



UNIVERSIDAD
DE LA REPUBLICA
URUGUAY



Algoritmos evolutivos para planificación energética en hogares

Giovanni Colacurcio

Ingeniería en computacion
Facultad de Ingeniería
Universidad de la República

Montevideo – Uruguay
Diciembre de 2019



UNIVERSIDAD
DE LA REPUBLICA
URUGUAY



Algoritmos evolutivos para planificación energética en hogares

Giovanni Colacurcio

Informe de Proyecto de Grado presentado al Tribunal Evaluador como requisito de graduación de la carrera de Ingeniería en Computación

Director de tesis:

Dr. Sergio Nesmachnow

Codirector:

Dr. Santiago Iturriaga

Montevideo – Uruguay

Diciembre de 2019

Colacurcio, Giovanni

Algoritmos evolutivos para planificación energética en hogares / Giovanni Colacurcio. - Montevideo: Universidad de la República, Facultad de Ingeniería, 2019.

XVI, 90 p.: il.; 29,7cm.

Director de tesis:

Sergio Nesmachnow

Codirector:

Santiago Iturriaga

Tesis de Grado – Universidad de la República, Programa de Ingeniería en computacion, 2019.

Referencias bibliográficas: p. 72 – 73.

1. Planificación energética, 2. Algoritmos evolutivos, 3. Optimización, 4. Ciudades inteligentes, 5. lanificación multiobjetivo. I. Nesmachnow, Sergio *et al.* II. Universidad de la República, Programa de Posgrado en Grado. III. Título.

Montevideo – Uruguay

Diciembre de 2019

RESUMEN

La administración energética es de crucial importancia en las sociedades de hoy en día. Diversas estrategias han sido propuestas para garantizar un acceso mayor a los recursos energéticos a un menor precio para los ciudadanos, asegurando la preservación de recursos naturales y la protección del medio ambiente. En este proyecto se presenta el diseño y la implementación de un sistema de planificación de uso de dispositivos eléctricos en el hogar, teniendo como objetivo optimizar la energía eléctrica, lo cual provoca un gran impacto en los usuarios finales y en las compañías eléctricas. El sistema propone utilizar información histórica y datos de sensores para definir instancias al problema de planificación. Con la información obtenida se propuso construir algoritmos evolutivos para la resolución del problema de planificación energética, con los objetivos de maximizar la preferencia del usuario, minimizando la energía eléctrica consumida y el precio de la energía.

Palabras claves:

Planificación energética, Algoritmos evolutivos, Optimización, Ciudades inteligentes, planificación multiobjetivo.

ABSTRACT

Energy management is a crucial issue in nowadays societies. Many strategies have been proposed to guarantee an increased access to the energy resources at affordable costs for citizens, while ensuring the preservation of natural resources and the protection of the environment. This project presents the advances in the design and implementation of a recommendation system for planning the use of household appliances, focused on improving energy efficiency from the point of view of both energy companies and end-users. The system proposes using historical information and data from sensors to define instances of the planning problem considering user preferences, which in turn are proposed to be solved using a multiobjective evolutionary approach, in order to maximize quality of service while minimizing the energy consumption and the price of the electric bill.

Keywords:

Energy Scheduling , Evolutionary Algorithms, Optimization, Smart Cities, Multiobjective Scheduling.

Lista de figuras

3.1	Operador de recombinacion 2PX.	12
3.2	Operador de mutacion de inversion de bit.	12
3.3	Frente de Pareto, soluciones dominadas y soluciones no dominadas para un problema de minimización de dos objetivos.	14
3.4	Objetivos de un AE para problemas de optimización multiobjetivo	15
5.1	Ejemplo de la representación utilizada.	26
5.2	Ejemplo de recombinación de una solución.	27
5.3	Ejemplo de mutación de una solución.	28
6.1	Resultados de la configuracion paramétrica del AE monoobjetivo	47
6.2	Resultados de la configuracion paramétrica del AE monoobjetivo	49
6.3	Test de rangos.	55
6.4	Frente de Pareto para la instancia chica en un día de semana	57
6.5	Frente de Pareto para la instancia chica fin de semana	58
6.6	Frente de Pareto para la instancia mediana en un día de semana	59
6.7	Frente de Pareto para la instancia mediana fin de semana	59
6.8	Frente de Pareto para la instancia grande en un día de semana	61
6.9	Frente de Pareto para la instancia grande fin de semana	61
6.10	Edificio día de semana	62
6.11	Edificio fin de semana	62
6.12	Corte instancia mediana, ejes Calidad Servicio, Precio.	63
6.13	Corte instancia mediana, ejes Calidad Servicio, Energía eléctrica contratada.	63
6.14	Corte instancia mediana, ejes Precio, Energía eléctrica contratada.	64

Lista de tablas

4.1	Resumen trabajos relacionados.	23
6.1	Detalle de resultados de la configuración de α y β del AE monoobjetivo	46
6.2	Calibración fitness algoritmo evolutivo monoobjetivo	46
6.3	Configuración parametrica AE	48
6.4	Detalle de resultados de la configuración paramétrica del AE monoobjetivo	49
6.5	Porcentaje de mejora AE.	50
6.6	Detalle de mejoras por objetivo del algoritmo evolutivo al algoritmo BaU	51
6.7	Configuración parametrica	53
6.8	Resultados obtenidos para el hipervolumen relativo en la configuración parametrica.	54
6.9	Resultados de las métrica DG para el algoritmo NSGA-II.	56
6.10	Resultados de las métrica Spread para el algoritmo NSGA-II.	56
6.11	Resultados de las métrica HR para el algoritmo NSGA-II.	56
6.12	Mejoras alcanzadas por la mejor solución de compromiso del algoritmo NSGA-II frente a los algoritmos propuestos.	64
6.13	Mejoras alcanzadas por el MOEA en comparación al algoritmo ávido QoS.	65
6.14	Mejoras alcanzadas por el MOEA en comparación al algoritmo ávido de Precio.	65
6.15	Mejoras alcanzadas por el MOEA en comparación al algoritmo ávido de energía eléctrica contratada.	66
6.16	Mejoras alcanzadas por el MOEA en comparación a la solución sin BaU.	66

6.17 Mejoras alcanzadas por el MOEA en comparación a la solución sin planificar.	67
6.18 Mejoras alcanzadas por el MOEA en comparación a la solución sin planificar.	68

Lista de Algoritmos

1	Funcionamiento de un algoritmo evolutivo	9
2	Esquema del algoritmo NSGA-II	16
3	Algoritmo ávido	34
4	Algoritmo ávido QoS	37
5	Algoritmo ávido energía eléctrica consumida	39
6	Algoritmo ávido energía precio	41
7	Estrategia BaU	42
8	Estrategia BaU	42

Tabla de contenidos

Lista de figuras	x
Lista de tablas	XI
Lista de algoritmos	XIII
1 Introducción	1
2 Descripción del problema	4
2.1 El problema de la planificación energética de hogares	4
2.1.1 Introducción y motivación	4
2.1.2 Formulación matemática	5
3 Algoritmos evolutivos	8
3.1 Algoritmos evolutivos	8
3.1.1 Introducción	8
3.1.2 Representación de soluciones	10
3.1.3 Función de fitness	10
3.1.4 Operadores evolutivos	11
3.2 Algoritmos evolutivos multiobjetivo	13
3.2.1 Problemas de optimización multiobjetivo	13
3.2.2 Algoritmos evolutivos multiobjetivo	14
3.2.3 El algoritmo NSGA-II	16
4 Trabajos relacionados	17
4.1 Resolución del problema de planificación energética en hogares utilizando algoritmos evolutivos	17
4.2 Resolución del problema de planificación energética en hogares utilizando otras técnicas	19

4.3	Estudio sobre desagregación de energía de consumos residenciales	21
4.4	Resumen	22
5	Implementación	24
5.1	Entorno de desarrollo	24
5.1.1	Introducción	24
5.1.2	Biblioteca de Desarrollo: ECJ	25
5.2	Implementación de los algoritmos para resolver el problema de planificación energética	25
5.2.1	Representación	25
5.2.2	Población inicial	26
5.2.3	Operadores evolutivos	26
5.2.4	Asignación de fitness	28
6	Evaluación experimental	29
6.1	Instancias del problema	29
6.1.1	Análisis de datos REDD	29
6.1.2	Generación de instancias del problema	30
6.2	Metodología de la evaluación experimental	31
6.2.1	Entorno de ejecución	31
6.2.2	Análisis estadístico	32
6.2.3	Configuración paramétrica	32
6.2.4	Algoritmos ávidos	33
6.2.5	Estrategia <i>Business as Usual</i>	42
6.2.6	Métricas del MOEA	43
6.3	Variante monoobjetivo	45
6.3.1	Asignación de fitness	45
6.3.2	Configuración paramétrica	47
6.3.3	Resultados Numéricos	50
6.4	Variante multiobjetivo	51
6.4.1	Configuración paramétrica	51
6.4.2	Resultados numéricos para el problema multiobjetivo	55
7	Conclusiones y trabajo futuro	69
7.1	Conclusiones	69
7.2	Trabajo futuro	70

Referencias bibliográficas	72
Anexos	74
Anexo 1 Publicaciones.	75

Capítulo 1

Introducción

La administración energética es de crucial importancia en las sociedades de hoy en día. Diversas estrategias han sido propuestas para garantizar un acceso mayor a los recursos energéticos a un menor precio para los ciudadanos, asegurando la preservación de recursos naturales y la protección del medio ambiente [15].

Para la implementación de políticas de administración energética efectiva, se debe integrar tecnología innovadora con sistemas que sean de fácil uso y eficientes, los cuales deben incluir características específicas que sean útiles tanto para compañías como para los ciudadanos.

De modo de que estos sistemas puedan ser utilizados por las compañías, deben ser capaces de realizar simulaciones realistas, controlar y planificar el mercado de electricidad. En cuanto al punto de vista de los ciudadanos, estos sistemas deben proveer aplicaciones de fácil uso que sean capaces de monitorear y administrar el consumo energético al nivel el hogar.

Las capacidades de monitorear, controlar y administrar el consumo energético son muy importantes para proveer un buen servicio (*Quality Of Service*, QoS) y experiencia de usuario, especialmente cuando consideramos la contratación del ciudadano, la protección del medio ambiente y los aspectos económicos en lo que a las ciudades inteligentes refiere [1].

Los edificios residenciales contribuyen significativamente a la energía total utilizada en el mundo. De acuerdo a las estadísticas de la Administración de Información Energética de Estados Unidos (USA, *Energy Information Administration*), el hogar promedio en Estados Unidos y Canadá utiliza alrededor de 12.000 kWh de electricidad. En Europa, esta cifra es menos de 10.000 kWh,

pero también es significativa.

Algunos electrodomésticos realizan la mayor contribución en lo que a consumo refiere, incluyendo calefactores y aires acondicionados (40-45%), electrodomésticos de la cocina (30%), calentadores de agua (15%-20%), y refrigerantes (5%-10%) [16]. Los patrones de utilización de energía no son tan diferentes en lo que refiere a países en desarrollo, donde el impacto del consumo energético es también muy importante.

La literatura que respalda el tema, indica que los sistemas basados en planificación del uso de electrodomésticos diferidos permiten mejorar la eficiencia energética al nivel del hogar [12, 11]. Los electrodomésticos diferidos son aquellos en los cuales su demanda de energía puede ser pospuesta o interrumpida (por ejemplo lavarropas, lava vajillas, etc.), causando un impacto despreciable en el QoS provisto a los usuarios. Éste es un acercamiento importante que toma en cuenta los diferentes precios de la electricidad y la disponibilidad de (no almacenable) energía provenientes de recursos renovables.

En esta línea de trabajo, este proyecto presenta la aplicación de algoritmos evolutivos (AEs) para resolver el problema de planificación energética de los electrodomésticos del hogar, considerando las preferencias del usuario. Un enfoque multiobjetivo apuntando a maximizar la satisfacción del usuario (evaluada en términos de QoS ofrecido de acuerdo a las preferencias especificadas) y simultáneamente minimizar el total de energía consumida, el cual está directamente relacionado al costo total de la factura eléctrica del usuario.

Los resultados principales indican que el enfoque propuesto es capaz de encontrar una planificación energética apropiada, el cual resulta superior a los casos en los que ningún tipo de planificación energética inteligente es utilizada (por ejemplo el modelo *Business as Usual*; BaU), así como también con un algoritmo ávido previamente propuesto en el marco de referencia del proyecto en cuestión.

Las mayores contribuciones de esta investigación abarcan: *i)* Formular una solución multiobjetivo al problema de planificación de energética del hogar teniendo en cuenta la satisfacción del usuario, el consumo energético y el precio de la energía eléctrica. *ii)* Idear AEs específicos que ataque el problema utilizando un enfoque de agregación lineal para los objetivos y un enfoque multiobjetivo considerando considerando cada objetivo de forma individual; *iii)* Evaluación del enfoque evolutivo ante escenarios realistas utilizando datos reales de consumición energética de los hogares provenientes de fuentes

confiables.

Lo que resta del documento se estructura como se describe a continuación. En el capítulo 2 se define el problema de planificación energética en hogares en su formulación multiobjetivo y monoobjetivo. El capítulo 3 introduce los algoritmos evolutivos como técnica para resolver el problema. El capítulo 4 presenta una breve reseña de los principales trabajos relacionados al problema de la planificación energética en hogares. Los algoritmos evolutivos implementados para resolver el problema de planificación energética en hogares se presentan en el capítulo 5. El capítulo 6 describe la evaluación experimental realizada así como los resultados alcanzados por los algoritmos propuestos. Por último las conclusiones del proyecto y las principales líneas del trabajo futuro se presentan en el capítulo 7.

Capítulo 2

Descripción del problema

Este capítulo presenta el problema abordado en el marco del proyecto, que consiste en planificar el uso de los dispositivos eléctricos teniendo como objetivos minimizar la factura eléctrica respetando la calidad de servicio del usuario final y la energía eléctrica contratada. La sección 2.1 presenta el problema en su versión multiobjetivo y monoobjetivo.

2.1. El problema de la planificación energética de hogares

2.1.1. Introducción y motivación

Minimizar costos es un claro objetivo desde el punto de vista del usuario al utilizar cualquier sistema de planificación. Sin embargo, las restricciones de uso de los dispositivos eléctricos en una planificación pueden condicionar la elección del usuario. Por tal motivo resulta de interés estudiar el problema de planificación de uso de los dispositivos eléctricos sujeto a las preferencias particulares de cada usuario al momento de realizar una carga eléctrica, la energía eléctrica contratada por los mismos y el precio de la energía.

Para cada instante de tiempo en el periodo a planificar de un día, el usuario final asignara distintos niveles de preferencia. Estos valores de preferencia representan la prioridad con la que se desea realizar una carga. Para cada uno de los intervalos de tiempo se conoce además el costo de la energía, se asume también que cada usuario tiene una cantidad contratada de Kw, por tanto la suma de la energía consumida en todos los instantes de tiempo no deberá superar

dicho valor, de lo contrario la solución será penalizada.

El objetivo del problema es desarrollar un algoritmo evolutivo multiobjetivo que tenga en cuenta los puntos mencionados anteriormente y de como resultado la asignación de carga de los distintos dispositivos electricos para los diferentes intervalos de tiempo del día. Se estudiarán también variantes monoobjetivo considerando objetivos económicos, de eficiencia energética y de calidad de servicio ofrecidos al usuario.

Se abordarán casos realistas del problema para analizar las características de consumo energético a diversos niveles urbanos (hogar, edificio, barrio, etc.)

2.1.2. Formulación matemática

La versión multiobjetivo y monoobjetivo del problema de planificación de uso de electrodomésticos abordada en este trabajo considera los siguientes elementos:

- Un conjunto de usuarios $U = \{u_1, \dots, u_N\}$, cada usuario representa a una unidad habitacional (casa) en una ciudad;
- Un conjunto de intervalos de tiempo (minutos) $T = \{t_1, \dots, t_M\}$ en el período de tiempo total a planificar;
- Un conjunto de electrodomésticos (genéricamente denominados *dispositivos*) $L = \{l_1, \dots, l_K\}$;
- Una función $E : U \rightarrow N$ donde $E(u_i)$ indica la potencia contratada por el usuario u_i ;
- Una penalización ρ por superar el máximo de potencia contratada;
- Una función $D : L \times U \rightarrow N$ donde $D(l_k, u_i)$ indica el tiempo promedio de utilización del dispositivo l_k para el usuario u_i ;
- Una función $C : T \rightarrow N$ donde $C(t_j)$ indica el costo de utilización (por kW) de la energía en el tiempo t_j ;
- Una función $P : L \rightarrow N$ donde $P(l_k)$ indica la energía (en kWh) consumida por el dispositivo l_k ;
- Una función $UP : U \times L \times T \rightarrow N$ donde $UP(u_i, l_k, t_j)$ indica la preferencia del usuario u_i en utilizar el dispositivo l_j en el tiempo t_k ;

Considérese la variable binaria x_{kj}^i , que indica si el dispositivo l_k del usuario u_i está encendido en el instante t_j ; y la función $y(x_{kj}^i)$ que indica el período

de tiempo que el aparato l_k del usuario u_i está encendido continuamente (sin apagarse) a partir del tiempo t_j (Ecuación 2.1)

$$y(x_{kj}^i) = m - j \text{ con } m = \max r / \forall h \in (j, r) \ x_{kh}^i = 1 \quad (2.1)$$

El problema multiobjetivo plantea hallar una planificación $X = \{x_{kj}^i\}$ de uso de electrodomésticos que simultáneamente maximice la satisfacción de los usuarios (dadas su funciones de preferencia) (Ecuación 2.2), minimice el costo de la factura eléctrica (Ecuación 2.3) y minimice la energía total consumida (Ecuación 2.4).

$$f(X) = \sum_{i=1}^N \sum_{j=1}^T \sum_{k=1}^K UP(u_i, l_k, t_j) \times \delta_{kj}^i \quad (2.2)$$

$$\text{con } \delta_{kj}^i = \begin{cases} 1 & \text{si } y(x_{kj}^i) \geq D(l_k, u_i) \\ 0 & \text{en otro caso} \end{cases}$$

$$h(X) = \sum_{i=1}^N \sum_{j=1}^T \sum_{k=1}^K x_{kj}^i \times P(l_k) \times C(t_k) \quad (2.3)$$

$$g(X) = \sum_{i=1}^N \sum_{j=1}^T \sum_{k=1}^K x_{kj}^i \times P(l_k) + \rho \times \psi_{kj}^i \quad (2.4)$$

$$\text{con } \psi_{kj}^i = \begin{cases} P(l_k) - E(u_i) & \text{si } P(l_k) - E(u_i) \geq 0 \\ 0 & \text{en otro caso} \end{cases}$$

El problema monoobjetivo plantea hallar una planificación $X = \{x_{kj}^i\}$ de uso de electrodomésticos que simultáneamente maximice la satisfacción de los usuarios (dadas su funciones de preferencia) y minimice el costo de la factura eléctrica (Ecuación 2.5). Para este propósito se estudiaron distintas combinaciones de α y β como parámetros para explorar diferentes valores de compromiso entre las funciones objetivo.

$$s(X) = \sum_{i=1}^N \sum_{j=M}^T \sum_{k=1}^K UP(u_i, l_k, t_j) \times \alpha - C(t_k) \times \beta \quad (2.5)$$

Ambos problemas aplican un modelo de penalización para los casos en que se supera la potencia máxima contratada. El modelo de penalización se divide en dos escenarios. El primer escenario el usuario supera la energía máxima contratada en un valor menor al 30 % de la misma. Este valor constituye al máximo valor de energía que puede haber sin que ocurra un cortocircuito. En dicho caso, la solución es penalizada por un 30 % de ρ . En el segundo escenario el usuario sobrepasa la energía en un valor mayor o igual al 30 % de la misma. Por lo cual las soluciones se ven penalizadas enteramente por la penalidad ρ .

La función UP se construyó utilizando las mediciones de carga de los dispositivos eléctricos tomadas por Kolter y Johnson [8]. Para cada minuto del día, en el periodo de un mes, se estudio cuantas veces un dispositivo se encontraba encendido. El valor calculado constituye a la preferencia que tiene el usuario de encender el dispositivo en dicho minuto.

La función D se construyó utilizando un día promedio de consumo de cada dispositivo del usuario. Para dicho día se estudia el valor de duración del dispositivo como la cantidad de minutos continuos en que el mismo permaneció encendido.

Capítulo 3

Algoritmos evolutivos

El presente capítulo describe la técnica utilizada para resolver el problema de planificación energética en hogares, tanto en su formulación monoobjetivo como en su variante multiobjetivo. En la sección 3.1 se introducen los conceptos generales de los algoritmos evolutivos como técnicas para resolver problemas de búsqueda y optimización. En la sección 3.2 se describen los algoritmos evolutivos para problemas multiobjetivo, en particular el utilizado para la resolución del problema de planificación energética en hogares

3.1. Algoritmos evolutivos

En esta sección se presentan los algoritmos evolutivos como técnica para resolver problemas complejos de búsqueda y optimización, y se introducen los conceptos fundamentales de esta técnica, los cuales serán utilizados a lo largo del documento.

3.1.1. Introducción

Los algoritmos evolutivos (AE) son algoritmos de búsqueda basados en los mecanismos de la selección natural y la genética [5]. Los primeros trabajos sobre los algoritmos evolutivos comenzaron a partir del año 1960 a cargo de John Holland y sus estudiantes en la Universidad de Michigan. La primera propuesta de algoritmo evolutivo fue presentada por Holland en 1975 en su libro *Adaptation in Natural and Artificial Systems*, introduciendo los conceptos de selección, cruzamiento y mutación, sentando las bases para la investigación en el área de la computación evolutiva [7]. Los AE se han popularizado en

los últimos 30 años y han sido exitosamente aplicados a resolver problemas de optimización subyacentes a problemas complejos del mundo real en múltiples áreas de aplicación.

Un AE es una técnica iterativa (cada iteración se denomina generación) que aplica operadores estocásticos sobre un conjunto de individuos (denominada población). Inicialmente, la población se genera a mediante un proceso aleatorio o utilizando alguna heurística diseñada para el problema en particular. Cada individuo en la población codifica una solución tentativa al problema y tiene asignado un valor de fitness, dado por una función de evaluación que determina su adecuación para resolver el problema. La búsqueda de aquellos individuos mas aptos, es decir, con mayor valor de fitness, se logra mediante la aplicación iterativa de operadores evolutivos como la selección de individuos según su fitness, la recombinación de partes de individuos y la mutación probabilística de su representación, conduciendo al AE hacia soluciones de mayor calidad. El funcionamiento básico de un AE se presenta en el Algoritmo 1.

Algoritmo 1 Funcionamiento de un algoritmo evolutivo

```
Inicializar población inicial ( $P(0)$ )
generación = 0
while no se cumpla criterio de parada do
    evaluar $P(\textit{generación})$ 
    padres = Selección( $P(\textit{generación})$ )
    hijos = OperadoresEvolutivos(padres)
    nuevaPoblación = reemplazo(hijos,  $P(\textit{generación})$ )
     $P(\textit{generación})$  = nuevaPoblación
    generación ++
end while
return mejor individuo encontrado
```

El criterio de parada puede involucrar un numero determinando en la cantidad de generaciones, una cota sobre el fitness alcanzado o la detección de una situación de convergencia en el AE. Se utilizan políticas específicas para determinar que individuos son seleccionados para realizar la recombinación y para determinar los individuos que serán insertados en la nueva población, en cada nueva generación. Finalmente, el AE retorna la mejor solución hallada durante el proceso iterativo, teniendo en cuenta la función de fitness definida en el problema.

3.1.2. Representación de soluciones

Al momento de trabajar sobre un problema, un AE trabaja sobre una abstracción de las soluciones candidatas al problema que se denomina cromosomas. Un cromosoma es un conjunto de genes que contienen información característica de una solución. Un gen puede tomar diferentes valores posibles, a cada valor se le denomina alelo. Un genotipo denota al conjunto de cromosomas que definen las características de un individuo. Dado que se suele utilizar un único cromosoma por individuo, los términos genotipo, cromosoma e individuo se utilizan de indistintamente. Por otra parte, el fenotipo representa un punto en el espacio de soluciones del problema.

