IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 24, 2017, accepted December 20, 2017, date of publication January 15, 2018, date of current version March 12, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2794198

Bacterial Foraging Optimization Algorithm for Menu Planning

BETANIA HERNÁNDEZ-OCAÑA[1], OSCAR CHÁVEZ-BOSQUEZ[1], JOSÉ HERNÁNDEZ-TORRUCO[1], JUANA CANUL-REICH[1], AND PILAR POZOS-PARRA[2]

[1]División Académica de Informática y Sistemas, Universidad Juárez Autónoma de Tabasco, Cunduacán 86690, México
[2]Centro de Innovación Social de Alto Impacto, Instituto Tecnológico de Estudios Superiores de Occidente, Tlaquepaque 45640, México

Corresponding author: José Hernández-Torruco (jose.hernandezt@ujat.mx).

ABSTRACT We propose a solution to the menu planning problem adapting the bacterial foraging-based optimization algorithm. We design a mathematical model which satisfies the nutritional needs of individuals while enforcing the ‘‘Laws of Nutrition’’, a set of postulates used by virtually all nutrition scientists in Latin America. This model corresponds to a constrained numerical optimization problem. We design a menu generator software prototype to set up a custom menu for 15 users with different characteristics. Results obtained by our proposal were satisfactory from an expert’s point of view.

INDEX TERMS Metaheuristics, optimization, diet planning, healthy menus, mathematical modeling.

I. INTRODUCTION

Currently, metaheuristic algorithms are a popular choice to solve complex optimization problems [40]. Among these, Evolutionary Algorithms (EAs) are based on emulating the process of natural evolution and the survival of the fittest, so that complex search spaces can be sampled [12]. On the other hand, in the middle of the 90’s a new group of metaheuristics emerged inspired by phenomena found in nature. These algorithms are grouped in an area called Collective Intelligence (CI), which base their functioning on social and cooperative behaviors of simple organisms such as insects, birds, and bacteria [14].

In 2002 Passino proposed the Bacterial Foraging Optimization Algorithm (BFOA) [32], a CI algorithm that arises from the initial ideas of Bremermann [3]. In BFOA each bacterium tries to maximize its energy obtained per unit of time, which is in the process of foraging, where it also evades harmful substances. Moreover, bacteria can communicate with each other through the segregation of substances. BFOA has four main processes: (1) chemotaxis (swim-tumble), (2) swarming, (3) reproduction and (4) elimination-dispersion. Bacteria are potential solutions to the problem, and their location represents the values of the decision variables of the problem. Bacteria can move (generate new solutions) through the chemotaxis cycle. Also, a motion is generated by the attraction of solutions in promising areas of the search space, the reproduction of the best solutions is allowed, and finally, bacteria in the cluster located in areas of low quality are eliminated. In 2009 a simplified version of the BFOA called Modified BFOA (MBFOA) was proposed [29] as a solution to constrained global optimization problems. MBFOA has a mechanism for handling constraints based on feasibility rules [9] on a decrease in the number of parameters with respect to those of the original BFOA. In addition, the MBFOA was applied to a set of design problems in the areas of chemical and mechanical engineering obtaining competitive results [29].

The MBFOA was applied to the resolution of a biobjective mechanical design problem in the presence of constraints in [30]. A more recent proposal based on MBFOA is Two Swim MBFOA (TS-MBFOA), which uses a mutation operator (similar to EAs) in the role of swimming operator within the chemotaxis process, in addition to a random swim in this same process and a skew mechanism for the initial population based on the range of decision variables of the problem [18].

Menu planning is a well-known optimization problem broadly identified as the process of organizing foods into meals according to specific constraints. Each element of the menu is quantified in rations, grams or milliliters. It allows obtaining a global quantification of the calories and nutrients, which must satisfy the user requirements to provide the body with the energy and nutrients in amounts enough to carry out the daily activities and to maintain the vital functions [26]. Menu planning is NP-Complete, as it can be reduced to a multidimensional knapsack problem (MDKP) [17].

There can be many constraints in menu planning, and these vary depending on the country, the user needs, the dietitian recommendation and so on. Then, to use a solid basis on nutrition, we rely on the called "Laws of Nutrition" or "Laws of Feeding" [10]. These rules were proposed by Pedro Escudero, which is considered the "father of nutrition" in Latin America [24], [36]. A formal model of the laws depicts each law as [5]:

- Law of Quantity: to take the amount of energy the body needs.
- Law of Quality: foods belonging to all food groups must be consumed.
- Law of Harmony: to keep a relationship between ingested nutrients.
- Law of Adequacy: adapt the diet to the nutritional, social and psychological needs of the user.

There are proposed solutions since the 60s of last century using mathematical programming [1], [11]. Research in the field include solutions using linear programming [23], [41], expert systems [4], [13], [35], EAs [16], [39], and more recently, CI techniques [6], [15]. Despite all of these proposals, as far as we know, there is no approach based on BFOA or MBFOA solving the menu planning problem in the literature, nor solutions based on the Laws of Nutrition. Moreover, the approaches are proposed for specific cases, for example in [6] the recommended nutritional values are proposed by a nutritionist for a specific user.

We propose to implement TS-MBFOA to generate healthy menus. We design a mathematical model of the menu planning problem resulting on a Constraint Numerical Optimization Problem (CNOP), based on the Laws of Nutrition. A CNOP can be defined as

$$\begin{aligned} &\text{Minimize } f(\vec{x}) \\ &\text{subject to: } g_i(\vec{x}) \leq 0, \quad i = 1, \ldots, m \\ &\qquad\qquad\quad h_j(\vec{x}) = 0, \quad j = 1, \ldots, p \end{aligned}$$

where $\vec{x} = [x_1, x_2, \ldots, x_n] \in R^n$ is the solution vector and each decision variable x_k, $k = 1, \ldots, n$ is bounded by lower and upper limits $L_k \leq x_k \leq U_k$, which define the search space S; m is the number of inequality constraints, and p is the number of equality constraints (in both cases, the constraints can be linear or nonlinear). If F denotes the feasible region, then it must be clear that $F \subseteq S$. As it is commonly found in the specialized literature of nature-inspired algorithms to solve CNOPs [7], [27], [28] equality constraints are transformed into inequality constraints by using a small tolerance $\varepsilon > 0$ as follows: $|h_j(\vec{x})| - \varepsilon \leq 0, j = 1, \ldots, p$.

The organization of this document is as follows. Section II details preliminary concepts used in this work. Section III defines the mathematical modeling of the problem. Section IV briefly describes TS-MBFOA. Section V presents the results of TS-MBFOA generating healthy menus. Finally, Section VI gives the conclusions and future work.

II. GENERAL SPECIFICATIONS

We consider the specifications about the target user, the foods considered and the menu characteristics before we model the problem.

TABLE 1. Nutritional information in the food database (Fruit group item example).

Key	Value
Index	6
Name	Jicama salad
Quantity	1
Measure	Cup
Grams/milliliters	100g
Ingredients	Jicama
Preparation	Cut the Jicama into strips and serve
Calories	126
Proteins	4.21
Lipids	3.01
Carbohydrates	20.5

TABLE 2. Number of foods per group in the food database.

