卒業論文

ウェルビーイングに有益なUesr eXperienceを考慮できる自立献立作成システム

Development of a Browser-based Automatic Menu Creation System that Supports 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 教授
提出年月: 令和7年 (2025年) 2月

目 次

図一覧

表一覧

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	$d : 1, 2, 3, \cdots, S$
日の番号	$k : 1, 2, 3, \cdots, 3D$
栄養素の番号	$l : 1, 2, 3, \cdots, N$
材料の番号	$m : 1, 2, 3, \cdots, Q$
レシピの番号	$i : 1, 2, 3, \cdots, R$
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

第1章

はじめに

§ 1.1　本研究の背景

戦後日本では，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．生活習慣病とは,「食習慣，運動習慣，休養， 喫煙，飲酒，ストレスなどの生活習慣を原因として発症する疾患の総称」のことであり，日本人の死因の上位を占める．生活習慣病は，脳血管疾患や心疾患などの深刻な疾患に深く関与している．

生活習慣病による疾患には，自覚症状がほとんどなく，気づかないうちに症状が進行し，血管や心臓，脳にダメージが蓄積していく．その結果突然として命に関わる疾患を引き起こす可能性がある．命に係わる疾患を引き起こしてからでは手遅れであるため，症状が全くないことから安心するのではなく，栄養バランスのとれた食事をとることや適度な運動，過度な飲酒や喫煙を抑えることや生活リズムの見直しなどの生活習慣病の予防に，日々努力する必要がある．

本研究では，上記の複数ある生活習慣病の予防方法のうち，バランスの取れた栄養を摂取できる健康的な食事をとることについて着目する．生活習慣病を予防，および改善するための具体的な食生活の要素として，1日3食を，朝昼晩の時間帯で規則正しく食べること，個人の身体状況にあった栄養素を十分に摂れること，食事に含まれる塩分や糖分を控えめにすること，ビタミン類や食物繊維，カルシウムを十分にとることなどが挙げられる [?].

さらに，血圧が高めであったり，肥満気味である，あるいは健康診断で生活習慣病予備軍と診断された場合でも食事療法で状態を改善することが期待され，生活習慣病と診断された場合でもさらなる悪化を防ぐことができる [?]．このように，栄養バランスのとれた食事をとるということは，健康的な生活を送るために必要不可欠な要素の1つであることがわかる．

また，近年，学校給食などの現場では，日々の食事は学生にとって整えるべき生活リズムのひとつであり，食事の時間は日々の楽しみの1つでもあることから，学生にとって摂取すべきである栄養のことを考え，様々な食材の組み合わせからなる献立を作成している．

その献立作成業務を担当している栄養士は，食事にかかる金額や栄養素や考慮すべきである献立作成条件を，毎日，何度も繰り返し見直しながら改善していく必要があり，また，献立を食べるすべての学生が満足するような献立を作成する必要があるため，献立作成業務は大変な作業であり，栄養士に対する負荷はかなり高いことがわかる [?]．それにともなって，これらの負担の高い業務を，AIや数理計画化によって自動化するシステムがある．

§ 1.2　本研究の目的

栄養の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，複雑な計算や必要な栄養価を考慮する必要があり，献立を考える時間が無かったり，自身で献立を考えることが面倒だと考える人は少なくない [?]．また，短い調理時間でお手軽にかつ食材コストを抑えられ，さらに自身の好みに合った献立を作成することは，時間がない人や空き時間を作りたい人，できるだけ献立作成の費用を節約をして料理を作成したい人にとっては理想的であるといえる．

また，献立作成業務を行っている学校給食や病院食の現場での栄養士は，煩雑な栄養計算や食材にかかる費用の計算などの様々な条件を考慮した上で何度も見直しながら献立を作成している．そのため，献立作成を自動化することによって，献立作成業務の負荷を軽減することが求められている．

他にも病院の現場では，毎日の食事は患者にとって生活リズムの中心であり，日々の楽しみの 1 つでもある．そのため，食の感動を大切にし，病院では医食同源の精神を基本に患者の好みに合ったメニューを提供することが求められている．食に関する専門性を高めるために，日々食に対して研究や開発，研修を行っていることから，献立作成する業務を行っている人にとって負荷の高い業務を行っていることが分かる．

そこで，本研究では，献立作成を組み合わせ多目的最適化問題として考えることにより，栄養のバランスがとれていて，調理時間，食材コストが少なくなるように，なおかつ，使用者の身体情報にあった栄養量や摂取カロリーについての要望を満たされるように，さらには使用者の好みや病態に最も適した献立を，自動的に作成するシステムを提案する．

また，最適化に用いる料理のデータは，Web 上に存在する複数のレシピサイトから Python のプログラムによるスクレイピングによって蓄積する．具体的な料理データの中身として，必要食材や摂取する栄養量，カロリーなどが挙げられる．また，料理のデータに対するコスト計算を行うため，食品価格の推移を調査している Web サイトから食材と販売単位あたりの価格をスクレイピングによってデータベースに蓄積する．上記の方法によって蓄積したデータを入力とした，組み合わせ多目的最適化問題を解く手法として，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 (Non-dominated Sorting Genetic Algorithm: NSGA-Ⅱ) を採用する．

本研究では，NSGA-Ⅱ や遺伝的アルゴリズムをはじめとした，様々な最適化手法について幅広く対応している，pymoo という Python ライブラリを利用して，最適化プログラムを記述する．さらに，最適化され，出力された料理の中から，ユーザ自身の希望する料理が選択できるように，分かりやすく感覚的にシステムを利用できるような対話型処理を用いる．具体的には，料理の合計調理時間と合計コストの候補を表示し，ユーザに選択してもらうものである．

また，本プログラムは大量のレシピデータやプログラムを使っているため，使用者が利用するたびにデータをダウンロードしていてはプログラムの実行する環境を整えるのに時間がかかってしまう．そのため本研究ではサーバー上にプログラムを置き，利用者にはブラウザでサーバーにアクセスするだけで利用できるようなシステムを実現する．

最後に，本研究によって提案された制限食や大人数料理を考慮した，自動献立作成システムをブラウザ上で動作させ，数値実験を行い，実験結果に基づき考察をする．

§ 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，並列分散処理による実行速度向上の意義について示す．目的は多目的遺伝的アルゴリズムによる最適な自動献立作成の並列分散処理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．また，並列分散処理に関する用語と手法についてまとめる．

