領域適応型遺伝的アルゴリズムの開発*

(精度および求解性の向上のためのオペレータの提案)

荒川 雅生^{*1}, 萩原 一郎^{*2}

Development of Adaptive Range Genetic Algorithms (Proposal of the New Operators for Efficient and High Accurate Solutions)

Masao ARAKAWA^{*3} and Ichiro HAGIWARA^{*3} Kagawa University, Faculty of Engineering, 1-1 Saiwaicho, Takamatsu, Kagawa, 760-8526 Japan

In design problems, design variables are roughly divided into three types; integer, discrete and continuous. And in practical design, they are composed by mixed variables of these three types of design variables. And these types of the problems are hard to optimize, because there exist no particular method to treat these kinds of problems. Although genetic algorithms have been said to be the powerful tools for mixed optimization, it cannot treat continuous variables by its nature and most of the cases they cannot search global solution right away. In order to overcome these situation, we have been developed the adaptive range genetic algorithms, which will change the searching range adaptively from the information obtained in the previous generation and showed its effectiveness to mixed optimization. However, there are some lack in certainty in finding global optimum and also the accuracy will change in each trial. In this study, we will propose some new operators to raise certainty to obtain global optimum and also to reduce the number of function call. We demonstrate the proposed method to two type of bench mark test and show the effectiveness of the proposed method.

Key Words: Design Engineering, Genetic Algorithms, Optimization, Mixed Problem

1. 緒 言

現在、遺伝的アルゴリズム (GAs) は非常に幅広い分野で利用されるようになっているが、その理由を考えてみると、

- 1) 多点同時探索である。すなわち、暗黙のうちに並列探索をしているため局所的最適解に陥りにくい。
- 2) 感度情報を利用しない。すなわち、感度が得られない (整数問題、離散変数問題) や得られにくい問題への適用に適している。
- 3) コーディングが容易である。
- 4) 集団として適応度が向上する。
- 5) 最適な、あるいは、満足のいく範囲で準最適解を導きだせる。

などが挙げられよう。GAs は上記のような理由ゆえに、本来得意としない分野にまで適用された感があり、その多くの研究が、GAs が本質的に持つ評価回数多さを補うために本来の特質を失うような性質を付加させられてきていたように思われる。例えば、筆者らが行った 2 次元構造物の位相決定問題⁽¹⁾においては部材の存在の有無の 2 値問題を各有限要素に割り当てた組合せ最適化問題として解いているが、このような問題に対しては、簡単なルールベースの方法^(2,3,4)のほうがより効果的で、解析回数も少なく済む。その上、均質化法⁽⁵⁾のような強力な数理的手法が存在す

るので、実際には GAs の利用に適していない事例と考えるほうが妥当であろう。

GAs の利用に適している問題には、よく言われているように、スケジューリング問題があり、そして本質的に離散変数の性質をもった問題に対する組合せ最適化問題が上げられる。それ以外では、感度が正確に求められない問題、多峰性が激しく、感度を用いた手法では解が求められない問題などが考えられよう。また、GAs は実際の設計問題においては、本数、規格値、材質など様々な整数、離散変数が存在し、その影響で従来からある数理計画法を使った最適化アプローチが困難であったような問題には非常に有効であり、実務設計への応用を可能とするものであった⁽⁶⁾。しかしながら、単純 GAs が取り扱えるのは基本的には離散変数のみであり、連続変数に対しては多くの情報の欠落を伴うものであった。実際の設計問題には、上記のような離散変数のほかにも連続変数が加わってくることが普通である。このように各種の変数からなる混合最適化に対する有力な解法は存在せず、最適化の研究の中で今後取り扱われなければならない問題として指摘されていた⁽⁷⁾。GAs はその有力な候補ではあるものの、連続変数を取り扱えないという弱点を抱えていた。連続変数を取り扱うためには、例えば古川ら⁽⁸⁾の方法に見られるように遺伝子として連続量を取り扱わなければならない。このような対応は非常に有効であるものの、本来の GAs が持つ、遺伝子型

* 原稿受付 1998 年 12 月 10 日。

^{*1} 正員, 香川大学 (☎ 760-8526 高松市幸町 1-1)。^{*2} 正員, 東京工業大学 (☎ 152-8552 東京都目黒区大岡山 2-12-1)。

E-mail: arakawa@eng.kagawa-u.ac.jp

での最適化の操作, 表現型での評価という枠組みから多少逸脱するものであり, また, 離散変数との混合問題を考えた場合には, 2通りのアルゴリズムを併用する形になり, 必ずしも好ましくないように思われる.

