ファジィクラスタリングを用いたシェアリングを含む遺伝的アルゴリズムによる 0-1 計画法

丹　羽　啓　一

1．はじめに

現実に存在する意思決定状況をモデル化すると0-1計画問題を含む整数計画問題に帰着することは少なくない。それらの例を挙げるとアルバイトのシフトを決めるための人員配置問題，公共施設の配置問題，荷物を輸送するための配送経路を最適化するような問題などがある[23]。本論文では，0-1計画問題の中から特に多次元0-1ナップサック問題を取り上げ，その解法を構築する。

Holland[4] によって提案された遺伝的アルゴリズム（Genetic Algorithm: GA）は，自然界のシステムの適応過程を説明し，生物の進化のメカニズムを模倣する人工モデルとして様々な分野の問題に適用され，それらの解法としてその有効性が検証されている[21]。最適化の分野においても離散変数を扱う整数計画問題や非線形関数を扱う非線形計画問題など，既存の解法では解決することが困難な問題に適用され，それらの問題の解法として現在も発展を続けている[11, 15, 16, 21]。

一方，データ分析のための手法の一つであるクラスタ分析は，経済学，経営学，工学，情報学といった様々な分野における諸問題を解決するために活用され，その有用性が認識されている[3, 5, 12, 13, 19]。上述した GA とクラスタ分析の手法を組み合わせた研究を分類すると次の二つに大別できる。一つは，GA を利用してクラスタ分析を行う研究であり[6, 12]，もう一つは GA のアルゴリズムの一部にクラスタ分析を取り込む研究である。後者については，主に最適化問題を解くための手法の構築に関する研究であり，

GA の再生オペレータの中にクラスタ分析の手法を取り込み，GA の性能を向上させることで良質な近似最適解を導出しようとしている[8, 14, 20, 24]。

本論文は，多次元ナップサック0-1計画問題に焦点をあて，GA を用いた計算手法を提案する。具体的に述べると著者[17] によって提案された手法には，個体の再生アルゴリズムに K-medoids 法[7] を用いたシェアリングが含まれていた。このシェアリングの手法には問題点が存在しており，それを改善するために Krishnapuram らによって提案された Fuzzy c-medoids 法[10] を採用したシェアリングを開発し，それを再生オペレータに組み込んだ GA による0-1計画法を構築する。その後，提案した手法の有効性を検証するために数値実験を行い，解の精度ならびに計算時間の両面において提案する解法と既存の解法の性能比較を行う。

本論文の構成は次のようになっている。第 1 章においては，本研究の背景について説明した後，次章以降の各章の概要を示す。第 2 章では，GA を用いた0-1計画法の研究について概観した後，クラスタ分析を用いたシェアリングについて簡単に解説し，このアルゴリズムを取り入れた GA による0-1計画法について説明する。第 3 章では，過去に提案されたクラスタ分析の手法を含むシェアリングの問題点について述べた後，その問題を解決するために Fuzzy c-medoids 法を用いたシェアリングを提案し，数値実験によってその手法の有効性を検証する。第 5 章では，本論文の結論を要約するとともに今後の研究課題について述べる。

2．0-1計画問題と遺伝的アルゴリズム

2.1. 0-1計画問題と遺伝的アルゴリズムを用いた0-1計画法

本論文では，多次元0-1ナップサック問題を取り上げ，その解法を構築する。多次元0-1ナップサック問題を定式化すると次のようになる[9]。

$$\left.\begin{array}{ll} \text{maximize} & \boldsymbol{cx} \\ \text{subject to} & \boldsymbol{Ax} \le \boldsymbol{b} \\ & x_i \in \{0,1\}, i = 1,\ldots,n \end{array}\right\} \qquad (2.1)$$

ここで，$\boldsymbol{x}=(x_1,\ldots,x_n)^T$ は意思決定者の決定変数ベクトルである。ただし，上付添字 T は転置を表す。$\boldsymbol{c}=(c_1,\ldots,c_n)$ は目的関数の係数ベクトルを表し，A は制約式の $m\times n$ 係数行列を表す。また，$\boldsymbol{b}=(b_1,\ldots,b_m)^T$ は制約式の右辺定数ベクトルとする。

本論文では，簡単化のため，A, b, c の各成分は正であると仮定する。

問題（2.1）の解法として以前から遺伝的アルゴリズム（Genetic Algorithm: 以下必要に応じて GA と略記する）が提案されている。具体的には，単純遺伝的アルゴリズム（Simple Genetic Algorithm: SGA）[2] や坂和ら[22] によって提案された二重構造文字列を個体表現に採用した GA（Genetic Algorithm with Double String: GADS）を用いた解法などが挙げられる。坂和らは，二重構造文字列とこの文字列に適用するためのデコーディングアルゴリズムを用いることによって問題（2.1）の制約式を満たす実行可能解のみを生成し，良好な近似最適解を短時間で導出することに成功した。

このような0–1計画問題に対する GA を用いた解法の基本的なアルゴリズムを示すと以下のようになる。

GA による0–1計画法のアルゴリズム

手順 1　GA の世代数を $t:=0$ とし，初期個体をランダムに N 個生成する。

手順 2　各個体の適合度の値を求める。

手順 3　現在の GA の世代数（反復回数）が予め設定された打ち切り世代数 M に到達していれば，アルゴリズムを終了し，それまでに得られた個体の中で適合度の値が最も大きな個体を最適な個体 $\boldsymbol{x}$ とみなす。そうでなければ，手順 2 で求めた適合度の値に応じて再生を行う。

手順 4　再生を適用した後の個体に対して，予め設定された交叉確率 p_c で交叉を行い，さらに，突然変異確率 p_m によって突然変異を実施し，新しい個体を生成する。その後，$t:=t+1$ として手順 2 に戻る。