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，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．

第2章

2　UXを考慮した献立作成支援

§ 2.1　ウェルビーイングと献立作成

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??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??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プレイピングという手法で取得する．

スクレイピング

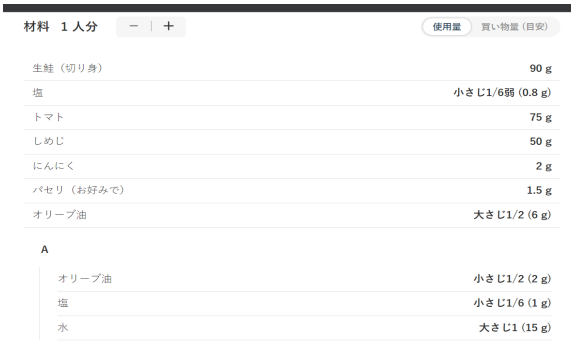
スクレイピングとは，データを収集し，かつ目的に合わせて加工することである．特に，Web上から必要なデータを取得することを，Webスクレイピングと呼ばれている．Webスクレイピングの流れについて図??に示す．様々なツールやプログラミングでスクレイピングを自動化することで，Webデータの収集にかかる手間や時間は大幅に削減が可能である．

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．クロージングとは，Web状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．クロージ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，クロージ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，

(a) 料理名とイメージ



(b) 得られる栄養素量と必要食材料

図 2.1: おいしい健康のレシピページ例 [?]

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．

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．

図 2.2: 食材価格動向調査サイトの例 [?]

図 2.3: Web スクレイピングの流れ

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する部分を抜きとり，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，

$$D_{ro}(S_1, S_2) = \frac{2K_m}{|S_1| + |S_2|} \tag{2.1}$$

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．

§ 2.2　献立作成での多目的最適化

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??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

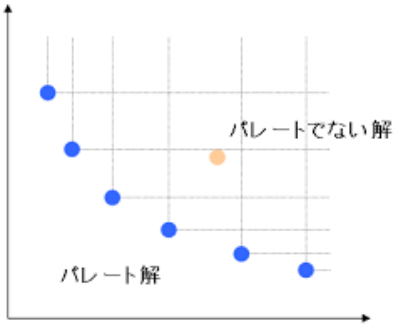


図 2.4: パレート最適解のイメージ

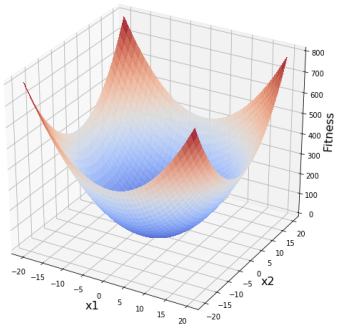


図 2.5: 解探索のイメージ（粒子群最適化）

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，$x = (x_1, x_2, \cdots, x_N)$，$f_i(x)$，$i = 1, 2, \cdots, n$ を目的関数として，$g_k(x), k = 1, 2, \cdots, m$ を制約条件式とすると，

$$\underset{x}{\text{minimize}} \qquad \{f_1(x), f_2(x), \cdots, f_n(x)\} \tag{2.2}$$

$$\text{subject to} \qquad g_k(x) \leq 0 \qquad k = 1, 2, \cdots, m \tag{2.3}$$

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 次元ユーグ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 SwarmOptimization: MOPSO) とは，粒子群最適化の手法 (Particle Swarm Optimization： PSO) 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??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，Zhang，Li らによって 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，Achevement 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　献立作成における UX

現在の情報社会では，経済や社会の問題を解決したり，業務を支援したり付加価値向上行うために用いる，ビッグデータをとり扱うために様々なアルゴリズムやアプリケーションによって，システム，組織などに関するデータは日々収集され，大量のデータが生成されている．しかし問題となってくるのは，この莫大な量のデータを，効率的かつ高速に処理する手段である．この問題を解決する方法として並列分散処理という技術がある．並列分散処理とは，複数台のコンピュータをリンクさせて，複数の CPU や，メモリを同時に使用することで一つの計算処理を行うことである．これにより，処理性能や計算速度を向上させることができる．並列分散処理を行っているときのイメージを図??に示す．並列分散処理を行うメリットとしては，1 台のコンピューターで実行するよりも短い時間で解を導くことができることや，1 台では実装が難しい大規模の処理を実現できることが挙げられる [?].

オーバーヘッド

並列分散処理における注意点としてオーバーヘッドが挙げられる．オーバーヘッドとは，コンピュータシステムで，何らかの処理を実行するときにかかる時間的，または空間的な費用やコンピュータにおける負荷のことを指す．並列分散処理におけるオーバーヘッドが，並列化によって得ることのできる性能の向上の恩恵を上回ってしまうことがありえる．

例えば，複数のプロセス，スレッド上で並列分散処理を行おうとする際に，複数のプロセスと各スレッドの起動，終了の処理，並列化するためのデータの分割と結果の統合の処理などにかかる時間の合計などが，並列化によって削減された時間合計を超えてしまう場合，並列化したことがかえって処理性能の低下を引き起こすことに繋がってしまうことがある．また，物理的なプロセッサコアの数以上のプロセス，スレッドを起動させて並列分散処理を行っても，現在実行している処理の流れを一旦停止し，別の処理に切り替え，実行を再開するときに発生するオーバーヘッドがかさんでしまうため，これもかえって処理性能を低下させてしまう要因となることがある．

また，とあるタスクをどう分散させ，どう実行するか，複数のコンピュータによる処理結果はどう１つの結果にまとめたらよいか，などの問題があり，導入は容易ではなかったが，Apache Hadoop や Apache Spark，Dask などの並列分散ソフトウェアが台頭したことによって並列分散処理の利用に対する敷居は低くなりつつある．現在，並列分散処理を行いたい場合に使われることの多いソフトウェアである Apache Hadoop，Apache Spark，Dask について説明する．

Apache Hadoop

Apache Hadoop とは，大規模なデータを効率的に管理し，分散処理するために用いられるソフトウェアの 1 つである．Hadoop は，オープンソースソフトウェアとして開発元のアパッチソフトウェア財団 (Apache Software Foundation: ASF) が公開しており，Java 言語での開発がされている [?].

