

卒業論文

パーソナライズAIと多目的最適化を用いた 献立提案システムの開発

Development of a Menu Proposal System Using Personalized AI
and Multi-objective Optimization

富山県立大学 工学部 情報システム工学科

2220039 辻琉玖

指導教員 António Oliveira Nzinga René 准教授

提出年月: 令和8年(2026年)2月

目 次

図一覧	ii
表一覧	iii
記号一覧	iv
第1章 はじめに	1
§ 1.1 本研究の背景	1
§ 1.2 本研究の目的	2
§ 1.3 本論文の概要	3
第2章 UXを考慮した献立作成支援	5
§ 2.1 テキストマイニングによる嗜好情報の抽出	5
§ 2.2 多目的最適化と NSGA-II	7
§ 2.3 適合性フィードバックと Rocchio アルゴリズム	10
第3章 アルゴリズムの一般的応用と関連研究	13
§ 3.1 多目的遺伝的アルゴリズムの応用分野	13
§ 3.2 ベクトル空間モデルと推薦システム	15
§ 3.3 パーソナライズとフィードバック学習	18
第4章 提案手法	21
§ 4.1 システムアーキテクチャとデータ基盤	21
§ 4.2 嗜好学習モデルと推論アルゴリズム	25
§ 4.3 多目的最適化問題の定式化と解探索	26
第5章 数値実験並びに考察	30
§ 5.1 数値実験の概要	30
§ 5.2 実験結果と考察	33
第6章 おわりに	36
謝辞	37
参考文献	38

図一覧

2.1	パレート最適解のイメージ	8
2.2	NSGA-II のアルゴリズムフロー	8
2.3	Rocchio アルゴリズムによる嗜好ベクトルの更新概念図	11
3.1	航空機主翼設計における多目的最適化の例（構造質量と空気抵抗係数のトレードオフ）[13]	14
3.2	ベクトル空間モデルにおけるレシピの表現例（TF-IDF による重み付け）	16
3.3	協調フィルタリングとコンテンツベースフィルタリングの比較	17
3.4	本研究におけるインタラクティブな学習サイクル（献立提案と評価のループ）	20
4.1	提案システムの詳細処理フローと数理モデル	22
4.2	ユーザー基本情報およびアレルギー・疾患情報の入力画面	24
4.3	調理時間制約および最適化目的の選択画面	24
4.4	レシピ詳細表示およびフィードバック入力画面	29
4.5	パレート最適解の可視化と対話的な選択インターフェース	29
5.1	栄養素データの例	31
5.2	最適化処理の実行画面	31
5.3	パレート解の出力	32
5.4	対話型処理による解の選択	32
5.5	ユーザ1のモデル評価の推移	35
5.6	ユーザ2のモデル評価の推移	35
5.7	ユーザ3のモデル評価の推移	35

表一覧

4.1	本システムの開発環境と使用ライブラリ	23
4.2	レシピデータベースの主要カラム定義	23
4.3	画像取得における信頼済みドメインリスト (Trusted Domains)	24
4.4	NSGA-II ハイパーパラメータ設定	27
5.1	モデル評価 (精度) の出力結果	33
5.2	精度の t 検定の結果	33
5.3	モデル評価 (適合率) の出力結果	33
5.4	適合率の t 検定の結果	33
5.5	モデル評価 (再現率) の出力結果	34
5.6	再現率の t 検定の結果	34
5.7	モデル評価 (F 値) の出力結果	34
5.8	F 値の t 検定の結果	34

記号一覧

以下に本論文において用いられる用語と記号の対応表を示す.

用語	記号
j 人目の使用者の名前	ϵ_j
j 人目の身長	α_j
j 人目の体重	β_j
j 人目の基礎代謝量 (下限)	B_j^L
j 人目基礎代謝量 (上限)	B_j^H
j 人目のアレルギー情報	x_j
j 人の有する生活習慣病	z_j
対象の日数	D
レシピの数	R
食材の数	Q
栄養素の数	N
データベース上の食材数	S
データベース上の食材番号	$d : 1, 2, 3, \dots, S$
日の番号	$k : 1, 2, 3, \dots, 3D$
栄養素の番号	$l : 1, 2, 3, \dots, N$
材料の番号	$m : 1, 2, 3, \dots, Q$
レシピの番号	$i : 1, 2, 3, \dots, R$
i 番目のレシピの名前	y_i
i 番目のレシピの献立フラグ	r_{ki}
i 番目のレシピの主菜フラグ	σ_i
i 番目のレシピの調理時間	T_i
i 番目のレシピの摂取カロリー	C_i
i 番目のレシピの調理コスト	G_i
i 番目のレシピの UX 項目の推定確率	Q_i
i 番目のレシピの m 番目の材料の名前	q_{im}
i 番目のレシピの m 番目の材料量	e_{im}
i 番目のレシピの l 番目の栄養素の名前	n_{il}
i 番目のレシピの l 番目の栄養素の量	f_{il}
d 番目の食材名	Z_d
d 番目の食材の販売単位	W_d
d 番目の食材の値段	M_d

はじめに

§ 1.1 本研究の背景

戦後の日本では急速に生活様式が欧米化し、特に食生活においてはジャンクフードやファストフードなど、手軽で高エネルギーながら栄養が偏った食品が普及するようになった [1]。この変化により、現代では生活習慣病を患う人々が増加しており、その予防と改善が大きな課題となっている。生活習慣病は、食習慣や運動習慣、喫煙、飲酒、ストレスなど、日々の生活習慣が原因となって発症する疾患であり、脳血管疾患や心疾患といった深刻な病気を引き起こすリスクがある [2]。特に栄養が偏った状態が続くと、そのリスクはさらに高まる。

生活習慣病の特徴的な問題は、その発症が徐々に進行するため、日々の生活の中で知らず知らずのうちに血管や心臓、脳にダメージが蓄積し、症状が現れるころにはすでに命に関わる疾患に進行していることが多い。このため、早期の予防が非常に重要であり、特にバランスの取れた食事の摂取が欠かせない要素となる [4]。しかし、現代の忙しい生活環境において、栄養バランスを考慮した食事を毎日作ることは多くの家庭にとって大きな負担となっている。

特に、共働きの家庭や時間に追われる家庭では、食事の準備にかける時間が限られており、手軽に食事を済ませる傾向が強くなる。結果的に、栄養が偏りがちで、外食や加工食品に頼ることが増えてしまう。これが生活習慣病を引き起こすリスクを高める要因となっている。外食や便利な加工食品が普及する一方で、家計や健康への影響を考慮すると、栄養バランスを意識した食事を効率的に作る方法が求められる。

家庭での食事作りは、栄養バランスを取ることだけでなく、家族全員の嗜好や食べる時間帯、調理の手間を考慮する必要がある。特に、忙しい日々の中で毎日の献立を考えることは、時間的な余裕がない家庭では大きな負担となり、簡単に済ませてしまいがちだ。これにより、栄養が偏った食事が続き、健康への影響が懸念される。家族全員が満足できる食事を提供するためには、食事作りの効率化が求められる。さらに、食事が日々の楽しみであるべきだという点も重要である。

このような背景の中で、個人の好みや UX（ユーザーエクスペリエンス）を重視した献立作成システムの導入が重要となる。家庭ごとのニーズに合わせたパーソナライズされた提案を行うことができれば、栄養バランスを保ちながらも、家族全員が満足できる献立を手軽に作成することが可能になる。ユーザーの過去の評価や好みを反映したシステムが提供されれば、献立作成の負担が軽減され、健康的な食生活が促進される。このようなシステムは、栄養士の負担を軽減すると同時に、家庭内での食事作りを効率化し、生活習慣病の予防に貢献することが期待される。

§ 1.2 本研究の目的

栄養バランスが取れた献立を作成するには、膨大なメニューの組み合わせや、それぞれの栄養素や摂取カロリーの計算など、多くの要素を考慮する必要がある。特に、忙しい日常の中で献立を考える時間がなかったり、食材や栄養を効率よく組み合わせることが難しいと感じる人々も少なくない。また、家計や時間的な制約を持つ家庭では、手軽に、かつ経済的に栄養バランスの良い食事を提供することが理想的である。しかし、献立作成における複雑な計算や調整は、時間がかかり面倒に感じられることが多い。

加えて、学校給食や病院食など、公共機関での献立作成を担う栄養士には、毎日の食事計画において、栄養計算や食材費用の計算、さらには食材の調達や調理の簡便さを考慮しながら、何度も献立を見直す作業が求められている。このプロセスは非常に時間を要し、繰り返し行う作業であるため、その負担は大きい。

このような背景を踏まえ、本研究では、献立作成の負担を軽減し、効率的に栄養バランスを取った食事を提供するシステムを提案する。具体的には、Web 上の複数のレシピサイトから収集した膨大な料理データを基に、調理時間やコスト、個々の身体情報や嗜好に基づいた最適な献立を自動提案することを目指す。収集データには、各料理の食材や栄養価、カロリーに加え、食材ごとの価格データも含まれる。

先行研究として、堀 [3] は、多目的最適化を用いた献立提案において、ユーザーの評価に基づくロジスティック回帰分析を取り入れ、UX 評価（調理しやすさ等）を推定して次回の生成に活かす手法を提案した。しかし、同手法は評価点数の予測・推定に主眼が置かれており、レシピに含まれる具体的な食材や味の傾向といった「コンテンツの中身」に基づく詳細な嗜好学習については改善の余地があった。

そこで本研究では、先行研究のアプローチを発展させ、レシピの食材や特徴を TF-IDF 法によりベクトル化し、Rocchio アルゴリズムを用いて「好みのベクトル」そのものを直接的に更新・学習するアプローチを採用する。これにより、単なる評価値のフィードバックにとどまらず、ユーザーが好む具体的な食材や味付けの傾向を数学的なベクトルとして学習・蓄積することが可能となり、より高度なパーソナライズを実現する。

また、組み合わせ多目的最適化問題として献立を作成する際、遺伝的アルゴリズムを応用した非優越ソート遺伝的アルゴリズム（NSGA-II）を使用し、最適な献立を提案する [5]。これにより、栄養バランスを保ちながらも、コストや時間といった個々のニーズに最適な献立が提供される。

さらに、本研究の特徴として、ベクトル空間モデル [6]（TF-IDF とコサイン類似度）を採用し、ユーザーの嗜好を学習する機能を実装する。ユーザーが過去に評価した献立のテキスト情報から「好みベクトル」を生成し、全レシピと比較することで、個人の好みに合った献立や、マンネリを防ぐ意外性のある献立をスコア化して提案に反映させる。また、提案された複数の献立候補を散布図として可視化し、ユーザー自身が状況に応じた選択を行える対話型のインターフェースを提供する。

最終的には、栄養バランスの取れた食事の提供がより効率的かつ満足度の高いものとなり、ユーザーのニーズに最適化された献立作成が実現される。このシステムにより、献立作成にかかる時間や手間を軽減し、家庭内での食事作りを効率化することで、生活習慣病の予防や健康的な食生活の促進に貢献することが期待される。

§ 1.3 本論文の概要

本論文は次のように構成される。

第1章 本研究の背景として生活習慣病予防と献立作成の負担について触れ, 目的として多目的最適化とベクトル空間モデルを用いたシステムの提案について述べる。

第2章 本研究の基盤技術として, 多目的遺伝的アルゴリズム (NSGA-II) , TF-IDF を用いた嗜好学習, および Web アプリケーション開発技術について解説する。

第3章 提案する自動献立作成支援システムの設計として, アレルギーや疾患に対応したフィルタリング, AI スコアリング, および献立生成アルゴリズムの詳細について述べる。

第4章 提案システムの実装とインターフェースについて説明する。データベース構築, 最適化エンジンの実装, および対話型 Web インターフェースの機能について詳述する。

第5章 数値実験およびその考察を行う。仮想ユーザーを用いたシミュレーションにより, 献立の栄養バランス, 多様性, および学習による嗜好反映を確認し, 有効性を検証する。

第6章 本研究の成果を総括し, 結論と今後の課題について述べる。

UX を考慮した献立作成支援

§ 2.1 テキストマイニングによる嗜好情報の抽出

本研究では、栄養バランスだけでなく、ユーザーの嗜好や献立の意外性（UX 項目）を考慮した提案を行うために、自然言語処理技術を導入する。具体的には、レシピデータに含まれる「材料名」や「料理の特徴を示すタグ」を文書データとして扱い、これらをベクトル空間モデルによって定量化する。本節では、その手法として用いる TF-IDF 法およびコサイン類似度について説明する。

レシピの特徴を数値ベクトルとして表現するために、情報検索分野で広く利用されている重み付け手法である TF-IDF (Term Frequency-Inverse Document Frequency) [5] を用いる。これは、「単語の出現頻度 (TF)」と「逆文書頻度 (IDF)」という 2 つの指標を組み合わせることで、ある単語が特定の文書（レシピ）においてどれほど重要かを評価する手法である。

レシピデータにおいて単純な単語の出現頻度 (TF) のみを用いると、「塩」「醤油」「水」といった、ほとんどの料理に含まれる一般的な調味料が高いスコアを持ってしまい、料理ごとの個性を適切に表現できないという問題が生じる。そこで、多くの文書に出現する単語の重要度を下げる指標である IDF を組み合わせることで、その料理を特徴づける食材（例えば「トマト」「バジル」など）の重みを相対的に高めることが可能となる。

各指標の定義は以下の通りである。

TF (Term Frequency)

ある文書（レシピ） d において、特定の単語（食材など） t が出現する頻度を表す。あるレシピ内で頻繁に使われる単語ほど、そのレシピを特徴づける要素として重要視される。

IDF (Inverse Document Frequency)

ある単語 t が、全文書集合（全レシピデータ）の中でどれほど希少かを表す指標である。「塩」や「醤油」のように多くのレシピに共通して現れる一般的な単語は IDF 値が低くなり、逆に特定の料理にしか現れない特徴的な単語は IDF 値が高くなる。

あるレシピ d における単語 t の TF-IDF 値 $w_{t,d}$ は、以下の式 (2.1) によって定義される。

$$w_{t,d} = tf(t, d) \times idf(t) \quad (2.1)$$

ここで, $tf(t, d)$ はレシピ d における単語 t の出現回数であり, $idf(t)$ は全レシピ総数を N , 単語 t を含むレシピ数を $df(t)$ としたとき, 以下の式 (2.2) で算出される.

$$idf(t) = \log \frac{N}{df(t)} + 1 \quad (2.2)$$

本システムでは, この手法を用いて各レシピを多次元のベクトル空間上の点として表現する. これにより, 定性的な情報である「食材」や「風味」を, 計算可能な定量値へと変換することが可能となる.

