Chapter

Intelligent Local Search Optimization Methods to Optimal Morocco Regime

Karim El Moutaouakil, Chellak Saliha, Baïzri Hicham and Cheggour Mouna

Abstract

In this paper, we compare three well-known swarm algorithms on optimal regime based on our mathematical optimization model introduced recently. Different parameters of this latter are estimated based on 176 foods and on who's the nutrients values are calculated for 100 g. The daily nutrients needs are estimated based on the expert's knowledge. Different experimentations are realized for different configurations of the considered swarm algorithms. Compared to Stochastic Fractal Search (SFS) and Particle Swarm Optimization Algorithm (PSO), the Firefly Algorithm (FA) produces the main suitable regimes.

Keywords: optimal regime, favorable nutrient, unfavorable nutrient, quadratic optimization, stochastic fractal search, firefly algorithm, optimization swarm algorithm

1. Introduction

For healthy individuals, balanced diets reduce the likelihood of developing chronic diseases; whereas for individuals with chronic diseases, balanced diets reduce the likelihood of entering dangerous stages, especially for diabetics, cardiovascular disease, obesity and cancer [1–6]. It is a matter of satisfying the body's demands in an optimal manner.

The earliest optimization model, relating to the diet issue, was suggested in [7] with the regime cost as an objective function. Within [8], the target function was minimization of weighted meal compositions, implicating case- and rule-based reasoning; in which any new daily vegan menu consisted of breakfast, lunch, dinner, a snack, and, in additional, a fruit serving. Further suggestions [9] involve minimizing the difference between the real and advised consumption whilst satisfying the nutritional needs. In studies [10], the authors suggest supplemental plans (children under the age of 2 years) and dietary plans (school age group 13–18 years) at the lowest total cost. To further investigate more features, various multi-objective driven schemes were suggested. While generating food meals, the authors of [11] tackled the economical and aesthetical aspects (taste, flavor, color …). When forming the objective functions of their mathematical optimization model, the authors of the article [12] included the price of regime, and other aspects like carbon dioxide emissions, land, and water consumption, etc. V. Mierlo have considered nearly the identical case by substitution of the regime cost and the fossil fuel depletion minimization [13]. At [14], the authors suggest a multi-objective programming framework which delivers a nutritional program plan and minimizes glycemic load and cholesterol consumption, seen as the major causes of childhood overweight.

Recently, we have proposed an original mathematical optimization model for the optimal diet problem. In this paper, we compare three well-known swarm algorithms on optimal regime based on our mathematical optimization model introduced recently [5]. Different parameters of this latter are estimated based on 176 foods who's the nutrients values are calculated for 100 g. The daily nutrients needs are estimated based on the expert's knowledge [6].

The remainder of the material is structured as follows: the second section concerns the mathematical model of the diet problem. The third section is about the three swarm optimization methods: SFS, FA, and PSO. In the fourth section, several experimental results are presented and analyzed. At the end, some conclusions and future propositions are discussed.

2. Optimal regime mathematical model

The quadratic optimization problem which permits the control the total glycemic load of the regime, the lack of positive nutrients, and overdose of negative nutrients in the regime is given by the coming Equations [5, 6, 15]:

$$(D):\begin{cases}\text{Min}\, g^{\intercal}x+\theta\,\text{dist}(Ax,b)+\sigma\,\text{dist}(Ex,f)\\ \textit{Subject to}:\\ c_i^{\intercal}x\geq\rho_i(C^tx)\quad,\ j\in\{car,p\}\\ c_j^{\intercal}x\leq\tau_j(C^tx),\ \ j\in\{tf,sf\}\\ \mathrm{x}\in[0\ 6]^{176}\end{cases}\tag{1}$$

In the problem (D), $\rho_{car}=0.55$, $\rho_p=0.18$, $\tau_{tf}=0.29$, and$\tau_{sf}=0.078$ represent the ratios recommended by WHO [16]; g represents the matrix of glycemic load of foods taking into account possible variations; A symbolizes the knowledge of foods in terms of positive nutrients; E gives the amount of negative nutrients in foods; f and b are the daily requirements of positive and negative nutrients, respectively; C is the vector of the foods calories extracted from A; c_{car}, c_p, c_{tf}, and c_{sf} are the calories from carbohydrate, potassium, total fat, and satured fat, respectively. Finally, θ and σ are parameters to control different components of the cost function.

In the Section 4, we will use three optimization swarm algorithms to estimate the optimal diet based on our model for different configurations.

3. Principles and complexity of firefly local search algorithm

This part concerns a brief description of the smart local search optimization methods, called firefly algorithm, we used to solve the diet problem (P).

Firefly algorithm: The Firefly Algorithm (FirA) was originally pioneered by Xin-She Yang [17, 18], on the basis of flashing and behavior models of fireflies. Essentially, FA employs three rules:

a. Fireflies are single-gender and a firefly might be attracting another firefly whatever its gender.

b. Attraction is directly correlated to brightness. If two fireflies are blinking, the darker one will move closer to the lighter one. If there is no firefly with more light, then a random firefly will change its place.

c. The luminosity of a firefly is decided based on the cost function of the problem to be solved.

