卒業論文

多目的遺伝的アルゴリズムによる制限食を考慮した自動献立作成システム開発と高速化

富山県立大学 工学部 電子・情報工学科
1915077 水上和秀
指導教員 Antonio Oliveira Nzinga Rene 講師
提出年月: 2023年2月

目 次

図一覧

表一覧

記号一覧

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，我が国の主要な死因である．生活習慣病は，脳血管疾患や心疾患，悪性新生物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いことから，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き，ある日突然として命に関わる疾患を引き起こす可能性がある．生活習慣病による疾患を引き起こしてからでは，既に手遅れであるため，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控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，日々意識して努めていく必要がある．

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，栄養バランスのとれた健康的な食事をとることについて着目する．生活習慣病を予防するための健康的な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，自分にとって適正なカロリーの食事をとるように心がけること，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 [1].

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 [2]．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，学校給食や病院食の現場では，毎日の食事は学生や患者にとって整えるべき生活リズムの１つの要素であり，食事の時間は日々の楽しみの１つでもあることから，学生や患者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，栄養バランス，食事にかかる金額などの考慮すべきである献立作成条件を，1日だけでなく週間や月間でのバランスを考えながら設定する作業を，何度も繰り返し見直しながら改善していく必要があるため，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 [4]．それに従って，こ

れらの負荷の高い業務を，AI や数理計画化によって自動化するツールが存在している．

§ 1.2　本研究の目的

栄養バランスの取れた献立を制作するには，莫大なメニューの組み合わせや，栄養価の複雑な計算を考慮する必要があり，献立を考える時間がそもそも無かったり，自分で献立を考えることが面倒だと考える人は少なくない [5]．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る献立を作成することは，忙しく時間がない人や空き時間を作りたい人，できるだけ節約をして料理を作りたい人にとっては理想的である．

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コストの計算などの様々な条件を加味した上で繰り返し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 つでもあることから，食の感動を大切にし，病院では医食同源の精神を基本に飽きのこない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，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捉えることにより，栄養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，使用者が摂取したい栄養量や摂取カロリーについての希望を叶えられるように，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 状に存在するレシピサイトから 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，必要食材や接種栄養量，カロリーなどが挙げられる．また，料理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．

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 ライブラリが存在しているため，Python によって最適化プログラムを記述する．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が使用できるような対話型処理を用いる．具体的には，調理時間とコストそれぞれのバーを表示し，重視したい方にバーを動かすことで，それに叶った料理が出力される処理を行う．

また，大量の料理データから最適な料理を選別し，組み合わせる献立作成の最適化には，膨大な時間がかかる可能性があるため，実生活でのシステムの使用を考えたとき，少ない空き時間に，多目的最適化によって献立を作成することは現実的ではないと考える．そこで並列分散をはじめとしたあらゆる処理を施しプログラムの高速化を図ることにより，プログラムを利用者にとって使いやすいように改良する．

最後に，本研究によって提案された自動献立作成システムを動作させ，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，

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

自動献立作成支援システムの概要

§ 2.1　web 上のレシピデータを活用した献立作成

現在，cookpad やクラシル，おいしい健康，ボブとアンジー [9] 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，料理のイメージ，和食や洋食，主菜や副菜などの料理のジャンル，汁物や丼もの，鍋料理などの料理タイプ，必要な材料名とその材料数，調理工程，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．

レシピサイトのひとつであるボブとアンジーの料理レシピ名，イメージ，得られる栄養素，必要な材料などが乗っているページの例を図 2.1 に示す．また，生鮮食品や加工食品，畜産品などの諸品の最低販売単位での価格動向を先月や前年同月と比較した情報を提供しているサイトが存在している（図 2.2 参照）．

本研究では，献立作成システムにて，摂取栄養量という条件のもと献立を作成したいため，料理から摂取できる栄養量を細かく記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，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 [10] から，食材とその価格のデータを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 2.3 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，必要なデータが API によって提供されてないかまず確認することが重要である．

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 サイトであったり，JavaScript で記述されている Web ページのスクレイピングを行うときは Selenium が用いら

