

Article

A Hybrid Food Recommendation System Based on MOEA/D Focusing on the Problem of Food Nutritional Balance and Symmetry

Shuchang Huang, Cungang Wang * and Wei Bian

School of Computer, Liaocheng University, Liaocheng 252000, China; huangshuchang703@gmail.com (S.H.); weibian355@gmail.com (W.B.)

* Correspondence: wangcungang@lcu-cs.com

Abstract: With the improvement of people's living standards, the issue of dietary health has received extensive attention. In order to simultaneously meet people's demands for dietary preferences and nutritional balance, we have conducted research on the issue of personalized food recommendations. For this purpose, we have proposed a hybrid food recommendation model, which can provide users with scientific, reasonable, and personalized dietary advice. Firstly, the collaborative filtering (CF) algorithm is adopted to recommend foods to users; then, the improved Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) is used to adjust the nutritional balance and symmetry of the recommended foods. In view of the existing problems in the current nutritional balance algorithm, such as slow convergence speed and insufficient local search ability, the autonomous optimization (AO) adjustment strategy, the self-adaptive adjustment strategy, and the two-sided mirror principle to optimize boundary strategy are introduced in the MOEA/D. According to the characteristics of the food nutrition regulation problem, an adaptive food regulation (AFR) adjustment strategy is designed to achieve more accurate nutritional regulation. Based on the above improvements, a food nutritional recommendation algorithm based on MOEA/D (FNR-MOEA/D) is proposed. Experiments show that compared with MOPSO, MOABC, and RVEA, FNR-MOEA/D performs more superiorly in solving the problem of nutritional balance in food recommendation.

Keywords: food recommendation; MOEA/D; CF algorithm; balanced nutritional regulation; multi-objective optimization



Citation: Huang, S.; Wang, C.; Bian, W. A Hybrid Food Recommendation System Based on MOEA/D Focusing on the Problem of Food Nutritional Balance and Symmetry. *Symmetry* **2024**, *16*, 1698. <https://doi.org/10.3390/sym16121698>

Academic Editors: Shi Cheng and Sergei Odintsov

Received: 15 November 2024

Revised: 13 December 2024

Accepted: 18 December 2024

Published: 21 December 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A Recommender System (RS) is an information filtering system that collects and analyzes a large amount of user data, including users' browsing history, purchase records, ratings, search behavior, etc., to build user profiles and understand users' preferences and interest patterns [1]. Based on this information, the recommender system applies various algorithms and models to accurately filter out items or content that may interest the user and recommends them to the user.

The amount of food consumed daily is increasing along with people's living standards, and people are paying more and more attention to the topic of dietary health [2–5]. It might be challenging to guarantee a balanced daily intake of nutrients because many people frequently lack a thorough awareness of the nutritional worth of food. Numerous studies have demonstrated that chronic diseases like diabetes, heart disease, and others can be brought on by an improper diet over an extended period of time [6–9].

Reasonable proportionality among foods is necessary for a healthy diet in order to provide a balanced intake of nutrients. Through the application of recommender technology, the food RS can assist users in logically planning their nutrient intake issues and guaranteeing their daily nutritional needs while accommodating their preferences.

Food RSs usually evaluate users' dietary preferences based on their eating records and suggest foods that they would enjoy based on those choices. Conventional recommendation methods work well with rectangular data sets, but they frequently fail to produce satisfactory results when it comes to food intake and nutritional balance [10,11].

To simultaneously meet the preferences of users and the requirements of daily nutritional intake, some scholars have introduced the Multi-Objective Optimization (MOO) algorithm into the field of food recommendation, transforming the original recommendation problem into a kind of multi-objective problem (MOP) [12–14]. In contrast to the majority of food RSs, our study focuses more on the problem of meal combinations' nutritional balance in the recommended outcomes. The nine primary nutritional components found in each food are used as the optimization objectives for this reason. The amount of each food consumed is adjusted to ensure that the discrepancy between the actual intake and the recommended standard amount of each nutritional component stays within a small range. A hybrid food RS based on the improved MOEA/D has been created as a result. The experimental findings demonstrate that this system is capable of both meeting the dietary requirements of users and successfully recommending foods that they enjoy.

The purpose of the hybrid food recommendation system is to ensure that daily diet can not only meet the needs of the human body but also prevent excessive nutrient intake by balancing the intake of various nutrients. From the aspect of nutritional balance, the system will consider the balanced combination of multiple nutritional elements such as protein, fat, vitamins, etc. This is like a symmetrical structure. Each nutritional element has its important position and is interrelated and mutually restricted, jointly maintaining the normal operation of the body. This kind of balanced relationship reflects a kind of symmetry. In future work, by considering different groups of people, the time dimension of diet, and food sources, etc., these all reflect the characteristics of symmetry in food recommendation. Through multi-faceted adjustments by the food recommendation system to achieve a balanced and symmetrical structure, we can thereby find the most suitable diet method for users.

The main contributions of this paper can be summarized as follows: (1) we proposed a hybrid food recommendation model based on MOEA/D, taking the nine nutritional elements in food as the optimization goals to regulate the intake of each food; (2) we improved the MOEA/D, and proposed four improvement strategies, which improved the performance of the MOEA/D; (3) we conducted a series of experiments based on the real data set, and the experiments proved the effectiveness of the improved MOEA/D in the optimization of food nutritional balance.

The remaining part of this paper is organized as follows: the second section reviews the relevant works in the field of food recommendation. The third section constructs a hybrid food RS based on MOEA/D. The fourth section describes the algorithm in detail. The fifth section presents the experimental data and results. The sixth section presents the conclusion and the outlook for future work.

2. Related Work

Personalized recommendation methods mainly include content filtering [15] and collaborative filtering (CF) [16]. Among them, the CF algorithm is used to calculate the similarity between users or items, so as to make preference recommendations for users, which can better discover users' potential interests and preferences, and thus recommend a wider range. As a tool to help people choose foods, the food RS is closely related to people's nutritional health. Compared with the traditional food RS, more nutritional and health requirements have been added [17–21]. Regarding the issue of food nutritional balance in food recommendations, Rostami et al. [22] developed a new hybrid food RS, which conducts food content recommendations and user recommendations, respectively, through two stages of graph clustering and deep learning. Given the significance of individual needs, Chen et al. [23] proposed a collaborative recipe knowledge network and defined a mechanism for user health degree scoring in order to construct a health-aware meal recommendation model that takes into account users' dietary preferences and health needs.

Based on a healthcare data set, Iwend et al. [24] presented a deep learning solution that uses machine learning to increase the accuracy of recommendations and automatically identify food products that patients require based on their characteristics. Mathur et al. [25] presented a graph-based recommender system that uses user feedback to identify similar products rather than explicit item evaluations to make recommendations. In order to achieve a more accurate balance of multiple nutrients, this paper uses an MOO algorithm to adjust the intake amount of each food.

In scientific application research, the problem of simultaneously optimizing multiple mutually conflicting objectives is usually called an MOP [26]. A simple MOP can be solved by means of mathematical programming, but when facing a high-dimensional MOP, it will become extremely difficult [27–30]. To this end, researchers have introduced an evolutionary algorithm with heuristic search characteristics to solve the MOP, such as NSGA-III [31], ACO [32], PSO [33], and so on.

Although the above algorithms have made considerable progress in the field of MOO, they continue to confront issues when dealing with high-dimensional MOPs [34]. Most evolutionary algorithms optimize the MOP as a whole, whereas the MOEA/D employs a decomposition strategy [35], decomposing the MOP into a set of single-objective problems and using the relationship between neighbors to share evolutionary information; this overcomes the limitations of evolutionary algorithms based on the Pareto dominance relationship, and has advantages such as high efficiency and strong optimization ability when dealing with MOPs.

With the in-depth research on the MOEA/D, many improvement strategies have also been frequently proposed. By using the binding mechanism between the person and the weight vector as well as the external population's replacement method for the invalid weight, Hu et al. [36] re-adjusted the weight vector. Wang et al. [37] successfully enhanced the search capability in the later phase of the algorithm by proposing three enhancement techniques for issues including the lack of species variety and the sluggish convergence speed present in the MOEA/D's later phase. By incorporating the useful information from the iterative MOEA/D process into the new update procedure, Zhang et al. [38] suggested an MOEA/D based on information feedback (MOEA/D-IFM). Cao et al. [39] strove to balance the convergence and diversity of the multi-objective evolutionary algorithm, adopted a two-stage evolutionary strategy, and proposed a new algorithm MOEA/D-TS for solving MOPs.

Although the MOEA/D has achieved great success in the field of MOO, according to the known literature, it has not yet been applied in the field of food recommendation. We introduced the MOEA/D algorithm into the food recommendation system and improved it in terms of convergence speed and population diversity. In view of the characteristics of food nutrition optimization, the AFR adjustment method has been added, which can make a balanced adjustment to the food consumption amount according to the nutritional content of the food. Based on the above improvement strategies, the hybrid food RS based on MOEA/D was designed.

3. The Hybrid Food RS Based on MOEA/D

When studying the issue of personalized food recommendation, both the user's preferences and the situation of nutritional balance are taken into consideration. Regarding user preferences, when users choose foods, they often choose the foods they like or are familiar with, while often ignoring the unfamiliar ones, which will lead to the monotony of the user's daily diet. Therefore, while considering recommending the foods that the user may like, the diversity of the diet also needs to be considered. For nutritional balance, users hope to achieve a healthy daily diet and meet the body's intake requirements for various nutritional elements, so the precise control of the nutrients ingested is required. In view of the above analysis, the hybrid food RS has been studied to solve the following problems: (1) recommend daily dietary recipes for users; and (2) conduct nutritional balance blending of the recommended foods every day to meet the needs of various nutrients.

