eng

Article

Improved Quantum Particle Swarm Optimization of Optimal Diet for Diabetic Patients

Abdellah Ahourag [1], Zakaria Bouhanch [1], Karim El Moutaouakil [1,*] and Abdellah Touhafi [2]

[1] Engineering Science Laboratory, FPT, Sidi Mohamed Ben Abdellah University, Fez 30000, Morocco; abdellah.ahourag@usmba.ac.ma (A.A.); zakaria.bouhanch@usmba.ac.ma (Z.B.)
[2] Department of Engineering Technology (INDI), Vrije Universiteit Brussel (VUB), 1050 Brussels, Belgium; abdellah.touhafi@vub.be
* Correspondence: karim.elmoutaouakil@usmba.ac.ma

Abstract: The dietary recommendations for individuals with diabetes focus on maintaining a balanced nutritional intake to manage blood sugar levels. This study suggests a nutritional strategy to improve glycemic control based on an analysis of a dietary optimization problem. The goal is to minimize the overall glycemic loads (GLs) of specific foods. Two variations of the particle swarm optimization (PSO) method, as well as random quantum process optimization (GQPSO), are introduced. The findings demonstrate that the quantum and random methods are more effective than the traditional techniques in reducing the glycemic loads of diets and addressing nutritional deficiencies while also aligning nutrient intake with the recommended levels. The resolution of this diet optimization model, executed multiple times with adjustments to the parameters of both methods, enables dynamic exploration and provides a wide range of diverse and effective food choices.

Keywords: nutritional optimization; glycemic load; quantum swarm particles; nutritional specifications; optimum diet

Citation: Ahourag, A.; Bouhanch, Z.; El Moutaouakil, K.; Touhafi, A. Improved Quantum Particle Swarm Optimization of Optimal Diet for Diabetic Patients. *Eng* **2024**, *5*, 2544–2559. https://doi.org/10.3390/eng5040133

Academic Editor: Juvenal Rodriguez-Resendiz

Received: 1 September 2024
Revised: 4 October 2024
Accepted: 8 October 2024
Published: 10 October 2024

1. Introduction

Diabetes is a chronic disease that affects millions of people worldwide. It is well understood that proper dietary management is crucial for diabetic patients to maintain their health and control their blood glucose levels [1]. To further improve in this area, we intend to address the complex dietary challenges faced by people with diabetes. This improvement involves testing the effectiveness of two optimization methods in a nutritional optimization model for diabetics: particle swarm optimization (PSO) and its enhanced variant, Gaussian quantum particle swarm optimization (GQPSO). Particle swarm optimization (PSO) is a heuristic optimization technique inspired by the collective movement of a swarm of birds. Developed by Kennedy and Eberhart in the 1990s [2], this method is particularly well able to solve complex non-linear optimization problems. Particle swarm optimization (PSO) is a non-linear optimization method based on social behavior, using swarm intelligence algorithms for global optimization on continuous search spaces, where particles collaborate and share knowledge within a swarm. Particle swarm optimization (PSO) is based on a set of randomly and homogeneously arranged particles that move in the research hyper-space, each considered a potential solution. Particles remember their best solution and can communicate with their environment. By iteratively updating their positions, particles gradually converge toward an optimal solution. A novel variant, quantum-behaved particle swarm optimization (QPSO), has recently emerged [3,4]. This method is based on the principles of quantum mechanics and particle swarm optimization techniques. The key difference between these two approaches is that the quantum particle swarm optimization (QPSO) method is inspired by the principles of quantum mechanics and particle swarm optimization techniques. The QPSO method is based on the concepts of quantum superposition and interference, allowing particles to jump between different potential solutions in a non-deterministic way [4]. On the other hand, by enhancing these methods with probability distributions [5,6], the behavior of quantum particles in the search for solutions can be stochastically and randomly modeled. This enables a more comprehensive and diversified exploration of the solution space.

Particle swarm optimization (PSO) and its variants, such as QPSO and GQPSO, have shown their versatility and effectiveness in solving a variety of real-world problems. These optimization techniques have been successfully applied in various fields, including engineering, healthcare, finance, and more. For example, the TrigAC-PSO variant of PSO has been utilized to optimize the solution for the fuzzy transportation problem [7]. The efficiency of this method is shown by incorporating trigonometric functions to adjust the particle velocities, potentially leading to faster convergence and improved exploration of the search space. Its adaptability to fuzzy data makes it suitable for solving the fuzzy transportation problem, and its global optimization capabilities are beneficial for finding near-optimal or optimal solutions. Similarly, in [8], the transportation problem was solved using the PSO method and its variants. Moreover, in [9], PSO was used to optimize the solutions for the urban transit routing problem, and the proposed PSO variant outperforms the existing techniques, providing superior solutions for the UTRP. Another application explored for PSO is in the field of school timetabling [10]; the results of this study suggest that PSO can be a valuable tool for optimizing school timetables and providing efficient and equitable scheduling. Furthermore, an application of PSO lies in the domain of cloud computing task scheduling [11]. The EECS strategy, utilizing APSO, offers an effective solution for managing IoT and non-IoT tasks. The strategy's focus on energy efficiency and resource optimization is a key advantage. In [12], a proposed hybrid PSO-OSELM model demonstrated to be an effective indoor temperature forecasting approach. Combining PSO for optimization and online sequential extreme learning machine (OSELM) for adaptive learning improve forecast accuracy and generalization performance over other methods. The successful application of PSO and its variants in diverse fields, ranging from engineering and finance to healthcare, has demonstrated their effectiveness in solving complex optimization problems. Given the complexity of finding an optimal diet for diabetic patients, which involves balancing multiple factors, such as nutritional needs, caloric intake, and glycemic control, PSO's ability to handle complex constraints and search for near-optimal solutions, coupled with the enhanced capabilities of its variants like QPSO and GQPSO, make it a promising candidate for addressing this challenge.

The article aims to assess the performance of two variants of particle swarm optimization—one with quantum and random behavior (GQPSO) and the other a traditional PSO—in solving an optimization problem aimed at minimizing the total glycemic loads of diets. The problem of minimizing the glycemic load represents an advanced nutritional optimization model, crucial for managing diabetes or for those looking to cut down on their carbohydrate intake [13]. Most of these models are constrained by various factors, like nutrient variability and interactions between macronutrients, making it tricky to solve these issues [14,15]. This study focuses on tackling these challenges within a quantum and random framework using a quantum-behaved swarm particle approach that incorporates Gaussian mutation (GQPSO). Quantum behavior encourages a more efficient exploration of the search space. Integrating a Gaussian mutation into the method (GQPSO) increases the likelihood of refining the solutions through iterations, enabling optimal exploitation and exploration. This helps in selecting safe and diverse diets, which are essential for both diabetics and others. Solving this model with different parameter values from both methods enables dynamic exploration, providing a variety of solutions. This makes it easier to compare the performance of the two methods across several aspects, including minimizing the glycemic load and nutritional intake.

In the upcoming section, we will provide detailed descriptions of the processes involved in two approaches: traditional particle swarm optimization (PSO) and quantum random behavior optimization. Following that, in the third section, we will outline the modeling of our nutritional optimization problem, including the consideration of data sets and parameters. Finally, in the last two sections, we will compare the performance of the PSO and GQPSO methods by analyzing the nutritional results. This will be followed by a conclusion and a discussion of the future perspectives in this field.

