Developing Tasty Calorie Restricted Diets Using a Differential Evolution Algorithm

Conference Paper *in* Lecture Notes in Computer Science · July 2016

DOI: 10.1007/978-3-319-42092-9_14

João Gabriel Rocha Silva
Federal University of Juiz de Fora

Iago Augusto Carvalho
University of Campinas

Michelli Marlane Silva Loureiro
Federal University of São João del-Rei

Vinicius da F. Vieira
Federal University of São João del-Rei

Developing tasty calorie restricted diets using a Differential Evolution algorithm

João Gabriel Rocha Silva[1], Iago Augusto Carvalho[2], Michelli Marlane Silva Loureiro[1], Vinícus da Fonseca Vieira[1], Carolina Ribeiro Xavier[1]

[1] Universidade Federal de São João del Rei, Brazil
[2] Universidade Federal de Minas Gerais, Brazil

Abstract. The classical diet problem seeks a diet that respects the indicated nutritional restrictions at a person with the minimal cost. This work presents a variation of this problem, that aims to minimize the number of ingested calories, instead of the financial cost. It aims to generate tasty and hypocaloric diets that also respect the indicated nutritional restrictions. In order to obtain a good diet, this work proposes a Mixed Integer Linear Programming formulation and a Differential Evolution algorithm that solves the proposed formulation. Computational experiments show that it is possible to obtain tasty diets constrained in the number of calories that respect the nutritional restrictions of a person.

Keywords: Differential evolution, diet problem, evolutionary algorithms, calories, obesity

1 Introduction

Metaheuristics are generic algorithms commonly used to solve optimization problems. Evolutionary Algorithms (EA) are metaheuristics based on the idea of the natural evolution of Darwin, that can be easily applied to problems that arise in several areas of the human knowledge. They are based on a set of solutions, called population. Each solution of the population is said to be an individual. The optimization process of EA are based on probabilistic transitions between two or more individuals of a population, instead the deterministic rule of classical optimization algorithms.

When solving linear or non-linear problems, one of the most prominent EA is the Differential Evolution algorithm (DE). In DE, as in others EA, the evolutionary process can be initiated from a population generated at random. At each generation, individuals suffer recombinations and mutations. Then, the most adapted individuals are selected to go to the next generation. The concept of population, mutation, and recombination turns DE into a fast, robust and stable algorithm, that can easily escape from local optima and achieve good solutions with a reasoning time.

DE can be easily adapted to a great number of optimization problems. One of the most classical optimization problem is the Diet Problem (DP), introduced by George Stigler in [17]. DP aims to build a diet for a 70 Kg man that respects the minimal quantity of nutrients recommended by the North American National Research Council in 1943, and has the minimum financial cost.

This work proposes a modified version of DP, denominated Caloric Restricted Diet Problem (CRDP). Instead the classic objective function that minimizes the diet finnancial cost, CRDP aims to build a diet that minimizes the number of ingested calories, while respecting the minimal quantity of nutrients. Moreover, this work is concerned to provide a tasty diet that can be easily adopted by a great number of individuals. We present a Mixed Integer Linear Programming (MILP) formulation for CRDP, and solve it through a DE algorithm. We also present how to develop a real life daily diet with low number of calories composed by six meals.

The remainder of this work is organized as follows. The background and motivation of this work is presented in Section 2. CRDP is formally defined in Section 3. A Differential Evolution algorithm for CRDP is presented in Section 4. Computational experiments are developed and discussed in Section 5. Finally, concluding remarks are drawn in the last section.

2 Background and motivation

2.1 The Diet Problem

DP is a classic linear optimization problem introduced in [17]. It aims to select a subset of foods, from a large set of 77 elements, a subset of they that should be eaten on a daily basis in order to produce healthy diet, taking in account nine different nutrients, and has the minimal cost [8]. Many variations of this problem can be found in Literature. As example, we can cite a fuzzy-logic DP [13], a multi-objective variant of DP [7], and a a robust optimization approach for DP [2].

Let F be the set of available foods. Moreover, let N be the set of nutrients considered at the problem. The LP formulation is defined with decision variables $p_i \geq 0$ that represent the amount of food $i \in F$ to be consumed. Moreover, c_i represents the cost per portion of food $i \in F$, m_{ij} represents the amount of nutrient $j \in N$ that is contained at food $i \in F$, and b_j is the minimal requirement of the nutrient $j \in N$. The corresponding LP formulation is defined by Equations (1)–(3).

$$min \sum_i c_i p_i \tag{1}$$

$$\sum_i m_{ij} p_i \geq b_j, \forall j \in \{0, \dots N\} \tag{2}$$

$$p_i \geq 0, \forall i \in \{0, \dots, F\} \tag{3}$$

The objective function (1) aims to minimize the cost of the aliments that are included on the diet. Inequalities (2) maintain all the necessary nutrients at a minimal level. Equation (3) defines the domain of the variable p.

