数理解析研究所講究録
第1584巻 2008年 90-101

ε制約 Differential Evolutionによる制約付き最適化

広島修道大学商学部　阪井　節子 (Setsuko Sakai)
Faculty of Commercial Science, Hiroshima Shudo University
広島市立大学情報科学部　高濱　徹行 (Tetsuyuki Takahama)
Faculty of Information Sciences, Hiroshima City University

1 はじめに

進化的アルゴリズム (Evolutionary Algorithm, EA) は，生物進化の過程をモデル化した最適化アルゴリズムの総称であり，遺伝的アルゴリズム (Genetic Algorithm, GA) や進化戦略 (Evolution Strategy, ES) がよく知られている．EAは，最適化の対象である目的関数の値だけを利用して解を求めることができる直接探索法であり，例えば微分可能性のような目的関数に対する制限がなく，アルゴリズムの実装が容易であることから，様々な最適化問題を解くために利用されるようになってきている．EAは基本的に制約のない最適化問題を解くためのアルゴリズムであるが，実世界の最適化問題の多くは与えられた制約の下で目的関数を最適化する制約付き最適化問題である．このため，EAに基づく制約付き最適化法も盛んに研究され，従来の数理的方法を超える結果が得られることも示されてきている [1, 2].

EAによる制約付き最適化の研究を，制約条件の扱い方に着目して分類すると以下のような4種類に分類できる．

(1) 目的関数のみを最適化する方法
制約を満足する一つ以上の探索点を初期点として準備し，制約を満足する探索点のみを考慮してゆくことにより，制約の最適化を省略する方法であり，death penalty法とも呼ばれる．探索の過程で得られた点が制約を満足しない場合には，単純に無視されるか，制約を満足するように修正される．GAにおいて，制約を満足した探索点を参照して制約を満足しない探索点を修正する方法 [3] やパーティクルスォーム最適化 (Particle Swarm Optimization, PSO)[4] において，制約を満足しない探索点を無視し，既知の制約を満足する探索点に置換する方法 [5] も提案されている．これらの方法は制約領域が比較的広い場合には有効であるが，等式制約など制約の厳しい問題では，初期点を準備したり探索点を修正することが非常に困難である．

(2) 目的関数と制約逸脱度 (constraint violation) の荷重和を最適化する方法
複数の制約条件を組み合わせて制約逸脱度を定義し，目的関数と制約逸脱度の荷重和を求め，その荷重和の一目的最適化問題として解く方法である．制約逸脱度は目的関数に対するペナルティと考えられるため，この方法は一般にペナルティ関数法 (penalty function method) と呼ばれる．ペナルティ関数法では，制約逸脱度の強さを調整するための荷重であるペナルティ係数 (penalty coefficient) を適切に選択することが困難であるという問題点がある．ペナルティ係数が大きいと，制約を満足する解は得られるが，目的関数の最適化が不十分になり，質の高い解を得ることが困難になる．逆にペナルティ係数が小さいと，目的関数は最適化されるが，制約の最適化が不十分になり，実行可能解を得ることが困難になる．ペナルティ係数を動的に調整する方法 [6]，探索の進行状況に応じてペナルティ係数を適応的に制御する方法 [7, 8] などが提案されている．しかし，これらの方法はペナルティ係数を自動的に調整するために時間がかかり，多くの計算量が必要となる．また，良好な結果を得るためには他のアルゴリズムパラメータを調整する必要があり，同じパラメータ設定で多くの問題に対応するのは困難である．

(3) 目的関数と制約逸脱度を分離して最適化する方法
この範疇における代表的な方法は，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方法である．GAにおいて，制約を満足しない探索点の適合度を，集団中の制約を満足する探索点における最悪の目的関数値と制約逸脱度との和として与える方法が提案されている [9]．ESにおいて，単に制約逸脱度を優先するのではなく，ある確率で制約逸脱度を無視し目的関数のみで比較を行うという拡張された辞書式比較により最適化を行うstochastic ranking法 [10] や，制約逸脱度を優先するが，制

約逸脱度が悪いが目的関数値が良い解も次世代に残す方法 [11] が提案されている．より一般的な方法として，直接探索法全般に対して，制約条件を緩和することができる辞書式比較である α レベル比較を使用する α 制約法 [12, 13, 14, 15] および ε 比較を使用する ε 制約法 [16] が提案されている．α 制約法および ε 制約法は，直接探索法における比較演算子を α レベル比較および ε レベル比較に置換することにより，制約のない問題に対する最適化アルゴリズムを制約付き問題に対する最適化アルゴリズムに変換する方法，すなわちアルゴリズム変換法である．α および ε 制約法は，制約条件を緩和することにより，等式制約を含むような制約条件の厳しい問題に対しても適用することができる．

また，辞書式比較以外の方法として，最初に制約のみを最適化して実行可能解を探索し，次に目的関数を最適化するという 2 段階の最適化法 [17] も提案されている．

この範疇の方法は，多様な問題に対して比較的良好な結果を得られることが示されている．

(4) 目的関数と各制約の多目的問題として解く方法
目的関数と一般に複数の制約関数を多目的最適化問題として解く方法である [18, 19]．制約が複雑な問題に有効であると期待されるが，多目的最適化問題は一目的問題と比較すると非常に困難な問題であり，一般に多くの計算量を必要とするという問題点がある．

