卒業論文

複数の評価基準から
合理的な総合評価による合意形成の支援

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

放送大学　教養学部情報コース
2010022122　奥原由利恵
指導教員
提出年月: 令和5年 (2023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 [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 [3]．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 [4]．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について示す．目的は制限食を考慮した多目的遺伝的アルゴリズムによる最適な自動献立作成について提案することを述べる．

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

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

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

その後，提案手法について説明する．

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

第 6 章 本研究で述べている提案手法をまとめて説明する．また，今後の課題について述べる．

第2章

サイバー空間からのデータ収集と処理

§ 2.1　多様な要因を考慮したデータセットの作成

地域経済分析システム（RESAS）

RESAS とは，地域創生の実現に向けて，内閣府地方創生推進室ビッグデータチーム，内閣官房まち・ひと・しごと創生本部事務局と経済産業省地域経済産業調査室が提供している WEB アプリケーションである．日本全国の各自治体区分における人口，産業，観光，まちづくり，医療・福祉，財政など幅広い分野のビッグデータを官民より集約し，表，グラフ，マップなどのフォーマットを用いてデータを可視化するシステムとして提供されている．

また，データを可視化するだけではなく，集積されたデータをもとに各テーマに沿った分析を行うことや，API を利用することによってシステムにデータを直接取り込むことも可能である．これらの機能は地方自治体をはじめ，その他地域の活性化に関心を持つ人々に対して一般に公開されており，地方自治体における地域課題の抽出， 地域版総合戦略の立案といった活用法に加えて，地方創生に関心のある民間の団体・個人による活用も期待される．

e-Stat

日本の政府統計に関する情報のワンストップサービスを実現することを目指した政府統計ポータルサイトです。これまで各府省等が独自に運用するＷｅｂサイトに散在していた統計関係情報を本サイトに集約、社会の情報基盤たる統計結果を誰でも利用しやすいかたちで提供することを目指し、各府省等が登録した統計表ファイル、統計データ、公表予定、新着情報、調査票項目情報、統計分類等の各種統計関係情報を提供していきます。

不動産情報ライブラリ

不動産情報ライブラリとは、不動産の取引価格、地価公示等の価格情報や防災情報、都市計画情報、周辺施設情報等、不動産に関する情報をご覧になることができる国土交通省の WEB サイトです。

犯罪データ

特定の施設やその近くは，犯罪の発生の要因となる可能性がある．施設データを取得できるサービスとして，Google Maps API が存在するが，無料で取得できる数に制限がある

ほか，たとえば遊園地や水族館など，レジャー施設としてジャンル分けできるものに対して,「レジャー施設」と検索しても，それらを網羅できるとは限らない点で，採用しなかった．そこで，ナビゲーションサービスのひとつである「NAVITIME」から施設データをスクレイピングして取得することとした．

スクレイピングとは，データを収集した上で利用しやすいように加工をすることである．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．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り，スクレイピングとは違い，これは，単に Web 上のデータを収集することを意味する．データを活用するために，使いやすく抽出や加工をしたりするのがスクレイピングの特徴である．

BeautifulSoup4 とは，Web サイト上の HTML から，必要なデータを抽出することができるライブラリである．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 関数などがある．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．

NAVITIME は，施設のジャンルごと，さらには都道府県ごとに一覧となって表示される．

説明変数の選定

犯罪の発生にはさまざまな要因が考えられる．そのため，できるだけ多くの説明変数を考慮することが望ましい．しかしながら，むやみやたらに目的変数と相関がない説明変数を追加しても，予測精度が上がるどころか，計算コストが増大するだけであろう．また，本システムで統計データを取得するために利用している e-Stat は，グリッドセル・小地域ごとのデータに絞っても，200 以上公開されている．さらに，それらを別のデータと計算し，新たな説明変数を作成することを許せば，その組み合わせは考慮しきれない．

そこで，機械学習による犯罪予測モデルを作成する際に，説明変数を容易に選定することができるようにしている．選定できるデータは，e-Stat で公開されているグリッドセル・小地域ごとの統計データ，および，NAVITIME で公開されている施設データである．前者については，異なるデータ間で計算した結果を説明変数として使用できるようになっている．

異なる空間的解像度のデータの結合

e-Stat で提供されている統計データは，集計されている区分ごとに，全国ごと，都道府県ごと，市区町村ごと，”…丁目”といったの小地域ごと，グリッドセルごとの５種類が存在する．本システムでは，予測する空間的な単位をグリッドセルとしているため，グリッドセルごとの統計データを使用するが，より考慮できる要因を増やすため，小地域ごとの統計データも使用できるようにした．小地域ごとのデータはそのまま用いることはできないため，グリッドセル単位に変換する必要がある．そのため，小地域ごとのデータについて，小地域全体に均等に分布していると仮定し，対象のグリッドセルに重なっている割合だけを足し合わせる．すなわち，対象のグリッドセル C に重なる小地域 $A_1, A_2, \cdots, A_n$ について，それぞれの全体の面積を $S_1, S_2, \cdots, S_n$，対象のグリッドセルと重なる面積を $s_1, s_2, \cdots, s_n$，データ値を $x_1, x_2, \cdots, x_n$ とすると，対象のグリッドセル C のデータ値 X を次のように算出する．

$$X = \sum_{k=1}^{n} x_k \frac{s_k}{S_k} \tag{2.1}$$

統計データとして公開されることの多い要素のなかには，犯罪発生の要因となり得るものが多く存在し，それらが豊富に公開されている e-Stat から，ドメイン知識をもとに自由に説明変数を選択できるようにしたことは，大きな利点と考える．

施設の最短距離と立地数の算出

さまざまなジャンルの施設について，NAVITIME からスクレイピングを行い，施設名と，その緯度と経度を取得する．本システムでは，それぞれのジャンルごとに，対象のグリッドセルに含まれる数と，最も近くにある施設までの距離を説明変数とする．

なお，地球は楕円体であるため，単純なユークリッド距離では誤差が生じてしまう．そこで，本システムではヒュベニの公式 [?] を用いて，距離を算出している．対象のグリッドセルの中心を $P_o(x_o, y_o)$，注目する施設を $P_n(x_n, y_n)$ とすると，それら 2 点間の距離 D は以下で求まる．

$$D = \sqrt{(D_y M)^2 + (D_x N cosP)^2} \tag{2.2}$$

$$M = \frac{R_x(1 - E^2)}{W^3} \tag{2.3}$$

$$N = \frac{R_x}{W} \tag{2.4}$$

$$W = \sqrt{1 - E^2 sinP^2} \tag{2.5}$$

$$E = \sqrt{\frac{R_x^2 - R_y^2}{R_x^2}} \tag{2.6}$$

多くの人が集まりやすいレジャー施設やショッピングモールは，犯罪生成・誘引要因となりやすい．逆に，人気が少ない駐車場は，犯罪可能要因となり得る．施設に関する説明変数を自由に選択できるようにしたことは，犯罪要因の特定に役立つことが期待できる．

地図画像にもとづく説明変数の抽出

Mapbox Static Tile API を利用して，それぞれのメッシュに対応する地図画像を取得する．本研究では，建物，道路，水，空き地の４つを色分けした地図画像を取得し，それぞれの画像の大きさに対する面積の比率を説明変数としている．他人による自然な監視は犯罪を抑制する．たとえば，道路や建物の面積比率が大きいほど，監視の量が増え，犯罪が起こりにくく，逆に空き地の面積比率が大きいほど，犯罪が発生しやすい傾向があるならば，それぞれの説明変数は有用なものとなるだろう．

対象の要素の面積比率 p_a は，地図画像の大きさを $n \times m$，その要素と同一の RGB 値をもつピクセル数を x とすると，以下のように算出する．

$$p_a = \frac{x}{nm} \tag{2.7}$$

なお，Mapbox Static Tile API によって取得する地図画像は，本来は画像処理を目的としていない．そのため，Mapbox Studio 上で指定した RGB 値と誤差があるピクセルがある．そのため，対象のピクセルの RGB 値と，それぞれの要素の RGB 値とのユークリッド距離を算出し，最も小さい要素を指定する．

また，地図画像から道路ネットワークを抽出し，道路に関連する属性を説明変数として抽出する．まず，道路とそれ以外の 2 値画像に変換し，ノイズを削除するためにオープニング処理を行う．その画像に対して，ネットワークを抽出するアルゴリズム [?] を使用し，ネットワークの属性であるノード数 N，エッジ数 E を取得する．また，それらから密度 d，平均次数 k を以下のとおり算出する．

$$d = \frac{2E}{N(N-1)} \tag{2.8}$$

$$k = \frac{2E}{N} \tag{2.9}$$

なお，密度 d と平均次数 k は，道路のネットワークとしてみたとき，それぞれ次のような特徴をもつ [?]．密度 d が大きい道路ネットワークは，道路が網目状に相互に接続された状態であり、幅員の狭い生活道路であると考えられる．また，平均次数 k が小さい道路ネットワークは，交差点の少ない直線的な道路が多いと考えられる．

動的データと静的データの組み合わせ

上で述べたものはすべて，1 日ごとに変化しない静的データであった．それらと，1 日ごとに変化する動的データから，空間（グリッドセル）軸 × 特徴量 × 時間軸の，3 次元のテーブルを作成する．本システムでは，動的データとして，対象のグリッドセルやその周囲で，過去一定期間に発生した犯罪発生件数や，平均気温，日照時間，降水量，降雪量などの天候データを用いる．前者については，犯罪の発生には近接反復被害効果があることから採用した．また，後者については，たとえば，雨や雪が降っている日は，自転車を使う人が減少し，それと同時に自転車盗難も減少するなど，天候も少なからず犯罪発生に寄与すると考え，採用した．

以上により，機械学習に用いるデータセットの作成が完了する．

本章では，実際の犯罪発生データを用いて，4 章で述べた手法で犯罪発生予測モデルを作成し，その精度を確認，および考察を行う．また，その予測モデルを用いて，要因を可視化したマップを作成し，考察を行う．

データセット

予測に用いる説明変数は，表??に示す計 65 個とした．

3,126 個のグリッドセルについて，静的データをまとめたテーブルを作成し，それに動的データを追加した 3 次元のテーブルを作成する．予測モデルの精度を検証するため，デー

タセットのうち，2020 年 8 月 31 日までの 10 年間を学習用，それ以降の 1 か月間を検証用とする（図??参照）.