Group	Number of foods
Vegetables	27
Fruits	48
Non fat Cereals	55
Cereals with fat	18
Legumes	27
Animal-derived foods for breakfast	47
Animal-derived foods in general	35
Breakfast beverages	27
Beverages in general	12
Side items	4

A. FOOD DATABASE

The food groups included in this study are: 1) vegetables, 2) fruits, 3) non-fat cereals, 4) cereals with fat, 5) legumes, 6) animal-derived foods for breakfast, 7) animal-derived foods in general, 8) breakfast beverages (only served at breakfast, may contain milk), 9) general beverages, and 10) side items (tortilla or bread roll). This last food group is mandatory in our model because tortilla and slices of bread are side items in virtually every meal. The tortilla is a Mexican culinary tradition since prehispanic times. We store foods data in a spreadsheet organized in 10 food groups according to their characteristics. For each food, the general and nutritional information is recorded, an example is shown in Table 1. Data was taken from the Mexican food composition tables [31].

Table 2 presents the number of foods in each group in the food database, containing mexican preparations, mainly. This database has a total of 300 foods and can be further increased by the user.

B. HEALTHY MENU SPECIFICATIONS

- A healthy menu will consist of at most 23 foods grouped into five meals: (1) breakfast, (2) morning snack, (3) lunch, (4) evening snack and (5) dinner.
- A snack is a small portion of food eaten in between breakfast and lunch or lunch and dinner. Snacks may be composed of cereal with fat or fruit or vegetables.
- Breakfast may consist of a breakfast drink (may contain milk) or non-fat cereal or fruits or vegetables or tortilla or animal-derived foods.

TABLE 3. Percentage added to the *BMR* according to physical activity.

Level of activity	Example	% of BMR
Repose	In bed	20%
Light	Office worker, professional	37.5%
Moderate	Housewife, student	55%
Intense	Worker, athlete	72.5%

- Lunch may include a general drink (no milk) or non-fat cereals or fruits or vegetables or tortilla or legumes or animal-derived foods.
- Dinner may consist of a general drink (no milk) or non-fat cereals or fruits or vegetables.

The menu must satisfy the constraints described in Section III.

C. USER SPECIFICATIONS

- The healthy menu and the nutritional parameters will be set up for people between 18 and 60 years old, with no overweight, malnutrition nor pathologies.
- The Basal Metabolic Rate (*BMR*) of an individual is calculated using the Harris-Benedict equation revised by Mifflin & St Jeor [38], which is stated by Equations 1 and 2 according to the user gender.

$$BMR_{female} = 10weight + 6.25height - 5age - 161 \quad (1)$$

$$BMR_{male} = 10weight + 6.25height - 5age + 5 \quad (2)$$

- The Thermic Effect of Food (*TEF*) consists of the energy expenditure needed to process food, which corresponds to 10% of *BMR* and must be added to it.
- Thermic Effect of Activity (*TEA*) is also added to the *BMR*, and the percentages vary depending on the physical activity of the individual according to Table 3 [34].
- Total Energy Expenditure (*TEE*) corresponds to the daily recommendation of calories for the user. TEF is added to BMR as well as TEA to compute TEE as shown in

$$TEE = BMR + TEF + TEA \quad (3)$$

III. MATHEMATICAL MODEL

We design an objective function subject to the Law of Quantity and the Law of Quality, among other useful constraints considering the remaining laws. Law of Quality is considered mandatory in our software prototype. It means to include foods from the five groups recommended by the "Plato del bien comer" ("eatwell plate") [8][1] in México. These food groups are fruits, vegetables, cereals, legumes and animal-derived foods.

This model represents an original formulation for the problem, as it formally represents general elements along with the Laws of Nutrition.

[1] http://www.promocion.salud.gob.mx/dgps/descargas1/programas/6_1 _plato_bien_c omer.pdf

TABLE 4. Caloric contribution of macronutrients.

Macronutrient	Calories
Carbohydrates	$4 kilocalories/gram$
Proteins	$4 kilocalories/gram$
Lipids	$9 kilocalories/gram$

A. OBJECTIVE FUNCTION (OBJ.F)

The objective function presented in Equation 4 seeks to minimize the difference between the number of calories required by an individual and the number of calories provided by the healthy menu found applying TS-MBFOA.

$$E = \left| \sum_{y=1}^{c} calFood_y(\vec{k}) - TEE \right| \quad (4)$$

In Equation 4, c is the number of meals in the menu. In this work the number of meals is always 5. $\vec{k}$ is a vector of integer values where each refers to an index $indexTgroup_t$, i.e., a food element from a given food group t where $t = 1, \ldots, 10$. The length of each $\vec{k}$ determines the number of foods allowed in each one of the five meals detailed in the healthy menu specifications (section II-B): $\vec{k}$ has six items in the breakfast, $\vec{k}$ has only three items in both snacks, $\vec{k}$ has seven items in the lunch and finally $\vec{k}$ has four items in the dinner. The range of each $indexTgroup_t$ is determined by the number of foods stored in a specific group of the database (Table 2). $calFood_y(\vec{k})$ are the calories of each food from the menu meals and *TEE* is computed using Equation 3 and the user specifications.

This objective function is subject to the constraints presented in the following section.

B. CONSTRAINTS

Constraint g_1 allows complying with the Law of Quantity (Equation 5) by the difference between the number of calories required by the individual and the number of calories provided by the healthy menu $\pm 10\%$. 100% adequacy in these calculations is technically impossible, a reason why a margin of error of $\pm$10-15% is commonly acceptable [19]. In this work, we use 10% as margin error:

$$g_1 : 0.90TEE \leq \sum_{y=1}^{c} calFood_y(\vec{k}) \leq 1.10TEE \quad (5)$$

Constraints g_2, g_3 and g_4 (Equations 6, 8, 10 respectively) allow complying with the Law of Harmony. It establishes the following daily caloric intake distribution: carbohydrates should cover between 55% and 60%, proteins between 10% and 15%, and lipids between 25% and 30% [31]. Carbohydrates, proteins, and lipids are denominated macronutrients, and from them, we compute the energy of foods. Table 4 shows the caloric contribution of these.

Constraints g_5 through g_9 (Equations 12 through 16 respectively) correspond to the energy distribution between meals.

Constraint g_2, carbohydrate intake, is represented by

$$g_2 : 0.90carbReq \leq \sum_{y=1}^{c} carbFood_y(\vec{k}) \leq 1.10carbReq \quad (6)$$

where $carbFood_y(\vec{k})$ is the total carbohydrates of the menu and $carbReq$ are the carbohydrates required by the individual according to Equation 7, based on the individual's TEE and the caloric contribution of carbohydrates (Table 4), $carb$ is a random variable in the range [55, 60], according to the daily caloric intake distribution:

$$carbReq = \frac{carb}{100} \times \frac{TEE}{4} \quad (7)$$

Constraint g_3 states the protein intake:

$$g_3 : 0.90protReq \leq \sum_{y=1}^{c} protFood_y(\vec{k}) \leq 1.10protReq \quad (8)$$

where $protFood_y(\vec{k})$ is the total proteins of the menu and $protReq$ is the total proteins required by the individual according to Equation 9, based on the individual's TEE and the caloric contribution of proteins specified in Table 4, $prot$ is a random variable in the range [10, 15], according to the daily caloric intake distribution:

$$protReq = \frac{prot}{100} \times \frac{TEE}{4} \quad (9)$$

Constraint g_4 validates the lipids intake with

$$g_4 : 0.90lipReq \leq \sum_{y=1}^{c} lipFood_y(\vec{k}) \leq 1.10lipReq \quad (10)$$

where $lipFood_y(\vec{k})$ is the total lipids of the menu and $lipReq$ is the total lipids required by the individual according to Equation 11, based on the individual's TEE and the caloric contribution of lipids (Table 4). lip is a random variable in the range [25, 30], according to the daily caloric intake distribution:

$$lipReq = \frac{lip}{100} \times \frac{TEE}{9} \quad (11)$$

Constraint g_5 corresponds to the daily energy requirement that an individual should take during breakfast. In this paper, the percentage established for breakfast is 25% of TEE ($pBreakfast = 0.25TEE$). This constraint is calculated with the Inequality 12, where n is the number of breakfast foods (6 foods from different food groups), $calFood_1(\vec{k})$ corresponds to the calories of breakfast foods (meal 1 according to our specifications).

