

Differential evolution as the global optimization technique and its computing

メタデータ	言語: jpn 出版者: 公開日: 2017-10-03 キーワード (Ja): キーワード (En): 作成者: メールアドレス: 所属:
URL	https://doi.org/10.24517/00008056

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 International License.



大域的最適化法としてのDifferential Evolutionと数値計算

北山哲士^{*1} 酒井 忍^{*1}
荒川雅生^{*2} 山崎光悦

Differential Evolution as the Global Optimization Technique and its Computing

Satoshi KITAYAMA, Shinobu SAKAI, Masao ARAKAWA, Koetsu YAMAZAKI

College of Science and Engineering, Kanazawa University

Kakuma-machi, Kanazawa, 920-1192, Japan

In this paper, Differential Evolution (DE) is examined through two aspects. Thus, one is the meta-heuristics, and the other is the global optimization technique. It is said that DE is the global optimization technique, and also belongs to the meta-heuristics. Indeed, DE can find the global minimum through numerical experiments. However, there are no proofs and useful examinations with respect to such comments. In this paper, DE is compared with the Generalized Random Tunneling Algorithm (GRTA) and the Particle Swarm Optimization (PSO), that are the global optimization techniques. Through the examination, some common characteristics as the global optimization technique are clarified in this paper. In addition, the difference of the neighborhood between DE and PSO is clarified. As the result, DE is possible to belong to the global optimization techniques. Additionally, DE is also examined as the meta-heuristics. Through benchmark test problems, the search ability of DE as the global optimization technique is examined.

Key Words : Global Optimization, Meta-heuristics, Differential Evolution, Engineering Optimization, Optimal Design, System Engineering

1. 緒言

近年、コンピュータを積極的に活用した設計が行われており、最適設計もその中の一つに位置づけることができる。最適設計では、各種最適化法を用いて最適解を求めることになるが、最適化法の中でも特に、メタヒューリスティクス（もしくはメタ戦略）と称される最適化法の研究は目覚ましい進化を遂げている。メタヒューリスティクスとは、理論的保証はないものの経験的に有効と思われる考え方を取り入れた最適化手法の枠組みであり⁽¹⁾、具体的な方法として例えば、遺伝的アルゴリズム(GA)、シミュレーテッドアニーリング(SA)、タブサーチ(TS)、Particle Swarm Optimization (PSO)、アントコロニー最適化(ACO)などが挙げられる。これらの方法の目的は、与えられた最適化問題の近似的な大域的最適解を求めることにあり、アルゴリズム内に乱数を用いているため、確率的最適化手法に属すると言えよう⁽²⁾。メタヒューリスティクスでは、経験的に有効と思われる考え方や法則を取り入れ、アルゴリズムを構築するが、それに加えて下記の項目が明確にされていることが望ましいと思われる。

(M1) 探索点の近傍領域・近傍点や、次の探索における探索点の生成方法が明確であること。

(M2) アルゴリズム内のパラメータの設定指針が明確であること。

(M3) アルゴリズムが様々な多峰性関数に対してもロバストであること。すなわち、様々な多峰性関数に対して、大域的最適解を求めることができること。

上記(M2)と(M3)、すなわちパラメータの設定指針や様々な多峰性関数の大域的最適解が求められるアルゴリズムであるか否かは、多くの数値計算を通じて決められることが多いが、(M1)は極めて重要な性質であると思われる。すなわち、最適化におけるメタヒューリスティクスとは、経験的に有効と思われる考え方を導入する一方で、(M1)~(M3)を有することが望ましいと著者らは考えている。

筆者らは従来より大域的最適化法に関する研究を行ってきているが、大域的最適化法は下記の点において必要である。

(G1) 離散変数問題や混合整数計画問題において、離散変数をペナルティ関数を用いて連続変数に変換する場合、拡大目的関数は多峰性となり、離散変数問題や混合整数計画問題の最適解探索は、大域的最適解を求める問題に変換される⁽³⁾。また、組合せ最適化問題は、基本的に大域的最適解を求める問題である。

* 原稿受付 平成??年?月 日

*¹正員, 金沢大学理工研究域 (〒920-1192 金沢市角間町)。

*²正員, 香川大学工学部 (〒761-0396 高松市林町2217-20)。

(G2) KrigingやRBFネットワークによる応答曲面は多峰性関数となりやすい⁽⁴⁾。そのため、応答曲面の最適解探索には大域的最適化法が必要となる。

(G3) min-max問題は基本的に多峰性関数であり、min-max問題の最適解を求める問題は、大域的最適解を求める問題である。例えば多目的最適化において対話型手法を用いた場合、パレート最適解を求めるためにはmin-max問題を解く必要があり⁽⁵⁾、大域的最適化法が必要となる。

(G4) 筆者らが提唱しているトレードオフ比を考慮した妥協点を得るための希求水準の算出は、多峰性関数となり、大域的最適化法が必要となる⁽⁶⁾。

さて、連続変数を対象とした関数の感度（勾配）を用いない多点同時探索型最適化手法の一つに、Differential Evolution (DE)がある⁽⁷⁾。DEは、突然変異・交叉・適者生存という操作を繰り返しながら、大域的最適解を求める方法であり、GAの考え方に強い影響を受けた方法であると思われ、メタヒューリスティクスの一つとして考えることができる⁽⁸⁾。DEではいくつかのバージョンが提唱されており⁽⁹⁾、また多目的最適化への適用も試みられており⁽¹⁰⁻¹⁵⁾、多くの数値計算例を通じて、その有効性が検討されているが、筆者らの調べる限りでは、DEをメタヒューリスティクスとして位置づけるような研究、すなわち先ほどのべた(M1)の観点から調べた研究や、他の大域的最適化法との類似構造に関して検討したような研究はない。このような視点でDEを検討することにより、最適化手法としてのDEの位置付けや、最適解を探索できる理由が明確になるものと思われる。

そこで本論文では、メタヒューリスティクスとしてのDE、すなわちDEの突然変異・交叉・適者生存といった操作について、それが意味することについて検討する。また筆者らが開発した一般化ランダム・トンネリング・アルゴリズム(GRTA)とPSOとの類似構造について言及する。すなわち、DEが大域的最適解を求めることができる理由について検討する。数値計算例では、DEの探索性能について検討する。本論文は、単純に数値計算を行い、DEの優位性を検討したものであるかもしれないが、最も重要な点は、メタ戦略としての位置づけや、GRTAやPSOとの類似構造、さらにはPSOとDEの近傍に関する相違点などを詳細に検討した論文であることをはじめに付記しておく。

