Automated and Personalized Meal Plan Generation and Relevance Scoring using a Multi-Factor Adaptation of the Transportation Problem

George Salloum[1] and Joe Tekli[1*]

[1] Lebanese American University (LAU), School of Engineering, E.C.E. Dept.

Byblos Campus, 36 Byblos, Lebanon

Abstract. Establishing a healthy lifestyle has become a very important aspect in people's lives. The latter requires maintaining a healthy nutrition by considering the nature and quantity of foods being consumed, allowing to regulate one's intake and consumption of calories and nutrients. As a result, people reach out for nutrition experts which services are costly, time consuming, and not readily available. While various e-solutions have been developed to perform meal planning, yet most of them lack a completely automated process and require domain expert intervention at different stages of the recommendation process (e.g., identifying macronutrient distribution, providing pre-defined meal plans, or combining recommended foods into meal structures). In addition, most solutions focus on fulfilling the patients' nutrition requirements (in terms of caloric intake and macronutrients) while disregarding other relevant factors such as patient food preferences, food variety, food-meal compatibility, and inter-food compatibility. Hence, there is a need for an automated solution to produce a full-fledged meal plan from scratch, based on a recommended caloric intake and considering multiple factors. In this study, we introduce a novel solution titled *MPG* for automated Meal Plan Generation recommendations, designed based on an adaptation of the transportation optimization problem to simulate the "human thought process" involved in generating daily meal plans. *MPG* allows to: (i) generate plans which fulfill a recommended caloric intake, given a set of available foods, while (ii) personalizing the plans following patient chosen factors (e.g., food preferences, variety, and compatibility), and (iii) evaluating the relevance of the produced plans following patient preferences. We have conducted various experiments involving 9 human testers and 124 meal plans to test the performance of *MPG*. Results highlight *MPG*'s effectiveness in producing "healthy" and personalized meal plans while complying with the testers' preferences.

Keywords: Personalized Meal Planning, Nutrition Health, Adapted Transportation Problem, Relevance Scoring, Parametric Model.

1. Introduction

Nowadays, establishing a healthy lifestyle has become a very important aspect in people's lives. The latter requires maintaining a healthy nutrition by considering the nature and quantity of foods being consumed, allowing to regulate one's intake and consumption of calories and nutrients. Poor nutrition has been shown to increase the risks of dangerous complications such as obesity, diabetes, and other health issues (Ayoub J. *et al.* 2015, Mattar L. *et al.* 2015). As a result, people reach out for nutrition experts to help them achieve healthy lifestyles. In this context, a few obstacles come to play: (i) the cost of seeking an expert's help which is recurring and non-trivial, (ii) the need to regularly meet with the expert which might not be always practical, and (iii) the need for readily accessible health services which might be difficult to provide by a human expert. An alternative popular approach is the use of electronic solutions, such as mobile applications and websites that are highly available and provide basic health and nutrition services (cf. literature review in Section 3). Yet most of these solutions share major weaknesses, namely: (i) lack of a complete automated process either requiring expert intervention, e.g., (Khan A. S. and Hoffmann A. 2003a, Khan A. S. and Hoffmann A. 2003b, Suksom N. and Buranarach M. 2010), or requiring the patient (user) to have some technical knowledge about nutrition health in order to utilize (and tune the parameters of) the e-solution, e.g., (Evans D. 2017, Livestrong Foundation 2020), (iii) performing meal planning or meal plan evaluation based on a combination of pre-fixed plans or pre-defined inputs set by the e-solution designer, e.g., (Khan A. S. and Hoffmann A. 2003a, Khan A. S. and Hoffmann A. 2003b, Suksom N. and Buranarach M. 2010), and (iii) allowing limited adaptability to the patient's preferences in terms of food affinity, variety, and compatibility, e.g., (MakeMyPlate Inc. 2020, Noor S. *et al.* 2018, Petot G. J. *et al.* 1998). Hence, there is a need for an automated solution to produce a full-fledged meal plan from scratch, based on a recommended caloric intake and considering multiple patient preference factors.

The main goal of this study is to create an intelligent agent that offers the same meal planning services offered by a human nutrition expert albeit doing it through a readily available, fully automated, and cheap e-solution. We aim to automate the process of producing personalized meal plans allowing patients to reach their target weight and BFP[1], while closely catering to their preferences. To achieve the latter services, we introduce a new solution for automated Meal Plan Generation titled *MPG*, designed based on an adaptation of the transportation optimization problem to simulate the "human thought process" involved in generating daily meal plans. *MPG* allows to: (i) generate meal plans which fulfil a recommended caloric intake (e.g., nutrition demand) given a set of available foods (e.g., nutrition supply), while (ii) personalizing the plans following patient chosen factors (e.g., assigning different weights for food preferences, variety, and compatibility), and (iii) evaluating their relevance following the patients'

[1] The Body Fat Percentage (BFP) is computed as the ratio of the patient's body fat weight over the total body weight. It is a common and expressive metric used in nutrition health practice (cf. Section 2).

* Corresponding author: joe.tekli@lau.edu.lb.

preferences. It consists of five main components: (i) *macronutrients calculator*: computes the amount of macronutrients (i.e., carbohydrates, protein, and fat) to fulfil the required caloric intake, (ii) *servings calculator*: computes the daily amount of servings for each of the six primary food categories (i.e., starch, fruits, milk, vegetables, lean meat, and fats) to fulfil the required macronutrients, (iii) *servings assignor*: splits the servings for each of the six primary food categories over the five daily meals (i.e., breakfast, snack one, lunch, snack two, dinner)[2] based on sample meal plan serving assignments, (iv) *food assignor*: allocates the foods to the five daily meals while meeting all the serving requirements and considering the patient's preferences, and (v) *meal plan evaluator*: computes the relevance score for every generated meal plan highlighting its compliance with patient preferences.

We have conducted various experiments involving 9 human testers and 124 meal plans to test the performance of *MPG*. Results highlight *MPG*'s effectiveness in generating healthy and personalized meal plans, producing recommendations that are on a par with those of human nutritionists, while complying with the testers' preferences.

The remainder of this paper is organized as follows. Section 2 introduces background notions on nutrition health. Section 3 briefly reviews the literature and existing e-solutions revolving around meal planning. Section 4 describes our *MPG* agent, its components, and functionalities. Section 5 describes our experimental evaluation and results. Finally, Section 6 concludes with future directions.

2. Background

2.1. Nutrition Health Preliminaries

A preliminary step to generating a meal plan is performing nutrition health assessment, where the human nutritionist collects input information regarding the patient, including gender, age, height, and weight. Next, the body mass index (BMI) and the body fat percentage (BFP) can be identified as follows (Mahan L. *et al.* 2012):

$$BMI = \frac{Weight(Kg)}{Height^2(m^2)} \quad \textbf{(1)}$$

$$BFP = \frac{Body\ Fat\ Weight\ (Kg) \times 100}{Total\ Body\ Weight\ (Kg)} \quad \textbf{(2)}$$

While BMI is an indication of the weight status of a person, yet it is not significant by itself since it disregards the patient's body fat composition. For instance, the author in (Khan A. S. and Hoffmann A. 2003b) demonstrated in a study of 486 subjects that about 87% of the patients classified as *normal* and *overweight* following BMI were actually *obese* according to their BFP. Hence, the combination of BMI and BFP is usually adopted and allows for a better assessment.

Once the patient's BFP is identified (through a simple test that can be performed at a specialized clinic or pharmacy), the expert nutritionist can decide on the target BFP and weight of the patient, and the daily caloric intake (CI) required to reach the latter. Weight change comes down to the gap between the caloric intake (CI) and the caloric expenditure (CE), meaning the difference between the amount of energy acquired from food and the energy consumed by the human body, both measured in Kilocalories (Kcal). If the daily CI is larger than the daily CE, the patient will gain weight in the long run, and vice versa (Hall K. D. et al. 2011). Once the daily CI has been determined, the expert nutritionist can produce a daily meal plan for the patient.

In this study, we automate the process of meal plan generation, assuming that nutrition health assessment (i.e., identifying the target BFP/weight and the required CI) has already been conducted (either by a human nutritionist or by an automated agent). We address the task of automating nutrition health assessment in a dedicated study (Salloum G. and Tekli J. 2021).

2.2. Meal Plan Recommendation

A major service provided by a nutrition expert is translating the daily CI recommendation into an actual daily meal plan, where foods can be selected based on many factors, including: macronutrient composition, energy density, patient food preferences, food variety, and food compatibility, among others. This can be performed in three main steps:

i. *Determining the amount of macronutrients*: based on the daily CI, the expert determines in grams (g) the needed amount of daily macronutrients, which fall into three categories: carbohydrates, protein, and fat. The latter are the basic constituents of any food (Rolls B. 2009). The relationship between macronutrients and energy density can be described as follows: 1 gram of carbohydrates or protein contains 4 Kcals, and 1 gram of fat contains 9 Kcals (Rolls B. 2009). Based on the latter, CI can be transformed into grams of macronutrients using a set of mathematical formulas (Kathleen M. and Janice R. 2017).
ii. *Determining the number of food servings*: based on the needed amount of macronutrients, the expert determines the number of servings from five well-defined food categories[3] (starch, fruits, milk, vegetables, lean meat, and fats), where each food category contains food items sharing similar macronutrient contents based on adjusted serving sizes (Kathleen M. and Janice R. 2017). Each serving of a food has about the same amount of macronutrients and calories as the other foods in the same category. This allows selecting multiple alternative options from the list of foods in a certain category by determining the number of servings from the category as a whole (Valdez-Pena H. and Martinez-Alfaro H. 2003). Converting the grams of macronutrients into food servings can also be computed using a set of well-defined mathematical formulas (Kathleen M. and Janice R. 2017).

[2] Our solution supports any number of daily meals. In the current study, we adopt a 5-meal plan which is typically adopted in health nutrition literature.
[3] Adopted based on the *Diabetic Exchange List* suggested by the American Dietetic Association (Kathleen M. and Janice R., 2017).

iii. *Selecting the food items*: The final step in meal planning is to transform the number of servings from each category into an actual meal plan. Here, meal planning goes beyond providing the correct numbers of servings and macronutrients. Several factors can be considered by the human nutrition expert: (i) food preferences of the patient, (ii) compatibility between meal and foods (e.g., *eggs* are compatible with *breakfast*, whereas *meat* is compatible with *lunch*), (iii) inner compatibility between foods assigned to the same meal, to make sure that they go well together (e.g., *eggs* and *milk* are compatible together, versus *eggs* and *fish* which might be less compatible), and (iv) variety of foods from day to day. Different from the previous tasks, there are no mathematical formulas to perform food item selection. This multi-factor process is intuitively resolved by human experts who rely on "common sense" reasoning and expertise to assign the necessary amount of food servings over the five meals of the plan, while attempting to respect the factors mentioned above.

While various computerized solutions (cf. Section 3) have attempted to solve the different aspects of the meal planning problem, yet, to our knowledge, there is no well-established process to produce a full-fledged meal plan from scratch: based on a recommended CI, and performing food selection considering the different factors mentioned above.

3. Related Works

This section briefly describes computerized solutions related to nutrition health and meal planning, organized following their computational techniques: (i) linear optimization, (ii) meta-heuristics, (iii) case-based reasoning, and (iv) fuzzy reasoning. We also briefly describe (v) existing e-solutions.

3.1. Linear Optimization Methods

One of the earliest approaches to solve the meal planning problem was put forward in 1945 in an attempt to find the least cost meal by modeling the problem as a mathematical model and adopting a trial-and-error based approach (Stigler G. 1945). The approach produced a list of unstructured foods that meet standard nutrition requirements while optimizing food price. The latter solution was later extended and formalized into the so-called *Stigler* diet problem model (Lancaster L. 1992), which comes down to a linear programming problem aiming to minimize $\sum_{j=1}^{J} c_j \times x_j$ subject to $\sum_{j=1}^{J} n_{ij} \times x_j \geq b_i$ for i=1, 2, …, n, with $x_j \geq 0$ for all j, where c_j is the cost of the jth food, x_j is the quantity of the jth food, n_{ij} is the ith nutrient content of the jth food, and b_i is the ith nutrient requirement. In the past two decades, the *Stigler* model has been adopted and extended using different variations of the linear programming paradigm (e.g., integer programming (Lenstra H. 1983), mixed integer-linear programming (Schniederjans M. 1995), and goal programming (McCann-Rugg M. *et al.* 1983)) to solve the meal planning problem, e.g.,(Bassi L. 1976, Foytik J. 1981, Silberberg E. Nutrition and the Demand for Tastes, Valdez-Pena H. and Martinez-Alfaro H. 2003). In (Valdez-Pena H. and Martinez-Alfaro H. 2003), the authors use an exchange system for menu planning, offering the option of serving food substitutions from the same food group (e.g., for instance substituting 1/2 a cup of pasta with 1/3 cup of white rice, since they both represent a serving from the *starch* food category). The problem is modeled using mixed integer-linear programming with special focus on preference maximization. Yet, the approach in (Valdez-Pena H. and Martinez-Alfaro H. 2003) does not consider food-meal compatibility and inter-food compatibility, and does not compute food portion sizes. In (McCann-Rugg M., White G. et al. 1983), the authors place more emphasis on user food preferences while considering nutrition requirements (e.g., calories and macronutrients) and disregard other factors (e.g., inter-food compatibility and food-meal compatibility). The proposed approach does not associate foods into meal structures (e.g., breakfast, lunch, dinner), and rather presents a set of recommended foods in the form of an unstructured list of suggestions.

A common problem with linear optimization approaches is that adding more variable constraints (i.e., more restrictions) to the problem significantly reduces its feasibility/solvability (Schniederjans M. 1995). This makes it difficult to consider multiple factors (e.g., nutrition requirements, food preferences, variety, and compatibility) using this family of approaches.

3.2. Meta-heuristic Methods

A few meta-heuristic approaches, using genetic algorithms or particle swarm optimization techniques, have been developed to produce personalized meal plans, e.g., (Ainsworth B. *et al.* 2011, Fister D. and Fister I. 2016, Noor S., Mohd A. et al. 2018, Seljak B. 2009). Compared with linear optimization algorithms, meta-heuristics do not guarantee a globally optimal solution, but rather provide a sufficiently good solution to the problem (Bianchi L. *et al.* 2009, El-Ghazi T. 2009).