以上で述べた EA に基づく制約付き最適化法には一般に以下のような問題がある．

1. 多峰性問題に対する探索性能が不十分である
制約付き最適化のための EA は，単峰性問題に対して実行可能領域を探索し，実行可能解を発見することができる．しかし，実行可能領域に多数の局所解がある多峰性問題を解くときは，たとえ実行可能領域を発見できても，局所解に陥り，最適解の探索ができなくなることがしばしばある．

2. 等式制約を持つ問題における解の実行可能性が不十分である
制約付き最適化に対する多くの EA は，等式制約を持つ問題を直接解くことができない．そのため，等式制約を不等式制約に緩和して不等式制約の問題に変換する必要がある．結果として，得られた解の実行可能性が不十分となる．そのため，等式制約を動的に緩和し，制約逸脱度が最小である解を発見できる方法が望まれる．

3. 探索の安定性と効率性が低い
同じ問題を対象にした場合でも，ランダムに発生した初期探索点集合の状態によって，良好な解を発見できる場合もあれば，かなり劣った解しか発見できない場合もあることが多い．したがって，平均的に安定して精度の高い近似解を発見できる方法が必要である．また，多くの EA では，ランキング選択や個体の入替，確率的選択，Gauss 分布や Cauchy 分布に基づく突然変異など，計算コストの高い操作が用いられる．より簡単な計算操作による最適化が可能な効率的な方法が望まれる．

本研究では，これらの問題を解決するために，差分進化 (Differential Evolution, DE) に対して ε 制約法を適用した εDE を提案する．DE では，ランダムに 1 親個体を選択し，この個体と新たに選んだ別の 2 個体の差の重みづけられた和を求めるという簡単な操作で交叉と突然変異を置き換える．DE を用いることによって，問題 1. と 3. が解決できる．等式制約問題を直接解くために，ε 制約法に対して等式制約の緩和を制御する簡単で新しい方法を提案することにより，問題 2. を解決できる．これらによって，εDE は多峰性問題や等式制約問題を解く安定的，かつ効率的な探索を実現する．本研究では，13 個のテスト問題を解き，得られた結果を他の方法と比較することによって，εDE の有効性を示す．

本研究は次のように構成されている．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 法を DE に組み込んだ εDE 法を説明する．4. で実験結果を示す．5. はまとめである．

2 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，不等式制約，等式制約，上下限制約を有しており，以下のように定義できる．

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

ここで，$\boldsymbol{x} = (x_1, \cdots, x_n)$ は n 次元決定変数ベクトル，$f(\boldsymbol{x})$ は目的関数，$g_j(\boldsymbol{x}) \leq 0$ は q 個の不等式制約，$h_j(\boldsymbol{x}) = 0$ 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，以下では全ての制約を満足する領域を実行可能領域 $\mathcal{F}$，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ために，制約逸脱度 $\phi(\boldsymbol{x})$ を導入する．制約逸脱度 $\phi(\boldsymbol{x})$ は，以下を満足する関数である．

$$\begin{cases} \phi(\boldsymbol{x}) = 0\ (\boldsymbol{x} \in \mathcal{F}) \\ \phi(\boldsymbol{x}) > 0\ (\boldsymbol{x} \notin \mathcal{F}) \end{cases} \tag{2}$$

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．

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

$$\phi(\boldsymbol{x}) = \sum_j \max\{0, g_j(\boldsymbol{x})\}^p + \sum_j |h_j(\boldsymbol{x})|^p \tag{4}$$

ただし，p は正数である．

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，制約逸脱度が ε 以下の場合は目的関数値の大小関係を優先し，それ以外の場合は制約逸脱度の大小関係を優先する比較である ε レベル比較を定義する．

点 $\boldsymbol{x}_1, \boldsymbol{x}_2$ 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，通常の大小関係である $<, \leq$ に対応する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 $<_\varepsilon$, $\leq_\varepsilon$ $(\varepsilon \in [0, \infty))$ は以下のようになる．

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

$$(f_1, \phi_1) \leq_\varepsilon (f_2, \phi_2) \quad \Leftrightarrow \quad \begin{cases} f_1 \leq f_2, \text{ if } \phi_1, \phi_2 \leq \varepsilon \\ f_1 \leq f_2, \text{ if } \phi_1 = \phi_2 \\ \phi_1 < \phi_2, \text{otherwise} \end{cases} \tag{6}$$

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 ($\mathrm{P}_{\leq_\varepsilon}$)，すなわち，$\varepsilon$ 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，$\mathrm{minimize}_{\leq_\varepsilon}$ は $\leq_\varepsilon$ の意味での最小化である．

$$(\mathrm{P}_{\leq_\varepsilon}) \quad \mathrm{minimize}_{\leq_\varepsilon} \quad f(\boldsymbol{x}) \tag{7}$$