筆者らは各世代の人口分布の状況を基に, 探索領域を自動的に変化させながら連続変数を表現する手法を開発し, その有効性を検討した⁽⁹⁾. 提示した手法は0, 1のバイナリ表現を基にしたGAsであり, 単純GAsの枠組みを守ったものである. さらに, 初期的に与える領域やビット数に関らず, 精度良く最適解を探索することができ, GAsの枠組の中で, 変数の性質から来る制約をほとんどなくすものであった. 筆者らはさらに整数, 離散変数に対しても領域適応型GAsの開発を進め, 混合最適化問題に提示した手法を適用し, その有効性を検討した⁽¹⁰⁾. その結果, 精度面, 求解の面いずれをとっても従来からある各種の手法と比較して優れた結果を得ることができ, 提示した手法の有効性の一端を確認することができた.

提示した手法は確かに非常に強力な手法のようであるが, 以下のような問題点も持ちあわせていた.

- 1) 試行毎に得られる結果のバラツキ
- 2) 制約条件付き問題の場合にペナルティ関数の設定
- 3) 設計変数の数への対応
- 4) 解析回数の問題
- 5) 確実に精度良く解を求めること

いずれの課題も手法の開発の当初には, 単純GAsの枠組みで連続変数を精度良く取り扱う大前提の下では, 問題にならなかった項目であり, 改良の過程にある⁽¹¹⁾. 本研究では個々の問題を手法の本質的な性質を悪戯にいじることなく, また, できる限り個別の問題にのみ当てはまる対策とならないような各種の改良を加えた. 提示した手法を混合設計変数問題のベンチマーク問題^(12,13), 及び, AIAAのホームページで紹介されたベンチマーク問題^(14,15)に適用し, その有効性を検討した.

2. 領域適応型GAs

詳細は, 参考文献^(9,10,11)に譲るものとし, ここでは既に示した基本的な手法, 各種パラメータについて極く簡単に示すものとする.

2.1 連続変数に対して 本手法では初期世代では従来までの手法と同じである. すなわち, 上下限値を設定し, それをビット数にあわせて等分割するものとする. 第2世代目からは, 前世代におけるGAsの操作の終了後に生き残った個体の前世代における各設計変数の平均値 μ_i と標準偏差 σ_i を用いて

$$R(p_i) = \begin{cases} \mu_i - \sqrt{-2\sigma_i^2 \ln \left(LB + \frac{(UB-LB)C(p_i)}{2^{m_i-1}-1} \right)} & \text{for } C(p_i) < 2^{m_i-1} \\ \mu_i + \sqrt{-2\sigma_i^2 \ln \left(LB + \frac{(UB-LB)(C(p_i)-2^{m_i-1})}{2^{m_i-1}-1} \right)} & \text{for } C(p_i) \geq 2^{m_i-1} \end{cases} \quad (1)$$

と表される. この概略は図1に示すとおりであり, システムパラメータLB, UBの間を等間隔に分割し, それぞれのy座標に対応するx座標の値をもって対応する p_i の連続変数とするものである. なお, システムパラメータにはLB, UBの他に, σ_i の上下限値 $\sigma_{max}, \sigma_{min}$ を設けている. σ_{max} は, 初期集団において探索領域がいたずらに大きくなるのを防ぐ目的で, σ_{min} は収束していったときに探索領域が狭まりすぎるのを防ぐ目的で設定したものである. 収束性の向上と安定へ向けた工夫として, 最良解の保証と外来種突然変異を導入している.

最良解の保証とは, 過去の世代において最も適応度の高かった解の組合せを必ず探索領域内に残しておくための操作で,

$$\begin{aligned} & \text{if } \text{Max value}_i > \mu_i + \sqrt{-2\sigma_i \ln(LB)} \\ & \text{then } \sigma_{i, \text{new}}^R = \frac{\text{Max value}_i - \mu_i}{\sqrt{-2\sigma_i \ln(LB)}} \\ & \text{if } \text{Max value}_i < \mu_i - \sqrt{-2\sigma_i \ln(LB)} \\ & \text{then } \sigma_{i, \text{new}}^L = \frac{-\text{Max value}_i + \mu_i}{\sqrt{-2\sigma_i \ln(LB)}} \end{aligned} \quad (2)$$

ここに, Max value_i とは, 最良解の各設計変数における実数値であり, $\sigma_{i, \text{new}}^R, \sigma_{i, \text{new}}^L$ はそれぞれ平均値の右側, 左側に新しく与えられたパラメータである. また, 外来種突然変異とは, 従来までのように特定のビットの反転を行うのではなく, 全く新しい個体をランダムに発生させて集団の中に残す方法である. こうすることにより, 多様性を保証し, 世代ごとに変化する探索領域に適応させることを目的としている. さらに, 各設計変数毎に, ある程度の幅を維持したいため, 各設計変数の標準偏差が σ_{min} を下回った場合には, 対応するビット列に関してのみ外来種突然変異をいれることとした. これは, 探索において必要ない部分への影響をなるべくなくし, かつ, 個体群の多様性を増