Es necesario tener un mecanismo que permita transformar el genotipo en fenotipo y viceversa, es decir, transformar soluciones del problema en individuos del AE. Este mecanismo es la representación de los individuos del problema a resolver y es un punto fundamental en la construcción y el diseño de un AE, debido a que la representación elegida debe ser capaz de modelar al problema de manera adecuada. Entre las representaciones más usuales en la literatura se encuentran la representación binaria y la representación entera. La representación binaria consiste en un vector de bits, donde cada alelo puede tomar el valor de 0 o 1 en codificación binaria. La representación entera consiste en un vector de enteros, donde cada alelo es un número entero.

3.1.3. Función de fitness

Cada individuo en un AE tiene asociado un valor de fitness que representa que tan apto es para resolver el problema. El valor de fitness viene dado por la función de fitness, la cual asigna un valor al individuo mediante la evaluación del fenotipo. Por lo tanto dicha función de fitness es la principal herramienta para guiar al AE a soluciones de calidad, seleccionando aquellos individuos más aptos de forma de transmitir su información genética a sus descendientes. Debido al proceso de codificación, es posible que un determinado genotipo tenga un fenotipo asociado que no represente una solución factible en el dominio del problema. En estos casos es necesario realizar una de las siguientes acciones: aplicar un mecanismo de corrección que los transforme en una solución válida, asignarles un valor de fitness adecuado de forma de reducir la posibilidad de que sobrevivan al proceso evolutivo, o simplemente descartarlos.

3.1.4. Operadores evolutivos

En su variante mas elemental, un AE cuenta con operadores de selección, recombinación y mutación los cuales son utilizados por el AE en forma probabilística.

Selección

El operador de selección es el encargado de seleccionar a los individuos más aptos para continuar en el proceso evolutivo. Generalmente los individuos con mayor valor de fitness tienen más posibilidad de ser seleccionados, pero puede ocurrir que individuos con menos valor de fitness sean seleccionados, la selección del individuo depende de la técnica de selección de utilizada. Múltiples técnicas han sido propuestas en la literatura. A continuación, se describen las mas utilizadas.

- **Selección elitista:** selecciona únicamente un porcentaje de los individuos más aptos en cada generación.
- **Selección proporcional:** a cada individuo de la población se le asigna una probabilidad de ser seleccionado. La probabilidad de selección está definida por el cociente entre su valor de fitness y la suma del fitness de todos los individuos de la población. Los individuos más aptos tienen mayor probabilidad de ser seleccionados.
- **Selección por torneo:** sorteas un subconjunto m de individuos de la población, que compiten entre ellos en base a su fitness para ser seleccionados. Luego, se eligen k individuos con mayor valor de fitness (comúnmente $k = 1$).

Recombinación

El operador de recombinación se aplica sobre dos cromosomas, de forma de generar descendientes con las características de ambos cromosomas padres. Este operador es análogo al proceso de reproducción en la evolución natural. Diferentes operadores de recombinación se utilizan dependiendo de la codificación de las soluciones y del problema a resolver. A modo de ejemplo, en la figura 3.1 se muestra la recombinación de dos puntos o 2PX (Two Point Crossover) para una representación de enteros. El operador se encarga de elegir

dos puntos al azar del cromosoma y crear dos descendientes tomando los genes entre los puntos de corte de un padre y el resto de los genes del otro padre.

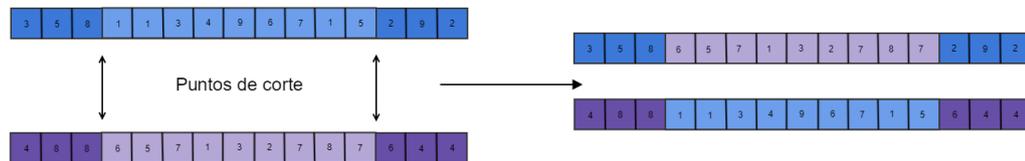


Figura 3.1: Operador de recombinacion 2PX.

Mutación

El operador de mutación es el encargado de modificar de forma aleatoria uno o varios genes del cromosoma, introduciendo diversidad en el proceso evolutivo, análogo al proceso de mutación en la evolución natural. De esta manera es posible obtener soluciones que no podrían ser obtenidas utilizando únicamente el proceso de recombinación. A modo de ejemplo, en la Figura 3.2 se muestra la mutación de inversión de bit (Flip Bit Mutation) para una representación binaria. Este operador cambia el valor a mutar por el valor opuesto binario.

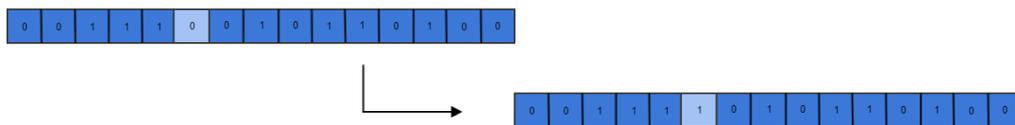


Figura 3.2: Operador de mutacion de inversion de bit.

3.2. Algoritmos evolutivos multiobjetivo

En esta sección se introducen los algoritmos evolutivos multiobjetivo aplicados a problemas de optimización. Además, se presenta el algoritmo multiobjetivo utilizado para la resolución del problema de planificación energética en hogares que es aplicado en este proyecto de grado (NSGA-II).

3.2.1. Problemas de optimización multiobjetivo

Los problemas de optimización multiobjetivo (*Multiobjective optimization problem, MOP*) son aquellos que involucran un conjunto de funciones objetivo a maximizar o minimizar, habitualmente en conflicto entre si, con restricciones a satisfacer por las soluciones obtenidas [4]. La ecuación 3.1 presenta la formulación matemática de un MOP con N_f objetivos para un vector $x = [x_1, x_2, \dots, x_d]$, donde d es la dimensión del espacio de variables de decisión,

$$\left. \begin{array}{l} \text{Minimizar/Maximizar} \quad \vec{f}(\vec{x}) = [f_i(\vec{x}), i = 1, \dots, N_f] \\ \text{Sujeto a} \quad g_j(\vec{x}) \leq 0 \quad j = 1, 2, \dots, J \\ \quad \quad \quad h_k(\vec{x}) = 0 \quad k = 1, 2, \dots, K \\ \quad \quad \quad x_m^{(L)} \leq x_m \leq x_m^{(U)} \quad m = 1, 2, \dots, d \end{array} \right\} \quad (3.1)$$

Muchos algoritmos de optimización multiobjetivo utilizan el concepto de dominancia para calcular soluciones. Se dice que una solución x^1 domina a otra solución x^2 si se cumplen las siguientes condiciones:

- x^1 no es peor que x^2 en todos los objetivos
- x^1 es estrictamente mejor que x^2 en al menos un objetivo

Es natural pensar que si una solución x^1 domina a otra solución x^2 , entonces la solución x^1 es mejor que la solución x^2 para el problema multiobjetivo que se esta resolviendo. Por este motivo muchos algoritmos evolutivos utilizan el concepto de dominancia como método para analizar y comparar soluciones de forma de encontrar soluciones no dominadas de buena calidad. Para un conjunto finito de soluciones es posible realizar todas las comparaciones entre ellas, de forma de establecer cuál solución domina a cuál y cuáles no son dominadas entre sí. Finalmente, es posible obtener un conjunto de soluciones

en el cual no existe un par de soluciones donde una domine a la otra. Este conjunto se denomina conjunto de soluciones no dominadas y cumple que, para toda solución que no pertenezca al conjunto, existe una solución “mejor” que la domina, perteneciente al conjunto. El conjunto de soluciones no dominadas del espacio de soluciones entero se denomina conjunto óptimo de Pareto. La región de puntos definida por el conjunto óptimo de Pareto en el espacio de valores de las funciones objetivo se conoce como frente de Pareto. La Figura 3.3 muestra un ejemplo de un problema multiobjetivo con dos funciones a minimizar, con sus respectivas soluciones dominadas, soluciones no dominadas y su frente de Pareto.

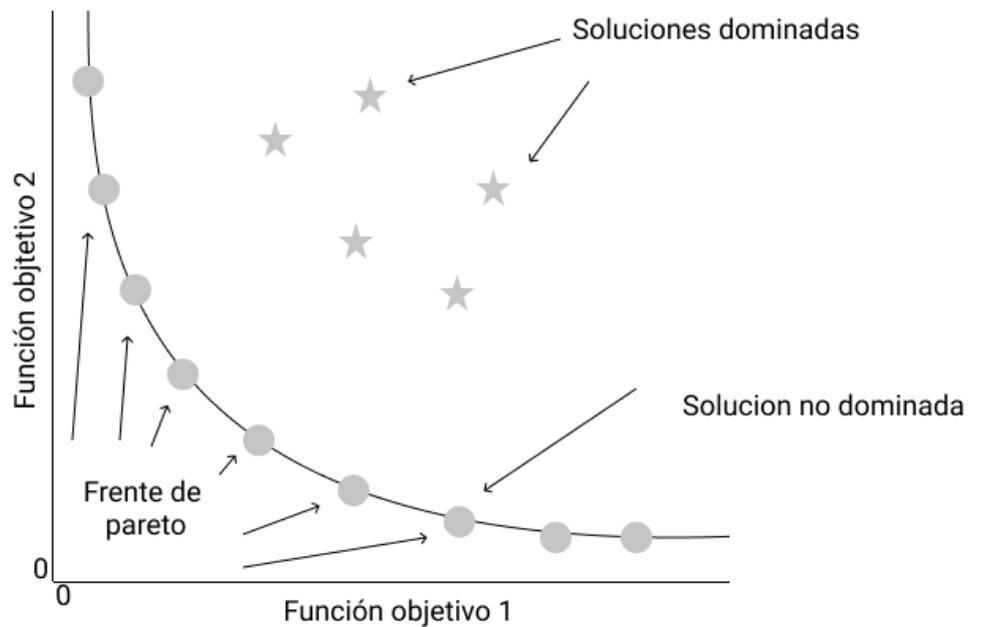


Figura 3.3: Frente de Pareto, soluciones dominadas y soluciones no dominadas para un problema de minimización de dos objetivos.

3.2.2. Algoritmos evolutivos multiobjetivo

Los algoritmos evolutivos para optimización multiobjetivo (*Multiobjective evolutionary algorithm, MOEA*) buscan cumplir con dos propósitos simultáneamente: obtener un conjunto de soluciones óptimas cercanas al frente de Pareto

to (convergencia) y encontrar adecuadamente el frente de soluciones que no convergen a una solución única o a una sección específica del frente de Pareto (diversidad) [2]. La figura 3.4 muestra de forma gráfica los dos propósitos mencionados, que deben ser tenidos en cuenta en la construcción de un MOEA. A partir del propio proceso evolutivo del MOEA se logra la convergencia, mientras que para mantener la diversidad es necesario agregar técnicas específicas (crowding, sharing).

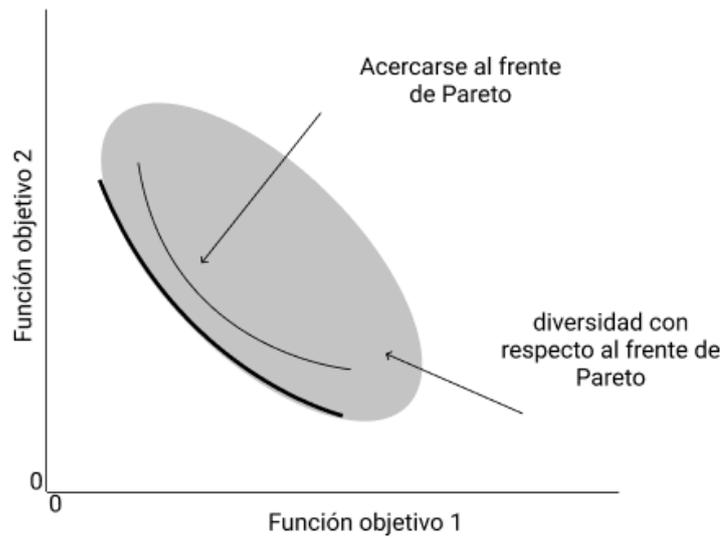


Figura 3.4: Objetivos de un AE para problemas de optimización multiobjetivo

3.2.3. El algoritmo NSGA-II

Uno de los algoritmos evolutivos multiobjetivo más utilizados es el *Non-dominated Sorting Genetic Algorithm* en su segunda versión (NSGA-II) [3], el cual provee una búsqueda evolutiva de acuerdo a las siguientes características:

- Utiliza un ordenamiento no dominado, elitista, que disminuye la complejidad asociada al chequeo de dominancia.
- Utiliza una técnica de crowding para preservar la diversidad de la población.
- Utiliza un método de asignación de fitness que considera además los valores de distancia de crowding para evaluar las soluciones.

El Algoritmo 2 muestra el esquema de funcionamiento del algoritmo NSGA-II.

Algoritmo 2 Esquema del algoritmo NSGA-II

```
Inicializar población inicial (P(0))
generación = 0
while no se cumpla criterio de parada do
    padres_hijos = Padres  $\cup$  Hijos
    frentes = Ordenamiento_No_Ordenado(padres_hijos)
    nueva_población = 0
    i = 1
    while nueva_población + frentes(i)  $\leq$  tamaño_población do
        padres_hijos = CalcularDistanciaCrowding(frentes(i))
        nueva_población = nueva_población  $\cup$  frentes(i)
        i ++
    end while
    Ordenar_frentes_por_distancia_crowding
    nueva_población = nueva_población  $\cup$  selección_elitista(frentes(i))
    generación ++
    P(generación) = nueva_población
end while
return mejor solución
```

El esquema de funcionamiento de NSGA-II ha sido aplicado para la resolución de problemas de optimización en diversas áreas, obteniendo muy buenos resultados en la práctica.

Capítulo 4

Trabajos relacionados

Este capítulo presenta un resumen de los principales trabajos relacionados a los problemas abordados en el proyecto. El problema de la planificación energética en hogares es de gran interés tanto para las empresas eléctricas como para los usuarios finales. Los beneficios que presenta permite a las empresas eléctricas planificar la producción de energía y a los usuarios finales la reducción del costo de la factura eléctrica. Estos beneficios han motivado la investigación de diferentes técnicas y heurísticas para su resolución. La sección 2.1 reseña trabajos que han abordado el problema utilizando algoritmos evolutivos. La sección 2.2 reseña la aplicación de otras técnicas para resolver el problema de planificación energética. La sección 2.3 muestra cómo fueron obtenidos los datos que fueron utilizados en el proyecto, para obtener información de los usuarios finales e información de los dispositivos eléctricos. Finalmente la sección 2.4 muestra un breve resumen de los artículos estudiados.

4.1. Resolución del problema de planificación energética en hogares utilizando algoritmos evolutivos

Soares et al. [13] propusieron un algoritmo evolutivo monoobjetivo para la resolución del problema de planificación energética en hogares. Los autores propusieron planificar las cargas de los dispositivos electrónicos del hogar de forma de minimizar la factura eléctrica, teniendo en cuenta las preferencias del usuario así como los precios de la electricidad y la energía eléctrica con-

tratada. El algoritmo busca lograr la planificación de dispositivos mediante la anticipación o el retraso de las cargas, respetando las preferencias del usuario. La complejidad del problema que se resuelve radica en la combinación de la asignación de las cargas desplazables en el tiempo, la variación en el precio del kWh, la existencia de diferentes niveles de energía contratada, la continuidad de los servicios de energía y la satisfacción personal del usuario. Se utilizó un algoritmo evolutivo para realizar la búsqueda en el espacio de soluciones. Las soluciones son codificadas como un arreglo de enteros, donde cada elemento representa el minuto en el cual el dispositivo comienza su ciclo. El tamaño del arreglo esta determinado por la cantidad de dispositivos a planificar. El operador de selección utilizado fue *ternary tournament*, para el cruzamiento se implementó una máscara aleatoria para crear dos descendientes utilizando las características de los padres. El operador de mutación cambia el instante en el que el dispositivo comienza a realizar su ciclo de carga dentro de un rango de 5 minutos. La función objetivo busca minimizar el costo total. Los factores que se consideraron para determinar el costo fueron: el costo de la energía consumida por los dispositivos administrados, la penalidad asociada a realizar las cargas de acuerdo a las preferencias del usuario, la penalidad por la cercanía entre el consumo actual y la energía contratada del usuario (de forma de asegurar que el consumo eléctrico no supere la energía contratada por el usuario), y una penalidad asociada a la energía disponible en cada intervalo de tiempo (la penalidad aumenta si la energía disponible disminuye). Todas las penalidades fueron monetizadas para incluirlas en la formulación que propone minimizar el costo total. La evaluación experimental considera una única casa con cinco diferentes dispositivos: dos lavarropas, un lavavajillas, un secarropas, un calefón y un vehículo eléctrico. Para estas cinco cargas, el control aplicado consiste en ubicar de forma adecuada las cargas en los intervalos establecidos de acuerdo a las condiciones establecidas. Los resultados mostraron un ahorro del 40 % frente a un caso tomado como referencia donde todas las cargas operan en los intervalos de preferencia del usuario respetando únicamente que la energía contratada no se vea superada.

Soares et al. [11] estudiaron un modelo multiobjetivo para optimizar la asignación de carga de los dispositivos eléctricos dentro de un intervalo de tiempo de 36 horas. El objetivo del enfoque propuesto es ayudar al usuario a tomar decisiones sobre la asignación de carga de los dispositivos eléctricos, dentro del intervalo de tiempo considerado. Las decisiones de asignación de

carga apuntan a minimizar la factura eléctrica y a minimizar la insatisfacción del usuario. La insatisfacción se divide en dos componentes. Un componente esta relacionado a los períodos de tiempo del día en los que el usuario prefiere no realizar la carga de un dispositivo eléctrico. El otro componente está asociado al riesgo de que la energía se vea interrumpida debido a que la energía eléctrica contratada por el usuario fue superada. Ambos componentes se incorporaron al modelo matemático de manera de penalizar soluciones que no cumplan con esas restricciones. El artículo plantea incorporar las características del problema a un algoritmo evolutivo multiobjetivo. Se propone utilizar *Non dominated sorting genetic algorithm II* (NSGA-II) modificado, para poder adaptar cierto tipo de características físicas relacionadas al problema de la planificación de carga, y así mejorar la usabilidad de los resultados. El modelo matemático explícitamente considera, minimizar el precio de la energía, minimizar la insatisfacción del usuario y que la energía consumida por los dispositivos no supere la energía eléctrica contratada. Las soluciones son codificadas como un arreglo de enteros, donde cada elemento representa el minuto en el cual cada dispositivo comienza a realizar su ciclo de carga. El largo del arreglo está determinado por la cantidad de dispositivos a planificar. Los operadores de selección, cruzamiento y mutación fueron los utilizados en [13]. La población inicial es inicializada de forma tal que ninguna supere la energía contratada por el usuario. Para la evaluación experimental se utilizaron seis diferentes dispositivos electrónicos: dos lavarropas, un secarropas, un lavavajillas, un calefón y un vehículo. Los resultados que se obtuvieron mostraron una reducción en la tarifa eléctrica de 23 % en promedio, frente a una asignación de cargas que no considera optimizaciones.

4.2. Resolución del problema de planificación energética en hogares utilizando otras técnicas

Tarish et al. [6] propusieron un enfoque de planificación en base a sistemas de respuestas a la demanda. Los sistemas de respuesta a la demanda existentes en la actualidad pueden ser categorizados de dos formas: en un esquema a base de incentivos o en un esquema a base de precios. En el enfoque a base de incentivos las compañías de servicios eléctricos manejan los diferentes disposi-

tivos de los usuarios durante tiempos de emergencia o durante tiempos donde se presentan picos de consumo eléctrico. En un esquema basado en precios los usuarios finales obtienen descuentos por reducir o desplazar la carga de los dispositivos durante períodos de alta demanda. El primero de los esquemas involucra un acceso directo a los dispositivos de los usuarios para realizar operaciones de encendido o apagado, resultando en un enfoque altamente invasivo. El segundo esquema no muestra este inconveniente, pero se le considera injusto por no ofrecer descuentos para los clientes cuyo consumo ya es bajo. Por estas razones, el artículo añade un sistema dinámico de planificación de carga residencial para poder optimizar la planificación de las cargas en una casa, basándose en un nivel de consumo y a un esquema de precios. El sistema propuesto alienta a los consumidores a administrar el consumo de energía dentro del límite de consumo permisible del esquema de precios propuesto, de forma de reducir la factura eléctrica. Los resultados de la simulación mostraron que utilizando este sistema se beneficia tanto a los clientes como a las empresas eléctricas. Los clientes se ven beneficiados ya que se reduce su factura eléctrica, mientras que las empresas logran disminuir el pico de consumo de carga. Para un caso de estudio particular estudiado por Setlhaolo et al.[10], el sistema propuesto por Tarish et al.[6] permite a los clientes reducir la factura eléctrica hasta en un 53% y disminuir el pico de carga hasta en un 35%.

Xiao Hui Li y Seung Ho Hong [9] estudiaron un algoritmo de respuesta a la demanda incluyendo el concepto de precio esperado por el usuario. Los autores propusieron un enfoque que explota la infraestructura de una red eléctrica inteligente de forma de responder ante precios dinámicos. Utilizando la información provista por la grilla sobre los precios de la energía y la información del usuario, el algoritmo puede determinar horas en las que el precio es más alto y horas en las que no. Con esta información, el algoritmo puede planificar automáticamente la operación de los electrodomésticos así como calcular cuánta energía deben almacenar los dispositivos en periodos donde el precio de la energía sea más baja, para que sea utilizada en periodos de precios más altos. La estrategia utilizada incluye un modelo de suavizado exponencial, de forma de poder predecir la energía requerida por los dispositivos electrónicos y utiliza el teorema de Bayes para computar la probabilidad de que un dispositivo realice una carga basándose en la información recolectada previamente. Con esa información, el algoritmo puede reducir significativamente el consumo en horas pico lo cual lleva a una reducción de costos en el uso de energía do-

mestica de un 39%. El caso de estudio analizado contiene información de seis dispositivos eléctricos provenientes de una casa. Estos son: una heladera, un televisor, luces, un aire acondicionado, un lavaropa y un secaropa y se utilizaron un esquema de precios con tres diferentes niveles: alto, medio y bajo.

4.3. Estudio sobre desagregación de energía de consumos residenciales

Los datos utilizados para este proyecto provienen de un estudio realizado por J. Zico Kolter y Matthew J. Johnson [8] sobre la desagregación de la energía en clientes residenciales. La desagregación permite obtener datos de consumo de diferentes aparatos domésticos a partir de un valor global de consumo instantáneo. La importancia en este tipo de estudio está en que la información sobre las cargas individuales aporta mayor información a los usuarios que la energía total utilizada. Con esta información se pueden inducir diferentes comportamientos en la utilización de los artefactos electrónicos por los usuarios. El trabajo discute diferentes enfoques de desagregación que fueron evaluados en publicaciones previas y como afectaron dichos enfoques a las decisiones de diseño para la obtención de datos. También describe el hardware utilizado y las configuraciones de software para la obtención de dichos datos. El sistema de hardware instalado en cada casa registró datos sobre el consumo de la casa y el voltaje de cada dispositivo individual. Los datos fueron almacenados en una base de datos central reúne información de todas las casas y puede ser accedida remotamente. Como información adicional, los autores proveen acceso al conjunto de datos recolectado en todas las casas para cada uno de los artefactos electrónicos en los que fue utilizado el hardware mencionado. Los datos incluyen información de consumo sobre seis diferentes casas ubicadas en Estados Unidos. De cada casa se tiene información de consumo de diferentes dispositivos electrónicos, durante un periodo promedio de cuatro semanas. Los datos pueden accederse en la dirección [<http://redd.csail.mit.edu/>]

4.4. Resumen

El relevamiento de trabajos relacionados muestra un creciente interés en la planificación energética de cargas domésticas. Las aproximaciones mediante sistemas de respuesta a la demanda logran reducciones en la factura eléctrica y reducciones en los picos de carga. Sin embargo, estos sistemas requieren que los usuarios se adapten al esquema de consumo de las empresas eléctricas, lo que puede reducir la calidad de servicio que se le ofrece. Los enfoques que utilizan algoritmos evolutivos toman en cuenta esta consideración incluyendo como restricción la calidad de servicio. En el enfoque monoobjetivo no se calculan soluciones de compromiso, por lo que diferentes usuarios con igual potencia contratada e iguales preferencias se deberán adaptar a una misma planificación de carga. En el enfoque multiobjetivo se ponderan las penalidades de preferencia de usuario y la penalidad de alcanzar la energía máxima contratada en una única función objetivo, por lo cual se reduce la información provista por los resultados del algoritmo. Adicionalmente, ninguno de los trabajos estudiados realizó estudios en diferentes niveles urbanos (edificios, barrios, etc). El análisis de los trabajos relacionados permite concluir que existen oportunidades para contribuir en soluciones enfocadas en los intereses de los usuarios finales teniendo en cuenta diferentes niveles urbanos, aplicando metaheurísticas eficaces como los AE y siguiendo un enfoque de optimización multiobjetivo explícito, como se propone en este proyecto.

