

Creación de dietas mediante AEs

Santiago Pereira

Facultad de Ingeniería, Universidad de la Republica
Montevideo, Uruguay
santiagopereira0812@gmail.com

Jorge Miguel Machado

Facultad de Ingeniería, Universidad de la Republica
Montevideo, Uruguay
jorge.machado.ottonelli@gmail.com

Abstract—In the pursuit of personalized and balanced dietary planning, this study explores the application of Evolutionary Algorithms (EAs) for generating customized meal plans. Addressing the complex task of satisfying daily and weekly nutritional needs while ensuring dietary variety, the proposed EA-based approach optimizes menus for individuals. The algorithm integrates considerations of caloric intake, proteins, carbohydrates, and fats, balancing the nutritional requirements with a strategic selection of foods suitable for specific meal times. The results demonstrate the efficiency of the proposed EA in producing nutritionally balanced and diverse meal plans tailored to individual preferences and dietary restrictions.

Index Terms—evolutionary algorithms, personalized diet planning, nutritional optimization, genetic algorithms, dietary restrictions, optimization techniques.

I. INTRODUCCIÓN

En una época en la que la conciencia de la salud y la nutrición es cada vez más prominente, la planificación de dietas equilibradas y personalizadas se ha convertido en un objetivo fundamental para individuos y profesionales de la salud por igual. Uno de los desafíos más comunes y cruciales en este contexto es la generación de una dieta que satisfaga las necesidades nutricionales y energéticas de una persona, en términos de una cantidad de calorías diarias, y una cantidad de macronutrientes determinados.

Al planificar dietas, la diversidad en la selección de alimentos se convierte en un criterio esencial. No solo se trata de cumplir con las necesidades nutricionales y energéticas individuales, sino también de crear menús variados.

Este problema no solo se relaciona con la salud y el bienestar de las personas, sino que también tiene un impacto directo en la prevención de enfermedades crónicas y en la promoción de un estilo de vida saludable.

II. ANÁLISIS DEL PROBLEMA

A. Descripción del problema

Las personas que desean mantener una vida saludable a menudo se cuestionan como obtener un menú para cada una de las comidas del día (desayuno, almuerzo, merienda y cena), de manera que se acerque a su cantidad de calorías diarias requeridas, a una cantidad de macronutrientes determinados y que además sea una dieta balanceada.

Se desea obtener una dieta semanal que poder seguir en busca de satisfacer unos requisitos nutricionales dados por la persona a la que va dirigida. Dicha dieta indica cual es el menú para cada una de las 4 comidas diarias, y además de cumplir

con los requisitos nutricionales, se busca que sea balanceada y que evite la repetición de alimentos a lo largo de la semana. Para fijar ideas definimos los siguientes conceptos:

- *Calorías diarias*: cantidad de calorías a consumir en un día.
- *Proteínas diarias*: cantidad de proteínas a consumir en un día.
- *Carbohidratos diarios*: cantidad de carbohidratos a consumir en un día.
- *Grasas diarias*: cantidad de grasas a consumir en un día.
- *Ranking de comida en el día*: beneficio de comer una comida según el horario.

De igual forma, se considera la variedad en la selección de alimentos en la planificación de dietas personalizadas, buscando evitar la repetición excesiva de comidas a lo largo de la dieta para proporcionar una experiencia diversa y agradable. Esta variedad no solo mejora la satisfacción, sino que también asegura una ingesta equilibrada de diferentes nutrientes y minimiza el riesgo de deficiencias nutricionales asociadas con la monotonía dietética.

B. Modelado del problema

El problema a resolver es la elección (posiblemente con repetición) de alimentos provenientes de un conjunto de datos [5] que contiene una amplia variedad de estos. A esta elección de alimentos se la debe distribuir a lo largo de la semana, de forma de cumplir con los requisitos nutricionales diarios de la persona a la que va dirigida la dieta. Además, se deberán penalizar aquellas dietas semanales que cuenten con alimentos repetidos, ya sea en la semana o en un mismo día.

Se trabaja con una representación matricial para simplificar y visualizar mejor el problema, donde la cantidad de columnas indican la cantidad de días que se consideran en la solución (típicamente 7 días, correspondientes a una semana), las filas representan cada una de las comidas en un día (típicamente 4: desayuno, almuerzo, merienda y cena), y además en cada celda de la matriz se alojará hasta un cierto número de alimentos correspondientes a una comida (en nuestro estudio un máximo de 4 alimentos por comida). Cabe destacar que si bien se toman como estándar los parámetros mencionado, todos estos son configurables en base a los gustos de la persona a la que va dirigida la solución. A continuación se muestra la forma de una solución:

Día	lun.	mar.	miérc.	juev.	vier.	sáb.	dom.
Desayuno	[1,2,3,4]						
Almuerzo							
Merienda							
Cena							

TABLE I

EJEMPLO DE LA REPRESENTACIÓN MATRICIAL DE UNA DIETA

C. Formulación matemática

Se definen a continuación los objetivos nutricionales diarios a los que se aproximará la dieta semanal:

- kc = calorías objetivo.
- p = proteínas objetivo.
- hc = carbohidratos objetivo.
- g = grasas objetivo.

