

評価関数の重みパラメータを推定する対話型遺伝的アルゴリズム

石川英太郎<sup>†a)</sup> 石田 崇<sup>††b)</sup> 後藤 正幸<sup>†††c)</sup>

Interactive Genetic Algorithm to Estimate Weight Parameters of Evaluation Function

Eitaro ISHIKAWA<sup>†a)</sup>, Takashi ISHIDA<sup>††b)</sup>, and Masayuki GOTO<sup>†††c)</sup>

あらまし 遺伝的アルゴリズムによる最適化問題の解法において、解（個体）の評価値が明示的に定式化できないとき、ユーザが直接解の評価を繰り返すことにより最適解を得ようとする対話型進化計算（IEC）が知られている。IECの研究は盛んに行われているが、デザインの良し悪しのように、対象の多くは評価関数が全く定式化できない場合を想定していることが多い。しかし、明示的な評価関数が存在するが、そのパラメータがユーザによって異なる場合においても IEC が有効であると考えられる。そこで、本研究では評価関数が個別評価値の線形和で表され、それらの重要度がユーザによって異なる場合の対話型遺伝的アルゴリズムを提案する。観光スケジュール作成問題などの実問題に適用して本手法の有効性を示す。

キーワード 対話型遺伝的アルゴリズム, パラメータ推定, 最小二乗法, リッジ回帰

1. ま え が き

遺伝的アルゴリズム (Genetic Algorithm : GA) [1] は解を複数の遺伝子から構成される個体で表現し、生物の進化プロセスを模倣したメタヒューリスティックな最適化アルゴリズムである。GA では優れた解を残し、劣った解を捨てるという考え方に基づいて評価、選択、交叉、突然変異などの遺伝的操作を繰り返すことで、準最適解を得ることができる。そのため通常は解を評価する指標（評価関数）が定式化できる問題を適用対象とすることが多い。

しかし、洋服のデザインなどの最適化 [2] では、ユーザの主観的评价が高い解の探索が目的となり評価関数を明示的に定式化することができない。このよう

な問題を解決するために対話型遺伝的アルゴリズム (Interactive Genetic Algorithm : iGA) [3] が提案されている。iGA では通常の GA のように定式化された評価関数により解が評価されるのではなく、提示された解に対してユーザ自身が評価値を付与する。iGA ではユーザへの解の提示、ユーザによる解の評価、選択、交叉、突然変異の操作を繰り返すことによりユーザ自身にとっての最適解を得ることを目指している。また、iGA ではユーザが直接解を評価することを繰り返し行うため、ユーザの負担が問題となっており、ユーザへの負担を減らすための研究が多くなされている [4], [5]。

一方、GA は観光スケジュール作成問題など、複数の評価要素を総合的に考え最適化する問題に適用されることも多い [6]。例えば観光スケジュール作成問題では「観光地の重要度」や「移動時間」、「到着時刻との差」を同時に考慮する必要があり、これらをバランス良くする解の発見が求められる。複数の評価要素が存在するとき、評価要素の重み付け和を評価関数とするスカラ化のアプローチと多くのパレート解を得ることができる多目的 GA のアプローチ [7], [8] が存在する。多目的 GA では一度の計算で多くのパレート解を得ることができるが、アルゴリズムに内在するパラメータの調整や計算コストが多いなどの問題があるため、スカラ化が用いられるケースも多い [9]~[11]。スカラ化

<sup>†</sup> 早稲田大学大学院創造理工学研究所経営システム工学専攻, 東京都 Graduate School of Creative Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan

<sup>††</sup> 早稲田大学メディアネットワークセンター, 東京都 Media Network Center, Waseda University, 1-104 Totsukacho, Shinjuku-ku, Tokyo, 169-8050 Japan

<sup>†††</sup> 早稲田大学創造理工学部経営システム工学科, 東京都 Faculty of Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan

a) E-mail: eitaro@fuji.waseda.jp

b) E-mail: ishida@it.mgmt.waseda.ac.jp

c) E-mail: masagoto@waseda.jp

を用いる場合、評価関数の重みを適切に設定した上で GA が適用される。このとき、GA による探索結果のパフォーマンスは重みパラメータのチューニングに大きく依存するが、その重みパラメータは各評価要素をどの程度重要視するかを表しており、ユーザの主観によって適切に設定されるべきケースも多い。そのため、重みパラメータをチューニングしてから何度も GA を繰り返し、結果を吟味するという試行錯誤が必要になってしまう。このような場合、iGA のような対話型のアルゴリズムにより、適切な重みパラメータを自動学習しながら探索を進める対話機能が備われば、ユーザの価値基準に見合った解の探索が可能となることを期待できる。

そこで、本研究では評価関数が各要素の線形和で表され、各要素の重視の度合いがユーザによって異なる問題に対して、重みパラメータを推定しながら解の探索を進める iGA を提案する。更に、多目的ナップサック問題、観光スケジュール作成問題、多目的巡回セールスマン問題に適用し、提案手法の有効性を検証する。

## 2. 関連研究

複数の評価項目の重み付き線形和を評価関数とし、GA を利用して最適化する適用例は多く存在する。例えば、山本らは燃料中心の温度や燃料利用効率などを評価要素として原子燃料の炉心内配置問題を GA を用いて定式化している [9]。森本らは勤務負荷や休日回数などを評価要素として看護師の勤務表作成を GA を用いて定式化した [10]。松島は沿道の騒音対策のためには、騒音対策のためにかかる費用と騒音防止の効果をバランスよく考慮した騒音対策の選択をする必要があると考え、その組合せを最適化する問題として、GA を用いた [11]。これらの研究では、評価関数を重み付き線形和の形で与え、各評価要素の重みは経験的にチューニングされている。

評価関数の重みパラメータを自動調整する研究も多くなされてきた。村田らは多制約条件を課した場合の、ペナルティ法を用いた目的関数作成手法を提案している [12]。また大筆はオセロゲームにおける強いコンピュータプログラムを作成するために、コンピュータがどの位置に駒を置くのが最適かを求めるための評価関数を作成している [13]。また金子らは将棋における強いコンピュータプログラムを作成するために将棋の棋譜を用いた評価関数を作成して、各項目の重み付けを自動化している [14]。これらの研究では重みパラ