2. Classical and Quantum Random Particle Swarm Approaches

2.1. Standard PSO

The particle swarm optimization method was originally developed by researchers Kennedy and Eberhart [2]. Particle swarm optimization (PSO) is a method that involves a group of individuals called particles, which are positioned randomly in a search space. Each particle represents a potential solution and moves throughout the space seeking the best solution. Additionally, each particle remembers the best solution it has encountered and can communicate with other particles in its vicinity. In the traditional PSO with M particles, each particle is considered as a potential solution within a D-dimensional space. The position of an individual particle at the k-th iteration is represented as $x_i(k) = [x_{i1}(k), x_{i2}(k), \ldots, x_{iD}(k)]$. Each particle retains information about its previous best position. The velocity of the moving particles is represented by a D-dimensional vector denoted $v_i(k) = [v_{i1}(k), v_{i2}(k), \ldots, v_{iD}(k)]$. At the $(k+1)$-th iteration, the velocity and position of particle i are updated by the following equations:

$$v_{ij}(k+1) = wv_{ij}(k) + c_1 r_{ij}(k)\left(P_{ij}(k) - x_{ij}(k)\right) + c_2 R_{ij}(k)\left(g_j^k - x_{ij}(k)\right) \quad (1)$$

$$x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k+1) \quad (2)$$

where c_1 and c_2 are two positive constants, called the cognitive and social coefficients, respectively. These constants control the relative influence of cognition and social interaction in the updating process. r_{ij} and R_{ij} are two random numbers uniformly distributed in the range $(0,1)$ for the j-th dimension of particle i. $P_i(k) = [P_{i1}(k), P_{i2}(k), \ldots, P_{iD}(k)]$ is the vector of the position with the best fitness found so far for the i-th particle, which is called (pbest). And, vector $g^k = [g_1^k, g_2^k, \ldots, g_D^k]$ records the best position discovered by the swarm so far, known as the global best (gbest) position. $x_{ij}(k)$, $v_{ij}(k)$ and $P_{ij}(k)$ are the j-th dimensions of the vectors of $x_i(k)$,$v_i(k)$ and $P_i(k)$, respectively. Parameter w represents the inertia weight utilized for striking a balance between global and local search abilities [16]. The most common strategy for controlling it is to first set it at 0.9 and then reduce it linearly to 0.4 [17]. Let f be the objective function we want to minimize. PSO can be described using the following Algorithm 1.

Algorithm 1 PSO Pseudo-code

```
Input: M : Swarm size
          D : Problem dimension.
          T: Maximum iterative number
Output: g^k : The best solution (position)
Start
Initialization : Generate an initial population with positions and velocities.
for i = 1 to M do
    if f(x_i) < f(P_i) then
        P_i = x_i
        g_i^k = argmin(f(P_i))
    end if
end for
while k ≤ T do
    for i = 1 to D do
        r_ik,R_ik two independent vectors randomly generated from [0,1]^D
        Apply Equation (1)
        Apply Equation (2)
```

Algorithm 1 *Cont.*

        if $f(x_i(k)) < f(P_i(k-1))$ **then**
            $f(P_i)(k) \leftarrow f(x_i(k))$
        end if
    end for
    $g_i^k = argmin(f(P_i))$
    $k \leftarrow k+1$
end while

2.2. *An Overview of Quantum-Behaved Particle Swarm Optimization*

In classical mechanics, any particle is characterized by its position vector x and velocity vector v, which determine its trajectory. In Newtonian mechanics, a particle follows a defined trajectory, but, in quantum mechanics, the concept of trajectory is meaningless. In this case, the notion of trajectory loses its meaning due to the uncertainty principle, making it impossible to simultaneously determine a particle's position x and velocity v. Consequently, if the particles within a PSO system exhibit quantum behavior, this implies that the PSO algorithm will operate differently.

A new variant of PSO, called quantum particle swarm optimization (QPSO) [4], is a method inspired by quantum mechanics and particle swarm optimization. For the QPSO method, the velocity vector does not appear; it only has the position vector and is, therefore, simpler than the standard particle swarm optimisation algorithm. Sun and et al. [3] introduced quantum theory into PSO and proposed the quantum particle swarm optimization algorithm (QPSO). This algorithm not only has fewer control parameters than PSO but is also more efficient, theoretically guaranteeing finding the optimal solution in the search space.

In the QPSO model, the state of a particle is provided by the wave function $\Psi(x,t)$ (Schrödinger equation) [18] instead of position and velocity. The probability density function $|\Psi(x,t)|^2$ provides a means to calculate the likelihood of a particle being found at a particular location, as determined by the statistical properties of its wave function [19]. The position of the particle can therefore be calculated using the probability density function. Using the Monte Carlo method, we can obtain the j-th component of position x_i of the particle at iteration $k+1$ as [3,19]

$$x_{ij}(k+1) = p_{ij}(k) \pm \beta |M_{best_j}(k) - x_{ij}(k)|.ln(1/u_{ij}(k)) \tag{3}$$

where $u_{ij}(k)$ is a random number generated using the uniform probability distribution functions in the range $[0,1]$, p_i is the local attractor and defined as [20] $p_i(k) = [p_{i1}(k), p_{i2}(k), \ldots, p_{iD}(k)]$, and

$$p_{ij}(k) = \frac{c_1 r_{ij}(k) P_{ij}(k) + c_1 R_{ij}(k) g_j^k}{c_1 r_{ij}(k) + c_1 R_{ij}(k)}$$

Or,

$$p_{ij}(k) = \phi p_{ij}(k) + (1-\phi) g_j^k$$

where

$$\phi = \frac{c_1 r_{ij}(k)}{c_1 r_{ij}(k) + c_1 R_{ij}(k)}.$$

β is called the contraction–expansion (CE) coefficient, which can be tuned to control the convergence speed of the algorithm. This coefficient should be controlled when using QPSO in practical applications. In [21], it has been shown that taking an interval $(0.5, 0.8)$ provides more interesting results for the majority of benchmark functions. The value of β can be calculated by

$$\beta = \beta_0 + (T-t).(\beta_1 - \beta_0)/T \tag{4}$$

where β_0 and β_1 are the final and initial values of β, respectively, T is the maximum number of iterations, and t is the current iteration number. M_{best} is known as the mean best (mbest) position defined as the mean of the *pbest* positions of all particles and is provided by

$$M_{best}(k) = [M_{best_1}(k), M_{best_2}(k), \ldots, M_{best_D}(k)] \tag{5}$$

$$= \left[\frac{1}{M} \sum_{i=1}^{M} P_{i1}(k), \frac{1}{M} \sum_{i=1}^{M} P_{i2}(k), \ldots, \frac{1}{M} \sum_{i=1}^{M} P_{iD}(k) \right]$$

In general, the PSO algorithm with Equation (3) is called quantum-behaved particle swarm optimization (QPSO), and it can be described by the following Algorithm 2.

Algorithm 2 QPSO Pseudo Code [3]

Start
Initialization : Generate an initial population (size $= M$), with positions and the dimensions of particles.
for $k = 1$ to T Maximum iteration **do**
    Compute the mean best position M_{best} by (5)
    $\beta = \beta_0 + (T - k).(\beta_1 - \beta_0)/T$
    for $i = 1$ to M **do**
        if $f(x_i) < f(P_i)$ **then**
            $P_i = x_i$
        end if
        $g_i^k = argmin(f(P_i))$
        for $j = 1$ to D **do**
            $\phi = rand(0,1)$, $u_{ij} = rand(0,1)$
            $p_{ij} = \phi.p_{ij} + (1 - \phi)g_j$
            if $rand(0,1) > 0.5$ **then**
                $x_{ij} = p_{ij} + \beta|M_{best_j} - x_{ij}|.ln(1/u_{ij})$
            else
                $x_{ij} = p_{ij}(k) - \beta|M_{best_j} - x_{ij}|.ln(1/u_{ij})$
            end if
        end for
    end for
end for

The illustration in Figure 1 shows how particles move in the PSO and QPSO methods towards the best global positions (central particles) in the search space. Thanks to Adaptive Particle Convergence in QPSO, particles that are far from the global position are able to move towards it. On the other hand, in PSO, a particle that does not find a better position than the global one no longer influences the others. As a result, QPSO enhances the contribution of these particles compared to traditional PSO. The arrows in particle swarm optimization (PSO) and quantum particle swarm optimization (QPSO) algorithms illustrate the behavior of particles. In the classic PSO algorithm, the red arrows show the direction of the particles towards the global position, which is determined by their best-known position (pbest) and the best position of the swarm (gbest) [5]. In contrast, in QPSO, the quantum behavior enables a position update that takes multiple factors into account, including the average best position (mbest), enabling the particles to move simultaneously in several directions. The blue arrows represent these directions, while the large red arrows show the most likely direction of movement in the QPSO algorithm.