3.1. The Framework of the Hybrid Food RS

In this section, the MOEA/D based hybrid food RS framework is demonstrated. The multi-objective hybrid food recommendation framework consists of two important modules: the recommendation module and the optimization module. First, the recommendation module searches for users with similar eating habits as the target user through a CF algorithm to filter out the food products that the target user may be interested in. In Section 3.2, the Pearson correlation coefficient is used to calculate the similarity between two users. In Section 3.3, MOEA/D is used to nutritionally optimize the foods so that the recommended foods meet the daily nutritional intake criteria. The model framework is shown in Figure 1.

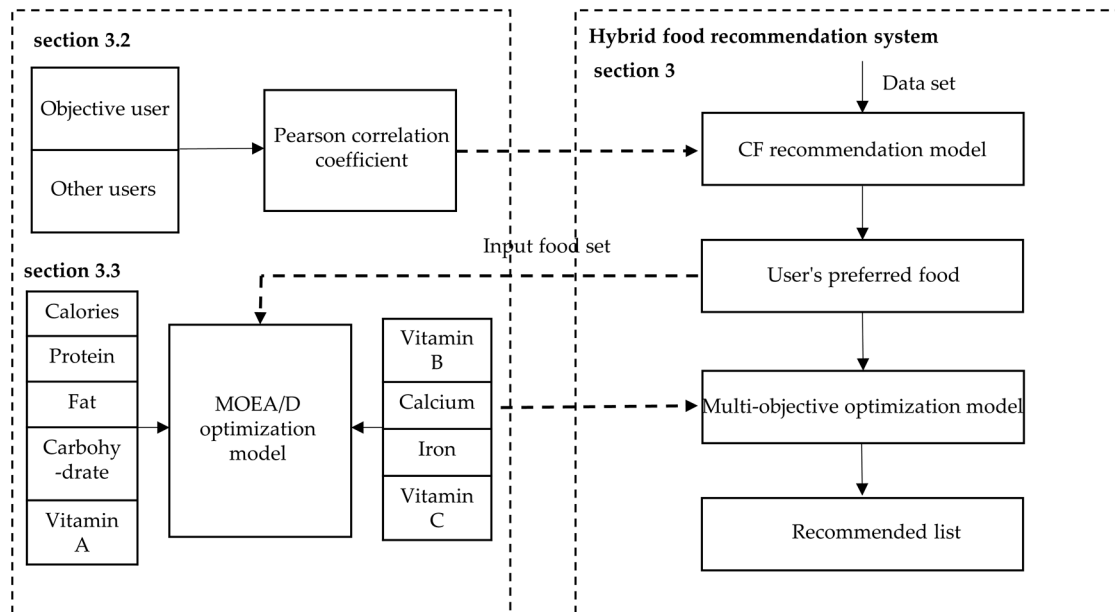


Figure 1. Framework of the hybrid food RS.

3.2. The CF Recommendation Model Based on User Preferences

(1) User preferences: track and record the daily dietary status of users, and finally, form the dietary data of users. In the CF model based on user preferences, it is necessary to calculate the similarity according to the dietary data of users. Due to the difference in the scale of recording user data, this will have an impact on the accuracy of food recommendation. Therefore, a user–dietary preference matrix $p(a, b)$ is constructed to record the dietary preferences of users, and the matrix is as follows:

$$p(a, b) = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1b} \\ p_{21} & p_{22} & \cdots & p_{2b} \\ \vdots & \vdots & & \vdots \\ p_{a1} & p_{a2} & \cdots & p_{ab} \end{bmatrix} \quad (1)$$

In the above matrix, $p(a, b)$ represents the b kinds of food that have been consumed by a group of users, and p_{ab} is the preference degree of user u_a for food F_b . F is the food set, and F_b represents food item b in the food set.

The solution of p_{ab} is as follows:

$$p_{ab} = D_{ab}/T_a \quad (2)$$

where D_{ab} is the consumption frequency of food F_a by user u_a , and T_a is the total consumption frequency of user u_a .

(2) Rating prediction: represent the frequency of the users' consumption of food in the form of scoring. The more a user eats a certain kind of food, the higher the score of this

food. In order to prevent the cold start problem caused by too little new user data, the Slop One algorithm [40] is adopted to score and predict the foods that the user has not eaten. The algorithm is as follows:

$$dev_{ji} = \sum_{u \in (I_i \cap I_j)} \frac{p_{uj} - p_{ui}}{card(I_i \cap I_j)} \quad (3)$$

$$p_{aj} = \frac{\sum_{i \in R(a) \cap D(j)} (p_{ai} + dev_{ji})}{card(R(a) \cap D(j))} \quad (4)$$

where dev_{ji} is the preference offset between the foods F_i and F_j . I is the set of users, and I_i denotes the set of users who prefer food F_i . $card(I_i \cap I_j)$ represents the total number of users who have a preference for both food F_i and food F_j , simultaneously, where p_{uj} represents the user's rating of the food F_j . $R(a)$ represents the set of food preferences by u_a , $D(j)$ represents the set of foods that can perform dev calculation with food F_j . $card(R(a) \cap D(j))$ represents the number of foods in the set of foods that are both foods eaten by u_a and can perform dev calculation with food F_j . Finally, the user u_a 's rating of food F_j is obtained p_{aj} .

(3) Similarity calculation: in order to better calculate the similar relationship between the target user and other users, the Pearson correlation coefficient [41] is adopted as the standard for judging the similarity between users. The similarity between user u_a and user u_b is defined as:

$$sim(u_a, u_b) = \frac{\sum_{n \in R(a) \cap R(b)} (p_{a,n} - \bar{p}_a)(p_{b,n} - \bar{p}_b)}{\sqrt{\sum_{n \in R(a)} (p_{a,n} - \bar{p}_a)^2} \sqrt{\sum_{n \in R(b)} (p_{b,n} - \bar{p}_b)^2}} \quad (5)$$

where \bar{p}_a and \bar{p}_b represent the mean values of the ratings of the food products by users u_a and u_b . $p_{a,n}$ and $p_{b,n}$ represent the ratings of the users u_a and u_b for the food F_n . $n \in R(a) \cap R(b)$ represents the food items that have been rated by u_a and u_b together.

(4) Recommendation selection: after calculating the similarity between other users and the target user, sort the other users according to the similarity. To prevent excessive recommendation due to overly high user similarity, the k nearest neighbors is adopted to take the first k_1 users, and the first k_2 foods recommended by these users are recommended to the target user. The size of k_1 and k_2 can be determined based on actual recommendations.

3.3. MOEA/D Optimization Model

3.3.1. Multi-Objective Problem Description

MOPs usually have multiple mutually conflicting objectives, which can be expressed as follows, where the decision variable $X = \{x_1, x_2, \dots, x_n\}, x \in R^n$ needs to meet the following constraint conditions:

$$\begin{cases} g_i(x) \geq 0, i = 1, 2 \dots q \\ h_i(x) = 0, i = 1, 2 \dots q \end{cases} \quad (6)$$

where R^n is an n -dimensional decision space that contains all the decision variables. It can be known that $g_i(x)$ defines q inequality constraints, and $h_i(x)$ defines q equality constraints. If the MOPs have m objectives, then the objective function can be expressed as:

$$F(X) = (f_1(X), f_2(X), \dots, f_m(X))^T \quad (7)$$

According to the constraints, find a solution set X^* that satisfies in the decision space such that $F(X^*)$ is minimized.

3.3.2. MOEA/D

MOEA/D decomposes a multi-objective problem into multiple single-objective sub-problems, and by using the neighbor relationship between the sub-problems, it assists the

neighbors in evolving while evolving itself, and finally achieves the optimization of the multi-objective problem. The process of the MOEA/D is as follows:

Step 1: Initialize.

Initialize population $Pop = \{x^1, x^2, \dots, x^n\}$ and fitness value $FV = \{f_1(x), f_2(x), \dots, f_n(x)\}$, and initialize the weight vector $\lambda = \{\lambda^1, \lambda^2, \dots, \lambda^n\}$.

Initialize the size T of individual neighbors according to the size of the population and initialize the neighbor $B(i) = \{b_1, b_2, \dots, b_T\}$ of each individual according to the weight vector.

Calculate the ideal point $z = \{z_1, z_2, \dots, z_m\}^T$, among which m is the target number, and z_i represents the optimal value of the i th target in the current population.

Step 2: Update the population.

Genetic recombination: randomly select two individuals x^j and x^k from the neighbor $B(i)$ of the current individual x^i , and take these individuals as parents to generate new offspring x_{new}^i through genetic variation.

Update the ideal point: calculate the fitness value $FV(x_{new}^i)$ of the new individual. If the fitness value of x_{new}^i is greater than z , then update the ideal point.

$$z = \begin{cases} z & f(z) < f(x_{new}^i) \\ x_{new}^i & f(z) > f(x_{new}^i) \end{cases} \quad (8)$$

Update the neighbor: for the neighbor $x^j, j \in B(i)$ of the individual x^i , if $g^{te}(x_{new}^i | \lambda, z) < g^{te}(x^j | \lambda, z)$, then the individual x^j will be updated, and the cumulative update is nr times, and nr is a fixed value.

Step 3: Iteration stops.

If the stop condition is satisfied, then stop the iteration and output the optimal solution set; otherwise, continue to execute Step 2.

3.3.3. Objective Function and Constraints

In this paper, the improved MOEA/D is used to adjust the multi-objective food nutrition. After providing the recommended food to the users, by calculating the nutritional content of the food, and according to the standard of the daily intake of nutrition, regulate the intake amount of the food. In this paper, it is set that the recommended food is eight different foods, and the nine nutritional elements among them are adjusted.