メータを推定しているが特定の問題のみに適用でき、一般的な最適化問題の評価関数の重み推定方法ではない。

次に、最適解がユーザによって異なる問題に対して、ユーザの価値基準に見合った解を見つけ出す研究について説明する。GA によって生成された解に対してユーザが評価を与えながら探索を進める方法論は対話型 GA (iGA) と呼ばれる。iGA は評価関数を明示的に定式化することができない問題において、ユーザにとっての最適解を見つけ出す方法として多くの研究がなされてきた。例えば、iGA のユーザの負担を軽減した iGA としては、渡辺らの一対比較評価による iGA [4] や Gong らの母集団サイズを大きくした iGA [5] などがある。また、伊藤らは嗜好の多峰性に対応した iGA の解生成手法 [2] を提案している。一方、中山は多目的計画法において複数の評価要素が存在するとき、ユーザがコンピュータに各評価要素の希求水準を直接入力することにより、ユーザの価値基準に見合った解を対話的に見つけ出す方法を提案している [15]。

## 3. 提案手法

### 3.1 提案手法の概要

前章で示したように、複数の評価項目の重み付き線形和を評価関数とする解法や対話型のアルゴリズムによりユーザの最適解を得ようとしている研究はそれぞれ存在する。

前者の重み付き線形和を評価関数として探索する方法では、一般に重みパラメータをユーザが適切に設定しなければならないが、そのためには、探索の終了後に得られた解を見ながら、試行錯誤的に重みパラメータをチューニングする作業を何度も繰り返さなければならない。一方で、後者の対話型のアルゴリズムのように、1 回の解探索のプロセス内で、ユーザが解評価を行い、それを活用しながら解探索を進めるような対話型のアルゴリズムが構築できれば、評価関数の重みパラメータをユーザの解評価から自動的にチューニングしつつ、効率的に優良な解を探索できる可能性がある。

評価関数が複数の評価項目の重み付け和で表され、個々の重みパラメータはユーザに依存する状況を想定する。今、 $m$  個の評価項目に対して重みパラメータのベクトルを  $\mathbf{w} = (w_1, w_2, \dots, w_m)^T$  とする。このとき、解  $s$  の評価関数  $f\mathbf{w}(s)$  を式 (1) のようにモデル化

する。

$$f\mathbf{w}(s) = \sum_{k=1}^m w_k x_k(s) + \varepsilon \quad (1)$$

$w_k$ :  $k$  番目の評価項目の重みパラメータ

$x_k(s)$ : 解  $s$  の  $k$  番目の評価項目の評価値

$\varepsilon$ : ユーザの評価の誤差

$f\mathbf{w}(s)$ : ユーザの解  $s$  に対する総合評価値

評価関数が式 (1) のように表されるとき、重みパラメータ  $w_k$  は解空間の規模や問題の複雑さまたはシステムを利用するユーザの主観によって異なることも多いと考えられる。評価関数が式 (1) で表されるような例として、鉄道を使って複数の駅を巡回すること（スタンプラリー）を考える。スタンプラリーは巡回セールスマン問題でコストが複数存在する問題と考えることができ、コストとして主に「運賃」、「移動時間」、「乗換回数」が挙げられる。「運賃」を重視するユーザもいれば、「移動時間」を重視するユーザもいるはずである。

本研究ではまず、ユーザにとっての最適な重みパラメータを推定することを考える。仮に最適に近い重みパラメータを推定できれば、推定された重みパラメータをもとに GA を実行して最適化することで、ユーザごとに異なった最適解を見つけることができる。iGA はユーザへの負担が問題となっているが、評価関数が存在しない状況を想定しているため、ユーザの評価回数が多くなってしまってもやむを得ない。本研究では評価関数が存在し、その重みパラメータがユーザや利用場面によって異なる状況を想定している。本研究が想定する状況では、重みパラメータを求めるためにユーザは提示された幾つかの解について評価を行うものとする。その際、なるべく少ない評価回数で済むことが望ましい。そこで、有限個の解評価から、最小二乗法によって、重みパラメータを推定する方法を考える。提案手法のイメージ図を図 1 に示す。

Step1 でシステムはまずユーザに対して複数の解を提示する。次に Step2 でユーザは提示された解に対して、評価点をつける。ユーザが解に対して評価値を付与した後 Step3 でシステムはユーザがつけた評価点をもとに最小二乗法によって最適な重みパラメータを求める。ここで、ユーザが評価した解の数と比べてパラメータ数が多い場合はリッジ回帰 [16] を用いて重みパラメータを推定する。Step4 で推定されたパラメータを用いた評価関数を使い、GA を実行し準最適解を求

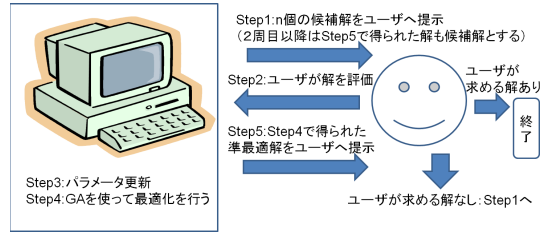


図 1 提案手法のイメージ

Fig. 1 Image of proposal method.

め、Step5 でその準最適解をユーザへ提示する。ここで、この解がユーザの求める解であればここで終了するが、ユーザが求める解でないと判断したら再度これらの処理を繰り返す。

### 3.2 提案手法の詳細

#### 3.2.1 提案アルゴリズム

ここでは提案するアルゴリズムを以下に示す。

[提案アルゴリズム]

Step0: アルゴリズムの繰り返し数を  $j = 1$ ,  $\hat{s}$  をランダムに生成した解とする。候補解の生成方法 (3.2.2) により  $n$  個の多様な解を生成し解集合  $S_n$  とする。

Step1: 解集合  $S_n$  をユーザへ提示する。

Step2: ユーザは  $S_n$  に対して相対評価を行う。

Step3: ユーザが与えた得点  $f(1), f(2), \dots, f(N)$  ( $N = n \times j$ ) に対して後述の式 (3) または式 (5) を用いて重みパラメータ  $\mathbf{w}$  を推定する。

Step4:  $\hat{s}$  とランダムに生成した解集合により、GA の初期母集団を生成する。Step3 で推定された  $\mathbf{w}$  を用いた評価関数 (式 (1)) を使い、GA を実行し準最適解を求める。