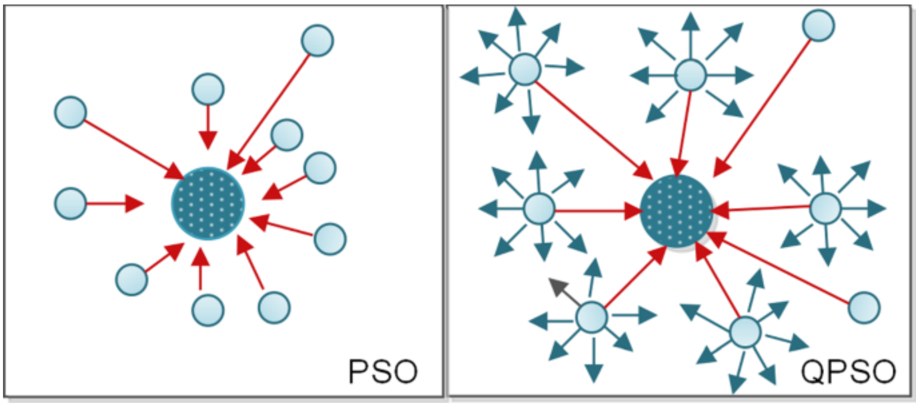

Figure 1. Illustration of PSO and QPSO methods.

2.3. Quantum Particle Swarm Optimization Using Gaussian Mutation (GQPSO)

In addition to the mechanisms for updating particle positions within QPSO methods that use a probability distribution (3), Maolong Xi et al. introduce a weighting parameter in calculating the optimal average position in WQPSO [22], which is determined by the fitness value of the particles. This approach involves a probabilistic evaluation aimed at measuring the influence of each particle on the average position. Several mutation mechanisms using probability distributions are integrated into quantum behavior particle swarm optimization to enhance the exploration of the search space and maintain diversity in solutions. In QPSO methods with mutations, the mutation is typically incorporated when updating the positions of the particles. For a mutation with a general distribution, the formula takes the following form:

$$x_{ij}^{\text{mut}}(k+1) = x_{ij}(k+1) + \mu \cdot Distribution \tag{6}$$

where $x_{ij}^{\text{mut}}(k+1)$ is the position of the particle after mutation, μ is a controlling factor determining the intensity of the mutation, and *Distribution* represents the applied probability distribution.

The use of Cauchy-distributed mutations in QPSO algorithms often enables particles to make "long jumps", thereby encouraging a more global and diverse exploration of solutions. In [5], a natural selection mechanism is also applied to enhance the efficiency of the QPSO-CD algorithm. There is also the integration of Gaussian mutation operators to enhance Gaussian quantum particle swarm algorithms (GQPSOs) in various engineering applications [6,23]. In GQPSO, the Gaussian distribution is used to update the positions of the particles.

Approach 1—GQPSO(1) [6]:

$$x_{ij}(k+1) = \begin{cases} P_i + b \cdot |Mbest(k) - x_{ij}(k)| \cdot \ln(1/Gu_{ij}(k)) & \text{if } k \geq 0.5 \\ P_i - b \cdot |Mbest(k) - x_{ij}(k)| \cdot \ln(1/Gu_{ij}(k)) & \text{if } k < 0.5 \end{cases} \tag{7}$$

The expression $Gu = |N(0,1)|$ represents the absolute value of a standard normal random variable. The term P_i denotes a point in the search space used to calculate the new position of particle $x_{ij}(k+1)$ based on the best position found by the population (*Mbest*) and the value of Gu . Parameter k affects the way the positions of the particles are updated.

Approach 2—GQPSO(2) [6]:

$$P_i = \frac{Gu \cdot p_{i,d} + g \cdot p_{g,d}}{Gu + g} \tag{8}$$

where $g = |N(0,1)|$. Notations $p_{i,d}$ and $p_{g,d}$ represent the best personal and global positions of the particles, respectively. This enables probabilistic exploration of the space, taking quantum effects into account, making the method more efficient and robust. In traditional optimization approaches, the mutation operators are applied in a fixed manner, and the types of distributions remain constant throughout the optimization process. However, this limits the exploration of the search space. To address these limitations, Ref. [24] introduces an adaptive mutation in the quantum behavior particle swarm optimization algorithm (AMQPSO). This algorithm, which is based on the q-Gaussian distribution, dynamically adjusts the mutation parameters based on the state of the population. The non-extensive entropy index q is calculated using the formula

$$q = q_{\max} - t \cdot \frac{q_{\max} - q_{\min}}{\text{MaxIter}}, \tag{9}$$

where $q_{\max}$ and $q_{\min}$ represent the maximum and minimum values of q, respectively, t is the current iteration (ranging from 0 to MaxIter), and MaxIter is the maximum number of iterations.

The study aims to use two techniques, including a traditional PSO and another that combines quantum and random principles, GQPSO, to enhance the nutritional optimization problem for individuals with diabetes, specifically in terms of nutrition.

3. Nutritional Optimization Problem Modeling for Diabetics

3.1. Information and Variables Related to Diet Problems

The most crucial element in the prevention and treatment of diabetes is diet, which requires the creation of an eating plan adapted to various social and economic factors and its integration into daily life. Foods contain essential nutrients for the body's proper functioning, Nutrients can be divided into two categories: positive nutrients, such as calories, protein, carbohydrates, and various vitamins and minerals, and negative nutrients, which include saturated fats, sodium, cholesterol, and total fat. In terms of daily nutritional requirements, we have chosen to follow the nutritional recommendations issued by the United States Department of Agriculture (USDA). These recommendations are designed to help individuals maintain a balanced diet to prevent diabetes and promote overall health. They are designed in Figures 2 and 3.

The variables and data for our diet problems are based on a selection of 177 foods from various groups (fruits, vegetables, meats, starches, etc.) that are most widely consumed by the Moroccan community. Nutritional values are calculated for a 100g portion of each food. Each meal, which is a combination of n foods, is represented by a vector $x = (x_1, \ldots, x_{177})$, where each element $(x_n)_{n=1,..,177}$ corresponds to the quantity of nutrient n in the meal per 100 g unit. The aim is to determine a food choice that provides a variety of nutritious foods while reducing the overall glycemic load of a meal (optimal diet). This food choice must also meet specific recommended requirements [14,25].

Here are the various variables and parameters used to model our problem:

$G^T x$: The glycemic load induced by foods included in the diet.

$A = [a_{ij}]$: The matrix in which a_{ij} represents the quantity of beneficial nutrients i per 100 g portion of food j.

$E = [e_{ij}]$: This is the matrix where e_{ij} indicates the amount of potentially harmful nutrients i per 100g portion of food j.

b: The minimum of positive nutrients required for the body to function properly.

f: Maximum acceptable levels of harmful nutrients can cause health problems if consumed in excess.

