

卒業論文

ウェルビーイングに有益な User eXperience を 考慮できる 自動献立作成支援システムの開発

Development of a Browser-based Automatic Menu Creation
System Dealing with Restricted Meals and Large Groups of People

富山県立大学 工学部 情報工学科

2120040 堀由隆

指導教員奥原浩之

提出年月: 令和7年(2025年)2月

目 次

圖一覽

表一覽

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 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。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。特に、食生活が不規則で栄養が偏った状態で続けると、生活習慣病のリスクが高まる。

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、初期段階では自覚症状がほとんどないことだ。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生活習慣病の予防には欠かせない要素となる。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、食事の準備にかか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り、食事作りに対するストレスや負担を軽減する方法が必要だ。

このような背景の中で、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。食材の入手しやすさ、調理の簡便さ、時間帯に適した食事提案など、家庭の実情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【1】。多忙な現代人にとって、献立作成に要する労力や時間は大きな障壁となっており、特に家庭での食事作りが負担となることが多い。

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。これにより、献立作成業務は非常に負担が大きく、特に多忙な現場ではその負荷が問題となっている【2】。病院などでは、食事が患者の日常生活の中心であり、食事が患者の楽しみの一部であるため、食事の内容や栄養だけでなく、患者の好みに合った献立を提供することが求められる。このような現場では、食に対する専門性を高めるために日々研究や開発が行われているが、それに伴い献立作成の業務負担も増大している。患者一人一人のニーズに応じた献立作成が求められる一方で、そのための業務負担の軽減も重要な課題となっている。

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、膨大な料理データを基に、調理時間や食材コスト、さらには個々の身体的な情報や嗜好に基づいた最適な献立を自動的に提案することを目指す。このアプローチにより、家庭や病院などの現場で、手間をかけずに効率的にバランスの取れた食事を提供できるようになる。献立作成のために収集するデータには、Web上の複数のレシピサイトからスクレイピングによって集めた料理データを活用する。具体的には、各料理に必要な食材やその栄養価、カロリーなどの情報に加え、食材ごとの価格データも蓄積する。これにより、食材のコストパフォーマンスを考慮した献立提案が可能となり、家計にも優しい食事の提供が実現できる。

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。これにより、栄養バランスを保ちながらも、個々のニーズに最適な献立が提供される。また、ロジスティック回帰分析を用いて、ユーザーが過去に評価した献立に基づく UX 項目（例：調理のしやすさ、食材の入手しやすさ、時間帯に適した献立かどうかなど）を予測し、その結果を次の献立提案に反映させる。このアプローチにより、ユーザーごとにパーソナライズされた献立推薦が可能となり、効率的かつ個別の好みに合った食事を提供することができる。

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 本研究の背景と目的について説明する。背景では栄養バランスの摂れた献立を作成することの難しさと、自動で献立を作成することの重要性について示す。目的は制限食を考慮した多目的遺伝的アルゴリズムによる最適な自動献立作成について提案することを述べる。

第2章 多目的最適化による自動献立作成システムの概要と、Web上のデータを活用した例について説明する。

第3章 多目的最適化と、GAを応用した多目的GAの仕組みを説明する。また、本研究で用いる制限食及びブラウザベースのシステムについて説明する。

第4章 提案手法の中で利用者が入力する部分と、NSGA-IIによる多目的最適化によって最適な献立を対話型で出力する部分について説明する。

その後、提案手法について説明する。

第5章 提案手法に基づいて自動献立作成システムを構築して、実際に献立の作成を行った結果を示す。そして、本研究の提案手法によって得られた結果が有意であることを示す。

第6章 本研究で述べている提案手法をまとめて説明する。また、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 ウェルビーイングとUX

ウェルビーイング（well-being）とは、個人の健康や幸福、生活全体の質を示す包括的な概念であり、身体的、精神的、社会的、経済的な側面が相互に関連する多次元的な状態を指す。一般的に、ウェルビーイングは単なる健康状態にとどまらず、個人が自らの生活に満足しているか、心理的に充実しているか、社会的に適応しているか、そして経済的に安定しているかをも含む広範な指標である。

ウェルビーイングの主要な構成要素としては、以下のものが挙げられる：

- **身体的ウェルビーイング**：健康的な生活習慣、適切な栄養摂取、運動習慣、十分な睡眠など、身体的健康を維持することが基本である。
- **精神的ウェルビーイング**：ストレスの管理、感情的安定、ポジティブな感情や自己肯定感、心理的回復力など、精神面の健康が重要な要素となる。
- **社会的ウェルビーイング**：他者との良好な関係、社会的支援、所属感やコミュニティとのつながりなど、社会的なつながりが個人のウェルビーイングに大きな影響を与える。
- **経済的ウェルビーイング**：経済的な安定性、収入の水準、労働の充実感など、経済的な側面もウェルビーイングに大きな役割を果たす。
- **自己実現および精神的充足感**：個人が自己の価値観や目標に沿った生活を送り、自己実現を感じることが精神的ウェルビーイングを高める。

このように、ウェルビーイングは多面的であり、単一の健康状態や幸福感に限定されるものではなく、個人が社会全体の中でどのように生き、どのように自分自身を実現していくかというプロセスを包括的に反映する概念である。また、ウェルビーイングは社会政策や公共の福祉、企業戦略など、さまざまな分野においても重要な指標となっており、国際的には国民の幸福度や生活満足度を測るための指標としても利用されている。

ウェルビーイングを高める UX デザインの例

マインドフルネスアプリ

ユーザーインターフェース（UI）の配慮：Headspace や Calm のようなアプリでは、視覚的に落ち着けるデザインが使われている。シンプルなレイアウトと柔らかな色調

が、ユーザーにリラックス感を与え、マインドフルネスを実践しやすくしている。
パーソナライズ機能: ユーザーの進捗に合わせて瞑想セッションをカスタマイズする機能があり、ストレスレベルに応じたコンテンツの提供や、習慣化を促すリマインダー機能が役立っている。

フィットネスアプリ

モチベーションを高める要素: Nike Training Club や MyFitnessPal では、ユーザーの運動記録や目標達成度をグラフやダッシュボードで可視化し、達成感を感じやすくしている。また、定期的にモチベーションを維持できる通知が届く仕組みがある。

ソーシャル機能: ユーザーが他のユーザーと成果を共有したり、チャレンジを通じて競い合ったりする機能が提供されており、健康維持への意欲を高める。

ソーシャルメディアの使用制限機能

利用時間のトラッキング: Instagram や Facebook などでは、1日の使用時間をトラッキングし、制限時間を超えると通知が表示される。これにより、過剰な使用を防ぐことができる。

健康的な休憩の提案: 長時間使用後に「休憩を取る」などのリマインダーが表示され、ユーザーが自分の使用状況を意識できるようになっている。

E コマースのストレス軽減機能

簡潔で透明な情報提供: Amazon のようなプラットフォームは、商品の詳細や料金、返品ポリシーを明確に表示しており、購入時の不安を減らしている。

フォローアップ機能: 購入後、配送状況の通知や、満足度調査が送られることで、ユーザーは安心感を得られる。

高齢者向けサービス

視覚・聴覚への配慮: 高齢者向けのアプリやウェブサイトでは、大きなフォントや音声ナビゲーションが使われており、視覚や聴覚に障害があるユーザーも使いやすくなっている。

簡易化されたインターフェース: 複雑な操作を避け、直感的に使えるインターフェースが提供されている。例えば、ボタンサイズや配置を工夫し、高齢者でも簡単に操作できるようになっている。

障害者支援ツール

スクリーンリーダー対応: 視覚障害者向けに、スクリーンリーダーが動作するアプリやサイトが増えており、テキストの読み上げ機能が利用されている。

カスタマイズ可能なインターフェース: 身体的制約があるユーザー向けに、文字サイズやインターフェースの色、操作方法をカスタマイズできるツールが提供されている。

ゲームとウェルビーイング

健康管理を取り入れたゲーム: フィットネスゲームの進化: ポケモン GO のように、ゲームを通じて歩いたり運動を促進したりすることができ、ユーザーは楽しみながら健康維持できる。ゲーム内の達成感や報酬が、継続的なモチベーションを提供している。

健康データの活用: ヘルスケアアプリとの連携で、ゲーム内での歩数や運動量が記録され、ユーザーは自分の健康状態を確認できる。

§ 2.2 多目的最適化としての献立作成

多目的最適化とは、「制約条件のもと、複数の選択肢を組み合わせて何か結果を出すとき、その結果（目的関数）を最小、もしくは最大にすること」である。多目的最適化の利点として自動化による結果が出るまでの作業時間が削減されることや、答えを導くのに現実的ではない時間がかかる問題を解くことができることがある。

最適化問題の種類の1つとして、組み合わせ最適化問題が挙げられる。本研究の自動献立作成システムはこれに分類される。組み合わせ最適化問題とは、様々な制約のもとで多くの選択肢の中から、ある評価(価値)を最もよくする変数の値(組み合わせ)を求めることである。

献立における制約条件として、何日分の献立を作成するか、カロリーをどのくらい制限するか、特定の栄養素を最低でもどのくらい取得するか、などが挙げられる。また、目的関数として、調理時間の最小化や個人の嗜好の最大化、材料コストの最小化などが挙げられる。

しかし、組み合わせ最適化を解く場合、目的関数がトレードオフになる関係がある場合がある。トレードオフとは、何かを得ると別の何かを失う相容れない関係のことである。食事を例に挙げるとすると、一般的に高カロリーな食生活によって食の満足度は上がるが、体重が増えて栄養に支障が生じる。逆に健康を意識してダイエットを行えば、食の満足感が減る。この場合、「高カロリーな食生活」と「健康」の関係性がトレードオフの関係になっている。

目的関数がトレードオフの関係である場合、一方の目的関数の最小化あるいは最大化が、他方の目的関数の最小または最大化に悪い影響を及ぼすため、単一目的の最適化問題とは異なり、複数の目的関数をすべて満たすような一つの最適解を得ることは困難である多目的最適化での探索では、パレート最適解と呼ばれる概念を導入する必要がある。

パレート最適解とは、ある目的関数を満たそうとしたときに他の目的関数が犠牲になり満たされなくなってしまう解のことであり、非劣解とも呼ばれる。反対に、パレート最適ではないような解のことは劣解と呼ばれている。

また、パレート最適解は一般的には1つとなることはほとんどなく複数となる場合がほとんどであるため、集合となる。パレート最適解の集合のイメージを図2.1に示す。複数のパレート最適解を、目的関数空間に添付したときに形成される曲線は、パレート最適フロントと呼ばれる。実際にはこのパレート最適フロントの中から解を選択することになるのである。

また、一般的に最適化問題には、実行可能領域という、制約条件を満たす領域の内側に複数の局所的最適解を持つ。局所的最適解とは、その近辺では最も良い解であるが、実行可能領域全体で考えたときに、最も良い解になるとは限らない解のことである。局所最適解に対して大域的最適解とは、制約条件を満たしなおか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ものとなる。

大域的最適化解は、通常は単に最適解と呼ばれることが多い。しかし目的関数が凸関数

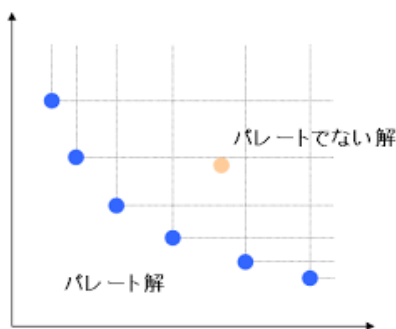


図 2.1: パレート最適解のイメージ

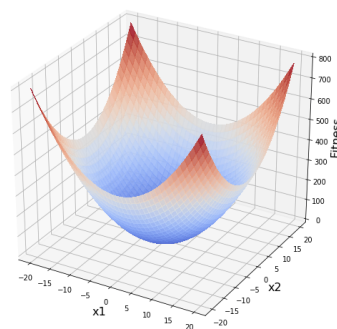


図 2.2: 解探索のイメージ (粒子群最適化)

で制約集合が凸集合である非凸計画問題や、組合せ最適化問題などにおいては、局所的最適解との区別を強調したい場合に、大域的という形容詞をつけて大域的最適解と呼ばれる場合が多い。

局所解を回避する方法のうち、例として遺伝的アルゴリズムを用いて最適化を解くという場合、突然変異率のパラメータを変更する方法がある。突然変異率は、低すぎると局所解に陥りやすくなる。しかし高すぎるとランダム探索になってしまうため、調整が必要となるパラメータとなる場合が多い。その他には、最適化アルゴリズムの1つである確率的勾配降下法は、確率的に局所解を抜け出せる可能性があるとされている。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、

$$\underset{x}{\text{minimize}} \quad \{f_1(x), f_2(x), \dots, f_n(x)\} \quad (2.1)$$

$$\text{subject to} \quad g_k(x) \leq 0 \quad k = 1, 2, \dots, m \quad (2.2)$$

のように定式することができる。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、最適化問題を解く様々な手法について説明する。

差分進化法

差分進化法とは、進化的アルゴリズムの一種であり、確率的な直接探索によって解集団を用いた多点探索を行うアルゴリズムである。差分進化法は非線形問題、微分不可能な問題などの、現実的な実行時間では厳密な解を導くことが困難な問題に対して近似解を求めることが可能であり、様々な最適化問題に適応されることが多い。差分進化法のアルゴリズムは、ベクトル集団の作成、変異、交叉、選択、終了判定、といった流れで構成されている。

Nelder-Mead 法

$n + 1$ 個の頂点からなる、 n 次元ユークリッド空間の実数値関数を $f(x)$ を与え、その

もとで $f(x)$ の最小値を微分に頼らずに求める方法の一つである。設計変数が2つの場合、2次元平面上の3つの点を初期値として与え、各ベクトルの目的関数を計算し、中でも最大値をとる点を、鏡像、拡大、収縮、縮小の4つの操作どれかを使い移動する。この操作を繰り返し行うことにより、全ての点を最小値に近づけていく、というアルゴリズムになっており、滑降シンプレックス法やアメーバ法とも呼ばれている。

多目的粒子群最適化

多目的粒子群最適化 (Multiobjective Particle Swarm Optimization: MOPSO) とは、粒子群最適化の手法を用いて多目的最適化を解く方法である [?].