Step5: Step4 で得られた準最適解を  $\hat{s}$  とし、ユーザへ提示する。

ここで、提示された準最適解をユーザが最適と判断すれば  $\hat{s}$  が最適となり終了。そうでなければ Step6 へ  
Step6:  $j = j + 1$  として、候補解の生成方法による  $(n - 1)$  個の多様な解と  $\hat{s}$  で構成された  $n$  個の解を新たな解集合  $S_n$  として、Step1 へ戻る。 □

#### 3.2.2 候補解の生成方法

Step1 では、ユーザに対して  $n$  個の候補解を提示している。本項ではどのようにして候補解を生成しているかについて説明する。提案手法でも述べたが、提示される解はユーザが相対評価を行うためのものなので、ユーザが評価しやすいうように生成する必要がある。できるだけ異なった (多様な) 解が出力されればユーザ

$p=0.3$ の場合の例

候補解	1	0	0	1	1	0	1	0
一様乱数	0.11	0.35	0.53	0.09	0.25	0.90	0.29	0.31

図 2 候補解出力方法

Fig. 2 Method to create a set of candidate individuals.

が相対評価がしやすいと考えられる。

・候補解が 0 または 1 の遺伝子で表される場合

候補解が遺伝子で表される例を図 2 に示す。図 2 のように、各遺伝子に対して一様乱数  $U$  を振る。ここで各遺伝子に対して  $U < p$  ならば選択とし、そうでなければ非選択とする。ここで対象問題としてナップサック問題 [17] を用いる場合、 $p$  は候補解が制約コストに近い値（ただし制約コスト以下）となるように設定する。

・候補解が番号で表される場合

巡回セールスマン問題 [18] のように候補解が番号で表される場合は前述の方法で候補解を生成することができない。そこでまず重みパラメータをある範囲に設定し、その範囲のもとで一様乱数を生成する。一様乱数で生成された重みパラメータを使って GA を用いた準最適解を候補解とする。ただ、準最適解を使うと解の多様性が失われることが多いので、GA の世代数を少なく設定した解を候補解としている。

3.2.3 重みパラメータの推定方法

前節で説明した最小二乗法を用いたパラメータ推定法について述べる。ユーザが解  $s$  に対して付与した得点  $f(s)$  が式 (1) に従うと仮定し、

$$\mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f(1) \\ \vdots \\ f(N) \end{bmatrix},$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(1)^T \\ \vdots \\ \mathbf{x}(N)^T \end{bmatrix} = \begin{bmatrix} x_1(1) & \cdots & x_m(1) \\ \vdots & \ddots & \vdots \\ x_1(N) & \cdots & x_m(N) \end{bmatrix}$$

とおけば、最小二乗法による重みパラメータの推定値は以下のように示される。

$$\mathbf{Q}(\mathbf{w}) = \|\mathbf{f} - \mathbf{w}^T \mathbf{X}^T\|^2 \quad (2)$$

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{f} \quad (3)$$

$\hat{\mathbf{w}}$ : 推定された重みパラメータのベクトル

ユーザが評価した解の個数と比べて、重みパラメータの数が少なければ上記の式 (2) (最小二乗法) を用いて推定すればよいが、ユーザが評価する解の個数と比べて、重みパラメータの数 (評価要素数) が多い場合は上記の式 (2) を求めることができない。そこでリッジ回帰を用いる。リッジ回帰では通常の最小二乗法に加えて、重みパラメータの絶対値が大きくなりすぎないようにするために以下の  $\mathbf{Q}(\mathbf{w})$  を最小化する。

$$\mathbf{Q}(\mathbf{w}) = \|\mathbf{f} - \mathbf{w}^T \mathbf{X}^T\|^2 + \lambda \mathbf{w}^T \mathbf{w} \quad (4)$$

$\lambda$ : 重みパラメータのペナルティを調節するための定数

式 (4) が最小となるように推定された重みパラメータは以下のように与えられる。

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{f} \quad (5)$$

ただし、 $\mathbf{I}$  は  $m \times m$  の単位行列を表す。

3.2.4 アルゴリズムが繰り返し行われる際の工夫  
上記のアルゴリズムを繰り返すと、Step5 で一度ユーザに最適候補が提示される。1 回目に提示される最適候補はユーザにとっての最適でないケースが多いため、一周でアルゴリズムが終了することは通常少なく、再び Step1 へ戻ることも多いと考えられる。しかし、1 回目に提示される最適候補もユーザにとっての最適ではないものの、ユーザにとってそれなりに良い解である可能性が高い。一周でアルゴリズムが終了しないとき再び Step1 へ戻り、Step4 で再び GA を回すことになる。その際、ユーザによる相対評価を行った後 ( $j-1$ ) 回目の Step5 で得られた準最適解  $\hat{s}$  を  $j$  回目の GA の初期個体群に混ぜることで GA の収束が早まると考えられる。

4. 評価実験

4.1 評価実験の対象問題

本節では評価実験の対象問題について説明する。対象問題としては多目的ナップサック問題 (スカラ化)、観光スケジュール作成問題、多目的巡回セールスマン問題 (スカラ化) とする。以下ではそれぞれについて詳しく説明する。

4.1.1 多目的ナップサック問題

多目的ナップサック問題 [19] はナップサック問題の値が複数存在する状況を想定したものである。多目的ナップサック問題では制約コストのもとで  $L$  個の品物の中から購入する品物を決定する問題を考える。こ

表 1 多目的ナップサック問題の問題設定  
Table 1 Setting of multiobjective knapsack problem.

