

制約付き最適化手法 ϵ 制約 Differential Evolution における 探索点のランク情報の利用法に関する一考察

広島修道大学商学部
Faculty of Commercial Sciences, Hiroshima Shudo University
広島市立大学大学院 情報科学研究科 高濱 徹行 (Tetsuyuki Takahama)
Graduate School of Information Sciences, Hiroshima City University

1 はじめに

差分進化 (Differential Evolution, DE) は 1995 年に Storn と Price [1, 2] によって提案された実数空間における最適化アルゴリズムであり, 進化的アルゴリズム (Evolutionary Algorithm, EA) の一つである. DE は非線形問題, 微分不可能な問題, 非凸問題, 多峰性問題など様々な最適化問題に適用されてきており, これらの問題に対して高速で頑健なアルゴリズムであることが示されてきている [3]. また, DE は進化的計算に関する国際会議のコンペティションにおいて優秀な成績を収めている [4, 5, 6].

DE は基本的には制約なし最適化手法であるが, 目的関数と制約逸脱度を分離して最適化する (特に制約逸脱度を目的関数より優先する辞書式比較を利用する) ϵ 制約法に基づく制約付き最適化法 ϵ DE が提案されている. ϵ 制約法は, 制約なし最適化手法を制約付き最適化手法へ変換するアルゴリズム変換法で実装が容易であり, DE の他にも GA, PSO に適用した ϵ GA, ϵ PSO が提案されているが, ϵ DE は特に高速かつ頑健であり探索性能が高いアルゴリズムであることが示されている.

最適化問題に対する DE の利用が増加してきている主な理由としては, 単純な算術演算に基づいているため高速に動作すること, 制御パラメータがスケリングファクター F , 交叉率 CR , 集団サイズ N の 3 つと単純であることが挙げられる. しかし, 制御パラメータについては, 問題によって適切なパラメータ設定が異なり, パラメータ設定によって DE の性能に大きな差が出るため, 非常に重要な検討課題となっている.

本研究では, 探索点のランク情報を利用することにより, F 及び CR に対する新しい固定的なパラメータ設定法を提案する. 従来のパラメータ設定法では, 集団全体に同じパラメータを使用していたが, 本研究では, 収束性と集団内の多様性維持に関する考察に基づき, ランク情報を利用して各探索点毎にパラメータ値を設定する.

本研究で提案するランク情報に基づきパラメータ値を設定する ϵ DE (ϵ constrained Rank-based DE, ϵ RDE) を, ϵ DE など多数の他手法と比較することにより, ϵ RDE の有効性を示す.

以下, 2. で最適化問題を定義し, ϵ 制約法について説明する. 3. で ϵ 制約法を DE に組み込んだ ϵ DE を説明し, ランク情報を用いたパラメータ設定法を提案する. 4. でこのパラメータ設定法を導入した ϵ RDE について説明する. 5. でテスト問題に対する実験結果を示し, 6. はまとめである.

2 制約付き最適化と ϵ 制約法

2.1 制約付き最適化問題

一般的な制約付き最適化問題 (P) は, 以下のように定義できる.

$$\begin{aligned} (P) \quad & \text{minimize} \quad f(\mathbf{x}) \\ & \text{subject to} \quad g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, q \\ & \quad \quad \quad h_j(\mathbf{x}) = 0, \quad j = q + 1, \dots, m \\ & \quad \quad \quad l_i \leq x_i \leq u_i, \quad i = 1, \dots, n \end{aligned} \tag{1}$$

ここで, $\mathbf{x} = (x_1, \dots, x_n)$ は n 次元決定変数ベクトル, $f(\mathbf{x})$ は目的関数, $g_j(\mathbf{x}) \leq 0$ は q 個の不等式制約, $h_j(\mathbf{x}) = 0$ は $m - q$ 個の等式制約であり, f, g_j, h_j は線形あるいは非線形の実数値関数である. l_i, u_i はそれぞれ, n 個の決定変数 x_i の下限値, 上限値である. さらに, 以下では実行可能領域を $\mathcal{F}$, 上下限制約のみを満足する領域を探索領域 S と呼ぶことにする.

