



**Escuela de Doctorado
y Estudios de Posgrado**
Universidad de La Laguna

Máster Universitario en Ingeniería Informática

Trabajo Fin de Máster

Planificación de menús escolares
con mínimo impacto ambiental
usando Algoritmos Evolutivos.

Computación y Nutrición.

Iván García Campos

Dña. **Coromoto León Hernández**, con N.I.F. 78605216-W profesora Catedrático de Universidad adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

D. **Carlos Segura Gonzalez**, con N.I.F. 78404244-S investigador tipo C, adscrito al Departamento de Ciencias de la Computación del Centro de Investigación Matemática (CIMAT) de México, como cotutor,

C E R T I F I C A (N)

Que la presente memoria titulada:

*“Planificación de menús escolares con mínimo impacto ambiental usando Algoritmos Evolutivos”.
Computación y Nutrición.*

ha sido realizada bajo su dirección por D. **Iván García Campos**, con N.I.F. 43833191-K.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 7 de septiembre de 2019.

Agradecimientos

Quiero dar las gracias a todas aquellas personas que han contribuido a la realización de este Trabajo Fin de Master: En primer lugar, quisiera agradecer a Coromoto León Hernández y Carlos Segura González, tutora y cotutor respectivamente de este proyecto, por la confianza que ha depositado en mí. Su apoyo, paciencia y compromiso ha sido parte esencial en el desarrollo de este trabajo.

Quisiera también, reconocer a todas aquellas personas que me han apoyado y dado ánimos para seguir hacia delante.

Por último me gustaría dar las gracias a mis padres y hermana por estar siempre ahí y confiar siempre en mí, por saber comprenderme, relativizar cualquier problema y enseñarme lo que realmente es importante.

Licencia

* Si quiere permitir que se compartan las adaptaciones de tu obra y NO quieres permitir usos comerciales de tu obra indica:



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional.

Resumen

El objetivo de este trabajo ha sido el estudio y resolución de un problema de asignación de menús para comedores escolares con variables multiobjetivo. Este estudio se enmarca dentro del ámbito de la Optimización Combinatoria y para su resolución se desarrollan técnicas aproximadas que permitan resolver el problema consiguiendo soluciones factibles en un tiempo razonable.

En este problema se debe asegurar un menú equilibrado y variado analizando diferentes restricciones nutricionales (energía, proteínas, azúcar, ...), máximo de repeticiones válido de un mismo plato en la misma semana y mes, o repetición del tipo de plato (arroz, pasta, vegetable, huevo, ...) semanalmente. Sin embargo, también se considera la optimización de dos valores relacionados con la Logística Verde, medidas medioambientales que ponen de manifiesto el impacto negativo que tienen para el planeta la adquisición de los diferentes alimentos y la generación de cada plato. En este caso se medirá la Huella Hídrica y la Huella de Carbono. Estos valores deberán minimizarse para obtener menús además de equilibrados, sostenibles.

Se ha considerado iniciar el procedimiento partiendo de soluciones aleatorias y utilizar un evaluador capaz de medir la infactibilidad y los valores para los distintos objetivos. A continuación, se utilizan distintas técnicas metaheurísticas como búsquedas locales y algoritmos evolutivos que en su conjunto pueden lograr alcanzar soluciones factibles.

Los resultados experimentados ponen de manifiesto que las técnicas aproximadas son capaces de proporcionar solución cercana a la exacta en un tiempo razonable.

Palabras clave: *Computación, Nutrición, Algoritmo Evolutivo, Problema de asignación de menús, Heurística, Optimización Lineal.*

Abstract

The objective of this work has been the study and resolution of a problem of assigning menus for school canteens with multiobjective variables. This study is part of the field of Combinatorial Optimization and approximate techniques are developed for its resolution which allow solving the problem, obtaining feasible solutions in a reasonable time.

In this problem, a balanced and varied menu must be ensured by analyzing different nutritional restrictions (energy, protein, sugar, ...), maximum repetitions valid for the same dish in the same week and month, or repetition of the type of dish (rice , pasta, vegetable, egg, ...) weekly. However, the optimization of two values related to Green Logistics is also considered, environmental measures that highlight the negative impact that the acquisition of different foods and the generation of each dish have for the planet. In this case, the Water Footprint and Carbon Footprint will be measured. These values should be minimized to obtain menus in addition to balanced, sustainable.

For this work it has been considered starting the procedure based on random solutions and using an evaluator capable of measuring the infeasibility and the values for the different objectives. Next, different metaheuristic techniques are used such as local searches and evolutionary algorithms that together can achieve feasible solutions.

Experienced results show that approximate techniques are capable of providing a near-exact solution in a reasonable time.

Keywords: *Computing, Nutrition, Evolutionary Algorithm, Menus assignment problem, Heuristics, Linear Optimization.*

Índice general

1. Introducción	1
1.1. Objetivos	2
1.2. Planificación	2
1.3. Conceptos previos.	3
1.3.1. ¿Qué es la Huella de Carbono?	3
1.3.2. ¿Qué es la Huella Hídrica?	3
1.3.3. ¿Qué es una Heurística?	3
1.4. El problema de la dieta	4
1.5. Problema a resolver	4
1.6. Datos del problema	5
2. Resolución del Problema de planificación de menús con mínimo impacto ambiental	11
2.1. Estructura del proyecto	11
2.2. Creación de un evaluador para el problema	12
2.3. Algoritmo aleatorio	13
2.4. Búsqueda Local	13
2.4.1. Búsqueda Local Recalculada	14
2.4.2. Búsqueda Local Incremental	14
2.5. Búsqueda Local Iterada	15
2.6. Algoritmo poblacional	15
2.6.1. Elección de la población	16
2.6.2. Estrategias de cruce	16
2.6.3. Elitismo	17
3. Resultados experimentales	18
3.1. Estudio entre Búsqueda Local Recalculada y Búsqueda Local Incremental	18
3.2. Estudio sobre Búsqueda Local Iterada	20
3.3. Estudio sobre Algoritmo Evolutivo	21
3.3.1. Población de 10 individuos, 60 segundos de búsqueda local iterada.	22
3.3.2. Población 4, 90 segundos	24
3.3.3. Población de 4 individuos, 30 segundos búsqueda	26
4. Conclusiones y líneas futuras	29
5. Summary and Conclusions	30
Bibliografía	31

Índice de figuras

1.1. Resultados previos.	5
1.2. Restricciones Ingesta Diaria Problema.	6
1.5. Restricciones Veces Categoría a la semana.	6
1.3. Platos.	7
1.4. Platos.	8
2.1. Arquitectura Ordenador Utilizado.	11
2.2. Distancia Cuadrática.	12
2.3. Espacio Soluciones de Búsqueda Local.	14
2.4. Espacio Soluciones Búsqueda Local Iterada.	15
3.1. Estudio sobre Algoritmo Recalculado e incremental. Datos Infactibilidad.	19
3.2. Estudio sobre Algoritmo Recalculado e incremental. Tiempos de ejecución.	19
3.3. Búsqueda Local Incremental Iterada. Datos de Infactibilidad	20
3.4. Tiempo de Ejecución, Algoritmo Búsqueda Local Incremental Iterada. Perturbación 5%.	21
3.5. Infactibilidad Grafo BoxPlot, 10 Individuos, 60 segundos	22
3.6. Infactibilidad Grafo Tiempo, 10 Individuos, 60 segundos	22
3.7. Agua Grafo Líneas 10 Individuos, 60 segundos	23
3.8. Carbono Grafo Líneas 10 Individuos, 60 segundos	23
3.9. Infactibilidad Grafo BoxPlot 4 Individuos, 90 segundos	24
3.10. Infactibilidad Grafo Tiempo 4 Individuos, 90 segundos	24
3.11. Agua Grafo Líneas 4 Individuos, 90 segundos	25
3.12. Carbono Grafo Líneas 4 Individuos, 90 segundos	25
3.13. Infactibilidad Grafo BoxPlot 4 Individuos, 30 segundos	26
3.14. Infactibilidad Tiempo 4 Individuos, 30 segundos	26
3.15. Agua Grafo Líneas 4 Individuos, 30 segundos	27
3.16. Carbono Grafo Líneas 4 Individuos, 30 segundos	27

Capítulo 1

Introducción

El presente Trabajo Fin de Master se enmarca en el estudio y optimización de problemas multiobjetivo [27], en el ámbito de la Programación Lineal. También plantea la utilización de algoritmos heurísticos y metaheurísticos [26], conjunto de técnicas de optimización que proporcionan soluciones “aceptables” en un tiempo razonable.

El problema en cuestión pretende la resolución de un menú para comedores escolares mensual capaz de ofrecer todas las características de una dieta sana y equilibrada; pero además, intentando minimizar la huella hídrica y de CO_2 equivalente producida por la producción de cada uno de los alimentos y la realización de cada plato. En este proceso, se deben tener en cuenta unas determinadas restricciones que indicarán el grado de satisfacción para un menú saludable, como por ejemplo, el máximo consumo diario recomendado de proteínas, sodio o azúcar. Además, se intentará minimizar dos variables convirtiéndose este, en un problema multiobjetivo con restricciones.

En la actualidad, existen multitud de problemas reconocidos como multiobjetivos al depender de diferentes criterios para declarar que una solución sea completamente satisfactoria. La mayoría de estos problemas afectan directamente la vida diaria y a menudo, algunos de los criterios de aceptación entran en conflicto unos con otros, no existiendo una única solución que simultáneamente satisfaga a todos a la vez. Por lo tanto, una solución factible debe obtenerse en concordancia con cada uno de los decisores con el fin de cumplir un conjunto de requisitos.

Cabe destacar que a diferencia de muchos estudios relacionados con la optimización de menús y dietas [29], la incorporación de aspectos medioambientales es un tema relativamente nuevo y de gran interés. La logística verde puede llevar asociado más de un criterio de eficiencia en el problema, convirtiéndolo en un problema multiobjetivo. Además, se ha demostrado que la producción de determinados alimentos como son la carne y productos lácteos son actualmente insostenibles. La agricultura, la deforestación y otros usos de la tierra son responsables de cerca del 25% de las emisiones de gases de invernadero relacionadas con la acción humana [3]. En nuestra formulación del problema, la huella hídrica y de CO_2 equivalente procedentes de la producción de los alimentos y la elaboración del plato deberán minimizarse. Resolver un problema multiobjetivo real con un gran número de datos puede llegar a ser costoso computacionalmente, siendo en muchos casos inabordable de manera exacta en tiempo polinomial.

Sin embargo, en este caso, todos los datos necesarios para la resolución de este problema han sido extraídos del artículo “*An optimal plan for food consumption with minimal environmental impact: the case of school lunch*” [24], de los autores *Luca Benvenuti, Alberto De Santis, Fabio Santesarti y Luigino Tocca* que contiene la definición del problema, la instancia de datos utilizados y una solución factible y óptima para la optimización de la huella hídrica y otra para la huella de carbono. En este trabajo se utilizó un enfoque basado en la formulación matemática y la utilización del sistema AMPL[14] para conseguir los resultados finales. **La intención de este Trabajo Fin de Máster es la resolución del mismo problema de asignación de menús para comedores escolares utilizando otro enfoque de resolución, utilizando metaheurísticas y técnicas aproximadas utilizando la optimización combinatoria y sin necesidad de utilizar un sistema de pago para conseguir soluciones factibles.**

En este Trabajo Fin de Máster han sido desarrollados diferentes métodos y algoritmos que no tienen por qué ofrecer la mejor solución posible, sino que obtienen soluciones de buena calidad. Se ha desarrollado un algoritmo final capaz de resolver el problema de asignación de menús con restricciones y dos objetivos en un tiempo prudencial. Hay que tener en cuenta que dependiendo de la configuración que se aplique a estas metaheurísticas se conseguirán unos resultados u otros. Debe equilibrarse la diversificación de soluciones pero también la intensificación. Estos serán cruciales para conseguir soluciones de buena calidad y determinar el tiempo de ejecución de las heurísticas.

De esta manera, a lo largo de este trabajo de investigación se estudiarán diferentes técnicas aproximadas que logren resolver el problema de asignación de menús planteado.

1.1. Objetivos

Nuestro objetivo es el estudio de técnicas de optimización lineal para obtener soluciones exactas para un problema de asignación de menús y la resolución de un sistema complejo utilizando algoritmos heurísticos y metaheurísticas con la necesidad además, de optimizar una serie de objetivos. Se buscarán métodos que cumplan las restricciones del problema logrando soluciones factibles y asimismo optimicen dos variables que mejoran la huella hídrica y de carbono de los platos asignados en la asignación del menú.

En este sentido, **el objetivo principal de este Trabajo de Fin de Grado es conseguir la resolución de un problema multiobjetivo con restricciones basándonos en el problema propuesto con título “An optimal plan for food consumption with minimal environmental impact: the case of school lunch”, un problema de asignación de menú para comedores infantiles, intentando lograr soluciones factibles en un módico tiempo de ejecución utilizando una estrategia de resolución diferente, basado en técnicas aproximadas[24].**

Para ello, primero se analizará el conjunto de datos y restricciones del problema además de conocer las implicaciones de cada una de las variables. Se desarrollará un generador de soluciones para este problema, un evaluador de infactibilidad y distintos procedimientos heurísticos y técnicas aproximadas para lograr conseguir soluciones factibles que aseguren buenas soluciones en un tiempo razonable y que sea independiente de la instancia del problema que se utilice.

1.2. Planificación

Este Trabajo se ha dividido en cinco actividades:

- **ACT1: Creación de una solución aleatoria para el problema señalado.**
- **ACT2: Creación de un evaluador capaz de medir la factibilidad de la solución actual.**
- **ACT3: Estudio y desarrollo de procedimientos para la optimización de problemas multiobjetivo.**
 - Búsqueda Local.
 - Búsqueda Local Iterativa.
 - Algoritmo Poblacional.
- **ACT4: Resultados y Validación final.** Análisis de los resultados conseguidos y comparación con los aportados en trabajo previo [24]. Conclusiones y trabajo futuro.
- **ACT5: Documentación.** Elaboración de material de soporte para el uso de la herramienta así como realización de una memoria de Trabajo Fin de Máster.

1.3. Conceptos previos.

Durante el desarrollo del Trabajo de Fin de Grado se utilizan algunos conceptos teóricos que se especifican a continuación:

1.3.1. ¿Qué es la Huella de Carbono?

La Huella de Carbono es un indicador de la cantidad de emisiones de gases de efecto invernadero que produce el ser humano al fabricar un producto o realizar sus actividades diarias, es la huella que deja nuestro paso por el planeta. Se expresa en toneladas de CO_2 emitidas [5]. La Huella de Carbono de los alimentos es una estimación de todas las emisiones causadas por la producción (por ejemplo, la agricultura o la ganadería), la producción y la distribución al consumidor hasta la eliminación de los envases. Tomemos el ejemplo de un tomate:

La producción en invernadero de 1 kg de tomates libera 3.5 kg de CO_2 equivalente, mientras que la misma cantidad producida en el campo es inferior a 0.05 kg, 70 veces menor. La producción en invernaderos requiere una enorme cantidad de energía para mantener las temperaturas de cultivo. Todo esto sin contar el transporte aéreo de productos de un extremo del planeta a otro. De ello se deduce que la compra de productos de temporada tiene un menor impacto en el medio ambiente. Datos de ImprontaWWF [2].

1.3.2. ¿Qué es la Huella Hídrica?