The main objective of the classical DP is to minimize the financial cost of a diet that provides all necessary nutrients. However, DP is only focused on healthy individuals. It is not concerned with individuals that have overweight. Moreover, the original solution provided in [17] lacks of food variety and palatability, thus being hard to follow the proposed diet. As the number of individuals with overweight has increased in the past 20 years [23], the development of computational methods to generate diets that are healthy and tasty at the same time is needed.

2.2 Diet, calories, and heath

It is notable that a diet that contains a high amount of calories is correlated with weight gain, both in humans [21] and animals [11]. More than simple affecting the body weight, it can greatly affects the health of an individual, being responsible for a great number of chronic diseases [22].

It was shown that a balanced diet, together with physical exercises, can help an individual to lose weight and improve it's health [16]. However, it is not easy to develop a diet with low calories amount that provides all necessary daily basis nutrients, such as proteins, zinc, and iron, among others. Moreover, the palatability of a meal is affected by the amount of calories present at the consumed foods, such that high amount of calories is directly related to a tastier food [21]. Thus, one of the main challenges for nutritionists all around the world is to develop a diet restricted in the number of calories that provides the necessary nutrients, and that is, at the same time, healthy and tasty.

It is recommended for an individual to daily consume from 2000 to 2500 calories. Caloric restricted diets generally uses 800 kilocalories as a lower bound for the daily calories intake. In order to develop a healthy diet, this caloric restricted diet also need to offer all the nutrients necessary for an individual. Moreover, it need to provide tasty foods, such that it will be easier for an individual to follow the recommended meals.

2.3 Evolutionary Algorithms

There exists many EA variants and implementations. However, all of they rely on one basic idea, back from the natural selection theory: given an initial population, the selection is realized at each generation such that the most fitted individuals have a greater chance to survive from one generation to another. Thus, it is possible to achieve a population most adapted to its environment.

A pseudo-code of a general EA is presented at Algorithm 1. The initial population is generated, commonly with individuals generated at random, at line 1. Next, at line 2, each individual have it fitness evaluated. The evolutionary process occurs from line 4 to 10, while a given stopping criterion is not met. First, at line 5, some individuals are selected as parents. Next, the selected individuals are recombined in order to generate a new individual, at line 6. These new individuals go through a mutation process at line 7. Next, their fitness function are evaluated again at line 8. Finally, the most fitted individuals are selected to continue to the new generation at line 9.

```
1  begin
2      initialize the population with individual;
3      evaluate each individual;
4      while stopping criterion not met do
5          select the parents;
6          cross the selected parents;
7          mutate the individuals;
8          evaluate the offspring;
9          evolute the most fitted individuals to survive at next generation;
10     end
11 end
```

Algorithm 1: Pseudo-code of an general EA

2.4 Differential evolution

The Differential Evolution (DE) algorithm is an instance of an EA. It was first presented in [18], with the objective to solve the polynomial adjust problem of Chebychev. DE is widely applied to non-linear optimization problems. According to [3], the choose of DE despite other EA is based on the following facts.

- DE is efficient to solve problems with discontinuously functions, because it does not require information about the derivate of a function.
- DE can run properly with a small population.
- DE can easily escape from local optima, due its efficiently mutation operator.
- DE input and output parameters can be manipulated as floating point number without any additional manipulation.

As DE was developed to handle real parameters optimization, it can be easily adapted to a large number of areas. In Literature, it is possible to find DE applications in engineering [10], chemistry [1], biology [19], finances [9], among others [14, 5]. In order to apply DE to a problem, it is just necessary to model the input of the problem as an input for DE.

DE follows the five main evolution stages of an EA, as presented in Algorithm 1. However, in DE, the evolution stages occur in a different order. First, the parents are selected. Next, occurs the mutation, recombination and evaluation phases, in order. Finally, the most fitted individuals are selected to go to the next generation.

Initialization This stage initializes a population with $|X|$ individuals. Each individual $x_i \in X$ is a d-dimensional vector, such that each position of the vector (also called a gene) represent one characteristic of that individual. Figure 1 shows, for an arbitrary problem, a representation of an individual with 6 genes. It is possible to see that each gene has a real value, thus representing one parameter or variable of the problem.

