An Island-Based Hybrid Evolutionary Algorithm for Caloric-Restricted Diets

João Gabriel R Silva
IFMT
Grasiele R. Duarte
UFRJ: Universidade Federal do Rio de Janeiro
Iago A. Carvalho
UFMG: Universidade Federal de Minas Gerais
Leonardo Goliatt (✉ goliatt@gmail.com)
UFJF: Universidade Federal de Juiz de Fora https://orcid.org/0000-0002-2844-9470
Vinicius F. Vieira
UFSJ: Universidade Federal de Sao Joao del-Rei
Carolina R. Xavier
UFSJ: Universidade Federal de Sao Joao del-Rei

Research Article

Keywords: Obesity, Diet Problem, Evolutionary Algorithms, Island Model

Posted Date: April 20th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-431029/v1

Noname manuscript No.
(will be inserted by the editor)

An Island-Based Hybrid Evolutionary Algorithm for Caloric-Restricted Diets

João Gabriel R. Silva · Grasiele R. Duarte · Iago A. Carvalho · Leonardo Goliatt* · Vinicius F. Vieira · Carolina R. Xavier

Received: date / Accepted: date

Abstract The most popular and successful way to maintain a healthy body is to have a rich and balanced diet, combined with physical exercise. Since it was proposed the diet dilemma, several works in the literature suggested calculating a diet that respects an individual's nutritional needs. In the Caloric-Restricted Diet Problem (CRDP), the goal is to find a reduced-calorie diet that meets an individual's dietary needs aiming for weight loss. This paper proposes a Hybrid Island-Based Evolutionary Algorithm (IBHEA) that combines a Genetic Algorithm (GA) with a Differential Evolution (DE) communicating through a migration policy to solve the CRDP. Computational experiments showed that IBHEA outperforms the non-distributed and non-hybrid implementations, generating a greater variety of diets with a small calorie count.

Keywords Obesity · Diet Problem · Evolutionary Algorithms · Island Model

João Gabriel R. Silva
Computational Modeling, Federal University of Juiz de Fora, Brazil
E-mail: joaogabriel.comp@gmail.com

Grasiele R. Duarte
Civil Engineering Program, COPPE/Federal University of Rio de Janeiro, Brazil
E-mail: grasiele.duarte@coc.ufrj.br

Iago A. Carvalho
Computer Science Program, Federal University of Minas Gerais, Brazil
E-mail: iago.august@gmail.com

L. Goliatt * (corresponding author)
Computational Modeling, Federal University of Juiz de Fora, Brazil
E-mail: leonardo.goliatt@ufjf.br

Vinicius F. Vieira
Computer Science Program, Federal University of Sao Joao del Rei, Brazil
E-mail: vfvieira@gmail.com

Carolina R. Xavier
Computer Science Program, Federal University of Sao Joao del Rei, Brazil
E-mail: carolrx@gmail.com

1 Introduction

A high-calorie diet and a lack of physical activities are the main causes of overweight and obesity [11]. An efficient treatment for obesity consists of the ingestion of a hypocaloric diet [1,6]. It is indicated that a hypocaloric diet with 1200 kilocalories (kcal) can be used to reduce their body fat [15]. Besides, a hypocaloric diet with 1000 to 1200kcal can improve the life quality of obese individuals in less than 30 days [5]. However, these diets also need to provide all the necessary vitamins and nutrients to the individuals, such as carbohydrates, iron, zinc, fibers, among others [18].

The Diet Problem (DP) proposed in [22] aims to calculate the minimum financial cost of a diet that satisfies an individual's nutritional needs. The CRDP, a DP variation, was introduced in [19] to minimize the calorie count of a diet instead to reduce the financial cost. More precisely, CRDP aims to build a diet that minimizes the number of ingested calories while respecting the minimal amount of nutrients necessary for an individual. The result of CRDP represents a diet composed of six daily meals applied in real-life nutrition.

It is difficult to develop a caloric-restricted diet that provides all necessary nutrients for an individual, such as proteins, zinc, and iron. The first attempt to develop an algorithm for computing a healthy diet was presented in [22]. However, it only seeks a minimum financial cost for the diet, not concerning the diet's application in the daily routine.In [19] it was proposed the CRDP, an optimization problem to compute healthy caloric-restricted diets while preserving the nutritional needs of an individual. The CRDP can be expressed by

$$\begin{cases} \text{minimize } f(kcal,p,y) = \left| 1200 - \sum_{i=1}^{T} kcal_i \times p_i \times y_i \right| \\ \text{subject to } g_j = \sum_{i=1}^{T} m_{i,j} \times p_i \times y_i \geq b_j, \qquad j = 1,2,...,N \end{cases}, \tag{1}$$

where T is the total of available foods to be included in diets, $kcal_i$ is the calorie count of one portion of food i, $p_i \in [0.5,3]$ is the portion of food i to be consumed in diet, $y_i \in \{0,1\}$, such that $y_i = 1$ if the food i is included in the diet and $y = 0$ otherwise, N is the total of nutrients considered in the problem and $m_{i,j}$ is the amount of nutrient j in food i and b_j is the minimal amount of the nutrient j required in diet. In Equation (1), the values for $kcal_i$ and b_j are properly indicated by professionals and regulatory agencies in the area.

The objective function $f(kcal,p,y)$ in Equation (1) aims to develop a diet with 1200kcal, the most indicated one to reduce weight [5,15]. A diet with a higher amount of calories does not greatly contribute to this objective. On the other hand, a diet with a smaller amount of calories can represent a risk to individual health. The model considers portions of 100g and 100mL for solid and liquid foods, respectively. Besides, $p_i \in [0.5,3]$ ensures that no food will be prescribed in relatively low or high quantity. It prevents real impracticable situations as cooking, for example, only 10g of fish at dinner, or that it dominates the diet due to its high quantity.