ここで，問題 (P) の制約条件を $\phi(\boldsymbol{x}) \leq \varepsilon$ 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．

$$\begin{array}{lll} (\mathrm{P}^\varepsilon) & \mathrm{minimize} & f(\boldsymbol{x}) \\ & \text{subject to} & \phi(\boldsymbol{x}) \leq \varepsilon \end{array} \tag{8}$$

問題 (P^ε) と問題 ($\mathrm{P}_{\leq_\varepsilon}$)，および問題 (P) に関して以下の定理が成り立つ．

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．$f(\boldsymbol{x})$, $\phi(\boldsymbol{x})$ を連続関数とし，問題 (P^0) に最適解 $\boldsymbol{x}^*$ 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 $\hat{\boldsymbol{x}}_n$ が存在すると仮定する．このとき，点列 $\{\hat{\boldsymbol{x}}_n\}$ の任意の集積点は問題 (P^0) の最適解である．

定理 1, 2 は，ε レベル比較を行うことにより，制約付き問題が等価な制約なし問題に変換されることを示している．したがって，既存の制約なし問題に対する最適化法に ε レベル比較を導入することにより，制約付き問題を解くことが可能となる．定理 3 は，ペナルティ法においてペナルティ係数を ∞ まで増加させるのと同様に，ε を 0 まで減少させながら最適化を行っても，最適解が得られることを示している．

α 制約法では，制約逸脱度の代わりに区間 $[0,1]$ の制約満足度を用い，制約満足度が 1 の場合に実行可能解とする．したがって，α 制約法と ε 制約法は理論的には等価である．しかし，計算機上においては，一般に 1 に近い数を表現する場合よりも 0 に近い値を表現する場合の方が表現精度が高いため，α レベルを制御する場合よりも ε レベルを制御する場合の方が制約に対する計算精度が高くなるという利点を持つ．

3 ε 制約 Differential Evolution

Differential Evolution に ε 制約法を適用したアルゴリズムである ε 制約 Differential Evolution(εDE) を説明する．

3.1 差分進化 (Differential Evolution)

Differential evolution (DE) は進化戦略 (evolution strategy) の一つであり，Storn and Price[20, 21] によって提案された．DE は，解集団を用いた多点探索を行う確率的直接探索法である．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，高速で頑健なアルゴリズムであることが示されている．

DE の重要な特徴として，進化戦略におけるガウス突然変異のステップ幅のような制御が不要で，単純な数学的演算が用いられる点が挙げられる．一般に，ガウス突然変異における理想的なステップ幅は，遺伝子あるいは各次元毎に異なり，また進化の状態によっても異なるため，何らかの方法でステップ幅を適応的に調整する必要がある．これに対し，DE はガウス突然変異の代わりに，基本ベクトル (base vector) と差分ベクトル (difference vectors) との重み付き和を突然変異として採用している．集団から選択された 1 個体が基本ベクトルとなり，集団からランダムに選択された個体対の差が差分ベクトルとなる．世代を経るに従

い，解集団が探索空間中で収縮したり拡張したりすることにより，差分ベクトルが変化し，差分ベクトルとして与えられる各次元におけるステップ幅が自動的に調整されるのである．

DE には幾つかの形式が提案されており，DE/best/1/bin や DE/rand/1/exp などがよく知られている．これらは，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親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親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*" は基本ベクトルを変異させるための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交叉 (exponential crossover) を用いる．

DE では，探索空間中にランダムに初期個体を生成し，初期集団を構成する．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．現在，親として選択された個体を除く個体群から互いに異なる $1+2\,num$ 個の個体を選択する．最初の個体が基本ベクトルとなり，残りの個体対が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られる．その結果得られたベクトルと親が交叉し，CR(crossover factor) により指定された確率で親の遺伝子をベクトルの要素で置換することにより，子のベクトル (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．

本研究では，差分ベクトル数を 1 ($num = 1$) とした DE/rand/1/exp を用いる．

3.2 ε 制約 DE のアルゴリズム

ε 制約 DE/rand/1/exp のアルゴリズムは以下のように記述できる [22, 23].

Step0 初期化．探索空間 S 内に，初期個体をランダムに N 個体生成し，初期集団 $P(0) = \{\boldsymbol{x}^i, i = 1, 2, \cdots, N\}$ を構成する．また，ε レベル制御関数 $\varepsilon(0)$ を与える．

Step1 終了判定．世代数が最大世代数 $T_{\max}$ を超えたとき，実行を終了する．

Step2 突然変異．各個体 $\boldsymbol{x}^i$ に対して，3 個体 $\boldsymbol{x}^{p1}$, $\boldsymbol{x}^{p2}$, $\boldsymbol{x}^{p3}$ を $\boldsymbol{x}^i$ および互いに重複しないようにランダムに選択する．新しいベクトル $\boldsymbol{x}'$ を基本ベクトル $\boldsymbol{x}^{p1}$ および差分ベクトル $\boldsymbol{x}^{p2} - \boldsymbol{x}^{p3}$ から以下のように生成する．

$$\boldsymbol{x}' = \boldsymbol{x}^{p1} + F(\boldsymbol{x}^{p2} - \boldsymbol{x}^{p3}) \tag{9}$$

