

トラス構造物重量最適化のための 効率的な遺伝的アルゴリズムの開発*

三村泰成^{*1}, 廣安知之^{*2}, 三木光範^{*2}

Development of efficient genetic algorithms for truss structure optimization*

Yasunari MIMURA^{*1}, Tomoyuki HIROYASU^{*2} and Mitsunori MIKI^{*2}

^{*1} Institute of Environmental Studies, Graduate School of Frontier Sciences, The University of Tokyo
^{*2} Knowledge Engineering and Computer Sciences, Doshisha University

This paper presents a new efficient genetic algorithm for truss structure optimization. In many optimum design problems, values of design variables are real numbers. We have proposed the "Center Neighborhood Crossover(CNX)" for real coded genetic algorithms. CNX is a powerful tool for function optimizations. Since it is uncertain that these operators are able to work for practical design problems efficiently, it is necessary to examine some crossover operators for practical use. In the present study, we verified and compared the performance of several crossover operators using benchmark functions. Furthermore, we tried to optimize a truss structure problem using these operators. By considering the results mentioned above, we developed the "two-stage method" for truss structure optimization. This method bring us the efficient search and the high-accurate optimum solution.

Key Words : Optimum Design, Structural Analysys, Probabilistic Method, Genetic Algorithms, Truss Structure, Crossover, Characteristic Preservation

1. 緒 言

構造物の最適化設計を行う場合、設計変数を実数と定義し、目的関数を実数ベクトル空間中の連続関数として取り扱うことが多い。連続関数の最小値(あるいは最大値)を探索する手法としては、傾斜法が効率的な手法の1つであり、高い精度で解を求めることができる。しかしながら、実際の設計問題では、極端な非線形性や非連続性を含むことが多く、傾斜法では良好な解探索が行えないことも少なくない。本研究で扱う「トラス構造物の重量最適化問題」も、座屈制約などが含まれることから、傾斜法の適用が困難な問題の一つである。それゆえ、この問題に対して、効率的で高精度な最適化手法を開発できれば、一般的な構造問題の最適設計の可能性を広げることができる。傾斜法以外の最適化手法で有力なものは、確率論的な手法であり、本研究では、その中でも遺伝的アルゴリズム⁽¹⁾(以下GAと称す)に注目する。

GAは生物の進化を模倣した確率的最適化手法であり、連続変数、離散変数を問わず評価関数さえ定義できれば、比較的簡単に利用できる手法である。これまでに、GAによる構造最適化に関する研究は数多く見られるが^{(2)~(4)}、そこで使用されているGAのほとんどはSimple GAと呼ばれる基本的なGAのアルゴリズムである。そのために、計算量が従来の傾斜法と比較して莫大に増加したり、精度の良い解が求まらないといった問題が存在する。本研究では、トラス構造物の重量最適化問題を対象として、精度の高い最適解を得ることのできるGAの開発を行う。対象問題は座屈制約を含んでいるため、極めて非線形性が強く、本問題を解決できるアルゴリズムであれば、その他の構造最適化問題にも拡張可能であると考えられる。精度の高い解を効率良く求めるためには、次の2点が必須となる。すなわち、良好な形質遺伝を行うことが可能な交叉法の開発と、高精度な解を求めることが可能な戦略の開発である。

良好な解をGAにて求めるためには、親個体が子個体を生成する際に、適切な形質遺伝を行う必要がある。適切な形質遺伝が行われない状態で探索を行った場合、ランダムサーチと同等の性能しか得られない

* 1999年5月27日第4回計算工学講演会にて講演、原稿受付2000年月日

^{*1} 正員、東京大学大学院新領域創成科学研究科環境学研究室(〒113-8656東京都文京区本郷7-3-1工学部12号館吉村忍教授気付)

^{*2} 正員、同志社大学工学部知識工学科
Email: mimura@cc.oita-u.ac.jp

ことがほとんどである。したがって、適切な形質遺伝が行われるようなコーディング / 交叉の設計を行う必要がある。適切な形質遺伝が行われるためには、積木仮説が成り立つ必要があり、積木仮説が成り立つためには「表現型が近い個体は遺伝子型も類似している」ことが必要である⁽⁵⁾。しかしながら、関数最適化のような連続変数を扱う場合には、必ずしもこのような設計が行われているとは言えない。一般的には、連続変数に対するコーディングとしてバイナリあるいはグレイコードが用いられ、交叉法として1点交叉 (One Point Crossover, 以下 1X と称す)、2点交叉 (Two Point Crossover, 以下 2X と称す) あるいは一様交叉 (Uniform Crossover, 以下 UX と称す) が用いられることが多い。これらのコーディング / 交叉では、表現型 (連続変数) は近くとも遺伝子型 (バイナリ or グレイコード) が必ずしも近くないことから適切な形質遺伝が行われず、要求する精度の最適解が求められないことも少なくない。以上の問題を解決するために、これまでに BLX- α ⁽⁶⁾、UNDX⁽⁷⁾、SPX⁽⁸⁾ などの交叉法が提案されてきた。それに対して、本研究ではそれらの手法の特徴を維持しつつ、コーディングが容易な重心近傍交叉 (Center Neighborhood Crossover, 以下 CNX と称す) を提案し、その有効性を検討する。

