

Discrete Differential Evolution の提案

北山哲士^{*1} 荒川雅生^{*2} 山崎光悦

Proposal of Discrete Differential Evolution

Satoshi KITAYAMA, Masao ARAKAWA, Koetsu YAMAZAKI

College of Science and Engineering, Kanazawa University

Kakuma-machi, Kanazawa, 920-1192, Japan

The Differential Evolution (DE), which is the population-based optimization technique, is one of the global optimizers for the continuous design variables such as the Particle Swarm Optimization (PSO). In the traditional DE, due to the mutation in the algorithm, the rounding-off and truncating are often employed to handle the discrete and integer variables. In this paper, the Discrete Differential Evolution (DDE) to handle the discrete or integer design variables is proposed. In the proposed DDE, the mutation is considered as the exchange possibility between two particles. By considering the mutation as the exchange possibility, it is easy and possible to handle the discrete and integer variables. In addition, the initialization of the population are also introduced in the proposed DE. It is possible to escape from local minimum by introducing the initialization of the population. The algorithm of the proposed DDE is very simple, and can be easily extend to the Mixed-Discrete Nonlinear Problems (MDNLPs). The proposed DDE can be applied to a variety of discrete and integer optimization problems. The validity is examined through typical benchmark problems.

Key Words : Differential Evolution, Engineering Optimization, Optimal Design, System Engineering. Discrete Design Variables

1. 緒言

近年、コンピュータ技術の飛躍的な発達により、進化計算と称される計算手法の研究が盛んに行われており、その代表的な手法として、Artificial Bee Colony (ABC)やParticle Swarm Optimization (PSO)、Ant Colony Optimization (ACO)などが挙げられる。進化計算の多くは、多点同時探索を行い、さらに関数の感度（勾配）を利用しないという特徴を有しており、進化計算を最適化問題へ適用すると、大域的最適解もしくはそれに相当する準最適解を求めることができることが数値実験を通じて報告されている⁽¹⁻⁵⁾。最適化手法の分類の仕方としては、大きく確定的もしくは確率的な方法という分類があり⁽⁶⁾、進化計算はアルゴリズム内に乱数を用いるため、確率的な方法に分類される。

連続型多峰性関数の大域的最適解もしくはそれに相当する準最適解を求める進化計算の一つの方法に、Differential Evolution (DE)がある⁽⁷⁻¹⁰⁾。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、連続変数の大域的最適解もしくはそれに相当する準最適解を求めることが可能である。DEも、他の進化計算法と

同様に、多くの数値実験を通じて、その有効性が検討されている⁽¹¹⁻¹³⁾。

離散変数問題（組合最適化問題）や混合整数計画問題の最適解を求めようとする場合、一旦、すべての変数を連続変数と見なし、最終的に得られた最適解の離

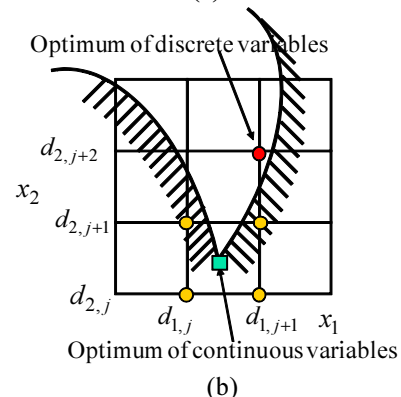
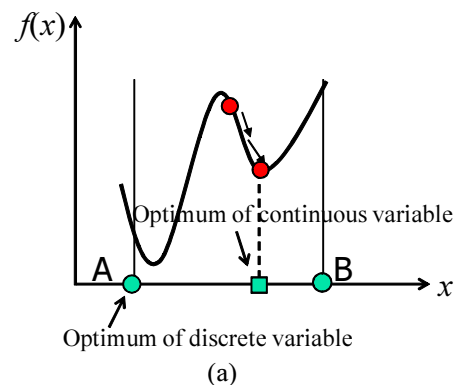


Fig.1 Nature of the discrete design variable problems

* 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)。

散変数の部分に対し、四捨五入や切り捨てを適用して、最適解を得ようとすることが多い^(14, 15)。しかし、離散変数の取り扱いには十分な注意が必要である。

目的関数に多峰性を許容する場合、例えば図1(a)に示すような1変数関数において、変数を連続変数とみなして得られた最適解(■)に対して四捨五入をすることで、近傍の離散値Bを離散変数の最適解とした場合、目的関数値が改悪されることがある。この程度であれば、準最適解として離散値Bを採用することもできるが、制約条件が付いた場合はより深刻な問題を引き起こす。例えば図1(b)のように、連続変数の最適解(■)に対して四捨五入や切り捨てを適用しても、制約条件を満足する離散変数の最適解が得られず、本来は離散変数の最適解は存在するにも関わらず、離散変数の最適解は存在しないという結論を導くことがある。

筆者らは、PSOによる離散変数問題や混合整数計画問題を解く方法を提案している⁽¹⁶⁾。この研究では、離散変数を直接的に扱うのではなく、ペナルティ関数を用いて離散変数を連続変数へ変換し、拡大目的関数を構築し、拡大目的関数の局所的小および大域的最適解が離散値に一致するという特徴を利用して、離散変数の最適解を求める方法であった。また、PSOが探索方向ベクトルを陰に有していること、探索点がg-best（全探索点の中で目的関数を最良にした探索点）へ向かって移動するという点、さらに連続型多峰性関数の大域的最適化法であることなど、PSOの特徴と拡大目的関数の特徴をうまく活用した方法であった。そこで、PSOと同様にDEが連続型多峰性関数の大域的最適化法であるということを考慮し、この方法を用いて離散変数最適化問題をDEで解いた結果、以下のような問題が生じ、離散変数の最適解を得ることができなかった。