El concepto de Huella Hídrica es similar al de la huella de carbono, pero en este caso se define como el volumen total de agua dulce utilizado para producir los bienes y servicios que habitualmente consumimos. Es una variable que nos permite conocer cuanta agua es necesaria para fabricar un producto desde la adquisición y su procedimiento, hasta el envasado y transporte. El uso del agua se mide en términos del volumen de agua consumida (evaporada) o contaminada, en relación con el área de origen de ese volumen de agua. De hecho, muchos problemas de agua dependen de la disponibilidad local del recurso y, por lo tanto, la información sobre el origen del agua dulce es esencial para determinar la salud ambiental o humana y las implicaciones de su uso. La huella hídrica es la suma de tres componentes[8]: la huella hídrica azul, la huella hídrica verde y la huella hídrica gris:

- La **Huella Hídrica azul** es el volumen de agua dulce (agua superficial y subterránea) que se extrae del ciclo natural y se utiliza en la producción.
- La **Huella Hídrica verde** es el volumen de agua de lluvia transpirada durante el cultivo.
- La **Huella Hídrica gris** es el volumen de agua contaminada, cuantificada como el volumen de agua necesario para diluir los contaminantes hasta tal punto que la calidad del agua vuelve por encima de normas de calidad. El concepto de huella hídrica se introdujo para satisfacer la necesidad de introducir indicadores de consumo sobre el uso de agua dulce y conocer el impacto en los recursos hídricos del planeta que produce el ser humano. Cuestiones como la escasez de agua y la contaminación de los recursos hídricos se puede entender y enfrentar mejor considerando el proceso de producción y la cadena de producción como una sola cosa.[30]

1.3.3. ¿Qué es una Heurística?

La palabra heurística[26] tiene su origen en la antigua palabra griega *heuriskein*, significa “El arte de descubrir nuevas estrategias (reglas) para resolver problemas”. El sufijo meta, también es una palabra griega que significa “*metodología de nivel superior*”. En el campo de la computación, el término metaheurístico fue introducido por F. Glover[4] en el documento[21].

Los métodos de búsqueda metaheurística se pueden definir como Metodologías generales (plantillas) de nivel superior que pueden utilizarse como estrategias de guía en el diseño de heurísticas subyacentes para resolver problemas específicos de optimización. Algunas de estas heurísticas son:

- Búsqueda exhaustiva.
- Estrategia Greedy.
- Algoritmos poblacionales.
- Búsqueda local.
- Algoritmos bioinspirados.

El principal objetivo de los párrafos siguientes es ubicar el problema de asignación de menús y problemas multiobjetivos además de introducir algunas definiciones sobre el entorno, problemática y las estrategias de resolución que se han aplicado a lo largo del tiempo para este tipo de problemas.

1.4. El problema de la dieta

El Problema de la Dieta [18] es un problema de Programación Lineal[12] resuelto por primera vez en 1947 por George Stigler [29, 31] que consiste en determinar las cantidades de distintos alimentos que deben ingerirse asegurando ciertas condiciones de nutrición y minimizando el coste de compra de cada uno.

Para un problema clásico, debe determinarse el número de unidades que debe comprarse de cada alimento que haga que se satisfagan los mínimos aconsejados de ingesta y se alcance un precio total mínimo. Los elementos que intervienen en el problema de la dieta son:

- **Datos.**
 - m: el número de nutrientes
 - n: el número de alimentos
 - a_{ij} : la cantidad del nutriente i en una unidad del alimento j
 - b_i : la cantidad mínima del nutriente i aconsejada
 - c_j : el precio de una unidad del alimento j
- **Variables.**
 - x_j la cantidad del alimento j que debe adquirirse
- **Restricciones.** Dado i , la cantidad total de un nutriente, es la suma de las cantidades de los nutrientes en todos los alimentos no pudiendo ser las cantidades de estos negativas:

$$\sum_{j=1}^n a_{ij} x_j \geq b_i; i = 1, \dots, m$$

$$x_j \geq 0; j = 1, \dots, n$$

- **Función a minimizar.** En este caso se debe minimizar el precio de la dieta:

$$\sum_{i=1}^n \sum_{j=1}^n c_i x_j$$

La información sobre los valores válidos de cada una de las restricciones están representados con rango de valores mínimo y máximo utilizando números difusos[11].

1.5. Problema a resolver

El problema que se desea resolver ha sido extraído de la investigación previa publicada por *Journal of Cleaner Production* [23] con título “An optimal plan for food consumption with minimal environmental impact: the case of school lunch”[24, 13].

En este trabajo, se pretende la resolución de un problema de asignación de menús proporcionándose un conjunto posible de 106 recetas de la cocina mediterránea divididos en 33 primeros platos(pasta,

arroz, sopa,...), 48 segundos platos(carne, pescado, huevo, ...), 23 guarniciones(ensalada, vegetable, ...), fruta y pan.

El conjunto de los platos ha sido diseñado a partir de 71 posibles ingredientes. La energía y nutrientes producidos por cada alimento fueron extraídos del Consejo Europeo de Información Alimentaria[19] y los datos de la huella de agua y carbono han sido extraídos de la base de datos de WWF en colaboración con la Universidad de Tuscia de Viterbo(Italia)[16] y la empresa Mutti S.p.A [25]. El valor total equivalente de cada plato se reflejan en las Figuras 1.3 y 1.4.

Alguno de los datos utilizados no se encuentran descritos en revista. Nos hemos puesto en contacto con los autores solicitándolos directamente. Se ha confirmado que existe una categoría llamada “Vegetable” que debe evaluarse de manera diaria teniendo que asignarse como mínimo una vez. Además se han proporcionado las categorías de cada uno de los platos que tampoco habían sido publicadas.

Los autores resolvieron el problema utilizando AMPL[14], un lenguaje de modelado algebraico privativo obteniendo las soluciones optimas si el objetivo es minimizar la huella hídrica o por el contrario, la huella de carbono, Figura 1.1.

	Huella Hídrica	Huella de Carbono
Optimizando Huella Hídrica	13.72	16.64
Optimizando Huella de Carbono	10.85	7.77

Figura 1.1: Resultados previos.

El análisis previo del problema determina lo siguiente:

▪ **Determinación de las restricciones:**

- Veces que se puede utilizar el mismo plato al mes.
- Veces que se puede usar el mismo plato a la semana.
- Veces que se puede usar la misma categoría de plato a la semana.
- Ingesta Recomendada diariamente.

▪ **Formulación de la función objetivo.**

Se trata de la función que mide la calidad de la solución y que hay que optimizar. En este caso minimizar dos costes.

- Minimizar el Gasto de Huella Hídrica de los platos asignados.
- Minimizar el Gasto de Huella de Carbono de los platos asignados.

1.6. Datos del problema

En este problema de asignación de menús se define la necesidad de realizar un menú escolar de lunes a viernes durante un total de cuatro semanas; veinte días. Cada menú diario esta constituido por un primer plato, un segundo plato, una guarnición, fruta y pan. Cada una de las recetas proporcionadas ofrece informacion sobre:

- | | | |
|----------------------|------------------|---------------------------|
| ▪ Energía total. | ▪ Carbohidratos. | ▪ Sodio. |
| ▪ Proteínas totales. | ▪ Fibra. | ▪ Huella Hídrica. |
| ▪ Lípidos totales. | ▪ Azúcar. | ▪ Huella CO2 Equivalente. |

También se proporcionan los datos relativo al pan y la fruta.

Para el conjunto de días debe lograrse un menú que no supere los límites inferiores o superiores de ninguna de las restricciones. Exclusivamente, el plato Lasaña, tendrá que ser asignado una vez al mes. Por último, cada plato tiene una o varias categorías asociadas de la forma:

- Pasta.
- Tomato Pasta.
- No Tomato Pasta.
- Arroz.
- Carne.
- Pescado.
- Huevo.
- Lácteo.
- Patata.
- Legumbre.
- Ensalada.
- Vegetable.

No debe superarse ni quedarse corto de cada una de las categorías por semana. Por ejemplo, un plato de Arroz puede utilizarse como mínimo una vez por semana y dos como máximo. Exclusivamente, en el caso de la categoría vegetable, esta tiene que consumirse mínimo una vez al día, ver Figuras 1.3 y 1.4. Por otro lado, en la Figura 1.2 se presentan los límites inferior y superior permitidos de la ingesta recomendada diaria.

Propiedad	Mínimo	Máximo
Energía	500	700
Proteínas	0	28
Lípidos	0	40
Carbohidratos	60	80
Fibra	5	15
Azúcar	0	40
Sodio	300	500

Figura 1.2: Restricciones Ingesta Diaria Problema.

En la Figura 1.5 se presentan los límites inferior y superior permitidos en la asignación de categorías de platos semanalmente, ver Figura 1.5.

Categoría Plato	Mínimo Veces por Semana	Máximo Veces por Semana
Pasta	1	3
Tomato Pasta	1	2
No Tomato Pasta	0	1
Arroz	1	2
Carne	1	2
Pescado	1	2
Huevo	1	1
Lácteo	1	1
Patata	0	2
Legumbres	0	3
Ensalada	1	3
Vegetable (Diario)	1	1

Figura 1.5: Restricciones Veces Categoría a la semana.

Por último, para cada uno de los platos se indica el valor de huella hídrica y huella de Carbono utilizada. Para calcular el valor total de estos objetivos se hará un sumatorio de los platos actualmente elegidos.

La solución se conseguirá cuando se asigne un menú para veinte días incluyendo primer plato, segundo plato, guarnición pan y fruta en el que se tengan en cuenta todas las restricciones propuestas y se minimice el consumo de huella hídrica y huella de CO_2 equivalente.

Plato	Energía	Proteína	Lípido	Carbs	Fibra	Azúcar	Sodio	H2O	CO2eq	Categoría
Agnolotti with tomato sauce	426,71	16,98	15,57	58,16	2,8	5,69	410,2	423,6	313,03	Pasta, TomatoPasta
Cream of chickpea soup with pasta	125,71	3,96	4,95	17,46	2,65	1,83	11,45	242,3	100,58	Pasta, Legumbre
Cream of bean soup with pasta	117,31	4,11	4,35	16,5	3,16	1,68	10,55	242,3	100,58	Pasta, Legumbre
Cream of lentil soup with pasta	117,31	3,93	4,35	16,68	3,39	1,65	10,55	242,3	100,58	Pasta, Legumbre
Cream of vegetable soup with pasta (summer)	144,56	5,99	5,85	18,12	2,73	6,05	77,35	172,58	131,51	Pasta, Vegetable
Cream of vegetable soup with pasta (winter)	142,26	5,61	5,83	17,99	2,81	5,63	76,15	172,58	131,51	Pasta, Vegetable
Lasagna	302,01	14,78	16,51	24,97	1,76	6,13	111,9	699,98	349,46	Pasta, TomatoPasta
Pasta with butter and parmesan	202,66	6,05	10,94	21,32	1,05	1,02	49,4	230,69	181,66	Pasta, NoTomatoPasta
Pasta with pesto	187,4	4,92	8,6	23,71	1,7	1,89	1,33	319,87	183,39	Pasta
Pasta with tomato sauce	171,46	6,08	6,17	24,32	2,01	4,02	45,95	250	260,73	Pasta, TomatoPasta
Pasta with tomato and basil	169,31	5,98	6,16	23,9	1,82	3,6	38,4	248,56	259,11	Pasta, TomatoPasta
Pasta with meat sauce	191,41	9,12	7,03	24,32	2,01	4,02	53,9	482,5	353,43	Pasta, TomatoPasta
Pasta with vegetable ragu	175,16	6,21	6,19	25,11	2,33	4,8	56,85	252,19	261,99	Pasta, TomatoPasta
Pasta with trout	169,66	8,33	5,27	23,61	1,77	3,31	17,5	222,28	309,06	Pasta, TomatoPasta
Pasta with tuna	196,66	10,55	7,27	23,61	1,77	3,31	86,9	372,36	285,75	Pasta, TomatoPasta
Pasta with vegetables	176,86	6,72	6,21	25	2,27	4,66	50,35	256,44	265,93	Pasta, TomatoPasta, Vegetable
Pasta all'Amatriciana	220,53	9,41	10,31	23,9	1,82	3,6	321,3	325,62	331,04	Pasta, TomatoPasta
Pasta with marinara sauce	148,66	4,25	4,75	23,61	1,77	3,31	7,9	222,28	245,46	Pasta, TomatoPasta
Pasta with tomato sauce and oregano	148,66	4,25	4,75	23,61	1,77	3,31	7,9	222,28	245,46	Pasta, TomatoPasta
Fettuccine with tomato sauce	148,76	5,37	6,3	18,65	1,62	3,53	45,85	257,81	183,72	TomatoPasta
Pasta with zucchini	163,31	6,3	5,84	22,86	1,62	2,48	40	227,2	159,11	Pasta, Vegetable
Pasta and potatoes soup	130,36	4,08	5,66	16,86	1,3	1,56	42,05	154,56	96,53	Pasta, Patata
Pasta with Mediterranean sauce	164,11	6,19	5,59	23,63	1,77	3,33	43,9	267,3	257,55	Pasta, TomatoPasta
Pasta with ricotta and tomato sauce	181,31	6,16	6,95	25,02	2,01	4,72	31,55	288,26	280,98	Pasta, TomatoPasta
Parmesan risotto	174,66	4,23	10,66	16,42	0,21	0,25	51,5	334,01	87,86	Arroz
Rice and potatoes porridge	118,36	3,3	5,54	14,76	0,94	1,23	42,95	198,94	56,23	Arroz, Patata
Tomato risotto	143,46	4,26	5,89	19,42	1,17	3,25	48,05	353,32	166,93	Arroz
Saffron rice	124,51	3,2	5,48	16,6	0,26	0,43	33,3	317,64	54,91	Arroz
Creamy pea risotto	134,11	4,28	5,54	17,88	1,52	1,83	81,7	324,08	110,71	Arroz, Legumbre
Endive risotto	129,31	3,47	5,57	17,41	0,74	1,24	36,3	324,75	74,71	Arroz, Vegetable
Pumpkin risotto	129,91	3,53	5,51	17,65	0,65	1,18	39,9	327,3	62,71	Arroz, Vegetable
Zucchini risotto	132,61	4,16	5,54	17,62	0,65	1,39	39,9	327,3	62,71	Arroz, Vegetable
Tortellini with butter and parmesan	456,86	16,89	20,31	55,01	1,8	2,54	413,2	402,14	227,06	Pasta, NoTomatoPasta
Braised lamb with potatoes	545,91	36,39	28,52	38,55	2,7	0,9	157,5	715,98	194,34	Carne
Lamb cacciatore	278,96	32,04	16,78	0	0	0	144	606,23	165,86	Carne
Roast beef	133,86	19,83	5,65	1,09	0,39	1,09	49,7	1451,76	566,78	Carne
Roast pork	190,76	18,72	12,91	0	0	0	53,1	485,36	215,36	Carne
Roast turkey	159,06	26,85	5,29	1,09	0,39	1,09	74	407,76	177,98	Carne
Balsamic beef stew	157,42	18,28	9,13	0,34	0	0,3	48,16	1564,8	810,08	Carne
Roasted chicken leg	350,96	43,5	19,75	0	0	0	177	638,36	287,36	Carne
Breaded pork cutlet	237,71	21,12	14,14	7	0,34	0,45	105,54	544,06	225,32	Carne
Cod fillet croquettes	151,17	18,79	5,21	7,71	0,34	0,36	103,06	84,2	369,47	Pescado
Hake fillet croquettes	151,17	18,79	5,21	7,71	0,34	0,36	103,06	84,2	369,47	Pescado
Roasted turkey breast with lemon	167,5	27,1	5,3	3,05	0,12	0,07	61,28	409,75	180,83	Carne
Fried turkey breast	153,86	26,64	5,26	0	0	0	61,2	404,36	175,76	Carne
Dab filets au gratin	140,86	19,48	5,52	3,61	0,17	0,34	150	59,21	362,12	Pescado
Breaded dab filets	190,22	22,47	7,24	9,45	0,46	0,71	204,66	193,46	371,78	Pescado
Cod filets au gratin	126,56	17,61	4,75	3,5	0,17	0,23	92,8	59,21	362,12	Pescado
Hake filets au gratin	126,56	17,61	4,75	3,5	0,17	0,23	92,8	59,21	362,12	Pescado
Breaded bass filets	190,22	22,47	7,24	9,45	0,46	0,71	204,66	139,46	371,78	Pescado
Asiago cheese	178	15,7	12,8	1	0	1	380	253	230	Lacteo
Caciotta cheese	192	12,25	15,5	0,9	0	0,9	257	158,9	84	Lacteo
Crescenza cheese	196,7	11,27	16,31	1,33	0	1,33	245	222,46	117,6	Lacteo
Montasio cheese	205,5	15,15	16,1	1	0	1	378,5	253	230	Lacteo
Provolone cheese	187	14,05	14,1	1	0	1	430	253	230	Lacteo
Omelets	99,96	6,2	8,35	0	0	0	68,5	249,19	16,86	Huevo
Beef burger	175,3	19,1	9,48	3,5	0,17	0,23	69,81	1465,96	569,03	Carne
Pork burger	196	17,93	12,36	3,5	0,17	0,23	87,81	502,96	219,83	Carne
Cod fillet burger	135,83	18,27	5,16	4,28	0,21	0,28	102,97	78,14	363,77	Pescado
Hake fillet burger	135,83	18,27	5,16	4,28	0,21	0,28	102,97	78,14	363,77	Pescado
Bass fillet burger	150,13	20,14	5,93	4,39	0,21	0,39	160,17	78,14	363,77	Pescado
Mozzarella cheese	187,2	10,86	15,86	0,26	0	0,26	130	206,57	45,5	Lacteo
Fried chicken breast	152,06	27,18	4,81	0	0	0	41,4	404,36	175,76	Carne
Bread crumbed chicken breast	199,01	29,58	6,04	7	0,34	0,45	93,84	463,06	185,72	Carne
Beef meatballs with tomato sauce	182,9	19,51	9,64	4,69	0,49	1,41	73,01	1479,78	610,73	Carne
Meatballs with tomato sauce	206,96	20,05	11,97	4,99	0,58	1,71	100,91	1017,74	457,94	Carne
Baked meatballs of cod fillet	135,83	18,27	5,16	4,28	0,21	0,28	102,97	78,14	363,77	Pescado
Beef meatloaf	175,3	19,1	9,48	3,5	0,17	0,23	69,81	1465,96	569,03	Carne
Cooked ham	107,5	9,9	7,35	0,45	0	0,45	32,4	240	218	Carne
Cooked ham (half portion)	53,75	4,95	3,68	0,23	0	0,23	162	120	109	Carne
Ham	134	12,75	9,2	0	0	0	1289	240	218	Carne
Ham (half portion)	67	6,38	4,6	0	0	0	644,5	120	109	Carne
Ricotta cheese with cooked ham	154,6	11,6	11,14	2,1	0	2,1	691,3	310,68	209,8	Carne, Lacteo
Escalope with ham and sage	152,64	22,19	7,13	0,02	0	0,02	294,84	1480,79	610,7	Carne
Beef escalope	139,48	20,1	5,33	3,07	0,12	0,09	37,12	1438,17	572,17	Carne
Pork cacciatore	176,36	17,1	12,01	0	0	0	65,7	485,36	215,36	Carne

