Optimizing Meal Planning via Mixed-Integer Optimization

Ismail Berbache
berbache@mit.edu

Alessandro Morosini
morosini@mit.edu

December 9, 2024

Contents

1 Introduction

1.1 Context and Purpose

Effective meal planning is essential for maintaining health and achieving specific dietary goals, particularly for individuals with targeted objectives such as muscle gain. These individuals often aim to consume a high caloric intake while maintaining a diverse and healthy diet. However, balancing macronutrient ratios, ensuring sufficient calorie intake, and minimizing undesirable components like added sugars can be a complex and time-consuming task. This project addresses these challenges by utilizing mixed-integer optimization to develop a framework for scientifically balanced, high-calorie weekly meal plans. The optimization prioritizes diverse, nutrient-dense foods while aligning with health standards and individual preferences.

Our focus is relevant for athletes and individuals looking to gain muscle mass, but also anyone interested in gaining weight in a healthy fashion. Additionally, our framework minimizes reliance on low-quality calories, such as those from added sugars, ensuring that the plans promote long-term health and performance. Beyond its immediate application, the methodology offers potential for broader uses, such as grocery optimization, highlighting its versatility and potential on dietary management.

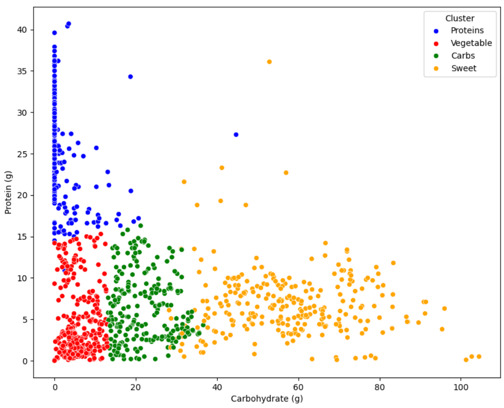
1.2 Data

This study utilizes McCance and Widdowson's The Composition of Foods Integrated Dataset (CoFID), a comprehensive resource that details the nutritional composition of thousands of foods consumed in the UK. CoFID provides nutrient data per 100 grams for over 3,000 food items, encompassing essential macronutrients like carbohydrates, proteins, and fats, alongside an extensive array of vitamins, minerals, and bioactive compounds. This granularity allows for precise tailoring of dietary optimization to meet diverse nutritional requirements, making it ideal for our modeling needs.

CoFID is available in user-friendly formats, including Excel (xlsx) and CSV, which facilitates seamless integration with computational tools and optimization models. Each food item is categorized into detailed sub-groups, such as cereals, dairy products, and vegetables, enabling efficient filtering and analysis based on dietary preferences or constraints. The dataset's robust coverage of both raw and processed foods ensures applicability across a wide range of dietary scenarios. These features make CoFID a reliable and versatile resource for constructing optimization models that prioritize nutritional adequacy, dietary variety, and meal balance.

2 Clustering Food Items

To enable the optimization of meal planning, it is necessary to categorize foods into distinct groups such as carbohydrates (e.g., rice, pasta, potato), proteins (e.g., fish, beef, chicken), and others. This is crucial because the optimization problem requires the selection of exactly one item from each group to construct a nutritionally balanced meal. However, the dataset lacks predefined labels for such categories. To address this, we run k-means on the nutritional attributes `Carbohydrates`, `Proteins`, `Total Sugars`, and `AUC Fibers`. The resulting clusters ensure that similar foods are grouped together in a nutritionally coherent way, allowing the optimization process to yield realistic and practical meal combinations. Investigating the scree plot we choose $k = 4$ clusters, which we later identify as `Proteins`, `Vegetables`, `Carbs` and `Sweets`.