$$g_5 : 0.90pBreakfast \leq \sum_{k=1}^{n} calFood_1(\vec{k}) \leq 1.10pBreakfast \quad (12)$$

Constraint g_6 corresponds to the daily energy requirement to be consumed by an individual during lunch. The percentage set for lunch is 35% of TEE ($pLunch = 0.35TEE$), and the constraint is calculated with the Inequality 13. The number of foods (n) at lunch is 7 and $calFood_3(\vec{k})$ corresponds to the calories of lunch foods (meal 3):

$$g_6 : 0.90pLunch \leq \sum_{k=1}^{n} calFood_3(\vec{k}) \leq 1.10pLunch \quad (13)$$

Constraint g_7 corresponds to the daily energy requirement to be consumed by an individual during the dinner. The percentage set for dinner is 20% of TEE ($pDinner = 0.20TEE$), and the constraint is calculated with the Inequality 14. The number of foods (n) at lunch is 4, and $calFood_5(\vec{k})$ corresponds to the calories of dinner foods (meal 5):

$$g_7 : 0.90pDinner \leq \sum_{k=1}^{n} calFood_5(\vec{k}) \leq 1.10pDinner \quad (14)$$

Finally, constraints g_8 and g_9 correspond to the daily energy requirement that an individual should take during snacks. In this paper, the percentage set for each snack is 10% of TEE ($pSnacks = 0.10TEE$). Both constraints apply in a similar way to morning and evening snacks. Inequality 15 corresponds to the morning snack (meal 2), and Inequality 16 corresponds to the evening snack (meal 4):

$$g_8 : 0.90pSnacks \leq \sum_{k=1}^{n} calFood_2(\vec{k}) \leq 1.10pSnacks \quad (15)$$

$$g_9 : 0.90pSnacks \leq \sum_{k=1}^{n} calFood_4(\vec{k}) \leq 1.10pSnacks \quad (16)$$

The number of foods in each snack is 3 and $calFood_2(\vec{k})$ and $calFood_4(\vec{k})$ corresponds to the calories of morning snack (meal 2), and evening snack (meal 4), respectively.

IV. TWO SWIM MODIFIED BACTERIAL FORAGING OPTIMIZATION ALGORITHM (TS-MBFOA)

Two Swim Modified Bacterial Foraging Optimization Algorithm (TS-MBFOA) is an improved version of MBFOA [29]. It characterizes two swims being intercalated in the chemotaxis process: the first swim features a random stepsize, and the second swim includes the mutation operator used in the evolutionary algorithms to improve the exploration and exploitation capacity of the algorithm. Although in the state of the art there are proposals based on BFOA that use the mutation operator within the search process [2], [20], [25], none of them include the mutation operator as a swimming mechanism. Main TS-MBFOA processes are described below. As all BFOA, TS-MBFOA complexity is exponential [33].

A. PRELIMINARIES

In TS-MBFOA a bacterium i represents a potential solution to the CNOP (i.e., an n-dimensional real-valued vector identified as $\vec{x}$), and it is defined as $\theta^i(j, G)$, where j is the chemotaxis loop index and G is the current generation of the algorithm.

B. SKEW MECHANISM FOR THE INITIAL SWARM

The initial swarm of bacteria S_b is created from three groups. The first group is comprised of bacteria randomly skewed towards the lower limit of the decision variables. The second group is comprised of bacteria randomly skewed towards the upper limit of the decision variables. Finally, a group of randomly located bacteria without skew, as in the original MBFOA, is used. The formulas to set the limits for the first and second group per variable are presented in

$$\left[L_i,\ L_i + \frac{U_i - Li}{ss}\right] \tag{17}$$

$$\left[U_i - \frac{U_i - Li}{ss},\ U_i\right] \tag{18}$$

where ss is the skew size ($ss > 1$), small values decrease the skew effect (when $ss \rightarrow 1$), increase otherwise. The aim of this skew in the initial swarm combined with the two swim operators and the random stepsize control is twofold: 1) to avoid the swarm of bacteria to converge prematurely, and 2) to improve the exploration and exploitation of the search space in the initial phase of the search. Premature convergence is a behavior observed in the original MBFOA caused by swarming and reproduction processes, along with the fixed step size. This process corresponds to line 1 of Algorithm 1.

C. CHEMOTAXIS

Preprocessing involve a bacteria evaluation before the chemotaxis process starts (line 2 of Algorithm 1).

Two swims are interleaved in this process (lines 5 to 7 of Algorithm 1), in each generation either the exploitation swim or exploration swim is performed. The process starts with the exploitation swim (classical swim), and next moves not necessarily interleave exploration and exploitation swims, because if the new position of a given swim, $\theta^i(j+1, G)$ has better fitness (based on the feasibility rules) than that of the original position $\theta^i(j, G)$, then another similar swim in the same direction will be carried out in the next generation. Otherwise, a new tumble for the other swim will be computed. The process stops after N_c attempts are completed.

The exploration swim uses the mutation between bacteria and is computed as indicated in

$$\theta^i(j+1, G) = \theta^i(j, G) + (\beta - 1)(\theta_1^r(j, G) - \theta_2^r(j, G)) \tag{19}$$

where $\theta_1^r(j, G)$ and $\theta_2^r(j, G)$ are two different randomly selected bacteria from the swarm; β is a user-defined parameter in the swarming process that defines how close a bacterium is located relative to the position of the best bacterium. In this operator, $\beta - 1$ is a positive control parameter for scaling the different vectors into (0,1], i.e., and it scales the area where a bacterium can move. Figure 1 depicts this process.

The exploitation swim is computed according to

$$\theta^i(j+1, G) = \theta^i(j, G) + C(i, G)\phi(i) \tag{20}$$

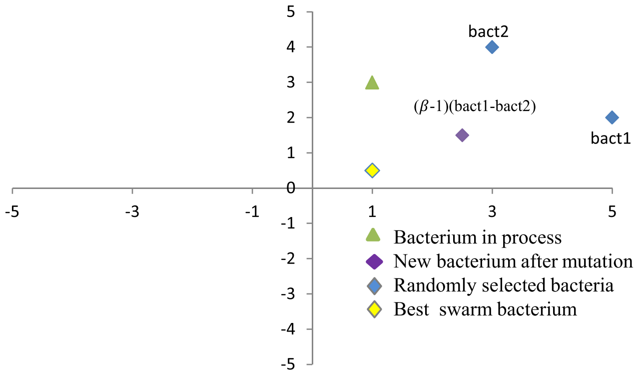

FIGURE 1. Exploration swim exemplification.

