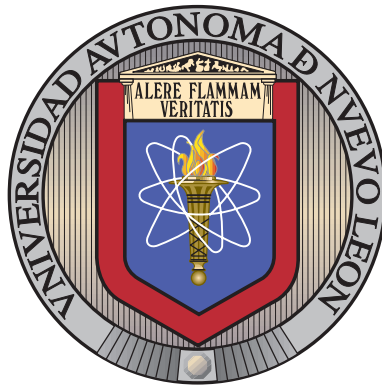


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS



UN PROBLEMA BI-NIVEL DE UBICACIÓN DE PUNTOS DE
VACUNACIÓN QUE CONSIDERA EL DISTANCIAMIENTO
SOCIAL Y LA EQUIDAD PARA LOS HABITANTES

POR

EDITH SALINAS DE LEÓN

EN OPCIÓN AL GRADO DE

DOCTOR EN CIENCIAS

CON ORIENTACIÓN EN MATEMÁTICAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

ENERO 2024

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS



UN PROBLEMA BI-NIVEL DE UBICACIÓN DE PUNTOS DE
VACUNACIÓN QUE CONSIDERA EL DISTANCIAMIENTO
SOCIAL Y LA EQUIDAD PARA LOS HABITANTES

POR

EDITH SALINAS DE LEÓN

EN OPCIÓN AL GRADO DE

DOCTOR EN CIENCIAS

CON ORIENTACIÓN EN MATEMÁTICAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

ENERO 2024

Universidad Autónoma de Nuevo León
Facultad de Ciencias Físico Matemáticas

Los miembros del Comité de Tesis recomendamos que la Tesis “Un problema bi-nivel de ubicación de puntos de vacunación que considera el distanciamiento social y la equidad para los habitantes”, realizada por la alumna Edith Salinas de León, con número de matrícula 1451084, sea aceptada para su defensa como opción al grado de Doctor en Ciencias con Orientación en Matemáticas.

El Comité de Tesis

Dr. José Fernando Camacho Vallejo
Director

Dr. Omar Jorge Ibarra Rojas
Revisor

Dr. Alvaro Eduardo Cordero Frnaco
Revisor

Dr. Samuel Moisés Nucamendi Guillén
Revisor

Dra. Dámaris Arizhay Dávila Soria
Revisor

Vo. Bo.

Dr. Omar Jorge Ibarra Rojas
Coordinar del Posgrado en Ciencias con
Orientación en Matemáticas

San Nicolás de los Garza, Nuevo León, Enero 2024

DECLARACIÓN DE AUTENTICIDAD

Declaro solemnemente que el documento que enseguida presento es fruto de mi propio trabajo, y hasta donde estoy enterada, no contiene material previamente publicado o escrito por alguien más, excepto aquellos materiales o ideas que por ser de otras personas les he dado el debido reconocimiento y los he citado cumplidamente en la bibliografía o las referencias. Declaro además que tampoco contiene material que haya sido aceptado para el otorgamiento de cualquier otro grado o diploma de alguna universidad o institución.

Edith Salinas de León

A mis pacientes padres, que han hecho posible todo lo que he conseguido.

A mis queridos hermanos, que siempre están presentes.

A mi esposo, que es el mejor apoyo que pueda tener.

ÍNDICE GENERAL

| | |
|---|-------------|
| Agradecimientos | XII |
| Resumen | XIII |
| 1. Introducción | 1 |
| 1.1. Objetivo | 3 |
| 1.2. Motivación | 3 |
| 1.3. Metodología | 4 |
| 1.4. Estructura | 5 |
| 2. Revisión de Literatura | 6 |
| 2.1. Localización de instalaciones | 6 |
| 2.2. Programación Bi-nivel | 11 |
| 2.3. Programación Multi-objetivo | 14 |
| 2.4. Aplicaciones relacionadas con el COVID-19 | 23 |
| 2.5. Resumen de la revisión de la literatura | 26 |
| 3. Descripción del problema | 28 |
| 4. Algoritmo bi-objetivo anidado de Entropía Cruzada | 34 |
| 4.1. Metaheurísticas | 34 |
| 4.1.1. Metaheurísticas Constructivas | 35 |

| | |
|--|-----------|
| 4.1.2. Metaheurísticas de Búsqueda | 37 |
| 4.1.3. Metaheurísticas Evolutivas | 38 |
| 4.1.4. Metaheurísticas de Relajación | 40 |
| 4.2. Algoritmo de Entropía Cruzada | 41 |
| 4.3. Algoritmo propuesto | 43 |
| 4.3.1. Descripción del BONXEA | 45 |
| 5. Experimentación computacional | 49 |
| 6. Conclusiones | 71 |
| 6.1. Líneas de investigación futuras | 74 |

