

高次元非線形スパース最適化問題に対する多様な近似解の導出
 ——食材・配合量最適化問題の解法——

佐藤 雅子^{†a)} 佐藤 仁樹^{††}

Derivation of Various Approximate Solutions for High-Dimensional Nonlinear Sparse Optimization Problems —— Optimization for Food Ingredients and Their Quantities ——

Masako SATOH^{†a)} and Hideki SATOH^{††}

あらまし 高次元非線形スパース最適化問題に対する複数の多様な局所近似解を、遺伝的アルゴリズムを用いて導出した。まず、状態変数の番号を遺伝子とした染色体を定義する。次に、高次元非線形スパース最適化問題を染色体で与えられた状態変数のみを変数とする問題に縮小する。縮小された非線形最適化問題の評価関数と染色体間の距離の和を染色体の適応度として遺伝的アルゴリズムにより染色体を改良し、縮小された非線形最適化問題を解くことにより、高次元非線形スパース最適化問題に対する複数の多様な局所近似解を導出する。この解法を食材及び食材配合量の最適化問題に適用し、食材及び食材配合量を栄養素バランスの目標値に対して最適化した。

キーワード 遺伝的アルゴリズム、スパース最適化、高次元、食材、レシピ

1. ま え が き

患者にとって最適な栄養バランスを保ち、かつ患者を飽きさせないバラエティに富んだ食事を提供する栄養食事療法は、疾病治療の手段として重要である。特に、腎臓食、糖尿食、潰瘍食、肥満食、腎炎食のように疾患別に患者に提供する特別食は、疾病治療に効果的である。また、特別な食事療法を必要としない患者に提供される一般食においても、通常の食事摂取基準 [1] に基づいて栄養バランスが考慮されているべきである。しかし、現実には、栄養管理が行われている病院でも、一般食において幾つかの栄養素が不足しているという報告がなされている [2]。また、調理やレシピ考案の手続きが複雑かつ非効率になるため、患者一

人一人に対して個別に最適な食事を提供することは困難である [3]。

食事の献立を決めるためには、まず、数千種類の食材から、栄養バランスを考慮して 10 種類程度の食材を選定し、主菜、副菜などのレシピが決められる。次に、主食、主菜、副菜を組み合わせて、1 回の食事の献立とする。最後に、1 日ないし数日の食事により数十種類の栄養素の必要量を満たし、かつ同じ料理の繰り返しにならないように献立を作成する。これらの作業は、一部の栄養計算を除き、栄養士による手作業で行われている。そのため、必要とされる数十種類の栄養のバランスを満たし、かつ患者を飽きさせない多様なレシピ及び献立を作成することは困難である。

制約条件付き非線形最適化問題を解くための非線形計画法は多くの分野で用いられており、様々な効率的な解法が提案されてきた [4]。前述のレシピ作成問題を非線形最適化問題として定式化した場合、数千種類の食材の配合量を状態変数とした制約付き非線形最適化問題となる。そのため、計算時間やメモリの制約から、この問題に通常の非線形計画法を適用することは困難である。

[†] (株) インテリジェントセンサーテクノロジー, 厚木市
 Intelligent Sensor Technology, Inc., 5-1-1 Onna, Atsugi-shi,
 243-0032 Japan

^{††} 公立はこだて未来大学システム情報科学部, 函館市
 School of Systems Information Science, Future University
 Hakodate, 116-2 Kamedanakano-cho, Hakodate-shi, 041-
 8655 Japan

a) E-mail: satohqueen@gmail.com

辛いことに、一食分の料理に使われる食材は高々10種類程度である。すなわち、数千種類の食材の配合量のほとんどが0となる。解の要素のほとんどが0になるような問題はスパース最適化問題と呼ばれ、関数近似、パターン認識、予測、信号処理等、多くの分野に存在する [5], [6]。この問題に対して、説明変数と目的変数の関係が線形の場合には、その線形性を利用してスパースな解を求める手法 [7] を利用できる。また、遺伝的アルゴリズム (GA) [8], [9] と粒子群最適化 (PSO) を組み合わせたスパース最適化手法 [10] は、説明変数ベクトルのうち、非0の値を解とする要素の位置に対応する遺伝子を1で、0を解とする要素の位置に対応する遺伝子を0で表し、GAの個体の染色体を作る。その後、GAにより染色体を改良し、PSOにより非0の値を解とする説明変数を最適化する。そのため、広い範囲のスパース最適化問題を解くことが可能である。

しかし、これらの方法は、非0の値を解とする説明変数の数を指定できないため、実際に一食分の料理に使われる食材数が非現実的に多くなる可能性がある。また、定式化に依存する限界のため、数千種類の食材に対応できない。更に、前述のレシピ作成問題では、所望の栄養バランス、コスト、及び味の条件を満たす多くのレシピや献立が必要となるにもかかわらず、文献 [5]~[7], [10] の手法では、一度に多くの解を得ることができない。そのため、これらの手法を、前述のレシピ作成問題に適用することは困難である。

そこで、我々は、GAを用いたスパース最適化手法 [11] を、数千種類の食材の中から一食分の料理に用いられる十種類程度の食材の組合せとその食材配合量を計算するレシピ作成問題に適用した。その際、染色体間の距離を適応度に導入することにより、多様な料理を提供するための食材の組合せを複数導出できた。また、適応度に食材間の相関を導入することにより、栄養バランスやコストのみならず食材の組合せ (味) を考慮できた。

本論文では、まず、2. で、GAを用いて多様な局所近似解を求めるためのスパース最適化手法について述べる。次に、3. で、2. で提案した手法を用いて、食材及び食材配合量を最適化する方法を述べる。4. では、提案手法の性能評価結果を示す。最後に、5. で、本論文の提案とその評価をまとめる。