ここで，F はスケーリングファクタである．

Step3 交叉．ベクトル $\boldsymbol{x}'$ を親 $\boldsymbol{x}^i$ と交叉し，子ベクトル $\boldsymbol{x}_i^{\text{new}}$ 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 $\boldsymbol{x}_i^{\text{new}}$ の j 番目の要素を $\boldsymbol{x}'$ の j 番目の要素から継承する．それ以降の次元は，交叉パラメータ CR によって指数関数的に減少する確率で，$\boldsymbol{x}'$ の要素から継承する．残りの部分は，親 $\boldsymbol{x}_i$ から継承する．実際の処理では，Step2 と Step3 は一まとまりの処理で実現される．

Step4 生存者選択．子ベクトル $\boldsymbol{x}_i^{\text{new}}$ を評価する．$\boldsymbol{x}_i^{\text{new}}$ が親ベクトル $\boldsymbol{x}^i$ よりも良けれ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．

Step5 ε レベルの制御．ε レベル制御関数 $\varepsilon(t)$ によって，ε レベルを更新する．

Step6 Step1 に戻る．

以下に擬似コードを示す．

```
εDE/rand/1/exp()
{
  P(0)=Generate N individuals {x^i} randomly;
  ε=ε(0);
```

```
for(t=1; t ≤ T_max; t++) {
  for(i=1; i ≤ N; i++) {
    (p1, p2, p3)=select randomly from [1, N]
                s.t. p1, p2, p3, i が全て異なる;
    x^new=x^i ∈ P(t − 1);
    j=select randomly from [1, n];
    k=1;
    do {
      x_j^new=x_j^p1+F(x_j^p2 − x_j^p3);
      j=(j + 1)%n;
      k++;
    } while(k ≤ n && u(0, 1) < CR);
    if((f(x^new), φ(x^new)) <_ε (f(x^i), φ(x^i)))
      z^i=x^new;
    else
      z^i=x^i;
  }
  P(t)={z^i, i = 1, 2, ···, N}
  ε=ε(t);
 }
}
```

ここで，$\varepsilon(t)$ は ε レベルを制御する ε レベル制御関数，F はスケーリングファクタ，CR は交叉パラメータ，$u(0,1)$ は区間 $[0,1]$ の一様乱数生成関数である．

3.3 ε レベルの制御

本研究では，等式制約を含む最適化問題に対して，次のような制御方法を採用する．ε レベルを制御する際には，そのレベル以下の個体が常にある程度含まれるようにしながら 0 まで減少させ，0 になった後もある程度の最適化を行うことが望ましい．そこで，初期値 $\varepsilon(0)$ を初期集団の制約逸脱度の良い個体の上位 20%番目の個体の制約逸脱度とし，最大世代数 T の 80%以降は常に 0 となる，以下のような世代 t のベキ乗関数による制御を提案する．

$$\begin{aligned} \varepsilon(0) &= \phi(x_\theta) \\ \varepsilon(t) &= \begin{cases} \varepsilon(0)(1-\frac{t}{T_c})^{cp}, & 0 < t < T_c, \\ 0, & t \geq T_c \end{cases} \end{aligned} \tag{10}$$

ただし，x_θ は制約逸脱度が上位 20%番目の個体 ($\theta = 0.2N$)，T_c は最大世代数の 80%世代 ($T_c = 0.8T_{\max}$) とし，ベキ乗の係数 cp は通常 5 とする．

4 εDE による制約付き最適化

本研究では，文献 [10, 24] をはじめとする幾つかの研究で取り上げられている 13 個の制約付き非線形最適化問題を取り上げ，εDE により得られた結果を文献 [24] による結果と比較する．

4.1 テスト問題と実験条件

13 個の制約付き非線形最適化問題 g01～g13 の内，g02, g03, g08, g12 は最大化問題であり，その他は最小化問題である．g03, g05, g11, g13 は等式制約を含む問題である．また，g12 は半径 0.25 の 9^3 個の円形制約領域を持つ，非連結な制約領域を持つ問題である [25]．

表 1 に各問題の概略を示す [11, 26]. 各問題について，決定変数の数 (n)，目的関数の形式，線形不等式制約 (LI)，非線形不等式制約 (NI)，線形等式制約 (LE)，非線形等式制約 (NE) の数，制約式の値が 0 となる有効制約の数 (active) を示した．また，最大化問題を上矢印 (↑) で示した．

表 1: テスト問題の概略

f	n	Form of f	LI	NI	LE	NE	active
g01	13	quadratic	9	0	0	0	6
↑ g02	20	nonlinear	1	1	0	0	1
↑ g03	10	polynomial	0	0	0	1	1
g04	5	quadratic	0	6	0	0	2
g05	4	cubic	2	0	0	3	3
g06	2	cubic	0	2	0	0	2
g07	10	quadratic	3	5	0	0	6
↑ g08	2	nonlinear	0	2	0	0	0
g09	7	polynomial	0	4	0	0	2
g10	8	linear	3	3	0	0	6
g11	2	quadratic	0	0	0	1	1
↑ g12	3	quadratic	0	9^3	0	0	0
g13	5	nonlinear	0	0	1	2	3