In (Noor S., Mohd A. et al. 2018), the authors introduce a self-adaptive hybrid genetic algorithm to perform menu planning for Malaysian adolescent school students aged between 13 and 18. The approach aims at producing daily structured meal plans to meet nutrient intake requirements while optimizing meal budget and food variety. Integer encoding is used to map the foods from each food category (e.g., fruits, vegetables, starch, etc.) to a chromosome associated with one of six meals (e.g., breakfast, first snack, lunch, etc.). The fitness function is evaluated to meet the budget constraint, where the total cost of the candidate solutions in the meal should not surpass the budget and the recommended nutrient intake. The produced meal plans were not empirically evaluated by nutrition experts. In (Gaál B. *et al.* 2005), the authors introduce a divide-and-conquer multi-level genetic algorithm to produce personalized weekly meal plans, aiming to satisfy macronutrient, user preference, and inter-food compatibility constraints. A global fitness function is defined as the sum of multiple individual functions evaluating each criterion. Empirical results show that the proposed solution can produce near-optimal meal plans after around 1000 iterations, requiring 10-to-15 minutes of processing time. A similar multi-level genetic algorithm is introduced in (Seljak B. 2009), considering economic cost and various aesthetic parameters (e.g., taste, color, temperature, and preparation method) to produce an *n*-day meal plan. Empirical results show that 250 iterations are needed to produce feasible solutions, requiring between several minutes and a couple of hours of processing time to generate a set of 21-day meal plans. In (Kaldirim E. and Kose Z. 2006), the authors introduce another multi-level genetic approach to generate meal plans by optimizing macronutrient fulfillment, user preferences, and food price. The proposed approach produces lists of recommended foods as feasible solutions, rather than producing a structured meal plan. In (Fister D. and Fister I. 2016), the authors introduce a particle swarm optimization solution to produce structured meal plans that can enhance an athlete's performance according to her/his exercise program. It accepts as input: (i) a description of the three-day exercise program of the athlete represented by the average heart rate and the duration of each exercise, which can be mathematically converted into the amount of burned calories, and (ii) the potential foods that can be assigned to each meal as well as the amount of calories in each food. The fitness function attempts to match the amount of calories burned with and the amount of calories contained in the meal. Caloric requirements are the only factor considered in the meal planning process, where other factors such as macronutrients, food preferences, variety, and compatibility are not considered. In a recent study in (Türkmenoglu C. et al., 2021), the authors model the diet problem as a many-objective multi-dimensional knapsack optimization problem. Given a set of available food items, the proposed solution aims at selecting a subset of the items allowing to optimize all objectives simultaneously, without exceeding the knapsack capacities. The authors utilize an existing popular many-objective evolutionary algorithm to solve the problem, i.e., non-dominated sorting genetic algorithm III (NSGA-III) (Deb K. and Jain H., 2014). They consider three objectives in their experiments: cost, preference, and preparation time, taking into account both upper-level and lower-level daily nutrient constraints for every meal. Experimental results highlight the quality of the recommended foods in evaluating energy and protein satisfaction, and the contribution of each food item to the objective functions. Note that the study in (Türkmenoglu C. et al., 2021) focuses on food-meal recommendation, and does not address the issues of producing meal plan structures and considering composite foods.

3.3. Case-based Reasoning Methods

Case-based reasoning methods, e.g., (Husain W. *et al.* 2011, Khan A. S. and Hoffmann A. 2003a, Khan A. S. and Hoffmann A. 2003b, Petot G. J., Marling C. et al. 1998), attempt to recommend meal plans by (i) identifying the best meal plan from a set of existing ones and then (ii) adapting and revising the plan to serve the target patient's needs.

In (Petot G. J., Marling C. et al. 1998), the authors consider minimum and maximum nutrient constraints and user food preferences to select the best meals from a pool of existing meal plans for similar patients. A rule-based pattern regulator mechanism is used to refine the selected meals when they do not properly fit their nutrient requirements. Nonetheless, the authors state that some of their generated plans contain odd food combinations and lack the human common sense factor. A similar approach is introduced in (Khan A. S. and Hoffmann A. 2003a, Khan A. S. and Hoffmann A. 2003b), combining case-based reasoning with a rule-based feedback mechanism to produce expert-tailored meal plans. The system accepts as input as a set of patient requirements (namely nutrient requirements, and special medical cases such as: liver mal function, constipation, etc.) and a set of possible solutions (i.e., candidate meal plans). A fuzzy scoring mechanism is used to score and rank the candidate meal plans based on the nutrient requirements of the patient. A domain expert is then tasked with selecting and modifying the "best" meal plan to better adapt to the patient's requirements, while providing an explanation for every modification. The explanations are represented as *if-else* rules, which are incrementally acquired and later used to generate meal plans for similar future patients. The study evaluates factors such as the number of actions needed and the number of rules needed to produce improved meal plans. For instance, 330 rules were added by the domain expert before the tool was able to produce its own meal plans, considering only two groups of similar patients (Khan A. S. and Hoffmann A. 2003a, Khan A. S. and Hoffmann A. 2003b). Furthermore, expert intervention is required to evaluate and determine the nutrient goals for every patient. In (Husain W., Wei L. et al. 2011), the authors combine case-based reasoning with a genetic algorithm. A database is set-up to acquire patient health information and their previously recommended meal plans. A meal plan generator mechanism is built using a typical genetic algorithm, where: (i) crossover is applied on the meal plans to exchange some of their foods, (ii) mutation is performed by stochastically altering the meal plan's food items, and (iii) selection is performed to identify the meals that fit best the user's nutrition requirements. The patient can choose to either accept or alter the suggested meal plan by substituting the recommended foods. Initial experiments show that the produced meal plans do not always meet the required nutrition constraints due to the fast convergence of the genetic algorithm. The authors suggest increasing the size of the patient database as a potential solution to the convergence problem. The proposed solution is mainly focused on fulfilling the patients' nutrition requirements and does not consider other factors such as food preferences, food variety, food-meal compatibility and inter-food compatibility.

3.4. Fuzzy Reasoning Methods

Various methods have been developed to infer the healthiness of meals and meal plans by combining fuzzy reasoning with food and patient profile ontologies, e.g., (Lee C. *et al.* 2009, Lee C.S. *et al.* 2010, Wang M. H. 2009). Most methods in this category address meal plan assessment of existing manually generated meal plans, rather than automated meal plan generation. They make use of the Fuzzy Logic (FL) paradigm (Lee C.S., Wang M.H. et al. 2010), as a robust solution designed to deal with uncertainties in real-life applications. For instance, the approach in (Wang M. H. 2009) receives as input: a meal suggested or consumed by a patient, and provides as output: a fuzzy score highlighting the likelihood of the meal being healthy or not (e.g., a meal plan can be considered as 0.3 *healthy* and 0.7 *unhealthy* simultaneously, instead of fully belonging to one single category solely). The authors in (Wang M. H. 2009) target Taiwanese foods, and utilize a dedicated food ontology describing meals, meal courses, and foods with their nutrition information. The latter feed into the fuzzy set definitions that are used in the FL inference mechanism. Experimental results on 20 recorded meals show high correlation ($\approx$ 90%) between human expert and system generated scores. The agent requires domain expert intervention to determine the caloric and nutrient requirements of each patient before the fuzzy decision making process can take place. The approach in (Wang M. H. 2009) is extended to include a type-2 fuzzy system[4] in a subsequent study (Lee C., Wang M. et al. 2009), without however showing major improvement over the type-1 fuzzy system approach, as stated by the authors. A similar study is introduced in (Lee C.S., Wang M.H. et al. 2010) to perform meal plan assessment for patients suffering from diabetes. It produces as output a fuzzy membership score associated with a linguistic variable (e.g., "*healthy*", "*very healthy*", "*not healthy*", etc.) to assess the healthiness of a given input meal w.r.t.[5] a target patient. Nonetheless, the proposed solution does not provide any decision making regarding the ideal energy requirements that patients should aspire to: promoting either weight loss or weight gain based on their needs. In (Lee C. *et al.* 2012, Lee C. and Lan S. 2015), the authors integrate evolutionary computation with fuzzy processing, attempting to learn the food ontology's fuzzy sets and fuzzy inference rules by mapping them into a chromosome representation and then applying genetic evolution (via dedicated crossing, mutation, and selection operators) to determine the desired output for each case. While the genetic process allows to automate part of the fuzzy model generation process, yet domain expert involvement is required in the selection phase, and in determining the caloric and nutrient requirements of the participants, as stated by the authors.

Different from previous fuzzy methods which only perform meal plan assessment, the authors in (Lee C. *et al.* 2010, Lee C. *et al.* 2008) introduce a solution that accepts as input the foods consumed in the breakfast and lunch meals of a diabetes patient, and infers as output the remaining caloric allowance and the needed food servings for the dinner meal. Using the fuzzy food and personal profile ontologies from (Lee C., Wang M. et al. 2012), the solution suggests multiple food options from each food category to fit the servings' requirements for dinner, without however producing a complete dinner meal. In other words, it does not combine foods to form a complete meal, but rather suggests food options to help the patient manually produce the final dinner meal.

3.5. Existing E-Solutions

Various nutrition and health related mobile and Web applications have been developed recently and are becoming increasingly available online. Calorie tracking tools, e.g., (Evans D. 2017, Livestrong Foundation 2020, MyNetDiary Inc. 2020, SparkPeople Inc. 2020), assist patients in monitoring their daily caloric intake (CI) and consumed macronutrients by accepting as input the patient's consumed foods, and producing/calculating as output the amount of calories and macronutrients contained in the consumed foods. Meal planning tools, e.g., (EatThisMuch Inc. 2020, Fitness Meal Planner 2020, MakeMyPlate Inc. 2020, Yang L. *et al.* 2017), generate daily meal plans based on patient provided CI requirements. One such tool is MakeMyPlate (MakeMyPlate Inc. 2020), a mobile application that provides patients with daily pre-defined meal plans fulfilling user specified CI levels. It allows the patient to replace a meal with an existing meal stored in the database, without verifying whether the replacement meal is calorically equivalent to the original one (which might result in surpassing or dropping below the recommended CI and macronutrient amounts). The tool does not consider the patient's food preferences. Another solution is EatThisMuch (EatThisMuch Inc. 2020), which accepts as input the patient's basic health state information (e.g., gender, age, height, weight, and level of activity) in addition to the BFP. It also accepts as input the user's destination weight in textual from (i.e., *maintain weight*, *lose weight*, *gain weight*, and *gain muscle*), the user's preferred diet type (e.g., mediterranean, vegetarian), as well as her food preferences (i.e., whether the user wants a food item to appear or not in the daily mean plans). The application then produces as output daily meal plans. While powerful, yet this solution has a few limitations, namely: (i) allowing the user to generate meal plans for the current day only (planning ahead required premium subscription), and (ii) considering user preferences in an "*include*" or "*do not include*" crisp fashion (rather than allowing a gradient rating of food preferences, e.g., $\in [0, 1]$ from "*not preferable*"–to–"*highly preferable*", which would be more useful in producing adapted meal plans). Fitness Meal Planner (Fitness Meal Planner 2020) is yet another online application sharing most of the functionality and limitations of the latter solution. The authors in (Yang L., Hsieh C.K. et al. 2017) describe an online framework to monitor foods consumed by the patient to help perform personalized meal planning, using food image recognition through machine learning. The system learns patient preferences by allowing them to select their favorite foods by uploading pictures of them. Then, image analysis is performed through a dedicated convolutional neural network to recognize the foods in the picture, allowing the system to later recommend similar foods from a pre-defined food database. The aim of the study is to improve the recommendations of survey-based systems (where food preferences are learned through manual patient surveys) by using image analysis-based preference learning. Nonetheless, the study in (Yang L., Hsieh C.K. et al. 2017) does not produce full-fledged meal plans that meet the CI requirements of a patient.

3.6. Discussion

To sum up, most meal planning solutions lack a completely automated process and require domain expert intervention at different stages of the recommendation process (e.g., identifying macronutrient distribution, providing pre-defined meal plans for case-based reasoning, or combining recommended foods into meal structures). Also, most solutions focus on fulfilling the patients' nutrition

[4] Type-2 FL is an extension of the original FL paradigm, referred to as type-1 FL, where every truth degree has an uncertainty degree associated with it (e.g., a person is considered 0.3 underweight with 0.9 certainty, i.e., we are 90% certain that the person is 30% overweight). If there is no uncertainty, then a type-2 fuzzy set is reduced to a type-1 fuzzy set (Karnik N. and Mendel J., 2001).

[5] With respect to

requirements (in terms of caloric intake (CI) and macronutrients) while disregarding other relevant factors such as patient food preferences, food variety, food-meal compatibility, as well as inter-food compatibility. A most recent study in (Türkmenoglu C. et al., 2021) does consider many factors including food item cost, preference, and preparation time, by modelling the diet problem as a many-objective optimization problem, yet the authors focus on food-meal recommendation and do not address the issues of producing meal plan structures and considering composite foods.

4. Proposal

In this study, we introduce an intelligent agent titled Meal Plan Generator (*MPG*) which aims at automating the meal plan recommendation services offered by a nutrition expert while addressing the limitations of existing e-solutions mentioned above. *MPG*'s overall architecture in shown in Figure 1, and consists of five main components. First, the *Macronutrients Calculator* computes the amount of macronutrients (i.e., carbohydrates, protein, and fat) in grams, based on the daily CI recommendation (provided by a human nutritionist or a dedicated automated agent (Salloum G. and Tekli J, 2021)). Second, the *Servings Calculator* computes the daily amount of servings for each of the six primary food categories (i.e., starch, fruits, milk, vegetables, lean meat, and fats)[6] based on the amounts of macronutrients produced by the macronutrient calculator. Third, the *Servings Assignor* splits the servings for each of the six primary food categories over the five daily meals (i.e., breakfast, snack one, lunch, snack two, dinner) based on sample meal plan serving assignments. Fourth, the *Food Assignor* allocates the foods to the five daily meals while meeting all the serving requirements and considering the patient's preferences. Finally, the *Meal Plan Evaluator* computes the relevance scores for the generated meal plans highlighting their compliance with the patient chosen factors (e.g., assigning different weights for food preferences, variety, and compatibility). While the first two components (performing macronutrient and serving calculation) are evaluated mathematically following well-established procedures from nutrition health literature (cf. Section 2), nonetheless, there is a lack of automated solutions to handle the last three components (performing servings assignment, food assignment, and relevance scoring). We solve the latter by introducing an adapted version of the transportation optimization problem, along with a set of relevance scoring functions specifically designed to solve and evaluate the meal planning task. We further describe each of the above components in the following sub-sections.