Figura 1.3: Platos.

Chicken breast strips	165,7	27,64	4,85	3,05	0,12	0,07	41,48	409,75	180,83	Carne
Beef strips	131,36	19,17	6,07	0	0	0	36	1448,36	564,56	Carne
Tuna in olive oil	96	12,6	5,05	0	0	0	158	300,15	80,58	Pescado
Potato mould	321,6	16,41	15,82	30,46	2,12	2,44	277,39	383,73	166,36	Patata
Scrambled eggs	94,32	6,23	7,69	0,04	0	0,04	68,78	218,05	21,94	Huevo
Sauteed chard	98,95	4,2	5,3	9	2,4	9	30	102,25	109,45	Vegetable
Boiled broccoli with olive oil	103,45	6,45	5,75	6,75	4,95	6,75	18	102,25	109,45	Vegetable
Stewed artichokes	77,95	4,05	5,3	3,75	7,5	2,85	199,5	115	428,95	Vegetable
Sliced carrots	80,25	1,11	5,2	7,67	3,1	7,67	95,1	92,6	20,1	Vegetable
Boiled string beans with olive oil	82,45	2,55	5,15	6,9	4,35	3,15	12	115	428,95	Legumbre, Vegetable
Fennels au gratin	76,72	2,77	5,62	3,83	3,41	1,65	30	109,97	27,43	Vegetable
Fried courgette flowers	200	5,92	9,04	23,28	0,96	0,32	2,16	70,01	36,99	Vegetable
Fennel salad	55,75	1,44	5	1,2	2,64	1,2	4,8	90,1	17,65	Vegetable, Ensalada
Tomatoes salad	65,35	1,44	5,24	3,36	1,2	3,36	3,6	90,58	16,45	Vegetable, Ensalada
Mixed salad with cucumbers	70,95	1,43	5,3	4,68	1,91	4,68	35,6	94,44	45,35	Vegetable, Ensalada
Mixed salad	71,15	1,55	5,18	4,84	2,55	4,84	43,7	92,2	47,65	Vegetable, Ensalada
Green salad	58,25	1,05	5,14	2,1	0,91	2,1	6,3	83,29	56,65	Vegetable, Ensalada
Mix of potatoes carrots and string beans	100,95	1,91	5,15	12,57	2,74	3,32	35,2	99,78	129,85	Patata, Vegetable
Baked potatoes	266,95	4,35	11,75	38,55	2,7	0,9	13,5	109,75	28,45	Patata
Mix of potatoes with olive oil	151,45	2,7	5,15	25,35	1,95	0,6	10,5	109,75	28,45	Patata
Roast potatoes	266,95	4,35	11,75	38,55	2,7	0,9	13,5	109,75	28,45	Patata
Sauteed potatoes	151,45	2,7	5,15	25,35	1,95	0,6	10,5	109,75	28,45	Patata
Stewed peas	84,65	4,37	5,24	5,41	5,09	5,89	194,1	93,44	233,95	Legumbre
Mashed potatoes	210,97	5,59	9,57	27,55	1,95	2,8	57,56	152,71	63,21	Patata
Spinach with butter and parmesan	87,88	5,58	5,29	4,56	2,7	0,73	109,85	96,31	445,98	Vegetable
Boiled spinaches with olive oil	79,45	4,2	5	4,5	2,7	0,68	85,5	115	428,95	Vegetable
Courgettes au gratin	99,49	5,2	5,38	8,21	2,1	5	49	120,2	52,97	Vegetable
Sauteed courgettes	85,45	4,8	5,3	5,1	1,95	4,8	33	115	49,45	Vegetable
Bread	110	3,24	0,2	25,4	1,52	0,8	117,2	52	35,2	Todos
Fruit	63,33	0,88	0,2	15,48	2,7	15,48	2,67	135,6	47,4	Todos

Figura 1.4: Platos.

▪ **Datos.**

- a: Número de propiedades.
- b: Número de platos.
- c: Número de categorías.
- d: Número de días.
- e_{ab} : Cantidad del nutriente. ab en una unidad del plato. j
- $f_{a,2}$: Cantidad mínima/máxima del nutriente a aconsejada.
- $g_{b,2}$: Veces mínimo/máximo repetición del plato b semanalmente.
- $h_{b,2}$: Veces mínimo/máximo repetición del plato b mensualmente.
- i_{cb} : Categorías c en una unidad del plato. b .
- $j_{cb,2}$: Veces mínimo/máximo repetición de la categoría c del plato b semanalmente.

▪ **Variables.**

- w Infactibilidad del menú.
- x Objetivo de Huella Hídrica del menú.
- y Objetivo de Huella de Carbono del menú.
- z_b la cantidad del alimento j que debe adquirirse

- **Restricciones.** Dado i , la cantidad total de un nutriente es la suma de las distancias cantidades de los nutrientes en todos los alimentos no pudiendo ser las cantidades de estos negativas:

- **Función a minimizar.** En este caso se debe lograr una infactibilidad con valor 0, minimizar el objetivo de huella hídrica y minimizar el objetivo de huella de carbono:

- $\min f_{H_2O(x)} x \in f$
- $\min f_{Carbono(x)} x \in f$
- $f_{Infactibilidad(x)} = 0 x \in f$

Formalmente:

$$f_{InfactibilidadIngestadiaria} = \sum_{b=1}^{106} \sum_{d=1}^{20} (x_i, x_j) * \sum_{a=1}^{12} \sum_{f_{a,2}=1}^2 distanciaFactibilidad(e_{ab}, f_{:,r})$$

$$f_{Infactibilidadplatosemanal(x)} = \sum_{b=1}^{106} \sum_{d=1}^{20} (x_i, x_j) \sum_{a=1}^{12} \sum_{g_{b,2}=1}^2 distanciaFactibilidad(e_{ab}, g_{:,r})$$

$$f_{Infactibilidadplatomensual(x)} = \sum_{b=1}^{106} \sum_{d=1}^{20} (x_i, x_j) * \sum_{a=1}^{12} \sum_{h_{b,2}=1}^2 distanciaFactibilidad(e_{ab}, h_{:,r})$$

$$f_{Infactibilidadcategoriaplatosemanal(x)} = \sum_{b=1}^{106} \sum_{d=1}^{20} (x_i, x_j) * \sum_{a=1}^{12} \sum_{j_{cb,2}=1}^2 distanciaFactibilidad(e_{ab}, j_{cb,(:,r)})$$

$$f_{Infactibilidad(x)} = f_{InfactibilidadIngestadiaria(x)} + f_{Infactibilidadplatosemanal(x)} + f_{Infactibilidadplatomensual(x)} + f_{Infactibilidadcategoriaplatosemanal(x)}$$

$$f_{HO_2(x)} = \sum_{i=1}^{106} \sum_{d=1}^{20} (x_i, x_j) * columnaHO_2 * elementos(:, i)$$

$$f_{Carbono(x)} = \sum_{i=1}^{106} \sum_{d=1}^{20} (x_i, x_j) * columnaCarbono * elementos(:, i)$$

La información relativa al rastro de agua y rastro de carbono fueron obtenidos por parte del trabajo previo[24] de la aplicación *impronta.wwf.it*[2] en colaboración entre WWF[32] y Mutti[25], además del apoyo de la Facultad de Agricultura y el Departamento de Ecología Forestal de la Universidad de Tuscia[16]. La información nutricional y datos de huella hídrica y huella de carbono se han determinado como la media de diferentes frutas: naranja, manzana, pera, uva, melocotón, cereza, mandarina, albaricoque y ciruela.

El objetivo de la aplicación ha sido sensibilizar al consumidor en la selección de alimentos. Para ello se desarrolló una versión del “*carrito de la compra*” en el que se indica de forma interactiva la información de huella hídrica y CO_2 Equivalente de cada alimento. Además, se detallan las pautas de la red de huella hídrica[6], la red internacional más autorizada que reúne a institutos de investigación, agencias gubernamentales, ONG y sector privado con el objetivo de promover el uso sostenible, equitativo y eficiente de los recursos[10].

Algunos de los datos utilizados no se encuentran en el artículo previo [24] y han sido proporcionados directamente a los autores como son las categorías de cada uno de los platos.

Capítulo 2

Resolución del Problema de planificación de menús con mínimo impacto ambiental

En este capítulo se indican los pasos que se han ido sucediendo a lo largo de este Trabajo Fin de Grado y se detalla el conjunto de procedimientos o técnicas aproximadas implementadas para la resolución del problema de la dieta con variables multiobjetivo.

2.1. Estructura del proyecto

El proyecto ha sido documentado utilizando la herramienta DOXYGEN[17] y está alojado en la plataforma Github[20], tanto el código en lenguaje C++ de la aplicación como la documentación[15]. Para facilitar el entendimiento de los algoritmos desarrollados, se proporcionan los pseudocódigos de cada uno de los métodos. Esta documentación se encuentra también alojada en el repositorio de Github. El proyecto ha sido desarrollado utilizando la programación orientada a Objeto.

El conjunto de estudios y ejecuciones necesarias para la resolución de este Trabajo Fin de Master se han desarrollado en un único ordenador. En la Figura 2.1 se detallan las características de la máquina utilizada en los experimentos.

```
Architecture:          x86_64
CPU op-mode(s):       32-bit, 64-bit
Byte Order:           Little Endian
CPU(s):                48
On-line CPU(s) list:  0-47
Thread(s) per core:   1
Core(s) per socket:   12
Socket(s):             4
NUMA node(s):         8
Vendor ID:             AuthenticAMD
CPU family:            16
Model:                 9
Model name:            AMD Opteron(tm) Processor 6164 HE
Stepping:              1
CPU MHz:               1699.987
BogoMIPS:              3400.13
Virtualization:       AMD-V
L1d cache:             64K
L1i cache:             64K
L2 cache:              512K
L3 cache:              5118K
NUMA node0 CPU(s):    0,4,8,12,16,20
NUMA node1 CPU(s):    24,28,32,36,40,44
NUMA node2 CPU(s):    3,7,11,15,19,23
NUMA node3 CPU(s):    27,31,35,39,43,47
NUMA node4 CPU(s):    2,6,10,14,18,22
NUMA node5 CPU(s):    26,30,34,38,42,46
NUMA node6 CPU(s):    1,5,9,13,17,21
NUMA node7 CPU(s):    25,29,33,37,41,45
```

Figura 2.1: Arquitectura Ordenador Utilizado.

2.2. Creación de un evaluador para el problema

Analizado los datos del problema debe plantearse la implementación de un evaluador capaz de medir la calidad de una solución calculando la infactibilidad o cumplimiento de las restricciones y su valor objetivo para la Huella Hídrica y la Huella de Carbono.

Algoritmo 1 Evaluador de soluciones

Datos: H, S, h, λ

Resultado: Valor objetivo de S

value = 0

for i in $\{0, 1, \dots, n - 2\}$ **do**

if $(h^i \neq H_{S_i}^i)$ **then**

 value = value + 1

end

if $(S_i \neq S_{i+1})$ **then**

 value = value + λ

end

end

if $(h^{n-1} \neq H_{S_{n-1}}^{n-1})$ **then**

 value = value + 1

end

return value

La infactibilidad de la solución se ha determinado como la suma de las distancias cuadrática de cada una de las propiedad cuando estas se encuentren fuera del rango válido.

$$\left[\begin{array}{ll} x < x_{\min} & (x - x_{\min})^2 \\ x > x_{\max} & (x - x_{\max})^2 \\ 0 & \end{array} \right.$$

Figura 2.2: Distancia Cuadrática.

Esto permite aumentar la infactibilidad de una restricción cuanto más lejos nos encontremos de la solución factible. Por otro lado, los objetivos de huella hídrica y huella de carbono se medirán como la suma de cada uno de los valores de los platos asignados para cada día.

Por ejemplo:

- El rango permitido para un menú diario para la propiedad 'Sodio' debe comprenderse entre 300mg y 500mg, para un valor diario de 234mg, el valor de infactibilidad de Sodio para ese día será de

$$(300 - 234)^2 = 4356$$

- El máximo número de veces por semana que se puede consumir carne es 2, Si tenemos en nuestra solución, sobre la misma semana, 4 platos de carne la distancia cuadrática será de

$$(4 - 2)^2 = 4$$

2.3. Algoritmo aleatorio

Para la resolución del problema se ha desarrollado un algoritmo inicial aleatorio que para el conjunto de días, asigna para cada uno de ellos de forma aleatoria un primer plato, segundo plato, guarnición, pan y agua. Esta solución no tiene porque cumplir el conjunto de restricciones. Simplemente obtenemos una solución que no ha sido consecuencia de ningún procedimiento de optimización. De esta manera, la generación de soluciones iniciales se implementó utilizando un generador aleatorio que sigue una distribución uniforme. El método recibe como parámetros el tamaño de las soluciones $|h|$ (número de días) y la cantidad de platos por día $|H|$. El funcionamiento es sencillo, en cada posición de la solución a generar se introduce un número entero aleatorio desde 0 hasta $|H| - 1$.