2.2. クラスタ分析の手法を用いたシェアリングを導入した遺伝的アルゴリズム

確率的に最適解を探索するアルゴリズムは大別すると二つに分類できる。一つは，山登法，Simulated Annealing（SA）や Tabu Search（TS）といった解の候補を一つだけ用意し，確率的にその候補を変化させながら最適解を導出する一点探索[18] のアルゴリズムである。もう一つは，本研究で扱う GA や粒子群最適化（Particle Swarm Optimization: PSO）のように複数の解候補（個体）を同時に変化させながら最適解を探索していく多点探索のアルゴリズムである。多点探索のアルゴリズムに分類される GA を用いて最適化問題を解く際，ある世代における個体群中の個体が GA を適用する問題の解空間においてどのように分布しているのかということと GA の性能の良し悪しが密接に関係していると言える。また，GA のような探索アルゴリズムを用いて良好な解を得るためには，一般的に探索初期において広大な解空間の中から最適解の存在する解空間を探索するような大域的探索能力が必要であり，さらに探索末期において絞り込んだ解空間の中から最適解を見つけ出す局所的探索能力も必要になる。この二つの能力をバランスよく組み合わせるために Goldberg ら[2] はシェアリングというアルゴリズムを提案した。このシェアリングを導入した GA は，個体群中の個体の過度な収束を防ぎ，その多様性を保つことによって，良好な近似解を導出できることがわかった。シェアリングの基本的な考え方は多くの個体が集中している部分において個体の評価値に比較的小さな重みをかけ，孤立している個体の評価値に対して相対的に大きな重みをかけることにある。このことによって，評価値の分布の相対的な均一化を図ることが可能になっている。

シェアリング関数 $sh(\cdot)$ は次の条件を満たすものとする[22]。

1） $0 \le sh(d) \le 1, \forall d \in [0, \infty)$

2） $sh(0) = 1$

3） $\lim_{d \to \infty} sh(d) = 0$

ここで，$d_{ij} = d\left(\boldsymbol{s}_i, \boldsymbol{s}_j\right)$ はコード化された文字列 $\boldsymbol{s}_i$ と $\boldsymbol{s}_j$ 間の距離とする。この $sh(\cdot)$ について Goldberg らは式（2.2）に示すシェアリング関数の導入を推奨している。

$$sh(d) = \begin{cases} 1 - \left(\dfrac{d}{\sigma_{share}}\right)^{\alpha} & d < \sigma_{share} \\ 0 & \text{その他} \end{cases} \tag{2.2}$$

ここで，σ_{share} と α は定数である。

GA のアルゴリズムにシェアリングを適用する際には，再生オペレータの一部として採用することになるが，具体的に個体の適合度を修正するためには次のような式（2.3），（2.4）を用いる。ここで，個体 $\boldsymbol{s}_i$ の適合度関数を $f\left(\boldsymbol{s}_i\right)$，個体群サイズを N とする。

$$f\left(\boldsymbol{s}_i\right) = \frac{f\left(\boldsymbol{s}_i\right)}{m_i} \tag{2.3}$$

$$m_i = \sum_{j=1}^{N} sh\left(d\left(\boldsymbol{s}_i, \boldsymbol{s}_j\right)\right) \tag{2.4}$$

シェアリングを導入すると適合度の値を修正する際，各世代において個体群中のすべての個体間の距離を計算し，その距離の度合いに応じて重みづけをする必要があり，多くの計算時間を要することになる。Yin ら[24] はクラスタリング手法[1] として K-means 法を採用したシェアリングによる再生オペレータを提案し，多目的最適化問題にそれを導入した GA を適用した。その結果，シェアリングを用いた場合に比べて計算時間を削減することが可能であり，近似最適解の精度も良好であることを示した。また，Niwa ら[14] は Yin らの方法を改良した GA を 2 レベル0–1計画問題に適用し，その効果を検証した。その際，個体間の距離を測るためにハミング距離[21] を用いている。さらに，丹羽[17] は，データユニットを分類する際に用いるクラスタの代表値を各クラスタのデータユニットとする K-medoids 法を採用したシェアリングを提案している。

ここでは，まず，丹羽によって提案された K-medoids 法をクラスタリングに採用したシェアリングについて概観するために K-medoids 法のアルゴリズム[1)] について述べる。

K-medoids 法の過程

手順 1　初期値 k を設定し，k 個の初期 medoid $\boldsymbol{V}=\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k\}$ をデータユニット $\boldsymbol{X}=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$ から選択する。ただし，$\boldsymbol{v}_c, c=1,\ldots,k$ は各クラスタの medoid を表す。

手順 2　残りの $N-k$ 個のデータユニットについて，k 個の medoid の中から最近傍となる medoid を見つけ，その medoid の属するクラスタにデータユニットを割当てる。その後，データユニットを割り当てたクラスタの medoid を式（2.5）に基づいて更新する。

$$\boldsymbol{v}_l=\arg\min_{i\in C_l}\sum_{j=1}^{n_l}d\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right) \tag{2.5}$$

ただし，$C_c, c=1,\ldots,k$ はクラスタを，k はクラスタの数を表す。また，$n_c, c=1,\ldots,k$ は c 番目のクラスタのデータユニットの数を表し，l は新しくデータユニットを割り当てたクラスタの番号とする。

この操作を medoid に割り当てるデータユニットがなくなるまで繰り返す。

次に，K-medoids 法を採用したシェアリングについて詳述する。丹羽のアルゴリズムでは，個体群中の個体の収束度合いを K-medoids 法を用いて測り，その度合いに応じて適合度の修正を行っている。上述した K-medoids 法のアルゴリズムをシェアリングに採用するためには，手順 1 における初期の medoid の選定方法とクラスタリングの結果を適合度にどのように反映させるのかということを考慮する必要があった。まず，前者については，初期個体群の中の個体の適合度を計算し，その値の良いものから順に k 個選定するという Yin ら[24)] によって提案された方法を採用した。

後者に関しては，以下の通りである。medoid と各個体の距離の計算には，Niwa らの手法と同様にハミング距離を採用し，さらに，適合度を修正する際には次の式（2.6）を用いた。

$$f_i' := \frac{f_i}{m_i}, i = 1,\ldots,N \quad \text{with} \quad m_i = n_c - n_c * \left(\frac{d_{ic}^m}{2d_{max}^m} \right)^{\alpha}, \boldsymbol{x}_i \in C_c \tag{2.6}$$

ここで，個体 i の適合度の値を f_i，修正後の適合度の値を f_i'，個体群サイズを N とする。また，α は定数であり，d_{ic}^m は個体 i とそれが含まれるクラスタの medoid $\boldsymbol{v}_c$ の距離を表す。

この式は Yin らによって提案された方法を改良したものである。ここで，d_{max}^m は，次式を用いて計算する。

$$d_{max}^m = \max_{c=1,\ldots,k} \max_{j=1,\ldots,n_c} d\left(\boldsymbol{v}_c, \boldsymbol{x}_j\right) \tag{2.7}$$