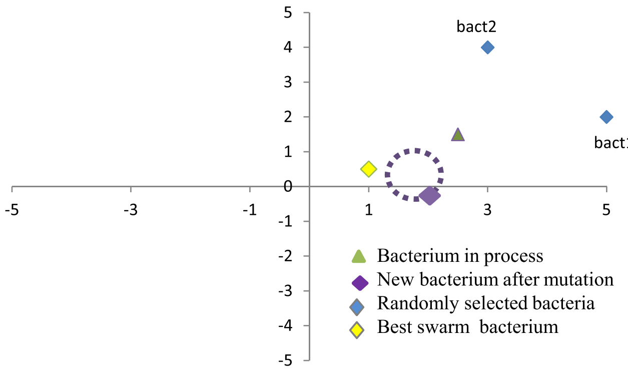

FIGURE 2. Exploitation swim exemplification.

where $\phi(i)$ is computed with the original BFOA (Equation 22) and $C(i, G)$ is a uniformly distributed random vector of size n with elements within the range of the decision variables limits. This vector determinates the step size of each bacterium updated in each generation with (line 13 of Algorithm 1)

$$C(i, G) = R * \Theta(i) \tag{21}$$

where $\Theta(i)$ is a randomly generated vector of size n with elements within the range of each decision variable: $[Upper_k, Lower_k]$, with $k = 1, \ldots .n$; and R is a user-defined parameter to scale the stepsize (this value should be close to zero, e.g. $5.00E-04$). The initial $C(i, 0)$ is generated using $\Theta(i)$. This random step size allows the bacteria to move in different directions of the search space and avoids the premature convergence as suggested in [22].

The following equation represents the original BFOA:

$$\phi(i) = \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{22}$$

where $\Delta(i)$ is a uniformly distributed random vector of size n with elements within the range $[-1, 1]$. Figure 2 depicts this process.

It is important to remark that the exploration swim (Equation 19) performs larger moves due to the mutation operator using randomly-selected bacteria. On the other hand, the exploitation swim (Equation 20) generates small moves using the random step size in the search process.

D. SWARMING

At the half number of iterations of the chemotaxis process, the swarming operator is applied with the Equation 23, where β is a user-defined positive parameter in the range (1, 2] (part of line 6 of Algorithm 1). However, in this proposal, and unlike MBFOA, when a solution violates the boundary of decision variables, a new solution x_i is randomly generated which is bounded by lower and upper limits $L_i \leq x_i \leq U_i$.

$$\theta^i(j+1, G) = \theta^i(j, G) + \beta(\theta^B(G) - \theta^i(j, G)) \tag{23}$$

where $\theta^i(j+1, G)$ is the new position of bacterium i; $\theta^i(j, G)$ is the current position of bacterium i; $\theta^B(G)$ is the current position of the best bacterium in the swarm so far at generation G; and β defines the closeness of the new position of bacterium i relative to the position of the best bacterium $\theta^B(G)$. The attractor move applies twice in a chemotaxis loop, while in the remaining steps the tumble-swim move is carried out.

E. REPRODUCTION

Reproduction takes place only at specific iterations of the algorithm (defined by the *RepCycle* parameter) deleting the S_r worst bacteria and duplicating the remaining $S_b - S_r$ ones. This process corresponds to lines 9 to 11 of Algorithm 1.

F. ELIMINATION-DISPERSAL

The worst bacterium $\theta^w(j, G)$ is removed according to the feasible rules, and a new bacterium is randomly generated (line 12 of Algorithm 1).

Figure 3 outlines the process performed by BFOAM. The corresponding algorithm of TS-MBFOA is introduced in Algorithm 1 along with the user-defined parameters.

Algorithm 1: Input Parameters Are Number of Bacteria S_b, Chemotaxis Loop Limit N_c, Number of Bacteria for Reproduction S_r, Scaling Factor β for Swarming, R Scaling Factor for Stepsize, Number of Reproduction Cycles *RepCycle* and Number of Generations *GMAX*

1 Create an initial swarm of bacteria $\theta^i(j, 0)\ \forall i$, $i = 1, \ldots, S_b$ by using the skew mechanism
2 Evaluate $f(\theta^i(j, 0))\ \forall i$, $i = 1, \ldots, S_b$
3 **for** *G=1 to GMAX* **do**
4   **for** *i=1 to* S_b **do**
5   **for** *j=1 to* N_c **do**
6   Perform the chemotaxis process by interleaving swims with Eqs. 19 and 20 and the attractor operator with Eq. 23 using β for bacteria $\theta^i(j, G)$
7   **end**
8   **end**
9   **if** *G mod RepCycle == 0* **then**
10   Perform the reproduction process by sorting the swarm based on the feasibility rules and deleting the S_r worst bacteria and duplicating the remaining $S_b - S_r$ ones
11   **end**
12   Perform the **elimination-dispersal process** by eliminating the $\theta^w(j, G)$ worst bacterium in the current swarm
13   **Update the step size vector** by using Eq. 21
14 **end**

V. TESTS AND RESULTS

A. TEST USERS

TS-MBFOA was implemented to generate healthy menus for 15 different users. Characteristics of each user are shown in Table 5.

TABLE 5. Data of the 15 test users.

Individual	Age	Weight (kg)	Height (m)	Gender	Physical activity level
1	21	60.5	1.46	Female	Moderate
2	21	98.0	1.68	Female	Light
3	28	78.0	1.56	Female	Moderate
4	24	61.5	1.52	Female	Intense
5	23	63.5	1.63	Male	Moderate
6	22	69.0	1.59	Female	Intense
7	19	119.0	1.72	Male	Light
8	20	67.0	1.54	Female	Moderate
9	20	61.5	1.59	Female	Intense
10	23	46.5	1.51	Female	Moderate
11	20	55.5	1.57	Female	Moderate
12	22	61.5	1.53	Female	Light
13	22	84.0	1.75	Male	Intense
14	22	64.0	1.54	Female	Intense
15	21	51.0	1.53	Female	Light

B. TS-MBFOA FOR MENU PLANNING

TS-MBFOA implementation was relatively straightforward. In this work, a bacterium $\vec{k}$ is a vector with at most 23 items where each one is a food element on the menu. This vector is composed of five subsets, each representing one of the five menu meals (breakfast, morning snack, lunch, evening snack, and dinner) including their respective foods. A solution to this problem (a bacterium) is represented as follows:

$$\begin{aligned}\theta^i(j, G) = \big\{&\{18, 48, 25, 19, 2, 47\}, \{17, 43, 23\},\\ &\{4, 39, 19, 19, 28, 2, 47\},\\ &\{13, 6, 27\}, \{12, 40, 26, 31\}\big\}\end{aligned}$$

where $\theta^i(j, G)$ is a bacterium i of the population in the chemotaxis process j in the generation G.

The default parameters of the TS-MBFOA based menu generator are shown in Table 6, which is the best combination of settings after a set of empirical tests. *Max_Evals* represents the number of solutions (obtained by the algorithm) that are evaluated in the objective function. *GMAX* is the maximum number of generations in the algorithm to gradually reach *Max_Evals*. It is important to mention that the execution time (response time) of the algorithm is relative to *Max_Evals*, i.e., a larger number of *Max_Evals* implies longer

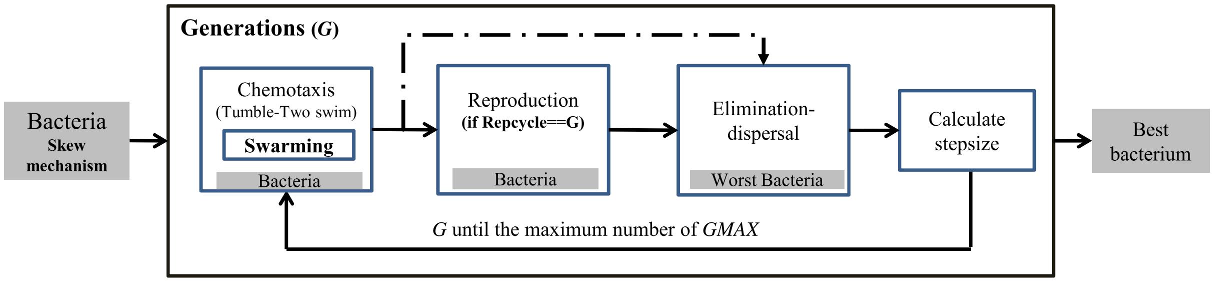

FIGURE 3. BFOAM full process.