In Equation (1) is considered 9 nutrients, namely dietary fibers (Df), carbohydrates (C), proteins (Pt), calcium (Ca), manganese (Mn), iron (Fe), magnesium (Mg), phosphor (P), and zinc (Zn). Table 1 presents the daily ingestion recommended for considered nutrients according to the Brazilian Table of Food Composition (TBCA) [13].

Table 1 Recommended nutrient daily intake considered in CRDP according to TBCA.

Unit	Requirement
(g)	$Df \geq 25$, $C \geq 300$, $Pt \geq 75$
(mg)	$Ca \geq 1000$, $Mn \geq 2.3$, $Fe \geq 14$, $Mg \geq 260$, $P \geq 700$, $Zn \geq 7$

The TBCA contains data regarding many foods and beverages, with a description of 25 of their nutrients. Each data refer to a portion of 100g or 100mL appropriately. This work only considers the nine most important nutrients from 25 in TBCA. Concerning the alternatives of food to built the diets, this work selected 9 groups of food products to CRDP, with the participation of a specialist in the area. They were classified into 9 categories according to their characteristics displayed in Table 2. In Table 2, beverages, except natural juices J and alcoholic beverages, are symbolized by B. Milk-derived products have the symbol L. Carbohydrates are separated into two different groups, symbolized by $C1$ and $C2$. $C1$ contains snacks, like bread, cookies, and crackers, while $C2$ are the main meal carbohydrates, such as rice, potato and cassava. Grains and leguminous foods, such as lentils and beans, are represented as G. The proteins category P contains high protein foods, such as meat, chicken and eggs.

Table 2 Classification of food products from TBCA [13]. In table, Interval is the range of codes associated to foods in each category.

Category	Symbol	Number of foods	Interval
beverages	B	21	1–21
juices	J	11	22–32
fruits	F	62	33–94
lacteal	L	19	95–113
carbohydrates 1	$C1$	21	114–134
carbohydrates 2	$C2$	12	135–146
grains	G	12	147–158
vegetables	V	41	159–199
proteins	P	95	200–294

When proposed in [19], the CRDP was solved by a DE. This work proposes to solve it by the IBHEA, which distributes the population into sub-populations to be evolved individually by DE and GA instances. However, they maintain periodic communication by the migration operator, by which sub-populations exchange solutions among themselves. The results demonstrate the better performance of IBHEA compared to isolated algorithms and non-hybrid implementations of the Island Model. The use of the IBHEA achieved significant results for the CRDP since it was able to reduce the amount of calories of diets without violating the constraints, contributing to the solution quality to optimization problems present in routine of anyone, such as the definition of a daily healthy diet.

2 Island-Based Hybrid Approach for CRDP

The concept of evolution has been considered in solving the optimization problem of different areas by Evolutionary Algorithms (EA) since the 1950s. Commonly these algorithms build a population of candidate solutions to a given problem and apply operators inspired by natural genetic variation and natural selection [14]. Over the years different alternatives to implement EAs have also been proposed in the literature. One of those alternatives is the Island Model (IM) by which it is possible to run EAs in parallel computing environments. An IM is a class of distributed EAs in which the population is divided into multiple sub-populations called islands. Each island run its own EA independently, but periodically they interact through the migration operator [3,21]. The islands exchange solutions between themselves according to topology and a set of rules defined by the user through IM parameters by migration.

To develop a populational algorithm, it is necessary to define the individual shape, i.e., how the solution is numerically represented in the algorithm. Each solution can be evaluated to give a fitness value that shows how fit is the specific individual to the expected solution. The following sections first present concepts and features to be considered for both Genetic Algorithm (GA) and Differential Evolution (DE). Then specific operators for each evolutionary algorithm are described.

2.1 Representation of Solutions

On the adjust of IBHEA to solve the CRPD, the candidate solutions is represented as proposed in [19]. Figure 1 illustrates the general model for a solution for CRPD. This model represents a daily routine of meals in diets. Figure 1 presents the organization of the 6 ordered daily meals, namely breakfast, two snacks, lunch, dinner and supper. In this sense, in addition to the number of food groups that must be in each meal, Figure 1 clarifies which are those food groups.

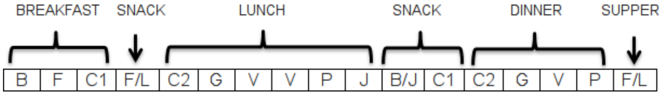
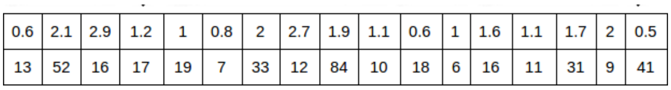


Fig. 1 Solution encoding for CRPD: 6 ordered different meals, each one compose by a specific number and groups of food products.

Based on the model presented in Figure 1, the dimension of CRPD is 17 and the candidate solution of EAs in IBHEA can be represented as illustrated in Figure 2. In this case, each solution consists of two vectors, both with 17 values which describe the 6 daily meals. The first vector (top vector in Figure 2), called *portions vector*, indicates the portions $p_i \in [0.5, 3]$ of each selected food product i, such that $i = 1, 2, ..., T$. The second vector (bottom vector in Figure 2), denoted as *ID vector*, contains integer values $\in [1, 294]$ corresponding to the code of food product included in diet according to Table 2. Each value in *ID vector* is mapped to the respective food and nutritional characteristics by a bijective function $f : \mathbb{N} \mapsto T$ applied in IBHEA. Besides, each value in *ID vector* is restricted to those in Table 2. Initially, each *portions vector* and *ID vector* in population is randomly generated according to the respective domains.



