

# Optimización Computacional

## Tarea 2

Diego A. Espinoza Apablaza, Felipe A. Morales Urrutia, and  
Sebastian M. Cabrera Torrejon

Universidad de Valparaiso

### 1. Resumen

Una empresa que se dedica a la venta de inmuebles quiere contratar una campaña publicitaria para su nuevo proyecto de vivienda con 5 tipos de anuncios: televisión tarde, televisión noche, diarios, revistas y radio ( $x_1, x_2, x_3, x_4, x_5$ ), cada uno con su propia valorización, cantidad de potenciales clientes y costo. Su objetivo es hacer rentable dicho proyecto, maximizando la valorización y minimizando los costos. Existe un límite máximo en la cantidad de cada tipo de anuncio, además de límites en cuantas personas se pueden alcanzar al final y cuánto costo se puede pagar por algunos de estos. Para resolver este problema, se modelará como un problema de programación lineal, luego se hizo uso de metodos y tecnicas de optimizacion tales como RSA, Simplex y consistencias. La solución más eficiente obtenida al aplicar estos métodos es: Contratar 0 anuncios de televisión tarde, 0 de televisión noche, 25 de diarios, 0 de revistas y 30 de radio. Resultando  $Z = 0,6571651090342678$ .

### 2. Introducción

ABC es una empresa del rubro de ventas de inmuebles y están en proceso de contratar el servicio de campaña publicitaria, para su nuevo proyecto de vivienda. El requisito que solicita ABC es tener 5 tipos de anuncios en medios de comunicación: televisión (tarde y noche), diarios, revista y radio. En su objetivo de hacer rentable el proyecto, ABC ha recolectado datos acerca del número de potenciales clientes a quienes les podrian ofrecer este nuevo proyecto: 1.000, 2.000, 1.500, 2.500 y 300, respectivamente.

La empresa publicitaria tomó ese estimado de clientes y valorizó el costo para cubrirlo y elaboro una escala del 0 al 100 de valorización dando:

1. anuncio televisión tarde, la valorización es 65pts y un costo de 150[um].
2. anuncio televisión noche, la valorización es 90pts. y un costo de 300[um].
3. anuncio diario, la valorización es 40pts y un costo de 40[um].
4. anuncio revistas, la valorización es 60pts y un costo de 100[um].
5. anuncio radio, la valorización es 20pts y un costo de 10[um].

Existe una limitante en torno a la cantidad máxima de anuncios que son posibles de emitir: 15, 10, 25, 4 y 30, respectivamente. ABC asesorado por una agencia de publicidad, decide utilizar no más de 20 anuncios en la televisión, para alcanzar cuanto mucho 50.000 clientes potenciales, además de no gastar más de 1.800[um] en anuncios en televisión, y si se hacen anuncios en el diario entonces no hacer anuncios en la televisión por la noche.

Este problema se resolverá a través de programación lineal que consiste en 3 puntos fundamentales, variables de decisión, función objetivo y restricciones, todo esto con el fin de hayar el balance entre maxima valorización y minimo costo. La resolución se realizará con el método Reptile Search Algorithm, que es un algoritmo de optimización inspirado por el comportamiento de los cocodrilos cuando cazan.

Principales contribuciones:

1. Se desarrollará una estrategia publicitaria óptima para el nuevo proyecto de vivienda de ABC.
2. Se maximizará la valorización de los anuncios y se minimizará el costo total de la campaña.
3. Se presentará una solución basada en programación lineal y el método Reptile Search Algorithm.

Objetivo principal:

Encontrar la combinación óptima de tipos de anuncios en medios de comunicación que maximice la valorización y minimice el costo total de la campaña publicitaria, dandoles la misma importancia/peso.

Objetivos específicos:

1. Modelar el problema como un problema de optimización lineal.
  - a) Desarrollar los modelos de optimización lineal que representen el problema.
  - b) Definir las funciones objetivo.
  - c) Establecer las restricciones que limitan las soluciones de los modelos.
2. Encontrar las soluciones óptimas para cada modelo utilizando técnicas de optimización lineal.
3. Aplicar técnicas de consistencia de restricciones al modelo de optimización lineal.
4. Normalización del ejercicio para combinar las funciones objetivo.
5. Utilizar el algoritmo RSA para encontrar la solución óptima del modelo.
6. Evaluar y presentar los resultados obtenidos para analizar su eficacia.