(a) Carbohydrates vs Proteins per Cluster

```
Cluster 1 has 247 elements:
80                         Bacon rashers, back, microwaved
2456                          Sardines, flesh only, grilled
682          Chicken, breast, strips, stir-fried in corn oil
2118                     Pork, fillet medallions, grilled lean
1947                           Peanuts, roasted and salted
667     Chicken in white sauce, made with semi-skimmed milk
1282                     Haddock, flesh only, smoked, steamed
713                       Chicken, skin, roasted/grilled, dry
2487                                 Sausages, beef, grilled
626                                 Cheese, Parmesan, fresh
Name: Food Name, dtype: object
```

(b) Random Items in Protein Cluster

Significant work went into yielding meaningful clusters - see code appendix. Yet, as this report focuses on optimization, we only discuss the results here. Figure 1a shows how nutritional attributes `Proteins` and `Carbohydrates` vary according to our newly found labels, and Figure 1b presents 10 randomly sampled items in the protein cluster. Notice some items, such as `Peanuts` or `Cheese` do not naturally belong to one of the 4 categories available. We do not expect this to be a problem as our optimization formulation should be strong enough to realize these aren't smart options for a balanced meal.

3 Single Meal Optimization

3.1 Formulation

We start with building a model to optimize caloric intake subject to nutritional constraints for a single meal. Let F be the set of all the food items in our dataset. Define K as the set of nutrients we consider (e.g Proteins, Fibers) with R_k and U_k the required lower and upper bounds for nutrient k in our meal. Let c_f and $n_{f,k}$ denote the caloric and nutritional content in 100 grams of item f. We also partition F into clusters C as defined in Section 2 with $F_C \subseteq F$ the items in group C. Introduce the decision variables $y_f \in \{0,1\}$ to indicate whether a food item $f \in F$ is selected, and $q_f \geq 0$ to represent its quantity in grams in our meal. We formulate the meal selection problem as the following.

$$
\begin{aligned}
\max \quad & \sum_{f \in F} \frac{c_f}{100} q_f \; - \lambda \sum_{f \in F} q_f && && (1)\\
\text{s.t} \quad & q_f \leq 300\, y_f && \forall f \in F && (2)\\
& q_f \geq 10\, y_f && \forall f \in F && (3)\\
& \sum_{f \in F_c} y_f \leq 1 && \forall c \in C && (4)\\
& \sum_{f \in F} \frac{n_{f,k}}{100} q_f \leq U_k && \forall k \in K && (5)\\
& \sum_{f \in F} \frac{n_{f,k}}{100} q_f \geq R_k && \forall k \in K && (6)\\
& y_f \in \{0,1\}, \; q_f \geq 0 && && (7)
\end{aligned}
$$

Interpretation. Constraints (4) is imposing to choose at most one item from each cluster and, in conjunction with (2) and (3), enforces the sparsity of our selection. Notice 300 and 10 were constants chosen for managerial reasons, as it would not make sense to eat more than 300 or less than 10 grams of a chosen item. Constraints (5) and (6) ensure that the overall meal provides a healthy amount of each nutrient, within recommended limits. We decided to model them as hard constraints rather than within the objective to ensure we are never outside of the healthy range.

Multi-objective. Our objective function (1) reflects the trade-off between calorie intake and meal weight. It is necessary to include this multi-objective component and fine-tune the hyper-parameter λ to ensure realistically consumable meals. As shown in Figure 2, the lower λ the heavier and calorie-intensive the solution. For the remaining of this study, we choose $\lambda = 0.1$ as it offers a balance between amount of calories and weight of the meal.

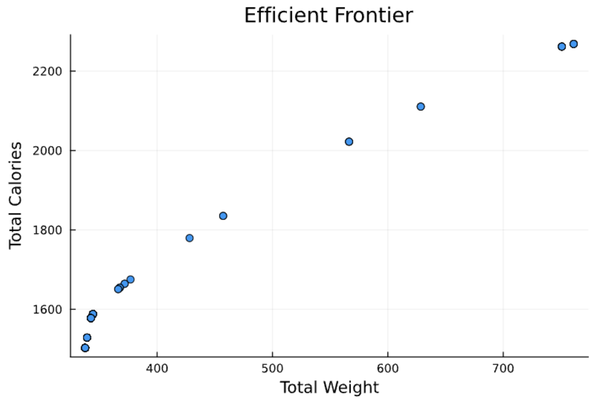


Figure 2: Calories vs Weight Pareto Frontier

3.2 Results

Implementing the above problem in Julia, after intensive data cleaning and pre-processing, we obtained the optimized meal composition presented in Table 1. We also present the nutrient profile of the optimized meal in Figure 3, confirming all specified nutritional requirements are fully met, as expected from our hard constraints.