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. 2 Representation of an IBHEA candidate solution for CRDP.

2.2 Population Initialization, Fitness Function and Constraint Handling

IBHEA initial population X is randomly generated with $|X| = pop$ individuals. For the *portions vector*, a random real number is generated in the interval $[0.5;3]$ for each gene. A random integer number is generated for each gene of the *ID vector* considering the interval of its class.

Concerning the evaluation of each IBHEA candidate solution according to Equation (1), when any constraints are violated, a mechanism to handle these violations must be adopted. Hence, the fitness function becomes the objective function $f(kcal,p,y)$ plus a penalization term [8].

Let N' represent the set of violated constraints and M is a large constant value used as a penalty term. The penalized fitness function is given by

$$\text{minimize } |Kcal - 1200| + M \times \sum_{j \in N'} \left(\frac{\left|\sum_{i \in T}(m_{i,j} \times p_i \times y_i) - b_j\right|}{b_j} \right), \qquad (2)$$

where $Kcal$ is the objective function $f(kcal,p,y)$ in Equation (1).

2.3 Evolutionary Algorithms

2.3.1 Genetic Algorithm

John Holland [10] described the Genetic Algorithm (GA) inspired in the evolution theory proposed by Darvin. GA is based on the operations selection, recombination (Crossover), mutation, and population update. GA was initially proposed for binary representation of the individuals, but then it was extended for several variations.

GA operations have many different implementations and, in general, GA works as shown in Figure 3. First, parents are selected. Then, the recombination, mutation and evaluation phases are executed, in this order. Finally, the most fitted individuals are selected to go to the next generation. The operations and some parameters considered in this were defined as follows.

The parent selection is performed considering the tournament selection strategy. Each tournament (which will generate two individuals of the intermediate population) involves two parents. For the selection of each of the two parents in the tournament,

```
Initialize the population
Evaluate the population
while stopping criterion is not met do
    Parents Selection
    Recombination
    Mutation
    Evaluate the intermediary population
    New population Selection
end while
return the best solution found
```

Fig. 3 GA pseudocode

two distinct candidates are chosen among all individuals in the population. The fittest candidate is selected as a parent.

The two vectors that represent a candidate solution are recombined in distinct ways. The $\alpha\beta-$blend crossover recombines portions vector with a randomly selected value between the parent's values. Two points-crossover recombination operator recombines the ID Vector. The mutation operator works as follows: for the portions vector, the mutation operator chooses a random value for a portion between 0.5 and 3. For the ID Vector, the mutation operator chooses another food ID for the chosen position.

To select the survivors for the next generation, each individual of the new generation was compared with the individual in the same position in the population of the previous generation, and the individual with the best fitness is selected for the next generation. This operation guarantees the monotonicity of the algorithm, and the best solution is never lost.

2.3.2 Differential Evolution Algorithm

The Differential Evolution algorithm (DE) [16] is a classical evolutionary algorithm originally designed to deal with real-valued variables. It is a general algorithm that can be easily adapted to a wide range of problems. In the literature, it is possible to find DE applications in engineering [12], chemistry [2], biology [4], finances [17], artificial intelligence [23] and other areas [7].

The application of DE algorithm operators is performed in a different order compared to other evolutionary algorithms, as shown in Figure 4. First, parents are selected. Then, the mutation, recombination, and evaluation phases are executed in this order. Finally, most fitted individuals are chosen to go to the next generation.

The mutation operator is in the core of DE, since it is the process that inserts random information in the population, thus expanding the search space. Mutation produces a new population of candidate individuals $V = \{v_0, \ldots, v_{pop}\}$, such that $|V| = |X| = pop$. The IBHEA for CRDP applies two different mutation operators, one for each vector of the solution.

The *portions vector* of a candidate individual $v_s \in V$ is generated as the sum of an individual from X to the difference between two other individuals, also from X. This process employs a factor of perturbation $F \in [0;2]$ that weighs the inserted randomness. This factor needs to be set as a parameter for IBHEA. There are many

```
Initialize the population
Evaluate the population
while stopping criterion is not met do
    Parents Selection
    Mutation
    Recombination
    Evaluate the intermediary population
    New population selection
end while
return the best solution found
```

Fig. 4 DE pseudocode

different mutation processes described in the literature, as reported in [16]. However, the most adopted is the *DE/rand/1*, described as follows.

Let X^g denote the IBHEA population at a generation g. Also, let x_α, x_β and x_γ be three distinct random individuals that belong to X^g and let x_j^k denote the k-th gene of an individual $x_j \in X^g$. *DE/best/1* generates a candidate individual v_s that belongs to the generation $g+1$ as show in Equation (3). This operator is applied to each gene separately. If v_s^k results in a value less than 0.5 or greater than 3, then it is rounded to the nearest acceptable value of the interval.

$$v_s^k = x_\alpha^k + F(x_\beta^k - x_\gamma^k) \tag{3}$$

The *ID vector* of the trial individual v_s is generated by a different operator. Each gene $k \in \{1, \dots, 17\}$ from v_s is generated by randomly selecting one of the k-th gene from solution x_α, x_β or x_γ with the same probability. Thus, each gene of the trial individual also respects its food category and does not compromise the modeling of the solution.