A_c: The vector represents the portion of calories from the positive nutrient.

c_i : Vector representing the calories from the nutrients in set $\{car, p, tf, sf\}$.

τ_i: The percentage of total calories of i nutrients from $\tau_p = 18\%$, $\tau_{car} = 55\%$, $\tau_{sf} = 7.8\%$, and $\tau_{tf} = 29\%$.

3.2. Mathematical Representation of Constraints

3.2.1. Requirements for Beneficial Nutrients

The total beneficial nutrients of diet x are expressed by vector $A.x$, and the requirements are represented by b, so it is necessary to impose constraint $A.x \geq b$.

The calorie ratio produced by carbohydrates must be determined by inequality $c_{car}^T x \geq 0.55\left(A_c^T x\right)$.

Calories from protein must be consistent with inequality $c_p^T x \geq 0.18\left(A_c^T x\right)$.

3.2.2. Requirements Concerning Harmful Nutrients

Total harmful nutrients in the diet x is Ex, and the needs are recorded in f; we thus have $Ex \leq f$.

Total calories from saturated fats are regulated by the following condition: $c_{tf}^T x \leq 0.29\left(A_c^T x\right)$.

Total calories from total fat are regulated by the following inequality: $c_{sf}^T x \leq 0.078\left(A_c^T x\right)$.

The research focuses on improving nutrition for people with diabetes, modeled as a linear optimization problem aimed at minimizing the total glycemic loads of diets found. Glycemic load is considered a more recent tool for better dietary management in the context of diabetes as it enables a diversified classification of foods according to their impact on blood glucose, thus facilitating diabetes control. Thus, our nutritional optimization problem is as follows:

$$(P): \begin{cases} Min\ G^T x \\ \quad Subject\ to: \\ \quad\quad A.x \geq b \\ \quad\quad E.x \leq f \\ \quad\quad c_i^T x \geq \tau_i\left(A_c^T x\right) \qquad i \in \{car, p\} \\ \quad\quad c_i^T x \leq \tau_i\left(A_c^T x\right) \qquad i \in \{tf, sf\} \\ \quad\quad x \geq 0 \end{cases} \tag{10}$$

Similar research addressing this issue has compared the performance of the PSO method with other intelligent optimization approaches, such as the stochastic fractal (SFS), the firefly optimization algorithm (FA), and the genetic algorithm (GA), which are covered in [15,26,27]. This problem was also modeled using fuzzy optimization techniques to deal with the imprecision of the associated parameters, which led to more efficient results and could be useful in the field of blood glucose management [28–32]. This issue is addressed through a multi-criteria optimization process that takes into account various factors, such as the cost of nutrition, and utilizes different methods to improve performance [33–37], making it possible to confront diabetes control with other criteria. With a view to vague nutritional recommendations, we have suggested a method based on the optimization and learning of artificial neural networks to better estimate people's nutritional dietary needs [38,39].

3.3. Evaluation of Diet Problem Parameter

We analyzed 177 commonly used foods in Moroccan cuisine to assess their macronutrient composition, including vitamins, minerals, lipids, and carbohydrates, as well as elements such as sugars and fiber. The nutritional values for certain foods are displayed in nutritional tables per 100g portion (see Tables 1 and 2).

The daily nutrient requirements for positive and potentially harmful nutrients are those recommended by the US Department of Agriculture [14,40], and these recommendations do not take into account the average amounts of positive and potentially harmful nutrients; they are Illustrated in Figures 2 and 3.

Table 1. Contents of vitamins and minerals.

Name of Foods	Vitamin A	Vitamin C	Vitamin E	Vitamin B6	Vitamin B12	Calcium (Ca)	Phosphorus	Magnesium	Potassium	Iron (Fe)	Zinc	Calories	Protein	Carbohydrate
Apricot	0	5.5	0.6	0.1	0	15.6	16.6	8.7	237	0.3	0.1	49	0.9	9
Dried Apricot	0	1	4	0.2	0	61.2	68.3	36.5	1090	4.3	0.3	271	3.1	53
Garlic	0	17	0	1.2	0	17.7	161	20.7	555	1.3	0.8	131	7.9	21.5
Almond	0	0.4	14.6	0.1	0	248	416	232	668	3	3.3	634	25.4	1.5
Pineapple	0	12	0.1	0.1	0	20.3	11	19.8	170	0.2	0.7	53	0.4	11
Canned Pineapple	0	10.4	0.1	0.1	0	14.3	5	13.3	105	0.2	0.1	82	0.4	19.1
Artichoke	0	10.3	0.2	0.1	0	39	49.2	29.5	380	0.7	0.5	44	2.8	4.9
Asparagus	0	16	0	0	0	19.9	51.5	6.3	198	0.7	0.4	30	2.7	3.2
Eggplant	0	1.3	0	0.1	0	20.1	15	15	123	0.3	0.1	35	0.8	6.3
Avocado	0	7.5	2.4	0.2	0	10.8	41.9	27.1	412	0.5	0.5	69	1.8	3.13
Baguette	0	0	0.1	0.1	0.0001	52.4	110	19.7	158	1.5	0.7	286	9.3	56.6
Banana	0	6.5	0.3	0.3	0	4.5	17.5	32.8	411	0.3	0.2	94	1.2	20.5
Beetroot	0	5	0	0	0	18.4	31.1	16.3	266	0.7	0.3	43	2.3	7.2
Cooked Egg White	0	0	0	0	0.00001	6.7	14.7	9.7	147	0.1	0	46	10.3	0.7
Cooked Broccoli	0.4	0.3	0.8	0.3	0.002	0	0	0	0	0	0.7	97	21.5	1.1
Broccoli	0	37.3	1	0.2	0	55.8	56	11.5	148	1	0.3	29	2.1	2.8
Peanut	0	0.7	12.2	0.5	0	4.9	370	70.6	54.2	0	2.8	636	25.9	14.8
Raw Carrot	0	16	0	0	0	19.9	51.5	6.3	198	0.7	0.4	30	2.7	3.2
Peeled, Cooked Carrot (boiled)	0	4	0.6	0.1	0	26.2	20.4	11.9	243	0.3	0.2	36	0.8	6.6
Celery	0	8	0.2	0.1	0	53.3	27.2	9.2	269	0.3	0.1	16	1.2	1.2
Cooked Celery Stalk	0	4	0.2	0.1	0	53.3	25	9	284	0.4	0.1	13	0.8	1.6

Table 2. Sodium, fat, cholesterol, and fatty acid content of foods.

Name of Foods	Sodium	Total Fat	Cholesterol	Saturated Fat
Apricot	1.00	0.39	0.100	0.027
Dried Apricot	10.00	0.51	0.195	0.017
Garlic	17.00	0.50	0.000	0.089
Almond	1.61	53.40	1.180	4.040
Pineapple	1.00	0.12	0.000	0.009
Canned Pineapple	0.10	0.10	0.000	0.100
Artichoke	94.00	0.15	0.000	0.036
Asparagus	14.00	0.22	0.000	0.048
Eggplant	1.00	0.23	0.000	0.044
Avocado	7.00	14.66	0.000	2.126
Baguette	711.00	1.30	0.100	0.280
Banana	1.00	0.33	0.100	0.112
Beetroot	0.10	0.20	0.200	0.100
Cooked Egg White	0.20	0.20	0.000	0.000
Cooked Broccoli	41.00	0.41	55.500	0.079
Broccoli	0.30	0.40	0.500	0.100
Peanut	2.10	49.60	0.000	1.000
Raw Carrot	69.00	0.24	0.000	0.037
Peeled, Cooked Carrot (boiled)	0.10	0.10	0.100	0.100
Celery	80.00	0.17	0.430	0.042
Cooked Celery Stalk	91.00	0.16	0.000	0.040

Potentially Harmful Nutrients	Maximum Acceptable Level
Sodium (s)	1779 mg
Total Fat (tf)	65 g
Cholesterol (ch)	230 mg
Saturated Fat (sf)	17 g

Figure 2. The negative nutrient requirements.