品物数	制約コスト	コスト生成方法	価値生成方法
15	250	30~50 の整数乱数	0~100 の整数乱数

ここで、各品物には幾つかの価値項目が設定されており、どの価値を重視するかはユーザによって異なる。評価関数と制約条件を式で表すと以下ようになる。ここで解  $s$  はユーザが購入した品物を 1、購入していない品物を 0 としたベクトルで表す。

$$s = (a_1, a_2, \dots, a_L)$$

$$a_j = \begin{cases} 1 & \text{品物 } j \text{ を購入} \\ 0 & \text{品物 } j \text{ を非購入} \end{cases} \quad (6)$$

$$f\mathbf{w}(s) = \sum_{k=1}^m w_k x_k(s) \quad (7)$$

$$x_k(s) = \sum_{j:a_j=1} v_k(j) \quad (8)$$

$$\sum_{j:a_j=1} c(j) < C \quad (9)$$

$x_k(s)$ : 解  $s$  の  $k$  番目の要素の価値

$v_k(j)$ : 品物  $j$  の  $k$  番目の要素の価値

$c(j)$ : 品物  $j$  のコスト

$C$ : 制約コスト

評価実験で使う多目的ナップサック問題の問題設定を以下の表 1 に示す。

#### 4.1.2 観光スケジュール作成問題

観光スケジュール作成問題 [6] は、主に市内観光で観光候補地が多数存在するとき、制約時間のもとで効率良く、しかもできるだけ多くの観光地を巡回可能なスケジュール案を探索する問題のことである。具体的には、ユーザ入力である出発地点を  $a$ 、到着地点を  $b$ 、出発時刻と到着時刻をそれぞれ  $time(a)$ 、 $time(b)$  とする。観光地データベースとして、総観光地数  $p$ 、観光地集合  $D = \{d_1, d_2, \dots, d_p\}$ 、観光地  $d_i$ 、 $d_j$  間の移動時間  $time(d_i, d_j)$ 、各観光地の重要度及び滞在時間のデフォルト値  $imp(d_i)$ 、 $stay(d_i)$  が存在し、各観光地の重要度と滞在時間はユーザが修正することが可能であり修正された値を  $imp'(d_i)$ 、 $stay'(d_i)$  とする。ただし、一つの観光スケジュール  $S$  は  $S = \{d_0(=a), d_1, d_2, \dots, d_q, d_{q+1}(=b)\}$  で表され  $k$  は巡回観光地数となる。また  $T = time(b) - time(a)$ 、

許容される観光時間オーバを  $t_{max}$  とおけば観光スケジュールの制約条件は式 (10) のようになる。

$$T_s \leq T + t_{max} \quad (10)$$

ここで  $T_s (= \sum_{i=1}^q stay'(d_i) + \sum_{i=1}^q time(d_i, d_{i+1}))$  は総観光時間を表す。評価実験では、対象を「京都市内の観光地」とし、観光地数は 22、観光時間は 8 時間とした。また、観光地の重要度はユーザが 10 点満点でつけると仮定し、移動時間の単位は「分」とする。

また観光スケジュール作成問題の評価関数は以下の式 (11) で表される。重みパラメータは  $\alpha$  (重要度の高い観光地をどれだけ多く訪れているかを表す項目)、 $\beta$  (移動時間の合計のペナルティに対する項目)、 $\gamma$  (希望到着時刻との差のペナルティに対する項目) である。

$$f(S) = \alpha \sum_{i=1}^q imp'(d_i) - \beta \sum_{i=1}^q time(d_i, d_{i+1}) - \gamma |T_s - T| \quad (11)$$

#### 4.1.3 多目的巡回セールスマン問題

多目的巡回セールスマン問題 [19] とは巡回セールスマン問題のコストが複数ある問題と考えることができる。本研究では、スタンプラリーなどのために鉄道を使って複数の駅を巡回する問題を対象とする。乗換案内によれば鉄道利用の際のコストは「運賃」、「移動時間」、「乗換回数」があるがどのコストを負担に感じるかはユーザによって異なると考えられる。地点  $i, j$  間の運賃、移動時間、乗換回数をそれぞれ  $cost(i, j)$ 、 $time(i, j)$ 、 $frequency(i, j)$  とする。また、巡回路  $S$  の枝集合を  $E(S)$  とすれば多目的巡回セールスマン問題の評価関数は式 (12) のようになる。なお、巡回セールスマン問題の制約条件は巡回すべき「都市」をすべて訪れていて「出発地=到着地 (出発地と到着地は決まったある地点)」であればよい。評価実験では首都圏の 10 箇所の駅をスタンプラリーで巡回すると仮定した。

$$f(S) = \sum_{\{i,j\} \in E(S)} (\alpha cost(i, j) + \beta time(i, j) + \gamma frequency(i, j)) \quad (12)$$

#### 4.2 実験内容

以下で各評価実験 (実験 1~実験 4) の概要を示す。実験 1 と実験 2 はパラメータ数が少ない場合を想定し、実験 3 と実験 4 はパラメータ数が多い場合を想定した実験である。実験 1 と実験 3 ではユーザが線形モ

デルに基づいて評価値を付けたと仮定し、実験 2 と実験 4 は実際にユーザに評価値を付与してもらった。なお、実験ではユーザが一度に評価する解の数を  $n = 5$  とする。

実験 1： 真のパラメータを用意し、真のパラメータで計算された評価値に揺らぎ（分散）を与えた値をユーザの評価値と仮定して実験を行う（多目的ナップサック問題と観光スケジュール作成問題：パラメータ数 3）。

実験 2： 実際に被験者に対して実験を行う（多目的巡回セールスマン問題：パラメータ数 3）。

実験 3： 実験 1 と同じ方法でパラメータ数が多い場合を想定して実験を行う（多目的ナップサック問題：パラメータ数 12）。

実験 4： 評価項目数が多い場合に実際に被験者に対して実験を行う（観光スケジュール作成問題：パラメータ数 22）。

#### 4.2.1 実験 1

真のパラメータ  $\mathbf{w}^*$  を設定し、ユーザ評価値は  $\frac{100}{M} \mathbf{w}^{*T} \mathbf{x}(s)$  に誤差が加わった値とする。ここで  $M = \max_s \mathbf{w}^{*T} \mathbf{x}(s)$  とする。ユーザの評価値の揺らぎを表す誤差は  $\varepsilon \sim N(0, \sigma^2)$  と仮定すれば、ユーザの評価値は正規分布  $N(\frac{100}{M} \mathbf{w}^{*T} \mathbf{x}(s), \sigma^2)$  により与えられる。  $\sigma$  の値を変えることで、本研究のモデルがどの程度の揺らぎまで対応できるのかを検証する。重みパラメータの推定精度として以下の「パラメータ比の二乗誤差」  $J$  と「最適解の評価値誤差」  $P$  を定義する。