次に, ベクトル化されたレシピデータを用いて, ユーザーの好みとレシピとの関連性を評価するために「コサイン類似度 (Cosine Similarity)」を用いる. ベクトル間の類似性を測る指標としてはユークリッド距離なども存在するが, レシピのような高次元かつ疎 (スパース) なデータにおいては, ベクトルの長さ (食材の総数や分量) の影響を受けやすいユークリッド距離よりも, ベクトルの方向 (食材の構成比率) の類似性を評価できるコサイン類似度が適しているとされる.

コサイン類似度は, ベクトル空間における 2 つのベクトルが成す角度の余弦 (コサイン) を計算することで, ベクトルの方向の類似性を評価する手法である. ベクトルの大きさに依存せず, 成分の構成比率 (レシピであれば食材の組み合わせ方) の類似性を測ることができるために, 文書分類や推薦システムに適している.

ユーザーの嗜好ベクトルを \vec{A} , 評価対象のレシピベクトルを \vec{B} とするとき, その類似度 $Similarity(\vec{A}, \vec{B})$ は以下の式 (2.3) で表される.

$$Similarity(\vec{A}, \vec{B}) = \cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (2.3)$$

ここで, $\vec{A} \cdot \vec{B}$ はベクトルの内積, $\|\vec{A}\|$ と $\|\vec{B}\|$ はそれぞれのベクトルのノルムを表す. 本研究では TF-IDF 値を用いているため各要素は非負となり, コサイン類似度は 0 (全く似ていない) から 1 (完全に一致) の範囲の値をとる.

本研究では, 上述したベクトル化と類似度計算を, 以下の 2 つのプロセスに応用し, 多目的最適化問題の目的関数として組み込んでいる.

1. 嗜好スコア (UX スコア) の算出

ユーザーが過去に高く評価したレシピ群から特徴ベクトルを抽出し, それらを合成することで「ユーザーの好みベクトル」を生成する. このベクトルと, データベース内の未提案レシピのベクトルとのコサイン類似度を計算し, 値が 1 に近いほどユーザーの好みに合致するレシピとして評価する.

2. 意外性 (Novelty) の算出

「定番レシピ (多くのユーザーが知っている一般的なレシピ)」のプロファイルベクトルと, 各レシピベクトルの類似度を計算する. ここでは逆に, 類似度が低い (0 に近い) ものを「意外性のあるレシピ」として評価する. これにより, ユーザーの好みを反映しつつも, マンネリ化を防ぐ新しい提案 (セレンディピティ) を実現する.

このように, テキストマイニング技術を用いて「好み」や「意外性」といった主観的な指標を数値化することで, 第 4 章で述べる NSGA-II による最適化プロセスにおいて, 栄養価やコストと同時にこれらを最適化することが可能となる.

§ 2.2 多目的最適化と NSGA-II

多目的最適化とは、「制約のある状況で複数の選択肢を組み合わせて結果を導き出す際に、その結果（目的関数）を可能な限り最小化または最大化すること」を指す。この手法の利点としては、自動化により結果が得られるまでの作業時間が短縮されることや、通常の方法では現実的な時間内で解決が難しい問題に対処できる点が挙げられる。

最適化問題の一種として「組み合わせ最適化問題」がある。本研究で開発する自動献立作成システムは、このカテゴリに属する。組み合わせ最適化問題とは、多くの選択肢の中から、与えられた制約のもとで評価値（目的関数）を最も良くする変数の組み合わせを求める問題である。

献立における制約条件としては、作成する日数、カロリー制限、特定の栄養素の最低摂取量などが考えられる。また、目的関数には調理時間の短縮、個人の嗜好を反映する度合いの向上、材料コストの削減などがある。

しかし、組み合わせ最適化を行う際には、目的関数同士がトレードオフの関係にある場合がある。トレードオフとは、一方の目的を満たそうとすると、別の目的の達成が難しくなる関係のことである。食事の例で言えば、高カロリーな食事をとることで満足度は上がるが、体重増加や栄養バランスの乱れを引き起こす可能性がある。逆に、健康を意識した食事制限をすれば満足度が下がる場合がある。このように、「高カロリーな食生活」と「健康」はトレードオフの関係になり得る。

目的関数がトレードオフの関係にある場合、单一の最適解を求めることが難しくなる。なぜなら、一方の目的関数を最適化しようとすると、もう一方の目的関数に悪影響を与える可能性があるからである。そのため、多目的最適化では、すべての目的関数を同時に満たす单一の解を求めるのではなく、パレート最適解と呼ばれる概念を用いる必要がある。

パレート最適解とは、ある目的関数を改善しようとすると、他の目的関数の評価が悪化してしまうような解のことであり、非劣解とも呼ばれる。逆に、他より優れた解が存在する場合、その解は劣解と呼ばれる。数理的には、ある解 x が他のいかなる解 y に対しても劣っていない（すなわち、すべての目的関数において y より悪い値をとらず、少なくとも一つの目的関数において y より良い値をとる）場合、その解 x はパレート最適であると定義される。これを「パレート支配」の概念を用いて定義すると、以下のようになる。

ある解 x が別の解 y をパレート支配する ($x \succ y$) とは、すべての目的関数 f_i において $f_i(x) \leq f_i(y)$ であり、かつ少なくとも一つの目的関数 f_j において $f_j(x) < f_j(y)$ である場合を指す（最小化問題の場合）。パレート最適解とは、実行可能領域内のどの解からも支配されない解のことである。

また、パレート最適解は通常1つに限定されることではなく、複数の解が存在するため、それらの集合を考えることになる。パレート最適解の集合の概念を図2.1に示す。図中の点はそれぞれの解が持つ目的関数の値をプロットしたものであり、左下の境界線上に並ぶ点がパレート最適解である。これらのパレート最適解を目的関数空間に配置したときに形成される曲線（または曲面）は「パレート最適フロント」と呼ばれ、実際の意思決定においては、このフロント上にある解の中から、ユーザーの状況や好みに応じて最終的な解を選択することになる。

さらに、最適化問題には一般的に「実行可能領域」という概念があり、これは制約条件を満たす領域のことを指す。この領域内には、複数の局所的最適解（ローカルミニマム）が

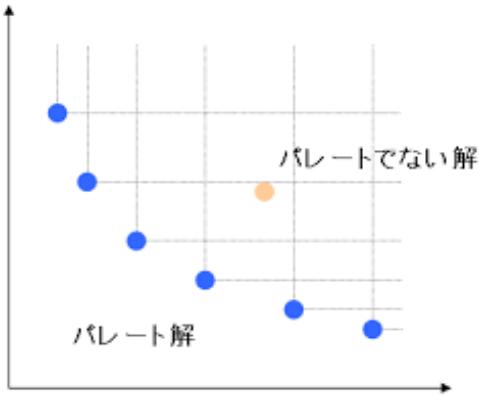


図 2.1: パレート最適解のイメージ

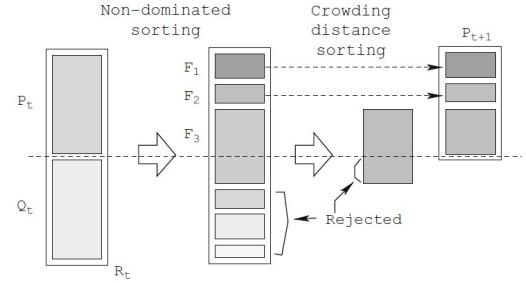


図 2.2: NSGA-II のアルゴリズムフロー

存在する場合がある。局所的最適解とは、その周囲のごく狭い範囲内では最も良い解であるが、実行可能領域全体で見た場合には必ずしも最適とは限らない解のことである。一方、大域的最適解（グローバルミニマム）とは、制約を満たしつつ、実行可能領域全体で最も良い解であることが保証されている解である。多目的最適化においては、局所的なパレートフロントに捕らわれず、真のパレートフロント（大域的なパレート最適解集合）を発見することが重要となる。

ここで、多目的最適化問題の定式化を行う。

多目的最適化問題の定式化

多目的最適化問題は、 $x = (x_1, x_2, \dots, x_N)$, $f_i(x)$, $i = 1, 2, \dots, n$ を目的関数として、 $g_k(x)$, $k = 1, 2, \dots, m$ を制約条件式とすると、

$$\underset{x}{\text{minimize}} \quad \{f_1(x), f_2(x), \dots, f_n(x)\} \quad (2.4)$$

$$\text{subject to} \quad g_k(x) \leq 0 \quad k = 1, 2, \dots, m \quad (2.5)$$

のように定式化することができる。式 (3.1) で与えられるベクトル関数を式 (3.2) の制約条件を満たした状態で最適化する問題を多目的最適化問題と呼ぶ。

次に、本研究で採用する最適化手法である遺伝的アルゴリズムおよび NSGA-II について説明する。

遺伝的アルゴリズム (Genetic Algorithm: GA)

遺伝的アルゴリズムとは、生物の進化過程を模倣して近似解を探索するメタヒューリティクスアルゴリズムである [9]。解の候補を遺伝子を持つ「個体」として表現し、集団（個体群）に対して「選択」「交叉」「突然変異」といった操作を繰り返し適用することで、環境（評価関数）に適応したより良い解を探索する。一般的な GA の流れは以下の通りである。

1. **初期化:** 初期個体群をランダムに生成する。
2. **評価:** 各個体の適応度（目的関数の値など）を評価する。

3. **選択 (Selection):** 適応度の高い個体を優先的に次世代の親として選ぶ. ルーレット選択やトーナメント選択などが用いられる.
4. **交叉 (Crossover):** 選ばれた親個体同士の遺伝子を組み合わせて, 新しい個体(子)を生成する. これにより, 親の持つ良い形質を継承しつつ, 新たな解の候補を生み出す.
5. **突然変異 (Mutation):** 一定の確率で遺伝子の一部をランダムに変化させる. これにより, 局所解からの脱出や, 集団の多様性維持を図る.
6. **終了判定:** 一定世代数に達するか, 十分な精度の解が得られるまで 2~5 を繰り返す.

この手法は, 勾配情報(微分の値)を必要としないため, 目的関数が微分不可能な場合や, 変数が離散値をとる組み合わせ最適化問題に対しても適用可能であるという利点を持つ. また, 多点探索(複数の解を同時に探索)を行うため, 局所解に陥りにくく, 大域的な最適解を発見しやすいという特徴もある.

NSGA-II (Non-dominated Sorting Genetic Algorithm-II)

NSGA-II は, Deb らによって 2002 年に提案された多目的遺伝的アルゴリズムであり [11], 従来の NSGA の計算コストの高さやエリート保存の欠如といった問題点を改善した手法である. 本手法は, パレート最適解に近い解を効率的に探索できることから, 多くの多目的最適化問題で標準的に利用されている. NSGA-II の処理の流れを図 2.2 に示す. 主な特徴は以下の 3 点である.

1. **高速な非優越ソート (Fast Non-dominated Sorting):**

集団内の全個体を, 優越関係に基づいてランク分けする手法. 他のどの個体にも劣っていない(支配されていない)個体をランク 1(パレートフロント)とし, ランク 1 を除いた中で支配されていない個体をランク 2 とする. これを繰り返すことで集団を層別化し, パレート最適解に近い個体を優先的に残すことができる. このアルゴリズムの計算量は $O(MN^2)$ (M は目的関数数, N は個体数) であり, 従来の $O(MN^3)$ に比べて大幅に高速化されている.

2. **混雑距離 (Crowding Distance):**

解の多様性を維持するための指標. 同一ランク内での個体の優劣を決める際に用いられる. 目的関数空間において, 隣接する個体との距離(混雑度)を計算し, 密集していない領域にある個体を優先的に選択する. これにより, 局所的な解への集中を防ぎ, パレートフロント全体に広く分布した解を得ることが可能となる. 混雑距離 $I[i]_{distance}$ は, 各目的関数 m についてソートされた個体群において, 個体 i の両隣の個体 $i+1$ と $i-1$ の目的関数値の差の総和として, 以下の式で定義される.

$$I[i]_{distance} = \sum_{m=1}^M \frac{f_m(i+1) - f_m(i-1)}{f_m^{max} - f_m^{min}} \quad (2.6)$$

ここで, f_m^{max} と f_m^{min} は目的関数 m の最大値と最小値である.

3. **エリート保存戦略 (Elitist Strategy):**

親個体と, そこから生成された子個体を合わせた集団(サイズ $2N$)の中から, 非優越ソートと混雑距離に基づいて優秀な個体 N 個を選抜して次世代に残す戦

略. これにより, 探索過程で得られた優れた解が交叉や突然変異によって失われることを防ぎ, 収束性能を向上させている.

本研究では, Python ライブリヤーである pymoo を用いて NSGA-II を実装し, 献立の組み合わせ最適化を行う.

§ 2.3 適合性フィードバックと Rocchio アルゴリズム

本システムでは, ユーザーに提案した献立に対する評価 (フィードバック) を受け取り, 次回の提案内容をよりユーザーの好みに近づけるための仕組みとして「適合性フィードバック」を採用している. 適合性フィードバックとは, 情報検索システムにおいて, 検索結果に対するユーザーの適合判定 (「この文書は求めていたものだ」「これは違う」といった評価) を利用して, 検索クエリ (検索条件) を自動的に修正し, 再検索を行う手法である.

