

# ナップサック問題における評価値変動に対応した 遺伝的アルゴリズムの提案

田島 友祐<sup>†1,a)</sup> 中田 雅也<sup>†1,b)</sup> 佐藤 寛之<sup>†1,c)</sup> 高玉 圭樹<sup>†1,d)</sup>

**概要:** 本論文では、評価値の変化にロバストな解を探索可能な確率モデル GA である Robustness-oriented compact Genetic Algorithm (RcGA) を提案する。提案手法の新規性としては、従来の確率モデル GA の一つである compact Genetic Algorithm (cGA) [1] に比べて、(1) 評価値の高い確率モデルを生成するのではなく、評価値の変化にロバストな確率モデルを生成する点、(2) 確率モデルに学習分類子システム の概念である一般化 (generalization) の概念を導入した点が挙げられる。評価値が変化するナップサック 問題に適用したところ、cGA よりも変化にロバストである解 (評価値が大きく変動しない解) を求めるこ とが明らかになった。

## 1. はじめに

遺伝的アルゴリズム (Genetic Algorithm : GA) [2] は生物群の環境への適応概念を問題解決方法に見立てた解探索手法である。適応概念として、生物群の多様性を保つことを目的とした染色体の交叉・突然変異、自然淘汰があり、これにより与えられた制約条件において単一の評価指標を最良にする解を求めることが可能である。GA は新幹線等の工学的な分野における形状設計 [4]、仕事のシフトにおける最良な人員配置、船やバスの路線網構築 [5] に用いられるが、制約条件である単一の評価指標が変化しないという前提が必要である。評価指標が変化しないとは、例えば、工学的設計において単一の評価指標をコストにした場合、製作する製品設計が一意に定まれば、材料のコストも一意に定まることを言う。しかし、現実において材料のコストは常に一意ではなく日々変化していくため、評価指標が変化しないという前提は成り立たない場合が多い。その際、GA は変化に対応することが出来ず学習が困難になるため、変化に対応した学習が可能な手法が必要である。本論文では、評価指標の変化に対応可能な解を学習する Robustness-oriented compact Genetic Algorithm (RcGA) を提案し、従来の確

率モデル GA である compact Genetic Algorithm (cGA) [1] との比較を通して学習性能を検証する。

本論文の構成は次の通りである。以下、第 2 章では従来の確率モデル GA (cGA) について述べ、第 3 章では評価指標の変化に対応可能な学習手法 (RcGA) について述べる。第 4 章では変動型ナップサック問題における実験結果を示し、最後に第 5 章で本論文をまとめる。

## 2. compact Genetic Algorithm

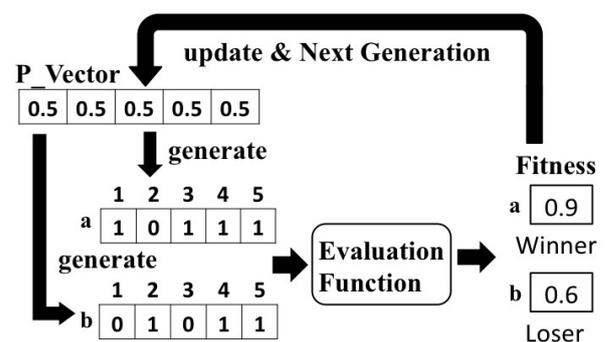


図 1 cGA の全体像

### 2.1 概念

図 1 は compact Genetic Algorithm (cGA) の全体の流れを示す。cGA は確率ベクトル (Probability Vector : P-Vector) を用いた遺伝的アルゴリズム (Genetic Algorithm :