Because the attractiveness of a firefly is shown to be proportional to the brightness seen by nearby fireflies, given to firefly i and j, the variability of attractiveness δ_{ij}, given the distance d_{ij}, is given by:

$$\delta_{ij} = \delta_0 \exp\left(-\sigma d_{ij}^2\right) \tag{2}$$

Where δ_0 is the basic attracness and σ is a parameter chosen by the user and σ can be chosen based on the formula $\sigma = \sqrt{L}^{-1}$, such that L depends on the large scale of the problem.

Given the current position of the ith x_i^t and jth x_j^t fireflies and the distance between these particles, noted d_{ij}, the position of the ith firefly is updated by:

$$\mathrm{x}_\mathrm{i}^{\mathrm{t}+1} = \mathrm{x}_\mathrm{i}^\mathrm{t} + \delta_{\mathrm{ij}}\left(\mathrm{x}_\mathrm{j}^\mathrm{t} - \mathrm{x}_\mathrm{i}^\mathrm{t}\right) + \alpha_\mathrm{t}\varepsilon_\mathrm{i}^\mathrm{t} \tag{3}$$

The **Figure 1** illustrates the behavior of the ith firefly considering the nearest strong firefly; The random term permits to explore more regions.

α_t is a global random serie of parameters and ε_i^t is personalized local random serie of of parameters linked to the ith firefly. The **Figure 2** gives different steps of the FA algorithm.

Parameters: A good way to control the algorithm randomness is consists on updating α_t based on the formula $\alpha_t = \alpha_0 a^t$ where $a \in [.95.97]$; α_0 represents the initial randomness control factor [18] and can be chosen using the formula $\alpha_0 = .001L$.

Complexity: Considering the two loops of FA, the complexity at the extreme case is $O\left(N^2T\right)$, where N is the number of generated individuals and T is the number of

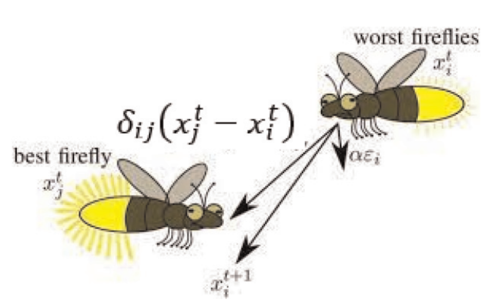

Figure 1.
Ullistration of the behavior of the ith firefly considering the nearst strong firefly.

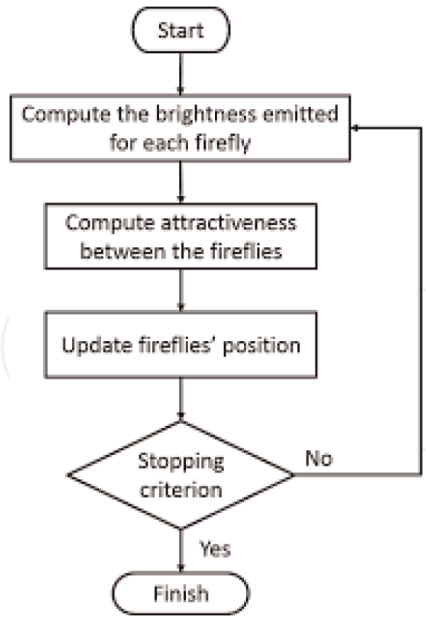

Figure 2.
Diagram of the FA algorithm.

iterations. To reduce the complexity of FA, we can rank the attractiveness or brightness using sorting algorithms and the complexity becomes $O(TNlog(N))$.

Variants: In the case of combinatorial optimization Problems, variants of FA were developed with improved efficiency [19–21].

4. Stochastic fractal search algorithm

SFS is inspired by the background process of development. This algorithm is a computational search method that utilizes a mathematical principle known as a fractal [22]. Fractal search uses 3 rules to come up with a solution: (a) every particle has an electrical potential energy, (b) every particle spread and induces the generation of more random particles, and the starting particle's energy is shared among the newly formed particles, and (c) just a small amount of the better particles stay in the next round, and the remaining particles are skipped. The **Figure 3** illustrates the diffusion

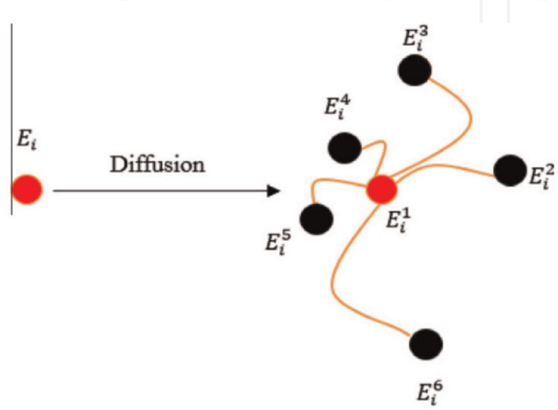

Figure 3.
Particle diffusion.

of the particle E_i . This strategy works well in identifying the solution; however, the method has its drawbacks.

The major problem is the high number of parameters required to be properly managed, and the additional issue is that the interchange of knowledge is not taking place between the individual. To overcome the above challenges, Salimi, H. introduced another version of fractal search called stochastic fractal search [22].

In the SFS algorithm, two main operations take place: the diffusion operation and the updating operation. In the first operation, each particle scatters around its current position to satisfy the intensification (exploitation) property. In the latter operation, the algorithm mimics the way an individual updates his location depending on the position of the remaining individual in this cluster.