Fig. 1 Illustrative sketch of adaptive range expression of continuous variables

すための試みである⁽¹¹⁾。

この手法では、各世代において探索領域が変動していく。そのために、場合によっては探索領域が各設計変数の持つ側面制約を破ってしまう可能性がある。このような場合、単純に適応度関数の中にペナルティ関数として付加することも可能である。しかしながら、一般に制約条件の取り扱いが騙し問題に見られるように必ずしも、容易でないために、この対応では、本質的な改善は不可能である。そこで、各変数ごとに状況に応じてシステムパラメータを操作することにより、探索領域が必ず側面制約 ($upper_i, lower_i$) を守るような工夫を加えるために、以下のような操作を行った。

for lower bound

$$LB_{i,new}^L = \exp\left(-\frac{(lower_i - \mu_i)^2}{2\sigma_i^2}\right)$$

if $LB_{i,new}^L > UB - margin$ then $LB_{i,new}^L = UB - margin$ and (3)

$$\sigma_{i,new}^L = \frac{lower_i - \mu_i}{\sqrt{-2\ln(LB_{i,new}^L)}}$$

for upper bound

$$LB_{i,new}^R = \exp\left(-\frac{(upper_i - \mu_i)^2}{2\sigma_i^2}\right)$$

if $LB_{i,new}^R > UB - margin$ then $LB_{i,new}^R = UB - margin$ and (4)

$$\sigma_{i,new}^R = \frac{upper_i - \mu_i}{\sqrt{-2\ln(LB_{i,new}^R)}}$$

ここで、 $LB_{i,new}^R$, $LB_{i,new}^L$ とはそれぞれ平均値の右側、左側の限界値を決定するために新しく設定されたパラメータであり、 $\sigma_{i,new}^R$, $\sigma_{i,new}^L$ はそれぞれ平均値の右側、左側に新しく与えられたパラメータである。margin とは $[LB, UB]$ 間の任意の値であり、LB だけの操作では、 $LB_{i,new}^R$, $LB_{i,new}^L$ が UB を上回る可能性があるため、それを回避する目的で設定したものである。LB_{i,new} が margin の値を越えたときは、 σ_i の値も同時に操作することで前世代の平均値 μ_i と上限の側面制約 $upper_i$ あるいは下限の側面制約 $lower_i$ との間で分割ができるように設定の変更を行っている。

2. 2 離散変数に対して

本研究では、式 (5) に示されるように $x_{i,min}$ の値を前世代の整数値の平均値 μ_i を基に操作する手法を提示する。すなわち、整数においては、

$$DI(p_i) = x_{i,min} + C(p_i) \quad (5)$$

である。ここで、

$$x_{i,min} = \text{Int}(\mu_i + 0.5) - 2^{m-i} \quad (6)$$

もし、この値が、設定した上下限界を越えてしまう場合、致死遺伝子を生じさせないために、

$$\text{if } x_{i,min} < lower_i \text{ then}$$

$$x_{i,min} = lower_i \quad (7)$$

$$\text{else if } x_{i,min} + 2^{m-i} > upper_i \text{ then}$$

$$x_{i,min} = upper_i - 2^{m-i} + 1$$

とすればよいことがわかる。連続変数の場合と同様に最良解を保証する場合、

$$\text{if } x_{i,min} > \text{Max value}_i \text{ then}$$

$$x_{i,min} = \text{Max value}_i \quad (8)$$

$$\text{else if } x_{i,min} + 2^{m-i} < \text{Max value}_i$$

$$x_{i,min} = \text{Max value}_i - 2^{m-i} + 1.$$

とすればよいことがわかる。

さらに、所与の離散値を扱う場合、式 (8) で与えられた整数値をさす離散値を与えればよい。すなわち、

$$DC(p_i) = \text{Database}[[DI(p_i)]] \quad (9)$$

となる。

この手法を用いる場合、従来まで問題になっていた離散点の数とビット数の不適合によって生じる致死遺伝子の問題点が生じないなどの利点があり、これは、収束性の向上へ向けて大きな寄与をするものと考えられる。

3. 提示する改良方法

3. 1 σ_{min} の設定方法

従来までの手法では、探索する変数の大きさによらず、 σ_{min} の値は一定であった。世代が進み収束に近づくにつれ、一般的には各設計変数の標準偏差値は小さくなり、 σ_{min} で一定となる。各世代において領域の幅は標準偏差値とシステムパラメータ LB によって決定されるために、結果的に終盤においては一定の幅の領域で解を探索することになり、精度の向上が望めなくなる。したがって、解の精度は σ_{min} の設定値次第ということになる。特に設計変数の絶対値が小さい問題では、容易に標準偏差が σ_{min} を下回ってしまうため、場合によっては従来までの手法と大差が無くなっていた。そこで、探索する設計変数の大きさを反映させるために、以後システムパラメータの推奨値として

$$\{\sigma_{max}, \sigma_{min}, LB, UB, margin, \epsilon\} = \{2.0, 1.0, 0.044, 0.99, 0.2, 0.001\}$$