この値は，クラスタごとに medoid とそのクラスタに属するデータユニットの距離の最大値を計算し，その最大値の中から最も大きな値を選出したものであり，クラスタの中心から近傍（そのクラスタの内部）として判断することのできる最大の距離を表す。

K-medoids 法を用いたシェアリングを採用した GA のアルゴリズムを示すと次のようになる。

K-medoids 法を用いたシェアリングを採用した GA による0–1計画法のアルゴリズム

手順 1　GA の世代数を $t := 0$ とし，初期個体をランダムに N 個生成する。

手順 2　各個体をデコードし，問題（2.1）の目的関数値を計算する。2.2 節で述べた K-medoids 法を適用することによって，個体の収束度合いを評価し，式（2.6），式（2.7）を用いて各個体の適合度の値を求める。

手順 3　現在の GA の世代数（反復回数）が予め設定された打ち切り世代数 M に到達していれば，アルゴリズムを終了し，それまでに得られた個体の中で適合度の値が最も大きな個体を最適な個体 $\boldsymbol{x}$ と

みなす。そうでなければ，手順 2 で求めた適合度の値に応じて再生を行う。

手順 4　再生を適用した後の個体に対して，予め設定された交叉確率 p_c で交叉を行い，さらに，突然変異確率 p_m によって突然変異を実施し，新しい個体を生成する。その後，$t:=t+1$ として手順 2 に戻る。

3．ファジィクラスタリングを用いたシェアリングの改良

3.1. Fuzzy c-medoids 法に基づいた個体の再生方法

丹羽[17] は K-medoids 法[7] を用いたシェアリングを提案し，それを再生オペレータに採用した GA の有効性を示した。K-medoids 法を用いることによって，以前，Niwa らによって提案された K-means 法を用いたシェアリング手法[14] における問題点の一部を解決している。しかしながら，予想した通り K-medoids 法をシェアリングに用いた再生オペレータは，K-means 法をシェアリングに用いた再生オペレータよりも計算時間を要しており，この計算時間を削減するという必要性が生じている。本論文では，K-medoids 法を用いたシェアリングを採用した GA の性能を維持もしくは改善しつつ，計算時間の短縮を目的として Fuzzy c-mediods 法[10] をシェアリングに用いた再生オペレータを設計し，それを組込んだ GA による0-1 計画法を提案する。計算時間の削減には，Fuzzy c-medoids 法のアルゴリズムにおいて設定する必要のあるパラメータを調整することによって対応することとする。

Fuzzy c-medoids 法を用いたシェアリングについて述べる前にまず Fuzzy c-medoids 法について概観する。

$X=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$ を n 個のデータユニットの集合とし，データユニット $\boldsymbol{x}_i$ とデータユニット $\boldsymbol{x}_j$ の非類似度を $d(\boldsymbol{x}_i,\boldsymbol{x}_j)$ とする。また，$\boldsymbol{V}=\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k\},\boldsymbol{v}_k\in X$ は濃度 k の X の部分集合を表す。このとき Fuzzy c-medoids 法では，以下の最小化問題が定義される[3]。

$$\begin{aligned} \min \quad & L_{fcm_{idl}} = \sum_{c=1}^{k}\sum_{j=1}^{n} u_{cj}^{m} d\left(\boldsymbol{x}_j, \boldsymbol{v}_c\right) \\ \text{s.t.} \quad & u_{cj} \in [0,1], c = 1,\ldots,k; j = 1,\ldots,n \\ & \sum_{c=1}^{k} u_{cj} = 1, j = 1,\ldots,n \end{aligned} \tag{3.1}$$

ここで，u_{cj} はクラスタ c に対する $\boldsymbol{x}_j$ の帰属の度合いを表すメンバシップ値である。

Krishnapuram ら[10] によると u_{cj} には様々な定義が存在することが紹介されている。しかしながら本論文においては，最も単純な Fuzzy c-medoids 法をシェアリングに適用した場合の GA の効果をみることを目的としているため，u_{cj} の定義をする際に最も単純な式（3.2）を用いることにする。

$$u_{cj} = \frac{\left(\dfrac{1}{d\left(\boldsymbol{x}_j, \boldsymbol{v}_c\right)}\right)^{\frac{1}{m-1}}}{\displaystyle\sum_{c=1}^{k}\left(\frac{1}{d\left(\boldsymbol{x}_j, \boldsymbol{v}_c\right)}\right)^{\frac{1}{m-1}}}, c = 1,\ldots,k; j = 1,\ldots,n \tag{3.2}$$

ここで $m \in [1,\infty)$ は重みを表し，帰属度合いのあいまいさを調整するために用いられる。

次に Fuzzy c-medoids 法のアルゴリズムについて述べる。アルゴリズムの手順を示すと以下の通りになる。

Fuzzy c-medoids 法の過程

手順 1　クラスタの数 k を設定し，反復回数を表す $iter := 0$ とする。k 個の初期 medoid $\boldsymbol{V} = \{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k\}$ をデータユニットから選択する。ただし，$\boldsymbol{v}_c, c = 1,\ldots,k$ は各クラスタの medoid を表す。

手順 2　メンバシップの値 $u_{cj}, c = 1,\ldots,k, j = 1,\ldots,n$ を式（3.2）によって導出する。

手順 3　$\boldsymbol{V}^{old} = \boldsymbol{V}$ として現在の medoid を保存する。その後，式（3.3）を用いて medoid の候補となる q 番目のデータユニット $\boldsymbol{x}_q$ を選定し，新しい medoid $\boldsymbol{v}_c, c = 1,\ldots,k$ に代入する。$iter = iter + 1$とする。

$$q = \arg\min_{1 \le l \le n} \sum_{j=1}^{n} u_{cj}^{m} \, d\left(\boldsymbol{x}_j, \boldsymbol{x}_l\right) \tag{3.3}$$

手順 4　もし，$\boldsymbol{V}^{old} = \boldsymbol{V}$ であるか $iter = MAX\_ITER$ であれば，手順を終了する。そうでなければ手順 2 に戻る。ここで，MAX_ITER は反復回数の最大値を表す。

上述したアルゴリズムの手順 1 を見ると初期 medoid を導出する必要がある。Krishnapuram ら[10] は三つの初期 medoid の選択方法を示しており，それらを示すと以下のようになる。