La tabla 4.1 presenta un breve resumen de los trabajos relacionados. Para cada artículo estudiado se reporta los autores, el año de publicación del artículo y una breve descripción del trabajo.

Autores	Año	Comentario
Soares et al. [13].	2012	Presenta un algoritmo evolutivo monoobjetivo que busca minimizar el costo de la factura eléctrica. Los factores que determinaron el costo fueron : el costo de la energía consumida por las dispositivos administrados, una penalidad asociada a realizar las cargas de acuerdo a las preferencias del usuario, una penalidad por la cercanía entre el consumo actual y la energía contratada del usuario (de forma de poder asegurar la continuidad de la energía), y una penalidad asociada a la energía disponible en cada intervalo de tiempo (la penalidad aumenta si la energía disponible disminuye). Todas las penalidades fueron monetizadas de forma de minimizar el costo total. Los resultados mostraron un ahorro del 40 % frente al caso que tomaron como referencia.
Soares et al. [11].	2014	Presenta un algoritmo evolutivo multiobjetivo que busca minimizar la factura eléctrica y minimizar la insatisfacción de los usuarios. La insatisfacción la mide en base a dos aspectos, los bloques de horario de preferencia para la carga y los riesgos de interrumpir el suplemento de energía. Los resultados que se obtuvieron mostraron una reducción promedio en la tarifa eléctrica de 23 %.
Tarish et al. [6].	2015	Propone un sistema dinámico de planificación de carga residencial para optimizar la planificación de las cargas en una casa en base a un nivel de consumo y a un esquema de precios. Para un caso particular el sistema permite a los clientes reducir la factura eléctrica hasta en un 53 % y disminuir el pico de carga hasta en un 35 %.
Li y Hong [9].	2013	Propone un algoritmo para planificar automáticamente la operación de electrodomésticos y calcular cuánta energía deben almacenar los dispositivos en periodos donde el precio de la energía sea más baja para que sea utilizada en periodos de precios más altos. El algoritmo reduce el consumo en horas pico lo cual lleva a una reducción de costos en el uso de energía domestica de un 39 %.
Kolter y Johnson [8].	-	Realiza un estudio sobre la desagregación de la energía. Describe el hardware y el software utilizado para la recoleccion de datos sobre el consumo de energía de múltiples casas.

Tabla 4.1: Resumen trabajos relacionados.

Capítulo 5

Implementación

El presente capítulo describe los algoritmos desarrollados para la construcción de datos de prueba y para el problema de planificación energética en hogares. La sección 5.1 presenta las herramientas y bibliotecas utilizadas para la implementación de los distintos algoritmos. La Sección 5.2 detalla las implementaciones realizadas para la construcción de los datos de prueba. También presenta los detalles de implementación de los AEs en su versión monoobjetivo y multiobjetivo para la resolución del problema de planificación energética en hogares.

5.1. Entorno de desarrollo

La siguiente sección presenta el entorno de desarrollo y las bibliotecas utilizadas para la implementación del análisis de datos y para el algoritmo evolutivo en su versión monoobjetivo y multiobjetivo.

5.1.1. Introducción

Para la generación de instancias del problema se utilizó Python en su versión 2.7.5. Dentro de la generación de instancias se encuentra comprendida las preferencias del usuario, los tiempos de carga de los dispositivos eléctricos y la potencia de carga de los dispositivos eléctricos. Este análisis fue realizado tanto para días de semana como de fin semana. Para la implementación de los AEs se utilizó Java en su versión 1.8 y la biblioteca ECJ.

5.1.2. Biblioteca de Desarrollo: ECJ

ECJ es una biblioteca de computación evolutiva desarrollada en la Universidad de George Mason de Fairfax, Estados Unidos [18]. Fue desarrollada en Java y pensada específicamente para soportar problemas de gran complejidad. La particularidad de este software, es que hace un gran énfasis en la programación genética proveyendo también soporte para una variedad de algoritmos evolutivos e incluso algoritmos específicos para problemas multiobjetivo. Ofrece también la posibilidad de realizar computo distribuido soportando el modelo maestro/esclavo y de islas. Además, brinda facilidades para implementar estrategias de evolución estacionarias, co-evolutivas y de evolución natural. Cuenta con una documentación completa y exhaustiva, soporte activo y liberaciones de nuevas versiones periódicamente. La biblioteca ECJ fue utilizada para la implementación de los AEs descritos en la sección 3.

5.2. Implementación de los algoritmos para resolver el problema de planificación energética

En esta sección se describen las particularidades de los AEs implementados para resolver el problema de planificación energética en hogares.

5.2.1. Representación

ECJ ofrece implementaciones para las representaciones más usuales, sin embargo la complejidad del problema deriva en que sea necesario definir una representación adaptada al mismo. La representación contempla para cada usuario un vector $X = (x_0, x_1, \dots, x_T)$ donde T es la cantidad de minutos en el periodo de tiempo total a planificar. Cada elemento x_j es un vector de bits de la forma $x_j = (b_1, b_2, \dots, b_L)$ donde L es la cantidad de dispositivos a planificar y cada b_i indica si el dispositivo está encendido en el minuto j .

La figura 5.1 muestra un ejemplo de codificación de la solución para una instancia con cinco usuarios en donde el usuario dos busca planificar cinco dispositivos eléctricos. En el ejemplo se muestra que en el minuto i , para el usuario dos, los dispositivos tres y cuatro están apagados.

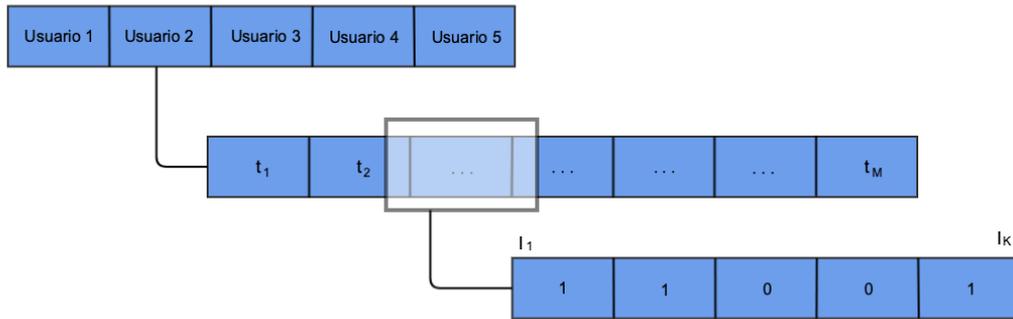


Figura 5.1: Ejemplo de la representación utilizada.

5.2.2. Población inicial

La población se inicializa de forma aleatoria de manera tal que cada dispositivo eléctrico se encuentre encendido con probabilidad $\alpha = 0.6$. Esto se debe a que el apagado de dispositivos influye en mayor parte al fitness que el encendido. Apagar un dispositivo eléctrico implica la disminución de la factura eléctrica y de la potencia contratada. Encender un dispositivo favorece únicamente la preferencia del usuario. Por lo tanto en las instancias iniciales los dispositivos eléctricos son encendidos con mayor probabilidad, de manera que el algoritmo explore las demás soluciones.

5.2.3. Operadores evolutivos

Los operadores evolutivos son un componente fundamental del proceso evolutivo ya que son los encargados de guiar al AE hacia soluciones de calidad. A continuación, se presentan los operadores evolutivos utilizados en el marco de este proyecto.

Selección

Se utilizó el operador de selección por torneo para otorgar una apropiada presión selectiva a las subpoblaciones, sorteando dos individuos de la población y eligiendo el mejor ($m = 2$ y $k = 1$ según lo explicado en la sección 3.1.4). Entre los beneficios que se encuentran de utilizar este tipo de selección esta la velocidad de aplicación (dado que no es necesario evaluar ni comparar la totalidad de la población) y la capacidad de prevenir, en cierto grado, la

convergencia prematura.

Recombinación

Para recombinar los individuos se utilizó el operador de recombinación de un punto (SPX). Dado dos elementos de la población (individuos padres), para cada usuario, se selecciona de forma aleatoria el intervalo de tiempo donde realizar la recombinación (punto de corte). A partir del punto de corte se forma una nueva planificación de uso de dispositivos eléctricos, tomando los datos previos al punto de corte del padre uno y los datos luego del punto de corte del padre dos. La figura 5.2 muestra un ejemplo de recombinación para tres usuarios con sus respectivos dispositivos eléctricos.

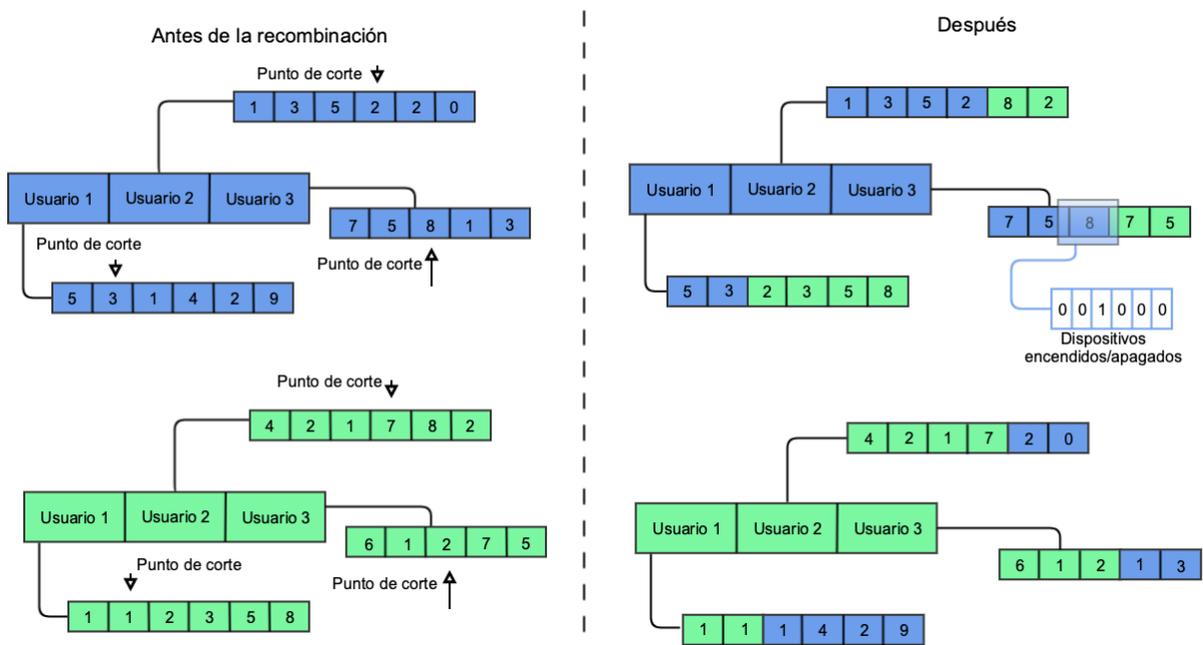


Figura 5.2: Ejemplo de recombinación de una solución.

Mutación

El operador de mutación cambia el estado actual de un dispositivo. Si el dispositivo se encuentra encendido el operador de mutación lo apaga, sino el operador de mutación lo enciende. Para cada usuario final, se selecciona en forma aleatoria el intervalo de tiempo donde realizar la mutación y el dispositivo eléctrico a el cual mutar. Experimentos iniciales mostraron que cambiar mayor cantidad de dispositivos reducía la diversidad en la población. La figura

5.3 muestra un ejemplo de mutación para tres usuarios con sus respectivos dispositivos eléctricos.

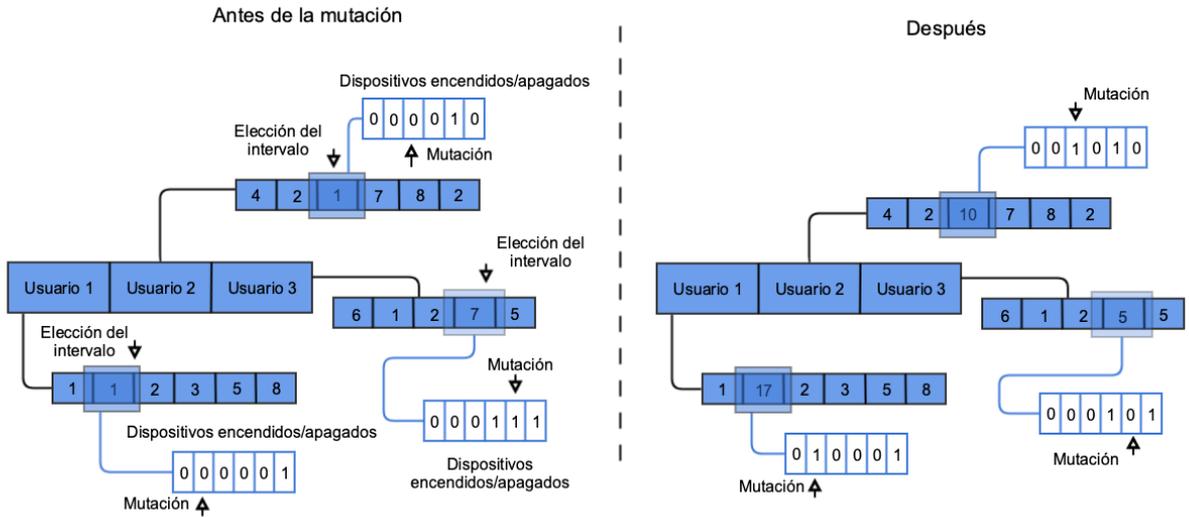


Figura 5.3: Ejemplo de mutación de una solución.

5.2.4. Asignación de fitness

El algoritmo evolutivo monoobjetivo buscó maximizar una función lineal que tuviese en cuenta la preferencia del usuario y el precio de la energía eléctrica en cada instante de tiempo. Se estudiaron distintas combinaciones de α y β que maximicen la función de agregación lineal descrita en la ecuación 2.5. El análisis realizado y la exposición de resultados se encuentran en la sección 6.3.1.

El algoritmo NSGA-II es multiobjetivo de forma explícita, por lo que se utilizan las funciones objetivo definidas en la Sección 2.1.2 al momento de asignar el fitness. El propio algoritmo es el encargado de clasificar a las soluciones según dominancia, de acuerdo a lo explicado en la sección 3.2.3.

Capítulo 6

Evaluación experimental

El presente capítulo detalla la evaluación experimental de los algoritmos implementados para resolver el problema de planificación energética en hogares. La sección 6.1 describe el proceso de generación de instancias utilizadas en la evaluación experimental, la sección 6.2 describe la metodología utilizada durante la evaluación experimental y las secciones 6.3 y 6.4 reportan los resultados obtenidos y la comparación con otras técnicas para los algoritmos monoobjetivo y multiobjetivo, respectivamente.

6.1. Instancias del problema

Con el fin de realizar la evaluación experimental de los AEs implementados, se estudió un conjunto de datos para generar instancias realistas del problema. La sección 6.1.1 describe los datos provistos por [8] y el análisis que se realizó en los mismos para generar instancias realistas del problema. La sección 6.1.2 detalla las instancias utilizadas en la evaluación experimental de los AE que resuelven el problema de planificación energética en hogares en su variante multiobjetivo y monoobjetivo.

6.1.1. Análisis de datos REDD

Los datos analizados por [8] sobre la desagregación de la energía contienen lecturas sobre la energía consumida por los dispositivos individuales de diferentes casas. Las medidas para los diferentes dispositivos fueron tomadas a una frecuencia promedio de 3 segundos durante un periodo de 28 días aproximadamente. Para las casas que formaron parte del análisis se estudio el tiempo

máximo de utilización continua de cada dispositivo. Con esta información se dedujeron los dispositivos más utilizados por el usuario. También para este estudio se tuvo en cuenta la cantidad de veces que se tomaron mediciones de un dispositivo. Un dispositivo que presenta un intervalo de utilización continuo y prolongado, pero de el que solo se tienen medidas en una sola casa, presenta menos información que uno en el cual se tienen mediciones de diferentes casas. Con la información obtenida y de forma de completar las cuatro categorías de clasificación de dispositivos que plantea Soares et al. [14], se obtuvieron los dispositivos que fueron utilizados en este proyecto: Lavavajillas, microondas, secarropa, aire acondicionado, horno y refrigerador. Debido a que la información de consumo de los dispositivos eléctricos fue tomada mediante intervalos pequeños de tiempo y en estos intervalos no se realizan grandes cambios de consumo, se realizó un promedio de los datos cada 15 minutos. Con los promedios de datos obtenidos se estudiaron los intervalos de tiempo en el día en el que cada usuario utiliza los diferentes dispositivos eléctricos. De esta forma se obtuvieron las preferencias de realización de una carga eléctrica para cada dispositivos para los diferentes usuarios. Por último, para cada día del mes se estudio el consumo de energía en términos de KWh utilizados. Este estudio se realizó para días de semana y para fines de semana. Luego, para cada dispositivo se determino un día de semana y un día en fin de semana en el que el consumo fuese representativo del consumo global, de forma de determinar la potencia consumida por el dispositivo durante un intervalo de tiempo de un día.

6.1.2. Generación de instancias del problema

Las variantes multiobjetivo y monoobjetivo del problema requieren como datos de entrada las preferencias de carga eléctrica de los usuarios para cada dispositivo eléctrico, el consumo de cada dispositivo, el precio de la energía y la energía eléctrica contratada. Las instancias se corresponden a los escenarios de casa, barrio y edificio. Debido a que la variante de barrio se corresponde a varias casas independientes entre si, las instancias chica, mediana y grande se puede corresponder tanto a casas, como a una instancias de barrio. En el caso de que la instancia se corresponda a un edificio los algoritmos reciben como entrada la energía eléctrica contratada por el edificio. Las preferencias de usuario y el consumo de cada dispositivo se obtienen del análisis de los datos de REDD.

El precio de la energía se obtuvo de las tarifas actuales de la administración nacional de usinas y transmisiones eléctricas (UTE) para un consumo medio con energía eléctrica contratada de 3,7 KWh disponibles en [17]. Para las instancia de edificio la energía eléctrica contratada se calcula como la energía eléctrica contratada por un usuario por la cantidad de usuarios en el edificio.

Se generaron ocho instancias de prueba:

- Dos instancias chicas: con dos usuarios con tres y dos dispositivos eléctricos respectivamente. Una instancia para día de semana y una instancia para fin de semana.
- Dos instancias medianas: con cuatro usuarios con cuatro, cuatro, tres, y dos dispositivos eléctricos respectivamente. Una instancia para día de semana y una instancia para fin de semana.
- Dos instancias grandes: con seis usuarios con cinco, cinco, cuatro, cuatro, tres y dos dispositivos eléctricos respectivamente. Una instancia para día de semana y una instancia para fin de semana.
- Dos instancias de edificios: con seis usuarios con cinco, cinco, cuatro, cuatro, tres y dos dispositivos eléctricos respectivamente. Una instancia para día de semana y una instancia para fin de semana.

Se generaron instancias para días de semana y fin de semana de forma de tener en consideración distintos patrones de consumo y de utilización de dispositivo eléctricos por parte del usuario.

6.2. Metodología de la evaluación experimental

En esta sección se presentan los lineamientos generales seguidos al momento de realizar la evaluación experimental de los algoritmos implementados. Se describe el ambiente de ejecución, las métricas utilizadas y los algoritmos ávidos utilizados para la comparación de resultados.

6.2.1. Entorno de ejecución

La evaluación experimental de los algoritmos propuestos en este proyecto fue realizada utilizando una computadora laptop TOSHIBA Sattelite S55-B. La misma utiliza como sistema operativo *Windows Home* 10, cuenta con

un procesador de cuatro núcleos con *multi threading* Intel(R) Core(TM) i7-4620HQ CPU @ 2,60GHz y 12 GB de memoria RAM disponible.

6.2.2. Análisis estadístico

Debido a la naturaleza estocástica de los AE, para diferentes ejecuciones sobre la misma instancia del problema se pueden obtener diferentes resultados. Con el fin de obtener resultados significativos, se realizaron 50 ejecuciones independientes para cada instancia.

6.2.3. Configuración paramétrica

Debido a la cualidad estocástica de los AEs, es necesario ajustar sus parámetros previamente al análisis experimental para determinar la configuración que permite calcular los mejores resultados. Para la configuración paramétrica se utilizaron tres instancias del problema (distintas de las utilizadas en la evaluación experimental para evitar sesgo). Luego de una evaluación inicial, el tamaño de población se fijó en 150 individuos.

El ajuste paramétrico se focalizó en estudiar las probabilidades de recombinación (p_C , valores candidato 0.1, 0.25, 0.5), de mutación (p_M , valores candidato 0.1, 0.05, 0.01) y cantidad de generaciones (G , valores candidato 2500, 5000, 10000). Se estudiaron todas las combinaciones de valores sobre las tres instancias de calibración, realizando 50 ejecuciones independientes en cada caso. Los resultados del análisis paramétrico se reportan en la sección 6.3.2 para el algoritmo evolutivo monoobjetivo y en la sección 6.4.1 para el algoritmo evolutivo multiobjetivo.

6.2.4. Algoritmos ávidos

Un algoritmo ávido (o *greedy*) es un método de construcción de soluciones que se basa en seguir una heurística enfocada en conseguir una solución óptima local en cada paso, con el objetivo de llegar a la solución general óptima del problema. Es relevante comparar el desempeño de los algoritmos implementados frente a un algoritmo ávido, en términos de soluciones alcanzadas por ambos métodos. Con este propósito se implementaron cuatro algoritmos ávidos, uno para el algoritmo evolutivo monoobjetivo y tres para el algoritmo evolutivo multiobjetivo. En el caso particular de que la instancia a evaluar se correspondan a un edificio los cuatro algoritmos agregan como restricción que para ningún intervalo de tiempo la energía consumida por todos los usuarios se vea superada por la energía contratada por el edificio.

Algoritmo ávido para AE monoobjetivo

El algoritmo ávido implementado para el algoritmo monoobjetivo buscó encontrar una planificación de dispositivos eléctricos que tuviese en cuenta la preferencia del usuario y el precio de la energía eléctrica en cada instante de tiempo. En cada paso el algoritmo selecciona para cada usuario y cada dispositivo eléctrico un intervalo de tiempo en donde realizar la carga del dispositivo siempre y cuando no se vea superada la energía eléctrica contratada por el usuario. Para el intervalo encontrado se toma el valor de aplicar la ecuación 2.5 (*PrefCosto*). El algoritmo continua buscando intervalos en donde realizar las cargas del dispositivo y comparando el valor de *prefCosto* de cada intervalo. Una vez encontrado el mejor intervalo de carga el algoritmo almacena el valor de *prefCosto* en *valorOptimo* y marca el intervalo como utilizado. Se continua la búsqueda de intervalos en donde realizar la carga del dispositivo siempre y cuando el valor de *prefCosto* mas *valorOptimo* hallado sea mayor al *valorOptimo* hallado hasta el momento. La planificación de carga del dispositivo eléctrico esta dada por aquellos intervalos con mejores *valorOptimo*.