主な Hadoop の機能として，複数のストレージ装置にペタバイト級の大量なデータを分散して保存すること，データを複数のコンピュータに分散して並列に解析処理を行ったりすることが Hadoop によって提供される．大規模なデータを解析することを目的とする開発

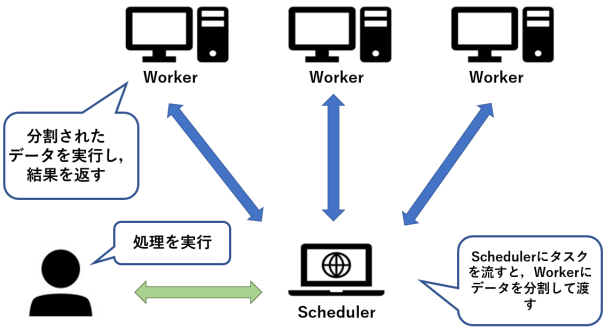


図 2.6: 並列分散処理のイメージ

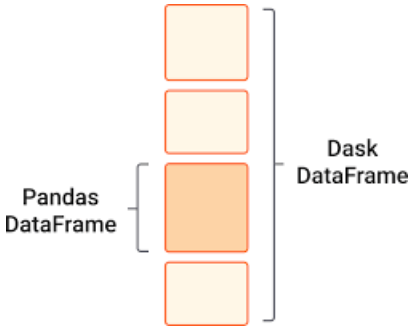


図 2.7: Dask のデータフレームのイメージ

者は，並列分散したい処理を記述するだけで，データをどうやって振り分けするかや，それぞれのコンピュータによって処理した結果の結合などを，自動的に Hadoop が行う．

数々の要素によって Hadoop は構成されており，それぞれの要素で共通の基盤となる Hadoop Common や，分散処理の管理を行う Hadoop MapReduce，分散ファイルシステムである HDFS（Hadoop Distributed File System: HDFS），複数のコンピュータで構成されるクラスタ上のリソースを管理したり，処理を最適な順番で行うスケジューリングなどを管理する YARN などの要素が組み込まれている．

Hadoop MapReduce は，Google 社によって発案された [?]，分散処理のメカニズムである，MapReduce を搭載したソフトウェアのことである．MapReduce は，与えられたデータを分割してそれぞれのノードに振り分ける Map プロセスと，それぞれのノードの計算処理による結果を収集して，最終的な処理結果を 1 つにまとめる，Reduce プロセスによって構成されている．2 つの処理は，両方とも並列化が可能であり，扱うデータが増加しても，計算処理を行うノード台数を増やすことによって，処理時間の短縮を図ることができる．

HDFS は，分散型のファイルシステムであり，分散処理のための大量なデータを，効率的に管理する機能を持っている．具体的には，一定の容量でブロック単位に大量のデータを分割し，多数のノードに複製することで保管するという仕組みになっている．ノードが故障したりなどして，一部のデータが消去されてしまっても，複数あるノードに複製されたデータから自動的に復元されるため，1 つの機器で保管や管理をすることが不可能な，ペタバイト級の大容量なデータを簡単に扱うことが可能となっている．

Hadoop MapReduce は，Google 社が発案した MapReduce 技術を，HDFS は Google File System [?] をそれぞれ模造することでデザインされている．しかし，元の技術との直接の互換性や，共通しているプログラムコードはないため，概念的，機能的な類似のみとなっている．

Apache Spark

Apache Spark は，大規模なデータを，複数のコンピュータに分散して並列で処理を行わせるソフトウェアの 1 つである．ASF によって開発され，Apache ライセンスに基づき，オープンソースソフトウェアとして公開されている [?]．

複数のコンピュータを束ね，大規模なデータを分散しながら保管し，並行して計算処理を行う．その際に，各コンピュータへのデータの振り分けや実行の指示，実行結果の統合などは，　管理用のサーバである Cluster Manager によって行われる．

Spark は，データを RDD（Resilient Distributed Datasets: RDD）と呼ばれる管理の単位に分割してそれぞれのノードで管理する．RDD は，メインメモリ上で管理され，必要に応じてストレージに記録される．そのため，ストレージへの記録と読み込みを毎回繰り返さなくても済むため，高速で処理を行うことができる．

Hadoop では 1 回の分散処理ごと実行結果をストレージに記録するため，機械学習などの用途では性能が落ちてしまうという問題があったが，Spark はそれぞれのノード上のメモリを有効活用できるため，用途によっては Hadoop よりも高速に機能する．

また，Spark は Windows や Linux などのプラットフォームに対応しており，Java や Scala などのプログラミング言語での記述がサポートされている．さらに，Java API を経由し，実行環境に Java 仮想マシンを利用する，JVM 言語のサポートも行われている．

拡張機能として，ノードやデータ，分散処理の管理をする Spark Core や，管理下のデータに対して SQL による問い合わせと処理を行う Spark SQL，グラフの処理などの機能を提供する GraphX などが用意されている．

Dask

Dask は，Matthew Rocklin によって開発されたコミュニティプロジェクトであり，並列分散処理行うために用いられるソフトウェアである．効率的な数値計算を行うための多次元配列のサポートとそれを操作できるように拡張された，Python のライブラリの 1 つである Numpy や，データ解析を支援するために，時系列データや数表を操作できるデータ構造とその演算を提供している，Numpy と同様に Python のライブラリである Pandas を，Dask は簡単に並列・分散して処理を行うことが可能である．また，Dask は，上記の 2 つのライブラリと競合するライブラリではなく，それらをより高機能にしたラッパーライブラリのようになっている．

Dask による分散処理は，大量のデータを複数のブロックについて分割してから，処理することにより実現される．この仕組みによって，分割されたブロックは，1 度に全てのデータを読み込む必要がなくなるため，メモリ消費のピーク値を大幅に抑えることが可能となる．

Dask では，いくつかの Numpy 配列を格子状に配置された状態を 1 つの Dask 配列とみなし，Numpy 配列単体が Dask 上でのチャンクサイズとなり，同様に，Dask のデータフレームは図**??**のように，いくつかの Pandas データフレームで構成される．

Dask は主にデータ分析や機械学習に利用されていて，本研究にも使われるモジュールの一つである Pandas は，大容量のデータを処理する際には，分析に使われるデータが，メモリに収まらないことや，基本的に単一のスレッドで処理が行われるため，処理速度が遅いことが問題に上げられる．Dask による並列分散処理を行うことで，それらの問題は解決され，プログラムの処理速度向上に繋がる．