Beneficial Nutrient	Minimum Necessary Level
Vitamin A (va)	1052 μg
Vitamin C (vc)	155 mg
Vitamin E (ve)	9.5 AT
Vitamin B6 (Vb6)	2.4 mg
Vitamin B12 (vb12)	8.3 μg
Calcium (ca)	1316 mg
Phosphorus (ph)	1740 mg
Magnesium (mg)	380 mg
Potassium (po)	4044 mg
Iron (ir)	18 mg
Zinc (z)	14 mg
Calories (c)	2000 kcal
Protein (p)	91 g
Carbohydrate (car)	271 g

Figure 3. The positive nutrient requirements.

The objective function to be minimized $G^T x$ is the glycemic load of food combinations x (optimal diet) for diabetic patients. It is a linear function whose G vector components represent the glycemic load content of food per 100 g unit, determined using a nutrition study. This requires a great deal of time and effort on the part of our research team due to the large number of foods and the difficulties encountered in estimating the nutritional values of different foods. They strive to complete this complex task and provide accurate and relevant information in the field of nutrition. Some of the estimates of glycemic load values are shown in Table 3. The glycemic load (GL) is a measure of the number of carbohydrates that can be absorbed by the digestive tract and have an impact on blood sugar levels, based on the glycemic index (GI) of food, which measures the speed at which the carbohydrates in a food are digested, transformed, and end up as glucose in the blood [41]; the glycemic load of food is obtained by multiplying the glycemic index of the food by the number of carbohydrates contained in a specific portion of this food, then dividing the result by 100 [25].

Table 3. Glycemic load content.

Name of Foods	Glycemic Load (Min)	Glycemic Load (Mean)	Glycemic Load (Max)
Apricot	5.13	5.13	5.13
Dried Apricot	15.9	18.55	21.2
Garlic	3.225	3.225	3.225
Almond	0.15	0.15	0.15
Pineapple	3.57	3.753	3.936
Canned Pineapple	0	0.313	0.626
Artichoke	0.735	0.735	0.735
Asparagus	0.48	0.48	0.48
Eggplant	0.945	0.945	0.945
Avocado	0.626	0.626	0.626
Baguette	39.62	39.62	39.62
Banana	9.22	10.76	12.3
Beetroot	1.08	2.088	3.096
Cooked Egg White	0	0	0
Cooked Broccoli	0.165	0.165	0.165
Broccoli	0.42	0.42	0.42
Peanut	2.07	2.07	2.07
Raw Carrot	0.48	0.48	0.48
Peeled. Cooked Carrot (boiled)	3.102	4.356	5.61
Celery	0.18	0.18	0.18
Cooked Celery Stalk	0.24	0.24	0.24

3.4. Discussion

In this section, we will be presenting the results obtained from using the GQPSO and PSO approaches to address a dietary problem for diabetics. Our objective is to assess the effectiveness of these methods by comparing their performances and analyzing the nutritional gaps, as well as the positive and negative aspects of each approach. Figures 4 and 5 depict the optimal food choices selected by the GQPSO and PSO methods with varying values for parameters w_1, w_2, c_1, and c_2. Parameters w_1 and w_2 in the particle swarm algorithms used represent the inertia weights that influence the particles' behavior during iterations. In the pseudo-code for the method GQPSO, these parameters are denoted as ($w_1 = \beta_0$ and $w_2 = \beta_1$). Conversely, for the method PSO, they are represented as ($w_1 = r$ and $w_2 = R$) (Section 2). Adjusting these parameters during optimization by gradually changing the inertia can affect the algorithm's effectiveness. Some problems may require more exploration, which can be achieved by using higher values of w_1 at the start, while others may require quick convergence, which can be obtained by using lower values of w_2 toward the end. This approach enables dynamic exploration and exploitation, thereby improving overall performance. c_1 and c_2 are acceleration coefficients that adjust the impact of the best personal solutions (pbest) and the best global solution (gbest) on the movement of particles [6]. In GQPSO, these parameters are distributed according to a Gaussian distribution. While optimizing our model, we adjusted the parameters for each method, particle swarm optimization (PSO) and quantum particle swarm optimization (QPSO). This helped us to assess their impact on improving the glycemic load by exploring a wide range of possibilities and selecting the best food combinations. We kept parameters c_1 and c_2 equal during each adjustment to maintain a balance between the cognitive and social tendencies of the particles. For each method, we calculated the optimal glycemic load and the positive and negative nutritional deviations associated with each diet. The yellow highlights indicate the food quantities chosen for the various optimal diet solutions. It is important to note that the values represent the quantity of each food per 100 g unit.

v1	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.46	0.46	0.46	0.46	0.46	0.46
v2	0.22	0.22	0.22	0.22	0.34	0.34	0.34	0.46	0.46	0.46	0.10	0.10	0.22	0.22	0.22	0.34	0.34	0.46	0.46	0.46	0.10	0.10	0.22	0.34	0.34	0.34	0.46	0.46	0.46	0.46	0.10	0.22	0.22	0.22	0.34	0.46
c1=c2	0.80	1.60	1.80	2.00	1.60	1.80	2.00	1.20	1.40	1.80	0.80	1.80	0.60	1.00	2.00	0.60	1.20	0.80	1.00	1.80	1.60	2.00	1.60	0.60	0.80	1.80	0.60	1.60	1.80	2.00	1.60	0.80	1.40	1.60	1.60	1.80
pineapple	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Asparagus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Aubergine	0	6	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cooked egg (white)	0	6	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	2	0	6	0	0	0	0	0	0	0	0
Broccoli (cooked)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0
Broccoli	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Carrot (raw)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0
Celery	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0
Cauliflower	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Raw kohlrabi	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Lemon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
Clementine	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0
Cooked zucchini	0	0	0	5	0	3	0	2	0	0	0	0	6	6	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	2	0
Crab	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Endive	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0
Fig	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Finouil	4	0	2	0	1	0	0	0	1	0	6	0	0	0	2	0	0	0	3	0	0	3	2	3	0	6	0	0	1	0	1	3	0	2	0	0
Raw lamb liver	0	0	1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	1	1	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0
Strawberry	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Raspberry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0
Sesame seed	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
Green beans (cooked)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Lobster	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
pineapple juice	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Grapefruit juice	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0
Kaki	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Full-fat goat's milk	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Soya milk	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Semi-skimmed milk UI	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Lettuce	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0
Rabbit cooked meat	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Leas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0
Mandarin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
Blackberries	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
Blueberry	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coconut	6	6	6	6	6	6	2	6	6	6	0	6	6	6	0	6	6	6	6	6	6	6	6	0	0	0	6	6	6	6	0	6	6	0	0	0
Grapefruit	0	0	0	6	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0
Papaya	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Watermelon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
Roast pigeon	2	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0
Pear	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fish (raw whiting)	0	1	0	0	3	0	0	5	0	0	0	0	0	0	0	3	4	0	0	0	4	0	4	3	0	0	0	0	3	0	3	0	4	4	6	3
Grape Juice	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Greengage	0	0	0	0	0	0	6	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Salad	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	1	0
Green salad (without	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Tea	0	0	0	6	0	6	4	6	0	0	0	0	0	0	0	0	6	0	0	6	6	0	0	0	0	0	0	6	0	0	0	0	0	0	0	6
Glycemic load	13.77	17.42	54.12	19.19	47.70	37.62	34.33	25.69	30.99	405.22	6.91	41.37	12.84	17.10	29.52	36.58	58.12	47.32	41.76	38.83	46.02	30.92	12.55	20.03	8.79	6.74	90.00	9.69	38.68	42.06	9.24	37.64	12.57	13.98	3.18	53.86
Positive nutrient Gap	613.79	857.92	786.83	695.54	838.61	903.22	954.18	981.90	760.27	900.99	502.04	793.26	594.13	659.49	859.54	690.76	871.75	954.94	723.46	993.17	976.97	824.73	538.53	657.99	485.47	489.25	902.69	899.72	762.10	933.11	862.23	708.69	569.30	602.13	714.19	798.51
Negative nutrient gap	173.85	171.22	308.76	422.39	93.15	422.97	404.17	270.08	93.84	193.15	320.41	13.25	118.31	183.75	206.00	127.49	289.97	407.37	276.17	471.21	403.63	129.31	162.44	96.94	275.49	286.31	194.65	683.71	103.19	0.00	549.58	229.83	235.81	116.14	404.70	201.70