El esquema del algoritmo ávido se presenta en el Algoritmo 3.

Algoritmo 3 Algoritmo ávido

```
1: procedure BUSCARINTERVALOMAXPREF(u,m,d,X, intervalosUtilizados)
2:   pref = 0
3:   duracion = 0
4:   for (min=m; min <  $t_M$ ; min++) do
5:     if duracion < D(d,u) then
6:       if intervalosUtilizados[m]==0 &&
7:          $\sum_{k=1}^K x_{km}^u * P(l_k) + P(d) < E(u)$ 
8:       then
9:         pref += UP(u,d,min)* $\alpha$  - C(min)* $\beta$ 
10:        duracion += (  $t_{min+1} - t_{min}$  )
11:      else
12:        pref = 0
13:        duracion = 0
14:      end ifend
15:    else
16:      return min,pref
17:    end ifend
18:  end forend
19:  return min,pref
20: end procedure end buscarintervalomaxpref
21:  $X = \vec{0}$ 
22: for ( $i = 1$ ;  $i \leq N$ ;  $i++$ ) do ▷ Para cada usuario
23:   for ( $k=1$ ;  $i \leq K$ ;  $k++$ ) do ▷ Para cada dispositivo eléctrico
24:     mejorPrefCosto, prefCosto = MIN_INT
25:     mejorMin, valorOptimo = 0
26:     intervalosUtilizados =  $\vec{0}$ 
27:     while valorOptimo==0 ||
28:       ( $\text{mejorPrefCosto} + \text{valorOptimo} \geq \text{valorOptimo}$ )
29:     do
30:       valorOptimo = mejorPrefCosto + valorOptimo
31:       for ( $m=t_1$ ;  $m < t_M - D(d_k, u_i)$ ;  $m++$ ) do
32:         min, prefCosto = buscarintervalomaxpref( $u_i, m, d_k, X,$ 
33:           intervalosUtilizados)
34:         if prefCosto > mejorPrefCosto then
35:           mejorPrefCosto = prefCosto
36:           mejorMin = min
37:           actualizar(intervalosUtilizados)
38:         end ifend
39:       end forend
40:       for ( $m=\text{mejorMin} - D(d_k, u_i)$ ;  $m \leq \text{mejorMin}$ ;  $m++$ ) do
41:          $x_{km}^i = 1$  ▷ encender dispositivos
42:       end forend
43:     end whileend
44:   end forend
45: end forend
```

El Algoritmo 3 utiliza una función auxiliar *BuscarIntervaloMaxPref* que recibe de entrada el parámetro u que indica un usuario, el parámetro m que indica el intervalo de tiempo en donde comenzar a buscar la carga del dispositivo, el parámetro d que indica un dispositivo eléctrico, el vector X que indica la planificación de uso de dispositivos eléctricos del usuario u y el vector *intervalosUtilizados* que indica en que momento el dispositivo d esta encendido (línea 1). Con la información provista por los parámetros de entrada la función *BuscarIntervaloMaxPref* busca un intervalo de tiempo de largo duración del dispositivo en donde realizar la carga del dispositivo d sin que la potencia máxima del usuario se vea superada (línea 2-15). En caso de encontrar un intervalo en donde realizar la carga la función *BuscarIntervaloMaxPref* retorna el valor de preferencia para el intervalo dado por la ecuación 2.5 y el minuto min que indica la finalización de la carga del dispositivo. En caso de no encontrar ningún intervalo desde el minuto m al final del intervalo de tiempo a planificar (t_M) la función *BuscarIntervaloMaxPref* retorna 0. El vector X se inicializa vacío para la cantidad de usuarios (línea 17), para cada usuario y para cada dispositivo eléctrico del usuario se buscan el intervalo de carga del dispositivo que maximicen el valor de *prefCosto* dado por la función *BuscarIntervaloMaxPref*. Una vez encontrado el mejor intervalo de carga el algoritmo almacena el valor de *prefCosto* en *valorOptimo* y marca el intervalo como utilizado. Se continua la búsqueda de intervalos en donde realizar la carga del dispositivo siempre y cuando el valor de *prefCosto* mas *valorOptimo* hallado sea mayor al *valorOptimo* hallado hasta el momento (línea 18-33). Por ultimo se actualiza el vector X agregando la utilización del dispositivo d a cada intervalo de tiempo tomando el *mejorMin* como la finalización de la carga del dispositivo (línea 33-34).

Algoritmos ávidos para MOEA

Se implementaron tres algoritmos ávidos, cada algoritmo busco optimizar una de las funciones objetivos analizadas en la sección 2.1.2. Los tres algoritmos tomaron como restricción que el dispositivo a encender esté encendido al menos la duración de carga del mismo dada por la función D definida en la sección 2.1.2. Otra restricción de los algoritmos fue que en para ningún intervalo de tiempo la energía eléctrica contratada por el usuario se vea superada. Los algoritmos ávidos se describen a continuación.

Algoritmo ávido QoS

El algoritmo ávido QoS buscó maximizar la preferencia del usuario. Para cada usuario y para cada dispositivo eléctrico, se buscó obtener el intervalo de tiempo en el que la preferencia del usuario de encender el dispositivo sea máxima sin tener en cuenta el precio de la energía consumida. En cada paso el algoritmo selecciona para cada usuario y cada dispositivo eléctrico un intervalo de tiempo en donde realizar la carga del dispositivo siempre y cuando no se vea superada la energía eléctrica contratada por el usuario. Para el intervalo encontrado se toma la preferencia del usuario (*pref*). El algoritmo continua buscando intervalos en donde realizar la carga del dispositivo y comparando el valor de *pref* de cada intervalo. La planificación de carga del dispositivo eléctrico esta dada por el intervalo que tiene mayor *pref*. El esquema del algoritmo ávido QoS se presenta en el Algoritmo 4.

Algoritmo 4 Algoritmo ávido QoS

```
1: procedure BUSCARINTERVALOMAXPREF(u,m,d,X)
2:   pref = 0
3:   duracion = 0
4:   for (min=m; min <  $t_M$ ; min++) do
5:     if duracion < D(d,u) then
6:       if  $\sum_{k=1}^K x_{km}^u * P(l_k) + P(d) < E(u)$  then
7:         pref += UP(u,d,min)
8:         duracion += (  $t_{min+1} - t_{min}$  )
9:       else
10:        pref = 0
11:        duracion = 0
12:      end ifend
13:    else
14:      return min,pref
15:    end ifend
16:  end forend
17:  return min,pref
18: end procedureend buscarintervalomaxpref
19:  $\vec{X} = \vec{0}$ 
20: for ( $i = 1$ ;  $i \leq N$ ;  $i++$ ) do ▷ Para cada usuario
21:   for ( $k=1$ ;  $k \leq K$ ;  $k++$ ) do ▷ Para cada dispositivo eléctrico
22:     mejorPref = MIN_INT
23:     pref = MIN_INT,
24:     mejorMin = 0
25:     for ( $m=t_1$ ;  $m < t_M - D(d_k, u_i)$ ;  $m++$ ) do
26:       min, pref = buscarintervalomaxpref( $u_i, m, d_k, X$ )
27:       if pref > mejorPref then
28:         mejorPref= pref
29:         mejorMin = min
30:       end ifend
31:     end forend
32:     for ( $m=mejorMin - D(d_k, u_i)$ ;  $m \leq mejorMin$ ;  $m++$ ) do
33:        $x_{km}^i = 1$  ▷ encender dispositivos
34:     end forend
35:   end forend
36: end forend
```

Algoritmo ávido energía eléctrica contratada

El algoritmo ávido energía eléctrica contratada implementado busco minimizar la energía eléctrica consumida. Para cada usuario y para cada dispositivo eléctrico, se busco obtener el intervalo de tiempo en el cual encender el dis-

positivo de forma tal que la energía consumida sea mínima. En cada paso el algoritmo selecciona para cada usuario y cada dispositivo eléctrico un intervalo de tiempo en donde realizar la carga del dispositivo siempre y cuando no se vea superada la energía eléctrica contratada por el usuario. Para el intervalo encontrado se toma la energía consumida por los dispositivos en el intervalo (*energia*). El algoritmo continua buscando intervalos en donde realizar la carga del dispositivo y comparando el valor de *energia* de cada intervalo. La planificación de carga del dispositivo eléctrico esta dada por el intervalo que tiene mayor *energia*. El esquema del algoritmo ávido energía consumida se presenta en el Algoritmo 5.

Algoritmo 5 Algoritmo ávido energía eléctrica consumida

```
1: procedure BUSCARINTERVALOMINENERGIA(u,m,d,X)
2:   energia = 0
3:   duracion = 0
4:   pref = 0
5:   for (min=m; min <  $t_M$ ; min++) do
6:     if duracion < D(d,u) then
7:       if  $\sum_{k=1}^K x_{km}^u * P(l_k) + P(d) < E(u)$  then
8:         energia += P(d)
9:         duracion += (  $t_{min+1} - t_{min}$  )
10:      else
11:        energia = 0
12:        duracion = 0
13:      end ifend
14:    else
15:      return min,energia
16:    end ifend
17:  end forend
18:  return min,energia
19: end procedure end buscarintervalominenergia
20:  $X = \vec{0}$ 
21: for ( $i = 1$ ;  $i \leq N$ ;  $i++$ ) do ▷ Para cada usuario
22:   for ( $k=1$ ;  $i \leq K$ ;  $k++$ ) do ▷ Para cada dispositivo eléctrico
23:     mejorEnergia = Max_INT
24:     energia = Max_INT,
25:     mejorMin = 0
26:     for ( $m=t_1$ ;  $m < t_M - D(d_k, u_i)$ ;  $m++$ ) do
27:       min, energia = buscarintervalominenergia( $u_i, m, d_k, X$ )
28:       if energia < mejorEnergia then
29:         mejorEnergia = energia
30:         mejorMin = min
31:       end ifend
32:     end forend
33:     for ( $m=mejorMin - D(d_k, u_i)$ ;  $m \leq mejorMin$ ;  $m++$ ) do
34:        $x_{km}^i = 1$  ▷ encender dispositivos
35:     end forend
36:   end forend
37: end forend
```

El Algoritmo 5 sigue la misma lógica del Algoritmo 4 cambiando únicamente como se toma el valor de *pref* de cada intervalo, en este caso el valor de preferencia está dado por la energía consumida en el intervalo.

Algoritmo ávido precio

El algoritmo ávido precio buscó minimizar el precio de la factura eléctrica. Para cada usuario y para cada dispositivo eléctrico, se buscó obtener el intervalo de tiempo en el cual encender el dispositivo de forma tal que el precio de la energía sea mínima. Para este algoritmo se tuvo en cuenta también la preferencia del usuario en distintos niveles, 0% para indicar que no se contempla la satisfacción del usuario, 100% para indicar que se desea obtener el precio de energía mínimo pero respetando enteramente las preferencias del usuario, y niveles intermedios para obtener diferentes soluciones (10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%). El esquema del algoritmo ávido precio se presenta en el Algoritmo 6.

Algoritmo 6 Algoritmo ávido energía precio

```
1: procedure BUSCARINTERVALOMINPRECIO(u,m,d,X, pref)
2:   precio = 0
3:   duracion = 0
4:   preferencia = 0
5:   for (min=m; min <  $t_M$ ; min++) do
6:     if duracion < D(d,u) then
7:       if  $\sum_{k=1}^K x_{km}^u * P(l_k) + P(d) < E(u)$  then
8:         precio += C(min)
9:         duracion += (  $t_{min+1} - t_{min}$  )
10:      else
11:        precio = 0
12:        duracion = 0
13:        preferencia = 0
14:      end ifend
15:    else
16:      return min,precop, pref
17:    end ifend
18:  end forend
19:  return min,precio, pref
20: end procedureend buscarintervalominprecio
21:  $X = \vec{0}$ 
22: prefUsuario=0.8                                ▷ Ejemplo 80 % de pref usuario
23: for (i = 1; i ≤ N; i++) do                      ▷ Para cada usuario
24:   for (k=1; i ≤ K; k++) do                       ▷ Para cada dispositivo eléctrico
25:     mejorPref = MIN_INT
26:     pref = MIN_INT
27:     for (m=t1; m <  $t_M - D(d_k, u_i)$ ; m++) do
28:       min, pref = buscarintervalomaxpref( $u_i, m, d_k, X$ )
29:       if pref > mejorPref then
30:         mejorPref= pref
31:       end ifend
32:     end forend
33:     mejorPrecio = Max_INT
34:     precio = Max_INT,
35:     mejorMin = 0
36:     prefAux = 0
37:     for (m=t1; m <  $t_M - D(d_k, u_i)$ ; m++) do
38:       min,precio,prefAux = buscarintervalominprecio( $u_i, m, d_k, X, pref$ )
39:       if precio < mejorPrecio && pref > prefUsuario * prefAux then
40:         mejorPrecio = precio
41:         mejorMin = min
42:       end ifend
43:     end forend
44:     for (m=mejorMin - D( $d_k, u_i$ ); m ≤ mejorMin; m++) do
45:        $x_{km}^i = 1$                                 ▷ encender dispositivos
46:     end forend
47:   end forend
48: end forend
```

El Algoritmo 6 sigue el misma lógica del Algoritmo 4 agregando la búsqueda del mejor intervalo de máxima preferencia del algoritmo 4 (lineas 21-26). La preferencia esta dada por el precio de cada intervalo teniendo en cuenta cierto nivel de preferencia del usuario al elegir el intervalo.

6.2.5. Estrategia *Business as Usual*

La estrategia *Business as Usual* (BaU) propone una planificación exactamente igual al consumo actual del usuario. Para todas las instancias evaluadas se realizo también la planificación BaU y se estudio el resultado de aplicar las funciones definidas en sección 2.1.2. Es relevante comparar el desempeño del MOEA y del algoritmo evolutivo monoobjetivo frente a la solución de BaU de forma de tener un valor comparativo frente a la planificación actual del usuario.

El esquema de la estrategia BaU se presenta en el Algoritmo 8.

Algoritmo 7 Estrategia BaU

```

1:  $X = \vec{0}$ 
2: for ( $i = 1; i \leq N; i++$ ) do                                     ▷ Para cada usuario
3:   for ( $k=1; k \leq K; k++$ ) do                                     ▷ Para cada dispositivo eléctrico
4:     for ( $m=t_1; m < t_M - D(d_k, u_i); m++$ ) do
5:       if UP( $i, k, m$ ) then
6:          $x_{km}^i = 1$                                              ▷ encender dispositivos
7:       end if
8:     end forend
9:   end forend
10: end forend

```

Algoritmo 8 Estrategia BaU

```

1:  $X = \vec{0}$ 
2: for u in Usuarios do                                           ▷ Para cada usuario
3:   for d in Dispositivos do                                       ▷ Para cada dispositivo eléctrico
4:     for m in Minutos do
5:       if UP( $u, d, m$ ) then
6:          $x_{dm}^u = 1$                                              ▷ encender dispositivos
7:       end if
8:     end forend
9:   end forend
10: end forend

```

6.2.6. Métricas del MOEA

Diferentes métricas han sido propuestas en la literatura para poder evaluar la calidad de los resultados obtenidos mediante los MOEA, teniendo en cuenta tanto la convergencia hacia el frente de Pareto como la extensión del frente de soluciones hallado. A continuación, se definen las métricas utilizadas en la evaluación experimental.

- Número de puntos no dominados: Evalúa la cantidad de soluciones distintas no dominadas encontradas por el algoritmo (ND).
- Distancia generacional: Evalúa la distancia promedio entre las soluciones no dominadas encontradas por el algoritmo y el frente de Pareto real (Ecuación 6.1). Por lo tanto, valores pequeños indican una mayor proximidad al frente de Pareto real (DG).

$$DG = \frac{1}{ND} \left(\sum_{i=1}^{ND} (d_i^{FP})^k \right)^{\frac{1}{k}} \quad (6.1)$$

- Spread: Evalúa la distribución de soluciones no dominadas en el frente de Pareto calculado, incluyendo la distancia hacia los extremos del frente de Pareto real de forma de obtener un valor más exacto de la distribución (6.2). Valores más cercanos a cero indican una mejor distribución de las soluciones no dominadas.

$$spread = \frac{\sum_{h=1}^k d_h^e + \sum_{i=1}^{ND} (\bar{d} - d_i)^2}{\sum_{h=1}^k d_h^e + ND \times \bar{d}} \quad (6.2)$$

- Hipervolumen relativo: Evalúa el cociente entre el volumen (en el espacio de las funciones objetivo) cubierto por el frente de Pareto calculado por el algoritmo y el volumen cubierto por el frente de Pareto real. Por lo tanto, el valor ideal del hipervolumen relativo es 1 (HR).

En la ecuación 6.1, d_i^{FP} es la distancia entre la i -ésima solución en el frente de Pareto calculado y el punto más cercano en el frente de Pareto real. En la ecuación 6.2, d_i es la distancia entre la solución i -ésima del frente de Pareto calculado y su vecino más cercano (la solución j -ésima), mientras que \bar{d} es el promedio de todos los d_i , d_h^e es la distancia entre el punto extremo del frente de Pareto real (considerando la h -ésima función objetivo) y el punto más cercano en el frente de Pareto calculado. Debido a que algunas métricas

utilizan el frente de Pareto real y éste es desconocido para las instancias del problema a resolver, se construye una aproximación combinando de los puntos no dominados obtenidos en las 50 ejecuciones independientes realizadas.

El número de puntos no dominados y la distancia generacional evalúan la calidad de los resultados obtenidos utilizando un MOEA, en términos de convergencia al frente de Pareto. Por otro lado, la métrica de spread evalúa la distribución de las soluciones no dominadas, midiendo la capacidad de muestrear el frente de Pareto del problema a resolver.

6.3. Variante monoobjetivo

En esta sección se detalla el análisis experimental correspondiente a la variante monoobjetivo del problema de planificación de dispositivos electrónicos en hogares.

6.3.1. Asignación de fitness

El algoritmo evolutivo monoobjetivo buscó maximizar una función lineal que tuviese en cuenta la preferencia del usuario y el precio de la energía eléctrica en cada instante de tiempo. Se estudiaron distintas combinaciones de α y β que maximicen la función 2.5

Fueron evaluadas tres instancias chicas del problema distintas a las utilizadas en la evaluación experimental. Cada una de las instancias consideró una única casa con dos, cuatro y seis dispositivos eléctricos respectivamente. Para el ajuste paramétrico se utilizaron valores medios de recombinación (p_C , valor candidato 0.25), mutación (p_M , valores candidato 0.05) y generaciones (G , valor candidato 5000) de los analizados en la sección 6.2.3. Se estudiaron las combinaciones de valores de α (α , valores candidatos 0.3, 0.5, 0.65, 0.75, 1.0) y β (β , valores candidato 0.3, 0.5, 1.0) sobre las tres instancias de calibración, realizando 30 ejecuciones independientes en cada caso. Se aplicó la prueba de Shapiro-Wilk, y se tomó como hipótesis nula que los resultados seguían una distribución normal; como se obtuvieron algunos p-valores menores a 0.05, se descartó la hipótesis nula y los resultados de fitness se consideraron como correspondientes a una distribución no normal.

Los resultados numéricos de las medianas y el rango intercuartílico (IQR) para cada combinación de valores de alfa y beta se muestran en detalle en la tabla 6.1. Se reportan valores promedios de mediana y de IQR para cada instancia.

α	β	Instancia 1		Instancia 2		Instancia 3	
		Mediana	IQR	Mediana	IQR	Mediana	IQR
0.3	0.3	15.81	1.09	179.83	0.55	132.35	6.68
0.3	0.5	11.02	1.54	167.94	2.03	110.10	4.54
0.3	1.0	4.27	6.39	128.55	28.66	54.82	13.56
0.5	0.3	34.06	0.81	311.32	28.66	253.95	5.11
0.5	0.5	26.75	1.06	298.80	23.71	224.47	12.21
0.5	1.0	13.53	10.28	269.83	26.03	175.11	40.39
0.65	0.3	47.28	2.08	409.91	1.76	342.66	6.57
0.65	0.5	39.87	0.83	398.45	1.34	316.32	8.40
0.65	1.0	25.48	6.12	369.21	3.91	257.55	39.03
0.75	0.3	54.85	2.79	475.64	0.39	405.86	10.03
0.75	0.5	48.52	3.20	462.74	35.73	373.95	4.11
0.75	1.0	30.18	3.43	433.82	3.23	304.31	21.64
1.0	0.3	81.21	4.41	640.62	0.59	548.20	5.64
1.0	0.5	71.46	4.55	628.42	2.76	524.55	9.10
1.0	1.0	52.05	2.89	597.33	0.24	454.48	20.50

Tabla 6.1: Detalle de resultados de la configuración de α y β del AE monoobjetivo

La tabla 6.2 reporta las distintas combinaciones de α y β y los resultados obtenidos de las funciones F y G .

α	β	$F(X)$	$G(X)$	α	β	$F(X)$	$G(X)$
0.3	0.3	442.56	79.55	0.65	1.0	433.72	67.87
0.3	0.5	428.20	65.70	0.75	0.3	452.73	93.95
0.3	1.0	355.75	48.81	0.75	0.5	431.02	88.01
0.5	0.3	452.45	90.18	0.75	1.0	432.80	73.64
0.5	0.5	423.00	79.03	1.0	0.3	454.34	103.85
0.5	1.0	385.64	58.51	1.0	0.5	434.58	91.07
0.65	0.3	431.85	90.55	1.0	1.0	428.74	78.09
0.65	0.5	450.09	82.45				

Tabla 6.2: Calibración fitness algoritmo evolutivo monoobjetivo

La figura 6.1 presenta de forma gráfica los resultados reportados en la tabla 6.2.

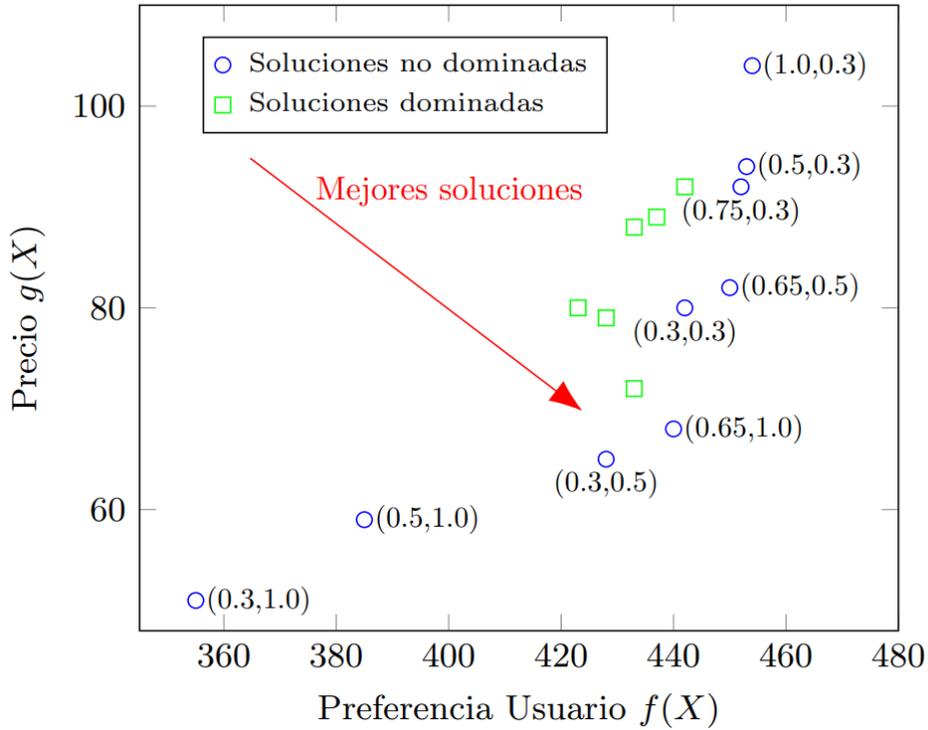


Figura 6.1: Resultados de la configuración paramétrica del AE monoobjetivo

A partir del análisis realizado la combinación de valores de α y β que optimiza tanto la preferencia del usuario como el precio de la energía es $\alpha = 0.65$ y $\beta = 1$.