The setting of the objective function is as follows: the recommended food types are eight kinds, and the nine nutrients involved are calories, protein, fat, carbohydrate, vitamin A, vitamin B, calcium, iron, and vitamin C, respectively. The absolute value of the difference between the values of these nine nutrients' contents and the values of the standard contents consumed per day is taken as the objective function of this experiment. These are defined as follows:

$$f(x^i) = \left| x_{1 \times 8}^i \times A_{8 \times 9} - S_{1 \times 9} \right| \quad (9)$$

where x^i represents the i 'th food combination, and each food combination contains eight different foods. A represents the relationship matrix of the nine nutritional elements corresponding to the eight foods, in which $a^{ij} (i = 1, 2, \dots, 8; j = 1, 2, \dots, 9)$ represents the content of the j 'th nutrition in the i 'th food. S is the recommended daily intake of the nine nutritional elements, and the standard refers to the "Reference Intakes for Dietary Nutrients for Chinese Residents" [42].

Constraint conditions are as follows: in this paper, considering the problem of excessive consumption of a single food caused by reducing errors in the process of optimization, it is set that the maximum amount of each food consumed is 400 g and the minimum is 10 g.

4. Description of the Algorithm

The hybrid food recommendation process is divided into the following two stages: firstly, through the Algorithm 1 to recommend the foods that the user may be interested in,

and secondly, through the Algorithm 2 to carry out the balanced deployment of nutrients for the recommended foods.

4.1. CF Algorithm Based on User Preferences

The CF algorithm recommends suitable foods to the target user through the similarity between users, prevents the cold start problem caused by the small amount of data of new users, and predicts the data of users through the Slop One function. According to the above model description, the framework of the CF algorithm is designed.

Algorithm 1. (Framework of the user-based CF).

Inputs: *data*- User dietary preference matrix, *k* nearby neighbors

Outputs: *ReFood*- The food combination recommended to users

Refining the user preference matrix with the Slop One function: $p_{aj} \leftarrow p_{ai} + dev_{ji}$.

for $i = 1$ to N **do**//Where N represents the number of users in the user preference matrix

$FR \leftarrow \text{sim}(u_a, u_b)$ //Calculate the similarity between target user and other users.

end for

$FR \leftarrow \text{Sort}(FR)$ //Sorting users by similarity size in matrix FR .

Select the k_1 users with the highest similarity to the user.

$ReFood \leftarrow \text{Select}(FR(u_i))$ //Each user individually selects the top k_2 food items that //have the highest ratings and have not been rated by the target user.

Finally obtain the *ReFood* that conforms to the preferences of the target users.

4.2. The Improvement Strategy of the MOEA/D

The MOEA/D updates the population through the differential evolution strategy. With the evolution of the population, the gap between individuals will gradually narrow, resulting in insufficient local search ability in the later phase of the algorithm. In the field of food recommendation, it is necessary to tune nine nutritional indicators. Facing the high-dimensional MOP, some improvements need to be made to the MOEA/D in order to expect to allocate food nutrition more accurately. For these problems, four improvement strategies are proposed: firstly, to improve the convergence speed of the algorithm, AO optimization is added in the MOEA/D. Secondly, to improve the later phase search ability of the algorithm and enrich the diversity of the population, the self-adaptive neighborhood adjustment strategy is improved, and the self-adaptive mutation probability adjustment is added while adjusting the size of the individual neighborhood. Thirdly, the two-sided mirror reflection algorithm is improved, and the use frequency of the algorithm is controlled through the iterative process. Fourthly, in view of the characteristics of the food recommendation problem, the AFR adjustment method is added to achieve more accurate adjustment of nutritional content.

4.2.1. AO Optimization Strategy

AO optimization is a kind of self-optimization strategy. When the population iterates to select individuals each time, AO optimization needs to be carried out for each selected individual. However, the time complexity of AO optimization is relatively high, so the probability of optimization needs to be adjusted. Whenever an individual is selected for AO optimization, the Tchebycheff value of the individual needs to be calculated first. The size of the adjustment variable is formulated according to the boundary value of each decision variable of the individual, and the value of the adjustment variable is determined by the normal distribution function. The normal distribution function is defined as follows:

$$\sigma = \frac{\sqrt{u-d}}{2} \quad (10)$$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (11)$$

where μ represents the mean of the normal distribution, and the value is 0. σ represents the standard deviation of the normal distribution, and the value of σ here is determined by the boundary value of the decision variable, among which u represents the upper boundary and d represents the lower boundary.

In order to verify the effectiveness of the AO optimization strategy, a controlled experiment was conducted on the test function ZDT1 [43]. Figure 2 shows the comparison of the results after running 100 generations with the test function ZDT1. It can be seen that the MOEA/D with the AO optimization strategy added has a faster convergence speed.

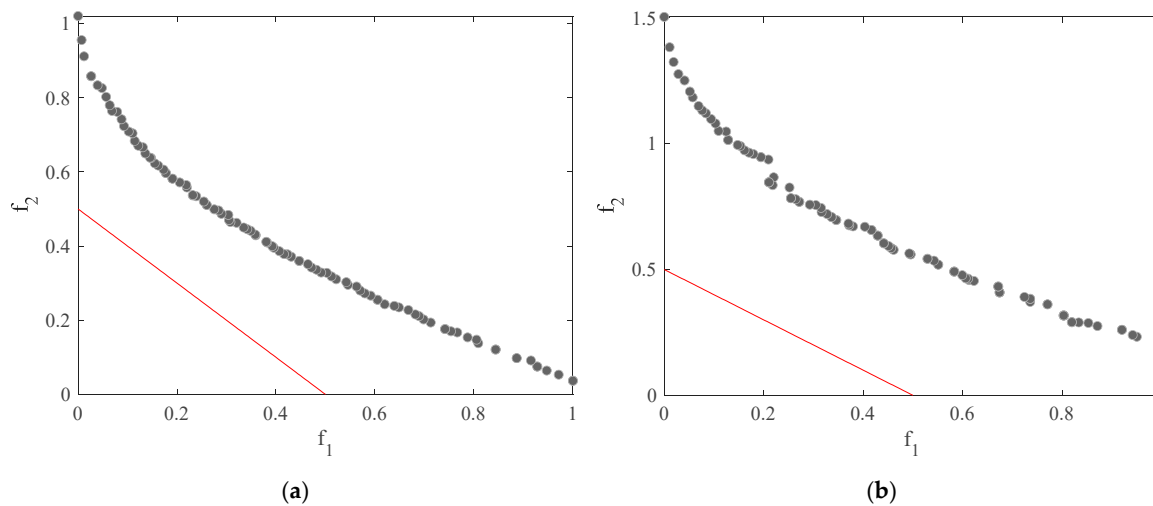


Figure 2. The distribution map of the Pareto optimal solution set of the MOEA/D. (a) With the AO optimization strategy added; (b) without adding the AO optimization strategy.

4.2.2. Self-Adaptive Adjustment Strategy

In the optimization strategy of MOEA/D, Xu et al. [44] proposed a strategy of adaptively adjusting the neighborhood. This strategy dynamically adjusts the neighborhood by judging the number of individuals around the sub-problem and solves problems such as sparse individuals around individual sub-problems and the aggregation of individuals in the optimization process. However, in the face of high-dimensional MOPs, simply changing the size of the neighborhood cannot optimize the poor sub-problems well. Therefore, in order to increase the diversity of the population, while adaptively adjusting the neighborhood, a strategy of adaptively adjusting the mutation probability is added.

The process of the self-adaptive adjustment strategy is as follows:

Step 1: find out the maximum value of all the targets. For all the sub-problems $i = 1, 2, \dots, N$, calculate the maximum value of each target $z_{max} = \{z_1, z_2, \dots, z_m\}$; m is the target number of the sub-problems, $z_j = \max_{1 \leq j \leq m, 1 \leq i \leq n} \{f_j(x^i)\}$.

Step 2: calculate the vertical distance of each individual to each sub-problem $xd(i, j) = D_{\perp}(x_i, \lambda_j)$.

$$D_{\perp}(x_i, \lambda_j) = \left\| \overline{L}(x_i) - \frac{\lambda_j^T \overline{F}(x_i) \lambda_j}{\lambda_j^T \lambda_j} \lambda_j \right\| \quad (12)$$

where $\overline{F}(x_i)$ is the regularization objective vector of individual x_i , and x_j objective function is regularized as follows:

$$L(x) = \frac{f_j(x) - z_j}{z_j - z_j^{nad}} \quad (13)$$

where $f_j(x)$ represents the objective function of the x_j , z_j indicates the current best point of the x_j , and z_j^{nad} represents the worst point.

Step 3: statistical sorting. For each individual, sort the distances to the sub-problems $k = \text{sort}(xd(i, j))$, and count the closest distance. Its density is represented by id . The density of individuals around the j 'th sub-problem is expressed as $id_j = id_j + \cdot$.

Step 4: adjust the neighborhood size. Through the size of the number of individuals around each sub-problem, adjust the neighborhood size of the sub-problem, and the individual closest to the sub-problem will therefore be the solution of this sub-problem.

Step 5: adjust the mutation probability. When an individual conducts polynomial mutation, the polynomial mutation probability is adjusted by calculating the size of the neighborhood of the sub-problem.

The method of adjusting the neighborhood size is as follows: when $id_j = 1$, it indicates that there is exactly one solution around this sub-problem, and at this time there is no need to adjust the size of the neighborhood. When $id_j = 0$, it indicates that there is no appropriate solution around this sub-problem, and at this time the neighborhood should be expanded as follows: $T(j) = T(j) + N$. When $id_j > 1$, it means that there are multiple solutions around the sub-problem at this time, and in this case, the size of the neighborhood needs to be reduced as follows: $T(j) = T(j) - N$. N is a fixed value, and the size is $T/2$.

The adjustment of the mutation probability is as follows: when $id_j = 1$, it indicates that there is exactly one solution around this sub-problem, and at this time there is no need to adjust the mutation probability. When $id_j = 0$, it indicates that there is no appropriate solution around this sub-problem, and the mutation probability needs to be increased as follows: $Prom = Prom \times (1 + p)$. When $id_j > 1$, it means that there are multiple solutions around the sub-problem at this time, and in this case, the mutation probability needs to be reduced as follows: $Prom = Prom \times (1 - p)$. p is a fixed value.