ε 制約法に関するパラメータは，次のように設定する．制約逸脱度は，式 (4) において $p = 1$ とする単純和を用いる．ε レベルについては，等式制約を含まない問題では $\varepsilon(t) = 0$ に固定する．等式制約を含む問題については式 (10) により ε レベルを制御する．ただし，ベキ乗の係数 $cp = 5$ とする．DE に関するパラメータは，次のように設定する．すべての問題に対して，個体 (探索点) 数 $N = 40$，スケーリングファクタ $F = 0.7$, 交叉率 $CR = 0.9$ とした．最大世代数 $T_{\max}$ は，g12 を除くすべての問題に対して $T_{\max} = 4,999$ とし，g12 では $T_{\max} = 499$ とした．したがって，目的関数の最大評価回数 $(T_{\max} + 1) \times N$ は，g12 以外の問題では 200,000 回 (g12 では，20,000 回) となる．本研究では，このような試行を 30 回行った．

4.2 実験結果

13 種類の問題について，十分な統計データが示されており，その他の方法と比較して良好な結果を残している研究として，Runarsson and Yao の Stochastic Ranking(SR) 法 [10]，Mezura-Montes and Coello の Simple Multimembered Evolution Sturategy(SMES) 法 [11]，高濱，阪井の突然変異を有する α 制約 Simplex(α Simplex) 法 [24] がある．ここでは，εDE と SR および α Simplex との性能比較を行った．

SR においては，個体数 $N = 200$ とし，g12 を除く問題について最大世代数 $T_{\max} = 1749$ まで，問題 g12 は最大世代数 $T_{\max} = 174$ まで，30 回の試行を行っている．このため目的関数の最大評価回数は，$N * (T_{\max} + 1) = 200 \times 1750 = 350,000$ 回 (g12 は $200 \times 175 = 35,000$ 回) となる．また，等式制約は多くの方法で提案されている式 (11) に基づいて緩和し，不等式制約に置き換えた問題に対する結果を示している．

$$|h_j(x)| \leq \delta,\ \delta > 0, \tag{11}$$

ただし，$\delta = 10^{-4}$ とした．α Simplex においては，g12 を除く問題については目的関数の最大評価回数は約 290,000 ~ 330,000 回，g12 については約 30,000 回であり，30 回の試行を行っている．なお，εDE では，等式制約を持つ問題すべてに対して同じ ε レベルの制御を行った．εDE および α Simplex では，他手法のように等式制約を不等式制約に緩和する必要はなく，制約付き問題を直接に解くことができる．しかも，制約逸脱度が 10^{-4} よりかなり小さい最適解の近似解を発見できる．しかし，ここでは，文献 [24] と比較するために，(11) において $\delta = 10^{-4}$ として，緩和された制約付き問題とした結果を示している．

表 2 と表 3 にεDE, SR, α Simplex で得られた平均値と標準偏差を示した．

表 2: εDE, SR, α Simplex の平均値の比較

f	optimal	εDE	SR	αSimplex
g01	-15.00	-15.000	-15.000	-15.000
↑g02	0.803619	**0.803613**	0.781975	0.784187
↑g03	1.00	**1.001**	1.000	**1.001**
g04	-30665.539	-30665.539	-30665.539	-30665.539
g05	5126.498	**5126.497**	5128.881	**5126.497**
g06	-6961.814	-6961.814	-6875.940	-6961.814
g07	24.306	**24.306**	24.374	**24.306**
↑g08	0.095825	0.095825	0.095825	0.095825
g09	680.630	**680.630**	680.656	**680.630**
g10	7049.248	**7049.248**	7559.192	**7049.248**
g11	0.750	0.750	0.750	0.750
↑g12	1.000000	1.000000	1.000000	1.000000
g13	0.053950	**0.053942**	0.067543	0.066770

表 3: εDE, SR, αSimplex の標準偏差の比較

f	εDE	SR	αSimplex
g01	**0**	**0**	6.4e-06
↑g02	**5.6e-06**	2.0e-02	1.3e-02
↑g03	6.5e-09	1.9e-04	**8.5e-14**
g04	**0**	2.0e-05	4.2e-11
g05	**0**	3.5	3.5e-11
g06	**0**	1.6e+02	1.3e-10
g07	**4.3e-09**	6.6e-02	1.3e-04
↑g08	**0**	2.6e-17	3.8e-13
g09	**0**	3.4e-02	2.9e-10
g10	**0**	5.3e+02	4.7e-06
g11	**0**	8.0e-05	4.9e-16
↑g12	**0**	**0**	3.9e-10
g13	**0**	3.1e-02	6.9e-02

g01, g04, g06, g08, g11 と g12 に対しては，すべての手法が 30 回の全試行で，最適解を発見している．平均値については，εDE が g02 と g13 に対して他のすべての手法より優れており，g03, g05, g07, g09, g10 に対しては，SR より優れている．安定性を表す標準偏差については，g03 を除くすべての問題においてεDE が他の手法より優れている．