Figure 4. Optimal diet choices proven using standard PSO method.

v1	0.1	0.1	0.1	0.1	0.1	0.1	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.34	0.34	0.34	0.34	0.34	0.34	0.46	0.46	0.46	0.46	0.46	0.46
v2	0.1	0.34	0.34	0.46	0.46	0.46	0.34	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.1	0.22	0.34	0.34	0.34	0.46	0.22	0.22	0.22	0.34	0.34	0.46
c1=c2	0.6	1.2	1.6	0.8	1	1.6	1.8	0.6	0.8	1	1.2	1.4	1.6	1.8	0.6	1.6	0.8	1	1.2	1	1	1.2	2	1.2	1.8	1
Apricot	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Apricot dried	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Canned pineapple	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Banana	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Carrot (raw)	0	0	0	0	6	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Celery	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0
Cabbage, white	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cauliflower	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
Raw kohlrabi	0	0	1	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
Zucchini	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	1	0	0	2	0	0	0	0	0	0
Cooked zucchini	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0
Shrimp	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Endive	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
Spinach	2	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	4	0	0	0	0	4	0	0
Barbary fig	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Finouil	0	0	0	4	0	3	2	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0
Sesame seed	0	0	1	0	1	1	0	0	0	0	1	0	1	1	0	0	1	0	0	1	1	1	1	0	0	1
Pomegranate	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Green beans	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Green beans (raw)	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
Lobster	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Carrot juice	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kiwi	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Full-fat goat's milk	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Semi-skimmed milk UH	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Lettuce	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
Mango	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Blackberries	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Blueberry	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Cooked turnip	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Boiled egg	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chili pepper	2	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	2	0	0	0	0	0	0	0	1
Leek	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fish (raw whiting)	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
Potato	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Octopus	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Red radish	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Salad	0	0	1	0	0	0	1	1	0	0	0	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Green salad (without	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
Tea	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tomato	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Plain whole milk yogurt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Lemon zest	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Glycemic load	27.673649	51.7108	64.4479	36.2466	35.2852	80.6241	53.4283	81.5425	18.6407	56.8102	43.2513	42.704	44.3756	42.8692	57.8294	51.5068	54.4208	45.1723	47.7303	46.8684	57.4768	30.4128	55.0155	51.7609	55.3713	54.6199
Positive nutrient Gap	849.73988	917.806	984.069	911.224	971.442	882.057	967.349	974.639	719.325	995.346	885.047	884.861	862.275	987.682	909.23	982.866	954.167	998.811	869.91	963.89	860.177	964.849	981.376	812.24	990.064	934.916
Negative nutrient gap	498.84513	428.273	566.001	525.111	658.193	597.902	259.446	367.228	846.269	756.166	458.177	501.541	370.022	792.989	736.191	744.241	447.997	589.548	709.527	469.696	541.888	398.512	131.804	674.746	465.601	466.302

Figure 5. Optimal diet choices proven using the standard GQPSO method.

The analysis of the diets created using the PSO method (see Figure 4) shows that they contain a variety of essential nutrients. These dietary plans commonly include foundational foods such as coconut, known for its beneficial fatty acids and dietary fiber; sesame seeds, which are a good source of protein, fiber, and various minerals; and fennel, a low-calorie vegetable rich in fiber, vitamins, and minerals. In addition to fruits and vegetables, these diets also contain protein-rich foods like raw lamb's liver, which is high in quality protein, B vitamins, and essential minerals. Raw whiting is another inclusion that adds protein, omega-3 fatty acids, as well as a range of vitamins and minerals to the nutritional profile. Raw lamb liver, selected by PSO, is recommended in small amounts (100 g per diet). While it is low in carbohydrates (0.16 g/100 g) and high in protein (21 g/100 g), it contains saturated fats and should be eaten in moderation. It is chosen to diversify meals. To address the issue of repeatedly recommending certain foods in dietary plans, we have developed classification algorithms that combine Fuzzy C-means (FCM) and deep neural networks with auto-encoders. This method enables each food to belong to multiple groups, each with varying degrees of membership. As a result, we can substitute each food with coherent alternatives within the same group. This is crucial for ensuring dietary diversity and optimizing diet management [42–44].

Thus, for the diets provided by the GQPSO method (Figure 5), we can see that they are characterized by remarkable diversity. These diets significantly include items such as sesame seeds, spinach, and salad. Sesame seeds provide protein, fiber, and minerals, while spinach offers essential vitamins, antioxidants, and minerals. The presence of salad suggests a contribution to vitamins, fiber, and hydration. The combination of these elements in these diets suggests a balanced and varied approach, promoting a complete and health-promoting diet.

The results obtained by the two methods are then compared in Figures 6–8. This comparison involves evaluating the glycemic load reduced by each method and analyzing the differences between the minimum amounts of beneficial nutrients calculated by the two methods. Additionally, it includes comparing the negative nutrients calculated and the acceptable minimum for a diet.

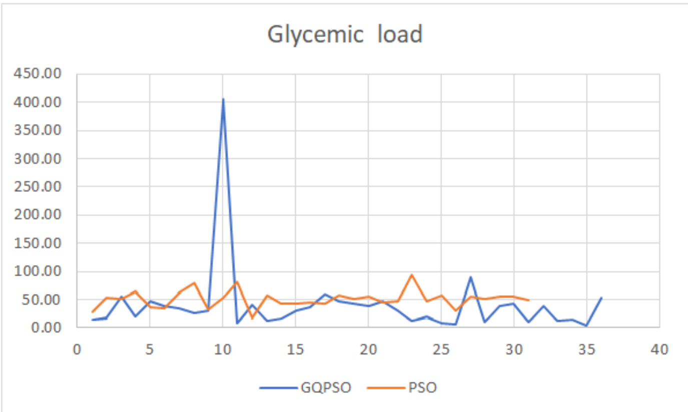

Figure 6. Comparison of improvements in glycemic load demonstrated by PSO and GQPSO methods.

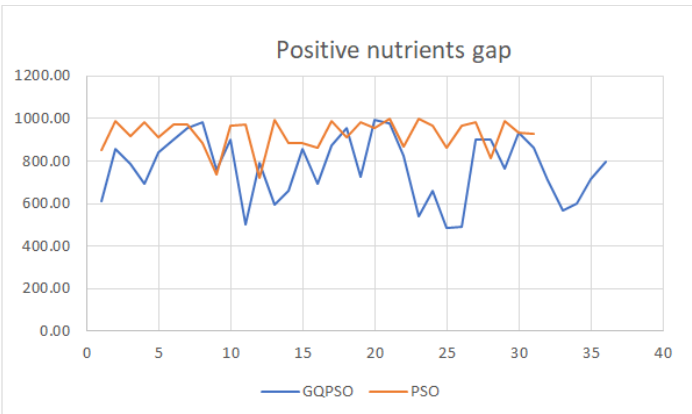

Figure 7. Comparison of positive nutrients gaps of diets generated by PSO and GQPSO techniques.