ÍNDICE DE FIGURAS

| | | |
|-----|--|----|
| 1. | Gráficas ilustrativas en programación Bi-nivel lineal | 15 |
| 2. | Gráficas ilustrativas de los espacios de decisión y de los objetivos | 16 |
| 3. | Gráficas ilustrativas de la Frontera de Pareto | 17 |
| 4. | Gráficas ilustrativas del punto ideal, nadir y utópico | 18 |
| 5. | Ejemplo Agente Viajero | 36 |
| 6. | Fases Metaheurística GRASP | 37 |
| 7. | Algoritmo Genético | 39 |
| 8. | Cruza | 39 |
| 9. | Mutación | 40 |
| 10. | Instancia 30-100-6-50 | 52 |
| 11. | Instancia 30-50-6-100 | 53 |
| 12. | Instancia 30-100-6-100 | 53 |
| 13. | Instancia 30-100-10-50 | 53 |
| 14. | Instancia 30-100-10-100 | 54 |
| 15. | Instancia 50-50-6-10 | 54 |
| 16. | Instancia 50-100-6-5 | 54 |
| 17. | Instancia 50-100-10-100 | 55 |
| 18. | Instancia 30-50-6-10 | 55 |

| | | |
|-----|----------------------------------|----|
| 19. | Instancia 30-100-6-10 | 55 |
| 20. | Instancia 30-100-10-10 | 56 |
| 21. | Instancia 50-50-10-10 | 56 |
| 22. | Instancia 50-50-10-50 | 56 |
| 23. | Instancia 50-100-3-10 | 57 |
| 24. | Instancia 50-100-10-50 | 57 |
| 25. | Instancia 20-20-3-1 | 57 |
| 26. | Instancia 20-20-6-10 | 58 |
| 27. | Instancia 30-30-4-10 | 58 |
| 28. | Instancia 30-50-6-50 | 58 |
| 29. | Instancia 30-50-10-50 | 59 |
| 30. | Instancia 30-50-10-100 | 59 |
| 31. | Instancia 30-100-3-50 | 59 |
| 32. | Instancia 50-100-6-100 | 60 |
| 33. | Instancia 30-100-3-100 | 60 |
| 34. | Instancia 50-100-3-50 | 60 |
| 35. | Instancia 20-20-3-10 | 61 |
| 36. | Instancia 20-20-6-1 | 61 |
| 37. | Instancia 20-20-6-50 | 61 |
| 38. | Instancia 30-30-3-1 | 62 |
| 39. | Instancia 30-30-3-10 | 62 |
| 40. | Instancia 30-30-3-100 | 62 |
| 41. | Instancia 30-30-4-100 | 63 |
| 42. | Instancia 30-50-3-50 | 63 |
| 43. | Instancia 50-50-3-50 | 63 |

| | | |
|-----|---|----|
| 44. | Instancia 100-100-3-1 | 64 |
| 45. | Instancia 100-100-3-50 | 64 |
| 46. | Instancia 50-50-10-100 | 64 |
| 47. | Instancia 20-20-3-50 | 65 |
| 48. | Instancia 30-50-3-10 | 65 |
| 49. | Instancia 30-50-3-100 | 65 |
| 50. | Instancia 30-50-10-10 | 66 |
| 51. | Instancia 50-50-3-10 | 66 |
| 52. | Instancia 50-100-3-100 | 66 |
| 53. | Instancia 100-100-3-10 | 67 |
| 54. | Instancia 50-50-3-100 | 67 |
| 55. | Convergencia de BONXEA con respecto a F_1 | 69 |
| 56. | Convergencia de BONXEA con respecto a F_2 | 70 |

ÍNDICE DE TABLAS

| | | |
|----|--|----|
| 1. | Métodos de escalarización | 22 |
| 2. | Métodos de no escalarización | 22 |
| 3. | Algoritmos evolutivos multiobjetivo | 23 |
| 4. | Resumen de las características consideradas en los problemas de localización de instalaciones estrechamente relacionados | 27 |
| 5. | Resumen de las soluciones obtenidas por el BONXEA | 51 |

AGRADECIMIENTOS

Agradezco primeramente a Dios por haberme dado la oportunidad de llegar hasta este momento de mi vida.