・パラメータ比の二乗誤差

$$J = \sum_{k=1}^{m-1} \left( \frac{w_k^*}{w_m^*} - \frac{\hat{w}_k}{\hat{w}_m} \right)^2 \quad (13)$$

・最適解の評価値誤差（割合）

$$P = \frac{f_{\mathbf{w}^*}(s^*) - f_{\mathbf{w}^*}(\hat{s})}{f_{\mathbf{w}^*}(s^*)} \quad (14)$$

$s^*$ ：最適解 ( $s^* = \arg \max_s f_{\mathbf{w}^*}(s)$ )

ただし、 $\hat{w}_k$  は提案法によって推定された  $k$  番目の重みパラメータ、 $\hat{s}$  は提案法で得られた解である。

対象問題は多目的ナップサック問題と観光スケジュール作成問題とする。重みパラメータ数は三つとし、ユーザが評価する解の数  $N$  を 5, 10, 15 と変化させた。またユーザの評価の揺らぎに相当する  $\sigma$  は 2, 4, ..., 16 と 2 きざみで変化させた。各問題の真のパラメータを表 2 に示す。また、GA の各パラメータと遺伝的操作の方法をそれぞれ表 3 と表 4 に示す。

表 2 真のパラメータ  
Table 2 True parameters.

真のパラメータ	
多目的ナップサック問題	$w_1 = 0.6, w_2 = 0.2, w_3 = 2.2$
観光スケジュール (1)	$\alpha = 2.5, \beta = 0.15, \gamma = 0.35$
観光スケジュール (2)	$\alpha = 1.8, \beta = 0.4, \gamma = 0.8$

表 3 GA のパラメータ  
Table 3 Parameters of GA.

個体数	世代交代数	交叉率	突然変異率
300	300	0.9	0.1

表 4 GA の遺伝的操作の方法  
Table 4 Inherited operation of GA.

交叉方法	選択方法	エリート保存個数
一点交叉	トーナメント選択	1 個

#### 4.2.2 実験 2

実験 1 ではユーザ評価値は正規分布による揺らぎを想定して、正規乱数で与えた。ここでは、実際に被験者（15 人）に解を評価してもらうモニタ実験により、提案手法の実問題への有効性を検証する。ユーザによる解評価終了後にユーザにとって最適重みパラメータを推定し、そのパラメータのもとでの最適解を提示する。その後、推定パラメータ付近から手動でパラメータを変化させ、提示された解よりもユーザにとってよい解が見つかるかどうかを確認し、以下のアンケート項目に回答するものとする。ユーザが評価する解の数を増やすことで提示された準最適解がユーザにとっての最適解に近づいていくかを検証する。なお、対象問題は多目的巡回セールスマン問題とし、重みパラメータの数は 3 とする。

・解評価終了後に提示された解はあなたにとっての最適解にどの程度近かったですか？

- (A) 最適解とほとんど一致していた。
- (B) 最適解とは一致しなかったが、その誤差は小さかった。
- (C) 最適解とはそれなりの誤差が感じられた。
- (D) 最適解と全く違う解が提示された。

#### 4.2.3 実験 3

実験 1 と実験 2 はいずれも重みパラメータの数が三つ程度の問題での実験であった。ここではパラメータ数が多い場合を想定して、3.2 で述べたリッジ回帰を用いた手法の有効性を検証する。実験 1 と同様に、真のパラメータをもとに計算した値に誤差が加わった値をユーザの評価値と仮定する。ここではパラメータの数を 12 とした多目的ナップサック問題を対象とし、

表 5 真のパラメーター一覧  
Table 5 True parameter table.

$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$
0.3	1.1	1.9	1.4	1.8	1.2
$w_7$	$w_8$	$w_9$	$w_{10}$	$w_{11}$	$w_{12}$
1.0	0.4	1.3	2.0	1.2	0.9

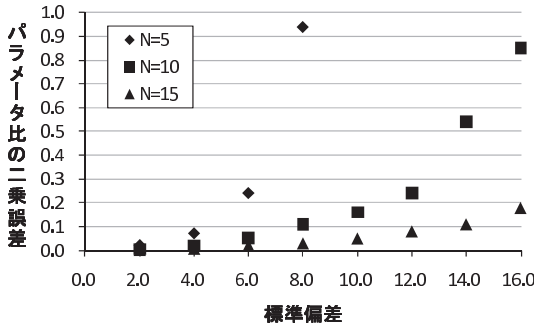


図 3 パラメータ比の二乗誤差 (多目的ナップサック問題)  
Fig. 3 Square error of parameter ratio (multiobjective knapsack problem).

ユーザが評価する解の数を  $N = 5, 10$  と変化させる。また、実験 1 と同様にユーザの評価の揺らぎに相当する  $\sigma$  を  $2, 4, \dots, 16$  と変化させた。以下の表 5 に真のパラメーター一覧を示す。

#### 4.2.4 実験 4

実験 3 では重みパラメータ数が多い場合の提案手法の有効性をシミュレーションにより検証した。実験 4 ではパラメータ数が多い場合に実際に被験者 (ユーザ) に解を評価してもらうことを考える。対象とする問題は観光スケジュール作成問題とする。観光スケジュール作成問題ではパラメータ数が三つのように思えるが、式 (11) の第 1 項の各観光地の重要度を重みパラメータとみなせばパラメータの数が多い場合と考えることができる。実験で使うパラメータ数は 22 とし、パラメーター一覧を以下に示す。

- 観光地の重要度 (観光地数: 20)
- 移動時間
- 到着時刻との差

#### 4.3 実験結果と考察

##### 4.3.1 実験 1 の結果と考察

実験 1 の結果 (100 回の平均) を図 3 と図 4 に示す。ここで図 3 はパラメータ比の二乗誤差と標準偏差  $\sigma$  の関係であり、図 4 は最適解の評価値誤差と  $\sigma$  の関係である。

図 3 及び図 4 から  $\sigma$  の値を大きくすればするほど、

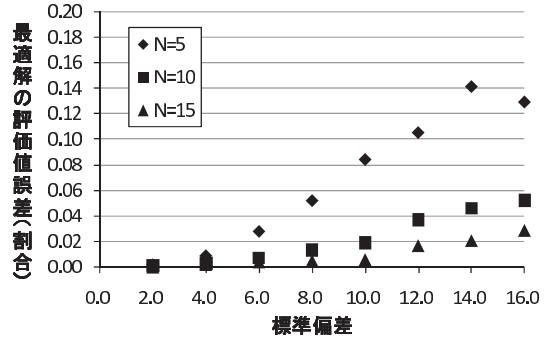


図 4 最適解の評価値誤差 (多目的ナップサック問題)  
Fig. 4 Error ratio of evaluation value on the optimum individual (multiobjective knapsack problem).