2.2 制約逸脱度

ε 制約法では、制約をどの程度逸脱しているかを表現する制約逸脱度 $\phi(\mathbf{x})$ を導入する。制約逸脱度 $\phi(\mathbf{x})$ は、以下を満足する関数である。

$$\begin{cases} \phi(\mathbf{x}) = 0 & (\mathbf{x} \in \mathcal{F}) \\ \phi(\mathbf{x}) > 0 & (\mathbf{x} \notin \mathcal{F}) \end{cases} \quad (2)$$

制約逸脱度関数は、ペナルティ関数法におけるペナルティと同様に以下のような定義が可能である。

$$\phi(\mathbf{x}) = \max\{\max_j \{0, g_j(\mathbf{x})\}, \max_j |h_j(\mathbf{x})|\} \quad (3)$$

$$\phi(\mathbf{x}) = \sum_j \max\{0, g_j(\mathbf{x})\}^p + \sum_j |h_j(\mathbf{x})|^p, \quad \text{ただし, } p \text{ は正数である。} \quad (4)$$

2.3 ε レベル比較

関数値と制約逸脱度の組 (f, ϕ) の集合上において、制約逸脱度の大小関係を優先する辞書式順序 ε レベル比較を定義する。

点 $\mathbf{x}_1, \mathbf{x}_2$ における関数値を f_1, f_2 、制約逸脱度を ϕ_1, ϕ_2 とすると、関数値と制約逸脱度の組 (f_i, ϕ_i) 間の大小関係である ε レベル比較 $<_\varepsilon, \leq_\varepsilon$ ($\varepsilon \in [0, \infty)$) は以下ようになる。

$$(f_1, \phi_1) <_\varepsilon (f_2, \phi_2) \Leftrightarrow \begin{cases} f_1 < f_2, & \text{if } \phi_1, \phi_2 \leq \varepsilon \\ f_1 < f_2, & \text{if } \phi_1 = \phi_2 \\ \phi_1 < \phi_2, & \text{otherwise} \end{cases} \quad (f_1, \phi_1) \leq_\varepsilon (f_2, \phi_2) \Leftrightarrow \begin{cases} f_1 \leq f_2, & \text{if } \phi_1, \phi_2 \leq \varepsilon \\ f_1 \leq f_2, & \text{if } \phi_1 = \phi_2 \\ \phi_1 < \phi_2, & \text{otherwise} \end{cases} \quad (5)$$

なお、 $<_0, \leq_0$ は制約逸脱度を優先する辞書式比較と一致し、 $<_\infty, \leq_\infty$ は目的関数値のみの比較と一致する。

2.4 ε 制約法の性質

ε 制約法は、制約付き最適化問題を直接探索法で解く際に、通常の比較の代わりに ε レベル比較を用いる方法である。通常の大小比較を ε レベル比較に置き換えた最適化問題 $(P_{\leq_\varepsilon})$ 、すなわち、 ε 制約法による最適化問題は以下のように定義できる。但し、 $\text{minimize}_{\leq_\varepsilon}$ は $\leq_\varepsilon$ の意味での最小化である。

$$(P_{\leq_\varepsilon}) \quad \text{minimize}_{\leq_\varepsilon} \quad f(\mathbf{x}) \quad (6)$$

ここで、問題 (P) の制約条件を $\phi(\mathbf{x}) \leq \varepsilon$ に緩和した問題 (P^ε) を以下のように定義する。なお、 (P^0) は問題 (P) と等価である。

$$\begin{aligned} (P^\varepsilon) \quad & \text{minimize} \quad f(\mathbf{x}) \\ & \text{subject to} \quad \phi(\mathbf{x}) \leq \varepsilon \end{aligned} \quad (7)$$