The strategy of adaptively adjusting the mutation probability is mainly to solve the insufficiency of the diversity of the population in the later phase, and to increase the mutation amplitude of the sparse individual sub-problem in order to expect to find a better solution. Figure 3 is the result of running on the DTLZ3 [45] test function.

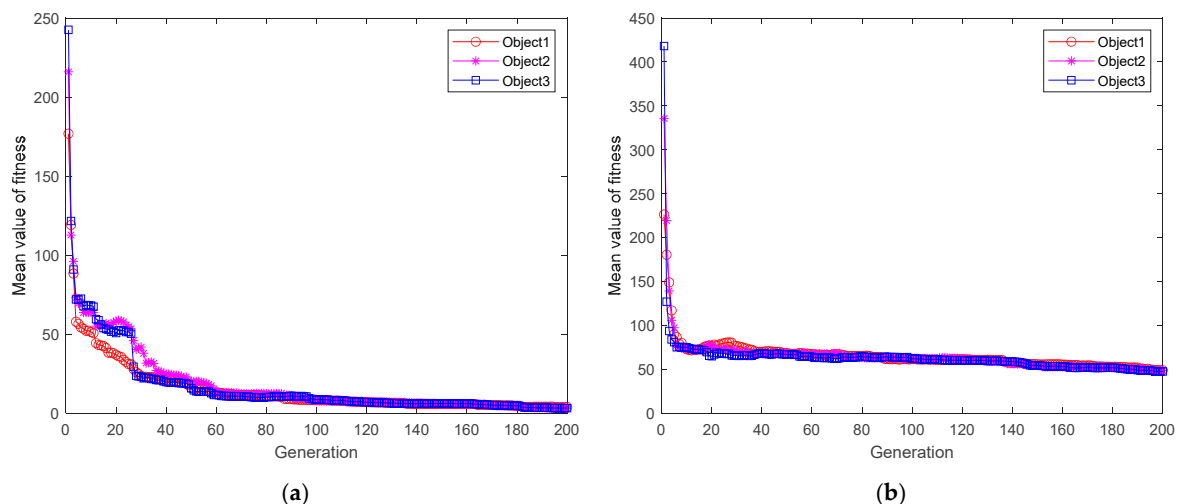


Figure 3. The mean curve of the Pareto optimal solution set of the improved MOEA/D. (a) With the self-adaptive adjustment strategy added; (b) without adding the self-adaptive adjustment strategy.

As shown in Figure 3, without adding the self-adaptive adjustment strategy, the algorithm reaches the 50'th generation, and due to the insufficient diversity of the population, the convergence trend of the population's Pareto optimal solution set gradually becomes smaller, and it is difficult for the algorithm to continue to optimize. After adding the self-adaptive adjustment strategy, the evolution of the population does not stagnate, and the optimization efficiency of the algorithm is improved.

4.2.3. The Principle of the Two-Sided Mirror to Optimize the Boundary Strategy

In the simulation experiment of MOEA/D, it is often found that the phenomenon of uneven distribution of the solution is set, and it is easy to aggregate at the edge. This situation leads to most of the middle area being unexplored. This is because the MOEA/D uses the result of marginalizing the out-of-bound individuals, as shown in the following formula:

$$y' = \begin{cases} d, & y < d \\ u, & y > u \\ y, & \text{other} \end{cases} \quad (14)$$

where y is the newly evolved individual, d is the upper limit of the boundary, and u is the lower limit of the boundary. In response to this problem, it is optimized and processed through the two-sided mirror reflection algorithm.

The principle of the two-sided mirror reflection algorithm is to regard the upper bound and the lower bound as two mirrors. When an individual is about to cross the boundary, it will be reflected according to the magnitude of the individual's crossing of the boundary, which is defined as follows:

$$y' = \begin{cases} u - w \times \text{mod}(y - u, u - d), & y > u \\ d + w \times \text{mod}(d - y, u - d), & y < d \end{cases} \quad (15)$$

The use of the two-sided mirror reflection algorithm can effectively avoid the aggregation of individual boundaries of the population, but it also brings new problems. Due to the randomness of the algorithm, the convergence speed will be slowed down at the later phase. In order to overcome this problem, this paper adds a linear probability to adjust the use of the algorithm. In the early phase of population iteration, the adjustment of the algorithm is increased to increase the diversity of the population. In the later phase of population iteration, the adjustment margin of the algorithm is decreased in order to increase the convergence speed of the algorithm, which is defined as follows:

$$w = 1 - (\text{gen}/\text{Generation})^2 \quad (16)$$

where w is the probability of using the two-sided mirror reflection boundary algorithm, gen is the current iteration number and Generation is the total iteration number. w increases as the iteration number increases.

Figure 4 shows the result of running the DTLZ4 [46] test function 200 times. As shown in Figure 4, the population with the two-sided mirror reflection boundary algorithm added will have the phenomenon of detaching from the boundary aggregation during the optimization process.

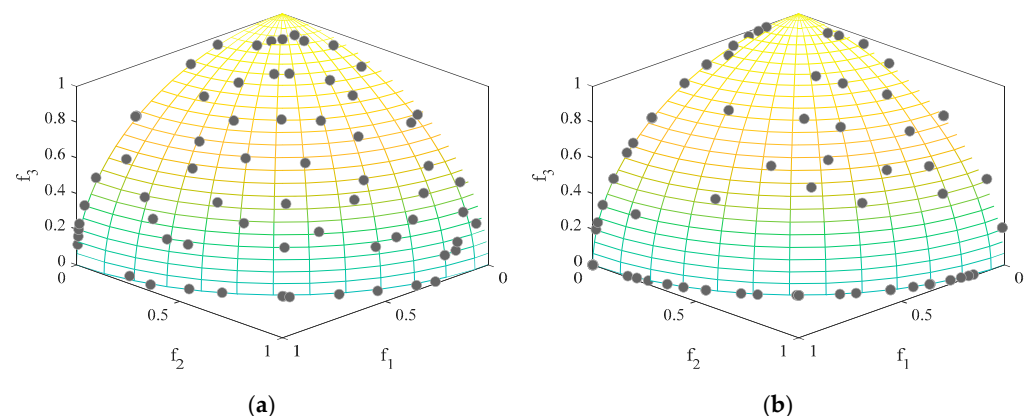


Figure 4. The distribution map of the Pareto optimal solution set of the improved MOEA/D. (a) With the two-sided mirror strategy added; (b) without adding the two-sided mirror strategy.

4.2.4. AFR Food Nutrition Regulation Strategy

Food nutrition regulation is a high-dimensional multi-objective optimization problem, and it is difficult to achieve a good result by only relying on the MOEA/D. For this reason, based on the MOEA/D and aiming at the characteristics of the food nutrition regulation problem, the strategy of AFR regulation is proposed. When the new individual has evolved completely, it is necessary to compare the nutritional elements contained in the food recommended by the individual with the standard nutritional elements consumed daily, which are counted by the two arrays, P and N , respectively,

$$\begin{cases} P, (x \times A)^i > S^i \\ N, (x \times A)^i < S^i \end{cases} \quad i = (1, 2, \dots, 9) \quad (17)$$

where P is the positive quantity array, N is the negative quantity array, and S is the intake standard of nine kinds of nutrients. When the nutrients contained in the food are greater than the standard value, this kind of nutrient is put into the positive quantity array, indicating that it exceeds the standard. When the nutrients contained in the food are less than the standard value, this kind of nutrient is put into the negative quantity value, indicating that it still fails to reach the standard. AFR realizes the optimization of nutritional balance through the adjustment of different arrays.

The flow chart of AFR regulation is shown in Figure 5.

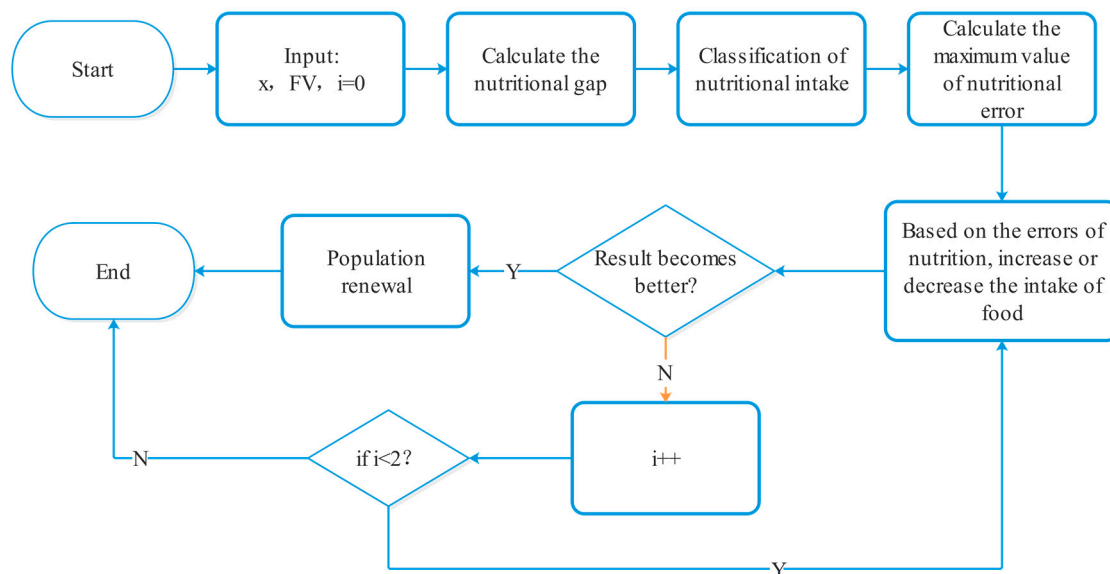


Figure 5. AFR adjustment flow chart.