Generar una solución supone recorrer un vector por lo que la complejidad temporal del método es $\mathcal{O}(n)$. En el Algoritmo 2.3 se muestra el pseudocódigo del método de generación de soluciones aleatorias.

Algoritmo 2 Generador de soluciones aleatorias

Datos: $|H|, |h|$

Resultado: Nueva solución aleatoria

```
 $S = [|h|]$  for  $i$  in  $\{0, 1, \dots, |h| - 1\}$  do
   $S_i = \text{RandomInteger}(0, |H| - 1)$ 
```

end

return S

Un dato importante a tener en cuenta reside en el hecho de que el generador aleatorio sigue una distribución uniforme para asignar los índices de cada una de las posiciones de la solución, esto implica que sea poco probable ($\frac{1}{|H|^2}$) que se asigne el **mismo índice a dos posiciones adyacentes**. Cabe destacar que precisamente éste es uno de los aspectos que se tienen en cuenta en la evaluación de soluciones y que por ello, la calidad de las soluciones generadas aleatoriamente mediante este método es sumamente baja.

2.4. Búsqueda Local

Partiendo de una solución inicial se aplicará un procedimiento llamado Búsqueda Local. En él, se intentará mejorar la solución modificando en cada iteración 1 elemento de la solución para introducir otro entre los posibles candidatos. Con estas perspectivas, solo cambiaremos de solución cuando se mejore la infactibilidad. Si esta no se ve influenciada negativamente y mejora el objetivo de Huella hídrica o Huella de Carbono, siendo estos los objetivos que se desean, también se cambiará de solución. Para desarrollar este procedimiento lo primero que debe elegirse es como se seleccionará a los elementos vecinos.

Recordar que el problema debe asignar una solución para 20 días y existen 33 primeros platos, 48 segundos platos y 23 guarniciones que hacen un total de 2080 posibles platos; 60 de ellos están ya siendo asignados en la creación de una solución por lo que existen un total de 2020 posibles candidatos o soluciones vecinas.

Existen multitud de estrategias para elegir un elemento entre toda una vecindad. Los tres procedimientos que más destacan son :

- Elegir el vecino que mejore en mayor cuantificación la solución actual.
- Elegir el primer vecino que mejore la solución actual.
- Elegir aleatoriamente entre alguno de los vecinos.

Realizada esta elección y siempre que el algoritmo mejore la solución actual, se volverá a repetir el procedimiento. En caso de que esto no ocurra, el proceso habrá terminado.

El procedimiento se describe a continuación:

Algoritmo 3 Pseudocódigo de Algoritmo Búsqueda Local

Result: Óptimo Local

```

s = s0 /** Generar una solución inicial s0 **/
while Criterio Parada do
  Generar(N(s)) /** Cálculo de vecindades **/
  if no hay una mejor solución entre los vecinos then
    | Parar la búsqueda
  else
    | s = si /** Seleccionamos el mejor vecino si ∈ N(s(i)) **/
  end
end

```

La búsqueda local plantea un problema que suele ser recurrente. Muchas veces el procedimiento se queda atascado en un óptimo local sin obtener la mejor solución posible. Esto hace que se obtengan soluciones de mayor calidad pero no asegura que la solución sea factible. Se describe mejor en la Figura 2.3.

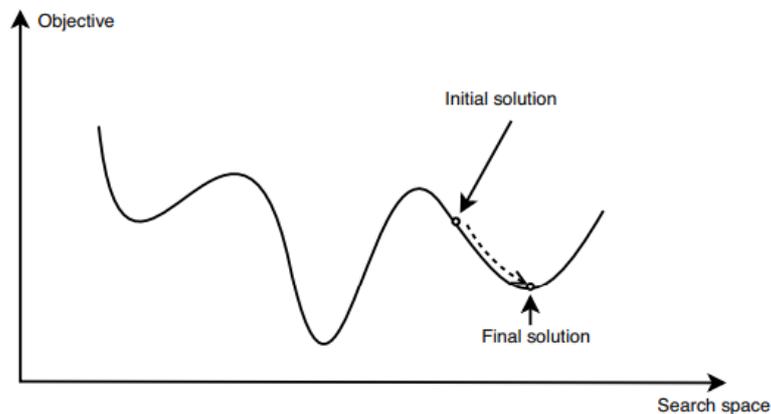


Figura 2.3: Espacio Soluciones de Búsqueda Local.

Aunque el principio básico de una búsqueda local es el que se ha explicado, se han desarrollado dos algoritmos distintos que cumplen los requisitos. Se describen a continuación:

2.4.1. Búsqueda Local Recalculada

Para cada uno de los vecinos, mientras no se cumpla el criterio de parada, se calcula la infactibilidad, objetivo de huella hídrica y huella de carbono total de la posible nueva solución. De esta manera, se realiza un análisis de la infactibilidad de todos los platos cuando se realiza la evaluación para un posible cambio en la solución.

2.4.2. Búsqueda Local Incremental

A diferencia de la Búsqueda Local Recalculada, en este algoritmo solo se calcula la diferencia entre la infactibilidad actual del día en el que se evalúa un posible cambio de plato, antes y después del cambio. El resultado se restará a la infactibilidad actual. Este método ofrece los mismos resultados que la búsqueda local recalculada pero con un número muy inferior de operaciones matemáticas pues no se calcula la infactibilidad y objetivos para el conjunto de 60 platos, sino la diferencia en el día que resulta afectado. Sin embargo, es un procedimiento más complicado de implementar al tener que desarrollar un método que calcule la diferencia de infactibilidad, huella hídrica y huella de carbono entre dos posibles platos para un día concreto. En el caso anterior, simplemente se calculaba el valor total de la solución antes y después del cambio.

2.5. Búsqueda Local Iterada

La calidad de un óptimo local depende de la solución inicial con la que se realice el proceso de búsqueda. La heurística Búsqueda Local Iterada busca escapar de los mínimos locales y mejorar la calidad de sucesidad búsqueda a lo largo del tiempo (Ver Figura 2.4).

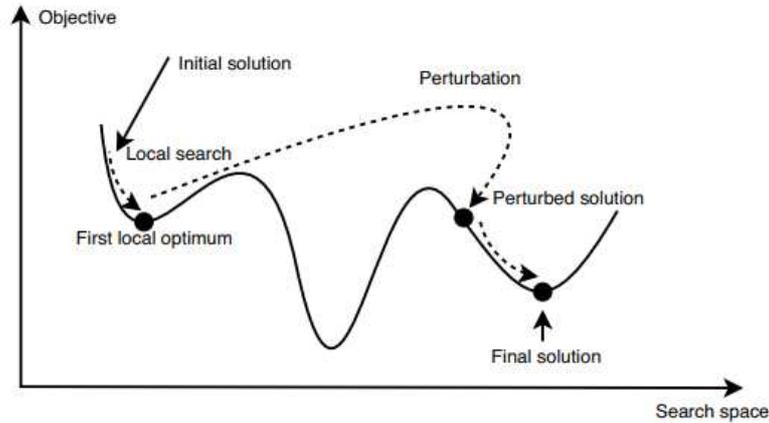


Figura 2.4: Espacio Soluciones Búsqueda Local Iterada.

En una Búsqueda Local Multiarranque[22], la solución inicial se elige siempre de forma aleatoria y se mejora utilizando una Búsqueda Local para generar multitud de óptimos locales distintos. De todos estos, la heurística elige el mejor de ellos como solución. La Búsqueda Local Iterada mejora la clásica Búsqueda Local Multiarranque perturbando el óptimo local de una solución actual.

El proceso de perturbación se base en realizar un determinado número de movimientos aleatorios a partir del óptimo local actual. El método de perturbación debe mantener parte de la solución y realizar una perturbación que sirva para escapar del entorno actual y explorar otro tipo de soluciones en búsqueda de soluciones de mayor satisfacción.

En el caso de encontrar una solución mejor que la actual la reemplazaremos y volveremos a repetir la Búsqueda Local Iterada. La heurística terminará cuando se cumpla una condición de parada, normalmente, cuando se cumpla un tiempo determinado de ejecución de la búsqueda o cuando se haya cumplido un número n de búsquedas.

Algoritmo 4 Pseudocódigo de Algoritmo Búsqueda Local Iterada

Result: Mejor solución encontrada

$s_i = \text{busquedaLocal}(s_0)$; // Aplicar un algoritmo de búsqueda local

while $!(\text{Criterio de parada})$ **do**

$s_i = \text{Perturbacion}(s_*, \text{tipoPerturbacion})$; /** Perturbar la solución actual

$s_i = \text{BúsquedaLocal}(s_i)$; /** Aplicar un algoritmo de búsqueda local $s_i = \text{evaluarSolucion}(s_i,$

s_{mejor})

end

2.6. Algoritmo poblacional

Una heurística poblacional parte de un conjunto de soluciones iniciales a las que considera individuos de una población. Esta población se irá modificando a lo largo de generaciones para intentar encontrar soluciones mejores a la actual.

2.6.1. Elección de la población

Para elegir cuales son las soluciones iniciales se pueden seguir distintas estrategias, dependerá mayormente del tipo de problema que se esté resolviendo.

- **Soluciones aleatoria.** Permite obtener poblaciones bastante dispares que aseguran que se inspeccionan distintos entornos de búsqueda. Sin embargo, si nos encontramos ante problemas complejos, puede llevar a no analizar en profundidad el entorno en el que se encuentra el óptimo global.
- **Estrategia voraz.** Partir de soluciones obtenidas de la ejecución previa de una heurística que haya obtenido óptimos locales sobre los individuos de la población. Si se aplica esta estrategia en problemas de sencilla resolución, se podría estar desperdiciando tiempo de computo. Por otro lado, en problemas complejos suele ser una estrategia recomendable pero podría tener un efecto negativo sobre la diversidad de los individuos que haría que la heurística converja rápidamente.

Sobre la población se debe realizar un proceso de reproducción basado en aplicar operadores de variación imitando una mutación que altere las soluciones actuales. Una estrategia muy usada se basa en elegir dos individuos de forma aleatoria y sobre estos, evaluar cual ofrece una mejor solución. Se vuelve a repetir el proceso sobre dos nuevos individuos incluyendo a los dos ya evaluados, que pueden volver a participar en la elección. Los dos individuos elegidos finalmente compartirán sus soluciones realizando diferentes tipos de cruce para generar dos nuevas soluciones hijas.

2.6.2. Estrategias de cruce

Existen diferentes tipos de cruce como son el Crossover y el Uniforme.

- **Cruce Crossover.** Se elige un punto de corte de forma aleatoria sobre el tamaño de la solución. Los individuos tendrán hasta el punto elegido la solución parcial del individuo padre. El resto de la solución se rellenará con el otro elemento padre. Por ejemplo:
Para dos soluciones distintas para ocho días representadas de la forma ♀♀♀♀♀♀♀♀ y ♂♂♂♂♂♂♂♂, determinando el punto de cruce el día tres, las soluciones resultantes del cruce serán las de combinar los tres primeros días de la primera solución y el resto de la segunda solución y por otro lado, la elección de los tres primeros días de la segunda solución y el resto de la primera solución. Con la representación anterior las soluciones hijas resultantes serán ♀♀♀♂♂♂♂♂♂♂♂ y ♂♂♂♀♀♀♀♀♀♀.
- **Cruce Uniforme.** Para cada uno de los elementos de las soluciones padres se elegirá aleatoriamente si se asigna un padre u otro. Por ejemplo:
Para dos soluciones distintas para ocho días representadas de la forma ♀♀♀♀♀♀♀♀ y ♂♂♂♂♂♂♂♂, las soluciones resultantes del cruce se elegirá evaluando para cada uno de los días que solución asignada en los padres, se elegirá. Con la representación anterior, las soluciones hijas resultantes podrían ser:
 - ♀♂♂♀♂♂♂♀ eligiendo los días uno, cuatro, siete y ocho del primer padre y el resto del segundo padre.
 - ♀♀♀♂♂♂♀♀♀ eligiendo los días uno, dos, tres, seis y ocho del primer padre y el resto del segundo padre.

Al implementar una estrategia de cruce debe asegurarse el concepto de herencia. El operador de cruce debe mezclar el material genético o solución de ambos individuos padres.

El cruce y la mutación deben ser elementos complementarios, siendo la mutación un elemento de diversificación. El número de hijos resultantes para una estrategia de cruce debe definirse. Puede quererse aumentar, disminuir o mantener la misma cantidad de elementos en la población. Cuando este número de individuos hijos haya sido generado, se repetirá el proceso con esta nueva población hasta cumplir un criterio de parada.

- **Estrategia Estática.** Antes de empezar la ejecución de la heurística se conoce cuándo terminará la ejecución. Normalmente utilizando un máximo tiempo de ejecución, un número máximo de iteraciones o límite en la asignación de recursos de la CPU.
- **Estrategia Adaptativa.** No se puede determinar a priori cuánto durará la ejecución de la heurística. Se puede usar como criterio de parada la ejecución de un determinado número de ejecuciones sin que se mejore la solución actual.

Con todo esto, el pseudocódigo de un algoritmo evolutivo es el siguiente.

Algoritmo 5 Pseudocódigo de Algoritmo Evolutivo

Result: Mejor solución encontrada

$P = P_0$; /*Generación de población inicial*/

$t = 0$;

while *!(Criterio de parada)* **do**

Evaluar(P_t);

$P'_t = \text{SeleccionarPoblacion}(P_t)$; /* Seleccionamos una nueva poblacion */

$P'_t = \text{Reproduccion}(P'_t)$; /* Seleccionamos una nueva poblacion */

evaluar(P'_t);

$P_{t+1} = \text{Reemplazo}(P_t, P'_t)$; /* Seleccionamos una nueva poblacion */

$t = t + 1$;

end

Salida: Mejor solución o población encontrada

2.6.3. Elitismo

En terminos generales el elitismo consiste en almacenar la mejor solución generada durante el proceso de selección si esta se encuentra entre los individuos padres. Estos se perderían en la siguiente generación. Este proceso garantiza obtener un comportamiento de optimización incremental. Durante el proceso nunca se perderá la mejor solución. Entre los inconvenientes, si se aplica un alto elitismo en la optimización de los individuos se podría producir una convergencia prematura en la generación de nuevas soluciones perdiendo la diversificación.

Capítulo 3

Resultados experimentales

En este capítulo se detallan los resultados más relevantes obtenidos en toda la experimentación y los estudios que se han hecho para conocer cuales son las variables heurísticas más adecuadas para lograr soluciones factibles sobre el problema de asignación de menús.

3.1. Estudio entre Búsqueda Local Recalculada y Búsqueda Local Incremental

Habiendo implementado dos Búsqueda Locales diferentes y habiendo sido explicados en el punto 2.4, se ha realizado un estudio para conocer cual de las dos estrategias ofrece soluciones en menor tiempo de ejecución. Para ello, se ha realizado un estudio en el que se ha realizado 30 ejecuciones utilizando las mismas semillas sobre ambos algoritmos.

La solución final de ambos algoritmos debe ser la misma tanto intentando optimizar la huella de agua, la huella de carbono o ambos. Sin embargo, se presupone que la Búsqueda Local Incremental debe ser inferior a la Búsqueda Local Recalculada con respecto a los tiempos de ejecución ya que utiliza un menor número de comparaciones para calcular el valor de infactibilidad.

En la Figura 3.1 se describen los resultados de infactibilidad que se obtienen de la ejecución del algoritmo recalculado e incremental sobre la optimización de huella hídrica, huella de carbono y ambos objetivos. Ninguno de ellos logra obtener soluciones factibles. Se demuestra que ambos algoritmos ofrecen los mismos resultados.

