A-022

多目的遺伝的アルゴリズムにおける制約条件の取り扱いの検討
Examination of Handling Constraint on Multiobjective Genetic Algorithm

廣安 知之 * 三木 光範 † 鈴木 和徳 ‡ 金 美和 §
Tomoyuki Hiroyasu Mitsunori Miki Kazunori Suzuki Kim Mifa

1. はじめに

多目的最適化問題とは，トレードオフの関係にある複数の評価基準のもとで最適解を求める問題である．この問題では評価基準が互いに競合することから，最適解は複数もしくは無数に存在する．そのため多目的最適化問題に対する手法には，多点探索である遺伝的アルゴリズム（GA）を適用した多目的GAの研究が数多く行われている．一方GAにおける交叉，突然変異といった遺伝的オペレータには，一般的に制約条件を考慮するメカニズムが組み込まれていない．そのため制約条件が存在する多目的最適化問題では，実行可能領域外に個体を生成してしまう可能性がある．また一般的にGAは初期個体をランダムに発生するため対象問題の実行可能領域が定義域に対して非常に小さい場合は，実行可能領域内に初期個体を生成させることができない．そのため，制約条件が存在する多目的最適化問題では，制約条件を取り扱う操作が重要となる．そこで本論文では，多目的GAにおける制約条件の取り扱い手法について検討を行う．

2. 制約条件を取り扱う手法

2.1 比較手法

本論文では制約条件を取り扱う手法として一般的に用いられる3つの手法について比較を行う．

デスペナルティ法

デスペナルティ法は，実行可能領域外の全ての個体に対して最も悪い評価値を与え，次世代への生存率を下げる手法である．

引き戻し法

引き戻し法とは実行可能領域外に子個体が生成された場合，式(1)によって実行可能領域内へ引き戻す手法である．k設計変数問題のt世代における親個体の設計変数を$x_i^t \ \ (i=1,...,k)$，子個体をx_i^{t+1}とした場合，新しく与えられる子個体の設計変数$x_{i_new}^{t+1}$は次のように求められる．

$$x_{i\_new}^{t+1} = \frac{x_i^t + x_i^{t+1}}{2} \tag{1}$$

ペナルティ法

ペナルティ法とは，ペナルティ関数によって個体の評価値を与える手法である．本論文ではDebによって提案されたペナルティ関数[1]を多目的最適化問題に拡張したものを用いる．m個の制約条件式$g_i \leq 0 \ \ (i=1,2,...,m)$を持つ問題において，式(2)により個体の評価値$E$を割り当てる．

$$E = \begin{cases} E & if \ \ g_i' = 0 \\ E_{worst} + \sum_{j=1}^{m} g_j' & otherwise \end{cases} \tag{2}$$

E_{worst}は実行可能領域内の個体における最も悪い評価値である．またg_i'は式(3)により求められる．

$$g_i' = \begin{cases} 0 & if \ \ g_i \leq 0 \\ \frac{g_i}{g_{i\_max}} & otherwise \end{cases} \tag{3}$$

g_{i_max}は個体群におけるg_iの最大値とする．

2.2 制約条件を考慮したバイナリトーナメント選択

NSGA-II [2]では制約条件にもとづいたバイナリトーナメント選択を用いている．バイナリトーナメント選択とは，ランダムに2個体を選び出し，優れた個体を探索母集団に加えるという手法である．NSGA-IIではバイナリトーナメント選択において以下の規則に従い優劣関係を定めている．

1. 実行可能領域内の個体iと実行可能領域外の個体jでは個体iを選択する．
2. 個体i，jが共に実行可能領域内に存在する場合，個体i，jのうち評価値の良い個体を選択する．
3. 個体i，jが共に実行可能領域外の場合，実行可能領域に近接している個体を選択する．

本論文の数値実験では，全ての手法においてこのバイナリトーナメント選択を導入したSPEA2[3]を用いている．

2.3 ペナルティ法と引き戻し法を組み合わせた手法

本研究ではペナルティ法と引き戻し法を組み合わせた手法を提案する．提案手法は実行可能領域内の親個体から生まれた子個体には引き戻しを行い，実行可能領域外から生まれた子個体にはペナルティを与えるという方法である．

3. 数値実験

3.1 実験内容

本実験では対象問題DEB，SRN，TNKにおいて，定義域に対する実行可能領域の割合が異なる問題をそれぞれ用意した．そして前節で述べた3つの制約条件取り扱い手法を各対象問題に適用し，得られた解集合について結果を比較する．結果の評価にはI_{RNI}[4]，I_{LI}[5]と呼ばれる評価手法を用いている．これらの評価手法は相対評価であり，2つの解集合の優劣を決定する．どちらの評価手法も，評価値が100%に近い解集合の方がもう一方の解集合を優越していることを意味する．DEB，SRNに対してはI_{LI}を用いて比較し，TNKに対してはI_{RNI}を用いて比較を行った．母集団サイズを100とし，各対象問題に対して評価計算回数を表1に示すように設定