Además se definen las siguientes funciones:

- $pond_horario(c, c.h)$ retorna que tan bien se ajusta una comida a un horario.
- $cant_rep_semana(c, indiv)$ retorna la cantidad de veces que aparece 'c' en la semana.
- $cant_rep_dia(c, d, indiv)$ retorna la cantidad de veces que aparece 'c' en el día 'd'.

Donde:

- c = comida elegida en un horario
- h = horario en que se come.

En base a lo detallado anteriormente, se define el problema que consta en: Por un lado minimizar la diferencia en cuanto a las calorías, los macronutrientes y las ponderaciones horarias entre el valor objetivo deseado y lo brindado por la dieta. Y por otro lado maximizar la variedad de alimentos tanto diariamente como semanalmente, lo que implica minimizar la cantidad de repeticiones de un alimento a lo largo del día o la semana.

Formalizando, se desean minimizar las ecuaciones (1) y (2) dadas a continuación:

$$f1 = \|f_col(col1), f_col(col2), \dots, f_col(col7)\| \quad (1)$$

$$f2 = \delta * cant_rep_dia(c, d, indiv) + \sigma * cant_rep_semana(c, indiv) \quad (2)$$

, donde f_col está definida de la siguiente forma:

$$f_col = \alpha * |kc - c.kcal| + \beta * [|p - c.p| + |hc - c.hc| + |g - c.g|] + \gamma * (1 - pond_horario(c, c.h))$$

y los hiperparámetros α , β y γ representan los pesos que se le dan a cada característica nutricional.

III. JUSTIFICACIÓN DE UTILIZAR ALGORITMOS EVOLUTIVOS

El uso de AE esta justificado por la naturaleza compleja y multiobjetivo del problema a resolver. El cual implica cumplir con las necesidades diarias y semanales de calorías, así como asegurar un equilibrio adecuado entre proteínas, carbohidratos y grasas.

De forma adicional, el problema presenta restricciones y objetivos múltiples conflictivos, ya que es necesario equilibrar la necesidad de cumplir con los requisitos diarios y semanales, mientras se evita la repetición excesiva de alimentos en el menú. Los operadores evolutivos, como la mutación y el cruzamiento, permiten introducir variabilidad y diversidad en la población de soluciones, permitiendo encontrar soluciones óptimas y diversas.

Por otra parte, el proceso de planificación de dietas no es estático; las necesidades nutricionales pueden cambiar con el tiempo, y es necesario adaptarse dinámicamente a estos cambios.

IV. ESTRATEGIA DE RESOLUCIÓN

Se resuelve este problema aplicando Algoritmos Evolutivos, mas precisamente, utilizando el algoritmo NSGA-II implementado en JMetalPy [3]. A continuación se describen los principales elementos de la solución.

A. Método de inicialización

Para la población inicial se toman alimentos del dataset [5] de forma aleatoria, de 100 individuos. Para la asignación aleatoria se utilizó la semilla propia de JMetalPy [3], que cambia en cada ejecución.

B. Representación de las soluciones

Cada individuo se representará como una lista de enteros de largo fijo igual al producto entre la cantidad de días \times cantidad de comidas \times cantidad de alimentos, cada lista es de la forma:

$$|D_{1,1}| \dots |D_{1,n}| A_{1,1}| \dots |A_{1,n}| M_{1,1}| \dots |M_{1,n}| C_{1,1}| \dots |C_{1,n}| D_{2,1}| \dots$$

, donde:

- $D_{i,j}$: representa el identificador del alimento j a consumir durante el desayuno del día i .
- $A_{i,j}$: representa el identificador del alimento j a consumir durante el almuerzo del día i .
- $M_{i,j}$: representa el identificador del alimento j a consumir durante la merienda del día i .
- $C_{i,j}$: representa el identificador del alimento j a consumir durante el cena del día i .
- $i \in [1, cantidad_de_días]$
- $j \in [1, n]$, con $n = cantidad_de_alimentos_por_comida$

Además, las variables $D_{i,j}$, $A_{i,j}$, $M_{i,j}$ y $C_{i,j}$ pueden tomar el valor -1 , lo que representa una posición sin un alimento asignado. Esto con el fin de permitir que se generen soluciones con menos alimentos que el largo fijo de la solución.

C. Condición de parada

La condición de parada será el alcanzar una cantidad máxima de generaciones. Esta condición es necesaria para que la ejecución del algoritmo termine.

D. Operadores evolutivos

- **Mutación:** Selecciono una celda aleatoria de un individuo y cambio su valor, reemplazándolo por otro alimento disponible.

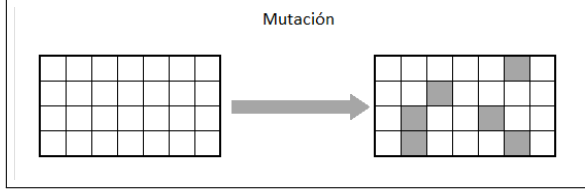


Fig. 1. Mutación de un individuo

- **Cruzamiento:** a continuación se presentan distintos tipos de cruzamiento que serán probados durante las diferentes configuraciones del algoritmo:
 - Para cada columna, seleccionar con probabilidad p la columna del mismo día del padre 1, y con probabilidad $(1-p)$ la columna del mismo día del padre 2.