εDE は，g02 を除くすべての問題に対して，比較的少ない評価回数で，平均的に優れたあるいは同等の解を発見できている．したがって，平均的には，εDE が他の手法に比べてより優れた探索性能を持ち，安定した最適化アルゴリズムであることが示された．

5 おわりに

DE は，進化戦略の 1 つであり，制約なし最適化問題に対する簡単で効率的かつ頑健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 ε 制約法を適用した制約付き最適化アルゴリズムであるεDE を提案した．さらに，数値的最適化が非常に困難な等式制約のある問題を解くために，等式制約を不等式

制約に緩和するのではなく，等式制約の緩和を制御する簡単な方法を提案した．εDE は，13 個のテスト問題を非常に高速に解くことができることを示した．さらに，制約付き最適化問題に対して高い性能を示す Stochastic Ranking 法および α Simplex 法と比較することによって，εDE が，他の手法より効率的で安定的な最適化アルゴリズムであることを示した．

今後は，アルゴリズムパラメータの設定に関する検討を行う．また，多数の決定変数や制約条件を持つ実際的な様々な問題に対してεDE を適用してゆきたいと考えている．

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. 17510139, 16500083) の援助のもとで行われた．

補遺

g01: minimize $f(\boldsymbol{x}) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i,$

subject to $g_1(\boldsymbol{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0,\quad g_2(\boldsymbol{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0,$
$g_3(\boldsymbol{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10, \leq 0,\quad g_4(\boldsymbol{x}) = -8x_1 + x_{10} \leq 0,$
$g_5(\boldsymbol{x}) = -8x_2 + x_{11} \leq 0,\quad g_6(\boldsymbol{x}) = -8x_3 + x_{12} \leq 0,\quad g_7(\boldsymbol{x}) = -2x_4 - x_5 + x_{10} \leq 0,$
$g_8(\boldsymbol{x}) = -2x_6 - x_7 + x_{11} \leq 0,\quad g_9(\boldsymbol{x}) = -2x_8 - x_9 + x_{12} \leq 0,$
$0 \leq x_i \leq 1\,(i = 1, \cdots, 9),\ 0 \leq x_i \leq 100\,(i = 10, 11, 12),\ 0 \leq x_{13} \leq 1$

最適解 $\boldsymbol{x}^* = (1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 3,\ 3,\ 3,\ 1)$, **最適値** $f(\boldsymbol{x}^*) = -15.$

g02: maximize $f(\boldsymbol{x}) = \left(\left|\sum_{i=1}^{n} \cos^4 x_i - 2\prod_{i=1}^{n} \cos^2 x_i\right|\right) \Big/ \left(\sqrt{\sum_{i=1}^{n} i\, x_i^2}\right),$

subject to $g_1(\boldsymbol{x}) = 0.75 - \prod_{i=1}^{20} x_i \leq 0,\quad g_2(\boldsymbol{x}) = \sum_{i=1}^{20} x_i - 7.5n \leq 0,$
$0 \leq x_i \leq 10\,(i = 1, \cdots, n),\ n = 20$

最適値は未知．既知の最良値 $f(\boldsymbol{x}) = 0.803619$ [10].

g03: maximize $f(\boldsymbol{x}) = (\sqrt{n})^n \prod_{i=1}^{n} x_i,$

subject to $h_1(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2 - 1 = 0,\quad 0 \leq x_i \leq 1\,(i = 1, \cdots, n),\quad n = 10$

最適解 $x_i^* = \dfrac{1}{\sqrt{n}}\,(i = 1, \cdots, n)$, **最適値** $f(\boldsymbol{x}^*) = 1.$

g04: minimize $f(\boldsymbol{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141,$

subject to $g_1(\boldsymbol{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_592 \leq 0,$
$g_2(\boldsymbol{x}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0,$
$g_3(\boldsymbol{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3x_3 - 110 \leq 0,$
$g_4(\boldsymbol{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3x_3 + 90 \leq 0,$
$g_5(\boldsymbol{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0,$
$g_6(\boldsymbol{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0,$
$78 \leq x_1 \leq 102,\quad 33 \leq x_2 \leq 45,\quad 27 \leq x_i \leq 45\,(i = 3, 4, 5)$

最適解 $\boldsymbol{x}^* = (78,\ 33,\ 29.995256025682,\ 45,\ 36.775812905788)$, **最適値** $f(\boldsymbol{x}^*) = -30665.539.$

g05: minimize $f(\boldsymbol{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + \dfrac{0.000002}{3}x_2^3,$

subject to $g_1(\boldsymbol{x}) = x_3 - x_4 - 0.55 \le 0,\quad g_2(\boldsymbol{x}) = -x_3 + x_4 - 0.55 \le 0,$

$h_3(\boldsymbol{x}) = 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0,$

$h_4(\boldsymbol{x}) = 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0,$

$h_5(\boldsymbol{x}) = 1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0,$

$0 \le x_i \le 1200\,(i = 1, 2),\quad -0.55 \le x_i \le 0.55\,(i = 3, 4)$

最適値は未知．既知の最良値は $f(\boldsymbol{x}) = 5126.4981.$

g06: minimize $f(\boldsymbol{x}) = (x_1 - 10)^3 + (x_1 - 20)^3,$

subject to $g_1(\boldsymbol{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0,$