2. 遺伝的アルゴリズムに基づくスパース最適化

2.1 遺伝的アルゴリズム

遺伝的アルゴリズム (GA) は、最適化問題を解決するための進化的アルゴリズムの一つである [8], [9]。i番目の個体は染色体に対応する m_{\max} 次元のベクトル \mathbf{g}_i をもち、個体の優劣は \mathbf{g}_i から計算される適応度 J_i で表される。ここで、 \hat{m}_{\max} を $g_{i;d}$ の最大値、染色体 \mathbf{g}_i の d 番目の遺伝子 $g_{i;d}$ を $g_{i;d} \in \{1, 2, \dots, \hat{m}_{\max}\}$ とする。GAは、優秀な個体を得るために、選択、交叉、突然変異を繰り返すことにより染色体 \mathbf{g}_i を改良し、適応度の高い染色体 (すなわち、優秀な個体) を次世代に残すアルゴリズムである。

エリート戦略に従い染色体を選択する場合、個体の適応度が高い順に \mathbf{g}_i をソートし、 $\mathbf{g}_1, \dots, \mathbf{g}_N$ の個体を残し、 $\mathbf{g}_{N+1}, \dots, \mathbf{g}_{b_{\max}}$ の個体を廃棄する。ここで、 b_{\max} は個体数、 $1 \leq N < b_{\max}$ である。次に、 $\mathbf{g}_1, \dots, \mathbf{g}_N$ からランダムに二つのベクトル $\mathbf{g}_i, \mathbf{g}_j$ を選び、 \mathbf{g}_i と \mathbf{g}_j を交叉し、次世代の染色体 $\mathbf{g}_{N+1}, \dots, \mathbf{g}_{b_{\max}}$ を作成する。その後、確率 p_M で $\mathbf{g}_{N+1}, \dots, \mathbf{g}_{b_{\max}}$ の遺伝子を突然変異により変更する。

2.2 高次元非線形最適化問題に対する複数のスパースな局所近似解の導出

以下の高次元非線形最適化問題に対する複数のスパースな局所近似解を求める問題を考える。

$$\begin{cases} \max_{\hat{\mathbf{x}}} \hat{f}_{\text{obj}}(\hat{\mathbf{x}}) \\ \text{subject to } \hat{\mathbf{f}}_{\text{eq}}(\hat{\mathbf{x}}) = \mathbf{0} \\ \hat{\mathbf{f}}_{\text{ne}}(\hat{\mathbf{x}}) \geq \mathbf{0} \end{cases} \quad (1)$$

ここで、 $\hat{\mathbf{x}} = (\hat{x}_1, \dots, \hat{x}_{\hat{m}_{\max}})^T$ は状態ベクトル、 \mathbf{T} は転置、 \hat{f}_{obj} は目的関数、 $\hat{\mathbf{f}}_{\text{eq}}$ は等式制約、 $\hat{\mathbf{f}}_{\text{ne}}$ は不等式制約であり、 $\hat{\mathbf{f}}_{\text{ne}}(\hat{\mathbf{x}}) \geq \mathbf{0}$ は $\hat{\mathbf{f}}_{\text{ne}}(\hat{\mathbf{x}})$ の全ての要素が0以上となることを表す。また、式 (1) の解 $\hat{\mathbf{x}}^*$ には、その m_{\max} 個 ($m_{\max} < \hat{m}_{\max}$) 以外の要素が0となるような制約 (スパース性) が要求されている。解にスパース性が要求されている最適化問題はスパース最適化問題と呼ばれ、従来から様々な手法が提案されてきた [5]~[7]。しかし、 $\hat{\mathbf{x}}$ の次元が数千~数万のような高次元の場合には、従来のスパース最適化手法により式 (1) の解を求めることは困難である。また、従来の手法では、解の0以外の要素の数を指定し、更に複数の局所近似解を一度に求めることは困難である。そこで、これらの問題を解決するために、上記スパース最適化

問題を GA を用いて解く方法が提案された [11]. 以下にその概要を述べる. まず, $\mathbf{g} = (g_1, \dots, g_{m_{\max}})^T$, $g_m \in \{1, 2, \dots, \hat{m}_{\max}\}$ なる変数を定義し, 式 (1) の解 $\hat{\mathbf{x}}^*$ の要素 \hat{x}_d^* には, 全ての $d \in \{1, 2, \dots, \hat{m}_{\max}\}$ に対して次式を満たすことが要求されているという問題を考える.

$$\hat{x}_d^* = 0 \quad \text{if} \quad d \notin \{g_1, g_2, \dots, g_{m_{\max}}\} \quad (2)$$

式 (2) は, $\hat{\mathbf{x}}^*$ の m_{\max} 個の要素以外は 0 となることを意味する. 次に, 新たに定義する変数 $\mathbf{x} \stackrel{\text{def}}{=} (\hat{x}_{g_1}, \hat{x}_{g_2}, \dots, \hat{x}_{g_{m_{\max}}})^T$ 及び

$$\hat{\mathbf{x}} = \mathbf{h}_{\text{tr}}(\mathbf{x}, \mathbf{g}) \quad (3)$$

を用いて, 式 (1) のスパース最適化問題を, 次式で表す.

$$\begin{cases} \max_{\mathbf{x}, \mathbf{g}} \hat{f}_{\text{obj}}(\mathbf{h}_{\text{tr}}(\mathbf{x}, \mathbf{g})) \\ \text{subject to} \quad \hat{f}_{\text{eq}}(\mathbf{h}_{\text{tr}}(\mathbf{x}, \mathbf{g})) = 0 \\ \hat{f}_{\text{ne}}(\mathbf{h}_{\text{tr}}(\mathbf{x}, \mathbf{g})) \geq 0 \end{cases} \quad (4)$$

ここで, \mathbf{h}_{tr} は, \mathbf{x} 及び \mathbf{g} から $\hat{\mathbf{x}}$ を構成する関数である. 式 (4) では, \mathbf{g} は自然数ベクトル, \mathbf{x} は実数ベクトルである. そのため, 式 (4) は \mathbf{g} に対する組合せ最適化問題と \mathbf{x} に対する制約付き非線形最適化問題の混合問題となり, \mathbf{x} と \mathbf{g} を同時に最適化することは困難である.