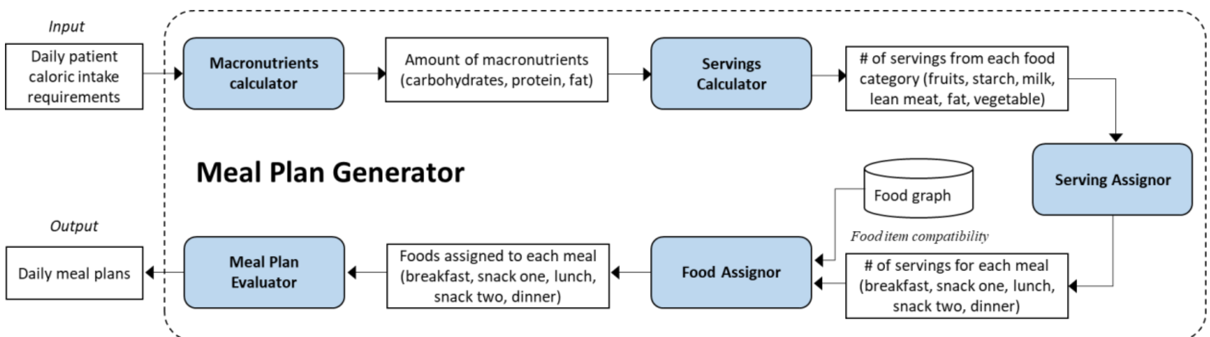

Figure 1. Simplified diagram describing *MPG*'s overall architecture

Note that *MPG* is developed as part of a comprehensive framework titled Personal Intelligent Nutritionist (*PIN*) (Salloum G. and Tekli J, 2021) which aims at automating the full nutrition heal assessment and recommendation pipeline, including: (i) *weight assessment and recommendation* based on various inputs (age, gender, height, weight, and BFP), and then recommending a target weight and BFP for the patient; (ii) *CI and exercise recommendation* based on the level of activity as well as the target weight and BFP of the patient; (iii) *progress evaluation and recommendation adjustment*, especially when the patient is not making the expected progress; leading to (iv) *meal plan generation* through *MPG*. This paper describes *MPG*, while *PIN*'s remaining modules are developed in (Salloum G. and Tekli J., 2021).

4.1. *Macronutrients Calculator* component

A normal diet translates the caloric intake (CI) into grams of macronutrients as follows. First, the CI's calories are distributed among the three macronutrient categories: (i) 45% to 55% to carbohydrates, 15% to 20% to protein, and 20% to 30% to fat (Kathleen M. and Janice R. 2017). In this study, we adopt the following commonly used percentage classification based on nutrition expert recommendations: 50% carbohydrates, 20% protein, and 30% fat. Second, the required amount of grams of each macronutrient is computed as follows (Kathleen M. and Janice R. 2017):

$$grams_{carbohydrates} = \frac{calories \times 50}{100 \times 4} \qquad grams_{protein} = \frac{calories \times 20}{100 \times 4} \qquad grams_{fat} = \frac{calories \times 30}{100 \times 9} \tag{3}$$

[6] Adopted based on the *Diabetic Exchange List* suggested by the American Dietetic Association (Kathleen M. and Janice R., 2017).

Example: Consider for instance a recommendation of CI=2107 Kcals. Following the above described process and formulas, the *Macronutrients Calculator* component produces the following macronutrient gram distribution:

$$grams_{carbohydrates} = \frac{2107 * 50}{100 * 4} = 263\ grams \qquad grams_{protein} = \frac{2107 * 20}{100 * 4} = 105\ grams \qquad grams_{fat} = \frac{2107 * 30}{100 * 9} = 70\ grams$$

4.2. *Servings Calculator* component

In this study, we adopt a widely utilized approach for servings calculation based on the exchange list system for diabetic meal planning (Kathleen M. and Janice R. 2017, Valdez-Pena H. and Martinez-Alfaro H. 2003). The exchange list organizes foods in categories, where each category groups food items which share similar nutrient contents based on adjusted serving sizes. Each serving of a food has about the same amount of carbohydrates, protein, fat, and calories as the other foods in the same category. Here, we adopt the six basic food categories shown in Table 1. By determining the number of servings from a specific category, multiple options can be selected from the list of foods available for that category.

Table 1. Food exchange categories adopted in MPG and their calorific and macronutrient properties per serving

Food Categories		Carbohydrates (grams)	Protein (grams)	Fat (grams)	Calories (Kcals)
cat_1	**Starch** (Bread, cereals, etc.)	15	0-3	0-1	80
cat_2	**Fruits**	15	-	-	60
cat_3	**Milk** (low-fat)	12	8	0-3	100
cat_4	**Non-starchy Vegetables**	5	2	-	25
cat_5	**Meat** (lean)	-	7	0-3	45
cat_6	**Fats**	-	-	5	45

In addition to basic foods (e.g., bread, beans, milk), we consider *composite foods* (e.g., grilled chicken, pizza, burger sandwich), consisting of dishes defined as combinations of different servings from different categories. For example, a grilled chicken sandwich is the equivalent of 3 servings of carbohydrates (starch) and 4 servings of lean meat.

The second step of the meal planning process consists in converting the grams of macronutrients into servings from each category. We compute the amount of servings using a process adapted from (Valdez-Pena H. and Martinez-Alfaro H. 2003):

i. Select one serving of milk.
ii. For CI below or equal to 2200 Kcals, select 3 servings of fruits and 3 servings of vegetables.
For CI above 2200 Kcals, fruit servings are increased from 3 to 4 and vegetable servings are increased from 3 to 5.
iii. Calculate the number of starch servings by subtracting the amount of carbohydrates obtained from milk, vegetables, and fruits, from the total required amount of carbohydrates. The amount of carbohydrates calculated is divided by 15 since each serving of starch contains 15 grams of carbohydrates (cf. Formula 4).
iv. Calculate the number of meat servings by subtracting the amount of protein from milk, vegetables, fruits, and starch servings, from the total required amount of protein. Fruits contain zero grams of protein. The amount of protein calculated is divided by 7 since each serving of meat contains 7 grams of protein (cf. Formula 5).
v. Finally, calculate the number of fat servings in the same fashion: based on the number of grams of fat in the pre-selected servings. Note that fruits and vegetables contain zero servings of fat. The amount of fat calculated is divided by 5 since each serving from the fat food category contains 5 grams of fat (cf. Formula 6).

$$s_{starch} = \frac{g_{carbohydrates-(12\times s_{milk} + 5\times s_{vegetable} + 15\times s_{fruit})}}{15} \qquad \textbf{(4)}$$

$$s_{meat} = \frac{g_{protein-(8\times s_{milk} + 5\times s_{vegetable} + 3\times s_{starch})}}{7} \qquad \textbf{(5)}$$

$$s_{fat} = \frac{g_{fat-(1\times s_{milk} + 2\times s_{meat} + 2\times s_{starch})}}{5} \qquad \textbf{(6)}$$

where *s* is the number of servings from a food category and *g* the amount of grams of a macronutrient.

Running example: Consider the same example from the previous section, where CI = 2107 Kcals was provided as input to the *Macronutrient Calculator*, resulting in the following macronutrient assignments: 263 grams of carbohydrates, 105 grams of protein, and 70 grams of fat. The latter are provided as input to the *Servings Calculator*, which produces the following numbers of servings per food category: s_{milk}= 1m s_{fruit} = 3, $s_{vegetable}$ = 3, $s_{starch} = \frac{263-12*1+5*3+15*3}{15}$ = 12.73=13, $s_{meat} = \frac{105-8*1+5*3+3*13}{7}$ = 7.42 = 7, and s_{fat} = $\frac{70-1*1+2*7+2*13}{5}$ = 6.

4.3. *Servings Assignor* component

This component distributes the number of servings from each food category over the daily meals. In this study, we adopt a five-meal approach: three main meals (i.e., *breakfast*, *lunch*, *dinner*) and two snacks in between (i.e., morning snack referred to as *snack one*, and afternoon snack referred to as *snack two*). Given the lack of well-established processes for assigning food category servings to daily meal servings, we adopt a template-based approach: where a *food category | daily meal* mapping template is automatically generated by aggregating a set of representative sample assignments provided by nutrition experts. The number of servings can be aggregated in several ways, using for instance the *maximum*, *minimum*, *average* or *weighted sum* aggregation functions. In our study, we make use of the *weighted sum* function since it enables the user to choose the weight of each sample mapping in accordance with her/his notion of mapping relevance (and comes down to the *average* function when no user preferences are provided). For each of the N samples provided by the nutrition expert, the aggregate number of servings $s_{Cat_i}^{Meal_j}$ for each food category cat_i and daily meal $meal_j$ is computed as follows:

$$s_{Cat_i}^{Meal_j} = \underset{n=1 \ldots N}{f_{Agg}}\left(s_{cat_i^n}^{meal_j^n} \right) = \sum_{n=1 \ldots N} w_n \times s_{Cat_i^n}^{Meal_j^n} \quad \textbf{(7)}$$

$$\text{where } \sum_{n=1 \ldots N} w_n = 1 \text{ and } w_{n=1 \ldots N} \in [0, 1]$$

Consequently, the template's values are proportionally scaled to the CI at hand producing the corresponding *food category | daily meal* servings matrix.

Running example: Consider the same example from the previous sections, where an input CI = 2107 kcals resulted in 1, 3, 3, 13, 7, and 6 servings of milk, fruits, vegetables, starch, meat, and fat respectively (cf. Section 4.2). By considering the *food category | daily meal* template provided in Table 2.a and after scaling it to the current example, the *Servings Assignor* component produces the servings plan shown in Table 2.b. We adopt linear scaling by multiplying the template's serving distribution percentages by the total number of required servings and rounding the fractions. For instance, following the 2000 Kcal template in Table 2.a, the serving distribution for starch over the five meals (i.e., *breakfast*, *snack one*, *lunch*, *snack two*, and *dinner*) is 3, 2, 3, 2, and 2 respectively, amounting to a total number of 12 starch servings. This results in the following serving distribution percentages over the five meals: 25%, 16%, 25%, 16%, and 16%. These percentages are multiplied by the total number of required starch servings =13 for the CI = 2107 kcals example, resulting in the following serving distribution for the five meals: 3.25, 2.08. 3.25, 2.08, and 2.08. These serving numbers are rounded to 3, 2, 3, 2, and 2, and a post calculation correction is performed to supply the remaining required serving fractions: 0.25+0.08+0.08+0.25+0.08=0.74, rounded to 1, by randomly sorting the meals and adding one serving per meal until the additional serving fractions are supplied. In this example, the additional 1 serving is added to the *snack two* meal, resulting in the final 3, 2, 3, 3, 2 serving distribution for starch.

Table 2. *Food category | Daily Meal* mapping example

a. Template for 2000 Kcal CI servings plan

Food Category		Total Servings	Breakfast	Snack one	Lunch	Snack two	Dinner
cat_1	***Milk***	1	1	0	0	0	0
cat_2	***Fruit***	3	0	1	0	2	0
cat_3	***Vegetable***	3	1	0	1	0	1
cat_4	***Starch***	12	3	2	3	2	2
cat_5	***Meat***	7	1	0	4	0	2
cat_6	***Fat***	5	1	1	2	0	1

b. Scaled servings plan for running example with CI = 2107 Kcals

Total Servings	Breakfast	Snack one	Lunch	Snack two	Dinner
1	1	0	0	0	0
3	0	1	0	2	0
3	1	0	1	0	1
13	3	2	3	3	2
7	1	0	4	0	2
6	2	1	2	0	1

4.4. *Food Assignor* component

The final and central step in meal planning is to transform the number of servings into an actual meal plan, by assigning them actual foods based on the required servings. Meal planning however goes beyond providing the numbers correctly. Important "logical" factors need to be considered, including: (i) the patient's food preferences (regarding what foods are preferred and what foods are not pleasurable), (ii) the food's compatibility with the meal (e.g., eggs and milk are compatible with breakfast, whereas fish is more compatible with lunch), (iii) the inter compatibility between foods assigned to the same meal (what foods go well with each other based on taste and appearance), and (iv) the variety of the foods recommended from day to day. Other factors could be considered based on the patient or on the nutrition expert's needs. Given the lack of well-established processes to solve this problem, we model the food assignment process as an adapted transportation optimization problem that fits all mentioned requirements, while accounting for multiple supply and demand types (i.e., the different food categories) as well as the different factors that play into the meal planning task.

4.4.1. Adapted Transportation Problem

In this sub-section, we present a modification of the transportation problem that fits the requirements of our meal plan generation problem[7]. The transportation problem is concerned with finding the minimum cost of transporting a single commodity from a given number of sources to a given number of destinations. The data required by the model include: (i) the amount of supply at each source and the amount of demand at each destination, as well as (ii) the unit transportation cost of the commodity from each source to each destination (Hira D. and Gupta P. 2014, Winston W. and Venkataramanan M. 2003). In our study, we model our adaptation of the transportation problem as follows (cf. Table 3):

i. We consider *m* different supply centers (sources) labeled with $food_{i=1..m}$, where every supply center designates an available food from the food categories.
ii. We consider n=5 different demand centers (destinations) labeled $meal_{j=1..n}$, where every demand center designates one of the 5 meals considered in our study (i.e., *breakfast*, *snack one*, *lunch*, *snack two*, and *dinner*).
iii. The demand required at each demand center $meal_j$ represents the number of servings form each food category required in the meal. This is modelled as a 6-dimentional vector, noted $\overrightarrow{D_j}$ corresponding to each of the 6 categories of basic foods considered in our study (i.e., starch, fruits, milk, vegetables, lean meat, and fat)[8].
iv. The supply capacity of each supply center $food_i$ represents the available amount of servings from the corresponding food. Since demand is modeled as a vector of serving requirements, supply is also modeled as a 6-dimensional vector representing the number of servings from each food category that $food_i$ is composed from, noted $\overrightarrow{S_i}$, multiplied by the available amount of servings of that food, noted s_i. We represent by $x_{(i,j)}$ the number of servings supplied from supply center $food_i$ to demand center $meal_j$.
v. The cost function, associating a cost value with every transportation operation, is defined as an extensible aggregation function that combines the different cost factors considered in our meal planning scenario (including, *patient preferences*, *meal-food compatibility*, *inter-food compatibility*, *food variety*, and *price*). We represent by $c_{(i,j)}$ the cost associated with delivering the servings from supply center $food_i$ to demand center $meal_j$.

Table 3. Food assignment transportation matrix

		Cost per Food Distributed				**Supply**
		Meal				
		1	2	…	n	
Food	1	$c_{(1,1)}$	$c_{(1,2)}$	…	$c_{(1,n)}$	$s_1 \times \overrightarrow{S_1}$
	2	$c_{(2,1)}$	$c_{2,2}$	…	$c_{(2,n)}$	$s_2 \times \overrightarrow{S_2}$
	…	…	…	…	….	…
	m	$c_{(n,1)}$	…	….	$c_{(m,n)}$	$s_m \times \overrightarrow{S_m}$
Demand		$\overrightarrow{D_1}$	$\overrightarrow{D_2}$	…	$\overrightarrow{D_n}$	

4.4.2. Demand and Supply Vectors

As previously described, each meal in our adapted transportation problem serves as a demand center. Following the food exchange list system adopted in our study (cf. Section 4.2), a meal has separate requirements for each of the six basic food categories noted cat_1-to-cat_6. In order to account for this multi-dimensional requirement[9], we model demand as a vector of serving requirements:

$$\vec{D} = (d_1, d_2, d_3, d_4, d_5, d_6) \quad \textbf{(8)}$$

where d_1-to-d_6 represent the requirements of the six food categories cat_1-to-cat_6 respectively[10].