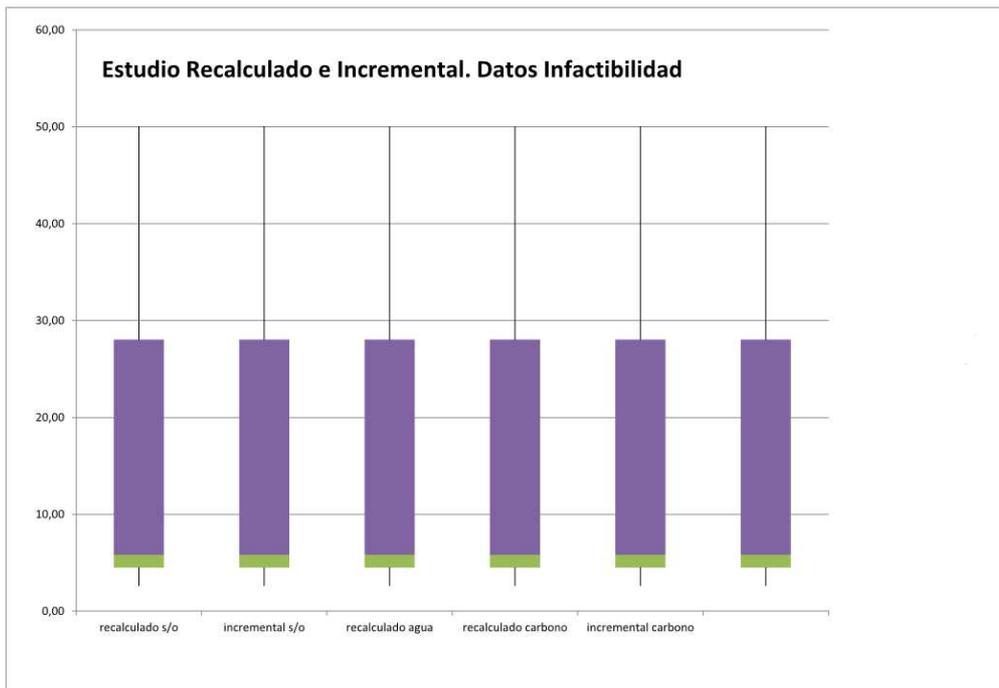


Figura 3.1: Estudio sobre Algoritmo Recalculado e incremental. Datos Infactibilidad.

En la Figura 3.2 se muestran los tiempos de ejecución de los algoritmos recalculados e incremental.

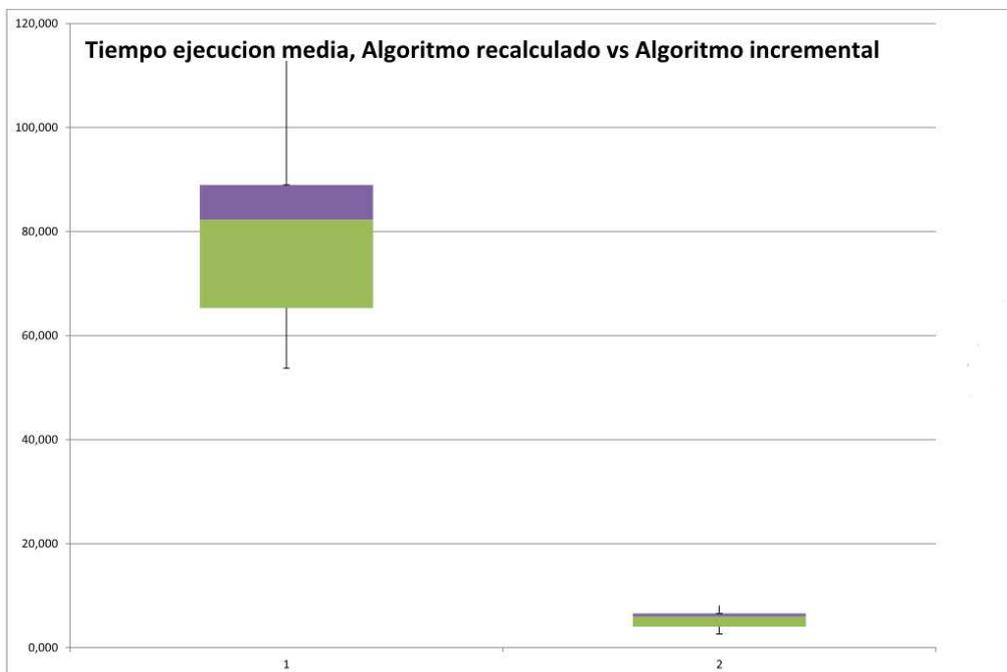


Figura 3.2: Estudio sobre Algoritmo Recalculado e incremental. Tiempos de ejecución.

Se demuestra que el algoritmo mas eficiente de los dos implementados es la Búsqueda Local Incremental ya que su tiempo de ejecución es de media 12 veces más rápida que la Búsqueda Local Recalculada.

3.2. Estudio sobre Búsqueda Local Iterada

Habiendo elegido en el proceso anterior la búsqueda local incremental se ha implementado una estrategia búsqueda local iterada utilizando la búsqueda local incremental.

Para conocer que valor de perturbación debe utilizarse en la Búsqueda Local Iterada para garantizar soluciones de mejor calidad, se ha hecho un estudio sobre 30 ejecuciones distintas y 5 perturbaciones. Se valorará qué perturbación es la adecuada y se conocerá el nivel de infactibilidad de las soluciones. Se propone una perturbación del 5 %, 7.5 %, 10 %, 12.5 % y 15 %. Se ha utilizado un criterio de parada basado en el tiempo, que ha sido de 1 hora. Sin embargo, las siguientes gráficas solo representan el tiempo de ejecución de los primeros 600 segundos ya que el resto de los datos representan datos muy cercanos a 0. En la Figura 3.3 se muestran las soluciones obtenidas para cada una de las perturbaciones empezando por la 5 % y avanzando de forma incremental hasta terminar en 15 %.

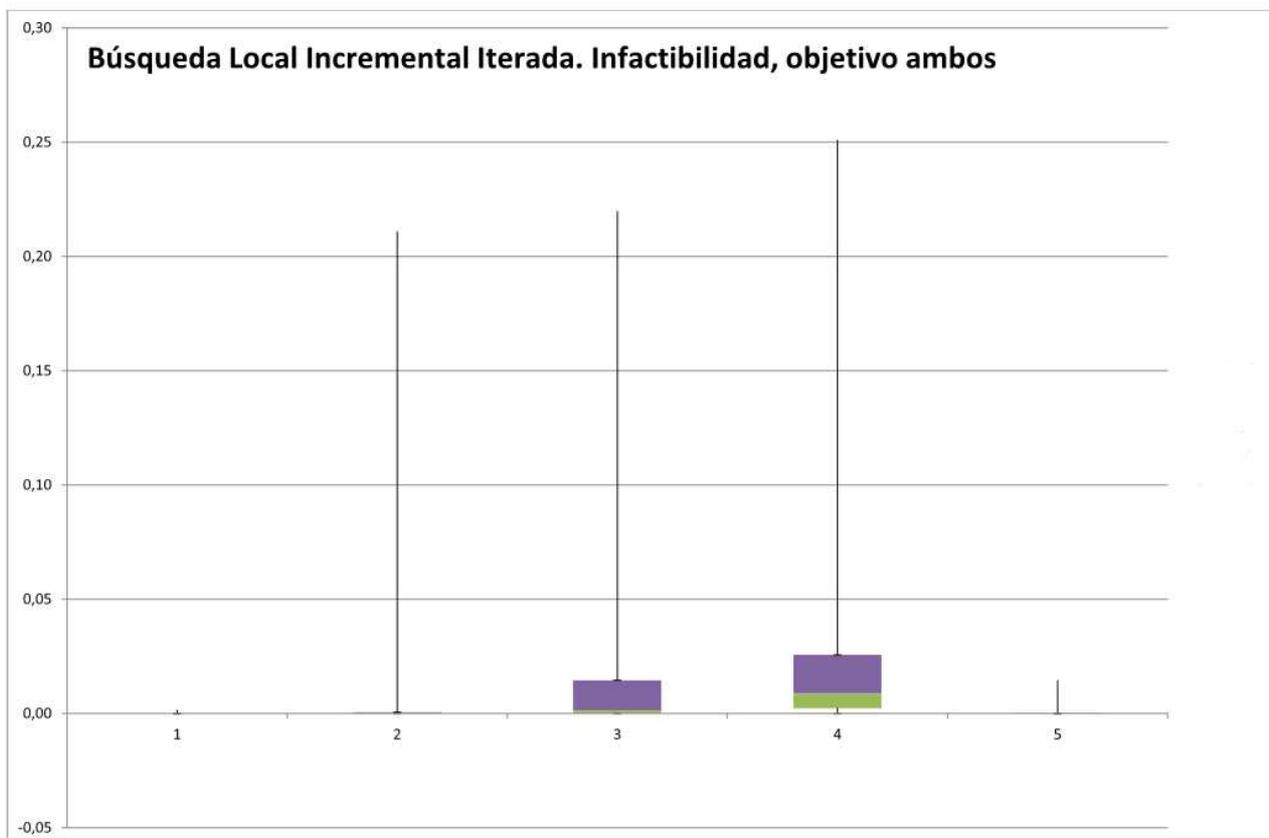


Figura 3.3: Búsqueda Local Incremental Iterada. Datos de Infactibilidad

Se detecta que el algoritmo consigue algunas soluciones factibles en casi todas las perturbaciones. Las mejores soluciones a priori se encuentran en la perturbación 5 %, 7,5 % y 15 %. Sin embargo, todas las soluciones se encuentran en valores inferiores a 0,25 para el valor de infactibilidad, lo que representa diferencias mínimas entre las distintas perturbaciones y en muchas ejecuciones se cumple el conjunto de restricciones, infactibilidad 0.

En la siguiente gráfica se expone el estudio de infactibilidad sobre tiempo para la Búsqueda Local Iterada con perturbación 5 %. las gráficas para la perturbacion 7.5 %, 10 %, 12.5 % y 15 % se encuentran en el repositorio Github [15].

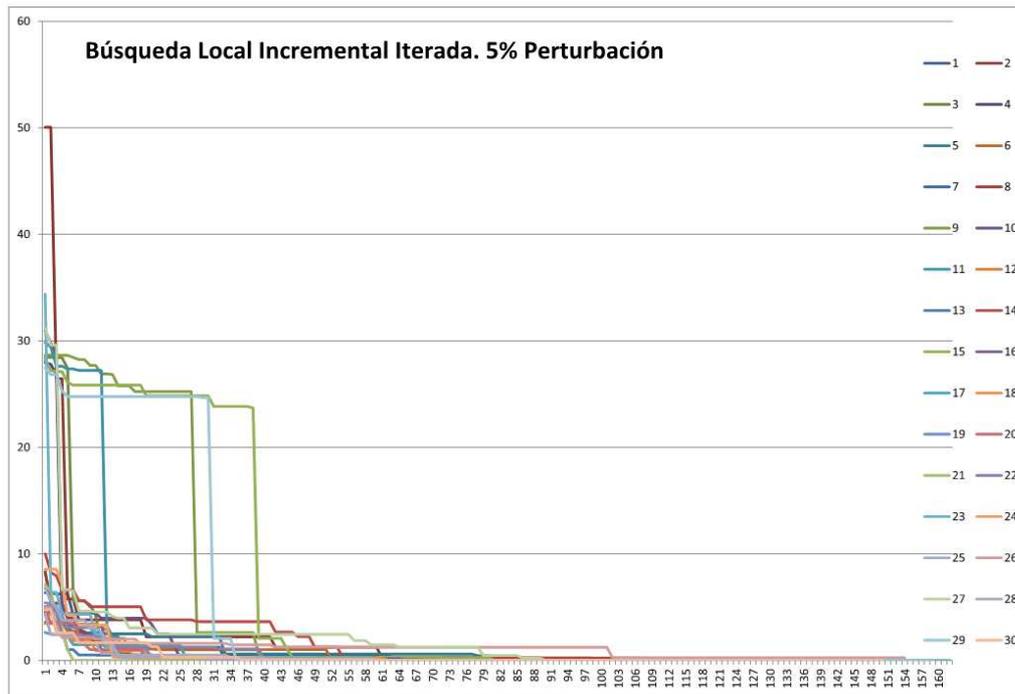


Figura 3.4: Tiempo de Ejecución, Algoritmo Búsqueda Local Incremental Iterada. Perturbación 5 %.

Analizando el tiempo que tarda cada uno de los algoritmos en lograr las mejores soluciones, queda patente que la mejor velocidad de convergencia se encuentra con la perturbación 5% consiguiendo soluciones de buena calidad en torno a la iteración 60 de 167 que representa el segundo 215 (3 minutos y medio). Esto no garantiza que para cualquier ejecución se consigan con total seguridad soluciones factibles en ese momento. Para este estudio se asegura soluciones factibles para el segundo 1059, casi 18 minutos de tiempo de ejecución.

3.3. Estudio sobre Algoritmo Evolutivo

Utilizando la búsqueda local incremental con perturbación 5% se ha decidido implementar un algoritmo evolutivo. Se ha realizado un estudio para conocer cuales son los valores de infactibilidad y los valores de huella hídrica, huella de carbono cuando se intentan minimizar 1 de estos o ambos. La intención es conocer cuan cercano se encuentra este algoritmo de la solución óptima proporcionada por el estudio previo.

Se ha realizado un estudio sobre distintas poblaciones aplicando distintos objetivos. En todos ellos se ha aplicado el elitismo para mantener la mejor solución de la población después de evaluar los elementos padres e hijos. En todos ellos, se ha utilizado un criterio de parada de 10 horas. Se han hecho los siguientes estudios:

- 10 Individuos en la población. 60 segundos de ejecución para cada Búsqueda Local Iterada.
- 4 Individuos en la población. 90 segundos de ejecución para cada Búsqueda Local Iterada.
- 4 Individuos en la población. 30 segundos de ejecución para cada Búsqueda Local Iterada.

Recordar que para búsquedas locales iteradas ya se habían encontrado algunas soluciones factibles para este problema. El algoritmo evolutivo implementado asegurará que se consigan siempre soluciones factibles. A continuación se presentan para cada estudio, los datos de infactibilidad, huella hídrica y huella de carbono. (Ver Figuras 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13, 3.14, 3.15 y 3.16).