PSO は、探索空間内において多くの粒子を用いて探索を行う、群知能の一種であり、魚群や鳥の群れにおいて、1匹が食料を発見できたり、安全であったりといった意味で、良さそうな経路を発見すると、群れの残りは素早くそれに倣うという特徴を応用したアルゴリズムである。粒子群最適化によって解の探索を行っている様子を図 2.2 に示す。

MOPSO の基本的なアルゴリズムは、PSO と同様で、パレート解は無数に存在し、今までの単目的のように最良な解が明らかではないため、粒子の最良解である personal best, 粒子群全体での最良解である global best をどのように選択するかが問題となる。

また, gbest はパレート解の中から、なるべく多様な解を求めるために、解空間での密度を考慮し、密度の薄い所から重点的にサンプリングする方法が存在する。

多目的進化アルゴリズム

多目的進化アルゴリズムは2007年に提案された [?] アルゴリズムのことであり、多目的最適化問題を、スカラー化する関数によって、複数の単目的問題に分割して解く手法のことである。スカラー化する関数には、Weighted Sum や Tchebycheff, Achievement Scalarizing Function などがあり [?], 中でも Weighted Sum は最も単純で、各目的関数に重み付けをした値を合計し、スカラー化した関数値とするものでとなっている。多目的進化アルゴリズムは初期化、交叉、突然変異、評価点の更新、解の更新といった流れを、終了条件を満たすまで繰り返すというアルゴリズムになっている。

§ 2.3 スクレイピングによるデータ収集

現在, cookpad や クラシル, おいしい健康, ボブとアンジー [?] などの料理レシピサイトと呼ばれるサイトが多数存在する。これらのサイトには、料理名、料理の画像、和食や洋食、主菜や副菜などの料理のジャンル、麺類や丼もの、鍋料理などの料理タイプ、料理につかう材料の名前とその材料の数、調理の工程、摂取カロリー、調理時間、得られるすべての栄養素などの情報がレシピサイトに掲載されている。

レシピサイトの1つであるおいしい健康の料理レシピ名、レシピの画像が乗っているページの例を図 2.3 に示す。また、生鮮食品や加工食品、畜産品などの最低、平均、最高販売価格の価格動向を先月や前年同月と比較している情報を提供している Web サイトも存在している (図 2.4 参照)。

本研究では、献立作成システムにおいて摂取栄養量をみたすかどうかという制約条件のもと献立を作成するため、料理から摂取できる栄養量をできるだけ細かく掲載されている複数の料理レシピサイトからレシピデータを取得する。また、食品価格動向を調査している Web サイトである小売物価統計調査による価格推移というサイト [?] から、食材とその価格のデータを取得する。また、これらのデータはスクレイピングという手法で取得する。

スクレイピング

スクレイピングとは、データを収集し、かつ目的に合わせて加工することである。特に、Web 上から必要なデータを取得することを、Web スクレイピングと呼ばれている。Web スクレイピングの流れについて図 2.5 に示す。様々なツールやプログラミングでスクレイピングを自動化することで、Web データの収集にかかる手間や時間は大幅に削減が可能である。

スクレイピングと似ている意味の言葉にクローリングがある。クローリングとは、Web 状態で様々なサイトを巡回し、情報の保存や複製など様々なことを行うことを指す。クローリングとスクレイピングはともに情報を収集手段ではあるが、クローリングが巡回に焦点を当てている一方でスクレイピングは情報の抽出に焦点を当てている。

また、企業や公共機関は、情報やデータを提供してくれることもあり、その際に使われている仕組みは API と呼ばれている。クローリングやスクレイピングをする前に、必要な情報が API によって提供されているかどうかまず確認することが大切になる。

Web スクレイピングに主に用いられるツールとして、BeautifulSoup4 や、Selenium がある。ログインやボタンのクリックなどの、マウス操作が必要な Web サイトや、JavaScript で記述されている Web ページのスクレイピングするときは Selenium が用いられている、それらの処理を必要としない Web サイトには、高速でスクレイピングができる BeautifulSoup4 が使用されることが多い。

Beautiful Soup4

BeautifulSoup4 とは、Web サイト上の HTML から、必要なデータを抽出するための Python のライブラリである。Beautifulsoup4 でスクレイピングする際、最初に対象の Web ページから HTML を取得する必要がある。

HTML を取得する方法として、同じく Python のライブラリである、Requests の get 関数や、Selenium の page_source 関数を使うなどの方法がある。上記の方法によって取得された HTML テキストを、BeautifulSoup4 の BeautifulSoup 関数

に渡すことで、BeautifulSoup オブジェクトを作成することができる。また、そのオブジェクトから class を検索することで Web サイトの必要な情報を抽出する。

class を検索するときに、条件を満たすひとつの要素を取得する select_one 関数や、条件に合う条件のすべてを取得する select 関数、find 関数などがある。select と find の違いは引数を指定する条件の指定方法がある。前者は、CSS セレクタを指定して要素を取得し、後者は class 名や属性キーワードを指定して検索し、class を取得する。これらの関数から取得した Tag オブジェクトである要素から、内部テキストのみを取得するためには、get_text 関数を使用することで取得することができる。



(a) 料理名とイメージ

材料 1人分		使用量	買い物量 (目安)
生鮭 (切り身)		90 g	
塩		小さじ1/6弱 (0.8 g)	
トマト		75 g	
しめじ		50 g	
にんにく		2 g	
パセリ (お好みで)		1.5 g	
オリーブ油		大さじ1/2 (6 g)	
A			
オリーブ油		小さじ1/2 (2 g)	
塩		小さじ1/6 (1 g)	
水		大さじ1 (15 g)	

(b) 得られる栄養素量と必要食材料

図 2.3: おいしい健康のレシピページ例 [?]

Selenium

Selenium は、Web ブラウザの操作を、自動的に操作することを可能にするライブラリである。元々は、Web アプリケーションの UI テストだったり、JavaScript のテストをする目的などで開発されていたが、テスト以外にも、Web サイトのクローリングや、タスクの自動化など、多岐にわたる用途で利用されている。

スクレイピングしたレシピ情報は、1つの料理につき1つの CSV ファイルで出力され、保存される。また、食材と価格データは1つの CSV ファイルにすべて出力する。それらの CSV ファイルを1つのデータベースに蓄積し、本研究で使用する自動献立作成システムの入力データとして利用する。

Web サイトからテキストをスクレイピングする際には、Python で記述したプログラムを使用する。まず、Python のライブラリである urllib を使って、目的の Web ページの URL を渡すことで、アクセスした際の HTML データを抽出する。次に、HTML や XML を解析することができる Python のライブラリの1つである BeautifulSoup4 を用いて Web ページ内の必要な要素を取得する。

上図のレシピデータ例に含まれている材料名を、食材のデータの材料名と照会し、その材料の必要な量と販売単位、販売価格から、各材料にかかる費用を全て計算し、料理にかかるコストを各レシピごとに計算する。レシピデータに含まれる材料名と食材価格データに含まれる材料名を照らし合わせる際に、微妙に違いが発生することがある場合、2つの材料名の文字列がどれほど一致しているかという類似度を計算し、類似度がしきい値よりも大きい場合に一致しているとしてコストの計算を行う。しきい値は、類似度計算に用いた Python のライブラリ関数にて、デフォルトの値である 0.65 を用いる。

文字の類似度を測定する際には、Python に標準で搭載されているライブラリである difflib を利用する。類似度計算をし、一致するものが見つからなかった場合は、ショッピングサイトである楽天市場の食品のカテゴリから、その材料名で検索をする。その後材料名とその材料の価格あたりの量をスクレイピングして食材価格データベースに追加する。ここで、類似度計算でしようした difflib とその技術について説明する。

difflib

difflib は文字比較を行うために使う python 標準モジュールである。difflib は、2つの文字列の類似度を表示する SequenceMatcher クラスや、リストからキーワードに類似



図 2.4: 食材価格動向調査サイトの例 [?]



図 2.5: Web スクレイピングの流れ

した文字列を抽出する `get_close_matches` 関数などの機能がある。SequenceMatcher クラスは、文字列同士の連続する共通部分を抜きとり、その抜き出した文字列の前後に対しても同様の処理を繰り返す、ゲシュタルトパターンマッチングというアルゴリズムを使用して、文字列の類似度計算処理とその表示を行う。 `get_close_matches` 関数は、特定のキーワードに類似した文字列を取得するために、マッチさせたい文字列と、マッチさせる文字列のリストを指定するほかに、マッチされた文字列のうち、上位の何件までを返すのか、何%以上の一致率ならば表示をするかななどの指定も可能となっている。

ゲシュタルトパターンマッチング

`difflib` で用いられている技術であるゲシュタルトパターンマッチングは、2つの文字列の類似度を判定するために用いられるアルゴリズムである。このアルゴリズムは Ratcliff, Obershelp によって 1983 年に考案された [?]. このアルゴリズムは、Ratcliff/Obershelp Pattern Recognition と呼ばれることもある。2つの文字列 S_1 , S_2 の類似度 D_{ro} は、

$$D_{ro}(S_1, S_2) = \frac{2K_m}{|S_1| + |S_2|} \quad (2.3)$$

で表すことができる。ここで、 K_m はマッチした文字の数であり、 D_{ro} は 0 から 1 の範囲となり、1 に近いほど類似度が高く、0 に近いほど類似度が低くなっている。

UXが反映されるシステム

§ 3.1 制限食を考慮した対話型献立作成

献立を作成するにあたって、人によってはアレルギーを含む食品や生活習慣病による制限食を考慮しなければならない。制限食とは、個人の健康状態、病気の状態に合わせてカロリーや塩分などを制限する食事のことである。身体の状態に応じてある程度の制限を加えた食事療法は、間接的な疾病の改善や病気を悪化させないための重要な役割をはたしている。食事療法には、生活習慣病をはじめとする病気の予防したり、健康診断で「要注意」と診断された場合に状態を改善することができたり、すでに病気と判断された場合に病気を悪化させないなどの効果がある。制限食には様々な種類があり、病態にあったものを選択する必要がある。ここで、代表的な制限食の種類を紹介する [?].

塩分制限食

塩分制限食とは、摂取塩分量を1日6g未満に制限した治療食のことである。食塩は、体内で生命維持に不可欠なナトリウムと塩素から構成されており、ナトリウムは体内水分量を維持し、神経や筋肉の生体機能の働きに必要である。しかし、摂りすぎると高血圧になる可能性になる。厚生労働省によると、高血圧や循環器疾患、胃がんなどの予防のため18歳以上男性8.0g/日未満、18歳以上女性7.0g/日未満と目標設定している。また、すでに高血圧症上のある場合は、「高血圧治療ガイドライン2014」では、1日の食塩摂取量を6.0g未満を目標としている [?].

脂質制限食

脂質制限食とは、食事に含まれる脂質をコントロールするとともに、ほかの栄養素等を必要量が確保できるように配慮された、一部の脂質異常症などに対応するための治療食である。脂質の摂りすぎると、中性脂肪や悪玉コレステロールを増加させ、肥満や脂質異常症を引き起こす場合がある。そのため、コレステロール摂取量を調整する必要がある。脂質制限食の目安として、1日のコレステロールの摂取量を200g以下にすることが推奨されている [?].

カリウム制限食

カリウム制限食とは、カリウムの代謝が困難になった人に向けた、カリウムの摂取量を制限した食事のことである。カリウムは細胞の浸透圧を維持したり、水分を保持したりする役割を果たしているが、浸透治療が必要な人はカリウムを摂取しすぎると不整脈を起こすことがあるため、カリウムの摂取量を調整する必要がある。カリウム制

限食の目安として食事一食当たりのカリウムの摂取量を 2000mg 以下に減らすことが推奨されている [?].

食事療法は生活習慣病治療の基本であり，合併症やさらなる悪化を防ぐには正しい食事療法を毎日続ける必要がある．また，食事療法による制限食は，患っている病気によって制限する栄養素が異なる場合がある．次に，食事療法によって予防や改善ができる主な生活習慣病を紹介する．

糖尿病

糖尿病とは，インスリンというホルモンの不足や作用低下が原因で高血糖状態が続く病気である．糖尿病は 1 型と 2 型に分けられており，1 型糖尿病は，主に自己免疫によって膵 β 細胞の破壊を生じ，インスリンの欠乏を来して発症する糖尿病である．2 型糖尿病はインスリン分泌量低下を来す複数の遺伝因子に，過食，運動不足などの生活習慣に起因する内臓脂肪型肥満が加わり発症する糖尿病である．

糖尿病を改善するために特に調整しなければならない栄養素は炭水化物，脂質，食物繊維である．炭水化物においては，1 日あたりの炭水化物摂取量を 100g 以下とする炭水化物制限が，肥満の是正に有効だとする研究結果から，糖尿病治療における炭水化物制限の有効性が注目されている [?]. また，脂質においては，日本糖尿病学会によると脂質の摂取量を必要推定エネルギーの 15～25% に抑えることが推奨されている．また，食物繊維において，糖尿病の発症リスクとの定量的解析を試みたメタ・アナリシスでは，食物繊維の平均摂取量は 20g/日を超えた時点から有意な低下傾向が見られている．

腎臓病

腎臓病とは，腎臓の糸球体や尿細管が冒されることで腎臓の働きが悪くなる病気のことである．腎臓の機能は一度失われると，回復することがない場合が多く，慢性腎不全と言われる病態になることがある．腎機能障害が進行してきた場合には，たんぱく質制限，塩分制限，カリウム制限などの食事療法を行うことにより，腎機能障害の進行を抑え，慢性腎臓病の合併症を予防することができる．たんぱく質の摂取量を制限することによって，腎機能低下の原因の一つである尿たんぱく，高リン血症の発生を軽減することができる [?]. 日本腎臓学会のガイドラインでは，たんぱく質制限を行う場合は，1 日のたんぱく質の摂取量を標準体重当たり 0.6～0.7g とすることが推奨されている [?].