(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: ボブとアンジーのレシピページ例

れている，それらの処理が必要でない Web サイトには，高速で処理ができる BeautifulSoup4 が用いられている．

BeautifulSoup4

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ができ，そのオブジェクトから要素検索をすることで必要な情報を抽出する．

要素を検索する際に，条件の合う最初の要素を取得する select_one 関数や，条件に合う全ての要素を取得する select 関数，find 関数などがある．select と find の違いは，引数に指定する条件の指定方法にあり，select 関数では，CSS セレクタを指定して要素を取得し，find 関数では要素名や属性キーワードを指定して検索し，要素を取得する．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．

Selenium

Selenium は，Web ブラウザにおける操作を，自動的に操作することが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 テストであったり，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の多彩な用途で利用されている．

スクレイピングしたレシピデータは，1 つの料理につき 1 つの CSV ファイルに出力する．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

図 2.2: 食材価格動向調査サイトの例

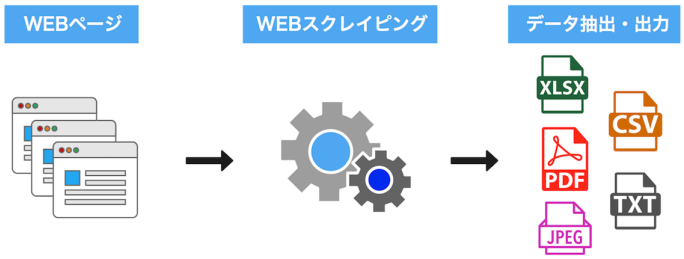

図 2.3: Web スクレイピングの流れ

ルをデータベースに蓄積し，本研究で使用する自動献立作成プログラムの入力データとして活用する．

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 情報を取得する．次に，HTML や XML を解析することができる，同じく 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，微妙な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 のライブラリの関数にて，デフォルトの値であった 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 を用いる．類似度計算をしたのちに一致するものが見つからなかった場合，ショッピングサイトである楽天市場の食品カテゴリから，その材料名で検索をかけて，材料名とそ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．次に，類似度計算に用いた difflib とその技術について説明する．

difflib

diffilib 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部分を抜き出して，その抜き出した文字列の前後に対しても同じ処理を繰り返す，ゲシュタルトパターンマッチングと呼ばれるアルゴリズムを使用して，文字列の類似度計算とその表示を行っている．get_close_matches 関数では，特定のキーワードに類似した文字列を抽出するために，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，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%以上の一致率であったら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り，Ratcliff, Obershelp によって 1983 年に考案された [11]．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，

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

で表される．ここで，K_m はマッチした文字数であり，D_{ro} は 0 から 1 の範囲をとり，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る．

§ 2.2　多目的最適化による最適な献立の提示

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」であり，メリットとして自動化による結果が出るまでの労力，作業時間が削減されることや，現実的ではない時間がかかる答えを導くことができることが挙げられる。

最適化問題の種類の一つとして、組み合わせ最適化問題が挙げられ、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 つにとどまらず複数存在するため，集合である．パレート最適解の集合のイメージを図 2.4 に示す．複数のパレート最適解を，目的関数空間にプロット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ており，実際にはこの中から解を選択することになる．

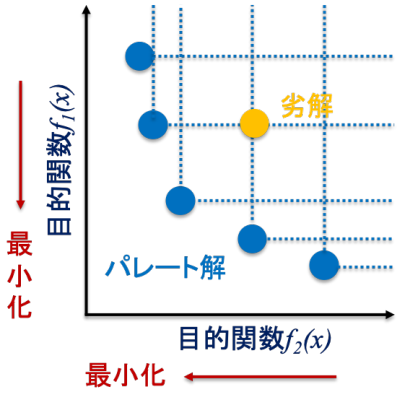

図 2.4: パレート最適解のイメージ

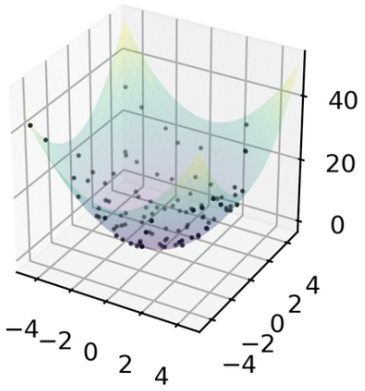

図 2.5: 解探索のイメージ（粒子群最適化）

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中には，複数の局所的最適解を持つ．局所的最適解とは，その近く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，制約条件を満たす，実行可能領域全体で，最も良いことが保証されている解となる．