(P1) DEでは基底ベクトルをランダムに選択するため、それまでの探索で目的関数値を最良にした探索点が保存されない。

(P2) そのため、離散変数に対するペナルティ関数のペナルティ係数の更新がされず、拡大目的関数が多峰性関数とならない場合もある。拡大目的関数が多峰性関数とならなければ、離散変数をはじめから連続変数と同様に扱っていることと同様であり、拡大目的関数の最適解（＝離散変数の最適解）すら見つからない。これはDEにはPSOにおけるg-bestとなる探索点が存在しないことに起因しており、DEでは必ずしも探索点が、それまでの探索で目的関数を最良とした探索点に集まらない。

(P3) 突然変異確率 F が存在するため、探索点を離散値上に移動させることができない。

離散変数をペナルティ関数として扱う方法では、ナップサック問題のような0-1のみをとる問題を対象とすると、極めて多くの局所的最適解が生成されてしまい、さらにペナルティ係数の更新方法も十分に検討されておらず、またDEを離散変数問題（組合最適化問題）へ適用する研究では、上記(P3)で述べた点について、何ら工夫することなく、突然変異によって生成される探索点の成分に対して四捨五入や切り捨てを適用されたり⁽¹⁷⁾、特定の問題（スケジューリング）に対してのみ有効な方法が提案されるなど^(18, 19)、いささか汎用性の欠ける方法が多いと思われる。

本論文では、DEが多点同時探索型であり、確率的な最適化法であることを踏まえ、離散変数を直接的に扱うDiscrete Differential Evolution (DDE)について報告する。DDEでは、突然変異確率 F が、探索点を持つ離散変数を交換するか否かを定める交換確率として考えられる。DEにおける突然変異確率 F は、新たな点を生成する際のステップ幅として作用しており、その役割は決定的に異なる。突然変異確率 F を交換確率としてみなすことで、DDEでは、離散変数を直接的に扱うことが可能となるので、DEのプログラムを大幅に変更する必要はない。また、離散変数をペナルティ関数により連続変数に変換するという作業がなくなる他、ペナルティ係数の更新方法も考える必要がない。さらに混合整数計画問題も容易に適用できるなどの特徴を有している。以降、2章ではDEの説明を行い、3章でDDEについて詳述する。アルゴリズムを示し、数値計算例を通じて、本論文で提案するDDEの有効性を検討する。

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/lrand/binを対象とする^(注)。

2.1 基本アルゴリズム DE/lrand/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) 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 C_r 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。

(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/lrand/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する。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。

(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。

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

である。ここで、 F は突然変異確率を表す。

(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。交叉による \mathbf{u}_d^k の生成の具体例を図2に示す。ここで、 \mathbf{x}_d^k の成分を $x_{d,i}^k$ ($i=1,2,\dots,n$) とし、 \mathbf{v}_d^k の成分を $v_{d,i}^k$ ($i=1,2,\dots,n$) とする。図2中の交叉点 (Crossover point) は、ランダムに選ばれる。また r は各設計変数毎に発生させる $[0,1]$ の乱数である。はじめに交叉点を決め、交叉点の設計変数の成分を \mathbf{v}_d^k から引き継ぎ、新たな点 \mathbf{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、 \mathbf{u}_d^k の残りの成分を決定してゆく。

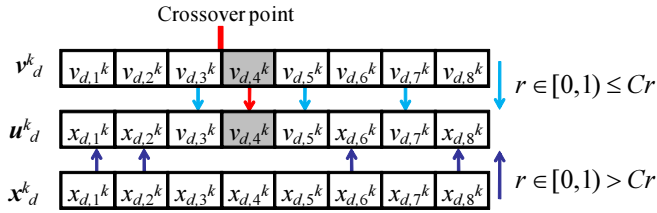


Fig.2 Crossover in DE

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

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

(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。

3. Discrete Differential Evolution

本論文で提案するDDEは、突然変異確率 F を受理確率とみなすことで、離散変数を直接的に扱うことが可能となる。この基本的な考え方はDiscrete Particle Swarm Optimization (DPSO)を参考に行っている。はじめにDPSOによる離散変数に取り扱いについて簡単に説明し、次に本論文で提案するDDEのアルゴリズムや特徴について説明する。

3.1 Discrete Particle Swarm Optimization PSOで

離散変数を直接的に扱う方法の一つとしてDPSOがある。DPSOの研究論文は、スケジューリング問題に関する適用事例として多く見受けられる。(20-22) PSOでは、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見つける方法である。 k 回目の探索における探索点 d の位置を \mathbf{x}_d^k 、速度を \mathbf{v}_d^k と表記すると、 $k+1$ 回目の探索における位置 \mathbf{x}_d^{k+1} を速度 \mathbf{v}_d^{k+1} はそれぞれ、以下の式によって表わされる。

$$\mathbf{x}_d^{k+1} = \mathbf{x}_d^k + \mathbf{v}_d^{k+1} \quad (3)$$

$$\mathbf{v}_d^{k+1} = w\mathbf{v}_d^k + c_1r_1(\mathbf{p}_d^k - \mathbf{x}_d^k) + c_2r_2(\mathbf{p}_g^k - \mathbf{x}_d^k) \quad (4)$$

式(4)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p-best、 \mathbf{p}_g^k はg-bestを表す。探索点 d の速度を制御する式(4)は、ベクトルの和と差、それと係数とベクトルの積から構成されていることがわかる。DPSOでは、離散変数を直接的に扱うため、ベクトルの和と差、係数とベクトルの積について、次のような規則を導入し、速度を制御している。なお、以下の式の中で、 r は $[0,1]$ の乱数、添え字 i はベクトルの i 番目の成分を表している。ベクトル \mathbf{b} と \mathbf{c} の和 (式(5)の右辺第2項と第3項の和に利用) :

$$\mathbf{a} = \mathbf{b} + \mathbf{c} \quad (5)$$

ここで、ベクトル \mathbf{a} の i 番目の成分 a_i は以下の式で決定される。

$$a_i = \begin{cases} c_i & r \leq 0.5 \\ b_i & r > 0.5 \end{cases} \quad (6)$$

ベクトル \mathbf{b} の \mathbf{c} 差 (式(5)の右辺第2項と第3項の中の差に利用) :

$$\mathbf{a} = \mathbf{b} - \mathbf{c} \quad (7)$$

ここで、ベクトル \mathbf{a} の i 番目の成分 a_i は以下の式で決定される。

$$a_i = b_i - c_i = \begin{cases} 0 & b_i = c_i \\ b_i & b_i \neq c_i \end{cases} \quad (8)$$