それぞれの交叉法には長所と短所が存在するために、精度の良い解を効率良く求めるには複数の交叉法を組み合わせて使用する戦略が必要となる。そこで本研究では、まず、CNX を用いた実数コード化 GA を含む数種の GA に対して、標準関数による数値実験を行う。そして、その結果を検証し、トラス構造物の重量最適化問題に対して効率的で高精度な最適解の探索を実現する「2段階手法」を開発した。また、本手法の開発の流れを明らかにすることで、「実数を設計変数とする最適化問題」に対して GA を適用する場合の指標も明らかとした。

2. 実数コード化遺伝的アルゴリズム

GA を用いた解探索では、探索の初期段階では大域探索が行われ、探索終盤では近傍探索が行われるのが理想である。このような探索を実現するためには適切な形質遺伝が行われるようなコーディング / 交叉を設計する必要がある。具体的には、

- (a) コーディング 表現型が近い個体は遺伝子型も類似させる。
- (b) 交叉 形質の離れた親からは、離れた子が生成され、形質が近い親からは、類似した子が生成される。

という2点を満たさねばならない。上記のことを考慮していない不適切なコーディング / 交叉の設計を施した GA を用いた探索は、ランダムサーチと同程度になってしまう。実際の応用ではこのようなケースも少なくなく、効率的な探索を実現することは難しい。

一般的な GA ではバイナリコーディングが用いられ、交叉法としては、1X, 2X あるいは UX が用いられる。これらの GA は、組合せ問題などに関しては、適切なスキーマ (意味のあるビットパターン) 保存が行われることが知られており⁽⁹⁾、ある程度は上記の条件を満たすと考えられる。しかしながら、連続変数では、ビット列 (遺伝子型) と実数 (表現型) の間には、なんら相互関係が存在しない。したがって、(a) の条件を満たしているとは言いがたく、1X, 2X, UX では (b) を満たすことも難しい。この欠点を補うためにグレイコーディングが考案されたが、グレイコーディングも「遺伝子型が隣接する個体は表現型も隣接している」ことが保証されているだけである。隣接していない個体については、バイナリコーディングと同様であり、なんら問題を解決していない。実装が容易なことから、連続変数に対してもバイナリコーディングが用いられることが多いが、効率的な探索が行われていないケースも少なくない。

実数を設計変数として用いる場合「実数ベクトル空間上の位置」に近い個体は、評価関数の値も近い事がほとんどである。したがって、継承すべき形質は「実数ベクトル空間上の位置」であると考えられ、良好な形質遺伝が行われるためには、

- コーディング 実数はできるだけ実数として扱う。
 - 交叉 確率的な揺らぎを持たせた上で、実数ベクトル空間上の位置に近い親からは、近い子を生成させ、離れた親からは離れた子を生成させる。
- という条件を満たすようにコーディング / 交叉を設計する必要がある。これらの条件を満たすために開発された、代表的な交叉法には以下のようなものがある。
- BLX- α (blend crossover)⁽⁶⁾ 2個の親を囲む各辺が座標軸に平行な超立方体の領域において一様分布にしたがってランダムに2つの子を生成する。
 - UNDX (Unimodal Normal Distribution Crossover)⁽⁷⁾ 2個の親を結んだ直線の中心近傍に正規分布にしたがって、ランダムに2つの子を生成する。正規分布の標準偏差は、両親を結ぶ主軸方向の成分は両親間の距離に比例させ、それ以外の軸の成分では、主軸と第3の親の距離に比例させる。

- SPX (Simplex Crossover)⁽⁸⁾ 設計変数が n 個の時, m 個 ($2 \leq m \leq n+1$) の親が構成する n 次元空間中の多面体の重心を \vec{O} とし, 親の位置ベクトルを $\vec{r}_i, (i = 1, \dots, m)$ とすると,

$$\vec{Y}_i = (1 - \epsilon)(\vec{r}_i - \vec{O})$$

が構成する多面体内に一様分布にしたがってランダムに子を生成するのが, SPX である. 設計変数 n , 親の数 m , パラメータ ϵ のとき, SPX- $n-m-\epsilon$ と表記される.

また, 交叉法ではないが, 実数コーディングに関する研究として, 以下のような手法が挙げられる.

- 実数領域適応型遺伝的アルゴリズム⁽¹⁰⁾ 個体群の平均と分散を用いて, 随時, 変数の定義域を調整する.

