数の数とする。

(STEP1) 探索点数 d_{max} と最大探索回数 k_{max} を設定。初期探索集団をランダムに生成。突然変異確率 F 、交叉確率 Cr を設定。探索回数を $k=1$ とする。

(STEP2) 全探索点に対し、以下の操作を繰り返す。

(STEP2-1) d 番目の探索点 \mathbf{x}_d^k が、ランダムに 3つの探索点 \mathbf{x}_{r1}^k 、 \mathbf{x}_{r2}^k 、 \mathbf{x}_{r3}^k を選ぶ。ただし $d \neq r1 \neq r2 \neq r3$ 。

(STEP2-2) 突然変異によって、新たな探索点 \mathbf{v}_d^k を以下の式により生成する。

$$\mathbf{v}_d^k = \mathbf{x}_{r1}^k + F(\mathbf{x}_{r2}^k - \mathbf{x}_{r3}^k) \quad (1)$$

(STEP2-3) 探索点 \mathbf{x}_d^k と \mathbf{v}_d^k が交叉確率 Cr によって交叉し, 新たな点 \mathbf{u}_d^k を生成する.

(STEP2-4) 目的関数 $f(\mathbf{x})$ を評価し, 探索点の位置を更新.

$$f(\mathbf{u}_d^k) \leq f(\mathbf{x}_d^k) \rightarrow \mathbf{x}_d^k = \mathbf{u}_d^k \quad (2)$$

$$f(\mathbf{u}_d^k) > f(\mathbf{x}_d^k) \rightarrow \mathbf{x}_d^k = \mathbf{x}_d^k \quad (3)$$

(STEP3) 探索回数を更新 ($k = k + 1$).

(STEP4) 終了条件を満足していれば, 終了. そうでなければ(STEP2)へ戻る.

2・2 交叉

DE の交叉について, 図 1 を用いて説明する. ここで \mathbf{x}_d^k , \mathbf{v}_d^k それぞれの成分を $x_{d,i}^k$, $v_{d,i}^k$ ($i = 1, 2, \dots, n$) と表記する. 交叉による \mathbf{u}_d^k の生成を図 1 に示す. 図 1 中の交叉点 (Crossover point) は, ランダムに選ばれる. また r は設計変数ごとに発生させる $[0,1)$ の乱数である. はじめに交叉点の設計変数の成分が \mathbf{v}_d^k から引き継がれ, 新たな点 \mathbf{u}_d^k の成分となる. 次に \mathbf{u}_d^k の他の成分を決定するため, 交叉点以外の設計変数に対して成分毎に乱数 r を発生させ, \mathbf{x}_d^k , \mathbf{v}_d^k から引き継いで残りの成分を決定していく.

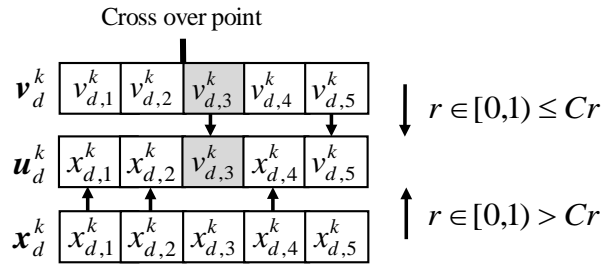


Fig.1 Crossover in DE

3. 領域適応型 Differential Evolution

本論文で提案する ARDE の探索領域は, ARPSO を参考にして設定する. はじめに ARPSO で提案された探索領域の基本的な考え方を示し, 次に ARDE の探索領域の設定指針や特徴について述べる.

3・1 領域設定の基本方針

ARPSO の探索領域の設定方法を示す. PSO は, 各探索点が位置と速度を持ち, それらを更新して最適解を求める方法である. k 回目の探索における d 番目の探索点の位置を \mathbf{x}_d^k , 速度を \mathbf{v}_d^k とすると, $k + 1$ 回目の探索における位置 \mathbf{x}_d^{k+1} と速度 \mathbf{v}_d^{k+1} はそれぞれ以下の式によって表される.

$$\mathbf{x}_d^{k+1} = \mathbf{x}_d^k + \mathbf{v}_d^{k+1} \quad (4)$$

$$\mathbf{v}_d^{k+1} = w\mathbf{v}_d^k + c_1r_1(\mathbf{p}_d^k - \mathbf{x}_d^k) + c_2r_2(\mathbf{p}_g^k - \mathbf{x}_d^k) \quad (5)$$

式 (5) において, w は慣性項と呼ばれる係数, c_1 , c_2 は定数であり, r_1 と r_2 は $[0,1)$ の乱数, \mathbf{p}_d^k は探索点 d の k 回目までの最良点 (p-best), \mathbf{p}_g^k は k 回目の探索における探索点全体の最良点 (g-best) を表す.

ARPSO における有効探索領域は, 式(6)で示される正規分布を基調とし, 式(7)によって与えられる. ここで i 番目の設計変数を x_i とし, i 番目の設計変数における各探索点の位置に対する標準偏差を σ_i , 平均を μ_i とする.

$$N(x_i) = \exp\left(-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right) \quad (6)$$

$$\mu_i - \sqrt{-2\sigma_i \log a_i} \leq x_i \leq \mu_i + \sqrt{-2\sigma_i \log a_i} \quad (7)$$

図2に ARPSO の有効探索領域を示す. ARDE でも式 (7) を基調とした有効探索領域を考えるが, 式 (7) は有効探索領域を設定する際の基本的な考え方を示すものであり, ARDE で探索に用いる領域については次節以降で述べる.