Food Item	Cluster	Quantity (g)	Calories
Cereal, plain rolled oats	Sweet	166.99	569.45
Pork chops in mustard and cream	Proteins	188.16	494.86
Sandwich, white bread	Carbs	128.20	220.50
Tomato puree	Vegetable	85.85	57.52

Table 1: Optimal Single Meal

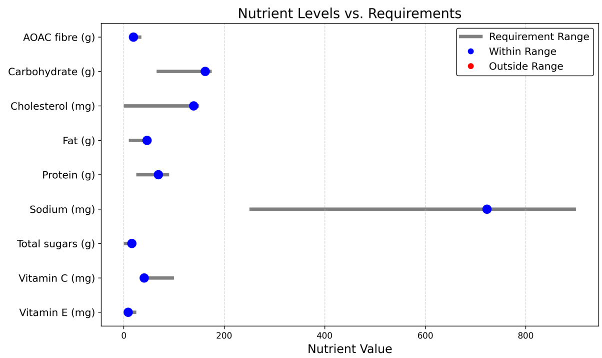
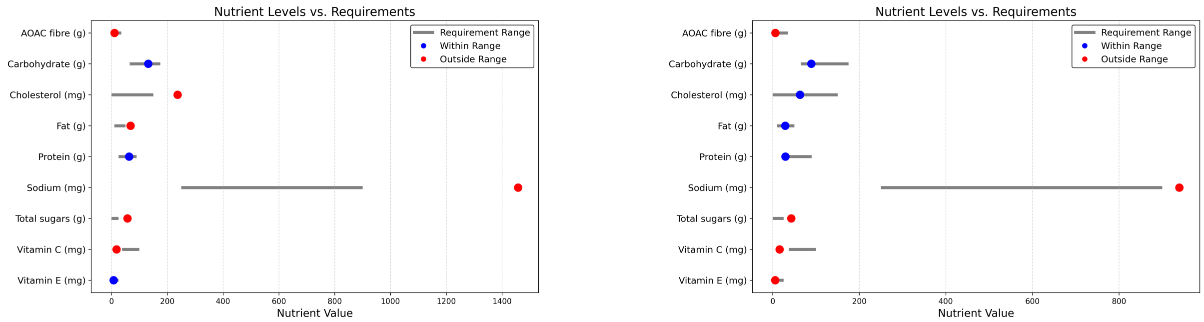


Figure 3: Optimal Meal Nutritional Profile

To evaluate the resulting meal, we compare its nutrient profile against two baselines; an average dish derived from sampling 1000 meals at random satisfying all constraints but the nutritional ones and a typical student lunch (details in appendix).

Figure 4 presents the nutrient profiles of these two comparison meals side-by-side. Red points indicate violations of nutritional constraints. We see both the average and the typical student meal often fail to meet the recommended nutrient requirements.



(a) Average Meal

(b) Typical Student Meal

Figure 4: Nutritional Profile of Baselines

Finally, Table 2 compares key caloric metrics across the three meal categories. The optimized meal is the most calorie-dense and achieves the highest calorie-to-weight ratio, reflecting our objective of maximizing caloric intake while minimizing overall meal weight, and satisfies all nutritional requirements, unlike the baselines.

Meal	Total Calories	Cal/Weight Ratio
Optimized Meal	1342.03	2.38
Average Meal	1260.81	2.11
Student Meal	653.26	1.45

Table 2: Calories (kcal) Across Meals

4 Robust Optimal Meal Planner

4.1 Formulation

Until now, we have restricted our modeling to a single meal. In practice, dietary choices are often influenced by previous servings, as consumers seek to have varied meals. From a health standpoint, nutritional stores accumulate or deplete over time, and represent mathematically interesting dynamics to add to our model. In this section, we introduce adaptive formulations that incorporate temporal structure, thereby offering a more realistic representation of sequential meal planning.