6.3.2. Configuración paramétrica

Se estudiaron los valores de G , p_C y p_M mediante el procedimiento descrito en la sección 6.2.3. La métrica considerada para el análisis fue el fitness (Ecuación 2.5). Las instancias utilizadas en la configuración paramétrica son las mencionadas en la sección 6.2.3. Para la elección de la mejor configuración se realizaron los siguientes pasos:

- Test de Normalidad: Se aplicó la prueba de Shapiro-Wilk, y se tomó como hipótesis nula que los resultados seguían una distribución normal; como se obtuvieron algunos p-valores menores a 0.05, se descartó la hipótesis nula y los resultados de fitness se consideraron como correspondientes a una distribución no normal.
- Test de Friedman: Se aplicó la prueba de rangos de Friedman y se tomó como hipótesis nula que las distribuciones de fitness para las dife-

rentes configuraciones eran no diferenciables; como se obtuvieron valores menores a 0.05, se descarto la hipótesis nula y los resultados se consideraron diferenciables.

- Se realizó un torneo entre los mejores valores de mediana, seleccionando como la mejor configuración la que alcanzó mejor valor de mediana.

La Tabla 6.3 reporta en detalle las distintas combinaciones de configuraciones paramétricas consideradas en el estudio.

Configuración	G	P_M	P_C	Configuración	G	P_M	P_C
#0	2500	0.01	0.1	#14	10000	0.1	0.5
#1	5000	0.01	0.1	#15	2500	0.1	0.25
#2	10000	0.01	0.1	#16	5000	0.1	0.25
#3	2500	0.01	0.5	#17	10000	0.1	0.25
#4	5000	0.01	0.5	#18	2500	0.05	0.1
#5	10000	0.01	0.5	#19	5000	0.05	0.1
#6	2500	0.01	0.25	#20	10000	0.05	0.1
#7	5000	0.01	0.25	#21	2500	0.05	0.5
#8	10000	0.01	0.25	#22	5000	0.05	0.5
#9	2500	0.1	0.1	#23	10000	0.05	0.5
#10	5000	0.1	0.1	#24	2500	0.05	0.25
#11	10000	0.1	0.1	#25	5000	0.05	0.25
#12	2500	0.1	0.5	#26	1000	0.05	0.25
#13	5000	0.1	0.5				

Tabla 6.3: Configuración paramétrica AE

La figura 6.2 presenta una gráfica de barras que compara el promedio tomando las medianas en cada instancia, para cada combinación de valores de la evaluación paramétrica. Los resultados en la figura sugieren utilizar la configuración 11 con valores $P_C = 0.1$, $P_M = 0.1$ y $G = 10000$.

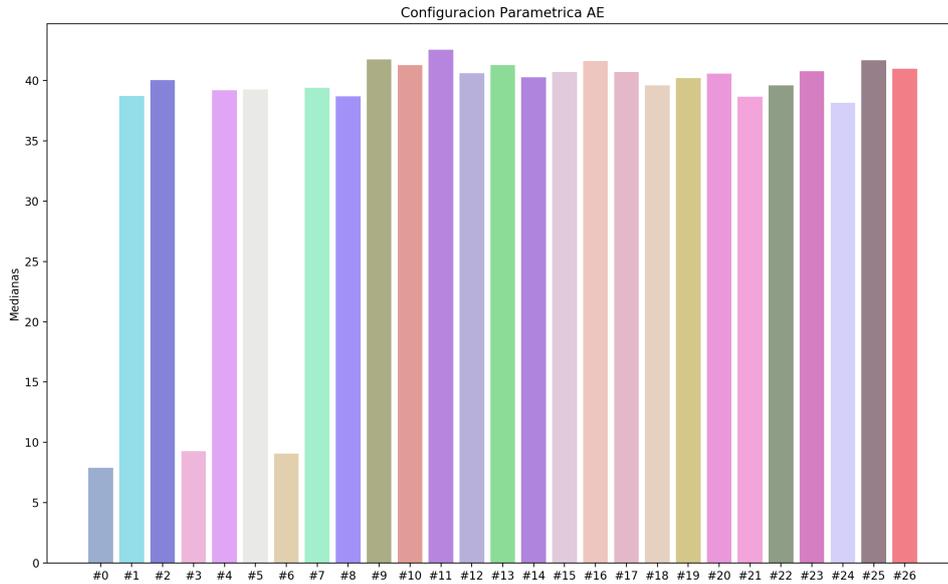


Figura 6.2: Resultados de la configuración paramétrica del AE monoobjetivo

Los resultados numéricos de las medianas y el rango intercuartílico (IQR) para cada combinación de valores de la evaluación paramétrica se muestran en detalle en la tabla 6.4. Se reportan valores promedios de mediana y de IQR para una instancia representativa del problema.

Configuración	Mediana	IQR	Configuración	Mediana	IQR
#0	7.88	10.45	#14	40.27	7.61
#1	38.74	11.71	#15	40.73	7.01
#2	40.04	6.98	#16	41.64	6.76
#3	9.26	11.17	#17	40.72	8.67
#4	39.22	9.37	#18	39.60	6.97
#5	39.28	9.88	#19	40.22	9.23
#6	9.06	9.22	#20	40.58	7.95
#7	39.39	10.18	#21	38.66	7.39
#8	38.69	11.76	#22	39.59	10.07
#9	41.77	8.88	#23	40.79	11.62
#10	41.30	8.46	#24	38.17	10.32
#11	42.57	6.46	#25	41.67	8.38
#12	40.62	7.22	#26	41.00	8.44
#13	41.28	8.65			

Tabla 6.4: Detalle de resultados de la configuración paramétrica del AE monoobjetivo

6.3.3. Resultados Números

La tabla 6.5 reporta la comparación de los promedios de los mejores fitness calculados por los algoritmos en cada una de las instancias ejecutadas, así como el porcentaje de mejora sobre la solución calculada por los algoritmos ávido y BaU. La mejora es calculada por la función $((f(AE) - f(algoritmo))/f(algoritmo)) \times 100$. Se destacan en negrita los mejores valores de mejora del algoritmo evolutivo con respecto al algoritmo ávido y a la estrategia BaU.

	f(AE)	f(Greedy)	%Mejora Greedy	f(BaU)	%Mejora BaU
chica	14.07	11.37	23 %	-7.62	284.65 %
mediana	340.02	259.71	30 %	69.54	388.96 %
grande	407.65	347.98	17 %	-187.18	317 %
edificio	417.16	347.98	19 %	-187.18	322.87 %
chica fin de semana	323.67	255.51	26 %	67.43	380.01 %
mediana fin de semana	253.39	197.18	28 %	153.79	64.76 %
grande fin de semana	369.81	351.90	5 %	-299.97	223.28 %
edificio fin de semana	351.90	297.18	18 %	-299.97	217.31 %

Tabla 6.5: Porcentaje de mejora AE.

La tabla 6.5 muestra que en todas las instancias estudiadas el AE mono-objetivo es capaz de alcanzar mejoras sobre las soluciones calculadas por el algoritmo ávido. En el mejor caso el algoritmo evolutivo supera al algoritmo ávido en un **30 %**; instancia mediana del problema. Los resultados también indican que los patrones de consumo de los usuarios en los fines de semana son mas difíciles de predecir por el AE ya que las mejoras con respecto al algoritmo ávido se reducen al **5 %**; instancia grande fin de semana. Esto ultimo puede deberse a que en fines de semana los usuarios interactúan continuamente y de forma distinta con los dispositivos eléctricos. Las mejoras del AE con respecto al algoritmo BaU son esperables debido a que el algoritmo BaU representa el consumo del usuario sin tener en cuenta ningún tipo de planificación. Además las soluciones propuesta por BaU no respetan el encendido de los dispositivos eléctricos durante el periodo de duración de los mismos, por lo que la ecuación de preferencia del usuario (Ecuación 2.2) penaliza la solución.

	Greedy		BaU	
	F(X)	G(X)	F(X)	G(X)
chica	48 %	25 %	16 %	81 %
mediana	50 %	29 %	271 %	52 %
grande	51 %	21 %	153 %	77 %
edificio	43 %	19 %	159 %	76 %
chica fin de semana	45 %	28 %	347 %	45 %
mediana fin de semana	37 %	24 %	98 %	64 %
grande fin de semana	42 %	23 %	142 %	79 %
edificio fin de semana	36 %	17 %	87 %	90 %

Tabla 6.6: Detalle de mejoras por objetivo del algoritmo evolutivo al algoritmo BaU

La tabla 6.6 muestra que el AE es capaz de computar planificaciones significativamente más accesibles (en términos de costos) que aquellas computadas por la estrategia BaU, que significativamente fallan en proveer una buena relación costo-beneficio, indicando que los usuarios no toman decisiones correctas a la hora de encender los dispositivos eléctricos. Con respecto al algoritmo ávido el algoritmo AE reporta notorias mejoras tanto en la calidad de servicio como en el costo de la energía eléctrica. Los resultados obtenidos sugieren que el enfoque evolutivo aporta notables mejoras al planificar el consumo energético del hogar, teniendo en cuenta tanto el costo energético y la satisfacción del usuario. El enfoque sugerido es un primer paso hacia el diseño de un sistema de recomendaciones inteligentes destinado a usuarios finales.

6.4. Variante multiobjetivo

En esta sección se detalla el análisis experimental correspondiente a la variante multiobjetivo del problema de planificación de dispositivos electrónicos en hogares.

6.4.1. Configuración paramétrica

Se estudiaron los valores de G , p_C y p_M mediante el procedimiento descrito en la sección 6.2.3. La métrica considerada para el análisis fueron las ecuaciones definidas en 2.1.2). Las instancias utilizadas en la configuración paramétrica

son las mencionadas en la sección 6.1.2. Para la elección de la mejor configuración se realizaron los siguientes pasos.

- Test de Normalidad: Se aplicó la prueba de Shapiro-Wilk, y se tomó como hipótesis nula que los resultados seguían una distribución normal; como se obtuvieron algunos p-valores menores a 0.05, se descartó la hipótesis nula y los resultados de fitness se consideraron como correspondientes a una distribución no normal.
- Test de Friedman: Se aplicó la prueba de rangos de Friedman y se tomó como hipótesis nula que las distribuciones de fitness para las diferentes configuraciones eran no diferenciables; como se obtuvieron valores menores a 0.05, se descartó la hipótesis nula y los resultados se consideraron diferenciables.
- Test de Rangos: Una vez que se obtuvo información sobre la diferenciableidad de los datos se realizó un test de rangos de Friedman para encontrar la mejor configuración paramétrica.

En la Tabla 6.7 se muestran en detalle las distintas combinaciones de configuraciones paramétricas.

Configuración	G	P_M	P_C	Configuración	G	P_M	P_C
#0	2500	0.01	0.1	#14	10000	0.1	0.5
#1	5000	0.01	0.1	#15	2500	0.1	0.25
#2	10000	0.01	0.1	#16	5000	0.1	0.25
#3	2500	0.01	0.5	#17	10000	0.1	0.25
#4	5000	0.01	0.5	#18	2500	0.05	0.1
#5	10000	0.01	0.5	#19	5000	0.05	0.1
#6	2500	0.01	0.25	#20	10000	0.05	0.1
#7	5000	0.01	0.25	#21	2500	0.05	0.5
#8	10000	0.01	0.25	#22	5000	0.05	0.5
#9	2500	0.1	0.1	#23	10000	0.05	0.5
#10	5000	0.1	0.1	#24	2500	0.05	0.25
#11	10000	0.1	0.1	#25	5000	0.05	0.25
#12	2500	0.1	0.5	#26	1000	0.05	0.25
#13	5000	0.1	0.5				

Tabla 6.7: Configuración paramétrica

En la Tabla 6.8 se reportan los resultados de la configuración paramétrica para cada instancia y cada configuración estudiada.

Configuración	Instancia 1	Instancia 2	Instancia 3
#0	0.08 ± 0.02 (0.12)	0.01 ± 0.00 (0.04)	0.02 ± 0.02 (0.09)
#1	0.16 ± 0.02 (0.23)	0.07 ± 0.02 (0.13)	0.03 ± 0.02 (0.09)
#2	0.27 ± 0.01 (0.31)	0.22 ± 0.04 (0.31)	0.16 ± 0.03 (0.27)
#3	0.08 ± 0.02 (0.18)	0.01 ± 0.00 (0.03)	0.01 ± 0.02 (0.10)
#4	0.16 ± 0.02 (0.21)	0.08 ± 0.03 (0.22)	0.03 ± 0.02 (0.08)
#5	0.27 ± 0.02 (0.32)	0.22 ± 0.03 (0.29)	0.15 ± 0.03 (0.23)
#6	0.08 ± 0.02 (0.15)	0.01 ± 0.01 (0.05)	0.02 ± 0.02 (0.13)
#7	0.16 ± 0.02 (0.24)	0.07 ± 0.02 (0.15)	0.02 ± 0.02 (0.09)
#8	0.27 ± 0.01 (0.32)	0.21 ± 0.03 (0.30)	0.16 ± 0.03 (0.23)
#9	0.47 ± 0.07 (0.63)	0.40 ± 0.02 (0.48)	0.33 ± 0.02 (0.40)
#10	0.78 ± 0.06 (0.89)	0.66 ± 0.07 (0.77)	0.66 ± 0.06 (0.77)
#11	0.92 ± 0.01 (0.95)	0.90 ± 0.01 (0.94)	0.92 ± 0.02 (0.95)
#12	0.45 ± 0.06 (0.62)	0.39 ± 0.01 (0.45)	0.33 ± 0.02 (0.39)
#13	0.77 ± 0.07 (0.87)	0.66 ± 0.07 (0.78)	0.63 ± 0.07 (0.76)
#14	0.92 ± 0.01 (0.95)	0.91 ± 0.01 (0.94)	0.92 ± 0.02 (0.95)
#15	0.45 ± 0.69 (0.68)	0.40 ± 0.02 (0.45)	0.33 ± 0.03 (0.41)
#16	0.77 ± 0.07 (0.86)	0.66 ± 0.06 (0.78)	0.64 ± 0.08 (0.81)
#17	0.92 ± 0.13 (0.95)	0.90 ± 0.01 (0.94)	0.91 ± 0.02 (0.95)
#18	0.30 ± 0.02 (0.35)	0.24 ± 0.02 (0.30)	0.16 ± 0.03 (0.26)
#19	0.46 ± 0.06 (0.60)	0.43 ± 0.03 (0.53)	0.38 ± 0.03 (0.45)
#20	0.78 ± 0.06 (0.88)	0.70 ± 0.07 (0.81)	0.69 ± 0.07 (0.82)
#21	0.30 ± 0.02 (0.43)	0.23 ± 0.02 (0.29)	0.15 ± 0.03 (0.25)
#22	0.48 ± 0.08 (0.68)	0.43 ± 0.03 (0.55)	0.38 ± 0.04 (0.50)
#23	0.78 ± 0.7 (0.92)	0.72 ± 0.06 (0.81)	0.68 ± 0.07 (0.81)
#24	0.30 ± 0.02 (0.38)	0.24 ± 0.03 (0.32)	0.15 ± 0.02 (0.21)
#25	0.46 ± 0.08 (0.67)	0.42 ± 0.02 (0.50)	0.38 ± 0.04 (0.52)
#26	0.77 ± 0.09 (0.87)	0.70 ± 0.07 (0.83)	0.72 ± 0.07 (0.82)

Tabla 6.8: Resultados obtenidos para el hipervolumen relativo en la configuración paramétrica.

La figura 6.3 muestra el resultado del test de rangos de Friedman, la cual sugiere utilizar la configuración #14 con valores $P_C = 0.5$, $P_M = 0.1$ y $G = 10000$.

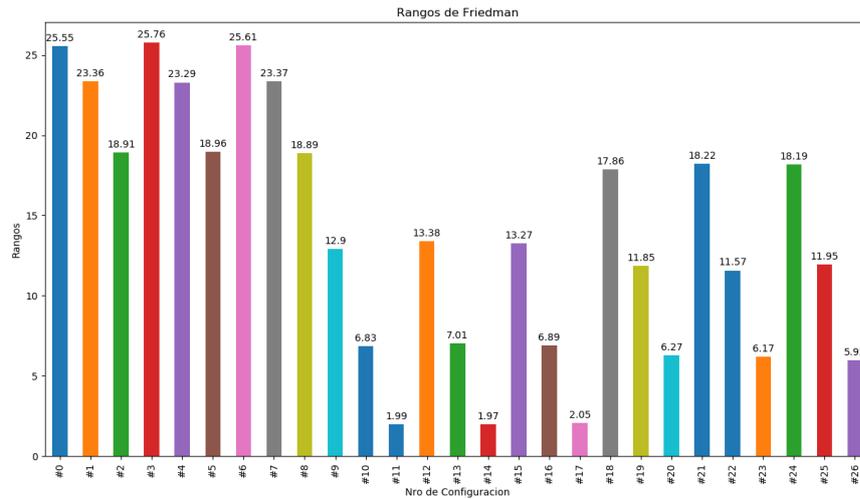


Figura 6.3: Test de rangos.

6.4.2. Resultados numéricos para el problema multiobjetivo

En esta sección se presentan los resultados obtenidos en la evaluación experimental del algoritmo NSGA-II implementado para la resolución del problema. Al igual que para la configuración paramétrica, se realizaron 50 ejecuciones independientes para cada una de las instancias del problema definidas en la sección 6.1.2.

Las Tablas 6.9, 6.10 y 6.11 muestran los resultados de las métricas obtenidos por el algoritmo NSGA-II agrupados por familia de instancias para las métricas definidas en la Sección 6.2.6. Se reportan los valores promedio, la desviación estándar y el mejor valor alcanzado entre paréntesis.

DG	
chica	0.0027 ± 0.0006 (0.0034)
mediana	0.0156 ± 0.0021 (0.0190)
grande	0.0124 ± 0.0051 (0.0217)
edificio	0.0125 ± 0.0044 (0.0271)
chica fin de semana	0.0031 ± 0.0003 (0.0033)
mediana fin de semana	0.0001 ± 0.0001 (0.0004)
grande fin de semana	0.0028 ± 0.0007 (0.0049)
edificio fin de semana	0.0166 ± 0.0055 (0.0282)

Tabla 6.9: Resultados de las métrica DG para el algoritmo NSGA-II.

Spread	
chica	0.8046 ± 0.0256 (0.8540)
mediana	0.8726 ± 0.0491 (0.9990)
grande	0.8151 ± 0.0351 (0.9289)
edificio	0.8133 ± 0.0437 (0.9080)
chica fin de semana	1.1784 ± 0.0149 (1.2154)
mediana fin de semana	0.8102 ± 0.0174 (0.8440)
grande fin de semana	0.8478 ± 0.0336 (0.9443)
edificio fin de semana	0.8085 ± 0.0423 (0.9322)

Tabla 6.10: Resultados de las métrica Spread para el algoritmo NSGA-II.

HR	
chica	0.9517 ± 0.0090 (0.9707)
mediana	0.7004 ± 0.0386 (0.95610)
grande	0.7374 ± 0.1008 (0.9675)
edificio	0.7364 ± 0.0885 (0.9699)
chica fin de semana	0.9980 ± 0.0003 (0.9991)
mediana fin de semana	0.9984 ± 0.0006 (0.9992)
grande fin de semana	0.9113 ± 0.0229 (0.9514)
edificio fin de semana	0.6561 ± 0.1110 (0.9581)

Tabla 6.11: Resultados de las métrica HR para el algoritmo NSGA-II.

Los valores de distancia generacional reportados en la tabla 6.9 son sumamente bajos, lo que sugiere una buena convergencia al frente de Pareto real. Los valores de spread alcanzados también son bajos (tabla 6.10), lo que evidencia una buena distribución en las soluciones no dominadas encontradas.

Por ultimo, los valores del hipervolumen reportados en la tabla 6.11 confirman una buena convergencia y diversidad en las soluciones halladas.

Las figuras 6.4 a 6.11, muestran los frentes de Pareto hallados por el MOEA y las soluciones de los algoritmos ávidos. También se muestra la solución correspondiente a no realizar ninguna planificación sobre el consumo del usuario. Los frentes de pareto reportados corresponden a aquellas ejecuciones que alcanzaron el mayor valor de mejora al finalizar la ejecución.

Las figuras 6.4 y 6.5 muestran los frente de Pareto hallado por el MOEA para la instancia chica en un día de semana y en fin de semana. También muestra las soluciones de los algoritmos evolutivos y la solución BaU.

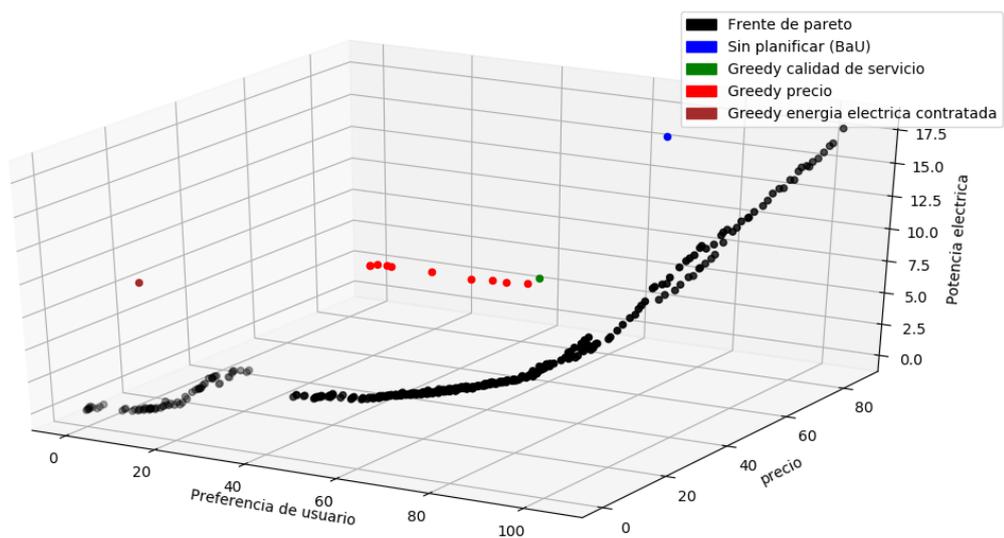


Figura 6.4: Frente de Pareto para la instancia chica en un día de semana

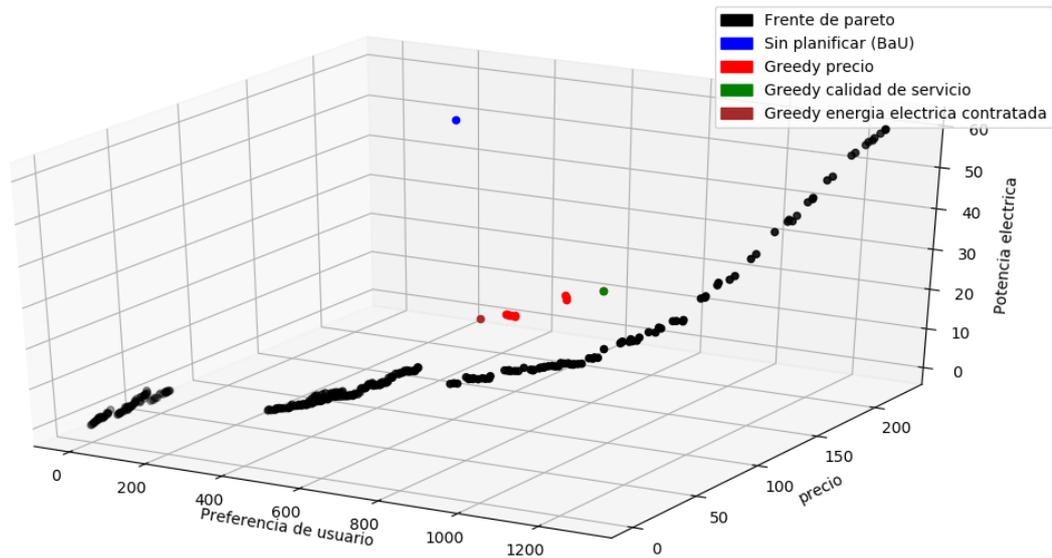


Figura 6.5: Frente de Pareto para la instancia chica fin de semana

Las figuras 6.4 y 6.5 muestran resultados similares. La figura 6.5 presenta puntos superpuestos en el algoritmo greedy precio. Esto se debe a que el algoritmo ávido no encontró una planificación distinta al mejorar la preferencia del usuario. La solución BaU tiene notables diferencias en la preferencia del usuario para día de semana como fin de semana, esto se debe a que la planificación BaU asigna el encendido o apagado del dispositivo eléctrico en base a los datos de encendido y apago del usuario, mientras que la planificación del MOEA busca encender o apagar un dispositivo no solo en base a la preferencia del usuario sino que también a la duración del dispositivo. La ecuación de preferencia del usuario asigna mayor preferencia a las asignaciones que respeten la duración del dispositivo. Por tanto la diferencia en la preferencia de usuario en ambas imágenes para la solución BaU se debe a que en días de semana el usuario realiza cargas de dispositivos eléctricos durante intervalos más cercanos a su duración.