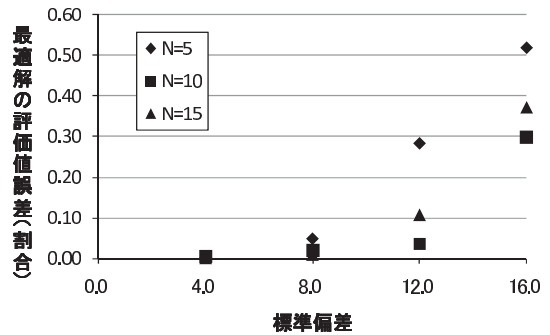


図 5 最適解の評価値誤差 (観光スケジュール作成問題 (1))  
Fig. 5 Error ratio of evaluation value on the optimum individual (travel scheduling problem (1)).

「パラメータ比の二乗誤差」, 「最適解の評価値誤差」ともに値が大きくなり精度が悪くなるのが分かる。その一方で、ユーザが評価する解の数  $N$  が増えれば誤差は小さくなり、 $\sigma$  の値が大きいつきでも  $N$  を増やせば、小さな誤差に抑えることができる。人間の評価値が式 (1) の線形モデルで付与されると仮定したら、 $\sigma$  が小さいとき最適解を推定できる可能性が高いことが分かる。

また観光スケジュール作成問題を使って同様の実験を行った。その結果 (50 回の平均) を図 5 及び図 6 に示す。

図 5 から分かるように、観光スケジュール作成問題を用いた場合も、多目的ナップサック問題を用いた場合と同様の結果を得ることができた。一方、図 6 はユーザの真のパラメータの値を「移動時間」と「到着時刻」を極端に重視したユーザを想定して設定した。



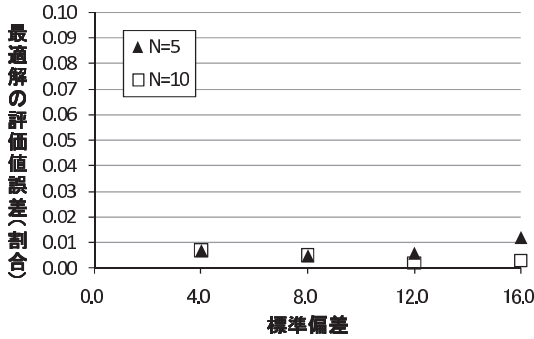


図 6 最適解の評価値誤差 (観光スケジュール作成問題 (2))

Fig. 6 Error ratio of evaluation value on the optimum individual (travel scheduling problem (2)).

表 6 ユーザ実験 (15 人) のアンケート結果 (数字は人数)

Table 6 Result of the questionnaire for a user test (15 respondents).

	(A)	(B)	(C)	(D)
N = 5	4	5	4	2
N = 10	7	4	4	0
N = 15	10	4	1	0
N = 20	11	3	1	0

その結果、ユーザが評価する解の数が少ないときや、 $\sigma$  が大きいときでもかなり少ない誤差で最適解を推定していることが分かる。これは、ユーザが重要視する要素が極端に偏っている場合は、ユーザから得た情報が少ないときや、ユーザの評価が揺らいだ場合でもユーザにとっての最適解を推定しやすいと考えられる。

#### 4.3.2 実験 2 の結果と考察

実験 2 のユーザ実験のアンケート結果を表 6 に示す。表 6 から分かるように、ユーザが評価する解の数  $N$  が増えれば、ユーザによる評価が良くなっていることが分かる。実験 2 の結果より人間の評価の揺らぎに対しても、 $N$  を増やすことにより対応できることを示した。このことから提案手法の重み推定方法は適切であり、ユーザ評価が高い解を推定可能であることが分かる。従来の iGA では評価回数が 100 回以上になることも多いが、表 6 では 20 回程度の評価でユーザにとっての最適解をシステムが見つげ出しているケースが多い。本研究が想定している状況では従来の iGA よりも提案手法を用いた方がよいことが分かる。

#### 4.3.3 実験 3 の結果と考察

最適解の評価値誤差と  $\sigma$  の関係 (100 回の平均) を図 7 に示す。図 7 から実験 1 のパラメータ数が三つの

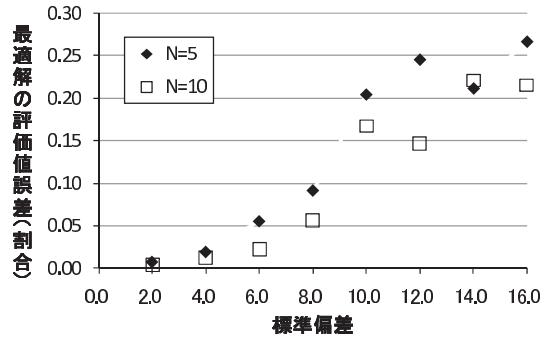


図 7 最適解の評価値誤差 (多目的ナップサック問題：パラメータ数 12)

Fig. 7 Error ratio of evaluation value on the optimum individual (multiobjective knapsack problem with 12 parameters).

表 7 ユーザ実験 (10 人) のアンケート結果 (数字は人数)

Table 7 Result of the questionnaire for a user test (10 respondents).