In the AFR reconciliation process, we categorize foods by calculating the gap between their nutrient intake and the standard daily intake, where N is an array of positive quantities that contain nutrients that are greater than the standard nutrient intake, and P is an array of negative quantities, which contain nutrients that are less than the standard nutrient intake. After categorizing the nutrients, the nutrients in the N and P groups with the largest error from the standard nutrient intake were calculated as shown in Formula (18),

$$E = \frac{|C_i - S_i|}{S_i} \quad (18)$$

where C represents the intake of nutrients in the diet, and S is the recommended daily intake of the nine nutritional elements. After selecting the two nutrients, the foods with the highest content of the two nutrients were calculated, and the intake of the nutrients was controlled by adding or subtracting the intake of the foods closest to the standard values.

4.3. FNR-MOEA/D

The results through improvement show that the first three strategies have improved the performance of the MOEA/D. On this basis, combined with the strategy of AFR food nutrition regulation, FNR-MOEA/D is proposed. The FNR-MOEA/D aims to comprehensively utilize the performance advantages of the three improved strategies and conforms to the unique characteristics of food nutrition optimization.

The FNR-MOEA/D adopts the basic framework of the MOEA/D and adds the mutation probability adjustment and the self-adaptive neighborhood adjustment mechanism in the evolutionary process. Before the evolutionary process, the selected population individuals need to adjust the adaptive neighborhood size and mutation probability and have the probability to self-adjust to select more excellent individuals for the next step of population evolution. In the evolutionary process, the evolutionary operator randomly selects individuals within the neighborhood and uses polynomial mutation to generate a new individual y . By comparing the Tchebycheff value of y with that of nr individuals in the neighborhood, if the Tchebycheff value of y is better than the value of the individuals in the neighborhood, then the new individual y will update these neighborhood individuals. In addition, when conducting individual evolution, the two-sided mirror theory will be used to deal with the problem of individual cross-border.

The implementation steps of the FNR-MOEA/D are as follows:

Step 1: initialize the population $Pop = \{x_1, x_2, \dots, x_n\}$ and fitness value $FV = \{f_1(x), f_2(x), \dots, f_n(x)\}$, initialize the weight vector $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$, initialize the size T of individual neighbors according to the size of the population, and initialize the neighbor $B(i) = \{b_1, b_2, \dots, b_T\}$ and $z = \{z_1, z_2, \dots, z_m\}$ of each individual according to the weight vector. m is the target number, indicating the optimal value of the i 'th target in the contemporary era.

Step 2: the density of the individuals of the current sub-problem is computed by a self-adaptive adjustment strategy, and the size of the current neighborhood T as well as the size of the mutation probability dm is adjusted according to the size of the density.

Step 3: the choice of whether the current individual's neighborhood takes its own neighborhood or all of the individuals is based on the magnitude of the probability δ , which is defined as follows:

$$p = \begin{cases} B(i) & \text{if } rand < \delta \\ \{1, 2, \dots, N\} & \text{otherwise} \end{cases} \quad (19)$$

Step 4: AO optimization. Optimize the current individual x^i .

Step 5: polynomial mutation. Adjust the size of the variation amplitude according to the variation parameter dm obtained in Step 2, and randomly select two individuals in $B(i)$ to polynomially mutate with the current individual to produce a new offspring x_{new}^i .

Step 6: in the process of performing polynomial variations, if there is a problem of data crossing the boundary, the adjustment is made using the two-sided mirror principle.

Step 7: solve to take the fitness value. Solve for the fitness value of the newly generated offspring $f(x_{new}^i)$.

Step 8: optimize new offspring x_{new}^i based on AFR regulation and calculate the fitness value $f(x_{new}^i)$.

Step 9: update the ideal point z . Calculate the fitness $f(z)$ of z , and determine whether to update.

$$z = \begin{cases} z & f(z) < f(x_{new}^i) \\ x_{new}^i & f(z) > f(x_{new}^i) \end{cases} \quad (20)$$

Step 10: update the neighbors of the current individual. If $g^{te}(x_{new}^i | \lambda, z) < g^{te}(x^j | \lambda, z)$, $j \in B(i)$, update the previous nr neighbors.

Step 11: judge whether the algorithm meets the termination conditions. If it meets the conditions, stop the iteration and output the optimal solution. If it does not meet the conditions, return to Step 3 and, according to the self-adaptive adjustment strategy, adjust the neighborhood size T of each individual.

The framework of FNR-MOEA/D is as follows:

Algorithm 2. (Framework of the FNR-MOEA/D).

Inputs: X - individual initial position, z - reference point, λ - weight vector of MOPs, B -neighborhood
Outputs: FV - object value of X
for gen = 1 to Generation **do**
 if gen > 1 **then**
 Calculate the distance between each x and the sub-problem, and sort them.
 $id_j \leftarrow id_j + +$ // Record the number of individuals around each subproblem.
 // the neighborhood and mutation probability of the X are adjusted adaptively.
 $T(j) \leftarrow T(j) \pm N$ // Adjusting neighborhood size
 $Prom(j) \leftarrow Prom(j) \times (1 \pm p)$ // Adjusting the probability of mutation
 end if
 for $i = 1$ to N **do** // Where N is the number of individuals of the population.
 if $rand() < \delta$ **then**
 $p = B(i)$ // Obtain the neighborhood of the current individual.
 else
 $p = X$ // Obtain all individuals of the entire population.
 end if
 if $rand() < \xi$ **then** // ξ is a number less than 1
 Carry out AO optimization for the current individual: $x^i \leftarrow AO(x^i)$.
 end if
 Randomly select two individuals x_j, x_k from P .
 $x_{new}^i \leftarrow x^i + CF \times (x_j - x_k)$. // Generation difference individual by using DE.
 if $rand() < Prom(i)$ **then**
 Polynomial variation: $x_{new}^i \leftarrow x_{new}^i + \beta_i$.
 end if
 Optimize the boundary through the two-sided mirror theory: $x_{new}^i \leftarrow x_{new}^i + \gamma_i$.
 $x_{new}^i \leftarrow AFR(x_{new}^i)$ // AFR regulation strategy.
 if $f(x_{new}^i) < f(z)$ **then**
 $z \leftarrow x_{new}^i$
 end if
 if $h < hr$ **then**
 Update neighborhood: $[B(i), FV] \leftarrow update(x_{new}^i, f(x_{new}^i))$.
 $h + +$
 end if
 end for
end for

5. Experiment

5.1. Experimental Data and Process

The experimental data come from a popular health and fitness tracking application called MyFitnessPal (MFP), which records the daily diet and exercise status of users and calculates the daily nutritional intake and the degree of calorie consumption accordingly. The MFP data set recorded the 1.9 million dietary statuses of 9800 users for 71 thousand kinds of foods from September 2014 to April 2015.

The data set is divided into two parts, among which the first part is the dietary records of the users; some of the data are shown in Table 1.

In Table 1, the user_id represents the serial number of the users. The food_ids represents the serial number of each food. The data represent the dining time of the users. The meal_sequence represents the meal sequence on a specific date, and meal_sequence = 1 indicates the first meal of each day.

Table 1. Users' dietary records.

Meal_Id	User_Id	Date	Meal_Sequence	Food_Ids
1	1	23 September 2014	1	17, 3, 20
2	1	24 September 2014	1	9, 3, 13, 7
3	1	25 September 2014	1	16, 9, 3, 22
4	1	26 September 2014	1	9, 3, 13, 7
5	1	27 September 2014	1	3, 7

The second part is the recording of food data, as shown in Table 2. The food_ids represents the sequential number of each kind of food. The title indicates the name of each kind of food. Each food name is composed of a triple of words related to food and cooking and is separated by double underscores. For example, “egg_dairy__dairy_product__milk” (egg, dairy product, milk) is a remarkable feature of a triple.

Table 2. MFT food record.

Food_Ids	Title
1	egg_dairy__dairy_product__milk
2	meat_beef_eye_fillet
3	staple_root_and_tuber__taro
4	dessert_cake__eggies

The experiment is divided into two stages. In the first stage, a CF algorithm based on user preferences is used to recommend foods that the user may be interested in. The recommended results are steak, carrot, walnut, rice, chicken, taco, yogurt, and espresso, which is called Combination 1. In the second stage, the nutritional balance is adjusted through the FNR-MOEA/D. Since the effectiveness of the CF algorithm in the first stage has been experimentally verified in ref. [47], the focus of the work in this paper is on the second stage. In order to verify the performance of the FNR-MOEA/D under different food combinations, this paper uses the recommended results of ref. [47] (rice, pork, lettuce, pear, eggplant, tofu, shrimp, and cucumber, which is called Combination 2) as the second combination.

The difference between these two food combinations lies in the different degrees of nutritional balance they contain. The nutritional elements used in the experiment in this article and the experiment in ref. [47] are exactly the same, which are calories, protein, fat, carbohydrates, vitamin A, vitamin B, calcium, iron, and vitamin C.

In order to judge the degree of nutritional balance of the two food combinations, we compare the amount of nutrients consumed by eating 100 g of each food with the daily standard intake of nutrients. Whichever food combination has a smaller error is judged to have a more balanced nutritional content.

As shown in Figure 6, according to the comparison result, the average error of the nutritional elements contained in various foods in Combination 1 is lower, and the nutritional content is more balanced.

According to whether the nutrition contained in the food is balanced, the experiment is carried out in two situations. The first combination is the one with a relatively balanced content of food nutrition, that is, Combination 1. The eight foods are steak, carrot, walnut, rice, chicken, taco, yogurt, and espresso, respectively. The second combination is the one with an unbalanced content of food nutrition, that is, Combination 2. The eight foods are rice, pork, lettuce, pear, eggplant, tofu, shrimp, and cucumber, respectively. The experiment mainly focuses on the optimization of the balance of food nutrition, and the final recommended standard is the individual with the smallest error and in the Pareto optimal solution set.