The recombination operator intensifies the search into good solutions by reusing previously successful individuals. For each target individual $x_s \in X$ a trial individual $v_s \in V$ is generated from mutation. Then, an individual u_s, denominated offspring, is generated as

$$u_s^k = \begin{cases} v_s^k, & \text{if} \quad r \leq R_r \\ x_s^k, & \text{if} \quad r > R_r \end{cases}, \quad \forall i \in \{1, \dots, 17\}, r \in [0;1],$$

where u_s^k, v_s^k and x_s^k are the k-th gene of the individuals u_s, v_s, and x_s, respectively. These solutions are inserted into a new population $U = \{u_0, \dots, u_{pop}\}$. The value $r \in [0;1]$ is randomly generated at each iteration of IBHEA. R_r is a parameter required by IBHEA that represents the *recombination ratio*, *i.e.* the probability that an offspring inherits the genes from the trial individual v_s. As r, R_r is in the interval $[0;1]$. When $R_r = 1$, the offspring will be equal to the trial individual v_s^k, and IBHEA does not intensify any solution. On the other hand, when $R_r = 0$, the offspring will be equal to the target vector x_s^k, and IBHEA does not insert any randomness in the population during its evolution process. If $0 < R_r < 1$, the offspring can receive genes from both v_s^k and x_s^k. One can see that this is a crucial parameter for IBHEA, as it controls the exploration/intensification ratio.

As the number of offsprings is the same as the number of individuals in the current generation, the selection operator compares one offspring with one individual from the current population. Then, the individual with the worst fitness is discarded, and the other undergoes to the next generation. This procedure ensures that generation $g+1$ will always be equal or better than the generation g.

2.4 The island-based implementation

The concept of evolution has been considered in solving the optimization problem of different areas by Evolutionary Algorithms (EA) since the 1950s. Commonly these algorithms build a population of candidate solutions to a given problem and apply operators inspired by natural genetic variation and natural selection [14]. Over the years different alternatives to implement EAs have also been proposed in the literature. One of those alternatives is the Island Model (IM) by which it is possible to run EAs in parallel computing environments. An IM is a class of distributed EAs in which the population is divided into multiple sub-populations called islands. Each island run its own EA independently, but periodically they interact through the migration operator [3,21]. Figure 5 represents an island graphically as a group of individuals evaluated separately. The islands exchange solutions between themselves according to topology and a set of rules defined by the user through IM parameters by migration.

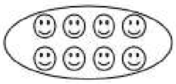


Fig. 5 Island representation.

Between user decisions on implementation of IM is the choice of EAs to be applied in islands. Note that it is possible to apply different EAs between islands, what characterizes the IM as hybrid. This type of implementation can be very beneficial for solving the optimization problem.

The IBHEA is composed by 24 islands with different EAs and different values defined for their respective parameters. The choice for hybrid implementation for IBHEA aims to provide different evolutionary behavior between islands. The DE and GA were the EAs applied in the islands of IBHEA. Thus, islands with DE exploit the population, while islands with GA acts on the population diversification.

A random ring topology arranges the islands of IBHEA. In this case, each island is randomly connected to other two ones. From this topology, in IBHEA the selected solutions for migration in each island are sent to the neighboring island on the right-hand side. Figure 6 illustrates the evolution process of IBHEA, basically composed by three steps. After distributing the population between islands, each one of them is randomly connected to two other ones to receive solutions from just one of them and send it to another. Then, each island evolves its population by its algorithm independently in the Evolution phase. Then, the Migration phase starts, and each island sends and receives the same number of solutions. After each Migration phase, the islands are randomly connected again for a new execution of the Evolution phase, followed by a new Migration phase by the new connections. This sequence of steps is repeated until the stop condition is met.

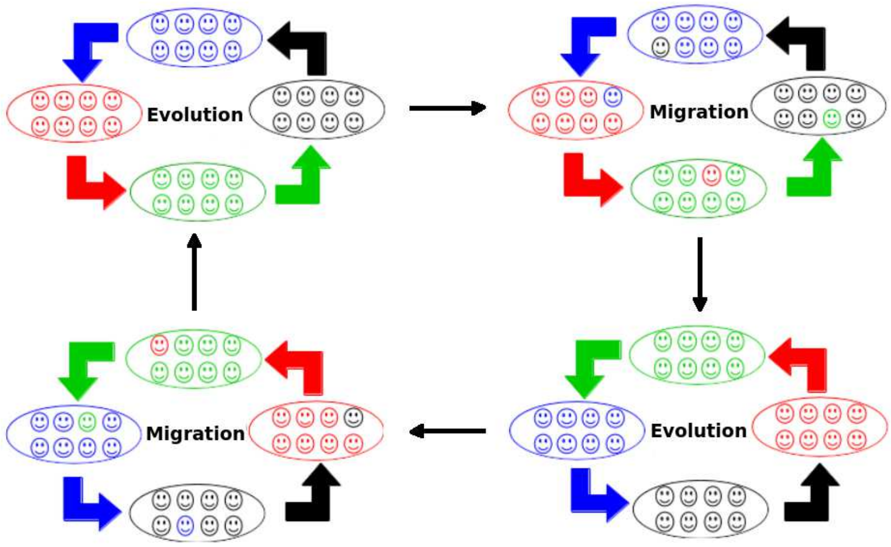


Fig. 6 Sequence of steps in execution of IBHEA.

3 Computational Experiments

The computational experiments were performed in a personal computer with Intel Core i5 CPU 5200U with 2.2GHz and 4GB of RAM, running a Linux operating system. The IBHEA was implemented in C and compiled with the GNU GCC 4.7.3 version. To the EAs in islands were defined 100 generations and 100 individuals. Different parameters setting were set for each of 12 DE and 12 GA applied in the 24 islands according to Table 3.

The GA has two main parameters to be defined by the user: the crossover probability $C_{GA} \in [0,1]$ and the mutation probability $M_{GA} \in [0,1]$. The DE has also two main parameters to be defined by the user: the perturbation factor $F_{DE} \in [0,2]$ and the crossover probability $C_{DE} \in [0,1]$. Table 3 shows the parameter settings for DE and GA applied in islands of IBHEA, defined according to the scenarios applied by Silva *et al.* [19].