一つ目の方法は，すべての初期 medoid をランダムに選択する方法である。本論文ではこの方法を Type 1 と呼ぶことにする。二つ目の方法は，全てのデータユニットの最も中心に位置するデータユニットを最初の medoid として選定し，その後，選定された medoid の数がクラスタの数 c になるまで，現時点において選定されている medoid から最も相違するデータユニットを選択し，medoid に選定していくというものである。この方法を Type 2 と呼び，アルゴリズムを以下に示す。

初期 medoid の選択方法（Type 2）

手順 1　medoid の数を $k > 1$とする。式（3.4）を用いて初期 medoid の候補となる q 番目のデータユニット $\boldsymbol{x}_q$ を選定し，$\boldsymbol{v}_1 = \boldsymbol{x}_q$ とする。

$$q = \arg\min_{1 \le j \le n} \sum_{l=1}^{n} d\left(\boldsymbol{x}_j, \boldsymbol{x}_l\right) \tag{3.4}$$

$\boldsymbol{V} = \{\boldsymbol{v}_1\}$, $iter := 1$ とする。

手順 2　$iter = iter + 1$とする。式（3.5）を用いて medoid の候補となる q 番目のデータユニット $\boldsymbol{x}_q$ を選定し，$\boldsymbol{v}_{iter} = \boldsymbol{x}_q$ とする。

$$q = \underset{1\le j\le n;\boldsymbol{x}_j\notin\boldsymbol{V}}{\arg\max}\ \min_{1\le c\le|V|} d\left(\boldsymbol{v}_c,\boldsymbol{x}_j\right) \tag{3.5}$$

$\boldsymbol{V}=\boldsymbol{V}\cup\{\boldsymbol{v}_{iter}\}$ とする。

手順 3　$iter=k$ であれば，手順を終了する。そうでなければ手順 2 に戻る。

三つ目の方法は，最初の medoid だけをランダムに選択し，その後は Type 2 のアルゴリズムを用いて c–1個の medoid を選定するというものである。この方法を Type 3 と呼ぶことにする。Krishnapuram らが提案した初期 medoid の選定方法はこの三つになるが，本論文では，Yin ら[24] が提案した初期のデータユニットの選択方法である個体群に属する個体の適合度を計算し，その値の良いものから順に c 個選定するという方法を Type 4として採用することにする。

本論文では，Fuzzy c-medoids 法を用いて，個体群中の個体の収束度合いを測り，その度合いに応じて適合度の修正を行う。まず，上述した Fuzzy c-medoids 法の過程に従って各クラスタの medoid を選定する。その際，medoid と各個体の非類似度（距離）の計算には，Niwa らの手法と同様にハミング距離を採用する。その後，Fuzzy c-medoids 法を適用した結果として得られる各クラスタへの各個体の帰属度を表すメンバシップの値 $u_{cj}, c=1,\ldots,k, j=1,\ldots,n$ を用いて各個体をクラスタに割り当てる。その際，現在の medoid V に選定されていないデータユニット（個体 $\boldsymbol{x}_j$）のメンバシップの値 u_{cj} と式（3.6）を用いてデータユニットをクラスタ q に割り当てる。

$$q = \underset{1\le c\le k}{\arg\max}(u_{c1},u_{c2},\ldots,u_{cn}) \tag{3.6}$$

さらに，適合度を修正する際には次の数式を用いる。

$$f_i' := \frac{f_i}{m_i}, i=1,\ldots,N \quad \text{with} \quad m_i = n_c - n_c * \left(\frac{d_{ic}^m}{2d_{max}^m}\right)^{\alpha}, \boldsymbol{x}_i \in C_c \tag{3.7}$$

この式は丹羽によって提案された式（2.6）と同様のものである。また，

d_{max}^{m} については式（2.7）と同様のものを用いて計算する。

$$d_{max}^{m} = \max_{c=1,\ldots,k} \max_{j=1,\ldots,n_c} d\left(\boldsymbol{v}_c, \boldsymbol{x}_j\right) \tag{3.8}$$

また，d_{ic}^{m} は個体 i とそれが含まれるクラスタの medoid $\boldsymbol{v}_c$ の距離を表す。最後に提案する手法の手順を以下に示す。

Fuzzy c-medoids 法を用いたシェアリングを採用した GA による0–1計画法のアルゴリズム

手順 1　GA の世代数を $t:=0$ とし，初期個体をランダムに N 個生成する。

手順 2　各個体をデコードし，問題（2.1）の目的関数値を計算する。3.1節で述べた Fuzzy c-medoids 法を適用することによって，個体の収束度合いを評価し，式（3.7），（3.8）を用いて各個体の適合度の値を求める。

手順 3　現在の GA の世代数（反復回数）が予め設定された打ち切り世代数 M に到達していれば，アルゴリズムを終了し，それまでに得られた個体の中で適合度の値が最も大きな個体を最適な個体 $\boldsymbol{x}$ とみなす。そうでなければ，手順 2 で求めた適合度の値に応じて再生を行う。

手順 4　再生を適用した後の個体に対して，予め設定された交叉確率 p_c で交叉を行い，さらに，突然変異確率 p_m によって突然変異を実施し，新しい個体を生成する。その後，$t:=t+1$ として手順 2 に戻る。

3.2. 数値実験

提案した手法の有効性を検証するために数値実験を行う。数値実験において用いる問題は，丹羽[14] と同じ問題とし，具体的には表3.1に示すような規模で制約式が 5 本の0–1計画問題とする。

ここで，0–1計画問題の A, c の各成分は［10, 99］の整数の乱数を発生

表3.1 数値実験で用いる問題の規模と制約の強さ

問題	変数数	制約の強さ
A	30	Ⅰ（70%）
		Ⅱ（90%）
B	60	Ⅰ（70%）
		Ⅱ（90%）
C	90	Ⅰ（70%）
		Ⅱ（90%）
D	120	Ⅰ（70%）
		Ⅱ（90%）
E	150	Ⅰ（70%）
		Ⅱ（90%）