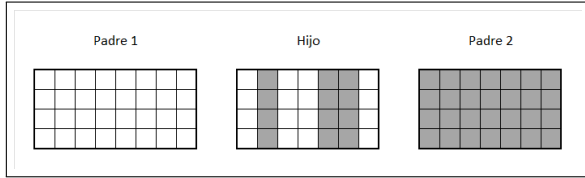


Fig. 2. Cruzamiento por columna

- Para cada fila, seleccionar con probabilidad p la fila del mismo horario del padre 1, y con probabilidad $(1-p)$ la fila del mismo horario del padre 2.

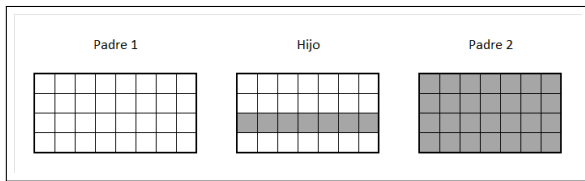


Fig. 3. Cruzamiento por fila

E. Operador de selección

Se utiliza el operador de selección "BestSolutionSelection" provisto por las librerías de JMetalPy [3], sin modificaciones. El mismo selecciona las soluciones más óptimas de la población actual, priorizando aquellas con los valores objetivos más bajos (dado que la naturaleza del problema propuesto es de minimización).

F. Operadores de evaluación

Se implementan las dos funciones objetivo (1) y (2) descritas anteriormente.

G. Restricciones

Se restringe en el problema la cantidad de alimentos que componen una comida a un máximo de cuatro. Adicionalmente, se tiene únicamente en cuenta que un día se distribuye en cuatro comidas.

Si bien es posible modificar los valores de configuración del algoritmo para aceptar un rango mayor de alimentos y comidas diarias, esto implica la creación de un nuevo dataset que pueda soportar una mayor cantidad de ponderaciones horarias.

V. PROPUESTA DE EVALUACIÓN EXPERIMENTAL

A. Generación de instancias

Se diseñaron dos instancias para realizar calibraciones, las cuales están inspiradas en dos personas de diferentes características, y con diferentes objetivos. De esta manera buscamos una dieta diaria en torno a las siguientes características:

Característica	Instancia 1	Instancia 2
Calorías	2000.0 kc	1704.0 g
Proteínas	153.0 g	128.0 g
Carbohidratos	205.0 g	170.0 g
Grasas	68.0 g	57.0 g

TABLE II

REQUERIMIENTOS DE CADA INSTANCIA

De forma adicional, se fijan los hiperparámetros definidos de la siguiente forma:

- $\alpha = 1.0$
- $\beta = 1.0$
- $\gamma = 0.75$
- $\delta = 100.0$
- $\sigma = 10.0$

Las características del PC ha utilizar para realizar estas pruebas son las siguientes:

- CPU: Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz
- RAM: 16.0 GB @ 2667 MHz
- SO: Windows 11 Pro 22H2

B. Calibración

Los parámetros del algoritmo seleccionados para calibrar son el Tipo de Cruzamiento, Probabilidad de Cruzamiento y Probabilidad de Mutación. En la tabla III se presentan los valores elegidos para los parámetros a calibrar. En todos los casos se efectúan 30 ejecuciones independientes y se evalúan sus resultado analizando el valor de las métricas de Hipervolumen en base al punto de referencia con peor valor obtenido de las funciones objetivo en todas las ejecuciones.

AE	Tipo Cruzamiento	Prob. Cruzamiento	Prob. Mutación
1	Por Columna	0.3	0.1
2	Por Columna	0.3	0.01
3	Por Columna	0.3	0.005
4	Por Columna	0.5	0.1
5	Por Columna	0.5	0.01
6	Por Columna	0.5	0.005
7	Por Columna	0.8	0.1
8	Por Columna	0.8	0.01
9	Por Columna	0.8	0.005
10	Por Fila	0.3	0.1
11	Por Fila	0.3	0.01
12	Por Fila	0.3	0.005
13	Por Fila	0.5	0.1
14	Por Fila	0.5	0.01
15	Por Fila	0.5	0.005
16	Por Fila	0.8	0.1
17	Por Fila	0.8	0.01
18	Por Fila	0.8	0.005

TABLE III

CONFIGURACIÓN DE LOS PARÁMETROS DEL ALGORITMO

Como se observa en los resultados obtenidos en las tablas IV y V, se obtiene un valor de hipervolumen considerablemente más bajo en aquellas configuraciones que presentan una probabilidad de mutación del 10%. Esto no es algo particular de una instancia individual, sino que se mantiene en ambas instancias presentadas.

AE	Media de Hipervolumen	Desviación Estándar
1	182436.497	16678.493
2	257581.449	10004.001
3	256515.423	12661.742
4	180247.477	20821.807
5	261369.979	10339.449
6	252488.461	16122.540
7	196792.741	19911.157
8	257359.802	12144.462
9	255193.462	19696.957
10	185564.542	19254.819
11	257409.514	11384.541
12	255862.311	10651.541
13	188918.477	16520.909
14	256227.259	10386.610
15	257425.702	12245.196
16	185021.378	21818.554
17	258149.349	11783.601
18	254805.897	11872.761

TABLE IV

HIPERVOLUMEN DE INSTANCIA 1

Adicionalmente, se observa que las aquellas configuraciones que presentan un 1% de probabilidad de mutación suelen presentar valores bajos en la desviación estándar de los hipervolumenes, cuando se comparan con el resto de configuraciones con probabilidades de mutación diferente.