To generate new individual from the scattering operation, Lévy and Gaussian flight are investigated as two statistical methods. Generally, a sequence of Gaussian treads participating in the scattering operation were listed in the next equations:

$$\mathrm{GW_1 = N(\mu_{BP}, \sigma) + \varepsilon BP - \varepsilon' P_i \text{ and } GW_2 = N(\mu_P, \sigma)} \tag{4}$$

Here $\varepsilon, \varepsilon' \sim U([0\ 1])$, BP denotes the global best position, P_i is the position of the current particle, $\mu_{BP} = BP$, $\mu_P = P_i$, and σ is given by $\sigma = \left|\frac{log\,(g)}{g}(P_i - BP)\right|$; g represents the number of iterations and $\frac{log\,(g)}{g}$ permits to reduce the size of the normal step.

To ensure a good exploration of the research domain, two statistical strategies are considered:

(a) A uniform probability weight is attributed to each individual i in the group:

$$Pa_i = \frac{the\ rank\ of\ the\ point\ i\ in\ the\ group}{the\ number\ of\ the\ points\ in\ the\ group} = \frac{rank(P_i)}{N}. \tag{5}$$

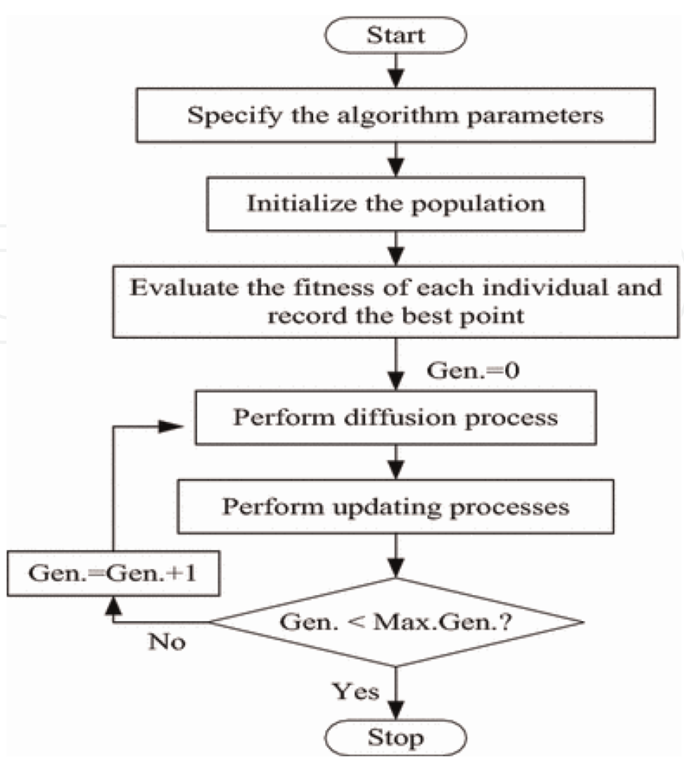

Figure 4.
Diagram of SFS algorithm.

In this sense, Pa_i is less than a given threshold, the position of the ith point, from the group G, is updated using the equation:

$$P_i' = P_{rand1\_G} - \varepsilon(P_{rand2\_G} - P_i) \text{ such that } \varepsilon \sim U([0\ 1]) \tag{6}$$

As in the first process, if the $Pa_i \leq \varepsilon$ holds, the current particle is changed:
If $\varepsilon' \leq .5$, then $P_i'' = P_i' - \hat{\varepsilon}\left(P'_{rand1\_G} - PB\right)$, else $P_i'' = P_i' - \hat{\varepsilon}\left(P'_{rand1\_G} - P'_{rand2\_G}\right)$,
Where $\hat{\varepsilon} \sim U([0\ 1]$
The **Figure 4** illustrates different steps of SFS algorithm; for more details, the reader can see the paper of Salimi [22].

5. Particle swarm algorithm

Particle swarm optimization, first introduced by Kennedy and Eberhart [23], is a synthetic meta-heuristic approach to global computer optimization, belonging to the swarm intelligence concept-based algorithm family of approaches.

5.1 Basic PSO algorithm

Each potential solution is known as a "particle" within PSO and the location of the ith particle may be determined by $\boldsymbol{p_i} = \left(\boldsymbol{p_{ij}}\right)_{j=1,\dots,n}$ where n is the dimension of the search space. From now on, we suppose that we have a swarm P of N particle $\boldsymbol{p_1}, \dots, \boldsymbol{p_N}$.
During the search process, the particles update their positions using the motion equation:

$$\mathrm{p}_i^{t+1} = \mathrm{p}_i^t + \mathrm{v}_i^{t+1} \tag{7}$$

The ith particle velocity is given by:

$$\mathrm{v}_i^{t+1} = \mathrm{v}_i^t + \mathbf{c}_1\left(\mathrm{bp}_i - \mathrm{p}_i^t\right)\mathbf{r}_1 + \mathbf{c}_2\left(\mathrm{bg} - \mathrm{p}_i^t\right)\mathbf{r}_2 \tag{8}$$

Such that $\boldsymbol{bp_i}$ is the best position of the particle i, g is the global best position of the swarm members, $\boldsymbol{c}_k, k = 1, 2$, is the acceleration parameters usually thoken from the interval $[0\ 4]$ named also "cognitive coefficient", and $\boldsymbol{r}_k = \mathrm{diag}(\mathrm{uniform}([0\ 1])), k = 1, 2$. The **Figure 4** illustrates the PSO formula used to update the particles positions (**Figure 5**).