また，塩分の摂取量を制限することにより血圧が低下し，末期腎不全に陥るリスクが低くなることがわかっている．日本腎臓学会のガイドラインでは，腎臓病患者の食塩摂取量として，1 日の摂取量が 3g 以上，6g 以下が推奨されている [?]. また，腎機能が低下すると，体内のカリウムの排泄も低下し，「高カリウム血症」を患う可能性がある．したがって，カリウム制限が必要となる．血清カリウム値 5.5mEq/L 以下を目標に 1 日カリウム摂取量を 1500mg 以下に制限する必要がある．

脂質異常症

脂質異常症とは，血液中の脂質の値が基準値から外れた状態のことをいう．血液中の

表 3.1: 考慮する生活習慣病

	糖尿病	腎臓病	脂質異常症	高血圧
たんぱく質(g)	健常者と同じ	標準体重当たり 0.6~0.7g	健常者と同じ	健常者と同じ
脂質(g)	総エネルギーの 15~25%	健常者と同じ	総エネルギーの 15%以下	健常者と同じ
炭水化物(g)	100g/日以下	健常者と同じ	健常者と同じ	健常者と同じ
塩分(g)	設定なし	3.0g/日以上6.0g/日以下	設定なし	6.0g/日未満
食物繊維(g)	20.0g/日以上	設定なし	20.0g/日以上	20.0g/日以上
カリウム(mg)	設定なし	1500mg/日以下	設定なし	3510mg/日以上
コレステロール(mg)	設定なし	設定なし	200mg/日以下	設定なし

LDL コレステロール（悪玉コレステロール）、HDL コレステロール（善玉コレステロール）、トリグリセライド（中性脂肪）の値のいずれかが異常値であれば、脂質異常症と診断される。脂質異常症は、動脈硬化生疾患、特に心筋梗塞及び脳梗塞の危険因子となる疾患である。コレステロール過剰に摂取すると血中のコレステロール値が上昇し、脂質異常症が重症化してしまうため、コレステロールの摂取量を調整する必要がある。

また、日本動脈硬化学会による「動脈硬化性膝下に予防ガイドライン 2017 年版」では、1 日のコレステロールの摂取量を 200mg とすることにより、コレステロール低下し、脂質異常症の重症化を防ぐことが期待できるとしている。ほかにも、厚生労働省によると、1 日の食物繊維の摂取量を 20g 以上にすることや 1 日当たりの脂質の摂取量を総エネルギーの 15%以下にする推奨している [?].

高血圧

高血圧とは、収縮期血圧及び拡張期血圧のいずれかが基準値を超えて上昇した状態で、診察室血圧では 140/90mmHg 以上と定義されている。高血圧が続いていて動脈硬化が進むと、動脈硬化が起こった部位ごとに様々な症状が現れる。

高血圧の要因として、塩分を過剰に摂取することによる血圧上昇が大きな要因となるため、塩分制限が必要となっている。日本高血圧学会による「高血圧治療ガイドライン 2019」によると、高血圧者の減塩目標を食塩 6g/日未満としている。また、カリウム摂取量増加によって高血圧者にとって血圧低下効果を認めた。厚生労働省によると、カリウムの摂取量を 3510mg 以上摂取することが推奨されている。さらに、1 日の食物繊維の摂取量を 20g 以上にすること推奨されている [?].

人によって食物アレルギーを有しておりアレルギーを含むレシピが含まれている場合がある。もしアレルギーに含まれる食品を摂取すると皮膚や、呼吸器、消化器など身体のさまざまな臓器にあらわれる。アレルギー症状は 1 つだけがあらわれる場合もあれば、急に複数の臓器に症状があらわれることもある。この症状に、さらに血圧低下や意識障害など急激に全身の症状が進行する場合を「アナフィラキシーショック」と呼び、生命の危険にまで及ぶことがある。

食物アレルギーを有する人には、症状が出ないように原因となる食品を除去する「除去療法」がある。除去療法とは、原因となる食べ物を除去することであり、例えば、卵アレルギーの場合は卵が含まれる食品を摂取しないなど除去をするなどがあげられる。また、除去療法をする場合、特に複数の食物アレルギーがある場合には栄養バランスが取れなくな



図 3.1: 考慮するアレルギー [?]

る場合があるので注意する必要がある [?]. 本研究は健常者のほかに、制限食を必要とする生活習慣病を患った人、生活習慣病を予防したい人、アレルギーを持っている人でもバランスが取れた献立が出力されるような献立作成システムを作成することを目的とする. 本研究で考慮する生活習慣病の数値をまとめたものを表 3.1 に示す. 対象となる献立作成システムは糖尿病, 高血圧, 脂質異常症, 腎臓病とする. また, アレルギーの対象項目として, 重い症状を引き起こしやすい, あるいは症例数が多く, 「特定原材料」として表示義務がされている 7 品目と, 「特定原材料」に準ずるものとして表示が推奨されている 21 品目の計 28 品目とする.

対話型とは, 利用者とシステムがディスプレイなどの出力装置, キーボードやマウスなどの入力装置を介して会話をするように互いに指示や応答をしながら作業を行う処理方式のことである. 対話型で処理を行うソフトウェアやシステムの具体例としては, ユーザが次に選択したいものをディスプレイ上の音声や画像, 動画などの形で提示することや, 操作するユーザの意図を汲み取り, それに対して反応を返したりすることなどが挙げられる.

対話型システムにおける「スライダー」は, ユーザーが視覚的に値を調整できるインターフェース要素の一つで, 特定の範囲内で数値を選択するためのツールだ. ユーザーはスライダーをドラッグすることで, 選択肢を細かく調整することができる. 例えば, 音量の調整, 明るさの変更, 評価の入力など, 数値的な選択が必要な場面でよく使われる.

スライダーの特徴範囲の指定: スライダーには最小値と最大値が設定され, その範囲内でユーザーが自由に値を選ぶことができる. 視覚的な操作: スライダーは横や縦に並べられたバーとして表示され, ユーザーがドラッグすることでリアルタイムで値が反映される. これにより, 直感的に操作ができる. 数値表示: スライダーの下や上に, 現在の値を数字として表示することも多い. これにより, ユーザーは具体的な数値を確認しながら調整を行える. ハプティックフィードバック: 一部のシステムでは, スライダーを操作する際に振動などのフィードバックを与えることがあり, 操作感を向上させる. 使用例音量調整: ユーザーが音量を増減するために, 音量スライダーを動かすことで, リアルタイムに音量の変化が反映される. フィルタリング: 商品検索やフィルタリング機能において, 価格帯や評価をスライダーで調整し, 検索結果を絞り込む. 設定調整: ウェブサイトやアプリの設定画面で, ユーザーがフォントサイズやテーマの色調整をスライダーで行う. スライダーはシンプル

でありながら、ユーザーにとって直感的に操作できるため、特に視覚的なインターフェースが重要なシステムで重宝される。

§ 3.2 分析データの前処理

ロジスティック回帰分析を行うために必要なデータの前処理について、以下の作業を行った。これらの手順は、スクレイピングした献立の情報をもとに、モデルが適切に学習できるようにデータを整形するための重要なステップだ。

1. 食材の有無の処理最初に行ったのは、各献立に含まれる食材の有無を特徴量として取り扱うことだ。献立情報をスクレイピングし、それぞれの献立に登場する食材があるかどうかを確認。例えば、「鶏肉」「玉ねぎ」「キャベツ」などの食材を列挙し、それぞれがその献立に含まれているかどうかを二値（1 または 0）で表現した。この処理により、各食材がその献立に存在するかないかを明示的に特徴量として扱えるようになる。この特徴量は、モデルにとって食材の有無が結果にどう影響するかを学習するために非常に有効だ。

2. 調理工程から調理動作のカウント次に、献立に含まれる調理工程をもとに調理動作の回数をカウントした。調理工程に記載されている動作（例：「炒める」「煮る」「焼く」など）をリストアップし、それぞれの献立がどれくらいの回数でその動作を行うのかを数えた。この特徴量を作成することで、献立がどれほど複雑か、あるいは調理にどれくらいの手間がかかるかといった情報を反映させることができる。調理動作の数が多ければ、一般的に調理時間が長くなる可能性があるため、時間やカロリーといった他の特徴量と組み合わせて分析できる。

Python の標準ライブラリである `re` は、正規表現（Regular Expression）を使って文字列の検索、抽出、操作を効率的に行うためのツールだ。正規表現は、文字列の中で特定のパターンを見つけたり、操作したりするのに使われ、テキスト処理やデータ解析でかなり重要な役割を果たす。

`re` ライブラリの概要正規表現の基本正規表現は、特定の文字列パターンを柔軟に定義して検索するための記法だ。たとえば、単純なキーワード検索だけでなく、「任意の文字列」「特定の文字の繰り返し」「数字やアルファベットのみに一致」など、複雑な条件を処理できる。

Python の `re` ライブラリには以下のような主な機能がある：

検索文字列中で特定のパターンが出現する場所を探す。抽出一致した部分をリストとして取得する。置換パターンに一致する部分を他の文字列に変える。分割パターンに基づいて文字列を分割する。

`re` ライブラリの主な関数 `re.search(pattern, string)` 文字列中で最初に一致する部分を探し、そのマッチオブジェクトを返す。

`re.findall(pattern, string)` 一致するすべての部分をリスト形式で返す。

`re.match(pattern, string)` 文字列の先頭部分がパターンに一致するかを確認する。

`re.sub(pattern, repl, string)` 一致する部分を指定した文字列に置き換える。

`re.split(pattern, string)` パターンに基づいて文字列を分割する。

正規表現を使うメリット柔軟性正規表現を使うことで、単純なキーワード検索だけでなく、より複雑な条件にも対応可能。例えば、「特定の文字列の前後に何かがある場合だけ一

致」なども簡単に書ける。

効率的な処理テキストデータの中からターゲット情報を迅速に抽出できるため、大量のデータでも適用しやすい。

非構造データを構造化データに変換テキストの中から有用な特徴を抽出し、モデルに入力できる構造化データを生成できる。今回のコードでは、非構造的な「作り方」テキストを調理動作のカウント値に変換している。

3. 栄養素量の単位削除スクレイピングした献立の情報には、栄養素量の単位（例えば、「g」や「ml」）が含まれていることがある。この単位をそのままにしておくと、数値として比較する際に問題が生じる。そのため、栄養素量の数値だけを抽出し、単位を削除した。これにより、異なる栄養素の量を単位に依存せずに比較でき、さらに数値データとして解析できるようになる。例えば、「100gの鶏肉」「200mlのスープ」といったデータから、単位を取り除き「100」「200」という純粋な数値のみを使用して分析を行う。

単位削除に使われるライブラリの説明

`applymap` メソッドは、`pandas` の `DataFrame` オブジェクトに対して、すべてのセルに関数を一括適用するために使用されるメソッドだ。`DataFrame` 内の各要素に対して効率的に処理を行いたい場合に有用で、例えば数値演算や文字列変換などができる。使い方としては、`df.applymap(func)` と書き、`df` は対象の `DataFrame`、`func` は各セルに適用したい関数だ。例えば、`DataFrame` のすべてのセルに対して二乗を計算したい場合、`df.applymap(lambda x: x ** 2)` と記述する。このメソッドは、列単位での処理が必要な場合は `apply` を使った方が適切で、複雑な処理には注意が必要だ。例えば、データの単位（g や kcal など）を削除するためには、`applymap` を使って各セルに対して処理を行うことができる。具体的には、文字列から数値部分を抽出する関数を定義し、`df.applymap(remove_units)` で `DataFrame` のすべてのセルから単位を削除することができる。

`map` メソッドは、`pandas` の `Series` オブジェクトに対して関数を適用するために使用される。一方、`applymap` は `DataFrame` の各セルに対して関数を適用するが、`map` は `Series` の要素に対してのみ関数を適用する。つまり、`map` は主に列単位での処理を行いたい場合に使用される。例えば、単一の列に対して値を変換したり、辞書やマッピング関数を使って置き換えを行いたい場合に便利だ。

`map` の使い方としては、`series.map(func)` と記述し、`series` は対象の `Series`、`func` は適用する関数だ。例えば、`df['列名'].map(lambda x: x ** 2)` のように、特定の列に対して演算を行うことができる。また、`map` は辞書や `Series` を使って値の変換にも使える。例えば、`df['列名'].map('old_value': 'new_value')` という風に、特定の値を別の値に置き換えることができる。

違いとして、`applymap` は `DataFrame` のすべてのセルに対して関数を適用するのに対し、`map` は `Series` のみを対象とするため、複数の列にわたる処理が必要な場合には `applymap` を、単一の列の処理を行いたい場合には `map` を使うべきだ。

4. 時間帯フラグの設定次に、献立が提供される時間帯（朝食、昼食、夕食など）を示すフラグを設定した。献立の提供時間に基づいて、それがどの時間帯に該当するかを判定し、時間帯ごとのフラグを立てた。例えば、朝食の献立には「朝食」フラグに1を、昼食の献立にはフラグに2を設定する。このようにすることで、献立が時間帯に応じた特徴を持っているかどうかをモデルに学習させることができる。時間帯によって、食事内容や栄養素、