よって，この時点におけるデータセットのレコード数は，$N = 11419278$ であり，式 2.14 における不均衡度 r は，$r \approx 0.0023$ である．そこで，学習用データについて，ランダムサンダーサンプリングを行い，$N = 51065$，$r \approx 0.477$ となった.

予測モデルの精度検証

検証用のデータを用いて，作成した犯罪発生予測モデルの精度を検証した結果を表??に，同地域における複数日の予測結果を図??に示す．犯罪が発生したグリッドセル，発生しなかったグリッドセルともに，正しく予測した確率（正解率）は約 0.970 であったが，発生したグリッドセルを，発生すると予測した確率（再現率）は約 0.318，発生すると予測したグリッドセルで，実際に発生した確率（適合率）は約 0.018 であった．すなわち，犯罪が発生しないグリッドセルは比較的正しく予測できているものの，犯罪が発生しているグリッドセルについては，それに多少のランダム性を持っていたとしても，実用的な精度であるとは到底いえない結果となった.

この理由として，図??で分かるように，この 2 日間で犯罪が発生すると予測したグリッドセルが変化していない．表??のうち，灰色で着色した説明変数は，Boruta によって選択されたものであることを示しているが，これから分かるように，1 日ごとに変化する説明変数のうち，選択されたものは「過去 1 か月間の犯罪発生件数」のみであった．このため，そのような静的データに対して，動的データが予測値に寄与する絶対量が小さくなり，この予測モデルは，1 日ごとに予測値が変化しにくい可能性が考えられる.

短期的リスク，特に近接反復という犯罪の特性は，大きな犯罪発生の要因となり得る．そのため，静的データと動的データと分けて，それぞれに対して予測モデルを構築し，前者の予測モデルの予測値に対して重みづけを行うことによって，1 日ごとに予測値を大きく変化させることによって，予測精度が改善する可能性がある.

また，図??に示している地域に含まれるグリッドセルは約 550 個であり，実際に犯罪が発生したグリッドセルは 3〜5 個である．すなわち，その割合は 0.01 を下回る．本研究では，適切なアプローチを行い，不均衡なデータであっても，時空間的に解像度の大きい予測を行うことを目指したが，今回の犯罪発生データは，その限界を超えており，それでもなお精度の向上が見込めなかった可能性がある.

そのため，この改善案として，犯罪が発生する例を異常な例として，異常検知問題として取り扱うことが挙げられる．異常検知問題とは，検知したい異常な例がかなり少ないか，まったくないときに，正常な例のみを用いて，異常な例か正常な例かを判断することである．異常検知問題を取り扱うために用いるアルゴリズムは，異常な例を必要としないため，極端な不均衡，もしくは，まったく例がないデータを前提としていることである．そのため，犯罪が発生する例を異常な例として，異常検知問題として予測を行うことで，精度の向上が期待できるだろう.

なお，本研究では，データセットとして使用した，または可視化された説明変数を「要因」と仮定し分析を行ったが，あくまでも予測値に対する説明変数の「関係性」を示しており，実際に因果関係を検証するためには，因果推論などの手法を用いる必要があることに注意する必要がある．さらに，解釈された結果は，予測モデルの精度に左右されるため，その精度が大きいことが前提となっていることにも留意すべきである.

しかしながら，単純にその場所で過去に発生した犯罪の件数を蓄積し,「ここは犯罪が発生しやすい」と判断するだけではなく，予測値に対する各説明変数の貢献度を算出し，どのような要素が犯罪の発生に寄与しているのか，または寄与していないのか，その傾向を可視化することで，上記のような新たな知見を得られる可能性があることは，本研究における有意性のひとつと言えるだろう．一方で，可視化される要因の精度も少なからず考慮しなければならない．そこで，今後の課題として，予測モデルを介さず，実際のデータから各説明変数の貢献度を算出し，可視化することが挙げられるだろう．

§ 2.2　データクリーニングによる前処理

機械学習における分類問題で，学習に用いるデータセットが，そのクラスの比率に極端な偏りが生じているものを，一般に「不均衡データ」という．

不均衡なデータが予測精度に及ぼす影響

実世界で観測されるデータは，そのクラス比率が不均衡になっているものが多い．分かりやすい例として，医療データがある．身体検査のデータから特定の疾患を発病しているかどうかを出力するモデルを作成しようとするとき，健常者のデータは比較的容易に収集できるが，それと比較して，その患者のデータは，健常者よりも母数が小さく，収集できるデータ数が少なくなってしまう．

しかしながら，一般的な機械学習アルゴリズムは，それぞれのクラスのデータ数が同一であることを仮定していることが多く，不均衡な学習データを用いて機械学習を行うと，少数派に対する分類精度が著しく低下する [?]．この理由を，線形判別分析（Linear Discriminant Analysis: LDA）を例に説明する．LDA は，2 つのクラスを，最も識別できる直線（識別境界）で分離する手法である．多次元であるときを考慮し，サンプル $\boldsymbol{x}_n \in \mathbb{R}^M$ を，識別境界に直行している軸 $\boldsymbol{w}$ で線形変換した

$$y_n = \boldsymbol{w}^\top \boldsymbol{x}_n \tag{2.10}$$

を用いて，クラスを分類する．このとき，目的関数 $J(\boldsymbol{w})$ について，

$$\text{maximize} \quad J(\boldsymbol{w}) = \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}} \tag{2.11}$$

となる射影軸 $\boldsymbol{w}$ を求める．なお，クラス間変動行列 $\boldsymbol{S}_B$ は,

$$\boldsymbol{S}_B = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^\top \tag{2.12}$$

であり，クラス内変動行列 $\boldsymbol{S}_W$ は,

$$\boldsymbol{S}_W = \sum_{n \in C_1} (\boldsymbol{x}_n - \boldsymbol{m}_1)(\boldsymbol{x}_n - \boldsymbol{m}_1)^\top + \sum_{n \in C_2} (\boldsymbol{x}_n - \boldsymbol{m}_2)(\boldsymbol{x}_n - \boldsymbol{m}_2)^\top \tag{2.13}$$

である．いま，クラス C_1, C_2 の平均 $\boldsymbol{m}_1, \boldsymbol{m}_2$ は変化せずに，C_2 のサンプル数が十分小さいとき，式 2.13 の第 2 項は，第 1 項と比べて非常に小さくなる．したがって，射影軸 $\boldsymbol{w}$ は,

ほとんどクラス C_1 の変動のみを考慮することとなり，クラス分類の精度は小さくなる（図**??**参照）．このような現象は，LDA に限ったことではない．

多数クラス C^{maj} のサンプル数を N^{maj}，少数クラス C^{min} のサンプル数を N^{min} とすると，データの不均衡度 r は

$$r = \frac{N^{min}}{N^{maj} + N^{min}} \tag{2.14}$$

と定義できる．$r < 0.2$ であると，そのデータは十分不均衡であり，機械学習に用いるときは，何らかの工夫が必要である [?]．以降，不均衡データに対するアプローチとして，サンプリングとアンサンブル学習を説明する．

サンプリング

サンプリングとは，多数クラス C^{maj} のデータを間引いたり，少数クラス C^{min} のデータを生成することで，式 2.14 における不均衡度 $r = 0.5$ に近づける手法である．

i. アンダーサンプリング

アンダーサンプリングとは，多数クラス C^{maj} のデータを間引く手法であり，選択型と生成型の 2 つのアプローチがある．

選択型は，既存のデータから間引くものを選択するアプローチであり，ランダムに選択するランダムアンダーサンプリング（Random Under Sampling: RUS），クラスをいくつかに分割して，それぞれからランダムに選択するクラスタ基準アンダーサンプリング，トメクリンク（距離が近い C^{maj} と C^{min} のデータのペア）の多数クラス C^{maj} 側のデータを間引くアンダーサンプリングがある．

生成型は，多数クラス C^{maj} の既存のデータをそのまま使用せずに，新たなデータを少数クラス C^{min} と同数生成するアンダーサンプリングである．新たなデータの生成には，k-平均法が用いられることが多い．

ii. オーバーサンプリング

オーバーサンプリングとは，少数クラス C^{min} のデータを新たに生成する手法であり，そのアルゴリズムは，さまざまなものが提案されている．

Synthetic Minority Oversampling Technique（SMOTE）は，注目している i 番目の少数クラス C^{min} のデータを $\boldsymbol{x}_i^{min}$ として，$\boldsymbol{x}_i^{min}$ と k 番目までに距離が近い k 個の $\boldsymbol{x} \in C^{min}$ を取り出す．これを，k-近傍サンプル $\boldsymbol{x}_k^{min}$ と呼ぶ．$\boldsymbol{x}_k^{min}$ のなかからランダムにひとつ選択し，これを $\boldsymbol{z}$ とする．$\boldsymbol{x}_i^{min}$ と $\boldsymbol{z}$ の内挿となる点に，$\boldsymbol{x}_i^{min}$ の新たなデータ $\boldsymbol{y}_i$

$$\boldsymbol{y}_i = \boldsymbol{x}_i^{min} + r(\boldsymbol{x}_i^{min} - \boldsymbol{z}) \tag{2.15}$$

を生成する．これを，すべての $\boldsymbol{x}_i^{min} \in C^{min}$ に対して行い，生成された $\boldsymbol{y}_i$ を学習データに追加する．ほかにも，SMOTE から派生した Adaptive Synthetic（ADASYN）や，ボーダーライン SMOTE などの手法が存在する．

アンサンブル学習

アンサンブル学習とは，複数のモデルを学習し，それらをもとに最終的な出力を決定する手法である．最終的な出力を決定するために学習させる複数のモデルを弱学習器と呼び，それぞれをどのように学習させるかによって，バギングとブースティングの 2 つに分けることができる．

バギングとは，ブートストラップ法（データからランダムに一部のサンプルを取り出すことを繰り返し，ひとつのデータセットから複数のデータセットを生成する手法）を行い，それぞれのデータセットで弱学習器を作成し，それぞれの出力をもとに，最終的な出力を決定する手法である．