係数 α とベクトル \mathbf{b} の積 (式(5)右辺のすべての項に利用) :

$$\mathbf{a} = \alpha \mathbf{b} \quad (9)$$

ここで、 \mathbf{a} の i 番目の成分 a_i は以下の式で決定される。

$$a_i = \begin{cases} b_i & r \leq \alpha \\ 0 & r > \alpha \end{cases} \quad (10)$$

特に、ベクトル \mathbf{b} が

$$\mathbf{b} = \mathbf{c} - \mathbf{d} \quad (11)$$

である場合は、さらに

$$a_i = \begin{cases} c_i & r \leq \alpha \\ d_i & r > \alpha \end{cases} \quad (12)$$

となる。(23) DPSOでは、探索点の速度を制御すること

で、離散値を得る．特に式(8), (10)において、速度の成分を強制的にゼロとおくことで、探索点の位置の更新の有無を決定している点、また係数 α を交換確率と見なしている点に特徴があるといえよう．

3.2 DEにおける離散変数の取り扱い さて、連続型多峰性関数を対象としたDEでは、1章の(P3)で述べたように、突然変異確率の存在が、離散変数の取り扱いを困難にしていた．しかし、DPSOにおける離散変数の取り扱いを考えれば、DEにおける突然変異確率を探索点 \mathbf{x}_{r2}^k と \mathbf{x}_{r3}^k の成分の交換確率としてとらえることができる．具体的に式(1)で考えると、式(1)右辺第2項は係数とベクトルの積の項、式(1)右辺第1項と第2項はベクトルの和から構成されていることがわかる．ここで、さらに簡略化のために

$$\mathbf{D}^k = F(\mathbf{x}_{r2}^k - \mathbf{x}_{r3}^k) \quad (13)$$

を導入すると、

$$\mathbf{v}_d^k = \mathbf{x}_{r1}^k + F(\mathbf{x}_{r2}^k - \mathbf{x}_{r3}^k) = \mathbf{x}_{r1}^k + \mathbf{D}^k \quad (14)$$

となる．そこで、DPSOにおける離散変数の取り扱いを参考に、DDEでは、以下のような規則を導入する．式(13)において、 \mathbf{D}^k の i 番目の成分を D_i^k と表記すると、突然変異確率 F を探索点 \mathbf{x}_{r2}^k と \mathbf{x}_{r3}^k の成分の交換確率とみなし、

$$D_i^k = \begin{cases} \mathbf{x}_{r3,i}^k & r \leq F \\ \mathbf{x}_{r2,i}^k & r > F \end{cases} \quad (15)$$

となる．また式(14)において、 \mathbf{v}_d^k の i 番目の成分を $v_{d,i}^k$ と表記すると、 \mathbf{v}_d^k は \mathbf{x}_1^k と \mathbf{D}^k の和であることから、

$$v_{d,i}^k = \begin{cases} D_i^k & r \leq 0.5 \\ \mathbf{x}_{r1,i}^k & r > 0.5 \end{cases} \quad (16)$$

となる．なお、交叉は探索点 \mathbf{x}_d^k と \mathbf{v}_d^k の成分の交換であるため、 \mathbf{x}_d^k と \mathbf{v}_d^k の成分が離散値であれば、交叉によって生成される新たな点 \mathbf{u}_d^k の成分も離散値となる．

3.3 探索集団の初期化 式(15), (16)の規則を導入し、離散変数問題へ適用した結果、多くの場合で、局所的最適解に陥るという問題が生じた．これは、PSOでは、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、式(2)から明らかなように目的関数値を改悪する点の生成を許容しないためである．局所的最適解への捕捉を防ぐために、すべての探索点をもつ目的関数値の標準偏差が一定値 ε 以下となった場合、探索点を初期化し、探索点の多様性を強制的に持たせることにした．一見すると、このような方法を取り入れることは、ある意味で総当り的な探索に近いことをしていることになるが、分枝限定法が、ある意味で総当り的な方法であることを考慮すれば、