調理時間が異なる可能性があるため、時間帯を考慮した分析ができるようになる。

5. SMOTE によるクラス不均衡の解消ロジスティック回帰分析の際、SMOTE (Synthetic Minority Over-sampling Technique) を用いてクラス不均衡を解消した。データセットには、ある特定のクラス (例えば「ユーザが作れる献立」と「ユーザが作れない献立」の分類) のサンプル数が大きく偏っていることがよくある。SMOTE は、少数派のクラスのサンプルを合成して増やす手法で、これによりクラス間のバランスを取ることができる。SMOTE を適用することで、モデルが少数派のクラスを無視せず、バランスの取れた学習ができるようになる。この手法は、クラス不均衡が原因で精度が低下するのを防ぐために非常に重要だ。SMOTE の手法 SMOTE のアルゴリズムは、まず少数クラスのサンプルを選択し、それらのサンプルを基に新しいサンプルを生成する。生成の過程は以下のように進行する：

近傍のデータポイントの選択: 各少数クラスのデータポイントについて、K 近傍法 (KNN) などを使い、近くに位置する他の少数クラスのサンプルを選ぶ。一般的に、指定された数 (例えば $K=5$) の近傍サンプルが選ばれる。

新しいサンプルの生成: 対象となる少数クラスのデータポイント (元のサンプル) と、その近傍サンプルの間に線形補間を行う。この補間により、元のサンプルと近傍サンプルを結ぶ線分上に新しいサンプルが生成される。補間にはランダム性があり、異なる補間を行うことで多様な新サンプルが作成される。

新しいサンプルの追加: 生成された新しいサンプルは、元の少数クラスデータセットに追加され、全体のデータセットにおける少数クラスの比率が改善される。このプロセスを複数回繰り返すことで、少数クラスのデータ数を増加させ、クラス間のバランスを取る。

SMOTE の利点 SMOTE の主な利点は、少数クラスのデータポイントを単純にコピーするのではなく、新しいサンプルを生成することだ。これにより、モデルが多様なパターンを学習できるようになり、過剰適合を避けることができる。新たに生成されたサンプルは、既存の少数クラスデータの間で分布するため、元のデータの特徴をより多様に反映させることができる。

性能向上の可能性クラス不均衡が解消されることで、分類器は少数クラスのサンプルを正確に予測できるようになる。特に、決定木やロジスティック回帰などのモデルでは、少数クラスのサンプルが学習データに過少反映されると、予測精度が低下することがよくある。SMOTE を使用することで、少数クラスに対する予測精度を改善し、全体的なモデルの性能を向上させることができる。

SMOTE の注意点 SMOTE は非常に強力な手法だが、いくつかの注意点がある。まず、生成されたサンプルは元のデータに基づいて補間するため、場合によってはノイズのあるデータや極端なサンプルが生成されることがある。このような新しいサンプルが学習に悪影響を与える可能性があるため、SMOTE を適用する前にデータの前処理をしっかりと行うことが重要だ。

また、SMOTE はあくまで少数クラスのサンプルを増加させる手法であり、モデルのバイアスを完全に解消するわけではない。特に、極端に不均衡なデータセットでは、他の手法 (例えば、アンダーサンプリングや異なるアルゴリズムの使用) と併用することが効果的だ。まとめこれらの前処理を行うことで、ロジスティック回帰分析において有効なデータセットが整った。食材の有無や調理動作のカウントなどの特徴量を作成し、栄養素量の単位削除や時間帯フラグの設定により、データがより分析しやすい形になった。さらに、SMOTE に

よってクラス不均衡を解消することで、ロジスティック回帰モデルがより精度高く学習できるようになる。これらのデータの前処理は、モデルの性能向上に直結する重要な作業だ。

§ 3.3 ロジスティック回帰分析

ロジスティック回帰分析 (Logistic Regression) は、従属変数 (目的変数) が2つのカテゴリに分類される場合 (バイナリ分類) に使用される統計手法である [?]. この手法は、入力変数 (説明変数) と目的変数の間に線形関係がない場合でも、分類問題を解決するために広く利用される。

ロジスティック回帰の基本的なアイデアは、従属変数があるカテゴリに分類される確率を予測することである。回帰分析と名前はついてはいるが、実際には回帰モデルではなく、分類問題を扱うために使用される。従属変数は、0 または 1 の2つの値を取る確率変数であり、モデルの出力はその確率に関連付けられている。

ロジスティック関数

ロジスティック回帰では、ロジスティック関数 (シグモイド関数) を用いて、モデルの出力を確率として解釈する。ロジスティック関数は、以下の式で表される。

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \dots + \beta_n X_n)}}$$

ここで、 $P(Y = 1|X)$ は、 X に基づいて $Y = 1$ となる確率、 e はネイピア数、 β_0 は切片、 $\beta_1, \beta_2, \dots, \beta_n$ は回帰係数、 X_1, X_2, \dots, X_n は説明変数である。この関数は、入力された値に対して、出力が0から1の範囲に収束するため、確率として解釈可能となる。

回帰係数とオッズ比

ロジスティック回帰では、各説明変数の影響を回帰係数 β_j によって表現する。 β_j は、説明変数 X_j が1単位変化したときに、従属変数 $Y = 1$ となる確率のオッズ比がどれだけ変化するかを示す。

オッズ比は、あるイベントが発生するオッズ (確率/1-確率) の比率であり、ロジスティック回帰モデルにおいて重要な役割を果たす。オッズ比は、回帰係数 β_j を指数関数で変換することによって得られる。すなわち、

$$\text{オッズ比} = e^{\beta_j}$$

例えば、ある回帰係数 β_j が1である場合、そのオッズ比は $e^1 \approx 2.718$ となり、 X_j が1単位増加することで $Y = 1$ のオッズが約2.7倍になることを示す。逆に、 β_j が負であれば、オッズ比は1未満となり、 X_j の増加が $Y = 1$ のオッズを減少させることを意味する。

オッズ比を用いることで、各説明変数が目的変数に与える影響を直感的に理解することができる。オッズ比が1より大きければ、その変数が目的変数に対して正の影響を持ち、1より小さければ負の影響を持つことを示す。

回帰係数の推定

ロジスティック回帰では、モデルのパラメータ（回帰係数）を最尤推定（Maximum Likelihood Estimation: MLE）によって推定する。最尤推定は、観測データが最も高い確率で得られるようなパラメータの値を求める方法であり、ロジスティック回帰においても広く用いられる。

最尤推定法では、以下の尤度関数を最大化することにより、 β を推定する。

$$L(\beta) = \prod_{i=1}^n P(y_i|x_i, \beta)^{y_i} (1 - P(y_i|x_i, \beta))^{1-y_i}$$

ここで、 y_i は観測された結果、 x_i は対応する説明変数の値、 $P(y_i|x_i, \beta)$ は、 x_i と β に基づく予測確率である。

モデル評価

ロジスティック回帰分析のモデル評価には、主に以下の指標が使用される。

- **精度 (Accuracy)**: 正しく予測されたデータの割合。
- **適合率 (Precision)**: 正と予測された中で実際に正であったデータの割合。
- **再現率 (Recall)**: 実際に正であるデータの中で正と予測された割合。
- **F 値 (F1-score)**: 精度と再現率の調和平均。モデルのバランスを評価するために使用される。

また、ロジスティック回帰モデルの適合度を評価するためには、**擬似決定係数**や **AIC（赤池情報量基準）**などの指標が使用されることがある。これらの指標は、モデルの精度や過剰適合を防ぐために役立つ。

解釈と応用

ロジスティック回帰分析の最大の特徴は、回帰係数が各説明変数の影響を示すことである。回帰係数 β_j は、説明変数 X_j が1単位変化したときに、従属変数 Y が1に分類される確率の対数オッズがどれだけ変化するかを示す。

例えば、 $\beta_j > 0$ であれば、 X_j が増加することで $Y = 1$ の確率が増加することを意味し、 $\beta_j < 0$ であれば逆に確率が減少することを意味する。オッズ比を用いることで、これらの影響を直感的に把握することができ、モデルの解釈が容易になる。

ロジスティック回帰分析は、医療分野での病気予測や、マーケティングにおける顧客の購入予測、金融分野でのクレジットカード不正使用の検出など、様々な分野で活用されている。

提案手法

§ 4.1 UX データの収集

本研究では、献立作成を多目的最適問題としてとらえる。目的関数を献立に含まれる料理の調理時間と調理にかかるコストの最小化とし、制約条件を必要栄養素や摂取カロリーなどとして多目的最適化を行う。そして、二つの目的関数の最小化と、複数の制約条件に基づいた、パレート最適である献立を出力する。なお、献立の日数や料理の準備にかかる時間などのユーザの選好がかかわる制約条件はユーザが選択できる。多目的最適化問題を解く手段として、NSGA-II という遺伝的アルゴリズムを多目的最適化に応用した手法を用いる。なお、このシステムは pymoo という Python プログラム用いて、NSGA-II によって組み合わせ最適化問題を解かせるよにプログラムの記述を行う。NSGA-II によって出力したパレート解である献立は、摂取栄養量やカロリー、主菜と副菜の数、アレルギー制限などの制約条件をみたし、調理にかかるコストおよび調理にかかる時間が最小化された、パレート最適な集合として複数出力される。この出力された複数の献立のうちユーザに最もあった献立をユーザ自身に選択してもらう。そして、その後 UX 項目についてユーザに回答してもらう。UX データ収集に使われる Flask について説明する。

Flask

Flask は、Python で書かれた軽量な Web アプリケーションフレームワークで、シンプルで拡張性に優れた設計が特徴。Flask は「マイクロフレームワーク」と呼ばれ、最小限のコア機能を提供し、その上に必要なライブラリや機能を追加していく形で柔軟に構築が可能。この特性により、開発者はプロジェクトのニーズに応じて自由に機能を追加したり、不要なものを省いたりでき、スケーラブルで効率的なアプリケーションの開発ができる。

Flask の基本的な特徴には以下の点がある

簡潔な設計: Flask は、シンプルな API と構造で、初心者でも理解しやすい。標準のテンプレートエンジンである Jinja2 や、URL ルーティングを容易に設定できる仕組みが組み込まれており、迅速な開発が可能。

拡張性: Flask は、必要に応じてデータベース、認証、フォーム処理、ファイルアップロード、セッション管理など、追加の機能を簡単に組み込むことができる。これにより、最初は小規模なアプリケーションとして開始しても、後に大規模なシステムへと成長させることができる。

テンプレートエンジン: Flask は Jinja2 という強力なテンプレートエンジンを使用して、HTML や XML などのドキュメントを動的に生成できる。これにより、ユーザーに対して動的に生成されるコンテンツを提供する Web アプリケーションの開発が可能。

デバッグ機能と開発サーバ: Flask は開発中に役立つデバッグモードを備えており、コードの変更を即座に反映できる開発サーバも内蔵。これにより、開発者は素早くフィードバックを得ることができ、アプリケーションの開発が効率的に行える。

軽量性と柔軟性: 他の Web フレームワークと比較して、Flask は非常に軽量で、無駄な機能を持たないため、最小限のリソースで動作する。そのため、非常に小さな Web アプリケーションから、大規模な API システムまで幅広い用途に対応できる。

Flask はそのシンプルさと拡張性から、プロトタイピングや小規模から中規模の Web アプリケーションの開発に非常に適しており、API を公開したり、データ分析の結果を Web 上で表示したりするシステムにもよく使用される。また、Flask の豊富なプラグインを活用することで、特定の機能を簡単に実装でき、開発スピードを大幅に向上させることができる。

このように、Flask は Web アプリケーション開発における強力なツールで、そのシンプルで柔軟な設計が多くの開発者に支持されている。

変数

各変数は、対象の日数を D 、日の番号を k 、レシピの数を R 、料理レシピが献立に含まれている場合に 1、含まれていない場合に 0 の値をとる献立フラグを r_{ki} 、料理レシピが主菜の場合に 1、副菜の場合に 0 の値をとる主菜フラグを σ_i 、 i 番目の料理レシピの調理時間を T_i 、 i 番目の料理レシピの食材コストを G_i 、 i 番目の料理レシピの l 番目の摂取栄養素を f_{il} 、 l 番目の栄養素の制約の最大値を F_l^H 、最小値を F_l^L 、 i 番目の料理レシピの摂取カロリーを C_i 、基礎代謝量の制約の最大値を B^H 、最小値を B^L 、朝食、昼食、夕食における最大調理時間をそれぞれ τ_1, τ_2, τ_3 とする。

また、入力画面でアレルギーが選択されていた場合に 1、選択されていない場合に 0 の値をとるアレルギーフラグを x_i 、各制限食が選択されていた場合に 1、選択されていない場合に 0 をとる制限食フラグを y_i 、制限食における栄養素の制約の最大値を E_l^H 、最小値を E_l^L とする。

本研究で提案する、自動献立作成システムにおける多目的最適化問題の目的関数と制約条件は、上記の変数を用いて下の式によって定式化される。

< 定式化 >

$$\text{minimize} \quad \sum_{k=1}^{3D} \sum_{i=1}^R r_{ki} T_i \quad (4.1)$$

$$\text{minimize} \quad \sum_{k=1}^{3D} \sum_{i=1}^R r_{ki} G_i \quad (4.2)$$

$$\text{subject to} \quad F_l^L \leq \sum_i^R r_{ki} f_{il} \leq F_l^H \quad (\forall k, \forall l) \quad (4.3)$$

$$B^L \leq \sum_i^R r_{ki} C_i \leq B^H \quad (\forall k) \quad (4.4)$$

$$\sum_i^R r_{ki} T_i \leq \tau_1 \quad (k \% 3 = 1) \quad (4.5)$$

$$\sum_i^R r_{ki} T_i \leq \tau_2 \quad (k \% 3 = 2) \quad (4.6)$$

$$\sum_i^R r_{ki} T_i \leq \tau_3 \quad (k \% 3 = 3) \quad (4.7)$$

$$0 < \sum_i^R r_{ki} \sigma_i \leq 1 \quad (\forall k) \quad (4.8)$$

$$0 \leq \sum_i^R r_{ki} (1 - \sigma_i) \leq 3 \quad (\forall k) \quad (4.9)$$

$$\sum_{k=1}^{3D} r_{ki} \leq 1 \quad (4.10)$$

$$0 \leq \sum_i^R r_{ki} x_i < 1 \quad (\forall k) \quad (4.11)$$

$$E_l^L \leq \sum_i^R y_i r_{ki} f_{il} \leq E_l^H \quad (\forall k, \forall l) \quad (4.12)$$