問題 (P^ε) と問題 $(P_{\leq_\varepsilon})$ 、および問題 (P) に関して以下の定理が成り立つ [7]。

定理 1 問題 (P^0) に最適解が存在するならば、問題 $(P_{\leq_\varepsilon})$ の最適解は問題 (P^ε) の最適解である。

定理 2 問題 (P) に最適解が存在するならば、問題 $(P_{\leq_0})$ の最適解は、問題 (P) の最適解である。

定理 3 $\{\varepsilon_n\}$ を、強い意味で単調減少し、0 に収束する点列とする。 $f(\mathbf{x}), \phi(\mathbf{x})$ を連続関数とし、問題 (P^0) に最適解 $\mathbf{x}^*$ が存在し、かつ、任意の ε_n に対する問題 $(P_{\leq_{\varepsilon_n}})$ の最適解 $\hat{\mathbf{x}}_n$ が存在すると仮定する。このとき、点列 $\{\hat{\mathbf{x}}_n\}$ の任意の集積点は問題 (P^0) の最適解である。

3 ϵ 制約 Rank-based Differential Evolution(ϵ RDE)

3.1 Differential Evolution

DE では、探索空間 S 中にランダムに初期個体 $\mathbf{x}^i = (x_1, \dots, x_n)$ を生成し、初期集団 $P = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$ を構成する。各個体は決定ベクトルに対応し、 n 個の決定変数を遺伝子として持つ。各世代において、全ての個体を親として選択する。各親に対して、次のような処理が行われる。突然変異のために、選択された親を除く個体群から互いに異なる $1 + 2 \text{ num}$ 個の個体を選択する。最初の個体が基本ベクトル (base vector) となり、残りの個体対が num 個の差分ベクトルとなる。差分ベクトルは F (scaling factor) が乗算され基本ベクトルに加えることにより、変異ベクトル (mutant vector) が生成される。変異ベクトルと親が交叉し、 CR (crossover rate) により指定された確率で親の遺伝子を変異ベクトルの要素で置換することにより、子 (trial vector) が生成される。最後に、生存者選択として、子が親よりも良ければ、親を子で置換する。

DE には幾つかの形式があり、DE/base/num/cross という記法で表現される。“base” は基本ベクトルとなる個体の選択方法を指定する。例えば、DE/rand/num/cross は基本ベクトルのための個体を集団からランダムに選択し、DE/best/num/cross は集団の最良個体を選択する。“num” は差分ベクトルの個数を指定する。“cross” は子を生成するために使用する交叉方法を指定する。例えば、DE/base/num/bin は一定の確率で遺伝子を交換する二項交叉 (binomial crossover) を用い、DE/base/num/exp は、指数関数的に減少する確率で遺伝子を交換する指数交叉 (exponential crossover) を用いる。本研究では、差分ベクトル数を 1 ($\text{num} = 1$) とした DE/rand/1/exp を用いる。

3.2 ランク情報を用いた DE の改良

一般に多点探索法において探索効率を向上させるためには、収束性と多様性の間のバランスを取る必要があるため、収束性と多様性について考える。

- 収束性: 良好な点の近傍に新しい点を発生させることにより、収束性は向上する。DE においては、基本ベクトルを最良個体に設定する DE/best 戦略をとる方法、 F を小さくする方法が代表的であるが、いずれの方法も局所解に陥りやすいという問題がある。
- 多様性: 大きな範囲で新しい点を発生させることにより、多様性は向上する。DE においては、基本ベクトルをランダムに設定する DE/rand 戦略をとる方法、 F を大きくする方法が代表的であるが、収束性が低下してしまうという問題がある。

本研究では、ランク情報を利用して上記の方法を組み合わせることで収束性と多様性のバランスを取ることを提案する。

- 収束性の向上: 基本ベクトルが良好な個体である場合に、 F を小さくすることにより、良好な個体の近くに変異ベクトルを発生させる。さらに、親の更新速度を速くし、子に変異ベクトルに近くなるように、 CR を大きくする。
- 多様性の向上: 基本ベクトルが良好でない場合は、 F を大きくすることにより、大きな範囲で変異ベクトルを発生させる。さらに、親の更新速度を遅くし、急速な収束を避けるために、 CR を小さくする。