2.1	2.8	0.5	1.3	2.3	1.7

Fig. 1. Example of an individual with 6 genes

Mutation Mutation is the process of perturbation of the population. Mutation produces new individuals, denominated trial individuals, from the addition of an aleatory individual γ from the population to the difference between two other aleatory individuals α and β from the same population. This process utilizes a factor of perturbation $F \in [0; 2]$ that ponderate the mutation operator. Let x_α, x_β, and x_γ be three distinct individuals that belongs to the population X at generation g. A trial individual V, that belongs to the generation $g+1$ can be generated as show in Equation (4).

$$V^{g+1} = x^g_\gamma + F(x^g_\alpha - x^g_\beta) \tag{4}$$

Crossing The crossing operator is used in order to increase the diversity of the trial individuals V generated at the mutation process. Consider that, for each target individual $x^g_s, s \in X, s \neq \alpha \neq \beta \neq \gamma$ a trial individual V^{g+1} is generated from mutation. Then, an individual U^{g+1}, denominated offspring, is generated as shown in Equation (5).

$$U^{g+1}_i = \begin{cases} V^{g+1}_i, & \text{if} \quad r_i \leq Cf \\ x^g_{s,i}, & \text{if} \quad r_i > Cf \end{cases}, \quad \forall i \in \{0, \ldots, N\}, r_i \in [0; 1] \tag{5}$$

where $U^{g+1}_i, V^{g+1}_i, x^{g+1}_{s,i}$ are the i-th gene of the individuals U^{g+1}, V^{g+1}, and x^g_s, respectively. Value r_i is a number generated at random, at each iteration of the algorithm. Let $Cf \in [0; 1]$ represents the possibility of the offspring to inherit the genes from the trial individual V^{g+1}_i. When $Cf = 1$, the offspring will be equal to the trial individual V^{g+1}. At the other hand, when $Cf = 0$, the offspring will be equal to the target vector x^g_s. If $0 < Cf < 1$, the offspring can receive genes from both V^{g+1} and x^g_s.

Evolution The evolution is the last stage of the evolutionary process. At evolution stage of generation g, the individuals that will be at the generation $g+1$ are selected. If the generated offspring fitness is better than the target vector offspring, then the target vector is replaced at the population by the offspring.

Otherwise, the population remains the same. This process occurs for all offspring generated at each generation.

3 Caloric Restricted Diet Problem

The Caloric Restricted Diet Problem (CRDP) aims to develop caloric restricted diets that provides all necessary nutrients for a daily basis intake. Moreover, it is also concerned to provide tasty meals, that can be easily accepted by a patient. Thus, one can lose weight by a healthy way, while consuming a great diversity of meals.

In Literature, it is possible to find a greater number of variations for the number of recommended calories into a hypocaloric diet. The work [15] suggest an initial diet with 1200 kilocalories (Kcal) for overweight women. Moreover, the work [4] suggests that diets from 1000 to 1200 Kcal can improve the health of a set of obese individuals within 25 days. Thus, this work adopt as objective to develop diets such that $Kcal \approx 1200$. CRDP objetive function is stated as Equation (6).

$$min \quad |1200 - Kcal| \tag{6}$$

where $Kcal$ denotes the number of calories in the diet. As this work is interested to obtain a diet with approximate 1200 $Kcal$, the objective function use the modulus of the result to minimize the distance between the number of calories at the formulated diet and the recommended number of calories. Let F be the set of available foods. The value of $Kcal$ is calculated as Equation (7).

$$Kcal = \sum_i Kcal_i p_i y_i \tag{7}$$

where $Kcal_i$ denotes the amount of $Kcal$ contained in one portion of the product $i \in F$. The value $p_i \in [0.5; 3]$ corresponds to the number portions of each product $i \in F$, and y_i is a binary variable such that $y_i = 1$ if the product i is part of the diet, and $y_i = 0$ otherwise, for all $i \in F$. We consider 100 g of a food, or 100 mL of a beverage as a portion. The value of p_i is limited at the interval $[0.5; 3]$ in order to avoid a greatly quantity of some product to appear, thus dominating the diet, or to consume a minor portion of some product, that may be impracticable at some real situation (imagine to cook only 10 g of fish at dinner, for example).

Besides the number of calories of the diet, CRDP takes into account the amount of proteins (pt), carbohydrates (c), sodium (Na), dietary fibers (f), calcium (Ca), magnesium (Mg), manganese (Mn), phosphor (P), iron (Fe), and zinc (Zn). Thus, CRDP have ten constraints, as shown in Table 1. Each couple of columns display the name of the nutrient and its necessary daily intake, respectively. Then, a set of constraints is inserted at the problem, as shown in Equation (8).

$$\sum_i m_{ij} p_i y_i \geq b_j \tag{8}$$

Table 1. Daily recommended nutrients intake [20]

nutrient	amount (g)	nutrient	amount (mg)	nutrient	amount (mg)
pt	≥ 75	Ca	≥ 1000	Mg	≥ 260
c	≥ 300	Mn	≥ 2.3	P	≥ 700
f	≥ 25	Fe	≥ 14	Zn	≥ 7
Na	≤ 2.4				

3.1 Food database

This work consider the Brazilian Table of Food Composition (TACO) (Tabela Brasileira de Composição de Alimentos, in portuguese) [12] as the information source about the nutritional data of foods. TACO was elaborated at the University of Campinas, Brazil. It contains data from a large number of foods and beverages, with a quantitative description about 25 of their nutrients and properties. Each data represents a portion of 100 g (or 100 mL, when appropriated) of a product. Moreover, TACO displays information about foods in a variety of states (frozen, cooked, with or without salt, for instance).