目的関数

本研究の献立作成における多目的最適化問題を構成する目的関数と制約条件式について説明する．まず，目的関数は，式 (4.1) と式 (4.2) の 2 つであり，(4.1) は調理時間の最小化であり，(4.2) は食材コストの最小化である．0-1 変数である献立フラグを用いて，設定した日数での料理の組み合わせを表現する．

制約条件

制約条件は，式 (4.3) から式 (4.12) の 10 つである．式 (4.3) は摂取栄養量制約，式 (4.4) は摂取カロリー量制約，式 (4.5) から式 (4.7) は朝食，昼食，夕食における最大の調理時間制約，式 (4.8) と式 (4.9) は主菜は 1 つ，副菜は 3 つ以下で献立を構成する制約，式 (4.10) は献立の中に，同じ料理が存在しないようにする制約である．また，式 (4.11) は，入力画面でアレルギーを選択した時に，そのアレルギーが含まれるレシピが含まれないようにする制約であり，式 (4.12) は，入力画面で制限食が選択されたときの摂取栄養素量の制約である．

摂取栄養量制約は，1 日あたりに摂取する特定の栄養量に，下限と上限を設定して表現する．摂取カロリー量制約は，1 日あたりに摂取するカロリー量に，下限と上限を設定して表現する．朝食，昼食，夕食における最大の調理時間制約は，入力画面で入力した朝，昼，夜の各時間帯における献立にかかる調理時間をそれぞれ上限に設定した．

主菜は 1 つ，副菜は 3 つ以下で献立を構成する制約は，その料理が主菜であるか，副菜であるかを表現する 0-1 変数の主菜フラグを用いて，献立に含まれる主菜と副菜の下限と上限を表現する．

§ 4.2 UX を考慮した献立作成

UX を考慮した献立作成のために本システムでは遺伝的アルゴリズムを使用する．遺伝的アルゴリズム (Genetic Algorithm: GA) とは，近似解を探索するためのメタヒューリスティックアルゴリズムである [?]．メタヒューリスティクスアルゴリズムとは，特定の問題

だけに限らず、どんな問題に対しても汎用的に対応できるように設計された、アルゴリズムの基本的な枠組みのことである。

GA は、解の候補であるデータサンプルを遺伝子で表した個体を複数体準備し、適応度関数によって計算された適応度の高い個体を優先して選択し、交叉、突然変異や淘汰などの操作を繰り返しおこなうことで最適な解を導出する。GA は生命の進化過程に似ている様子からその名が付けられた。

GA の基本的な流れを以下に示し、各ステップの説明をする。また、この処理の 1 回の繰り返し単位は「世代」と呼ばれる。

1. ランダムで複数の個体を生成する
2. 集団の各個体それぞれの適応度を計算する
3. 各個体から「選択」、「交叉」、「突然変異」をし、次世代の個体を生成する
4. 現在の集団を新たに生成された集団で置き換える。
5. ステップ 2. にもどる

最初に生成される 1 世代目の個体はランダムで作成される。このとき、個体の各遺伝子のパラメータも乱数によってランダムに設定がされている場合が多い。

次に、その世代全ての個体の適応度を計算する。適応度とは、初期に生成した個体が評価値に対してにどれだけ適応できているかを評価する値である。一般に GA では、適応度関数は最大化、または最小化の形で定義されることが多い。

また、選択とは新しい世代を生み出すときの遺伝子の操作の 1 つであり、適応度によって次の世代の母集団を作成することである。適応度の高い個体ほど多くの子どもを生成するように選択を行われる。選択の方法としてルーレット選択、ランキング選択、エリート選択があり、どの方法も、最終的に生き残る個体を限定して母集団を最適解へ収束させる役割がある。

次に、本研究で扱う手法である、非優越ソート遺伝的アルゴリズム (Non-dominated Sorting Genetic Algorithm: NSGA-II) について説明する。NSGA-II とは、Deb らによって 2002 年に提案された [?], GA を、多目的最適化問題に拡張したものである。

NSGA は個体の評価方法を、Goldberg により提案された非優越ランキングソート [?] と、シェアリングを組み合わせたものを用いており、パレート最適的なアプローチによる手法の 1 つである。

非優越ソート (Non-Dominated Sort) とは、1989 年に Goldberg により提案されたアルゴリズムで、NSGA-II において適応度の高い個体を抽出するために用いられている個体のランク付けである。個体をランク付けし、同じランクの中でシェアリングを行う際に、ランクレベルのみでシェアリングを行うことによって、全個体についてシェアリングを行うよりも、計算にかかる負荷を軽減させることができる。

次に、多目的最適化で解いたパレート解のうち、ユーザにどのようにして最適な献立を出力させるかについて説明する。複数の目的関数の最小化または最大化を考える多目的最適化において、複数の目的関数を同時に満たすような解は存在せず、一方の目的関数が高い評価を得た場合、他方の目的関数は犠牲になってしまうトレードオフの関係になってしまうことが普通なため、目的関数が複数にある場合における解は、意思決定者にとって、最も好ましいものを選択できるようにすることが大事である。



図 4.1: 対話型処理のイメージ

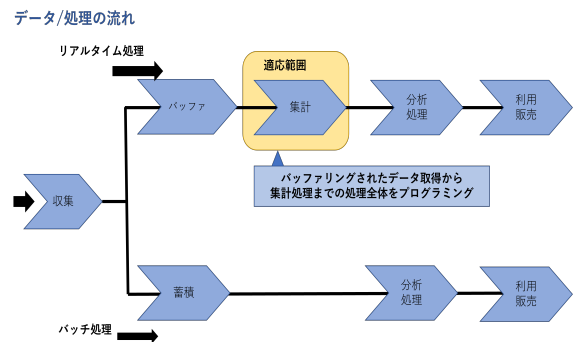


図 4.2: バッチ処理とリアルタイム処理

対話型処理を用いたパレート最適解を選好する従来事例として、多目的最適化問題に関して、制約式と目的関数に含まれるパラメータの決定などの問題の設定時に含まれるあいまい性と、意思決定者があいまいな目標を持つことを考慮した、対話型ファジィ満足化手法がある。この手法では、個体の作成から最適化処理の部分はアルゴリズムが担い、その最適化処理の過程における評価の部分行っている。このシステムでは、ランダムで生成された個体をユーザに画像で提示し、提示された画像に対してユーザが5段階評価をし、その評価に従って近似最適解を再度作成している [?].

また、単一目的の大規模な多目的離散最適化問題を、効率的に解を探索するためのアルゴリズムである、モジュラアプローチを用いて解き、それによって求められたパレート最適解集合の大きさを表示したのち、パレート最適解集合の大きさが決められた値以内になるまで繰り返しモジュラアプローチを用いて解き、縮小されたパレート最適解集合の中から各目的関数の重要度などの自分の選好条件に基づいて選好最適解を決定する手法が挙げられる。

他には、対話型 GA による近似最適解の探索を基本としつつも、GA の最適化処理の過程における、個体の適応度を評価をする部分を人間が行うといった手法も提案されている。一般的な GA での評価の役割は、評価関数が担っているが、対話型 GA では、この評価関数により個体の適応度を決定する部分を、人間が評価を行うようにしている。人間の意思決定を個体の適応度評価の過程に組み込むことにより、人間による主観的な評価が1つのシステムの要素となることから、対話型 GA は人の感性をシステムに落とし込むことが可能な手法である。

対話型 GA は、感覚や個人の好みなどといった、数値では表すことが困難な個人の感性を、対話型 GA による設計やデザインに取り入れることが可能となっているため、服飾やオフィスデザインや感性による様々な事柄への推薦、補聴器を使用する人の、聞こえに合わせるフィッティングなどへの研究に応用することが可能となっている。

意思決定者の選好解を求めるために、大きく分けて、以下の3つのアプローチがある。

1. 全て、もしくは十分に多くパレート最適解を求め、それを意思決定者に提示し、選好解を自分自身で決定してもらう。
2. 意思決定者の選好を表す実数値関数である、価値関数または効用関数を求め、それを最適化するような数理計画問題を解く。
3. コンピュータによって導出されたパレート最適解と、その解に基づく意思決定者の

局所的な選好情報を用いて、ユーザとコンピュータの対話を繰り返すことによって、選好解を決定する。

最初のアプローチでは、目的関数の数が少ない場合や、実行可能解が少数で、有限個しか存在しない場合に有効であるとされる。この方法で代表的なものとして、各目的関数に対する重みを用いて、問題を解く加重和最小化や、1つの目的関数を残し、他の目的関数に対する要求水準を制約条件に用いる、制約変換法などがある。

2番目のアプローチの、価値関数もしくは効用関数の同定について、多属性効用理論が知られており、目的関数間の独立性が十分確保されていることが重要となる [?]. 1番目のアプローチで挙げた、加重和目的関数を、価値関数もしくは効用関数として想定して、そのパラメータを同定するといった、このアプローチの簡略版も考えられる。

最後のアプローチは、対話型解法と呼ばれており、意思決定者が、システムとの対話をすることによって、複数ある目的関数をどのように選り好みするかといった、局所的な選好情報を用いて、パレート最適解から解を自動的に選択する、という方法である。

この方法は、コンピュータとユーザの両者の情報交換の仕方によるので、いくつかの方法が考えられるが、ユーザという人間が関わっているということから、ヒューマンフレンドリーである方法が望まれる。提案されてきた対話型解法として、意思決定者が目的関数に対する、望ましいと考える値である希求水準を設定して、それに最も近い解をパレート最適解から得るという、希求水準法などが挙げられる [?].

本研究では、対話型処理によって、ユーザに対して分かりやすく献立を、選択してもらいたいと考えたため、3番目のアプローチをとる。また、ユーザからのフィードバックをもとに、ロジスティック回帰分析を行う。しかし、回答データに偏りが生じた場合、不均衡なデータを扱うことになるため、分析結果の信頼性が低下する可能性がある。そこで、本研究では解析の前段階としてSMOTE (Synthetic Minority Over-sampling Technique) を使用し、データのバランスを調整することで、モデルの予測精度を向上させることを目指す。SMOTEは少数派クラスのデータを合成的に増加させ、より均衡の取れたデータセットを作成する手法であり、不均衡データによる問題を軽減できると期待される。

SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) は、不均衡なデータセットに対処するための手法で、少数派クラスのサンプルを合成的に生成する方法である。具体的には、少数派クラスのデータポイント間で新たな合成サンプルを作成することで、データセットのバランスを取る。SMOTEは、各少数派クラスのデータポイントを選び、その近隣のサンプルとの線形補間を行って新しいデータポイントを生成する。この過程により、少数派クラスのデータ数を増やし、学習アルゴリズムが少数派クラスの特徴をより学習しやすくなる。

SMOTEの具体的な処理の流れは以下の通りである：

少数派クラスの各サンプルに対して、その近隣にある k 個のサンプルを見つける。そのサンプルとの間に直線的な補間を行い、指定された数だけ新しいサンプルを生成する。生成されたサンプルは元のデータセットに追加され、バランスが取れたデータセットを形成する。この手法の利点は、単に少数派クラスのデータを複製するのではなく、合成サンプルを生成する点にある。これにより、過学習を防ぎつつ、少数派クラスの特徴を豊富に学

習させることができる。しかし、SMOTE には注意が必要で、特に高次元データやノイズの多いデータでは、生成されたサンプルが効果的でない場合もある。そのため、適切なパラメータ設定や前処理を行うことが重要である。

本研究では、SMOTE を適用することで、不均衡なデータセットの問題を解消し、ロジスティック回帰モデルの精度を向上させることを目指す。これにより、ユーザからのフィードバックを基にした分析結果がより信頼性の高いものになると期待される。

§ 4.3 提案システムの流れ

本研究で提案する制限食と多人数考慮した自動献立作成システムの流れを図??に示す。また、本システムの流れを説明する。

Step 1: 料理レシピ、食材価格のデータベースの作成

はじめに、Python を使って Web サイトからスクレイピングし、料理レシピと食材価格のデータベースを作成する。本研究で用いるスクレイピング手法として、Python のライブラリである BeautifulSoup4 を扱う。BeautifulSoup4 は Web サイト上の HTML から、取得したいデータを HTML 内のクラスや ID などの要素検索して抽出することができる。また、最初に対象の Web ページから HTML を取得する必要があるため、その際に HTML パーサーである Requests を用いる。これらのスクレイピング手法を用いて、レシピ情報を取得する。料理レシピサイトとして、「ボブとアンジー」のみを参考に行っている研究も存在しているが[?], 1つのレシピサイトでは出力されるレシピに偏りがあるという問題点があったため、本研究においては「ボブとアンジー」、「EatSmart」、「おいしい健康」の3つのサイトからスクレイピングした情報をレシピサイトとしている。また、料理情報とその食材の価格、販売価格の情報は、食材とその価格動向を載せているサイト、「小売物価統計調査による価格推移」からスクレイピングしている。

それぞれの料理レシピサイトからは、その料理から摂取することができる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、作り方などのデータがスクレイピング可能である。それらの各料理レシピデータはそれぞれ CSV ファイルに出力され保存されるようになっている。また、それぞれのレシピサイトから取得できる栄養素の単位などが異なる場合があるので、すべてのレシピサイトから取得できる栄養素の単位、順番を統一している。

また、食材価格データは全て1つの CSV ファイルに出力されるようになっている。食材から得られる栄養素は、食材価格データベースに付随して出力される。その後、料理レシピにて必要な食材名を、食材価格データベースから一致するものを検索し、見つかったときに、その必要な食材量を食材価格データベースの販売単位と価格から計算し、食材コストを算出する。