本研究などの組み合わせ最適化は膨大な量のデータから近似最適解を探索し，処理のたびに評価し，さらなる最適解を検討しているため，最適解を求める時間が長くなり，システムとして実用的ではないことが多い．そのため，組み合わせ最適化を適応したシステムに並列分散処理を施して短い時間で解を導いている事例もいくつか存在する．

第3章

ユーザ意見が反映されるシステム

§ 3.1　ユーザの身体情報を考慮した制限レシピ

遺伝的アルゴリズム（Genetic Algorithm: GA）とは，1975 年に，ミシガン大学の John Holland が提案した，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．

ランキング選択

ランキング選択とは，適応度の多さでソートし，その順番によってあらかじめ与え

た確率で選択する方法である．問題点として，適応度にあまり差がない個体でも選択確率に大きな差が生じる可能性があることである．また個体にランク付けするため，次世代がそろうたびにソートを行う必要がある．

ルーレット選択

ルーレット選択とは，個体群におけるそれぞれの個体の適応度と，それらの統計を導出し，適応度の合計に対する，各個体の適応度の割合を選択確率として個体を選択するという考え方である．これのメリットとして適応度の高い個体が次世代の個体として選ばれる可能性が高くなるが，適応度の低い個体であっても次世代の個体に選ばれる可能性も 0 ではないので，個体群の多様性を保つことができ，局所的な最適解に陥ってしまうことを防止できるということが挙げられる．

エリート保存選択

エリート保存選択とは，一つ世代の最も優位性の個体を，無条件にそのままの世代に残す方法である．確率のみにしたがって個体を選択する．また，交叉や突然変異のステップに進む場合，非常に適応度の高い個体が現れても消滅してしまうことがある．これは，局所的な最適解に陥るのを防ぐことにも繋がるが，良い解を少ない回数で得たい場合には障害になってしまう．そのため，エリート保存選択が提案されてきた．

交叉

交叉とは，生物が交配によって子孫を残すことをモデル化したものである． 基本的には選択によって選ばれた個体に対して，ある交叉位置における双方の染色体の一部ずつを組み換え，さらに，新しい個体の染色体を作成する遺伝子操作のことである．

突然変異

突然変異とは，ある一定の確率で，個体の染色体上の遺伝子を別の遺伝子に置き換える操作である．交叉によって生成される個体は，その親遺伝子に依存した限られた範囲の個体であるため，突然変異は個体群の多様性を維持する役割を担っているといえる．

突然変異の操作の後，適応度関数によって各個体の適応度を求め，選択，交叉，突然変異の過程を繰り返し，世代を交代しながら最適解を探索する．世代交代の過程で最適解が得られたとプログラムが判断された場合や，世代交代数を設定しておき，最終世代数に達したときに生き残っていた個体集団の中で，もっとも高い適応度の個体（個体群）を最適解と見なす場合に GA は終了する．

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

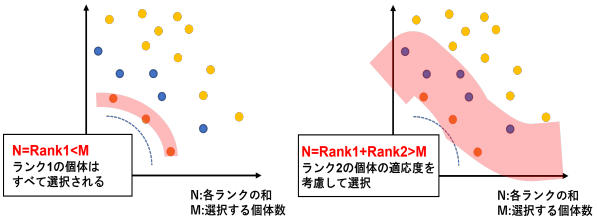


図 3.1: 混雑度トーナメント選択 [?]

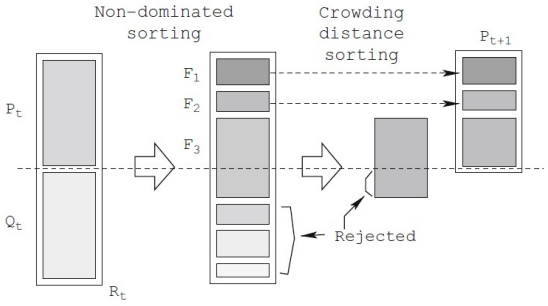


図 3.2: NSGA-Ⅱの流れ [?]

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．また，NSGA-II は NSGA 同様，非優越ソートを用いている．NSGA-Ⅱは，NSGA と比較して，以下の 3 点において変更と改良が行われている．

混雑度トーナメント選択の導入

混雑度トーナメント選択とは，ランクが高い個体を優先的に選択し，各ランクによって形成されるパレートフロントの中でも，特定フロントから限られた数の個体を取得する際に，そのパレートフロントから一様に分布するように個体を選択する手法である．混雑度トーナメント選択のイメージを図 3.1 に示す．NSGA-Ⅱでは，パレート保存する個体数は常に一定である．このことから保存したパレート個体を選択に反映させる方法を使用している．

混雑度ソートの導入

混雑度ソートとは，個体がどれほどばらつきがあるのかを評価する方法である．これは従来までのシェアリングに代わる手法となっており，解の両隣にある 2 つの解の平均の距離を表している．ここで，i 番目の解 $x^{(i)}$ の混雑度は，$x^{(i)}$ に関数値が最も近接する，$i-1$ 番目と $i+1$ 番目の解を $x^{(i-1)}$，$x^{(i+1)}$ として，$\overline{f}_j(\cdot)$ は，i 番目の目的関数 $f_j(\cdot)$ をその値域の幅で正規化した目的関数とすると，次のように定義される．

$$CD(x^{(i)}) = \frac{1}{k}\sum_{j=1}^{k} |\tilde{f}_j(x^{i+1}) - \tilde{f}_j(x^{i-1})| \tag{3.1}$$

NSGA-II におけるアルゴリズムの流れについて，図??に示す．また，各手順ごとの説明を以下に示す．

1. サイズ N の親母集団 P_t をランダムに作成したのち，サイズ N の子母集団 Q_t を作成し，P_t と Q_t を組合せて，集団 R_t を作成する．
2. 集団 R_t に対し，非優劣ソートを施し，全個体をランク毎（$F1$, $F2$,$F3$ ･･･ ）に分類する．
3. 新規の空白の親母集団 P_{t+1} を作成する．2 の非優劣ソートによる，ランクが上位の個体だけを親母集団 P_{t+1} とする．