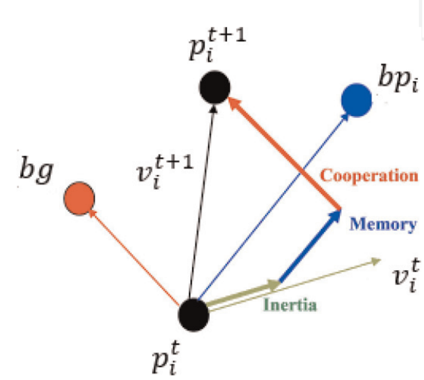

Figure 5.
PSO learning equation illustration.

The basic PSO pseudo-code can be the following:

1. Initialization. For each of the N particles:

   a. Initialize the position p_i^0;

   b. Initialize the particle's best position to this initial position $bp_i^0 = p_i^0$;

   c. Calculate the fitness of each particle and $\mathrm{bg} = p_j^0$ with $f\left(p_j^0\right) \geq f\left(p_i^0\right)$.

2. Repeat the coming steps until convergence:

   a. Update the velocity using:

$$\mathrm{v}_\mathrm{i}^{\mathrm{t}+1} = \mathrm{v}_\mathrm{i}^\mathrm{t} + \mathbf{c}_1\left(\mathrm{bp}_\mathrm{i} - \mathrm{p}_\mathrm{i}^\mathrm{t}\right)\mathbf{r}_1 + \mathbf{c}_2\left(\mathrm{bg} - \mathrm{p}_\mathrm{i}^\mathrm{t}\right)\mathbf{r}_2 \tag{9}$$

   b. Update the particle position using:

$$\mathrm{p}_\mathrm{i}^{\mathrm{t}+1} = \mathrm{p}_\mathrm{i}^\mathrm{t} + \mathrm{v}_\mathrm{i}^{\mathrm{t}+1} \tag{10}$$

   c. Evaluate the ith particle fitness $f\left(p_i^{t+1}\right)$;

   d. If $f\left(p_i^{t+1}\right) \geq f\left(p_i^{t+1}\right),; bp_i = p_i^{t+1}$

   e. If $f\left(p_i^{t+1}\right) \geq bg),; bg = p_i^{t+1}$

3. At the convergence the best solution is bg.

5.2 PSO meta parameters

Initialization: PSO involves an initial estimate of the positions and velocities. For the initial positions, a general consensus is to cover the solution space on a uniform basis: $p_{ij}^0 \sim U\left(\left[LB_j, UB_j\right]\right)$. For initial velocities, it is suggested to use a uniform distribution to ensure a uniform coverage of the search space. But this could augment the probability of particles being infeasible solutions. To defeat this inconvenience, the velocities may be set to zero or to very tiny arbitrary numbers.

Acceleration constants: The parameters c_1 and c_2 have a very large impact on the particle's paths and on the algorithm convergence. In this sense, the larger these constants are, the more the oscillation of the particle around the optimum increases, whereas very small values give rise to sinusoidal patterns. In general, it is recommended to set these parameters to 2 [24].

Swarm size: A large swarm size improves the variety of the swarm and its exploration ability, but in another way, it may also increase the risk of an early convergence and the calculation costs. Nevertheless, in most situations, it has actually been found that once the swarm size is higher than 50 particles, PSO becomes insensitive to the swarm size [24].

6. Experimentation and analysis

We utilize FA, PSO, and SFS algorithms to establish optimal regimes based on the proposed mathematical model in [5] where $\theta = 0.67$ and $\sigma = 1.34$. The WHO recommendations concerning the nutrients daily needs were token into considerations [6, 25, 26]. We work on 176 aliments considered as the most consumed in Morocco. The linear part of our model is estimated using the means glycemic load of the considered foods. From now on, we adopt the symbols: TGL for Total Glycemic Load, FTG for Favorable Totale Gap, and UFTG for UFavorable Totale Gap.

a. We used the SFS algorithm to solve problem (D). We tested this algorithm for different values of the parameters: walk probability, maximum diffusion, and the number of iterations. Th **Table 1** gives TG, FTG, and UFTG of diets produced by SFS for max diffusion equals to 5, start points equals to 50, number of iterations of 200, and different values of walk probability from the interval [0.3 0.9] adopting 0.1 as step.

The best diet is the one produced by SFS for walk probability value equals to 0.7 with glycemic load in the interval [82.2152 92.5292] and nutrients requirements gaps

SFS walk probability	Diet total glycemic load			FTG	UFTG
	min	mean	max		
0.3	95.4564	116.5515	120.6926	210.7442	18.9232
0.4	87.0935	94.1033	97.3494	190.9682	30.2345
0.5	89.6853	97.8046	98.4658	216.7432	37.7244
0.6	101.0979	113.8293	110.0329	170.6037	22.8605
0.7	**82.2152**	**87.4453**	**92.5292**	**143.3103**	**30.9554**
0.8	86.3963	94.3195	94.5612	151.0829	30.5128
0.9	86.3344	94.2951	99.5339	164.5838	46.2813

Table 1.
TG, FTG, and UFTG of diets produced by SFS for max diffusion = 5, start points = 50, number of iterations of 200, and different values of walk probability.