<sup>†1</sup> 現在、電気通信大学情報理工学研究所 東京都調布市調布が丘 1-5-1

Presently with Graduate School of Informatics and Engineering, The University of Electro-Communications, Chofu, Tokyo

a) monteishi.usk@gmail.com

b) masaya.n04@gmail.com

c) sato@hc.uec.ac.jp

d) keiki@inf.uec.ac.jp

GA) であり, GA での操作である染色体の交叉, 突然変異, 自然淘汰は存在しない. cGA が対象とするデータは 2 値 (0, 1) のバイナリデータである. 確率ベクトルが持つ実数値はデータが "1" となりうる確率を示し, 確率ベクトルを基にした個体生成, 生成された個体間の比較による確率モデルの更新により学習が進められる. 確率ベクトルを基にした個体生成, 更新は GA での交叉, 突然変異, 自然淘汰と同等の効果を出す操作である. 更に, 確率ベクトルは GA での母集団の役割を担うため, GA の一番シンプルな形である Simple GA よりも少ないメモリ数で学習を進めることが可能である. 次章では cGA の学習操作である確率ベクトルを基にした個体生成, 確率ベクトルの更新について説明する.

## 2.2 メカニズム

### 2.2.1 個体生成

cGA は確率ベクトルの確率に従い, 問題により与えられる配列要素を持つ個体を 2 つ生成する. 確率ベクトルが持つ確率はデータが "1" になる確率を示している. 図 2 において確率ベクトルの 1 番目の実数値は 0.0 であるため, データは必ず "0" になる. 対して, 3 番目の実数値は 1.0 であるため, データは必ず "1" になる. 4,5 番目のように実数値が 0.5 である場合, 同確率でデータが "0" と "1" になることを表している. 確率ベクトルを基にした個体生成法は GA における交叉の役割を担っており, 多様な個体を用いた更新を可能にする.

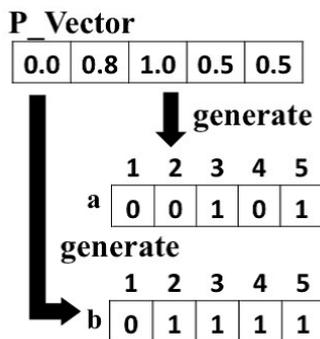


図 2 cGA における個体生成

### 2.2.2 確率ベクトルの更新

生成された 2 個体に対し, 単一の評価関数を用いて 2 個体の評価値をそれぞれ算出し, 評価値が高い個体を winner 個体, 低い個体を loser 個体と設定する. 次に, winner 個体と loser 個体の遺伝子情報 (ビット構成) を比較することで, winner の遺伝子情報を次世代に残すように確率ベクトルを更新する. 具体的には, 遺伝子情報が 2 個体間で異なる箇所において, 1) winner 個体側が "1" の時は確率ベクトルの値を更新率だけ増加させ, 2) winner 個体側が "0"

の時は確率ベクトルの値を更新率 ( $\alpha$ ) だけ減少させる. 図 3 において, 生成された個体 a, b の評価値は 0.9, 0.6 のため, winner 個体は a, loser 個体は b となる. 2 個体間で遺伝子情報が異なる箇所は 2 番目と 4 番目の遺伝子であり, 2 番目の遺伝子に関しては winner 個体側が "1" の時, 4 番目の遺伝子に関しては winner 個体側が "0" の時であるため, それぞれ更新率分増減している. 図での更新率は 0.1 に設定しているが, 更新率は問題に応じて, 事前に設定するパラメータである. この更新の操作を繰り返すことにより, 確率ベクトルは単一の評価関数を満たす良好な解の遺伝子情報を推定することが可能となる.

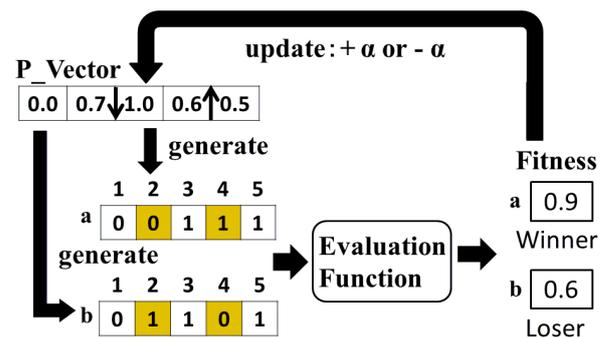


図 3 確率ベクトルの更新

## 2.3 アルゴリズム

上記のようなメカニズムを備えた cGA のアルゴリズムを図 4 に示す. 確率ベクトルの実数値は 0.5 に初期化され, 個体生成, 更新を繰り返していく. この操作は確率ベクトルの全ての实数値が "0" もしくは "1" に収束するまで繰り返される.