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が，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ときに，大域的という形容詞をつけて大域的最適解と呼ばれることが多い．

局所解を回避する方法は，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が，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である．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．次に，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，

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

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

のように定式化される．式 (3.1) で与えられるベクトル関数を式 (3.2) の制約条件のもと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求める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ている．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 次元の単体を，アメーバのように動かしながら目的関数の最小値を探索する方法である．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 [12].

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 2.5 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 1 に提案された [13]，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 [14]，中でも Weighted Sum は最もシンプルで，各目的関数に重み付けをした値の合計をスカラー化した関数値とするものである．初期化，交叉，突然変異，理想点の更新，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　並列分散処理による解法

現在の情報社会では，社会や経済の問題解決や，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，人，プロセス，システム，組織などに関するデータは年中収集され，膨大な量のデータが生成されている．しかし，問題となるのは，この膨大な量のデータを，高速かつ効率的に処理する方法である．そこで，並列分散処理という技術が存在する．並列分散処理とは，複数台のコンピュータを用いて複数の CPU や，メモリを使うことで一つの計算処理を行い，性能や計算速度の向上を図ることである．並列分散処理を行っているときのイメージ図を図 2.6 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素早く結果を得られること，１台では困難な大規模な処理を実現できる性能の向上が挙げられる [?].

オーバーヘッド

並列分散処理における注意点として，コンピュータシステムで，何らかの処理を実行する際にかかる時間的，または空間的なコストやコンピュータにおける負荷のことを指す，オーバーヘッドが挙げられる．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向上の恩恵を上回ってしまうことがありえてしまう．

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場合に，複数のプロセスと各スレッドの起動，終了処理や，並列化するためのデータ分割と結果の統合処理などにかかる時間の合計が，並列化によって削減された時間合計を超えてしまうと，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．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て，実行を再開する際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．

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，MPICH，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，MPICH，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 [18].

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

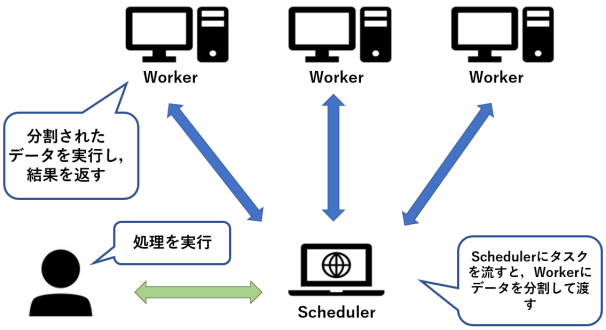

図 2.6: 並列分散処理のイメージ

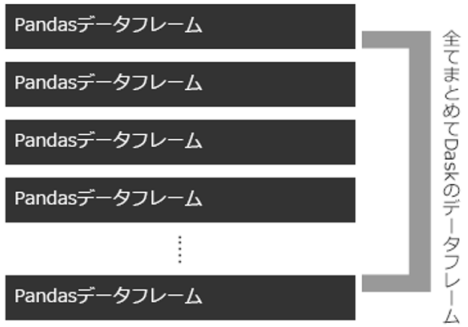

図 2.7: Dask のデータフレームのイメージ

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 [19]，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 [20] 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 [21]．

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 2.7 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．

第3章

健康のための制限レシピの考慮

§ 3.1　遺伝的アルゴリズムによる解法

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 [15]．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を遺伝子で表した「個体」を複数体用意し，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，生命の進化過程である交叉，突然変異や淘汰などの操作を繰り返し行うことで最適な解の探索をする．

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」の操作により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 1 世代目の個体はランダムに生成する．このとき，個体の各遺伝子のパラメータはランダムに設定がされている．

次に，その世代のすべ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環境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．

選択

選択とは，新しい世代を生み出す際の遺伝子操作の一つであり，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むように選択を行う．選択の方法として，ルーレット選択，トーナメント選択，ランキング選択，エリート選択があるが，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，それらの統計を求め，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．従って，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一方，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ゼロではないので，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メリットがある．