4. 3で親母集団 P_{t+1} を構成する途中で，個体サイズが N を越えるとき，サイズ N を越えているランクの個体に混雑度ソートを施して，混雑度が高い個体を，個体のサイズが N となるまで排除していく．
5. 親母集団 P_{t+1} に混雑度トーナメント選択を施し，交叉・突然変異をすべき個体を選択し，遺伝子操作を行う．そのとき新たな子集団 Q_{t+1} を作成する．
6. 親母集団 P_{t+1} と子集団 Q_{t+1} を組み合わせ，新たな集団 R_{t+1} を作成する．
7. 2へ戻り，これまでの操作を終了条件が満たすまで繰り返し終了する．

§ 3.2 ロジスティ回帰分析によるUX項目の推定

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いている．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，胃がんとうの予防のため 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．2型糖尿病は，内臓脂肪型肥満によるインスリン抵抗性により発症することから，2型糖尿病の発症やさらなる悪化を防ぐためには肥満の是正が重要となる．

糖尿病を改善するために特に調整しなければいけ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．腎臓病が進行して腎機能が低下すると，腎臓から排泄されるべき物質が体内に蓄積し，高カリウム血症，高リン血症，尿毒症などの代謝異常を生ずる．これらに対して食事療法により対処する必要がある．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3 g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，血液中の資質の値が基準値から外れた状態のことをいう．血液中の 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．もしアレルギーを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??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えび，かに，小麦，そば，卵，乳，落花生（ピーナッツ）の 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

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

ている，アーモンド，あわび，いか，いくら，オレンジ，カシューナッツ，キウイフルーツ，牛肉，くるみ，ごま，さけ，さば，大豆，鶏肉，バナナ，豚肉，まつたけ，もも，やまいも，りんご，ゼラチンの計 28 品目とする．

§ 3.3　UX 項目の評価関数への組み込み

ブラウザベースとは，操作や利用の手段として Web ブラウザを利用することである．一般的にブラウザベースは Web アプリケーションの性質を表す語として用いられる．ブラウザベースと似た表現に「クラウドベース」がある．クラウドベースは，システムのプラットフォームがクラウド上に構築されていることをさす語である．

両者の違いとして前者はブラウザを通じで動作するため特定のソフトウェアやアプリを端末にインストールする必要がないがブラウザを通じてインターネット接続を行う必要があるためオフラインの環境では動作できない，という特徴がある．対して後者は専用アプリを端末にダウンロードした場合，オフラインの環境下でも一部機能が使えるという特徴がある．

また，ブラウザベースのアプリケーションの主な利点の 1 つとして，デスクトップアプリケーションのようにコンピュータにローカルにインストールするソフトウェアを購入する必要がないことが挙げられる．例えば Microsoft Office のようなソフトウェアは，コンピュータのハードドライブにローカルにインストールする必要があるが，ブラウザベースのアプリケーションでは，ソフトウェアがコンピュータ上でホストされていないため，このインストールプロセスは必要ない点がある．

Web ベースのアプリケーションのもう 1 つの利点として Web ブラウザとインターネットに接続するだけでシステムにアクセスできることがあげられる．また，それと同時にこれらのアプリケーションは，Web サイトまたは Web ベースのサービスが実行され，アクセス可能であれば，いつでも使用したいいつでもアクセスできる．

次に，ブラウザベースのアプリケーションの例を紹介する．利用可能な Web ベースのアプリケーションは幅広くあり，その数は増え続けてる． Web ベースのバージョンでは，電子メールアプリケーション，ワードプロセッサ，スプレッドシートアプリケーション，その他多数のオフィス生産性ツールなど，よく知られている種類のソフトウェアがある．具体的なブラウザベースのアプリケーションの例として,「Google Chrome」や「Yahoo! Japan」などが挙げられる (図??，図??).

「Google Chrome」とは Google が開発したクロスプラットフォームのウェブブラウザ

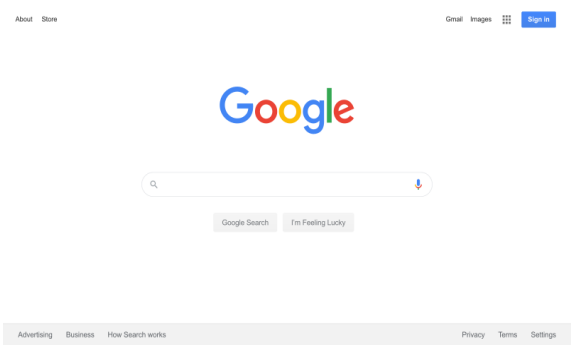


図 3.3: Google のホームページ

図 3.4: Yahoo! Japan のホームページ

である．Google Chrome のソースコードは，そのほとんどが Google のオープンソースの Chromium プロジェクトからのものであるが，Google Chrome は Google 社所有のフリーウェアとしてライセンスするようにした．また，開発言語としては，C/C++，Java，JavaScript，Python，Go が挙げられる．

「Yahoo! Japan」とは，Z ホールディングス傘下のヤフー株式会社が運営するポータルサイトのことである．サーチエンジンを中心に，ニュース配信，ネットオークションなどのサービスを提供している．開発言語として，Swift，Kotlin，JavaScript，Node.js，PHP，Java がある．

本研究においては，システムを HTML 上に適用し，システムの作成といったサーバーサイドの部分を「Flask」といわれる Python の Web アプリケーションフレームワークを使用する．

Flask

Flask とは，プログラミング言語 Python 用の，ウェブアプリケーションフレームワークである．標準で提供する機能を最小限に保っているため，自身を「マイクロフレームワーク」と呼んでいる．Flask は，必要最低限の機能しか搭載されていないため，他のウェブアプリケーションフレームワークに比べて動きが軽くなるという利点がある．そのため，多くの機能は必要としない単純な Web アプリケーションなどに適している．また，Flask は搭載されている機能が少ないためシンプルで学習が簡単であるという特徴がある．また，一般的な Web アプリケーションフレームワークでは，大体搭載されている機能をもとにアプリケーション設計していくため，カスタマイズすることは難しいが，Flask は多くの機能を自分で実装していく必要があるため，要件に応じた細かなカスタマイズをすることが可能である．

HTML

HTML とは，コンテンツの構造を定義するマークアップ言語のことである．HTML は一連の要素 で構成されており，これらの要素がコンテンツのさまざまな部分を囲み，一定の表示や動作をさせることができる．HTML は別ページに移動することができる「ハイパーリンク」をテキストを埋め込むことができることや，見出しや強調などのなどの印づけ (マークアップ) することができる特徴をもつ．