Agradezco el apoyo, dedicación y paciencia del Dr. José Fernando Camacho Vallejo quien me dirigió en el desarrollo de esta tesis.

Gracias a todos los maestros que me dieron clases, por compartir sus conocimientos y permitirme crecer profesionalmente. También agradezco a mis compañeros, que siempre me apoyaron a lo largo del doctorado.

Agradezco infinitamente a mi familia y esposo, que me dieron motivación y apoyo en todo momento.

Agradezco en especial a mi padre por siempre estar junto a mi, apoyarme en este camino y siempre impulsar el gusto por las matemáticas.

Finalmente agradezco a la Universidad Autónoma de Nuevo León y a la Facultad de Ciencias Físico Matemáticas por brindarme los conocimientos y herramientas necesarias para formarme profesionalmente.

Así mismo agradezco a CONACYT por darme la oportunidad de continuar estudiando el doctorado y por el apoyo que me brindó con la beca durante el doctorado.

RESUMEN

Diseñar programas de vacunación eficientes que consideren las necesidades de la población es muy relevante para prevenir la reaparición de la pandemia de COVID-19. El Gobierno necesita habilitar puntos de vacunación para aplicar dosis de vacunas a la población. En este trabajo se analizan la ubicación de los puntos de vacunación, pero atendiendo a las preferencias de los habitantes. Se consideran dos objetivos que eviten la aglomeración de habitantes. El Gobierno pretende maximizar la distancia mínima entre los puntos de vacunación ubicados; además, apunta a que el número de habitantes que acuden a los diferentes puntos de vacunación sea equitativo. Uno de los aspectos clave de este problema es la suposición de que los habitantes eligen libremente el punto de vacunación al que acudir. Esa decisión afecta a los objetivos del Gobierno ya que pueden aparecer aglomeraciones en los puntos de vacunación por decisiones de los habitantes. Este problema se modela como un programa bi-objetivo bi-nivel, en el que el nivel superior está asociado al Gobierno y el nivel inferior a los habitantes. Para aproximar el frente de Pareto de este problema, se propone una metaheurística de Entropía Cruzada. El algoritmo incorpora criterios para manejar dos funciones objetivo de manera simultánea y resolver de manera óptima el problema de nivel inferior para cada decisión del Gobierno. El algoritmo propuesto se prueba en un conjunto adaptado de instancias de referencia y se incluye el análisis pertinente de los resultados. Una idea gerencial importante es que ubicar puntos de vacunación alejados no nos lleva a una distribución más equitativa de los habitantes.

Palabras clave: programación bi-nivel; COVID-19; ubicación de instalaciones públicas; sistema de vacunación; metaheurística de Entropía Cruzada.

CAPÍTULO 1

INTRODUCCIÓN

En diciembre de 2019, la Organización Mundial de la Salud dio a conocer sobre la existencia de un nuevo virus: el SARS-CoV-2, causante de la enfermedad COVID-19. Algunos de sus síntomas comunes son fiebre, tos seca y cansancio. Las medidas para prevenir la propagación de COVID-19 incluyen el distanciamiento físico de otras personas, mascarillas faciales, lavarse las manos periódicamente y cubrirse la boca con el codo flexionado o con un pañuelo al toser. Hasta el 11 de septiembre de 2022, se han notificado 605 millones de casos confirmados y 6.4 millones de muertes en todo el mundo.

Sin embargo, la vacuna contra este virus ya está disponible. La vacunación de los habitantes parece ser la forma más segura y eficaz de controlar la pandemia de COVID-19. En Duijzer et al. (2018b), afirman que una sola dosis de vacuna puede ser una gota en el océano, pero varias dosis juntas pueden salvar a una población.

Durante lo peor de la pandemia y hasta el día de hoy, ha habido una gran aceptación de la vacuna COVID-19 y una fuerte demanda de la misma. Esta es la razón principal por la que el diseño de programas de vacunación apropiados es de suma importancia. Contar con programas de vacunación adecuados a las necesidades de la población a vacunar es un elemento fundamental para evitar el rebrote de esta pandemia.

Nuestra investigación está orientada a la logística humanitaria, analizando casos donde el gobierno necesita ubicar diferentes puntos de vacunación para los habitantes de una determinada región, de tal manera que exista el acceso más equitativo a las vacunas, ya que la vacunación de los habitantes parece ser la forma más segura y eficaz de controlar la pandemia de COVID-19.