2. Differential Evolution

DEは連続変数を対象とした関数の感度を用いない多点同時探索型手法である。はじめに基本アルゴリズム

を示す。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^(注)。

2.1 基本アルゴリズム

DE/1/rand/binの基本アルゴリズムを以下に示す。なお、 k 回目の探索において、 i 番目の探索点の設計変数ベクトルを \mathbf{x}_i^k と表記し、設計変数の数を n とする。

(STEP1) 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) すべての探索点に対し、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 を生成。ここで、 \mathbf{v}_d^k は以下の式によって得られる。

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

(STEP2-3) 探索点 \mathbf{x}_d^k と \mathbf{v}_d^k が交叉し、新たな点 \mathbf{u}_d^k を生成。

(STEP2-4) 目的関数 $f(\mathbf{x})$ の評価をし、探索点の位置を更新。

$$\begin{aligned} f(\mathbf{u}_d^k) \leq f(\mathbf{x}_d^k) &\rightarrow \mathbf{x}_d^k = \mathbf{u}_d^k \\ f(\mathbf{u}_d^k) > f(\mathbf{x}_d^k) &\rightarrow \mathbf{x}_d^k = \mathbf{x}_d^k \end{aligned} \quad (2)$$

(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。

上の流れの中で、(STEP2-2)が突然変異と呼ばれ、(STEP2-3)が交叉と呼ばれる。

2.2 突然変異と大域的探索

図1に示すように、DEでは d 番目の探索点 \mathbf{x}_d^k がランダムに3つの探索点 $\mathbf{x}_{r1}^k, \mathbf{x}_{r2}^k, \mathbf{x}_{r3}^k$ を選択し \mathbf{v}_d^k を生成する。

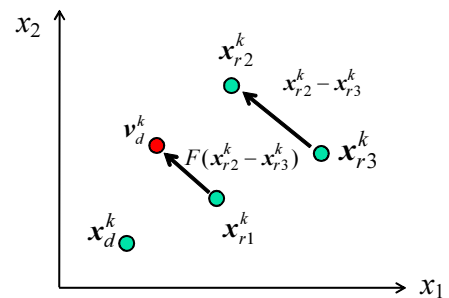


Fig.1 New point \mathbf{v}_d^k in two dimensions

このとき、無制約における勾配法の探索点更新式と式(1)を比較すると、 \mathbf{x}_{r1}^k が基底ベクトルであり、 $\mathbf{x}_{r2}^k - \mathbf{x}_{r3}^k$ が探索方向ベクトル、 F がステップ幅を表し

(注) 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の意味している。

ているものと考えることができる。また F は一律な値を適用しているが、各設計変数毎に適用する方法もあり、例えば、 d 番目の探索点の i 番目の設計変数を $x_{d,i}^k$ ($i=1,2,\dots,n$) と表記すれば、

$$v_{d,i}^k = x_{r_1,i}^k + F_i(x_{r_2,i}^k - x_{r_3,i}^k) \quad (3)$$

$$F_i = F + d(\text{rand}(0,1) - 0.5) \quad (4)$$

$$d < 2F \quad (5)$$

とするものもある⁽¹⁰⁾。ここで $\text{rand}(0,1)$ は $[0,1)$ の乱数であり、式(5)は F_i を正にするための条件である。式(1)もしくは式(3)を用いても、DEにおける突然変異は、無制約における勾配法の探索点更新式と類似構造を持っているとみなすことができる。また、勾配法では、現在の設計点を基底ベクトルに選び、勾配計算を行い、目的関数を常に改善する方向に向けて局所探索を行うのに対し、DEではランダムに基底ベクトルを選んでいる。基底ベクトルや探索方向ベクトルをランダムに選択するという点がDEにおける大域的探索能力を表しているものと考えられる。

2.3 交叉と近傍点生成 DEにおける交叉を図2を用いて説明する。 x_d^k の成分を $x_{d,i}^k$ ($i=1,2,\dots,n$) とし、突然変異によって生成された点 v_d^k の成分を $v_{d,i}^k$ ($i=1,2,\dots,n$) とする。DEの交叉では、はじめに交叉点を決める。そして、交叉点の設計変数の成分を $v_{d,i}^k$ から引き継ぎ、新たな点 u_d^k の成分とする。以降、交叉点を除くすべての設計変数の成分毎に乱数 r を発生させ、

$$r \in [0,1) \leq Cr \quad (6)$$

であれば、 $v_{d,i}^k$ から設計変数の成分を引き継ぎ、

$$r \in [0,1) > Cr \quad (7)$$

であれば、 $x_{d,i}^k$ から設計変数の成分を引き継ぐ。

設計変数の数が8である場合を考える (図2参照)。

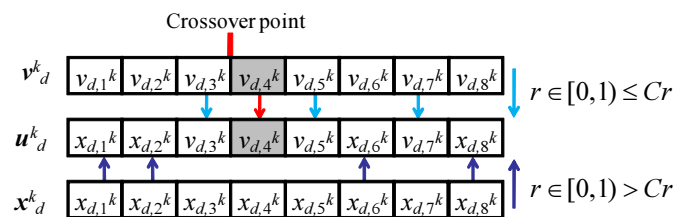


Fig.2 Crossover in DE

このとき、交叉点が4番目である時、新しい点 $u_{d,i}^k$ の4番目の設計変数の成分は $v_{d,i}^k$ の4番目の設計変数の成分となる。以降、4番目の設計変数を除くすべての設計変数の成分に対して、乱数を発生させ、交叉確率と比較することにより、新たな点 u_d^k のすべての設計変数の成分を決める。

ここで、交叉の意味について考えてみる。簡単のため、図3に示すように2変数で考える。このとき、交叉

によって生成される点 u_d^k は図3中の▲で表わすどちらかの点、もしくは確率的に v_d^k となることもありうる。すなわち、DEにおける交叉とは、 x_d^k の近傍点を直接的に生成していることに他ならない。また別の考え方として、交叉が x_d^k の摂動を与えていると考えることもできる。GAにおける交叉とは考え方や交叉方法が根本的に異なることに注意されたい。