Diversity over Time. To extend the problem to a planning horizon of M meals, we index both the binary and continuous decision variables by meal, i.e., $y_{f,m}$ and $q_{f,m}$ for each $f \in F$ and $m \in \{1, \dots, M\}$. All previous constraints apply at the level of each meal, ensuring that we always satisfy nutrient requirements, cluster-level restrictions, and portion size bounds. To encourage diversity, we introduce a constraint that prevents any item from being repeated in consecutive meals.

$$y_{f,m} + y_{f,m-1} \leq 1 \quad \forall f \in F, \quad m = 2, \dots, M.$$

Moreover, to prevent undesirable cyclic patterns (e.g even meals being identical), we introduce r_f as an implicit penalty on the number of repeated selections of food f. Hence, we add $\mu \sum_{f \in F} r_f$ to our objective, where $\mu > 0$, as well as the following constraint.

$$r_f \geq \sum_{m=1}^{M} y_{f,m} - 1 \quad \forall f \in F.$$

Nutrient Stored and Adaptation. Introduce variables $s_{k,m}$ as the amount of nutrient k in the body shortly after meal m. For a nutrient-specific depletion factor $0 < \delta_k \leq 1$, we model the dynamics of $s_{k,m}$ as the following.

$$s_{k,m} = \delta_k \, s_{k,m-1} + \sum_{f \in F} \frac{n_{f,k}}{100} q_{f,m}, \quad s_{k,0} = 0.$$

The use of $s_{k,m}$ provides a mechanism to consider the cumulative effect of past meals, allowing for adaptive meal planning: if previous meals left a surplus of a given nutrient, the next meal may need less of it, and vice-versa. For simplicity, we choose two depletion rates: $\delta_k = 0.5$ for nutrients that are more readily stored (e.g., fat-soluble vitamins and fats), and $\delta_k = 0.0$ for nutrients that are not stored and must be replenished each meal (e.g., water-soluble vitamins and carbohydrates).

Robustness. While previous formulations assume that nutrient content values $n_{f,k}$ are known and exact, our data is subject to (little) uncertainty. Let's introduce robustness to our formulation. Let $\gamma > 0$ be such that $n_{f,k}^{TRUE} \in [(1-\gamma)n_{f,k}, (1+\gamma)n_{f,k}]$. We update constraints (5) and (6) in the previous formulation to their robust and adaptive counterparts. To do so, we define two sets of store variables, $s_{k,m}^L$ and $s_{k,m}^U$, representing the lower and upper bound scenarios for nutrient k after meal m.

$$s_{k,m}^L = \delta_k \, s_{k,m-1}^L + \sum_{f \in F} \frac{(1-\gamma)\, n_{f,k}}{100} q_{f,m}, \quad s_{k,0}^L = 0,$$

$$s_{k,m}^U = \delta_k \, s_{k,m-1}^U + \sum_{f \in F} \frac{(1+\gamma) n_{f,k}}{100} q_{f,m}, \quad s_{k,0}^U = 0.$$

We also ensure the nutrient stores remain acceptable in adversarial scenarios:

$$s^L_{k,m} \geq R_k \quad \text{and} \quad s^U_{k,m} \leq U_k \quad \forall k \in K, \quad \forall m = 1, ..., M$$

Final Formulation After careful investigation, we set $\lambda = 1$, $\mu = 50$ and $\gamma = 5\%$. Our problem can then be written as follows.

$$\begin{aligned}
\max \quad & \sum_{m=1}^{M} \sum_{f \in F} \frac{c_f}{100} q_{f,m} \; - \; \sum_{m=1}^{M} \sum_{f \in F} q_{f,m} \; - \; 50 \sum_{f \in F} r_f && (8) \\
\text{s.t} \quad & \sum_{f \in F} y_{f,m} \leq 4 \quad \forall m = 1, \ldots, M && (9) \\
& q_{f,m} \leq 300 \, y_{f,m} \quad \forall f \in F, m = 1, \ldots, M && (10) \\
& q_{f,m} \geq 10 \, y_{f,m} \quad \forall f \in F, m = 1, \ldots, M && (11) \\
& \sum_{f \in F_c} y_{f,m} = 1 \quad \forall c \in C, m = 1, \ldots, M && (12) \\
& y_{f,m} + y_{f,m-1} \leq 1 \quad \forall f \in F, m = 2, \ldots, M && (13) \\
& r_f \geq \sum_{m=1}^{M} y_{f,m} - 1 \quad \forall f \in F && (14) \\
& s^L_{k,0} = 0, \quad s^U_{k,0} = 0 \quad \forall k \in K && (15) \\
& s^L_{k,m} = 0.8 \, s^L_{k,m-1} + \sum_{f \in F} \frac{0.95 \, n_{f,k}}{100} q_{f,m} \quad \forall k \in K, m = 1, \ldots, M && (16) \\
& s^U_{k,m} = 0.8 \, s^U_{k,m-1} + \sum_{f \in F} \frac{1.05 \, n_{f,k}}{100} q_{f,m} \quad \forall k \in K, m = 1, \ldots, M && (17) \\
& s^L_{k,m} \geq R_k \quad \forall k \in K, m = 1, \ldots, M && (18) \\
& s^U_{k,m} \leq U_k \quad \forall k \in K, m = 1, \ldots, M && (19) \\
& y_{f,m} \in \{0, 1\}, \; q_{f,m} \geq 0, \; r_f \geq 0, \; s^L_{k,m} \geq 0, \; s^U_{k,m} \geq 0 && (20)
\end{aligned}$$