本研究では、このアイデアを実現するために、DE/rand 戦略を採用し、ランクに基づく線形的な F と CR の設定法を提案する。親を $\mathbf{x}^i$ 、基本ベクトルを $\mathbf{x}^b$ 、基本ベクトルのランクを R_b とすると、 $\mathbf{x}^i$ に対するスケーリングファクター F_i と交叉率 CR_i は以下のように定義できる。なお、最良個体のランクは 1 である。

$$F_i = F_{\min} + (F_{\max} - F_{\min}) \frac{R_b - 1}{N - 1}, \quad CR_i = CR_{\max} - (CR_{\max} - CR_{\min}) \frac{R_b - 1}{N - 1} \quad (8)$$

ただし、 $F_{\min}$, $F_{\max}$ は F の最小値、最大値を指定するパラメータ、 $CR_{\min}$, $CR_{\max}$ は CR の最小値、最大値を指定するパラメータである。これにより、基本ベクトルが最良個体の場合は F が最小 CR が最大になり、最悪個体の場合は F が最大 CR が最小となり、上記のアイデアを実現することができる。

3.3 ϵ RDE のアルゴリズム

Differential Evolution のパラメータ設定に探索点のランク情報を利用するアルゴリズムである ϵ 制約 Rank-based DE (ϵ RDE) を提案する. ϵ RDE/rand/1/exp のアルゴリズムは以下のように記述できる.

- (0) 初期化. N 個の初期個体 $\mathbf{x}^i$ を初期探索点として生成し, 初期集団 $P = \{\mathbf{x}^i, i = 1, 2, \dots, N\}$ を構成し, 全ての個体を評価する.
- (1) 終了判定. 世代数が最大世代数 $FE_{\max}$ を超えたとき, 実行を終了する.
- (2) ランキング. 各個体 $\mathbf{x}^i$ に対してランク R_i ($\mathbf{x}^i \in \{1, 2, \dots, N\}$) を付与する.
- (3) 突然変異と交叉. 全ての個体 $\mathbf{x}^i$ に対して, 下記の突然変異と交叉操作を行う.
- (3a) 突然変異. 各個体 $\mathbf{x}^i$ に対して, 3 個体 $\mathbf{x}^{p_1}, \mathbf{x}^{p_2}, \mathbf{x}^{p_3}$ を $\mathbf{x}^i$ および互いに重複しないようにランダムに選択する. 基本ベクトルのランク R_{p_1} により, 式 (8) に基づきスケールリングファクター F_i を決定する. 変異ベクトル $\mathbf{x}'$ を基本ベクトル $\mathbf{x}^{p_1}$ および差分ベクトル $\mathbf{x}^{p_2} - \mathbf{x}^{p_3}$ から以下のように生成する.

$$\mathbf{x}' = \mathbf{x}^{p_1} + F_i(\mathbf{x}^{p_2} - \mathbf{x}^{p_3}) \quad (9)$$

- (3b) 交叉. 基本ベクトルのランク R_{p_1} により, 式 (8) に基づき交叉率 CR_i を決定する. 変異ベクトル $\mathbf{x}'$ を親 $\mathbf{x}^i$ と交叉し, 子ベクトル $\mathbf{x}^{\text{child}}$ を生成する. 交叉点 j を全ての次元 $[1, n]$ からランダムに選択する. 子ベクトル $\mathbf{x}^{\text{child}}$ の j 番目の要素を $\mathbf{x}'$ の j 番目の要素から継承する. それ以降の次元は, 交叉率 CR_i によって指数関数的に減少する確率で, $\mathbf{x}'$ の要素から継承する. 残りの部分は, 親 $\mathbf{x}^i$ から継承する. 実際の処理では, (3a) と (3b) は一まとまりの処理で実現される.
- (4) 生存者選択. 子ベクトルを評価する. 子ベクトル $\mathbf{x}^{\text{child}}$ と親ベクトル $\mathbf{x}^i$ を ϵ レベル比較により比較し, 子ベクトルの方が良ければ, すなわち

$$(f(\mathbf{x}^{\text{child}}), \phi(\mathbf{x}^{\text{child}})) <_{\epsilon} (f(\mathbf{x}^i), \phi(\mathbf{x}^i)) \quad (10)$$

ならば, 子ベクトルが生存者となり, 親を子ベクトルで置換する. ここでは, 全ての親について操作を行った後に集団 P 全体を置換する離散世代モデルではなく, より効率が良いとされている親を子に直接置き換える連続世代モデルを用いる [5, 8].