	(A)	(B)	(C)	(D)
N = 10	0	5	5	0
N = 15	3	5	2	0
N = 20	6	4	0	0

ときと同様の結果が得られることが分かった。ただし、パラメータ数が増えたため、ユーザの評価値の揺らぎに相当する  $\sigma$  を 10 程度に増やすと最適解に近い解を求めることが困難となっている。また、実験 1 の結果と比べると相対的に最適解の評価値誤差が大きくなっていることが分かる。また、 $\sigma$  が比較的小さい状況では実験 1 と同様にユーザが評価する解の数  $N$  を増やせば最適解の評価値誤差は小さくなることが分かった。

#### 4.3.4 実験 4 の結果と考察

実験 4 では実験 2 と類似した条件でパラメータ数が多い場合を想定してユーザ (人数：10 人) に対してモニタ実験を行った。その結果を表 7 に示す (質問項目は実験 2 と同じである)。実験 2 と同様に  $N$  を増やすにつれてユーザの評価はよくなった。しかし、実験 2 と比べるとユーザの評価は下がった。これはパラメータ数が多くなったことで、パラメータ数が少ないときと比べて推定精度が下がり、これが GA の解探索の結果にも結び付いたためと考えられる。

### 5. 全体的な考察

本章では提案手法、評価実験の結果を踏まえて本研究の全体的な考察を行う。

#### (1) アルゴリズム改良の効果についての考察



表 8 実験条件  
Table 8 Experimental conditions.

品物数	制約コスト	標準偏差	混ぜる割合
30	500	$\sigma = 10$	1 個/個体数 100

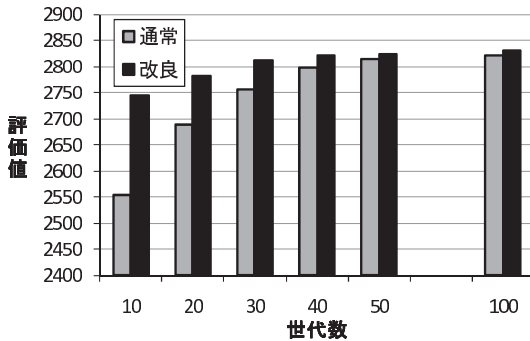


図 8 世代数と評価値の関係 (個体数: 100)

Fig. 8 Relation between number of generations and evaluation value (100 individuals).

提案手法において GA を 2 回以上繰り返すとき、前回得られた準最適解を初期個体群に混ぜることで GA の収束が早まることを考えた。本章ではこのアルゴリズム改良の効果について考察する。実際にどの程度 GA の収束が早まったかを確認する実験を行った。ユーザ評価値は実験 1 と同様に自動で与え、真のパラメータは表 2 に示した値 (多目的ナップサック問題) を用いた。この実験の実験条件を表 8 に、実験結果を図 8 に示す。

図 8 より、世代数が多いときは、アルゴリズムを改良した効果は見られなかったが、世代数が少ないときは改良した効果が大きいことが分かった。世代数が多いと計算コストが大きいので、この方法を用いることで世代数が少ない計算コストがかからない条件でも準最適解が得られることが分かる。

#### (2) 提案手法の有用性についての考察

評価実験の結果から重みパラメータの数が少ない場合、多い場合ともにユーザが評価する解の数を増やすことで、ユーザにとっての最適解を得ることができる可能性が高いことが分かった。このことから、ユーザが解を直観的に評価する際に、問題によってはユーザがある程度線形モデルに基づいて評価していることが分かる。ユーザが評価する解の数は 20 回程度であり、iGA と比べれば圧倒的に少ない。よって本研究の提案手法は実問題に適用できると考えられる。

しかし、線形モデルの評価関数に加えて、線形モデ

ル以外の要素がユーザの評価に影響すると、誤差が大きくなりユーザにとっての最適解を見つけることは難しくなると考えられる。よって、線形モデル以外の要素の影響が大きい場合は iGA を適用することが望ましいと考えられる。

本研究の提案手法は評価要素間にスケールの違いが存在するとき特に有用であると考えられる。ここで、スケールの違いとは評価要素の単位の違いのことである。例えば「運賃 (円)」と「時間 (分)」は単位が違う。仮に単位が同じであれば、ユーザは重みを自分で設定しやすいが、単位が違うことで多くの試行錯誤による重みパラメータの調整が必要となると考えられる。実際に、関連研究で紹介した本研究の適用対象となる問題の多くも評価要素間にスケールの違いが存在している。

本研究ではユーザによって重みが異なる場合を想定しているが、問題の規模や制約条件が変わることで重みは変化していく。関連研究で紹介した三つの論文 [9]~[11] はいずれもこのケースにあてはまる。

#### (3) アプリケーションのユーザビリティ

提案手法をアプリケーションとして実装する際に考慮すべき点として、ユーザへの解の提示方法が挙げられる。提案手法ではユーザの解に対する評価により、最適解が決まってしまうため、ユーザが評価しやすいように解を提示する必要がある。具体的には図や写真を用いて他の解との違いを明確にすることや 1 回に提示する解の数は少なめにすることなどが挙げられる。

また、ユーザ実験においては解に対して点数を付けることが負担であるという意見もあった。よって、一対比較による解の評価も検討する必要があると考えられる。

#### (4) 提案手法の発展性についての考察

本研究の提案手法は GA を用いた最適化問題を対象としているが、他のメタヒューリスティックにも適用可能であると考えられる。例えば GA の派生形である GP [20] などにはほとんど改良することなく適用できると思われる。

一方、提案手法は GA を複数回繰り返すため、SA [21] のような探索時間が非常に長いメタヒューリスティックに適用することは望ましくないと考えられる。

また、最適化問題の中には解候補の数がメタヒューリスティックを使うほど多くないため、全探索を使用する状況も存在する。そのようなケースでも重みがユー

ザによって異なる状況は存在し、GAを全探索に置き換えることで提案手法は適用できると考えられる。

## 6. む す び

本研究では、複数の評価項目の重みパラメータがユーザによって異なる状況を想定し、重み推定と解探索を対話的に行う方法を提案した。具体的には評価関数が評価要素の線形和で表される場合を想定し、ユーザが解に対して付与した評価値をもとに、最小二乗法により重みパラメータを逐次推定しながら、GAによる解探索を行う方法を提案した。評価実験により、ユーザの評価は線形モデルで表現できるケースが存在し、パラメータが少ないときはユーザにとって最適解を出力できる可能性が高いことが分かった。

今後の課題としては、重みパラメータ数が多い場合の手法の改善が挙げられる。また、今回はユーザは解に対して得点を付けたが、実際にユーザ実験の際に得点付けを負担に感じるユーザも見受けられた。今後は、一対比較を用いた場合を想定した提案手法も検討したい。