3. 重心近傍交叉 (CNX)

3.1 提案アルゴリズム 前章で示した各交叉法は, 精度面で優れていたり, 処理量が少なかったりと, それぞれ長所短所を持っている. 本研究では各交叉法の特徴を合わせもつような重心近傍交叉 (CNX) を提案する.

Fig. 1 に設計変数が 2 個 ($n = 2$) の場合の CNX の概念図を示す. CNX では, 設計変数が n 個の場合, n

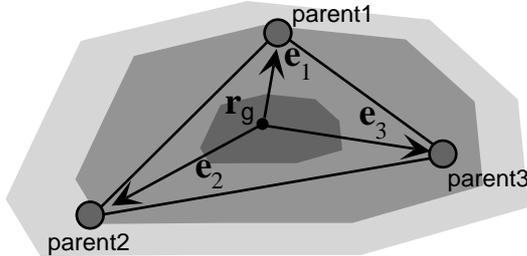


Fig. 1 CNX ($n = 2$)

次元空間中に $n+1$ 個の親が形成する多角形の重心近傍に $n+1$ 個の子を生成する. 重心の位置ベクトル \vec{r}_g は,

$$\vec{r}_g = \frac{1}{n+1} \sum_{i=1}^{n+1} \vec{r}_i \quad (1)$$

\vec{r}_i : 親の位置ベクトル

であり, 重心から親へ方向ベクトル \vec{e}_i は,

$$\vec{e}_i = \frac{\vec{r}_i - \vec{r}_g}{|\vec{r}_i - \vec{r}_g|}, \quad (i = 1, \dots, n) \quad (2)$$

となる. 式 (1), 式 (2) を用いて, 子の位置ベクトル \vec{r}_{child} を,

$$\vec{r}_{child} = \vec{r}_g + \sum_{i=1}^{n+1} t_i \vec{e}_i \quad (3)$$

として定義する. ここで t_i は, 標準偏差 σ_i , 平均 0 の正規分布に従う乱数である. また, σ_i は,

$$\sigma_i = \alpha |\vec{r}_i - \vec{r}_g|, \quad (i = 1, \dots, n) \quad (4)$$

と仮定する. ここで α はパラメータである. α の値を調整することで確率的な揺らぎの大きさを調整でき, 親の外側にもある程度の確率で子が生成される. CNX は, 設計変数間に依存がある場合にも有効であり, 設計変数の数が増加しても容易に実装することができる. $n = 2$ のとき, 子は Fig. 2 のような分布に従う確率で生成される. CNX を使えば, 探索の初期段階では,

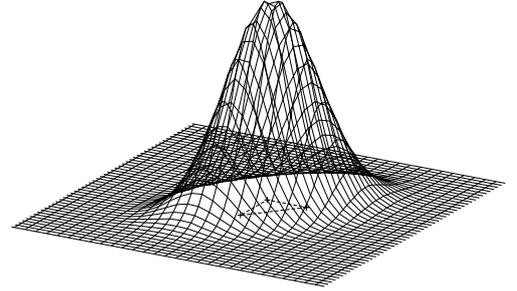


Fig. 2 Probability Distribution of CNX ($n = 2$)

親が離れた位置にあるために, ある程度ランダムな探索が行われ, 探索終盤では, 親の近傍を探索することになる. 親の選び方は SPX の $m = n+1$ の時と同等であり, 子の生成には UNDX と同様に正規分布を用いていることから, 両手法の中間的な手法と言える.

次節では, CNX がどのような場合に有効に機能するかを標準関数を使って検証する.

3.2 標準関数を用いたアルゴリズムの評価

3.2.1 検証方法 ここでは, 標準関数を用いて以下の 2 種類の数値実験を行い, アルゴリズムを検証する.

- 数値実験 1 交叉法の探索性能を検証するために, 突然変異率を 0% として, 1X, UX, CNX の 3 種類の交叉法を用いた解探索を行う. 2X は多点交叉の一種であり, UX に含まれるものと考えられることから, ここでは対象外とする.
- 数値実験 2 数値実験 1 では突然変異を用いなかったが, 1X や UX を用いた従来の GA では, 若干の突然変異を設けた方が, 実用上, パフォーマンスが上がるのがしばしばある. そこでここでは, 1X, UX, CNX の 3 種類の交叉法に対して, 突然変異がどのような効果を及ぼすかを検証する. ここでの突然変異率は, およそ 1 個体に一回突然変異が発生する程度とする.