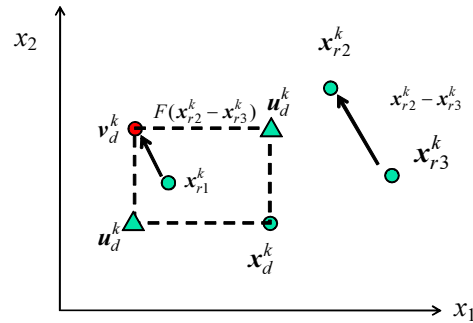


Fig.3 Generation of the neighborhood by crossover in DE

2.4 集団降下特性 多点同時探索型手法では、集団としての降下特性を持たせることにより、目的関数の改善を図り、さらには近似的な最適解を得ようとする。すなわち、多点同時探索型手法が最適解を探索できる理由は、集団としての降下特性にある。DEでは式(2)から明らかのように、 x_d^k と突然変異・交叉を経て生成された新たな点 u_d^k の目的関数値を比較することが、集団としての降下特性を示しており、これにより、近似的な最適解の探索が可能となっている。

2.5 多様性と集中化 多点同時探索型手法で重要となる要素の一つとして、集団の多様性と集中化が挙げられる。集団に多様性を持たせることで、幅広い探索を行い、一方で集中化により、精度の高い近似的な最適解を得ることが可能となる。DEの探索能力の一つの要素は、2.2節で述べた基底ベクトルの選び方であり、探索点 x_d^k とランダムに選ぶ基底ベクトル $x_{r_1}^k$ の位置関係が重要である。図3のように、 x_d^k と $x_{r_1}^k$ が設計変数空間における距離が近い場合、突然変異・交叉によって得られる点 u_d^k は、 x_d^k の近くに生成される。 u_d^k

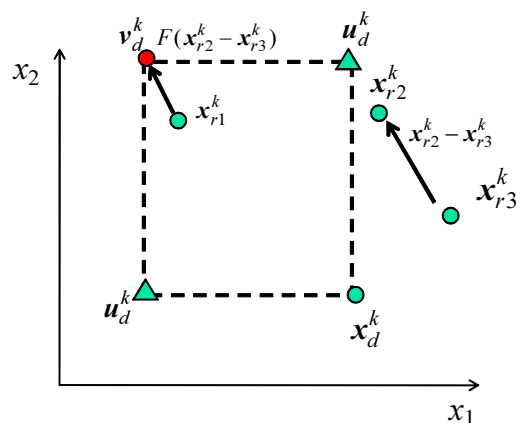


Fig.4 Selection of basis vector and diversity of search points

は x_d^k に対する試行点であり、さらに集団としての降下特性を考慮すれば、このような場合は、ある領域の集中的な探索を意味する。

一方、図4のように、 x_d^k と x_{r1}^k が設計変数空間における距離が遠い場合は、突然変異・交叉によって得られる点 u_d^k は、 x_d^k から離れた点に生成される。集団としての降下特性を考慮すれば、この場合は幅広い探索、すなわち多様性を維持しながら探索を進めていることになる。

2.6 メタヒューリスティクスとしてのDE はじめに述べた通り、メタヒューリスティクスとして望ましい性質の中で、(M1)について考えてみると、DEは、探索点の更新式が突然変異・交叉によって与えられており、また交叉によって、近傍点が明確にされているため、メタヒューリスティクスとして望ましい性質を有していることがわかる。

3. 大域的最適化法としてのDE

本章では、著者らが開発した関数の感度を用いて大域的最適解もしくは次善の局所最適解を求める方法であるGRTA、多点同時探索型最適化手法であるPSOと、DEの類似構造について検討し、DEが大域的最適解もしくは次善の局所最適解を求めることができる理由について検討する。

3.1 GRTAとDE GRTAは、関数の感度を用いて一旦、局所最適解 x_L を求め、そこから摂動 δx を与えることにより、効率的に大域的最適解もしくは次

善の局所最適解を求める方法である⁽¹⁶⁾。紙面の都合上、アルゴリズムのみを図5に示す。詳細については文献(12)を参照されたい。

GRTAで重要な一つの要素は、局所最適解から脱出するための摂動の与え方である。局所最適解 x_L の i 番目の成分を $x_{L,i}$ ($i=1,2,\dots,n$) とすれば、摂動を与えることによって生成される新たな点 x_i ($i=1,2,\dots,n$) は

$$x_i = x_{L,i} + T \tan(p_i) \quad (8)$$

で表わされる。ここで p_i は i 番目の設計変数に対して生成される $-\pi/2 < p_i < \pi/2$ の乱数であり、 T はステップ幅を表している。式(8)によって生成された新たな点は局所最適解 x_L と比較され、目的関数値が改善されれば、新たな点が制約条件を満足しているかを検討するステップに移る。無制約最適化問題の場合は、目的関数が改善された点を新たな探索初期点として、新たな局所最適解を探索することになる。

ここで、DEとの類似構造について検討すると、GRTAおよびDEは共に目的関数値の改善を検討している点において、アルゴリズム内に降下特性を有していることがわかる。また、GRTAにおいて基底ベクトルは局所最適解 x_L であり、探索方向ベクトルが $\tan(p_i)$ ($i=1,2,\dots,n$)、ステップ幅が T であると考えれば、式(3)と類似構造を持っていると考えることが可能である。式(1)と異なり、式(3)では各設計変数毎に突然変異確率が設定されていることに注意されたい。