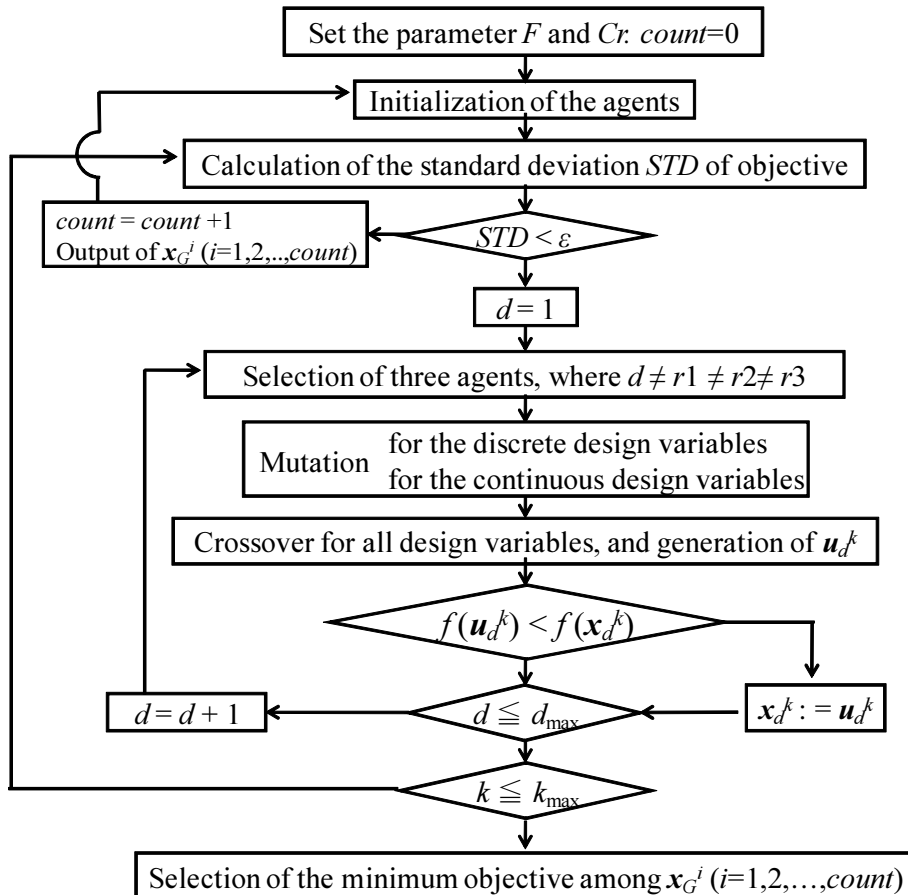


Fig.3 The algorithm of Discrete Differential Evolution

多点同時探索型手法では探索点の集中化と多様性の維持が重要な要素の一つであるため、このような方法は有効な方法の一つであると考えている。

3.4 混合整数計画問題の場合 設計変数ベクトル \mathbf{x} が連続変数ベクトル \mathbf{x}^c と離散変数ベクトル \mathbf{x}^d から構成されているとする。すなわち、

$$\mathbf{x} = (\mathbf{x}^c, \mathbf{x}^d)^T \quad (17)$$

と表記されたとする。このとき、離散変数ベクトル \mathbf{x}^d の部分に対してのみ式(13)～(16)を、連続変数ベクトル \mathbf{x}^c の部分については式(1)を適用して、 \mathbf{v}_d^k を決定すればよい。

3.5 アルゴリズム 本論文で提案するDDEのアルゴリズムを図3に示す。DDEで入力するパラメータは、2.1節で記述したアルゴリズムに加え、目的関数値の収束を判断する値 ε が必要となるだけである。

図3 中における STD は目的関数値の標準偏差を表し、 $count$ は収束した回数を表している。本論文で提案するアルゴリズムは、探索回数 k が最大探索回数 k_{\max} に到達した時点で終了となる。

4. 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DDEの有効性を検討する。本論文で用いた数値計算例において、突然変異確率を $F=0.6$ ，交叉確率を $Cr=0.5$ ，収束判定に用いた ε の値を $\varepsilon=1.0 \times 10^{-5}$ とした。また、探索点数および最大探索回数は、問題に応じて変更しており、すべての数値計算例では、乱数の種を変更し、10回試行した。

4.1 歯車列設計問題 文献(24)で用いられている歯車の歯数列の設計問題を考える。図4に示すように、4つの歯車 D, B, A, F があり、それらの歯数（離散変数） $T_d(=x_1)$ ， $T_b(=x_2)$ ， $T_a(=x_3)$ ， $T_f(=x_4)$ とする。

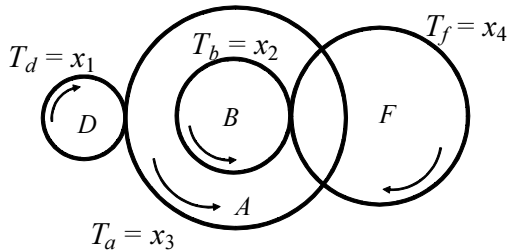


Fig.4 Design of gear train

このとき、所望のギア比と現状のギア比の差を最小化問題は、次のように定式化される。

$$f(\mathbf{x}) = \left(\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4} \right)^2 \rightarrow \min \quad (18)$$

$$12 \leq x_i \leq 60 \quad i=1,2,3,4 \quad (19)$$

探索点数を30, 最大探索回数を200として、DDEで探索を行った。最良の結果を、他のいくつかの研究結果と合わせ表1に示す。

表1 より、現時点で報告されていると考えられる大域的最適解は得られている。また、この問題は4つの大域的最適解が存在し、以下に示す大域的最適解も得られた。

$$\mathbf{x}_{opt}^D = (19, 16, 43, 49)^T \quad (20)$$

$$\mathbf{x}_{opt}^D = (19, 16, 49, 43)^T \quad (21)$$

$$\mathbf{x}_{opt}^D = (16, 19, 49, 43)^T \quad (22)$$

図5に、最良の目的関数値と目的関数値の標準偏差の履歴をそれぞれ、破線と実線で示す。図5より、およそ50回程度で大域的最適解は見つかっているが、3.3節で述べた探索点の初期化により、探索点が離合集散を繰り返しながら、大域的最適解を見つけていることがわかる。もし探索点の初期化を導入しなければ、局所的最適解に捕捉されていることは明らかであり、本論文で提案した探索点の初期化の効果が表れているものと考えられる。

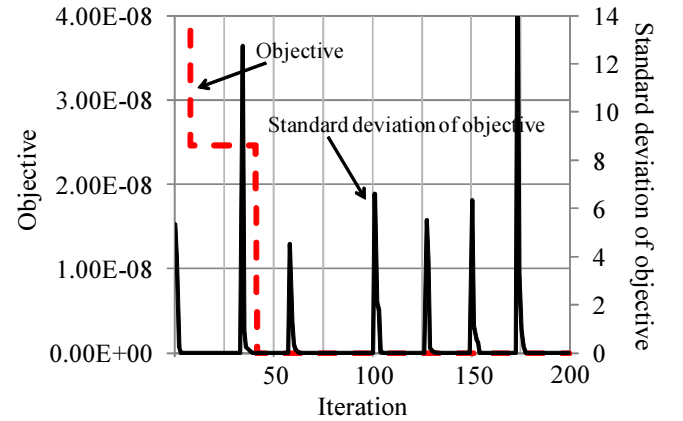


Fig.5 History of objective and standard deviation of objective

4.2 圧力器の最適設計問題 広く研究されている混合整数計画問題の一つとして、図6に示す圧力器の最適設計問題が挙げられる。設計変数は半径 $R(=x_1)$ と長さ $L(=x_2)$ （共に連続変数）と、圧力器の厚さ