3.3.1. Población de 10 individuos, 60 segundos de búsqueda local iterada.

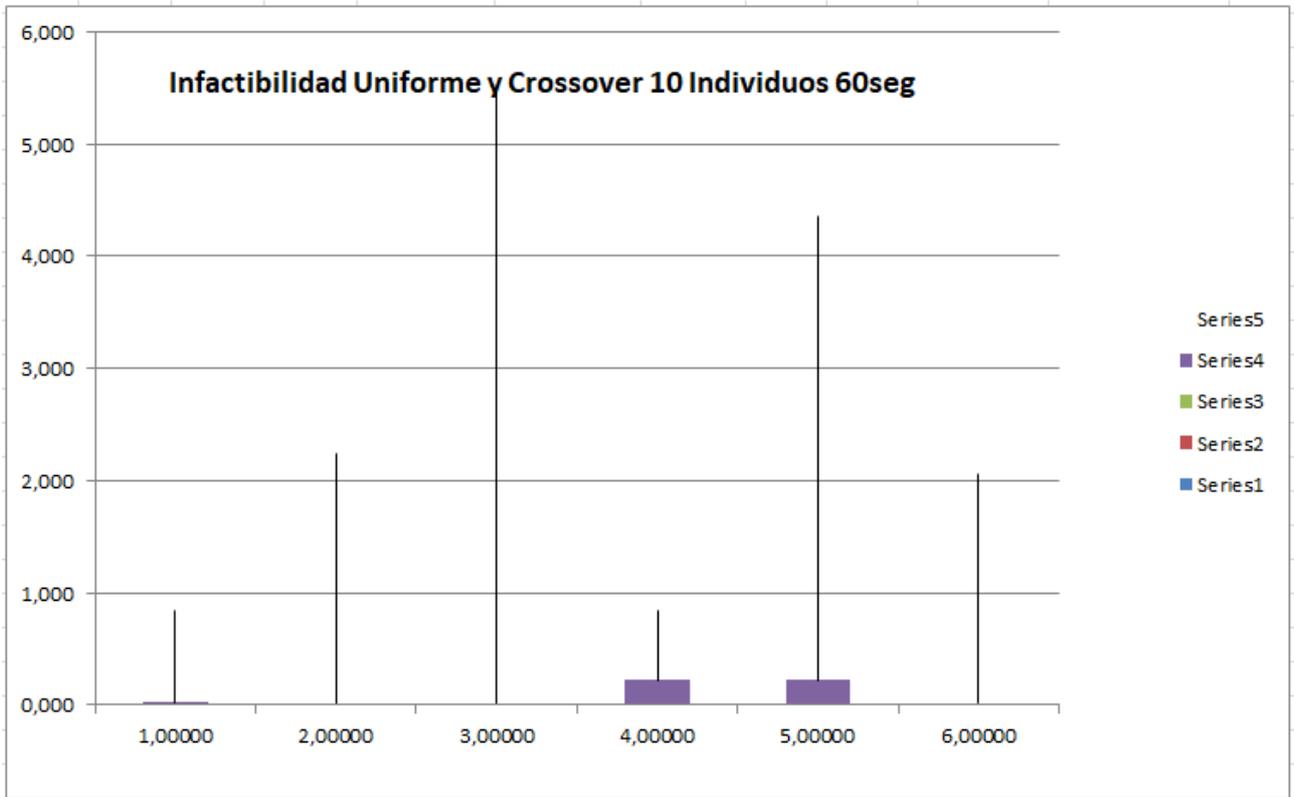


Figura 3.5: Infactibilidad Grafo BoxPlot, 10 Individuos, 60 segundos

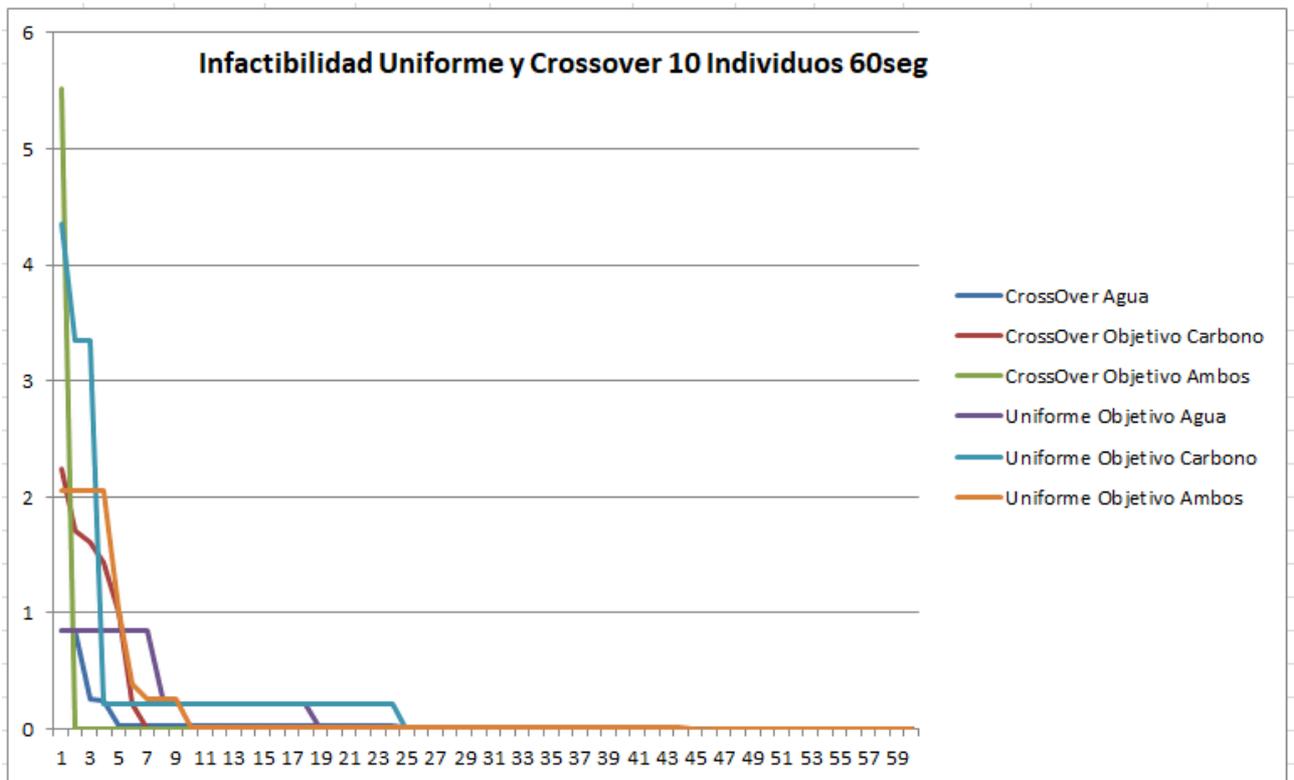


Figura 3.6: Infactibilidad Grafo Tiempo, 10 Individuos, 60 segundos

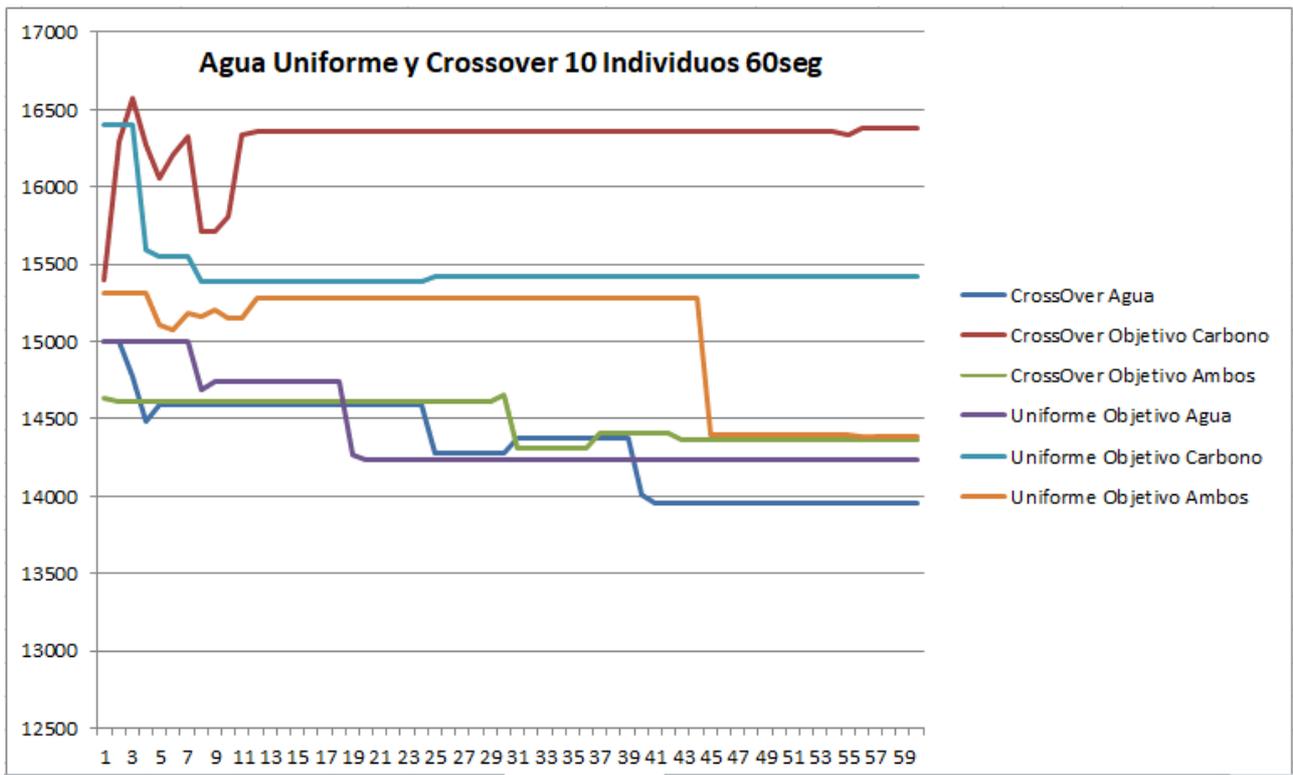


Figura 3.7: Agua Grafo Líneas 10 Individuos, 60 segundos

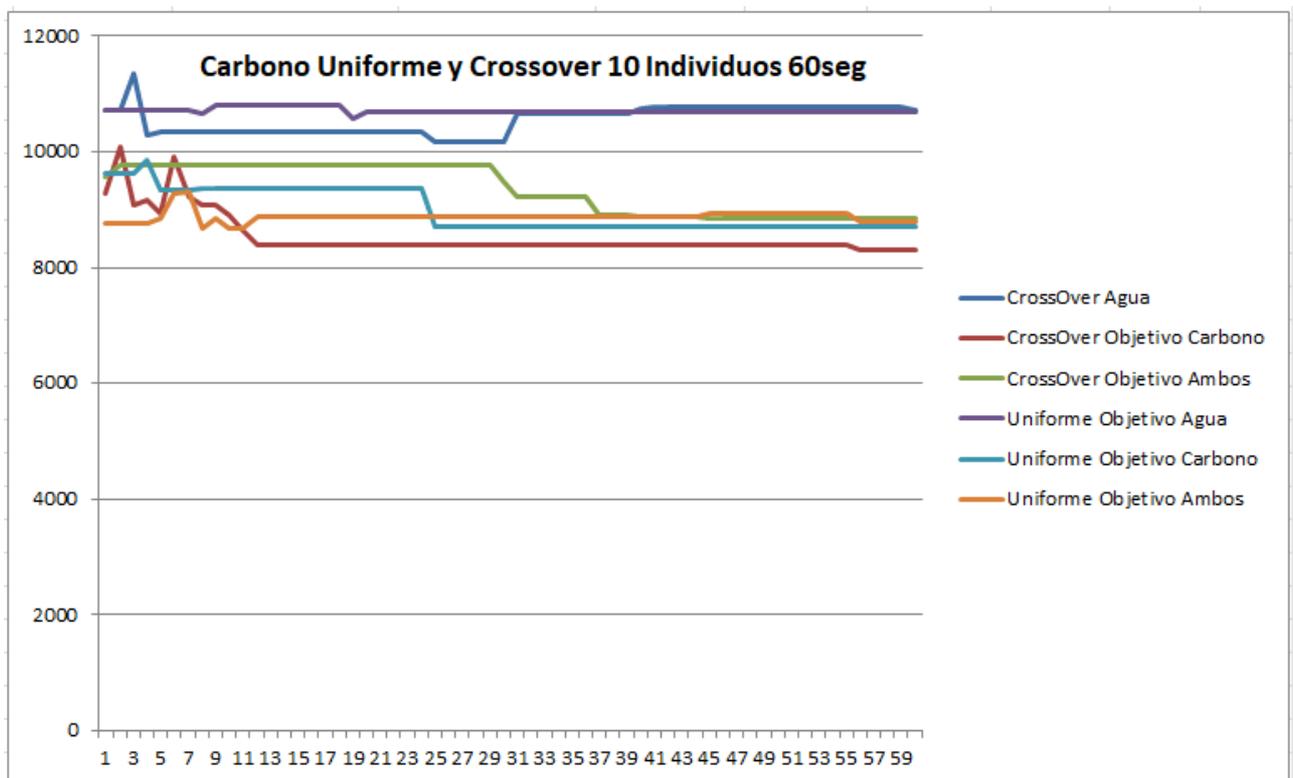


Figura 3.8: Carbono Grafo Líneas 10 Individuos, 60 segundos

Con este estudio, se demuestra que se consiguen soluciones factibles a partir del segundo 600 intentando optimizar ambos objetivos con una estrategia de cruce crossover. Se consiguen soluciones aceptables para los valores de huella hídrica a partir del segundo 18267 con un valor de 14376 utilizando el cruce Crossover con objetivo de minimizar la huella hídrica. El algoritmo termina con un valor final

de 13956,3. Se consigue una solución aceptable para la huella de carbono en el segundo 6749 con un valor de 8392,71 utilizando el cruce crossover con objetivo huella hídrica. El algoritmo termina con un valor de 8315,83. La solución final obtenida se acerca a la solución óptima pero tarda casi cinco horas en lograr valores cercanos para la optimización de Huella Hídrica y 2 horas para la Huella de Carbono.