3.2 PSOとDE PSOは多点同時探索型の大域的最適化法の一つである。PSOでは各探索点が「位置」

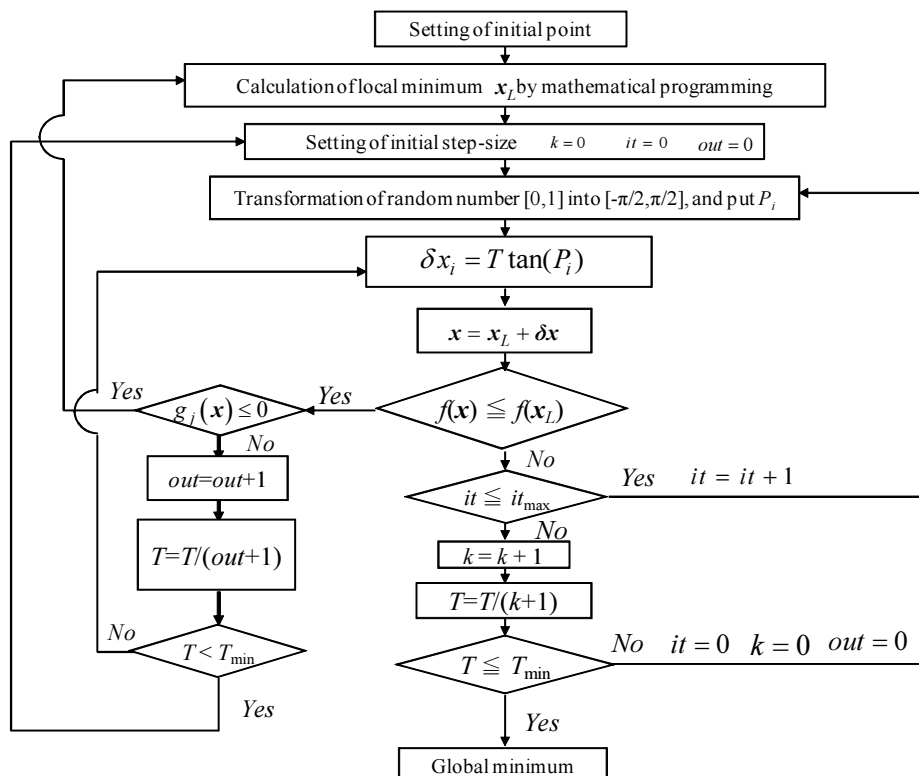


Fig.5 The algorithm of GRTA

と「速度」を持ち、集団で探索を行い、各探索点の位置と速度を更新しながら、最適解を探索する方法である。 k 回目の探索において、探索点 d の位置 \mathbf{x}_d^k と速度 \mathbf{v}_d^k を用いて、 $k+1$ 回目の位置 \mathbf{x}_d^{k+1} と速度 \mathbf{v}_d^{k+1} は、次の式を用いて更新される。

$$\mathbf{x}_d^{k+1} = \mathbf{x}_d^k + \mathbf{v}_d^{k+1} \quad (9)$$

$$\mathbf{v}_d^{k+1} = w\mathbf{v}_d^k + c_1r_1(\mathbf{p}_d^k - \mathbf{x}_d^k) + c_2r_2(\mathbf{p}_g^k - \mathbf{x}_d^k) \quad (10)$$

式 (10) において、 r_1 と r_2 は $[0, 1)$ の乱数である。また c_1 、 c_2 と w はパラメータである。 \mathbf{p}_d^k は、探索点 d が k 回目までの探索において、今までで訪れた最良の解 (p-best) を表す。一方、 \mathbf{p}_g^k は k 回目の探索における群れ全体の中での最良の解 (g-best) を表す。式 (9)、(10) を用いて探索点を更新するモデルは通常、g-best モデルと呼ばれる。また式 (10) 中の \mathbf{p}_g^k を、 k 回目までの探索で目的関数値を最良にした探索点 \mathbf{p}_g で置き換えたモデルは、最良値保存型モデルと呼ばれる⁽³⁾。最良値保存型モデルのアルゴリズムを簡単にまとめると次のようになる。

(STEP1) 探索点数、最大探索回数を決める。また側面制約条件を設定する。

(STEP2) 各探索点に対して、ランダムに初期位置 \mathbf{x}_d^k と初期速度 \mathbf{v}_d^k を決める。 $k=1$ とする。

(STEP3) 各探索点に対して、目的関数値を計算する。

(STEP4) \mathbf{p}_d^k と \mathbf{p}_g を求める。 $f(\mathbf{p}_d^k)$ と $f(\mathbf{p}_g)$ が改善されていれば、置き換える。

(STEP5) 各探索点の速度と位置を式 (9)、(10) に従い更新。また慣性項 w を以下の式に従い更新。

$$w = w_{\max} - (w_{\max} - w_{\min}) / k_{\max} \times k \quad (11)$$

(STEP6) 探索回数 k が最大探索回数以下なら $k = k+1$ としてSTEP3へ戻る。そうでなければ、探索終了。

PSOでは、STEP4において、探索集団の降下特性を検討する要素が取り入れられている。また、式(9)、(10)を変形すると次式が得られる。

$$\mathbf{x}_d^{k+1} = \mathbf{x}_d^k + w\mathbf{v}_d^k + \alpha(\mathbf{q} - \mathbf{x}_d^k) \quad (12)$$

$$\alpha = c_1r_1 + c_2r_2 \quad (13)$$

$$\mathbf{q} = \frac{c_1r_1\mathbf{p}_d^k + c_2r_2\mathbf{p}_g^k}{c_1r_1 + c_2r_2} \quad (14)$$

式 (5) はPSOが感度を用いた最適化手法との類似構造をもっていることを示しており、探索方向ベクトルは $\mathbf{q} - \mathbf{x}_d^k$ で与えられることを示している。また α は確率的ステップ幅を表していると考えられる。また \mathbf{q} は \mathbf{p}_d^k と \mathbf{p}_g^k を内分する点を表しており、最良値保存モデルでは、必ず

$$f(\mathbf{p}_g) \leq f(\mathbf{p}_d^k) \quad (15)$$

という関係が成立しており、ここでも探索集団としての降下特性を満足していることがわかる。

DEとの対比を考えれば、PSOとDEは多点同時探索型最適化手法として、下記の点で類似構造をもっていると考えることが可能である。

- (1) PSOにおける基底ベクトルは式(12)から明らかのように、 $\mathbf{x}_d^k + w\mathbf{v}_d^k$ である。
- (2) 探索方向ベクトルを有している。
- (3) 式(3)と式(13)を比較した場合、PSOとDEは共に確率的ステップ幅を有している。
- (4) 探索集団として降下特性を有している。

3.3 類似点のまとめ

GRTA, PSOとDEのそれぞれについて、探索点の生成 (更新式)、探索方向ベクトル、基底ベクトル、ステップ幅という点でまとめると表1のようになる。なお、表1中の灰色部分は乱数による確率的要素を陽に含むものを示している。

表1より、各手法はそれぞれ、類似構造があることがわかり、またアルゴリズム内に目的関数値を比較することにより探索点を置きかえる降下特性を含んでいるため、大域的最適解もしくは次善の局所的最適解を探索できるものと考えられる。