Table 1 Comparison of the results on the gear train problem

	Sandgren ⁽²⁴⁾	Loh ⁽²⁵⁾	Zhang ⁽²⁶⁾	Wu ⁽²⁷⁾	Lin ⁽²⁸⁾	Guo ⁽²⁹⁾	Proposed DDE
x_1	18	19	30	19	19	16	16
x_2	22	16	15	16	16	19	19
x_3	45	42	52	43	49	43	43
x_4	60	50	60	49	43	49	49
obj.	5.70E-06	2.33E-07	2.36E-09	2.70E-12	2.70E-12	2.70E-12	2.70E-12

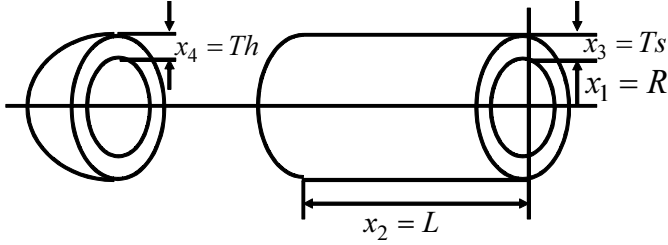


Fig.6 Optimum design of the pressure vessel

$Ts(=x_3)$ と $Th(=x_4)$ (共に離散変数) であり, 離散値 x_3 と x_4 の間隔は共に 0.0625 である. この問題の定式化は, 文献(16)の表記に合わせると次のようになる.

$$f(\mathbf{x}) = 0.6224x_1x_2x_3 + 1.7781x_1^2x_4 + 3.1661x_2x_3^2 + 19.84x_1x_3^2 \rightarrow \min \quad (23)$$

$$g_1(\mathbf{x}) = \frac{0.0193x_1}{x_3} - 1 \leq 0 \quad (24)$$

$$g_2(\mathbf{x}) = \frac{0.00954x_1}{x_4} - 1 \leq 0 \quad (25)$$

$$g_3(\mathbf{x}) = \frac{x_2}{240} - 1 \leq 0 \quad (26)$$

$$g_4(\mathbf{x}) = \frac{1296000 - \frac{4}{3}\pi x_1^3}{\pi x_1^2 x_2} - 1 \leq 0 \quad (27)$$

$$10 \leq x_1, x_2 \leq 200 \quad (28)$$

$$0.0625 \leq x_3, x_4 \leq 6.1875 \quad (29)$$

式(24)～(27)の制約条件は文献(31)の方法を用いて, ペナルティ関数として扱い, 拡大目的関数を最小化する問題に帰着させて解いた. 探索点数を50, 最大探索回数を1000としたとき最良の結果を表2に, また図7に, そのときの目的関数値と目的関数値の標準偏差の履歴をそれぞれ破線と実線で示す.

表2から, 現時点で報告されている大域的最適解は得られなかったが, 次善の局所的最適解は得られたものと考えられる. また図7より, 探索回数がおおよそ

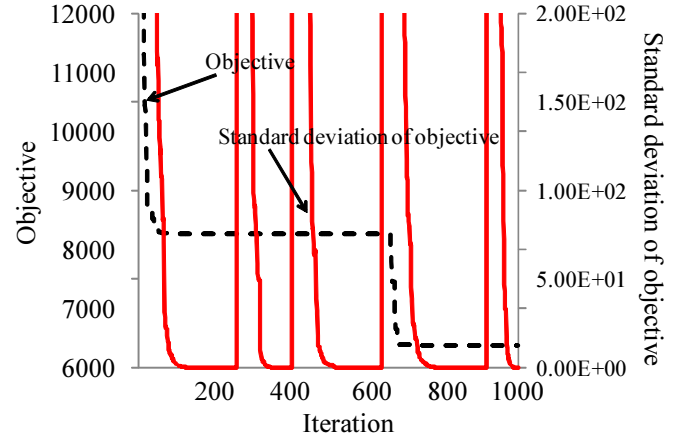


Fig.7 History of objective and standard deviation of objective

600回程度を超えた時点で, 探索点の初期化が行われ, 結果的には, 目的関数値が改善されていることがわかる.

4.3 研究室配属問題

輸送問題の典型的な例の一つとして, 研究室配属問題があげられる. ここでは, 簡単な例として, 8名の学生が3研究室を希望している場合を考える. 各学生の希望, 各研究室の定員は表3の通りとする. 表3中の数値は学生の希望順位を表している.

Table 3 Assignment of laboratory

	Lab.1	Lab.2	Lab.3
student1	1	2	3
student2	3	1	2
student3	2	1	3
student4	3	2	1
student5	3	1	2
student6	2	1	3
student7	1	3	2
student8	3	1	2
Capacity	3	2	3

このとき, 学生の希望の総和が最小となるように, 学生を研究室へ割り振ることを考える. このとき, 変数 $x_{i,j} \in \{0,1\}$ ($i=1,2,\dots,8, j=1,2,3$) を導入し, 学生の

Table 2 Comparison of the results on the pressure vessel

	Sandgren ⁽²⁴⁾	Zhang ⁽²⁶⁾	Wu ⁽²⁷⁾	Lin ⁽²⁸⁾	Guo ⁽²⁹⁾	Proposed Method	He ⁽³⁰⁾	Kitayama ⁽¹⁶⁾
x_1 (R[inch])	47.0000	N/A	58.1978	N/A	58.2900	45.3368	42.0980	42.3710
x_2 (L[inch])	117.7010	N/A	44.2930	N/A	43.7000	140.2539	176.6360	173.4170
x_3 (Ts[inch])	1.1250	N/A	1.1250	N/A	1.1250	0.8750	0.8125	0.8125
x_4 (Th[inch])	0.6250	N/A	0.6250	N/A	0.6250	0.5000	0.4375	0.4375
$g_1(\mathbf{x})$	-0.1937	N/A	-0.0016	N/A	0.0000	0.0000	0.0000	0.0000
$g_2(\mathbf{x})$	-0.2826	N/A	-0.1117	N/A	-0.0689	-0.1350	-0.0820	-0.0760
$g_3(\mathbf{x})$	-0.5096	N/A	-0.8154	N/A	-0.8179	-0.4156	-0.2640	-0.2770
$g_4(\mathbf{x})$	0.0542	N/A	-0.0021	N/A	-0.0001	0.0000	0.0000	0.0000
Objective[\$]	8129.8000	7197.9000	7207.7748	7197.7000	7198.3058	6318.9492	6059.7143	6029.8740