上記の数値実験では, それぞれの手法で 10 回づつ探索を行い, 同時点での最良個体の平均値を評価関数値

とする．そして，この値の推移を観察することで性能を評価した．

3.2.2 Rosenbrock 関数 Rosenbrock 関数とは，

$$F(x) = \sum_{i=1}^m [(1 - x_{2i-1})^2 + 100(x_{2i} - x_{2i-1}^2)^2], \quad (5)$$

$$(m = n/2, \quad n = 4).$$

で表され，変数間に依存関係が存在する関数である．別名バナナ関数と呼ばれており，文字通りバナナの先のように若干複雑な形状をしているが，単峰性である．

ここでは遺伝子長を 80bit としたことから，探索空間は約 10^{24} である．GA では最大値探索を行うので，実際にはこの関数の符号を反転させて，最適値「0」を探索する．また， α は予備実験から経験的に 0.5 とした．

数値実験 1 の結果を Fig. 3 に示す．1X, UX と比べて CNX が圧倒的に探索性能が高く，10 試行の探索全てが最適値に到達した．また，1X, UX については，全ての試行で最適解に到達することはなかった．

数値実験 2 の結果を Fig. 4 に示す．比較のために，数値実験 1 で最も良い結果を得た CNX の結果も示す．この実験では，突然変異率を 1% に設定している．これは 1 個体に約 1bit 反転する確率である．数値実験 2 では，1X が最も良好な探索性能を示しているが，数値実験 1 の CNX の方が圧倒的に優れている．また，CNX については突然変異を設けると探索性能が低下している．これは，突然変異によって実数ベクトル空間上の優れた情報が破壊されたことが原因であると考えられ，Rosenbrock 関数では，実数ベクトル空間の情報が非常に有用であると言える．

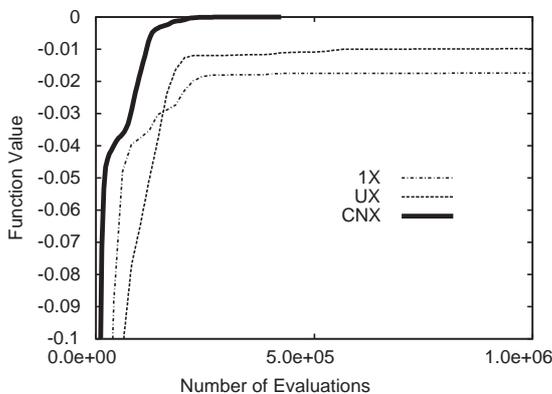


Fig. 3 Performance on rosenbrock function (Comparison among crossover operators)

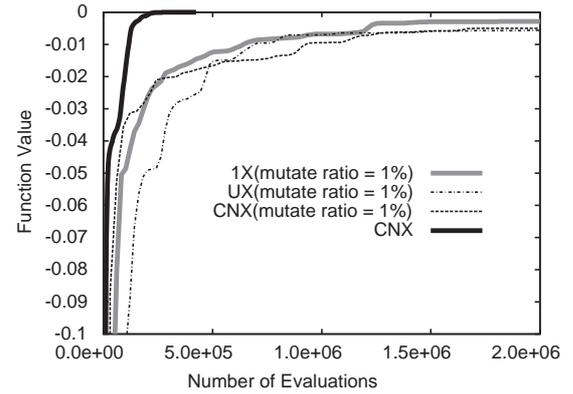


Fig. 4 Performance on rosenbrock function (Effect of mutation)

3.2.3 De Jong の第 5 関数 ⁽¹¹⁾ De Jong の第 5 関数は

$$F(x) = - \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right]^{-1} \quad (6)$$

のような関数で，変数間に依存性のない多峰性関数であり，最適値は「-1」である．ここでは遺伝子長を 34bit としたので，探索空間は約 2×10^{10} である．また， α は予備実験から経験的に 0.3 とした．

数値実験 1 の結果を Fig. 5 に示す．UX が最も高い探索性能を示しているが，どの交叉法を用いても，必ずしも最適値に到達しなかった．特に CNX については，1 度も最適値に到達できなかった．

数値実験 2 の結果を Fig. 6 に示す．比較のために数値実験 1 の UX と CNX の結果も示す．この実験では，突然変異率を 3% と設定した．これは 1 個体に約 1bit 反転する確率である．突然変異を設けた場合，1X が最も探索性能が高く，10 試行の探索全てが最適解に到達した．これは数値実験 1 で最も探索性能が良かった UX を上回っている．また，CNX に対しても突然変異を設けた方が探索性能が上がっており，この問題では，実数ベクトル空間上の情報は，ほとんど探索性能に寄与しないと考えられる．これは，CNX を用いるよりもランダムサーチの方が探索性能が優れていることを意味し，この問題に対しては，CNX は有効に機能しないことが明らかとなった．

De Jong の第 5 関数では，最適値が存在する「山」を探索することが重要であり，関数の最適値を探索すると言うよりは，むしろ最も良い評価を与える「組合せ」を探索していると言える．組合せ問題の場合，1X や UX の方が優れたスキーマ保存が行われるので，このような結果が得られたと考えられる．