ブースティングとは，n 個の弱学習器をそれぞれ直列に学習させ，それぞれの出力をもとに最終的な出力を決定するが，k 番目（ただし，$k = 1, 2, 3 \cdots, n-1$）に学習させた弱分類器の出力を，$k+1$ 番目に学習させる弱分類器に応用する手法である．

機械学習，たとえば，近年急速に注目されている深層ニューラルネットワークといったアルゴリズムは，複雑かつ非線形な性質であってもモデリングすることができる．すなわち，より予測精度の大きいモデルを作成することができる．しかしながら，一般にモデルの精度が大きくなるほど，その解釈性は小さくなる性質がある．

予測モデルを解釈する重要性

近年，深層ニューラルネットワークなどの表現力の高いモデルを作成できるアルゴリズムの登場により，多くの分野で機械学習が活用されるようになってきた．医療分野では，網膜の画像から，糖尿病網膜症かどうかを診断するシステムが，米国で認可されている [?]．そのような責任が大きい判断の場合は，予測精度が大きいことはもちろん，なぜその予測値を出力したのか，その根拠も人間が知る必要がある．そのため，総務省が示している「AI 利活用原則」や，EU が施行している「一般データ保護規則（General Data Protection Regulation: GDPR）」においても，機械学習モデルの説明責任について言及しており，予測モデルを解釈することは，国際的に重要視されているといえるだろう．

予測精度と解釈性のトレードオフ

以下のような線形回帰モデルを考える．

$$f(X_1, X_2) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \tag{2.16}$$

このとき，X_1 が 1 だけ大きくなると，$f(X_1, X_2)$ は β_1 倍だけ大きくなることが明示的に分かる．このように，線形回帰モデルは，目的変数と説明変数とのあいだに単純な関係を仮定しており，モデルに対する透明性が高いと言える．これを，一般に「解釈性が高い」と言う．

一方で，比較的近年に発表されたアルゴリズム，例えば深層ニューラルネットワークやランダムフォレストなどは，目的変数と説明変数とのあいだに線形性などの仮定を置いていない．よって，より複雑な関係をモデリングできるようになり，一般に線形回帰モデルよりも予測精度は大きくなりやすい．しかしながら，線形回帰モデルと違い，その複雑さから，なぜその予測値を出力するのかを理解することができず，その中身はブラックボックスとなりやすい．これを，一般に「解釈性が低い」と言う．

予測モデルを解釈する主な手法

機械学習によって作成されたモデルに対して，何らかの解釈を与える手法はいくつか存在するが，特に有用なものとして，以下の 4 つが挙げられるだろう．

- Permutation Feature Importance（PFI）
- Partical Dependence（PD）
- Individual Conditional Expectaion（ICE）
- Shapley Additive Explanations（SHAP）

それぞれは何を解釈できるのかが異なり，用途によって使い分ける必要がある（図**??**参照）．例えば，モデル全体の傾向など，マクロな視点から解釈する場合は PFI を，出力されたひとつの予測値に対する根拠など，ミクロな視点を知りたい場合は ICE を用いるべきだろう．本研究では，ミクロな視点から解釈できるものの，マクロな視点からの解釈も可能な SHAP [**?**] を用いることとする．

SHAP

$\mathbf{X} = (X_1, \ldots, X_J)$ を説明変数とする学習済みのモデルを $\hat{f}(\mathbf{X})$ とする．インスタンス i の説明変数が $\mathbf{x_i} = (x_{i,1}, \ldots, x_{i,J})$ とすると，インスタンス i の予測値は $\hat{f}(\mathbf{x}_i)$ である．ここで，予測の期待値を $\mathbb{E}\left[\hat{f}(\mathbf{X})\right]$，インスタンス i の説明変数 $x_{i,j}$ の貢献度 $\phi_{i,j}$ としたとき，

$$\hat{f}(\mathbf{X}_i) - \mathbb{E}\left[\hat{f}(\mathbf{X})\right] = \sum_{j=1}^{J} \phi_{i,j} \tag{2.17}$$

のように，期待値からの差分を貢献度の総和で表現できるように，貢献度を分解することが，SHAP の基本的な考え方である．線形モデルであれば，比較的容易に分解することができるが，非線形モデルではこのままでは難しい．そのため，SHAP では，協力ゲーム理論の Shapley 値の考え方を用いて，貢献度を分解する．

ここで，協力ゲーム理論のひとつであるアルバイトゲームを説明する．アルバイトの参加者として，A，B，C の 3 つのプレイヤーを仮定し，アルバイトの参加者とそのときに得られる報酬には，表**??**のような関係があるとする．

A・B・C の 3 プレイヤーが参加したときの報酬は 24 である．より貢献度が大きいプレイヤーに，より多くの報酬を配分するとすれば，その貢献度はどのように算出すべきだろうか．ここで，限界貢献度という概念を導入する．限界貢献度とは，あるプレイヤーが参加したときの報酬と，参加する直前の報酬との差を表す．例えば，B・C がすでに参加しているときに A が参加した場合の限界貢献度は，$24 - 10 = 14$ である．しかし，各プレイヤーがどのような順序で参加するかにより，限界貢献度は異なる．例えば，A の限界貢献度について，A，B，C という順番で参加したときは 6 であるが，B，C，A という順序で参加したときは 14 である．

この影響を解消するため，考えられるすべての順序で限界貢献度を算出し，その平均を求めることにする．例えば，A の限界貢献度の平均値は，$(6+6+16+14+13+14)/6 = 11.5$ である．この限界貢献度の平均値を Shapley 値といい，これをもとに報酬を分配する．一般に，J つのプレイヤーが存在するとき，プレイヤー j の Shapley 値 ϕ_j は以下のように算出される．

$$\phi_j = \frac{1}{|\mathcal{J}|!} \sum_{\mathcal{S} \subseteq \mathcal{J} \setminus \{j\}} (|\mathcal{S}|!(|\mathcal{J}| - |\mathcal{S}| - 1)!)(v(\mathcal{S} \cup \{j\}) - v(\mathcal{S})) \tag{2.18}$$

SHAP は，この Shapley 値の考え方を機械学習のモデルに適用している．例えば，説明変数が X_1, X_2 であるモデルにおいて，インスタンス i の予測値 $v(\{1,2\})$ の，説明変数を $x_{i,1}, x_{i,2}$ とすると，

$$v(\{1,2\}) = \hat{f}(x_{i,1}, x_{i,2}) \tag{2.19}$$

である．また，$x_{i,1}$ と $x_{i,2}$ のいずれも未知の場合は，予測値の期待値とし，

$$v(\emptyset) = \mathbb{E}\left[\hat{f}(X_1, X_2)\right] \tag{2.20}$$

である．では，$x_{i,1}$ は既知であり，$x_{i,2}$ は不明であるときの予測値 $v(\{1\})$ は，後者について周辺化を行い，

$$v(\{1\}) = \mathbb{E}\left[\hat{f}(x_{i,1}, X_2)\right] = \int \hat{f}(x_{i,1}, x_2) p(x_2) dx_2 \tag{2.21}$$

である．よって，$x_{i,1}$，$x_{i,2}$ という順序で説明変数が判明したときの，それぞれ時点における限界貢献値 $\Delta_{i,1}, \Delta_{i,2}$ は，

$$\Delta_{i,1} = \mathbb{E}\left[\hat{f}(x_{i,1}, X_2)\right] - \mathbb{E}\left[\hat{f}(X_1, X_2)\right] \tag{2.22}$$

$$\Delta_{i,2} = \mathbb{E}\left[\hat{f}(x_{i,1}, x_{i,2})\right] - \mathbb{E}\left[\hat{f}(x_{i,1}, X_2)\right] \tag{2.23}$$

である．Shapley 値と同様に，考え得るすべての順番で算出し，それらを平均する．すなわち，説明変数 $x_{i,1}, x_{i,2}$ について，その平均値 $\phi_{i,1}, \phi_{i,2}$ は，

$$\phi_{i,1} = \frac{1}{2}\left(\left(\mathbb{E}\left[\hat{f}(x_{i,1}, X_2)\right] - \mathbb{E}\left[\hat{f}(X_1, X_2)\right]\right) + \left(\hat{f}(x_{i,1}, x_{i,2}) - \mathbb{E}\left[\hat{f}(X_1, x_{i,2})\right]\right)\right) \tag{2.24}$$

$$\phi_{i,2} = \frac{1}{2}\left(\left(\hat{f}(x_{i,1}, x_{i,2}) - \mathbb{E}\left[\hat{f}(x_{i,1}, X_2)\right]\right) + \left(\mathbb{E}\left[\hat{f}(X_1, x_{1,2})\right] - \mathbb{E}\left[\hat{f}(X_2, X_2)\right]\right)\right) \tag{2.25}$$

である．このとき，$\phi_{i,1}, \phi_{i,2}$ は，協力ゲーム理論においては Shapley 値と呼ぶが，SHAP においては SHAP 値と呼ぶ．式 3.17 と式 3.18 より，$\phi_{i,1}$ と $\phi_{i,2}$ を足すと，

$$\phi_{i,1} + \phi_{i,2} = \hat{f}(x_{i,1} + x_{i,2}) - \mathbb{E}\left[\hat{f}(X_1, X_2)\right] \tag{2.26}$$

であり，式 3.10 と同様に，インスタンス i の予測値と，予測の期待値との差分になっていることが分かる．

SHAP の有用性

SHAP は，ブラックボックスなモデルであっても，なぜその予測値を出力したのか，説明変数ごとにその貢献度を出力できる．その貢献度がどれほど的確に推定できているかを検証したところ，回帰問題について，既存の解釈手法より正しく貢献度が推定できることが示された [?].

