

## 進捗報告

多目的最適化を用いた献立推薦システムの段階的デバッグ

辻 琉玖

Ruku Tsuji

u220039@st.pu-toyama.ac.jp

富山県立大学 工学部 情報システム工学科

13:10-14:50, Tuesday, June 10, 2025

N516, Toyama Prefectural University

## はじめに

本発表では, Python で開発された多目的最適化 (NSGA-II) を用いた献立推薦システムのデバッグプロセスについて報告する.

初期のプログラムは, 複数の問題を抱えており, それらを段階的に特定し, 解決することでシステムの安定動作を実現した.

## 主な課題

- データ処理の不整合
- 外部ファイル参照のエラー
- プログラム間連携におけるデータの非互換性

デバッグ過程で直面した主な問題点は、以下の4つに大別される。

- 1 **インデックス管理の問題：** レシピ名に依存した検索による不整合
- 2 **データ同期の問題：** スライダー操作後の複数リストの非同期
- 3 **ファイルパスの問題：** 外部データソースの参照エラー ('FileNotFoundException')
- 4 **データ連携の問題：** 文字コード不一致とファイルI/O ロジックの欠陥

本発表では、これらの問題を一つずつ詳述する。

# 課題 1：インデックス再取得による不整合

4/10

## 問題の状況

最適化後の献立候補からユーザーが一つを選択した際、そのレシピ名をキーにして、データリストからインデックスを再検索していた。

## 問題があったコード

```
# レシピ名リストからインデックスを再検索
number = []
for i in kondatename:
    # 同名のレシピがあるとバグの原因になる
    number.append(yi.index(i))
```

# 解決策1：インデックスリストの直接利用

5/10

## 修正方針

最適化の段階で、レシピ名リスト ('paratename') と並行して、対応する**インデックスのリスト ('parate')** も保持・ソートするようにした。ユーザーが候補を選択した後は、この 'parate' リストから直接インデックスを取得する方法に変更した。

## 修正後のコード

```
# paratename(名前) ではなく parate(インデックス) を直接利用
kondatename = paratename[h]
number = parate[h]
```

## 問題

GUIのスライダーで候補を並べ替える際、表示用のリスト群は更新されるものの、内部で管理するインデックスリスト ('parate') が一緒にソートされていなかった。これにより、表示と内部データが不一致になった。

## 解決策

Python の 'zip'関数やリスト内包表記を用いて、関連する全てのリストをグループ化し、一度に並べ替える処理に修正。これにより、リスト間の同期が常に保たれるようになった。

## 課題3と解決策：ファイルパスの誤り (‘FileNotFoundError’)

7/10

### 問題

‘howto.csv’を読み込む際、プログラムの実行場所を基準（‘./’）にしたパスを指定していたため、ファイルを見つけれずエラーが発生した。

### 解決策

「一つ上の階層」を意味する‘../’を用いて、正しい相対パスを指定し直すことでエラーを解消した。

# 修正前

```
pd.read_csv("./data/howto_data/howto.csv", ...)
```

# 修正後

```
pd.read_csv("../howto_data/howto.csv", ...)
```

## 課題 4：データ連携時の列ズレ問題

8/10

### 問題の状況

- 1 **文字コードの不一致**： 献立生成プログラムが CSV を 'cp932' で保存する一方、サーバープログラムは 'shift-jis' で読み込んでいた.
- 2 **ファイル I/O ロジックの不備**： 献立生成プログラムが、  
'data/user/' フォルダ内に処理対象のファイルが存在しない場合、ファイルを新規作成せずにスキップするロジックだったため、ファイルの不整合を助長していた.



## 修正方針

- 1 文字コードの統一：** 'server1.py'の仕様に合わせ、'2 献立作成.py'が出力する全ての CSV ファイルを 'encoding='shift-jis'' に統一した.
- 2 ファイル作成ロジックの堅牢化：** 過去のデータを保護するため、「ファイルが存在すれば更新, 存在しなければ元データから新規作成する」という 'if-else' ロジックに変更した.

## 堅牢化したファイル書き出しロジック

```
if os.path.exists(dest_file_path):  
    # 存在すれば、過去データを保持しつつ更新  
    df_to_write = pd.read_csv(...)  
else:  
    # 存在しなければ、マスターデータから新規作成  
    df_to_write = pd.read_csv(...)
```

# 最終的に shift-jis で保存

## これからすること

すべての修正が完了したため、次に応用可能な機能の調査（サーベイ）を行う。

案として、物価が高騰している現在の食材の価格をリアルタイムで取得する機能が考えられる。しかし、現在地から遠い場所の店舗の価格を取得しても実用性が低いという課題がある。