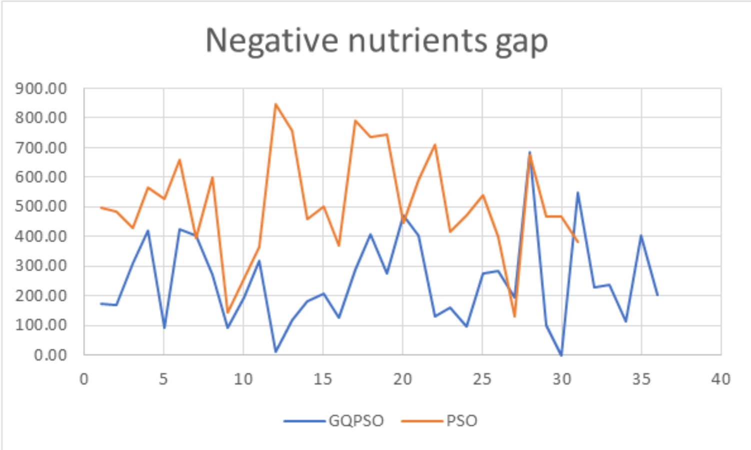

Figure 8. Comparison of negative nutrients gaps of diets generated by PSO and GQPSO techniques.

Figure 6 shows the GLs of the diets produced by PSO and GQPSO considering the different cases of study associated with different values of parameters (w_1, w_2, c_1, c_2). Except for the two diets, the diets produced by GQPSO have lower GLs than those provided by PSO. The fact that the GQPSO method produces diets with a minimal GL quantity compared to the PSO method indicates that GQPSO is more efficient in minimizing GL, and the improvement from GQSPO to PSO demonstrates an evolution in solving the problem of GL minimization.

Figure 7 highlights the gap between the quantities of positive nutrients generated by the PSO and GQPSO techniques considering the different cases of study associated with different values of parameters (w_1, w_2, c_1, c_2). This figure shows that the PSO method generates quantities of positive nutrients that are considered sufficient, exceeding those provided by the GQPSO method. In some cases, GQPSO outperforms PSO regarding positive nutrients. However, in other cases, adjusting the parameters can lead to minimizing the difference between the quantities generated by the two methods. Although the GQPSO method remains better at minimizing the glycemic load than the PSO method, it is notable that the latter can provide diets containing positive nutrients in higher quantities.

Figure 8 highlights the gap between the quantities of negative nutrients generated by the PSO and GQPSO techniques considering the different cases of study associated with different values of parameters (w_1, w_2, c_1, c_2).. We can see that, by modifying the parameters specific to each of the two methods, the GQPSO method stands out by generating diets with smaller deviations from tolerable values for negative nutrients than those produced by the PSO method. This feature shows the efficiency of the GQPSO method in designing diets that are more in accordance with the acceptable limits for negative nutrients. This makes it possible to assess the ability of the algorithms to control negative nutrients within acceptable limits, thereby contributing to the overall quality of the diet .

4. Conclusions

In addition to medical treatments, diet plays a fundamental role in glycemic control for diabetic patients, so many studies focus on nutritional optimization problems in this context, and the use of glycemic load concepts as optimization targets certainly makes an outstanding contribution in this field. In this work, we carried out a comparative study of a diet optimization problem in which we compared the performance of two optimization approaches: a classical swarm particle approach and one with quantum behavior and Gaussian mutation. Our study demonstrates that combining quantum physics principles with random processes in GQPSO yields significant improvements in managing the glycemic load and nutritional deficiencies within dietary patterns. Due to the variability regarding foods' nutritional values, it is often difficult to estimate them accurately; this complicates data collection and decision-making regarding the results. In our subsequent work, we will seek to integrate particle swarm optimization principles with artificial intelligence techniques in these methods to improve their performance.

Author Contributions: Conceptualization, A.A. and Z.B.; Methodology, K.E.M.; Software, A.A.; Validation, A.T.; Formal analysis, Z.B.; Investigation, K.E.M.; Writing—original draft, A.A. and A.T.; Writing—review & editing, Z.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Ministry of National Education, Professional Training, Higher 707 Education and Scientific Research (MENFPESRS) and the Digital Development Agency (DDA) and 708 CNRST of Morocco (Nos. Alkhawarizmi/2020/23).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. International Diabetes Federation. *IDF Diabetes Atlas*, 10th ed.; Citeseer: Princeton, NJ, USA, 2023. Available online: https://diabetesatlas.org/ (accessed on 11 January 2024).
2. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
3. Sun, J.; Feng, B.; Xu,W.-B. Particle Swarm Optimization with Particles Having Quantum Behavior. In Proceedings of the Congress on Evolutionary Computation, Piscataway, Portland, OR, USA, 19–23 June 2004; pp. 325–331.
4. Sun, J.; Xu, W.B.; Feng, B. A global search strategy of quantum behaved particle swarm optimization. In Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, Singapore, 1–3 December 2004; pp. 111–116.
5. Bhatia, A.S.; Saggi, M.K.; Zheng, S.; Nayak, S.R. QPSO-CD: Quantum-behaved Particle Swarm Optimization Algorithm with Cauchy Distribution. *Quantum Inf. Process.* **2020**, *19*, 345. [CrossRef]
6. Dos Santos Coelho, L. Gaussian quantum-behaved particle swarm optimization approaches for constrained enginee ring design problems. *Expert Syst. Appl.* **2010**, *37*, 1676–1683. [CrossRef]
7. Aroniadi, C.; Beligiannis, G.N. Solving the Fuzzy Transportation Problem by a Novel Particle Swarm Optimization Approach. *Appl. Sci.* **2024**, *14*, 5885. [CrossRef]
8. Aroniadi, C.; Beligiannis, G.N. Applying Particle Swarm Optimization Variations to Solve the Transportation Problem Effectively. *Algorithms* **2023**, *16*, 372. [CrossRef]
9. Kourepinis, V.; Iliopoulou, C.; Tassopoulos, I.X.; Aroniadi, C.; Beligiannis, G.N. An Improved Particle Swarm Optimization Algorithm for the Urban Transit Routing Problem. *Electronics* **2023**, *12*, 3358. [CrossRef]
10. Tassopoulos, I.X.; Iliopoulou, C.A.; Katsaragakis, I.V.; Beligiannis, G.N. An Effective Local Particle Swarm Optimization-Based Algorithm for Solving the School Timetabling Problem. *Algorithms* **2023**, *16*, 291. [CrossRef]
11. Adhikari, M.; Srirama, S.N. Multi-objective accelerated particle swarm optimization with a container-based scheduling for internet-of-things in cloud environment. *J. Netw. Comput. Appl.* **2019**, *137*, 35–61. [CrossRef]
12. Kumar, S.; Pal, S.K.; Singhm R. A novel hybrid model based on particle swarm optimisation and extreme learning machine for short-term temperature prediction using ambient sensors. *Sustain. Cities Soc.* **2019**, *49*, 101601. [CrossRef]
13. He, F.; Chen, C.; Li, F.; Qi, Y.; Lin, X.; Liang, P.; Ren, M.; Yan, L. An optimal glycemic load range is better for reducing obesity and diabetes risk among middle-aged and elderly adults. *Nutr. Metab.* **2021**, *18*, 31. [CrossRef]
14. Bas, E. A robust optimization approach to diet problem with overall glycemic load as objective function. *Appl. Math. Model.* **2014** *38*, 4926–4940. [CrossRef]
15. El Moutaouakil, K.; Cheggour, M.; Chellak, S.; Baizri, H. Metaheuristics optimization algorithm to an optimal Moroccan diet. In Proceedings of the 2021 7th Annual International Conference on Network and Information Systems for Computers (ICNISC), Guiyang, China, 23–25 July 2021; pp. 364–368.
16. Shi, Y.; Eberhart, R.C. A modified particle swarm optimizer. In Proceedings of the IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
17. Eberhart, R.C.; Shi, Y. Comparing inertia weights and constriction factors in particle swarm optimization. In Proceedings of the 2000 Congress on Evolutionary Computation, La Jolla, CA, USA, 16–19 July 2000; Volume 81, pp. 84–88.
18. Schweizer, W. *Numerical Quantum Dynamics*; Springer Science & Business Media: Hingham, MA, USA, 2001.
19. Liu, J.; Xu, W.; Sun, J. Quantum-behaved particle swarm optimization with mutation operator. In Proceedings of the 17th International Conference on Tools with Artificial Intelligence, Hong Kong, China, 14–16 November 2005.