[7] An early version of *MPG*'s transportation optimization solution is mentioned in (Salloum G. et al., 2018), where the authors consider: i) basic food items only (the current study introduces a new model integrating both basic and composite foods, cf. Section 4.4), ii) the traditional transportation problem only (the present paper introduces a new multi-factor adaptation of the transportation problem, specifically designed to handle composite foods, cf. Section 4), iii) pre-defined static food-meal cost values defined by experts (the present paper introduces a dynamic approach consisting of a battery of novel mathematical cost functions – cf. Section 4.4.3 – and meal plan evaluation functions – cf. Section 4.5).
[8] Compared with typical transportation problem formulations where demands are represented as 1-dimentional scalar values.
[9] Compared with traditional transportation problem formulations where every demand center has one single requirement from a given supply center.
[10] In the formula, we represent the vector as its transpose for ease of presentation.

Running example: Considering the serving plan presented in Table 2, the corresponding demand vectors for every meal are defined as follows: $\overrightarrow{D_{Breakfast}} = (1,0,1,3,1,2)$, $\overrightarrow{D_{SnackOne}} = (0,1,0,2,0,1)$, $\overrightarrow{D_{Lunch}} = (0,0,1,3,4,2)$, $\overrightarrow{D_{SnackTwo}} = (0,2,0,3,0,0)$, and $\overrightarrow{D_{Dinner}} = (0,0,1,2,2,1)$. A demand is considered met once all the requirements in the demand vector are met.

In addition, each food serves as a supply center. Here, we distinguish between: (i) basic foods which can be mapped 1-to-1 with the six food categories (cat_1-to-cat_6) mentioned above, and (ii) composite foods which consist of combinations of basic foods. Foods, both basic and composite, will serve as supply centers aiming to fulfill the servings of each meal. Hence, we model supply as a vector of available servings as follows:

$$\vec{S} = (s_1, s_2, s_3, s_4, s_5, s_6) \quad \textbf{(9)}$$

where s_1-to- s_6 represent the supply from each of the food categories, cat_1-to-cat_6, respectively. Note that a basic food will only supply the category it belongs to, whereas a composite food can supply many categories simultaneously (based on the nature of the food and its constituents). For example, considering basic foods on the one hand: one serving of *yogurt* will supply category *milk* (i.e., cat_1), and one serving of *chicken* will supply category *meat* (i.e., cat_5), with the following supply vectors: $\overrightarrow{S_{milk}} = (1,0,0,0,0,0)$ and $\overrightarrow{S_{chicken}} = (0,0,0,0,1,0)$ respectively. On the other hand, *chicken sandwich*, which is a composite food, is equivalent to 3 servings of starch (i.e., cat_4) and 4 servings of meat (i.e., cat_5) and is represented as: $\overrightarrow{S_{ChikenSandwich}} = (0,0,0,3,4,0)$.

As for the supply amount, it is determined based on two factors: (i) the recommended amount of servings to be consumed per day, and (ii) the available food stock. We define supply as follows:

$$s = \begin{Bmatrix} Recommended & \text{if } Recommended < InStock \\ InStock & \text{otherwise} \end{Bmatrix} \quad \textbf{(10)}$$

This is necessary to make sure that we select the amount of servings from the available stock without surpassing the recommended amounts. Hence, the supply vector is weighted by supply amount *s* as follows:

$$\vec{S} = s \times (s_1, s_2, s_3, s_4, s_5, s_6) \quad \textbf{(11)}$$

4.4.3. Cost Function

The transportation matrix is a cost-based matrix, where each supply center (food) is related to each demand center (meal) through the cost of supplying the demand center (meal) from the supply center (food). Here, we define *cost* as *the inverse of the likelihood of a food being associated with a meal*; meaning the lower the cost the more likely the food will be assigned to a meal, and vice versa. This is inspired from human experts' thought process, who tend to consider foods that are more likely to be assigned to a meal, based on an overall likelihood estimation considering different factors. More formally, we define the cost function for delivering servings from supply center $food_i$ to demand center $meal_j$ as an extensible weighted sum of multiple cost factors:

$$c_{Total}(\text{i}, \text{j}) = \sum_{r=1}^{4} w_r \times c_r(\text{i}, \text{j}) \quad \in [0,1] \quad \textbf{(12)}$$

where *i* and *j* respectively represent $food_i$ and $meal_j$'s indices in the transportation matrix, c_r is the cost associated with each of the following 5 factors considered in our current study (more factors can be later added following the user's needs): (i) food preference, (ii) food occurrence, (iii) food-meal compatibility, and (iv) inter-food compatibility; and w_r is the weight assigned to each cost factor, such that $\forall\, w_r \in [0, 1]$ and $\sum_{r=1}^{4} w_r = 1$. We describe each of the cost factors below.

Food preference cost: This cost simulates the human decision-making process in choosing foods based on patient preferences. Each food is associated with a preference cost determined by the patient, using linguistic qualifies: e.g., *likes*, *extremely likes*, or *dislikes* (which are easier for the patient to provide, rather than providing scalar inputs). Here, we adopt five linguistic qualifiers (more qualifiers can be added to increase cost granularity) and heuristically convert them to normalized numeric values ∈ [0, 1]:

$$\text{c}_1(i,j) = \text{c}_{\text{FoodPref}}(food_i) = \begin{Bmatrix} 0 & \text{if patient } \textit{extremely likes} \text{ food} \\ 0.25 & \text{if patient } \textit{likes} \text{ food} \\ 0.5 & \text{if patient is } \textit{neutral} \text{ toward food} \\ 0.75 & \text{if patient } \textit{dislikes} \text{ food} \\ 1 & \text{if patient } \textit{extremely dislikes} \text{ food} \end{Bmatrix} \in [0,1] \quad \textbf{(13)}$$

Note that we use the same numerical scale/granularity $\in \{0, 0.25, 0.5, 0.75, 1\}$ in defining all cost factors for fairness of comparison. Yet any other numerical scale can be adopted as needed.

Food occurrence cost: The occurrence cost factor is included to avoid repetitive occurrences of foods, which is one of the main factors required for healthy meal planning. This is achieved by increasing the costs of foods that are repetitively selected using a certain scaling factor. After reaching high enough cost values (or the maximum cost =1) such that the food item is no longer selected, the cost is reset to 0 to favor its re-selection in the following iteration. More formally:

$$\mathrm{c}_2(i,j) = \mathrm{c}_{\mathrm{FoodOcc}}(food_i) \ = Nb_{occ}(food_i) \times \alpha = \begin{Bmatrix} 0 & \text{if occurrence is } \textit{very likely} \\ 0.25 & \text{if occurrence is } \textit{likely} \\ 0.5 & \text{if occurrence is } \textit{somehow likely} \\ 0.75 & \text{if occurrence is } \textit{unlikely} \\ 1 & \text{if occurrence is } \textit{very unlikely} \end{Bmatrix} \in [0, 1] \tag{14}$$

where Nb_{occ} represents the number of consecutive daily occurrences of $food_i$, and α its scaling factor. We heuristically choose α's initial value to be = 0.25 and bound the maximum number of consecutive occurrences at 4, in order to produce occurrence costs $\in \{0, 0.25, 0.5, 0.75, 1\}$ such that Nb_{occ}=1 produces cost = 0.25, Nb_{occ}=2 produces cost =0.5, Nb_{occ}=3 produces cost =0.75 and Nb_{occ}=4 produces maximum cost =1, following the same numerical scale adopted in defining the other cost factors. Users can choose to increase/decrease the scaling factor and the maximum bound following their preferences.

Food-meal compatibility cost*:* Each food is associated a food-meal compatibility cost, determined by expert nutritionists to express the food's best fit in the meal planning task. Similar to the food preference cost, we adopt common sense linguistic qualifiers in describing food-meal compatibility as follows:

$$\mathrm{c}_3(i,j) = \ \mathrm{c}_{\mathrm{FoodMeal}}(food_i, meal_j) \ = \begin{Bmatrix} 0 & \text{if compatibility is } \textit{very likely} \\ 0.25 & \text{if compatibility is } \textit{likely} \\ 0.5 & \text{if compatibility is } \textit{somehow likely} \\ 0.75 & \text{if compatibility is } \textit{unlikely} \\ 1 & \text{if compatibility is } \textit{very unlikely} \end{Bmatrix} \in [0, 1] \tag{15}$$

Numerical values are heuristically defined in accord with the previous two cost scores defined above, and are normalized $\in$ [0, 1].

Inter-food compatibility cost: This cost parameter designates the compatibility between foods assigned to the same meal. Given a food item $food_i$ that is considered for inclusion in $meal_j$, we define the inter-food compatibility cost of $food_i$ in $meal_j$ as the average compatibility cost relating $food_i$ with all food items $food_{i'=1, 2, 3, \ldots, k}$ already assigned to $meal_j$:

$$c_4(i,j) = c_{InterFood}(food_i, meal_j) = \sum_{i'=1}^{k} \frac{c_{InterFood}(food_i, food_{i'})}{k} \quad \in [0, 1] \tag{16}$$

Here, the compatibility cost factor between two individual food items, $c_{Inter\text{-}Food}(food_i, food_{i'})$, represents the distance separating the foods in a certain referential space. To compute the latter, we build a dedicated food graph connecting foods following their *direct compatibility* relationship identified by nutrition experts[11] (cf. Figure 2). If food items are directly connected in the graph (inter-distance =1 edge), it means they are directly compatible together. And if the foods are not directly connected, we navigate the graph to identify the shortest distance between them highlighting their (indirect) compatibility.

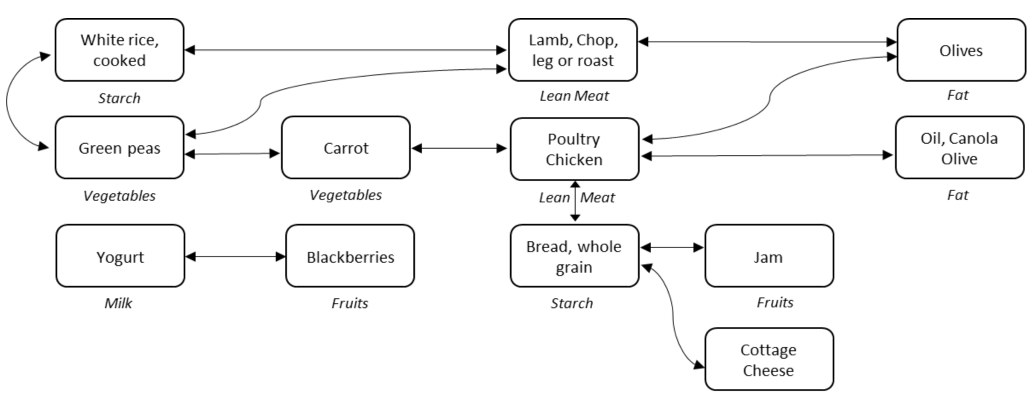

Figure 2. Extract of our food compatibility graph (the complete graph is provided in (Salloum G. and Tekli J. 2020))

[11] The food compatibility graph was developed with the help of Dr. Maya Bassil (Associate Professor of Human Nutrition in the Department of Natural Sciences at LAU) and Ms. Eva-Maria Kahwaji (M.Sc. in Sports and Exercise Nutrition at Loughborough University).

The distance between two food nodes in the graph is identified using an adaptation of the *Dijskstra* shortest path computation algorithm (Cormen T.H. *et al.* 2009). We multiply distance by a scaling factor β to reduce it to the same scale adopted by the other cost factors. More formally, the inter-food compatibility cost between two individual food items is defined as:

$$c_{Inter-Food}(food_i, food_{i'}) = (Dist(i, i') - 1) \times \beta = \left\{\begin{array}{ll} 0 & \text{if compatibility is } \textit{very likely} \\ 0.25 & \text{if compatibility is } \textit{likely} \\ 0.5 & \text{if compatibility is } \textit{somehow likely} \\ 0.75 & \text{if compatibility is } \textit{unlikely} \\ 1 & \text{if compatibility is } \textit{very unlikely} \end{array}\right\} \in [0, 1] \qquad \textbf{(17)}$$

where *i* and *i'* represent the graph node identifiers of the two food items being compared, and β is the distance scaling factor. We heuristically choose β's initial value to be = 0.25 and bound the maximum distance at 5 (i.e., any distance >5 is reduced to 5), in order to produce occurrence costs $\in$ {0, 0.25, 0.5, 0.75, 1} such that $Dist(i, i')$=1 (comparing a food with its direct neighbor[12]) produces minimal cost 0, $Dist(i, i')$=2 produces cost =0.25, $Dist(i, i')$=3 produces cost =0.5, $Dist(i, i')$=4 produces cost =0.75, and $Dist(i, i') \geq 5$ produces maximum cost =1, following the same numerical scale adopted in defining the other cost factors. Users can choose to increase/decrease the scaling factor and the maximum bound following their preferences.

Note that every time a food is assigned to a meal, the corresponding cost values in the transportation matrix are updated automatically based on the already assigned foods. In other words, the cost values are updated at each iteration of the transportation problem[13]. This simulates the human expert's thought process: assigning foods while trying to match the next selected food with the already assigned ones.