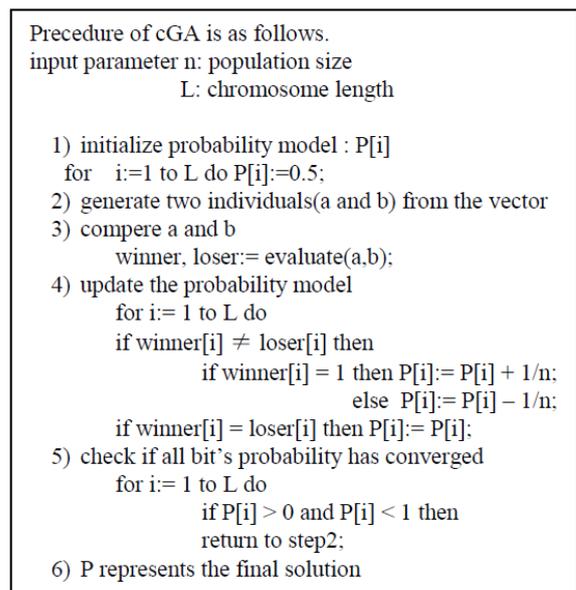


図 4 cGA のアルゴリズム



図 5 RcGA の全体像

### 3. Robustness-oriented compact Genetic Algorithm

#### 3.1 概念

図5はRobustness-oriented compact Genetic Algorithm (RcGA) の全体の流れを示す。RcGAはcGAと同様に確率モデルを用いた評価値変化に対応可能なGAである。cGAと異なる点としては以下の点が挙げられる。(1) 学習前に生成した個体と評価関数により算出した評価値のデータを集めたデータセットを生成する点、(2) 更新時に使用する個体を生成するのではなくデータセットから選択をする点、(3) 確率モデルに学習分類子システム概念である一般化 (generalization) の概念 [3] を取り込んでいる点。次章ではRcGAのメカニズムである確率モデルの更新、データセットの生成、確率モデルの一般化について説明する。

#### 3.2 メカニズム

##### 3.2.1 確率モデルの更新

確率モデルの更新方法はcGAと同様にwinner個体とloser個体の遺伝子情報 (ビット構成) を比較することで、遺伝子情報が異なる箇所に関して、1) winner個体側が”1”の時は確率モデルの値を更新率だけ増加させ、2) winner個体側が”0”の時は確率モデルの値を更新率 ( $\alpha$ ) だけ減少させる。この際、cGAでは確率モデルを基に生成した個体により比較をしたが、RcGAではデータセットから選択された個体より選択をする。データセットに関しては次節で説明をする。

##### 3.2.2 データセット生成

RcGAでは、変動にロバストな組み合わせ解を求めるために個体生成でなくデータセットからの選択を用いる。このデータセットには個体の組み合わせ情報とその組み合わせによる評価値を合わせ持ったものになっており、同じ組み合わせ情報を持ち、評価値が異なる情報を入れることが可能である。評価値が異なる同等の組み合わせ情報をデータセットに持つことで、変動度合いが小さく評価値が高い

組み合わせ情報を学習することが出来る。

##### 3.2.3 確率モデルの一般化

RcGAでは、確率モデルに学習分類子システム概念である一般化 (generalization) の概念を取り入れている。これは、データセットからの選択により確率モデルを更新するため、cGAのように最適解に収束することがなく、そのため2値問題への適用が困難のためである。具体的には、学習後の確率モデルの値が”0.25”以下である場合は確率モデルの値を”0”に、”0.75”以上である場合は確率モデルの値を”1”に、確率モデルの値が”0.25~0.75”の間であれば”#”と言う、”0”, ”1” どちらでも取りうる事が可能なドントケアと呼ばれるものにする。