3.2.4 検証結果の考察 標準関数の解探索の結果より，以下のことが言える．

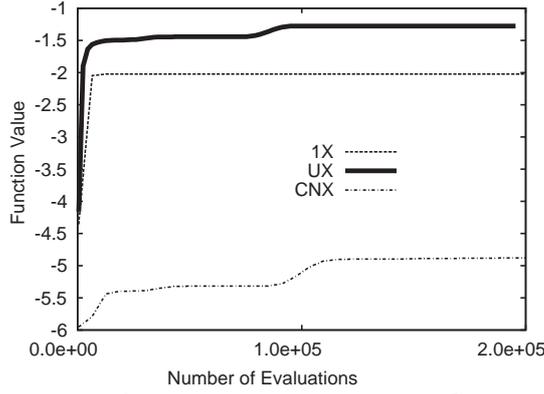


Fig. 5 Performance on De Jong F5 (Comparison among crossover operators)

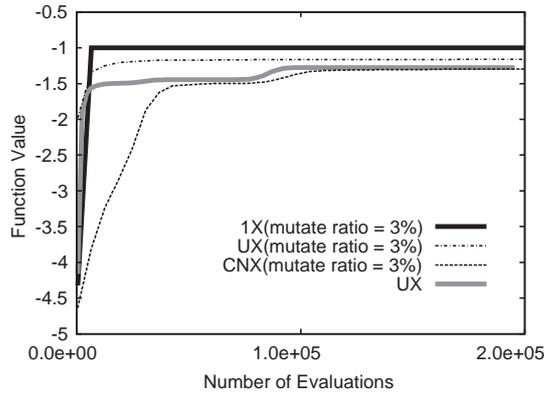


Fig. 6 Performance on De Jong F5 (Effect of mutation)

- 単峰性関数では、突然変異を設けない CNX のみの GA が最も高い探索性能を示す。
- 極端な多峰性関数の場合、1X に若干の突然変異を設けた GA の探索性能が高い。

実際の設計問題では、関数形状が既知でないことがほとんどであり、上記をふまえた試験を行い、適切なアルゴリズムを選択する必要がある。以降、トラス構造物の重量最適化への適用について述べる。

4. トラス構造物の重量最適化

本章では、トラス構造物の重量最適化問題における各交叉法の性能評価を行い、精度の高い解を効率よく探索することが可能な戦略を提案する。

4.1 解析モデル 本研究では、Fig. 7 に示すような 10 部材のトラス構造物についての重量最適化を行った。

設計変数は部材の断面積とし、連続変数として扱う。1 部材の断面積あたり 12bit を使用し、定義域は 1 ~ 4096mm² とした。遺伝子長は、120 ビットで表され、探索空間は、約 10³⁶ である。最小化すべき評価関数を式 (7) に示す。

$$H = (w_v \times V) + P_L + P_G \quad (7)$$

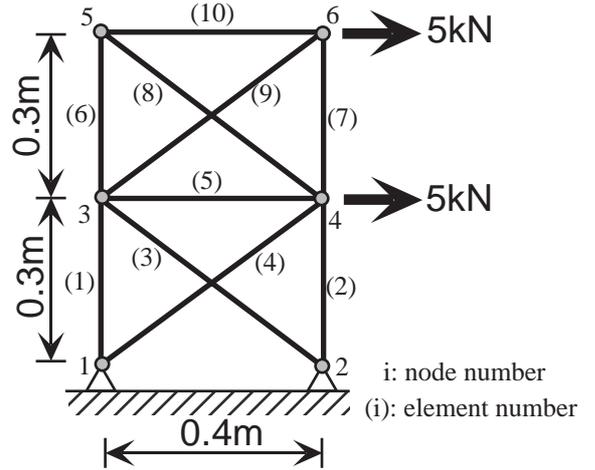


Fig. 7 10-member truss structure

$$P_G = w_d \times d^2 \quad \text{if } (d > d^*) \quad (8)$$

$$P_L = \sum_{k=1}^N P_k \quad (9)$$

$$P_k = \begin{cases} 1 & \text{if } ((\sigma_k > \sigma_y) \text{ or } (L_k > L_k^*)) \\ \text{else } 0 \end{cases} \quad (10)$$

ここで、 V はトラス構造物の総体積、 w_v 、 w_d は重み係数、 d^* は節点 6 の変位の上限值、 P_G 、 P_L はそれぞれ全体制約と局所制約に関するペナルティ関数である。また、 σ_y は引張強度であり、 L_k^* は部材 k の座屈荷重である。本問題では、座屈を制約条件に含めたことから、対象とする最適化問題の非線形性が極めて強くなっている⁽¹²⁾。具体的な値は、

- ヤング率 $E = 100 \text{ MPa}$
- 引張強度 $\sigma_y = 40 \text{ MPa}$
- $d^* = 0.003m$
- $w_v = 10^3$, $w_d = 10^6$