SFS diffusion	Diet total glycemic load			FTG	UFTG
	min	mean	max		
5/45	76.9257	82.9567	83.2560	133.2240	40.1505
5/50	82.2152	87.4453	92.5292	143.3103	30.9554
6/45	73.0082	84.9846	87.1250	196.1147	48.0560
7/45	94.8373	98.9290	101.1763	77.5158	6.1873
9/45	84.0789	99.0630	100.9079	106.8640	18.2439
10/45	**68.8041**	**74.0633**	**76.3373**	**52.0240**	**47.8260**

Table 2.
TG, FTG, and UFTG of diets produced by SFS for start points = 50 (45), number of iterations of 200, walk probability of 0.7, and different values of diffusion.

143.3103 mg (for positive nutrients) and 30.9554 mg (for negative nutrients). These diets still bad considering the considered three criterions. To investigate possible improvements, we set the walk probability to 0.7 and, start points to 45, and number of iterations to 200, and we variate the value of diffusion.

The **Table 2** give TGL, FTG, and UFTG of diets produced by SFS for start points equals to 50(45), number of iterations of 200, walk probability of 0.7, and different values of diffusion from [5 10] by adopting 1 as step. The obtained diets become to be acceptable and the best diet is the one who's TGL is in [**68.8041 76.3373**], FTG = **52.0240**, and UFTG = **47.8260.**

To investigate more improvements, we set max diffusion to 10, walk probability to 0.7, start points to 45, and we vary different number of iterations; see **Table 3**.

Indeed, we detect a very good diet (produced by SFS) for 600 number of iterations with TG is in [53.8780 66.0715], FTG = 50.1917, and UFTG = 28.5891. The **Figure 6** illustrates the behavior of (D) objective function when solving the diet problem using SFS for max diffusion equals to10, walk probability equals to 0.7, start points equals to 45, and the number of iterations equals to 600; it is clear that the algorithm has not yet

SFS iterations number	The diet total glycemic load			FTG	UFTG
	min	mean	max		
300	68.8041	74.0633	76.3373	52.0240	47.8260
400	91.5940	95.7307	98.1429	86.6777	10.5893
500	82.1371	88.2860	93.6195	119.4418	12.2939
600	**53.8780**	**60.5048**	**66.0715**	**50.1917**	**28.5891**
700	79.8448	82.8554	85.6542	68.4567	16.4858
800	84.6661	95.5110	105.3748	23.6773	21.2563

Table 3.
TG, FTG, and UFTG of diets produced by SFS for max diffusion 10, walk probability 0.7, start points 45, and different number of iterations.

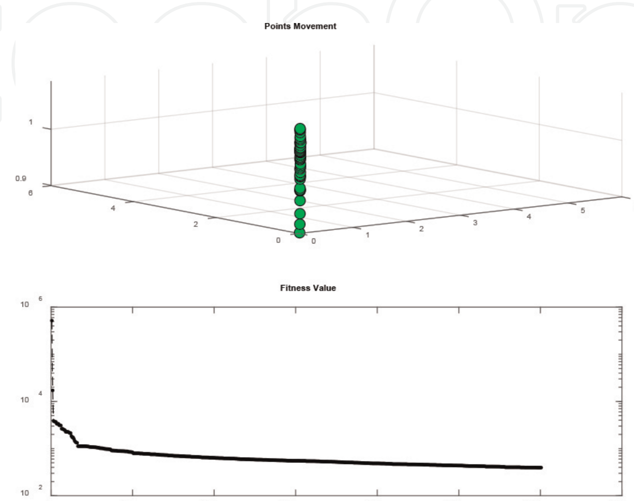

Figure 6.
Evolution of the model (D) fitness with iterations by SFS for walk probability of 0.7, maximum diffusion of 10, and number of iteration equals to 600.

converged and an additional number of iterations will allow more improvement, but we compare the algorithms for a very small number of iterations to get a good diet in real time.

b. We used the FA algorithm to solve problem (D). We tested this algorithm for different values of the parameter's population, attraction coefficient base value, iterations, and of Mutation coefficient damping ratio.

The **Table 4** give TG, FTG, and UFTG of diets produced by FA for: population 40, attraction coefficient base value of 2.25, iterations of 300, variation of mutation coefficient damping ratio from in [0.1 0.9] with 0.1 as step.

All the produced diets are acceptable and the best diet is the one produced for Mutation Coefficient Damping Ratio equals to 0.4. To investigate more improvements

FA Mutation Coefficient	Diet total glycemic load			FTG	UFTG
	min	mean	max		
0.1	52.6591	53.0527	53.4451	32.7860	3.1337
0.2	52.9493	54.3853	55.8213	10.0024	12.5275
0.3	77.3769	78.5940	79.7333	10.0022	11.3670
0.4	**68.7673**	**71.0372**	**73.1670**	**14.8004**	**4.7598**
0.5	69.6771	70.7284	71.7793	19.9331	7.4906
0.6	59.7460	61.6316	63.3772	5.1746	19.7053
0.7	64.4724	65.4970	66.3804	16.5202	20.7693
0.8	59.0634	60.0857	61.0316	12.5387	1.0147
0.9	63.6272	64.4156	65.0629	11.0490	18.8717

Table 4.
Diet produced by FA for population equals to 40, attraction coefficient base value of 2.25, iterations equals to 300, and variation of mutation coefficient damping ratio.