El presente informe detalla de manera exhaustiva los aspectos relevantes relacionados con el tema en cuestión. El documento se estructura siguiendo un orden y formato adecuados, que se describen a continuación:

1. **Resumen:** En esta sección se presenta un breve resumen del contenido del informe, destacando los aspectos más relevantes y los resultados obtenidos.
2. **Introducción:** En la sección de introducción se brinda una visión general del tema tratado, se establecen los objetivos del informe y se presenta el contexto en el que se desarrolló el estudio o análisis.
3. **Implementación:** En esta sección se detalla la metodología y el enfoque utilizado para llevar a cabo la implementación del estudio. Se describen las herramientas, técnicas o algoritmos utilizados, así como los pasos seguidos para obtener los resultados.
4. **Pruebas:** La sección de pruebas se dedica a presentar los resultados obtenidos a través de la implementación. Se muestran los datos recopilados, se analizan y se presentan gráficos, tablas u otros elementos visuales que respalden los hallazgos.
5. **Conclusión:** En esta sección se resumen y discuten los resultados obtenidos, se destacan las principales conclusiones del estudio y se brindan recomendaciones o consideraciones adicionales.
6. **Bibliografía:** La última sección del informe está destinada a listar todas las fuentes bibliográficas, referencias o recursos consultados durante el proceso de investigación y redacción del informe.

Este enfoque estructurado y ordenado garantiza la claridad y coherencia del informe, proporcionando una presentación formal y organizada de la información.

### 3. Implementación

Para resolver el problema y encontrar la mejor solución, se ha decidido emplear técnicas como: Inteligencia de enjambre, Simplex, Nodo-Arco Consistencia, Optimización Multiobjetivo y el Algoritmo Reptile Search.

#### Inteligencia de enjambre

La programación lineal es una técnica ampliamente utilizada en la optimización de problemas que involucran la maximización o minimización de una función lineal sujeta a restricciones lineales. Sin embargo, existen situaciones en las que los enfoques matemáticos tradicionales no son aplicables o resultan ineficientes. Funciona como una alternativa para abordar problemas de programación lineal, ofreciendo una metodología dinámica y eficiente para buscar soluciones óptimas. A través de la búsqueda global, la interacción entre las entidades, la adaptabilidad y el aprovechamiento del paralelismo, la inteligencia de enjambre [1] puede proporcionar un enfoque prometedor para maximizar la valorización y minimizar los costos en este escenario específico.

#### Simplex

El método Simplex [2] fue creado para hallar la solución óptima en un problema de programación lineal, es decir, encontrar los valores para las variables de decisión que maximicen o minimicen una función objetivo lineal con restricciones lineales. Para este ejercicio se hará uso de este método para calcular el costo

máximo posible de los anuncios y la valorización máxima posible de la campaña dentro del dominio de las variables de decisión, estos datos serán utilizados para calcular el equilibrio entre la maximización de la valorización y la minimización del costo utilizando el enfoque multiobjetivo, normalizando los modelos para obtener una única función objetivo, para luego utilizar el algoritmo de búsqueda de reptiles.

### **Nodo-Arco Consistencia**

Nodo-Arco Consistencia [3] son técnicas de optimización que se aplican a restricciones unarias y binarias de un modelo. Son utilizadas como etapas de pre-procesamiento donde se identifican y descartan valores para las variables de decisión que no se deben considerar, con esto se logra disminuir la complejidad, inconsistencias y reducir el espacio de búsqueda de soluciones.

### **RSA**

RSA o Reptile Search Algorithm [4], es un algoritmo de optimización basado en el comportamiento de los cocodrilos cuando están cazando. Aplica dos estrategias para resolver el problema.

#### **1. Encircling Phase**

- High Walking: "High walking" se refiere a una estrategia de movimiento en donde el algoritmo explora el espacio de búsqueda con pasos o saltos largos. Esta estrategia es similar a cuando los reptiles levantan el estómago para poder desplazarse rápidamente. Al dar pasos más grandes, el algoritmo puede cubrir el espacio de búsqueda rápidamente, potencialmente alcanzando regiones más prometedoras que contengan soluciones más óptimas que las demás.
- Belly Walking: "Belly walking", por otro lado, es una estrategia de movimiento donde el algoritmo explora el espacio de búsqueda dando pequeños pasos. Esta estrategia está inspirada en reptiles que se desplazan reptando, examinando cuidadosamente su entorno. Al tomar pequeños y precisos movimientos, el algoritmo puede explorar minuciosamente el espacio de búsqueda y así potencialmente encontrar la solución más óptima local.