と設定した。節点変位、部材の応力は有限要素法で計算し、 L_k^* は、部材を円断面と仮定してオイラーの座屈荷重より計算する。全体制約条件はトラス構造物全体に関わる制約条件で、節点 6 の変位が d^* より小さいならペナルティは 0、 d^* 以上であれば変位の 2 乗の w_d 倍がペナルティとして課せられる。局所制約は、圧縮または伸長に関する条件を満たしていない部材の数がペナルティとなる。GA では最大値を探索することから、実際に用いる評価関数 F は式 (7) の逆数で表され、次式の形となる。

$$F = \frac{1}{H} = \frac{1}{(w_v \times V) + P_L + P_G} \quad (11)$$

4.2 交叉法の性能評価 個体数を 500，評価関数の繰り返し数の限界を 1.5×10^6 回として，関数最適化の数値実験 1 と同様な探索を行った．また，CNX で用いるパラメータ α は，予備実験より経験的に 0.3 とした．評価関数の推移を Fig. 8 に示す．関数最適化と同様に，それぞれの手法で 10 回の探索を行い，同時点での最良個体の平均値を評価関数値とした．

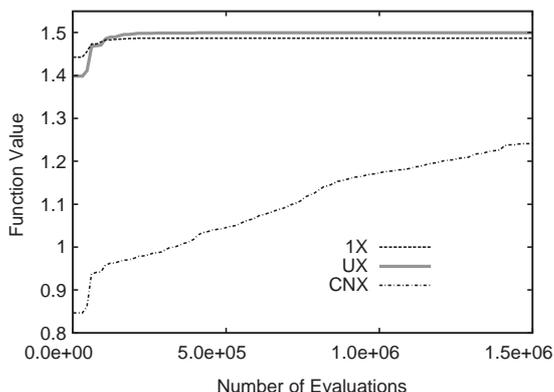


Fig. 8 Performance on truss structure optimization (Comparison among crossover operators)

UX が最も高い探索性能を示しているが，1X についても，UX とかなり近い性能を示している．これに対し CNX は，かなり低い探索性能であった．

この問題は，応力制約，座屈制約，変位制約を含んでおり，単純な単峰性ではないと考えられる．それゆえ，組合せ問題的な性質が大きく，CNX では十分な大域探索が行われず，探索性能が低くなったと考えられる．しかしながら，De Jong の第 5 関数と異なり，この問題での CNX の探索は，評価関数が増加方向にあり，実数ベクトル空間中の関数としての性質も持っていると考えてもよいであろう．

交叉法のみを用いた探索では，10 試行で得られた最良の評価関数値にばらつきがあり，応力も制約に近づいておらず，安定して最適解が得られているとはいえないものであった．

4.3 従来の GA との比較 (突然変異の効果)

ここでは関数最適化の数値実験 2 と同様な実験を行い，トラス構造物の重量最適化における突然変異の効果を明らかにする．GA のパラメータは，突然変異率 1%，個体数は 500 とした．評価関数の推移を Fig. 9 に示す．比較のために，突然変異を設けない上述の数値実験の UX，CNX の結果も示す．

最も高いパフォーマンスを示したのは突然変異率 1% を設けた 1X であり，これは交叉法の評価で最も良い結果を得た UX よりも良い結果が得られた．トラス

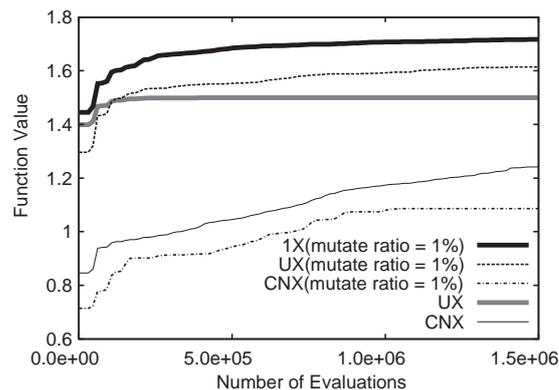


Fig. 9 Performance on truss structure optimization (Effect of mutation)

構造物の重量最適化については，実数問題として扱うよりも単純に組合せ問題と考えて，一般的な GA を用いた方が大域探索において高いパフォーマンスが得られるようである．しかしながら，探索終盤では個体群の多様性が失われ，類似した個体が交叉するだけになり，1X のみで優良なスキーマを発見するのは難しいはずである．それゆえ，近傍探索は突然変異のみに依存した単なるランダムサーチしか行っていないと考えられ，改善の余地がある．

一方，CNX については，

- 突然変異を設けない CNX では，評価関数値が増加方向にある．
- CNX に突然変異を設けると探索性能が低下している．それゆえ，CNX に突然変異を設定しても，優良な情報を破壊するだけで，探索性能の向上には寄与しないとみなせる．

という結果が得られた．これはトラス構造物の重量最適化問題が，De Jong の第 5 関数のような単なる「組合せ問題」ではなく，実数ベクトル空間中の関数としての性質も有していることを意味し，近傍探索においては，CNX が有効に機能すると予測される．