Example: Consider two food items from the *meat* category: *cottage cheese* and *poultry chicken*; and two meals: *breakfast* and *lunch*. Also, consider that the *carrot* food item from category *vegetable* is already assigned to the *lunch* meal while no foods are assigned to *breakfast* yet. The detailed and total cost factor values for both *cottage cheese* and *poultry chicken* are shown in Table 4, and are described below. Consider the *cottage cheese* food item with:

$c_1 = c_{FoodPref}$(*cottage cheese*) = 0.5 (i.e., patient is neutral toward *cottage cheese*)

$c_2 = c_{FoodOcc}$ (*cottage cheese*) = 0 × 0.25 (i.e., no previous occurrence the day before)

$c_3 = c_{Food-Meal}$(*cottage cheese*, *breakfast*) = 0.25 (i.e., likely) and $c_{Food-Meal}$(*cottage cheese*, *lunch*) = 0.5 (i.e., somehow likely)

$c_4 = c_{Inter-Food}$(*cottage cheese*, *breakfast*) = 1 (i.e., *breakfast* does not include any foods yet) and $c_{Inter-Food}$(*cottage cheese*, *lunch*) = $c_{Inter-Food}$(*cottage cheese*, *carrot*) = (3-1)*0.25 = 0.5 (i.e., somehow likely – since *cottage cheese* has a distance of 3 from *carrot* following our food compatibility graph, cf. Figure 2)

Consider the *poultry chicken* food item with:

$c_1 = c_{FoodPref}$(*poultry chicken*) = 0.25 (i.e., patient likes *poultry chicken*)

$c_2 = c_{FoodOcc}$ (*poultry chicken*) = 1 × 0.25 (i.e., one previous occurrence)

$c_3 = c_{Food-Meal}$(*poultry chicken*, *breakfast*) = 1 (i.e., very unlikely) and $c_{Food-Meal}$(*poultry chicken*, *lunch*) = 0 (i.e., very likely)

$c_4 = c_{Inter-Food}$(*poultry chicken*, *breakfast*) = 1 (i.e., since *breakfast* does not include any foods yet) and $c_{Inter-Food}$(*poultry chicken*, *lunch*) = $c_{Inter-Food}$(*poultry chicken*, *carrot*) = (1-1)*0.25= 0 (i.e., very likely – since *lunch* includes *carrots* already, and *poultry chicken* is at distance of 1 from *carrots* following our food compatibility graph, cf. Figure 2)

Table 4. Food assignment example

	Costs of Breakfast Meal					Costs Of Lunch Meal				
	c_1	c_2	c_3	c_4	A_{verage}	c_1	c_2	c_3	c_4	A_{verage}
Cottage Cheese	0.5	0	0.25	1	0.4375	0.5	0	0.5	0.5	0.375
Poultry Chicken	0.25	0.25	1	1	0.625	0.25	0.25	0	0	0.125

Following the above cost computations, and considering equal cost factor weights when computing $c_{Average}$, *poultry chicken* will be assigned to *lunch* in the next iteration since it produces the lowest cost. After each selection, the inter-food compatibles will be updated dynamically for foods that can still supply before performing the next selection. The weights assigned to each cost factor can be modified to favor certain factors among others, and thus modify the decision making process accordingly. For example, by increasing the weight factor for food-occurrence (c_2) and reducing other weight factors, *cottage cheese* could be assigned to *lunch* instead of *poultry chicken*. This alteration of weight factors allows determining multiple approaches for meal planning following the user's needs.

4.4.4. Solving the Transportation Matrix

Solving the transportation matrix comes down to finding the number of supply units (i.e., number of servings) to be transported from supply center (source) $food_i$ to demand center (destination) $meal_j$ such that the total transportation cost is minimum:

[12] Note that $Dist(i, i')$ =0 will never occur in our computations since it amounts to comparing a food with itself.

[13] This is different from having static cost values that remain unchanged throughout the whole computation process of typical transportation problem solutions.

$$min \sum_{i=1}^{m} \sum_{j=1}^{n} x_{(i,j)} \times c_{(i,j)} \quad \textbf{(18)}$$

where $x_{i,j}$ represents the number of servings supplied from $food_i$ to $meal_j$, and $c_{i,j}$ the cost of delivering one serving from $food_i$ to $meal_j$. The latter should be achieved while satisfying the following constraints: (i) the total amount of food servings supplied from a supply center (food) must not exceed the available supply amount (cf. Formula 19), and (ii) the total amount of servings supplied to a demand center (meal) must not exceed the required demand (cf. Formula 20):

$$\sum_{j=1}^{n} x_{(i,j)} \leq s_i \text{ for all } i = 1, \dots, m \quad \textbf{(19)}$$

$$\sum_{i=1}^{m} x_{(i,j)} \leq \overrightarrow{D_j}.d_i \text{ for all } j = 1, \dots, n \quad \textbf{(20)}$$

where s_i represents the supply (i.e., available number of servings) for supply center $food_i$, and $\overrightarrow{D_j}.d_i$ the demand (i.e., number of required servings) for demand center $meal_j$ from $food_i$'s category[14] cat_i.

Following transportation problem literature, e.g., (Hira D. and Gupta P. 2014, Winston W. and Venkataramanan M. 2003), when the total demand is equal to the total supply, the transportation problem is said to be *balanced*. If the demand exceeds the supply, then the problem cannot be solved. If the supply exceeds the demand, then the problem can be solved by adding a dummy demand center where the demand of this center is equal to the total excess supply, thus making the transportation problem balanced. In our approach, we consider the supply constraint introduced in the adapted vector-based model presented in Section 4.4.2, which states that the supply center (i.e., food) is capable of supplying all the requirements of the demand center (i.e., meal). As for the computational process to solve the transportation problem, we use the *minimum* (*least*) *cost method* widely adopted in the literature, e.g., (Hira D. and Gupta P. 2014, Winston W. and Venkataramanan M. 2003). We briefly describe the process as follows: (i) assign as much supply units as possible to the cell with the smallest unit cost in the entire matrix, (ii) cross-out the row where the supply center has supplied all available supply centers, (iii) cross-out the column where the demand center's demands were satisfied, (iv) adjust the supply and demand for those rows and columns which are not crossed based on the amount of units supplied, and (v) assign the remaining units to the feasible allocations when exactly one row or one column is left. Note that other approaches can be used to solve the transportation problem, such as *penalty-based* or *correction-based* methods (Hira D. and Gupta P. 2014).

Running example: A sample meal plan produced for our CI = 2107 Kcals running example is presented in Figure 3. We assume equal cost factors $w_i = 0.2 \ \forall i \in \{1, 2, 3, 4, 5\}$, and consider that the patient preference toward all foods is neutral (i.e., $c_1 = c_{FoodPref} = 0.5$). Here, we explain the first iteration of the process considering an extract of the transportation matrix consisting of two meals and nine food items shown in Table 5, where the cost of assigning every food at every meal is computed based on the description of the running example provided in Table 2.

Table 5. Extract transportation matrices highlighting the first iteration of the computation process

a. Input of Iteration #1

	Snack Two	**Dinner**	**Supply**
Demand	$\vec{D} = (0,0,1,3,0,0)$	$\vec{D} = (0,1,0,1,2,2)$	
Bread, whole grain	0.9	0.75	$2 * \vec{S} = (0,0,0,1,0,0)$
Milk	0.75	0.75	$2 * \vec{S} = (0,0,0,1,0,0)$
Biscuit	0.25	0.75	$1 * \vec{S} = (0,0,0,1,0,0)$
Green peas	1	1	$2 * \vec{S} = (0,0,0,1,0,0)$
Carrot	0.75	0.5	$2 * \vec{S} = (0,0,0,1,0,0)$
White rice, cooked	1	0.5	$5 * \vec{S} = (0,0,0,1,0,0)$
Lamb: chop, leg or roast	1	0.5	$3 * \vec{S} = (0,0,0,1,0,0)$
Strawberries	0.75	1	$5 * \vec{S} = (0,0,0,1,0,0)$
Blueberries	0.5	0.5	$5 * \vec{S} = (0,0,0,1,0,0)$
Rice cake	0.25	0.75	$2 * \vec{S} = (0,0,0,1,0,0)$
Olives	0.75	0.25	$2 * \vec{S} = (0,0,0,1,0,0)$

b. Output of Iteration #1 (used as input for Iteration #2)

	Snack Two	**Dinner**	**Supply**
Demand	$\vec{D0} = (0,0,1,1,0,0)$	$\vec{D} = (0,1,0,1,2,2)$	
Bread, whole grain	0.85	0.75	$2 * \vec{S} = (0,0,0,1,0,0)$
Milk	0.7	0.75	$2 * \vec{S} = (0,0,0,1,0,0)$
Biscuit	0.25	0.75	$1 * \vec{S} = (0,0,0,1,0,0)$
Green peas	1	1	$2 * \vec{S} = (0,0,0,1,0,0)$
Carrot	0.75	0.5	$2 * \vec{S} = (0,0,0,1,0,0)$
White rice, cooked	0.925	0.5	$5 * \vec{S} = (0,0,0,1,0,0)$
Lamb: chop, leg or roast	0.925	0.5	$3 * \vec{S} = (0,0,0,1,0,0)$
Strawberries	0.7	1	$5 * \vec{S} = (0,0,0,1,0,0)$
Blueberries	0.465	0.5	$5 * \vec{S} = (0,0,0,1,0,0)$
Rice cake	0.25 → 2 supplied	0.75	$2 * \vec{S} = (0,0,0,1,0,0)$
Olives	0.725	0.25	$2 * \vec{S} = (0,0,0,1,0,0)$

[14] Recall that that $\overrightarrow{D_j} = (d_1, d_2, d_3, d_4, d_5, d_6)$ represents a 6-dimentional vector where every dimension corresponds to one of the 6 categories of basic foods considered in our study, cat_1-to-cat_6 (i.e., starch, fruits, milk, vegetables, lean meat, and fat).

Meal	Food item	# of Servings	Serving size
Breakfast	Bread, whole grain	2	1 slice (28 grams)
	Oil: canola, olive,	1	1 tea spoon
	Cottage cheese	2	1/4 cup
	Peanuts	1	10 nuts
	Carrot	1	One piece (60 grams)
	Milk	1	1 cup
Snack one	Jam	1	1 table spoon
	Biscuit	1	1 piece
	Peanut Butter	1	1/2 table spoon
Lunch	Green peas	4	1/2 a cup
	Carrot	1	One piece (60 grams)
	White rice, cooked	1	1/3 cup
	Oil: canola, olive	2	1 tea spoon
	Lamb: chop, leg or roast	4	28 grams
	Strawberries	1	1.25 cup
Snack two	Blueberries	1	3/4 cup
	Rice cake	2	2 pieces
	Biscuit	1	1 piece
Dinner	White rice, cooked	2	1/3 a cup
	Lamb: chop, leg or roast	3	28 grams
	Olives	1	8 pieces
	Carrot	1	1 piece (60 grams)

Figure 3. Sample meal plan obtained based on our running example

In the first iteration, we find the minimal cost of supplying a demand center (meal) from a supply center (food) and we check if the provided supply vector meets the demand vector. Ties are broken randomly to insure variety. Consider that *rice cake* ↔*snack two* is selected in the first iteration (cf. Table 5.a). Since we need 3 servings of *starch* in *snack two* and given that *rice cake* includes 2 servings only, the latter 2 servings will be supplied from *rice cake* and then it will be marked as no longer having any supply. Subsequently, the demand vector will be updated in the second iteration to reflect that only 1 serving of starch is now needed (cf. Table 5.b). As a result of assigning *rice cake* to *snack two*, all food costs will be recomputed w.r.t. *snack two* based on the food graph connections between the food items and *rice cake* in this case. If a food is not connected to *rice cake*, its cost will remain the same and will not be reduced. The same process is repeated in the second iteration until all the demands are met (cf. Figure 3).

4.5. *Meal Plan Self-Evaluation* component

As described previously, *MPG's* meal plan generation process is designed in a flexible manner that offers the patient a variety of healthy meal plans to choose from, similarly to a human nutritionist's way of recommending multiple healthy solutions. Yet, providing various options could become confusing for the patient, if not presented properly. To address this issue, we introduce a relevance function to rank the generated meal plans w.r.t. the cost factors considered previously:

$$Rel_{Total}(\text{MP}) = \sum_{i=1}^{4} w_i \times Rel_i(\text{MP}) \quad \in [0, 1] \qquad \textbf{(21)}$$

where *MP* is a generated meal plan being evaluated, Rel_i the relevance function of each cost factor, and w_i the weight assigned to each cost factor such that $\forall\ w_i \in [0, 1]$ and $\sum_{i=1}^{4} w_i = 1$. The weight factors are the same used in solving the transportation problem and can be fine-tuned following patient preferences.

As for the individual relevance functions, we compute them as the inverse of an error rate comparing the actual meal plan score with the best possible meal plan score considering every individual factor:

$$Rel_i(MP) = 1 - \frac{|\, score_i(MP) - score_i(MP_{Max})|}{\max(score_i(MP), score_i(MP_{max}))} \quad \in [0, 1] \qquad \textbf{(22)}$$

where $score_i(MP)$ represents the score of meal plan *MP* w.r.t. cost factor *i*, and $score_i(MP_{Max})$ the score of the "best" meal plan that can be generated following the considered factor *i*. The best meal plan MP_{Max} following a certain factor *i* is generated by running the system while setting the weight of the considered factor $w_i = 1$, with all other cost factor weights being set to 0[15]. This insures that the system performs a relative and fair evaluation against the best meal plans that can be generated from the foods available in stock. $Rel_i(MP)$ will reach its maximum (=1) value when the generated meal plan scores the same as the best possible meal plan, and will decrease (and tend to minimum =0) as the generated meal plan score deviates from the best possible score. We describe each of the individual relevance scores in the following subsections.

[15] *MPG* can generate multiple different MP_{Max} solutions for the same cost factor, considering the nature of our computation process. Hence, we perform multiple runs for every individual factor separately, compute the score of each produced MP_{Max} solution in each run, and then average them out to produce $score(MP_{Max})$.

4.5.1. *Food Preference score*

This score evaluates how much the produced meal plan meets the patient food preferences, and is computed as the inverse of the overall average preference cost for every food item selected in the meal plan:

$$score_1(MP) = score_{FoodPref}(MP) = 1 - \sum_{i=1}^{m} \frac{c_{FoodPref}(food_i)}{m} \quad \in [0, 1] \tag{23}$$

where *m* is the total number of foods considered in the target meal plan *MP*, and $c_{FoodPref}(food_i)$ is the preference cost for food item $food_i$. The value of $score_{FoodPref}(MP)$ will increase/decrease in a manner which is inversely proportional to the average food preference cost, and will reach maximum (=1)/minimum (=0) values when the average cost is minimal/maximal respectively.

4.5.2. *Food Occurrence score*

This score evaluates the relevance of the generated meal plan in terms of *food occurrence* repetitions, and is computed as the inverse of the overall average occurrence cost for every food item selected in the meal plan, where the costs are produced at the time of the generation of the meal plan based on the previously selected foods:

$$score_2(MP) = score_{FoodOcc}(MP) = 1 - \sum_{i=1}^{m} \frac{c_{FoodOcc}(food_i)}{m} \quad \in [0, 1] \tag{24}$$

where *m* is the total number of foods considered in the target meal plan *MP*, and $c_{FoodOcc}(food_i)$ is the occurrence cost for food item $food_i$. The value of $score_{FoodOcc}(MP)$ will increase/decrease in a manner which is inversely proportional to the average food occurrence cost of the target meal plan, and will reach maximum (=1)/minimum (=0) values when the average cost is minimal/maximal respectively.