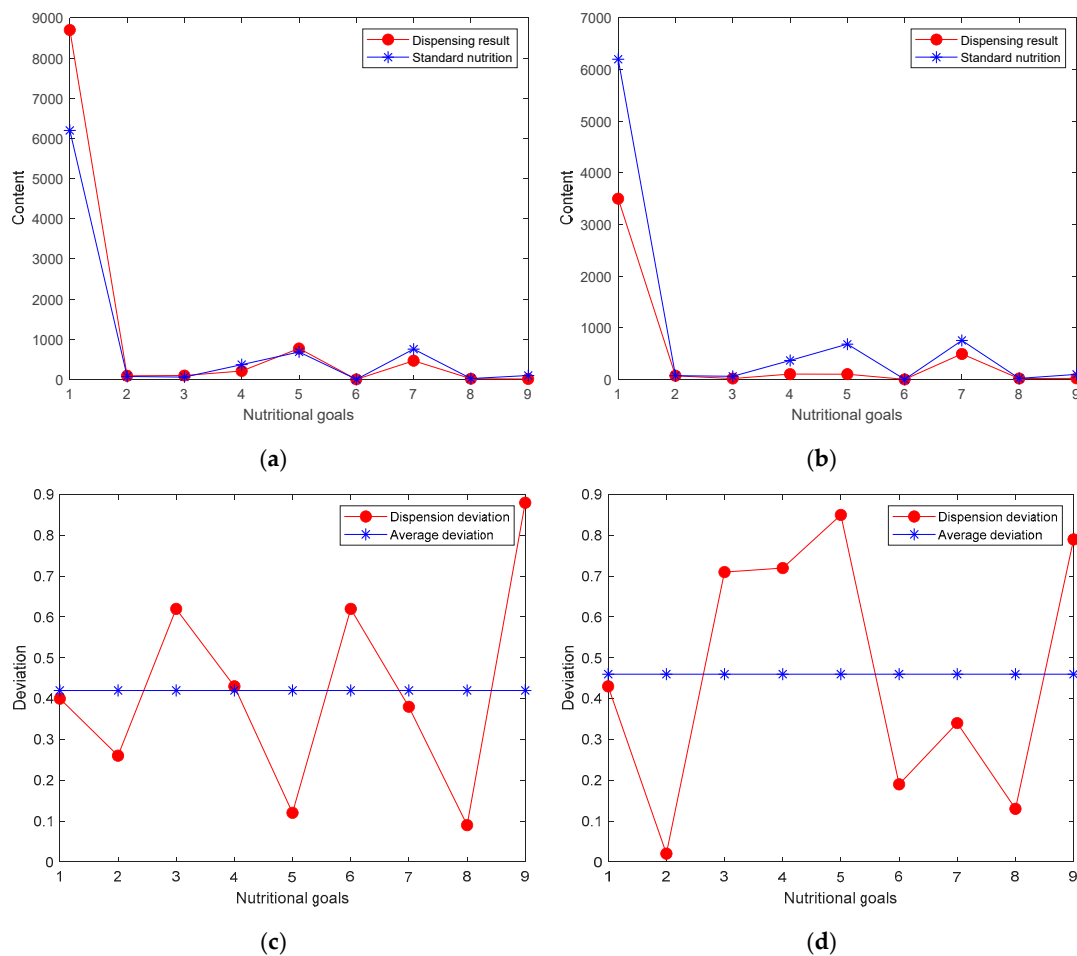


Figure 6. The comparison of error between two kinds of food combinations. (a) Comparison of nutritional intake (Combination 1). (b) Comparison of nutritional intake (Combination 2). (c) Comparison of nutritional errors (Combination 1). (d) Comparison of nutritional errors (Combination 2).

In order to highlight the advantages of the FNR-MOEA/D, it will be compared with the MOPSO algorithm, the MOEA/D, the MOABC algorithm, and the RVEA. The superiority of the tuning results of each algorithm was judged according to the size of the sum of the errors between the tuned food nutrient intake and the standard daily nutrient intake.

The specific parameter settings of the FNR-MOEA/D are shown in Table 3, where some of the parameters are set with reference to the paper [48].

Table 3. The parameter settings of FNR-MOEA/D.

Parameter	Meaning	Value
Generations	The Quantity of Generations	150
N	The Population Size	120
M	The Number of Optimization Objectives	9
T	The Neighbor Size	12
CR	The Crossover Probability	0.5
ProM	The Mutation Probability	1/D

5.2. Experimental Results

(1) The combination of balanced food nutritional content (Combination 1).

The optimization results of the six algorithms under the condition that the eight food combinations contained relatively balanced nutrients are shown in Figure 7. The results of FNR-MOEA/D stabilized around the 40th generation. MOEA/D and RVEA recommended

similar results, the MOABC algorithm gave the worst results, and FNR-MOEA/D was the best recommended result among the five algorithms.

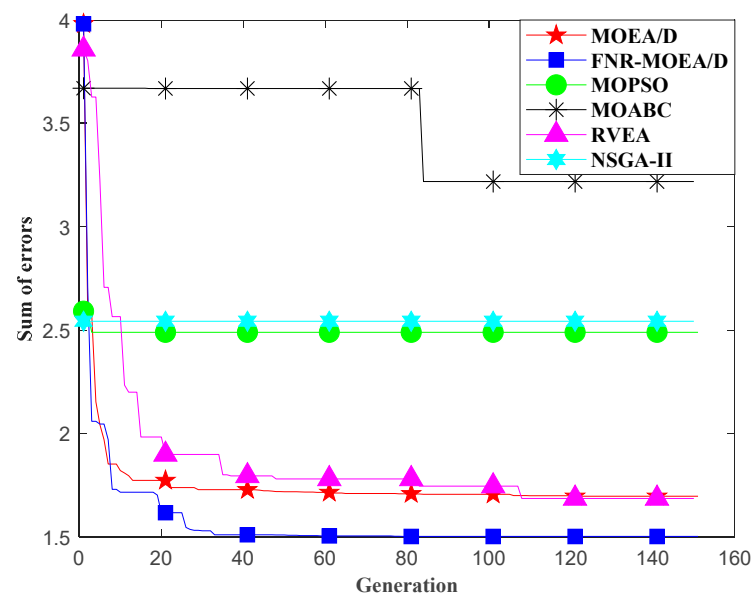


Figure 7. Food nutrition regulation error (Combination 1).

The eight kinds of foods recommended by the FNR-MOEA/D are 24 g of steak, 91 g of carrot, 77 g of walnut, 30 g of rice, 25 g of chicken, 22 g of taco, 160 g of yogurt, and 334 g of espresso. The total error of the nine kinds of nutritional contents is about 154%. The errors of each nutritional content are shown in Table 4 as follows:

Table 4. The results of nutritional allocation (Combination 1).

Nutritional Elements	Standard	Allocate	Deviation	Error
Calories (KJ)	6200	7450.4	1250.4	20.16%
Protein (g)	72.3	71.1	−1.2	1.66%
Fat (g)	59.71	60.3	0.59	0.98%
Carbohydrate (g)	368.03	227.7	−140.33	38.13%
Vitamin A (μg)	683.25	684.3	1.05	0.15%
Vitamin B (mg)	1.4	1.4	0	0
Calcium (mg)	755.32	756.5	1.18	0.15%
Iron (mg)	20.24	21.4	1.16	5.73%
Vitamin C (mg)	97.21	11.6	−85.61	88.07%

In Table 4, the error of vitamin C is the largest, which is 88.07%. The reason is that among the eight foods, only carrots contain a certain amount of vitamin C, and the rest of the foods only contain trace amounts or no such nutrient element, so it is difficult to blend. Then, the error of carbohydrate is 38.13%, which is because there is a large amount of carbohydrates in rice and corn tortillas; however, the consumption amount of the two foods is relatively small, thus resulting in a relatively large error. The error of calories is 20.16%, because walnuts contain extremely high calories, and excessive intake of walnuts leads to a relatively large error. The errors of the other six nutrient elements are within 10%, which belongs to the normal intake range.

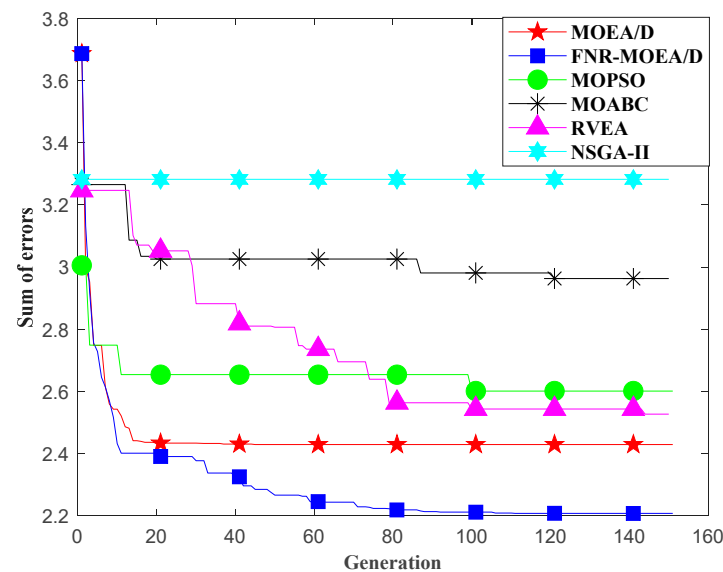
In Table 5, the nutritional errors of all the test algorithms in Combination 1 are counted. Among them, the data marked in bold are the minimum error of each nutritional element, and the data underlined are the maximum error of each nutritional element. On the whole, the average error of the recommended results of the FNR-MOEA/D is smaller, and the errors of various nutrients are more balanced.

Table 5. Comparison of algorithms (Combination 1).