とし、標準偏差の上下限界は、設計変数の探索領域 w (一般には上下限界の幅、上下限界がない場合には、初期的に与えた幅) を考慮し、

$$\sigma_{min} = \sigma_{min} * w / 10 \quad (10)$$

$$\sigma_{max} = \sigma_{max} * w / 10 \quad (11)$$

とすることにした。こうすることで設計者はシステムパラメータの設定から解放される。

さらに、収束が進んだ状態では、さらに精度の向上が望まれる。ここで、収束の判定が問題となるが、一定の世代数最良値が変化しない場合に σ_{min} を小さくすることとした。

$$\sigma_{min} = \sigma_{min} * a \quad (0 < a < 1.0) \quad (12)$$

ただし、有効な探索領域を確保するために、影響は小

さいもののLBの値も小さくすることとした。

$$LB=LB*a \quad (0<a<1.0) \quad (13)$$

この対応は、計算機によっても異なるが、計算機エラーが生じるまで続けることが可能である。こうすることによって、自動的に精度の向上が望まれる。

3. 2 最良値周りの探索 提示した手法では探索領域が変動するために、最良値と各世代における最大値とは大きく異なる場合がある。これは、図2のような状態に陥った場合であり、従来までは突然変異によってこの状態の解消を待っていた。しかしながら、多くの場合過去の世代を通じて得られた最良解にはそれなりの意味があるものと考え、その近傍に一定数の個体のみ探索領域を設定することにした。具体的には、

$$\mu_i = \text{Max Value}_i \quad (14)$$

$$\sigma_i = (0.1 \sim 10) * \sigma_{\min} \quad (15)$$

とした。ここで、標準偏差値は $\sigma_{\min}$ の0.1倍から10倍までをランダムにとれるものとした。

3. 3 領域の保存 提示した手法では各世代ごとに探索領域が可変していた。つまり、同じ遺伝子でも世代によって異なった適応度をもつために、世代毎でいちいち適応度の評価をしなければならなかった。そのため、評価回数はおのずと多くなってしまっていた。実際の生物を見てみると、親と子ではほとんど同じ遺伝子を持っていても、現れる形は当然異なり、このことは自然なことである。しかしながら、このままでは、遺伝子操作後に生き残った個体（親）までもが、次世代で表現体が異なることになる。これは実際の生物を見ても不自然である。そこで、領域の変更は各世代毎で、その世代で新しく生まれた個体、突然変異によって表現体が変わる個体のみを対象とすることにした。こうすることで、適応度の高い個体は数世代に渡って生存が可能となり、個体群としての多様性を維持するうえでも、有益であると考えられる。

3. 4 ペナルティ関数の可変調整 G A sでは評価関数は適応度のみであり、一般的には制約条件は含まれない。そこで、制約条件付き問題への対応が必要になってくるが、例えば、免疫アルゴリズム⁽¹⁶⁾や劣性遺伝子⁽¹⁷⁾を考える等の対策が見られるものの、一般的にはペナルティ関数を設定して、適応度関数の中に入れていく。多くの研究事例では一見するといとも簡

Fig.2 Operator to add special range for Max Value

単にペナルティ関数を決めているようであるが、実際にはペナルティ関数の設定は精度問題、求解性において非常に重要であり、設計者への大きな負担となっている。例えば、ペナルティ係数を必要以上に大きく設定してしまうと、騙し問題を発生してしまい、なかなか最適解が見つからなかったりする。そこで、式(16, 17)のような可変調整法を提示する。

$$fitness = f(x) + Coef \sum_{j=1}^M P[1.0 - g_j(x)]^{a - (a-1/a)gen/cgen} \quad (16)$$

ここで、Coefとはペナルティ係数であり、本研究では経験的に初期世代における人口の中で目的関数値の絶対値が最も大きいものの0.4倍に設定した。この値は本来、初期的には制約条件を多少破った個体でも高い評価を与え、最終的には厳しい評価を与える大きさならば任意に設定できるものである。P[.]はペナルティ関数であり、関数内が0以下ならば0である。式(16)の場合、制約条件が1.0以上であることを基本に定式化を考えている。ペナルティ関数の指数部分では、世代によって図3のように変化させている。cgenは指数を何世代まで変化させるのかを決めるパラメータであり、genは0世代からcgen世代までの間を段階的に変化させるように、

$$gen = \text{Int}(i / \text{Int}(cgen/div)) \quad (17)$$

で与えられる。ここで、Intは整数部分を取り上げる関数であり、iは世代数、divは分割数を表す。ただし、世代数がcgenを越えた場合、gen=cgenで一定とする。