#### **2. Hunting Phase**

- Hunting Coordination: "Hunting coordination", en RSA involucra la coordinación y sincronización de varios individuos (reptiles) durante la búsqueda. Esta estrategia imita el comportamiento colectivo observado en ciertos reptiles que cazan en grupo de manera coordinada. En el algoritmo, los individuos intercambian información, comparten conocimiento o influyen en las decisiones de movimiento de los demás para mejorar el rendimiento general de la búsqueda. Esta coordinación ayuda

a explorar diversas regiones del espacio de búsqueda y puede llevar al descubrimiento de soluciones mejores.

- Hunting Cooperation: "Hunting cooperation", en RSA involucra el esfuerzo colaborativo de los individuos en el algoritmo para realizar un objetivo en común. Esta estrategia fue inspirada por reptiles que cooperan o colaboran durante la caza para incrementar las posibilidades de éxito. En RSA, los individuos pueden cooperar a través del compartimiento de recursos, combinando sus fuerzas o coordinando sus estrategias de búsqueda. A través del trabajo en equipo, el algoritmo se beneficia de la inteligencia colectiva y sinergia de los individuos, potencialmente conduciendo a una mejor exploración y explotación del espacio de búsqueda.

El algoritmo usa las siguientes fórmulas:

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \cdots & \cdots & x_{i,j} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix} \quad (1)$$

**Fig 1.** La matriz inicial

$$x_{ij} = rand \times (UB - LB) + LB, j = 1, 2, \dots, n$$

**Fig 2.** Se rellena cada posición de la matriz inicial con un random multiplicado por la resta entre la cota superior e inferior de las respectivas variables de decisión

$$x_{(i,j)}(t+1) = \begin{cases} Best_j(t) \times -\eta_{(i,j)}(t) \times \beta - R_{(i,j)}(t) \times rand, & t \leq \frac{T}{4} \\ Best_j(t) \times x_{(r,j)} \times ES(t) \times rand, & t \leq 2\frac{T}{4} \text{ and } t > \frac{T}{4} \end{cases} \quad (3)$$

**Fig 3.** Encircling phase (exploration): y high walking and belly walking

$$\eta_{(i,j)} = Best_j(t) \times P_{(i,j)}, \quad (4)$$

**Fig 4.** Este valor es utilizado en el algoritmo para guiar la búsqueda y exploración de soluciones óptimas en un espacio de búsqueda.

$$R_{(i,j)} = \frac{Best_j(t) - x_{(r_{2,j})}}{Best_j(t) + \epsilon}, \quad (5)$$

**Fig 5.** Es un valor ocupado para reducir el área de búsqueda y utiliza al mejor como área de interés.

$$ES(t) = 2 \times r_3 \times \left(1 - \frac{1}{T}\right), \quad (6)$$

**Fig 6.** Evolutionary Sense: como dice su nombre lo indica esta ecuación permite evaluar el progreso y desempeño del algoritmo.

$$P_{(i,j)} = \alpha + \frac{x_{(i,j)} - M(x_i)}{Best_j(t) \times (UB_{(j)} - LB_{(j)}) + \epsilon}, \quad (7)$$

**Fig 7.** Es la diferencia en porcentaje de la solución actual y la mejor solución, a esto se le suma alpha para explorar más allá de la mejor solución.

$$M(x_i) = \frac{1}{n} \sum_{j=1}^n x_{(i,j)}, \quad (8)$$

**Fig 8.** Promedio de la distancia de las soluciones

$$x_{(i,j)}(t+1) = \begin{cases} Best_j(t) \times P_{(i,j)}(t) \times rand, & t \leq 3\frac{T}{4} \text{ and } t > 2\frac{T}{4} \\ Best_j(t) - \eta_{(i,j)}(t) \times \epsilon - R_{(i,j)}(t) \times rand, & t \leq T \text{ and } t > 3\frac{T}{4} \end{cases} \quad (9)$$

**Fig 9.** Hunting phase (exploitation): hunting coordination and hunting cooperation

Como se puede apreciar, todos los métodos incorporados en RSA son utilizados para obtener una solución óptima de un problema complejo de forma eficiente, por lo tanto se escogio por su utilización para encontrar la mejor solución al problema.