Las figuras 6.6 y 6.7 muestran los frentes de Pareto hallados por el MOEA para la instancia mediana en un día de semana y en fin de semana. También muestra las soluciones de los algoritmos evolutivos y la solución BaU.

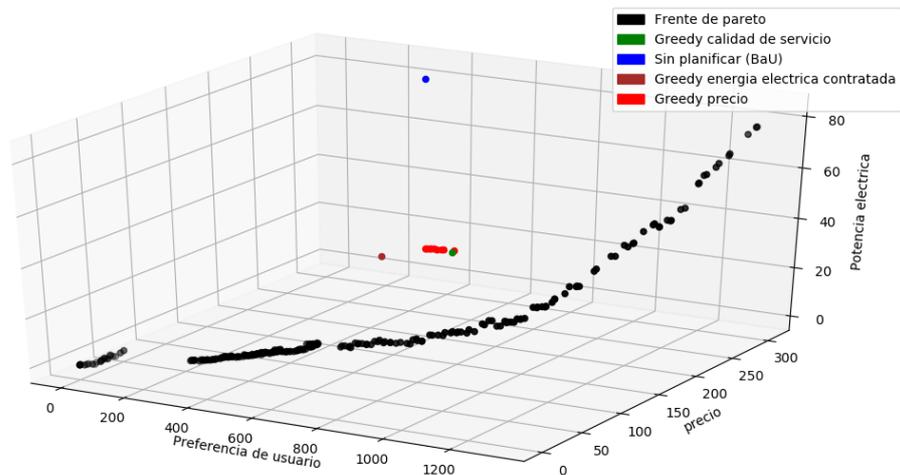


Figura 6.6: Frente de Pareto para la instancia mediana en un día de semana

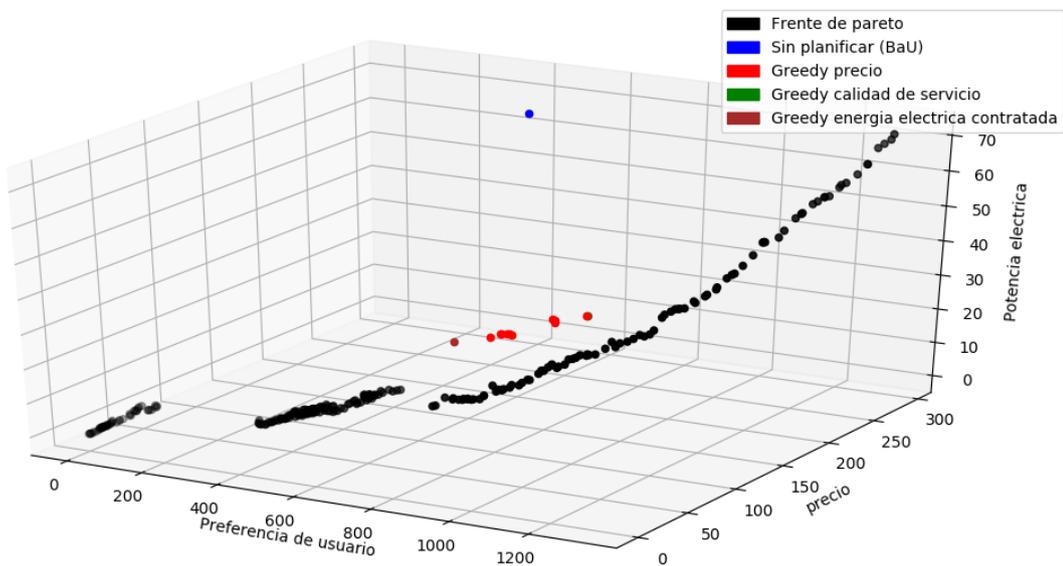


Figura 6.7: Frente de Pareto para la instancia mediana fin de semana

Las figuras 6.6 y 6.7 presentan resultados similares. La figura 6.6 muestra mayor distancia de los puntos del frente de Pareto a los puntos de los algoritmos

ávidos y de la solución BaU, de lo que se deduce mayores mejores. En figura 6.7 se ve que tanto el algoritmo ávido QoS coincide con el algoritmo ávido precio, esto se debe a que uno de los puntos del algoritmo ávido precio considera la preferencia del usuario al 100 % y que en este caso la solución propuesta por el algoritmo QoS es también la de menor precio. Las figuras 6.8 y 6.9 muestran los frente de Pareto hallado por el MOEA para la instancia grande en un día de semana y en fin de semana. También muestra las soluciones de los algoritmos evolutivos y la solución BaU.

Las figuras 6.8 y 6.9 muestran los frente de Pareto hallado por el MOEA para la instancia edificio en un día de semana y en fin de semana. También muestra las soluciones de los algoritmos evolutivos y la solución BaU.

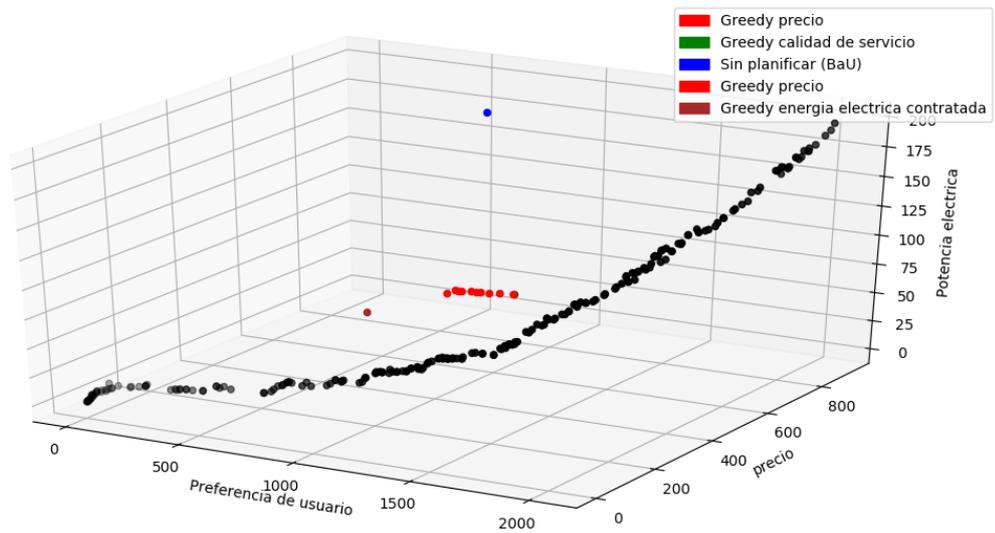


Figura 6.8: Frente de Pareto para la instancia grande en un día de semana

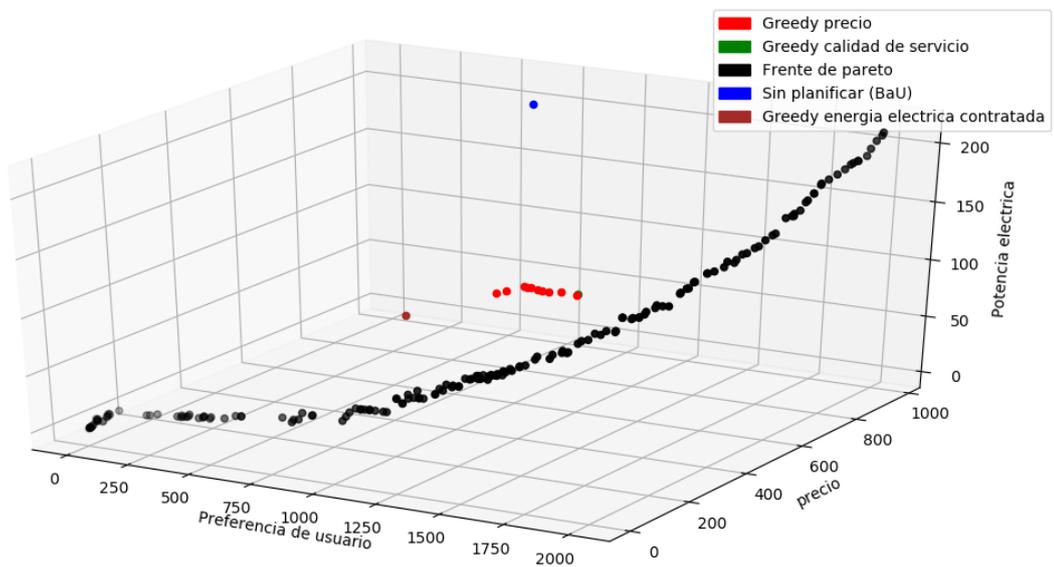


Figura 6.9: Frente de Pareto para la instancia grande fin de semana

Las figuras 6.8 y 6.9 presentan resultados similares tanto en la distancia al frente de pareto como en los puntos de los algoritmos ávidos y la solución BaU.

Las figuras 6.10 y 6.11 muestran los frente de Pareto hallado por el MOEA para la instancia edificio en un día de semana y en fin de semana. También muestra las soluciones de los algoritmos evolutivos y la solución BaU.

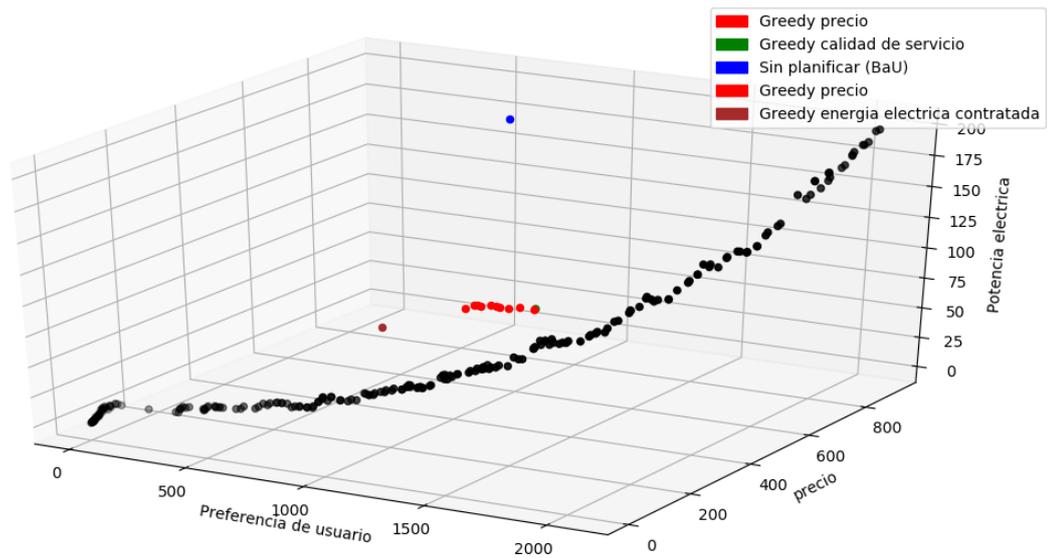


Figura 6.10: Edificio día de semana

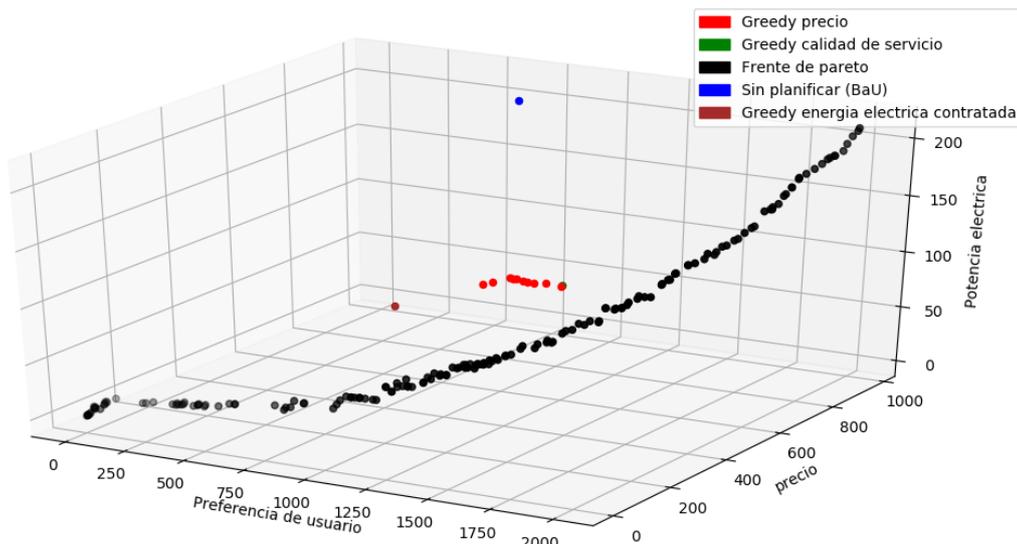


Figura 6.11: Edificio fin de semana

Las figuras 6.10 y 6.11 presentan resultados similares tanto en la distancia al frente de pareto como en los puntos de los algoritmos ávidos y la solución BaU.

Las figuras 6.12, 6.13 y 6.14 detallan los cortes en dos dimensiones para la instancia mediana. Se presenta tanto el frente de Pareto alcanzado, como las soluciones de los algoritmos ávidos y la solución BaU.

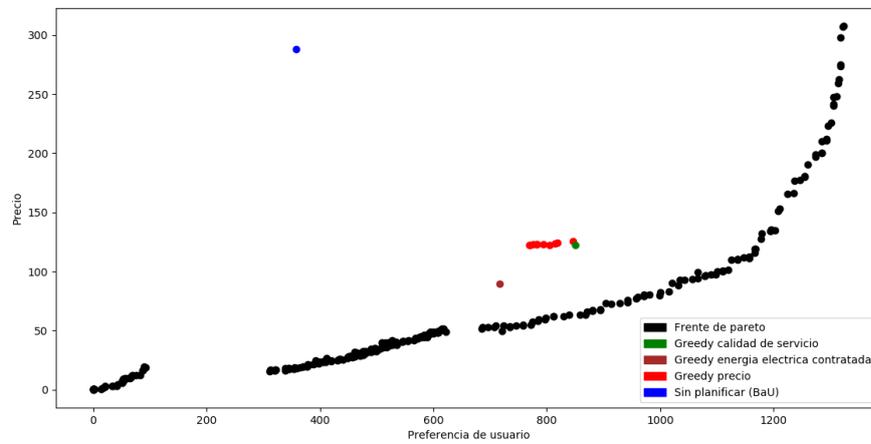


Figura 6.12: Corte instancia mediana, ejes Calidad Servicio, Precio.

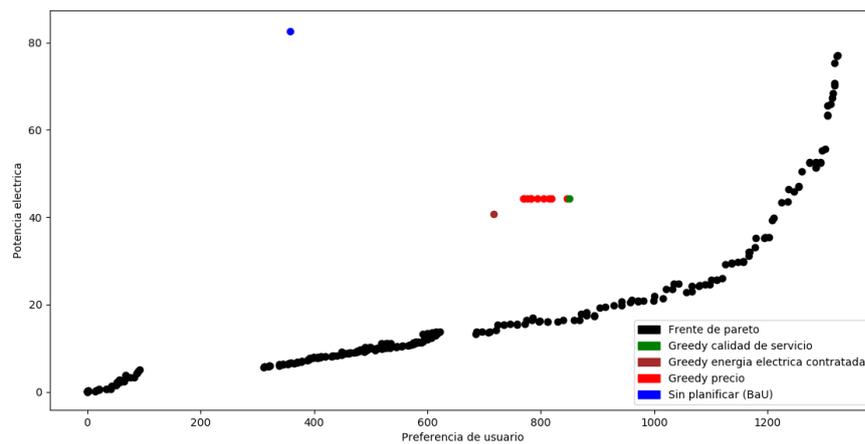


Figura 6.13: Corte instancia mediana, ejes Calidad Servicio, Energía eléctrica contratada.

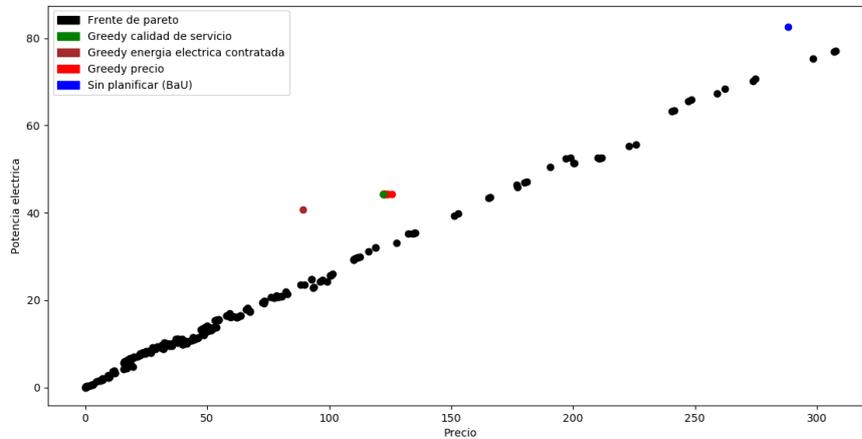


Figura 6.14: Corte instancia mediana, ejes Precio, Energía eléctrica contratada.

Las figuras 6.12 a 6.14 presentan una buena dispersión en el frente de Pareto así como diversidad en las soluciones.

Para la instancia mediana se estudio también las mejoras de la mejor solución de compromiso del algoritmo NSGA-II en comparación a las soluciones ávidas y a la solución BaU, las cuales son presentadas en la tabla 6.12.

Solución	QoS	Precio	Energía eléctrica
ávido precio	31 %	20 %	42 %
ávido QoS	29 %	21 %	41 %
ávido energía eléctrica	54 %	12 %	36 %
BaU	211 %	65 %	69 %

Tabla 6.12: Mejoras alcanzadas por la mejor solución de compromiso del algoritmo NSGA-II frente a los algoritmos propuestos.

La tabla 6.12 muestra notorias mejoras del algoritmo NSGA-II con respecto a la soluciones de los algoritmos ávidos y la solución BaU.

Para la solución encontrada por el algoritmo ávido QoS se busco la solución en el frente de Pareto con QoS mas cercano, para esa solución la tabla 6.13 reporta los porcentajes de mejora de precio y energía eléctrica contratada para cada una de las instancias estudiadas en comparación al algoritmo ávido QoS.

Instancia	Mejora Precio	Mejora energía eléctrica contratada
chica	76.79 %	58.18 %
mediana	49.23 %	63.74 %
grande	46.21 %	38.80 %
edificio	36.46 %	33.40 %
chica fin de semana	49.16 %	36.17 %
mediana fin de semana	55.16 %	35.57 %
grande fin de semana	46.13 %	38.00 %
edificio fin de semana	43.93 %	35.19 %

Tabla 6.13: Mejoras alcanzadas por el MOEA en comparación al algoritmo ávido QoS.

El algoritmo NSGA-II es capaz de alcanzar mejoras en todas las instancias estudiadas sobre las soluciones calculadas por el algoritmo ávido QoS. En el mejor caso mejora el precio en un **76.79 %** (50.38 % promedio) y la energía eléctrica contratada en un **38.12 %** (42.38 % en promedio) manteniendo la calidad de servicio alcanzada por el algoritmo ávido QoS.

La tabla 6.14 reporta los porcentajes de mejora de calidad de servicio y energía eléctrica contratada para cada una de las instancias estudiadas en comparación al algoritmo ávido precio.

Instancia	Mejora QoS	Mejora energía eléctrica contratada
chica	38.72 %	17.21 %
mediana	51.79 %	27.78 %
grande	43.34 %	5.22 %
edificio	51.64 %	3.79 %
chica fin de semana	25.36 %	27.89 %
mediana fin de semana	20.64 %	20.05 %
grande fin de semana	43.37 %	4.62 %
edificio fin de semana	42.61 %	0.21 %

Tabla 6.14: Mejoras alcanzadas por el MOEA en comparación al algoritmo ávido de Precio.

El algoritmo NSGA-II es capaz de alcanzar mejoras en todas las instancias estudiadas sobre las soluciones calculadas por el algoritmo ávido precio. En el mejor caso es capaz de mejorar en un **51.79 %** (39.68 % promedio) la calidad de servicio y en un **27.78 %** (13.34 % promedio) la energía eléctrica contratada manteniendo el mismo precio que el alcanzado por el algoritmo ávido de precio.

La tabla 6.15 reporta los porcentajes de mejora de calidad de servicio y precio para cada una de las instancias estudiadas en comparación al algoritmo ávido energía eléctrica contratada.

Instancia	Mejora QoS	Mejora Precio
chica	52.16 %	29.52 %
mediana	61.05 %	26.98 %
grande	47.27 %	3.37 %
edificio	43.34 %	1.17 %
chica fin de semana	42.03 %	41.68 %
mediana fin de semana	30.66 %	24.12 %
grande fin de semana	49.98 %	8.04 %
edificio fin de semana	42.87 %	3.66 %

Tabla 6.15: Mejoras alcanzadas por el MOEA en comparación al algoritmo ávido de energía eléctrica contratada.

El algoritmo NSGA-II es capaz de alcanzar mejoras en todas las instancias estudiadas sobre las soluciones calculadas por el algoritmo ávido energía eléctrica contratada. En el mejor caso, el precio en un **26.98 %** (17.31 % promedio) y la calidad de servicio en un **61.05 %** (46.17 % en promedio) manteniendo la energía eléctrica contratada alcanzado por el algoritmo ávido de energía eléctrica contratada.

La tabla 6.16 reporta los porcentajes de mejora de precio y energía eléctrica contratada para cada una de las instancias estudiadas en comparación a la solución BaU.

Instancia	Mejora Precio	Mejora energía eléctrica contratada
chica	83.54 %	75.80 %
mediana	93.61 %	91.96 %
grande	91.76 %	87.71 %
edificio	87.67 %	86.05 %
chica fin de semana	90.29 %	87.27 %
mediana fin de semana	90.72 %	87.82 %
grande fin de semana	90.64 %	88.63 %
edificio fin de semana	92.01 %	88.42 %

Tabla 6.16: Mejoras alcanzadas por el MOEA en comparación a la solución sin BaU.

El algoritmo NSGA-II es capaz de alcanzar mejoras en todas las instancias

estudiadas sobre la solución BaU. En el mejor caso mejora el precio en un **93.61 %** (90.02 % promedio) y la energía eléctrica contratada en un **91.96 %** (86.70 % en promedio) manteniendo la calidad de servicio reportada por la solución BaU.

La tabla 6.17 reporta los porcentajes de mejora de calidad de servicio y energía eléctrica contratada para cada una de las instancias estudiadas en comparación a la solución BaU.