トーナメント選択

トーナメント選択とは，個体同士のトーナメント方式によって優位性を判断し，より優位性を持つ個体を選択していく方法である．トーナメントサイズの設定により，どれくらい淘汰するかといった具合を調整できる．

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．確率のみに従って個体を選択し，交叉や突然変異のステップに進むと，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ということは，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，新しい個体の染色体を作成する遺伝子操作である．

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，上記の選択，交叉，突然変異の過程を繰り返すことで，世代を交代しながら最適な解を探索する．世代交代の過程で最適解が得られたと判断された場合や，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 GA(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 [16]，GA を，多目的最適化問題に拡張したものである．

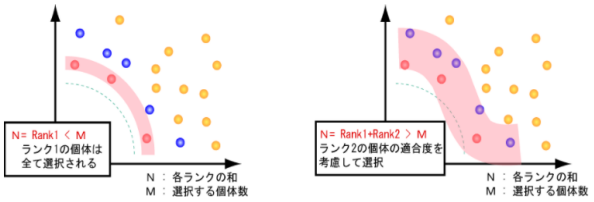

図 3.1: 混雑度トーナメント選択のイメージ

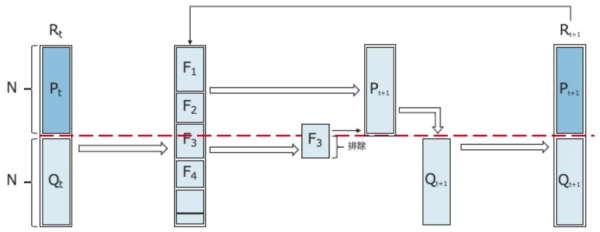

図 3.2: NSGA-Ⅱ の流れ

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 [17] 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 つとなっている．

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．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，さらに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，特定フロンとから限られた数の個体を抽出する場合，そのフロントから一様に分布するように個体を選択する方法である．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用いている．

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り，従来までのシェアリングに代わる手法となっていおり，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 3.2 に示す．また，各 Step ごとの説明を以下に示す．

1. サイズ N の親母集団 P_t をランダムに生成したのち，サイズ N の子母集団 Q_t を生成し，P_t と Q_t を組合せて，集団 R_t を生成する．

2. 集団 R_t に対し，非優劣ソートを実施することで，全個体をランク毎（$F1, F2, \cdots$）に分類する．
3. 新しい空の親母集団 P_{t+1} を生成する．2 の非優劣ソートによる，ランクが上位の個体だけで親母集団 P_{t+1} を構成する．
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実施して，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実施し，交叉・突然変異すべき個体を選択したのち，遺伝子操作を行い，新たな子集団 Q_{t+1} を生成する．
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} を生成する．
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す.

§ 3.2 健康のための制限レシピの考慮

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．[30]

塩分制限食

塩分制限食とは，摂取塩分量を１日６ g 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。[32]

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。[34]

糖質制限食

糖質制限食とは、エネルギー源となる３大栄養素のうち、糖質の摂取を原料あるいは調整する食事療法のことである。糖質は、生きていくうえで必要なエネルギー源となっている重要な栄養素である。しかし、糖質を過剰に摂取することにより動脈硬化や心筋梗塞、糖尿病になるリスクが高まり、逆に制限しすぎると頭痛・イライラなど、様々な体調不良を引き起こす可能性があるので、適度な量の糖質を摂取することが必要である。糖質制限食の目安として、食事一食当たりの糖質量を 25〜40g とし、それとは別に感触からの糖質 10g を含めた１日の糖質量を 70〜130g にすることが推奨されている。[33]

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、新党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。[35]

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、血糖値の情報を抑える働きが低下してしまい、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。

アメリカで行われた糖尿病予防プログラムでは、糖尿病発症リスクの高い対象において、３年間で５％の体重の低下は、糖尿病の発症を 55 %抑制したとしている。[36] このことから、体重を落とす食事療法は糖尿病の改善に貢献することがわかる。

糖尿病を改善するために調整しなければいけない栄養素は炭水化物、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。[36] 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。[37] 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。[38]

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、６ g/日未満、３ｇ以上が推奨されている。[38]