- (5) (1) に戻る.

4 数値実験

ここでは, ペナルティ関数法を含んだ数多くの方法で解かれている制約付き最適化を取り上げる.

4.1 実験条件

実験条件は次の通りである. ϵ RDE の設定は, 個体数 $N = 20$, $F_{\min} = CR_{\min} = 0.7$, $F_{\max} = CR_{\max} = 1.0$ とし, 目的関数の最大評価回数 $FE_{\max}$ が 2,500 回, 5000 回, 10,000 回の 3 通りの場合について実験を行った. 本研究では, 各問題について 30 回独立に試行を行った.

4.2 Himmerblau Problem

Himmerblau 問題は以下のように定式化される。

$$\begin{aligned}
 &\text{Minimize} \quad f(\mathbf{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \\
 &\text{subject to} \\
 &\quad g_1(\mathbf{x}) = 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.0022053x_3x_5, \\
 &\quad g_2(\mathbf{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2, \\
 &\quad g_3(\mathbf{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4, \\
 &\quad 0 \leq g_1(\mathbf{x}) \leq 92, 90 \leq g_2(\mathbf{x}) \leq 110, 20 \leq g_3(\mathbf{x}) \leq 25, \\
 &\quad 78 \leq x_1 \leq 102, 33 \leq x_2 \leq 45, 27 \leq x_3, x_4, x_5 \leq 45.
 \end{aligned}$$

表 1 に実験結果を示す。本研究では、Himmerblau 問題について、13 種類の手法との比較を行った。 ϵ DE, ϵ PSO, ϵ GA は、自然アルゴリズムである DE, PSO, GA に対して ϵ 制約法を適応した手法である。 ϵ RDE 及びこれらの手法以外の結果は、文献 [9] に基づくものである。MGA は多目的 GA と解の優越関係によるアルゴリズム、Gen は GA を利用したアルゴリズム、GRG は Generalized Reduced Gradient 法、Death は death penalty 法、その他はペナルティに基づくアルゴリズムであり、ペナルティ係数を固定する static penalty, 探索ステップ数によりペナルティ係数を変化させる dynamic penalty, simulated annealing のように温度によりペナルティ係数を変化させる annealing penalty, 複数の探索点の状態によりペナルティ係数を決定する adaptive penalty, 解の集団と 2 種類のペナルティ係数のための集団を用いて共進化させる Coevolutionary penalty 法である。各アルゴリズムによる試行における最良値、平均値、最悪値および標準

表 1: Result of Himmerblau's problem

Algorithm	Best	Average	Worst	S.D.
ϵ RDE (2,500)	-31025.4405	-31023.3898	-31017.0982	1.9417e+00
ϵ RDE (5,000)	-31025.5601	-31025.5563	-31025.5311	5.1273e-03
ϵ DE (2500)	-31011.7391	-30979.33	-30925.707	20.4843
ϵ DE (5000)	-31025.0348	-31023.8356	-31018.9192	1.2120
ϵ PSO (5000)	-31011.999	-30947.3262	-30762.889	55.8631
ϵ GA (5000)	-30987.1366	-30945.5421	-30904.1387	19.9409
MGA	-31005.7966	-30862.8735	-30721.0418	73.240
Co-evolutionary	-31020.859	-30984.2407	-30792.4077	73.6335
Gen	-30183.576	N/A	N/A	N/A
GRG	-30373.949	N/A	N/A	N/A
Static	-30790.2716	-30446.4618	-29834.3847	226.3428
Dynamic	-30903.877	-30539.9156	-30106.2498	200.035
Annealing	-30829.201	-30442.126	-29773.085	244.619
Adaptive	-30903.877	-30448.007	-29926.1544	249.485
Death	-30790.271	-30429.371	-29834.385	234.555