Instancia	Mejora QoS	Mejora energía eléctrica contratada
chica	45.39 %	1.32 %
mediana	268.76 %	8.86 %
grande	209.15 %	1.60 %
edificio	215.85 %	2.42 %
chica fin de semana	364.26 %	4.41 %
mediana fin de semana	203.18 %	2.68 %
grande fin de semana	207.83 %	2.66 %
edificio fin de semana	255.82 %	1.84 %

Tabla 6.17: Mejoras alcanzadas por el MOEA en comparación a la solución sin planificar.

El algoritmo NSGA-II es capaz de alcanzar mejoras en todas las instancias estudiadas sobre la solución BaU. En el mejor caso es capaz de mejorar en un **364.26 %** (221.18 % promedio) la calidad de servicio y en un **4.41 %** (3.22 % promedio) la energía eléctrica contratada manteniendo el mismo precio que el reportado por la solución BaU.

La tabla 6.18 reporta los porcentajes de energía eléctrica contratada y precio para cada una de las instancias estudiadas en comparación a la solución BaU.

Instancia	Mejora QoS	Mejora Precio
chica	46.22 %	8.46 %
mediana	270.21 %	6.84 %
grande	208.86 %	4.70 %
edificio	213.17 %	1.48 %
chica fin de semana	363.63 %	3.73 %
mediana fin de semana	206.97 %	8.48 %
grande fin de semana	203.25 %	2.88 %
edificio fin de semana	255.82 %	1.15 %

Tabla 6.18: Mejoras alcanzadas por el MOEA en comparación a la solución sin planificar.

El algoritmo NSGA-II es capaz de alcanzar mejoras en todas las instancias estudiadas sobre la solución BaU. En el mejor caso, el precio en un **3.73 %** (4.715 % promedio) y la calidad de servicio en un **363.63 %** (228.81 % en promedio) manteniendo la energía eléctrica contratada la alcanzado por el algoritmo ávido de energía eléctrica contratada.

Las tablas 6.17 y 6.18 presentan notables mejoras en la calidad de servicio del algoritmo NSGA-II sobre la solución propuesta por BaU. Esto se debe a que la solución propuesta por BaU no respeta el encendido de los dispositivos eléctricos durante el periodo de duración de los mismos, por lo que la ecuación de preferencia del usuario (Ecuación 2.2) penaliza la solución.

Los resultados obtenidos sugieren que el enfoque evolutivo multiobjetivo aporta notables mejoras al planificar el consumo energético del hogar, teniendo en cuenta las preferencias del usuario, el precio de la energía eléctrica y la energía eléctrica contratada por el usuario final. El enfoque sugerido multiobjetivo muestra también diferentes soluciones de compromiso en donde el usuario de acuerdo a sus necesidades particulares es capaz de elegir una planificación que se adapte a él.

Capítulo 7

Conclusiones y trabajo futuro

El presente capítulo presenta las conclusiones del trabajo realizado en este proyecto y las principales líneas de trabajo futuro.

7.1. Conclusiones

En este proyecto se presentan algoritmos evolutivos para la resolución del problema de planificación energética en hogares con el fin de obtener un sistema de planificación de uso eléctrico de los dispositivos eléctricos de los usuarios. Este problema es de gran relevancia tanto para las compañías como para los usuarios finales.

La formulación matemática propuesta considero optimizar la preferencia de realización de carga de los usuarios finales, la energía consumida y el precio de la misma. El algoritmo MOEA considero la optimización de los objetivos definidos mientras que el algoritmo monoobjetivo busco optimizar una función lineal teniendo en cuenta las preferencias del usuario y el costo.

Se utilizo un conjunto de datos realistas al problema para la evaluación experimental de los algoritmos evolutivos propuestos. Los resultados obtenidos fueron comparados con otros métodos de planificación (algoritmos ávidos y BaU).

Los resultados experimentales del AE monoobjetivo mostraron mejoras de hasta un **30 %** sobre al algoritmo ávido. Los resultados también indicaron que los patrones de consumo de los usuarios en los fines de semana son mas difíciles de predecir por el AE ya que las mejoras con respecto al algoritmo ávido se redujeron al **5 %**. Esto ultimo puede deberse a que en fines de semana

los usuarios interactúan continuamente y de forma distinta con los dispositivos eléctricos. Las mejoras del AE con respecto al algoritmo BaU fueron esperables debido a que el algoritmo BaU representa el consumo del usuario sin tener en cuenta ningún tipo de planificación. Además las soluciones propuesta por BaU no respetan el encendido de los dispositivos eléctricos durante el periodo de duración de los mismos, por lo que la ecuación de preferencia del usuario (Ecuación 2.2) penaliza la solución.

Los resultados experimentales del algoritmo NSGA-II mostraron que éste es capaz de alcanzar mejoras sobre las soluciones calculadas por los algoritmos ávidos y por la solución BaU en todas las instancias estudiadas. El algoritmo evolutivo NSGA-II logró mejorar hasta en **76.79 %** y la energía eléctrica contratada en un **38.12 %** manteniendo la calidad de servicio alcanzada por el algoritmo ávido QoS. También logró mejorar hasta en un **51.79 %** la calidad de servicio y en un **27.78 %** la energía eléctrica contratada manteniendo el mismo precio que el alcanzado por el algoritmo ávido de precio. Por ultimo, logró mejorar hasta en un **26.98 %** el precio de la energía y la calidad de servicio hasta un **61.05 %** manteniendo la energía contratada por el algoritmo ávido de energía contratada. Con respecto a la solución BaU, el algoritmo NSGA-II logró mejorar hasta un **93.61 %** el precio y hasta un **91.96 %** la energía eléctrica contratada manteniendo la calidad de servicio reportada por la solución BaU. También logro mejorar hasta un **364.26 %** la calidad de servicio y en un **4.41 %** la energía eléctrica contratada manteniendo el mismo precio que el reportado por la solución BaU. Por ultimo, logró mejorar hasta un **3.73 %** el precio y hasta un **363.63 %** la calidad de servicio manteniendo la energía eléctrica contratada.

En el marco de este proyecto se presentó un artículo científico en el congreso iberoamericano de ciudades inteligentes (ICSC-CITIES 2019), celebrado del 7 al 9 de octubre del 2019 en Soria, España. El artículo titulado "Multiobjective household energy planning using evolutionary algorithms" presenta la versión monoobjetivo del problema de planificación energética en hogares y su evaluación experimental. El artículo se adjunta en el anexo 1.

7.2. Trabajo futuro

A continuación, se detallan las principales líneas de trabajo futuro que surgen a a partir del trabajo realizado en el marco del proyecto.

En primer lugar las preferencias del usuario consideran únicamente el deseo por parte del usuario de encender o apagar un dispositivo en cierto instante de tiempo, sin tener en cuenta el tiempo de consumo de los dispositivos. Esto hace que se reflejen notorias mejoras al comparar los algoritmos evolutivos propuestos contra la planificación BaU debido a la penalización que realiza la ecuación 2.2 al no cumplir con el tiempo mínimo de carga. Por tanto las principales líneas de trabajo futuro con respecto a las preferencias del usuario serían de extender el modelo de forma de proporcionar una mejor visión de los distintos patrones de consumo de los usuarios.

En segundo lugar, aplicar los algoritmos evolutivos a instancias provenientes de datos analizados de casas, barrios y edificios de Uruguay, teniendo también en cuenta otros dispositivos eléctricos del usuario.

En tercer lugar aplicar el modelo de planificación de energía eléctrica a otros ámbitos de planificación como puede ser el gas.

Por último se podría considerar en el modelo, bancos de baterías en los usuarios finales, de donde poder consumir energía sin recurrir al administrador de energía eléctrica y poder recargarlos en intervalos de tiempo donde sea conveniente.

Referencias bibliográficas

- [1] M. Deakin and H. Al Waer. From intelligent to smart cities. *Intelligent Buildings International*, 3(3):140–152, 2011.
- [2] K. Deb. Multi-objective optimization using evolutionary algorithms: An introduction. *Springer*, pages 2–3, 2001.
- [3] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature PPSN VI*, pages 849–858. Springer Berlin Heidelberg, 2000.
- [4] Z. Garroussi and R. Ellaia. An enhanced multiobjective particle swarm optimization algorithm for economic environmental dispatch problem. In *2015 3rd International Renewable and Sustainable Energy Conference*, pages 1–5, 2015.
- [5] D. Goldberg and J. Holland. Genetic algorithms and machine learning. *Machine Learning*, 3(2):95–99, 1988.
- [6] H. T. Haider, O. H. See, and W. Elmenreich. Dynamic residential load scheduling based on adaptive consumption level pricing scheme. *Electric Power Systems Research*, 133:27–35, 2016.
- [7] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.
- [8] J. Kolter and M. Johnson. REDD : A public data set for energy disaggregation research. In *Workshop on Data Mining Applications in Sustainability*, volume 25, pages 59–62, 2011.

- [9] X. H. Li and S. H. Hong. User-expected price-based demand response algorithm for a home-to-grid system. *Energy*, 64:437–449, 2014.
- [10] D. Setlhaolo, X. Xia, and J. Zhang. Optimal scheduling of household appliances for demand response. *Electric Power Systems Research*, 116:24–28, 2014.
- [11] A. Soares, C. H. Antunes, C. Oliveira, and Á. Gomes. A multi-objective genetic approach to domestic load scheduling in an energy management system. *Energy*, 77(C):144–152, 2014.
- [12] A. Soares, Á. Gomes, C. Antunes, and H. Cardoso. Domestic load scheduling using genetic algorithms. In *European Conference on the Applications of Evolutionary Computation*, pages 142–151, 2013.
- [13] A. Soares, Á. Gomes, C. Henggeler Antunes, and H. Cardoso. Domestic load scheduling using genetic algorithms. In A. I. Esparcia-Alcázar, editor, *Applications of Evolutionary Computation*, pages 142–151. Springer Berlin Heidelberg, 2013.
- [14] A. Soares, Álvaro Gomes, and C. H. Antunes. Categorization of residential electricity consumption as a basis for the assessment of the impacts of demand response actions. *Renewable and Sustainable Energy Reviews*, 30:490–503, 2014.
- [15] W. Turner and S. Doty. *Energy management handbook*. The Fairmont Press, 2007.
- [16] U.S. Energy Information Administration (EIA). Energy use in homes. <https://www.eia.gov/>, June 2018.
- [17] UTE. Tarifas inteligentes. Comunicación personal, 2017.
- [18] D. R. White. Software review: the ECJ toolkit. *Genetic Programming and Evolvable Machines*, 13(1):65–67, 2012.

ANEXOS

Anexo 1

Publicaciones

En este anexo se encuentra la publicación "Multiobjective household energy planning using evolutionary algorithms". El artículo fue presentado en el congreso iberoamericano de ciudades inteligentes (ICSC-CITIES 2019), celebrado del 7 al 9 de octubre del 2019 en Soria, España.

Multiobjective household energy planning using evolutionary algorithms

Giovanni Colacurcio¹, Sergio Nesmachnow¹,
Jamal Toutouh², Francisco Luna³, and Diego Rossit⁴

¹ Universidad de la República, Uruguay,
{giovanni.colacurcio,sergion}@fing.edu.uy

² Massachusetts Institute of Technology, USA
toutouh@mit.edu

³ Universidad de Málaga, Spain
flv@lcc.uma.es

⁴ Universidad Nacional del Sur, Argentina
diego.rossit@uns.edu.ar

Abstract. This article presents the advances in the design and implementation of a recommendation system for planning the use of household appliances, focused on improving energy efficiency from the point of view of both energy companies and end-users. The system proposes using historical information and data from sensors to define instances of the planning problem considering user preferences, which in turn are proposed to be solved using a multiobjective evolutionary approach, in order to minimize energy consumption and maximize quality of service offered to users. Promising results are reported on realistic instances of the problem, compared with situations where no intelligent energy planning are used (i.e., ‘*Bussiness as Usual*’ model) and also with a greedy algorithm developed in the framework of the reference project. The proposed evolutionary approach was able to improve up to 29.0% in energy utilization and up to 65.3% in user preferences over the reference methods.

1 Introduction

Energy management is a crucial issue in nowadays societies. Many strategies have been proposed to guarantee an increased access to the energy resources at affordable costs for citizens, while ensuring the preservation of natural resources and the protection of the environment [15].

For implementing effective energy management policies, innovative technologies must be integrated in easy-to-use and efficient systems, which must include specific features to be useful for both energy companies and citizens. In order to be applied by energy companies, management systems must include capabilities for performing realistic simulations, controlling, and planning the electricity market. From the point of view of citizens, system must provide easy-to-use applications to monitor and manage the energy consumption at household level.

The capabilities of monitoring, controlling, and managing the energy consumption and generation are very important to provide good quality of service (QoS) and user experience, especially when considering the emphasis on citizen engagement, environment protection, and economic considerations, under the novel paradigm of smart cities [4].

Residential buildings significantly contribute to the total energy used in the world. According to statistics from the USA Energy Information Administration, the average household in the USA and Canada uses about 12,000 kWh of electricity. In Europe the figure is less than 10,000 kWh, but also significant. Some household appliances make the biggest contributions to consumption, including heaters and air conditioning (40–45%), electronic and kitchen appliances (~30%), water heaters (15%–20%), and cooling (5%–10%) [16]. Energy utilization patterns are not too different in developing countries, where the impact of energy consumption is also very important.

The related literature indicates that systems based on planning the use of *deferrable appliances* allow improving energy efficiency at domestic level [12,14]. Deferrable appliances are those whose demand for energy can be postponed or interrupted (such as dishwashers, washing machines, etc.), causing a negligible impact on the QoS provided to users. This is an important approach that takes into account the different prices of electricity and the availability of (non-storable) energy from renewable sources.

In this line of work, this work presents the application of evolutionary algorithms (EAs) to solve the problem of planning the use of household appliances considering user preferences. A multiobjective approach is considered, aiming at simultaneously maximizing user satisfaction (evaluated in terms of the QoS offered according to the specified preferences) and minimizing the total energy consumed, which is directly related to the total cost of the electricity bill for the user. The main results indicate that the proposed approach is able to find appropriate plannings that improve over situations where no intelligent energy planning are used (i.e., ‘*Business as Usual*’ model) and also with a greedy algorithm previously proposed in the framework of the reference project.

The main contributions of this research are: *i*) formulating a novel multiobjective household energy planning problem accounting for user satisfaction and energy consumption; *ii*) devising a specific EA to address the problem, using a linear aggregation approach for the objectives; and *iii*) evaluating the proposed evolutionary approach over realistic scenarios, built by using real data of household energy consumption from well-known repositories.

The research reported in this article is developed in the context of the Cloud Computing for Smart Energy Management (CC-SEM) project [6,7], which proposes building an integrated platform for smart monitoring, controlling, and planning energy consumption and generation in urban scenarios. The project integrates cutting-edge technologies (Big Data analysis, computational intelligence, Internet of Things, High Performance Computing and Cloud Computing) and specific hardware for energy monitoring/controlling built within the project.

The article is organized as follows. Section 2 presents the formulation of the multiobjective household energy consumption planning and a review of related works. The proposed evolutionary approach for household energy planning is described in Section 3. The experimental analysis is reported in Section 4. Finally, Section 5 presents the conclusions and the main lines of future work.

2 The household energy planning problem

This section introduces the household energy planning problem, the multiobjective formulation addressed in this article, and a review of related works.

2.1 General considerations

The goal of the study is to develop a system to help end-users to take appropriate decisions concerning the use of household appliances in a given planning period (e.g., daily, weekly, etc.). The problem consists in scheduling the use of different household appliances to minimize the electric bill, taking into account the end-user preferences, electricity prices, and the available contracted power.

The planning period is divided in slots considering the user preferences. For every slot, each user can indicate a value that represents the priority of using a certain appliance in that time. Higher values of priority represent a higher desire of using the appliance. In case that users do not indicate their preferences, machine learning is applied to infer preferences from the analysis of historical utilization data. Classification methods can also be applied to characterize the household power consumption, regarding neighboring houses and socio-economical data, such as for other public services [10].

The problem formulation assumes that the energy cost is known for each time interval. These values are publicly available from the energy companies, for example from the National Electricity Company (UTE) in Uruguay. Also, the maximum contracted power for each user is known, from the contract details provided by the energy company. The contracted maximum power can only be surpassed by a small amount (up to 30%) in a short period of time, without causing a short circuit. Schedules that includes such a surplus are penalized.

2.2 Problem formulation

The multiobjective version of the household energy planning problem addressed in this article considers the following elements:

- a set of users $U = (u_1 \dots u_N)$, each user represents a house in a city;
- a set of time slots $T = (t_1 \dots t_M)$ in the planning period;
- a set of domestic appliances $L = (l_1 \dots l_K)$ for each user;
- a function $E : U \rightarrow \mathbb{N}$, where $E(u_i)$ indicates the maximum electric power contracted by user u_i ;
- a penalty term ρ applied to those users that surpass the maximum electric power contracted;

- a function $D : L \times U \rightarrow \mathbb{N}$, where $D(l_k, u_i)$ indicates the average time of utilization of appliance l_k for user u_i ;
- a function $C : T \rightarrow \mathbb{N}$, where $C(t_j)$ indicates the utilization cost (per kW) of the energy in time t_j ;
- a function $P : L \rightarrow \mathbb{N}$, where $P(l_k)$ indicates the power (in kWh) consumed by appliance l_k ;
- and a function $UP : U \times L \times T \rightarrow \mathbb{N}$, where $UP(u_i, l_k, t_j)$ indicates the preference of user u_i to use the appliance l_k at time t_j ;

Lets consider the binary variable x_{kj}^i , that indicates if appliance l_k of user u_i is turn on at time t_j ; and function $y(x_{kj}^i)$ that indicate the time period in which appliance l_k of user u_i is turned on continuously (without intermediate turn off) from time t_j : $y(x_{kj}^i) = m - j$ with $m = \max_{r \in (j, r)} x_{kh}^i = 1$

The problem proposes finding a planning function $X = \{x_{kj}^i\}$ for the use of each household appliance that simultaneously maximizes the user satisfaction defined in Eq. 1 (given the users' preference functions) and minimize the total energy consumed (see Eq. 2).

$$f(X) = \sum_{i=1}^N \sum_{j=M}^T \sum_{k=1}^K UP(u_i, l_k, t_j) \times \delta_{kj}^i \quad (1)$$

$$\text{with } \delta_{kj}^i = \begin{cases} 1 & \text{if } y(x_{kj}^i) \geq D(l_k, u_i) \\ 0 & \text{otherwise} \end{cases}$$

$$g(X) = \sum_{i=1}^N \sum_{j=M}^T \sum_{k=1}^K x_{kj}^i \times C(t_j) \times P(l_k) + \rho \times \psi_{kj}^i \quad (2)$$

$$\text{with } \psi_{kj}^i = \begin{cases} P(l_k) - E(u_i) & \text{if } P(l_k) - E(u_i) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Two scenarios are defined for defining the penalty model used for those situations in which the household consumption exceeds the maximum power contracted. The first scenario (*soft penalty*) is when the user exceeds the maximum power contracted for less than 30% of it. This is the maximum value of energy consumption that can exist without a short circuit occurring. In that case, the solution is penalized by a 30% of ρ . The second scenario (*hard penalty*) is when the user exceeds the maximum power contracted in a value greater than or equal to 30%. Therefore, that plannings are penalized entirely by the penalty term ρ .

Function UP considers the energy consumption measurements of electrical devices reported by Kolter and Johnson [5]. For each minute of the day, in the period of a month, the user preference is defined considering how many times each appliance was turned on for each appliance at that minute.

Function D uses consumption values of user appliances from a representative day. The duration of using for the appliance was studied, defined as the number of consecutive minutes in which it remained powered on [2].

2.3 Related works

The analysis of the related literature allows identifying several hardware- and software-based methods for household energy consumption characterization and planning. The main related works are reviewed next.

The main line of work related to the proposed research has been developed by Soares et al., who studied the household electricity demands and categorized a set of appliances, according to their use and management strategies that can be applied to them [13]. An initial work [14] introduced a model based on integer non-linear programming for energy utilization planning, with the aim of reducing cost. The authors applied an EA to minimize the cost of invoice and violations to the maximum contracted power. The EA allowed to reduce up to 40% the energy cost for the users with respect to a reference scenario without demand management. Later, the authors proposed minimizing cost and maximizing user satisfaction [12], which is the main motivation for the work proposed in our research. Results showed that the cost reduction was 22–24%. However, no trade-off solutions were computed, so different users with equal contracted power and equal preferences should adapt to the same planning. Additionally, no studies were carried out in different urban levels (buildings, neighborhoods, etc.) or used real data.

Our previous work [11] presented a hardware and software platform for intelligent monitoring and planning of energy consumption in homes. The proposed system integrates a hardware controller for energy efficiency, a communication protocol to improve data transmission, and a software module for planning and managing household devices. The proposed solution was implemented applying the Internet of Things (IoT) paradigm, allowing the integration of computational intelligence techniques. A greedy algorithm was proposed for planning, considering user preferences and a maximum allowed power consumption. Results showed that it is possible to reduce the energy consumption of a water heater to 38.9% and that two water heaters and an air conditioner can be optimized simultaneously without reducing QoS. These results suggest that the proposed approach is useful for energy consumption planning in homes.

Bilil et al. proposed a characterization of household appliances and a dynamic planning method for collaborative microgrids [1]. Two multiobjective optimization problems were studied, accounting for the activation and power profiles of appliances. A simulation procedure was applied to generate the instances of these problems and NSGA-II was used to solve them. The instances consisting in 40 microgrids that include a flexible deferrable appliance, such as a water heater, and a non-flexible one (i.e., dishwasher). For the experiments, a residential load curve based on U.S. user profiles was used. The results showed that the load curve can indeed become very flat by applying the proposed bi-level multiobjective optimization scheduling approach.

The analysis of the related works indicates that there is room to contribute with solutions focused on the development of systems to implement the management of domestic demand through the integration of IoT technologies and computational intelligence algorithms.

3 The proposed EA for household energy planning

This section describes the proposed EA to solve the household energy planning problem.

3.1 Evolutionary algorithms

EAs are stochastic techniques that emulate natural evolution to solve optimization, search, and learning problems. They are useful for solving complex real-world problems in multiple application areas [9].

An EA is an iterative technique (each iteration is called *generation*). In each generation, probabilistic operators are applied on a set of individuals (the *population*). The initial population is generated by applying a random procedure or using a specific heuristic for the problem to be solved. Each individual encodes a tentative solution to the problem and has a *fitness* value that determines its suitability to solve the problem. The goal of the EA is to improve the fitness of individuals in the population. In order to achieve this objective, *evolutionary operators* are applied iteratively, such as the *recombination* of parts of two individuals and the random *mutation* of an individual's coding. These operators are applied to individuals selected according to their fitness, thus guiding the EA toward tentative solutions of higher quality that replace old individuals. The stop criterion usually involves a fixed number of generations, a quality level on the fitness of the best individual, or detecting convergence. The EA returns the best solution found in the iterative process, taking into account the fitness function considered for the problem. Algorithm 1 presents the generic schema of an EA with a population P .

Algorithm 1 Schema of an evolutionary algorithm.

```

1: initialize( $P(0)$ )
2:  $t \leftarrow 0$  ▷ generation counter
3: while not stop criterion do
4:   evaluate( $P(t)$ ) ▷ evolutionary cycle
5:   parents  $\leftarrow$  selection( $P(t)$ )
6:   children  $\leftarrow$  variation operators(parents)
7:   newpop  $\leftarrow$  replacement(children,  $P(t)$ )
8:    $t++$ 
9:    $P(t) \leftarrow$  newpop
10: end while
11: return best individual found ▷ best fitness value

```

In this article, a traditional EA using a linear aggregation approach is used to solve household energy planning problem. Although the aggregation approach is often outperformed by Pareto-based methods for multiobjective optimization, it is a common approach in the literature, because of two main advantages: i) it is suitable for multiobjective optimization problems with a convex Pareto front, and ii) it is computationally efficient, so it is recommended when the times available to perform the planning is short [3].