The parameters that describe a migration policy are the migration rate (the number of individuals that will migrate) and the migration frequency (the frequency of generations in which the migrations occur). In this work, IBHEA was evaluated under different configurations regardin combinations of these parameters. The migration rate varied from 1% to 20% of solutions, randomly chosen from each island. The migration frequency varied from 1 to 20 generations. Each IBHEA configuration was executed 100 times.

Table 3 Parameter setting for EAs applied in islands of IBHEA.

DE Island	F_{DE}	C_{DE}	**GA Island**	C_{GA}	M_{GA}
1	0.30	0.40	13	0.75	0.05
2	0.30	0.60	14	0.75	0.10
3	0.30	0.80	15	0.75	0.15
4	0.60	0.40	16	0.75	0.20
5	0.60	0.60	17	0.95	0.05
6	0.60	0.80	18	0.95	0.10
7	0.80	0.40	19	0.95	0.15
8	0.80	0.60	20	0.95	0.20
9	0.80	0.80	21	1.00	0.05
10	1.30	0.40	22	1.00	0.10
11	1.30	0.60	23	1.00	0.15
12	1.30	0.80	24	1.00	0.20

Table 4 presents the Minimum, Mean and Maximum values of the objective function of the solutions obtained in the 100 runs of different IBHEA configurations identified in first column. Table 4 indicates that the migration operator in IM can positively contribute to the solution quality of EAs. Table 4 demonstrates that the IBHEA configuration that produced the best set of solutions, according to the Minimum and Mean metrics, was the one in which the migration was performed at each generation of the EAs and which exchanged the largest number of solutions between the islands. The configuration associated to the lowest value for the Maximum metric also presents similar characteristic of IBHEA regarding the application of migration.

The main results of this work are shown in Tables 5 and 6, where it is verified that, for the best parameters set for GA and DE islands the island migrations improves the average calories of the final solutions Notice that the hybrid model implementing GA and DE runs the same number of evaluations of the objective function. IBHEA results achieved better performance when compared to the results present by Silva in [20], which used a DE with island implementation to solve the CRDP.

From Tables 5 and 6 it is possible to state that the goals of the work were achieved comparing with DE and GA. The proposed implementation using islands shows better results than the original model in 100% of the tests. One of the major and most likely reasons for the improvement in the quality of the results arises from the diversity of the population introduced by the migration between islands with different evolutionary algorithms. Table 7 shows an example of a daily diet generated by IBHEA, in which a large diversity of foods and drinks in all meals can be noticed.

Another relevant feature of IBHEA is its applicability to real-world contexts since it can solve CRPD with short execution time. Moreover, IBHEA can solve many instances of CRPD in seconds, thus generating different diets at each execution. The great diversity of foods and drinks makes diets easier to be followed, so the taste will not be tired with repetitive meals, thus becoming more comfortable with losing weight.

Table 4 Results obtained in 100 runs for different configurations of IBHEA identified in first column according to their migration frequency and migration rate values.

(Migration frequency/migration rate(%))	Minimum	Mean	Maximum
(1/1)	1424.56	1468.03	1490.21
(1/5)	1427.24	1454.76	1486.02
(1/10)	1419.26	1448.87	1476.93
(1/15)	1416.69	1447.92	**1470.78**
(1/20)	**1407.66**	**1444.01**	1472.58
(5/1)	1438.37	1483.89	1519.26
(5/5)	1419.95	1469.58	1505.01
(5/10)	1422.48	1466.59	1494.37
(5/15)	1419.43	1464.97	1485.59
(5/20)	1430.04	1462.65	1490.32
(10/1)	1424.03	1490.35	1545.22
(10/5)	1425.74	1478.06	1521.72
(10/10)	1409.04	1474.12	1509.82
(10/15)	1415.03	1474.21	1519.49
(10/20)	1428.40	1470.88	1506.06
(15/1)	1436.98	1492.30	1544.97
(15/5)	1435.20	1484.92	1547.35
(15/10)	1419.54	1478.14	1521.11
(15/15)	1433.95	1477.32	1517.95
(15/20)	1422.39	1475.44	1517.68
(20/1)	1423.08	1496.98	1609.26
(20/5)	1425.79	1488.26	1560.74
(20/10)	1428.68	1484.57	1539.71
(20/15)	1423.89	1482.40	1535.03
(20/20)	1433.14	1481.78	1525.89

Table 5 Mean - Island-based model × DE model.

F_{DE}	C_{DE}	**DE result**	**Difference (%)**
0.3	0.4	1505.64	-61.64
0.3	0.6	1492.04	-48.04
0.3	0.8	1485.73	-41.73
0.6	0.4	1506.92	-62.92
0.6	0.6	1499.66	-55.66
0.6	0.8	1483.31	-39.31
0.8	0.4	1512.12	-68.12
0.8	0.6	1500.36	-56.36
0.8	0.8	1507.57	-63.57
1.3	0.4	1521.05	-77.05
1.3	0.6	1510.33	-66.33
1.3	0.8	1510.02	-66.02

4 Conclusions

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

The use of the hybrid island-based model achieved significant results since it was able to reduce the amount of calories of the diets without violating the restrictions in

Table 6 Mean - Island-based model x GA model.