そこで, まず, 式 (4) をある \mathbf{g} に対して \mathbf{x} を最適化する制約付き非線形最適化問題に縮小する. すなわち, 式 (4) から \mathbf{g} に対する最適化操作を除くことにより, 次式を得る.

$$\begin{cases} \max_{\mathbf{x}} \hat{f}_{\text{obj}}(\mathbf{h}_{\text{tr}}(\mathbf{x}, \mathbf{g})) \\ \text{subject to} \quad \hat{f}_{\text{eq}}(\mathbf{h}_{\text{tr}}(\mathbf{x}, \mathbf{g})) = 0 \\ \hat{f}_{\text{ne}}(\mathbf{h}_{\text{tr}}(\mathbf{x}, \mathbf{g})) \geq 0 \end{cases} \quad (5)$$

式 (5) では \mathbf{x} の次元 m_{\max} が十分小さいことを仮定しているため, 式 (5) の最適解は通常の制約付き非線形最適化問題に対する解法を適用することにより得られる.

次に, 式 (5) の \mathbf{g} を GA を用いて最適化するために, 式 (5) における \mathbf{g} を GA における i 番目の個体の染色体 \mathbf{g}_i に置き換える. また, \mathbf{g}_i に対する式 (5) の解を \mathbf{x}^*_i として, \mathbf{g}_i に対する適応度 J_i を次式で定義する.

$$J_i \stackrel{\text{def}}{=} \hat{f}_{\text{obj}}(\mathbf{h}_{\text{tr}}(\mathbf{x}^*_i, \mathbf{g}_i)) \quad (6)$$

このようにして定義された J_i を用いて, GA を実行する. GA では, $J_i (i \in \{1, 2, \dots, b_{\max}\})$ の中で J_1 が最大の値を取る. すなわち, \mathbf{g}_1 が最適な染色体となる. したがって, 式 (1) の最適解の局所近似解の第一候補 $\hat{\mathbf{x}}_1^*$ は \mathbf{g}_1 と \mathbf{x}^*_1 を用いると, 以下のように表される.

$$\hat{\mathbf{x}}_1^* = \mathbf{h}_{\text{tr}}(\mathbf{x}^*_1, \mathbf{g}_1) \quad (7)$$

同様に, 局所近似解の第 i 番目の候補は

$$\hat{\mathbf{x}}_i^* = \mathbf{h}_{\text{tr}}(\mathbf{x}^*_i, \mathbf{g}_i) \quad (8)$$

となる. ここで, あらかじめ指定された $N_{\hat{c}} (N_{\hat{c}} \geq 1)$, $\hat{c}_1, \dots, \hat{c}_{N_{\hat{c}}}$ に対して, 局所近似解 $\hat{\mathbf{x}}_i^*$ の要素が $\hat{x}_{i;\hat{c}_1}, \dots, \hat{x}_{i;\hat{c}_{N_{\hat{c}}}} > 0$ とならなければならない場合, すなわち, $\hat{c}_1, \dots, \hat{c}_{N_{\hat{c}}}$ の全てが \mathbf{g}_i の要素に含まれなければならない場合, $m_{\max} > N_{\hat{c}}$ として, 染色体 \mathbf{g}_i を $\mathbf{g}_i \stackrel{\text{def}}{=} (\hat{c}_1, \dots, \hat{c}_{N_{\hat{c}}}, g_{i;N_{\hat{c}}+1}, \dots, g_{i;m_{\max}})^T$ のように定義する. $g_{i;N_{\hat{c}}+1}, \dots, g_{i;m_{\max}}$ を対象として, 初期化, 交叉, 及び突然変異を実行することにより, 全ての染色体が $\hat{c}_1, \dots, \hat{c}_{N_{\hat{c}}}$ を常に含まないように GA を実行できる.

2.3 多様な局所近似解を導出するための適応度の補正

前節で述べたスパース最適化手法 [11] は GA のエリート戦略を用いている. そのため適応度の大きい局所近似解の遺伝子が広まり, 局所近似解 $\hat{\mathbf{x}}_i^*$ の差が小さい. 本論文のテーマであるレシピ作成問題の場合について考えると, これは得られたレシピの多くが類似していることを意味する. 同じような献立が毎日続くと, その献立に飽きてしまい食事を楽しむことができな.

この問題を解決し, できるだけ異なる複数の局所近似解を導出するために, 次式で定義される適応度 J'_i を導入する.

$$J'_i \stackrel{\text{def}}{=} \begin{cases} C_J & i = 1 \\ J_i + w_M |J_i|^{w_E} \text{dist}_i & 2 \leq i \leq b_{\max} \end{cases} \quad (9)$$

ここで, dist_i は, i 番目 ($i > 1$) の染色体 \mathbf{g}_i と $\mathbf{g}_j (j < i)$ との平均距離であり, **3.2** で定義される. この補正により, 上位の局所近似解と似ていない局所近似解の適応度が大きくなり, 次世代に残りやすくなる. また, C_J を十分大きな値に設定することにより, $\hat{\mathbf{x}}_1^*$ が必ず次世代に残る. w_M, w_E は距離による補正項 (式 (9) 第 2 項) のパラメータである.