Se resalta además que para la instancia 1 la configuración con mayor media de hipervolumen es la AE 5. Dicha configuración también obtiene valores altos para la instancia 2, siendo solamente superada por las configuraciones AE 2 y AE 14, siendo esta última la que mayor valor tiene en esta segunda instancia.

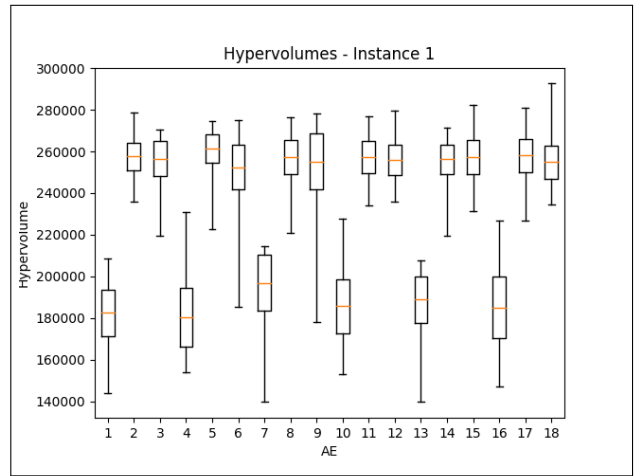


Fig. 4. Boxplot de los hipervolumenes obtenidos en la instancia 1.

AE	Media de Hipervolumen	Desviación Estándar
1	213785.683	24449.738
2	288489.967	8506.793
3	284155.613	24697.097
4	198163.763	18301.253
5	288152.137	15812.217
6	280848.208	14454.271
7	204315.093	25588.174
8	284388.356	12758.854
9	282153.947	20686.297
10	196274.627	23404.402
11	284876.715	6084.816
12	281895.440	19329.63
13	210342.056	22367.169
14	289030.709	6610.640
15	283266.435	19429.432
16	204904.670	23444.205
17	285826.956	6639.281
18	277163.861	30141.490

TABLE V

HIPERVOLUMEN DE INSTANCIA 2

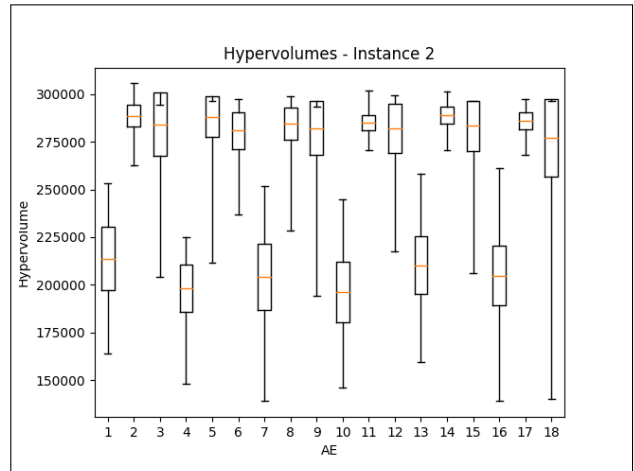


Fig. 5. Boxplot de los hipervolumenes obtenidos en la instancia 2.

C. Test de normalidad sobre las muestras

En todos los casos se aplicaron los tests de Anderson-Darling y Shapiro-Wilk para comprobar la normalidad de

los datos obtenidos y así poder compararlos. Esto se realizó con la biblioteca de Python, SciPy [8], con los paquetes stats.anderson y stats.shapiro, los cuales proveen el test directamente y retornan si la distribución es normal o no con un nivel de significancia de $\alpha = 5\%$.

Se aplicaron los tests a los resultados de las 30 ejecuciones para cada una de las 18 configuraciones definidas para ambas instancias, obteniendo los siguientes resultados:

AE	Estadístico	Resultado al 5%
1	0.499	True
2	0.144	True
3	0.725	False
4	0.339	True
5	0.588	True
6	1.643	False
7	0.791	False
8	0.780	False
9	1.664	False
10	0.157	True
11	0.296	True
12	0.377	True
13	0.865	False
14	0.846	False
15	0.260	True
16	0.388	True
17	0.407	True
18	0.370	True

TABLE VI

TEST DE NORMALIDAD DE ANDERSON-DARLING PARA LA INSTANCIA 1.

AE	Estadístico	Resultado al 5%
1	0.485	True
2	0.526	True
3	2.483	False
4	0.415	True
5	2.512	False
6	0.838	False
7	0.424	True
8	1.893	False
9	2.835	False
10	0.401	True
11	0.361	True
12	1.568	False
13	0.220	True
14	2.150	False
15	2.426	False
16	0.472	True
17	0.387	True
18	2.592	False

TABLE VII

TEST DE NORMALIDAD DE ANDERSON-DARLING PARA LA INSTANCIA 2.