本研究では, この概念を献立推薦に応用する. 具体的には, ユーザーの「好み」をベクトルとして表現し, ユーザーが「良い」と評価したレシピのベクトルを足し合わせ, 「悪い」と評価したレシピのベクトルを引くことで, 好みベクトルを更新していく. このベクトル更新の計算には, 情報検索の分野で標準的に用いられている Rocchio アルゴリズム [12] を使用する.

Rocchio アルゴリズムは, 適合性フィードバックを実現するための代表的なアルゴリズムであり, 1971 年に J.J. Rocchio によって提案された. このアルゴリズムでは, 元のクエリベクトルに対して, 適合文書のベクトルを加算し, 不適合文書のベクトルを減算することで, 新しいクエリベクトルを生成する.

本研究における Rocchio アルゴリズムを用いたユーザー嗜好ベクトル (プロファイルベクトル) の更新式は, 以下の式 (2.7) で表される.

$$\vec{Q}_{new} = \alpha \vec{Q}_{old} + \beta \frac{1}{|D_r|} \sum_{\vec{d} \in D_r} \vec{d} - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d} \in D_{nr}} \vec{d} \quad (2.7)$$

ここで, 各変数の定義は以下の通りである.

- \vec{Q}_{new} : 更新後のユーザー嗜好ベクトル (新しいプロファイル)
- \vec{Q}_{old} : 更新前のユーザー嗜好ベクトル (現在のプロファイル)
- D_r : ユーザーが「適合 (好き)」と評価したレシピ文書の集合
- D_{nr} : ユーザーが「不適合 (嫌い)」と評価したレシピ文書の集合
- \vec{d} : 各レシピの特徴ベクトル (TF-IDF ベクトル)
- $|D_r|, |D_{nr}|$: それぞれの集合に含まれるレシピ数
- α, β, γ : 各項の重みを調整するパラメータ

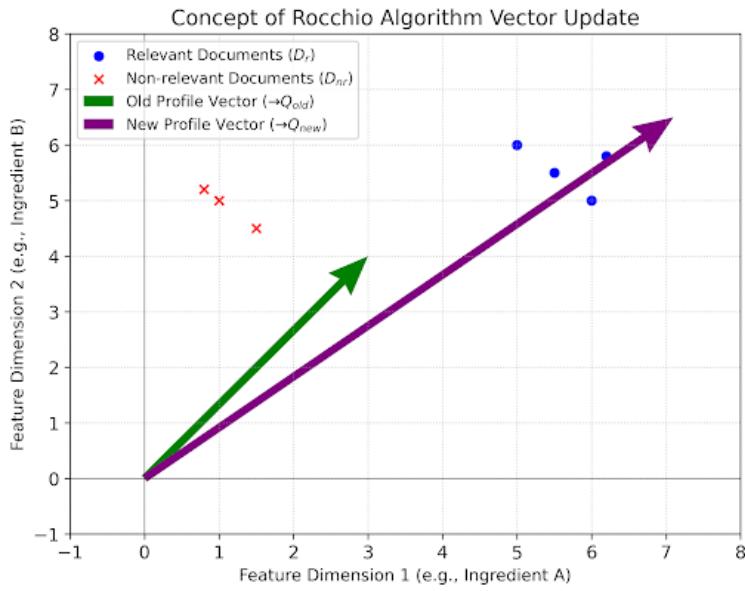


図 2.3: Rocchio アルゴリズムによる嗜好ベクトルの更新概念図

図 2.3 に、このアルゴリズムによるベクトル更新の概念図を示す。初期の嗜好ベクトル（緑色の矢印）が、ユーザーが高評価したレシピ群（青色の点）の重心方向に引き寄せられ、逆に低評価したレシピ群（赤色の点）の重心からは遠ざかる方向に修正され、新しい嗜好ベクトル（紫色の矢印）が生成される様子を表している。

式 (2.7) におけるパラメータ α, β, γ は、それぞれ以下の意味を持つ。

α (慣性係数) 現在の嗜好プロファイルをどれだけ維持するかを決定する。値が大きいほど、過去の好みが保存され、急激な変化が抑制される。

β (適合性重み) ユーザーが高評価したレシピ（正のフィードバック）の影響度を決定する。

γ (不適合性重み) ユーザーが低評価したレシピ（負のフィードバック）の影響度を決定する。

一般的な情報検索のタスクでは、 $\alpha = 1.0, \beta = 0.75, \gamma = 0.15$ 程度の値が用いられることが多い。しかし、食事推薦の文脈においては、「嫌いなものを避ける」ことよりも「好きなものを積極的に提案する」ことの方が UX 向上に寄与しやすいと考えられる。例えば、「唐揚げが好き」という情報は献立決定に直接役立つが、「セロリが嫌い」という情報は、セロリを含まない膨大なレシピすべてが候補となるため、特徴の絞り込みとしては弱い場合がある。そのため本システムでは、 $\beta > \gamma$ の関係を維持しつつ、正のフィードバックを重視したパラメータ設定を行う。

また、ベクトル加算を繰り返すとベクトルのノルム（大きさ）が肥大化し、計算上の不都合が生じる可能性がある。そのため、更新後のベクトル \vec{Q}_{new} に対しては、以下の式 (2.8) を用いて正規化を行い、常に単位ベクトルとして扱う。

$$\vec{Q}_{final} = \frac{\vec{Q}_{new}}{\|\vec{Q}_{new}\|} \quad (2.8)$$

実際のシステム運用において、Rocchio アルゴリズムは以下の手順で適用される。

1. 初期プロファイルの生成:

初回利用時など、学習データが存在しない段階では、嗜好ベクトル \vec{Q}_{old} はゼロベクトル、もしくは一般的な人気レシピの平均ベクトルとして初期化される。

2. 献立の提案と評価:

システムが提案した献立に対し、ユーザーは5段階評価（1: 全く良くない～5: とても良い）を行う。

3. レシピの分類:

ユーザーの評価値に基づき、レシピを以下の3つのグループに分類する。

- **適合レシピ (D_r):** 評価値が4または5のレシピ。ユーザーの好みを反映しているとみなす。
- **不適合レシピ (D_{nr}):** 評価値が1または2のレシピ。ユーザーの好みに合わないとみなす。
- **中立レシピ:** 評価値が3のレシピ。特徴更新には使用しない。

4. 特徴ベクトルの更新:

分類された D_r と D_{nr} に含まれるレシピのTF-IDFベクトルを用い、式(2.7)に従って嗜好ベクトルを更新する。これにより、「鶏肉」や「トマト」といった、高評価レシピに共通して含まれる食材の次元の重みが増加し、逆に低評価レシピに含まれる特徴の重みは減少する。

5. 再提案:

更新された嗜好ベクトル \vec{Q}_{new} を用いて、データベース内の全レシピとのコサイン類似度を再計算し、スコアが高い順に次回の献立候補を選出する。

このサイクルを繰り返すことで、システムは利用回数を重ねるごとにユーザーの潜在的な好みを学習し、より精度の高いパーソナライズされた献立提案が可能となる。

アルゴリズムの一般的応用と関連研究

§ 3.1 多目的遺伝的アルゴリズムの応用分野

本研究で採用した NSGA-II をはじめとする多目的遺伝的アルゴリズム (Multi-Objective Genetic Algorithm: MOGA) は、その高い探索能力と汎用性から、工学、経済学、社会基盤システムなど、トレードオフ関係にある複数の目的を同時に最適化する必要がある極めて幅広い分野で応用されている。本節では、代表的な応用事例として「航空・宇宙工学」、「エネルギー・社会インフラ」、「物流・サプライチェーン」、「金融工学」の4つの領域を取り上げ、それぞれの問題構造と進化計算の有用性について詳述する。

航空・宇宙および自動車工学における構造最適化

工学設計の分野において、多目的最適化は製品の性能を決定づける最も重要なプロセスの一つである。特に航空機や自動車の設計においては、安全性、環境性能、コストといった相反する要件を高い次元でバランスさせる必要がある。

航空機の主翼設計はその典型例である。図3.1に示すように、主翼の設計には「構造質量の最小化（燃費向上）」と「空気抵抗の最小化（速度・燃費向上）」という2つの主要な目的が存在する。一般に、主翼の強度を高めるために構造部材を厚くしたり、リブやスパー（補強材）を増やしたりすると、質量が増加してしまう。逆に、軽量化を優先して部材を薄くすると、空気抵抗を減らすための理想的な流線型を維持することが構造力学的に困難になる場合がある。このように、一方の性能を向上させようとすると他方が悪化するというトレードオフ関係が顕著であるため、設計者は単一の最適解ではなく、パレート最適解集合（パレートフロント）を導出し、その中からミッション要件に合致した妥協点を見出す必要がある。NSGA-IIは、設計変数が連続値（翼の厚みや角度）と離散値（補強材の本数）の混合となる場合でも効率的に解を探索できるため、この分野で標準的に利用されている[13]。

また、自動車の車体設計においても、「衝突安全性の最大化」と「車体重量の最小化」の同時最適化に応用されている。衝突時のエネルギー吸収効率を高めるためには複雑なフレーム構造や高強度な部材が必要となるが、これは重量増を招き、燃費や加速性能を悪化させる。進化計算を用いることで、数万通りの部材配置パターンの中から、軽量でありながら高い衝突安全性能を持つパレート最適な構造案を自動的に生成することが可能となっている。

エネルギー管理とスマートグリッド

持続可能な社会の実現に向けたエネルギー分野においても、多目的最適化の重要性は増している。特に、再生可能エネルギー（太陽光、風力など）を導入した電力網（スマートグリッド）の運用計画においては、「発電・運用コストの最小化」と「環境負荷（CO₂排出

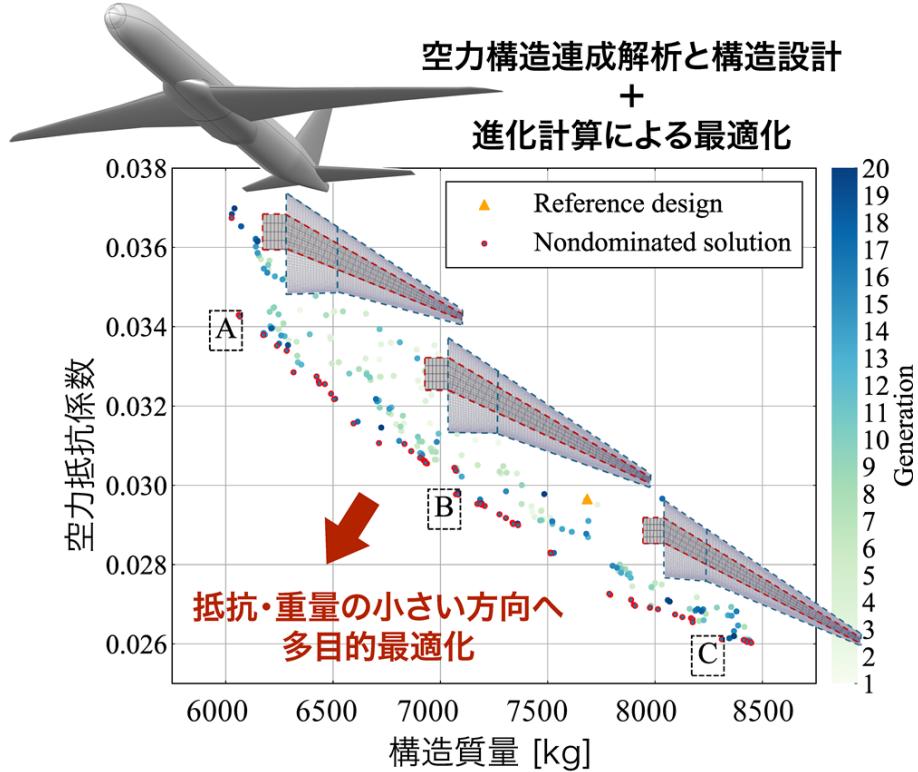


図 3.1: 航空機主翼設計における多目的最適化の例（構造質量と空気抵抗係数のトレードオフ）[13]

量) の最小化」が主要な目的となる。

火力発電は安定した電力供給が可能であるが、燃料コストがかかり CO₂ 排出量も多い。一方、太陽光発電や風力発電は運転コストや環境負荷は低いが、天候に左右されるため供給が不安定である。電力会社やグリッド管理者は、これらの電源を最適に組み合わせ（ベストミックス），電力需要を満たしつつ、コストと環境負荷のバランスをとる必要がある。[14]。さらに、蓄電池の充放電スケジュールや、デマンドレスポンス（需要制御）といった要素も加わるため、問題は非常に複雑な組み合わせ最適化問題となる。NSGA-II 等のアルゴリズムは、このような大規模かつ非線形な制約を持つ問題に対しても、実用的な時間内で多様な運用シナリオ（パレート解）を提示できるため、エネルギー・マネジメントシステム（EMS）の中核技術として活用されている。

物流・サプライチェーンの最適化

物流分野においては、配送計画問題（Vehicle Routing Problem: VRP）への応用が盛んである。配送業務においては、「総配送距離・時間の最小化（コスト削減）」と「配送遅延の最小化（サービスレベル維持）」あるいは「ドライバーの労働負荷の平準化」といった複数の目的が存在する。

例えば、配送トラックの台数を減らして積載率を高めようすると、一台あたりの配送ルートが長くなり、指定された配送時刻に遅れるリスクが高まる。逆に、配送品質を重視して余裕を持ったスケジュールを組むと、必要な車両数やドライバー数が増加し、コストが増大する。また、近年では「CO₂ 排出量の削減」も重要な目的関数として追加される傾向にある。このような多目的 VRP に対し、遺伝的アルゴリズムを用いることで、都市部の複雑な道路網や交通渋滞、配送先の指定時間帯（タイムウィンドウ）といった厳しい制約

条件を満たしつつ、コストとサービス品質のバランスが取れた配送ルートを自動生成するシステムが実用化されている。さらに、土木計画の分野においても、遺伝的アルゴリズムは離散的な組み合わせ最適化問題の有効な解法として認識されている。有村ら [15] は、社会基盤整備における最適化だけでなく、環境に適応して学習するシステムとしての GA の可能性を論じており、本研究が目指す「ユーザーの嗜好に適応する献立作成」と数理的な共通点が見られる。