式 (9) で定義された適応度 J'_i を指標に個体を選択する遺伝的アルゴリズムを以下に示す。

[Algorithm 1] 多様な局所近似解を得るための遺伝的アルゴリズム (GA)

- (1-1) $\mathbf{g}_1, \dots, \mathbf{g}_{b_{\max}}$ の要素に 1 から \hat{m}_{\max} までの整数を乱数で与え、初期化する。
- (1-2) $\mathbf{g}_1, \dots, \mathbf{g}_{b_{\max}}$ から $J_1, \dots, J_{b_{\max}}$ を計算する。
- (1-3) J_i の値により, $J_1, \dots, J_{b_{\max}}$ 及び $\mathbf{g}_1, \dots, \mathbf{g}_{b_{\max}}$ をソートする. ($J_i > J_{i+1}$)
- (1-4) J'_1 に十分に大きい値を設定する. 式 (9) により J'_i を計算する ($2 \leq i \leq b_{\max}$).
- (1-5) J'_i の値により $J'_1, \dots, J'_{b_{\max}}$, 及び $\mathbf{g}_1, \dots, \mathbf{g}_{b_{\max}}$ をソートする. ($J'_i > J'_{i+1}$)
- (1-6) $|J_i - J_{\text{old}}|$ が指定した値より小さくなれば終了. そうでない場合, Step(1-7) へ.
- (1-7) $\mathbf{g}_1, \dots, \mathbf{g}_N$ を残し, $\mathbf{g}_{N+1}, \dots, \mathbf{g}_{b_{\max}}$ を廃棄する.
- (1-8) $n = N$ とする.
- (1-9) $\mathbf{g}_1, \dots, \mathbf{g}_N$ から, 二つのベクトル $\mathbf{g}_i, \mathbf{g}_j$ をランダムに選ぶ.
- (1-10) \mathbf{g}_i と \mathbf{g}_j を交叉し, $\mathbf{g}_{n+1}, \mathbf{g}_{n+2}$ を作成する. 交叉ポイントは 2 から $m_{\max} - 1$ までの乱数とする.
- (1-11) $n+2 \geq b_{\max}$ ならば, Step(1-12) へ. $n+2 < b_{\max}$ ならば, $n = n+2$ として Step(1-9) へ.
- (1-12) $\mathbf{g}_i (N+1 \leq i \leq b_{\max})$ の各要素を確率 p_M の突然変異により変更する.
- (1-13) $\mathbf{g}_i (2 \leq i \leq b_{\max}) \in \{\mathbf{g}_1, \dots, \mathbf{g}_{i-1}\}$ の場合, $\mathbf{g}_i \notin \{\mathbf{g}_1, \dots, \mathbf{g}_{i-1}\}$ となるように \mathbf{g}_i を乱数で生成する.
- (1-14) $J_{\text{old}} = J_1$ として, Step(1-2) へ戻る.

また, Algorithm 1 の有効性を確認するために, Algorithm 1, Step(1-4) の式 (9) を次式に置き換える。

$$J'_i \stackrel{\text{def}}{=} \begin{cases} C_J & i = 1 \\ J_i + N(0, (\sigma_M |J_i|^{\sigma_E})^2) & 2 \leq i \leq b_{\max} \end{cases} \quad (10)$$

$N(0, \sigma^2)$ は平均 0, 分散 σ^2 の正規乱数, σ_M 及び σ_E はそのパラメータである. 式 (10) 第 2 項により, 適応度 J_i が低い場合でも, その個体が次世代に残る確率が高くなり, ルーレット戦略 [8], [9] と同様な効果が得られる. 本アルゴリズムを Algorithm2 とする.

上記 Algorithm 1 では, エリート戦略 [8], [9] に従い適応度が最も大きい染色体 \mathbf{g}_1 を次世代に残すため

に, Step(1-4) で J'_1 に十分に大きい値を設定する. これにより, \mathbf{g}_1 は Step(1-5) のソートにかかわらず, 必ず次世代に残る. 残りの $\mathbf{g}_2, \dots, \mathbf{g}_{b_{\max}}$ については, 距離または乱数による補正を適用した J'_i の値によって選択される.

3. 食材・配合量の最適化

3.1 食材・配合量最適化問題の定式化

\hat{m}_{\max} 個の食材から m_{\max} 個の食材 (調理食材) を選び, それらを用いて料理を作る際に, 調理食材の選択と食材配合量の調整により, 栄養素, 調理食材の原価, 及び味を最適化する問題を考える.

使用可能な食材の番号 $(1, 2, \dots, \hat{m}_{\max})$ の集合を使用可能食材集合 \hat{M} , 調理食材の食材番号の集合 (\hat{M} の部分集合) を調理食材集合 M_i (要素数 m_{\max}), M_i の全ての要素からなる調理食材番号ベクトルを \mathbf{g}_i とする. ここで, $i \in \{1, \dots, b_{\max}\}$ は調理食材集合の番号を表す. また b_{\max} は調理食材集合 M_i の数であり, 2.1 で定義された GA の個体数に対応する. $\hat{\mathbf{x}}$ を \hat{M} に属する全ての食材に対する食材配合量ベクトル, \mathbf{x}_i を M_i に属する全ての食材に対する食材配合量ベクトルとする. 例えば, $\hat{m}_{\max} = 10$, $\hat{M} \stackrel{\text{def}}{=} \{1, 2, \dots, 10\}$, $m_{\max} = 3$, $b_{\max} = 4$ としたときの \hat{M} と M_i の関係の一例を図 1 に示す. また, $\mathbf{g}_2 = (1, 3, 8)^T$, $\mathbf{x}_2 = (16, 50, 40)^T$ としたとき, \mathbf{g}_2 と \mathbf{x}_2 に対応する $\hat{\mathbf{x}}$ は, $\hat{\mathbf{x}} = (16, 0, 50, 0, 0, 0, 0, 40, 0, 0)^T$ となる. $\hat{\Psi}$ を \hat{M} に属する全ての食材の栄養素を表す \hat{m}_{\max} 行 d_y 列の食材栄養素行列として, 次式で定義する.

$$\hat{\Psi} \stackrel{\text{def}}{=} \begin{bmatrix} \hat{\psi}_1 & \hat{\psi}_2 & \dots & \hat{\psi}_{\hat{m}_{\max}} \end{bmatrix}^T \quad (11)$$

ここで, $\hat{\psi}_m$ は \hat{M} に属する m 番目の食材 ($m \in \{1, 2, \dots, \hat{m}_{\max}\}$) の栄養素を表す d_y 次元の栄養素ベ

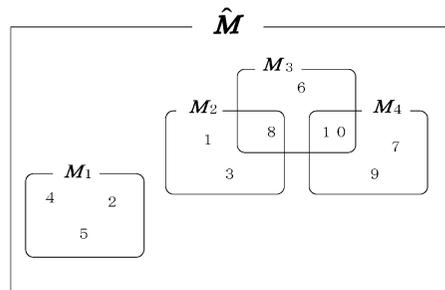


図 1 使用可能食材集合 \hat{M} と調理食材集合 M_i の一例
Fig.1 Example of \hat{M} and M_i .