4.5.3. *Food-Meal Compatibility score*

This score evaluates how much the foods are compatible with their assigned meals in the generated meal plan, and is computed as the inverse of the overall average food-meal compatibility cost for every food item selected in the meal plan:

$$score_3(MP) = score_{FoodMeal}(MP) = 1 - \sum_{i=1}^{m} \frac{c_{FoodMeal}(food_i, meal_j)}{m} \quad \in [0, 1] \tag{25}$$

where *m* is the total number of foods considered in the target meal plan *MP*, and $c_{Food\text{-}Meal}(food_i, meal_j)$ is the food-meal compatibility cost for food item $food_i$ occurring in $meal_j$ of meal pan *MP*. The value of $score_{FoodMeal}(MP)$ will increase/decrease in a manner which is inversely proportional to the average food-meal compatibility cost of the target meal plan, and will reach maximum (=1)/minimum (=0) values when the average cost is minimal/maximal respectively.

4.5.4. *Inter-Food Compatibility score*

This score evaluates how much the foods are compatible with each other in the generated meal plan, and is computed as the inverse of the overall average inter-food compatibility cost for every pair of food items selected in the meal plan:

$$score_4(MP) = score_{InterFood}(MP) = 1 - \sum_{i=1}^{p} \frac{c_{InterFood}(food_i, food_{i'})}{p} \quad \in [0, 1] \tag{26}$$

where *p* is the total number of food item pairs considered in the target meal plan *MP* ($p = \frac{m \times (m-1)}{2}$ where *m* is the number of food items in *MP*), and $c_{Food\text{-}Meal}(food_i, food_{i'})$ is the inter-food compatibility cost for food items $food_i$ and $food_{i'}$ occurring in *MP*. The value of $score_{InterFood}(MP)$ will increase/decrease in a manner which is inversely proportional to the average inter-food compatibility cost of the target meal plan, and will reach maximum (=1)/minimum (=0) values when the average cost is minimal/maximal respectively.

4.5.5. *Running Example*

Consider in Figure 4 an extract of our running example meal plan from Figure 3, consisting of two meals: *snack two* and *dinner*, and their constituent food items. For clarity of presentation, we demonstrate how the different relevance scores for each of the cost factors are generated using this extract meal plan (computations for the complete running example are provided in (Salloum G. and Tekli J. 2020)).

Meal	Food item	# of Servings	Serving size
Snack two	Blueberries Rice cake Biscuit	1 2 1	3/4 cup 2 pieces 1 piece
Dinner	White rice, cooked Lamb: chop, leg or roast Olives Carrot	2 3 1 1	1/3 a cup 28 grams 8 pieces 1 piece (60 grams)

Figure 4. Extract meal plan $MP_{Extract}$ taken from our running example in Figure 3

Table 6. Sample transportation matrix considering meal plan $MP_{Extract}$ from Figure 4

Food	Food Preference Cost	Food Occurrence cost	Food-Meal Compatibility cost		Inter-Food compatibility cost
			with *snack two*	with *dinner*	Cost
Blueberries	0.25	0.5 (occ=2)	0.5	0.5	Rice cake = 1 (dist=*NA*) Biscuit = 1 (dist=8)
Rice Cake	0.25	0.25 (occ=1)	0	1	Biscuit = 1 (dist=*NA*)
Biscuit	0.5	0 (occ=0)	0	1	-
White Rice	0	0 (occ=0)	1	0	Lamb = 0 (dist=1) Olives = 0.25 (dist=2) Carrot = 0.25 (dist=2)
Lamb	0.5	0.5 (occ=2)	1	0	Olives = 0.5 (dist=3) Carrot = 0.25 (dist=2)
Olives	0.75	0 (occ=0)	1	0	Carrot = 0.25 (dist=2)
Carrot	0	0 (occ=0)	0	0.5	-

Considering the transportation matrix in Table 7, produced at meal plan generation time, the extract meal plan's individual scores for each cost factor are computed as follows:

- Food preference: $score_{FoodPref}(MP_{Extract}) = 1 - \sum_{i=1}^{m} \frac{c_{FoodPref}(food_i)}{m} = 1 - \frac{0.25+0.25+0.5+0+0.5+0.75+0}{7} = 0.68$
- Food occurrence: $score_{FoodOcc}(MP_{Extract}) = 1 - \sum_{i=1}^{m} \frac{c_{FoodOcc}(food_i)}{m} = 1 - \frac{0.5+0.25+0+0+0.5+0+0}{7} = 0.82$
- Food-meal compatibility: $score_{FoodMeal}(MP_{Extract}) = 1 - \sum_{i=1}^{m} \frac{c_{FoodMeal}(food_i,\ meal_j)}{m} = 1 - \frac{0.5+0+0+0+0+0+0.5}{7} = 0.86$
- Inter-food compatibility: $score_{InterFood}(MP_{Extract}) = 1 - \sum_{i=1}^{p} \frac{c_{InterFood}\left(food_i, food_{i'}\right)}{p}$
  $= 1 - \frac{1+1+1+0+0.25+0.25+0.5+0.25+0.25}{9} = 0.5$

Considering the following scores produced for the best possible meal plans for every individual cost factor[16]: $score_{FoodPref}(MP_{Max})$=1, $score_{FoodOcc}(MP_{Max})$=1, $score_{FoodMeal}(MP_{Max})$=0.96, and $score_{InterFood}(MP_{Max})$=82, we compute $MP_{Extract}$'s relevance functions as follows:

- Food preference: $Rel_{FoodPref}(MP_{Extract}) = 1 - \frac{|0.68-1|}{1} = 0.68$
- Food occurrence: $Rel_{FoodOcc}(MP_{Extract}) = 1 - \frac{|0.82-1|}{1} = 0.82$
- Food-meal compatibility: $Rel_{FoodMeal}(MP_{Extract}) = 1 - \frac{|0.86-0.96|}{0.96} = 0.90$
- Inter-food compatibility: $Rel_{InterFood}(MP_{Extract}) = 1 - \frac{|0.5-0.82|}{0.82} = 0.61$

The overall relevance function for $MP_{Extract}$ considering equal weights for all cost factors (w_1-to-w_4 = 0.25) is computed as:

- Overall relevance: $Rel_{Total}(MP_{Extract}) = \sum_{i=1}^{4} w_i \times Rel_i(MP_{Extract}) = 0.7525$

To sum up, the self-evaluation results show that meal plan $MP_{Extract}$ is considered 0.68 % relevant in terms of food preferences, 0.82 % relevant in terms of food occurrences, 0.90 % relevant in terms of food-meal compatibility, and 0.61 % relevant in terms of inter-food compatibility. All in all, the system considers $MP_{Extract}$ to be ≈0.75 % relevant considering all factors combined with equal weights. Recall that the latter are internal relevance scores produced by the *MPG* agent itself, allowing it to rate and rank the generated meal plans following their relevance w.r.t. patient chosen factors.

[16] The scores for the best possible meal plans are computed experimentally, considering a pool of 15 experimental runs where the produced score for every individual cost factor is computed as the average of the maximum scores obtained in every experimental run (cf. Section 5).

5. Experimental Evaluation

We have implemented our *MPG* agent as part of the Personal Intelligent Nutritionist (*PIN*) framework (Salloum G. and Tekli J. 2021), which aims at automating the full nutrition health recommendation process: starting from health assessment and caloric intake (CI) recommendations, to physical exercise recommendation and adjustment, leading to meal plan generation through *MPG*[17]. We have empirically tested *MPG* using three sets of experiments covering: (i) *patient preference satisfaction*: evaluating *MPG*'s ability to generate meal plans that satisfy patient food preferences and occurrence variety, (ii) *meal plan quality*: evaluating *MPG*'s ability to generate "healthy" meal plans following nutrition experts, and (iii) *cost weight variation*: evaluating the effect of changing the cost weight factors on the agent's self-evaluation component. A total of 9 human testers: 4 nutrition experts and 5 non-experts (patients) were involved in the experiments. Experimental results are described and discussed in the below sub-sections. The system implementation, experimental data, and test results are available online[18].

5.1. Experiment 1: *Patient Preference Satisfaction* evaluation

The objective of this experiment is to evaluate *MPG*'s ability to generate meal plans that satisfy patient preferences. To do so, we provide every non-expert tester (patient) with 4 sets of 3-day meal plans where every set targets the CI needs and cost factor preferences set by the testers themselves. The testers are then asked to evaluate and rate their satisfaction of the generated meal plans on an integer scale from 0 (strong disagreement) to 4 (strong agreement), by considering two criteria: (i) *food preference satisfaction*, i.e., if the foods selected meet the preferences set by the tester, and (ii) *food occurrence variety*: if the daily meal plans have a variety of non-repetitive foods that satisfy the tester for every daily meal plan. Figure 5 shows the tester rating scores averaged for every evaluation criterion over the 3 days covered by the meal plans.

Results show that *MPG* fairly satisfies tester *food preference* and *food occurrence variety*, producing an overall average rating of 3.27 with an average standard deviation of 0.49. *MPG* seems to perform slightly better in satisfying *food occurrence variety* with an average rating of 3.47, compared with *food preference* with an average rating of 3.07. The latter results can be further improved by increasing the size of the food graph considered in our study (it currently consists of 56 food items, cf. Figure 2), which would provide the testers with a larger selection of food items that meet their preferences.

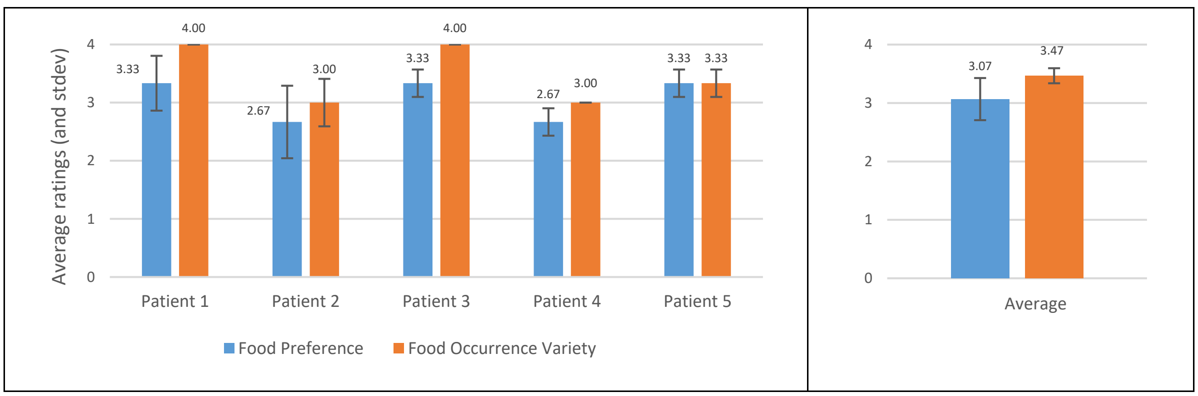

Figure 5. Average non-expert tester ratings evaluating their satisfaction with the generated meal plans

5.2. Experiment 2: *Meal Plan Quality* evaluation

The objective of this experiment is to evaluate *MPG*'s ability to generate healthy meal plans. To do so, we sought the participation of 4 nutrition experts who volunteered to participate in this experiment. Each expert is provided 4 sets of 3-day meal plans, where every set targets one of the following 4 typical patient caloric intake (CI) requirements: 1200, 1600, 2000, and 2400 Kcals[19]. Every set is divided in two equal groups (referred to as A and B) following the nature of the food items involved: i) group A: *basic foods only* (e.g., bread, milk, apples), and ii) group B: *basic and composite foods* (e.g., pizza, hamburger, lasagna). The latter is necessary to evaluated *MPG*'s ability of matching foods of different natures, especially when involving meal-food compatibility and inter-food compatibility factors. The meal plans are generated using equal weights assigned to all cost factors. The expert testers are asked to evaluate the quality (i.e., healthiness) of the meal plans by considering three criteria: (i) *food occurrence variety*: if the meal plans have a variety of non-repetitive foods, (ii) *food-meal compatibility*: if the foods are correctly assigned to the meals, and (iii) *inter-food compatibility*: if the foods are matched well together within the same meal[20]. Testers are asked to rate every criterion on an integer scale from 0 (strong disagreement) to 4 (strong agreement). Figures 6, 7, and 8 show the quality ratings for meal plans including *basic foods only* (group A), *basic and composite foods* (group B), and all foods combined (group A and B). Rating scores are averaged for every evaluation criterion, considering expert tester scores produced over the 3 days covered by the meal plans.

[17] This paper describes *MPG*, while *PIN*'s remaining modules are developed in (Salloum G. and Tekli J., 2020).
[18] http://sigappfr.acm.org/Projects/PIN/
[19] The CI requirement cases considered in this experiment are chosen based on common practices in health nutrition literature (Kathleen M. and Janice R. 2017).
[20] Note that the *food preference* criterion does not reflect meal plan healthiness, and is evaluated by non-expert testers in Experiment 2 (Section 5.2).

Results show that *MPG* performs well, producing an overall quality rating of 2.67 by averaging the results of the four CI requirements (i.e., 1200, 1600, 2000, and 2400) and the three evaluation criteria (e.g., food occurrence variety, food-meal compatibility, and inner-food compatibility) for both *basic foods only* and *basic and composite foods* combined, with an average standard deviation of 0.57. More specifically, we make the following observations:

- *MPG* seems to perform slightly better for meal plans consisting of *basic foods only* (group A, producing a higher average rating score of 2.73 considering all CI requirements and all evaluation criteria combined) compared with plans including *basic and composite foods* (group B, with avg. = 2.45). We also notice higher standard deviation levels for group B (std. dev. = 0.65 considering all CI requirements and all evaluation criteria combined, versus std. dev = 0.50 for group A), highlighting a wider diversity of opinions among testers when composite foods are included in the evaluation process.