3.2 The proposed EA for household appliances planning

The main features of the proposed EA for household appliances planning are described next.

Solution encoding. A problem-specific encoding is used to represent solutions. The proposed encoding considers for each user a vector $X = (x_0, x_1, \dots, x_T)$, where T is the total number of timesteps (i.e., minutes) in the planning period. Each element x_j in the encoding is a vector of binary values $x_j = (b_1, b_2, \dots, b_L)$, where L is the number of appliances considered in the planning and each value b_i indicates if the appliance is on on timestep j .

Fig. 1 presents an example of solution encoding for an instance of the problem considering five appliances. In the example, at timestep (minute) i , appliances #1, #2, and #5 are ON, while appliances #3 and #4 are OFF.

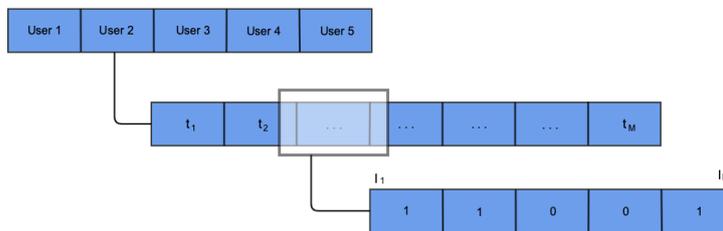


Fig. 1: An example of the proposed solution encoding.

Fitness assignment. The fitness function of the proposed EA corresponds to an linear aggregation of the power consumption and user satisfaction functions: $F = \alpha \times f(X) - \beta \times g(X)$. Several combinations of weights (α , β) were studied in order to properly weight each objective function and provide a useful search pattern. The main results of the analysis are reported in Section 4.3. The combination that allowed computing the best results was ($\alpha = 0.65$, $\beta = 1$).

Initialization. The population of tentative solutions is initialized by applying a randomized method that assigns to each appliance a probability $\gamma = 0.6$ (value tuned in preliminary experiments) to be ON at each time step, following a discrete non-uniform distribution. Assigning a slightly larger probability to each appliance to be ON than to be OFF allows starting the evolutionary search for a more diverse set of solutions. The value of γ was set to provide an equal pressure to both objectives, considering the weights defined in the previous paragraph.

Selection. The standard tournament selection was applied in the proposed EA. Preliminary experiments demonstrated that tournament selection provides an appropriate selection pressure to guide the search. After a preliminary configuration analysis, the size of the tournament was set to two individuals, and the best of them is selected.

Evolutionary operators. Ad-hoc evolutionary operators were conceived to provide efficacy and diversity to the search, working with the proposed solution encoding. The proposed evolutionary operators are:

- Recombination.** An ad-hoc version of the Single Point Crossover operator was conceived to recombine solutions. A cutting point is selected for each user and a new planning is created for each user, using information from the first parent (before the cutting point) and from the second parent (after the cutting point). Fig. 2 presents an example of the application of the proposed recombination operator between two solution for a problem instance with three users and six appliances for each user.

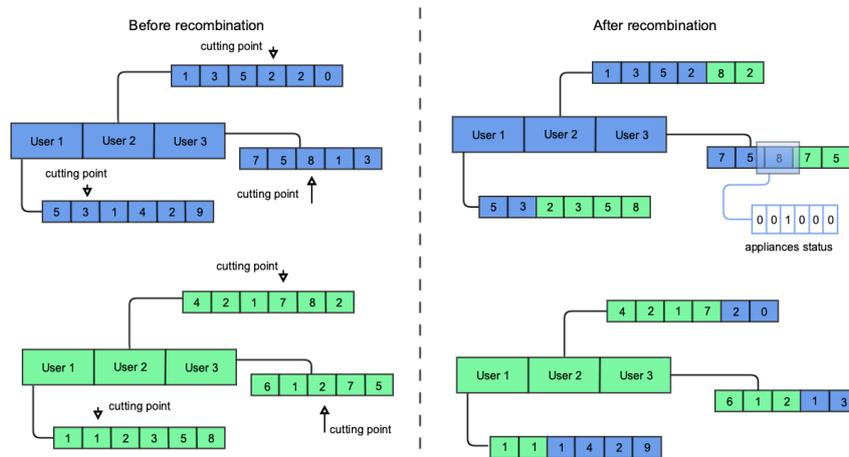


Fig. 2: An example of the proposed recombination operator.

Mutation. The mutation operator modifies the current state of an appliance. First, a specific time interval is randomly selected for every user, according to a uniform distribution. An appliance is then randomly selected (applying a uniform distribution) from all belonging to that user, and its state is changed (on/off or viceversa). Fig. 3 presents an example of the mutation operator.

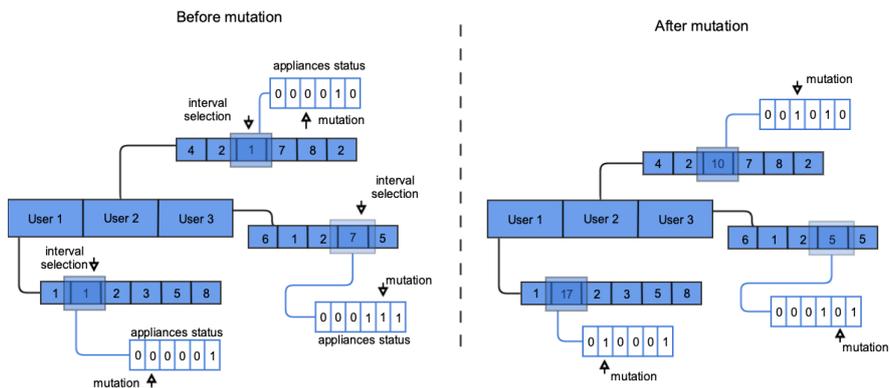


Fig. 3: An example of the proposed mutation operator.

3.3 Development and execution platform

The evolutionary approach was implemented using the ECJ library, a Java-based evolutionary computation system developed at George Mason University (cs.gmu.edu/~eclab/projects/ecj). ECJ includes easily modifiable classes for solving optimization problems. The experimental evaluation was performed on a Dell Power Edge server, Quad-core Xeon E5430 processor at 2.66GHz, 8 GB RAM, from Cluster FING, Universidad de la República, Uruguay [8].

4 Experimental analysis

This section presents the experimental analysis of the proposed EA for household energy consumption planning.

4.1 Problem instances

A set of problem instances was build using real data.

Raw energy consumption data for appliances were obtained from the REDD dataset [5], sampled every three seconds for a period of one month. Appliances included in the problem instances were selected considering: i) the categorization of household appliances according to their operating profiles and purposes [14]; ii) the average and maximum time of use of each appliance; and iii) several other parameters, including the number of activations and energy consumption measurements, the number of houses where each appliance is present, the frequency of activation, etc. Six appliances were selected: dishwasher, microwave, dryer, air conditioning, oven, and refrigerator. The planning period is one day.

Energy consumption data in the REDD dataset was averaged over intervals of 15 minutes, and the resulting values were analyzed to define the preferences of using each appliance for each user. The overall energy consumption data were studied for each day. One representative weekday and one representative weekend day was determined for each appliance, in order to define the daily power consumption in each case. Real energy prices from the National Electricity Company (UTE) in Uruguay (<https://portal.ute.com.uy>), considering an average residential plan (contracted power of 3.7 kWh) were used.

Six problem instances were generated (Table 1) accounting for different number of users, appliances, and consumption patterns (weekday/weekend).

Table 1: Proposed problem instances

#	<i>name</i>	<i>users</i>	<i>appliances in the instance</i>	<i>consumption pattern</i>
1	small.1 (s1)	2	(2,3)	weekday
2	small.2 (s2)	2	(2,3)	weekend
3	medium.1 (m1)	4	(4,4,3,2)	weekday
4	medium.2 (m2)	4	(4,4,3,2)	weekend
5	large.1 (l1)	6	(5,5,4,4,3,2)	weekday
6	large.2 (l2)	6	(5,5,4,4,3,2)	weekend

4.2 Baseline algorithms for results comparison

Two baseline strategies were implemented for evaluating the results of the proposed EA for household appliances planning: a greedy algorithm and a Business-as-Usual (BaU) planning strategy, which are described next.

Greedy planning strategy. Greedy algorithms iteratively build solutions based on a taking optimal local decisions in each step. A greedy algorithm from the literature [11] was adapted for baseline comparison. The proposed strategy searches the best time intervals to switch on each appliance d_k , according to the user satisfaction and cost, considering the linear aggregation fitness function using $\alpha = 0.65$, $\beta = 1$ (Algorithm 2).

Algorithm 2 Greedy algorithm for household appliances planning

```

procedure INTERVALMAXPREFCOST(initMin,ui,d,X)
  prefCost  $\leftarrow$  0; duration  $\leftarrow$  0
  for (m=initMin; m < tM; m++) do
    if duration < D(d, ui) then
      if  $\sum_{k=1}^K x_{km}^i \times P(d_k) + P(d) < E(u_i)$  then
        prefCost  $\leftarrow$  prefCost +  $\alpha \times UP(u, d, m) - \beta \times C(m)$ 
        duration  $\leftarrow$  duration + (tm+1 - tm)
      else
        prefCost  $\leftarrow$  0
        duration  $\leftarrow$  0
      end if
    else
      return [m, prefCost] ▷ interval found
    end if
  end for
  return [m, prefCost] ▷ no interval found
end procedure

X  $\leftarrow$   $\vec{0}$ 
for (i = 1; i ≤ N; i++) do ▷ for each user
  for (k=1; k ≤ K; k++) do ▷ for each appliance
    prefCost  $\leftarrow$  0; bestPrefCost  $\leftarrow$  -1; bestmin  $\leftarrow$  0 ▷ search best interval
    for (m=t1; m < tM - D(dk, ui); m++) do
      [min, prefCost] = IntervalMaxPrefCost(m, dk, ui, X)
      if prefCost > bestPrefCost then
        bestPrefCost  $\leftarrow$  prefCost
        bestmin  $\leftarrow$  min
      end if
    end for
    for (m=bestmin - D(dk, ui); m ≤ bestmin; m++) do
      xkmi  $\leftarrow$  1 ▷ set appliance ON
    end for
  end for
end for

```

BaU planning strategy. The BaU strategy proposes assigning ON times to each appliance without planning. These plannings have good user preference values, but suboptimal cost values.

4.3 Linear aggregation fitness function

The analysis considered the candidate values $\alpha \in \{0.3, 0.65, 0.75, 1.0\}$ and $\beta \in \{0.3, 0.5, 1.0\}$. The EA was executed over three medium-size instances of the problem (two, four, and six devices). Table 2 reports the mean and interquartile range (IQR) of the best fitness value computed in 30 independent executions of the proposed EA for the three instances solved, using the studied configurations.

Table 2: Best fitness values computed using different values of (α, β)

α	β	instance #1		instance #2		instance #3	
		median	IQR	median	IQR	median	IQR
0.3	0.3	15.81	1.09	179.83	0.55	132.35	6.68
0.3	0.5	11.02	1.54	167.94	2.03	110.10	4.54
0.3	1.0	4.27	6.39	128.55	28.66	54.82	13.56
0.5	0.3	34.06	0.81	311.32	28.66	253.95	5.11
0.5	0.5	26.75	1.06	298.80	23.71	224.47	12.21
0.5	1.0	13.53	10.28	269.83	26.03	175.11	40.39
0.65	0.3	47.28	2.08	409.91	1.76	342.66	6.57
0.65	0.5	39.87	0.83	398.45	1.34	316.32	8.40
0.65	1.0	25.48	6.12	369.21	3.91	257.55	39.03
0.75	0.3	54.85	2.79	475.64	0.39	405.86	10.03
0.75	0.5	48.52	3.20	462.74	35.73	373.95	4.11
0.75	1.0	30.18	3.43	433.82	3.23	304.31	21.64
1.0	0.3	81.21	4.41	640.62	0.59	548.20	5.64
1.0	0.5	71.46	4.55	628.42	2.76	524.55	9.10
1.0	1.0	52.05	2.89	597.33	0.24	454.48	20.50

Fig. 4 presents a trade-off analysis of solutions computed using different values of (α, β) for instance #3 (results are representative of those obtained for other instances). The combination $(0.65, 1.0)$ allows computing the best trade-off solutions regarding user satisfaction and total energy/price.

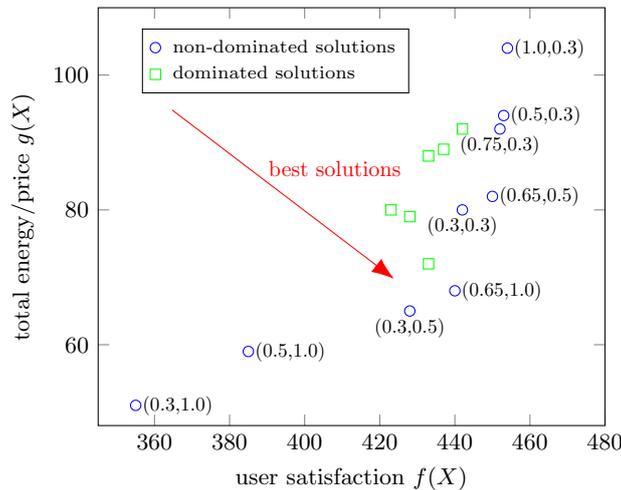


Fig. 4: Trade-off analysis of solutions computed using different values of (α, β)

4.4 Parametric configuration analysis

EA parameters must be adjusted to determine the configuration that allows computing the best results. The analysis was performed over three problem instances, different from those used in the evaluation to avoid bias. After an initial evaluation, the population size was fixed at 150 individuals.

Three relevant parameters of the proposed EA were studied: number of generations used as stopping criterion (G), recombination probability p_C and mutation probability p_M . Candidate values for each parameter were: $p_C \in \{0.1, 0.25, 0.5\}$; $p_M \in \{0.1, 0.05, 0.01\}$; and $G \in \{2500, 5000, 10000\}$. All combinations of parameter values were studied by performing 50 independent executions of the proposed EA for the three problem instances considered in the analysis. The metric considered in the analysis was the linear aggregation fitness function defined in the previous subsection.

The methodology for selecting the best configuration included: i) the Shapiro-Wilk statistical test was applied to check normality, taken as a null hypothesis that the results followed a normal distribution; as p-values less than 0.05 were obtained, the null hypothesis was discarded and it was assumed that the fitness results follow a non-normal distribution; ii) the Friedman's rank test was applied, taken as a null hypothesis that the fitness distributions for the different configurations were not different, as p-values less than 0.05 were obtained, the null hypothesis was discarded and the results significantly differ from each other.

Table 3 reports the fitness values computed in the parameter setting experiments for a representative problem instance. Overall, the best results (i.e., largest fitness median and also lower IQR) were obtained using configuration #11 (values $G = 10000$, $p_C = 0.1$, and $p_M = 0.1$). Henceforth, these values were used in the validation experiments of the proposed EA.

Table 3: Parameter setting results for the proposed EA

<i>configuration</i> (G, p_C, p_M)	<i>fitness</i>		<i>configuration</i> (G, p_C, p_M)	<i>fitness</i>	
	<i>median</i>	<i>IQR</i>		<i>median</i>	<i>IQR</i>
(2500, 0.1, 0.01)	7.88	10.45	(10000, 0.5, 0.1)	40.27	7.61
(5000, 0.1, 0.01)	38.74	11.71	(2500, 0.25, 0.1)	40.73	7.01
(10000, 0.1, 0.01)	40.04	6.98	(5000, 0.25, 0.1)	41.64	6.76
(2500, 0.5, 0.01)	9.26	11.17	(10000, 0.25, 0.1)	40.72	8.67
(5000, 0.5, 0.01)	39.22	9.37	(2500, 0.1, 0.05)	39.60	6.97
(10000, 0.5, 0.01)	39.28	9.88	(5000, 0.1, 0.05)	40.22	9.23
(2500, 0.25, 0.01)	9.06	9.22	(10000, 0.1, 0.05)	40.58	7.95
(5000, 0.25, 0.01)	39.39	10.18	(2500, 0.5, 0.05)	38.66	7.39
(10000, 0.25, 0.01)	38.69	11.76	(5000, 0.5, 0.05)	39.59	10.07
(2500, 0.1, 0.1)	41.77	8.88	(10000, 0.5, 0.05)	40.79	11.62
(5000, 0.1, 0.1)	41.30	8.46	(2500, 0.25, 0.05)	38.17	10.32
(10000, 0.1, 0.1)	42.57	6.46	(5000, 0.25, 0.05)	41.67	8.38
(2500, 0.5, 0.1)	40.62	7.22	(10000, 0.25, 0.05)	41.00	8.44
(5000, 0.5, 0.1)	41.28	8.65			

4.5 Experimental results

Table 4 reports the median of the best fitness computed by the EA and the comparison with the reference algorithms (greedy and BaU). The relative improvement on fitness values (Δ_f) and on each objective function (Δ_{cost} , Δ_{pref}) over each reference algorithm refA is computed as $\Delta = (f(\text{EA}) - f(\text{refA})) / f(\text{refA})$.

Table 4: Experimental results: fitness values and improvements of the proposed EA over the baseline greedy algorithm and the BaU strategy.

<i>instance</i>	$f(\text{EA})$	<i>greedy</i>				<i>BaU</i>			
		f	Δ_f	Δ_{cost}	Δ_{pref}	f	Δ_f	Δ_{cost}	Δ_{pref}
<i>weekday</i>									
small.1	14.1	11.0	28.3%	22.6%	56.2%	-7.6	284.6%	81.1%	16.0%
medium.1	340.0	239.4	42.0%	27.2%	65.3%	69.5	388.9%	77.9%	19.4%
large.1	407.8	347.9	17.2%	20.8%	47.3%	-187.1	317.0%	70.6%	22.0%
<i>weekend</i>									
small.2	323.7	252.1	28.4%	25.1%	44.9%	67.4	383.1%	76.6%	25.8%
medium.2	253.4	197.2	28.5%	29.0%	48.1%	153.8	64.7%	60.6%	20.8%
large.2	369.8	351.9	5.1%	19.7%	37.4%	-299.9	224.2%	72.2%	31.6%

Results in Table 4 indicate that the proposed EA is able to improve significantly over the greedy algorithm regarding the fitness values.

Considering the baseline results computed by the proposed greedy algorithm, improvements of up to 42.0% were obtained in instance medium.1. Results also suggest that consumption patterns during the weekend are harder to plan for the EA, as the improvements over the greedy algorithm reduced to 5.1% in instance large.2. This can be explained due to the interactive utilization of household appliances in weekends, when people are at home a significantly larger periods than in weekdays. Regarding the improvements on user satisfaction and cost, the plannings computed by the proposed EA allow reducing more than 20% the electric bill, and preferences improve more than 40% in all the studied scenarios.

The EA computed significantly cheaper plannings than those of BaU, which systematically failed to provide good cost values, indicating that users do not take the correct decisions to turn on home appliances in this regard, and they can benefit of having an automated planning offered by a recommendation system. In addition, preferences on the solutions computed by the EA were 16–31% better than BaU. The obtained improvements over a BaU strategy are consistent with results reported in previous works for a reduced subset of home appliances (air conditioner and water heater) [11].

The obtained results suggest that the proposed evolutionary approach is accurate for computing household energy consumption plannings accounting for both energy costs and user satisfaction at the same time. The proposed approach is a first step towards designing an intelligent recommendation system for end-users.

5 Conclusions and future work

This article presented an evolutionary approach to address the problem of household energy planning, as a first approach to develop an automated recommendation system for end-users. This is a relevant problem for both energy companies and citizens under the novel smart city paradigm.

A mathematical formulation for the problem was proposed, considering the optimization of user preferences and energy consumed/cost. A specific EA was proposed to solve the problem, simultaneously optimizing both criteria using a linear aggregation multiobjective function and ad-hoc evolutionary operators.

A set of six realistic problem instances built using real data were considered in the experimental evaluation of the proposed EA. The analysis compared the EA results with two baseline planning methods (greedy and business-as-usual).

The experimental results showed that the proposed EA is able to compute accurate plannings, accounting for significant improvements on the problem objectives. Regarding the baseline greedy algorithm, improvements of up to 42.0% were obtained in the proposed multiobjective function, accounting for an average reduction of more than 20% in the energy consumption (and thus, on the electric bill) and preferences improved more than 40% in all the studied scenarios. Regarding the BaU strategy, the EA computed significantly cheaper plannings and user preferences improved up to 31%, in line with previous results from our research group.

The obtained results suggest that the proposed evolutionary approach is accurate for computing household energy consumption plannings accounting for both energy costs and user satisfaction at the same time. Overall, the proposed algorithm showed to be effective for addressing the considered optimization problem. The analysis demonstrated that users can significantly benefit of having an automated planning offered by a recommendation system.

The main lines for future work are related to study explicit multiobjective algorithms to solve the problem, in order to compute several trade-off solutions at the same time. The problem formulation can be extended to include the noisy nature of user preferences in order to define an uncertainty optimization problem. In this regard, robust evolutionary approaches should be studied to solve this problem variant. Finally, new real problem instances can be generated, especially using data from the National Electricity Administration (UTE), in Uruguay.

References

1. Bilil, H., Aniba, G., Gharavi, H.: Dynamic appliances scheduling in collaborative microgrids system. *IEEE Transactions on Power Systems* 32(3), 2276–2287 (2016)
2. Chavat, J., Graneri, J., Nesmachnow, S.: Energy disaggregation of household appliances based on pattern consumption similarities. In: *Iberoamerican Congress on Smart Cities* (2019)
3. Coello, C., Van Veldhuizen, D., Lamont, G.: *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic, New York (2002)

4. Deakin, M., Al Waer, H.: From intelligent to smart cities. *Intelligent Buildings International* 3(3), 140–152 (2011)
5. Kolter, J., Johnson, M.: REDD: A public data set for energy disaggregation research. In: *Workshop on Data Mining Applications in Sustainability*. vol. 25, pp. 59–62 (2011)
6. Luján, E., Otero, A., Valenzuela, S., Mocskos, E., Steffemel, L., Nesmachnow, S.: *Cloud Computing for Smart Energy Management (CC-SEM Project)*. In: *Smart Cities, Communications in Computer and Information Science*, vol. 978. Springer, Cham (2019)
7. Luján, E., Otero, A., Valenzuela, S., Mocskos, E., Steffemel, L., Nesmachnow, S.: An integrated platform for smart energy management: the CC-SEM project. *Revista Facultad de Ingeniería* (2019)
8. Nesmachnow, S.: Computación científica de alto desempeño en la Facultad de Ingeniería, Universidad de la República. *Revista de la Asociación de Ingenieros del Uruguay* 61(1), 12–15 (2010)
9. Nesmachnow, S.: An overview of metaheuristics: accurate and efficient methods for optimisation. *International Journal of Metaheuristics*, 3(4), 320–347 (2014)
10. Nesmachnow, S., Baña, S., Massobrio, R.: A distributed platform for big data analysis in smart cities: combining intelligent transportation systems and socioeconomic data for Montevideo, Uruguay. *EAI Endorsed Transactions on Smart Cities* 2(5), 153478 (2017)
11. Orsi, E., Nesmachnow, S.: Smart home energy planning using IoT and the cloud. In: *IEEE URUCON* (2017)
12. Soares, A., Antunes, C., Oliveira, C., Gomes, A.: A multi-objective genetic approach to domestic load scheduling in an energy management system. *Energy* 77(1), 144–152 (2014)
13. Soares, A., Gomes, A., Antunes, C.: Categorization of residential electricity consumption as a basis for the assessment of the impacts of demand response actions. *Renewable and Sustainable Energy Reviews* 30, 490–503 (2014)
14. Soares, A., Gomes, A., Antunes, C., Cardoso, H.: Domestic load scheduling using genetic algorithms. In: *European Conference on the Applications of Evolutionary Computation*. pp. 142–151 (2013)
15. Turner, W., Doty, S.: *Energy management handbook*. The Fairmont Press (2007)
16. U.S. Energy Information Administration (EIA): Energy use in homes, <https://www.eia.gov/>, June 2018