TACO presents each product in more than one presentation. For instance, it presents uncooked meat, uncooked meath with salt, and cooked meat. Thus, only subset of the products of TACO (that can be served in a meal) are selected to be in F. Moreover, TACO presents 25 different information about each product. This work considers only the most important of then, thus considering 11 from 25 nutrients in the problem. The selected nutrients are displayed at the odd columns of Table 1.

The subset of selected products were classified into nine different categories, as shown in Table 2. The first column displays the class of the product. The second column displays the symbol that represents the class. The third column displays the number of products at each class. The last column displays some extra information about the products contained in each class.

Table 2. Different classification of products from TACO

product	symbol	#	information
beverages	B	21	beverages, except natural juices and alcoholic beverages
juices	J	11	natural fruits juice
fruits	F	62	fruits, in general
lacteal	L	19	products derived from milk
carbohydrates 1	C1	21	snacks, as bread, cookie, and cracker
carbohydrates 2	C2	12	main meals carbohydrates, as rice, potato, and cassava
grains	G	12	leguminous foods, such as lentils and beans
vegetables	V	41	vegetables, in general
proteins	P	95	high proteic foods, as meat, chicken, and eggs

4 Differential Evolution for CRDP

4.1 Solution representation

In order to represent correctly the diet of the people, an individual, also called a solution of DE, needs to represent the different meals that exists during a day. Thus, a solution is represented as a combination of the most common types of products consumed in each meal. As products are divided in categories, it is possible to build a diet that has a great variety of products every day.

Figure 2 shows the diversity of products in each meal. This work considers 6 different meals at a day, namely the breakfast, two snacks, lunch, dinner, and supper. Breakfast is the first meal by the morning. Snacks are small meals that can exists between two major meals. Supper is the last meal, after the dinner and before sleep, that is common to a great number of individuals.

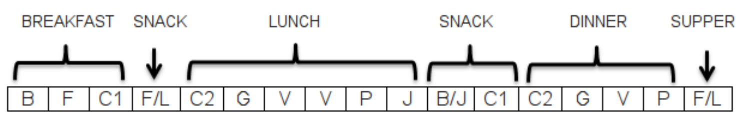

Fig. 2. Representation of a daily diet based on their food classes

Based at Figure 2, a solution for CRDP is expressed as shown in Figure 3. A solution consists in two vectors of 17 genes each, that aggregates 6 different meals, as expressed by Figure 2. The first vector contains the quantity of portions p_i of each product $i \in F$ to be consumed. The second vector contains a bijective function $f : \mathbb{N} \mapsto F$ that maps each number as a product.

0.6	2.1	2.9	1.2	1	0.8	2	2.7	1.9	1.1	0.6	1	1.6	1.1	1.7	2	0.5
13	52	16	17	19	7	33	12	84	10	18	6	16	11	31	9	41

Fig. 3. Representation of an individual

4.2 Penalty function

When the constraint is not violated, the CRPD employs the objective function (Equation (6)) as the fitness function. However, when constraints are violated, a mechanism to handle these constraints must be adopted. It is worth mentioning that the DE method does not deal with the constraints properly and, thus, some technique must be adopted to deal with this issue. Hence, the fitness function becomes the objective function plus a penalization term [6]. This term is defined in accordance with Equation (9).

$$min \quad |Kcal - 1200| + \sum_j \left(\frac{|\sum_{i \in F}(m_{ij} p_i y_i) - b_j|}{b_j} \right) \cdot M, \quad \forall j \in N \qquad (9)$$

where m_{ij} denotes the amount of the nutrient $j \in N$ at product $i \in F$, M is a big constant number, used as a penalty term for violated constraints. Both p_i, y_i, and b_j are the same as denoted at Equation (7). One can see that there is no penalty for the value of y_i. The limits of y_i are handled by DE, such that a value out of the interval $[0.5; 3]$ is rounded to the closest value inside the interval.

4.3 Population initialization, mutation, crossing, and evolution

The population is initialized at random. For each gene of an individual $x_i \in X$, a product is selected at random, always respecting the group that it belongs. Next, a random value $[0.5; 3]$ is assigned to each product.

At each generation, $|X|$ new offsprings are generated through the mutation and crossing operators. The mutation operator is the same as described at Subsection 2.4. The crossing operator uses Equation (5) to produce new values for the first vector of the individual. The number of the product at the trial vector is selected at random among the chosen individuals α, β, and γ. Figure 4 shows an example of the crossing operator between individuals with 3 genes. It can be see that the product of the trial vector is one of the products of individuals α, β, or γ. The offspring products, as the products from the trial vector, are random chosen between the target and the trial vector.