偏差を示した。なお、 ϵ RDE および ϵ DE については関数評価回数が 2,500 回と 5,000 回の結果、 ϵ PSO, ϵ GA および MGA については 5,000 回の結果を示している。Co-evolutionary を除くその他については 50,000 回の結果であり、Co-evolutionary については 900,000 回の結果である。良い結果を示したアルゴリズムは ϵ 制約法を用いた ϵ RDE, ϵ DE, ϵ PSO, ϵ GA であり、特に DE に対して ϵ 制約法を適用した ϵ RDE と ϵ DE が安定的に良い結果を示している。関数評価回数が 5,000 回の ϵ RDE と他のすべての手法を比較すると、全ての項目で ϵ RDE の方が優れている。また、評価回数が 2,500 回の ϵ RDE と他のすべての手法を比較しても全ての項目で ϵ RDE が同等あるいは優れた性能を示している。したがって、 ϵ RDE は効率よく探索を行える安定したアルゴリズムであるといえる。

4.3 Welded Beam Design

梁の剪断応力 (τ), 曲げ応力 (σ), 台の座屈荷重 (P_c), 梁の端のたわみ (δ) などの制約の下で費用が最小となる梁の溶接を設計する. 図1のように, 4つの決定変数 $h(x_1)$, $l(x_2)$, $t(x_3)$, $b(x_4)$ により設計する. この問題は以下のように定式化される.

$$\begin{aligned} & \text{Minimize } f(\mathbf{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\ & \text{subject to} \end{aligned} \quad (11)$$

$$\begin{aligned} g_1(\mathbf{x}) &= \tau(\mathbf{x}) - \tau_{\max} \leq 0, \quad g_2(\mathbf{x}) = \sigma(\mathbf{x}) - \sigma_{\max} \leq 0, \quad g_3(\mathbf{x}) = x_1 - x_4 \leq 0, \\ g_4(\mathbf{x}) &= 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0, \quad g_5(\mathbf{x}) = 0.125 - x_1 \leq 0, \\ g_6(\mathbf{x}) &= \delta(\mathbf{x}) - \delta_{\max} \leq 0, \quad g_7(\mathbf{x}) = P - P_c(\mathbf{x}) \leq 0, \quad 0.1 \leq x_1, x_4 \leq 2, \quad 0.1 \leq x_2, x_3 \leq 10, \end{aligned}$$

where

$$\begin{aligned} \tau &= \sqrt{\tau'^2 + 2\tau'\tau''\frac{x_2}{2R} + \tau''^2}, \quad \tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J}, \quad M = P\left(L + \frac{x_2}{2}\right), \\ R &= \sqrt{\frac{x_2^2 + (x_1 + x_3)^2}{4}}, \quad J = 2\sqrt{2}x_1x_2\left(\frac{x_2^2}{12} + \frac{(x_1 + x_3)^2}{4}\right), \quad \sigma(\mathbf{x}) = \frac{6PL}{x_4x_3^2}, \quad \delta(\mathbf{x}) = \frac{4PL^3}{Ex_3^3x_4}, \\ P_c(\mathbf{x}) &= \frac{4.013E\sqrt{x_3^2x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right), \quad P = 6000lb, \quad L = 14in, \quad \delta_{\max} = 0.25in, \\ E &= 30 \times 10^6psi, \quad G = 12 \times 10^6psi, \quad \tau_{\max} = 13600psi, \quad \sigma_{\max} = 30000psi. \end{aligned}$$

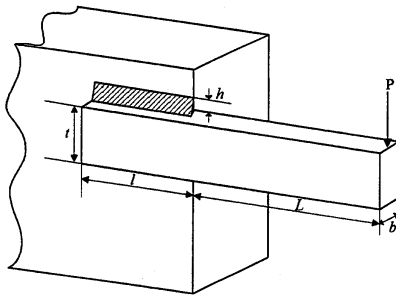


図 1: Welded beam design

表 2: Result of welded beam problem

Algorithm	Best	Average	Worst	S.D.
ϵ RDE (2500)	1.7261	1.7297	1.7469	0.0041
ϵ RDE (5000)	1.7249	1.7249	1.7249	0.0000
ϵ DE (2500)	1.7267	1.7339	1.7423	0.0039
ϵ DE (5000)	1.7249	1.7249	1.725	0.0000
ϵ PSO (5000)	1.7258	1.8073	2.1427	0.1200
ϵ GA (5000)	1.7852	1.8236	1.9422	0.0329
MGA	1.8245	1.9190	1.9950	0.05377
Co-evolutionary	1.7483	1.7720	1.7858	0.01122
Static	2.0469	2.9728	4.5741	0.6196
Dynamic	2.1062	3.1556	5.0359	0.7006
Annealing	2.0713	2.9533	4.1261	0.4902
Adaptive	1.9589	2.9898	4.84036	0.6515
Death	2.0821	3.1158	4.5138	0.6625