3. 5 ノイズの制御 遺伝的アルゴリズムの収束において、大局的な解を求めるために、評価関数のノイズが世代の初期段階では、強力な武器になりうるということが予測される⁽¹⁸⁾。そこで、本研究では、初期段階において乱数を発生させ、ある一定の世代後の発生を行わない手法を提示する。乱数の発生法として一様に小さい乱数を発生する方法と、一定の世代まで線形に徐々にノイズを小さくしていく方法の2通りを考えてみた。この手法を前節に示したペナルティ関数の制御の有無の両方に適用した。ノイズを乗せることによって以下のような効果が期待される。

1. 個体の多様性を維持できる。

Fig. 3 Changes in penalty function

2. 環境の変化に対してロバストである。
 3. ノイズの大きさを徐々に減らすことによって、大域的な解に導きやすい。これは、初期的な段階では制約条件を多少破っていても許容する場合があります、そのような個体が大域的な解近傍に導き、その後、正確に評価をすることによって精度を上げると考えられるからである。

評価関数にノイズを乗せるということは表現体に操作を加えるだけであり、遺伝的アルゴリズムの本質である遺伝子には何の操作も加えていない。したがって、この対策は突然変異とは根本的に異なるものである。

4. 数値計算例

提示した手法の有効性を検討するために以下の2題に適用し、従来法との比較を通じて有効性の検討を行った。

4. 1 問題1 混合最適化問題のベンチマーク問題である圧力器問題を取り上げる。詳細は文献(10, 12, 13)に譲るものとして定式化は以下の通りである。

Find

- R: inner radius(continuous)
 L: length of the cylindrical section(continuous)
 T_s : thickness of shell(discrete)
 T_h : thickness of head(discrete)

Constraints

- g1: minimum shell wall thickness
 g2: minimum head wall thickness
 g3: maximum length
 g4: minimum volume of tank

Side constraints

- $R_{LB} \leq R \leq R_{UB}$
 $L_{LB} \leq L \leq L_{UB}$
 $T_{SLB} \leq T_s \leq T_{SUB}$
 $T_{HLB} \leq T_h \leq T_{HUB}$

Minimize

$$f = 37981.2T_sRL + 108506.3T_hR^2 + 193207.3T_s^2L + 1210711T_s^2R \quad (18)$$

離散変数と連続変数が含まれており、また、それが故に多峰性の激しい問題となっている。

4. 2 問題2 混合最適化問題ではあるが、離散変数が一つであり、比較的多峰性のすくない問題としてGolinskiの変速機問題^(14, 15)を取り扱った。

- x_1 : width of the gear face, cm
 x_2 : teeth module, cm
 x_3 : number of pinion teeth (integer)
 x_4 : shaft 1 length between bearings, cm
 x_5 : shaft 2 length between bearings, cm
 x_6 : diameter of shaft 1, cm
 x_7 : diameter of shaft 2, cm

であり、目的関数はギアボックスの体積の最小化で

$$f = C_{11}x_1x_2^2(C_{12}x_3^2 + C_{13}x_3 - C_{14}) - C_{15}(x_6^2 + x_7^2)x_1 + C_{16}(x_6^3 + x_7^3) + C_{17}(x_6^2 + x_7^2) \quad (19)$$

ここで、

$$C_{11} = 0.7854 \quad C_{12} = 3.3333 \quad C_{13} = 14.9334$$

$$C_{14} = 43.0934 \quad C_{15} = 1.5079 \quad C_{16} = 7.477$$

である。設計変数には側面制約があり、

$$2.6 \leq x_1 \leq 3.607 \leq x_2 \leq 0.817 \leq x_3 \leq 28$$

$7.3 \leq x_4 \leq 8.373 \leq x_5 \leq 8.3 \quad 2.9 \leq x_6 \leq 3.950 \leq x_7 \leq 5.5$
 である。制約条件は、

g_1 : upper bound on the bending stress of the gear tooth

g_2 : upper bound on the contact stress of the gear tooth

g_3, g_4 : upper bounds on the transverse deflection of shafts 1, 2

g_5, g_6 : upper bounds on the stresses in shaft 1, 2

g_7, g_8, g_9 : dimensional restriction based on space and experiment

g_{10}, g_{11} : dimensional requirements for shafts based on experience

であり、

$$g_1 = x_1 x_2^2 x_3 / 27.0$$

$$g_2 = x_1 x_2^2 x_3^2 / 397.5$$

$$g_3 = x_2 x_3 x_6^4 / 1.93 x_4^3$$

$$g_4 = x_2 x_3 x_7^4 / 1.93 x_5^3$$

$$g_5 = \frac{110.0 x_1'}{\sqrt{\left(\frac{745 x_1}{x_2 x_3}\right)^2 + 16.9}} \times 10^7$$