FA population size	Diet total glycemic load			FTG	UFTG
	min	mean	max		
20	64.2545	65.9005	67.5535	13.5829	32.0750
25	42.2206	43.8853	45.4469	56.8449	7.4848
30	54.6090	56.0057	57.3672	72.9725	5.3529
35	79.2812	81.8633	84.3053	3.0136	15.7434
40	68.7673	71.0372	73.1670	14.8004	4.7598
45	76.3777	78.0148	79.6519	26.6040	3.2404
50	**53.5439**	**54.9658**	**56.3875**	**10.6000**	**7.8365**

Table 5.
Diets produced by FA for attraction coefficient base value equals to 2.25, iterations equals 300, mutation coefficient damping ratio = 0.4, and variation of population.

of this diets, we variate the number of iterations will setting the mutation coefficient damping ratio to 0.4; see **Table 5**.

In fact, the quality of diets were improveded and the best one is obtained for attraction coefficient base value equals to 2.25, iterations equals to 300, mutation coefficient damping ratio equals to 0.4, size population = 50 with TGL is in [**53.5439 56.3875**], FTG = **10.6000 mg**, and UFTG = **7.8365 mg.**

The **Figure 7** illustrates the behavior of (D) objective function when solving the diet problem using FA for coefficient base value equals to 2.25, iterations equals to 200, mutation coefficient damping ratio equals to 0.4, and size of population equals to 50. We remark that FA algorithm reaches early a very good local solution.

c. We used the PSO algorithm to solve problem (D). We tested this algorithm for different values of iterations, self-adjustment weight, social-adjustment weight, and population size.

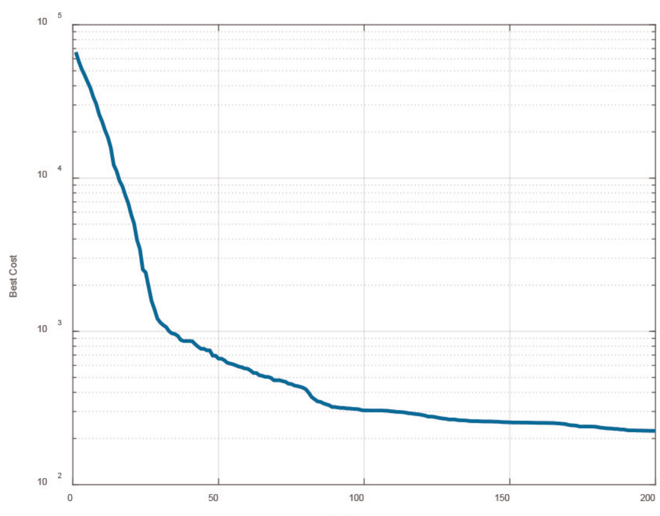

Figure 7.
Behavior of (D) objective function when solving by FA for: Firefly attraction coefficient base value = 2.25, iterations = 200, mutation coefficient damping ratio = 0.4, variation of population = 50.

PSO population size	Diet total glycemic load			FTG (mg)	UFTG (mg)
	min	mean	max		
20	59.6241	63.5302	68.5703	170.9513	22.2901
30	75.4830	85.8035	88.7775	170.4762	77.3999
40	68.5199	78.5372	80.0771	102.6863	36.4461
50	**70.8154**	**75.5430**	**80.1564**	**61.6584**	**19.1466**
60	83.5883	91.8249	98.7967	469.1408	184.4479
70	72.0223	81.4192	81.8198	170.7936	29.8207
80	69.5418	74.5632	80.3492	133.7875	37.3952

Table 6.
Diets produced by PSO for number of iterations = 200, self-adjustment weight = social-adjustment weight = 2, and variation of the population size.

PSO Adjustment Weight	Diet total glycemic load			FTG (mg)	UFTG (mg)
	min	mean	max		
1	89.4023	100.8713	111.2694	1.3158e+03	122.9172
1.1	87.1844	95.9035	98.1930	1.0457e+03	79.9678
1.2	74.1234	83.6180	84.1389	230.2951	482.5612
1.3	81.7219	91.7305	94.7066	117.2526	55.7460
1.4	81.9234	94.7501	101.9684	236.6845	45.4220
1.5	70.6824	82.5010	85.0437	551.0106	52.5807
1.6	77.2499	87.0125	89.6039	116.2240	61.8163
1.7	73.5639	79.0021	83.3030	66.6849	25.3209
1.8	58.8132	60.9824	62.9015	110.0112	57.2320
1.9	71.0009	73.7154	79.7176	146.4381	127.1389
2	**70.8154**	**75.5430**	**80.1564**	**61.6584**	**19.1466**

Table 7.
Diets produced by PSO for number of iterations = 200, variation of self-adjustment weight = SocialAdjustmentWeight, and population size =50.

The **Table 6** give TG, FTG, and UFTG of diets produced by FA for number of iterations equals to 200, self-adjustment weight = social-adjustment weight = 2, and population size variation between 20 and 80 particles.

The best diet is the one produced by PSO for population size of 50 with TG in [70.8154 80.1564], FTG = 61.6584 mg, and UFTG = 19.1466 mg. To investigate more improvements of this diets, we vary the Adjustment Weight coefficients in [1 2] will setting the population size to 50 (**Table 7**).