El codigo es publico y se puede acceder a el desde:  
<https://github.com/zhonnner/Optimizacion-RSA>

## 4. Solucion

Se presentan 2 casos para el problema planteado, donde  $x_2 = 0$  y  $x_3 \leq 25$  o  $x_2 \leq 10$  y  $x_3 = 0$ . Por esto se debe obtener el mejor valor para las funciones de maximización utilizando estas combinaciones. La mejor solución para las 2 funciones es cuando  $x_2 = 0$  y  $x_3 \leq 25$ . Utilizando el método **SIMPLEX** se obtienen los siguientes resultados.

Maximizar  $p = 65x_1 + 90x_2 + 40x_3 + 60x_4 + 20x_5$  sujeto a

$x_1 \leq 15$   
 $x_2 \leq 0$   
 $x_3 \leq 25$   
 $x_4 \leq 4$   
 $x_5 \leq 30$   
 $1000x_1 + 2000x_2 + 1500x_3 + 2500x_4 + 3000x_5 \leq 50000$   
 $x_1 + x_2 \leq 20$   
 $150x_1 + 300x_2 \leq 1800$   
entera  $x_1, x_2, x_3, x_4, x_5$

Solution:

Solución óptima:  $p = 2140$ ;  $x_1 = 12$ ,  $x_2 = 0$ ,  $x_3 = 19$ ,  $x_4 = 0$ ,  $x_5 = 30$

Resolver Ejemplos Borrar todo

☒ Ocultar tablas (más rápido).  
☐ Mostrar tablas.  
☐ Mostrar tablas y soluciones intermedias.

Modo de tablas: ☒ decimal  
☐ fracción  
☐ integral

Redondeando: 6 cifras significativas

Fig 10. Simplex 1, Max

Maximizar  $p = 150x_1 + 300x_2 + 40x_3 + 100x_4 + 10x_5$  sujeto a

$x_1 \leq 15$   
 $x_2 \leq 0$   
 $x_3 \leq 25$   
 $x_4 \leq 4$   
 $x_5 \leq 30$   
 $1000x_1 + 2000x_2 + 1500x_3 + 2500x_4 + 300x_5 \leq 50000$   
 $x_1 + x_2 \leq 20$   
 $150x_1 + 300x_2 \leq 1800$   
 entera  $x_1, x_2, x_3, x_4, x_5$

Solution:

Solución óptima:  $p = 3000$ ;  $x_1 = 12$ ,  $x_2 = 0$ ,  $x_3 = 13$ ,  $x_4 = 4$ ,  $x_5 = 28$

☒ Ocultar tablas (más rápido).  
☐ Mostrar tablas.  
☐ Mostrar tablas y soluciones intermedias.

Modo de tablas: ☒ decimal  
☐ fracción  
☐ integral

Redondeando:  cifras significativas

Fig 11. Simplex 2, Max(Min)

Con esta información se puede **NORMALIZAR** el ejercicio para obtener una única función objetivo. En este caso se escogió el mismo peso para ambas funciones, para obtener el equilibrio entre estas.

$$f_{\min} = 150x_1 + 300x_2(1 - y) + 40x_3(y) + 100x_4 + 10x_5$$

$$f_{\max} = 65x_1 + 90x_2(1 - y) + 40x_3(y) + 60x_4 + 20x_5$$

$$f_{\text{obj}} = \left( \frac{f_{\max}}{2140} \right) \cdot 0,5 + \left( \frac{3000 - f_{\min}}{3000 - 0} \right) \cdot 0,5$$

Además, tenemos las funciones que se encuentran dentro de nuestra función objetivo y las restricciones que son la base para nodo-consistencia y arco-consistencia.

$$\max z = 65x_1 + 90x_2(1 - y) + 40x_3y + 60x_4 + 20x_5$$

$$\min z = 150x_1 + 300x_2(1 - y) + 40x_3y + 100x_4 + 10x_5$$

$$x_1 \leq 15$$

$$x_2 \leq 10(1 - y)$$

$$x_3 \leq 25y$$

$$x_4 \leq 4$$

$$x_5 \leq 30$$

$$y \leq 1$$



$$1000x_1 + 2000x_2 + 1500x_3 + 2500x_4 + 300x_5 \leq 50000$$

$$x_1 + x_2 \leq 20$$

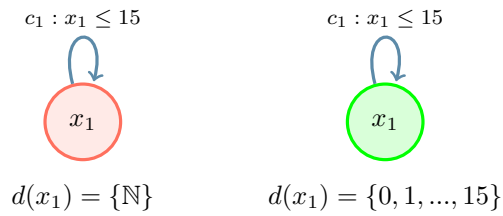
$$150x_1 + 300x_2 \leq 1800$$

Antes de aplicar el algoritmo RSA se aplicará Nodo-Arco consistencia.