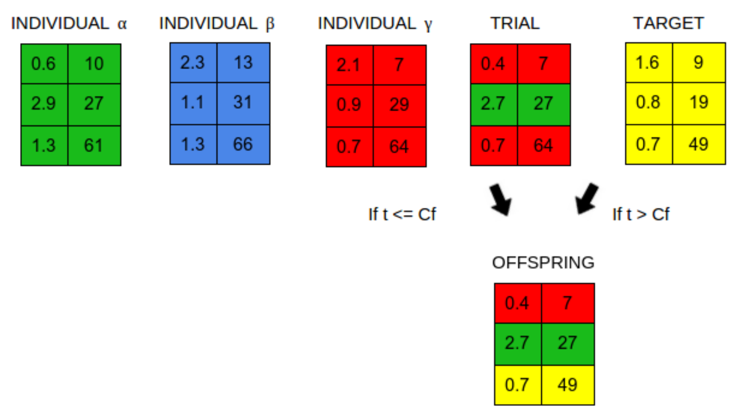

Fig. 4. Illustration of a mutation operation and crossing at the DE

As the number of offsprings is the same of the number of individuals at the current generation, the evolution process compares one offspring with one individual from the current population. Then, the individual with the worst value of fitness is discarded, and the other individual is included at the population. This procedure ensures that a generation $g+1$ will be equal or better than generation g.

5 Computational experiments

Computational experiments have been performed on a single core of an Intel Core i5 CPU 5200U with 2.2 GHz clock and 4 GB of RAM memory, running Linux operating system. DE was implemented in C and compiled with GNU GCC version 4.7.3.

5.1 Parameter tuning

DE has three different parameters, namely the perturbation factor, the crossing factor, and the population size. The perturbation factor F varies in the interval $[0; 2]$. The crossing factor is a percent value, and varies in the interval $[; 1]$. The population size $|X|$ is an integer number greater than 2. In order to tune the algorithm, a complete factorial experiment was developed with their three parameters, in order to choose the best value for each one.

The value of the perturbation factor F was varied in the set $\{0.3, 0.8, 1.3, 1.8\}$. The crossing factor Cf was varied in the set $\{0.4, 0.6, 0.8\}$. Finally, the population size $|X|$ was varied in the set $\{10, 50, 100\}$. Using these values, a complete factorial experiment was performed in order to evaluate DE parameters. As DE is not a deterministic algorithm, it is necessary to execute each experiment repeatedly, in order to minimize its non-determinist nature. Thus, each experiment was repeated 20 times.

5.2 Results and discussion

The results of the complete factorial experiment are shown in Table 3. Each row of the table corresponds to 20 repetitions of a experiment. The first, second, and third columns show the parameter values. The fourth, fifth, sixth, and seventh columns show the results for 20 executions of the experiments with the defined parameter values. There is a horizontal line that split the results according to the population size, and the experiment that has the best average fitness value for each population size is highlighted. As DE ran with up to 2 seconds, the computational time was disregarded from this experiment.

In Table 3, one can see that the population size is positive correlated with the solution value. There is no one other apparent correlation between the perturbation factor F and the crossing factor Cf with the solution value. Based on this experiment, in order to solve CRDP, the best set of parameter for DE is $F = 0.3, Cf = 0.6$, and $|X| = 100$.

Table 3. Results for the complete factorial experiment

parameters			fitness			
population size	F	Cf	average	best	worst	standard deviation
10	0.3	0.4	1665.81	1544.91	1820.98	75.96
10	0.3	0.6	1761.16	1552.56	2166.99	142.26
10	0.3	0.8	1869.79	1685.96	2393.99	153.65
10	0.8	0.4	1631.13	1524.63	1778.55	67.24
10	0.8	0.6	1659.09	1495.48	1915.12	99.23
10	0.8	0.8	1675.09	1545.51	1846.58	89.65
10	**1.3**	**0.4**	**1630.83**	**1512.67**	**1793.14**	**71.12**
10	1.3	0.6	1633.73	1528.59	1740.91	66.66
10	1.3	0.8	1696.47	1586.12	1847.05	82.50
10	1.8	0.4	1650.69	1537.55	1782.75	64.80
10	1.8	0.6	1667.26	1520.58	1849.50	91.60
10	1.8	0.8	1749.45	1611.98	1911.00	83.72
50	**0.3**	**0.4**	**1507.89**	**1456.03**	**1579.59**	**26.23**
50	0.3	0.6	1508.93	1449.48	1608.58	36.26
50	0.3	0.8	1522.09	1449.70	1578.57	38.45
50	0.8	0.4	1519.45	1461.62	1570.28	28.50
50	0.8	0.6	1514.38	1479.55	1558.03	22.77
50	0.8	0.8	1522.01	1434.66	1596.34	42.71
50	1.3	0.4	1530.11	1488.79	1578.24	26.67
50	1.3	0.6	1518.93	1470.49	1567.46	25.05
50	1.3	0.8	1528.54	1478.57	1579.48	28.65
50	1.8	0.4	1517.64	1483.21	1588.33	24.99
50	1.8	0.6	1527.42	1485.89	1594.77	28.71
50	1.8	0.8	1531.60	1469.17	1632.94	43.39
100	0.3	0.4	1491.56	1450.89	1528.48	20.88
100	**0.3**	**0.6**	**1479.08**	**1430.06**	**1531.20**	**24.66**
100	0.3	0.8	1484.84	1442.14	1567.03	31.84
100	0.8	0.4	1496.23	1459.74	1527.54	17.44
100	0.8	0.6	1487.97	1435.62	1520.59	19.29
100	0.8	0.8	1493.18	1452.27	1532.95	26.63
100	1.3	0.4	1507.45	1461.57	1546.01	21.19
100	1.3	0.6	1499.52	1441.36	1564.64	28.28
100	1.3	0.8	1509.47	1474.18	1546.89	22.49
100	1.8	0.4	1511.40	1483.83	1541.32	14.18
100	1.8	0.6	1503.08	1471.72	1551.44	23.44
100	1.8	0.8	1507.61	1468.57	1558.26	25.49