Nivel de Significancia	Valor Crítico
15%	0.521
10%	0.593
5%	0.712
2.5%	0.83
1.0%	0.988

TABLE VIII

NIVELES DE SIGNIFICANCIA Y SUS VALORES CRÍTICOS DEL TEST DE NORMALIDAD DE ANDERSON-DARLING PARA UNA DISTRIBUCIÓN NORMAL.

AE	Estadístico	p-valor	Resultado
1	0.943	0.111	True
2	0.989	0.988	True
3	0.929	0.0477	False
4	0.972	0.608	True
5	0.907	0.0126	False
6	0.799	6.528e-05	False
7	0.918	0.0236	False
8	0.917	0.0227	False
9	0.813	1.143e-04	False
10	0.981	0.851	True
11	0.967	0.480	True
12	0.970	0.553	True
13	0.908	0.013	False
14	0.878	0.002	False
15	0.979	0.801	True
16	0.958	0.278	True
17	0.964	0.402	True
18	0.951	0.181	True

TABLE IX

TEST DE NORMALIDAD SHAPIRO-WILK PARA LA INSTANCIA 1.

AE	Estadístico	p-valor	Resultado
1	0.953	0.206	True
2	0.964	0.385	True
3	0.782	3.170e-05	False
4	0.949	0.161	True
5	0.665	5.084e-07	False
6	0.905	0.011	False
7	0.966	0.427	True
8	0.733	4.885e-06	False
9	0.709	2.146e-06	False
10	0.965	0.402	True
11	0.974	0.656	True
12	0.850	0.000	False
13	0.984	0.935	True
14	0.706	1.973e-06	False
15	0.767	1.758e-05	False
16	0.961	0.326	True
17	0.968	0.499	True
18	0.691	1.171e-06	False

TABLE X

TEST DE NORMALIDAD SHAPIRO-WILK PARA LA INSTANCIA 2.

Dados los resultados de los test de normalidad, se rechaza la hipótesis nula, y se concluye que los resultados no siguen una distribución normal.

D. Elección de la mejor configuración

Se aplica un test de rango, en particular se usa el Test de Friedman en base a los hipervolumenes obtenidos anteriormente con el fin de obtener la mejor configuración del algoritmo entre las 18 posibles.

Al observar la tabla XI vemos que la configuración mejor rankeada para la primera instancia se corresponde con el AE 5, mientras para la segunda instancia se corresponde con el AE 14.

Si hacemos un análisis global (para ambas instancias en conjunto) del ranking de configuraciones, notamos que los primeros puestos del ranking suelen estar ocupados por aquellas configuraciones cuya probabilidad de mutación es del 1%, al igual de como se observó previamente con los altos valores de media de hipervolumen.

De manera adicional, se puede identificar que la configuración con mejor ranking global es la AE 2, la cual presenta la segunda posición del ranking en ambas instancias.

AE	Instancia 1	Instancia 2
1	17	13
2	2	2
3	9	11
4	18	17
5	1	5
6	12	7
7	13	16
8	6	6
9	11	10
10	15	18
11	4	4
12	8	9
13	14	14
14	7	1
15	5	8
16	16	15
17	3	3
18	10	12

TABLE XI

RANKING DE LAS CONFIGURACIONES PARA LAS INSTANCIAS 1 Y 2.

Al analizar la tabla XII, vemos que los *p-valores* obtenidos del test de Friedman son menores que el nivel de significancia establecido ($\alpha = 0.05$). Cuando esto sucede, nos indica que hay diferencias estadísticamente significativas entre al menos dos configuraciones, y por lo tanto, nos proporciona evidencia para rechazar la hipótesis nula de que todas las configuraciones son iguales.

Instancia	Estadístico	p-valor
1	450.702	4.418e-85
2	400.856	1.228e-74

TABLE XII

RESULTADOS DEL TEST NO PARAMÉTRICO DE FRIEDMAN PARA AMBAS INSTANCIAS

Al realizar esta observación global de los resultados, y los comentarios anteriores, se finaliza el proceso de calibración seleccionando como mejor configuración al algoritmo AE 2, indicado en la tabla XIII.

AE	Tipo Cruzamiento	Prob. Cruzamiento	Prob. Mutación
2	Por Columna	0.3	0.01

TABLE XIII

VALORES DE LA CONFIGURACIÓN SELECCIONADA

VI. EVALUACIÓN EXPERIMENTAL

A. Instancias de Evaluación

Se definen a continuación dos modelos de personas referenciados en la tabla XIV con características bien distintas, donde sus requerimientos funcionales se usaran como instancias para realizar la evaluación.

A partir de estas, se generan sus respectivos requerimientos nutricionales utilizando la fórmula de Harris-Benefict [9] (Instancia 1E e Instancia 2E) referenciados en la tabla XV para evaluar la performance de la configuración seleccionada.