3.4 近傍の相違

DEとPSOは共に連続変数を対象とした多点同時探索型手法であるが、生成される近傍は大きく異なる (図6参照)。図6は、2変数における近傍の相違を示している。

PSOの近傍は図6の灰色部で示した領域であり、この

Table 1 Comparison of three global optimization techniques

	GRTA	PSO	DE
Generation of the new search point	$\mathbf{x}_i = \mathbf{x}_{L,i} + T \tan(p_i)$	$\mathbf{x}_d^{k+1} = \mathbf{x}_d^k + w\mathbf{v}_d^k + \alpha(\mathbf{q} - \mathbf{x}_d^k)$ $\alpha = c_1r_1 + c_2r_2$ $\mathbf{q} = \frac{c_1r_1\mathbf{p}_d^k + c_2r_2\mathbf{p}_g^k}{c_1r_1 + c_2r_2}$	$\mathbf{v}_{d,j}^k = \mathbf{x}_{r_1,j}^k + F_i(\mathbf{x}_{r_2,j}^k - \mathbf{x}_{r_3,j}^k)$ + Mutation
Search direction vector	$\tan(p_i)$	$\mathbf{q} - \mathbf{x}_d^k$	$\mathbf{x}_{r_2}^k - \mathbf{x}_{r_3}^k$
Step-size	T	$c_1r_1 + c_2r_2$	$F + d(\text{rand}(0,1) - 0.5)$
Basis vector	\mathbf{x}_L	$\mathbf{x}_d^k + w\mathbf{v}_d^k$	$\mathbf{x}_{r_1}^k$

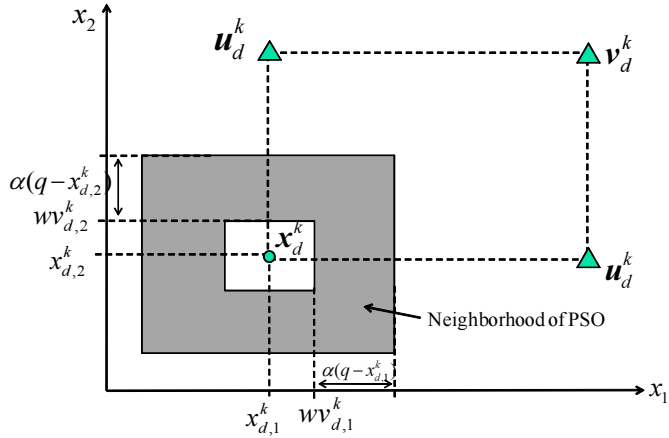


Fig.6 Difference of the neighborhood between PSO and DE

領域内に新たな点 x_d^{k+1} を生成を生成する. 一方, DEでは, 近傍点を直接的に生成する (図6中の▲). また, DEでは基底ベクトルをランダムに選び, さらに近傍点を生成しながら探索を行うため, 比較的幅広い探索を可能にしているのに対し, PSOでは, 表1に示した通り, 基底ベクトルは $x_d^k + wv_d^k$ であり, DEと比べると, 幅広い探索は難しい. また, PSOは図6の近傍領域内に次の探索点を必ず生成するため, 目的関数値の改善を許容しながら大域的最適解の探索を行うのに対し, DEでは, 式(2)から明らかなように, 目的関数値を改善する点のみを許容するため, PSOに比べ, 探索性能が高いと予想される.

4. 数値計算

DEの探索能力をいくつかの代表的な数値計算例を通じて検討する. DEの中では, 突然変異確率 F , 交叉確率 C_r などのいくつかのパラメータを設定しなければならないが, 本論文ではGRTAとDEの類似構造を考え, 突然変異確率として式(3), (4)を用い, 式(4)中の数値は次のように設定した.

$$F = 0.8 \quad (16)$$

Table 2 Some benchmark problems for unconstrained optimization problem

No.	Name	Number of design variables	Objective	Side constraints	Objective at global minimum
1	2^n minima	10	$f(x) = \frac{1}{2} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i) \rightarrow \min$	$-5 \leq x \leq 5$	$f(x_G) = -391.661$
2	Griewank	10	$f(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \rightarrow \min$	$-10 \leq x \leq 10$	$f(x_G) = 0$
3	Ackley	10	$f(x) = 22.71828 - 20 \exp\left[-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right] - \exp\left[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right] \rightarrow \min$	$-30 \leq x \leq 30$	$f(x_G) = 0$
4	Michalewicz	5	$f(x) = -\sum_{i=1}^n \sin(x_i) \times \left\{ \sin\left(\frac{i x_i^2}{\pi}\right) \right\}^{20} \rightarrow \min$	$0 \leq x \leq \pi$	$f(x_G) = -4.687658$

$$d = 0.0001 \quad (17)$$

また交叉確率は一様乱数を用いることを考慮し, $C_r = 0.5$ とした.

4.1 無制約最適化問題 表2に示す代表的な無制約最適化問題のベンチマーク問題を考える. 探索点数を30, 最大探索回数を500とした.

DEが連続変数を対象にした多点同時探索型最適化手法であること, また最終的に得られる最適解の精度を検討するため, PSOや筆者らが開発した領域適応型PSO (ARPSO)⁽¹⁷⁾などと比較する. PSOやARPSOの速度更新式 (式(10)および式(11))の中のパラメータは次のように設定した.

Table 3 Result of 2^n minima function

	ARPSO	PSO	DE
Best objective	-391.661656	-391.65783	-391.66158
Worst objective	-391.661597	-390.98815	-391.65466
Mean value of objective	-391.661648	-391.510196	-391.66019
Standard Deviation of objective	1.83E-05	2.00E-01	2.15E-03
Average of function call	6879	13014	9729

Table 4 Result of Griewank function

	ARPSO	PSO	DE
Best objective	3.50E-09	8.93E-04	6.82E-07
Worst objective	9.83E-08	7.30E-02	4.47E-05
Mean value of objective	2.78E-08	2.91E-02	1.11E-05
Standard Deviation of objective	3.16E-08	2.52E-02	1.55E-05
Average of function call	7557	14799	11997

Table 5 Result of Ackley function

	ARPSO	PSO	DE
Best objective	6.5100E-04	5.7838E-02	2.4170E-03
Worst objective	1.5700E-03	9.8645E-02	7.3280E-03
Mean value of objective	1.1947E-03	8.3662E-02	3.9370E-03
Standard Deviation of objective	3.1302E-04	1.2248E-02	1.7340E-03
Average of function call	12249	14919	13527