$$g_6 = \frac{85.0 x_1'}{\sqrt{\left(\frac{745 x_1}{x_2 x_3}\right)^2 + 157.5}} \times 10^7$$

$$g_7 = 40.0 / x_2 x_3 \quad g_8 = x_1 / 850.0 x_2$$

$$g_9 = 5.0 x_2 / x_1$$

$$g_{10} = x_4 / (1.5 x_6 + 1.9)$$

$$g_{11} = x_5 / (1.1 x_7 + 1.9)$$

である。この問題は、全て連続変数として解かれた結果があり、整数が一つしか含まれておらず、しかも、下限値で最適解を得ているため、残念ながら x_3 を連続変数として取り扱っても問題はなかった。

4. 3 GAsの設定 いずれの例題も基本的な設定として、特に記述のない場合には各設計変数につき4ビット、一世代あたり80個体、交叉率70%、突然変異率1%、外来種率2.5%、1設計変数あたりの外来種率2.5%、最良値周りの探索率10%（使用しない場合0%）、エリート保存率35%、4段階のペナルティ関数の変更とし、初期的に設定した領域には、最終的な解（世代を1200世代まで続けた場合に得られた最良の結果として、問題1で適応度関数値が5850.38、問題2で2994.49を得ている）が入っていない条件とした。

4. 4 σ_{min} の設定方法の有無による比較 問題2を用いて同じ条件で以下の場合について解いた。

case 1:従来法

case 2:提示した設定法

case 3:case 2+絞り込み有

いずれも二回試行したが、傾向は変わらなかった。そこで、それぞれの場合で収束性の悪かった場合を比較する。（図4）case 1と2の比較から、適正な設定をすることで、求解の早さが増していることがわかる。また、Case 2と3の比較から、精度の向上に絞り込みが与える影響が少なからずあることがわかる。図中のAzarmの結果を見ると、計算機の精度、収束の判定条件の問題もあるが絞り込むことで数値計画法の結果よりも精度良く解を求めていることがわかり、提示した手法の有効性の一端が確認できる。

4. 5 最良値周りの探索の効果 最良値周りの探索の効果をみるために、上記case 2と同じ設定で、最良値周りの探索をいれてた場合を設定した。

case 4: with local search on "Max Value"

その結果を図5に示す。明らかに最良値と世代の最大値の隔たりが減っていることがわかる。このことから、当初の目的は達せられたものと思われる。さらに、この図以外の試行においても、求解の精度面でも向上する傾向が認められた。

4.6 領域の保存 領域を保存することによる効果をみるために、領域の保存に関するオペレータの有無を問題2を用いて比較する。前節までの設定では、このオペレータは利用しておらず、以後の設定では利用するものとする。

case 5: with local search on "Max Value", remain region

この場合の結果を図6に示す。また、各世代において必要となった解析回数の比較を図7に示す。問題2の場合、いずれの試行においても解析回数はほぼ3/4倍、問題1では2/3倍になっており、合理的に解析回数を減らせていることがわかる。また、収束の状況を見ても、最良値と世代の最大値の隔離はcase4の場合と比較して少なく、安定していることがわかる。これは、隔離が起こるのが、突然変異により最良値を持った個体の変更されてしまう場合に限ることもあり、極く自然な事である。また、そのことによって収束の状況は安定し、より高い精度に早期にたどり着けるものと考えられる。