C_{GA}	M_{GA}	**GA result**	**Difference (%)**
0.75	0.05	1575.70	-131.7
0.75	0.10	1531.17	-87.17
0.75	0.15	1512.37	-68.37
0.75	0.20	1502.37	-58.37
0.95	0.05	1553.95	-109.95
0.95	0.10	1515.55	-71.55
0.95	0.15	1501.73	-57.73
0.95	0.20	1507.23	-63.23
1.00	0.05	1550.50	-106.5
1.00	0.10	1507.14	-63.14
1.00	0.15	1502.74	-58.74
1.00	0.20	1503.98	-59.98

Table 7 Diet Example

1444.72kcal	
breakfast	
coconut water	1.492
banana	1.638
whole wheat bread	0.743
snack	
lime orange	1.163
lunch	
sweet potato	1.170
split bean	0.872
alfavaca	1.877
white cabbage	1.491
crab	0.868
lemon suice	2.374
snack	
lemon clove suice	1.501
green corn cream	0.623
dinner	
cooked potato	1.383
bean	1.862
broccoli	1.609
crab	0.790
supper	
banana	2.079

Nutrition Facts	
Proteins	75.6
Carbohydrate	300.0
Food Fiber	57.8
Calcium	1613.2
Magnesium	634.9
Manganese	4.9
Phosphorus	1156.0
Iron	16.9
Sodium	1020.9
Zinc	16.7

the majority of the scenarios. This work can be applied in real situations, as it can generate a lot of diets in a short time.

For future works, we propose to develop a multi-objective version of CRDP, to minimize, in addition to the number of calories, the financial cost, or the sodium consumption, while maximizing the amount of fiber and protein, for example. Another approach is to turn this work a configurable toll for use in specific diets for diabetic or hypertensive patients.

Compliance with Ethical Standards

Funding

The authors acknowledge the support of Graduate Program in Computational Modeling at Federal University of Juiz de Fora (UFJF) and the Brazilian funding agencies CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico (grant number 429639/2016), FAPEMIG (grant number APQ-00334/18), and CAPES -Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, finance code 001.

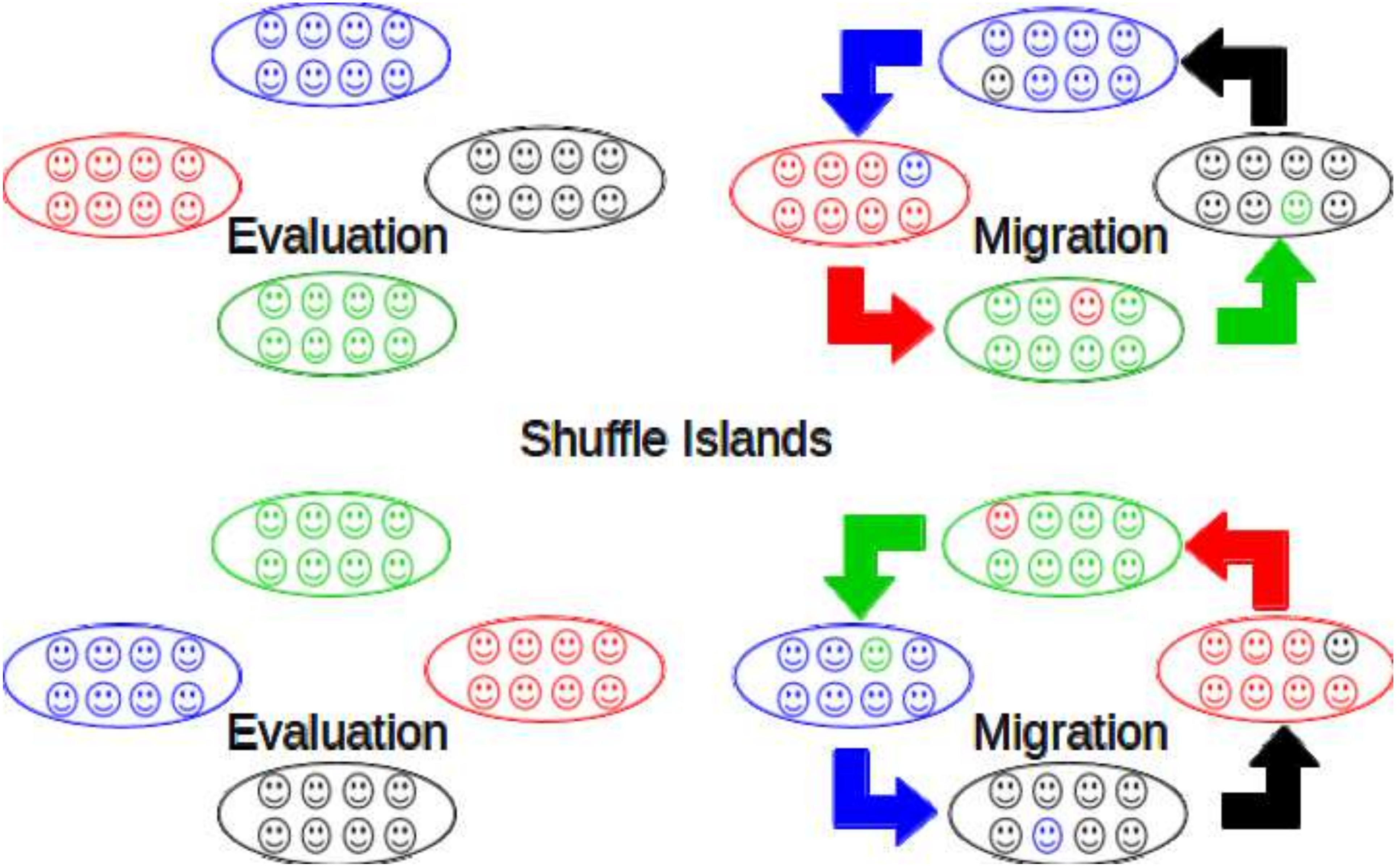
Conflict of insterest/Competing Interest

The authors declare that they have no conflicts/competing of interest.