平均 0，標準偏差 30 の正規分布 $N(0, 30^2)$ に従う 5 つの説明変数 x_1, x_2, x_3, x_4, x_5 と，平均 0，標準偏差 10 の正規分布 $N(0, 10^2)$ に従うノイズ b を生成し，式 3.20 および式 3.21 で教師データをそれぞれ 10000 行作成する．式 3.21 で作成した教師データについては，x_2 だけ十の位で四捨五入し，ほかの説明変数とデータの解像度が異なるケースを再現する．

$$y = 15x_1 + 10x_2 + 5x_3 + x_4 + 0.3x_5 + b \tag{2.27}$$

$$y = 10x_1 + 10x_2 + 5x_3 + x_4 + 0.3x_5 + b \tag{2.28}$$

それぞれの教師データを，決定木をベースとした機械学習アルゴリズムである XGBoost でモデルを学習した．そのモデルを SHAP で解釈した結果と，従来手法である Future Importance で解釈した結果を比較する．式 3.20 による教師データの結果を図 3.6，式 3.21 の結果を図 3.7 に示す．なお，Future Importance は，ある説明変数が予測精度をどれだけ向上させたかを，その説明変数の「重要度」として示した値である．図中の赤字で書かれた値は，x_4 の大きさを 1 としたときの各説明変数の比率であり，Future Importance と比較しても，おおむね正確に貢献度を推定できていることが分かる．

Boruta による説明変数の選択

本システムでは，ユーザが自由に説明変数を選択することができるが，過度に説明変数の数が大きかったり，目的変数と相関がない説明変数があると，過学習などによって，かえって予測精度が低下してしまう可能性がある．そこで，本システムでは，犯罪発生予測モデルを作成する前に，Boruta [?] と呼ばれるアルゴリズムを用いて，適切な説明変数を選択する．

Boruta のアルゴリズムは，以下のような流れである．

1 もともとのテーブルをコピーし，各列をシャッフルする．もとのテーブルの説明変数を Original features，シャッフルした説明変数を Shadow features と呼ぶことにする．このとき，Shadow features は，なんら目的変数に寄与しないはずである．

2 Original features と Shadow features を結合し，ランダムフォレストでモデルを作成する．

3 そのモデルにおいて，それぞれの説明変数の重要度を算出し，Shadow features における最大値よりも大きい Original features を見つける（hit する）.

4 ランダムフォレストの性質により，モデルを作成するごとに重要度は変化するため，1〜3 を n 回くり返す.

5 各 Original features について，Shadow features の重要度と同じことを帰無仮説，より大きい・より小さいことを対立仮説とし，hit した合計を検定統計量 T，$p = 0.5$ としたときの二項分布を用いて検定を行う.

検定の結果，説明変数が，Confirmed，Tentative，Rejected の 3 つに分類される．本システムでは，Confirmed，Tentative の 2 つを，説明変数として用いることとする.

§ 2.3　ネットワーク構造における入力と出力

因果探索とは，観測データを用いて，そのデータ群の因果グラフ（複数の観測データにおいて，それぞれの値がお互いに及ぼしあっている影響の度合いを構造的に示したもの）を導出するための教師なし学習のことである.

また，類似する手法として因果推論が挙げられるが，因果推論では因果関係の向きが既知である場合にその因果関係が本当に有意であるのかをデータから分析する手法であるのに対し，因果探索は因果関係が不明かつ因果関係の向きも不明であるデータ群に対して，それらの間に因果関係が成立するかを導く手法である.

例えば,「ある小売店 A でアイスクリームの安売りを行った際にアイスクリームの売り上げが向上した．また，同日の小売店 A の来客数は前日より 100 人多かった」というケースがあったとする．このとき，アイスクリームの安売りを行ったことが売り上げの向上につながったかどうかを調べるのが因果推論である．これに対して，アイスクリームが安かったから来客数が増加したのか，来客数が多かったためにアイスクリームの売り上げが向上したのかという因果の方向性も含めて分析を行うのが因果探索である.

このような特徴を持つため，因果探索は適用されるデータの分野に対しての制約が少なく，様々な分野のデータに適用することができる．それゆえ，因果探索を用いた応用研究も盛んにおこなわれており，疫学，経済学，神経科学，化学，医学をはじめとした幅広い分野のほか土木計画学 [?] の研究でも用いられている.

LiNGAM

近年，因果探索の手法における研究が活発化したことで，因果探索における様々なモデルが提唱されている．代表的なものとしては独立主成分分析の手法を用いたセミパラメトリックなもので，非時系列データに対しても適用可能な LiNGAM が挙げられる．LiNGAM とは，一般的に以下の式（2.29）のように定式化され,

$$x_i = \sum_{i \neq j} b_{ij} x_j + e_i \qquad i, j = 1, \cdots, p \tag{2.29}$$

以下のような仮定を用いることで因果グラフを導出する手法である [?].

1. 外生変数と内生変数をつなぐ関数は線形関数とする.（内生変数とは実際に観測されている変数，外生変数とは内生変数以外の変数で内生変数のそれぞれに関する未知の値である.）
2. 外生変数の分布は非ガウス連続分布とする.
3. 因果グラフは非巡回とする.
4. 外生変数は互いに独立とする.

ここで，図??のような内生変数が 4 つの因果グラフがあると考える．このとき，仮定 3 があることによって 4 つの内生変数のうち，どの変数からも因果的影響を受けない変数が少なくとも 1 つ必ず同定される.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_4 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_4 \\ x_3 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_4 \\ e_3 \end{bmatrix} \tag{2.30}$$

つまり，式（2.30）のように因果的影響を受ける場合にその値，受けない場合には 0 を入れたパス係数行列を考えると必ず右肩が逆三角形に全て 0 になる行列となる．そのため，式（2.30）における x_1 のように全てのパスに対する係数が 0 となる内生変数を因果グラフから除外し，再度パス係数行列を求めるという操作を繰り返すことによって未知である因果グラフを同定することが可能になる.

また，上記の同定法を成立させるにあたって，仮定 4 がなくてはならない．前述のとおり，LiNGAM における因果関係の同定では内生変数同士の因果関係のみに着目して因果グラフの最も外側に位置する内生変数を順に除外する方法をとるため，内生変数同士の間に成立する因果関係以外の因果関係が内生変数間に発生してはならない.

ここで，もし外生変数同士が独立ではなければ外生変数同士の間に因果関係が生じてしまい，図??の例に見られるように内生変数同士がそれぞれの内生変数に関わる外生変数同士の因果関係を介在として内生変数間に存在しない新たな因果関係を持ってしまう．このような場合には前述のような因果関係の同定法が成り立たなくなるため，LiNGAM における仮説 4 は必ず必要となる.

Direct-LiNGAM

前述のようなアルゴリズムによって内生変数間の因果関係を推定する LiNGAM であるが，推定時の計算方法の違いによって現在までにいくつかのアプローチが提唱されている．代表的な例として，独立成分分析によるアプローチである ICA-LiNGAM や回帰分析と独立性評価によるアプローチである Direct-LiNGAM などが挙げられる．その中でも，本研究で取り扱う Direct-LiNGAM に関する解説を行う．Direct-LiNGAM によるアプローチの基本的な考え方は

- 観測変数群から 2 変数を取り出しそれらの変数間に成り立つ因果関係を同定すことを繰り返して観測変数群全体における因果の始まりとなる変数を探す.
- その変数を観測変数群から除外し，残った変数のみで再度，観測変数群を形成する.

という 2 つの操作を観測変数群に属する変数が存在しなくなるまで繰り返すことによって元の観測変数群の因果グラフを同定するというものである．例として，観測変数群から 2 変数 x_1, x_2 を取り出し，以下の構造方程式モデルが背後にあるものと想定する．

$$\begin{cases} x_1 = e_1 \\ x_2 = b_{21}x_1 + e_2 \end{cases} \tag{2.31}$$

ここで e_1, e_2 は互いに独立かつ非ガウス分布に従い，$b_{21} \neq 0$ とする．これについて単回帰分析を行うことによって，因果順序の同定を行う．まず x_2 を目的変数，x_1 を説明変数として回帰する場合を考える．この場合元の構造方程式モデルの第 2 式がそのまま成り立つことになる．そのため，回帰残差は e_2 となりこれは $x_1 = e_1$ と独立となる．一方，x_1 を目的変数，x_2 を説明変数とした場合，回帰残差 r_1 は

$$r_1 = \left\{1 - \frac{b_{21}cov(x_1, x_2)}{var(x_2)}\right\} e_1 - \frac{b_{21}var(x_1)}{var(x_2)}e_2 \tag{2.32}$$

となり e_2 の項が出てくる．一方冒頭の構造方程式に戻ると，x_2 は式として e_2 を含むので，この回帰残差と説明変数 x_2 とは従属する．この従属性の成立に関しては前述の仮定 2 にて示した「外生変数が非ガウス分布とする」というきまりに基づいており，以下に示すダルモア・スキットビッチの定理を用いている．

ダルモア・スキットビッチの定理

2 つの確率変数 y_1, y_2 が，互いに独立な確率変数 $s_i (i = 1, ..., q)$ の線形和で下記のように表されているとする．この時，もし y_1, y_2 が独立なら，$\alpha_j \beta_j \neq 0$ となるような変数 s_j はガウス分布に従う

$$y_1 = \sum_{i=1}^{q} \alpha_i s_i \tag{2.33}$$

$$y_2 = \sum_{i=1}^{q} \beta_i s_i \tag{2.34}$$

上記の考察から，両方のパターンで回帰分析を行い残差と説明変数の独立性を判定することで因果の向きを推定することが可能となる．なお独立性の評価には相互情報量という量を用いる．この量が 0 となるときに独立であると判定するが，実際には推定誤差があり正確には 0 にはならないため，相互情報量が 0 に近い方を独立とみなして因果の順序を決定する．

計量経済学は経済現象を数理・統計的手法を用いて分析・モデル化する学問分野である．計量経済モデルとは，この計量経済学の考え方に基づき，経済のメカニズムや変数間の関係性を数学的な式や方程式系によって定式化したものを指す．

計量経済モデルには大きく分けて，経済構造そのものを表現する構造形モデルと，特定の経済変数間の総合的な相関関係を表現する縮小形モデルがある．前者は経済主体の意思決定や市場の働きなど，経済の根本メカニズムを体系的にモデル化する点に特徴がある．後者は実証分析に基づいて，複数経済変数間の相関関係を簡潔に定式化することにが特に重要である．

これらのモデルは消費者や企業の最適化行動の仮定、市場の均衡条件の設定、複数の内生変数や外生変数の導入などの方式で構築される．その際、個々の変数の挙動はパラメータによって定量的に規定される．これらパラメータは実際の時系列データや断面データを用いた回帰分析などの計量的手法に基づいて推定されることが多く，可能な限り実態を反映するよう設定される．