TABLE 6. Parameters and values of TS-MBFOA.

Parameter	Variable	values
Bacteria	S_b	60
Limit of the chemotaxis cycle	N_c	10
Stepsize	R	0.05
Scaling factor	β	1.75
Bacteria to reproduce	S_r	5
Reproduction frecuency	$RepCyle$	100
Bacteria to remove	S_e	5
Evaluations	Max_Evals	80,000
Generations	$GMAX$	Value to reach Max_Evals

execution time and a major precision improvement in the results.

C. MENU GENERATOR PROTOTYPE

An interface for the TS-MBFOA was designed in Matlab's Graphical User Interface [43], to visualize the output of the algorithm. There exist many tools solving the menu planning problem [21], [37], [42], but these frameworks do not rely on the Laws of Nutrition.

The Menu generator prototype has an intuitive interface so any individual with basic computer skills can use it. It only requires the user data as input to work. We use Matlab R2009b, and all test were performed on a PC with a Core 2 Duo 3.5 processor, 4GB RAM, and 64 bit Windows 7 Operating System.

D. RESULTS

We executed the Menu generator prototype once per user using default parameters for TS-MBFOA. According to results shown in Table 7, all generated menus are within the $\pm 10\%$ margin error. TS-MBFOA found healthy menus that satisfy all the modeled constraints in 4 of the 15 users. For the rest of users, TS-MBFOA found healthy menus with a minimum Constraint Violation Sum (CVS), which is close to zero (second last column in Table 7). This value corresponds mainly to constraint g_4, lipids, in each of the menu meals (Figure 4).

The time required by the menu generator was different throughout all 15 cases, but the average run-time was 2.24 minutes (Figure 5).

Figure 6 shows the menu generated by TS-MBFOA with user 3 data and menu details. The user of the menu generator can visualize the convergence of the TS-MBFOA algorithm and adjust the parameters of this algorithm to create a new menu for the same user, as shown in Figure 7.

Table 8 shows a menu generated by TS-MBFOA with user 3 data. Our software prototype displays the particular details of each personalized menu generated.

TABLE 7. Results of menu planning using TS-MBFOA.

Indi-vidual	Lipids %	Proteins %	Carbohydrates %	TEE	TS-BFOA cal.	Obj.F.	CVS	Time (min.)
1	29	12	59	1892	1852	40	1.46	3.3
2	28	14	58	2328	2298.25	29.75	2.95	2.1
3	27	14	59	2191	2186.25	4.75	0	2.16
4	29	12	59	2189	2131.5	57.5	2.36	3
5	29	12	59	2333	2293.8	39.25	0	1.8
6	28	13	59	2407	2257.8	149.25	15.96	2.3
7	28	13	59	2871	2772.8	98.25	5.65	1.8
8	28	13	59	2073	2003.3	69.75	0.024	1.9
9	28	14	58	2296	2215	81	0.64	2.1
10	28	13	59	1712	1644	68	6.86	1.8
11	29	14	57	1928	1888	40	0	1.8
12	29	14	57	1716	1724	8	1.33	1.8
13	27	14	59	3468	3375	93	11.35	2.1
14	29	13	58	2525	2471.8	53.25	0	1.9
15	28	13	59	1584	1563	21	5.83	2.6

TABLE 8. A full menu generated by TS-MBFOA.

Food	Quality	Portion	Calories	Proteins	Lipids	Carbohydrates
Breakfast						
Strawberry smoothie	1 cup	–	200	9	8	24.5
Pasta with tuna	1 cup	210 gr	282.5	11	12.5	30
Cauliflower cooked	3/4 cup	94 gr	21	1.7	0.4	3.9
Raspberry	1 cup	123 gr	64	1.5	0.8	14.7
Morning snack						
Miniature cajeta bread	2 piece	26 gr	119	2.8	5.8	10.5
Raspberry	1 cup	123 gr	64	1.5	0.8	14.7
Cauliflower cooked	3/4 cup	94 gr	21	1.7	0.4	3.9
Lunch						
Water	1 cup	–	0	0	0	0
French fries	1 cup	210 gr	207.5	4	7.5	30
Cooked lima beans	1/2 cup	85 gr	94	6.5	0.3	16.7
Chicken with nopales	3/4 piece	100 gr	247.5	20.5	15	6
Corn tortilla	3 piece	100 gr	180	3.6	1.2	39.6
Dehydrated apple	9 slices	23 gr	55	0.2	0.1	15
Evening snack						
Bisquet	1/2 piece	33gr	118	2	5.4	15.8
Seedless grape	1 cup	92 gr	55	0.6	0.3	15
Chopped raw carrot	1/2 cup	64 gr	26	0.6	0.2	4.3
Dinner						
Water	1 cup	-	0	0	0	0
Red rice	1 cup	195 gr	168.75	4.5	2.5	31
Raw gherkins	1/3 cup	80 gr	23	1.1	1	3.3
Prune	1 cup	100 gr	240	2.18	0.38	63.88
TEE	–	–	2186	74.98	62.58	342.78
TS-MBFOA	–	–	2191	76.685	65.73	323.1725