References

1. Anderson, J. W., Konz, E. C., Frederich, R. C., Wood, C. L.: Long-term weight-loss maintenance: a meta-analysis of US studies. The American Journal of Clinical Nutrition, 74(5), 579–584. (2001)
2. Babu, B. V., Angira, R.: Modified differential evolution (MDE) for optimization of non-linear chemical processes. Computers & Chemical Engineering, 30(6), 989–1002. (2006)
3. Cantú-Paz, E.: A Survey of Parallel Genetic Algorithms. CALCULATEURS PARALLELES, reseaux et systems repartis, 10(2), 141–171. (1998)
4. Chowdhury, A., Rakshit, P., Konar, A.: Protein-protein interaction network prediction using stochastic learning automata induced differential evolution. Applied Soft Computing 49, 699–724 (2016)
5. 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)
6. Curioni, C., Lourenco, P.: Long-term weight loss after diet and exercise: a systematic review. International Journal of Obesity, 29(10), 1168–1174. (2005)
7. Das, S., Mullick, S. S., Suganthan, P. N.: Recent advances in differential evolution–an updated survey. Swarm and Evolutionary Computation, 27, 1–30. (2016)
8. Datta, R., Deb, K.: Evolutionary Constrained Optimization, Infosys Science Foundation Series in Applied Sciences and Engineering, 1st Ed. Springer India (2015)
9. Heerspink, H. J. L., Holtkamp, F. A., Parving, H. H., Navis, G. J., Lewis, J. B., Ritz, E., De Graeff, P. A., De Zeeuw, D.: Moderation of dietary sodium potentiates the renal and cardiovascular protective effects of angiotensin receptor blockers. Kidney international, 82(3), 330–337. (2012)
10. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, Cambridge, MA, USA. (1992)
11. Ladabaum, U., Mannalithara, A., Myer, P.A., Singh, G.: Obesity, abdominal obesity, physical activity, and caloric intake in us adults: 1988 to 2010. The American Journal of Medicine, 127(8), 717–727. (2014)
12. Liao, T.W.: Two hybrid differential evolution algorithms for engineering design optimization. Applied Soft Computing 10(4), 1188–1199 (2010)
13. 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 – TBCA (2011),
14. Mitchell, M.: An Introduction to Genetic Algorithms. A Bradford book, Bradford Books. (1998)
15. Pasquali, R. et. al.: 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). (2000)
16. Price, K., Storn, R.M., Lampinen, J.A.: Differential evolution: a practical approach to global optimization. Springer Science & Business Media. (2006)

17. Ravi, V., Kurniawan, H., Thai, P.N.K., Kumar, P.R.: Soft computing system for bank performance prediction. Applied soft computing 8(1), 305–315 (2008)
18. Rolfes, S.R., Pinna, K., Whitney, E.: Understanding normal and clinical nutrition, chap. An overview of nutrition, 10 edn., Cengage Learning, 3–34. (2014)
19. Silva, J.G.R., Carvalho, I.A., Loureiro, M.M.S., da Fonseca Vieira, V., Xavier, C.R.: Developing tasty calorie restricted diets using a differential evolution algorithm. In: ICCSA 2016. 171–186. Springer (2016)
20. Silva, João Gabriel Rocha, Carvalho, Iago Augusto, Goliatt, Leonardo, da Fonseca Vieira, Vinícius, Xavier, Carolina Ribeiro.: A Differential Evolution Algorithm for Computing Caloric-Restricted Diets - Island-Based Model. In: ICCSA 2017. Springer, 385–400. (2017)
21. Skolicki, Z.M.: An Analysis of Island Models in Evolutionary Computation. Ph.D. thesis, Fairfax, VA, USA. (2007)
22. Stigler, G.J.: The cost of subsistence. Journal of farm economics, 27(2), 303–314. (1945)
23. Wang, L., Zeng, Y., Chen, T.: Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. Expert Systems with Applications 42(2), 855–863 (2015)


Evaluation
Migration
Shuffle Islands
Evaluation
Migration

Figures

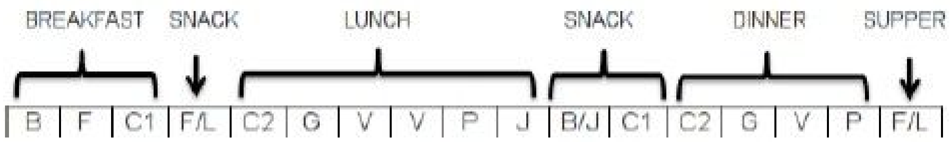
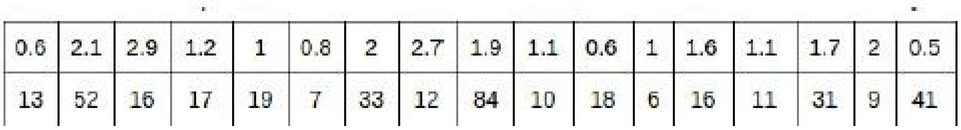


Figure 1

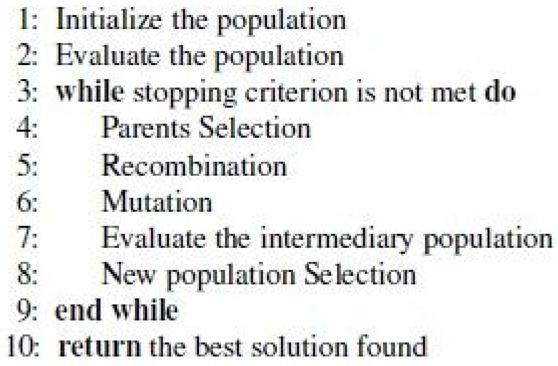
Solution encoding for CRPD: 6 ordered different meals, each one compose by a specific number and groups of food products.