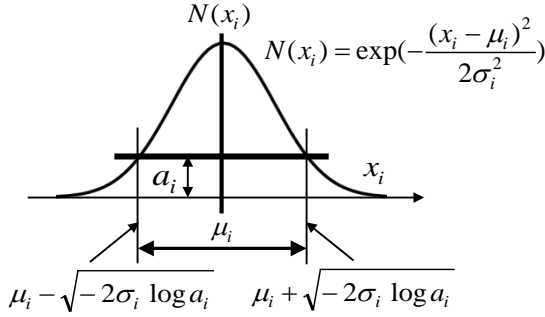


Fig.2 The active search domain of i -th design variable

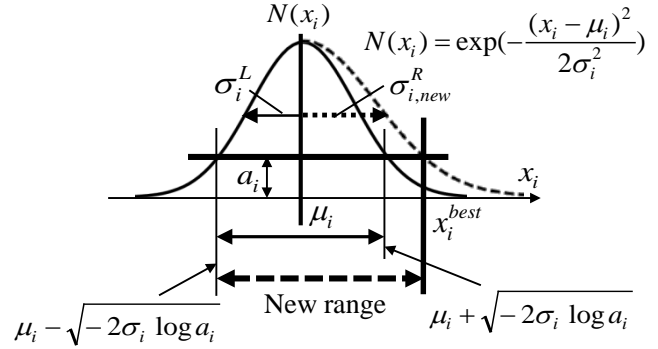


Fig.3 New range of ARPSO by changing the standard deviation

3・2 ARDE における有効探索領域

ARPSO では, さらに最良値の保存という操作を行い, 過去の探索において最良値をとった探索点 \mathbf{p}_g の座標が探索領域内に必ず含まれるようにしていた. そのため強制的に正規分布の形状を変更することで有効探索領域を更新して探索を行うため, 統計的要素を含むとは言え, どこか不自然さが残っていた. 図3に ARPSO で最良点座標が式 (7) の右側に位置した時に, 最良値の保存を行った場合の有効探索領域を示す. ここで x_i^{best} は \mathbf{p}_g の i 番目の成分であり, $\sigma_{i,new}^R$ は式 (6) の x_i を x_i^{best} に, $N(x_i)$ を a_i に置き換えて σ_i について解くと得られる値で, σ_i^L は σ_i である. ARDE における有効探索領域は, 以下のようにして生成する.

全探索点の中で目的関数を最良にした探索点 \mathbf{x}^{best} の i 番目の成分を x_i^{best} とし, 最良点を中心とした領域を作成する.

$$x_i^{best} - \sqrt{-2\sigma_i \log a_i} \leq x_i \leq x_i^{best} + \sqrt{-2\sigma_i \log a_i} \quad (8)$$

そして式 (7) と式 (8) それぞれの上下限を用いて有効探索領域を決定する. ARDE で最終的に探索に用いる領域は式 (11) である.

$$x_i^{Right} = \max\left\{\mu_i + \sqrt{-2\sigma_i \log a_i}, x_i^{best} + \sqrt{-2\sigma_i \log a_i}\right\} \quad (9)$$

$$x_i^{Left} = \min\left\{\mu_i - \sqrt{-2\sigma_i \log a_i}, x_i^{best} - \sqrt{-2\sigma_i \log a_i}\right\} \quad (10)$$

$$x_i^{Left} \leq x_i \leq x_i^{Right} \quad (11)$$

3・3 有効探索領域の性質と ARPSO との相違

式 (11) で与えられる有効探索領域の一例を図4に示す. 破線の曲線が各設計変数の平均値, 最良点を中心とした正規分布を, 実線の曲線がそれら2つの曲線を重ね合わせたものを表している. 図4(a)の場合, 実線部は大きな正規分布のような形状となり, 巨大な正規分布から有効探索領域を決定していると見なすことができる. 本来正規分布は平均を中心左右対称であり, 基準となる値を中心としたばらつきを表すものである. このようなことを踏まえれば不自然にこの形状を変更する ARPSO より, 本来の正規分布に基づく左右対称な一つの巨大な正

規分布から有効探索領域を決定している ARDE の有効探索領域の方が考え方が自然である。ただし図 4(b)にあるように、実線部の曲線は必ずしも正規分布の形状をとらないことに注意されたい。平均値と最良点の位置が大きく離れている場合、重ね合わせた曲線は正規分布の形状とはならず中央がくぼんだ形状となる。その場合、有効探索領域が巨大な正規分布から決定されるという意味は薄れ、広い有効探索領域を用いて探索の多様性が高まるという意味合いが強くなる。

また図 4 には ARPSO で最良値の保存を行った場合の有効探索領域も示されている。図 4 から明らかなように、ARPSO の有効探索領域より ARDE の有効探索領域の方が有効探索領域を広くとることができる。有効探索領域が広い場合、探索点の移動範囲が増加するため有効探索領域は探索の多様性が高くなる。多様性の高い有効探索領域は局所的最適解に捕らわれにくい幅広い探索を可能とし、大域的最適解の発見が期待できる。探索初期は探索点の位置がばらつき、正規分布のすそ野が広く、最良点と平均値が離れていることが予想される。探索初期で探索の多様性が求められる場合に有効探索領域は図 4(a)のようになることが予想され、ARDE の方が ARPSO よりも、大域的最適解を発見しやすい有効探索領域になることが期待される。また最良点が局所的最適解から離れた位置に発見された場合、有効探索領域は図 4(b)のようになることが予想される。この場合局所的最適解付近を集中的に探索するよりも、新たに発見された最良点付近の探索も考慮して有効探索領域を広げたほうが大域的最適解の発見が期待できる。このような状況でも ARDE は ARPSO より多様性のある効果的な有効探索領域を設定することができる。このように ARDE では最良点を有効探索領域に残すという ARPSO の最良値の保存の設定指針を常に含んだまま、探索の多様性が特に求められる状況で有効探索領域を ARPSO より多様性のあるものにする事が期待できる。