Ind	Sexo	Edad	Altura	Peso	Actividad	Objetivo
1	H	35	175 cm	75 kg	Ligera	Manten. Peso
2	M	20	155 cm	60 kg	Muy Activo	Ganar Peso

TABLE XIV

INDIVIDUOS A REALIZAR LA DIETA

Características	Instancia 1E	Instancia 2E
Calorías	2385.0 kc	2888.0 kc
Proteínas	179.0 g	217.0 g
Carbohidratos	238.0 g	325.0 g
Grasas	79.0 g	80.0 g

TABLE XV

INSTANCIAS A UTILIZAR PARA EVALUAR LA CONFIGURACIÓN

B. Reporte de resultados

Cada instancia es evaluada efectuando 30 ejecuciones. Se selecciona la métrica de Hipervolumen, reportando su media y desviación estándar. Adicionalmente, se realiza una comparación de resultados en cuanto a los valores obtenidos en las funciones objetivo y tiempo de ejecución con el algoritmo ávido diseñado, y además se muestran los porcentajes de mejora. Al igual que en la calibración, se realizan las ejecuciones utilizando el mismo PC, con las siguientes características:

- CPU: Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz
- RAM: 16.0 GB @ 2667 MHz
- SO: Windows 11 Pro 22H2

Algoritmo	hv_min	hv_media	hv_max	hv_std
AE 2	215905.9	241350.48	266532.15	12504.87

TABLE XVI

VALORES DE HIPERVOLUMEN PARA LA INSTANCIA 1E

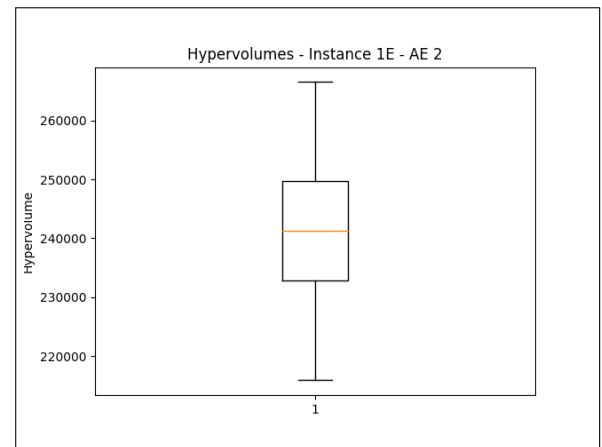


Fig. 6. Boxplot de los hipervolumenes obtenidos para la instancia 1E utilizando el AE 2.

Algoritmo	hv_min	hv_media	hv_max	hv_std
AE 2	11823.09	26915.42	36492.71	7142.83

TABLE XVII

VALORES DE HIPERVOLUMEN PARA LA INSTANCIA 2E

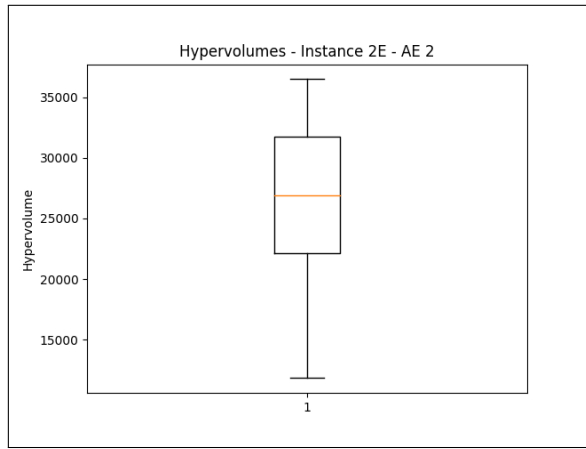


Fig. 7. Boxplot de los hipervolumenes obtenidos para la instancia 2E utilizando el AE 2.

C. Comparación con otros algoritmos

Se lleva a cabo una comparación entre los resultados obtenidos por el algoritmo propuesto y los generados por el algoritmo ávido (Greedy) implementado. El objetivo principal es evaluar el rendimiento y la eficacia del enfoque basado en Algoritmos Evolutivos (AE) en comparación con una estrategia ávida. El algoritmo ávido implementado se caracteriza por tomar decisiones locales óptimas en cada etapa del proceso sin considerar la totalidad del problema. En este caso, se evaluarán los resultados teniendo en cuenta las funciones objetivo $f1$ y $f2$ y la eficiencia computacional.

Algoritmo	f1_min	f1_media	f1_max	f1_std
AE 2	248.91	357.66	465.97	57.56
Greedy	761.78	761.78	761.78	-
Mejora (%)	67.33%	53.05%	38.83%	-

TABLE XVIII

COMPARACIÓN DE RESULTADOS DE F1 ENTRE AE 2 VS GREEDY INSTANCIA 1E

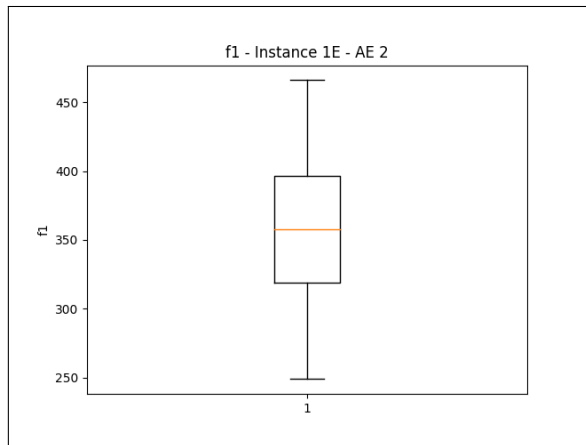


Fig. 8. Boxplot de los resultados de la función objetivo $f1$ obtenidos sobre la instancia 1E utilizando el AE 2.