させることで値を設定し，b の成分 $b_i, i=1,\cdots,m$ については，

$$b_i = r_i \times \left(\sum_{j=1}^{n} a_{ij} \right), i = 1,\ldots,m$$

にしたがって値を設定する。ただし，$a_{ij}, i=1,\cdots,m, j=1,\cdots,n$ は行列 A の ij 成分を表す。ここで，r_i は制約の強さの度合いを表しており，制約が強くなるほどパーセントの値が小さくなる。それゆえ，制約の強さによって r_i に用いる乱数の値の範囲が異なり，問題Ⅰでは［0.65, 0.75］を採用し，問題Ⅱでは，［0.85, 0.95］の区間を用いる。その後，r_i を発生させて b_i の値を算出した後，小数点以下を四捨五入することで整数値に丸め，その値を制約右辺値に設定する。また，問題 A から E については係数値の異なる問題を 5 種類ずつ用意する。

数値実験には，次の三つの GA を用いる。具体的には，提案手法である Fuzzy c-medoids 法をシェアリングに用いた GA（以下，FMed-GA と略記する），比較対象となる SGA ならびに K-medoids 法をシェアリングに用いた GA（以下，Med-GA と略記する）である。その際，共通する GA のパラメータとして個体群サイズ100，交叉率0.9，突然変異率0.02を採用する。

打ち切り世代数は，問題 A, B, C, D については1000世代，問題 E については2000世代に設定した。また，Med-GA と FMed-GA に共通して使用するパラメータとして，シェアリング用のパラメータ α と初期クラスタの数があり，それぞれを0.25と 5 とした。最後に FMed-GA において用いるパラメータとして，各個体の各クラスタへの帰属の度合いを表すメンバシップ値を計算する際に用いられる重み m と Fuzzy c-medods 法の手順の最大反復回数 *MAX_ITER* があり，それぞれを 2 と20に設定した。

次に実験環境について述べる。数値実験には，1.40 GHz の CPU を搭載し，Windows XP を OS としたコンピュータを用いた。また，プログラム言語として Visual C＋＋ 6.0を採用した。シミュレーションの試行回数は，各問題に対して10回ずつとする。

数値実験を実施し，得られた結果の中から代表的な例を示しながら考察する。提案した解法と既存の解法の性能を比較するために表3.2から表3.6には，各問題に対して各解法を適用して得られた目的関数の値を示し，表3.7から表3.11には，それぞれの問題を解くために要した計算時間を示す。表3.2から表3.6における最大値は，10回の試行において得られた最も良い目的関数値を表し，最小値は最も悪い目的関数値を表す。平均値と分散については10回の試行結果から算出したものである。また，最適解到達回数は，10回の試行において厳密な最適解に何度到達したかを表した数値である。この厳密な最適解を求めるために分枝限定法[23] のアルゴリズムを実装した混合整数計画問題に対するフリーソルバーの lp_solve[25] を用いる。表3.7から表3.11における最大値は，10回の試行において得られた最も悪い計算時間を表し，逆に最小値は最も良い計算時間を表す。平均値と分散については，目的間数値の場合と同じく10回の試行結果から算出したものになる。

最後に表中の FMed-GA に付加された表記について説明すると，括弧内の Type 1 から Type 4 の表記は，初期 medoid の選定方法を表している。

表3.2の結果を見ると制約の強さが I の問題においては，SGA を除くほ

表3.2 問題 A に GA を適用した際の実験結果（解の精度）

（制約の強さ I）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	1397	1399	1399	1399	1399	1399
最小値	1348	1399	1399	1399	1397	1399
平均値	1380.4	1399.0	1399.0	1399.0	1398.6	1399.0
分散	309.8	0.0	0.0	0.0	0.6	0.0
最適解到達回数	0	10	10	10	8	10
（制約の強さ II）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	1570	1570	1570	1570	1570	1570
最小値	1536	1570	1570	1570	1570	1570
平均値	1564.0	1570.0	1570.0	1570.0	1570.0	1570.0
分散	102.4	0.0	0.0	0.0	0.0	0.0
最適解到達回数	6	10	10	10	10	10

かの 4 つの手法によって最適解を得ることができている。しかしながら FMed-GA（Type 3）の計算結果を見ると最適解への到達回数が他の手法に比べて若干劣っていることが分かる。平均値と分散の値を見ると SGA と FMed-GA（Type 3）以外の手法においては性能に大差がないと言える。制約の強さが II の問題の実験結果を見ると SGA 以外の 4 つの手法では，10回の試行のすべてにおいて最適解を得ることができていることがわかる。このことから提案した FMed-GA は良好な結果を得ることができたと言える。

表3.3の計算結果より，Med-GA は，他のすべての手法の実験結果よりも優れていることがわかる。Med-GA に関しては問題の規模が倍になっているにも関わらず，それほど性能が劣化していないと言える。それに比べると SGA ほど劣化はしていないものの FMed-GA の 4 つの手法は解の精度の点でやや劣化していると言える。ただ，どちらの制約の問題に対しても FMed-GA（Type 1）から FMed-GA（Type 4）を用いて最適解を導出することはできていると言える。SGA に関しては解の精度の面において急

表3.3　問題 B に GA を適用した際の実験結果（解の精度）

（制約の強さⅠ）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	2796	2857	2857	2857	2857	2857
最小値	2689	2842	2814	2816	2821	2833
平均値	2738.1	2853.3	2838.6	2831.0	2837.8	2846.7
分散	998.3	33.0	193.4	200.0	136.8	47.8
最適解到達回数	0	7	2	1	1	2
（制約の強さⅡ）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	3180	3194	3194	3194	3194	3194
最小値	3034	3194	3170	3162	3161	3164
平均値	3129.6	3194.0	3184.6	3182.3	3181.2	3180.1
分散	1449.0	0.0	52.6	98.0	80.6	51.1
最適解到達回数	0	10	2	3	1	1

速に劣化していると言える。

表3.4の結果を見ると表3.3に示した結果と同様に Med-GA の解の精度が最も優れていることがわかる。特に制約の強さⅡの結果を見ると最適解への到達回数が 7 割に達していることから問題 A に比べると問題の規模が 3 倍になっているにも関わらず，それほど性能が劣化していないと言える。しかしながら制約の強さⅠの結果を見ると Med-GA についても若干の性能の劣化が見受けられる。FMed-GA に関しては，最適解の導出という点では Med-GA に劣るものの，制約の強さⅠの問題において最大値と平均値を見ると FMed-GA（Type 1）と FMed-GA（Type 2）の結果はそれほど悪くないと言える。また制約の強さⅡの問題において同じ指標を見ると FMed-GA（Type 2）と FMed-GA（Type 4）の結果について同様のことが言える。