20. Clerc, M.; Kennedy, J.F. The particle swarm: Explosion, stability and convergence in a multi-dimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [CrossRef]
21. Sun, J.; Fang, , W.; Wu, X.J.; Palade, V.; Xu, W.B. Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection. Evol. Comput. 2012, 59, 3686–3694. [CrossRef]
22. Xi, M.; Sun, J.; Xu, W. An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position. *Appl. Math. Comput.* **2008**, *205*, 751–759. [CrossRef]
23. Moon, J.H.; Lee, K.H.; Kim, H.; Han, D.I. Thermal-Economic Optimization of Plate–Fin Heat Exchanger Using Improved Gaussian Quantum-Behaved Particle Swarm Algorithm. *Mathematics* **2022**, *10*, 2527. [CrossRef]
24. Wei, Z.; Ye, S. Quantum-Behaved Particle Swarm Optimization Algorithm with Adaptive Mutation Based on q-Gaussian Distribution. *Chin. J. Electron.* **2012**, *21*, 449–452.
25. Available online: https://mail.glycemicindex.com/faqsList.php (accessed on).
26. El Moutaouakil, K.; Saliha, C.; Hicham, B.; Mouna, C. Intelligent Local Search Optimization Methods to Optimal Morocco Regime. In *Swarm Intelligence-Recent Advances and Current Applications*; IntechOpen: London, UK, 2023.
27. Abdellatif, E.O.; Karim, E.M.; Hicham, B.; Saliha, C. Intelligent local search for an optimal control of diabetic population dynamics. *Math. Models Comput. Simul.* **2022**, *14*, 1051–1071. [CrossRef]
28. Ahourag, A.; Chellak, S.; Cheggour, M.; Baizri, H.; Bahri, A. Quadratic Programming and Triangular Numbers Ranking to an Optimal Moroccan Diet with Minimal Glycemic Load. *Stat. Optim. Inf. Comput.* **2023**, *11*, 85–94.
29. El Moutaouakil, K.; Ahourag, A.; Chakir, S.; Kabbaj, Z.; Chellack, S., Cheggour, M.; Baizri, H. Hybrid firefly genetic algorithm and integral fuzzy quadratic programming to an optimal Moroccan diet. *Math. Model. Comput.* **2023**, *10*, 338–350. [CrossRef]
30. El Moutaouakil, K.; Yahyaouy, A.; Chellak, S.; Baizri, H. An optimized gradient dynamic-neuro-weighted-fuzzy clustering method: Application in the nutrition field. *Int. J. Fuzzy Syst.* **2022**, *24*, 3731–3744. [CrossRef]
31. El Moutaouakil, K.; Touhafi, A. A new recurrent neural network fuzzy mean square clustering method. In Proceedings of the 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech), Marrakesh, Morocco, 24–26 November 2020; pp. 1–5
32. El Moutaouakil, K.; Palade, V.; Safouan, S.; Charroud, A. FP-Conv-CM: Fuzzy Probabilistic Convolution C-Means. *Mathematics* **2023**, *11*, 1931. [CrossRef]
33. Ahourag, A.; El Moutaouakil, K.; Chellak, S.; Baizri, H.; Cheggour, M. Multi-criteria optimization for optimal nutrition of Moroccan diabetics:* Note: Sub-titles are not captured in Xplore and should not be used. In Proceedings of the 2022 International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 18–20 May 2022; pp. 1–6.
34. Ahourag, A.; El Moutaouakil; K.; Cheggour, M.; Chellak, S.; Baizri, H. Multiobjective Optimization to Optimal Moroccan Diet Using Genetic Algorithm. *Int. J. Eng. Model.* **2023**, *36*, 67–69.
35. El Moutaouakil, K.; El Ouissari, A.; Palade, V.; Charroud, A.; Olaru, A.; Baïzri, H.; Cheggour, M. Multi-objective optimization for controlling the dynamics of the diabetic population. *Mathematics* **2023**, *11*, 2957. [CrossRef]
36. El Moutaouakil, K.; El Ouissari, A.; Hicham, B.; Saliha, C.; Cheggour, M. Multi-objectives optimization and convolution fuzzy C-means: Control of diabetic population dynamic. *RAIRO-Oper. Res.* **2022**, *56*, 3245–3256. [CrossRef]
37. Ahourag, A.; El Moutaouakil, K.; Elkari, B.; Hammouni, A.; Ourabah, L.; Chellak, S. A Multiobjective Diet Planning Model for Diabetic Patients in the Moroccan Health Context Using Particle Swarm Intelligence. *Stat. Optim. Inf. Comput.* **2024**, *12*, 605–616. [CrossRef]
38. El Moutaouakil, K.; Ahourag, A.; Chellak, S.; Baïzri H.; Cheggour, M. Fuzzy Deep Daily Nutrients Requirements Representation. *Rev. D'intell. Artif.* **2022**, *36*, 263. [CrossRef]
39. El Moutaouakil, K.; Baizri, H.; Chellak, S. Optimal fuzzy deep daily nutrients requirements representation: Application to optimal Morocco diet problem. *Math. Model. Comput.* **2022**, *9*, 607–615. [CrossRef]
40. U.S. Department of Health and Human Services, U.S. Department of Agriculture. *Dietary Guide- lines for Americans*; U.S. Department of Health and Human Services, U.S. Department of Agriculture: Washington, DC, USA, 2010.
41. Atkinson, F. S.; Foster-Powell, K.; BranMiller, J. C.International tables of glycemic index and glycemic load values: 2008. *Diabetes care.* **2008**, *31*, 2281–2283. [CrossRef]
42. Bounabi, M.; Moutaouakil, K. E.; Satori, K. The Optimal Inference Rules Selection for Unstructured Data Multi-Classification. *Stat. Optim. Inf. Comput.* **2022**, *10*, 225–235. [CrossRef]
43. El Moutaouakil, K.; Ahourag, A.; Belhabib, F.; Hammoumi, A.; Patriciu, A.-M.; Chellak, S.; Baizri, H. Fuzzy Modeling to Personalized Nutritional Menu. *Curr. Nutr. Food Sci.* **2025**, *21*, e210324228226. [CrossRef]
44. El Moutaouakil, K.; Roudani, M.; Ahourag, A.; Aayah, H.; Mouna, C.; Chellak, S.; Baizri, H. A Deep Fuzzy Neural Network System to Group Moroccan Foods: Towards a Personalized Menu for Type 2 Diabetes Patient. *Nutr. Food Sci. Int. J.* **2024**, *13*, 555860. [CrossRef]