表2に実験結果を示す. 良い結果を示したアルゴリズムは ϵ RDE, ϵ DE, ϵ PSO である. 関数評価回数が 5,000 回の ϵ RDE とその他の結果を比較すると, 全ての項目で ϵ RDE の方が優れており, 非常に精度の高い最良解を見つけている. また, 評価回数が 2,500 回の ϵ RDE とその他の結果を比較すると, 評価回数が 5,000 回の ϵ DE を除くすべての手法に対して優れている. したがって, ϵ RDE は効率よく探索を行えるアルゴリズムであるといえる.

4.4 Pressure Vessel

半球状のキャップが両端に付いている円筒状の容器において, 材料, 形成, 溶接に必要なコストを最小化する問題である. 図2に示すように, T_s (シェルの厚み, x_1), T_h (キャップの厚み, x_2), R (内径, x_3), L (円筒状の長さ, x_4) の4変数を設計する. このうち, T_s と T_h は利用可能な筒状鋼板の厚みから, 0.0625 インチの整数倍である. この問題は以下のように定式化される.

表3に実験結果を示す. Deb は Genetic Adaptive Search, Kannan は拡張 Lagrange 乗数法, Sandgen は分枝限定法によるものである. なお, ϵ RDE については関数評価回数が 2,500 回と 5,000 回の結果, ϵ DE

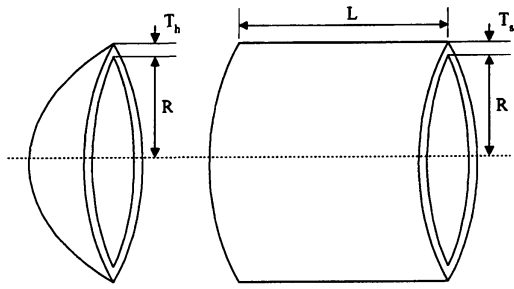


図 2: Pressure Vessel design

$$\begin{aligned} \text{Minimize } f(\mathbf{x}) = & 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 \\ & + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \end{aligned} \quad (12)$$

subject to

$$\begin{aligned} g_1(\mathbf{x}) &= -x_1 + 0.0193x_3 \leq 0, \\ g_2(\mathbf{x}) &= -x_2 + 0.00954x_3 \leq 0, \\ g_3(\mathbf{x}) &= -\pi x_3^2x_4 - 4\pi/3x_3^3 + 1296000 \leq 0, \\ g_4(\mathbf{x}) &= x_4 - 240 \leq 0, \\ x_1, x_2 &= 0.0625i, i \in \{1, 2, \dots, 99\}, \\ 10 &\leq x_3, x_4 \leq 200. \end{aligned}$$

については 5,000 回, 10,000 回の結果, ϵ PSO, ϵ GA および MGA については 50,000 回の結果を示している。Co-evolutionary を除くその他については 250,000 回の結果であり, Co-evolutionary については 900,000 回の結果である。

良い結果を示したアルゴリズムは ϵ RDE, ϵ DE である。関数評価回数が 5,000 回の ϵ RDE とその他の結果を比較すると, すべての項目で ϵ RDE の方が優れており, 良質な解を安定的に見つけている。また, 評価回数が 2,500 回の ϵ RDE についても, 評価回数が 5,000 回及び 10,000 回の ϵ DE とほぼ同等の結果が得られており, 評価回数が 50,000 回以上の他手法よりすべての項目で 2,500 回の ϵ RDE が優れている。したがって, ϵ RDE は効率よく探索を行えるアルゴリズムであるといえる。