上述の結果は，大域探索と近傍探索では，それぞれ異なる GA が有効に機能することを意味しており，以降では，これらを考慮した改善策について述べる．

4.4 2 段階手法 上述の数値実験の結果をふまえて，本研究では，トラス構造物の重量最適化問題にもっとも適していると考えられる，以下の「2 段階手法」を提案する．

1. 第 1 段階 突然変異を設けた 1X による GA で探索を行う．
2. 第 2 段階 十分に大域探索が行えたと判断できれば，第 1 段階で求められた解を初期個体中にエリート個体として保存し，CNX のみによる GA で近傍探索を行う．

これにより、大域探索、近傍探索ともに良好な解探索が実行できると期待できる。

アルゴリズムの検証のために、第1段階では、個体数を500、突然変異率を1%とし、第2段階では、個体数を100、突然変異率を0%、CNXで用いるパラメータを $\alpha = 0.3$ として、Fig. 7のトラス構造物の重量最適化を行った。大域探索が十分に行えたかどうかの判断は、経験的に「世代交替が100世代(評価回数にして約50000回弱)行われても評価関数値に変動がない」というものを用いた。

評価関数の推移をFig. 10に示す。参考のために1X(突然変異率1%)のみによる探索結果も示す。評価関数の実行回数が、 2×10^5 回近辺で、CNXによる探索に移行しており、探索性能が明らかに向上していることが分かる。

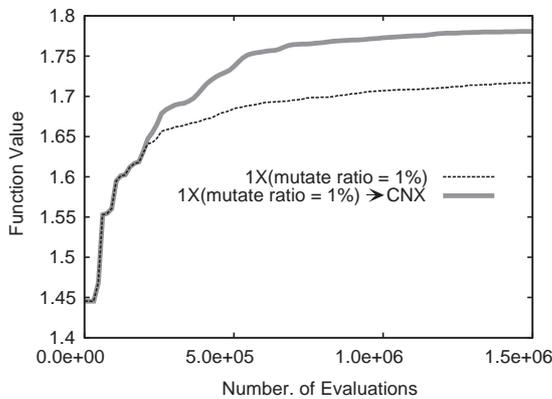


Fig. 10 Performance on truss structure optimization (Using two-stage method)

第1段階で得られた典型的なトラス構造の断面積分布と第2段階で得られた最適解の断面積分布をFig. 11に示す。第1段階での探索では、部材6, 8, 10が残っているが、第2段階の探索によって余分な部材がほぼ取り除かれ、全体的にスリムになっているのが分かる。

「UXのみのGA」、「1Xに突然変異を設けたGA(表中では1X(*mutate*)と表記する)」、「2段階手法」で求められたトラス構造物の総体積の比較をTable 1に示す。各値は10試行で求められた最大値、最小値、平均値、標準偏差である。2段階手法の標準偏差のオーダーが他の手法よりも1桁小さく、10試行の結果が、ほぼ同じ値に到達していることがわかる。Table 2には、2段階手法で得られた最も小さい総体積 $9.329 \times 10^{-4} m^3$ の時の、各部材の断面積、応力、座屈応力および節点6の変位を示す。ここで、応力は引張を正として表示している。全ての値が制約条件を満たしており、節点

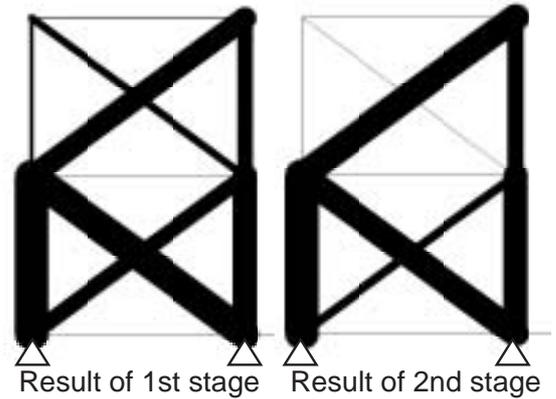


Fig. 11 Distribution of cross-section area

6の変位が $0.003m$ 、部材5の応力が40MPaにほぼ等しく、制約条件に到達しているのがわかる。2段階手法で求められた10試行の全ての解は、Table 2の結果とほぼ同等のものであったことから、総体積 $9.329 \times 10^{-4} m^3$ が、本数値実験の最適値と考えてよいであろう。

Table 1 Total volume of truss structure

	UX	1X(<i>mutate</i>)	1X(<i>mutate</i>) → CNX
max.	1.187×10^{-3}	9.995×10^{-4}	9.415×10^{-4}
min.	1.042×10^{-3}	9.517×10^{-4}	9.329×10^{-4}
ave.	1.113×10^{-3}	9.711×10^{-4}	9.359×10^{-4}
std. dev.	4.141×10^{-5}	1.487×10^{-5}	2.308×10^{-6}