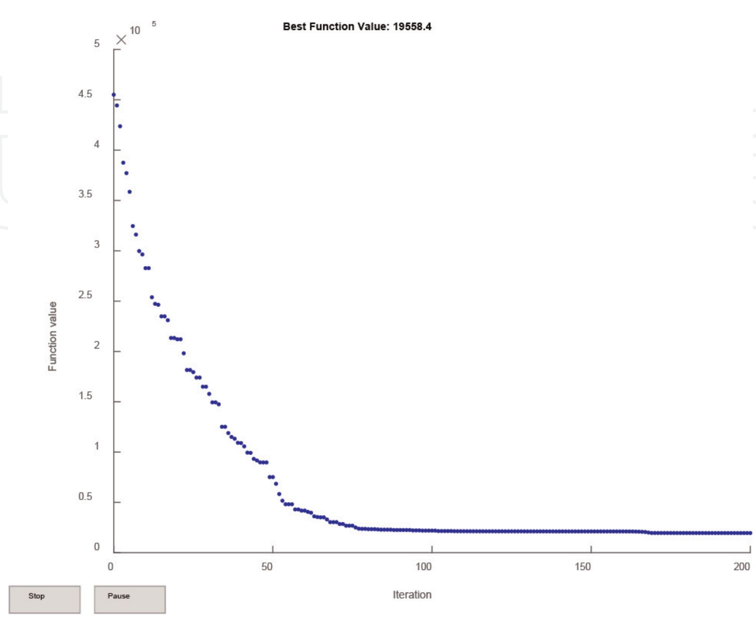

Figure 8.
The behavior of (D) objective function when solving the diet problem using PSO for number of iterations = 200, self-adjustment weight = social-adjustment weight = 2, and population size = 50.

Method	Parameters values	Diet total glycemic load			FTG (mg)	UFTG (mg)
		min	mean	max		
SFS	• Walk probability = 0.7 • Diffusion = 45 • Maximum diffusion = 10 • Number of iteration = 600	53.8780	60.5048	66.0715	50.1917	28.5891
FA	• Attraction coefficient base value = 2.25, • Iterations = 300, • Mutation coefficient damping ratio = 0.4, • Variation of population = 40	53.5439	54.9658	56.3875	10.6000	7.8365
PSO	• Iterations = 200, • Adjustement weight = 2, • Population size = 50	70.8154	75.5430	80.1564	61.6584	19.1466

Table 8.
Comparison between the diets produced by PSO, FA, and SFS.

Indeed, the quality of diets were improveded and the best one is obtained for PSO with iterations = 200, variation of self adjustment weight = social adjustment weight = 2, and population size =50; the Diet total glycemic load is in [70.8154 80.1564] and FTG = 61.6584 mg, and UFTG = 19.1466 mg, which meets the recommandations given in [24] . It should be noted that the first height diets are unacceptable.

The **Figure 8** illustrates the behavior of (D) objective function when solving the diet problem using PSO for self-adjustment weight = social-adjustment weight = 2, and population size =50. We remark that PSO was attracted very early to a very bad diet.

d. We compared the best diets produced by SFS, FA, and PSO based on the considered three criteria: TGL, FTG, and UFTG; see **Table 8**.

We remark that the best diet is the one produced by firefly algorithm for the configuration shown by the column 2 of the **Table 8** for a small number of iterations.

We can repeat all this study will consider additional quality measures such as the satiety rate and the applicability of the considered diets.

7. Conclusion

In this work, we used well-known swarm algorithms to solve the optimal diet problem based on the optimization mathematical model proposed recently in [5]. The inputs of our model were estimated based on 176 Morocco foods. Based on different paper search and the WHO's recommendations, we have estimated the daily nutrients requirements [6]. Different experimentations were realized for different configurations of the considered algorithms. Concerning SFS algorithm, we solved the problem (D) for different values of walk probability (0.7*), maximum diffusion (10*), and number of iteration (600*). Concerning FA algorithm, we solved the problem (D) for different values of attraction coefficient base value (2.25*), iterations (300*), mutation coefficient damping ratio (0.4*), and variation of population (40*). Concerning PSO, we solved the problem (D) for different values of Iterations (200*), adjustment weight (2*), and population size (50*). The best diets were produced by Firefly algorithm.

We can replicate that this entire investigation will consider further metrics of quality like satiety rate and feasibility of the examined diets.

In the future, we will propose a hybrid algorithm based on the SFS, FA, and PSO; this algorithm will be used to solve the diet problem and other well-known problems.

Acknowledgements

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

Author details

Karim El Moutaouakil[1*], Chellak Saliha[2], Baïzri Hicham[3] and Cheggour Mouna[4]

1 Faculty Polydisciplinary, Engineering Science Laboratory, USMBA, Taza, Morocco

2 MorphoSciences Research Laboratory of the Faculty of Medicine and Pharmacy of Marrakech, Université Cadi Ayyad, Marrakech, Morocco

3 Faculty of Medicine and Pharmacy Université Cadi Ayyad, Hicham Baïzri The Biosciences and Health Laborator, Marrakech, Morocco

4 FMPM, Cadi Ayyad University, Marrakech, Morocco

*Address all correspondence to: karim.elmoutaouakil@usmba.ac.ma

References

[1] Fukuoka Y, Komatsu J, Suarez L, Vittinghoff E, Haskell W, Noorishad T, et al. The mPED randomized controlled clinical trial: Applying mobile persuasive technologies to increase physical activity in sedentary women protocol. BMC Public Health. 2011;**11**(1):1-8