その際、料理レシピにおける食材名と食材価格データベースにおける食材名が微妙に違っていた時には見つからないため、Python のライブラリである difflib を用いて、ゲシュタルトパターンマッチングという手法で、2つの食材名の類似度を計算し、類似度の近い食材を検索して見つけるようにしている。

この手法を用いても食材価格データベースから見つけることができなかったときは、ショッピングサイトである楽天市場から、カテゴリを食品に設定してその食材について検索をか

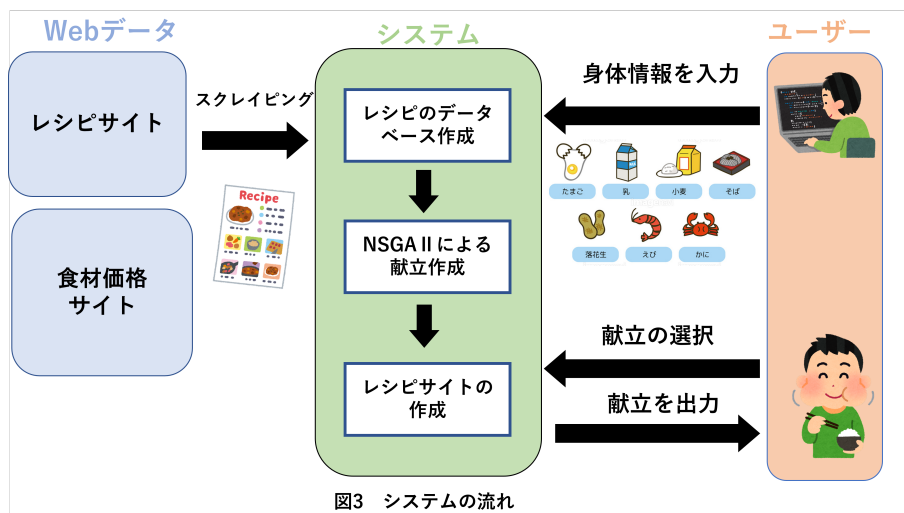


図 4.3: 提案手法の流れ

け、販売単位と値段をスクレイピングし、スクレイピングした食材価格データは、データベースの CSV ファイルに追加している。

Step 2: ユーザ情報と制約条件の入力

組み合わせ最適化を解くに当たって、設定する制約条件がユーザーの身体情報によって異なる。そのため、献立作成をする前にユーザに、入力情報を入力しておく必要がある。ユーザ情報を入力する手順としてまず、献立作成に必要な人数分のユーザ情報を利用者に入力してもらう。その後入力する情報として入力する人数を入力すると人数分のユーザ情報を入力する画面が出てくる。その中でそれぞれのユーザの名前、身長と体重、年齢、性別、アレルギー情報、予防したいまたはすでに患っている生活習慣病を入力する。入力された身体情報は身体情報データベースに蓄積されており、最適化処理を実行するときに使用する。

その次に、出力する献立の日数及び、朝、昼、夜に調理の準備にかかる時間を選択する。入力した日数は多目的最適化を解く際の選ぶ献立数を設定している。1日あたりに出力する献立は7品としているため入力した日数によって選択するレシピの数が異なる。また、朝、昼、夜に調理の準備にかかる時間は多目的最適化を解く際の朝、昼、夜にかかる時間の最大値として設定される。

Step 3: NSGA-II による多目的最適化と最適な献立の出力

次に、料理レシピデータ群から入力された情報をもとに作成された制約条件と目的関数に沿った料理レシピを選択するという組み合わせ最適化問題と捉え献立作成を行う。献立作成に用いるデータまたは変数として、料理レシピサイトと食材とその価格を載せているサイトからスクレイピングしたレシピデータ、食材価格、栄養素データと、ユーザーによって入力された身長と体重、年齢、性別から計算された基礎代謝量、推定エネルギー必要量、アレルギー情報、疾患情報を用いる。これらの情報を多目的最適化問題を NSGA-II によって解き、パレート最適な献立を出力する。

目的関数には UX 項目の最小化が与えられ、制約条件には、3 大栄養素の摂取量、摂取カロリー量、朝、昼、夕の時間帯別の調理時間合計、献立に含まれる主菜と副菜の数、アレルギーがある場合にそのアレルギーの材料が含まれないようにする、疾患を患っている場

合、その疾患にあった栄養素の摂取量の制限、出力される日数のうち、料理が被らないようにする、などの条件が設定されている。

自動献立作成における多目的最適化問題を定式化し、それをプログラム上に記述する際には、Python のライブラリである pymoo を用いた。pymoo は PSO や GA、多目的進化アルゴリズムや NSGA-II などの、単目的最適化問題や多目的最適化問題を解くための様々な手法をサポートしている。目的関数や制約条件が視覚的にわかりやすく記述できることや、自作関数の作成や用意した変数を最適化処理に組み込むことが容易なこと、今までスクレイピングで扱ってきた Python のプログラムで実装ができることから、本研究のシステムにて組み込むことにした。

Step 4: 対話型処理による献立の選択

h

Step 5: 出力された献立のユーザによる評価

e

Step 6: ロジスティック回帰分析

l

Step 7: 評価関数への組み込み

p

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となっている。まず、使用するレシピデータ数は3000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を使った。

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種類からスクレイピングする。urllibにより、目的のレシピサイトと食材価格サイトの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に、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??に示す。

3つのレシピサイトからはスクレイピングする情報として、その料理から摂取することができる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、作り方などをスクレイピングする。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の中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している際の実行画面を図??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、本研究はブラウザでシステムを用いている。そのため本研究では表示されない。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を可能と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を設定する。制約条件は、健康者の場合と制限食が必要な人の場合で異なる。まず、健康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、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。超えて摂取すると、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[?]、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のサイト[?]を参考にして、基礎代謝量と身体活動レベルの係数をかけ合わせたものを使用し

レシピの名前	主菜フラグ	調理時間	摂取カロリー・材料名	材料量	栄養素名	栄養量	コスト	朝食
うずら卵の蒸し物	0	45	170 (ひき肉蒸し)	たんぱく質11.5g		241	0	
うずら卵の蒸し物	0	45	170 ・うずら8個	炭水化物 6.2g		241	0	
うずら卵の蒸し物	0	45	170 ・生しいち8枚	糖質 5.1g		241	0	
うずら卵の蒸し物	0	45	170 ・片栗粉 少々	脂質 9.7g		241	0	
うずら卵の蒸し物	0	45	170 ・鶏ひき肉100g	食塩相当量1.2g		241	0	
うずら卵の蒸し物	0	45	170 ・酒 大さじ2	食物繊維 1.1g		241	0	
うずら卵の蒸し物	0	45	170 ・卵白 少々	ビタミンA160μg		241	0	
うずら卵の蒸し物	0	45	170 ・塩 少々	ビタミンB0.13mg		241	0	
うずら卵の蒸し物	0	45	170 ・こしょう少々	ビタミンB0.44mg		241	0	
うずら卵の蒸し物	0	45	170 ・うま味調味料少々	ビタミンB0.27mg		241	0	
うずら卵の蒸し物	0	45	170 (れんげ蒸し)	ビタミンB2.2μg		241	0	
うずら卵の蒸し物	0	45	170 ・うずら8個	ビタミンC4mg		241	0	

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	1.64265E+03	1	-	-
2	800	8.49100E+02	1.34299E+03	1	2.47000E+02	idea
3	1200	8.48100E+02	1.23841E+03	1	2.00000E+02	f
4	1600	8.20700E+02	1.16736E+03	1	3.16300E+03	idea
5	2000	7.87800E+02	1.09891E+03	1	2.02200E+03	idea
6	2400	7.79000E+02	1.03477E+03	1	4.85000E+02	idea
7	2800	7.16900E+02	9.78273E+02	1	8.54000E+02	idea
8	3200	5.68000E+02	9.29777E+02	1	1.58300E+03	idea
9	3600	4.22400E+02	8.85090E+02	1	1.97800E+03	idea
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	6.63149E+02	1	0.00000E+00	f
16	6400	3.96300E+02	6.33423E+02	1	1.19800E+03	idea
17	6800	3.96300E+02	6.06926E+02	1	0.00000E+00	f
18	7200	2.86900E+02	5.78641E+02	1	4.70000E+02	idea
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	4.93082E+02	1	0.00000E+00	f
23	9200	2.86900E+02	4.75208E+02	1	0.00000E+00	f
24	9600	2.86900E+02	4.57897E+02	1	0.00000E+00	f
25	10000	2.38800E+02	4.42169E+02	1	2.76000E+02	idea

図 5.2: 最適化処理の実行画面

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，アレルギーを持っている人と，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だが，4.2 章で述べた通り，糖尿病は，内臓脂肪型肥満によってインスリン抵抗性により発症する．そのため糖尿病の予防と改善には脂肪の是正が重要となってくる．また，厚生労働省によると，1 日あたりの炭水化物摂取量を 100 g 以下とする炭水化物制限が，肥満の是正に有効だとし，糖尿病の予防に有効だとしている．また，食物繊維の 1 日の平均摂取量が 20g を超えた時点から糖尿病の発症リスクに有意な低下傾向が見られている [?]. そのため，本研究における糖尿病の患者に対する制約条件として，エネルギー量，タンパク質摂取量は健常者と同じだが，1 日の炭水化物摂取量，脂質の摂取量，食物繊維の摂取量をそれぞれ 100g 以下，必要推定エネルギーの 15～25%，20g 以上とする．

次に，腎臓病を患っている人については，4.2 章で述べた通り腎臓病はたんぱく質制限，塩分制限，カリウム制限などの食事療法を行うことにより，腎機能障害の進行を抑え，慢性腎臓病の合併症を予防することができる．具体的な数値として日本腎臓学会によると 1 日のタンパク質の摂取量を標準体重当たり 0.6～0.7g とし，塩分の 1 日の摂取量は 3g 以上 6g 未満とし，カリウムの 1 日の摂取量が 1500mg 以下に制限することが推奨されている [?]. そのため，本研究における腎臓病の患者に対する制約条件として，エネルギー量，脂質，炭水化物の摂取量は健常者と同じだが，1 日のタンパク質と塩分と，カリウムの摂取量をそれぞれ標準体重当たり 0.6～0.7g，3g 以上 6g 未満，1500mg 以下とする．

脂質異常症を患っている人については，4.2 章で述べた通り脂質異常症は，コレステロール，食物繊維，脂質の摂取量を調整することにより脂質異常症の予防と改善に役に立つとされている [?]. 本研究における具体的な制約条件としては，エネルギー量，炭水化物，タンパク質の摂取量は健常者と同じだが，1 日のコレステロール，脂質，食物繊維の摂取量をそれぞれ 200mg 以下，総エネルギーの 15%未満，20g 以上にする．

高血圧と診断されている人は，4.2 章で述べた通り高血圧の要因として，塩分を過剰に摂取することによる血圧上昇が大きな要因となるため，塩分制限が必要となっている．日本高血圧学会による「高血圧治療ガイドライン 2019」によると，高血圧者の減塩目標を食塩 6 g/日未満としている．また，カリウム摂取量増加によって高血圧者にとって血圧低下効果を認めた．厚生労働省によると，カリウムの摂取量を 3510mg 以上摂取することが推奨

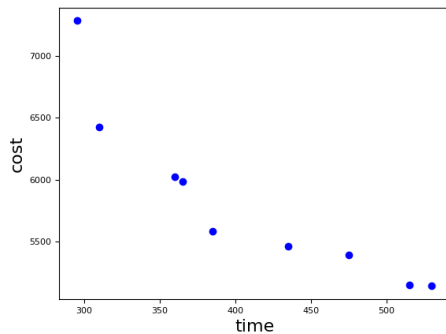


図 5.3: パレート解の出力

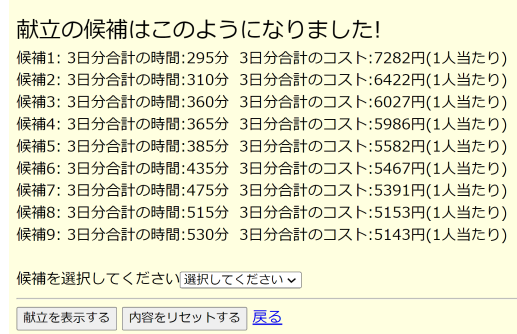


図 5.4: 対話型処理による解の選択

されている。さらに、1日の食物繊維の摂取量を 20g 以上にすること推奨されている [?]. 以上のことから、本研究における高血圧者に対する制約条件として、エネルギー量、脂質、炭水化物、タンパク質の摂取量は健常者と同じだが、1日のと塩分、カリウム、食物繊維の摂取量をそれぞれ 6g 未満、3510mg 以上、20g 以上とする。

次に、NSGA-II による最適化を行っている際の、実行画面について説明すると、n_gen は現在の世代数、n_level はこれまでの個体を評価した数、cv (min), cv (avg) はそれぞれ現在の母集団における最小の制約違反、現在の母集団における平均の制約違反、n_nds は多目的最適化問題の場合の非劣解の数、eps は過去数世代にわたるインジケータの変化、indicator はパフォーマンスインジケータを表す。

次に、NSGA-II による最適化処理が終わり、パレート最適解が出力された様子を図??に示す。縦軸は、指定した日数分の献立の合計調理時間を表しており、横軸は指定したに数の合計の食材コストを表している。

次に、パレート最適解から、対話型処理によって献立を選択する画面を図??に示す。図??は、図??のパレート解を数値として表示している画面である。ユーザは、画面に表示されている選択ボタンで、表示されている候補を選択する。「献立を表示する」というボタンをクリックすると選択した候補に対応した献立が出力される。

本研究においては以下の数値実験を行う。まず、健常者のユーザー像を想定し、システムを動かし、出力された献立が制約条件を満たしているかを考察する。次に、それぞれの生活習慣病を患っているユーザー像を想定し、同様の検証をする。さらに、複数人のユーザー像を想定し入力した全員の制約条件を満たしているかを考察する。