表3.5の結果についても表3.4の結果とほぼ同様の結果になっていると言える。最適解の到達回数を見ると Med-GA が最も優れていると言える。

表3.4 問題 C に GA を適用した際の実験結果（解の精度）

（制約の強さ I）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	4079	4269	4232	4234	4219	4221
最小値	3959	4229	4156	4162	4135	4173
平均値	4038.5	4252.3	4202.0	4197.5	4177.3	4199.0
分散	1145.1	132.8	477.2	476.9	473.2	214.2
最適解到達回数	0	1	0	0	0	0
（制約の強さ II）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	4677	4734	4701	4716	4704	4714
最小値	4508	4728	4653	4672	4657	4669
平均値	4582.7	4732.2	4682.7	4691.6	4684.0	4693.1
分散	2113.0	7.6	201.8	193.6	287.4	261.1
最適解到達回数	0	7	0	0	0	0

表3.5 問題 D に GA を適用した際の実験結果（解の精度）

（制約の強さ I）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	5633	5834	5756	5737	5730	5741
最小値	5373	5781	5598	5607	5591	5603
平均値	5472.5	5806.3	5678.5	5659.9	5668.0	5663.5
分散	8193.7	179.2	1975.1	2196.9	2029.4	1670.1
最適解到達回数	0	1	0	0	0	0
（制約の強さ II）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	6407	6537	6493	6465	6446	6481
最小値	6150	6529	6366	6346	6329	6322
平均値	6313.8	6533.9	6430.4	6411.9	6391.9	6403.8
分散	4986.0	6.5	1720.8	1321.5	1021.9	2187.8
最適解到達回数	0	3	0	0	0	0

しかしながら制約の強さⅡの結果を見ると到達回数が若干減っていることがわかる。制約の強さⅠの結果についてはほぼ同等である。FMed-GA については，制約の強さⅠとⅡの双方ともに最大値と平均値の結果を見るとFMed-GA（Type 1）が優れていることがわかる。ただし，制約の強さⅡの問題の最小値で見るとFMed-GA（Type 2）とFMed-GA（Type 4）よりは劣っていると言える。

表3.6 の結果を見ると制約の強さⅠと制約の強さⅡの双方の結果においてどの手法を用いても最適解を得ることができていないことがわかる。Med-GA については問題 D までの結果に比して性能が劣化していることがわかる。それに比べるとFMed-GA の性能はそれほど劣化していないと言える。制約の強さⅠの問題の最大値を見るとFMed-GA（Type 1）の性能が最も良く，その他の指標まで含めて見るとFMed-GA（Type 4）の性能が優れていることがわかる。制約の強さⅡの問題においても最大値ではFMed-GA（Type 1）の性能が優れていると言える。また，その他の指標

表3.6　問題 E に GA を適用した際の実験結果（解の精度）

（制約の強さⅠ）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	6601	6683	6804	6777	6801	6802
最小値	6341	6505	6583	6627	6643	6635
平均値	6491.1	6573.4	6718.9	6722.1	6720.2	6727.1
分散	5250.7	3754.0	5311.3	1549.9	2457.4	2644.9
最適解到達回数	0	0	0	0	0	0
（制約の強さⅡ）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	7563	7608	7704	7683	7684	7683
最小値	7280	7350	7538	7548	7527	7568
平均値	7446.1	7517.3	7627.5	7625.6	7630.2	7646.9
分散	6200.7	5055.8	3278.3	1270.4	2173.0	1210.1
最適解到達回数	0	0	0	0	0	0

を見ると総合的には FMed-GA（Type 4）が安定して良い結果を得ることができていると言える。細部の結果の差はあるものの FMed-GA の 4 つの手法は比較的良好な結果を得ることができていると言える。

次に計算時間について考察する。表3.7から表3.11の結果を見ると SGA については，問題の規模が大きくなるにつれて少しずつ計算時間は延びるものの非常に短い時間で計算結果を得ることができていると言える。

表3.7の結果を見ると制約の強さⅠとⅡの問題において 4 つの FMed-GA の計算時間に大差はなく，約30秒程度で一つの問題を解くことが出来ていると言える。それに比べて Med-GA については，FMed-GA と比較して遅く，約1.8倍程度の計算時間を要していることがわかる。

表3.8の結果を見ると FMed-GA の中で差が生じていることがわかる。特に FMed-GA（Type 3）は，平均値の指標で見た場合に最も計算速度の速い FMed-GA（Type 2）と比較して制約の強さⅠの問題で約 2 倍，制約の強さⅡの問題で約 3 倍の差があることがわかる。それ以外に特筆すべき点を挙げると制約の強さⅡの問題の結果における FMed-GA（Type 4）の最大値が非常に大きいことである。ただし，平均値を見ると最小値に近い値

表3.7　問題 A に GA を適用した際の実験結果（計算時間）

（制約の強さⅠ）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	0.86	54.94	32.36	32.06	31.66	30.28
最小値	0.78	50.27	30.22	29.86	29.88	29.22
平均値	0.81	52.71	30.85	30.50	30.37	29.49
分散	0.00	2.93	0.52	0.36	0.32	0.10
（制約の強さⅡ）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	0.83	62.03	30.99	31.14	30.17	32.97
最小値	0.81	56.38	30.06	29.66	29.64	28.97
平均値	0.81	58.93	30.34	30.30	29.78	29.64
分散	0.00	2.58	0.07	0.22	0.03	1.26

表3.8　問題 B に GA を適用した際の実験結果（計算時間）

（制約の強さⅠ）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	1.17	114.64	64.94	54.44	109.70	56.56
最小値	1.13	101.80	54.17	54.03	105.73	53.59
平均値	1.14	108.45	55.92	54.18	107.80	54.67
分散	0.00	10.38	9.14	0.02	1.66	0.84
（制約の強さⅡ）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	1.19	129.64	56.80	54.67	165.58	108.44
最小値	1.17	116.74	54.11	53.31	159.31	71.27
平均値	1.18	123.80	55.32	53.85	161.92	77.49
分散	0.00	13.51	0.91	0.12	3.11	127.22

になっており，分散の値もそれほど大きくないことから乱数の揺らぎによるものと言える。Med-GA の計算時間を見ると制約の強さⅠと制約の強さⅡの双方の結果ともに最速の FMed-GA（Type 2）の約 2 倍の計算時間を要することがわかる。