Table 6 Result of Michalewicz function

	ARPSO	PSO	DE
Best objective	-4.687658	-4.687521	-4.687658
Worst objective	-4.687737	-4.645192	-4.687658
Mean value of objective	-4.68769	-4.669190	-4.687658
Standard Deviation of objective	-7.947280E-08	2.047007E-02	0.000000E+00
Average of function call	8697	14547	7578

$$c_1 = c_2 = 2.0 \quad (18)$$

$$w_{\max} = 0.9, \quad w_{\min} = 0.4 \quad (19)$$

各ベンチマーク問題に対し、乱数の種を変更し、20回試行したときの結果を表3~6に示す。なお、目的関数値の平均や標準偏差は、アルゴリズムのロバスト性を検討する一つの評価項目である。

これらの結果より、ARPSOの結果が比較的良好であることがわかる。またDEとPSOを比較した場合は、すべての結果においてDEの結果が優れていることがわかる。

そこで、多峰性の激しいGriewank関数およびAckley関

数について、目的関数の収束状況を検討する。具体的には目的関数値の標準偏差と探索回数との関係を調べることで、DEがどのように探索を行っているかを検討することが目的である。図7にGriewank関数、図8にAckley関数の収束状況を示す。

図7, 8より、DEが基底ベクトルと探索方向ベクトルをランダムに選択しているため、探索の中盤あたりまでは大域的な探索を行い、それ以降では、降下特性によって、大域的最適解周辺に探索点が集まっていることがわかる。また両関数とも大域的最適解はロバスト性のないピーキーな関数であり、このような問題でも大域的最適解が見つかったのは、探索点が集まりはじめても、交叉の影響が大きいものと思われる。

4.2 コイルバネの重量最小化問題 文献(18)

で取り上げられているコイルバネの重量最小化問題を考える。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。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。最適設計問題は次のように定式化される。

$$f(x) = (2 + x_3)x_1^2 x_2 \rightarrow \min \quad (20)$$

$$g_1(x) = 1 - x_2^3 x_3 / (71785 x_1^4) \leq 0 \quad (21)$$

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0 \quad (22)$$

$$g_3(x) = 1 - 140.45 x_1 / (x_2^2 x_3) \leq 0 \quad (23)$$

$$g_4(x) = (x_1 + x_2) / 1.5 - 1 \leq 0 \quad (24)$$

$$0.05 \leq x_1 \leq 2.00 \quad (25)$$

$$0.25 \leq x_2 \leq 1.30 \quad (26)$$

$$2.00 \leq x_3 \leq 15.0 \quad (27)$$

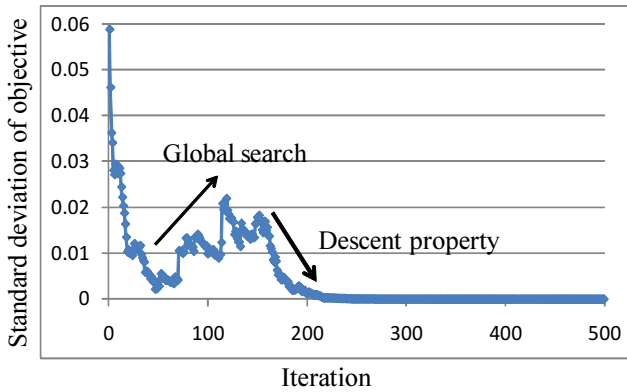


Fig.7 Convergence of Griewank function

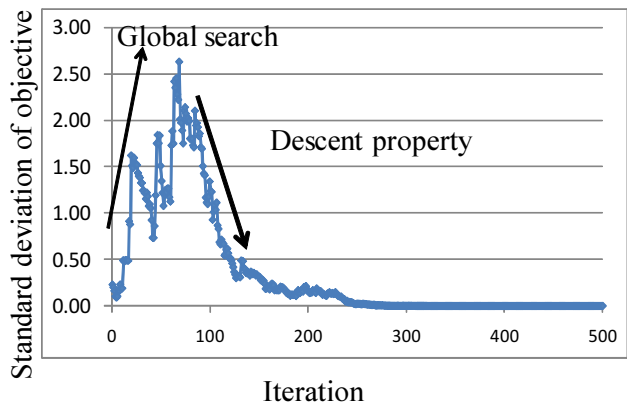


Fig.8 Convergence of Ackley function

Table 7 Comparison of result of minimum weight design of tension/compression spring

Design Variables	Best solutions found					
	Arora ⁽¹⁸⁾	Coello ⁽¹⁹⁾	Ray ⁽²⁰⁾	Hu ⁽²¹⁾	ARPSO ⁽¹⁷⁾	DE
$x_1 (d)$	0.053396	0.05148	0.050417	0.051466	0.051679	0.0516868
$x_2 (D)$	0.39918	0.351661	0.321532	0.351384	0.356477	0.3566636
$x_3 (N)$	9.1854	11.632201	13.979915	11.608659	11.299395	11.2878946
$g_1(x)$	0.000019	-0.00208	-0.001926	-0.003336	-0.000037	-8.22116E-10
$g_2(x)$	-0.000018	-0.00011	-0.012944	-0.00011	-0.000008	-1.1952E-11
$g_3(x)$	-4.123832	-4.026318	-3.89943	-4.026318	-4.054976	-4.0555802
$g_4(x)$	-0.698283	-0.731239	-0.752034	-0.731324	-0.727895	-0.7277664
$f(x)$	0.01273	0.012705	0.01306	0.012667	0.012661	0.0126612
Function Call	N/A	900000	1291	N/A	5804	5696
Average of $f(x)$	N/A	0.012769	0.013436	0.012719	0.012675	0.0126612
Worst of $f(x)$	N/A	0.012822	0.01358	N/A	0.012696	0.0126612
Standard Deviation of $f(x)$	N/A	3.9390E-05	N/A	6.4660E-05	1.1740E-05	2.41E-09

この問題において、ARPSOと同様に、探索点数を20、最大探索回数を500とした。また制約条件はペナルティ関数として扱い、文献(17)の方法を採用した。乱数の種を変更し、11回試行した結果を、いくつかの研究結果と併せて表7に示す。

表7より、DEでは極めて精度よく大域的最適解が求まっており、ファンクションコール(関数呼び出し回数)の平均もARPSOとほぼ同等であることがわかる。また、目的関数値の平均や標準偏差を比較すると、DEは他の方法と比べ、ロバストなアルゴリズムであることがわかる。