Nutritional Elements	FNR-MOEA/D	MOEA/D	MOPSO	MOABC	RVEA	NSGA-II
Calories (KJ)	20.16%	17.72%	18.64%	22.61%	20.66%	39.16%
Protein (g)	1.66%	5.14%	12.82%	31.42%	5.12%	26.83%
Fat (g)	0.98%	0.54%	6.53%	17.23%	2.86%	0.70%
Carbohydrate (g)	38.13%	41.03%	43.15%	44.87%	37.27%	78.07%
Vitamin A (μg)	0.15%	0.04%	19.35%	16.55%	1.79%	4.17%
Vitamin B (mg)	0	17.14%	26.19%	92.86%	2.86%	3.57%
Calcium (mg)	0.15%	0.44%	20.16%	9.57%	0.62%	79.65%
Iron (mg)	5.73%	3.71%	24.58%	1.19%	8.73%	2.57%
Vitamin C (mg)	88.07%	88.76%	88.74%	85.64%	88.68%	19.55%
Average error	17.22%	19.39%	28.91%	35.77%	18.73%	27.25%

(2) The combination of unbalanced food nutritional content (Combination 2).

In Figure 8, the superiority of FNR-MOEA/D is obvious compared with the other five algorithms, while the difference between the results of the MOPSO algorithm and RVEA is relatively small; NSGA-II has the worst result. From the experimental results, it can be concluded that FNR-MOEA/D performs better in dealing with the problem of the nutritional imbalance of food. The eight food items recommended by FNR-MOEA/D are rice 253 g, pork 13 g, lettuce 399 g, pear 400 g, eggplant 400 g, tofu 105 g, shrimp 36 g, and cucumber 400 g. The total errors of the nine nutrient contents of FNR-MOEA/D are about 225%.

**Figure 8.** Food nutrition regulation error (Combination 2).

The food nutritional content allocated by the FNR-MOEA/D is shown in Table 6.

From Table 6, the errors of the two nutritional elements, fat and vitamin A, are both very large, which are 72.03% and 67.99%, respectively. This is because among the eight foods, only pork and tofu contain relatively more fat, and the fat content of the remaining foods is very low. Moreover, the content of vitamin A in these eight kinds of foods is very low, thus resulting in the large errors of these two nutritional elements. The error of the iron element is 38.83%, because the consumption of rice and tofu with relatively more iron elements is relatively large. The error of the carbohydrate is 25.66%, because of the low carbohydrate content of all the food items except rice, thus resulting in a relatively large error of the carbohydrate. Except for these four nutrients, the errors of the remaining five nutrients are all around 15%, which is an acceptable range.

Table 6. The results of nutritional allocation (Combination 2).

Nutritional Elements	Standard	Allocate	Deviation	Error
Calories (KJ)	6200	6131.2	−68.8	1.11%
Protein (g)	72.3	73.1	0.8	1.11%
Fat (g)	59.71	16.7	−43.01	72.03%
Carbohydrate (g)	368.03	273.6	−94.43	25.66%
Vitamin A (μg)	683.25	218.7	−464.55	67.99%
Vitamin B (mg)	1.4	1.4	0	0
Calcium (mg)	755.32	753.2	−2.12	0.28%
Iron (mg)	20.24	28.1	7.86	38.83%
Vitamin C (mg)	97.21	80.9	−16.31	16.78%

From Table 7, the parts marked in bold are the minimum values, and the parts underlined are the maximum values. Judging from the nutritional errors of all the tested algorithms in the second food combination, the average error of the recommended result of the FNR-MOEA/D is the smallest, and there is no maximum error among all the nutritional elements.

Table 7. The comparison of algorithms (Combination 2).

Nutritional Elements	FNR-MOEA/D	MOEA/D	MOPSO	MOABC	RVEA	NSGA-II
Calories (KJ)	1.11%	3.34%	3.78%	36.77%	0.18%	94.74%
Protein (g)	1.11%	6.50%	10.65%	3.18%	7.88%	39.28%
Fat (g)	72.03%	69.85%	18.26%	66.50%	66.50%	85.41%
Carbohydrate (g)	25.66%	31.30%	61.10%	64.70%	29.67%	30.14%
Vitamin A (μg)	67.99%	70.29%	15.90%	70.29%	70.14%	32.05%
Vitamin B (mg)	0	0	46.72%	2.86%	0.71%	1.43%
Calcium (mg)	0.28%	0.31%	12.94%	5.87%	0.31%	13.99%
Iron (mg)	38.83%	38.34%	44.51%	3.75%	48.22%	2.37%
Vitamin C (mg)	16.78%	27.99%	47.22%	42.39%	29.02%	28.80%
Average error	24.86%	27.54%	29.01%	32.92%	28.07%	36.47%

5.3. Experimental Analysis

From the experimental results, when the FNR-MOEA/D is used to balance the food combinations with relatively balanced nutritional content, the overall error is the lowest, and the errors of the six nutritional elements are below 10%, which is an acceptable range. For food combinations with unbalanced nutritional content, the overall error after optimization is also the lowest. Excluding the four nutritional elements with the highest errors, the errors of the remaining nutritional elements are about 15%. Experiments show that in the issue of food nutrition balance regulation, compared with the MOPSO algorithm, MOABC algorithm, and RVEA, the FNR-MOEA/D performs better.

6. Conclusions

In this study, to better tackle the issue of personalized food recommendations and meet users' nutritional needs, we developed a hybrid food recommendation model based on MOEA/D. The main contributions of the study are as follows: (1) using the Slop One algorithm can predict the dietary score of users and prevent the cold start problem; (2) taking the nine nutritional elements in food as the optimization goals to regulate the intake of each food can finely adjust the matching proportion of each kind of food, thereby ensuring that the human body can obtain a comprehensive and balanced supply of nutrition; (3) aiming at the problem of nutritional balance adjustment, this paper makes a series of improvements to the MOEA/D through three strategies; (4) in combination with the food recommendation issue, by adding the AFR method to optimize and adjust the food nutrition, a more accurate dietary nutrition balance has been achieved.

The experimental results show that the overall error of FNR-MOEA/D is much smaller compared to other multi-objective algorithms, and the errors for most of the nutrients remain at around 10%, which is within a normal intake range. From this it can be seen that the hybrid food recommendation model constructed in this paper is effective in solving the problem of daily dietary nutritional balance.

In the future, we will address the problem of relatively large errors of certain nutrients in the case of unbalanced food nutrition, consider the use of targeted nutritional optimization methods, and improve the existing food recommendation model by combining advanced data analysis techniques, thereby introducing a more accurate nutritional assessment index system and combining the deep learning capability of artificial intelligence. The implementation of the model in real scenarios may have problems such as low user acceptance, slow system operation speed, system compatibility, and adding a user feedback mechanism to the model, which will be added to our future work.

Author Contributions: S.H.: conceptualization, software, methodology, validation, roles/writing—original draft, and writing review and editing. C.W.: conceptualization, formal analysis, investigation, methodology, resources, validation, and writing—review and editing. W.B.: supervision, validation, roles/writing—original draft, and writing review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China project (52205529).

Data Availability Statement: The original data utilized in this paper are available at <https://www.kaggle.com/datasets/zvikinozadze/myfitnesspal> (accessed on 24 October 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MOEA/D	Multi-Objective Evolutionary Algorithm Based on Decomposition
MOPSO	Multi-Objective Particle Swarm Optimization
MOABC	Multi-Objective Artificial Bee Colony algorithm
RVEA	Reference Vector Guided Evolutionary Algorithm
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
MOP	Multi-Objective Problem
MOO	Multi-Objective Optimization

References

1. Gao, C.; Zheng, Y.; Wang, W.; Feng, F.; He, X.; Li, Y. Causal inference in recommender systems: A survey and future directions. *ACM Trans. Inf. Syst.* **2024**, *42*, 1–32. [[CrossRef](#)]
2. Zmora, N.; Suez, J.; Elinav, E. You are what you eat: Diet, health and the gut microbiota. *Nat. Rev. Gastroenterol. Hepatol.* **2019**, *16*, 35–56. [[CrossRef](#)] [[PubMed](#)]
3. Wu, D.; Zheng, J.; Mao, G.; Hu, W.; Ye, X.; Linhardt, R.J.; Chen, S. Rethinking the impact of RG-I mainly from fruits and vegetables on dietary health. *Crit. Rev. Food Sci. Nutr.* **2020**, *60*, 2938–2960. [[CrossRef](#)]
4. Wu, G. Dietary protein intake and human health. *Food Funct.* **2016**, *7*, 1251–1265. [[CrossRef](#)]
5. Clemente-Suárez, V.J.; Beltrán-Velasco, A.I.; Redondo-Flórez, L.; Martín-Rodríguez, A.; Tornero-Aguilera, J.F. Global impacts of western diet and its effects on metabolism and health: A narrative review. *Nutrients* **2023**, *15*, 2749. [[CrossRef](#)] [[PubMed](#)]
6. Hu, F.B.; Liu, Y.; Willett, W.C. Preventing chronic diseases by promoting healthy diet and lifestyle: Public policy implications for China. *Obes. Rev.* **2011**, *12*, 552–559. [[CrossRef](#)]
7. Mendez-Zorrilla, A. Systematic Review of Nutritional Recommendation Systems. *Appl. Sci.* **2021**, *11*, 12069. [[CrossRef](#)]
8. Jayedi, A.; Soltani, S.; Abdolshahi, A.; Shab-Bidar, S. Healthy and unhealthy dietary patterns and the risk of chronic disease: An umbrella review of meta-analyses of prospective cohort studies. *Br. J. Nutr.* **2020**, *124*, 1133–1144. [[CrossRef](#)]
9. Cecchini, M.; Sassi, F.; Lauer, J.A.; Lee, Y.Y.; Guajardo-Barron, V.; Chisholm, D. Chronic Diseases: Chronic Diseases and Development 3 Tackling of unhealthy diets, physical inactivity, and obesity: Health effects and cost-effectiveness. *Lancet* **2010**, *376*, 1775–1784. [[CrossRef](#)] [[PubMed](#)]