$g_2(\boldsymbol{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0,$

$13 \le x_1 \le 100,\quad 0 \le x_2 \le 100$

最適解 $\boldsymbol{x}^* = (14.095,\ 0.84296)$，最適値 $f(\boldsymbol{x}^*) = -6961.81388.$

g07: minimize $f(\boldsymbol{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2$

$+2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45,$

subject to $g_1(\boldsymbol{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0,\ g_2(\boldsymbol{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0,$

$g_3(\boldsymbol{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0,$

$g_4(\boldsymbol{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \le 0,$

$g_5(\boldsymbol{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0,$

$g_6(\boldsymbol{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0,$

$g_7(\boldsymbol{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \le 0,$

$g_8(\boldsymbol{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0,$

$-10 \le x_i \le 10\,(i = 1, \cdots, 10)$

最適解 $\boldsymbol{x}^* = (2.171996,\ 2.63683,\ 8.773926,\ 5.095984,\ 0.9906548,\ 1.430574,\ 1.321644,\ 9.828726,$
$8.280092,\ 8.375927)$，最適値 $f(\boldsymbol{x}^*) = 24.306209.$

g08: maximize $f(\boldsymbol{x}) = \dfrac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)},$

subject to $g_1(\boldsymbol{x}) = x_1^2 - x_2 + 1 \le 0,\ g_2(\boldsymbol{x}) = 1 - x_1 + (x_2 - 4)^2 \le 0,\ 0 \le x_i \le 10\,(i = 1, 2)$

最適解 $\boldsymbol{x}^* = (1.2279713,\ 4.2453733)$，最適値 $f(\boldsymbol{x}^*) = 0.095825.$

g09: minimize $f(\boldsymbol{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7$

$-10x_6 - 8x_7,$

subject to $g_1(\boldsymbol{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \le 0,$

$g_2(\boldsymbol{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \le 0,$

$g_3(\boldsymbol{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \le 0,$

$g_4(\boldsymbol{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0,$

$-10 \le x_i \le 10\,(i = 1, \cdots, 7)$

最適解 $\boldsymbol{x}^* = (2.330499,\ 1.951372,\ -0.4775414,\ 4.365726,\ -0.6244870,\ 1.038131,\ 1.594227),$
最適値 $f(\boldsymbol{x}^*) = 680.6300573.$

g10: minimize $f(\boldsymbol{x}) = x_1 + x_2 + x_3,$

subject to $g_1(\boldsymbol{x}) = -1 + 0.0025(x_4 + x_6) \le 0,\ g_2(\boldsymbol{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \le 0,$

$g_3(\boldsymbol{x}) = -1 + 0.01(x_8 - x_5) \le 0,$

$g_4(\boldsymbol{x}) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \le 0,$

$g_5(\boldsymbol{x}) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \le 0,$

$g_6(\boldsymbol{x}) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \le 0,$

$100 \le x_1 \le 10000,\ 1000 \le x_i \le 10000\,(i = 2,\ 3),\ 10 \le x_i \le 1000\,(i = 4, \cdots, 8)$

最大値は未知．既知の最良値は $f(\boldsymbol{x}) = 7049.3307$.

g11: minimize $f(\boldsymbol{x}) = x_1^2 + (x_2 - 1)^2$,
subject to $h(\boldsymbol{x}) = x_2 - x_1^2 = 0,\ -1 \leq x_i \leq 1\,(i = 1, 2)$

最適解 $\boldsymbol{x}^* = (\pm\frac{1}{\sqrt{2}}, \frac{1}{2})$, 最適値 $f(\boldsymbol{x}^*) = 0.75$.

g12: maximize $f(\boldsymbol{x}) = \frac{1}{100}\{100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2\}$
subject to $g(\boldsymbol{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0,$
$0 \leq x_i \leq 10\,(i = 1, 2, 3),\ \ p, q, r = 1, 2, \cdots, 9$

最適解 $\boldsymbol{x}^* = (5,\ 5,\ 5)$, 最適値 $f(\boldsymbol{x}^*) = 1$.

g13: minimize $f(\boldsymbol{x}) = e^{x_1 x_2 x_3 x_4 x_5}$,
subject to $h_1(\boldsymbol{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0,\ h_2(\boldsymbol{x}) = x_2 x_3 - 5x_4 x_5 = 0,$
$h_3(\boldsymbol{x}) = x_1^3 + x_2^3 + 1 = 0,\ -2.3 \leq x_i \leq 2.3\,(i = 1, 2),\ -3.2 \leq x_i \leq 3.2\,(i = 3, 4, 5)$

最適解 $\boldsymbol{x}^* = (-1.717143,\ 1.595709,\ 1.827247,\ -0.7636413,\ -0.763645)$, 最適値 $f(\boldsymbol{x}^*) = 0.0539498$.

参考文献

[1] Z. Michalewicz: "Genetic algorithm + data structures = evolution programs 3rd ed.", Springer-Verlag, Berlin (1996).