構築されたモデルは政策変更がマクロ経済・ミクロ経済に与える影響の定量的分析や経済変数の将来予測に利用される．特に政策当局にとって異なる政策オプションを比較検討する上での有用性が高いとされている．ここでは現在使われている計量経済モデルの例について紹介する．

VAR モデル

VAR モデルは多変量時系列データを分析する統計的手法であり，柔軟性や汎用性から将来予測の分野などで特に注目を集めている．この手法では複数の変数が同時に相互作用する複雑なシステムを捉えることができる．経済の分野においては，VAR モデルは異なる経済指標や変数間の相互作用を捉え，経済の動向や政策の影響を理解するために頻繁に活用されている．為替変動や売上高成長率の予測 [16] などの研究があり，経済の分野でも幅広く利用されていることがわかる．
VAR モデルの中でも一般的な VAR モデル (p) の構造は以下のようになる．

$$Y_t = A_1 Y_{t-1} + A_2 Y_{t-2} + \ldots + A_p Y_{t-p} + \varepsilon_t \tag{2.35}$$

$$\begin{aligned}
\begin{bmatrix} Y_{1,t} \\ Y_{2,t} \\ \vdots \\ Y_{k,t} \end{bmatrix} &= \begin{bmatrix} a_{11,1} & a_{12,1} & \ldots & a_{1k,1} \\ a_{21,1} & a_{22,1} & \ldots & a_{2k,1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1,1} & a_{k2,1} & \ldots & a_{kk,1} \end{bmatrix} \begin{bmatrix} Y_{1,t-1} \\ Y_{2,t-1} \\ \vdots \\ Y_{k,t-1} \end{bmatrix} \\
&+ \begin{bmatrix} a_{11,2} & a_{12,2} & \ldots & a_{1k,2} \\ a_{21,2} & a_{22,2} & \ldots & a_{2k,2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1,2} & a_{k2,2} & \ldots & a_{kk,2} \end{bmatrix} \begin{bmatrix} Y_{1,t-2} \\ Y_{2,t-2} \\ \vdots \\ Y_{k,t-2} \end{bmatrix} + \ldots \\
&+ \begin{bmatrix} a_{11,p} & a_{12,p} & \ldots & a_{1k,p} \\ a_{21,p} & a_{22,p} & \ldots & a_{2k,p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1,p} & a_{k2,p} & \ldots & a_{kk,p} \end{bmatrix} \begin{bmatrix} Y_{1,t-p} \\ Y_{2,t-p} \\ \vdots \\ Y_{k,t-p} \end{bmatrix} + \begin{bmatrix} u_{1,t} \\ u_{2,t} \\ \vdots \\ u_{k,t} \end{bmatrix}
\end{aligned}$$

近年では VAR モデルは単体として使われることは少なく，構造 VAR モデルなど拡張されて使われることが多いが，未だに最も基本的な多変量時系列データを分析するための手法として注目されている．経済分析に特に頻繁に用いられるのは構造 VAR である．構造 VAR モデルの構造については以下の式にして示す．

$$Y_t = A_1 Y_{t-1} + A_2 Y_{t-2} + \ddots + A_p Y_{t-p} + BX_t + \varepsilon_t \tag{2.36}$$

構造 VAR モデルでは VAR モデルに外部説明変数を取り入れることで経済ショックが内政変数にもたらす動学的な影響をインパルス応答などで数量的に表すことができる．また変数間の因果関係の方向性や即時効果の程度を知ることができる．これにより，金融政策の効果検証や政策分析への研究に適用されることがあり，構造 VAR モデルを用いた日本経済の資産蓄積，所得分配，負債の動態分析などの研究が行われている [17].

VAR-LiNGAM

因果探索も計量経済モデル計量経済モデルの一種として LiNGAM によるものがある．VAR-LiNGAM は因果探索手法の一つであり，時系列データに対する因果関係の推定を行うためのモデル化手法である．VAR-LiNGAM は 2010 年に提唱され [18]，様々な分野で活用されている．VAR-LiNGAM は VAR モデルと Linear Non-Gaussian Acyclic Model (LiNGAM) を組み合わせたものであり，図のような形になる．LiNGAM は因果探索のセミパラメトリックなアプローチの手法であり定式化は以下のようになる．

$$x_i = \sum_{i \neq j} b_{ij} x_j + e_i \qquad (i = 1, \ldots, p) \tag{2.37}$$

そしてこれは行列表現で以下のように表すことができる．

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pp} \end{bmatrix}, \quad E = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_p \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pp} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_p \end{bmatrix}$$

VAR-LiNGAM ではこれに VAR モデルを加える．VAR-LiNGAM のイメージは図に示す．VARLiNGAM における仮定として VAR-LiNGAM の一般的な定式化としては以下のようになる

$$x(t) = \sum_{\tau=0} b_{ij} B_r x(t-\tau) + e(t) \tag{2.38}$$

これは VAR モデルと似た形になっているが，時間差を表す τ が 0 から始まっている．B_0 は LiNGAM における B のように現時点での多変数からの因果を表す係数であり，$B_\tau (\tau = 1, \ldots, k)$ は過去時点の $x(t-\tau)$ からの直接的な影響を表す係数であり，これにより現在のベクトル変数の値 $x(t)$ を現時点の他の変数だけでなく過去時点のベクトル変数も用いて推定することができる．

第3章

複数の評価基準からの総合評価

§ 3.1　データ包絡分析の種類と効率の評価

DEA とは，ある分野における組織の集合において，対象の組織の業績を評価するために生み出されたノンパラメトリックな手法の 1 つで 1978 年に Charnes，Cooper と Rhodes によって提唱された．ここでいう組織とは，その活動においていくつかの種類の入力（投入）をいくつかの出力（産出）に変換することに携わる生産体（Decision Making Unit: DMU）のことである．DEA での分析の利点の一つとして，複数の入力・出力があるデータを扱うことができることが挙げられる．

DMU における活動の例として，半導体生産工場における毎期の生産活動や，様々な市場に対する宣伝や販売活動等が挙げられる．DEA はこれらの活動に対して，自らを基準とした集合内の他の DMU との比較によって，その業績を評価することが可能である．

1978 年に提唱されて以来，DEA に関する研究，応用は世界中の研究機関で積極的に行われており [?]，CCR，BCC などの基本的なモデルをはじめとして，現在までに様々なモデルが発表されている．本節では，多数存在する DEA のモデルの中で最も基本的であり，本研究の分析にも用いる CCR モデルについて，そのアルゴリズムを解説する．

また，本研究で DEA におけるデータの分析を行う際に前述のとおり多数存在するモデルの中から CCR モデルによる分析を選択した理由については次のようなことが挙げられる．CCR モデルは最もはじめに考えられた DEA のモデルであり，原点からの距離の比を用いて値の最小を定める「比率尺度データ」を対象とした分析法である．

そして，それに次いで考えられたモデルは CCR モデルに各種制約を追加することで「感覚尺度データ」など，対象のデータに適用できるように拡張されたものである．ここで，本研究で分析に用いるオープンデータを考えると，すべてが「比率尺度データ」であったため，本研究では CCR モデルを適用することとした．

CCR モデル

DEA における DMU の評価法の基本的な考え方は「どれだけ少ない入力を用いてどれだけ多くの出力を生み出しているか」である．そのため，DEA における評価値は対象の DMU における各入出力に対して重みをつけたうえで出力の総和を入力の総和で割ることによってもとめられる．

また，評価値を算出する際に各入出力に対して付与される重みには他の DMU の入出力に基づく制約式が設けられており，それらに基づいて入力・出力の重みを最適化する線形

計画問題を解くことによってDEAにおける評価値を算出することができる．CCRモデルにおける制約条件は以下の二つである．

- すべてのDMUに対する評価値はいずれも1を超えない．
- 入力・出力に対する重みはいずれも0以上である．

これらをもとにCCRモデルを線形計画問題として定式化すると以下のようになる [?].

<CCRモデルの主問題 >

$$\text{maximize} \quad \frac{u^T y_o}{v^T x_o} = z \tag{3.1}$$

$$\text{subject to} \quad -v^T X + u^T Y \leq 0 \tag{3.2}$$

$$u \geq 0 \tag{3.3}$$

$$v \geq 0 \tag{3.4}$$

ここで，DEAにおける分析が持つもう一つの利点を考える．それは，算出された各入力・出力に対するウェイトを用いることで，対象のDMUの入力・出力をどのように増減させれば評価値がより優れたものになるかを数学的に示すことができることである．このDMUにおける評価値をより良くするための入力・出力の値（本論文では以後，改善案と呼ぶ）を算出するためには，評価値における入力（分母）もしくは出力（分子）を1と仮定し，前述の線形計画問題を主問題とする双対問題を考える必要がある．

これらをもとに，入力に対する改善値を算出することを目的とした入力指向モデルと出力に対する改善値を算出することを目的とした出力指向モデルを定式化すると以下のようになる．

< 入力指向モデル >

$$\text{minimize} \quad w = \theta \tag{3.5}$$

$$\text{subject to} \quad Y\lambda \geq y_o \tag{3.6}$$

$$-X\lambda + x_o\theta \geq 0 \tag{3.7}$$

$$\lambda \geq 0 \tag{3.8}$$

< 出力指向モデル >

$$\text{maximize} \quad w = \eta \tag{3.9}$$

$$\text{subject to} \quad X\mu \leq x_o \tag{3.10}$$

$$-Y\mu + y_o\eta \leq 0 \tag{3.11}$$

$$\mu \geq 0 \tag{3.12}$$

ここで，入力指向モデルを取り上げて解説を行うと，その線形計画問題は対象のDMUにおける出力の大きさを維持しつつ，最小の入力を達成することができるDMUの集合（参

表 3.1: 各病院におけるパラメータ

病院	A	B	C	D	E	F	G
医師数	17	58	72	19	11	54	8
患者数	266	661	1,695	514	543	1,447	390
医師一人当たり患者数	15.6	11.4	23.5	27.1	49.4	26.8	48.8

照集合）をすべての DMU の中から探すという操作ととることができる．つまり，線形計画問題の解においてその重みが正の数となる DMU が参照集合に属し，その場合における効率的な DMU ととらえることができる．