### Nodo-consistencia

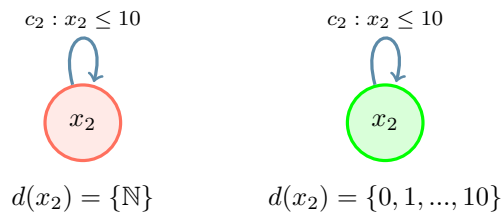
$x_1$  :

**Antes de aplicar — Después de aplicar**



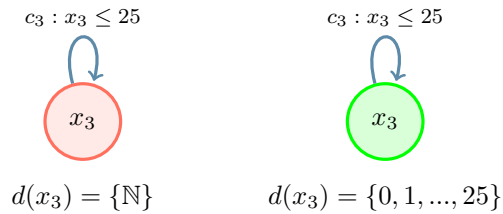
$x_2$  :

**Antes de aplicar — Después de aplicar**



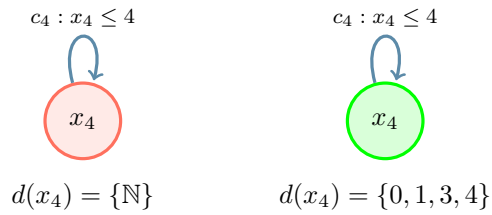
$x_3$  :

**Antes de aplicar — Después de aplicar**



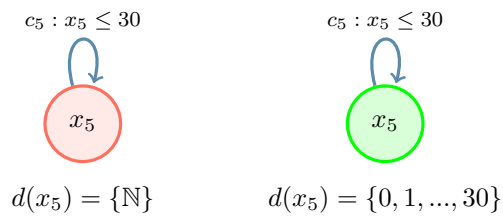
$x_4$  :

**Antes de aplicar — Después de aplicar**



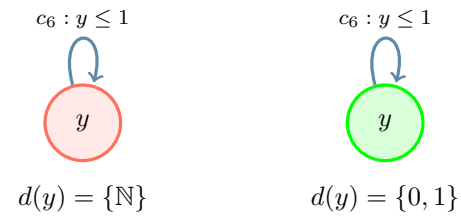
$x_5$  :

**Antes de aplicar — Después de aplicar**

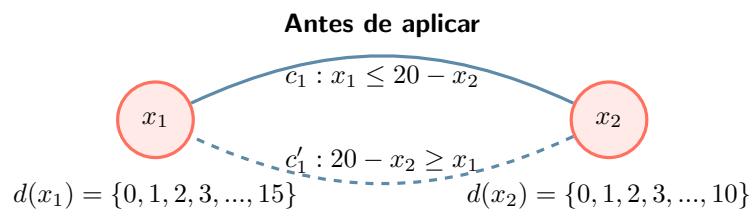


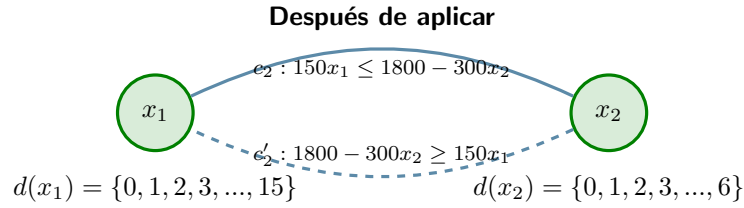
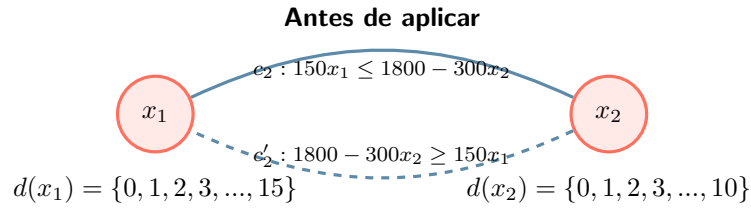
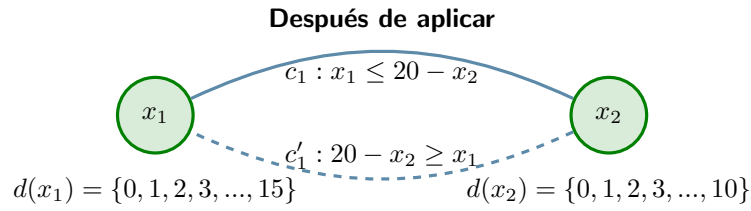
$y$  :