Figure 5 shows a detailed convergence curve for the three highlighted parameter set in Table 3. One can see a difference of almost 300 $Kcal$ between the best and worst results with a population size $|X| = 10$. Moreover, as the population size increases, the algorithm convergence curve also increases. It shows that the solutions diversity has an important role in DE, preventing it to be stuck at a local minima prematurely.

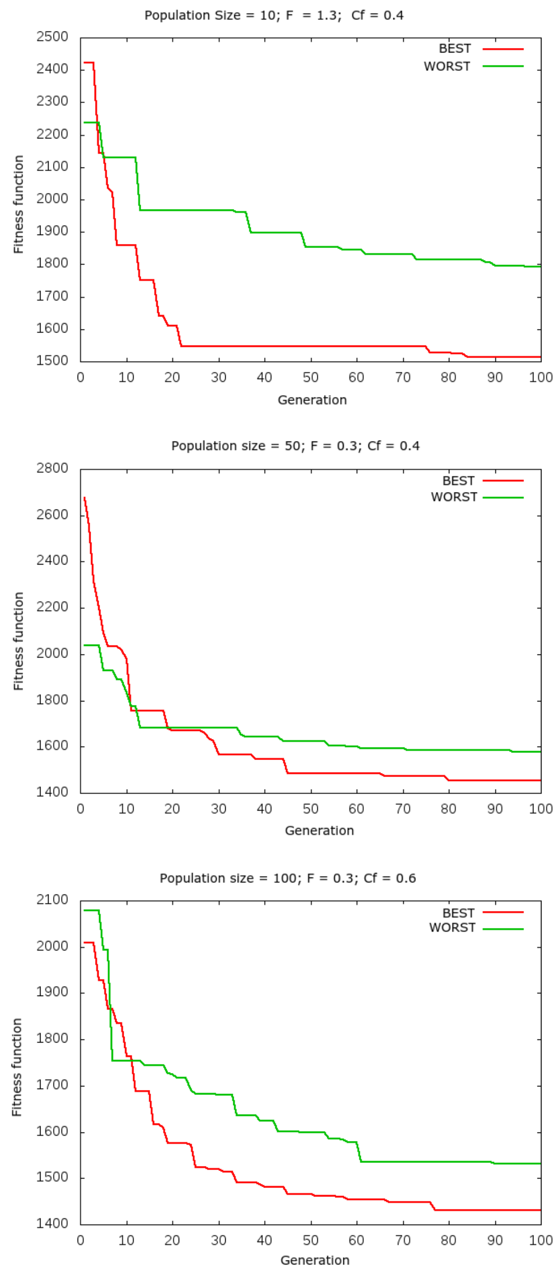

Fig. 5. Convergence curve for the best and the worst executions, for each set of highlighted parameters

In order to show the importance of the constraints, a DE with the best set of parameters was executed another 20 times at CRDP without the 10 nutrients constraints. Thus, DE aims only to minimize the caloric cost of the diet, without considering the basic nutrient needs of a person. Table (4) summarizes the results of this experiment. It shows that a diet that just only minimizes the number of calories can be very dangerous to the health, as 8 of the 10 nutritional requirements are not satisfied.

Table 4. Violations of the diet without constraint

nutrient	pt	c	f	Ca	Mg	Mn	P	Fe	Na	Zn
constraint	≥ 75	≥ 300	≥ 25	≥ 1000	≥ 260	≥ 2.3	≥ 700	≥ 14	≤ 2400	≥ 7
reached	62.7	174.4	22.4	736.5	253.0	1.84	1050.5	10.0	819.0	5.8
violation (%)	16.4	41.8	10.2	23.6	2.69	20.0	0.0	28.5	0.0	17.4

Figure 6 shows the convergence curve of 20 independent executions of DE with the best set of parameters found. One can see that all the convergence curves are close to each other. Thus, it indicates that DE is a stable method to solve CRDP.