希望を行列 \mathbf{c} (行列の要素を $c_{i,j}$ とする), 研究室の定員を b_j ($j=1,2,3$) と表記すれば, 研究室配属問題は次のように定式化される.

$$\sum_{i=1}^8 \sum_{j=1}^3 c_{i,j} x_{i,j} \rightarrow \min \quad (30)$$

$$\sum_{i=1}^8 x_{i,j} = b_j \quad j=1,2,3 \quad (31)$$

$$\sum_{j=1}^3 x_{i,j} = 1 \quad i=1,2,\dots,5 \quad (32)$$

$x_{1,1} = x_1, x_{1,2} = x_2$ などとみると, 24変数問題となる.

この問題は等式制約条件となるため, 求解は比較的困難となる. 最小化する目的関数を $f(\mathbf{x})$, 等式制約条件を $h_j(\mathbf{x})$ ($j=1,2,\dots,m$) と表記すれば, 次の拡大目的関数 $F(\mathbf{x})$ を最小化する問題へ変換する.

$$F(\mathbf{x}) = f(\mathbf{x}) + \sum_{j=1}^m r_j h_j(\mathbf{x}) \rightarrow \min \quad (33)$$

ここで r_j ($j=1,2,\dots,m$) はペナルティ係数であり,

$$r_j = |h_j(\mathbf{x})|^2 \quad (34)$$

とした. 探索点数を20, 最大探索回数を100としたときの結果 (学生の配属研究室) を表3中の灰色で示す. このときの目的関数値は11であった.

4.4 数値計算のまとめ 本論文で扱った数値計算例の中でも特に圧力器の最適設計問題では, 提案したDDEは必ずしも従来の研究よりも良い大域的最適解を見つけることはできなかった. この問題に関して言えば, PSOによる方法, すなわち離散変数をペナルティ関数として扱う方法が, 比較的よい結果を出している. この理由の一つとして, PSOでは, 探索点更新式が陰に勾配ベクトルを含んだ形となっており, 関数の多峰性が激しくなればなるほど, 目的関数値を更新する方向へ探索点が移動しやすくなる. また探索点の更新によって, 離散変数に対するペナルティ係数が更新され, 結果的に拡大目的関数が多峰性の激しい関数になり, PSOの特徴をよくとらえた方法であったためと考えられる. しかし, 離散変数問題や混合整数計画問題の最適解を求めることが連続変数問題に比べ, 困難であることを考慮すれば, 少なくとも本論文で提案したDDEでは, 大域的最適解に相当する準最適解を見つけることができている.

5. 結言

本論文では, 連続型多峰性関数の最適解を求める確率的な手法の一つであるDEについて, 離散変数問題および混合整数計画問題を扱うことができるDDEを提案した. DEが確率的な手法であるということを踏まえ, DDEにおける離散変数の取り扱いは, DPSOを参考にし, 突然変異確率を探索点間の成分の交換確率とみなすことで, 離散変数を直接的に扱う方法である. また, DEが目的関数の改悪を許容しないため, 局所的最適解へ陥るという問題点を克服するために, 一旦探索

集団が収束した場合, 探索点の初期化を導入することで, 局所的最適解への捕獲を回避する方法を取り入れた. 多点同時探索型手法の課題の一つが, 探索点の多様性の維持と集中化であるという点を踏まえれば, このような方法は有効な方法の一つであると考えている. 本論文で提案したDDEのアルゴリズムの特徴は, DEのアルゴリズムを大幅に変更することなく, 簡単に離散変数を扱えるという点にある. そのため混合整数計画問題も容易に扱うことができる.

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.

付 録

本論文で提案したDDEを下記問題に適用した結果を以下に示す. 制約条件の取り扱い文献(31)の方法により, 拡大目的関数を作成し, 無制約最適化問題へ変換している. また, 探索点数を20, 最大探索回数を50としている. 以下のすべての問題に対して, 20回試行した結果, すべての試行において大域的最適解が得られており, また0-1計画問題に対しても適用可能である. 本論文で提案したDDEは, 離散変数や混合整数計画問題に対し, 強力な解法の一つであると考えている.

(P1) 文献(32)より引用.

$$\begin{aligned} f(\mathbf{x}) &= 7x_1^2 + 6x_2^2 + 8x_3^2 - 6x_1x_2 + 4x_2x_3 \\ &\quad - 15.8x_1 - 93.2x_2 - 63x_3 + 500 \rightarrow \min \\ g_1(\mathbf{x}) &= 142x_1 + 172x_2 + 118x_3 - 1992 \leq 0 \\ g_2(\mathbf{x}) &= 98x_1 + 114x_2 + 44x_3 - 1162 \leq 0 \\ g_3(\mathbf{x}) &= 40x_1 + 72x_2 + 34x_3 - 703 \leq 0 \\ 0 &\leq x_1, x_2, x_3 \leq 10 \\ x_1, x_2, x_3 &: \text{int} \\ \mathbf{x}_G &= (2, 7, 3)^T, \quad f(\mathbf{x}_G) = 69 \end{aligned}$$

(P2) 文献(33)より引用.

$$\begin{aligned} f(\mathbf{x}) &= 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \rightarrow \min \\ -0.55 &\leq x_1, x_2 \leq 4.95 \end{aligned}$$

x_1, x_2 の間隔は0.55.

$$\mathbf{x}_G = (1.65, 2.75)^T, \quad f(\mathbf{x}_G) = 0.498125$$

(P3) 文献(34)より引用.

$$\begin{aligned} f(\mathbf{x}) &= -9x_1^2 + 10x_1x_2 - 50x_1 + 8x_2 + 460 \rightarrow \min \\ g_1(\mathbf{x}) &= x_1 - (0.2768x_2^2 - 0.235x_2 + 3.718) \leq 0 \\ g_2(\mathbf{x}) &= x_1 + 0.019x_2^3 - 0.446x_2^2 \\ &\quad + 3.98x_2 - 15.854 \leq 0 \\ 0 &\leq x_1, x_2 \leq 10 \\ x_1, x_2 &: \text{int} \end{aligned}$$

$$\mathbf{x}_G = (5, 3)^T, \quad f(\mathbf{x}_G) = 159$$

(P4) 文献 (35) より引用.