En nuestro trabajo se estudia un problema relacionado con las decisiones estratégicas de vacunación. El Gobierno necesita ubicar diferentes puntos de vacunación. Al tratar de mantener la *sana distancia* y evitar las aglomeraciones, se persiguen dos objetivos:

- (i) que los puntos de vacunación ubicados estén lo más alejadas posible entre sí, y
- (ii) que los habitantes estén repartidos equitativamente (lo más posible) dentro de los puntos de vacunación abiertos.

La estrategia de vacunación impone algunas restricciones de movilidad, como que solo los habitantes actuales del municipio recibirán la vacuna en los puntos de vacunación de ese municipio específico. Sin embargo, el Gobierno no puede imponer la restricción a un habitante de a cual punto de vacunación asistir, cada habitante es libre de asistir al punto de vacunación que prefiera. Este supuesto nos lleva a tener en cuenta las preferencias de los habitantes hacia los puntos de vacunación. Por lo tanto, la asignación de habitantes a los puntos de vacunación la elige cada habitante y no el Gobierno.

En resumen, el Gobierno decide la ubicación de los puntos de vacunación dentro del municipio y los habitantes asisten a los puntos de vacunación abiertos que prefieran. Es evidente que las decisiones de los habitantes inciden directamente en el hacinamiento en los puntos de vacunación, lo que interesa al Gobierno. Por lo tanto, debido a la jerarquía existente en esta situación y la relación entre las decisiones, un enfoque de programación bi-nivel es el adecuado para estudiar este problema. Se consideran dos objetivos en el nivel superior, simultáneamente, y esto impide aplicar de manera eficiente la conocida reformulación de un solo nivel. Es decir, a pesar de que el modelo bi-nivel puede reformularse como uno equivalente de un solo nivel utilizando las condiciones de optimalidad del nivel inferior, el análisis no debe perder de vista que se están optimizando dos funciones objetivo al mismo tiempo en el nivel superior. Por lo tanto, se debería resolver un modelo bi-objetivo muy complejo de un solo nivel.

Para manejar la complejidad inherente de este problema bi-objetivo bi-nivel, se propone un algoritmo de Entropía Cruzada para aproximar el frente de Pareto de este problema de ubicación de los puntos de vacunación. Se necesitan varias adaptaciones para diseñar el algoritmo propuesto. En primer lugar, se realiza una adaptación para orientar la búsqueda considerando las dos funciones objetivo. En segundo lugar, se diseña una estrategia eficiente para tratar con el problema parametrizado del nivel inferior en cada etapa del algoritmo.

1.1 OBJETIVO

La pandemia de COVID-19 por la cual hemos atravesado durante los últimos años y hasta el día de hoy ha traído con ella innumerables efectos, como el contagio de millones de personas en todo el mundo y la muerte de un gran número de personas. En efecto, la vacunación pretende ser la forma más segura y eficaz para controlarla, por lo cual ha habido una gran aceptación de la vacuna COVID-19 y una fuerte demanda de la misma. Dicho lo anterior, el objetivo de este trabajo es estudiar un problema relacionado con las decisiones estratégicas de vacunación.

OBJETIVO GENERAL

El objetivo general planteado es proponer un programa bi-objetivo bi-nivel para modelar un sistema de vacunación que apunte a mantener la *sana distancia* entre los habitantes sin imponer restricciones en cuanto al punto de vacunación que debe asistir. Es importante destacar que dentro de la literatura no existen trabajos de estas características.

OBJETIVOS ESPECÍFICOS

Dentro de nuestros objetivos específicos se establecen los siguientes:

- Proponer una metaheurística de entropía cruzada que considere simultáneamente dos objetivos y maneje la estructura bi-nivel del problema. Es decir, que resuelva el problema del nivel inferior mientras considera las decisiones de nivel superior. Es importante destacar que esta es la primera vez que se aplica la Entropía Cruzada para resolver un problema bi-objetivo bi-nivel.
- Llevar a cabo una extensa experimentación computacional para validar la metaheurística propuesta y obtener información gerencial importante con respecto a la ubicación de los puntos de vacunación y el distanciamiento social.
- Analizar los resultados obtenidos y presentar conclusiones sobre el modelo propuesto y de la metaheurística de Entropía Cruzada utilizada para resolver el problema bajo estudio.