10. Elswailer, D.; Trattner, C.; Harvey, M. Exploiting food choice biases for healthier recipe recommendation. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; Volume 51, pp. 575–584. [\[CrossRef\]](#)
11. Koolen, M.; Bogers, T.; Mobasher, B.; Said, A.; Tuzhilin, A. Overview of the Workshop on Recommendation in Complex Scenarios 2019 (ComplexRec 2019). In Proceedings of the Conference on Recommender Systems, Copenhagen, Denmark, 16–20 September 2019; ACM: New York, NY, USA, 2019. [\[CrossRef\]](#)
12. Zuo, Y.; Gong, M.; Zeng, J.; Ma, L.; Jiao, L. Personalized Recommendation Based on Evolutionary Multi-Objective Optimization [Research Frontier]. *IEEE Comput. Intell. Mag.* **2015**, *10*, 52–62. [\[CrossRef\]](#)
13. Zheng, Y.; Wang, D.X. A survey of recommender systems with multi-objective optimization. *Neurocomputing* **2022**, *474*, 141–153. [\[CrossRef\]](#)
14. Cui, L.; Ou, P.; Fu, X.; Wen, Z.; Lu, N. A novel multi-objective evolutionary algorithm for recommendation systems. *J. Parallel Distrib. Comput.* **2017**, *103*, 53–63. [\[CrossRef\]](#)
15. Elkin-Koren, N. Contesting algorithms: Restoring the public interest in content filtering by artificial intelligence. *Big Data Soc.* **2020**, *7*, 2053951720932296. [\[CrossRef\]](#)
16. Alharbe, N.; Rakrouki, M.A.; Aljohani, A. A collaborative filtering recommendation algorithm based on embedding representation. *Expert Syst. Appl.* **2023**, *215*, 119380. [\[CrossRef\]](#)
17. Yera, R.; Alzahrani, A.A.; Martinez, L. A food recommender system considering nutritional information and user preferences. *IEEE Access* **2019**, *7*, 96695–96711. [\[CrossRef\]](#)
18. Sahoo, A.K.; Pradhan, C.; Barik, R.K.; Dubey, H. DeepReco: Deep Learning Based Health Recommender System Using Collaborative Filtering. *Computation* **2019**, *7*, 25. [\[CrossRef\]](#)
19. Tran, T.N.T.; Atas, M.; Felfernig, A.; Stettinger, M. An overview of recommender systems in the healthy food domain. *J. Intell. Inf. Syst.* **2018**, *50*, 501–526. [\[CrossRef\]](#)
20. Trattner, C.; Elswailer, D. Food Recommender Systems: Important Contributions, Challenges and Future Research Directions. *arXiv* **2017**, arXiv:1711.02760. [\[CrossRef\]](#)
21. Mckensy-Sambola, D.; Rodriguez-García, M.Á.; García-Sánchez, F.; Valencia-García, R. Ontology-Based Nutritional Recommender System. *Appl. Sci.* **2021**, *12*, 143. [\[CrossRef\]](#)
22. Rostami, M.; Oussalah, M.; Farrahi, V. A novel time-aware food recommender-system based on deep learning and graph clustering. *IEEE Access* **2022**, *10*, 52508–52524. [\[CrossRef\]](#)
23. Chen, Y.; Guo, Y.; Fan, Q.; Zhang, Q.; Dong, Y. Health-Aware Food Recommendation Based on Knowledge Graph and Multi-Task Learning. *Foods* **2023**, *12*, 2079. [\[CrossRef\]](#)
24. Iwendi, C.; Khan, S.; Anajemba, J.H.; Bashir, A.K.; Noor, F. Realizing an efficient IoMT-assisted Patient Diet Recommendation System through Machine Learning Model. *IEEE Access* **2020**, *8*, 28462–28474. [\[CrossRef\]](#)
25. Mathur, A.; Juguru, S.K.; Eirinaki, M. A Graph-Based Recommender System for Food Products. In Proceedings of the 2019 First International Conference on Graph Computing, Laguna Hills, CA, USA, 25–27 September 2019. [\[CrossRef\]](#)
26. Madani, A.; Engelbrecht, A.; Ombuki-Berman, B. Cooperative coevolutionary multi-guide particle swarm optimization algorithm for large-scale multi-objective optimization problems. *Swarm Evol. Comput.* **2023**, *78*, 101262. [\[CrossRef\]](#)
27. Daulton, S.; Eriksson, D.; Balandat, M.; Bakshy, E. Multi-Objective Bayesian Optimization over High-Dimensional Search Spaces. *arXiv* **2021**, arXiv:2109.10964. [\[CrossRef\]](#)
28. Qi, S.; Wang, R.; Zhang, T.; Dong, N. Cooperative coevolutionary competition swarm optimizer with perturbation for high-dimensional multi-objective optimization. *Inf. Sci.* **2023**, *644*, 119253. [\[CrossRef\]](#)
29. Song, Z.; Wang, H.; Xue, B.; Zhang, M. Balancing Different Optimization Difficulty Between Objectives in Multi-Objective Feature Selection. *IEEE Trans. Evol. Comput.* **2023**, *28*, 1824–1837. [\[CrossRef\]](#)
30. Xia, H.; Zhuang, J.; Yu, D. Novel soft subspace clustering with multi-objective evolutionary approach for high-dimensional data. *Pattern Recognit.* **2013**, *46*, 2562–2575. [\[CrossRef\]](#)
31. Gu, Q.; Xu, Q.; Li, X. An improved NSGA-III algorithm based on distance dominance relation for many-objective optimization. *Expert Syst. Appl.* **2022**, *207*, 117738. [\[CrossRef\]](#)
32. Rivera, G.; Cruz-Reyes, L.; Fernandez, E.; Gomez-Santillan, C.; Rangel-Valdez, N. An interactive ACO enriched with an eclectic multi-criteria ordinal classifier to address many-objective optimisation problems. *Expert Syst. Appl.* **2023**, *232*, 120813. [\[CrossRef\]](#)
33. Moazen, H.; Molaei, S.; Farzinvasht, L.; Sabaei, M. PSO-ELPM: PSO with elite learning, enhanced parameter updating, and exponential mutation operator. *Inf. Sci.* **2023**, *628*, 70–91. [\[CrossRef\]](#)
34. Qian, H.; Yu, Y. Solving high-dimensional multi-objective optimization problems with low effective dimensions. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31. [\[CrossRef\]](#)
35. Zhang, Q.; Li, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.* **2008**, *11*, 712–731. [\[CrossRef\]](#)
36. Hu, Y.; Lyu, C.; Yuan, J. Parameter optimization of Slave-Master PID controller parameters based on improved MOEA/D algorithm. *Appl. Res. Comput.* **2024**, *41*, 1434–1440. [\[CrossRef\]](#)
37. Wang, W.; Li, K.; Tao, X.; Gu, F.-H. An improved MOEA/D algorithm with an adaptive evolutionary strategy. *Inf. Sci.* **2020**, *539*, 1–15. [\[CrossRef\]](#)

38. Zhang, Y.; Wang, G.G.; Li, K.; Yeh, W.-C.; Jian, M.; Dong, J. Enhancing MOEA/D with information feedback models for large-scale many-objective optimization. *Inf. Sci.* **2020**, *522*, 1–16. [[CrossRef](#)]
39. Cao, J.; Zhang, J.; Zhao, F.; Chen, Z. A two-stage evolutionary strategy based MOEA/D to multi-objective problems. *Expert Syst. Appl.* **2021**, *185*, 115654. [[CrossRef](#)]
40. Wang, Q.X.; Luo, X.; Li, Y.; Shi, X.-Y.; Gu, L.; Shang, M.-S. Incremental Slope-one recommenders. *Neurocomputing* **2018**, *272*, 606–618. [[CrossRef](#)]
41. Zhou, H.; Deng, Z.; Xia, Y.; Fu, M. A new sampling method in particle filter based on Pearson correlation coefficient. *Neurocomputing* **2016**, *216*, 208–215. [[CrossRef](#)]
42. Cheng, Y. Dietary Reference Intake of Nutrients for Chinese Residents, Introduction to the 2013 revision. *J. Nutri.* **2014**, *36*, 313–317. [[CrossRef](#)]
43. Cheng, R.; Jin, Y.; Olhofer, M. Test problems for large-scale multiobjective and many-objective optimization. *IEEE Trans. Cybern.* **2016**, *47*, 4108–4121. [[CrossRef](#)] [[PubMed](#)]
44. Xu, M.; Zhang, M.; Cai, X.; Zhang, G. Adaptive neighbourhood size adjustment in MOEA/D-DRA. *Int. J. Bio-Inspired Comput.* **2021**, *17*, 14. [[CrossRef](#)]
45. Li, Y.; Liu, H.; Xie, K.; Yu, X. A method for distributing reference points uniformly along the Pareto front of DTLZ test functions in many-objective evolutionary optimization. In Proceedings of the 2015 5th International Conference on Information Science and Technology (ICIST), Changsha, China, 24–26 April 2015; IEEE: New York, NY, USA, 2015; pp. 541–546. [[CrossRef](#)]
46. Cheng, R.; Li, M.; Tian, Y.; Zhang, X.; Yang, S.; Jin, Y.; Yao, X. A benchmark test suite for evolutionary many-objective optimization. *Complex Intell. Syst.* **2017**, *3*, 67–81. [[CrossRef](#)]
47. He, J.; Luo, F.; Yuan, Z.; Huang, H. The application of collaborative filtering and particle swarm algorithm in dietary recommendation. *Comput. Appl. Softw.* **2019**, *36*, 36–40. [[CrossRef](#)]
48. Li, H.; Zhang, Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans. Evol. Comput.* **2009**, *13*, 284–302. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.