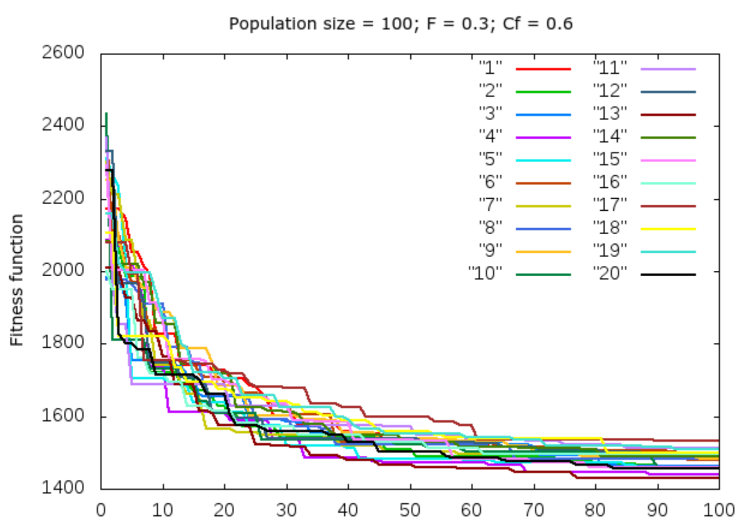

Fig. 6. Evolution of 20 independent runs of DE with the the best set of parameters

5.3 Example diets

Table 5 shows two possible caloric restricted diets generated though DE. One can see the diversity of foods and beverages among diets. DE can be executed many times to solve CRDP, thus generating different diets at each execution. Thus, a weekly diet can be generated in seconds. The great diversity of foods and beverages makes generated diets easy to follow, such that the palate will not be tired with repetitive meals.

Table 5. Example of two possible diets

diet 1 **1464.50 Kcal**		diet 2 **1490.36 Kcal**	
breakfast			
Nonfat plain yogurt	190 ml	Coconut water	170 ml
Banana Pacovan (raw)	285 g	Plum (raw)	265 g
Morning cereal, corn flavor	50 g	Morning cereal, corn flavor	115 g
snack			
White guava with peel	50g	Pear (raw)	80g
lunch			
Potatoes (baked)	145 g	Sweet potato (baked)	70 g
Beans (baked)	95 g	Carioca beans (cooked)	130 g
Alfavaca (raw)	210 g	Broccoli (cooked)	270 g
Tomato (raw)	230 g	Alfavaca (raw)	275 g
Corvina fish (cooked)	65 g	Sirloin tri tip roast (grilled)	50 g
Galician lemon juice	220 g	Rangpur lime juice	280 ml
snack			
Tangerine juice	220 ml	Rangpur lime juice	85 ml
Green corn	50 g	Green corn	50 g
dinner			
Sweet potato (baked)	210 g	Sautéed potatoes	100 g
Lentil (cooked)	50 g	Beans (baked)	80 g
Spinach (raw)	275 g	Celery (raw)	125 g
Shrimp (boiled)	60 g	Sardines (grilled)	60 g
supper			
Pitanga (raw)	225 g	Pear (raw)	225 g

6 Conclusions and future works

In this work, the Caloric Restricted Diet Problem (CRDP) was proposed. A Mixed Integer Linear Programming formulation was proposed, so as a Differential Evolution (DE) algorithm that solves the proposed formulation. Computational experiments with the proposed algorithm shows that it is possible to generate tasty and health diets with a reduced number of calories. Another contribution of this work is to model meals as a set of different foods and beverages classes. This modeling guarantee that generated meals are closer to the reality.

The key point to solve CRDP is to generate a great diversity of solutions. Thus, various diets can be easily generated through DE. Deterministic methods, such as branch-and-bound or mathematical programming approaches, can not be properly used to solve CRDP, as they always generate the same diet.

Although CRDP is not a hard problem, from the computational point of view, it can be a hard problem in practice. It can takes an expensive amount of time to generate a weekly diet, with a great variety of meals, as the ones generated though DE. Thus, this work can be applied to real situations. It can be used by practitioners to generate a great amount of diets, in a small amount of time.

As future works, it is proposed to generate different types of diets. This model and algorithm can be used to generate an uncountable number of diets, from high-protein diets, for those who are interested in muscle mass gain, to balanced vegan diets. Another future work is to develop a multi-objective version of CRDP, that minimizes, besides the number of calories, the financial cost or the sodium intake, while maximizes the amount of dietary fibers and proteins, for example.