本研究は制限食と大人数料理に対応した献立作成システムをブラウザベースで実現することを目標とする．

第4章

提案手法

§ 4.1　調理時間とコストを最小化するパレート最適な献立

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ーが選択できる．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．今回用いるライブラリの１つであるpymooについて説明する．

pymoo

pymooとは，単目的最適化や多目的最適化などの最適化アルゴリズムを解くために使われるPythonライブラリの一つである．pymooはGAや，粒子群最適化，NSGA-Ⅱ，Nelder-Mead法などの最適化手法を，ライブラリからインポートすることによって簡単に使うことができる．pymooが扱える最適化手法の一部の例を表**??**に示す．また，多目的最適化問題であるZDT2からDTL6などの数十個のベンチマーク関数が数多く実装されており，自分で関数を自作し，その問題に制約条件などを設けることによりその問題に対して最適化を行うこともできる．さらに，関数をカスタマイズすることによって，ZDT5などのバイナリ決定変数の問題や，混合変数問題などの最適化問題も解くことが可能となる．また，多目的最適化による結果は，さまざまな視覚化手法を利用することにより出力できる．pymooで扱える視覚化の手法の一部を図**??**に示す．各手法は，それぞれで異なる目的を持っており，低次元もしくは高次元で表現される目的関数空間にも適応することができる．具体的な手法を挙げると，散布図や平行座標プロット，ヒートマップやレーダープロットなどが実装されている [**?**].

散布図は，2つの要素からなる1組のデータが得られたときに，2つの要素の関係を見るために作られたプロットしたグラフで，1つ目の要素が変化したときに，2つ目の要素はど

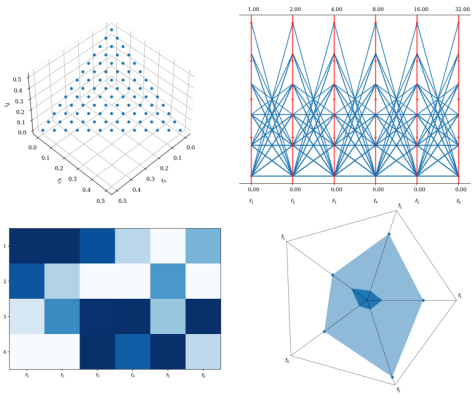


図 4.1: pymoo における様々な視覚化手法

表 4.1: pymoo の実装アルゴリズムの例

アルゴリズム	目的関数
GA	単目的
DE	単目的
BRKGA	単目的
Nelder-Mead	単目的
MOEAD	多目的
NSGA-II	多目的
CTAEA	多目的

のように変化するかを確認することができるものである．2 つの要素の間に何らかの関係がある場合，これらのデータ間には「相関関係」があるといえる．

3 変数以上の関係を把握することが困難であり，また各散布図の間で，点がどのような対応関係にあるのかは分からないことが多い．高次元データを，直交座標に映して把握すれば可能だが人間は 3 次元以上の空間は認識できないため，不可能である．それを何とかして可能にしようと考えられたものが，平行座標プロットである．平行座標プロットとは，各変数軸 (座標軸) を平行に並べたものである．これは，直交に並べることが不可能な場合に用いられる．これによって，ある程度の多変数間の繋がりを視覚化することが可能となる．使われている例として，医学研究者は，異なるそれぞれの薬物が様々な種類の細菌の増殖に与える影響を評価するため，平行座標プロットを作成して評価するということが挙げられる．

さらに，pymoo は並列処理，分散処理にも対応している．pymoo では，1 台の PC 内で各スレッドに処理ををどう割り当てるかなどの操作を行ったり，複数台の PC をクラスタとして用意して，処理の管理をするスケジューラとする PC から実際に処理をするワーカーとする PC に最適化処理に関するタスクを分散し処理することが可能となっている．

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 >

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

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

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

$$B^L \leq \sum_{i}^{R} r_{ki}C_i \leq B^H \qquad (\forall k) \tag{4.4}$$

$$\sum_{i}^{R} r_{ki}T_i \leq \tau_1 \qquad (k\%3 = 1) \tag{4.5}$$

$$\sum_{i}^{R} r_{ki}T_i \leq \tau_2 \qquad (k\%3 = 2) \tag{4.6}$$

$$\sum_{i}^{R} r_{ki}T_i \leq \tau_3 \qquad (k\%3 = 3) \tag{4.7}$$

$$0 < \sum_{i}^{R} r_{ki}\sigma_i \leq 1 \qquad (\forall k) \tag{4.8}$$

$$0 \leq \sum_{i}^{R} r_{ki}(1-\sigma_i) \leq 3 \qquad (\forall k) \tag{4.9}$$

$$\sum_{k=1}^{3D} r_{ki} \leq 1 \tag{4.10}$$

$$0 \leq \sum_{i}^{R} r_{ki}x_i < 1 \qquad (\forall k) \tag{4.11}$$

$$E_l^L \leq \sum_{i}^{R} y_i r_{ki}f_{il} \leq E_l^H \qquad (\forall k, \forall l) \tag{4.12}$$

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　対話型処理による利用者にとって最適な献立の出力

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．

また，他のシステムの利用形態としてバッチ処理，リアルタイム処理がある．バッチ処理とリアルタイム処理の流れを図??に示す。バッチ処理とはプログラム（データ）を処理目的ごとにまとめ，そのデータを順次処理していく一連の流れ，システムを指す．バッチ処理は特定の処理に人まとめて実行するので，大量のデータを一括に計算することに向いている．しかし一定のデータがたまったときにそのデータをまとめて処理をするため，リアルタイム性が求められる機能にはバッチ処理は向いていない．バッチ処理が向いている処理の事例として，事務所における給与計算，振込・請求処理などがある．

リアルタイム処理とは，端末から入力されたデータ，発生したデータを，処理要求が発生した時点から即時にコンピュータで処理する処理方法である．リアルタイム処理はデータの遅延が極めて少なく，処理に使われる情報が最新なものであるため，受け取ったデータの情報をすぐ知ることができ，問題も特定しやすいという利点がある．しかし，受け取ったデータをきわめて短い時間で処理をするため，処理をするハードウェアが高性能になり，単純なシステムで実装することが難しくなる．バッチ処理が向いている処理の事例として，GPS 追跡アプリケーションや，株式のリアルタイム売買などがある．