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

Figure 2

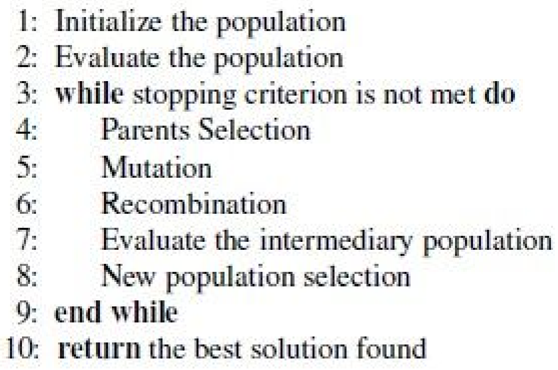
Representation of an IBHEA candidate solution for CRDP.



```
Initialize the population
Evaluate the population
while stopping criterion is not met do
    Parents Selection
    Recombination
    Mutation
    Evaluate the intermediary population
    New population Selection
end while
return the best solution found
```

Figure 3

GA pseudocode



```
Initialize the population
Evaluate the population
while stopping criterion is not met do
    Parents Selection
    Mutation
    Recombination
    Evaluate the intermediary population
    New population selection
end while
return the best solution found
```

Figure 4

DE pseudocode

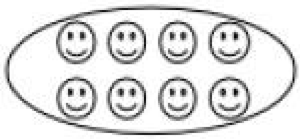


Figure 5

Island representation

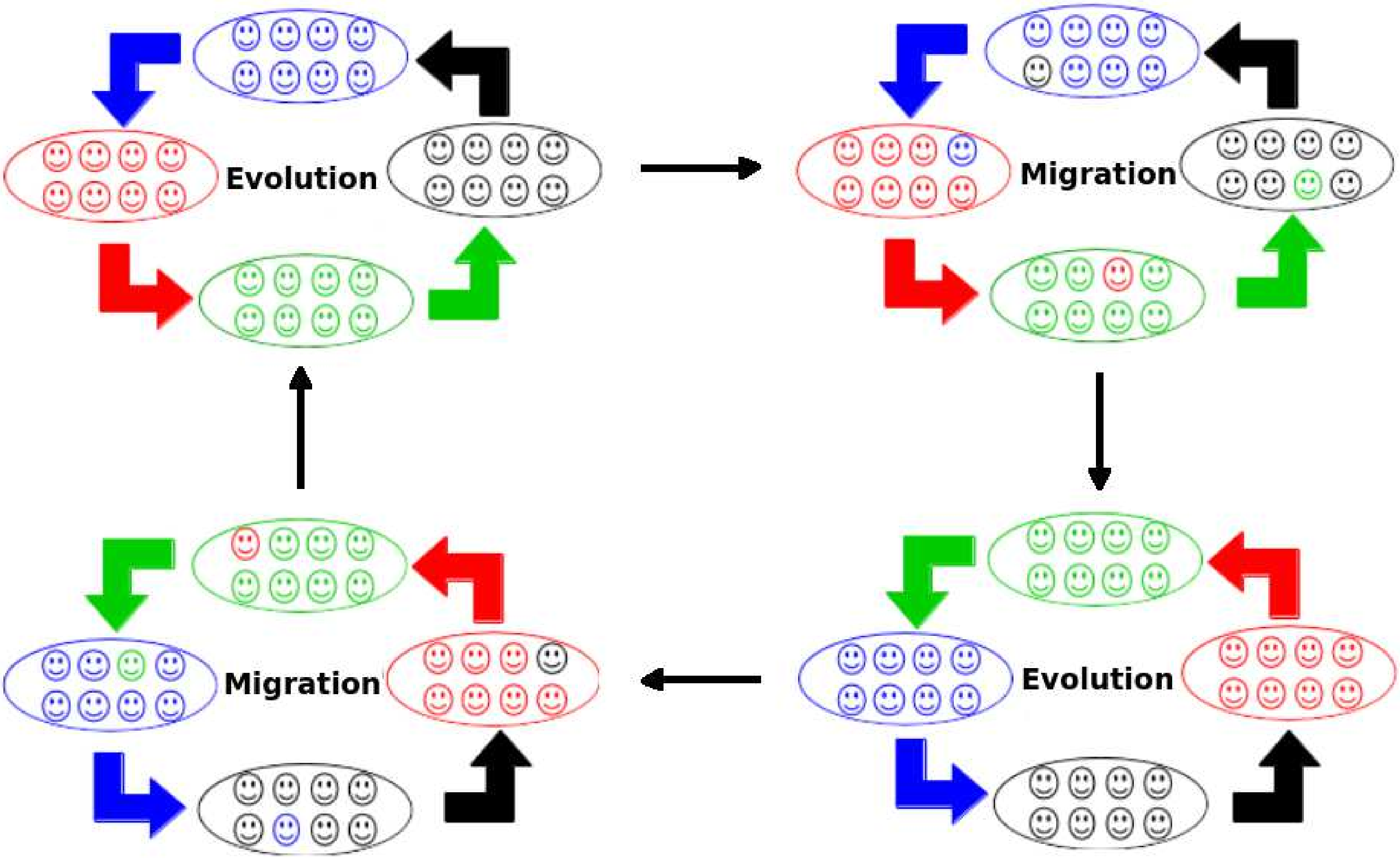


Figure 6

Sequence of steps in execution of IBHEA

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- ilhasexecucaomigracaolegenda.png