3.3.2. Población 4, 90 segundos

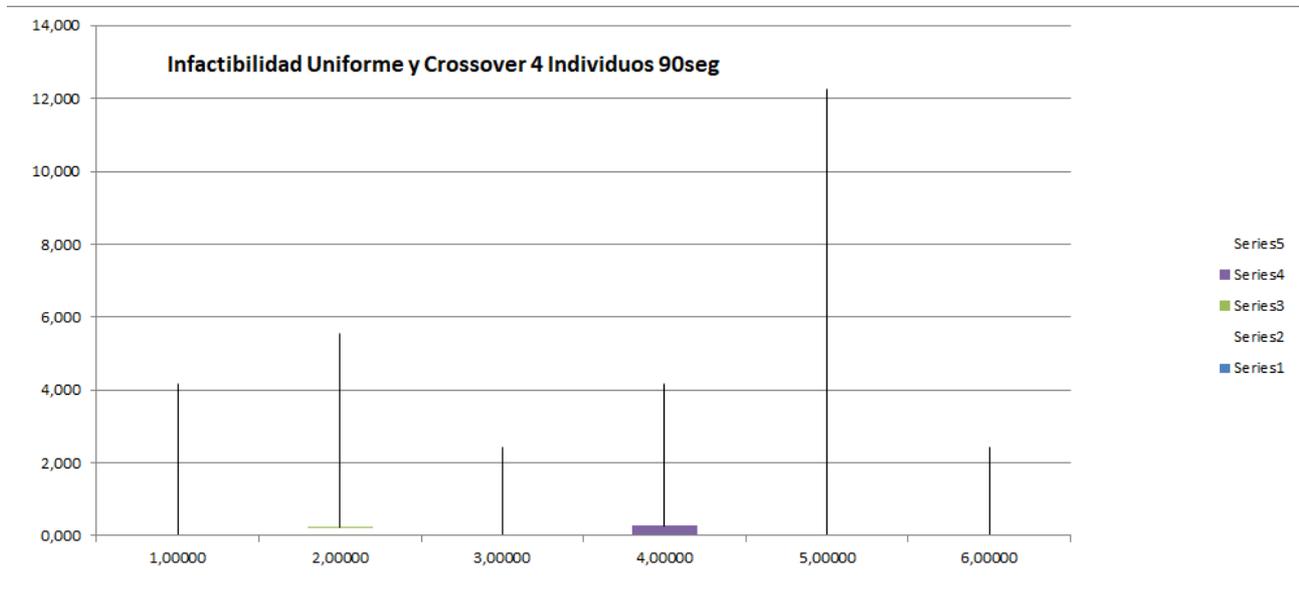


Figura 3.9: Infactibilidad Grafo BoxPlot 4 Individuos, 90 segundos

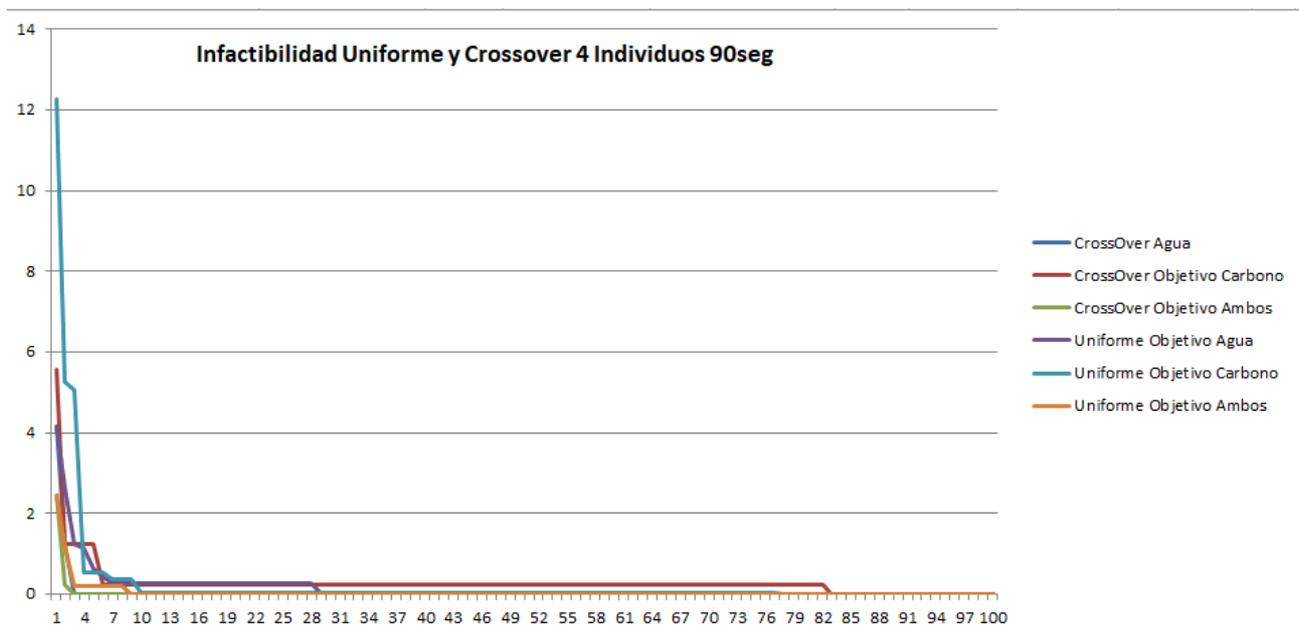


Figura 3.10: Infactibilidad Grafo Tiempo 4 Individuos, 90 segundos

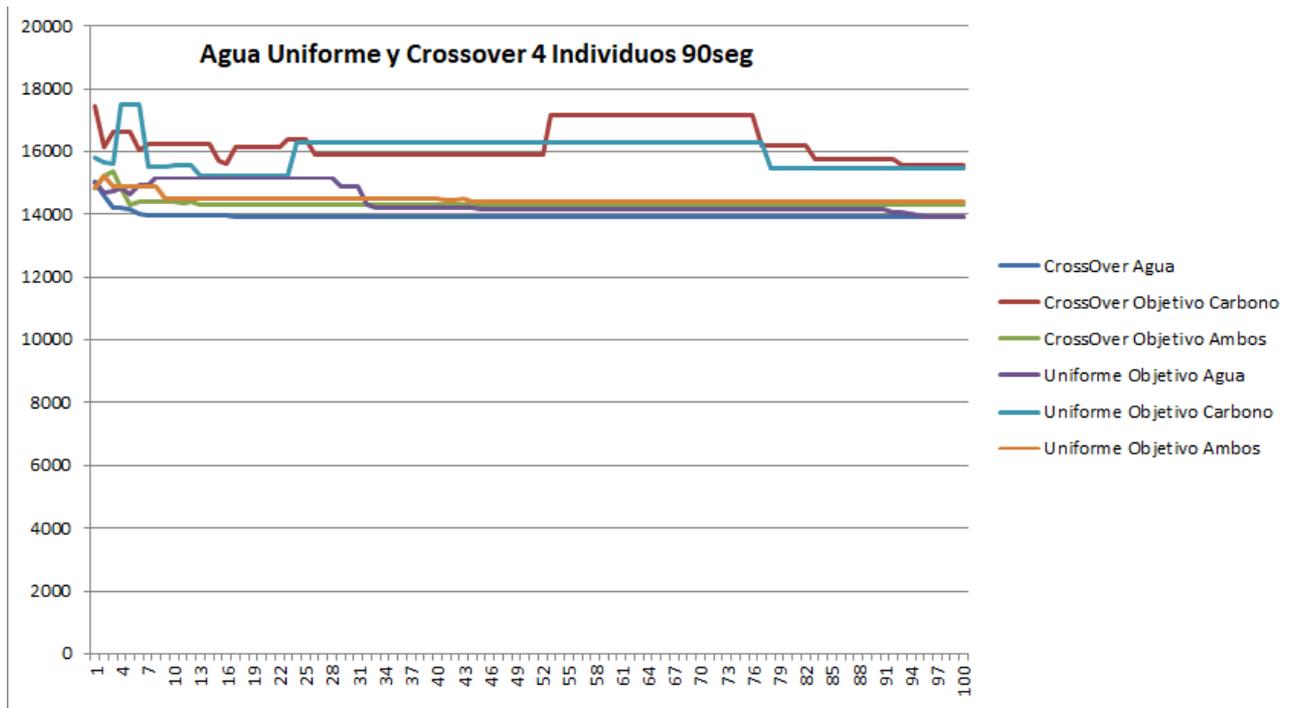


Figura 3.11: Agua Grafo Líneas 4 Individuos, 90 segundos

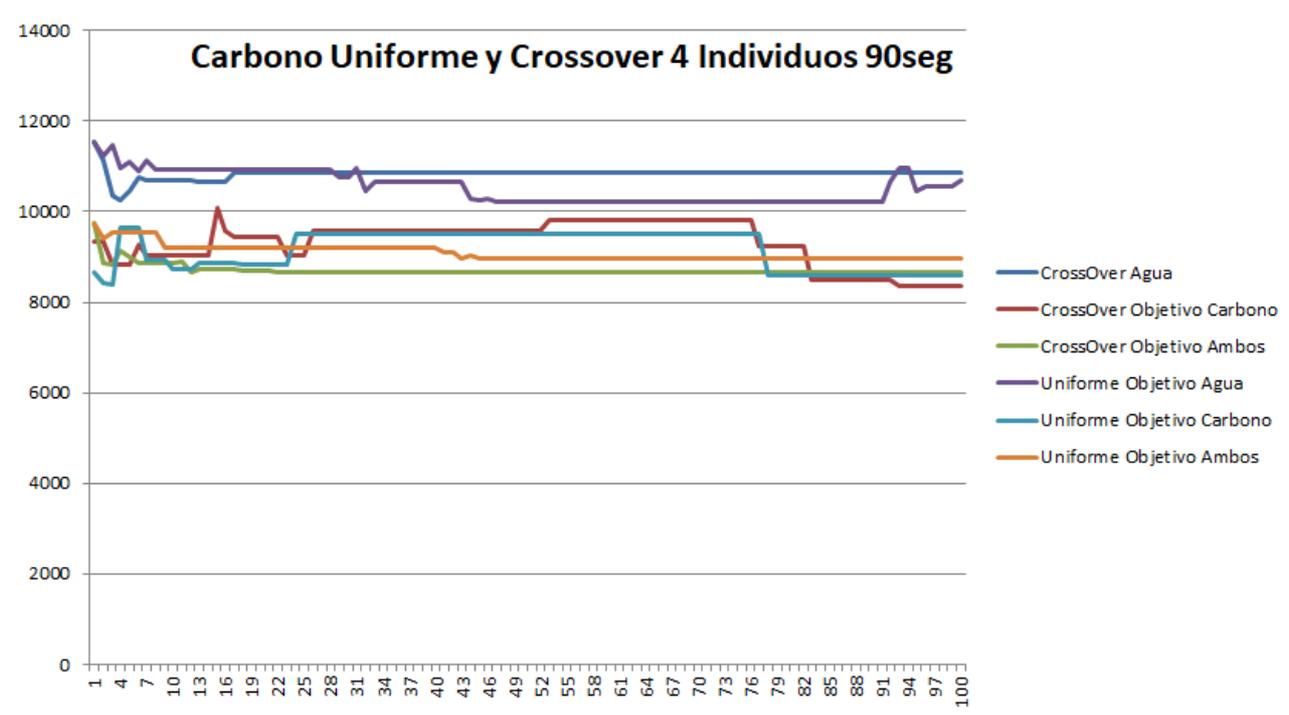


Figura 3.12: Carbono Grafo Líneas 4 Individuos, 90 segundos

Con este estudio, se demuestra que se consiguen soluciones factibles a partir del segundo 724 intentando optimizar ambos objetivos con una estrategia de cruce Crossover. Se consiguen soluciones aceptables para los valores de huella hídrica a partir del segundo 724 con un valor de 14207,5 utilizando el Cruce Crossover con objetivo de minimizar la huella hídrica. El algoritmo termina con un valor final de 13941,9. Se consigue una solución aceptable para la huella de carbono en el segundo 6217 con un valor de 8392,71 utilizando el cruce Crossover, objetivo ambos. El algoritmo termina con un valor de 8315,83 para el cruce Crossover con objetivo minimización de huella de carbono. La solución final obtenida se acerca a la solución óptima y mejora considerablemente el tiempo de convergencia del

anterior algoritmo poblacional. El algoritmo tarda 12 minutos en conseguir soluciones cercanas a la óptima para la huella hídrica y 2 horas para la huella de carbono.