これらのことから，参照集合とは対象の DMU が見本とすべき DMU の集合であり，参照数号に属する DMU におけるそれぞれの要素の値とその重みの大きさを用いることによって，対象の DMU に対する入力および出力の改善案を算出することができる．

参照集合における DMU の数を K，入力・出力の項目数をそれぞれ m，n とすると対象の DMU のある項目 i, j における入力・出力における改善案はそれぞれ次のようにしてもとめることができる．

< 入力改善案 >

$$\hat{x_i} = \sum_{k=1}^{K} x_{ik}\lambda_k \qquad i = 1, 2, \cdots m \tag{3.13}$$

< 出力改善案 >

$$\hat{y_j} = \sum_{k=1}^{K} y_{jk}\mu_k \qquad j = 1, 2, \cdots n \tag{3.14}$$

医療機関における DEA の例

DEA の一般的な適用例として病院における医師数と患者数という 1 入力-1 出力の場合を挙げる??．本例では医師数を入力，患者数を出力として各病院の運営効率を CCR モデルで評価する．また，単純化のために本例では少ない医師でより多くの患者を治療する病院こそが運営において効率的であるとする．

各病院における医師数，一日の平均患者数，医師一人が一日に治療する平均患者数を表 3.1，図 3.1 に示す．グラフにおける横軸が入力，縦軸が出力であり，太い実線が効率フロンティア，破線は回帰直線を表す．表 3.1 における医師一人当たりの患者数が多くなっている，つまり，効率的な運営を行っている病院 E，G がグラフにおいても効率フロンティア上に存在していることが分かる．

また，細い実線の様子から病院 F がより効率的に運営を行うためには医師数を減らす，患者数を増やす，またはその両方を同時に行うことが必要であることが分かり，詳細な値についてはグラフ上における点 F と参照集合の直線の距離で求められる．

太い直線と破線の差からも分かるように最も評価の高い組織を基準に各組織の評価を行うという点が組織の集合全体における平均像から各組織の評価を行う統計学における回帰分析を用いるアプローチと大きく異なる DEA の特徴である．

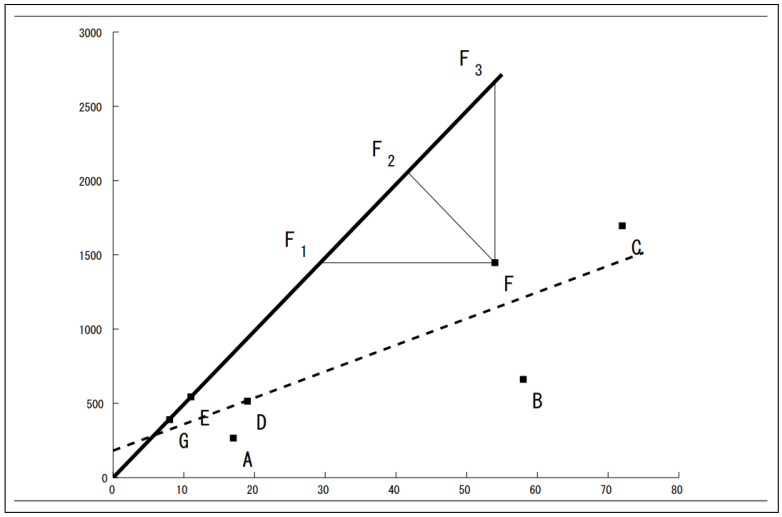

図 3.1: 適用例における CCR モデルのグラフ [?]

本研究における提案手法において，入力指向モデルおよび出力指向モデルによって算出された対象の市区町村における評価値と式（3.13），式（3.14）によって算出された入力および出力の改善案を提示することを政策に対する意思決定を支援する手法の 1 つとして用いることとする．

§ 3.2　データ包絡分析による改善策の導出

平均クロス効率値による評価では，DMU 全体として合理することはできないことについて述べた．本節では，クロス効率行列を使って効率評価を算出する際に，協力ゲームとして捉えることで DMU 全体が合意できるような仕組みについて述べる．そのため，まずはゲーム理論について述べる．

ゲーム理論とは，社会や経済，ビジネスなどのさまざまな問題を，そこに参加する個人や企業・政府をプレイヤーと見做して，どういう行動をとるのかを数理的に分析する理論である．こうした現実の問題をゲームと考えて，そのゲームのプレイヤーがどういう戦略を選択するかを分析することからゲーム理論と名付けられた．ゲーム理論は，数学者のジョン・フォン・ノイマンと経済学者のオスカー・モルゲンシュテルンによって発明された．ゲーム理論は，非協力ゲーム理論と協力ゲーム理論に分けられる．非協力ゲーム理論は，プレイヤーが目的達成のために独自に行動する状況を扱う理論で，協力ゲーム理論は，複数のプレイヤーがプレイヤー同士の協力 (連携) が許された状況を扱う理論である．

非協力ゲームでは，さらに展開形ゲームと戦略形ゲームに分けられる．戦略形ゲームは，ゲーム理論の基礎の形で，全てのプレイヤーが他のプレイヤーの行動を観察できずに同時に行動するゲームのことである．例として，じゃんけんが挙げられる．また展開形ゲームは，戦略形ゲームを拡張して時間と情報を含めた状況の中で，他のプレイヤーの行動を観察して自分の行動を選ぶことができるゲームである．展開型ゲームは，主に将棋などのボードゲームを例として上げることができる．また，展開型ゲームにおいては過去の戦略である行動選択を全プレイヤーが知ることができるゲームを完全情報ゲームといい，そうでな

いゲームを不完全情報ゲームという．

非協力ゲーム理論の中で特に有名なのが囚人のジレンマと呼ばれるモデルである．これは，2 人の囚人が意思疎通できない中で尋問された状況のゲームで，囚人がどの行動を選択するか明らかにするゲームである．同時に尋問がされると考えるので，このモデルは，戦略形の非協力ゲーム理論といえる．また，囚人に与えられた条件は，尋問に対して 2 人ともが自白した場合は 2 人の懲役が 5 年となり，1 人が自白もう 1 人が黙認した場合は自白したほうが無罪で黙認したほうが懲役 10 年となり，2 人ともが黙認した場合は懲役 2 年という条件である．

このモデルでは，囚人の都合は最も良い条件の無罪を取ることである．ただ一方で，2 人の囚人が無罪を取るために自白をしてしまうと懲役が 5 年となってしまう．この結果は，2 人ともが黙認するときの結果の懲役 2 年と比べると悪い結果である．こうしたお互い協力するほうが協力しないよりも良い結果が得られるにもかかわらず，片方が自分を優先して裏切ることで利益が得られる状況では，お互いは協力しなくなるジレンマを囚人のジレンマと呼ぶ．このような非協力ゲームに対する行動選択に対する解として，ナッシュ均衡や支配戦略均衡，被支配戦略逐次排除均衡，サブゲーム完全均衡といったものがある．

また，協力ゲームでは戦略形ゲームと連携形ゲームに分けられる．連携型ゲームは，特性関数型ゲーム [?] とも言われ，プレイヤーの連携を集合を使って表し，連携によって得られる利益が事前に与えられるため特性関数を作ることができるゲームとなっている．協力ゲームに対する解としては，安定集合，コア，交渉集合，仁，Shapley 値，カーネルなどがある．この中で，コアを用いるとプレイヤー全員で協力することで最も利益を得ることができるとして，誰も協力を裏切るようなことのない安定した配分をすることができるが，配分の集合が空や非常に大きい場合もありプレイヤーの配分を事前に予測することができなかった．そこで，利益を公正に分配する方法として Shapley 値がある．

Shapley 値とは，協力ゲームのもとでプレイヤーが連携して，その連携で得られた利益を分配する状況を考えたときにプレイヤー間で貢献度が異なることがあるため，貢献度に応じて利益を公正に分配する手法としてロイド・シャープレーによって導入された．貢献度は，プレイヤーそれぞれの限界貢献度から計る．限界貢献度の導出には，プレイヤー i を含む連携 S において特性関数が D で与えられたとき，$D(S) - D(S \setminus \{i\})$ によって得ることができる．また，その連携 S の発生のしやすさをかけたものを，プレイヤー i を含む全ての連携の組み合わせで足し合わせることで，次のように Shapley 値を求めることができる．

$$\phi_i = \sum_{j \in S \subset N} \frac{(s-1)!(n-s)!}{n!} \{D(S) - D(S \setminus \{i\})\} \tag{3.15}$$

ここで，s は連携 S に対する要素数を表し，特性関数 D は連携 S に含まれる DMU で再構成された新たな DMU の組に対しての DEA 分析を行う．ここでプレイヤーが N 人いたとき，特性関数 D は 2^N 個を得ることができ，連携 S によって特性関数の値は，変わる．DEA のクロス効率行列からこの Shapley 値を解くことで，DMU のそれぞれの貢献度を本節では導出する．

また，DEA の入出力項目の評価ベクトルを決める被評価対処 DMU_o と DMU の評価ベクトルを決める DMU 全体の評価者との間のゲームとして捉えることを DEA ゲームという [?, ?]．DEA ゲームに持ち込むために，まずクロス効率行列のクロス効率値を行和で割

表 3.2: 行で正規化したクロス効率値

評価 DMU	被評価 DMU					行和
	DMU_1	DMU_2	DMU_3	$\ldots$	DMU_N	
DMU_1	θ'_{11}	θ'_{12}	θ'_{13}	$\cdots$	θ'_{1N}	1
DMU_2	θ'_{21}	θ'_{22}	θ'_{23}	$\cdots$	θ'_{2N}	1
DMU_3	θ'_{31}	θ'_{32}	θ'_{33}	$\cdots$	θ'_{3N}	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
DMU_N	θ'_{N1}	θ'_{N2}	θ'_{N3}	$\cdots$	θ'_{NN}	1