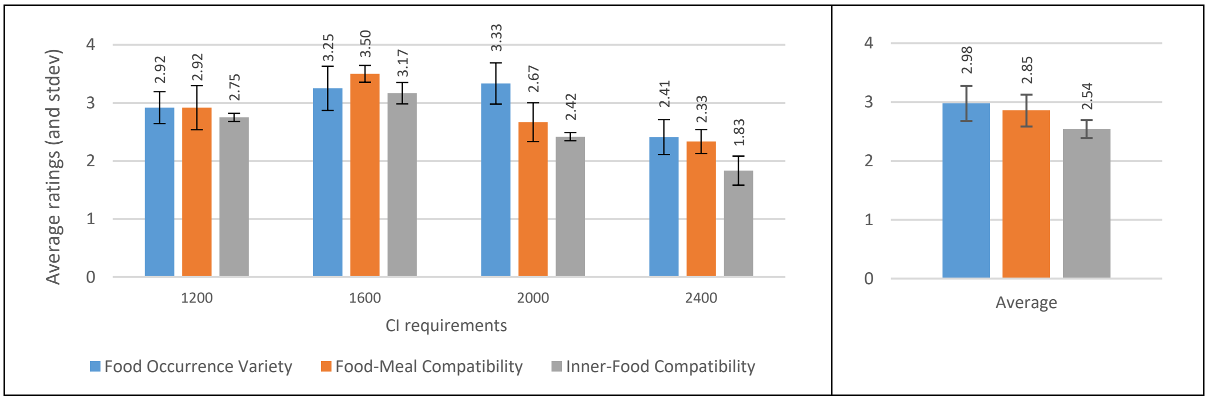

Figure 6. Average expert tester ratings evaluating the quality of meal plans including *basic foods only* (group A)

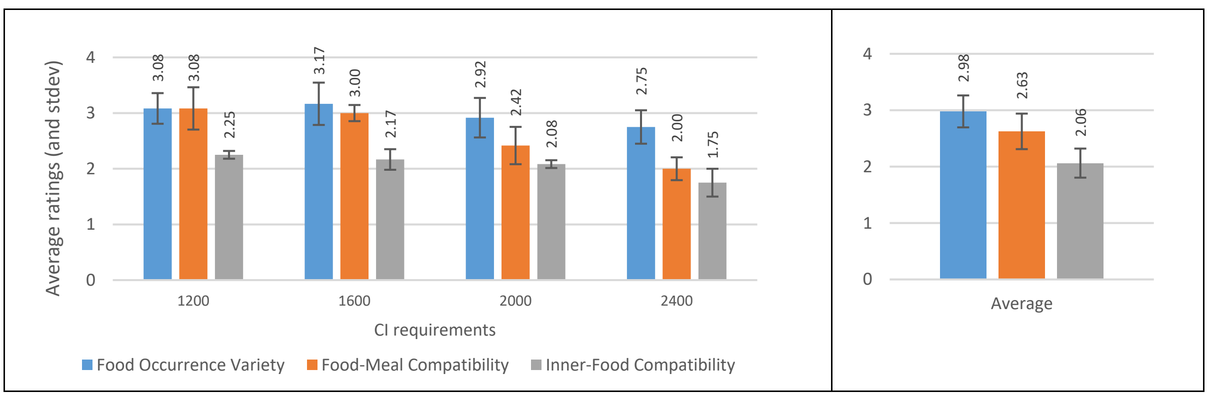

Figure 7. Average expert tester ratings evaluating the quality of meal plans including *basic and composite foods* (group B)

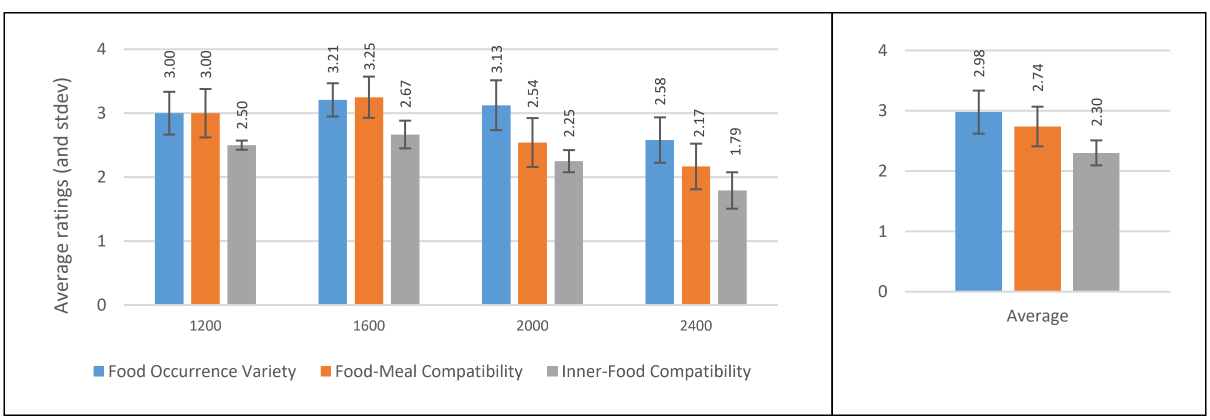

Figure 8. Average expert tester ratings evaluating the quality of meal plans considering both *basic foods only* (group A) and *basic and composite foods* (group B) combined

- Considering both groups A and B (cf. Figure 8), *MPG* seems to perform best when evaluating *food occurrence variety* (avg. = 2.98 considering all CI requirements for both *basic foods only* and *basic and composite foods* combined), followed by *meal-food compatibility* (avg. = 2.74) and then *inter-food compatibility* (avg. = 2.30). The lower scores for *inter-food compatibility* can be attributed to the food compatibility graph used as reference for computing food compatibility scores (cf. Section 4.4.3). The graph was constructed with the help of two nutrition experts who are different from the ones who participated in the evaluation process. Here, subjectivity in matching food items could be a key factor affecting the results, especially since different experts might have different opinions concerning food compatibility. A possible approach would to allow the patients to define their own personal food matchings, instead of relying on a pre-defined food graph, which would produce more personalized inter-food compatibility results.

5.3. Experiment 3: *Cost Weight Variation* evaluation

In this experiment, we vary the weights of the cost factors considered in our approach, and evaluate their effect on *MPG*'s meal plan self-evaluation component (cf. Section 4.5). We consider 5 sets of meal plans generated using the following weight configurations:

- Set 1: $w_{FoodPref} = 0.7$ and $w_{FoodOcc} = w_{FoofMeal} = w_{InterFood} = 0.1$, where the *food preference* factor is emphasized over the others,
- Set 2: $w_{FoodOcc} = 0.7$ and $w_{FoodPref} = w_{FoofMeal} = w_{InterFood} = 0.1$, where the *food occurrence* factor is emphasized,
- Set 3: $w_{FoodMeal} = 0.7$ and $w_{FoodPref} = w_{FoodOcc} = w_{InterFood} = 0.1$, where the *food-meal compatibility* factor is emphasized,
- Set 4: $w_{InterFood} = 0.7$ and $w_{FoodPref} = w_{FoodOcc} = w_{FoofMeal} = 0.1$, where the *inter-food compatibility* factor is emphasized,
- Set 5: $w_{FoodPref} = w_{FoodOcc} = w_{FoodMeal} = w_{InterFood} = 0.25$, where all factors are emphasized and assigned equal weights,

Every set is made of 20 meal plans (i.e., totaling 100 meal plans all-in-all), generated for each of the typical CI requirements considered in the previous experiments: 1200, 1600, 2000, 2400 Kcals. We do not distinguish between *basic foods only* and *basic and composite foods* in this experiment since they do not make a difference in this evaluation: they both fit as supply vectors in the transportation problem where the same cost factors produce the same numerical results. Table 7 shows *MPG*'s meal plan self-evaluation results for individual cost factors, and Figure 9 depicts the corresponding average and standard deviation results.

Table 7. *MPG*'s self-evaluation results considering different CI requirements and cost factor configurations

a. Set 1 (*food preference* factor is emphasized)

Caloric Intakes	Food Preference	Food Occurrence	Meal-Food Compatibility	Inter-food Compatibility
1200	1.00	0.55	0.83	0.85
1600	1.00	0.64	0.77	0.71
2000	1.00	0.63	0.76	0.82
2400	1.00	0.63	0.76	0.75

b. Set 2 (*food occurrence* factor is emphasized)

Caloric Intakes	Food Preference	Food Occurrence	Meal-Food Compatibility	Inter-food Compatibility
1200	0.93	1.00	0.81	0.80
1600	0.87	1.00	0.88	0.72
2000	0.85	1.00	0.88	0.81
2400	0.90	1.00	0.86	0.76

c. Set 3 (*food-meal compatibility* factor is emphasized)

Caloric Intakes	Food Preference	Food Occurrence	Meal-Food Compatibility	Inter-food Compatibility
1200	0.83	0.65	1.00	0.81
1600	0.84	0.73	1.00	0.68
2000	0.82	0.73	1.00	0.77
2400	0.85	0.75	1.00	0.78

d. Set 4 (*inter-food compatibility* factor is emphasized)

Caloric Intakes	Food Preference	Food Occurrence	Meal-Food Compatibility	Inter-food Compatibility
1200	0.93	0.70	0.85	0.98
1600	0.90	0.61	0.90	0.80
2000	0.91	0.70	0.88	0.92
2400	0.88	0.68	0.92	0.86

e. Set 5 (all factors are equally emphasized)

Caloric Intakes	Food Preference	Food Occurrence	Meal-Food Compatibility	Inter-food Compatibility
1200	0.90	0.86	0.91	0.86
1600	0.90	0.87	0.92	0.75
2000	0.90	0.88	0.90	0.86
2400	0.90	0.85	0.90	0.79

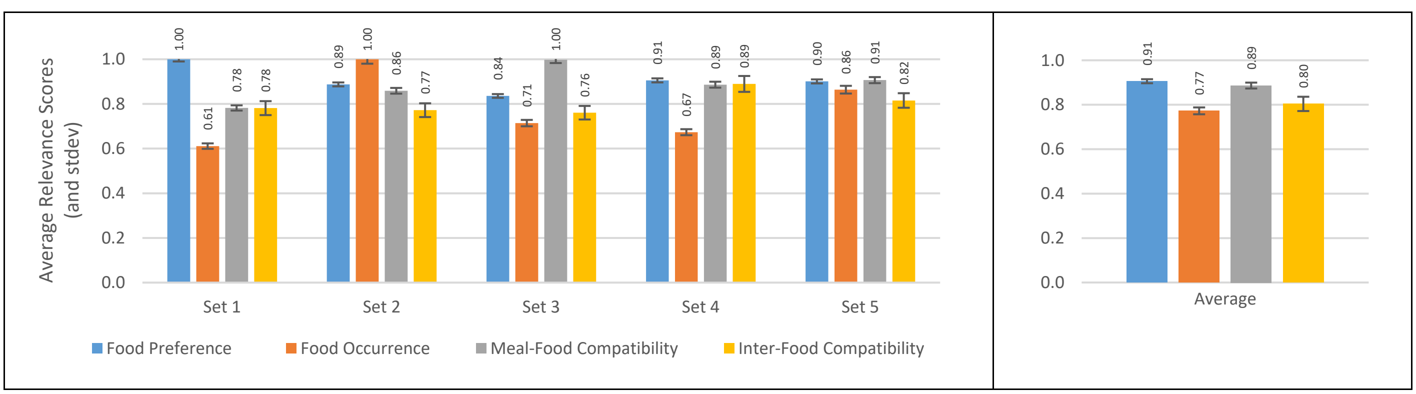

Figure 9. MPG's self-evaluation results averaged for each cost factor configuration

Relevance levels for most test sets show consistent and high quality results across all CI requirement categories and cost factor configurations, highlighting *MPG*'s self-evaluation performance. More specifically, we highlight the following observations:

- **Set 1** emphasizing *food preference*: *MPG* produces maximum relevance scores (=1) for the mentioned factor compared with the others, and this is achieved consistently throughout all four CI requirements (cf. Table 7.a).
- **Set 2** emphasizing *food occurrence*: *MPG* produces maximum relevance scores (=1) for the mentioned factor, compared with the other factors across all CI requirement categories. We also notice that relevance score for *food preference* is significantly improved with the smallest CI requirement of 1200 Kcals, compared with the other CI requirements (cf. Table 7.b) This is because a small CI requirement produces a reduced number of foods per meal plan, which in turn increases the probability of choosing foods that match the patient preferences. In fact, choosing less foods from the food graph while limiting repetitive choices (e.g., the same food should not be selected in consecutive days) increases the probability of having more convenient foods (that the patient likes and which are compatible with the meals and with each other) allowing to fulfil the demands of the meal plan. However as CI requirements increase, so does the number of foods to be included per meal plan, which reduces the probability of finding non-repeating foods that satisfy the preference of the patient. In addition, we notice a significant increase in *meal-food compatibility* from Set 1 to Set 2 (cf. Figure 9). This can be attributed to the fact that the *food occurrence* factor is less restrictive on the foods being assigned to a certain meal, where any food can be assigned to a meal as long as it has not occurred enough times previously. In contrast, the *food preference* factor is stricter on food assignments, and might induce sub-optimal food assignments simply because the foods are highly preferred by the patient.
- **Set 3** emphasizing *food-meal compatibility*: *MPG* produces maximum relevance scores (=1) for the mentioned factor compared with the other factors across all CI requirement categories. Relevance scores for the other factors vary between 0.71 for *food occurrence* and 0.84 for *food preference*, which we consider as fair scores given that these factors are not emphasized in this experimental run**.**
- **Set 4** emphasizing *inter-food compatibility*: *MPG* produces consistently high scores for the mentioned factor across all CI requirement categories. Yet, the produced scores are sometimes slightly surpassed by other factors (like *food preference* and *food-meal compatibility* for CI=1600 Kcals, and *food preference* for CI=2400 Kcals, cf. Table 7.d). Following our discussion with nutrition experts, this result resonates with manual meal planning where it is very unlikely to match all items in a meal – one-to-one – in an optimal way, especially when CI grows and the meal grows in size accordingly. We also notice relatively high relevance scores for *food preference* (average = 0.9) and *meal-food compatibility* (average = 0.89, cf. Figure 9). The former demonstrates that foods can be matched well together while meeting patient preferences, while the latter can be attributed to the fact that foods that go well together are more likely to go well in the same meal.
- **Set 5** emphasizing all factors equally: *MPG* produces consistent (almost equal) scores and relevance levels for each factor across all CI requirement categories. We also notice that the average relevance score for each factor are within the same range of the optimized results for that category. This shows that a balanced weight configuration combining all cost factors can result in meal plans where all factors are well satisfied without favoring one factor over the others.

To sum up, this experiment provides a self-evaluation of *MPG*, showing that the system behaves as expected according to the weight factor choices made by the patient, where each factor can be individually emphasized, and all factors can be equally emphasized, following the patient's weight factor preferences.

6. Conclusion

In this paper, we introduce a novel solution for Meal Plan Generation titled *MPG*, allowing to automate the meal plan generation service offered by a nutrition expert. *MPG* allows to: (i) generate meal plans which fulfil a recommended caloric intake (e.g., nutrition demand) given a set of available foods (e.g., nutrition supply), while (ii) personalizing the plans following patient chosen factors (e.g., food preferences, compatibility, variety, and price), and (iii) evaluating their relevance following the patients' preferences. Experimental results reflect *MPG*'s effectiveness and quality in producing "healthy" and personalized meal plans which largely comply with human tester preferences. They also display *MPG*'s ability to accurately self-evaluate is own meal plans in order to provide the patients with a set of recommendations that fit their needs.