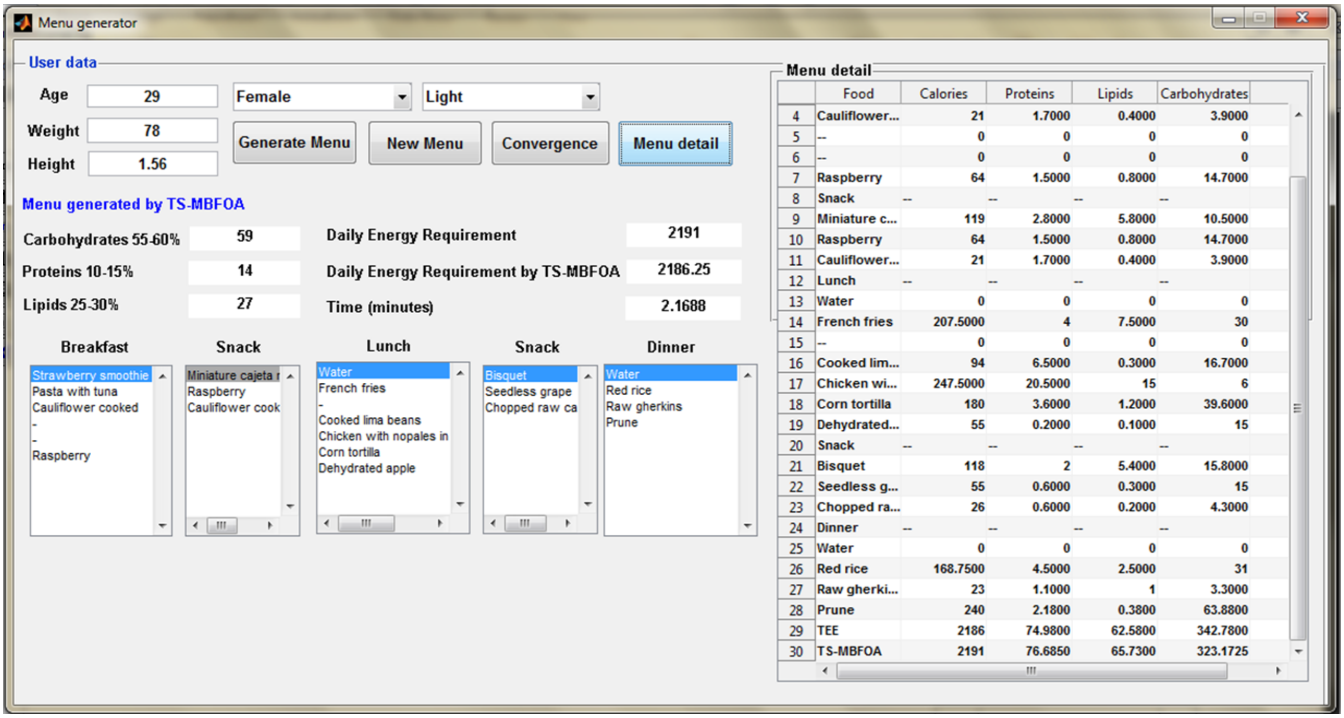

FIGURE 4. Constraint violations.

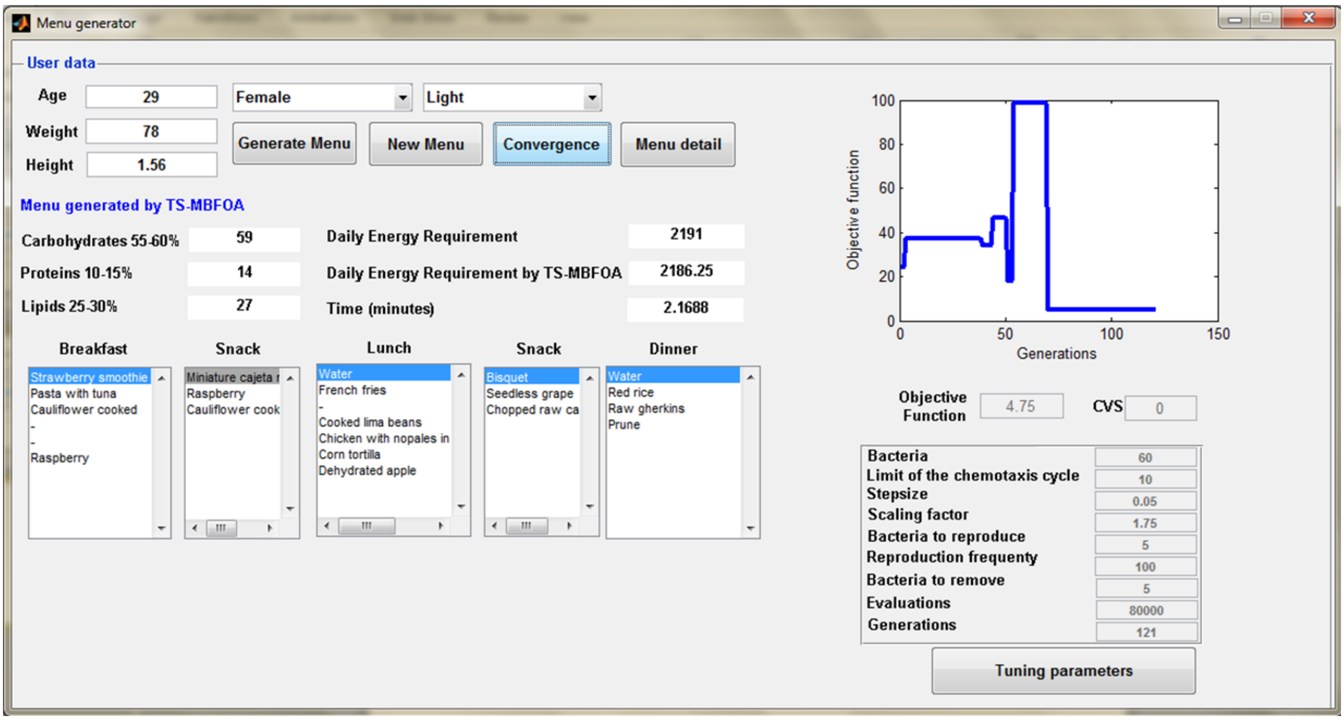

FIGURE 5. Menu generator prototype execution time.

Our results are compared against those from two proposals found in the specialized literature [16], [39], both using an evolutionary algorithm. It is crucial to mention that is difficult to make a fair comparison because each proposal defines its mathematical model, the number of cases to solve (users) is different and the number of days covered by the menu differs as well. Another important issue is the type of user: in some cases, menus are generated according to the needs of users

TABLE 9. TS-MBFOA compared against to NSGA-II menu planning.

Algorithm	Patient					Required calories	Results (Percentage of energy)			Calories by algorithm
	Gender	Characteristics	Weight	Height	Age		Lipids %	Protein %	Carbohydrates %	
TS-MBFOA	Female	Non-Pathology	61.2kg	164.9cm	28	2284	29	12	59	2223
NSGA-II	Female	Chronic kidney	61.2kg	164.9cm	N/A	2100	29	7.3	63.7	2013
TS-MBFOA	Male	Non-Pathology	70.2kg	1.77cm	35	2159	29	12	59	2162.5
NSGA-II	Male	Chronic kidney	70.2kg	177cm	N/A	2400	32	6.8	61.2	2373

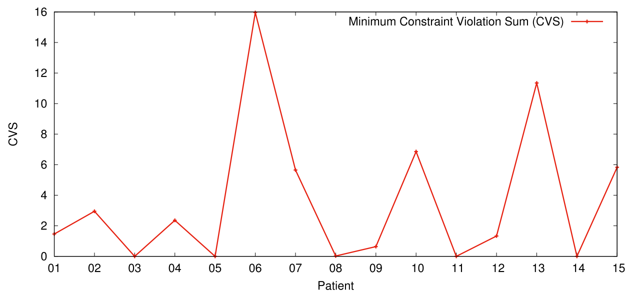

FIGURE 6. Example of a healthy menu generated, result and details.

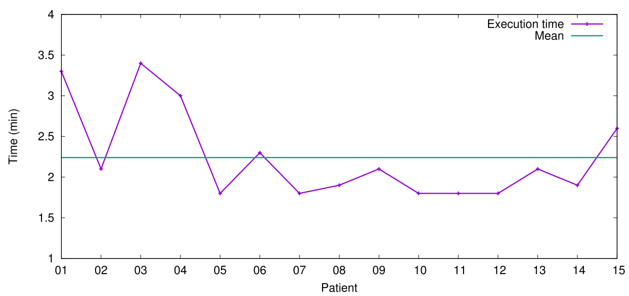

FIGURE 7. Example of TS-BFOA convergence results and configuration options.

who suffer chronic kidney, cardiovascular diseases or with no pathologies. Moreover, menus are generated using databases with foods and meals specific to the region or country of authors. Also, certain regional traditions bias the mathematical model.

According to the results of the proposal based on GA [16], so-called *MenuGene*, the automated menu generator is successful on generating menus with meal plans that satisfy all constraints for non-pathological users. Healthy users consider only one or two out of four constraints in their mathematical model. However, their proposal finds nearly optimal solutions for users with cardiovascular diseases, i.e., when considering three or four constraints. Unlike the TS-MBFOA, *MenuGene* needs an initialized population (a base menu corresponding to a feasible solution) to generate better menus during its iterations.