§ 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。また、以下は今回の数値実験に使用する 1 日に必要なエネルギーである、必要推定エネルギー量の計算式である。

< 基礎代謝基準値 >

$$\text{基礎代謝基準値} = \frac{\text{基準体重での基礎代謝量 (kcal/日)}}{\text{基準体重 (kg)}} \quad (5.1)$$

< 基礎代謝量 >

$$\text{基礎代謝量 (kcal/日)} = \text{基礎代謝基準値} \times \text{体重 (kg)} \quad (5.2)$$

< 必要推定エネルギー量 >

$$\text{必要推定エネルギー量 (kcal/日)} = \text{基礎代謝量} \times \text{身体活動レベル指数} \quad (5.3)$$

また、栄養素の制約条件に関しては、3大栄養素であるたんぱく質、脂質、炭水化物について設定した。具体的に設定した制約値としては、厚生労働省によると健常者におけるたんぱく質、脂質、炭水化物の必要摂取エネルギー量はそれぞれ必要推定エネルギーの40%以上、15%以上、13%以上とされているため[?], そのように設定した。以下は、最低でも1日に摂取すべき3大栄養素の量を計算する式である。

< 必要たんぱく質 >

$$\text{たんぱく質 (g/日)} = \frac{\text{必要推定エネルギー量 (kcal/日)} \times 0.13}{4(\text{kcal/g})} \quad (5.4)$$

< 必要脂質 >

$$\text{脂質 (g/日)} = \frac{\text{必要推定エネルギー量 (kcal/日)} \times 0.15}{9(\text{kcal/g})} \quad (5.5)$$

< 必要炭水化物 >

$$\text{炭水化物 (g/日)} = \frac{\text{必要推定エネルギー量 (kcal/日)} \times 0.4}{4(\text{kcal/g})} \quad (5.6)$$

今回はこれらの数式をもとに複数のユーザー像を想定して数値実験を行う。

1: 健常者に対する出力結果の考察

今回想定したユーザー像は以下のとおりである。まず、健常者のユーザーとして22歳の平均男性の平均値である、身長を172.3cm、体重を65.3kg、活動レベルを普通とした。

式(5.1)~(5.3)から1日に必要推定エネルギー量を計算すると、2695kcalとなるため、1日に摂取するカロリーは2695kcal プラスマイナス100kcalの範囲に入るように制約が設定されるようにした。また、(5.4)~(5.6)から、1日に最低でも摂取すべきたんぱく質は、84.33g以上の値であり、1日に最低でも摂取すべき脂質は、摂取エネルギーの2595kcalの15%以上であるから、1kcalに対して9g摂取できるとしたときに43.25g、炭水化物は摂取エネルギーの2595kcalの40%以上より、1kcalに対して4g摂取できるとしたときに259.5gである。各3大栄養素の上限値に対して、栄養素を摂取するための指標の1つであり、健康障害をもたらすリスクは医学的にないとみなされ、摂取量の上限を与える量と定義される耐容上限量は、それぞれ厚生労働省によって設定がされていないため、最低でも1日に摂取すべきである栄養素量を下限に設定している。今回設定した、摂取カロリーとたんぱく質、脂質、炭水化物の各3大栄養素、朝、昼、夜での各時間帯の調理時間合計、主菜と副菜による制約条件と、実験にて設定した値について、表??に示す。

梅干しとわかめのスープ



調理時間	摂取カロリー	食材コスト
10分	38kcal	148円

栄養名	栄養素量	食材名	食材量	作り方 (1)わかめは塩を洗い、水で戻した後、サッと湯通しし、冷水に取ってひと口大に切ります。(2)鶏ささ身は細切りにし、塩、酒をふります。(3)チキンスープを温め、(2)を入れ、(A)で調味し、(1)、梅干しを加えて水溶き片栗粉でとろみを付けます。
たんぱく質	5.5g	梅干し	4個	
脂質	0.2g			
炭水化物	3.7g	塩わかめ	60g	
糖質	2.7g			
食塩相当量	5.5g	鶏ささ身	1本	
食物繊維	1.0g			
ビタミンA	5.0μg			
ビタミンB1	0.05mg			

図 5.5: 献立作成システムによる出力結果

表 5.1: 設定した制約条件

制約条件	設定した値
摂取カロリー(kcal)	2595~2795
たんぱく質(g)	84.33~
脂質(g)	43.25~
炭水化物(g)	259.5~
調理時間合計[朝](分)	15
調理時間合計[昼](分)	45
調理時間合計[夕](分)	60
主菜	1
副菜	2

本研究で提案する自動献立作成システムにおける、献立を作成した出力結果について図 5.5 献立結果 に示す。最適化処理によって出力された献立は、flask により作成した Web サーバー上で、HTML ファイルによって表示される。

また、以上の制約条件を入力したときの献立の出力結果およびパラメータはそれぞれ表 5.1, 表 5.2 に示す。

次に、自動献立作成システムによって実際に出力した献立が設定した制約条件を満たしているか比較を行う。最初に、摂取エネルギーの比較を行うと、出力された献立の 1 日に摂取エネルギーは 2621kcal であり、これは制約条件である、2595kcal 以上 2795kcal 以下を満たしている。

次に、3 大栄養素の制約条件の比較を行う。出力された献立から得られる 1 日のたんぱく質、脂質、炭水化物はそれぞれ 92.3g, 72.1g, 276.3g であり、これは 1 日に摂取すべきであるたんぱく質、脂質、炭水化物量である 84.33g, 43.25g, 259.5g を満たしていることがわかる。

また、各時間帯別の、1 日の献立の調理時間合計についての制約条件について比較を行う。出力された献立の朝、昼、夕の調理時間の合計はそれぞれ 15 分、40 分、45 分となった。これは制約した朝、昼、夕の調理時間の合計の 15 分以下、45 分以下、60 分以下を満たしている。

2: 生活習慣病を患っている人に対する出力結果の考察

まず、糖尿病を患っている人の数値実験に関して説明する。厚生労働省によると、BMI が 23 以上になると糖尿病のリスクがなりやすいと報告されている。また 2009 年の糖尿病の平均年齢が 71 歳であることから、年齢を 71 歳、身長は 71 歳の男性の平均身長である 163.1cm、体重は BMI が 23 で身長が 163.1cm の場合の 61.18kg、身体活動レベルを低いとする。

また、糖尿病を患っている人に対する必要推定エネルギーは健常者と同じ式を使う。以上の身体情報とし、式 (5.1)~(5.3) から 1 日に必要推定エネルギー量を計算すると 1919kcal となるから 1 日に摂取するカロリーは 1919kcal プラスマイナス 100kcal の範囲に入るように制約が設定されるようにした。また、5.1 章よりたんぱく質の式は (5.4) を使い、炭水化物の摂取量を 100g 以下、脂質の摂取量を必要推定エネルギーの 15~25%、食物繊維の摂取量を 20g 以上とする。その結果、タンパク質の摂取量は 62.36g、脂質の摂取量は 31.98g~53.3g となった。

表 5.2: 献立の出力結果

	出力されたレシピ
朝	ひまわりご飯
	なんちゃってピザ
昼	イワシのガーリックトマトソース
	絹さやの卵とじ
	ブリのもぐり飯
夕	ほうれん草とえのきのお浸し
	ホットブレッドサラダ

表 5.3: 献立のパラメータ

献立のパラメータ	
摂取カロリー	2621
たんぱく質	92.3
脂質	72.1
炭水化物	276.3
調理時間合計[朝](分)	15
調理時間合計[昼](分)	40
調理時間合計[夕](分)	45

次に、腎臓病を患っている人の数値実験に関して説明する．糖尿病と同様，厚生労働省によると，BMI が 23 以上になると腎臓病のリスクがなりやすいと報告されている．また 2019 年の腎臓病の平均年齢が 70 歳であることから，年齢を 70 歳，身長は 71 歳の男性の平均身長である 163.1cm，体重は BMI が 23 で身長が 163.1cm の場合の 61.18kg，身体活動レベルを低いとする．

腎臓病を患っている人に対する必要推定エネルギーは健常者と同じ式を使う．以上の身体情報とし，式 (5.1)～(5.3) から 1 日に必要推定エネルギー量を計算すると 1919kcal となる．よって 1 日に摂取するカロリーは 1919kcal プラスマイナス 100kcal の範囲に入るように制約が設定されるようにした．脂質，炭水化物の制約条件は式 (5.5)，(5.6) から 30.31g，181.9g となった．タンパク質の制約条件は，腎臓病の場合標準体重当たり 0.6～0.7g となっているため 61.18kg の場合 36.7～42.82g となる．また，塩分の摂取量の制約は 3～6g，カリウムの摂取量は 1500mg 未満である．

続いて，脂質異常症を患っている人の数値実験に関して説明する．厚生労働省によると，BMI が 35 以上になると脂質異常症を発症するリスクが高まるとされている．また 2019 年の脂質異常症の平均年齢が 45 歳であることから，年齢を 45 歳，身長は 45 歳の男性の平均身長である 171.5cm，体重は BMI が 35 で身長が 171.5cm の場合の 102.94kg，身体活動レベルを低いとする．

また，厚生労働省総エネルギーを減らすことによる脂質異常症の抑制のを示す直接的なエビデンスはないとされている．よって必要推定エネルギーは式 (5.1)～(5.3) を使用して求める．以上の身体情報とし，式 (5.1)～(5.3) から 1 日に必要推定エネルギー量を計算すると 2838kcal となる．よって 1 日に摂取するカロリーは 2838kcal プラスマイナス 100kcal の範囲に入るように制約が設定されるようにした．また，たんぱく質，炭水化物の摂取量は式 (5.4)，(5.6) を使用してそれぞれと 92.23g，283.8g になった．また，コレステロールは 200mg 未満，脂質は総エネルギーの 15%未満であるから 47.3g 未満，食物繊維の摂取量は 20g 以上である．

最後に，高血圧を患っている人の数値実験に関して説明する．厚生労働省によると，肥満の人が高血圧になりやすいとされている．そのため，肥満の平均値である BMI が 30 の人を想定する．また 2019 年の高血圧の平均年齢が 37 歳であることから，年齢を 37 歳，身長は 37 歳の男性の平均身長である 171.5cm，体重は BMI が 30 で身長が 171.5cm の場合の 88.24kg，身体活動レベルを低いとする．

表 5.4: 各生活習慣病に対する制約条件

	摂取カロリー(kcal)	たんぱく質(g)	脂質(g)	炭水化物(g)	塩分(g)	食物繊維(g)	カリウム(mg)	コレステロール(mg)
糖尿病	1819~2019	62.36~	31.98~53.3	0~100.0	設定なし	20.0~	設定なし	設定なし
腎臓病	1819~2019	36.7~42.82	30.31~	181.9~	3.0~6.0	設定なし	~1500	設定なし
脂質異常症	2738~2938	92.23~	0~47.3	283.8~	設定なし	20.0~	設定なし	0~200.0
高血圧	2524~2724	82.03~	42.06~	252.4~	0~6.0	20.0~	3510~	設定なし

表 5.5: 各生活習慣病患者のパラメータ

	摂取カロリー(kcal)	たんぱく質(g)	脂質(g)	炭水化物(g)	塩分(g)	食物繊維(g)	カリウム(mg)	コレステロール(mg)
糖尿病	1926	80.1	42.3	96.2	8.1	22.5	3210	141
腎臓病	1830	35.7	32.6	194.5	3.9	18.3	1465	72
脂質異常症	2925	93.5	40.5	294.6	8.2	26.7	2988	156
高血圧	2655	92.3	43.5	265.1	4.5	26.4	3612	204

また、高血圧を患っている人に対する必要推定エネルギーは健常者と同じ式を使う。以上の身体情報とし、式 (5.1)~(5.3) から 1 日に必要推定エネルギー量を計算すると 2624kcal となるから 1 日に摂取するカロリーは 2624kcal プラスマイナス 100kcal の範囲に入るように制約が設定されるようにした。また、たんぱく質、脂質、炭水化物の摂取量は式 (5.4), (5.5), (5.6) を使用してそれぞれ 82.03g, 42.06g, 252.4g となった。また、塩分の摂取量を 6g 未満、カリウムの摂取量を 3510mg 以上、食物繊維の摂取量を 20g 以上とする。

以上の制約条件から数値実験を行った時の結果を以下の表??に示す。また、その時の制約条件を以下の表??に示す。

次に、自動献立作成システムによって実際に出力したそれぞれのパラメータが設定した制約条件を満たしているか比較を行う。最初に、摂取エネルギーの比較を行うと、出力された献立の 1 日に摂取エネルギーは糖尿病、腎臓病、脂質異常症、高血圧の順に行くと 1926kcal, 1830 kcal, 2925kcal, 2655kcal であり、これは制約条件の 1 日に摂取すべきカロリーである 1819kcal 以上 2019kcal 以下, 1819kcal 以上 2019kcal 以下, 2738kcal 以上 2938kcal 以下, 2524kcal 以上 2724kcal 以下の範囲内に入っているため制約条件を満たしている。

次に、それぞれの栄養素の制約条件の比較を行う。まず糖尿病患者の比較を行う。出力された献立から得られる 1 日のたんぱく質、脂質、炭水化物、塩分、食物繊維、カリウム、コレステロールはそれぞれで 80.1g, 42.3g, 96.2g, 8.1g, 22.5g, 3210mg, 141mg であり、これは制約条件で設定した 1 日に摂取すべきたんぱく質 62.36g 以上, 脂質 31.98g 以上 53.3g 以下, 炭水化物 100g 以下, 食物繊維 20g 以上を満たしていることがわかる。

次に腎臓病患者の比較を行う。出力された献立から得られる 1 日のたんぱく質、脂質、炭水化物、塩分、食物繊維、カリウム、コレステロールはそれぞれで 35.7g, 32.6g, 194.5g, 3.9g, 18.3g, 1465mg, 72mg であり、これは 1 日に摂取すべきたんぱく質 36.7g 以上 42.82g 以下, 脂質 30.31g 以上, 炭水化物 181.9g 以上, 塩分 3g 以上 6g 未満, カリウム 1500mg 未満を満たしていることがわかる。

続いて、脂質異常症患者の比較を行う。出力された献立から得られる 1 日のたんぱく質、脂質、炭水化物、塩分、食物繊維、カリウム、コレステロールはそれぞれで 93.5g, 40.5g,

表 5.6: 大人数における制約条件

	摂取カロリー(kcal)	たんぱく質(g)	脂質(g)	炭水化物(g)
モデル1	2509~2709	81.54~	41.82~	250.9~
モデル2	1876~2076	60.97~	31.27~	187.6~
モデル3	2953~3153	95.97~	49.22~	295.3~
モデル4	1575~1775	51.19~	26.25~	157.5~

表 5.7: 大人数料理のパラメータ

	摂取カロリー(kcal)	たんぱく質(g)	脂質(g)	炭水化物(g)
モデル1	2620	85.15	43.67	262.00
モデル2	2017	65.57	33.63	201.74
モデル3	3039	98.77	50.66	303.92
モデル4	1755	57.05	29.26	175.54

294.6g, 8.2g, 26.7g, 2988mg, 154mg であり, これは1日に摂取すべきたんぱく質 92.23g 以上, 脂質 47.3g 未満, 炭水化物 283.8g 以上, 食物繊維 20g 以上, コレステロール 200mg 未満を満たしていることがわかる.