[2] Chen XC, Wei TR, Guo JH, Zhou WX, Song A, Chen WN, et al. August. Multiobjective evolutionary algorithm with double-level archives for nutritional dietary decision problem. In: 2019 9th International Conference on Information Science and Technology (ICIST). 2019. pp. 419-426

[3] Mortada S, Abou Najm M, Yassine A, El Fadel M, Alamiddine I. Towards sustainable water-food nexus: An optimization approach. Journal of Cleaner Production. 2018;**178**: 408-418

[4] Seligman HK, Laraia BA, Kushel MB. Food insecurity is associated with chronic disease among low-income NHANES participants. The Journal of Nutrition. 2010;**140**(2):304-310

[5] El Moutaouakil K, Cheggour M, Chellak S, et al. Metaheuristics optimization algorithm to an optimal Moroccan diet. In: 2021 7th Annual International Conference on Network and Information Systems for Computers (ICNISC). IEEE; 2021. pp. 364-368

[6] El Moutaouakil K, Ahourag A, Chellak S, Baïzri H, Cheggour M. Fuzzy deep daily nutrients requirements representation. Revue d'Intelligence Artificielle. 2022;**36**(2):263-269. DOI: 10.18280/ria.360210

[7] Stigler GJ. The cost of subsistence. Journal of Farm Economics. 1945;**27**(2): 303-314

[8] Orešković P, Kljusurić JG, Šatalić Z. Computer-generated vegan menus: The importance of food composition database choice. Journal of Food Composition and Analysis. 2015;**37**: 112-118

[9] Masset G, Monsivais P, Maillot M, Darmon N, Drewnowski A. Diet optimization methods can help translate dietary guidelines into a cancer prevention food plan. The Journal of Nutrition. 2009;**139**(8):1541-1548

[10] Briend A, Darmon N, Ferguson E, Erhardt JG. Linear programming: A mathematical tool for analyzing and optimizing children's diets during the complementary feeding period. Journal of Pediatric Gastroenterology and Nutrition. 2003;**36**(1): 12-22

[11] Seljak BK. June. Dietary menu planning using an evolutionary method. In: 2006 International Conference on Intelligent Engineering Systems. 2006. pp. 108-113

[12] Donati M, Menozzi D, Zighetti C, Rosi A, Zinetti A, Scazzina F. Towards a sustainable diet combining economic, environmental and nutritional objectives. Appetite. 2016;**106**:48-57

[13] Van Mierlo K, Rohmer S, Gerdessen JC. A model for composing meat replacers: Reducing the environmental impact of our food consumption pattern while retaining its nutritional value. Journal of Cleaner Production. 2017;**165**: 930-950

[14] Bello P, Gallardo P, Pradenas L, Ferland JA, Parada V. Best compromise nutritional menus for childhood obesity. PLoS One. 2020;**15**(1):0216516

[15] Bas E. A robust optimization approach to diet problem with overall glycemic load as objective function. Applied Mathematical Modelling. 2014; **38**(19–20):4926-4940

[16] You A. Dietary Guidelines for Americans. USA: US Department of Health and Human Services and US Department of Agriculture; 2015

[17] Yang XS. Nature-Inspired Metaheuristic Algorithms. Luniver Press; 2010

[18] Yang XS. Firefly algorithms for multimodal optimisation. In: Watanabe O, Zeugmann T, editors. Proc. 5th Symposium on Stochastic Algorithms, Foundations and Applications. Vol. 5792. Lecture Notes in Computer Science; 2009. pp. 169-178

[19] dos Santos Coelho L, de Andrade Bernert DL, Mariani VC. A chaotic firefly algorithm applied to reliability-redundancy optimisation. In: 2011 IEEE Congress on Evolutionary Computation (CEC'11). 2011. pp. 517-521

[20] Fister J, Fister I, Brest J, Yang XS. Memetic firefly algorithm for combinatorial optimisation. In: Filipič B, Silc J, editors. Bioinspired Optimisation Methods and their Applications (BIOMA2012). Bohinj, Slovenia; 2012. pp. 75-86

[21] Hassanzadeh T, Vojodi H, Moghadam AME. An image segmentation approach based on maximum variance intra-cluster method and firefly algorithm. In: Proc. of 7th Int. Conf. On Natural Computation (ICNC2011). 2011. pp. 1817-1821

[22] Salimi H. Stochastic fractal search: A powerful metaheuristic algorithm. Knowledge-Based Systems. 2015;**75**: 1-18

[23] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks. Vol. 4. 1995. pp. 1942-1948

[24] Eberhart RC, Shi Y, Kennedy J. Swarm Intelligence. Burlington, MA: Morgan Kaufmann; 2001

[25] Humayun MA, Elango R, Ball RO, Pencharz PB. Reevaluation of the protein requirement in young men with the indicator amino acid oxidation technique. The American Journal of Clinical Nutrition. 2007;**86**(4):995-1002

[26] Verma H, Garg R. Effect of magnesium supplementation on type 2 diabetes associated cardiovascular risk factors: A systematic review and meta-analysis. Journal of Human Nutrition and Dietetics. 2017;**30**(5):621-633