クトルであり、食品成分表 [13] をもとに作成される。料理の栄養素ベクトル \mathbf{y} は、 $\hat{F}_B \stackrel{\text{def}}{=} \hat{\Psi}^T$ を用いて次式により求められると仮定する。

$$\mathbf{y} = \hat{F}_B \hat{\mathbf{x}} \quad (12)$$

$\mathbf{y}^* = (y^*_1, y^*_2, \dots, y^*_{d_y})^T$ を料理の栄養素ベクトルの目標値、 W_y を d_y 行 d_y 列の正定値行列として、 \mathbf{y}^* と \mathbf{y} の誤差 ε (料理の栄養素ベクトルの誤差) を次式で定義する。

$$\varepsilon \stackrel{\text{def}}{=} (\mathbf{y} - \mathbf{y}^*)^T W_y (\mathbf{y} - \mathbf{y}^*) / d_y^2 \quad (13)$$

\hat{F}_C を \hat{M} に属する食材の価格を要素とする 1 行 \hat{m}_{\max} 列の行列とすると、料理の原価 $cost$ は次式により得られる。

$$cost \stackrel{\text{def}}{=} \hat{F}_C \hat{\mathbf{x}} \quad (14)$$

\hat{W}_r を \hat{M} に属する食材の相関 \hat{r}_{ij} を要素とする \hat{m}_{\max} 行 \hat{m}_{\max} 列の行列として、料理に使われている食材の相関の総和 r_{all} を次式で定義する。

$$r_{\text{all}} \stackrel{\text{def}}{=} \frac{\hat{\mathbf{x}}^T \hat{W}_r \hat{\mathbf{x}}}{\sum_{i=1}^{\hat{m}_{\max}} \hat{x}_i^2} \quad (15)$$

ここで、 \hat{r}_{ij} を既存レシピの食材配合量の相関係数とすると、 \hat{r}_{ij} が大きいほど i 番目の食材と j 番目の食材が同時に使用されているレシピが多いことを示す。 r_{all} が大きくなるように食材を選択することは、既存レシピでよく使用されている食材の組合せを選択することを意味するため、既存料理と同様の味を保証できる可能性が高くなる。

また、 \hat{M} の食材 m の食材配合量の最小値を $\hat{x}_{\min;m}$ 、食材 m の食材配合量の最大値を $\hat{x}_{\max;m}$ 、 $\hat{\mathbf{x}}_{\min} \stackrel{\text{def}}{=} (\hat{x}_{\min;1}, \dots, \hat{x}_{\min;\hat{m}_{\max}})^T$ 、及び $\hat{\mathbf{x}}_{\max} \stackrel{\text{def}}{=} (\hat{x}_{\max;1}, \dots, \hat{x}_{\max;\hat{m}_{\max}})^T$ とすると、 $\hat{\mathbf{x}}$ に関して以下の制約条件が成り立つ。

$$\hat{\mathbf{x}}_{\min} \leq \hat{\mathbf{x}} \leq \hat{\mathbf{x}}_{\max} \quad (16)$$

式 (12)、式 (14)、及び式 (16) は、 \mathbf{x}_i 、 \mathbf{g}_i 、及び式 (3) を用いると、以下のように表される。

$$\mathbf{y} = \hat{F}_B \mathbf{h}_{\text{tr}}(\mathbf{x}_i, \mathbf{g}_i) \quad (17)$$

$$cost = \hat{F}_C \mathbf{h}_{\text{tr}}(\mathbf{x}_i, \mathbf{g}_i) \quad (18)$$

$$\hat{\mathbf{x}}_{\min} \leq \mathbf{h}_{\text{tr}}(\mathbf{x}_i, \mathbf{g}_i) \leq \hat{\mathbf{x}}_{\max} \quad (19)$$

式 (13) の栄養素ベクトルの誤差、式 (18) の原価、及び式 (15) の食材の相関の総和を用いて、 $\hat{f}_{\text{obj}}(\mathbf{h}_{\text{tr}}(\mathbf{x}_i, \mathbf{g}_i))$ を次式で定義する。

$$\hat{f}_{\text{obj}}(\mathbf{h}_{\text{tr}}(\mathbf{x}_i, \mathbf{g}_i)) \stackrel{\text{def}}{=} -w_\varepsilon \varepsilon - w_c cost - w_r \exp(-r_{\text{all}}) \quad (20)$$

ここで、 $\exp(-r_{\text{all}})$ は、目的関数 \hat{f}_{obj} の最大値を 0 とするために r_{all} を変換した項である。また、 w_ε は料理の栄養素ベクトルの誤差 ε に対する重み、 w_c は料理の原価 $cost$ に対する重み、 w_r は食材の相関の総和 r_{all} に対する重みを表す。 M_i に対する最適な食材配合量ベクトルを \mathbf{x}^*_i とすると、 \mathbf{x}^*_i は以下の最適化問題を解くことにより得られる。

$$\left\{ \begin{array}{l} \max_{\mathbf{x}_i} \hat{f}_{\text{obj}}(\mathbf{h}_{\text{tr}}(\mathbf{x}_i, \mathbf{g}_i)) \\ \text{subject to } \varepsilon_{\max} - \varepsilon \geq 0 \\ c_{\max} - cost \geq 0 \\ r_{\text{all}} - r_{\text{all}_{\min}} \geq 0 \\ \mathbf{h}_{\text{tr}}(\mathbf{x}_i, \mathbf{g}_i) - \hat{\mathbf{x}}_{\min} \geq \mathbf{0} \\ \hat{\mathbf{x}}_{\max} - \mathbf{h}_{\text{tr}}(\mathbf{x}_i, \mathbf{g}_i) \geq \mathbf{0} \\ \mathbf{y} - \mathbf{y}_{\min} \geq \mathbf{0} \\ \mathbf{y}_{\max} - \mathbf{y} \geq \mathbf{0} \end{array} \right. \quad (21)$$