3.3.3. Población de 4 individuos, 30 segundos búsqueda

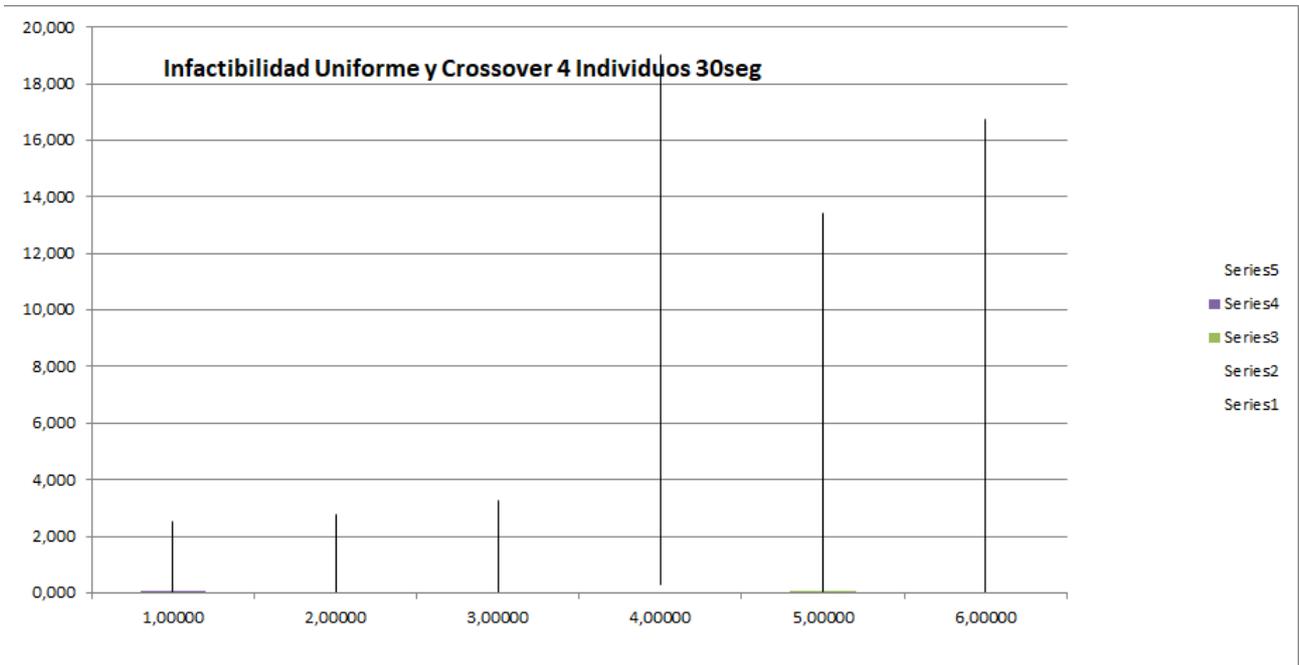


Figura 3.13: Infactibilidad Grafo BoxPlot 4 Individuos, 30 segundos

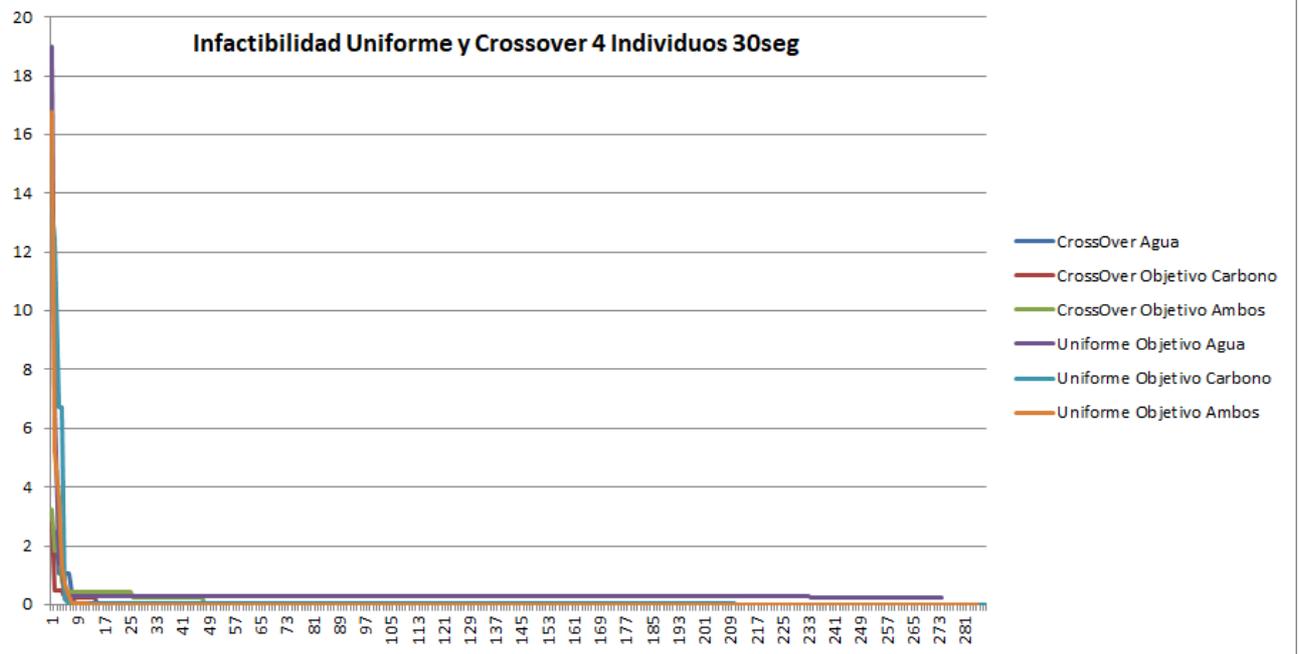


Figura 3.14: Infactibilidad Tiempo 4 Individuos, 30 segundos

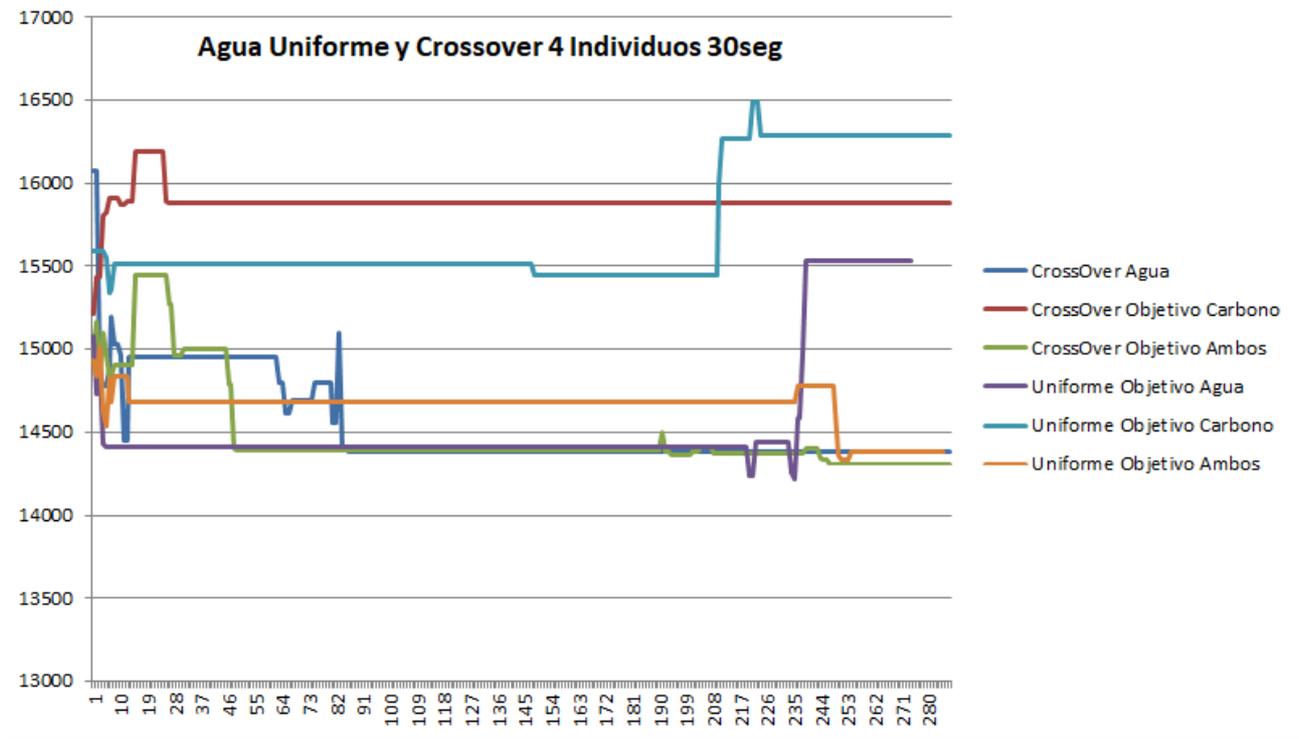


Figura 3.15: Agua Grafo Líneas 4 Individuos, 30 segundos

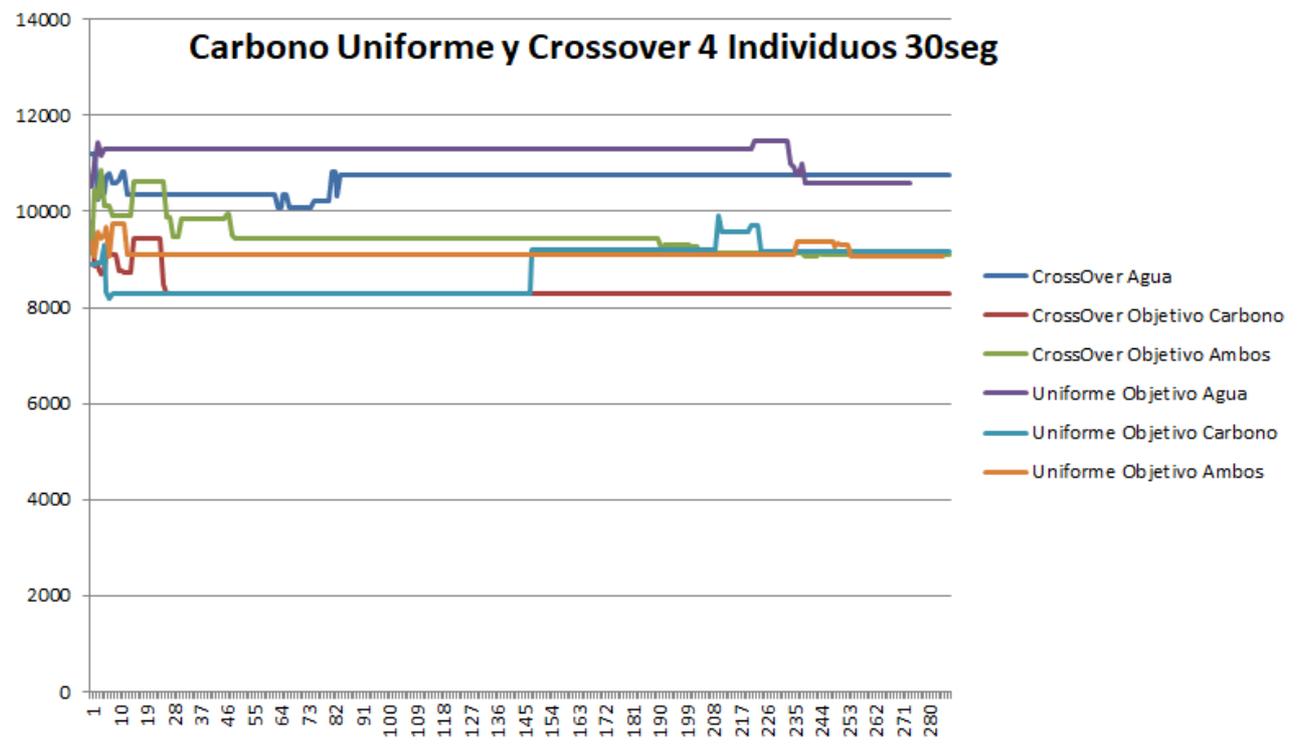


Figura 3.16: Carbono Grafo Líneas 4 Individuos, 30 segundos

En este estudio, queda evidenciado que los datos son poco fiables ya que en muchos casos no se consiguen soluciones factibles. Sobre los datos validos se deduce que se logra factibilidad 0 en el segundo 3010 para el Cruce Crossover objetivo carbono. Para la optimización de la huella hídrica se consiguen soluciones aceptables en el segundo 10407 con valor 14378,7 terminando con valor 14398,7. Se consigue una buena solución para el objetivo huella de carbono en el segundo 3010 con valor 8476,52 terminando con valor 8299,2.

Tras estos estudios queda demostrado que entre los cruces, el que ofrece soluciones de mayor calidad a lo largo de todas las ejecuciones es el cruce Crossover.

En cuanto a los valores de huella hídrica y de CO_2 , se consiguen soluciones de mayor calidad cuando se utiliza una población con pocos individuos pero se utiliza un tiempo de ejecución para la Búsqueda Local Iterada cercano a los 60 segundos. Se han conseguido soluciones factibles para el algoritmo poblacional utilizando 10 individuos en unos 12 minutos (724 segundos). Uno de los elementos consigue solución factible ejecutando la Búsqueda Local Iterada durante un minuto y medio.

A partir de estos resultados, si se aplican otro tipo de procedimientos que permitan diversificar el análisis filtrando algunos elementos o paralelizando algunos cálculos, posiblemente se mejoraría aun más la solución actual.

Capítulo 4

Conclusiones y líneas futuras

En este trabajo se ha conseguido analizar un problema real de programación lineal, concretamente un problema de asignación de menús en comedores escolares con restricciones e intentando minimizar dos objetivos distintos. Se ha desarrollado una estrategia de resolución basado en la programación de metaheurísticas y algoritmos aproximados analizando distintas perspectivas, realizando estudios y partiendo de soluciones aleatorias hasta conseguir soluciones factibles y de buena calidad para ambos objetivos. La utilización de estas técnicas permite escalar el problema permitiendo que ante entrada de datos mucho mayores, no empeoren significativamente el tiempo de ejecución, hecho que en otro tipo de estrategias, es lo más usual.

Con el algoritmo poblacional se ha conseguido analizar diferentes soluciones factibles cercanas a la solución óptima global. En muchos de los casos, se han conseguido tiempos prudencialmente bajos. Sin embargo, para conseguir encontrar la solución óptima global, deben aplicarse mayor número de procedimientos de los actualmente desarrollados.

Por último, ha resultado bastante interesante poder analizar el comportamiento de distintas heurísticas sobre un problema real. Conocer la implicación de cada una de las restricciones y la importancia de las diferentes variables de decisión como son la perturbación, estrategia de cruce o elitismo resultan fundamentales para poder resolver eficientemente este tipo de problemas,

Sobre el actual estado del proyecto podrían realizarse nuevos estudios para intentar mejorar el tiempo de ejecución final y lograr que los valores de huella hídrica y huella de carbono fueran óptimos globales además de minimizar aun más el tiempo necesario para obtener estas soluciones. Por ejemplo, en la heurística poblacional se podrían aplicar un método de diversificación eligiendo los individuos factibles cuya elementos de la solución se alejen más entre sí. Esta estrategia podría mejorar el tiempo en el que se determina la mejor solución actual. Si un elemento de la población ya contempla esa evaluación, deberá elegirse otro vecino. De esta manera, se buscaría analizar el mayor número distinto de soluciones factibles existentes. Por otro lado, se podrían aplicar medidas de paralelización de tareas entre procesadores utilizando MPI o OpenMP. Se podría mejorar exponencialmente el tiempo de ejecución del algoritmo poblacional actual. Por último, y si con esto no consiguiéramos obtener la solución óptima en un tiempo razonable se podría hacer uso del algoritmo genético NSGA-II [1] (Non-dominated Sorting Genetic Algorithm II), un procedimiento eficiente para la búsqueda de soluciones óptimas partiendo de soluciones óptimas locales.

Capítulo 5

Summary and Conclusions

In this work it has been possible to analyze a real problem of linear programming, specifically a problem of assigning menus in school canteens with restrictions and trying to minimize two different objectives. A resolution strategy based on the programming of metaheuristics and approximate algorithms has been developed, analyzing different perspectives, carrying out studies and starting from random solutions until achieving feasible and good quality solutions for both objectives.

The use of these techniques allows scaling the problem that requires much larger data entry, without worsening the execution time, a fact that in other types of strategies, is the most common. With the population algorithm, it has been possible to analyze different feasible solutions close to the global optimal solution. In many cases, prudently low times have been achieved. However, in order to find the optimal global solution, a greater number of procedures than those currently developed must be applied.

Finally, it has been quite interesting to analyze the behavior of different heuristics on a real problem. Knowing the implication of each of the restrictions and the importance of the different decision variables such as disturbance, crossing strategy or elitism are essential to efficiently solve these types of problems. On the current state of the project, new studies could be carried out to try to improve the final execution time and to make the water footprint and carbon footprint values global optimal, and further minimize the time needed to obtain these solutions. For example, in population heuristics, a diversification method could be applied by choosing the feasible individuals whose elements of the solution are further apart from each other. This strategy could improve the time in which the best current solution is determined. If an element of the population already contemplates this evaluation, another neighbor must be chosen. In this way, we would seek to analyze the largest number of existing feasible solutions. On the other hand, we can use task parallelization measures among processors using MPI or OpenMP. The execution time of the current population algorithm could be exponentially improved. Finally, if we fail to obtain the optimal solution in a reasonable time, the NSGA-II [1] (Non-dominated Sorting Genetic Algorithm II) genetic algorithm could be used. This is an efficient procedure to find optimal solutions based on local optimal solutions.

Bibliografía

- [1] A fast and elitist multiobjective genetic algorithm: NSGA-II. <https://ieeexplore.ieee.org/document/996017> [Online; accessed 07 September 2019].
- [2] Aplicación WWF, calculo de huella hídrica y CO2. improntawwf.it/, [(Online; accessed 07 September 2019)].
- [3] Cambio climático: por qué el consumo de carne y lácteos tiene tanto impacto. <https://www.bbc.com/mundo/noticias-49279749> [Online; accessed 07 September 2019].
- [4] Fred Glover. https://en.wikipedia.org/wiki/Fred_W._Glover, [(Online; accessed 07 September 2019)].
- [5] Huella de Carbono. <https://www.sostenibilidad.com/cambio-climatico/claves-para-entender-la-huella-de-carbono/>, [(Online; accessed 07 September 2019)].
- [6] Huella Hídrica. <http://www.huellahidrica.org/?page=files/home>, [(Online; accessed 07 September 2019)].
- [7] Huella Hídrica 2. <https://www.iagua.es/noticias/espana/aquafides/16/04/28/que-es-huella-hidricapara-que-sirve>, [(Online; accessed 07 September 2019)].
- [8] Huella Hídrica 3. es.wikipedia.org/wiki/Huella_h%C3%ADdrica, [(Online; accessed 07 September 2019)].
- [9] LateX. https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings, [(Online; accessed 07 September 2019)].
- [10] Manual Huella Hídrica. <https://waterfootprint.org/media/downloads/ManualEvaluacionHH.pdf>, [(Online; accessed 07 September 2019)].
- [11] Números difusos. https://www.researchgate.net/publication/257139258_MANIPULACION_EFICIENTE_DE [(Online; accessed 07 September 2019)].
- [12] Programación lineal. https://www.programacionlineal.net/programacion_lineal.html, [(Online; accessed 07 September 2019)].
- [13] Trabajo Previo. <https://www.researchgate.net/requests/r52390257>, [(Online; accessed 07 September 2019)].
- [14] AMPL, 2008. <https://www.android.com/> [Online; accessed 07 September 2019].
- [15] Documentacion Trabajo Fin de Máster. <https://github.com/PAL-ULL/tfm-ivan-garcia> [Online; accessed 07 September 2019].
- [16] Donald Knuth. Universidad de Tuscia. <http://www.unitus.it/>, [(Online; accessed 07 September 2019)].
- [17] Doxygen. DOXYGEN. <http://www.doxygen.nl/>, [(Online; accessed 07 September 2019)].

- [18] Pablo Pedregal Ricardo García y Natalia Alguacil Enrique Castillo, Antonio J. Conejo. *Formulación y Resolución de Modelos de Programación Matemática en Ingeniería y Ciencia*. <http://www.dia.fi.upm.es/~jafernán/teaching/operational-research/LibroCompleto.pdf>. Universidad De Castilla – La Mancha, 1986.
- [19] Consejo Europeo de Información Alimentaria. <https://www.eufic.org/en> [Online; accessed 07 September 2019].
- [20] Github. <https://github.com/> [Online; accessed 07 September 2019].
- [21] F. Glover. *Future paths for integer programming and links to artificial intelligence*. *Computers and Operation Research*. Computers OR, 1986.
- [22] J. Marcos Moreno Vega, Rafael Martí. Métodos Multi-arranque. <https://www.uv.es/rmarti/paper/docs/multi1.pdf>, [(accessed 07 September 2019)].
- [23] Jiří Jaromír Klemeš, Cecília Maria Villas Bôas de Almeida, Yutao Wang. *Journal of Cleaner Production*. <https://www.journals.elsevier.com/journal-of-cleaner-production> [(Online; accessed 07 September 2019)].
- [24] Benvenuti L, De Santis A, Santesarti F, and Tocca L. An optimal plan for food consumption with minimal environmental impact: the case of school lunch menus. *Journal of Cleaner Production*, 129(10):704–713, 2016.
- [25] MUTTI PARMA. MUTTI PARMA. <https://www.mutti-parma.com>, [(Online; accessed 07 September 2019)].
- [26] El-Ghazali Talbi. *Metaheuristics: From Design to Implementation*. WILEY A JOHN WILEY SONS, PUBLICATION, 2009.
- [27] Universidad de Malaga. Introducción a la Optimización Evolutiva Multiobjetivo. <http://neo.lcc.uma.es/pdf-charlas/MOEA.pdf>, [(Online; accessed 07 September 2019)].
- [28] varios. Optimalidad de Pareto. https://es.wikipedia.org/wiki/Eficiencia_de_Pareto, [(Online; accessed 07 September 2019)].
- [29] varios. Stigler. https://en.wikipedia.org/wiki/Stigler_diet/, [(Online; accessed 07 September 2019)].
- [30] varios. Water Footprint network. <http://www.waterfootprint.org/>, [(Online; accessed 07 September 2019)].
- [31] Varios. Ejemplos, Problema de la dieta. <https://www.uv.es/~sala/dieta.pdf>, (Online; accessed 07 September 2019), 2018.
- [32] WWF. WWF. <https://www.wwf.es/>, [(Online; accessed 07 September 2019)].