金融工学におけるポートフォリオ最適化

金融分野では、投資配分の決定問題（ポートフォリオ最適化）において多目的最適化が古典的かつ中心的な役割を果たしている。ハリー・マーコウイッツが提唱した現代ポートフォリオ理論に基づき、投資家は「期待リターンの最大化」と「投資リスク（分散や標準偏差）の最小化」という2つの相反する目的を追求する。

リスクを抑えようとして国債などの安全資産ばかりに投資すれば、期待できるリターンは低くなる。高いリターンを狙って株式などの変動資産に集中投資すれば、元本割れのリスクは増大する。この「ハイリスク・ハイリターン」の関係を数学的に表現し、投資家が許容できるリスクレベルに応じた最高のリターン（効率的フロンティア）を導出するためには、最適化アルゴリズムが用いられる [16]。現実の市場においては、売買単位の制約（整数制約）や、特定の銘柄への投資比率制限（基数制約）など、解析的な手法では解くことが難しい複雑な制約条件が存在する。進化計算を用いたアプローチは、こうした制約を柔軟に扱うことができ、かつ局所解に陥ることなく大域的なパレートフロントを探索できるため、ロボアドバイザーや機関投資家の資産運用システムに実装されている。

§ 3.2 ベクトル空間モデルと推薦システム

ベクトル空間モデルと TF-IDF

テキストマイニングや情報検索の分野において、非構造化データである「文書」を定量的に扱うための標準的な手法として、ベクトル空間モデル（Vector Space Model: VSM）が広く利用されている。このモデルでは、文書を単語（Term）の集合として捉え、各単語の重要度を成分とする多次元ベクトルとして表現する。対象となる文書集合に含まれる全単語数を N とした場合、一つの文書は N 次元のベクトル空間上的一点（あるいは原点からの矢印）として配置される。これにより、これまで定性的にしか扱えなかった文書間の類似性や、検索クエリとの関連度を、ベクトル間の距離や角度として数学的に計算することが可能となる [17]。

ベクトル化の際、単に単語の出現回数を数えるだけでは、頻出する一般的な単語（「て」「に」「を」「は」などの助詞や、レシピにおける「塩」「水」など）の影響が大きくなりすぎ、文書ごとの特徴を正しく捉えられないという問題がある。そこで、各単語の重要度を適切に評価するための重み付け手法として、TF-IDF（Term Frequency-Inverse Document Frequency）法が一般的に用いられる。

TF-IDF は、ある文書内での単語の出現頻度（TF）と、全文書集合におけるその単語の希少性（IDF）を掛け合わせた指標であり、以下の式で定義される。

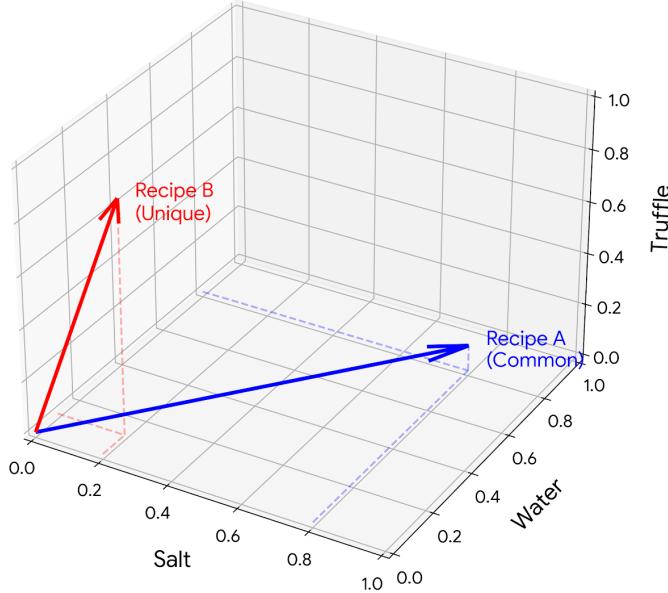


図 3.2: ベクトル空間モデルにおけるレシピの表現例 (TF-IDF による重み付け)

$$\text{TF-IDF}_{t,d} = \text{TF}_{t,d} \times \text{IDF}_t \quad (3.1)$$

$$\text{IDF}_t = \log \frac{N}{df_t} \quad (3.2)$$

ここで, $\text{TF}_{t,d}$ は文書 d における単語 t の出現頻度, N は全文書数, df_t は単語 t を含む文書の数を表す. IDF 値は, 多くの文書に出現する単語ほど分母の df_t が大きくなるため, 値が小さくなる (0 に近づく) 性質を持つ. 逆に, 特定の文書にしか現れない希少な単語ほど IDF 値は大きくなる.

図 3.2 に, 本研究の対象であるレシピデータにおける TF-IDF の概念図を示す. この空間は, 「Salt (塩)」「Water (水)」「Truffle (トリュフ)」という 3 つの単語を軸 (次元) として構成されている.

- **Recipe A (Common):** 一般的なレシピの例である. 「塩」や「水」は多くの料理で使用されるため出現頻度 (TF) は高いが, 多くのレシピに含まれるため IDF 値は低く抑えられる. 結果として, 図中の青い矢印のように, 特定の軸に突出することなく, 原点に近い, あるいは平均的な方向にベクトルが伸びる傾向がある.
- **Recipe B (Unique):** 特徴的なレシピの例である. 「トリュフ」のような高級食材や特定の料理にしか使われないスパイスは, 出現頻度 (TF) 自体は少なくとも, 全レシピ中での希少性が高いため IDF 値が非常に大きくなる. そのため, 図中の赤い矢印のように, 「Truffle」軸の方向に強く引き伸ばされたベクトルとなり, 空間上で他のレシピと明確に区別することが可能となる.

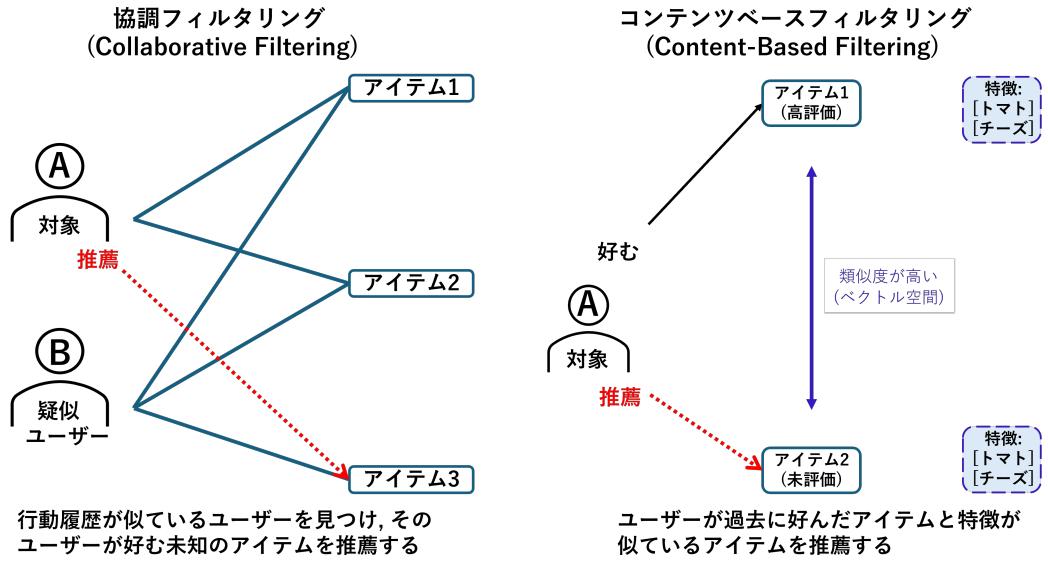


図 3.3: 協調フィルタリングとコンテンツベースフィルタリングの比較

このように、TF-IDF を用いることで、ありふれた食材の影響を抑えつつ、その料理を特徴づける食材や調味料を強調したベクトル表現を得ることができる。本研究においても、レシピの食材や特徴をベクトル化する際の基盤技術として、この TF-IDF 法を採用している。

類似度計算と推薦システム

ベクトル空間モデルによって数値化されたデータは、推薦システム (Recommender System) の中核技術として応用されている。図 3.3 に示すように、推薦システムは主に、「協調フィルタリング (Collaborative Filtering)」と「コンテンツベースフィルタリング (Content-Based Filtering)」の 2 つに大別される [18]。

図 3.3 左側の協調フィルタリングは、Amazon などの EC サイトでよく見られる「この商品を買った人はこんな商品も買っています」というアプローチである。ユーザーの過去の購買履歴や評価データを蓄積し、行動パターンが似ている他者（類似ユーザー）を見つける、その他者が好むアイテムを推薦する。この手法は、アイテムの中身（内容）を分析する必要がないという利点がある一方で、新規ユーザーや新規アイテムに対して十分なデータがないと推薦精度が下がる「コールドスタート問題」が課題となる。

一方、図 3.3 右側のコンテンツベースフィルタリングは、アイテム自体の特徴（コンテンツ）に着目する手法である。例えば、映画推薦であれば「ジャンル」や「監督」、料理レシピであれば「食材」や「味付け」といった属性をベクトル化し、ユーザーが過去に好んだアイテムとベクトル空間上で距離が近い（類似度が高い）アイテムを推薦する。ベクトル間の類似度判定には、コサイン類似度 (Cosine Similarity) が標準的に用いられる。コサイン類似度は、2 つのベクトルが成す角度の余弦（コサイン）をとることで、ベクトルの大きさ（文章の長さや単語数）に依存せず、方向（成分の構成比率）の類似性を測ることができるため、テキストやレシピの比較に適している。

本研究では、個人の嗜好に特化した献立提案を行うため、他者のデータを必要とせず、ユーザー個人の評価履歴のみから学習が可能なコンテンツベースのアプローチを主軸としている。

§ 3.3 パーソナライズとフィードバック学習

本節では、ユーザー個人の嗜好に特化した献立提案を実現するための核心技術である「適合性フィードバック」と、その具体的なアルゴリズムについて詳述する。

適合性フィードバックの理論的枠組み

コンテンツベースフィルタリングにおいて、推薦精度を決定づける最大の要因は、システム内部で保持される「ユーザープロファイル (User Profile)」が、現実のユーザーの関心や意図をどれだけ正確に反映しているかにある。本システムにおいて、ユーザープロファイルとは単なる静的な属性情報ではなく、レシピの特徴（食材、味付け、調理法など）と同じベクトル空間上に存在する嗜好ベクトルとして表現される。しかし、システム利用開始初期の段階では、このプロファイルは空であるか、あるいは全ユーザーの平均的な傾向を示しているに過ぎず、個人の詳細かつ微細な嗜好を捉えきれていない（コールドスタート問題）。

さらに、ユーザーの食の好みは不变の属性ではなく、極めて動的な性質を持つ。例えば、夏には酸味のあるさっぱりした料理が好まれ、冬には濃厚で温かい料理が好まれるといった「季節性」、あるいはその日の体調や気分による「短期的な嗜好の変化」、さらには加齢や生活環境の変化に伴う「長期的な嗜好の変遷 (Concept Drift)」などが挙げられる。したがって、固定的なプロファイルではユーザーの満足度を持続的に維持することは困難である。

この「システムが持つユーザー像（内部モデル）」と「実際のユーザーの意図（真の要求）」とのギャップを埋めるための手法として、情報検索の分野では古くから「適合性フィードバック (Relevance Feedback)」が研究されてきた [17]。これは、一度の検索や推薦で完結するのではなく、システムが提示した結果に対するユーザーの反応を取り込み、クエリ（要求）を反復的に修正・最適化していくプロセスである。

適合性フィードバックのアプローチには、大きく分けて二つの種類が存在する [18]。一つは、ユーザーが「好き／嫌い」や「5段階評価」といった明確な評価を下す「明示的フィードバック (Explicit Feedback)」である。もう一つは、閲覧時間、クリック数、購入履歴などの行動ログからユーザーの関心を推定する「暗黙的フィードバック (Implicit Feedback)」である。ウェブ検索や動画推薦などの分野では、ユーザー負担の少ない暗黙的フィードバックが主流となりつつあるが、食の推薦においては以下の理由から明示的フィードバックが優れていると考えられる。

第一に、レシピにおける「閲覧」は必ずしも「喫食」や「満足」を意味しないためである。例えば、レシピを長時間閲覧していたとしても、それは「美味しいだから」ではなく「手順が複雑で理解するのに時間がかかっただけ」である可能性があり、暗黙的なデータにはノイズが多く含まれる。第二に、食の推薦は情報の消費とは異なり、実際に食材を購入し調理するという「物理的・時間的コスト」をユーザーに強いる。誤った推薦（ユーザーが嫌いな味や食べられない食材の提案）が行われた場合の不利益が大きいため、曖昧な推定よりも、確実な評価に基づく学習が求められる。特にアレルギー食材の混入などは健康被害に直結するため、システムはユーザーの拒絶反応を明確に学習する必要がある。

以上の理由から、本研究のような献立作成支援においては、ユーザーが「食べたい」と感じるかどうかの主観的な判断を直接システムに伝えることが重要であり、明示的フィー

ドバックを採用することで、高精度かつ安全なパーソナライズを実現する。

Rocchio アルゴリズムの数理と幾何学的解釈

ベクトル空間モデルにおける適合性フィードバックの標準的な手法として、本研究では「Rocchio アルゴリズム」を採用する。このアルゴリズムの基本的なアイデアは、ベクトル空間上において、ユーザープロファイルベクトルを「適合 (Relevant)」と判定された文書群の重心に近づけ、「不適合 (Non-relevant)」と判定された文書群の重心から遠ざけることである。これにより、ユーザーの関心領域の中心へとクエリベクトルを移動させ、検索（推薦）精度を向上させる。

第 m 回のフィードバック後の新しいユーザープロファイルベクトル \vec{q}_m は、以下の式によって定義される。

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_k \in D_{nr}} \vec{d}_k \quad (3.3)$$