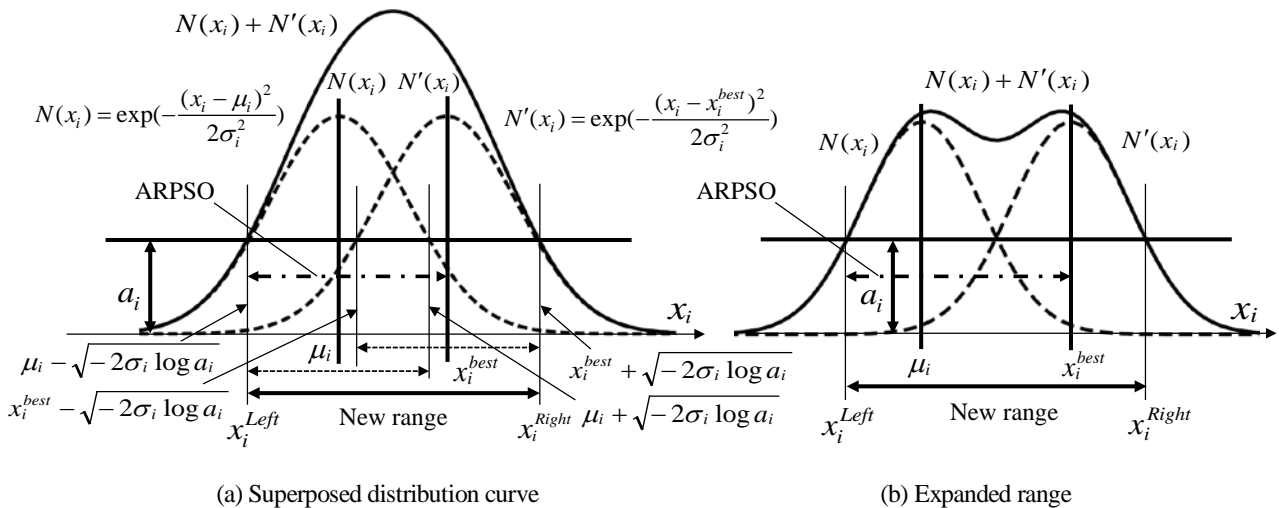


Fig.4 Active search domain range of ARDE

3・4 有効探索領域の確保

探索終盤に探索点が集中して探索領域が極端に小さくなるのを防ぐために、各設計変数の標準偏差の最小値を、側面制約条件を利用して式 (12) のように決定する。

$$\sigma_{i,\min} = \varepsilon_i (x_i^U - x_i^L) \tag{12}$$

ここで ε_i は $[0, 1]$ の値である。また x_i^U 、 x_i^L は側面制約条件の上限值、下限値である。

3・5 有効探索領域の多様性と集中化

式 (11) では常に最良点と探索点の平均値を含むため、探索初期では図 5(a)に示すような広い探索領域となる。探索が進むにつれ、探索点の集中化が進めば探索点の平均値と最良点は近づくことが期待され、図 5(b)のように

徐々に有効探索領域は小さくなる。最終的には図 5(c)のように探索点の平均値と最良点はほぼ一致して、有効探索領域は十分小さくなる。ARDE は、DE がランダム性を強く意識した方法であるということを踏まえて、正規分布の形状を維持しながら探索を行う方法であり、図 5 に示したように有効探索領域は幅広くとることができる。探索が進んでも、探索点がばらついていれば、有効探索領域は広いま探索を続けることになるため、大域的最適解の発見確率が高く、さらに最終的に得られる最適解の精度も十分高いことが期待できる。

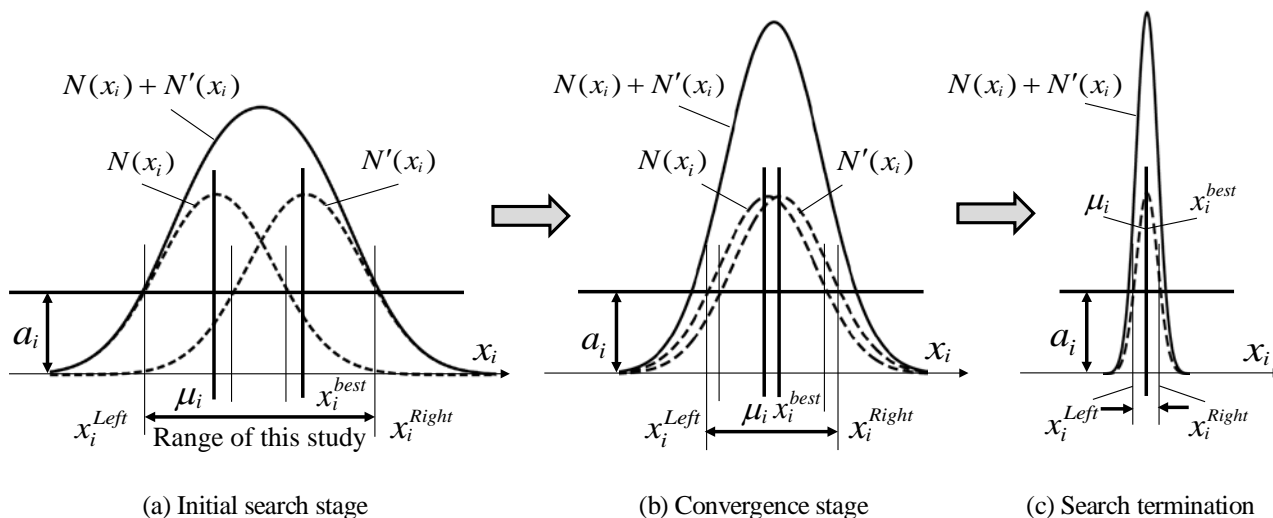


Fig.5 Shrinkage of active search domain

4. アルゴリズム

提案する手法のアルゴリズムを示す。ARDE では 2 章で述べた DE の基本的なアルゴリズムにおける(STEP1)～(STEP2-4)および(STEP3), (STEP4)が同じであるため、変更を加える(STEP2-4)以降について示す。ただし(STEP1)においては 2 章の設定パラメータに加えて、新たに a_i と ε_i を設定する。そして以下に示す(STEP2-6)が終了したら 2 章の(STEP3)へ移動する。