**Antes de aplicar — Después de aplicar**



**arco-consistencia:**





## 5. Pruebas

**Luego de aplicar RSA**  
 Tabla de resumen descriptivo.

variables	mejor	peor	promedio	mediana	desviación estándar	rango inter-cuartílico
$x_1$	0	0	0	0	0	0
$x_2$	0	0	0	0	0	0
$x_3$	25	25	25	25	0	0
$x_4$	0	0	0	0	0	0
$x_5$	30	30	30	30	0	0
$y$	1	1	1	1	0	0
$Z \approx$	0,657	0,657	0,657	0,657		

**Cuadro 1.** Realizado con 30 ejecuciones del algoritmo RSA.

```
PS C:\Users\baamz\OneDrive\Documents\Codigos> python3 .\test.py
0 0.6571651090342678
1 0.6571651090342678
2 0.6571651090342678
3 0.6571651090342678
4 0.6571651090342678
5 0.6571651090342678
6 0.6571651090342678
7 0.6571651090342678
8 0.6571651090342678
9 0.6571651090342678
10 0.6571651090342678
11 0.6571651090342678
12 0.6571651090342678
13 0.6571651090342678
14 0.6571651090342678
15 0.6571651090342678
16 0.6571651090342678
17 0.6571651090342678
18 0.6571651090342678
19 0.6571651090342678
20 0.6571651090342678
21 0.6571651090342678
22 0.6571651090342678
23 0.6571651090342678
24 0.6571651090342678
25 0.6571651090342678
26 0.6571651090342678
27 0.6571651090342678
28 0.6571651090342678
29 0.6571651090342678
PS C:\Users\baamz\OneDrive\Documents\Codigos> |
```