[2] C. A. C. Coello: "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, **191**, pp. 1245–1287 (2002).

[3] Z. Michalewicz and M. Schoenauer: "Evolutionary algorithms for constrained parameter optimization problems", Evolutionary Computation, **4**, 1, pp. 1–32 (1996).

[4] J. Kennedy and R. C. Eberhart: "Swarm Intelligence", Morgan Kaufmann, San Francisco (2001).

[5] K. E. Parsopoulos and M. N. Vrahatis: "Particle swarm optimization method for constrained optimization problems", Intelligent Technologies — Theory and Application: New Trends in Intelligent Technologies (Eds. by P. Sincak, J. Vascak and et al.), Vol. 76 of Frontiers in Artificial Intelligence and Applications, IOS Press, pp. 214–220 (2002).

[6] J. Joines and C. Houck: "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs", Proceedings of the first IEEE Conference on Evolutionary Computation (Ed. by D. Fogel), Orlando, Florida, IEEE Press, pp. 579–584 (1994).

[7] D. Coit, A. E. Smith and D. M. Tate: "Adaptive penalty methods for genetic optimization of constrained combinatorial problems", INFORMS Journal on Computing, **8**, 2, pp. 173–182 (1996).

[8] S. B. Hamida and M. Schoenauer: "An adaptive algorithm for constrained optimization problems", Parallel Problem Solving from Nature – PPSN VI (Eds. by M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo and H.-P. Schwefel), Berlin, Springer, pp. 529–538 (2000).

[9] K. Deb: "An efficient constraint handling method for genetic algorithms", Computer Methods in Applied Mechanics and Engineering, **186**, 2/4, pp. 311–338 (2000).

[10] T. P. Runarsson and X. Yao: "Stochastic ranking for constrained evolutionary optimization", IEEE Trans. on Evolutionary Computation, **4**, 3, pp. 284–294 (2000).

[11] E. Mezura-Montes and C. A. C. Coello: "A simple multimembered evolution strategy to solve constrained optimization problems", IEEE Trans. on Evolutionary Computation, **9**, 1, pp. 1–17 (2005).

[12] 高濱, 阪井 : "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, **J82-A**, 5, pp. 658–668 (1999).

[13] T. Takahama and S. Sakai: "Learning fuzzy control rules by α-constrained simplex method", The Transactions of the Institute of Electronics, Information and Communication Engineers, **J83-D-I**, 7, pp. 770–779 (2000). in Japanese.

[14] T. Takahama and S. Sakai: "Constrained optimization by α constrained genetic algorithm (αGA)", The Transactions of the Institute of Electronics, Information and Communication Engineers, **J86-D-I**, 4, pp. 198–207 (2003). in Japanese.

[15] T. Takahama and S. Sakai: "Constrained optimization by the α constrained particle swarm optimizer", Journal of Advanced Computational Intelligence and Intelligent Informatics, **9**, 3, pp. 282–289 (2005).

[16] T. Takahama and S. Sakai: "Constrained optimization by ε constrained particle swarm optimizer with ε-level control", Proc. of the 4th IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST'05), pp. 1019–1029 (2005).

[17] S. Venkatraman and G. G. Yen: "A generic framework for constrained optimization using genetic algorithms", IEEE Trans. on Evolutionary Computation, **9**, 4, pp. 424–435 (2005).

[18] E. Camponogara and S. N. Talukdar: "A genetic algorithm for constrained and multiobjective optimization", 3rd Nordic Workshop on Genetic Algorithms and Their Applications (3NWGA) (Ed. by J. T. Alander), Vaasa, Finland, University of Vaasa, pp. 49–62 (1997).

[19] P. D. Surry and N. J. Radcliffe: "The COMOGA method: constrained optimisation by multiobjective genetic algorithms", Control and Cybernetics, **26**, 3, pp. 391–412 (1997).

[20] R. Storn and K. Price: "Minimizing the real functions of the ICEC'96 contest by differential evolution", Proc. of the International Conference on Evolutionary Computation, pp. 842–844 (1996).

[21] R. Storn and K. Price: "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, **11**, pp. 341–359 (1997).

[22] T. Takahama, S. Sakai and N. Iwane: "Solving nonlinear constrained optimization problems by the ε constrained differential evolution", Proc. of the 2006 IEEE Conference on Systems, Man, and Cybernetics, pp. 2322–2327 (2006).

[23] T. Takahama and S. Sakai: "Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites", Proc. of the 2006 World Congress on Computational Intelligence, pp. 308–315 (2006).

[24] T. Takahama and S. Sakai: "Constrained optimization by applying the α constrained method to the nonlinear simplex method with mutations", IEEE Transactions on Evolutionary Computation, **9**, 5, pp. 437–451 (2005).

[25] S. Koziel and Z. Michalewicz: "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization", Evolutionary Computation, **7**, 1, pp. 19–44 (1999).

[26] R. Farmani and J. A. Wright: "Self-adaptive fitness formulation for constrained optimization", IEEE Trans. on Evolutionary Computation, **7**, 5, pp. 445–455 (2003).