最後に高血圧患者の比較を行う. 出力された献立から得られる1日のたんぱく質, 脂質, 炭水化物, 塩分, 食物繊維, カリウム, コレステロールはそれぞれで 92.3g, 43.5g, 265.1g, 4.5g, 26.4g, 3612mg, 204mg であり, これは1日に摂取すべきたんぱく質 82.03g 以上, 脂質 42.06g 以上, 炭水化物 252.4g 以上, 塩分 6g 未満, 食物繊維 20g 以上, カリウム 3510mg 以上を満たしていることがわかる.

3: 大人数を想定した場合に対する出力結果の考察

次に, 大人数を想定したときの考察する. 今回実験した大人数として4人家族世帯を想定する. 1人目のモデルとして年齢は52歳, 身長と体重は52歳の平均身長, 平均体重である 170.8cm, 70.4kg とし, 性別は男, 身体活動レベルは普通とした. 2人目のモデルとして年齢は48歳, 身長と体重は48歳の平均身長, 平均体重である 158.3cm, 55.2kg とし, 性別は女, 身体活動レベルは普通とした. 3人目のモデルとして年齢は22歳, 身長と体重は22歳の平均身長, 平均体重である 172.6cm, 64.0kg, 性別は男, 身体活動レベルは高いとした. 4人目のモデルとして年齢は17歳, 身長と体重は17歳の平均身長, 平均体重である 154.8cm, 47.2kg とし, 性別は女, 身体活動レベルは低いとした.

上記の実験と同様に式 (5.1)~(5.6) を用いてそれぞれのモデルの必要エネルギー, 必要たんぱく質, 必要脂質, 必要炭水化物をまとめたものを表??に示す. また, 出力するにあたって出力日数を1日, 朝の調理時間の合計を20分, 昼の調理時間の合計を45分, 夜の調理時間の合計を60分とした. また, 本研究で提案する自動献立作成システムにおけるパラメータを表??に示す.

次に, 自動献立作成システムによって実際に出力した結果が設定した制約条件を満たしているか比較を行う. 最初に, 摂取エネルギーの比較を行うと, モデル1, モデル2, モデル3, モデル4の出力された献立の1日に摂取エネルギーはそれぞれ 2620kcal, 2017kcal, 3039kcal, 1755kcal であり, これは制約条件の1日に摂取すべきカロリーである 2509kcal 以

表 5.8: 1 日ごとの献立作成の処理時間

	1日	2日	3日	4日	5日	6日	7日
1回目(sec)	930.58	909.66	899.54	899.89	866.91	935.38	946.33
2回目(sec)	929.34	938.16	1051.69	1452.54	933.714	959.39	977.4
3回目(sec)	980.27	978.95	919.22	924.49	913.84	890.18	879.92
平均(sec)	946.73	942.257	956.817	1092.31	904.821	928.317	934.55

上 2709kcal 以下, 1876kcal 以上 2076kcal 以下, 2953kcal 以上 3153kcal 以下, 1575kcal 以上 1775kcal 以下の範囲内に入っているため制約条件を満たしている。

次に, それぞれの栄養素の制約条件の比較を行う。まずモデル 1 の比較を行う。出力された献立から得られる 1 日のたんぱく質, 脂質, 炭水化物はそれぞれで 85.15g, 43.67g, 262g であり, これは制約条件で設定した 1 日に摂取すべきたんぱく質 81.54g 以上, 脂質 41.82g 以上, 炭水化物 250.9g 以上を満たしていることがわかる。

次にモデル 2 の比較を行う。出力された献立から得られる 1 日のたんぱく質, 脂質, 炭水化物はそれぞれで 65.57g, 33.63g, 201.74g であり, これは制約条件で設定した 1 日に摂取すべきたんぱく質 60.97g 以上, 脂質 41.82g 以上, 炭水化物 187.6g 以上を満たしていることがわかる。

続いてモデル 3 の比較を行う。出力された献立から得られる 1 日のたんぱく質, 脂質, 炭水化物はそれぞれで 98.77g, 50.66g, 303.92g であり, これは制約条件で設定した 1 日に摂取すべきたんぱく質 95.97g 以上, 脂質 49.22g 以上, 炭水化物 295.3g 以上を満たしていることがわかる。

最後にモデル 4 の比較を行う。出力された献立から得られる 1 日のたんぱく質, 脂質, 炭水化物はそれぞれで 80.1g, 42.3g, 96.2g であり, これは制約条件で設定した 1 日に摂取すべきたんぱく質 57.05g 以上, 脂質 29.26g 以上, 炭水化物 175.54g 以上を満たしていることがわかる。

4: 出力する日数を変更した場合に対する出力にかかった時間の比較

自動献立作成にあたって, 出力する日数を変更した場合に対する出力にかかった時間の結果を表??に示す。表??より, 1 日分の献立を出力したときの出力時間の平均は 946.73 秒, 2 日分の献立を出力したときの出力時間の平均は 942.257 秒, 3 日分の献立を出力したときの出力時間の平均は 956.817 秒, 4 日分の献立を出力したときの出力時間の平均は 1092.31 秒, 5 日分の献立を出力したときの出力時間の平均は 904.821 秒, 6 日分の献立を出力したときの出力時間の平均は 928.317 秒, 7 日分の献立を出力したときの出力時間の平均は 934.55 秒であることが分かった。

この結果から, 出力する日数を変更しても実行時間はあまり変化しないことがわかる。

5: 出力する人数を変更した場合に対する出力にかかった時間の比較

自動献立作成にあたって, 出力する人数を変更した場合に対する出力にかかった時間の結果を表??に示す。ここで用いたモデルは大人数の時に想定したモデルと同じものを使用する。

1 人分の献立を出力したときの出力時間の平均は 946.73 秒, 2 人分の献立を出力したときの出力時間の平均は 950.23 秒, 3 人分の献立を出力したときの出力時間の平均は 919.59 秒, 4 人分の献立を出力したときの出力時間の平均は 976.007 秒あることが分かった。この結果より, 出力する人数を変更は処理時間に大きく影響しないことがわかった。

表 5.9: 1 日ごとの献立作成の処理時間

	1人	2人	3人	4人
1回目(sec)	930.58	959.47	913.3	933.142
2回目(sec)	929.34	977.4	899.14	1031.59
3回目(sec)	980.27	913.82	946.33	963.29
平均(sec)	946.73	950.23	919.59	976.01

おわりに

急激な生活様式の欧米化に伴い、ジャンクフードといった、余分にエネルギーを摂取してしまうような食生活が大きく広まったことから、現在、生活習慣病を患う人々が増加している。生活習慣病を予防する一つの方法として、栄養バランスのとれた食事をとることが推奨されている。しかし、栄養バランスの取れた献立作成には、その人の身体情報、疾患情報などによってメニューや料理の分量を調整しなければならず、献立作成業務の負荷は高いことがわかる。

これらの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価格を活用し、制約条件を考慮できる多目的遺伝的アルゴリズムによって自動的に献立を作成をするシステムを考案した。

本研究で用いるレシピデータとして、3つのレシピサイトからスクレイピングを行うことによってレシピデータベースに多様性を持たせることができた。また、この献立作成システムは健常者だけではなく、生活習慣病を患っている人やアレルギーを患っている人でも利用できるようにした。さらに、プログラム実行に必要なすべてのプログラムをサーバーに置き、実行に必要なURLを用意することによって、ユーザはそのURLをクリックするだけでプログラムを実行できるようにした。

また、プログラムの実行にはレシピデータなどの大量のデータが必要なため、プログラムの環境を整えるための手間が大変になってしまう問題があった。そのためプログラムをサーバー上に置くことでプログラム実行の環境を整える手間を省くことができた。

本研究で提案した制限食と大人数料理に対応した自動献立作成システムを実際に動作させた実験結果として、多目的最適化によって作成された献立は調理時間、料理コストを最小化しながら、設定した制約条件を満たしながら出力することができた。

本研究の課題として、摂取栄養素や摂取カロリーの上限、下限の設定などの制約条件を、ユーザ自身で決められるようにすることや、並列分散処理などを施すことにより、最適化プログラムの実行処理時間を向上し、よりユーザに快適に利用できるようにプログラムを改良する必要がある。また、ユーザが好みの料理を入力することによって、出力する料理がユーザの好みに近いもの出るようにすることや、ユーザが現在持っている食材を入力することによって、その食材を含む料理が出力されるようにする必要があると考えられる。

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電子・情報工学科情報基盤工学講座の António Oliveira Nzinga René 講師，奥原浩之教授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．

2023 年 2 月

水上和秀

参考文献

- [1] 公益社団法人 千葉県栄養士会, “生活習慣病の予防、食生活 生活習慣病の予防と食事”, <https://www.eiyou-chiba.or.jp/commons/shokuji-kou/preventive/seikatusyukan/>, 閲覧日 2023.1.7.
- [2] 国立研究開発法人 国立循環器病研究センター, “食事療法について”, <https://www.ncvc.go.jp/hospital/pub/knowledge/diet/diet02/>, 閲覧日 2023.1.7
- [3] 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 2022.12.28.
- [4] 貝沼やす子, 江間章子, “日常の献立作りの実態に関する調査研究 (第 1 報)”, 日本調理学会誌, Vol.30, No. 4, pp. 364-371, 1997.
- [5] 株式会社おいしい健康, “おいしい健康”, <https://oishi-kenko.com/>, 閲覧日 2022.10.16.
- [6] 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 2022.10.11.
- [7] J. W. Ratcliff and D. Metzener, “Pattern Matching: The Gestalt Approach”, *Dr. Dobbs' s Journal*, p.46, 1988.
- [8] 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.
- [9] 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.
- [10] 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 2023.1.6.
- [11] J. H. Holland, “Adaptation in Natural and Artificial Systems”, 1975.
- [12] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, “A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II”, *IEEE Tran. on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197, 2002.
- [13] D. E. Goldberg, “Genetic algorithms in search, optimization and machine learning ”, *Addison-Wesly*, 1989.
- [14] メディカル・ケア・サービス株式会社, “制限食にはどんな種類があるの?”, 健達ネット, <https://www.mcsg.co.jp/kentatsu/health-care/12106>, 閲覧日 2023.1.6.

- [15] ときわ会栄養指導課, “減塩について”, 栄養指導,
<http://www.tokiwa.or.jp/nutrition/diet/low-salt.html>, 閲覧日 2023.01.15
- [16] 全国健康保険協会, “ちょっとした工夫で脂質をコントロール”,
<https://www.kyoukaikenpo.or.jp/g4/cat450/sb4501/p004/>, 閲覧日 2023.01.15
- [17] 厚生労働省, “日本人の食事摂取基準 (2020 年度版)”,
<https://www.mhlw.go.jp/content/10904750/000586559.pdf>, 閲覧日 2023.01.15
- [18] 東京医科大学病院, “カリウムは調理のくふうで減らせます”, 内臓内科,
<https://articles.oishi-kenko.com/syokujinokihon/dialysis/05/>, 閲覧日 2023.01.15
- [19] 厚生労働省, “糖尿病”, <https://www.mhlw.go.jp/content/10904750/000586592.pdf>,
閲覧日 2023.01.17
- [20] 厚生労働省, “慢性腎臓病”, <https://www.mhlw.go.jp/content/10904750/000586595.pdf>,
閲覧日 2023.01.17
- [21] 腎臓内科, “慢性腎臓病の食事療法”, 東京女子医科大学,
<https://www.twmu.ac.jp/NEP/shokujiryohou.html>, 閲覧日 2023.01.17
- [22] 厚生労働省, “脂質異常症”, <https://www.mhlw.go.jp/content/10904750/000586590.pdf>,
閲覧日 2023.01.17
- [23] 厚生労働省, “高血圧”, <https://www.mhlw.go.jp/content/10904750/000586583.pdf>,
閲覧日 2023.01.17
- [24] 厚生労働省, “食べ物アレルギー”, アレルギーポータル,
<https://allergyportal.jp/knowledge/food/>, 閲覧日 2023.01.17
- [25] J. Blank, “pymoo: Multi-objective Optimization in Python ”,
<https://www.egr.msu.edu/kdeb/papers/c2020001.pdf>, 閲覧日 2023.1.22.
- [26] 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.
- [27] 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 2022.12.26.
- [28] 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
2023.1.22.