ここで、 ε_{\max} は ε の最大値、 c_{\max} は $cost$ の最大値、 $r_{\text{all}_{\min}}$ は r_{all} の最小値を表す。また、 $\mathbf{y}_{\min} = (y_{\min;1}, y_{\min;2}, \dots, y_{\min;d_y})^T$ は料理の栄養素の最小値ベクトル、 $\mathbf{y}_{\max} = (y_{\max;1}, y_{\max;2}, \dots, y_{\max;d_y})^T$ は料理の栄養素の最大値ベクトルを表す。

3.2 染色体の距離

2.3 で導入された染色体 \mathbf{g}_i の距離 $dist_i$ は、問題に応じて様々な定義される。本節では、食材配合量最適化問題に適した距離 $dist_i$ の定義を述べる。

まず、 \mathbf{g}_i の 2 値表現 $\hat{\mathbf{g}}_i = (\hat{g}_{i;1}, \hat{g}_{i;2}, \dots, \hat{g}_{i;\hat{m}_{\max}})^T$ を定義する。ここで、 $\hat{\mathbf{g}}_i$ の要素 $\hat{g}_{i;d}$ は次式を満たす。

$$\hat{g}_{i;d} = \begin{cases} 0 & d \notin \{g_{i;1}, \dots, g_{i;\hat{m}_{\max}}\} \\ 1 & d \in \{g_{i;1}, \dots, g_{i;\hat{m}_{\max}}\} \end{cases} \quad (22)$$

$\hat{\mathbf{g}}_i$ を用いて以下の 2 種類の $dist_i$ を定義する。

a) 各染色体との距離の平均

1, \dots , $i-1$ 番目 ($i > 1$) の各染色体との距離の平均を $dist_i$ とする。

$$dist_i = \frac{1}{i-1} \sum_{j=1}^{i-1} \sum_{d=1}^{\hat{m}_{\max}} (\hat{g}_{i;d} - \hat{g}_{j;d})^2 \quad (23)$$

b) 各染色体の平均との距離

1, ..., i-1 番目 ($i > 1$) の各染色体の平均と i 番目の染色体との距離を $dist_i$ とする.

$$dist_i = \sum_{d=1}^{\hat{m}_{\max}} \left(\left(\frac{1}{i-1} \sum_{j=1}^{i-1} \hat{g}_{j;d} \right) - \hat{g}_{i;d} \right)^2 \quad (24)$$

式 (21) の解 \mathbf{x}^*_i から式 (6), (9), 及び (20) により計算される適応度 J'_i を用いて Algorithm 1 を実行することにより, 適応度 J'_i が高い順に, $\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, \dots$ 及び, $\mathbf{x}^*_1, \mathbf{x}^*_2, \mathbf{x}^*_3, \dots$ が得られる. 適応度 J'_i は, \mathbf{g}_i が示す食材及びその配合量 \mathbf{x}^*_i により与えられる料理の栄養ベクトルの誤差, 原価, 及び食材の相関に対する評価値 J_i と, \mathbf{g}_i と \mathbf{g}_j ($j < i$) との距離に関する評価値の和である. したがって, Algorithm 1 により, レシピの評価値が高くかつ上位のレシピとの距離が大きい順に, 多様なレシピ (食材及びその配合量) が得られる.

なお, Algorithm 1 は適応度ができるだけ高くなるような食材及びその配合量を与えるのみである. 今後, 料理手順を与える手法についても検討していく予定である.

4. 性能評価

本節では, 2. 及び 3. で提案された手法を用いて食材及びその配合量を最適化し, 提案手法の有効性を示す.

4.1 局所近似解の多様性の評価

局所近似解の多様性を評価するために, 式 (9) で定義された適応度 J'_i のパラメータ w_M , w_E , 及び $dist_i$ の計算方法に対する上位 N_σ 個の局所近似解の分散 σ_g^2 の変化を調べた. ここで, σ_g^2 は次式で定義される.

$$\sigma_g^2 = \sum_{d=1}^{\hat{m}_{\max}} \frac{1}{N_\sigma} \sum_{i=1}^{N_\sigma} (\bar{g}_d - \hat{g}_{i;d})^2 \quad (25)$$

$$\bar{g}_d = \frac{1}{N_\sigma} \sum_{i=1}^{N_\sigma} \hat{g}_{i;d} \quad (26)$$

ここで, $N_\sigma = 10$ である. 以下に, その結果を示す. まず, $dist_i$ を各染色体との距離の平均 (式 (23)) とした場合のシミュレーション結果を図 2 に示す. w_E が 0.0, 0.2, 0.4 のいずれの場合でも w_M の増加に従い局所近似解の分散 σ_g^2 が増加した. 一方, $w_M \geq 10^{-2}$ では, w_E の増加に従い, J_i の平均 $E[J_i]$ の減少が顕

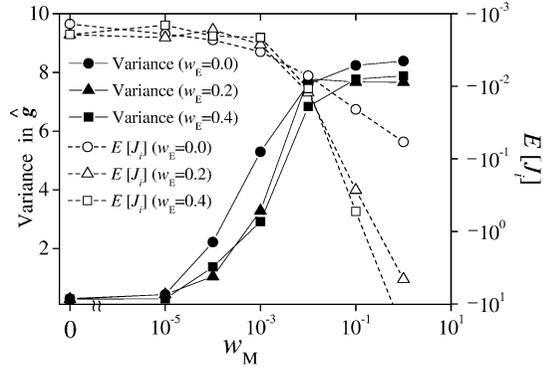


図 2 w_M に対する J_i の平均 $E[J_i]$ 及び局所近似解の分散 σ_g^2 (距離による補正: 式 (23))

Fig. 2 Effect of w_M on average $E[J_i]$ and variance σ_g^2 (correction with distance: Eq. (23)).