4.3 トラス構造の最適位相決定問題 文献(16)で取り上げている二次元トラス構造物の最適位相決定問題を考える(図9参照)。

最適位相問題は一般に複数の極小値を持つ多峰性問

題である。設計変数は各部材の断面積、目的関数はトラス構造の総体積であり、荷重点変位が許容変位以下となるような制約条件を課している。定式化は次のようになる。

$$f(\mathbf{A}) = \sum_{i=1}^n A_i L_i \rightarrow \min \quad (28)$$

$$g_j(\mathbf{A}) = u_k(\mathbf{A})/u_a(\mathbf{A}) - 1 \leq 0 \quad j=1,2,\dots,m \quad (29)$$

$$A_{i,\min} \leq A_i \leq A_{i,\max} \quad i=1,2,\dots,n \quad (30)$$

ここで $\mathbf{A} = (A_1, A_2, \dots, A_n)^T$ は設計変数である各部材の断面積、 $f(\mathbf{A})$ はトラスの総体積、 $g_j(\mathbf{A})$ は第 k 番目の変位 u_k に関する挙動制約条件であり、 u_a は許容変位の値である。また式(30)の $A_{i,\max}$ と $A_{i,\min}$ はそれぞれ各部材の断面積の上下限值であり側面制約条件である。

図9に示す二次元トラス構造は9節点28部材で構成される。節点1と3の水平鉛直方向の変位が固定されており、節点4と7に荷重 $P = 1.00 \times 10^3 [N]$ が作用するものと

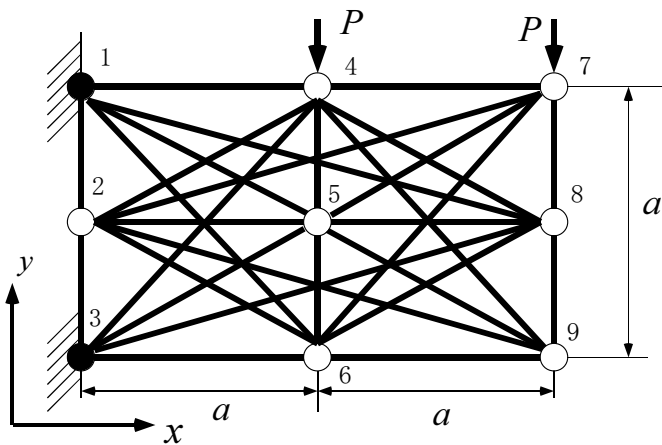


Fig.9 Truss structure in two dimensions

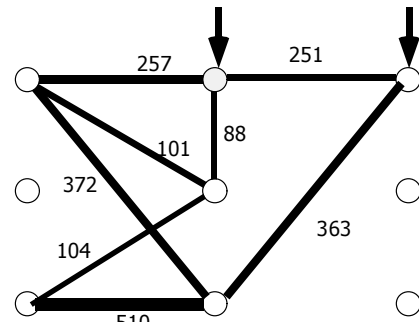


Fig.10 Optimum topology by DE

Table 8 Comparison of optimum topology by some global optimization techniques

Optimum topology at global minimum by GRTA	PSO (Best position keeping model)	Simple GA
	Objective 2.37×10^5	Objective 2.56×10^5
	SA	Distributed GA
Objective 2.60×10^5	Objective 2.40×10^5	

する。節点間の距離を $a=100[mm]$ とし、変位制約として荷重点の鉛直変位を考え、許容変位値 u_a を $1.50 \times 10^{-2}[mm]$ とし、すべての設計変数の上下限値をそれぞれ $A_{i,max}=1.00 \times 10^3[mm^2]$ と $A_{i,min}=1.00[mm^2]$ とする。なおヤング率を $E=210[GPa]$ とした。

表8は、いくつかの最適化手法によって求めた最適位相の結果であり、表中のFCはファンクションコールを意味している。各種手法で用いたパラメータの詳細を表9にまとめて示す。

探索点数を50、最大探索回数を4000としたとき、DEによって得られた結果を図9に示す。設計変数の下限値となったもの部材については非表示としてあり、このときの目的関数値は $2.35 \times 10^5[mm^3]$ であった。

図10より、GRTAによる結果とほぼ同等の結果が得られていることがわかる。多点同時探索型最適化手法では、効率的に大域的最適解を求めることは困難であるが、数値計算結果からは、ほぼ大域的最適解が得られていると考えてよい。

また目的関数値の標準偏差と探索回数を図11に示す。図11において、PSOの半分のファンクションコールの時点(探索回数:2000)におけるトラス構造の最適位相を図12に示す。図12では、断面積が $5[mm^2]$ 以下の部材は非表示としてある。

図12より、PSOの半分のファンクションコールで、ほぼ同等の結果が得られていることがわかり、また図11より、この時点では探索点が大域的探索を行っている様子がうかがえる。

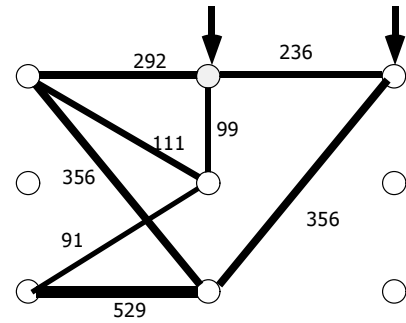


Fig.12 Optimum topology by DE at 2000 iteration

4.4 数値計算のまとめ 本論文で扱った範囲の数値計算の結果から、下記のことがわかる。すなわち、多峰性関数に対し、大域的探索能力は、連続型多峰性関数の大域的最適解を求めるために開発された多点同時探索型のPSOよりもDEの探索能力は優れている。PSOの場合、探索点全体がg-bestへ向かう。PSOではg-bestが局所的最適解にとらわれると、局所的最適解からの脱出はDEに比べると困難である。一方、DEでは基底ベクトルおよび探索方向ベクトルをランダムに選んでいるため、多様性と集中化のバランスがとれていると考えられる。この一つの要因としては、3.4節で示したように、近傍生成の方法がPSOとDEでは異なり、また、PSOが目的関数の改悪する領域に探索点を生成するのに対し、DEでは目的関数の改善した点のみを許容して探索を進めるという点が大きく影響しているものと考えている。さらに、設計変数の増加に対して、比較的ロバストなアルゴリズムであることもわかる。