Algoritmo	f1_min	f1_media	f1_max	f1_std
AE 2	402.55	506.39	640.12	60.78
Greedy	913.36	913.36	913.36	-
Mejora (%)	55.92%	44.56%	29.92%	-

TABLE XIX

COMPARACIÓN DE RESULTADOS DE F1 ENTRE AE 2 VS GREEDY INSTANCIA 2E

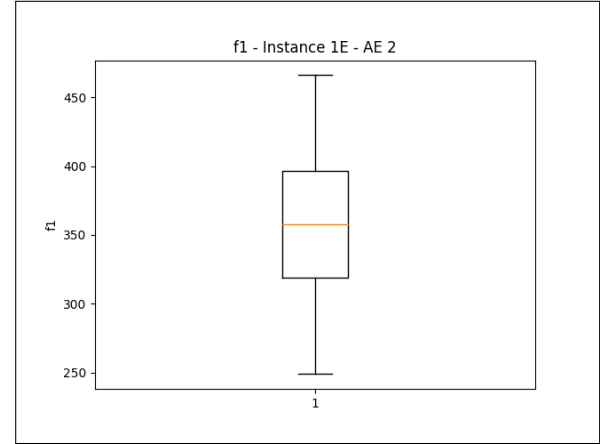


Fig. 9. Boxplot de los resultados de la función objetivo $f1$ obtenidos sobre la instancia 2E utilizando el AE 2.

Observación: El algoritmo Greedy fue diseñado para que una vez que elige un alimento a incluir en la dieta semanal, este se quita de la lista de alimentos posibles en la siguiente iteración. Por lo tanto, dado que el dataset tiene muchos mas alimentos que los que lleva una dieta semanal, nunca se repetirán alimentos en la solución resultado. Obteniendo siempre una dieta donde no se repiten alimentos y donde el valor funcional de $f2 = 0$.

Algoritmo	tiempo_media (s)	tiempo_std
AE 2	6.20	0.167
Greedy	3.43	-

TABLE XX

COMPARACIÓN DE TIEMPOS DE EJECUCIÓN DE AE 2 VS GREEDY INSTANCIA 1

Algoritmo	tiempo_media (s)	tiempo_std
AE 2	6.48	0.45
Greedy	3.17	-

TABLE XXI

COMPARACIÓN DE TIEMPOS DE EJECUCIÓN DE AE 2 VS GREEDY INSTANCIA 2

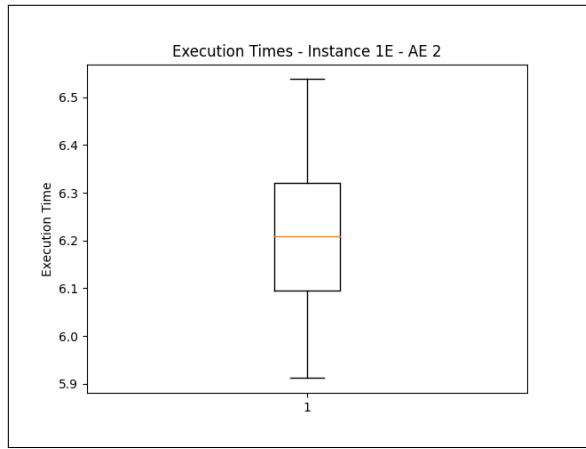


Fig. 10. Boxplot de los tiempos de ejecución del AE 2 sobre la instancia 1E.

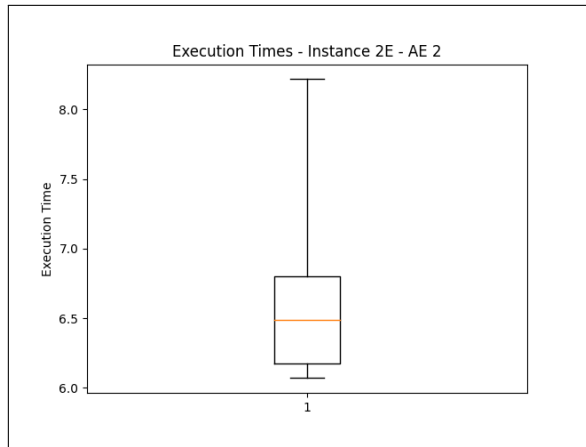


Fig. 11. Boxplot de los tiempos de ejecución del AE 2 sobre la instancia 2E.

VII. EJEMPLO DE DIETAS SEMANALES

Con fin de mostrar los resultados del algoritmo, a continuación se proveen los valores de las funciones objetivos de dos soluciones tomados de forma aleatoria del frente de pareto para la instancia 1E e instancia 2E. Además para ambas soluciones tomadas, los valores en f_2 fueron 0. Lo que implica que no se repitió ningún alimento en la semana.