表 3: Result of Pressure Vessel problem

Algorithm	Best	Average	Worst	S.D.
ϵ RDE (2500)	6060.073	6074.403	6104.089	14.1990
ϵ RDE (5000)	6059.714	6062.796	6090.526	9.2434
ϵ DE (5000)	6059.7144	6065.878	6090.5266	12.3242
ϵ DE (10000)	6059.7143	6065.8767	6090.5262	12.3248
ϵ GA (50000)	6074.6305	6172.9887	6335.7302	72.0513
ϵ PSO (50000)	6059.7143	6154.4386	6410.0868	132.6205
MGA	6069.3267	6263.7925	6403.4500	97.9445
Deb	6410.3811	N/A	N/A	N/A
Kannan	7198.0428	N/A	N/A	N/A
Sandgen	8129.1036	N/A	N/A	N/A
Co-evolutionary	6288.7445	6293.8432	6308.1497	7.4133
Static	6110.8117	6656.2616	7242.2035	320.8196
Dynamic	6213.6923	6691.5606	7445.6923	322.7647
Annealing	6127.4143	6660.8631	7380.4810	330.7516
Adaptive	6110.8117	6689.6049	7411.2532	330.4483
Death	6127.4143	6616.9333	7572.6591	358.8497

5 終わりに

DE は制約なし最適化問題に対する単純で高速かつ頑健な最適化アルゴリズムであることが知られている。一方, DE に対してアルゴリズム変換法である ϵ 制約法を適用した ϵ DE は, 制約付き最適化問題に対する効率的かつ安定的な最適化アルゴリズムとして知られている。本研究では, DE における制御パラメータであるスケーリングファクタ F と交叉率 CR について, 集団全体に同じ値を使用するのではなく, 基本ベクトルのランク情報に基づき, 個体毎に値を固定的に設定する新たなパラメータ値設定法を提案した。この設定法を ϵ DE に適用した ϵ RDE を提案し, ϵ 制約法を適用した ϵ DE, ϵ PSO, ϵ GA を含む他手法と比較することにより, ϵ RDE の有効性を示した。

今後は、 ϵ RDE における動的なパラメータ値設定法を検討するとともに、決定変数や制約条件が多い実問題への適用について研究する予定である。

謝辞 この研究の一部は、日本学術振興会科学研究費補助金 基盤研究 (c) (No. 22510166, 20500138) および広島市立大学特定研究費（一般研究）の援助を受けた。

参考文献

- [1] Storn, R. and Price, K.: Minimizing the Real Functions of the ICEC'96 Contest by Differential Evolution, in *Proc. of the International Conference on Evolutionary Computation*, pp. 842–844 (1996).
- [2] Storn, R. and Price, K.: Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, Vol. 11, pp. 341–359 (1997).
- [3] Chakraborty, U. K. ed.: *Advances in Differential Evolution*, Springer (2008).
- [4] Das, S. and Suganthan, P.: Differential Evolution: A Survey of the State-of-the-Art, *IEEE Transactions on Evolutionary Computation*, Vol. 15, No. 1, pp. 4–31 (2011).
- [5] Takahama, T. and Sakai, S.: Constrained Optimization by the ϵ Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites, in *Proc. of the 2006 IEEE Congress on Evolutionary Computation*, pp. 308–315 (2006).
- [6] Takahama, T. and Sakai, S.: Constrained Optimization by the ϵ Constrained Differential Evolution with an Archive and Gradient-Based Mutation, in *Proc. of the 2010 IEEE Congress on Evolutionary Computation*, pp. 1680–1688 (2010).
- [7] 高濱徹行, 阪井節子: 制約付き非線形最適化手法 α 制約法によるファジー制御ルールの最適化, *電子情報通信学会論文誌*, Vol. J82-A, No. 5, pp. 658–668 (1999).
- [8] Takahama, T. and Sakai, S.: Efficient Nonlinear Optimization by Differential Evolution with a Rotation-Invariant Local Sampling Operation, in *Proc. of the 2011 IEEE Congress on Evolutionary Computation*, pp. – (2011).
- [9] Coello, C. A. C.: Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art, *Computer Methods in Applied Mechanics and Engineering*, Vol. 191, No. 11-12, pp. 1245–1287 (2002).