バッチ処理，リアルタイム処理はどちらも大量のデータを形式的な処理方法で処理をする事例には向いているが，ユーザの選好を反映した処理は難しい．対して処理対話型処理は，ユーザにとって最も適した解を出力したい場合によく使われている．そのため，本研究においては対話型処理によるシステムを開発している．

また，対話型の処理のシステムは，機能が増え，複雑化したシステムを専門的な知識を持っていない人でも利用できるようにすることを目的としている．そのため，表示の分か

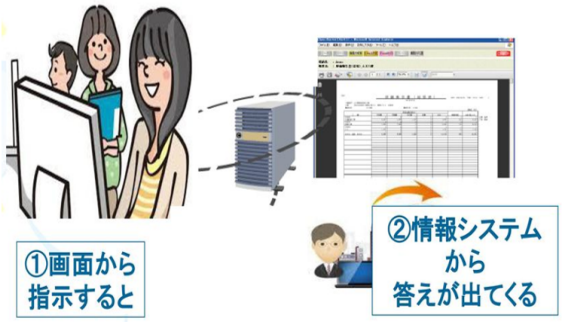


図 4.2: 対話型処理のイメージ

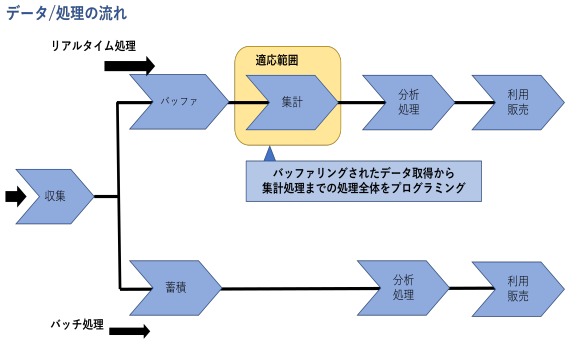


図 4.3: バッチ処理とリアルタイム処理

りやすさ，見やすさ，感覚的な操作が可能であることなどが重要な要素として挙げられる．そのため，対話型処理を行うソフトウェアやシステムには，GUI が用いられることが多い．対話型処理を行うソフトウェアやシステムのイメージを図?? に示す．GUI は，Graphical User Interface の略であり，コンピュータの画面上に表示されるウィンドウ，アイコンやボタン，ドロップダウンメニューなどを用いることで，マウスやタッチパネルなどのポインティングデバイスで感覚的な操作を可能とするインターフェースである．これと反対に，文字でコマンドを入力して操作するインターフェースは，Character User Interface の略で CUI と呼ばれている．また，現在の PC のインターフェースにはほとんど全てに GUI が採用されている．

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と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いての解は，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．

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，いくつも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．多目的最適化によって調理時間と材料コストを最小化するように得られた，入力した日数分の料理レシピをそれぞれ表すパレート最適集合の中から，ユーザがコンピュータとの対話型処理によって献立を選択する．

§ 4.3　提案システムの構成

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．

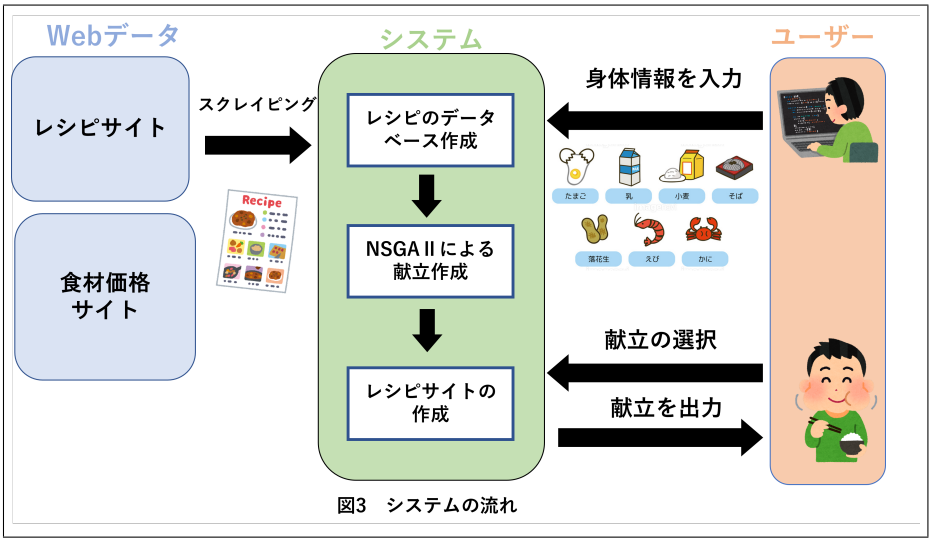


図 4.4: 提案手法の流れ

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」のみを参考にし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ける時間を選択する．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ける時間は多目的最適化を解く際の朝，昼，夜にかけ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．

NSGA-Ⅱ は，NSGA をエリート保存選択，混雑距離の導入，高速ソートの 3 点について変更と改良を施した手法であり，多目的最適化問題を解くアルゴリズムの１つである．目的関数には調理時間の最小化，使用する材料のコスト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

自動献立作成システムにおける，多目的最適化によって得られたパレート最適な献立から，ユーザ自身の希望に叶った献立を選択できるような対話型処理を行う．具体的には，調理時間と食材コストの 2 つの目的関数が最小化されたパレート解と，最適解におけるそれぞれのコストと調理時間を候補として表示し，ユーザに提示させる．ユーザは候補番号を選択し，最適解を評価する．これらの対話型処理によってパレート最適な献立の中から選好された，入力した日数分の献立の情報は，HTML 上に表示するようになっている．

自動献立作成の最適化プログラムにて出力された，入力した日数分の献立を構成する要素である料理レシピデータの料理名，料理イメージ画像，調理時間，食材コスト，摂取栄養素量や摂取カロリー，調理時間，必要な食材とその量，作り方などのデータを HTML に渡し，献立データを HTML 上に表示する．HTML にはその日に作るべきレシピの名前が表示されており，また，それぞれの日にちのページにジャンプするリンクが設置されている．ジャンプ先では，その日にちについての献立を構成する要素である料理が朝，昼，夕に分けられて表示される．なお，HTML は，本サーバー上に配置されている．これらの方法によって．ユーザに自動献立作成の最適化処理によって HTML 上に出力された，入力した日数の献立のデータを提示する．