Acknowledgments

This work was partially supported by the Brazilian National Council for Scientific and Technological Development (CNPq), the Foundation for Support of Research of the State of Minas Gerais, Brazil (FAPEMIG), and Coordination for the Improvement of Higher Education Personnel, Brazil (CAPES).

References

1. Babu, B., Angira, R.: Modified differential evolution (mde) for optimization of non-linear chemical processes. Computers & chemical engineering 30(6), 989–1002 (2006)
2. Bas, E.: A robust optimization approach to diet problem with overall glycemic load as objective function. Applied Mathematical Modelling 38(19), 4926–4940 (2014)
3. Cheng, S.L., Hwang, C.: Optimal approximation of linear systems by a differential evolution algorithm. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 31(6), 698–707 (2001)
4. Crampes, F., Marceron, M., Beauville, M., Riviere, D., Garrigues, M., Berlan, M., Lafontan, M.: Platelet alpha 2-adrenoceptors and adrenergic adipose tissue responsiveness after moderate hypocaloric diet in obese subjects. International journal of obesity 13(1), 99–110 (1988)
5. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. Evolutionary Computation, IEEE Transactions on 15(1), 4–31 (2011)
6. Datta, R., Deb, K.: Evolutionary Constrained Optimization. Springer (2015)
7. Gallenti, G.: The use of computer for the analysis of input demand in farm management: A multicriteria approach to the diet problem. In: First European conference for information technology in agriculture (1997)
8. Garille, S.G., Gass, S.I.: Stigler's diet problem revisited. Operations Research 49(1), 1–13 (2001)

9. Hachicha, N., Jarboui, B., Siarry, P.: A fuzzy logic control using a differential evolution algorithm aimed at modelling the financial market dynamics. Information Sciences 181(1), 79–91 (2011)
10. Kim, H.K., Chong, J.K., Park, K.Y., Lowther, D.A.: Differential evolution strategy for constrained global optimization and application to practical engineering problems. Magnetics, IEEE Transactions on 43(4), 1565–1568 (2007)
11. Kohsaka, A., Laposky, A.D., Ramsey, K.M., Estrada, C., Joshu, C., Kobayashi, Y., Turek, F.W., Bass, J.: High-fat diet disrupts behavioral and molecular circadian rhythms in mice. Cell metabolism 6(5), 414–421 (2007)
12. Lima, D.M., Padovani, R.M., Rodriguez-Amaya, D.B., Farfán, J.A., Nonato, C.T., Lima, M.T.d., Salay, E., Colugnati, F.A.B., Galeazzi, M.A.M.: Tabela brasileira de composição de alimentos – taco (2011), http://www.unicamp.br/nepa/taco/tabela.php?ativo=tabela, acessed: 2016-03-16
13. Mamat, M., Rokhayati, Y., Mohamad, N., Mohd, I.: Optimizing human diet problem with fuzzy price using fuzzy linear programming approach. Pakistan Journal of Nutrition 10(6), 594–598 (2011)
14. Neri, F., Tirronen, V.: Recent advances in differential evolution: a survey and experimental analysis. Artificial Intelligence Review 33(1-2), 61–106 (2010)
15. Pasquali, R., Gambineri, A., Biscotti, D., Vicennati, V., Gagliardi, L., Colitta, D., Fiorini, S., Cognigni, G.E., Filicori, M., Morselli-Labate, A.M.: Effect of long-term treatment with metformin added to hypocaloric diet on body composition, fat distribution, and androgen and insulin levels in abdominally obese women with and without the polycystic ovary syndrome. The Journal of Clinical Endocrinology & Metabolism 85(8), 2767–2774 (2000)
16. Stefanick, M.L., Mackey, S., Sheehan, M., Ellsworth, N., Haskell, W.L., Wood, P.D.: Effects of diet and exercise in men and postmenopausal women with low levels of hdl cholesterol and high levels of ldl cholesterol. New England Journal of Medicine 339(1), 12–20 (1998)
17. Stigler, G.J.: The cost of subsistence. Journal of farm economics 27(2), 303–314 (1945)
18. Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 11(4), 341–359 (1997)
19. Thomsen, R.: Flexible ligand docking using differential evolution. In: Evolutionary Computation, 2003. CEC'03. The 2003 Congress on. vol. 4, pp. 2354–2361. IEEE (2003)
20. de Vigilância Sanitária (ANVISA), A.N.: Resolution rdc number 360. Diário Oficial da União (2003), http://goo.gl/PZAChm, acessed: 2016-03-16
21. Warwick, Z.S., Schiffman, S.S.: Role of dietary fat in calorie intake and weight gain. Neuroscience & Biobehavioral Reviews 16(4), 585–596 (1992)
22. Who, J., Consultation, F.E.: Diet, nutrition and the prevention of chronic diseases. World Health Organ Tech Rep Ser 916(i-viii) (2003)
23. Yang, L., Colditz, G.A.: Prevalence of overweight and obesity in the united states, 2007-2012. JAMA internal medicine (2015)