(STEP2-5) 各設計変数の標準偏差 σ_i と平均 μ_i を計算し、 x_i^{best} を求める。 σ_i が $\sigma_{i,\min}$ を下回っているのなら置換する。

$$\sigma_i < \sigma_{i,\min} \Rightarrow \sigma_i = \sigma_{i,\min} \quad (13)$$

(STEP2-6) 設計変数毎に、式 (7), (8) から平均を中心とした領域、最良点を中心とした領域の上下限値を求め、それら 4 つの値から有効探索領域の幅が最大となるように式 (9), (10) を用いて有効探索領域の上限値・下限値を決定し、式 (11) のように有効探索領域を設定する。

5. 数値計算

代表的な数値計算例を通じて本論文で提案する手法の有効性を検討する。本論文では有効探索領域の変更が基本的な DE にもたらす有効性を確認するため、ARDE と有効探索領域の変更を行わない通常の DE で数値計算を行った。また本論文では参考とした ARPSO の数値計算結果との比較は極力行わなかった。これははじめに述べたように、PSO が力学系に基づく内挿領域の集中探索であり、一方で DE はランダム性を基調とした外挿領域の探索も可能な多点同時探索法であるため、ARPSO と ARDE では基本的な探索の性質が異なるためである。

すべての数値計算例で、突然変異率 $F=0.7$ 、交叉率 $Cr=0.5$ とし、探索領域の作成で必要となるパラメータは、 $\varepsilon_i=0.01$ （側面制約の1%）として、乱数の種を変え10回試行した。なお以下では標準偏差（Standard Deviation）をSDと略記する。

5・1 有効探索領域の変化

有効探索領域の変化を可視化するため、式(14)、(15)で与えられる Schwefel 関数を用いる。関数の様子と等高線を図6に示す。

$$f(\mathbf{x}) = 418.9829n + \sum_{i=1}^n \left\{ -x_i \sin \sqrt{|x_i|} \right\} \rightarrow \min \tag{14}$$

$$-500 \leq x \leq 500 \tag{15}$$

大域的最適解は $x_{i,G} = 420.9687$ ($i=1,2,\dots,n$) であり、そのときの目的関数値は0である。多くの局所的最適解が含まれており、ARDEのように初期設定領域にとらわれずに外挿領域の探索をも意識した方法であれば、局所的最適解にとらわれずに大域的最適解の発見が期待できる。有効探索領域の可視化のため、設計変数の数は2とし、探索点数を20、最大探索回数を100とした時、目的関数の履歴を図7に、有効探索領域の変化の一例を図8に示す。また数値計算の結果を表1に示す。

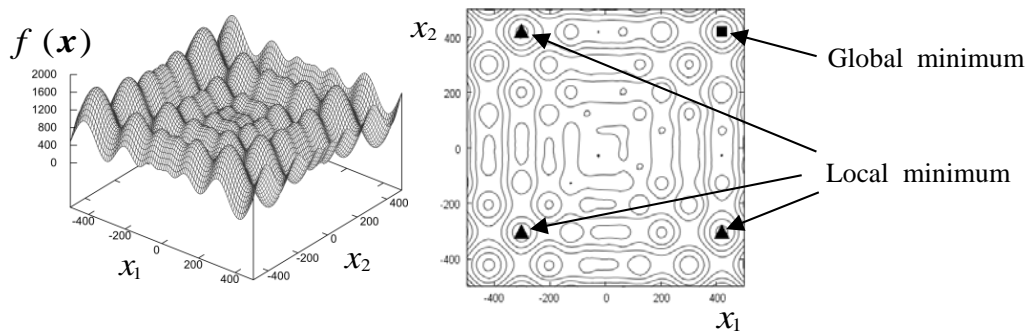


Fig.6 The behavior and contour of objective function

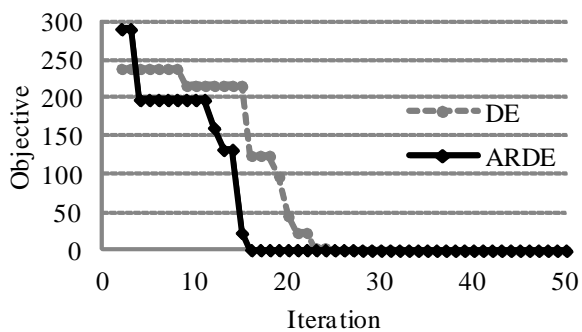


Fig.7 Convergence of objective function

Table 1 Result of Schwefel function

	ARDE	DE
Best objective	2.55E-05	2.55E-05
Worst objective	2.55E-05	2.55E-05
Mean value of objective	2.55E-05	2.55E-05
SD of objective	6.74E-09	8.33E-09
Average of function call	1528	1588