A more clear comparison can be carried out against the NSGA-II proposal [39]. It presents statistical results on the runs with two users. Table 9 presents some similar criteria to make a comparison. We experimented our TS-MBFOA with similar users as the NSGA-II, and both algorithms found competitive results for the test users. However, the menu of NSGA-II is specific for users with chronic kidney disease.

In general, and according to the opinion of a nutrition expert, the results of TS-MBFOA are competitive against the results an expert would get manually. Nevertheless, the generator has the advantage of providing a menu in order of minutes while a nutrition expert may take hours to generate a personalized menu for a single user. Besides, the food database is still limited and may lead to dissatisfaction of the constraint on the lipids of each meal on the menu.

VI. CONCLUSION AND FUTURE WORK

We adapted the Two Swim-Modified Bacterial Foraging Optimization Algorithm (TS-MBFOA) to generate healthy menus. This algorithm is part of the group of Collective Intelligence (CI) algorithms that simulate the collaborative behavior of certain simple species such as birds, fish, bees, ants, bacteria, which intelligently search for food or shelter. As far as we know, no approach based on this specific CI algorithm creating healthy menus had been reported in the literature.

We design a novel mathematical model of menu planning based on the definition of a Constraint Numerical Optimization Problem (CNOP) to tackle this problem computationally using TS-MBFOA. The model relies upon the Laws of Nutrition, widely used in Latin America by nutritionist and dietitians to plan diets. Also, we created a custom food database including local dishes, to test the mathematical model in TS-MBFOA.

We created a custom graphical interface for TS-MBFOA, called Menu generator prototype. We test our prototype on 15 different users resulting in satisfactory menus according to the point of view of a nutrition expert. It is important to mention that TS-MBFOA was executed only once per user. In numerical data, the algorithm obtained four fully compliant menus, that is, the menus satisfy the constraints of the problem as well as the objective function. The rest of the menus fill all but one constraints of the mathematical model, these menus carried a minimum amount of violation of constraints (*CVS*), mainly on the constraint corresponding to the control of the lipids in each menu meal (constraint g_4). This behavior may correspond to a lack of food elements with lipids as its main macronutrient.

TS-MBFOA's results are competitive when compared against those of a proposal called NSGA-II. Both algorithms generated healthy menus taking into account test user's characteristics. According to a nutrition specialist, solutions generated by TS-MBFOA may be similar to those made by an expert hand. Also, the Menu generator has the advantage of providing menus within minutes and allows regenerating them until the user gets satisfied.

Future work remains to be done: our food database may be enriched to get better menus, to make a comparison of the menus generated using our proposal against menus generated using non-commercial systems. We also need to adjust the parameters of the algorithm to obtain feasible results for any case type. We are also interested in finding the correlation between the user characteristics and the *CVS* of the generated menus, along with the execution time of our Menu generator.

REFERENCES

[1] J. L. Balintfy, "Menu planning by computer," *Commun. ACM*, vol. 7, no. 4, pp. 255–259, 1964.

[2] A. Biswas, S. Dasgupta, S. Das, and A. Abraham, "A synergy of differential evolution and bacterial foraging optimization for global optimization," *Neural Netw. World*, vol. 17, no. 6, pp. 607–626, 2007.

[3] H. Bremermann, "Chemotaxis and optimization," *J. Franklin Inst.*, vol. 297, no. 5, pp. 397–404, 1974.

[4] G. Brewka, I. Niemelä, and M. Truszczyński, "Answer set optimization," in *Proc. 18th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2003, pp. 867–872. Kaufmann, 2003.

[5] O. Chávez-Bosquez and P. Pozos-Parra, "The latin American laws of correct nutrition: Review, unified interpretation, model and tools," *Comput. Biol. Med.*, vol. 70, pp. 67–79, Mar. 2016.

[6] V. Chifu, R. Bonta, E. St Chifu, I. Salomie, and D. Moldovan, *Particle Swarm Optimization Based Method for Personalized Menu Recommendations*. Cham, Switzerland: Springer, 2017, pp. 232–237.

[7] C. A. Coello-Coello, "Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art," *Comput. Methods Appl. Mech. Eng.*, vol. 191, nos. 11–12, pp. 1245–1287, Jan. 2002.

[8] Secretará de Economía, "Norma Oficial Mexicana para la promoción y educación para la salud en materia alimentaria: Official mexican standard for health promotion and education in food matters)," (in Spanish), Ministry Economy, NOM-043.SSA2-2005, 2005.

[9] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, 2000.

[10] B. C. Deon, L. B. Medeiros, A. L. de Freitas Saccol, L. H. Hecktheuer, S. Saccol, and M. Naissinger, "Good food preparation practices in households: A review," *Trends Food Sci. Technol.*, vol. 39, no. 1, pp. 40–46, 2014.

[11] E. F. Eckstein, "Menu planning by computer: The random approach," *J. Amer. Dietetic Assoc.*, vol. 51, no. 6, pp. 529–533, 1967.

[12] A. Eiben and J. E. Smith, Eds., *Introduction to Evolutionary Computing* (Natural Computing Series). Berlin, Germany: Springer-Verlag, 2003.

[13] H. S. Ejtahed, M. M. Sarsharzadeh, P. Mirmiran, G. Asghari, E. Yuzbashian, and F. Azizi, "Leemoo, a dietary assessment and nutritional planning software, using fuzzy logic," *Int. J. Endocrinol. Metabolism*, vol. 11, no. 4, 2013, Art. no. e10169.

[14] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. Hoboken, NJ, USA: Wiley, 2005.

[15] D. Fister, S. Rauter, and I. Fister, Jr., "Generating eating plans for athletes using the particle swarm optimization," in *Proc. IEEE 17th Int. Symp. Comput. Intell. Inform. (CINTI)*, Nov. 2016, pp. 193–198.

[16] B. Gaál, I. Vassányi, and G. Kozmann, "A novel artificial intelligence method for weekly dietary menu planning," *Methods Inf. Med.*, vol. 44, no. 5, pp. 655–664, 2005.

[17] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, 1st ed. New York, NY, USA: Freeman, 1979.

[18] B. Hernández-Ocaña *et al.*, "Two-swim operators in the modified bacterial foraging algorithm for the optimal synthesis of four-bar mechanisms," *Comput. Intell. Neurosci.*, vol. 2016, Jan. 2016, Art. no. 4525294.

[19] F. O. F. Herrán, L. D. C. Quintero, and M. F. L. Ardila, "Fuentes y magnitud de la variación en la dieta de adultos de Bucaramanga," *Rev. Chilena Nutrición*, vol. 33, pp. 55–64, Apr. 2006.

[20] H. C. Huang, Y. H. Chen, and A. Abraham, "Optimized watermarking using swarm-based bacterial foraging," *Inf. Hiding Multimedia Signal Process.*, vol. 1, no. 1, pp. 51–58, 2010.

[21] N. Jothi, W. Husain, and N. Zakaria, "Application of knowledge-based system in personalized health management portal," *J. Public Health Frontier*, vol. 2, no. 4, pp. 193–200, 2013.

[22] A. Kasaiezadeh, A. Khajepour, and S. L. Waslander, "Spiral bacterial foraging optimization method: Algorithm, evaluation and convergence analysis," *Eng. Optim.*, vol. 46, no. 4, pp. 439–464, 2014.

[23] P. Leung, K. Wanitprapha, and L. A. Quinn, "A recipe-based, diet-planning modelling system," *Brit. J. Nutrition*, vol. 74, no. 2, pp. 151–162, 1995.