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。[39]

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。脳血管の動脈硬化が進むと、脱力感や激しい頭痛が起き、場合によっては意識を失うこともある。高血圧の発症・憎悪は、生活習慣と遺伝要因の相互作用から成り立っており、食事を含めた生活習慣改善は高血圧の改善・重症化のみではなく、発症予防においても重要である。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６ g/日未満としている。[40]

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利用でき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。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「特定原材料等」に指定されている，えび，かに，小麦，そば，卵，乳，落花生（ピーナッツ），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 28 品目とする．

§ 3.3　自動献立作成の実用化のための短時間化

先述した通り先行研究においては「ボブとアンジー」から献立作成のレシピを参照していた．しかし「ボブとアンジー」には主食が掲載していないため, 出力結果に主食が出力されないという問題が生じていた．

そこで本研究では「ボブとアンジー」に加え,「eatsmart」と「おいしい健康」の二つのサイトからスクレイピングを行い, 多種多様なレシピを参照し, より実用的な献立作成を提案する．

「eatsmart」は株式会社 EatSmart が運営しているレシピサイトであり, 主に主食の栄養素を掲載している．また,「おいしい健康」は管理栄養士が考案したレシピが掲載されているレシピサイトである．

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を目的関数，必要栄養素や摂取カロリーなどを制約条件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、献立の日数はユーザーが選択できるものとする。

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応用した手法を用いる．なお，Python のライブラリである pymoo を用いて，NSGA-Ⅱによって多目的最適化問題を解かせるよう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最適である献立は，摂取栄養量やカロリー，主菜と副菜の数の制限などの制約条件をみたし，調理にかかるコストと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．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 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サポートする Python ライブラリである．pymoo は，遺伝的アルゴリズムや，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扱うことができる．pymoo にて扱える最適化手法の一部の例を表 4.1 に示す．また，多目的最適化問題である，ZDT1 や，DTLZ1 などの数十個のベンチマーク関数が用意されているほか，自分で関数を自作し，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にも対応しており，様々な問題に対して定式化が可能である．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 4.1 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適応できる．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 [27].

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 つ目の要素はどのように変化するかを確認することができる．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 変数以上の関係は把握するのが困難であり，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できるこ

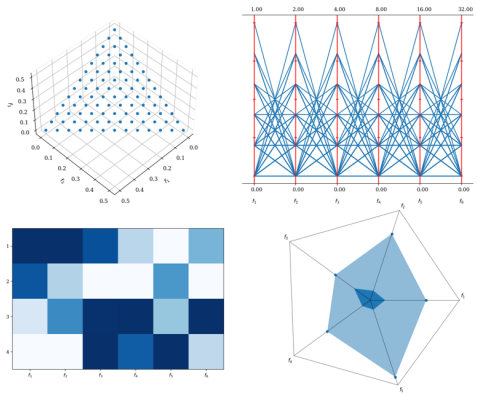

図 4.1: pymoo における様々な視覚化手法

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

とができたら良いが，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．平行座標プロットは，その名の通り，各変数軸 (座標軸) を平行に並べたものであり，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，ある程度は多変数間の繋がりを視覚化することができる．使われている例として，医学研究者は，異なる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に，平行座標プロットを作成して評価する．

また pymoo は並列処理、分散処理にも対応しており、1 台の PC 内で各スレッドにタスク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，スケジューラとする PC から PC クラスタからなるワーカー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．

次に，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を定式化する．先に各変数の説明を行ってから，目的関数と制約条件式を提示し，最後にそれぞれの式についての説明を行う．

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における最大値を 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，基礎代謝量の制約における最大値を 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．

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 ＞

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

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

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

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

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

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

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

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

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

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

目的関数

最後に，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，上から調理時間の最小化と，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，期間 1 週間での料理の組み合わせを表現する．

制約条件

次に，制約条件は，式 (4.3) から式 (4.10) の 8 つであり，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 1 週間で構成する献立の中に，同じ料理が存在しないようにする制約である．

摂取栄養量制約は，1 日当たりに摂取する，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 日当たりに摂取する，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，朝，昼，夜の各時間帯における献立にかかる調理時間に，それぞれ上限を設定し，朝は調理時間を少なくしたい，夜は調理時間を多くしてもいいなどの希望を叶えられるようにする．