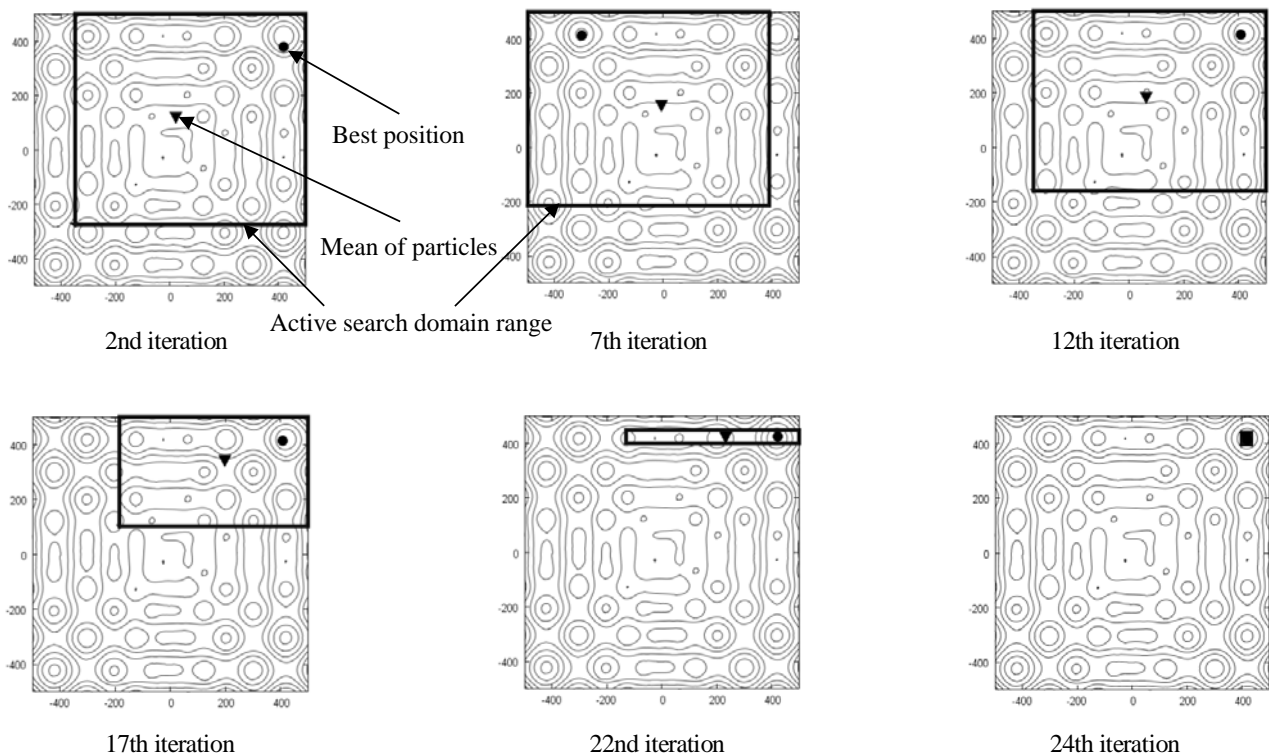


Fig.8 Change of search domain through search process

図7から、ARDEは探索が進むと早い段階でDEよりも良い最良値を求めることができている。図8では、探索が進むにつれて最良点が移動しながら大域的最適解を発見する様子が見て取れる。有効探索領域は初期の探索回では広いままであったが、その後大域的最適解を発見してから、探索点の座標平均値と最良点座標が近付いて縮小していく様子が見られる。

5・2 無制約最適化問題

無制約最適化問題の代表的なベンチマーク問題を考える。(用いたベンチマーク問題は付録に記載)。探索点数を30点、最大探索回数 $k_{max} = 500$ とした。探索性能を比較するため、有効探索領域を操作しない通常のDEで試行した結果と合わせて数値計算の結果を表2~5に示す。

これらの結果から、本論文で提案する手法は通常のDEよりも少ないファンクションコール(目的関数値の計算回数)で大域的最適解を求めることができていることがわかる。また目的関数値の平均、標準偏差も小さな値となっているため、安定的に大域的最適解の探索が行われていることがわかる。本節で用いたベンチマーク問題は多峰性が激しかったり、大域的最適解と局所的最適解の位置が離れていたり、各設計変数で大域的最適解が異なる値をとったりするなどの特徴を有するものである。このような問題でも大域的最適解が見つかったのは有効探索領域が探索状況に合わせて適切に行われて、探索の多様性を維持したまま探索が行われているためと考えられる。

Table 2 Result of 2ⁿ minima function

	ARDE	DE
Best objective	-391.661657	-391.661657
Worst objective	-391.661654	-391.661654
Mean value of objective	-391.661656	-391.661655
SD of objective	5.42E-07	6.75E-06
Average of function call	8250	10611

Table 3 Result of Griewank function

	ARDE	DE
Best objective	3.95E-09	4.23E-09
Worst objective	1.93E-08	2.16E-08
Mean value of objective	1.05E-08	1.19E-08
SD of objective	3.54E-09	4.03E-09
Average of function call	8091	10242

Table 4 Result of Ackley function

	ARDE	DE
Best objective	1.71E-07	2.97E-07
Worst objective	2.54E-06	2.66E-06
Mean value of objective	1.29E-06	1.46E-06
SD of objective	5.69E-07	5.82E-07
Average of function call	12633	14082

Table 5 Result of Michalewics function

	ARDE	DE
Best objective	-4.68766	-4.68766
Worst objective	-4.68765	-4.68765
Mean value of objective	-4.68766	-4.68766
SD of objective	1.86E-06	1.90E-06
Average of function call	4029	4311

5・3 コイルばねの重量最小化設計

文献 (14) で取り上げられているコイルばねの重量最小化設計問題を考える. この問題は, 探索効率や最適解の精度を検討するために多点同時探索型最適化手法でしばしば用いられる問題である^(13,15,16). 設計変数はワイヤの直径 $d(=x_1)$, コイルの平均直径 $D(=x_2)$, コイルの巻数 $N(=x_3)$ であり, すべて連続変数である. 最適設計問題は以下の式 (16) ~ (23) のように定式化される. 探索点数を 20 とし, 最大探索回数を 500 として計算したときの結果を, 他の研究報告と併せ, 表 6 に示す. なお, 制約条件はペナルティ関数として扱い, 文献 (13) の方法を採用した.