ここで、各項およびパラメータは以下の意味を持つ。

- \vec{q}_0 : 初期クエリベクトル（更新前のプロファイル）。
- D_r : ユーザーによって「高評価（好き）」と判定されたレシピベクトルの集合。 $|D_r|$ はその要素数である。
- D_{nr} : ユーザーによって「低評価（嫌い）」と判定されたレシピベクトルの集合。
- α, β, γ : 学習の挙動を制御する重みパラメータ。

パラメータ α は「過去の学習履歴の維持」を、 β は「正のフィードバックの取り込み」を、 γ は「負のフィードバックの排除」を司る。一般的に情報検索の分野では $\alpha = 1.0, \beta = 0.75, \gamma = 0.15$ 程度の値が用いられることが多いが [17]、本研究のようなレシピ推薦においては、「嫌いな食材（アレルギーや嗜好）」が含まれる場合のペナルティを重くすべきであるため、 γ の値を相対的に高く設定するなどの調整が必要となる。

幾何学的には、この更新式は第 2 章で示した概念図と同様に、プロファイルベクトルを、好まれた料理の特徴語（例えば「トマト」「パスタ」）の次元方向へ伸ばし、嫌われた料理の特徴語（例えば「唐辛子」「レバー」）の次元方向へ縮める操作に他ならない。これを繰り返すことで、ベクトルはユーザーの好みの「重心」へと収束していく。

本システムにおける学習サイクルと課題への対処

本研究の献立作成システムでは、上述の理論を図 3.4 に示すインタラクティブな学習サイクルとして実装している。

具体的なプロセスは以下の通りである。

1. **Menu Proposal (提案)**: NSGA-II により、栄養価・コスト・現在の嗜好適合度を最適化したパレート最適解（献立案）を生成し、ユーザーに提示する。
2. **User Evaluation (評価)**: ユーザーは提示された献立に対し、主観的な評価（5 段階評価など）を行う。このフェーズは、ユーザーがシステムに対して自身の好みを教える重要なステップである。

献立計画における対話型学習サイクル

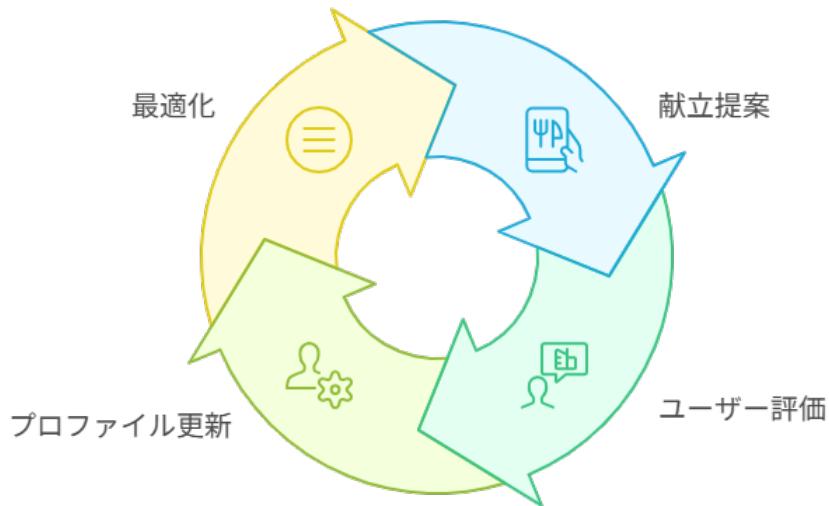


図 3.4: 本研究におけるインタラクティブな学習サイクル（献立提案と評価のループ）

3. **Profile Update (学習):** ユーザーからの評価結果に基づき, 式 (3.3) を用いてユーザーの嗜好ベクトル \vec{q} を更新する.
4. **Re-optimization (再最適化):** 更新された嗜好ベクトル \vec{q} と, 各レシピベクトル \vec{d} のコサイン類似度 $sim(\vec{q}, \vec{d})$ を新たな目的関数の一つとして設定し, 次回の献立生成を行う.

このサイクルを回す上で, 推薦システム特有の課題である「コールドスタート問題 (Cold Start Problem)」への対処も考慮する必要がある [18]. システム利用開始直後は D_r や D_{nr} が空集合であるため, 学習が機能しない. そこで本研究では, 初期状態において一般的な人気レシピの平均ベクトルを \vec{q}_0 として設定するか, あるいは初回起動時に簡易的なアンケート（「和食派? 洋食派?」など）を行い, 大まかな初期ベクトルを生成するアプローチをとることで, 初期段階から一定の推薦品質を担保する設計としている.

また, 長期的な運用においては, ユーザーの好みが変化する「嗜好のドリフト (Concept Drift)」が発生する可能性がある. これに対しては, 過去のフィードバックの影響力 (α 項) を時間経過とともに減衰させることで, 直近の好みをより強く反映させる拡張も可能である.

提案手法

本章では、第2章で述べた献立作成の多目的最適化問題と、第3章で述べたベクトル空間モデルによる嗜好学習を統合し、ユーザーのUX (User Experience) を最大化する献立提案システムの具体的な構成と実装詳細について述べる。

§ 4.1 システムアーキテクチャとデータ基盤

本研究で構築したシステムの全体処理フローを図4.1に示す。本システムは、Webアプリケーションとして実装されており、ユーザー情報の入力から始まり、データベースの読み込み、フィルタリング、AIモデルによる学習とスコアリング、そしてNSGA-IIによる最適化を経て、最終的にユーザーによる評価をフィードバックする一連のサイクルで構成される。

処理シーケンスの詳細

図4.1に示した各処理ステップの詳細は以下の通りである。

Step 1: ユーザー情報入力 (User Input)

システムはまず、ユーザー定義集合 $U = \{U_{info}, U_{const}, U_{pref}, U_{obj}\}$ を受け付ける。これには、身体情報 (U_{info})、アレルギーや疾患などの制約条件 (U_{const})、初期の好みキーワード (U_{pref})、および「コスト重視」や「時間重視」といった最適化の目的 (U_{obj}) が含まれる。これらの入力値は、後続の最適化プロセスにおける制約条件式 $G(x) \leq 0$ の閾値を決定する重要なパラメータとなる。

Step 2: データベース読み込みと前処理 (DB Loading & Preprocessing)

全レシピ集合 $R_{all} = \{r_1, \dots, r_N\}$ をデータベースからメモリ上に展開する。本システムでは、ディスクI/Oのボトルネックを解消するため、列指向フォーマットであるParquet形式を採用し、数千件のレシピデータと特徴ベクトルを高速に読み込む。また、読み込み時にはデータクレンジングが実行される。後述する画像リンクの自動修復や、欠損値の補完処理を経たクリーンなデータのみが最適化エンジンに渡される。

Step 3: フィルタリング (Filtering)

読み込んだ全レシピに対し、ユーザーの制約に基づいたフィルタリングを行う。具体的には、アレルギー食材 A を含むレシピや、疾患 D に基づく栄養基準を満たさない

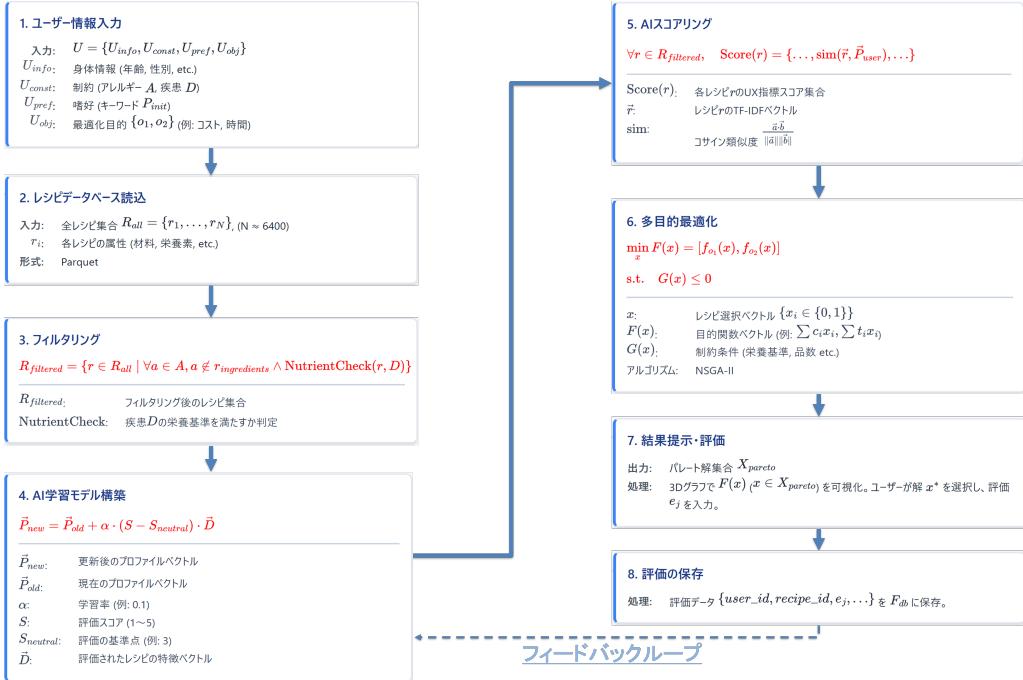


図 4.1: 提案システムの詳細処理フローと数理モデル

レシピを除外し、安全な候補集合 $R_{filtered}$ を生成する。

$$R_{filtered} = \{r \in R_{all} \mid \text{Safe}(r, A) \wedge \text{NutrientCheck}(r, D)\}$$

Step 4: AI学習モデル構築 (Model Construction)

ユーザーの過去の評価データに基づき、Rocchio アルゴリズムを用いて嗜好プロファイルベクトル \vec{P}_{user} を更新・構築する。これにより、ユーザーの好む食材や味付けの傾向を数学的なベクトルとして表現する。

Step 5: AIスコアリング (AI Scoring)

候補レシピ集合 $R_{filtered}$ の各レシピに対し、ユーザープロファイルとのコサイン類似度を計算し、これを UX スコア (嗜好適合度) として付与する。また、ジャンル (和食・洋食・中華) ごとの適合度も算出する。

Step 6: 多目的最適化 (Optimization)

NSGA-II アルゴリズムを用い、コスト、調理時間、UX スコアを目的関数とする組み合わせ最適化を行う。数万通りの組み合わせの中から、パレート最適解 (劣後しない解の集合) を探索する。

Step 7: 結果提示・評価 (Presentation)

導出されたパレート解集合を可視化してユーザーに提示する。ユーザーはその中から自身の状況に最適な献立を選択し、実際に喫食した後に評価を入力する。提示画面には、各解の特性 (コスト重視型、バランス型など) が直感的に分かるネットワークグラフが描画される。

表 4.1: 本システムの開発環境と使用ライブラリ

項目	詳細
PC Model	MacBook Air (M1)
OS	macOS Sonoma 14.2
Language	Python 3.9.13
Framework	Flask 3.0.0
主要ライブラリ	
pymoo	多目的最適化
scikit-learn	TF-IDF, 類似度
pandas	データ処理
BeautifulSoup4	スクレイピング

表 4.2: レシピデータベースの主要カラム定義

カラム名	説明
recipe_id	レシピ識別子
title	料理名
ingredients	食材リスト
time	調理時間 (分)
cost	推定コスト (円)
nutrients	栄養価 (25 項目)
vector	TF-IDF ベクトル

Step 8: 評価の保存 (Feedback)

ユーザーによる評価データをデータベースに保存し、次回の Step 4 における学習データとして利用する (フィードバックループ)。

開発環境とデータベース構造

本システムの開発および数値実験を行った環境を表 4.1 に、構築したレシピデータベースのスキーマ定義を表 4.2 に示す。Web フレームワークには「Flask」を採用し、バックエンドの科学計算ライブラリ (NumPy, pymoo 等) とのシームレスな統合を実現している。また、データベースの構築において、複数のレシピサイトから収集したデータの整合性を保つため、difflib ライブラリを用いたゲシュタルトパターンマッチングによる名寄せ処理を実装した。これにより、「鶏もも肉」と「若鶏モモ」のような表記ゆれを同一食材として認識させ、正確なコスト算出を可能にしている。

画像データの自動修復と信頼性担保

本システムは、先行研究で構築されたレシピデータを基盤としているが、Web 上の情報の変動により、多くの料理画像リンクが無効 (リンク切れ) となっている問題があった。料理画像は、ユーザーが献立を選択する際の UX (User Experience) に直結する重要な要素であるため、本研究では画像リンクの自動修復モジュールを新たに実装した。

画像収集にあたっては、検索エンジンの API (DuckDuckGo Search) を利用してレシピ名に基づく画像検索を行うが、不適切な画像の混入を防ぐため、表 4.3 に示す「信頼済みドメイン (Trusted Domains)」リストを定義した。実装したアルゴリズムは以下の手順で動作する。

- リンク切れ検知:** データベース内の画像 URL を走査し、無効なリンクまたは「No Image」のレコードを抽出する。
- バッチ検索処理:** 抽出されたレコードに対し、レシピ名をクエリとして画像検索を実行する。この際、サーバー負荷を考慮し、一定行数ごとのバッチ処理 (Start Row ~ End Row) を行う。
- ドメインフィルタリング:** 検索結果の中から、表 4.3 のリストに含まれるドメインの画像のみを採用する。これにより、質の低い画像や無関係な画像の混入を防止する。

表 4.3: 画像取得における信頼済みドメインリスト (Trusted Domains)

ドメイン名	サイト概要
cookpad.com	クックパッド (ユーザー投稿型レシピ)
kikkoman.co.jp	キッコーマン (公式レシピ)
kurashiru.com	クラシル (動画レシピ)
delishkitchen.tv	DELISH KITCHEN
orangepage.net	オレンジページ net
lettuceclub.net	レタスクラブ
sirogohan.com	白ごはん.com
nhk.or.jp	NHK (みんなのきょうの料理など)
ajinomoto.co.jp	味の素パーク
oceans-nadia.com	Nadia (プロのレシピ)