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

$$g_1(\mathbf{x}) = 1/x_1 + 1/x_2 - 2 \leq 0$$

$$x_1 \in \{0.3, 0.7, 0.8, 1.2, 1.5, 1.8\}$$

$$x_2 \in \{0.4, 0.8, 1.1, 1.4, 1.6\}$$

$$\mathbf{x}_G = (0.8, 1.4)^T, \quad f(\mathbf{x}_G) = 2.6$$

(P5) 文献 (36) より引用.

$$f(\mathbf{x}) = -5x_1 - 10x_2 - 13x_3 - 4x_4$$

$$-3x_5 - 11x_6 - 13x_7 \rightarrow \min$$

$$g_1(\mathbf{x}) = 2x_1 + 5x_2 + 18x_3 + 3x_4$$

$$+ 2x_5 + 5x_6 + 10x_7 - 21 \leq 0$$

$$x_j \in \{0, 1\} \quad j = 1, 2, \dots, 7$$

$$\mathbf{x}_G = (0, 1, 0, 0, 0, 1, 1)^T, \quad f(\mathbf{x}_G) = -34$$

(P6) 文献 (36) より引用.

$$f(\mathbf{x}) = -5x_1 - 2x_2 - 6x_3 - 3x_4 - 8x_5 \rightarrow \min$$

$$g_1(\mathbf{x}) = 8x_1 - 3x_2 - 9x_3 + 4x_4 + 2x_5 + 2 \leq 0$$

$$g_2(\mathbf{x}) = 6x_1 + 4x_2 + 3x_3 + 2x_4 + 5x_5 - 10 \leq 0$$

$$x_j \in \{0, 1\} \quad j = 1, 2, \dots, 5$$

$$\mathbf{x}_G = (0, 0, 1, 1, 1)^T, \quad f(\mathbf{x}_G) = -17$$

(P7) 文献 (37) より引用.

$$f(\mathbf{x}) = (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + (x_4 - 1)^2$$

$$+ (x_5 - 2)^2 + (x_6 - 1)^2 - \ln(x_7 + 1) \rightarrow \min$$

$$g_1(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 - 5 \leq 0$$

$$g_2(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_6^2 - 5.5 \leq 0$$

$$g_3(\mathbf{x}) = x_1 + x_4 - 1.2 \leq 0$$

$$g_4(\mathbf{x}) = x_2 + x_5 - 1.8 \leq 0$$

$$g_5(\mathbf{x}) = x_3 + x_6 - 2.5 \leq 0$$

$$g_6(\mathbf{x}) = x_1 + x_7 - 1.2 \leq 0$$

$$g_7(\mathbf{x}) = x_2^2 + x_5^2 - 1.64 \leq 0$$

$$g_8(\mathbf{x}) = x_3^2 + x_6^2 - 4.25 \leq 0$$

$$g_9(\mathbf{x}) = x_3^2 + x_5^2 - 4.64 \leq 0$$

$$0 \leq x_i \leq 3 \quad i = 1, 2, 3 \text{ (continuous)}$$

$$x_j \in \{0, 1\} \quad j = 4, 5 \text{ (integer)}$$

$$\mathbf{x}_G = (0.2, 0.8, 1.908, 1, 1, 0, 1)^T, \quad f(\mathbf{x}_G) = 4.5796$$

(P8) 文献 (38) より引用.

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

$$g_1(\mathbf{x}) = \frac{\sqrt{3}x_2 + 1.932x_3}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_1x_3} - 1 \leq 0$$

$$g_2(\mathbf{x}) = \frac{0.634x_1 + 2.828x_3}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_1x_3} - 1 \leq 0$$

$$g_3(\mathbf{x}) = \frac{0.5x_1 - 2x_2}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_1x_3} - 1 \leq 0$$

$$g_4(\mathbf{x}) = -\frac{0.5x_1 - 2x_2}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_1x_3} - 1 \leq 0$$

$$\mathbf{x} \in \{0.1, 0.2, 0.3, 0.5, 0.8, 1.0, 1.2\}$$

$$\mathbf{x}_G = (1.2, 0.5, 0.1)^T, \quad f(\mathbf{x}_G) = 3.0414$$

参考文献

- (1) Karaboga, D., Basturk, B., A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm, *J. of Global Optimization*, Vol.39, (2007), pp.459-471.
- (2) Karaboga, D., Basturk, B., On the Performance of Artificial Bee Colony (ABC) Algorithm, *Applied Soft Computing*, Vol.8, (2008), pp.687-697.
- (3) Perez, R.E., Behdinan, K., Particle Swarm Approach for Structural Design Optimization, *Computers and Structures*, Vol.85, (2007), pp.1579-1588.
- (4) Li, L.J., Huang, Z.B., Liu, F., A Heuristic Particle Swarm Optimization Method for Truss Structures with Discrete Variables, *Computers and Structures*, Vol.87, (2009), pp.435-443.
- (5) 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.
- (6) Nemhauser, G.L., Rinnooy Kan, A.H.G., Todd, M.J., (eds.), Handbook in Operations Research and Management Science, Vol.1, *Optimization*, Elsevier Science, (1989).
- (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) Price, K.V., Storn, R., Lampinen, J.A., *Differential Evolution -A Practical Approach to Global Optimization-*, (2006), Springer.
- (9) Feoktistov, V., *Differential Evolution -In Search of Solutions-*, Springer, (2006).
- (10) Chakraborty U.K. ed., *Advances in Differential Evolution*, (2008), Springer.
- (11) Zaharie, D., Influence of Crossover on the Behavior of Differential Evolution Algorithms, *Applied Soft Computing*, Vol.9, (2009), pp.1126-1138.
- (12) Tvrdik, J., Adaptation in Differential Evolution: A Numerical Comparison, *Applied Soft Computing*, Vol.9, (2009), pp.1149-1155.
- (13) Tang, H., Xue, S., Fan, C., Differential Evolution Strategy for Structural System Identification, *Computers and Structures*, Vol.86, (2008), pp.2004-2012.