Fig 12. RSA en el tiempo

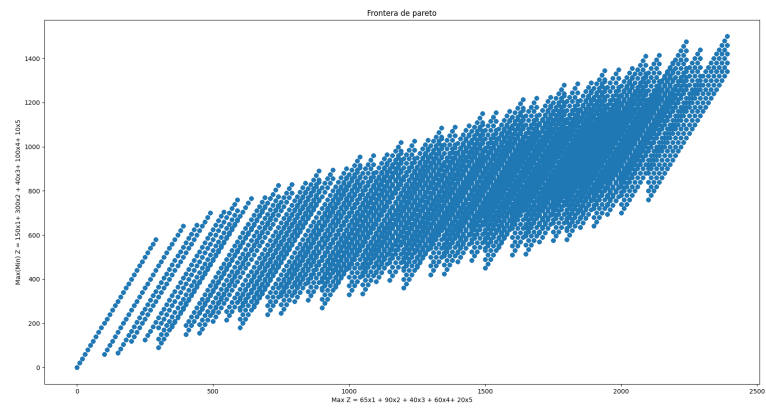


Fig 13. Soluciones factibles

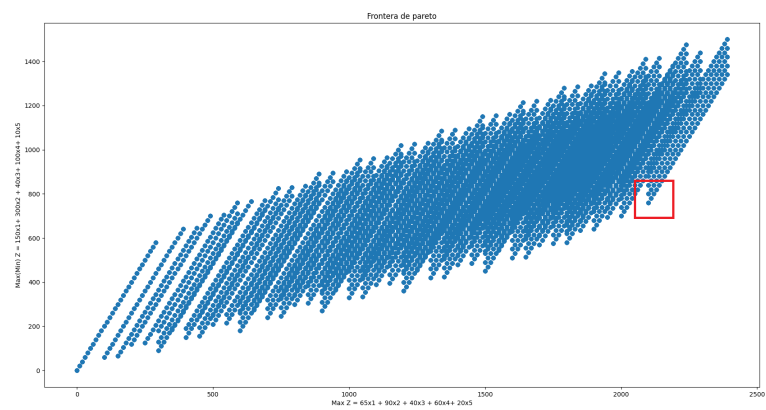
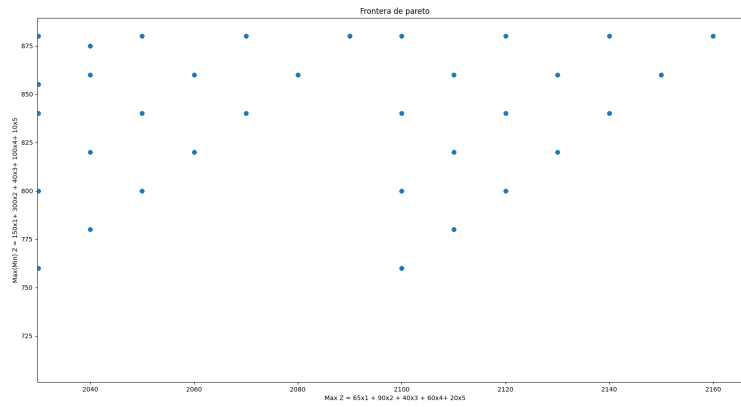
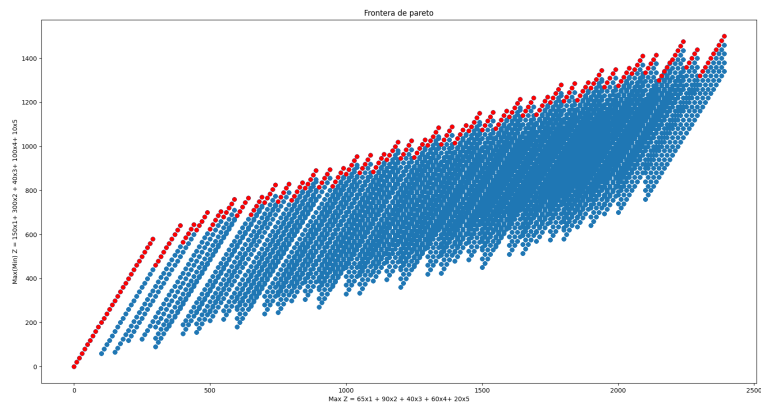


Fig 14. Analizando soluciones factibles



**Fig 15.** Soluciones factibles dominadas

Como ambas funciones son de maximización, la frontera de pareto va a estar por encima de la gráfica, donde esté la mayor maximización para el costo y la mayor maximización para el puntaje.



**Fig 16.** Frontera de pareto

A continuación, en la figura 1, el gráfico de dispersión caja y bigote.

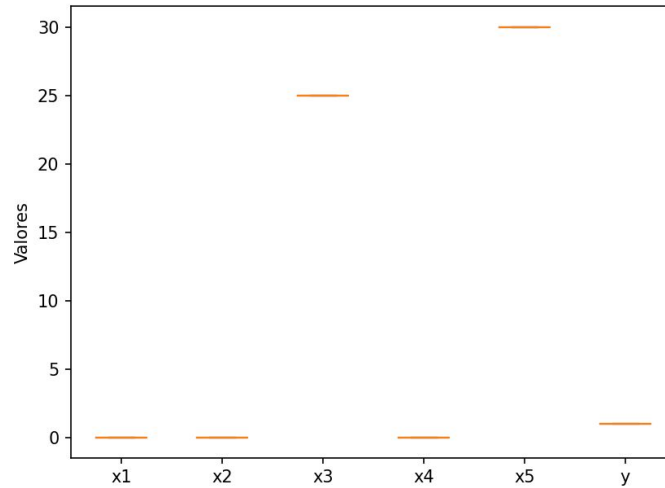


Fig 17. Gráfico de caja bigote. Dispersión de variables de decisión

Al no tener desviación estándar o rango intercuartílico el gráfico de caja y bigote termina solamente con bigotes.

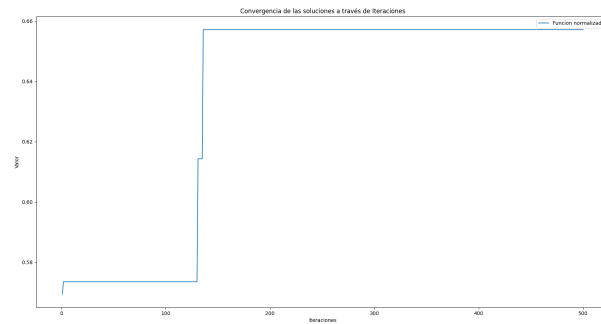


Fig 18. Convergencia de las soluciones a través del tiempo

## 6. Conclusiones

Para resolver este problema, utilizamos las tecnologías presentadas anteriormente en conjunto y así poder encontrar la solución paso a paso.