図 4.2: ユーザー基本情報およびアレルギー・疾患情報の入力画面

図 4.3: 調理時間制約および最適化目的の選択画面

4. レートリミット対策: 検索エンジンからのアクセス遮断を防ぐため、リクエストごとにランダムな待機時間 (4~8秒) を設ける処理 (Rate Limiting) を実装している。

このモジュールにより、数千件に及ぶレシピ画像の整合性を回復し、システムの視覚的な質を担保している。

入力インターフェースとフィルタリング

ユーザーは図 4.2 に示す画面で身体情報およびアレルギー情報を入力する。入力された身体情報に基づき、厚生労働省の基準を参照して個別に必要な栄養素量が自動算出される。続いて図 4.3 の設定画面において、作成日数や調理時間の制約、および最適化で重視する項目を選択する。これらの入力に基づき、データベースから不適切なレシピを事前に除外するフィルタリング処理が実行される。

§ 4.2 嗜好学習モデルと推論アルゴリズム

本節では、ユーザーの動的な嗜好を学習する AI モデルと、その出力スコアを用いて献立を構成する推論アルゴリズムについて述べる。

自然言語処理による特徴量抽出

本システムでは、レシピのテキスト情報を定量化するために TF-IDF 法を採用しているが、その前処理として日本語形態素解析エンジン MeCab を用いた高度なトークナイゼーションを行っている。レシピの材料名や手順には「大さじ」「適量」といった、嗜好抽出においてはノイズとなる単語が多く含まれる。そこで本研究では、IPADIC 辞書を用いて形態素解析を行い、以下の品詞フィルタリングを適用した。

- **抽出対象:** 名詞（一般、固有名詞）、動詞（自立）
- **除外対象:** 助詞、助動詞、記号、および「少々」「お好みで」などのストップワード

この処理により抽出された単語群を用いてコーパスを構築し、scikit-learn の `TfidfVectorizer` により約 4,000 次元の疎行列（Sparse Matrix）としてベクトル化を行っている。高次元かつ疎（Sparse）なデータを効率的に扱うことで、計算コストを抑えつつ詳細な特徴抽出を実現している。

TF-IDF 計算において、あるレシピ d における単語 t の重み $w_{t,d}$ は以下の式で算出される。

$$\text{tf}(t, d) = \frac{n_{t,d}}{\sum_k n_{k,d}}, \quad \text{idf}(t) = \log \frac{N}{\text{df}(t)} + 1 \quad (4.1)$$

$$w_{t,d} = \text{tf}(t, d) \cdot \text{idf}(t) \quad (4.2)$$

この重み付けにより、多くのレシピに共通して現れる一般的な食材（例：塩、水）の影響度を下げ、その料理を特徴づける食材（例：トリュフ、パクチー）の重要度を高めている。

フィードバックループと Rocchio アルゴリズム

ユーザーは提案された献立に対し、図 4.4 に示す画面を通じて詳細なフィードバックを行う。蓄積された評価データに基づき、以下の Rocchio アルゴリズムを用いてプロファイルを更新する。

$$\vec{P}_{new} = \vec{P}_{old} + \alpha \cdot (S - S_{neutral}) \cdot \vec{D}_{recipe} \quad (4.3)$$

ここで、 \vec{D}_{recipe} は評価対象レシピの TF-IDF ベクトル、 S は評価値（1~5）である。この更新式により、ユーザーが高評価を与えたレシピの特徴（食材や味付け）はプロファイルベクトル内で強化され、低評価の特徴は減衰する。更新されたプロファイル \vec{P}_{new} と、未提案の各レシピ r とのコサイン類似度を計算し、これを UX スコア（嗜好適合度）として定義する。

$$\text{Score}(r) = \frac{\vec{r} \cdot \vec{P}_{new}}{\|\vec{r}\| \|\vec{P}_{new}\|} \quad (4.4)$$

コサイン類似度はベクトル間の角度に基づく指標であり、ベクトルの長さ（食材の数）に依存せずに嗜好の方向性の一致度を測ることができるので、レシピ推薦に適している。

AIスコアに基づく時間帯割り当てヒューリスティクス

最適化エンジンによって選出された N 個のレシピ集合（パレート解）は、そのままでは「どのレシピをいつ食べるか」が決まっていない。食事の満足度を高めるためには、単に栄養を満たすだけでなく、「朝食は手早く済ませたい」「夕食はしっかり食べたい」といった人間らしいリズムに合わせる必要がある。そこで本システムでは、AIが算出した「適合度スコア」と「調理時間制約」に基づき、以下のヒューリスティックなアルゴリズムを実装し、自動割り当てを行っている。

1. **候補プールの生成:** 選出された全レシピについて、調理時間がユーザーの設定した各時間帯（朝・昼・夕）の上限時間 T_{limit} 以内であるかを判定し、割り当て可能な候補プール $P_{breakfast}, P_{lunch}, P_{dinner}$ を作成する。

$$P_{meal} = \{r \in R_{selected} \mid \text{Time}(r) \leq T_{limit}^{meal}\}$$

2. **朝食の決定 (Priority 1):** 最も時間制約が厳しくなりがちな朝食を優先的に決定する。朝食プール $P_{breakfast}$ の中から、AIによる「朝食適合度スコア」が最も高いレシピを1つ選択し、朝食とする。この適合度スコアは、朝食らしい食材（パン、卵、ヨーグルト等）を含むレシピが高くなるよう事前に学習されている。
3. **夕食の決定 (Priority 2):** 残りのレシピのうち P_{dinner} に含まれるものから、「夕食適合度スコア」が高い順に主菜・副菜（計3品程度）を選択し、夕食とする。
4. **昼食の決定 (Priority 3):** 残ったレシピを昼食に割り当て、時間制約を満たすか最終確認を行う。

このアルゴリズムにより、パレート最適解としての栄養学的整合性を維持しつつ、実際の食生活において実行可能な献立スケジュールを自動生成することが可能となる。

§ 4.3 多目的最適化問題の定式化と解探索

本システムの中核となる献立生成エンジンは、多目的最適化問題として定式化されている。本節ではその数理モデルの詳細と、Python ライブラリ `pymoo` を用いた実装詳細について述べる。

NSGA-II アルゴリズムの実装詳細

多目的最適化問題の解法には、非優越ソート遺伝的アルゴリズム (NSGA-II) を採用した。NSGA-II は、解の優越関係に基づくランキング (Ranking) と、解の多様性を維持するための混雑距離 (Crowding Distance) を用いることで、広範囲に分布するパレート最適解を効率的に探索する。本システムにおける実装の詳細は以下の通りである。

1. パレート支配の定義 (Pareto Domination):

ある解 x_1 が別の解 x_2 を支配する ($x_1 \prec x_2$) とは、全ての目的関数 f_m において x_1 が x_2 よりも優れているか同等であり、かつ少なくとも一つの目的関数で優れている状態を指す。数学的には以下のように定義される。

$$\forall m \in \{1, \dots, M\}, f_m(x_1) \leq f_m(x_2) \wedge \exists n \in \{1, \dots, M\}, f_n(x_1) < f_n(x_2) \quad (4.5)$$

表 4.4: NSGA-II ハイパーパラメータ設定

パラメータ	設定値	説明
Population Size	200	個体数
Generations	30	世代数
Sampling	Random	初期個体生成手法
Crossover	Binary Crossover	二点交叉
Mutation	Bit-flip Mutation	ビット反転突然変異
Selection	Tournament (k=2)	トーナメント選択
Eliminate Duplicates	True	重複個体の排除

2. ランク付け (Fast Non-dominated Sorting):

集団内の各個体 p に対し, 他の個体 q に優越される回数 (被優越数 n_p) と, 自分が優越する個体のリスト S_p を計算する. $n_p = 0$ となる個体をランク 1 (パレートフロント) とし, これらを集合から除いた後に再度 n_q を更新して次のランクを決定する. これを全個体がランク付けされるまで繰り返す.

3. 混雑距離の計算 (Crowding Distance Calculation):

同一ランク内での解の多様性を保つため, 各個体 i の混雑距離 d_i を以下の式で算出する.

$$d_i = \sum_{m=1}^M \frac{f_m(i+1) - f_m(i-1)}{f_m^{\max} - f_m^{\min}} \quad (4.6)$$

ここで, M は目的関数の数 (本研究では 3), f_m は m 番目の目的関数値である. 両端の個体には無限大の距離を与え, 距離が大きい (混雑していない) 個体を優先的に選択することで, 解空間全体に広がった献立案を生成する.

最適化エンジン pymoo の設定

実装には pymoo ライブラリを用い, ‘SubsetProblem’ クラスとして問題定義を行った. 表 4.4 に, 本実験で採用した遺伝的アルゴリズムのハイパーパラメータ設定を示す. 個体数 (Population Size) は Web アプリケーションの応答速度を考慮して 200 とし, 世代数 (Generations) は 30 に設定した.

数理モデルと制約条件の定式化

決定変数を $x_{k,i} \in \{0, 1\}$ (スロット k にレシピ i を採用するか否か) とし, 以下の目的関数と制約条件を設定する.

1. 目的関数 (Objectives):

ユーザーが選択した 2 つの指標を最小化 (または最大化) する.

- 食材コストの最小化: $f_1 = \sum_k \sum_i x_{k,i} \cdot \text{Cost}_i$
- 調理時間の最小化: $f_2 = \sum_k \sum_i x_{k,i} \cdot \text{Time}_i$
- UX スコアの最大化: $f_3 = \sum_k \sum_i x_{k,i} \cdot \text{Score}(r_i)$

2. 制約条件 (Constraints):

本システムでは, 単なる栄養価の合計だけでなく, 食事の多様性やバランスを保つために, 以下の 6 つの詳細な制約条件 ($g_1 \dots g_6$) を実装した. pymoo はこれらの制約違反量の総和 (Constraint Violation) を最小化するよう動作する.



図 4.4: レシピ詳細表示およびフィードバック入力画面

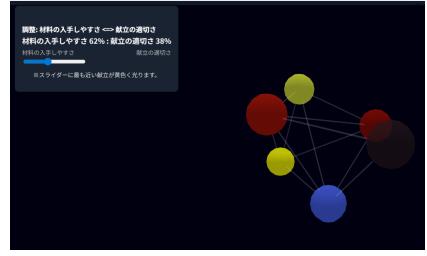


図 4.5: パレート最適解の可視化と対話的な選択インターフェース

- **栄養素下限制約** (g_1, g_2, g_3): タンパク質, 脂質, 炭水化物の摂取量が, ユーザーの必要量 ($Req_{protein}, Req_{fat}, Req_{carb}$) を下回らないこと.

$$g_1 = Req_{protein} - \sum x \cdot \text{Protein} \leq 0$$

$$g_2 = Req_{fat} - \sum x \cdot \text{Fat} \leq 0$$

$$g_3 = Req_{carb} - \sum x \cdot \text{Carb} \leq 0$$

- **エネルギー範囲制約** (g_4, g_5): 総摂取カロリーが, 推定エネルギー必要量 (EER) の許容範囲内 ($\pm 200\text{kcal}$) に収まること.

$$g_4 = (\text{EER} - 200) - \sum x \cdot \text{Cal} \leq 0$$

$$g_5 = \sum x \cdot \text{Cal} - (\text{EER} + 200) \leq 0$$

- **ジャンル多様性制約** (g_6): 提案される献立が特定のジャンル (和食のみ等) に偏らないよう, 和・洋・中の3ジャンルすべてが含まれることを制約とする.

$$g_6 = 3 - \text{CountUniqueGenres}(x) \leq 0$$

これらの厳密な制約条件により, 栄養バランスが完璧でありながら, 飽きのこない多様な献立の生成を保証している.

パレート解の提示と対話的選択

導出されたパレート最適解集合は, 図 4.5 に示すインターフェースにて可視化される. 画面上には各解の目的関数値の分布が表示され, ユーザーはスライダー操作により重み付けを調整できる. システムはスライダーの値に最も近いパレート解をリアルタイムで強調表示するため, ユーザーは「コスト」と「満足度」のトレードオフを視覚的に理解しながら, 自身の状況に最適な献立を選択することが可能である.

また, 可視化に際しては, ネットワークグラフ構造を用いた. パレート解同士の類似度 (共通するレシピの割合) をエッジの重みとして計算し, 似た献立同士が近くに配置されるようにレイアウトしている. これにより, ユーザーは「全く異なる献立案」を直感的に探し出すことができる.