#### 3.3 アルゴリズム

上記のようなメカニズムを備えたRcGAのアルゴリズムを図6に示す。確率ベクトルの実数値は0.5に初期化され、個体選択、更新を繰り返していく。この操作は予めパラメータとして設定する世代数を満たすまで繰り返される。

```

Procedure of Proposed method is as follows
input parameter G: generation
                L: chromosome length

1) To generate dataset
2) To initialize probability model : P[i]
    for i:=1 to L do P[i]:=0.5;
3) To select two individuals(a and b) from the dataset
4) To compare a and b
    winner, loser:= evaluate(a,b);
5) To update the probability model
    for i:= 1 to L do
    if winner[i] ≠ loser[i] then
        if winner[i] = 1 then P[i]:= P[i] + 1/n;
        else P[i]:= P[i] - 1/n;
6) To check generation_count
    if generation_count != G
        return to step3;
7) P represents the final solution
    
```

図 6 RcGA のアルゴリズム

### 4. 実験

#### 4.1 実験内容

RcGAの性能を評価するために、組み合わせ最適化問題として代表的なナップサック問題を拡張した変動型ナップサック問題を用い、従来の確率モデルGAであるcGAと比較する。実験方法としては、価値・重量にランダム値を与えた基準のアイテムセットを1セット分生成し、事前に定めた変動率に従い、基準のアイテムセットを変動させることによりアイテムの価値・重量を変化させたアイテムセットを学習データとして複数作り学習をする。その後、学習

データと同等の変動率により価値・重量を変化させたアイテムセットを評価データとして作り評価をする。

#### 4.2 変動型ナップサック問題

変動型ナップサック問題は組み合わせ最適化問題として代表的なナップサック問題を拡張した問題である。一般的なナップサック問題は、ある制約条件の基、複数の要素を取捨選択することで、評価値が最大となる組み合わせを探索する問題である。具体的には、アイテム数を  $n$ 、ナップサックの重量制限を  $W$ 、 $i$  番目のアイテムの価値を  $p_i$ 、重量を  $w_i$ 、アイテムの選択・非選択を表す遺伝子中の  $i$  番目の値を  $x_i$  ( $x_i = 0$  or  $x_i = 1$ ) としたとき、ナップサックに入れられたアイテムの総価値は式 (1) の形で定式化され、この  $f(x)$  が最大となる組み合わせを探索する問題である。

$$f(x) = \begin{cases} \sum_{i=1}^n p_i x_i & \text{if } \sum_{i=1}^n w_i x_i \leq W \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

変動型ナップサック問題は一般的なナップサック問題のような静的環境な問題でなく、アイテムの価値・重量の値が変動する動的な問題である。そのため、あるアイテムセットの価値・重量において  $f(x)$  の値が最大になる解の選択方法では、アイテムセットの価値・重量の変動に対応できず、価値の低下・重量増加による制約条件の違反が起きてしまう。そのため、変動型ナップサック問題では変動時の影響が少なくなるようなロバストな組み合わせの探索が必要になる。



図7 変動型ナップサック問題

図7は変動型ナップサック問題の例である。 $x$ はアイテムの選択する組み合わせを示し、“1”の箇所のアイテムをナップサックに入れる、つまり“1”の箇所の価値・重量を加えることを意味する。図7での変動前のアイテムセットにおいては、 $f(x) = 66 + 50 + 66 + 40 = 222$ の評価値を持ち、この時の制約条件は  $\sum_{i=1}^n w_i x_i = 56 + 48 + 56 + 41 = 201 \leq W$  であるため制約条件を満たす。しかし、変動後の価値・重量を持つアイテムセットにおいては、 $f(x) = 43 + 38 + 45 + 40 = 166$ の評価値であり、制約条件は  $\sum_{i=1}^n w_i x_i = 67 + 51 + 53 + 68 = 239 > W$  となり制約条件を満たさず、式(1)により  $f(x) = 1$  となる。