4.2 Results

We implement the model in Julia, choosing a horizon of $M = 4$ meals. This length balances computational feasibility, allows the model to meaningfully optimize over time, and reflects a realistic planning horizon for forward-looking dietary decisions. Results are reported in Table 3.

Meal	Food Item	Cluster	Quantity (g)	Calories
Meal 1	Almonds, weighed with shells	Vegetable	10.00	20.50
	Pakora/bhajia, cauliflower, fried	Carbs	146.37	468.39
	Pork, diced, kebabs, grilled, lean	Proteins	102.27	204.54
	Wafers, plain ice cream wafers	Sweet	175.45	645.67
Meal 2	Beans, red kidney, dried, boiled	Carbs	10.00	10.00
	Crackers, wholemeal, homemade	Sweet	213.37	898.30
	Pepper, capsicum, red, boiled in water	Vegetable	44.13	7.50
	Tuna, flesh only, baked	Proteins	152.65	207.60
Meal 3	Blackcurrants, stewed with sugar	Carbs	34.32	19.91
	Breakfast cereal, wheat biscuits	Sweet	222.17	737.61
	Ghee made from vegetable oil	Vegetable	14.71	132.42
	Venison, roast	Proteins	103.81	171.28
Meal 4	Cereal, plain rolled oats	Sweet	223.33	761.57
	Cheese, Parmesan, fresh	Proteins	61.78	256.38
	Pepper, capsicum, yellow, boiled	Vegetable	45.35	11.79
	Potato products, shaped, frozen	Carbs	10.34	19.65

Table 3: Four-meal plan generated by the adaptive optimization model.

Overall, the resulting plan demonstrates the advantages of our time-adaptive, robust approach: it consistently meets nutritional targets, promotes variety, and adapts its selection as the sequence of meals unfolds, reflecting realistic and health-conscious meal planning.

5 Conclusion

In this work, we successfully developed a meal planner using mixed-integer optimization that consistently outperforms baseline methods, including random meal generation and typical team meals. The planner ensures nutritionally balanced, high-calorie meals designed to meet specific dietary goals, achieving superior adherence to nutritional requirements while maximizing caloric intake and minimizing meal weight. These results showcase the practical advantages of our framework in promoting effective and health-conscious meal planning. To extend the planner's applicability, we enhanced the framework to support sequential meal planning and robust optimization. By accounting for nutrient accumulation dynamics and variability over time, the sequential model ensures dietary diversity and long-term compliance with nutritional goals. Furthermore, the robust formulation addresses uncertainties in nutrient data, increasing the model's reliability in real-world scenarios.

6 Future Work

We recognize some limitations in our current framework, particularly in the formulation of the objective function. Nevertheless, this project provides a solid foundation for further development. For future work, we aim to incorporate additional factors such as price and utility, which are critical drivers of meal choices. Price information proved challenging to include due to the nature of the dataset, which focuses on prepared dishes rather than granular food types, making the retrieval of corresponding price data overly complex. Similarly, utility, which reflects individual preferences, was difficult to model. While we attempted to manually label hundreds of food items based on taste preferences and train machine learning models to predict utilities for the remaining items, the resulting accuracy was too low to be practically useful. This suggests the presence of more intricate dynamics underlying utility, which warrants deeper exploration. As a workaround, we used calories as a proxy for utility and meal weight as a proxy for price. While these assumptions are admittedly strong, they are not entirely unreasonable in specific contexts, such as a pay-by-weight buffet, where meal composition directly impacts both cost and satisfaction. An example could be the MIT Sloan Cafeteria, which is where the idea of this project generated from.