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　高速化のために用いた技術

§ 4.3　提案システムの構成

本研究で提案する, 制限食を考慮した自動献立作成システムの流れを図に示す。まず最初に, 献立作成の最適化に必要な, レシピデータと食材価格データを,Web サイトからスクレイピングし, データベースに蓄積する。

次に, ユーザーが身体情報やアレルギー情報, 患ている生活習慣病を入力する。その際にアレルギーや嫌いな食品が含まれるレシピをデータベースから削除する。

そして蓄積されたレシピデータ, ユーザの身体情報を入力として, 摂取栄養素やカロリーなどの制約条件のもと, 調理時間, 調理コストの最小化を目的関数に設定した最適化問題を, 制約条件を考慮した遺伝的アルゴリズムによって解く。最後に, 設定した日にち分献立をユーザに出力する。さらに, 最適化の工程で複数の PC を利用し, 並列処理を行うことでプログラム実行時間の速度向上を図る

第5章

数値実験並びに考察

§ 5.1　数値実験の概要

§ 5.2　実験結果と考察

第6章

おわりに

mitei

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] 工藤一彦, “からだの不調を食事で治す”. 女子栄養大学出版部, 2001.

[4] “【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.

[5] 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.

[6] 日本栄養士会編 (梶本雅俊, 大谷八峰, 白鷹増男, 他), “栄養指導に役立つコンピュータ入門 ”, 第一出版, 1983.

[7] Joseph L. Balintfy, G. Terry Ross, Prabhakant Sinha and Andris A. Zoltners, “A Mathematical Programing System for Preference and Compatibility Maximized Menu Planning and Scheduling”, Mathematical Programming, Vol.15, No. 1, pp. 63-76, 1978.

[8] 辻 明日夏, 倉重 賢治, 亀山 嘉正. “ファジィ数理計画法を用いた料理の選択”, 知能と情報（日本知能情報ファジィ学会誌）, Vol. 20, No. 3, pp. 337-346, 2008.

[9] “料理レシピ ボブとアンジー 管理栄養士監修の健康ヘルシーレシピ”, https://www.bob-an.com/, 閲覧日 2022.10.16.

[10] “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.

[11] John W. Ratcliff and David Metzener, “Pattern Matching: The Gestalt Approach”, Dr. Dobb ’s Journal, p.46, 1988.

[12] 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.

[13] 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.

[14] 安藤祐斗,“ Web 情報を活用した自動献立作成のための多目的遺伝的アルゴリズムによる並列分散処理 ”富山県立大学学位論文，2022.

[15] 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.

[16] John H. Holland, “Adaptation in Natural and Artificial Systems”, 1975.

[17] 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.

[18] D.E.Goldberg, “Genetic algorithms in search, optimization and machine learning ”, Addison-Wesly, 1989.

[19] “Apache Hadoop ”, https://hadoop.apache.org/, 閲覧日 2023.01.11.

[20] Jeffrey Dean, Sanjay Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters ”, OSDI’04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, pp. 137-150, 2004.

[21] Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung, “The Google File System”, Google, 2003.

[22] “ Unified engine for large-scale data analytics”, Apache Spark, https://spark.apache.org/, 閲覧日 2022.12.26.

[23] 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.

[24] 田村坦之, 中村豊, 藤田眞一, “効用分析の数理と応用”, コロナ社, 1997.

[25] 中山弘隆, 谷野哲三, “多目的計画法の理論と応用”, 計測自動制御学会, 1994.

[26] “食品成分データベース”, https://fooddb.mext.go.jp/, 閲覧日 2023.1.28.

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

[29] “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.

[30] “主食・主菜・副菜とは?献立作りのポイントとあわせて紹介”, https://www.morinaga.co.jp/protein/columns/detail/?id=166&category=health, 閲覧日 2023.1.22.

[31] “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

[32] “塩分制限食 京都市立病院”, https://www.kch-org.jp/outline/section/eiyo/塩分制限食, 閲覧日 2023.01.15

[33] “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

[34] “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

[35] “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

[36] “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

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

[38] “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

[39] “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

[40] “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

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