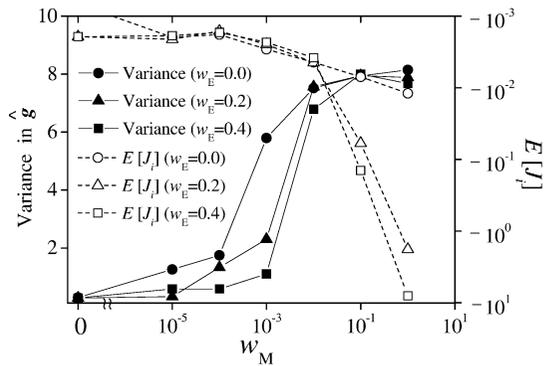


図 3 w_M に対する J_i の平均 $E[J_i]$ 及び局所近似解の分散 σ_g^2 (距離による補正: 式 (24))

Fig. 3 Effect of w_M on average $E[J_i]$ and variance σ_g^2 (correction with distance: Eq. (24)).

著になる. 特に w_E が 0.2, 0.4 の場合では, $E[J_i]$ の減少幅が大きくなった. $dist_i$ を各染色体の平均との距離 (式 (24)) とした場合のシミュレーション結果 (図 3) についても同様である. したがって, $dist_i$ の定義にかかわらず $w_E = 0.0$, $w_M = 10^{-2} \sim 10^{-1}$ とすることにより $E[J_i]$ の値を大幅に下げることなく解の分散を大きくできることが分かる.

比較のため, Algorithm2 (エリート+ルーレット戦略と同等) により得られた $E[J_i]$ と σ_g^2 を図 4 に示す. 図 4 より, エリート+ルーレット戦略では十分大きな σ_g^2 を得るためには, 図 2 及び図 3 の場合と比較して $E[J_i]$ が小さくなる. 以上の結果より, 多様な局所近似解を得るために, 染色体の距離を導入した適応度が有効であることが分かる.

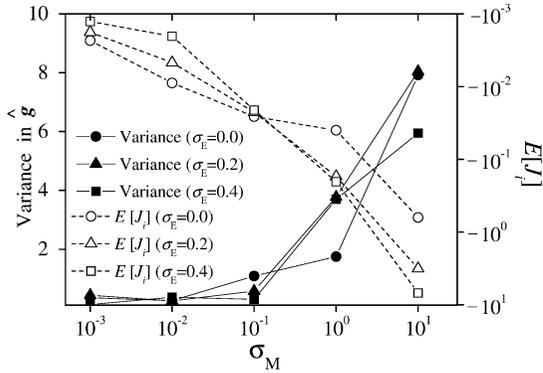


図 4 σ_M に対する J_i の平均 $E[J_i]$ 及び局所近似解の分散 σ_g^2 (エリート+ルーレット戦略)
 Fig. 4 Effect of σ_M on average $E[J_i]$ and variance σ_g^2 (elite+roulette strategy).

4.2 食材・配合量問題の局所近似解

まず、成人男性の一食分の栄養素の摂取基準値 [1] を用いて、栄養素ベクトルの目標値 \mathbf{y}^* を作成した。次に、食品成分表 [13] をもとに食材栄養素行列 $\hat{\Psi}$ を作成した。 $\hat{\Psi}$ の栄養素の種類は、栄養素ベクトルの目標値の栄養素の種類に合わせて決められる。表 1 に栄養素の目標値の例を、表 2 に食材に対する栄養素の例を示す。

本論文では、栄養素を 14 種類として作成された栄養素ベクトルの目標値 \mathbf{y}^* 、及び成分表に記載された食材の中から比較的入手が容易で料理に使いやすい 128 種類の食材に対する食材栄養素行列 $\hat{\Psi}$ を用いた。ここで、計算誤差を減らすため、食材栄養素行列 $\hat{\Psi}$ の各要素を目標値 \mathbf{y}^* の各要素をもとに正規化した。また、日本人の食事摂取基準 [1] に基づき、 $y_{\min;m}$ を推定平均必要量または目標量の最小値に、 $y_{\max;m}$ を耐容上限量または目標量の最大値に設定した。これらの指標が示されていない栄養素に対しては、 $y_{\min;m}$ を y_m^* の 0.9 倍の値、 $y_{\max;m}$ を y_m^* の 1.1 倍の値とした。栄養素ベクトルの目標値の要素を 1 としたときの提案手法により得られた \mathbf{g}_1 に対する栄養素ベクトル \mathbf{y}_1 を図 5 に示す。図 5 より、 \mathbf{y}_1 が栄養素ベクトルの目標値 \mathbf{y}^* に近く、 \mathbf{y}_{\min} と \mathbf{y}_{\max} の間にあることが分かる。ただし、今回の計算では調理による栄養成分の変化は考慮されていないため、調理後の栄養素ベクトルは目標値と異なる可能性がある。また、3. で述べたように、本論文の提案手法は食材及びその配合量を与えるのみである。調理による栄養素の変化の導入及び

表 1 栄養素の目標値の例
 Table 1 Example of target nutritional values.

栄養素 単位	エネルギー [kcal]	たんぱく質 [g]	炭水化物 [g]	脂質 [g]
目標値	600	50	130	30

表 2 食材に対する栄養素の例
 Table 2 Example of nutritional values for food ingredients.

栄養素 単位	エネルギー [kcal]	たんぱく質 [g]	炭水化物 [g]	脂質 [g]
食材 1	358	12.7	64.9	6
食材 2	364	10.5	73.1	2.7
食材 3	211	4.4	46.5	0.8
食材 4	356	10.6	73.1	1.7
食材 5	341	10.9	72.1	2.1

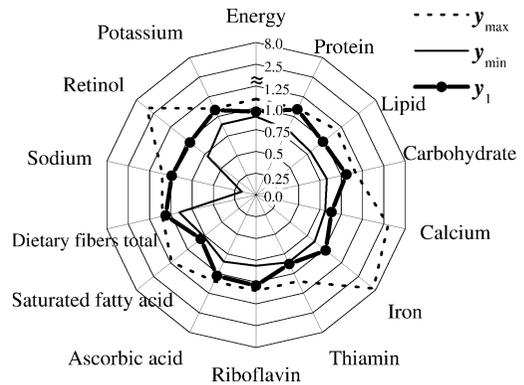


図 5 栄養素ベクトル \mathbf{y}_1 (栄養素の目標値を 1 として正規化)
 Fig. 5 Nutritional value \mathbf{y}_1 (normalized as target nutritional values equal one).