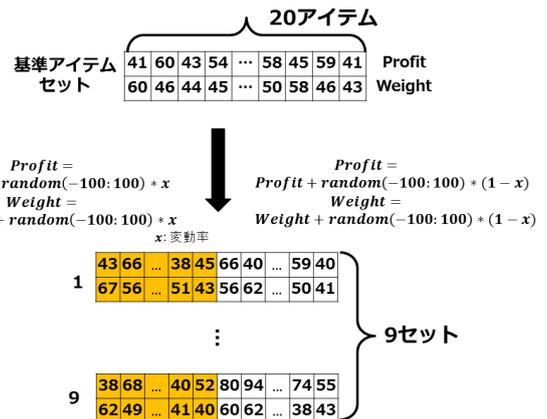


図8 学習アイテムセットデータの生成

#### 4.3 評価方法とパラメータ設定

評価方法としては、あらかじめ用意した学習アイテムセットデータを用いて学習をし、その後、評価アイテムデータセットにおいて評価値を求める。具体的には、学習アイテムセットデータとして9つのアイテムセットを生成する。これらのアイテムセットは、基準となるアイテムセットに対し、半分のアイテムは変動率  $x$  により重量と価値を変動させ、残りのアイテムは変動率  $1 - x$  により変動させる。 $x$ の値が1、もしくは0に近い程、アイテム間での変動具合が大きく、 $x$ の値が0.5に近い程、全アイテムの変動具合は同じものとなる。評価アイテムデータセットも学習アイテムセットデータと同様に生成する。評価値の出し方は式(1)と同様である。評価の際、RcGAで学習した確率モデルを一般化し、“#”よって取りうる全通りの解を生成する。これにより生成される同数の解をcGAでも生成し全個体に対し評価をする。パラメータ設定は、世代数を10000000、更新率を  $1.0/100000 = 0.000001$ 、変動率： $x = 0.9$ としている。RcGAのデータセットは、学習データセットの制約を満たす全通りの解と制約違反をする解の半分のデータを持つ。制約違反をする解を半分アイテムセットに入れる選択方法はランダム選択を用いる。

#### 4.4 結果

実験結果を図9に示す。縦軸は評価値を示し、横軸にはcGA, RcGAを示し、黒いバーは分散値を示す。図より、RcGAでは変動にロバストな解を学習できているのに対し、cGAでの学習では変動に対応が出来ず、解である全個体で制約違反になってしまっている。

#### 5. おわりに

本論文では、評価値の変化にロバストな解を探索可能な確率モデルGAであるRobustness-oriented compact Genetic Algorithm (RcGA)を提案した。変動型ナップサック問題を用いて従来の確率モデルGAの一つであるcompact

Genetic Algorithm (cGA) と比較したところ、cGA では変化に対応できず制約違反解が多く存在してしまったのに対し、RcGA では変化に対応可能な解が生成された。今後の課題として、変動率の異なるアイテムを自動的に判別し特徴を抽出する手法に拡張する。

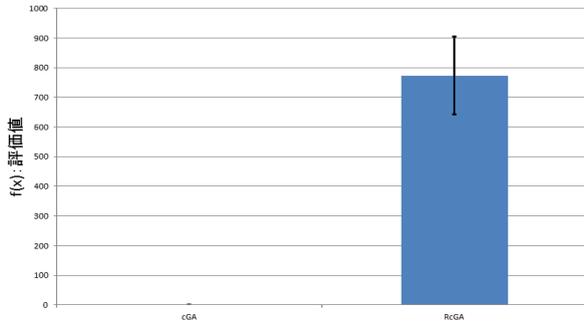


図 9 変動率： $x = 0.9$  での実験結果

## 参考文献

- [1] G. R. Harik, F. G. Lobo, and D. E. Goldberg :The Compact Genetic Algorithm, IEEE Transactions on Evolutionary Computation, Vol. 3, No. 4,(1999)
- [2] D. E. Goldberg : Genetic Algorithms in Search, Optimization & Machine Learning, Addison-wesley, Reading(1989)
- [3] Holland, J. H. : Escaping Brittleness: The Possibilities of General Purpose Learning Algorithms Applied to Parallel Rule-based System, Machine Learning, 2, 593 623 (1986)
- [4] Kei,SAKANOUE : Series N700 Shinkansen Train and Its Effect of Energy Saving, Journal of the Japan Society for Precision Engineering Vol. 76 No.1 P41-45 (2010)
- [5] Hiroto Kitagawa, Keiji Sato, Keiki Takadama, Hiroyuki Sato, and Kiyohiko Hattori : Robust Bus Route Optimization to Destruction of Roads, WEIN (2013)