5. 結言

本論文では、多点同時探索型最適化手法であるDEについて、メタヒューリスティクスとしての位置づけを行い、また大域的最適化手法としてのDEについて検討を行った。特に、GRTAやPSOと類似構造について検討し、大域的最適解もしくは次善の局所的最適解を求めることができる理由について検討した。DEはランダムに選ばれる探索方向ベクトルや確率的なステップ幅などを有し、また探索集団としての降下特性を持ち合わ

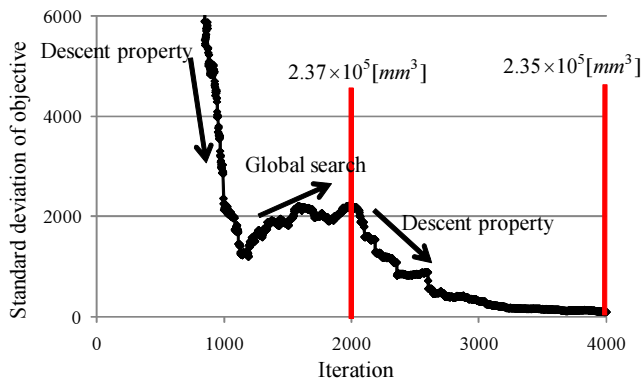


Fig.11 Convergence in the case of topology optimization problem

Table 9 Some parameters used in the algorithms

Method	Initial step size (Initial temperature)	Minimum step size (Minimum temperature)	Iteration of tunneling phase	Number of search agents	Numer of search iteration	Mutation	Crossover	Number of islands
GRTA	1.00	1.00E-05	20	/	/	/	/	/
PSO	/	/	/	100	2,000	/	/	/
Simple GA	/	/	/	100	10,000	0.01	0.6	/
Distributed GA	/	/	/	100	10,000	0.01	0.6	4
SA	1.00	1.00E-05	/	/	/	/	/	/

せていることが、最適解探索能力に大きく影響しているものと考えられる。さらに、PSOとDEでは近傍の取り方が大きく異なり、PSOが目的関数値の改悪を許容しながら探索を進めるのに対し、DEでは目的関数値の改善する点のみを許容して探索を行うという点が大きく異なる。連続変数を対象とした多点同時探索型最適化手法の一つであるPSOに対し、最終的に得られる目的関数値、探索効率、アルゴリズムのロバスト性などの項目について、数値計算例を通じ比較検討したが、本論文で扱った範囲の数値計算例では、これらの側面で、DEはPSOよりも優れていることがわかった。また、設計変数の規模が大きくなっても、DEの探索性能はPSOよりも優れていることがわかった。

最後に、本研究を遂行するにあたり、杉本博之先生（北海学園大学）、中山弘隆先生（甲南大学）、山川宏先生（早稲田大学）からは、適切なお助言をいただいた。この場を借り、謝意を表す。

参考文献

- (1) Yasuda, K., The Present and Future of Meta Heuristics, *J. of the Society of Instrument and Control Engineers*, Vol.47, No.6, (2008), pp.453-458.
- (2) Nemhauser, G.L., Rinnooy Kan, A.H.G., Todd, M.J. ed., *Handbooks in Operations Research and Management Science Vol.1 OPTIMIZATION*, (1989), Elsevier Science Publishers.
- (3) Kitayama, S., Arakawa, M., 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.
- (4) Nakayama, H., Arakawa, M., Sasaki, R., Simulation-Based Optimization Using Computational Intelligence, *Optimization and Engineering*, Vol. 3, (2002), pp.201-214.
- (5) Miettinen, K.M., *Nonlinear Multiobjective Optimization*, (1998), Kluwer Academic Publishers.
- (6) Kitayama, S., Yamazaki, K., Arakawa, M., Yamakawa, H., Trade-Off Analysis on the Multi-Objective Design Optimization, *Trans. of the Japan Society of Mechanical Engineers, Series C*, 75-754, (2009), pp.1828-1836.
- (7) Storn, R., Price, K.V., Differential Evolution - a simple and efficient heuristic for global optimization over continuous spaces- *J. of Global Optimization*, Vol.11, (1997), pp.341-359.
- (8) Feoktistov, V., *Differential Evolution - In Search of Solutions-*, Springer, (2006).
- (9) Coello Coello, C., Lamont, G.B., Van Veldhuizen, D., *Evolutionary Algorithms for Solving Multi-Objective Problems* (2nd Edition), (2007), Springer.
- (10) Price, K.V., Storn, R., Lampinen, J.A., *Differential Evolution - A Practical Approach to Global Optimization-*, (2006), Springer.
- (11) Chakraborty U.K. ed., *Advances in Differential Evolution*, (2008), Springer.
- (12) Liu, H., Cai, Z., Wang, Y., Hybridizing Particle Swarm Optimization with Differential Evolution for Constrained Numerical and Engineering Optimization, *Applied Soft Computing*, Vol.10, (2010), pp.629-640.
- (13) Zaharie, D., Influence of Crossover on the Behavior of Differential Evolution Algorithms, *Applied Soft Computing*, Vol.9, (2009), pp.1126-1138.
- (14) Tvrdik, J., Adaptation in Differential Evolution: A Numerical Comparison, *Applied Soft Computing*, Vol.9, (2009), pp.1149-1155.
- (15) Tang, H., Xue, S., Fan, C., Differential Evolution Strategy for Structural System Identification, *Computers and Structures*, Vol.86, (2008), pp.2004-2012.
- (16) Kitayama, S., Yamazaki, K., Generalized Random Tunneling Algorithm for Continuous Design Variables, *Trans. of ASME/ Journal of Mechanical Design*, Vol.127, No.3, (2005), pp.408-414.
- (17) Kitayama, S., Arakawa, M., Yamazaki, K., Proposal of Adaptive Range Particle Swarm Optimization, *Trans. of the Japan Society of Mechanical Engineers, Series C*, 73-725, (2007), pp.280-287.
- (18) Arora, J.S., *Introduction to Optimum Design*, (1989), McGraw-Hill, New York.
- (19) Coello Coello, C.A., Use of a Self-Adaptive Penalty Approach for Engineering Optimization Problems, *Computers in Industry*, Vol.41, (2000), pp.113-127.
- (20) Ray, T., Saini, P., Engineering Design Optimization Using Swarm with an Intelligent Information Sharing among Individuals, *Engineering Optimization*, Vol.33, (2001), pp.735-748.
- (21) Hu, X. H., et. al., Engineering Optimization with Particle Swarm, *IEEE Swarm Intelligence Symposium*, (2003), pp.53-57.