[m^3]

Table 2 Optimum solution

element No.	area [mm^2]	stress [MPa]	buckling stress [MPa]
1	577.0	13.06	-50.35
2	413.0	-18.08	-36.04
3	474.0	-13.30	-14.89
4	167.0	37.09	-5.246
5	1.000	39.98	-0.04909
6	7.000	-0.5414	-0.6109
7	275.0	-13.65	-24.00
8	1.000	6.316	-0.03142
9	451.0	13.87	-14.17
10	11.00	-0.4594	-0.5400
displacement of node 6			3.000 mm

上述のように、本手法を用いれば、探索性能が向上するだけでなく、高い精度で最適解を探索することができる。確率的手法であるGAで解探索を行ったにもかかわらず、安定して最適解が得れるということは、特筆すべきことである。

4.5 2段階手法におけるパラメータの設定 本研究で提案した2段階手法では、

1. 第1段階：
 - (a) 突然変移率
 - (b) 初期個体数
 - (c) 十分な大域探索が行えたかどうかの判断基準
2. 第2段階：
 - (a) 初期個体数
 - (b) CNXのパラメータ α

というGAパラメータが必要とされる。本研究では、Fig. 7のような10部材のトラス構造物についてのGAパラメータを経験的に求めている。任意のトラス構造物の最適化を容易に実現するためには、

- 設計変数の数 (部材数)
- 探索空間の規模 (部材数 \times 1変数のbit数)

などから、探索前にパラメータを同定する手段を開発すべきである。パラメータ同定の手段を見出せれば、本手法は常に安定した最適解を提供してくれるだろう。

5. 結 言

本研究では、まず、標準関数について著者らが提案しているCNXを用いた実数コード化GAを含む数種のGAの性能評価を行った。さらに、トラス構造物の重量最適化に対しても、同様な性能評価を行い、それらの結果をふまえて、

- 1Xを用いた従来のGAにより大域探索を行う。
- CNXを用いた実数コード化GAにより近傍探索を行う。

という「2段階手法」を提案した。数値実験を行った結果、この手法は、トラス構造物の重量最適化に対して高い探索性能を示すとともに、高精度な最適解が得られることが明らかとなった。GAはモンテカルロシミュレーションであり、膨大な計算時間を費したとしても、良好な解を探索できないことも少なくない。本研究の手法を用いれば、安定した最適解を得ることができ、実用的な設計に対しても安全に適用することが可能である。

設計変数が実数の場合、可能であるならば、当然ながら傾斜法を用いた方が近傍探索性能は優れていると思われる。しかしながら、実際の設計の現場では、目

的関数は複雑で不明確なことがほとんどであり、傾斜法で有効な探索が実現できないことも少なくない。特に非連続性を有するときは、傾斜法は適用できない。本研究では対象問題をトラス構造重量最適化に限定した。この問題は座屈制約を含んでおり、非線形性が極めて強いため、傾斜法で良好な探索を行なうには難しい問題の一つである。この問題で良好な解を求めることができることから、一般的な構造問題にも2段階手法は有効であると考えられる。問題の性質が不明確なときでも、設計変数が実数であれば、容易に適用できることが本手法の長所であると言える。

文 献

- (1) Holland, J. H.: *Adaptation in Natural and Artificial System*, University of Michigan Press, (1975), MIT Press, (1992).
- (2) 朝山・他2名, 機論, 62-597, A, (1996), 1234.
- (3) 稲川・他2名, 機論, 61-587, C, (1995), 2901.
- (4) 尾田・他2名, 機論, 61-582, A, (1995), 460.
- (5) 坂和正敏, 田中雅博, 遺伝アルゴリズム, 朝倉書店, (1995).
- (6) Eshleman, L. and Schaffer, J. D., *Real-Coded Genetic Algorithms and Interval-Schemata*, Foundations of Genetic Algorithms 2, (1993), 187-202.
- (7) Ono, I. and Kobayashi, S., *A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover*, Proceedings of 7th International Conference on Genetic Algorithms, (1997), 246-253.
- (8) Tsutsui, S and Ghosh, A., *A Study on the Effect of Multi-parent Recombination in Real Coded Genetic Algorithms*, Proceedings of the 1998 IEEE ICEC, (1998), 828-833.
- (9) 山本公洋, 内藤昭三, 交叉のスキーム保存機構に関する考察, 電子情報通信学会論文誌, D-II, Vol. J81-D-II, No.12, (1998), 2790-2801.
- (10) 荒川, 萩原, 実数領域適応型 (ARRange) 遺伝的アルゴリズムの開発, 機論, 63-616, C, (1997), 138-145.
- (11) Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programming*, Springer, (1992).
- (12) Miki, M., *Object-Oriented Approach to Modeling and Analysis of Truss Structures* AIAA Journal, Vol. 33, No. 2, (1994), 348.