$$f(\mathbf{x}) = (2 + x_3)x_1^2x_2 \rightarrow \min \quad (16)$$

$$g_1(\mathbf{x}) = 1 - x_2^3x_3 / (71785x_1^4) \leq 0 \quad (17)$$

$$g_2(\mathbf{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_3^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \quad (18)$$

$$g_3(\mathbf{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \quad (19)$$

$$g_4(\mathbf{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \quad (20)$$

$$0.05 \leq x_1 \leq 2.00 \quad (21)$$

$$0.25 \leq x_2 \leq 1.30 \quad (22)$$

$$2.00 \leq x_3 \leq 15.0 \quad (23)$$

Table 6 Comparison of results of minimum weight design of tension/compression spring

Design Variables	Best solutions found					
	Arora ⁽¹⁴⁾	Coello ⁽¹⁵⁾	Ray ⁽¹⁶⁾	ARPSO ⁽¹³⁾	DE	This study
$x_1(d)$	0.053396	0.05148	0.050417	0.051679	0.051668	0.051638
$x_2(D)$	0.039918	0.351661	0.321532	0.356477	0.35620	0.35549
$x_3(N)$	9.1854	11.632201	13.979915	11.299395	11.320	11.362
$g_1(\mathbf{x})$	0.000019	-0.00208	-0.001926	-0.000037	-1.01E-02	-1.06E-06
$g_2(\mathbf{x})$	-0.000018	-0.00011	-0.012944	-0.000008	-5.29E-06	-6.53E-06
$g_3(\mathbf{x})$	-4.123832	-4.026318	-3.89943	-4.054976	-4.0527	-4.0513
$g_4(\mathbf{x})$	-0.698283	-0.731239	-0.752034	-0.727895	-0.72809	-0.72858
$f(\mathbf{x})$	0.01273	0.012705	0.01306	0.012661	0.012665	0.012665
Average of $f(\mathbf{x})$	N/A	0.012769	0.013436	0.012675	0.012666	0.012666
Worst of $f(\mathbf{x})$	N/A	0.012822	0.01358	0.012696	0.012668	0.012667
SD of $f(\mathbf{x})$	N/A	3.94E-05	N/A	1.17E-05	4.75E-07	4.00E-07
Function call	N/A	900000	1291	5804	6880	5900

表 6 より ARDE では少ないファンクションコールで効率的に大域的最適解を求めることができている。標準偏差と平均値も小さいため、ARDE はロバスト性が高いアルゴリズムであると考えられる。

5・4 溶接梁のコスト最小化問題

文献 (17) にある溶接梁のコスト最小化問題を考える。図 9 に対象とする溶接梁を示す。目的関数は式 (24) にあるように溶接梁の重量に単位重量当たりのコスト係数 c_1 , c_2 をかけたものである。設計変数は溶接部の寸法 $h(=x_1)$, 溶接する長さ $l(=x_2)$, はりの断面寸法 $t(=x_3)$, $b(=x_4)$ の 4 つである。また側面制約条件は式 (25), (26) である。制約条件は、はりのせん断応力や曲げ応力、たわみ、座屈等である。それらを式 (27) ~ (41) に示す。ここで $c_1 = 67413.5[\$/\text{m}^3]$, $c_2 = 2935.85[\$/\text{m}^3]$, $L = 357[\text{mm}]$, $\tau_{\max} = 93.769[\text{MPa}]$, $\sigma_{\max} = 206.84[\text{MPa}]$, $\delta_{\max} = 6.35[\text{mm}]$, $P = 26.689[\text{kN}]$, $E = 206.84[\text{GPa}]$, $G = 82.737[\text{GPa}]$ である。探索点数を 20, 最大探索回数を 500 として DE と ARDE で計算した結果を、他の文献に記された結果⁽¹⁸⁻²⁰⁾と併せて表 7 に示す。

表 7 より ARDE はファンクションコールが少なく、探索が効率的であることが分かる。前節のコイルばねの重量最小化問題の結果と合わせて考えると、ARDE は有制約最適化問題に対しても有効であると考えられる。

$$f(\mathbf{x}) = c_1 x_1^2 x_2 + c_2 x_3 x_4 (L + x_2) \quad (24)$$

$$2.54 \leq x_i \leq 50.8 \quad (i=1,4) \quad (25)$$

$$2.54 \leq x_i \leq 254 \quad (i=2,3) \quad (26)$$

$$g_1(\mathbf{x}) = \tau(\mathbf{x}) - \tau_{\max} \leq 0 \quad (27)$$

$$g_2(\mathbf{x}) = \sigma(\mathbf{x}) - \sigma_{\max} \leq 0 \quad (28)$$

$$g_3(\mathbf{x}) = x_1 - x_4 \leq 0 \quad (29)$$

$$g_4(\mathbf{x}) = 3.175 - x_1 \leq 0 \quad (30)$$

$$g_5(\mathbf{x}) = \delta(\mathbf{x}) - \delta_{\max} \leq 0 \quad (31)$$

$$g_6(\mathbf{x}) = P - P_c(\mathbf{x}) \leq 0 \quad (32)$$

$$\tau(\mathbf{x}) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2} \quad (33)$$

$$\tau' = P / (\sqrt{2} x_1 x_2) \quad (34)$$

$$\tau'' = MR / J \quad (35)$$

$$M = P(L + (x_2 / 2)) \quad (36)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \quad (37)$$

$$J = 2 \left\{ \frac{x_1 x_2}{\sqrt{2}} \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2 \right] \right\} \quad (38)$$

$$\sigma(\mathbf{x}) = 6PL / (x_4 x_3^2) \quad (39)$$

$$\delta(\mathbf{x}) = 4PL^3 / (Ex_3^3 x_4) \quad (40)$$

$$P_c(\mathbf{x}) = \frac{4.013 \sqrt{EG(x_3^2 x_4^6 / 36)}}{L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right) \quad (41)$$