数値実験並びに考察

§ 5.1 数値実験の概要

本研究の流れは、料理レシピ・食材価格のデータベースの作成、ユーザ情報と制約条件の入力、NSGA-IIによる多目的最適化と最適な献立の出力、対話型処理による献立の選択、ユーザによるレシピの評価、ロジスティック回帰分析の結果を用いた献立作成の繰り返しとなっている。まず、使用するレシピデータ数は1000個とした。Pythonによるスクレイピングを行う際は、PythonのライブラリであるurllibとBeautifulSoup4を使用した。

使用したレシピサイトは「ボブとアンジー」、「EatSmart」、「おいしい健康」の3種類であり、これらのサイトからスクレイピングを行う。urllibにより、目的のレシピサイトおよび食材価格サイトのWebページのURLを渡し、そのページのHTML情報を取得したのち、BeautifulSoup4を用いてWebページ上の料理レシピ名や摂取栄養素、食材とその価格などの必要な要素を、class名やid名などで指定し取得する関数を用いてスクレイピングを行う。Webサイトからのスクレイピングによって作成した料理レシピデータベースの例を図5.1に示す。

3つのレシピサイトからスクレイピングする情報として、料理から摂取可能な全栄養素やカロリー、調理時間、必要な材料名、材料量、料理のイメージ、アレルギー情報、作り方などが含まれる。また、各料理レシピの食材コストについては、料理に必要な食材と食材価格データベース内の食材を照らし合わせ、必要食材量と食材の価格、販売単位を用いて計算する。次に、NSGA-IIによる多目的最適化を行っている際の実行画面を図5.2に示す。これはプログラムの内部で行われている処理を可視化したものであり、本研究ではブラウザでシステムを用いているため、表示はされない。NSGA-IIを用いた多目的最適化プログラムは、Pythonのライブラリであるpymooを利用して記述した。pymooは、多目的最適化や単目的最適化などの様々な解法をサポートするライブラリである。

今回の実験で設定した目的関数と制約条件について説明する。目的関数は、調理時間の最小化と食材コストの最小化である。制約条件は、健常者の場合と制限食が必要な人の場合で異なる。まず、健常者の場合について説明する。摂取栄養素については、3大栄養素であるたんぱく質、脂質、炭水化物のそれぞれについて、1日に最低でも摂取すべき量を摂取できるように設定した。設定した値は、それぞれの3大栄養素に対して、たんぱく質は1日に必要な推定エネルギーの13%以上、脂質は15%以上、炭水化物は40%以上である。超過摂取による健康障害のリスクが高まると定義される耐容上限量は、3大栄養素に関しては設定されていないため[21]、制約条件として上限値は設定しないことにした。

摂取カロリーについては、1日に必要なエネルギー量の目安を掲載している農林水産省

エビと豆腐の煮物	1	10	174	むきエビ	120g	たんぱく質 15.9g
エビと豆腐の煮物	1	10	174	納ごし豆腐	1/2丁	炭水化物 6.4g
エビと豆腐の煮物	1	10	174	グリンピ	大さじ1	糖質 5.8g
エビと豆腐の煮物	1	10	174	調味料等		脂質 8.5g
エビと豆腐の煮物	1	10	174	酒	大さじ1/2	食塩相当量 1.5g
エビと豆腐の煮物	1	10	174	だし汁	2/3カップ	食物繊維 0.6g
エビと豆腐の煮物	1	10	174	(A)		ビタミンA 4 µg
エビと豆腐の煮物	1	10	174	砂糖	小さじ1	ビタミンB 0.12mg
エビと豆腐の煮物	1	10	174	・ うすくち 大さじ3/4		ビタミンC 0.09mg
エビと豆腐の煮物	1	10	174	水溶き片栗粉	適量	ビタミンE 0.13mg
エビと豆腐の煮物	1	10	174	サラダ油	適量	ビタミンK 0.9 µg
エビと豆腐の煮物	1	10	174	NAN	NAN	ビタミンC 2.2mg
エビと豆腐の煮物	1	10	174	NAN	NAN	ビタミンE 1.2mg
エビと豆腐の煮物	1	10	174	NAN	NAN	ビタミンK 1.9mg
エビと豆腐の煮物	1	10	174	NAN	NAN	ビタミンB 21.1µg
エビと豆腐の煮物	1	10	174	NAN	NAN	カリウム 351mg
エビと豆腐の煮物	1	10	174	NAN	NAN	カルシウム 82mg
エビと豆腐の煮物	1	10	174	NAN	NAN	鉄 1.4mg
エビと豆腐の煮物	1	10	174	NAN	NAN	葉酸 49 µg
エビと豆腐の煮物	1	10	174	NAN	NAN	マグネシウム 67mg
エビと豆腐の煮物	1	10	174	NAN	NAN	リン 248mg
エビと豆腐の煮物	1	10	174	NAN	NAN	亜鉛 1.1mg
エビと豆腐の煮物	1	10	174	NAN	NAN	パンチテ 0.38mg
エビと豆腐の煮物	1	10	174	NAN	NAN	銅 0.33mg
エビと豆腐の煮物	1	10	174	NAN	NAN	マンガン 0.33mg

図 5.1: 栄養素データの例

n_gen	n_eval	cv (min)	cv (avg)	n_nds	eps	indicator
1	400	1.05290E+03	1.64285E+03	1	2.47000E+02	-
2	800	8.49100E+02	1.34299E+03	1	0.00000E+00	f
3	1200	8.49100E+02	1.2941E+03	1	3.16300E+03	ideal
4	1600	8.20700E+02	1.16736E+03	1	0.00000E+00	ideal
5	2000	7.87800E+02	1.09991E+03	1	2.02200E+03	ideal
6	2400	7.79000E+02	1.03477E+03	1	4.85000E+02	ideal
7	2800	7.15900E+02	9.73739E+02	1	8.54000E+02	ideal
8	3200	5.95000E+02	9.29777E+02	1	1.97600E+03	ideal
9	3600	4.22400E+02	8.85000E+02	1	0.00000E+00	f
10	4000	4.22400E+02	8.42549E+02	1	0.00000E+00	f
11	4400	4.22400E+02	8.03732E+02	1	0.00000E+00	f
12	4800	4.22400E+02	7.62277E+02	1	0.00000E+00	f
13	5200	4.22400E+02	7.24457E+02	1	0.00000E+00	f
14	5600	4.22400E+02	6.92382E+02	1	0.00000E+00	f
15	6000	4.22400E+02	6.61212E+02	1	0.00000E+00	f
16	6400	3.96300E+02	6.33425E+02	1	0.00000E+00	f
17	6800	3.96300E+02	6.06926E+02	1	0.00000E+00	f
18	7200	3.86900E+02	5.78641E+02	1	4.70000E+02	ideal
19	7600	2.86900E+02	5.55043E+02	1	0.00000E+00	f
20	8000	2.86900E+02	5.31442E+02	1	0.00000E+00	f
21	8400	2.86900E+02	5.10590E+02	1	0.00000E+00	f
22	8800	2.86900E+02	4.93032E+02	1	0.00000E+00	f
23	9200	2.86900E+02	4.75298E+02	1	0.00000E+00	f
24	9600	2.86900E+02	4.57564E+02	1	0.00000E+00	f
25	10000	2.38800E+02	4.42169E+02	1	2.76000E+02	ideal

図 5.2: 最適化処理の実行画面

のサイト [22] を参考にし、基礎代謝量と身体活動レベルの係数を掛け合わせた値を用いた。そのため、上限値は 2536 キロカロリーに設定した。

次に、制限食が必要な人の制約条件について説明する。本研究で対象となる制限食が必要な人は、アレルギーを持つ人と生活習慣病を患っている人である。また、対象となる生活習慣病は糖尿病、腎臓病、脂質異常症、高血圧とする。まず、糖尿病を患っている人について説明する。3.1 章で述べたように、糖尿病は内臓脂肪型肥満が原因となり、インスリン抵抗性を引き起こすことで発症する。そのため、糖尿病の予防や改善には脂肪の適切な管理が重要である。また、厚生労働省の指針によれば、1日あたりの炭水化物摂取量を 100g 以下に抑えることが、肥満の改善に有効であり、糖尿病予防にも役立つとされている。さらに、食物繊維の1日あたりの摂取量が 20g を超えると、糖尿病の発症リスクが有意に低下することが報告されている [?]. 以上のことから、本研究では糖尿病患者に対する制約条件として、エネルギー摂取量とタンパク質摂取量は健常者と同じ基準を維持しつつ、1日の炭水化物摂取量を 100g 以下、脂質摂取量を推定エネルギー必要量の 15~25% の範囲、食物繊維摂取量を 20g 以上に設定する。

次に、腎臓病患者に対する制約条件を示す。3.1 章で説明したように、腎臓病の進行を抑制し、慢性腎疾患の合併症を防ぐためには、タンパク質制限、塩分制限、カリウム制限を含む食事療法が有効とされる。日本腎臓学会の指針では、1日のタンパク質摂取量は標準体重 1kg あたり 0.6~0.7g、塩分摂取量は 3g 以上 6g 未満、カリウム摂取量は 1500mg 以下に制限することが推奨されている [?]. このため、本研究では腎臓病患者の制約条件として、エネルギー、脂質、炭水化物の摂取量は健常者と同様としつつ、1日のタンパク質摂取量を標準体重 1kg あたり 0.6~0.7g、塩分摂取量を 3g 以上 6g 未満、カリウム摂取量を 1500mg 以下と設定する。

脂質異常症の患者については、3.1 章で述べたように、コレステロール、食物繊維、脂質の摂取バランスを調整することが、予防および改善に役立つとされている [?]. 本研究では、脂質異常症の患者に対する制約条件として、エネルギー摂取量、炭水化物摂取量、タンパク質摂取量は健常者と同様とし、1日のコレステロール摂取量を 200mg 以下、脂質摂取量を総エネルギーの 15% 未満、食物繊維摂取量を 20g 以上とする。

高血圧と診断されている人については、3.1 章で説明したように、塩分の過剰摂取による血圧上昇が主な要因の一つとされるため、減塩が不可欠である。日本高血圧学会の「高血圧治療ガイドライン 2019」では、高血圧患者の塩分摂取量を1日 6g 未満に制限することが推奨されている。また、カリウム摂取量の増加が高血圧患者の血圧低下に寄与することが報

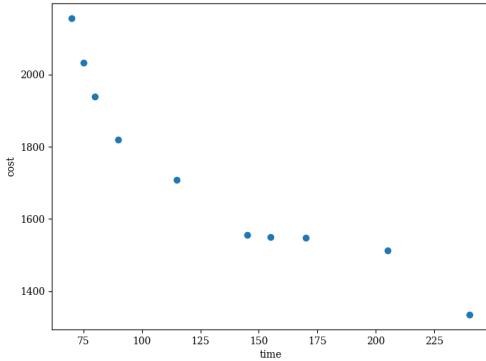


図 5.3: パレート解の出力



図 5.4: 対話型処理による解の選択

告されている。厚生労働省によると、1日のカリウム摂取量は3510mg以上が望ましいとされている。さらに、食物繊維の摂取量を20g以上とすることが推奨されている[?].これらを踏まえ、本研究では高血圧患者の制約条件として、エネルギー、脂質、炭水化物、タンパク質の摂取量は健常者と同様とし、1日の塩分摂取量を6g未満、カリウム摂取量を3510mg以上、食物繊維摂取量を20g以上とする。

次に、NSGA-IIによる最適化プロセスについて説明する。最適化の実行画面において、`n_gen`は現在の世代数、`n_eval`は評価が完了した個体数、`cv(min)`および`cv(avg)`はそれぞれ現在の母集団における最小の制約違反と平均の制約違反、`n_nds`は非劣解の個数、`eps`は過去の世代における指標の変化量、`indicator`は最適化のパフォーマンス指標を示す。

また、最適化処理の結果として得られたパレート最適解を図5.3に示す。縦軸は指定した期間の献立にかかる合計調理時間を表し、横軸は同期間における食材コストの合計を示している。

次に、パレート最適解から対話型処理を用いて献立を選択する画面を図5.4に示す。図5.4は、図5.3で表示されたパレート解を数値として表示する画面である。ユーザは、画面上の選択ボタンを使用して提示された候補を選択することができる。「献立を表示する」ボタンをクリックすると、選択した候補に対応する献立が出力される。その後、ユーザは各レシピについて4つのUX項目を評価し、ロジスティック回帰分析を行うことで、UX項目を考慮した献立作成が可能となる。

本研究では、以下の数値実験を実施する。まず、ユーザに身体情報を入力してもらい、時間とコストを目的関数とした最適化問題を解き、その結果得られた献立を提示する。次に、その献立に対してUX項目に関する質問を行い、得られた評価データをCSVファイルに保存する。このデータを基にロジスティック回帰分析を行い、ユーザごとに重要視するUX項目を抽出する。さらに、UX項目を考慮した献立を再度作成し、ユーザに評価を依頼する。このプロセスを繰り返すことで、ユーザが重要視するUX項目を反映した献立が徐々に洗練されていく仕組みとなっている。また、ロジスティック回帰分析を行うごとにモデルの評価を行う。

表 5.2: 精度の t 検定の結果

表 5.1: モデル評価 (精度) の出力結果

精度	推薦前	推薦後	変化率
q1	0.42857	0.59524	0.38889
q2	0.49206	0.77778	0.58065
q3	0.36508	0.5873	0.6087
q4	0.66667	0.7619	0.14286
平均値	0.4881	0.68056	0.43027

表 5.3: モデル評価 (適合率) の出力結果

適合率	推薦前	推薦後	変化率
q1	0.41667	0.75304	0.80729
q2	0.60684	0.88442	0.45742
q3	0.53704	0.75966	0.41455
q4	0.79206	0.85307	0.07702
平均値	0.58815	0.81255	0.43907

表 5.4: 適合率の t 検定の結果

	推薦前	推薦後
平均	0.4881	0.68056
分散	0.01686	0.01068
観測数	4	4
ピアソン相関	0.78149	
仮説平均との差異	0	
自由度	3	
t	-4.7501	
P(T<=t) 片側	0.00885	
t 境界値 片側	2.35336	
P(T<=t) 両側	0.0177	
t 境界値 両側	3.18245	

表 5.4: 適合率の t 検定の結果

	推薦前	推薦後
平均	0.58815	0.81255
分散	0.02465	0.00438
観測数	4	4
ピアソン相関	0.72211	
仮説平均との差異	0	
自由度	3	
t	-3.79	
P(T<=t) 片側	0.01611	
t 境界値 片側	2.35336	
P(T<=t) 両側	0.03222	
t 境界値 両側	3.18245	

§ 5.2 実験結果と考察

今回数値実験をするにあたって、朝、昼、夜の合計の調理時間の合計と主菜、副菜の数を共通にしておく。具体的な数値として、朝、昼、夜の合計の調理時間の合計をそれぞれ 15 分、45 分、60 分とし、主菜、副菜の数を 1 つ、2 つとした。

被験者は大学生 21 才から 22 才の男性 6 名に対して実験を行った。それぞれにロジスティック回帰分析の後の目的関数をユーザ 1 は q1, q2, ユーザ 2 は q1, q3, ユーザ 3 は q1, q4, ユーザ 4 は q2, q3, ユーザ 5 は q2, q4, ユーザ 6 は q3, q4 を選択してもらった。アレルギーを持つ人はいなく、ロジスティック回帰分析を行うためのテストデータを集めるために目的関数に時間とコストが設定された献立を 1 週間分提示して評価を行ってもらう。その後ロジスティック回帰分析を行い、UX 項目の推定、考慮された献立の評価を行う。分析と評価を 3 回繰り返し、ロジスティック回帰分析前後の精度、適合率、再現率、F 値についての変化を考察する。