料理手順を与える手法は今後の課題である。

また、 $w_e = w_c = 0, w_r = 1$ とし食材の相関を考慮した場合の食材及びその配合量の一例を表 3 に、 $w_e = 1, w_c = w_r = 0$ とし食材の相関を考慮しない場合の食材及びその配合量の一例を表 4 に示す。ここで、相関行列 \hat{W}_r は既存レシピのデータをもとに作成された。表 3 より、 $w_r = 1$ とし食材の相関を考慮した場合は、既存レシピでよく使用されている食材の組合せが選択されていることが分かる。一方、表 4 より、食材の相関を考慮しない場合は、既存レシピでは使用されない食材の組合せが選択される場合があることが分かる。このように、評価関数に食材の相関を導入することにより、既存レシピと同様の味を保証する解を得られる確率が高くなる。

表 3 食材の相関を考慮した場合の食材及びその配合量

Table 3 Ingredients and their quantities obtained using correlations of ingredients.

食材	配合量 [g]
こめ	214.0
たまねぎ	103.0
豚肉	69.3
なたね油	15.3
こいくちしょうゆ	30.0

表 4 食材の相関を考慮しない場合の食材及びその配合量

Table 4 Ingredients and their quantities obtained without using correlations of ingredients.

食材	配合量 [g]
食パン	140.0
こめ	146.0
クリーム	13.8
ナチュラルチーズ	5.9
マヨネーズ	18.1

5. む す び

高次元非線形スパース最適化問題に対する複数の多様な局所近似解を得るために、遺伝的アルゴリズムの適応度に染色体間の距離を導入した。この解法を用いて、使用可能な食材の中から一食分の料理に用いられる十種類程度の食材を選択し、選択された食材を用いて所望の栄養バランスを満たす食材配合量を計算した。その結果、所望の栄養バランスを満たし、かつ多様な料理を提供するための食材及び食材配合量を複数列挙できた。

謝辞 レシピ作成問題に対してスパース最適化プログラムの使用をお認めくださった(株)インテリジェントセンサーテクノロジーの池崎秀和社長、及びプログラムの作成に協力していただいた公立はこだて未来大学佐藤仁樹研究室の内藤優貴君、長坂明輝君に深謝いたします。

文 献

- [1] 第一出版編集部, 日本人の食事摂取基準 (2010年版), 第一出版, 2010.
- [2] 岡村吉隆, 原登久子, “病院給食における給与栄養量の評価: 特にミネラルについて,” 大阪樟蔭女子大学論集, vol.37, pp.93-105, 2000.
- [3] 加藤哲子, “病院食栄養基準における成分別栄養管理導入の試み,” 山形県立米沢女子短期大学紀要, vol.38, pp.23-33, June 2003.
- [4] D.G. Luenberger, Linear and Nonlinear Programming, Addison-Wesley Publishing Company, USA, 1984.
- [5] R. Tibshirani, “Regression shrinkage and selection via the lasso,” J. Roy. Stat. Soc. B, vol.58, no.1,

pp.267-288, 1996.

- [6] S.S. Chen, D.L. Donoho, and M.A. Saunders, “Atomic decomposition by basis pursuit,” SIAM J. Sci. Comput. vol.20, no.1, pp.33-61, 1998.
- [7] T. Zhang, “Adaptive forward-backward greedy algorithm for learning sparse representations,” IEEE Trans. Inf. Theory, vol.57, no.7, pp.4689-4708, July 2011.
- [8] D.E. Goldberg, Genetic Algorithms in Search Optimization and Machine Learning, Addison-Wesley, USA, 1989.
- [9] J.H. Holland, “Genetic algorithm,” Scientific American, pp.66-72, July 1992.
- [10] R. Dong and W. Pedrycz, “Sparse optimization using a mixed GA-PSO optimization framework,” Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp.1862-1866, 2010.
- [11] 佐藤雅子, 佐藤仁樹, 池崎秀和, “味覚センサの出力データに基づく原料・ブレンド比の最適化,” 電学論 (E), vol.135, no.2, pp.57-64, 2015.
- [12] J.A. Nelder and R. Mead “A simplex method for function minimization,” Comput. J., vol.7, pp.308-313, 1965.
- [13] 香川芳子, 食品成分表 (2013), 女子栄養大学出版社, 2013.
- [14] 北野宏明, 遺伝的アルゴリズム, 産業図書, 1993.
- [15] 電気学会 GA 等組合せ最適化手法応用調査専門委員会 (C 部門), 遺伝的アルゴリズムとニューラルネットワークスケジューリングと組み合わせ最適化, コロナ社, 1998.

(平成 27 年 5 月 5 日受付, 12 月 2 日再受付)



佐藤 雅子

1989 年筑波大学第三学群情報学類卒業。同年(株)東芝研究開発センター入社。ATM 交換機の OS 開発及びモバイル端末のアプリケーション開発に従事。1997 年同社退職。2012 年より(株)インテリジェントセンサーテクノロジーテクニカルスタッフ。原料・ブレンド比最適化ツールの開発に従事。



佐藤 仁樹 (正員)

1987 年早稲田大学理工学研究科修士課程修了。同年(株)東芝研究開発センター入社。音声のパケット化, ATM 網のトラフィック制御, 及びインターネットの輻輳制御の研究に従事。2000 年 4 月より(株)ワイ・アール・ピー移動通信基盤技術研究所に出向。移動通信網の送信電力制御, 輻輳制御, 及びインターネット TV 会議システムの研究に従事。2002 年より公立はこだて未来大学にて, 非線形システムの解析及び最適化の研究に従事。博士(情報科学)早稲田大学。