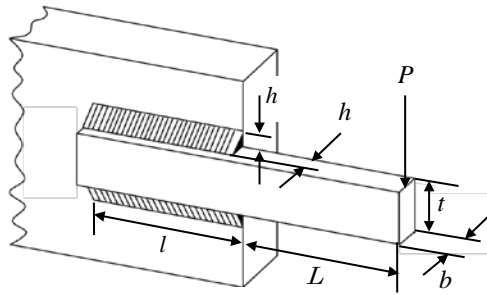


Fig.9 Design problem of welded beam

Table 7 Comparison of results of welded beam design

Design Variables	Best solutions found				
	Ray and Liew ⁽¹⁸⁾	He et al ⁽¹⁹⁾	Wang and Yin ⁽²⁰⁾	DE	ARDE
$x_1(h)$ [mm]	6.2087	6.2070	6.2070	6.2067	6.2068
$x_2(l)$ [mm]	158.44	157.92	157.92	157.91	157.93
$x_3(t)$ [mm]	210.53	210.60	210.60	210.63	210.61
$x_4(b)$ [mm]	6.2120	6.2070	6.2070	6.2069	6.2069
$g_1(\mathbf{x})$ [Pa]	-22346E+01	-1.9262	-1.6981	-354.96	-2215.6
$g_2(\mathbf{x})$ [Pa]	-22376	-3.5293	-4.0282	-47426	-19434
$g_3(\mathbf{x})$ [mm]	-3.2488E-03	0.0000E+00	0.0000E+00	-1.4073E-04	-1.0780E-04
$g_4(\mathbf{x})$ [mm]	-3.0337	-3.0320	-3.0320	-3.0317	-3.0318
$g_5(\mathbf{x})$ [mm]	-5.9496	-5.9497	-5.9497	-5.9499	-5.9498
$g_6(\mathbf{x})$ [N]	-58.180	-1.3745E-03	-1.3968E-03	-1.1409	-0.47144
$f(\mathbf{x})$ [\$]	2.3854	2.3810	2.3810	2.3810	2.3810
Function call	40000	30000	30000	8100	7560
Average of $f(\mathbf{x})$	3.2551	2.3819	2.3810	2.3813	2.3811
Worst of $f(\mathbf{x})$	6.3997	N/A	2.3810	2.3815	2.3813
Standard Deviaton of $f(\mathbf{x})$	0.959	5.24E-03	1.14E-05	1.28E-04	5.92E-05

5・5 数値計算のまとめ

提案する手法は通常の DE に比べ、少ない計算回数で効率的に精度良く大域的最適解を求めることができている。DE では探索方向のランダム性により、探索は時として外挿領域にも及び幅広い探索が行われるが、提案手法では有効探索領域内に常に最良点を含みつつ平均まで考慮するため、有効探索領域を広く取ることができ、外挿領域への探索を妨げない。また仮に探索の途中で見つけた局所的最適解に最良点が移動した場合でも、平均値と局所的最適解の位置がずれていたり、あるいは探索点のばらつきが大きかったりする場合、簡単に有効探索領域は縮小しない。そして平均値と最良点が近付き、全体のばらつきが少なくなってきたときに、はじめて探索点の集中化がおき有効探索領域は縮小する。このため局所的最適解の発見に留まることなく、大域的最適解を求めることができると言える。また、5.2 節で扱った問題に関して、Michalewics 関数を除くベンチマーク問題に対して、最大探索回数を固定 ($k_{\max}=1000$) し、探索点数を 20、設計変数の数を 20 とした場合、ARDE において大域的最適解が発見できない結果も見受けられた。5.2 節で扱った問題に関しては、本手法の有効範囲は設計変数の数が 10 程度の問題までと考えられる。なお、文献(21)では、探索の初めに設定する突然変異確率 F と交叉確率 Cr を、探索状況に応じて自動的に制御する研究が行われており、数多くのベンチマーク問題が示されているが、ファンクションコール（探索点数と最大探索回数の積）は本論文で提案した方法に比べ非常に多いため、一概に比較することは困難であるが、最適解を求めるまでの効率（ファンクションコール）を考慮すれば、本論文で提案した方法の有効性の一端は確認できるものと考えられる。

6. 結 語

本論文では、先行研究である領域適応型 Particle Swarm Optimization (ARPSO) を参考に、領域適応型 Differential Evolution (ARDE) を提案した。提案する手法は DE に各設計変数の平均と標準偏差、そして最良点から構成される有効探索領域を適用して探索を行う。この有効探索領域は ARPSO の強制的な正規分布の変更をなくし、本来の正規分布を利用して構成され、探索状況をより反映したものとなっている。提案手法は各設計変数の最良点と平均それぞれを中心とした領域を重ね合わせるため有効探索領域を広く取ることができ、DE の持つ外挿領域をも探索に含む探索の多様性を維持したまま探索を行うことが可能である。数値計算例を通じ、計算コスト（ファンクションコール）を下げつつ局所的最適解の発見に留まらずに精度良く大域的最適解を発見できることが確認できた。