*同志社大学工学部助教授
†同志社大学工学部教授
‡同志社大学工学部知識工学専攻修士1年
§同志社大学工学部知識工学専攻修士2年

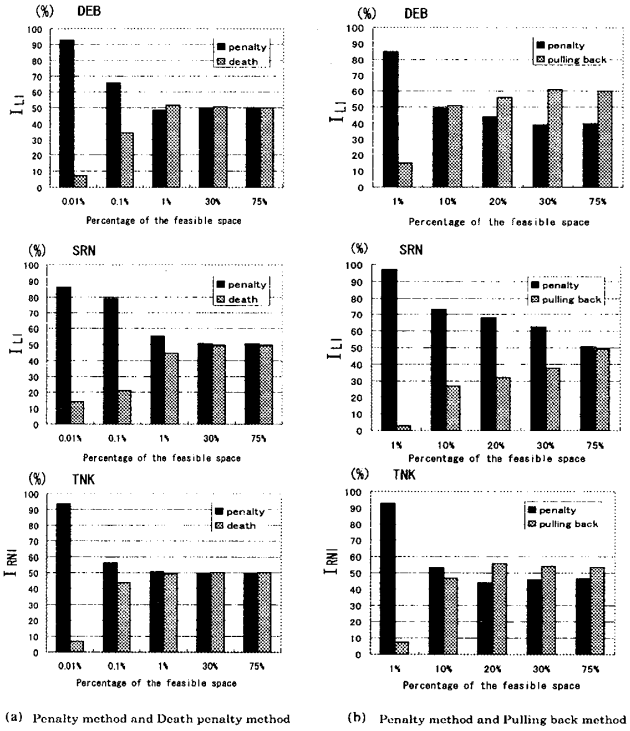
した (ただし制約領域 0.01%の SRN では評価計算回数 10000 とする). また本実験では個体の設計変数値の変化を 1 回評価としてカウントする.

表 1: 各対象問題における評価計算回数

対象問題	DEB	SRN	SRN(0.01%)	TNK
評価計算回数	6000	2000	10000	25000

3.2 ペナルティ法，デスペナルティ法，引き戻し法

図 1 に，ペナルティ法とデスペナルティ法の比較，およびペナルティ法と引き戻し法の比較結果を示す.



(a) Penalty method and Death penalty method (b) Penalty method and Pulling back method

図 1: 実験結果

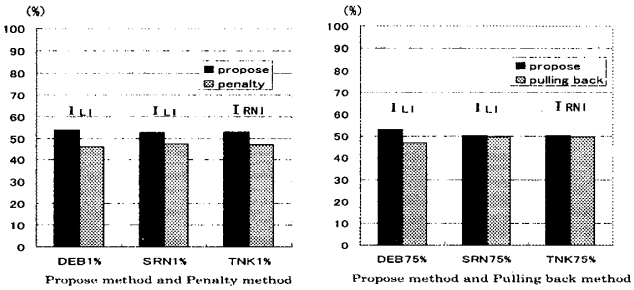
図 1 よりデスペナルティ法は定義域に対する実行可能領域が小さくなるにつれペナルティ法に劣る. デスペナルティ法では実行可能領域からの近接度合を考慮しないため，実行可能領域から遠く離れた個体も次世代に残る. よって実行可能領域内に個体を生成することが困難となる. 一方，ペナルティ法と引き戻し法の比較では，引き戻し法は実行可能領域の境界上に最適解が存在する DEB，TNK においては，定義域に対する実行可能領域の割合が大きい問題ほど良好な結果を示している. これは実行可能領域を外れた場合，実行可能領域内に引き戻す操作を行う引き戻し法は，実行可能領域境界付近に個体を生成しやすいためである.

この実験により以下のような結果が得られた.

- ペナルティ法によって実行可能領域外に存在する個体群を実行可能領域に集めることができる.
- 引き戻し法により実行可能領域の境界付近の探索を効率的に行える.

3.3 提案手法の結果

図 2 に，提案手法と，ペナルティ法，引き戻し法の比較を行った結果を示す. 左が各対象問題の実行可能領域を 1%とした提案手法とペナルティ法との比較，右が各対象問題の実行可能領域を 75%とした提案手法と引き戻し法との比較である.



Propose method and Penalty method　Propose method and Pulling back method

図 2: 提案手法

前節までの結果より，ペナルティ法は実行可能領域が小さい場合に優れた解探索性能を示し，引き戻し法は実行可能領域が大きい場合に優れた解探索性能を示していた. 図 2 より提案手法では実行可能領域が小さい場合にも大きい場合にも優れた解探索を行えていることがわかる.

4. 結論

- デスペナルティ法では制約条件からの近接度合を考慮しないため，実行可能領域が小さい問題では実行可能領域内に個体を生成することが難しい.
- ペナルティ法では実行可能領域からの近接度合を考慮した評価値割り当てによって，実行可能領域により近い個体を選択していくため，実行可能領域が小さい問題においても，実行可能領域内に個体を集めることができる.
- 引き戻し法は定義域に対する実行可能領域の割合が大きい問題ほど有効である. 特に制約境界付近を探索するのが得意であり，実行可能領域境界上に最適解が存在する DEB，TNK では実行可能領域の割合が比較的小さな 10%，20%の場合にもペナルティ法より優れた結果を示す. 実行可能領域が少なくとも 75%以上あれば SRN においてもペナルティ法と同等の性能を示す.

提案手法により，実行可能領域が大きい場合にも小さい場合にも優れた結果を得られることが確認できた.

参考文献

[1] Kalyanmoy Deb, An Efficient Constraint Handling Method for Genetic Algorithms, 1999

[2] K. Deb and S. Agrawal and A. Pratab and T. Meyarivan, A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, 2000

[3] E. Zitzler and M. Laumanns and L. Thiele, SPEA2: Improving the Performance of the Strength Pareto Evolutionary Algorithm, 2001

[4] K.C.Tan, T.H.Lee, and E.F.Khor. Increamenting Multi-objective Evolutionary Algorithms Performance Studies and Comparisons. In First International Conference on Evolutionary Multi-Criterion Optimization, pp. 111-125, 2001.

[5] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. Evolutionary Computation, Vol. 8, No. 2, pp.149?172, 2000.