表3.9より，FMed-GA（Type 3）と FMed-GA（Type 2）の差はさらに大きくなっており，制約の強さⅠの問題では，約 3 倍，Ⅱの問題では約 2 倍である。FMed-GA（Type 3）を除く 3 つの手法の計算時間はほぼ同等であると言える。Med-GA については，表3.8の結果と同様に制約の強さⅠと制約の強さⅡの双方の結果ともに FMed-GA（Type 2）の約 2 倍の計算時間を要していることがわかる。

表3.10の結果を見ると表3.8と表3.9の結果の傾向と異なる傾向が見受けられる。これまでの 3 つの問題に対する実験結果とは異なり，制約の強さⅡの問題において FMed-GA（Type 2）の計算時間がかなり増えていることである。逆に FMed-GA（Type 3）の計算時間については，FMed-GA（Type 1）と FMed-GA（Type 4）とほぼ同等の結果を得ていることがわかる。Med-GA の計算時間については，これまでと同じく FMed-GA の最も

表3.9 問題 C に GA を適用した際の実験結果（計算時間）

（制約の強さ I）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	1.50	169.39	82.58	81.13	238.97	81.06
最小値	1.42	164.36	80.42	77.97	232.92	78.38
平均値	1.46	167.01	81.66	78.89	236.12	79.10
分散	0.00	3.14	0.61	1.32	3.48	0.57
（制約の強さ II）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	1.70	201.09	82.00	79.91	234.44	81.17
最小値	1.45	188.19	78.89	77.84	155.61	78.55
平均値	1.52	194.67	80.51	78.48	164.59	79.27
分散	0.01	20.53	1.09	0.38	542.99	0.48

表3.10 問題 D に GA を適用した際の実験結果（計算時間）

（制約の強さ I）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	1.81	241.27	109.81	210.36	116.88	105.61
最小値	1.73	226.47	103.95	102.41	103.53	102.80
平均値	1.77	236.13	105.77	114.53	109.07	103.83
分散	0.00	23.45	2.99	1021.68	21.62	0.67
（制約の強さ II）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	1.97	289.83	106.50	314.52	106.03	104.19
最小値	1.81	259.98	103.08	307.89	102.88	103.20
平均値	1.86	274.31	103.99	310.68	103.71	103.74
分散	0.00	80.29	0.98	3.43	0.77	0.13

計算時間の短いものと比較すると約 2 倍の計算時間を要していると言える。

表3.11の結果を見ると制約の強さ I の問題においては FMed-GA の 4 つの手法の差はそれほど見受けられず，Med-GA との差は平均値で見ると約

表3.11　問題 E に GA を適用した際の実験結果（計算時間）

（制約の強さ I）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	4.36	645.22	263.05	260.27	259.84	260.95
最小値	4.13	590.55	260.67	257.38	256.52	257.19
平均値	4.18	620.18	261.47	258.73	257.39	258.29
分散	0.01	230.84	0.49	1.24	0.84	1.71
（制約の強さ II）手法	SGA	Med-GA	FMed-GA（Type1）	FMed-GA（Type2）	FMed-GA（Type3）	FMed-GA（Type4）
最大値	4.41	787.50	274.58	520.78	299.89	302.25
最小値	4.27	704.20	256.58	514.38	255.92	256.83
平均値	4.31	734.05	260.53	516.66	261.00	267.19
分散	0.00	600.47	28.14	3.77	168.23	161.27

2.2倍程度である。制約の強さ II で見ると FMed-GA（Type 2）は他の FMed-GA に比べて約 2 倍の計算時間を要していると言える。また FMed-GA（Type 2）を除く 3 つの手法と Med-GA を比較した場合，Med-GA の方が約2.8倍の計算時間を要していると言える。

次に制約の強さ I と II の問題を解く際に各手法が要した平均計算時間をグラフにしたものをそれぞれ図3.1と図3.2に示す。

図3.1と図3.2の双方ともに問題 E の計算結果については，他の問題に比べて打ち切り世代数が 2 倍になっていることに注意しておく必要がある。

図3.1を見ると FMed-GA（Type 3）を除く他の手法については多少の差はあるもののほぼ線形で計算時間が増えていると言える。FMed-GA（Type 3）については，問題 C の計算時間が非常にかかっているものの，規模の大きな問題 D や問題 E の結果を見ると他の FMed-GA の手法とほぼ同等の結果になっていることがわかる。

図3.2についても FMed-GA（Type 2）を除くとほぼ同様の傾向が見られる。FMed-GA（Type 2）については，制約の強い問題については，問題の規模が大きくなると急激に計算時間が増える傾向にあると言える。

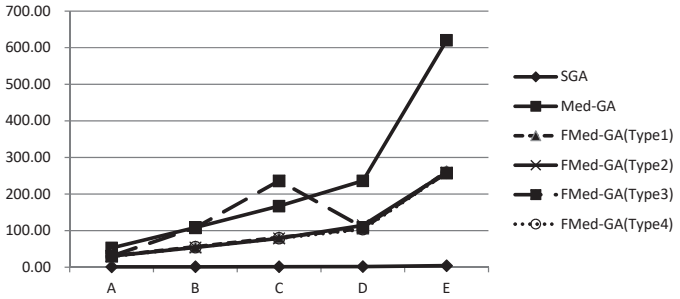

図3.1 制約の強さⅠの問題を解く際に要する各手法の平均計算時間の推移

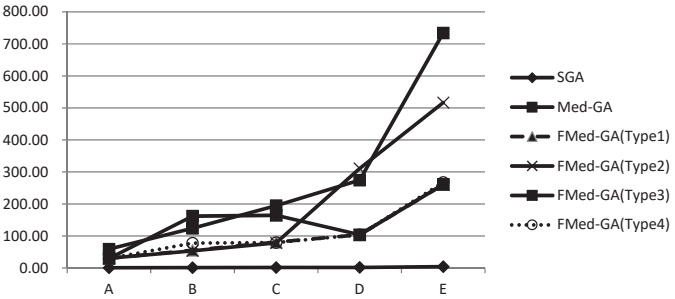

図3.2 制約の強さⅡの問題を解く際に要する各手法の平均計算時間の推移

最後に数値実験によって得られた結果を解の精度と計算時間の双方の視点から総合的に考察する。提案した FMed-GA については，解の精度と計算時間を考慮すると全体的には FMed-GA（Type 1）の性能が優れていると言える。また，規模の大きな問題においては，FMed-GA（Type 4）も比較的良好な結果を得ることが出来ている。Med-GA については，問題の規模が大きくなると急速に解の精度が悪くなっていることと FMed-GA よりも計算時間が長くなっていることが挙げられる。計算時間に関しては一般的に K-medoids 法と Fuzzy c-medoids 法を比較すると前者の方が短い時間でデータユニットをクラスタに分割することができる。しかしながら，FMed-GA のアルゴリズムに含まれている Fuzzy c-medoids 法のアルゴリ