Para empezar, este problema multiobjetivo lo convertimos en un problema de programación lineal calculando por separado cada función objetivo: una de minimización de costos y otra de maximización de valorización. Esto lo vimos en la sección de pruebas junto con la consistencia nodo-arco, obteniendo todas las restricciones y cumpliendo así el primer y tercer objetivo específico.

Ambas funciones comparten el valor mínimo que pueden tomar, ya que el dominio no incluye números negativos, pero llega hasta el 0. Por lo tanto, si reemplazamos todas las variables de decisión con este número, para este problema obtenemos  $z = 0$ .

Para encontrar el valor máximo que pueden tomar, utilizamos el simplex mostrado en la sección anterior, obteniendo así los resultados de la maximización de ambas funciones, cumpliendo el segundo objetivo específico.

Cada función objetivo tiene una escala diferente. La función  $\max z$  tiene un rango de  $[0, 2140]$  y la función  $\min z$  tiene un rango de  $[0, 3000]$ . Esto genera una función multiobjetivo más inclinada hacia la minimización, lo que se traduce en una mayor importancia al costo de la campaña. Para evitar este desequilibrio, normalizamos cada función de acuerdo a lo descrito en la sección anterior, y ahora ambas tendrán el mismo peso de 0.5 de un total de 1, cumpliendo así el cuarto objetivo específico.

Con todas nuestras preparaciones hechas, utilizamos RSA para resolver nuestra función objetivo de la forma más eficiente que podamos encontrar. Comenzamos por añadir las restricciones y las funciones objetivo, y luego, utilizando las dos estrategias descritas en la implementación, obtenemos el resultado más eficiente encontrado en las 500 iteraciones que ocupa el algoritmo:  $Z = 0,6571651090342678$ . Cumpliendo así el quinto objetivo específico y el objetivo principal.

Finalmente, utilizamos la frontera de Pareto, tabla de resumen descriptivo, gráfico de convergencia y gráfico de divergencia para analizar los resultados, cumpliendo el sexto y último objetivo específico.

Nuestra solución indica que la empresa ABC no debe comprar anuncios televisivos o en revistas si su prioridad es un balance valorización/costo, ya que no son los usos más eficientes de sus recursos. En cambio, los anuncios en diarios y radios tienen la mejor relación valorización/costo y, por lo tanto, la mejor inversión. Con esta información, ABC está mejor posicionada para tomar una decisión informada y rentable que le ayude a crecer como empresa, teniendo un impacto positivo en su desarrollo y abriendo posibilidades para más estudios que creen un círculo virtuoso.

Además, con nuestra solución es posible evaluar más tipos de publicidad y ex-



plorar nuevas formas de llegar a potenciales clientes. Por otro lado, podemos investigar y aumentar la precisión de nuestro modelo, teniendo en cuenta más variables y otros factores relevantes, como por ejemplo la demografía de cada anuncio o si una hora específica hace los anuncios en televisión más atractivos para nuestra solución.

Finalmente, es claro que esta solución tiene mucho potencial para la empresa ABC, ya que puede ser el inicio de muchas investigaciones que le darán información a la empresa para poder tomar decisiones más rentables. También es posible mejorar el modelo actual con más información y así obtener resultados con mayor precisión. En ambos casos, esta investigación es solo el primer paso en un proceso más amplio sobre problemas multi-objetivos en optimización computacional.

## 7. Bibliografía

### Referencias

1. R. O. Órdenes, “OPTIMIZACIÓN COMPUTACIONAL: Inteligencia de Enjambre,” Profesor, Escuela de Ingeniería Informática, Universidad de Valparaíso, Chile, Mayo 2023.
2. —, “OPTIMIZACIÓN COMPUTACIONAL: Programación Lineal,” Profesor, Escuela de Ingeniería Informática, Universidad de Valparaíso, Chile, Abril 2023.
3. —, “OPTIMIZACIÓN COMPUTACIONAL: Programación con Restricciones,” Profesor, Escuela de Ingeniería Informática, Universidad de Valparaíso, Chile, Mayo 2023.
4. A. Laith, A. E. Mohamed, S. Putra, W. G. Zong, and H. G. Amir, “Reptile search algorithm (rsa): A nature-inspired meta-heuristic optimizer,” *www.elsevier.com*, p. 116158, 2021.