We are currently completing an extended study, building on *MPG*'s meal plan generation process to perform nutrition health monitoring and meal plan adjustment: assessing the patient's health state evolution over time and recommending adjusted meal plans accordingly (Salloum G. and Tekli J. 2021). In the near future, we aim to extend *MPG* to include exercise recommendations, providing patients with the choice to follow leaner or heavier meal plans following the amount of physical exercise they wish to undertake (e.g., patients who do not exercise, versus patients who regularly exercise), taking into account different age, gender, and food preference groups (e.g., vegetarian or vegan), and multiple health nutrition measurements including body mass index and body fat percentage. Integrating image analysis-based preference learning (Yang L., Hsieh C.K. et al. 2017, Salameh K., Tekli J., et al., 2014) in order to monitor foods consumed by the patient would help perform personalized and interactive meal planning. On the long run, we plan to develop a dedicated exercise planning mechanism that incorporates and schedules multiple exercise types (e.g., jogging, swimming) based on the patient's time availability and exercise preferences. Using alternative computation techniques such as non-parametric and lazy machine learners (e.g., fuzzy *k*-nearest neighbors, or fuzzy support vector machines)

(Abboud R. and Tekli J. 2018, Abboud R. and Tekli J. 2019, Fahmi A. *et al.* 2019) could be most useful in this context, in order to compensate for the lack of formal rules and lack of sizeable training data coining physical exercise fitness with meal plan recommendation.

Declarations

An early version of *MPG*'s prototype is described in (Salloum G. et al., 2018). The latter is a short 4-page technical paper which briefly describes the initial version of the prototype, consisting of: i) a short one-paragraph description of the state of the art (the current paper provides a comprehensive review of exiting solutions), ii) a short one page description of the initial theoretical proposal (the current paper provides an in-depth description of the new proposal), and iii) a very limited and preliminary experimental evaluation, consisting of one experiment described in one paragraph (the present paper performs an in-depth experimental evaluation of the new proposal – cf. Section 5, considering three major experiments involving a total of 9 human testers: 4 nutrition experts and 5 non-experts). In addition, the initial proposal in (Salloum G. et al., 2018) considers: iv) basic food items only (the current paper introduces a new model integrating both basic and composite foods, cf. Section 4.4), v) the traditional transportation problem only (the present paper introduces a new multi-factor adaptation of the transportation problem, specifically designed to handle composite foods, cf. Section 4), iv) pre-defined static food-meal cost values defined by experts (the present paper introduces a dynamic approach consisting of a battery of novel mathematical cost functions – cf. Section 4.4.3 – and meal plan evaluation functions – cf. Section 4.5).

Funding
This study is partly funded by the National Council for Scientific Research (CNRS-L) – Lebanon, and the Lebanese American University (LAU).

Acknowledgements

We would like to thank all nutritionists who volunteered to participate in this study, namely: Dr. Maya Bassil (Associate Professor of Human Nutrition in the Department of Natural Sciences at LAU), and Ms. Eva-Maria Kahwaji (M.Sc. in Sports and Exercise Nutrition at Loughborough University), for their help in preparing the food compatibility graph, as well as Ms. Haneen Boughanem (Licensed Dietitian), Mr. Omar Makki (Research Assistant in Nutrition and Dietetics Coordinated Program, Natural Sciences Department, LAU), Ms. Fatima kawtharani (M.Sc. in Human Nutrition and UNICEF field worker), and Ms. Rym Kalo (Licensed dietitian) for participating in the meal plan assessment tests.

Conflicts of interest/Competing interests
Not Applicable.

Compliance with Ethical Standards

- Disclosure of potential conflicts of interest
  1. Funding: *The study was partly funded by:*
     - *National Council for Scientific Research – Lebanon, CNRS-L (grant number: NCSR-LAU#887)*
     - *Lebanese American University, LAU (grant number: SOERC1516R003)*
  2. Conflict of Interest: *The authors declare that they have no conflict of interest.*

- Research involving human participants and/or animals
  1. Statement of human rights: *Ethical approval: For this type of study formal consent is not required.*
  2. Statement on the Welfare of Animals: *Ethical approval: This article does not contain any studies with animals performed by any of the authors.*

- Informed consent: *Additional informed consent was obtained from all individual participants for whom identifying information is included in this article.*

Availability of Data

The datasets generated and analyzed during the current study are described in this published thesis report (Salloum G. and Tekli J., 2020). They are also available from the authors on reasonable request.

Availability of Code

A software demo and an executable version of the prototype are available at the following link: http://sigappfr.acm.org/Projects/PIN/.

References

Abboud R. and Tekli J. (2018). *MUSE Prototype for Music Sentiment Expression.* IEEE International Conference on Cognitive Computing (ICCC'18), part of the IEEE World Congress on Services 2018 pp. 106-109.

Abboud R. and Tekli J. (2019). *Integration of Non-Parametric Fuzzy Classification with an Evolutionary-Developmental Framework to perform Music Sentiment-based Analysis and Composition.* Springer Soft Computing DOI: 10.1007/s00500-019-04503-4.

Ainsworth B., Haskell W., Herrmann S., B. D. J. Meckes N., Tudor-Locke C., V. J. Greer J., Whitt-Glover M. and Leon A. (2011). *2011 Compendium of Physical Activities: A Second Update of Codes and MET Values.* Medicine and Science in Sports Exercise 43(8):1575–1581.

Ayoub J., Samra A., Hlais A., Bassil M. and Obeid O. (2015). *Effect of Phosphorus Supplementation on Weight Gain and Waist Circumference of Overweight/Obese Adults: a Randomized Clinical Trial.* Nutrition & Diabetes 5(12), e189.

Bassi L. (1976). *The Diet Problem Revisited.* Americain Economist 20(2): 35–39.

Bianchi L., Dorigo M., Gambardella L. and Gutjahr W. (2009). *A Survey on Metaheuristics for Stochastic Combinatorial Optimization.* Natural Computing 8(2):239–287.

Cormen T.H., Leiserson C.E., Rivest R.L. and S. C. (2009). *Introduction to Algorithms (3rd Ed.).* MIT Press and McGraw-Hill.

Deb K. and Jain H. (2014). *An Evolutionary Many-Objective Optimization Algorithm using Reference-Point-based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints*. IEEE Transactions on Evolutionary Computation, 18(4): 577–601.

EatThisMuch Inc. (2020). *"Eat This Much".* Available: https://www.eatthismuch.com/ (Accessed: 03-April-2020).

El-Ghazi T. (2009). *Metaheuristics: From Design to Implementation.* Wiley 624 p.

Evans D. (2017). *"MyFitnessPal".* Br. J. Sports Med. 51(14): 1101–1102.

Fahmi A., Abdullah S., Amin F. and K. M.: (2019). *Trapezoidal Cubic Fuzzy Number Einstein Hybrid Weighted Averaging Operators and its Application to Decision Making.* Soft Computing 24(14): 5753-5783.

Fister D. and R. S. Fister I. (2016). *Generating Eating Plans for Athletes using the Particle Swarm Optimization.* IEEE Int. Symp. Comput. Intell. Informatics (CINTI'16) pp. 193–198.

Fitness Meal Planner (2020). *"Fitness Meal Planner".* Available: http://www.fitnessmealplanner.com/, (Accessed: 03-April-2020).

Foytik J. (1981). *Devising and Using a Computerized Diet: An Exploratory Study.* J. Consumer Affairs 15(1):158–169.

Gaál B., Vassányi I. and Kozmann G. (2005). *A Novel Artificial Intelligence Methodfor Weekly Dietary Menu Planning.* Methods of Information in Medicine 44:655–64.

Hall K. D. et al. (2011). *Quantification of the Effect of Energy Imbalance on Bodyweight.* Lancet, 378(9793): 826–837.

Hira D. and Gupta P. (2014). *Operations Research.* Published by S. Chand & Company Ltd. ISBN10:8121902819

Husain W., Wei L. and Z. N. Cheng S. (2011). *Application of Data Mining Techniques in a Personalized Diet Recommendation System for Cancer Patients.* IEEE Colloq. Humanit. Sci. Eng. CHUSER 2011 pp. 239–244.

Kaldirim E. and Kose Z. (2006). *Application of a Multi-Objective Genetic Algorithm to the Modified Diet Problem.* Genetic and Evolutionary Computation Conf. pp. 10–13.

Karnik N. and Mendel J. (2001). *Operations on Type-2 Fuzzy Sets.* Fuzzy Sets and Systems 122:327–348.

Kathleen M. and Janice R. (2017). *Food & the Nutrition Care Process.* 14th Edition, Saunders p. 1152.

Khan A. S. and Hoffmann A. (2003a). *An Advanced Artificial Intelligence Tool for Menu Design.* Nutritial Health journal 17(1):43-53.

Khan A. S. and Hoffmann A. (2003b). *Building a Case-based Diet Recommendation System without a Knowledge Engineer.* Elsevier journal of Artificial Intelligence in Medicine 27(2):155-79.

Lancaster L. (1992). *The History of the Application of Mathematical Programming to Menu Planning.* European Journal of Operational Research 57(3):339–347.

Lee C., Wang M., Habras H., Chen Z., Lan S., Hsu C., Kuo S., Kuo H. and Cheng H. (2012). *A Novel Genetic Fuzzy Markup Language and Its Application to Healthy Diet Assessment.* Int. J. Uncertainty, Fuzziness Knowledge-Based Syst. 20:247–278.

Lee C., Wang M. and Hagras H. (2010). *A Type-2 Fuzzy Ontology and its Application to Personal Diabetic-Diet Recommendation.* IEEE Trans. Fuzzy Syst. 18(2):374-395.

Lee C., Wang M., Hsu C. and Hagras H. (2009). *A Novel Type-2 Fuzzy Ontology and its Application to Diet Assessment*. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol. - Work. WI-IAT Work. 3(1):417–420.

Lee C., Wang M., Li H. and Chen W. (2008). *Intelligent Ontological Agent for Diabetic Food Recommendation.* IEEE Int. Conf. Fuzzy Syst. pp. 1803–1810.

Lee C., W. M. and Lan S. (2015). *Adaptive Personalized Diet Linguisti Recommendation Mechanism Based on Type-2 Fuzzy Sets and Genetic Fuzzy Markup Language.* EEE Trans. Fuzzy Syst. 23(5):1777-1802.

Lee C.S., Wang M.H., Acampora G., Hsu C.Y. and Hagras H. (2010). *Diet Assessment based on Type-2 Fuzzy Ontology and Fuzzy Markup Language.* International Journal of Intelligent Systems pp. 1187-1216.

Lenstra H. (1983). *Integer Programming with a Fixed Number of Variables.* Mathematics of Operational Research 8(4):538–548.

Livestrong Foundation (2020). *"MyPlate".* Available: https://www.livestrong.com/myplate/ (Accessed: 1-April-2020).

Mahan L., Escott-Stump S. and Raymond L. (2012). *Krause's Food & the Nutrition Care Process.* Philadelphia: Saunders pp. 1248.

MakeMyPlate Inc. (2020). *"Make My Plate".* Available: http://www.makemyplate.co/ (Accessed: 01-April-2020).

Mattar L., Zeeni N. and Bassil M. (2015). *Effect of Movie Violence on Mood, Stress, Appetite Perception and Food Preferences in a Random Population.* European Journal of Clinical Nutrition 69(8), 972.

McCann-Rugg M., White G. and Endres J. (1983). *Using Goal Programming to Improve the Calculation of Diabetic Diets.* Computers and Operational Ressearch 10(4): 365–373.

MyNetDiary Inc. (2020). *"MyNetDiary".* Available: ttps://www.mynetdiary.com/ (Accessed: 01-April-2020).

Noor S., Mohd A., Ruhana K., Mahamud K. and Arbin N. (2018). *Self-Adaptive Hybrid Genetic Algorithm (SHGA).* Far East J. Math. Sci. 103: 171–190.

Petot G. J., Marling C. and Sterling L. (1998). *An Artificial Intelligence System for Computer-Assisted Menu Planning.* J. Am. Diet. Assoc. 98(9): 1009-1014.

Rolls B. (2009). *The Relationship between Dietary Energy Density and Energy Intake.* Physiol. Behav. 97(5):609–615.

Salameh K., Tekli J., and Chbeir R. (2014), *SVG-to-RDF Image Semantization*, Inter. Conf. on Similarity Search and Applications (SISAP'14), 214-228

Salloum G., Semaan E. and Tekli J. (2018). *PIN Prototype for Intelligent Nutrition Assessment and Meal Planning.* IEEE Internaitonal Conference on Cognitive Computing (ICCC'18) - Part of the IEEE Congress on Services pp. 110-113.

Salloum G. and Tekli J. (2021). *Automated and Personalized Nutrition Health Assessment, Recommendation, and Progress Evaluation using Fuzzy Reasoning.* Elsevier International Journal of Human-Computer Studies (IJHCS), 151:102610.

Salloum G. and Tekli J. (2020). *Automated Knowledge-based Nutrition Health Assessment, Recommendation, Progress Evaluation, and Meal Planning.* Master's Thesis Report, LAU Lib., https://laur.lau.edu.lb:8443/xmlui/bitstream/handle/10725/10468/George_Kamil_Salloum_Thesis_Redacted.pdf

Schniederjans M. (1995). *Goal Programming: Methodology and Applications* 1st ed. Springer, 1995. 219 p.

Seljak B. (2009). *Computer-based Dietary Menu Planning.* J. Food Composition and Analysis 22(5):414–420.

Silberberg E. (Nutrition and the Demand for Tastes). *1985.* J. Political Economy 93(5):881-900.

SparkPeople Inc. (2020). *"SparkPeople".* Available: https://www.sparkpeople.com/resource/mix-and-match-meal-planner.asp (Accessed: 01-April-2020).

Stigler G. (1945). *The Cost of Subsistence.* J. Farm Economics 27(2): 303–314.

Suksom N. and Buranarach M. (2010). *A Knowledge-based Framework for Development of Personalized Food Recommender System.* Proc. of the 5th Int. Conf. on Knowledge, Information and Creativity Support Systems p. 2.

Türkmenoglu C., Etaner-Uyar A.S., and Kiraz B. (2021). *Recommending Healthy Meal Plans by Optimising Nature-inspired Many-Objective Diet Problem*. Health Informatics Journal 27(1):146045822097671

Valdez-Pena H. and Martinez-Alfaro H. (2003). *Menu Planning using the Exchange Diet System.* Int. Conf. on Systems, Man and Cybernetics Vol 3, pp. 3044-3049.

Wang M. H. (2009). *Intelligent Ontological Multi-Agent for Healthy Diet Planning.* IEEE Inter. Conf. on Fuzzy Systems doi:10.1109/FUZZY.2009.5277049.

Winston W. and Venkataramanan M. (2003). *Introduction to Mathematical Programming : Operations Research Volume.* Thomson Learning, 4th Edition 924 p.

Yang L., Hsieh C.K., Yang H., Dell N., Belongie S., Cole C. and Estrin D. (2017). *Yum-Me: A Personalized Nutrient-Based Meal Recommender System.* ACM Transactions on Information Systems 36(1): 7.