ることで，各行を正規化する．正規化したクロス効率値を θ'_{ij} とおくと，表 3.2 の θ'_{ij} は次のように求めることができる．

$$\theta'_{ij} = \frac{\theta_{ij}}{\sum_{i=1}^{N} \theta_{ij}} \tag{3.16}$$

ここで，DMU_i の C.W. を次のように与える．

$$w_i = w_i^j \tag{3.17}$$

このとき，j の特性関数 C は次のようになる．

$$C(j) = \text{Maximize} \quad \sum_{i=1}^{N} w_i^j \theta'_{ij} \tag{3.18}$$

$$\text{Subject to} \quad \sum_{i=1}^{J} w_i^j = 1 \tag{3.19}$$

$$w_i^j \geqq 0 \qquad \forall i \tag{3.20}$$

$\theta'_i(S)$ は，i と j による評価をされた正規化後の θ'_{ij} の和として次のように表される．

$$\theta'_i(S) = \sum_{j \in S} \theta'_{ij} \tag{3.21}$$

$$\tag{3.22}$$

双対形は同じ解を得ることができることから，j の連携の要素数が 1 のときの特性関数 D の値は次のようになる．

$$D(j) = \text{Minimize} \quad \sum_{i=1}^{N} w_i^j \theta'_{ij} \tag{3.23}$$

$$\text{Subject to} \quad \sum_{i=1}^{N} w_i^j = 1 \tag{3.24}$$

$$w_i^j \geqq 0 \qquad \forall i \tag{3.25}$$

ここで，DMU_i の連携を考慮したときの C.W. を次のように与える．

$$w_i = w_i^S \tag{3.26}$$

よって，連携 S のときの特性関数は次のようになる．

$$D(S) = \text{Minimize} \quad \sum_{i=1}^{N} w_i^S \theta_i'(S) \tag{3.27}$$

$$\text{Subject to} \quad \sum_{i=1}^{N} w_i^S = 1 \tag{3.28}$$

$$w_i^S \geqq 0 \qquad \forall i \tag{3.29}$$

本研究において任意の連携 S と i について考えたときに，$S \cap i$ であるとき次の式が成り立つ．

$$D(S \cup i) + D(S) \geqq 0 \tag{3.30}$$

よって，特性関数は有加法性を満たすことが示すことができる．
さらに，j の配分を x_j とすると，

$$x_j \geqq D(j) \tag{3.31}$$

をみたすことから，個人合理性を満たす．
そして，行和でクロス効率値を割ったことから，

$$D(N) = 1 \tag{3.32}$$

よって，全体合理性を満たす．
以上から本研究でのシステムでは，有加法性を持ち個人合理性と全体合理性を満たすことから Shapley 値による分析ができる．そして，本研究でのクロス効率行列を使った C.W. の導出について述べる．まず，DMU_i の C.W. を次のように与える．

$$w_i = w_i^S \tag{3.33}$$

このとき特性関数は，次の LP となる．

$$D(k) = \text{Minimize} \quad u_1\alpha_d + u_2\mu_d \tag{3.34}$$

$$\text{Subject to} \quad w_1\beta_d + w_2\sigma_d + w_3\gamma_d = 0 \tag{3.35}$$

$$(u_1\alpha_{d'} + u_2\mu_{d'}) - (w_1\beta_{d'} + w_2\sigma_{d'} + w_3\gamma_{d'}) \leqq 0 \qquad \forall d' \tag{3.36}$$

$$u_1, u_2, w_1, w_2, w_3 \geqq 0 \tag{3.37}$$

ここで得られた特性関数を 4.20 式に当てはめることで，それぞれの DMU の Shapley 値を求めることができる．

表 3.3: クロス効率性行列と異なる重みの C.W.

評価 DMU	被評価 DMU					C.W.
	DMU_1	DMU_2	DMU_3	$\dots$	DMU_N	
DMU_1	θ'_{11}	θ'_{12}	θ'_{13}	$\cdots$	θ'_{1N}	w_1^N
DMU_2	θ'_{21}	θ'_{22}	θ'_{23}	$\cdots$	θ'_{2N}	w_2^N
DMU_3	θ'_{31}	θ'_{32}	θ'_{33}	$\cdots$	θ'_{3N}	w_3^N
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
DMU_N	θ'_{N1}	θ'_{N2}	θ'_{N3}	$\cdots$	θ'_{NN}	w_N^N
Shapley 値	ϕ_1	ϕ_2	ϕ_3	$\cdots$	ϕ_N	

表 3.4: 導出された Sharpley 値と共通の重み

	被評価 DMU				
	DMU_1	DMU_2	DMU_3	$\dots$	DMU_N
Shapley 値	0.000330174	0.0005206707	0.0008710043	$\cdots$	0.0000951163
共通の重み	0.0	0.0	0.0086897253	$\cdots$	0.0

§ 3.3 数法則の発見

本節では，共通の重みを適用した総合評価システムの開発を行う．個別の DMU に対して，総合評価の参照値が配分で示されていると考えられるので，クロス効率値に対して各 DMU の重みを乗じた値の和がその配分となるように最小二乗法により各 DMU の重みを推定する．

Shapley 値から，導出される DMU の C.W. は次の最小二乗法による計算で決定される．ここで最小二乗法とは，誤差を伴う測定値の処理系において，その誤差の二乗の和を最小にすることで，最も確からしい関係式を求める手法である．最小二乗法による導出は以下である．

$$\text{Minimize} \qquad \epsilon \tag{3.38}$$

$$\text{Subject to} \qquad \sum_{i=1}^{N} w_i^N \theta'_{ij} + s_j^+ - s_j^- = \phi_j \quad (\forall j) \tag{3.39}$$

$$\sum_{i=1}^{N} w_i^N = 1 \tag{3.40}$$

$$0 \leqq s_j^+ \leqq \epsilon,\ 0 \leqq s_j^- \leqq \epsilon \quad (\forall j) \tag{3.41}$$

$$w_i^N \geqq 0 \quad (\forall i) \tag{3.42}$$

表 3.3 に正規化したクロス効率行列と異なる重みの C.W. と Shapley 値をまとめている．

ここで，実際に導出された重みは，表 3.4 に示す．平均クロス効率値では，C.W. の値が同じであったのに対して，C.W. の値は貢献度に応じて値が変わるようになった．

最小二乗法により，C.W. が決定されたため効率値 $\tilde{\theta}_i$ は次の式で求められる．

$$\tilde{\theta}_i = \sum_{j=1}^{N} w_j^N \theta'_{ji} \tag{3.43}$$

DEA は，DMU ごとに自己都合の重みを決定することで，効率値を導出していた．そのため，この通常の DEA では D 効率的と判断される DMU が多数存在することがあり，最大効率の DMU 同士は，区別がつけられず比較ができないため意思決定の妨げとなる．そこで，DMU の順位付けの方法として，クロス効率値を用いたクロス効率性行列の研究がある [?]．これは，通常の DEA で導出される DMU ごとの最適な入力項目と出力項目に対する重みを用いて、自身以外の DMU を相互評価することで得られるクロス効率値を得る．得られたクロス効率値からなるクロス効率行列を用いてそれぞれの DMU の効率値を導出する手法となっている．

ここで，DMU が相互評価することでクロス効率行列は表 3.7 のように与えられる．このクロス効率行列上にある評価 DMU に対して被評価 DMU が一致している部分，すなわち行列の対角成分に注目する．これは，評価 DMU が自分自身を評価していることになる．すなわち，自己都合の重みで自己を評価しているため，いわば通常の DEA が対角成分では行われている．対角成分では通常の DEA が行われていることから，各 DMU の効率値は対角成分が最大効率値となっている．

相互評価を行うことで，他人に取って都合の良い評価項目の重みは，自分にとっては必ずしも良い重みではないため相対的に効率値が下がる．そのため，全 DMU が相互評価を行うことで，評価値を下げることができると考えられる．そして，各 DMU の重みを等価と見做すとことで DMU の効率値は，クロス効率値に対して各 DMU の重みを乗じた値の和の平均 (加重平均) により，平均クロス効率値として導出される．

本研究では，3.3 節で用いた従来の DEA を拡張し開発されたクロス効率性分析を用いた評価分析を行う．3.3 節のように，DMU の数が N 個で入力項目と出力項目の数がそれぞれ P と Q 個を持つ DMU_o(o 番目の DMU) の効率値を ϕ_o とおく．また，x_{op} を $\mathrm{DMU}_o(o = 1, \cdots, N)$ の p 番目 $(p = 1, \cdots, P)$ の入力，y_{oq} を $\mathrm{DMU}_o(o = 1, \cdots, N)$ の q 番目 $(q = 1, \cdots, Q)$ の出力とおく．

また，各 DMU の重みからクロス効率値を得るために，w^*_{op} を x_{op} に対する重み,u^*_{oq} を y_{oq} に対する重みとして新たな変数として設定する．このとき 3.3 節で用いた通常の FP の式を用いて DMU_o の効率値 ϕ_o は次のように定式化される．

$$\text{Maximize} \qquad \phi_o = \frac{\displaystyle\sum_{q=1}^{Q} u^*_{oq} y_{oq}}{\displaystyle\sum_{p=1}^{P} w^*_{op} x_{op}} \tag{3.44}$$

表 3.5: DMU の評価項目ごとの重み

対象 DMU	各入力項目の重み			各出力項目の重み		
	x_1	$\cdots$	x_P	y_1	$\cdots$	y_Q
DMU_1	w^*_{11}	$\cdots$	w^*_{1P}	u^*_{11}	$\cdots$	u^*_{1Q}
DMU_2	w^*_{21}	$\cdots$	w^*_{2P}	u^*_{21}	$\cdots$	u^*_{2Q}
DMU_3	w^*_{31}	$\cdots$	w^*_{3P}	u^*_{31}	$\cdots$	u^*_{3Q}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
DMU_N	w^*_{N1}	$\cdots$	w^*_{NP}	u^*_{N1}	$\cdots$	u^*_{NQ}

$$\text{Subject to} \qquad \frac{\sum_{q=1}^{Q} u^*_{oq} y_{o'q}}{\sum_{p=1}^{P} w^*_{op} x_{o'p}} \leqq 1 \qquad o' = 1, \cdots, N \tag{3.45}$$

$$w^*_{ip}, u^*_{iq} \geqq 0 \qquad \forall p, \forall q \tag{3.46}$$

またこのとき FP は, 次の LP に変換される.

$$\text{Maximize} \qquad \theta_o = \sum_{q=1}^{Q} u^*_{oq} y_{oq} \tag{3.47}$$

$$\text{Subject to} \qquad \sum_{p=1}^{P} w^*_p x_{op} = 1 \tag{3.48}$$

$$\sum_{q=1}^{Q} u^*_{oq} y_{o'q} \leqq \sum_{p=1}^{P} w^*_p x_{o'p} \qquad o' = 1, \cdots, N \tag{3.49}$$

$$w^*_p, u^*_q \geqq 0 \qquad \forall p, \forall q \tag{3.50}$$

この LP から，得られた DMU ごとの評価項目に対する重みを表 3.5 にまとめた．この重みを使って，相互評価を行っていく．$\theta_{oo'}$ を DMU_o の重みによる $\mathrm{DMU}_{o'}$ の評価値として次のように定式化する．

$$\theta_{oo'} = \frac{\sum_{q=1}^{Q} u^*_{oq} y_{o'q}}{\sum_{p=1}^{P} w^*_{op} x_{o'p}} \tag{3.51}$$

つぎに，相互評価によって得られたクロス効率値 $\theta_{oo'}$ で作られるクロス効率行列を用いて，それぞれの $\theta_{oo'}$ の共通の重み (Common Weight: C.W.) を等価として，平均クロス効率

表 3.6: 等価な C.W. の重み

	被評価 DMU				
	DMU_1	DMU_2	DMU_3	$\ldots$	DMU_N
共通の重み	0.03125	0.03125	0.03125	$\cdots$	0.03125

表 3.7: クロス効率性行列と等価な重みの C.W.