第5章

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，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	・うずら卵	8個	炭水化物	6.2g	241	0
うずら卵の蒸し物	0	45	170	・生しいたけ	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A	160μg	241	0
うずら卵の蒸し物	0	45	170	・塩	少々	ビタミンB	0.13mg	241	0
うずら卵の蒸し物	0	45	170	・こしょう	少々	ビタミンB	0.44mg	241	0
うずら卵の蒸し物	0	45	170	・うま味調	少々	ビタミンB	0.27mg	241	0
うずら卵の蒸し物	0	45	170	(れんげ蒸し)		ビタミンB	2.2μg	241	0
うずら卵の蒸し物	0	45	170	・うずら卵	8個	ビタミンC	4mg	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	ideal
3	1200	8.49100E+02	1.23941E+03	1	0.00000E+00	f
4	1600	8.20700E+02	1.16736E+03	1	3.16300E+03	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	9.76273E+02	1	8.54000E+02	ideal
8	3200	5.68000E+02	9.29777E+02	1	1.58300E+03	ideal
9	3600	4.22400E+02	8.85080E+02	1	1.97600E+03	ideal
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	1.13600E+03	ideal
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	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	4.93082E+02	1	0.00000E+00	f
23	9200	2.86900E+02	4.75208E+02	1	0.00000E+00	f
24	9600	2.86900E+02	4.57967E+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

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 以上にすることする．

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

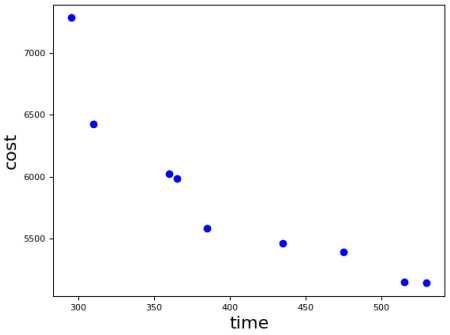


図 5.3: パレート解の出力

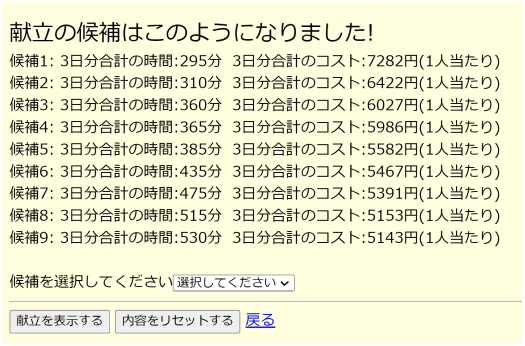


図 5.4: 対話型処理による解の選択

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_evel 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ーの変化，indicator はパフォーマンスインジケーターを表す．

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＞

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

< 基礎代謝量 >

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

< 必要推定エネルギー量 >

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

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 >

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

< 必要脂質 >

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

< 必要炭水化物 >

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

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
たんぱく質	5.5g
脂質	0.2g
炭水化物	3.7g
糖質	2.7g
食塩相当量	5.5g
食物繊維	1.0g
ビタミンA	5.0μg
ビタミンB1	0.05mg

食材名	食材量
梅干し	4個
塩わかめ	60g
鶏ささ身	1本

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。

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 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??，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，各時間帯別での，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 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 k cal，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．

第6章

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らばならなく，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し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] “生活習慣病の予防、食生活 生活習慣病の予防と食事-公益社団法人　千葉県栄養士会 ”, https://www.eiyou-chiba.or.jp/commons/shokuji-kou/preventive/seikatusyukan/, 閲覧日 2023.1.7.

[2] “食事療法について　国立研究開発法人　国立循環器病研究センター ”,https://www.ncvc.go.jp/hospital/pub/knowledge/diet/diet02/, 閲覧日 2023.1.7

[3] “【前編】給食業界で高まる AI 活用ニーズ〜「献立作成」「食数予測」課題とユースケース！”, 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] “管理栄養士観衆のレシピ検索・献立作成：おいしい健康”, https://oishi-kenko.com/, 閲覧日 2022.10.16.

[6] “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] John W. Ratcliff and David Metzener, “Pattern Matching: The Gestalt Approach”, Dr. Dobb ’ 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] Qingfu Zhang and Hui 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] John 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] “制限食にはどんな種類があるの? 健康ネット”, https://www.mcsg.co.jp/kentatsu/health-care/12106

[15] “減塩について ときわ会”, http://www.tokiwa.or.jp/nutrition/diet/low-salt.html, 閲覧日 2023.01.15

[16] “ちょっとした工夫でをコントロール　全国健康保険協会”, https://www.kyoukaikenpo.or.jp/g4/cat450/sb4501/p004/, 閲覧日 2023.01.15

[17] “日本人の食事摂取基準 (2020 年度版)　厚生労働省”, https://www.mhlw.go.jp/content/10904750/000586559.pdf, 閲覧日 2023.01.15

[18] “カリウムは調理の工夫で減らせます 東京医科大学病院”, https://articles.oishi-kenko.com/syokujinokihon/dialysis/05/, 閲覧日 2023.01.15

[19] “糖尿病 厚生労働省”, https://www.mhlw.go.jp/content/10904750/000586592.pdf, 閲覧日 2023.01.17

[20] “慢性腎臓病 厚生労働省”, https://www.mhlw.go.jp/content/10904750/000586595.pdf, 閲覧日 2023.01.17

[21] “慢性腎臓病の食事療法　東京女子医科大学”, https://www.twmu.ac.jp/NEP/shokujiryouhou.html, 閲覧日 2023.01.17

[22] “脂質異常症　厚生労働省”, https://www.mhlw.go.jp/content/10904750/000586590.pdf, 閲覧日 2023.01.17

[23] “高血圧　厚生労働省”, https://www.mhlw.go.jp/content/10904750/000586583.pdf, 閲覧日 2023.01.1

[24] “アレルギー　厚生労働省”, https://allergyportal.jp/knowledge/food/

[25] “pymoo: Multi-objective Optimization in Python ”, https://www.egr.msu.edu/ kde-b/papers/c2020001.pdf

[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. ll, pp. 1182-1189, 1982.

[27] “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] “一日に必要なエネルギー量と摂取の目安 - 農林水産省”, https://www.maff.go.jp/j/syokuiku/zissen_navi/balance/required.html, 閲覧日 2023.1.22.