謝辞 本論文を執筆するにあたり、早稲田大学東基衛名誉教授と早稲田大学平澤茂一名誉教授に多くの有益な助言を頂きました。心より感謝致します。また、本研究の内容について活発に議論して頂いた早稲田大学創造理工学部経営システム工学科後藤研究室の皆様

## 文 献

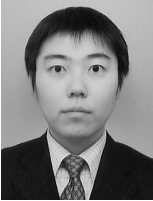
- [1] D.E. Goldberg, Genetic Algorithms in search, Optimization, and Machine Learning, Addison-Wesley Professional, 1989.
- [2] 伊藤冬子, 廣安知之, 三木光範, 横内久猛, “対話型遺伝的アルゴリズムにおける嗜好の多峰性に対応可能な個体生成方法,” 人工知能学会論文誌, vol.24, no.1, pp.127-135, 2009.
- [3] H. Takagi, “Interactive evolutionary computation fusion of the capabilities of EC optimization and human evaluation,” Proc. IEEE, vol.89, no.9, pp.1275-1296, Sept. 2001.
- [4] 渡辺芳信, 吉川大弘, 古橋 武, “一対比較評価に基づく対話型遺伝的アルゴリズム,” 情処学研報, 2007-MPS-63, pp.69-72, March 2007.
- [5] D.W. Gong, J. Yuan, and X.P. Ma, “Interactive genetic algorithm with large population size,” Proc. 2008 IEEE Congress on Evolutionary Computation, pp.887-894, HongKong, Sept. 2008.
- [6] 丸山敦史, 柴田直樹, 村田住洋, 安本慶一, 伊藤 実, “P-Tour: 観光スケジュール作成支援とスケジュールに沿った経路案内を行うパーソナルナビゲーションシステム,” 情処学論, vol.45, no.12, pp.2678-2687, Dec. 2004.
- [7] C.M. Fonseca and P.J. Fleming, “Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization,” Proc. Fifth International Conference on Genetic Algorithms, pp.416-423, July 1993.
- [8] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-2,” IEEE Trans. Evol. Comput., vol.6, no.2, pp.182-197, Aug. 2002.
- [9] 山本章夫, 橋本 寛, “原子燃料の炉心内配置の最適化—GA, SAの適用事例,” オペレーションズ・リサーチ, vol.44, no.11, pp.599-603, Nov. 1999.
- [10] 森本章雄, 三宅宏典, 高之瀬文明, 大北正昭, 大木 誠, “遺伝的アルゴリズムによる看護師勤務表の最適化,” FIT2005 (第4回情報科学技術フォーラム), A024, pp.57-58, 2005.
- [11] 松島佑介, “沿道騒音対策の最適選択手法に対する遺伝的アルゴリズムの適用に関する研究,” 九州大学大学院人間環境学研究院都市・建築学部門 藤本研究室卒業論文, 2003.
- [12] 村田幸弘, 安藤竜弥, 阿部充志, “ペナルティ法による目的関数生成における重み付け自動化,” 情処学論, vol.48, no.SIG19(TOM19), pp.99-106, Dec. 2007.
- [13] 大筆 豊, “オセロプログラムの評価関数の改善について,” 鳥取環境大学紀要, pp.65-79, Feb. 2004.
- [14] 金子知適, 田中哲郎, 山口和紀, 川合 慧, “駒の関係を利用した将棋の評価関数の学習,” 情処学論, vol.48, no.11, pp.3438-3445, Nov. 2007.
- [15] 中山弘隆, “多目的計画に対する満足化トレードオフ法の提案,” 計測自動制御学会論文集, vol.20, no.1, pp.29-35, Jan. 1987.
- [16] A.E. Hoerl and R.W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” Technometrics, vol.12, no.3, pp.55-67, Feb. 1970.
- [17] 橋達 弘, 村田住洋, 柴田直樹, 安本慶一, 伊藤 実, “FPGA上への遺伝的アルゴリズムの柔軟な実装手法の提案,” 信学論 (D), vol.J89-D, no.6, pp.1182-1191, June 2006.
- [18] 山村政幸, 小野貴久, 小林重信, “形質の遺伝を重視した遺伝的アルゴリズムに基づく巡回セールスマン問題の解法,” 人工知能学会論文誌, vol.7, no.6, pp.1049-1059, 1992.
- [19] 橋達 弘, 村田住洋, 柴田直樹, 安本慶一, 伊藤 実, “ハードウェア化のための多目的 GA アーキテクチャの提案,” 情処学論, vol.49, no.1, pp.381-392, Jan. 2008.
- [20] 伊庭斉志, “遺伝的プログラミングと進化論的な学習,” 人工知能学会誌, vol.9, no.4, pp.512-517, July 1994.
- [21] 喜多 一, “シミュレーテッドアニーリング,” 日本ファジィ学会誌, vol.9, no.6, pp.870-875, Dec. 1997.

(平成 23 年 4 月 11 日受付)



石川英太郎

平 21 早大・理工・経営システム卒. 平 23 同大・創造理工学研究科・修士課程了. 遺伝的アルゴリズムなど最適化の研究に興味をもつ.



石田 崇 (正員)

平 11 早大・理工・経営システム卒. 平 13 同大・理工学研究科・修士課程了. 平 14 同大・理工学研究科・博士後期課程入学. 平 17~19 同大・理工・経営システム工学科助手. 平 20 同大・メディアネットワークセンター助教, 現在に至る. 博士(工学). 情報源符号化, 統計的学習理論, ベイズ統計応用などの研究に従事. IEEE 会員.



後藤 正幸 (正員)

平 4 武蔵工大・工・経営卒. 平 6 同大大学院修士課程了. 平 6 早大・理工学研究科・博士後期課程入学. 平 8~11 同大・理工・経営システム工学科助手. 平 12~14 東大・工・環境海洋工学専攻・助手. 平 14~20 武蔵工大・環境情報・助教授. 平 20~23 早大・理工・経営システム工学科・准教授. 平 23 より同教授. 情報理論, 統計的学習理論, ベイズ統計などの情報数理を基盤研究とし, マーケティング分析やリコメンデーションシステムの理論評価, 経営情報や経営工学分野の応用研究にも従事. 博士(工学). IEEE, 人工知能学会, OR 学会, 日本経営工学会, 経営情報学会等各会員.