4.7 ペナルティ関数の自動設定 ペナルティ関数の自動設定による影響を見るために、問題1を用いて以下のような5通りの場合に適用した。

case 6: without penalty function control

Fig. 4 Comparison of "Max Value" of case 1, 2, and 3

Fig. 5 Comparison of process of case 2, and 4

case 7: control up to 100 generation, without step

case 8: control up to 200 generation, without step

case 9: control up to 100 generation, with step

case 10: control up to 200 generation, with step

それぞれの場合で15回試行したときの結果の比較を表1に示す。収束の状況の変化の様子を最良値の場合で比較したものを図8に示す。case 6はペナルティ指数の変更が内容に全ての場合で1/2.5乗している。ペナルティ係数は初期世代の最大値の0.4倍としており、その他の場合と同じである。この場合、表1からもわかるように、なかなか大局的最適解を見いだせていないことがわかる。このことは、ペナルティ関数の設定が如何に難しいかを物語るものと考えられる。次に、case7とcase8では段階ごとにペナルティ指数を変更するのではなく毎世代毎に変化をさせたものである。大局的最適解を見いだす確率はペナルティ指数の変更が緩やかなほうが確実であることがわかる。case7の場合で大局的最適解へ近づいた場合の有無の比較をすると、case 6とは近づき方が異なることが見て取れる。すなわち、case 6の場合は、最初から高いペナルティが課せられているために徐々に適応度が改善され、偶然よい解が見つかったときに、これに引っ張られて大局的最適解の近傍へと近づいている。これに対して、それ以外の場合では、初期段階においては制約条件を破るものも許容している。ペナルティ指数部分の変更にともない徐々に最良値が減少しているのは徐々にペナルティを大きく課していつているからである。この傾向が図中aと記した部分である。成功している場合はいずれも、制約条件は破るものの適応度の高い個体が最良解と見なされており、一端適応度が下がるものの(図中cの状態)、その個体の近傍で制約条件を守り、適応度の高い個体を見だし、これを基に精度の向上へとつなげていつている。一方、大局的最適解を見いだせていない場合は、図中cのような落ち込み

Fig. 6 Process of the results in case 5

Fig. 7 Comparison of the number of function call

Table 1 Comparison of the results

trial	case 6	case 7	case 8	case 9	case 10
1	6059.72	5850.38	5879.63	6059.96	5862.35
2	6370.94	6418.80	5850.39	5850.41	5850.38
3	5850.99	5850.38	5852.71	5853.19	5851.95
4	6424.06	5850.38	5864.63	5852.73	5850.38
5	6101.26	6075.91	5850.42	5871.70	5850.38
6	6067.22	6092.15	5852.47	5850.49	5857.21
7	6111.38	5883.07	5850.39	6090.53	5850.38
8	6059.74	6059.71	5850.44	6059.84	5856.47
9	6090.57	6144.32	6059.93	6113.22	5850.38
10	6165.58	6078.41	5853.04	6381.66	5852.84
11	5854.15	6107.30	6031.76	5852.23	5850.48
12	5850.39	5850.67	5896.91	6191.88	5850.43
13	6429.23	6370.80	5850.38	5851.39	5850.42
14	5850.38	6095.52	5855.90	6713.63	5874.96
15	5850.48	6688.61	5850.38	5860.56	6059.72

Fig. 8 Comparison of the convergence

から、この個体の近傍で制約条件を守る個体を見いだせずに、他の局所解の近傍へと最良解が移ってしまい、その後、その近傍から抜け出せずに収束している。しかしながら、この場合においてもcase 6とは異なり、収束の過程において、大全局最適解の近傍を一端は通過している。例えば、case 10のbのような部分を見ることで、失敗した場合でも大全局解の概算値を見いだせる。case 10の失敗例の場合、146世代から184

世代までは実は大全局最適解の近傍であり、制約を若干破っているものである。その結果、適応度関数値が6099.87（目的関数値は5791.29）となっており、結局これが6081.87という局所的最適解に淘汰される。このように収束の過程を通じてより良い解の可能性を見いだせることは、case 6に見られない特徴の一つであり、このペナルティ関数の設定法の利点の一端でもある。さて、ペナルティ指数の変更を急しゅんにした場合、最適解近傍を見いだす確率が下がる。これは、各世代において急激にペナルティを課すことによって、制約を破りはするものの大全局最適解の近傍にいる個体を中心とした一定の数の個体群を形成できないうちに適応度の評価が下がってしまったためと考えられ、その結果、これらの個体群が消滅してしまったためと考えられる。その意味からも段階的にペナルティ指数を変更することで、ある程度個体群の形成に余裕ができ、大全局最適解をより確実に見いだしているものと考えられる。表1のcase 10を見るとほぼ確実に大全局最適解を十分に高い精度で得られており、また、バラツキも非常に小さい。このことから提示した手法の有効性の一端を見て取れる。

4. 8 ノイズの制御

問題1に対して、ペナルティ関数の制御無しの場合で、

Case 11 Coef*1.0, with 1.0% random noise

Case 12 Coef*1.0, with 0.1% random noise, up to 200th generation

Case 13 Coef*1.0, with 1.0% at initial 0.0% at 200th change linearly, from 201 generation no random noise

の3通りに関して、制御有の場合で、

Case 14 Coef*1.0, with control, 0.1% random noise till 200 th

Case 15 Coef*1.0, with control, 1.0% at initial, 0.0 % at 200th change linearly, from 201 generation no noise

の2通りの試行を行った。Case 10は乱数をどれくらいの大きさから始めればよいかを試行するためである。結果を表2, 3, 図9, 10に示す。制御有の場合はいずれも200世代までにペナルティ関数の変更を終えるものとした。表2より、ノイズを徐々に減らすことで大全局最適値へ導く確率が非常に高くなっていることがわかる。また、ペナルティ関数の制御有の場合でも、ノイズを乗せた方が若干ではあるが大全局最適解へ導く確率が上がっている。特にノイズを徐々に減らした場合は、いずれの試行においても解を得ている。これは、図9, 10示すように、局所解に陥る