ズムにおいて *MAX_ITER* の値を適切に設定することができたため，計算時間を削減することができたと考えられる。SGA については，非常に単純なアルゴリズムゆえ，計算時間は他の 5 つの手法に比べて非常に短いものの，解の精度においてはかなり劣る結果になったと言える。

4．おわりに

本論文では，ファジィクラスタリングを用いたシェアリングを含む GA による0–1計画法を提案した。具体的には，丹羽によって提案された K-medoids 法をシェアリングに用いた GA の問題点を改良するために Fuzzy c-medoids 法を用いたシェアリングを提案し，それを組み込んだ GA の有効性を検証するために数値実験を実施し，解の精度ならびに計算時間の両面からそれを示した。

今後の研究課題としては，解の精度を改善するためにパラメータのチューニングを実施することが挙げられる。また，Fuzzy c-medoids 法の medoid の選定において，設定する必要のある *MAX_ITER* の値が適切であったことから提案した手法は，K-medoids 法をシェアリングに用いた GA よりも計算時間の面で優れていた。しかしながら，SGA と比較すると多大な計算時間を要していることからさらなる計算時間の短縮が必要であると考えられる。そのためには，すでに開発されているロバストな Fuzzy c-medoids 法[10] を採用したシェアリングを開発することなどが考えられる。また，それ以外にも過去に Niwa らによって提案された K-means 法を用いたシェアリングと同様に世代によって変化する個体群に応じて自動的に適切なクラスタ数を導出できるようなアルゴリズムを開発することなども挙げられる。

参　考　文　献

1）M. R. Anderberg: *Cluster Analysis for Applications*, Academic press,（1975）.
2）D. E. Goldberg: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley,（1989）.

3）坪和，本田，市橋，野津：“Fuzzy c-Medoids 法の応用による線形クラスタリングと関係データからの部分空間学習，”知能と情報，Vol. 21, No. 1, pp. 151-159（2009）.
4）J. H. Holland: *Adaptation in Natural and Artificial Systems*, University of Michigan Press（1975）, MIT Press（1992）.
5）本多，S. Tong，野津：“k-Medoids 法の概念に基づくファジィ共クラスタリング，”第 5 回横幹連合コンファレンス論文集，pp. 162-164（2013）.
6）加藤，小沢：“遺伝的アルゴリズムを用いた非階層クラスタリング，”情報処理学会論文誌，Vol. 37, pp. 1950-1959（1996）.
7）L. Kaufman, and P. J. Rousseeuw: *Finding Groups in Data: An Introduction to Cluster Analysis,* Wiley（1990）.
8）Hee-Su Kim, and Sung-Bae Cho: “An Efficient Genetic Algorithm with Less Fitness Evaluation by Clustering,”in *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, Vol. 2, pp. 887-894（2001）.
9）今野，鈴木：整数計画法と組合せ最適化，日科技連（1982）.
10）R. Krishnapuram, A. Joshi, O. Nasraoui, and L. Yi: “Low-Complexity Fuzzy Relational Clustering Algorithms for Web Mining,”*IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 4, pp. 595-607（2001）.
11）Z. Michalewicz: *Genetic Algorithms + Data Structures = Evolution Programs*, Third, revised and extended edition, Springer-Verlag（1996）.
12）宮本：クラスター分析入門，森北出版（1999）.
13）宮本：“ファジィクラスタリングの有用性について，”知能と情報，Vol. 21, No. 6, pp. 1008-1017（2009）.
14）K. Niwa, I. Nishizaki, and M. Sakawa: “Two-Level 0-1 Programming Using Genetic Algorithms and a Sharing Scheme Based on Cluster Analysis,”in *International MultiConference of Engineers and Computer Scientists 2008 Proceedings*, Vol. 2, pp. 1931-1936（2008）.
15）K. Niwa, T. Hayashida, and M. Sakawa: “Computational Methods for Two-Level 0-1 Programming Problems through Distributed Genetic Algorithms,”*Journal of Telecommunications and Information Technology*, Vol. 2, pp. 78-87（2010）.
16）K. Niwa, T. Hayashida, and M. Sakawa: “Multiobjective Two-Level 0-1 Programming through Distributed Genetic Algorithms,”in *Proceedings of IEEE International Conference on Fuzzy Systems*, pp. 1767-1773,（2011）.
17）丹羽：“K-medoids 法を用いたシェアリングを含む遺伝的アルゴリズムによる 0-1計画法，”広島経済大学経済研究論集，Vol. 37, No. 1, pp. 25-45（2014）.
18）S. M. Sait, H. Youssef（著），白石（訳）：組合せ最適化アルゴリズムの最新手法，丸善（2002）.

19）齋藤，宿毛：関連性データの解析法，共立出版（2006）.
20）坂倉，谷口，星野，亀井：“ファジィクラスタリングを用いた遺伝的アルゴリズムに関する基礎研究，”第21回ファジィシステムシンポジウム講演論文集，pp. 171–174（2005）.
21）坂和，田中：遺伝的アルゴリズム，朝倉書店（1995）.
22）M. Sakawa: *Large Scale Interactive Fuzzy Multiobjective Programming*, Springer-Verlag（2000）.
23）坂和：離散システムの最適化，森北出版（2000）.
24）X. Yin and N. Germay: “A Fast Genetic Algorithm with Sharing Scheme Using Cluster Analysis Methods in Multimodal Function Optimization,” in *Artificial Neural Nets and Genetic Algorithms, Proceedings of the International Conference in Innsbruck, Austria, 1993*, Springer-Verlag, pp. 450–457（1993）.
25）lp_solve, https://groups.yahoo.com/neo/groups/lp_solve/.