[24] UNC Health Sciences Library. *Pedro Escudero*. Accessed: May 26, 2016. [Online]. Available: http://hsl.lib.unc.edu/specialcollections/bios/escudero

[25] Y. Luo and Z. Chen, "Optimization for PID control parameters on hydraulic servo control system based on the novel compound evolutionary algorithm," in *Proc. 2nd Int. Conf. Comput. Modeling Simulation*, 2010, pp. 40–43.

[26] L. K. Mahan, S. Escott-Stump, and J. L. Raymond, *Krause's Food & the Nutrition Care Process* (Krause's Food & Nutrition Therapy), 13th ed. Readfield, ME, USA: Saunders, 2011.

[27] E. Mezura-Montes, Ed., *Constraint-Handling in Evolutionary Optimization* (Studies in Computational Intelligence), vol. 198. Berlin, Germany: Springer-Verlag, 2009.

[28] E. Mezura-Montes and C. A. Coello-Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm Evol. Comput.*, vol. 1, no. 4, pp. 173–194, 2011.

[29] E. Mezura-Montes *et al.*, "Modified bacterial foraging optimization for engineering design," in *Proc. Artif. Neural Netw. Eng. Conf. (ANNIE)*, vol. 19. St. Louis, MO, USA, Nov. 2009, pp. 357–364.

[30] E. Mezura-Montes, E. P. Flores, and B. H. Ocaña, "Optimization of a mechanical design problem with the modified bacterial foraging algorithm," in *Proc. Argentine Congr. Comput. Sci.*, Oct. 2011, pp. 171–180.

[31] M. Muñoz de Chávez *et al.*, "Miriam Muñoz de Chávez," in *Valor nutritivo de los alimentos de mayor consumo*, 2nd ed. New York, NY, USA: McGraw-Hill, 2010.

[32] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 52–67, Mar. 2002.

[33] K. M. Passino, "Bacterial foraging optimization," *Int. J. Swarm Intell. Res.*, vol. 1, no. 1, pp. 1–16, Jan. 2010.

[34] O. P. Perera, "Manual de lineamientos para la práctica de la nutrición clýnica," in *Spanish: Manual of Guidelines for the Practice of Clinical Nutrition*, 1st ed. New York, NY, USA: McGraw-Hill, 2005.

[35] G. J. Petot, C. Marling, and L. Sterling, "An artificial intelligence system for computer-assisted menu planning," *J. Amer. Dietetic Assoc.*, vol. 98, no. 9, pp. 1009–1014, 1998.

[36] R. E. Pite, *Creating a Common Table in 20th-Century Argentina*, 1st ed. Chapel Hill, NC, USA: Univ. North Carolina Press, 2013.

[37] T. Pizzuti, G. Mirabelli, M. A. Sanz-Bobi, and F. Goméz-Gonzaléz, "Food track & trace ontology for helping the food traceability control," *J. Food Eng.*, vol. 120, pp. 17–30, Jan. 2014.

[38] R. A. Roth, *Nutrition & Diet Therapy*, 10th ed. Mason, OH, USA: Cengage Learning, 2010.

[39] B. K. Seljak, "Computer-based dietary menu planning," *J. Food Composition Anal.*, vol. 22, no. 5, pp. 414–420, 2009.

[40] P. Siarry and Z. Michalewicz, Eds. *Advances in Metaheuristic Methods for Hard Optimization*. Berlin, Germany: Springer, 2008.

[41] D. Sklan and I. Dariel, "Diet planning for humans using mixed-integer linear programming," *Brit. J. Nutrition*, vol. 70, no. 1, pp. 27–35, 1993.

[42] C. Snae and M. Bruckner, "FOODS: A food-oriented ontology-driven system," in *Proc. 2nd IEEE Int. Conf. Digit. Ecosyst. Technol. (DEST)*, 2008, pp. 168–176.

[43] "MATLAB guide," The MathWorks, Inc., Natick, MA, USA, 2016. [Online]. Available: https://www.mathworks.com/discovery/matlab-gui.html

BETANIA HERNÁNDEZ-OCAÑA received the B.Sc. degree in computational systems from the Universidad Juárez Autónoma de Tabasco, México, in 2009, the M.Sc. degree in applied computing from the Laboratorio Nacional de Informática Avanzada, México, in 2012, and the Ph.D. degree in computer science from the Universidad Juárez Autónoma de Tabasco, México, in 2016. She is currently a Research Professor with the División Académica de Informática y Sistemas, Universidad Juárez Autónoma de Tabasco, México. Her areas of interest are related to swarm intelligence algorithms and evolutionary computation (Bacterial foraging, particle swarm optimization, differential evolution), global optimization, and constraint-handling.

OSCAR CHÁVEZ-BOSQUEZ received the Ph.D. degree in computer science from the Universidad Juárez Autónoma de Tabasco, México, involving in the belief merging and metaheuristics applications under the supervision of P. Pozos-Parra. He is currently an Associate Professor with UJAT, teaching courses mainly in artificial intelligence and software development, mainly, He has published a number of papers about solving real life problems via computational methods. He is a certified Software Engineer, holds a number of certifications in software development and tools, and he has developed open source software and open models. In 2016, he received membership of the National System of Researchers (Candidate) by CONACYT, México.

JOSÉ HERNÁNDEZ-TORRUCO received the Ph.D. degree in computer science from the Universidad Juárez Autónoma de Tabasco, (UJAT) México, involved in a descriptive and predictive model for Guillain-Barré syndrome based on clinical data using machine learning algorithms under the supervision of J. Canul-Reich. He is currently an Associate Professor with UJAT, teaching courses mainly in programming. He has directed a number of under graduate theses. He holds a number of certifications in software development and tools. His area of interest is machine learning and data mining, especially its application in medicine. He has published a number of articles in English and Spanish in refereed and indexed journals, and national and international conferences. He has been responsible for some institutional projects and has participated as a collaborator in funded projects.

JUANA CANUL-REICH received the Ph.D. degree in computer science and engineering from the University of South Florida in 2010. She is a Fulbrighter. She is currently a Faculty Member with the Academic Division of Informatics and Systems, Universidad Juárez Autónoma de Tabasco, México, where she is a Graduate Advisor of master and Ph.D. students. Her research interests include data mining and machine learning with the focus on microarray data analysis, medical data analysis, and seismic data analysis, and performance evaluation of algorithms. Mainly interested in problems of dimensionality reduction, classification, data exploration, preprocessing, and visualization. In 2015, she received the membership of the National System of Researchers level 1 (SNI 1) in México. She is an active reviewer for the *Journal of Ambient Intelligence and Smart Environments*, for Sinectica -a Mexican Journal in Education, and for diverse international conferences.

PILAR POZOS-PARRA received the B.S. degree in computer science from the Benemérita Universidad Autónoma de Puebla, México, and the M.S. and Ph.D. degrees in computer science (knowledge representation and formalization of reasoning) from SUPAERO College, France. She is currently a Researcher with the Centro de Innovación Social de Alto Impacto, Instituto Tecnológico de Estudios Superiores de Occidente, Tlaquepaque, Jalisco, México. Her research interests include knowledge representation and decision-making systems. She is currently studying nutrition science with the División Académica de Ciencias de la Salud (DACS), UJAT. Her research interests are DACS focuses on advancing nutrition treatments, clinical decision support and processes in nutrition evaluation. She is member of the National System of Researchers level 1 (SNI 1) in México.