ユーザごとに重要視する UX 項目を選択してもらい、それに基づく評価を収集することで、モデルの精度向上を図る。本研究では、ユーザが繰り返し評価を行うプロセスを通じて、モデルが输出する献立が UX 項目においてより高い評価を得られるようになる過程を示す。さらに、一回目のモデル評価結果と四回目のモデル評価結果について対応のある t 検定を実施し、有意差が認められるかどうかを検証する。

本プログラムのモデルの考察

モデル評価項目ごとの結果とそれらの t 検定の結果を表 5.1 から 5.8 まで、ユーザごとのモデル評価の回数ごとの推移の例を図 5.5 から 図 5.7 まで示す。本研究では、UX 項目を考

表 5.6: 再現率の t 検定の結果

表 5.5: モデル評価 (再現率) の出力結果

再現率	推薦前	推薦後	変化率
q1	0.83766	0.82108	-0.0198
q2	0.52894	0.85	0.60699
q3	0.39145	0.65694	0.67822
q4	0.74164	0.8634	0.16417
平均値	0.62492	0.79786	0.3574

	推薦前	推薦後
平均	0.62492	0.79786
分散	0.04087	0.00914
観測数	4	4
ピアソン相関	0.69836	
仮説平均との差異	0	
自由度	3	
t	-2.2798	
P(T<=t) 片側	0.05348	
t 境界値 片側	2.35336	
P(T<=t) 兩側	0.10695	
t 境界値 兩側	3.18245	

表 5.7: モデル評価 (F 値) の出力結果

F値	推薦前	推薦後	変化率
q1	0.65518	0.77718	0.1862
q2	0.52983	0.864	0.6307
q3	0.45208	0.69818	0.54439
q4	0.76441	0.85352	0.11657
平均値	0.60038	0.79822	0.36946

表 5.8: F 値の t 検定の結果

	推薦前	推薦後
平均	0.60038	0.79822
分散	0.01896	0.00595
観測数	4	4
ピアソン相関	0.56868	
仮説平均との差	0	
自由度	3	
t	-3.4933	
P(T<=t) 片側	0.01984	
t 境界値 片側	2.35336	
P(T<=t) 兩側	0.03967	
t 境界値 兩側	3.18245	

慮した献立推薦の効果を検証するため、ロジスティック回帰分析の結果を用いて、推薦の前後で精度、適合率、再現率、および F 値を比較し、片側の t 検定を実施した。帰無仮説として「UX 項目を考慮した推薦によって指標に有意な差が生じない」を設定し、対立仮説として「UX 項目を考慮することで指標が改善される」を設定した。その結果、再現率を除く 3 つの指標（精度、適合率、F 値）で帰無仮説が棄却され、有意差が認められた。

まず、精度、適合率、および F 値が有意に向上した点は、UX 項目を考慮することで推薦アルゴリズムの性能が改善し、ユーザーの期待やニーズに応じた献立選択が可能になったことを示している。特に適合率の向上は、推薦結果の精度が高まり、誤った推薦が減少したことを示唆している。これにより、UX 項目の考慮がユーザー体験の向上に寄与する可能性が示唆される。

一方、再現率では有意差が認められなかったものの、p 値は 0.5348 と比較的惜しい結果となった。これは、UX 項目を考慮することで、ある程度再現率に影響を及ぼした可能性を示唆しているが、その効果が統計的に有意となるほどの強さではなかったと考えられる。再現率は、推薦システムがどれだけ多くの適切な項目を網羅的に捕捉できるかを測る指標であるため、UX 項目の導入がこの側面に与える影響が限定的であった可能性がある。

この結果を踏まえ、UX 項目が精度、適合率、および F 値に対して顕著な効果をもたらした一方で、再現率においてはさらなる改善の余地があると考えられる。

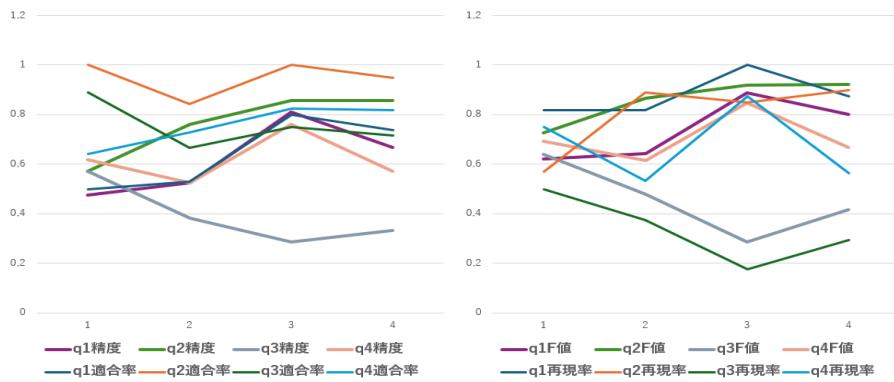


図 5.5: ユーザ 1 のモデル評価の推移

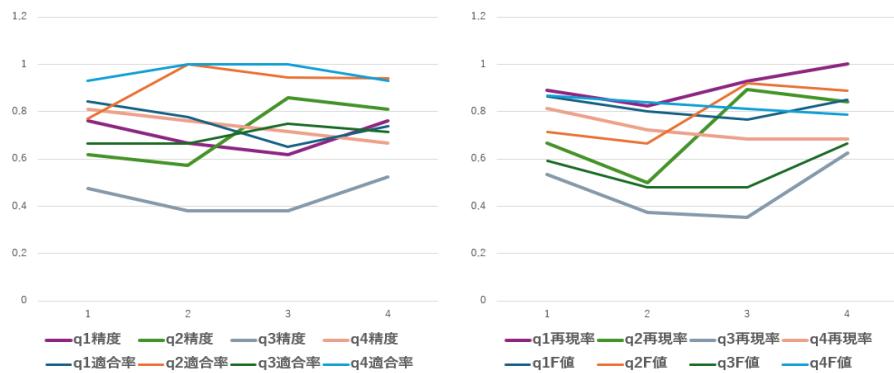


図 5.6: ユーザ 2 のモデル評価の推移

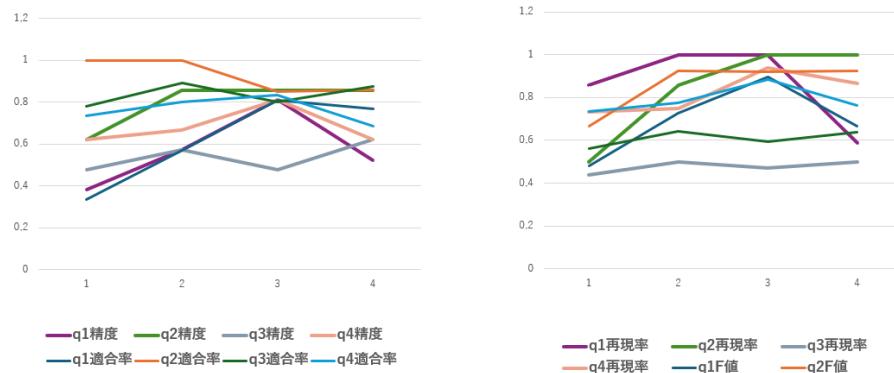


図 5.7: ユーザ 3 のモデル評価の推移

おわりに

本研究では、急激な生活様式の欧米化に伴い、ジャンクフードなどの過剰なエネルギー摂取を伴う食生活が広まり、生活習慣病の罹患率が増加しているという現状を踏まえ、生活習慣病の予防に向けたアプローチとして、栄養バランスの取れた食事の重要性に着目した。栄養バランスの取れた献立を作成することは、健康的な生活を送る上で非常に重要であるが、実際にはその作成には個々の身体情報や疾患情報を元にメニューや料理の分量を調整する必要があり、その作業は非常に手間がかかるものである。

このような問題を解決するために、本研究では、Webサイトから得られるレシピ情報や食材の価格などを活用し、ユーザーの健康状態や個々のニーズに合わせて栄養バランスを考慮した献立を自動的に作成できるシステムを提案した。具体的には、制約条件を考慮できる多目的遺伝的アルゴリズムを用いて、個々のユーザーに最適な献立を生成し、その過程でロジスティック回帰分析を活用して、UX項目を反映させることができるシステムを設計した。このシステムにより、ユーザーの健康状態や好みに応じて柔軟に献立を調整できるようになり、より個別化された食事提案が可能となる。

本研究で使用したレシピデータは、3つの異なるレシピサイトからスクレイピングを行い、データベースの多様性を持たせることに成功した。これにより、システムが提供する献立の選択肢が豊富になり、ユーザーの好みに応じた食事の提案が可能となった。また、このシステムは健常者だけでなく、生活習慣病やアレルギーを持つ人々にも対応できるように設計されており、幅広いユーザー層に対応した健康管理支援を提供できる点が大きな特徴である。

実際にシステムを動作させた実験結果に基づき、帰無仮説として「UX項目を考慮した推薦によって指標に有意な差が生じない」を設定し、対立仮説として「UX項目を考慮することで指標が改善される」を立てて検証を行った。その結果、再現率を除く3つの指標（精度、適合率、F値）について帰無仮説が棄却され、いずれの指標でも有意差が認められた。これにより、UX項目を考慮した献立推薦が、ユーザーにとってより精度の高い、適切な献立を提供することに寄与していることが示された。

この結果から、本研究の提案するシステムは、献立作成における最適化を実現し、単なる画一的な献立にとどまらず、ユーザーの身体情報やUX項目を考慮した個別化された提案を行えることが明らかになった。これにより、ユーザー一人ひとりの健康状態に適した献立の作成が可能となり、より効果的な健康管理が支援されることが期待される。

一方で、本研究にはいくつかの課題も残されている。例えば、摂取栄養素や摂取カロリーに関する上限・下限の設定など、制約条件をユーザーが柔軟に調整できるようにすることが今後の改良点として挙げられる

謝辞

本研究を遂行するにあたり、多大なご指導と終始懇切丁寧なご鞭撻を賜った富山県立大学工学部情報システム工学科情報基盤工学講座の奥原浩之教授, António Oliveira Nzinga René講師に深甚な謝意を表します。最後になりましたが、多大な協力をしていただいた研究室の同輩諸氏に感謝致します。

2025年2月

堀由隆

参考文献

- [1] ソフトム株式会社, ”ソフトム通信 第 79 号「給食業界における A I 活用」”, https://data.nifcloud.com/blog/food-service-provider_ai-use-case_01/, 閲覧日 2024.12.28.
- [2] C. A. Coello Coello and M. S. Lechuga, ”MOPSO: a proposal for multiple objective particle swarm optimization, ”*Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)*, Vol. 2, pp. 1051-1056, 2002.
- [3] 堀由隆, ”ウェルビーアイングに有益な User eXperience を考慮できる自動献立作成支援システム”, 富山県立大学卒業論文, 2025.
- [4] Q. Zhang and H. Li, ”MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition”, *IEEE Trans. Evolutionary Computation*, Vol. 11, No. 6, pp. 712–731, 2007.
- [5] LeftLetter, ”多目的進化型アルゴリズム MOEA/D とその改良手法”, <https://qiita.com/LeftLetter/items/a10d5c7e133cc0a679fa>, 閲覧日 2025.1.6.
- [6] John W. Ratcliff and David Metzener, ”Pattern Matching: The Gestalt Approach”, *Dr. Dobb's Journal*, p.46, 1988.
- [7] 株式会社おいしい健康, ”おいしい健康”, <https://oishi-kenko.com/>, 閲覧日 2023.10.16.
- [8] 総務省統計局, ”小売り物価統計調査による価格調査”, <https://jpmarket-conditions.com/>, 閲覧日 2024.10.11.
- [9] John H. Holland, ”Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, 1975.
- [10] G. Salton and M. J. McGill, ”Introduction to Modern Information Retrieval”, McGraw-Hill, Inc., New York, NY, USA, 1986.
- [11] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, ”A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II”, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197, 2002.
- [12] J. J. Rocchio, ”Relevance feedback in information retrieval”, in *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall, pp. 313-323, 1971.
- [13] 東北大学工学研究科・工学部, ”複合材航空機主翼の空力構造最適設計”, <https://www.eng.tohoku.ac.jp/news/detail-,id,3094.html>, 閲覧日 2025.12.05.
- [14] 楠原一紀, ”電力の地産地消を目指した自律分散型スマートグリッド”, システム/制御/情報, Vol. 56, No. 8, pp. 433-441, 2012.

- [15] 有村幹治, 田村亨, 井田直人, ”土木計画分野における遺伝的アルゴリズム：最適化と適応学習”, 土木学会論文集 D, Vol. 62, No. 4, pp. 505-518, 2006.
- [16] 青木博明, ”遺伝的アルゴリズムによる効率的フロンティアの導出”, 情報科学研究, 第 17 号, pp. 19-34, 2003.
- [17] C. D. Manning, P. Raghavan and H. Schütze, ”Introduction to Information Retrieval”, Cambridge University Press, 2008.
- [18] G. Adomavicius and A. Tuzhilin, ”Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, pp. 734-749, 2005.
- [19] M. Pazzani and D. Billsus, ”Content-Based Recommendation Systems”, in *The Adaptive Web*, Springer, pp. 325-341, 2007.
- [20] 和正敏, ”多目的線形計画問題に対する対話型ファジィ意思決定手法とその応用”, 電子情報通信学会論文誌 Vol. J 65-A, No. 11, pp. 1182-1189, 1982.
- [21] 厚生労働省, ”日本人の食事摂取基準 (2020 年版)”, <https://www.mhlw.go.jp/content/10904750/000586553.pdf>, 閲覧日 2024.12.26.
- [22] 農林水産省, ”一日に必要なエネルギー量と摂取の目安”, https://www.maff.go.jp/j/syokuiku/zissen_navi/balance/required.html, 閲覧日 2025.1.22.
- [23] 電力土木技術協会, ”遺伝的アルゴリズム”, <https://www.jepoc.or.jp/tecinfo/library.php>, 閲覧日 2025.1.22.