- (14) Venter, G., Sobieski, J.S., Particle Swarm Optimization, *AIAA Journal*, Vol.41, No.8, (2003), pp.1583-1589.
- (15) Venter, G., Sobieski, J.S., Multidisciplinary Optimization of a Transport Aircraft Wing Using Particle Swarm Optimization, *Structural and Multidisciplinary Optimization*, Vol.26, No.1-2, (2004), pp.121-131.
- (16) 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.
- (17) Onwubolu, G.C., Davendra, D., (eds), *Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization*, (2009), Springer.
- (18) Qian, B., Wang, L., Hu, R., Huang, D.X., Wang, X., A DE-based Approach to No-Wait Flow-Shop Scheduling, *Computers and Industrial Engineering*, Vol.57, (2009), pp.787-805.
- (19) Pan, Q.K., Tasgetiren, M.F., Liang, Y.C., A Discrete Differential Evolution Algorithm for the Permutation Flowshop Scheduling Problem, *Computers and Industrial Engineering*, Vol.55, (2008), pp.795-816.
- (20) Xia, W.X., Wu, Z.M., A Hybrid Particle Swarm Optimization Approach for the Job-Shop Scheduling Problem, *International Journal of Advanced Manufacturing Technology*, Vo.29, (2006), pp.360-366.
- (21) Pan, Q.K., Wang, L., No-idle Permutation Flow Shop Scheduling Based on a Hybrid Discrete Particle Swarm Optimization Algorithm, *International Journal of Advanced Manufacturing Technology*, Vo.39, (2008), pp.796-807.
- (22) Pan, Q.K., Wang, L., Tasgetiren, M.F., Zhao, B.H., A Hybrid Discrete Particle Swarm Optimization Algorithm for the No-wait Flow Shop Scheduling Problem with Makespan Criterion, *International Journal of Advanced Manufacturing Technology*, Vo.38, (2008), pp.337-347.
- (23) Engelbrecht, A.P., *Fundamentals of Computational Swarm Intelligence*, (2005), Wiley.
- (24) Sandgren, E., Nonlinear Integer and Discrete Programming in Mechanical Design Optimization, *ASME/J. of Mechanical Design*, Vol.112, (1990), pp.223-229.
- (25) Loh, H.T., Papalambros, P.Y., Computational Implementation and Tests of a Sequential Linearization Algorithm for Mixed-Discrete Nonlinear Design Optimization, *ASME/J. of Mechanical Design*, Vol.113, (1991), pp.335-345.
- (26) Zhang, C., Wang, H.P., Mixed-Discrete Nonlinear Optimization with Simulated Annealing, *Engineering Optimization*, Vol.21, (1993), pp.277-291.
- (27) Wu, S.J., Chow, P.T., Genetic Algorithms for Nonlinear Mixed Discrete-Integer Optimization Problems via Meta-Genetic Parameter Optimization, *Engineering Optimization*, Vol.24, (1995), pp.137-159.
- (28) Lin, S.S., Zhang, C., Wang, H.P., On Mixed-Discrete Nonlinear Optimization Problems: A Comparative Study, *Engineering Optimization*, Vol.23, (1995), pp.287-300.
- (29) Guo, C.X., Hu, J.S., Ye, B., Cao, Y.J., Swarm Intelligence for Mixed-Variable Design Optimization, *Journal of Zhejiang University SCIENCE*, Vo.5, No.7, (2004), pp.851-860.
- (30) He, S., Prempan, E., Wu, Q.H., An Improved Particle Swarm Optimizer for Mechanical Design Optimization Problems, *Engineering Optimization*, Vol.36, No.5, (2004), pp.585-605.
- (31) Kitayama, S., Yamazaki, K., Arakawa, M., Adaptive Range Particle Swarm Optimization, *Optimization and Engineering*, Vol.10, No.4, (2009), pp.575-597.
- (32) Loh, H.T., Papalambros, P.Y., A Sequential Linearization Approach for Solving Mixed-Discrete Nonlinear Design Optimization Problems, *ASME/J. of Mechanical Design*, Vol.113, (1991), pp.325-334.
- (33) Fu, J.F., Fenton R.G., Cleghorn, W., A Mixed Integer-Discrete-Continuous Programming Method and its Application to Engineering Design Optimization, *Engineering Optimization*, Vol.17, (1991), pp.263-280.
- (34) Loh, H.T., Papalambros, P.Y., Computational Implementation and Test of a Sequential Linearization Algorithm for Mixed-Discrete Nonlinear Design Optimization, *ASME/J. of Mechanical Design*, Vol.113, (1991), pp.335-345.
- (35) Olsen, G.R., Method for Nonlinear Optimization with Discrete Design Variables, *AIAA J.*, Vol.27, No.11, (1989), pp.1584-1589.
- (36) Sakawa, M., *Optimization of the Discrete Systems*, Morikita Pub., (2000), (in Japanese).
- (37) Floudas, C.A., (ed), *Handbook of Test Problems in Local and Global Optimization*, Kluwer Academic Publishers, (1999).
- (38) Rao, S.S., *Engineering Optimization* (Theory and Practice) 3rd edition, Wiley Interscience, (1996).