Table 2 Comparison of pressure vessel problem

Trial	Case 11	Case 12	Case 13
1	6591.86 (h)	6102.91(c)	5801.60(a)
2	6777.38 (g)	6090.53(c')	5855.38(a)
3	7252.63 (h)	6059.71(c)	5850.38(a)
4	5866.00 (a)	5850.39(a)	5850.39(a)
5	6565.89 (h)	6382.86(d)	6821.06(h)

Table 5 Comparison of the results

Trial	Case 10	Case 11	Case 12
1	5862.35	5850.52	5851.31
2	5850.38	5850.45	5850.38
3	5851.95	5850.38	5850.38
4	5850.38	5850.38	5851.73
5	5850.38	5850.38	5881.70
6	5857.21	5861.33	5854.50
7	5850.38	5856.35	5850.38
8	5856.47	5850.41	5851.69
9	5850.38	5850.83	5851.40
10	5852.84	5851.82	5862.29
11	5850.48	5850.39	5850.45
12	5850.43	5850.38	5856.65
13	5850.42	5853.30	5850.43
14	5874.96	6078.58	5851.01
15	6059.72	5850.58	5850.39

場合に多くの場合でペナルティ関数の変更後の環境変化に対して、対応できず、過去の世代における最大値と各世代における最大値との間のかい離が見られるが、ノイズを載せることによってこのような現象が減ることが期待されるためである。また、ノイズを徐々に減らした場合の方がより求解性の安定性が向上するのは、ペナルティ関数の制御とノイズの乗せ方の相乗効果によるものと考えられる。

5. 結 言

本研究では、既に提示した領域適応型遺伝的アルゴリズムにおいて収束性と精度の向上を目的とし、新た

Fig. 9 Process of convergence (case 14)

Fig. 10 Process of convergence (case 15-4)

に3つのパラメータと制約条件付き問題に対するペナルティ関数の自動設定法、適応度のにせるノイズの制御法を提示した。数値計算例に適用した結果、各オペレータを利用することで、収束性、精度面での向上、また、合理的に解析回数を減らすことができることを明らかにした。特に、大局的最適解を得るためのペナルティ関数の自動設定法にノイズを乗せそれを制御する方法を用いることで、大局的最適解にいたる確率が高くなり求解の安定性が向上するなど、提示した手法の有効性の一端を明らかにした。

謝 辞

本研究は東京工業大学J R 東日本寄附講座の研究の一環である。ここで関係各位に謝意を表したい。また、有益な御助言を頂いた早稲田大学山川宏教授、北海学園大学杉本博之教授に謝意を表したい。

参考文献

- (1) 稲川, 荒川, 山川, 機論, 61-587C, (1995), 2901-2908.
- (2) M. Arakawa, N. Ohkubo and H. Yamakawa, ASME Proc. of Design Automation Conference'95, DE-Vol. 83, 2, (1995), 733-738.
- (3) 伊能, 上杉, 機論, 61-585A, (1995), 1109-1114.
- (4) 伊能, 下平, 小林, 機論, 61-586A, (1995), 1416-1422.
- (5) 菊池, 応用数理, 3-1, (1993), 2-26.
- (6) 古田・杉本, 遺伝的アルゴリズムの構造工学への応用, 森北出版, (1997) .
- (7) Papalambros, P. Y., Journal of Mechanical Design, **117**, Special 50th Anniversary Design Issue, (1995), pp. 55-62.
- (8) 古川, 矢川, 機論, 61-586A, (1995), 1409-1415.
- (9) 荒川, 萩原, 機論, 63-616C, (1997), 4216-4223.
- (10) 荒川, 萩原, 機論, 64-621C, (1998), 1626-1635.
- (11) 荒川, 萩原, 日本計算工学会講演論文集, Vol. 3, No. 2, (1998), 577-580.
- (12) Sandgren, E., Journal of Mechanical Design, **112**-2, (1990), pp. 223-229.
- (13) Lewis, K. and Mistree, F., Proc. of ASME Design Engineering Technical Conference and Computers in Engineering Conference, Irvine, 1996, in CD-ROM, 96-DETC/DAC-1601.
- (14) Azarm, S., Papalambros, P., Tran. of ASME, Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 106, No.1, (1984), pp.82-89.
- (15) NASA homepage : <http://fmad-www.larc.nasa.gov/mdob/mdo.test/class2prob4/dscr.html>
- (16) Hajela, P., Lee, J., Structural Optimization, **12**, (1996), 11-15.
- (17) 萩・Haftka, 機論, 64-618A, (1998), 266-273.
- (18) 荒川, 萩原, 機論, No. 98-8, Vol. B, 489-492.