	Calorías	Proteínas	Carbohidratos	Grasas
Objetivo Ins. 1E	2385.0 kc	179.0 g	238.0 g	79.0 g
Solución Ins. 1E	2383.4 kc	98.2 g	282.7 g	96.1 g
Objetivo Ins. 2E	2888.0 kc	217.0 g	325.0 g	80.0 g
Solución Ins. 2E	2892.1 kc	147.8 g	301.2 g	121.4 g

TABLE XXII

RESULTADOS PROMEDIO DE VALORES NUTRICIONALES

VIII. CONCLUSIONES Y TRABAJO A FUTURO

En este trabajo se presentó el problema de obtener una dieta que cumpla con los requisitos funcionales de una persona dada y que esta proporcione una variabilidad de alimentos significativa. Se abstraigo el problema y se diseñaron distintas configuraciones paramétricas donde participaron el tipo de

cruzamiento, la probabilidad de cruzamiento y la probabilidad de mutación. En el proceso de configuración paramétrica se incluyeron análisis no paramétricos sobre los resultados obtenidos en base al hipervolumen. De este proceso se logro seleccionar la configuración definitiva para continuar hacia la evaluación experimental.

Se generaron nuevas instancias para realizar la evaluación experimental con el fin de omitir el sesgo que podría tener seleccionar las mismas instancias que en la etapa de calibración, obteniendo valores de hipervolumen de la misma magnitud en la etapa de evaluación. Además se obtuvieron soluciones del frente de pareto de forma aleatoria con el fin de comparar si los resultados en términos de valores nutricionales se correspondían con los objetivos de cada instancia, obteniendo buenos resultados en cuanto a ambas funciones objetivo.

A su vez, la configuración seleccionada fue evaluada en conjunto al algoritmo Greedy presentado, de donde se observó un promedio del 50% de mejora con respecto al algoritmo ávido. De forma adicional, se destaca el hecho de que si bien el algoritmo ávido realiza una buena aproximación a los objetivos definidos, la obtención de una dieta mediante este método no parece ser una buena solución, ya que implica que cada ejecución arrojará los mismos resultados, teniendo la misma dieta todas las semanas.

Como trabajos a futuro se plantea:

- Explorar nuevas configuraciones de los parámetros, sobre todo en valores de probabilidad de mutación.
- Búsqueda de un nuevo dataset que contenga información nutricional ya calculada de una comida, entendiéndose una comida como un conjunto de alimentos.
- Optimizar el algoritmo para intentar alcanzar tiempos de ejecución aún mas bajos, y que se acerque a los tiempos del algoritmo ávido.
- Analizar las diferentes implementaciones de MOEAs sobre el problema presentado, en busca de observar como impactan en las soluciones obtenidas por cada uno.

IX. ANEXO

Adjunto a este informe se encuentran todos los archivos y código fuente utilizado; en conjunto se encuentran las instancias de pruebas con la correspondiente configuración y el dataset de alimentos utilizado. El script principal main.py requiere dos argumentos obligatorios y uno opcional:

```
python3 main.py instance proyect_name
[--ae ae]
```

- *instance*: selecciona la instancia a ejecutar (los valores disponibles son: '1', '2', '1E' y '2E').
- *proyect_name*: ingresa el nombre del proyecto, el cual sera además el nombre de la carpeta en donde se retornaran los resultados.
- *ae*: campo no obligatorio, selecciona una AE en particular a ejecutar, en caso de ser '0' ejecuta todas.

Es posible obtener ayuda adicional sobre este script mediante el comando:


```
python3 main.py -h
```

Además, se pone a disposición ejecutables .cmd para la fácil ejecución de los scripts implementados, los mismos ya presentan los comandos utilizados.

REFERENCES

- [1] "Material Teórico – Curso Algoritmos Evolutivos". EVA FING. <https://eva.fing.edu.uy/course/view.php?id=1049> (accessed: Oct. 21, 2023).
- [2] Babalola, A.E., Ojokoh, B.A., Odili, J.B.: Diet optimization techniques: A review, Computer Engineering and Computer Science (ICMCECS), pp. 1-5, IEEE (2020).
- [3] "jMetalPy". GitHub. <https://github.com/jMetal/jMetalPy> (accessed: Oct. 21, 2023).
- [4] "Calorías de los Alimentos - Información Nutricional". Medicina Información. <https://medicinainformacion.com/wp-content/uploads/2018/04/tabla-de-alimentos.pdf> (accessed: Oct. 21, 2023).
- [5] "Food and Nutrient Database for Dietary Studies". Agricultural Research Service. U.S. Department of Agriculture. <https://www.ars.usda.gov/northeast-area/beltsville-md-bhnrc/beltsville-human-nutrition-research-center/food-surveys-research-group/docs/fndds-download-databases/> (accessed: Dec. 12, 2023).
- [6] "Calculadora de Calorías y Macros para Calcular tu Metabolismo" TRX España. <https://trxspain.es/calculadora-de-calorias-y-macros/> (accessed: Dec. 12, 2023).
- [7] "Recomendaciones de Ingesta Energía y Nutrientes para la población uruguaya". Ministerio de Salud Pública. https://www.gub.uy/ministerio-salud-publica/sites/ministerio-salud-publica/files/documentos/publicaciones/RECOMENDACIONES_INGESTA_ENERGIA_NUTRIENTES_POBLACION_URUGUAYA.pdf (accessed: Dec. 12, 2023).
- [8] "Scipy." <https://www.scipy.org/> (accessed: Dec. 11, 2023)
- [9] J. Arthur Harris and Francis G. Benedict. Proceedings of the National Academy of Sciences. Vol. 4 (December 1918)