付 録

本論文で扱った無制約最適化問題を以下に示す。なお以下では設計変数の数を n とする。

(P1) 2ⁿ minima 関数

大域的最適解は $n = 10$ で $\mathbf{x}_G = (-2.90354, \dots, -2.90354)^T$ 、そのときの目的関数値は $f(\mathbf{x}_G) = -391.661$ である。本論文の表 2 は $n = 10$ の場合の計算結果である。

$$f(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i) \rightarrow \min \quad (\text{A1})$$

$$-5 \leq \mathbf{x} \leq 5 \quad (\text{A2})$$

(P2) Griewank 関数

大域的最適解は $n = 10$ で $\mathbf{x}_G = (0, 0, \dots, 0)^T$ 、そのときの目的関数値は $f(\mathbf{x}_G) = 0$ である。本論文の表 3 は $n = 10$ の場合の計算結果である。

$$f(\mathbf{x}) = 1 + \frac{1}{400} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \rightarrow \min \quad (\text{A3})$$

$$-10 \leq \mathbf{x} \leq 10 \quad (\text{A4})$$

(P3) Ackley 関数

大域的最適解は $n = 10$ で $\mathbf{x}_G = (0, 0, \dots, 0)^T$ 、そのときの目的関数値は $f(\mathbf{x}_G) = 0$ である。本論文の表 4 は $n = 10$ の場合の計算結果である。

$$f(\mathbf{x}) = 22.71828 - 20 \exp\left[-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right] - \exp\left[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right] \rightarrow \min \quad (\text{A5})$$

$$-30 \leq \mathbf{x} \leq 30 \quad (\text{A6})$$

(P4) Michalewics 関数

本論文では $n = 5$ として計算し、大域的最適解は $\mathbf{x}_G = (2.202906, 1.570796, 1.284992, 1.923059, 1.720470)^T$ であり、そのときの目的関数値は $f(\mathbf{x}_G) = -4.687658$ である。

$$f(\mathbf{x}) = -\sum_{i=1}^n \sin(x_i) \times \left\{ \sin\left(\frac{ix_i^2}{\pi}\right) \right\}^{20} \rightarrow \min \quad (\text{A7})$$

$$0 \leq \mathbf{x} \leq \pi \quad (\text{A8})$$

文 献

- (1) Storn, R., and Price, K.V., “Differential Evolution – a simple and efficient adaptive scheme for global optimization over continuous space”, *Technical Report TR-95-012*, (1995), ICSI.
- (2) Storn, R., and Price, K.V., “Differential Evolution – a simple and efficient heuristic for global optimization over continuous spaces”, *Journal of Global Optimization*, Vol.11, (1997), pp. 341-359.
- (3) Kennedy, J., and Eberhart, R., “Particle Swarm Optimization”, *IEEE International Conference on Neural Networks*, (1995), pp.1942-1948.
- (4) Kitayama, S., Arakawa, M., and Yamazaki, K., “Penalty Function Approach for the Mixed Discrete Non-Linear Problems by Particle Swarm Optimization”, *Structural and Multidisciplinary Optimization*, Vol.32, No.3, (2006), pp.191-202.
- (5) 安田恵一郎, 石亀篤司, “非線形計画アルゴリズム—実用的視点から—”, *システム/制御/情報*, Vol.50, No.9, (2006), pp.1-7.
- (6) Zaharie, D., “Influence of Crossover on the Behavior of Differential Evolution Algorithms”, *Applied Soft Computing*, Vol. 9, (2009), pp. 1126-1138.
- (7) Tvrdik, J., “Adaptation in Differential Evolution: A Numerical Comparison”, *Applied Soft Computing*, Vol.9, (2009), pp. 1149-1155.
- (8) 北山哲士, 酒井忍, 荒川雅生, 山崎光悦, “大域的最適化手法としての Differential Evolution と数値計算”, *日本機械学会論文集 C 編*, Vol. 76, No. 771, (2010), pp. 73-82.
- (9) 北山哲士, 荒川雅生, 山崎光悦, “Discrete Differential Evolution の提案”, *日本機械学会論文集 C 編*, Vol. 76, No. 772, (2010), pp. 3828-3836.
- (10) Pan, Q.K., Tasgetiren, M.F., and Liang, Y.C., “A Discrete Differential Evolution Algorithm for the Permutation Flowshop Scheduling Problem”, *Computers and Industrial Engineering*, Vol.55, (2008), pp.795-816.
- (11) Qian, B., Wang, L., Hu, R., Huang, D.X., and Wang, X., “A DE-based Approach to No-Wait Flow-Shop Scheduling”, *Computers and Industrial Engineering*, Vol.57, (2009), pp.787-805.
- (12) Dasgupta, S., Das, S., Biswas, A., and Abraham. A., “On Stability and Convergence of the population-dynamics in Differential Evolution”, *AI Communications*, Vol.22, (2009), pp.1-20.
- (13) 北山哲士, 荒川雅生, 山崎光悦, “領域適応型 Particle Swarm Optimization の提案”, *日本機械学会論文集 C 編*, Vol. 73, No. 725, (2007), pp. 280-287.
- (14) Arora, J.S., *Introduction to Optimum Design* (1989), McGraw-Hill, New York.
- (15) Coello Coello C.A., “Use of a Self-Adaptive Penalty Approach for Engineering Optimization Problems”, *Computers in Industry*, Vol.41, (2000), pp.113-127.
- (16) Ray, T., and Saini, P., “Engineering Design Optimization Using Swarm with an Intelligent Information Sharing among Individuals”, *Engineering Optimization*, Vol.33, (2001), pp.735-748.
- (17) Rao, S.S., *Engineering Optimization, Theory and Practice, Third Edition* (1996), Wiley Interscience.
- (18) Ray, T., and Liew, K.M., “Society and civilization: An optimization algorithm based on the simulation of social behavior”, *IEEE Transactions on Evolutionary Computation*, Vol.7, (2003), pp.386-396.
- (19) He, S., Prempan, E., and Wu, Q.H., “An improved particle swarm optimizer for mechanical design optimization problems”, *Engineering Optimization*, Vol.36, (2004), pp.585-605.
- (20) Wang, J., and Yin, Z., “A ranking selection-based particle swarm optimizer for engineering design optimization problems”, *Structural and Multidisciplinary Optimization*, Vol.37, (2008), pp.131-147.
- (21) J. Brest., S. Greiner., B. Boskovic., M. Mernik, and V. Zumer., “Self-Adapting Control Parameters in Differential Evolution: A Comparative study on Numerical Benchmark Problems”, *IEEE Transactions on Evolutionary Computation*, Vol.10, (2006), pp. 646-657.