評価 DMU	被評価 DMU					C. W.
	DMU_1	DMU_2	DMU_3	$\ldots$	DMU_N	
DMU_1	θ_{11}	θ_{12}	θ_{13}	$\cdots$	θ_{1N}	$1/N$
DMU_2	θ_{21}	θ_{22}	θ_{23}	$\cdots$	θ_{2N}	$1/N$
DMU_3	θ_{31}	θ_{32}	θ_{33}	$\cdots$	θ_{3N}	$1/N$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
DMU_N	θ_{N1}	θ_{N2}	θ_{N3}	$\cdots$	θ_{NN}	$1/N$
平均	$\bar{\theta}_1$	$\bar{\theta}_2$	$\bar{\theta}_3$	$\cdots$	$\bar{\theta}_N$	

値を導出する．このとき，i 番目の生徒の C.W. である w_i は次のようになる．

$$w_i = \frac{1}{N} \tag{3.52}$$

よって，生徒の数 N が例えば 32 人であるようなときの C.W. は表 3.6 のようになる．C.W. は等価であるので平均クロス効率値は次のように加重平均で求めることができる．

$$\bar{\theta_o} = \frac{1}{N}\sum_{o'=1}^{N}\theta_{o'o} \tag{3.53}$$

導出されたクロス効率値 $\theta_{oo'}$ と平均クロス効率値 $\bar{\theta_o}$ と C.W. をまとめたものが表 3.7 となる．つぎに，本研究で生徒間でクロス効率行列をつくるために各変数を定義する．3.3 節の条件のもとで DMU ごとの重みを区別するために，DMU_i の α の重みを u^*_{i1}, μ の重みを u^*_{i2} とし，出力では β の重みを w^*_{i1}，γ の重みを w^*_{i2}, σ の重みを w^*_{i3} とした. すると，3.3 節の FP は次のようになる．

$$\text{Maximize} \qquad \phi_i = \frac{u^*_{i1}\alpha_i + u^*_{i2}\mu_i}{w^*_{i1}\beta_i + w^*_{i2}\sigma_i + w^*_{i3}\gamma_i} \tag{3.54}$$

$$\text{Subject to} \qquad \frac{u^*_{i1}\alpha_j + u^*_{i2}\mu_j}{w^*_{i1}\beta_j + w^*_{i2}\sigma_j + w^*_{i3}\gamma_j} \leqq 1 \qquad j = 1, \cdots, N \tag{3.55}$$

$$u^*_{i1}, u^*_{i2}, w^*_{i1}, w^*_{i2}, w^*_{i3} \geqq 0 \tag{3.56}$$

また，この FP を LP に変換したものが，次のようになる．

$$\text{Maximize} \qquad \theta_i = u^*_{i1}\alpha_i + u^*_{i2}\mu_i \tag{3.57}$$

$$\text{Subject to} \qquad w^*_{i1}\beta_i + w^*_{i2}\sigma_i + w^*_{i3}\gamma_i = 0 \tag{3.58}$$

$$(u^*_{i1}\alpha_j + u^*_{i2}\mu_j) - (w^*_{i1}\beta_j + w^*_{i2}\sigma_j + w^*_{i3}\gamma_j) \leqq 0 \qquad j = 1, \cdots, N \tag{3.59}$$

$$u^*_{i1}, u^*_{i2}, w^*_{i1}, w^*_{i2}, w^*_{i3} \geqq 0 \tag{3.60}$$

つぎに，導出された DMU の評価項目ごとの重みを用いて，生徒のクロス効率値を求める．ここで，θ_{ij} は，生徒 i の評価項目の重みで生徒 j の評価したクロス効率値とする．

$$\theta_{ij} = \frac{u^*_{i1}\alpha_j + u^*_{i2}\mu_j}{w^*_{i1}\beta_j + w^*_{i2}\sigma_j + w^*_{i3}\gamma_j} \tag{3.61}$$

さらに，クロス効率値から C.W. を等価にした生徒 i の平均クロス効率値 $\bar{\theta}_i$ は次のようになる．

$$\bar{\theta}_i = \frac{1}{N}\sum_{j=1}^{N}\theta_{ji} \tag{3.62}$$

この平均クロス効率値を生徒の評価値として用いる．この平均クロス効率値は，DMU のそれぞれのクロス効率値の重み C.W. を等価として導出された．しかし，通常の DEA は各 DMU が最大効率値をもつように各評価項目に対して重みをつけていたことで，必然的に平均クロス効率値は通常の DMU の効率値の値を上回ることはない．他の都合のいい重みで評価項目を評価されることに加えて，得られたクロス効率値を単に平均を取ることに対して DMU 全体としては合意することができない．次の節では，DMU 全体が合理できるような仕組みを考慮して DMU の効率値を導出する．

第4章

提案手法

§ 4.1　ODペアからの入出力抽出

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4.1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 [25].

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

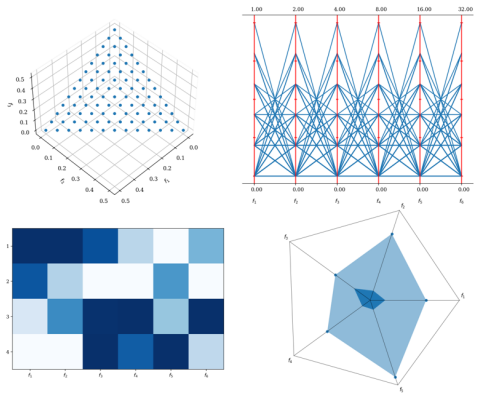

図 4.1: pymoo における様々な視覚化手法

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　モデル式の出し方

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

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 [26].

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 提案手法の流れ

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

Step 1: 料理レシピ，食材価格のデータベースの作成

はじめに，Python を使って Web サイトからスクレイピングし，料理レシピと食材価格のデータベースを作成する．本研究で用いるスクレイピング手法として，Python のライブラリである BeautifulSoup4 を扱う．BeautifulSoup4 は Web サイト上の HTML から，取得したいデータを HTML 内のクラスや ID などの要素検索して抽出することができる．また，最初に対象の Web ページから HTML を取得する必要があるため，その際に HTML パーサーである Requests を用いる．これらのスクレイピング手法を用いて，レシピ情報を取得する．料理レシピサイトとして,「ボブとアンジー」のみを参考に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 [27]，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 [28] 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 [19]．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 [20]．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 [22]．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 [23]．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 [27]，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.

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．

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．

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，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．この結果より，出力する人数を変更は処理時間に大きく影響しないことがわかった．

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

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

[3] ソフトム株式会社, “ソフトム通信　第 79 号「給食業界におけるＡＩ活用」”, https://data.nifcloud.com/blog/food-service-provider_ai-use-case_01/, 閲覧日 2022.12.28.

[4] 貝沼やす子, 江間章子, “日常の献立作りの実態に関する調査研究（第１報）”, 日本調理学会誌, Vol.30, No. 4, pp. 364-371, 1997.

[5] 株式会社おいしい健康, “おいしい健康”, https://oishi-kenko.com/, 閲覧日 2022.10.16.

[6] 総務省統計局, “小売り物価統計調査による価格調査”, https://jpmarket-conditions.com/, 閲覧日 2022.10.11.

[7] J. W. Ratcliff and D. Metzener, “Pattern Matching: The Gestalt Approach”, *Dr. 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] Q. Zhang and H. Li, “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition”, *IEEE Trans. Evolutionary Computation*, *Vol. 11*, No. 6, pp. 712–731, 2007.

[10] LeftLetter, “多目的進化型アルゴリズム MOEA/D とその改良手法”, https://qiita.com/LeftLetter/items/a10d5c7e133cc0a679fa, 閲覧日 2023.1.6.

[11] J. H. Holland, “Adaptation in Natural and Artificial Systems”, 1975.

[12] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”, *IEEE Tran. on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197, 2002.

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

[14] メディカル・ケア・サービス株式会社, “制限食にはどんな種類があるの？”, 健達ネット, https://www.mcsg.co.jp/kentatsu/health-care/12106, 閲覧日 2023.1.6.

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

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

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

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

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

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

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

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

[23] 厚生労働省，“高血圧”，https://www.mhlw.go.jp/content/10904750/000586583.pdf，
閲覧日 2023.01.17

[24] 厚生労働省，“食べ物アレルギー”，アレルギーポータル，
https://allergyportal.jp/knowledge/food/， 閲覧日 2023.01.17

[25] J. Blank，“pymoo: Multi-objective Optimization in Python ”，
https://www.egr.msu.edu/ kdeb/papers/c2020001.pdf，閲覧日 2023.1.22.

[26] 和正敏，“多目的線形計画問題に対する対話型ファジィ意思決定手法とその応用 ”，電子情報通信学会論文誌 Vol. J 65-A， No. ll， pp. 1182-1189， 1982.

[27] 厚生労働省，“日本人の食事摂取基準 (2020 年版) ”，
https://www.mhlw.go.jp/content/10904750/000586553.pdf， 閲覧日 2022.12.26.

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