1.2 MOTIVACIÓN

La situación actual por la que están pasando todos los países en general, despertó la inquietud de modelar esta situación para ayudar de forma eficiente y rápida a mitigar

la pandemia. Como se mencionó antes, la mejor forma de disminuir los fallecimientos a causa del COVID-19 es la aplicación de la vacuna a la mayor cantidad de personas. De tal manera que proponemos un modelo para contribuir en la ubicación de puntos de vacunación a los que pueden asistir las personas a recibir la dosis de la vacuna. El modelo propuesto es un problema bi-objetivo bi-nivel, que por su estructura lo hace muy retador e interesante de estudiar. Al contar con dos objetivos en el nivel superior no se llega a una única solución óptima; es decir se encuentra un conjunto de soluciones eficientes que forman la frontera de Pareto.

Además de lo anterior, proponer una metaheurística de Entropía Cruzada para resolver el problema definido anteriormente y saber que es la primera vez que se aplica para resolver un problema bi-objetivo bi-nivel nos motiva a realizar la presente investigación.

1.3 METODOLOGÍA

Para realizar esta investigación, lo primero que se realizó fue definir nuestro problema. Nos interesaba abordar una situación de logística humanitaria, por esta razón y por la presente situación que se vivía a nivel mundial presentamos un modelo bi-objetivo bi-nivel. Dicho modelo se plantea en el contexto de la pandemia COVID-19. Se busca localizar puntos de vacunación que permita a las personas asistir a recibir la dosis de la vacuna respetando todas las recomendaciones sanitarias. Dadas las características de nuestro problema nuestro modelo cuenta con dos niveles de decisión, de los cuales se consideran dos funciones objetivo en el nivel superior y solo una en el nivel inferior. Dentro del modelo se consideraron las restricciones más adecuadas y cercanas a la realidad, por ejemplo, el número de puntos de vacunación que se deben localizar, que los habitantes sólo pueden elegir un solo punto de vacunación abierto para asistir a recibir la vacuna, y que solamente pueden recibir una sola dosis de la vacuna.

Después de tener el modelo bien definido, el paso siguiente fue plantear la forma en que se debe resolver. Para este paso se eligió la metaheurística de Entropía Cruzada. Por consiguiente, se elaboró un algoritmo para resolver nuestro modelo. El algoritmo propuesto de Entropía Cruzada sigue tres pasos muy sencillos: primero genera un conjunto de soluciones; segundo, particiona esas soluciones en una malla del espacio de objetivos; tercero, en cada cuadrícula del mallado se seleccionan las soluciones eficientes, se obtiene información de la frecuencia de los componentes y a partir de ellas se vuelve a generar

un nuevo conjunto de soluciones de manera sesgada. En cada iteración se guarda la aproximación de la Frontera de Pareto. Nuestro algoritmo termina al completar un número máximo de iteraciones.

Al tener implementada la metaheurística para resolver nuestro problema, el siguiente paso fue realizar la experimentación computacional para medir el rendimiento del algoritmo propuesto y su capacidad para aproximar el frente de Pareto. La experimentación computacional se llevó a cabo en base a un conjunto de instancias adaptadas de la literatura. Paso siguiente, se llevó a cabo un análisis de las soluciones obtenidas por nuestro algoritmo propuesto.

1.4 ESTRUCTURA

El resto de este documento está organizado de la siguiente manera. La Sección 2 revisa la literatura relacionada con programación bi-nivel, teoría de localización de instalaciones y en particular, sobre logística de vacunas, pero con especial énfasis en los sistemas para la pandemia de COVID-19. La Sección 3 presenta la formulación matemática del problema bajo estudio. La Sección 4 describe la metaheurística propuesta y sus componentes. La Sección 5 presenta los resultados computacionales en instancias de prueba adaptadas y los análisis respectivos. Finalmente, la Sección 6 describe las conclusiones de nuestra investigación y enumera algunas direcciones de investigación adicionales interesantes.

CAPÍTULO 2

REVISIÓN DE LITERATURA

En este capítulo se presenta la revisión de literatura, la cual se divide en cuatro direcciones principales. En primer lugar, se presenta la descripción general de los problemas de localización de instalaciones y algunos de los problemas clásicos más importantes de esta área. También, se presta especial atención al problema de la p -dispersión y a los problemas de localización de instalaciones que buscan equilibrar la asignación de usuarios. En segundo lugar, se presentan las bases de la teoría relevante sobre los problemas de programación bi-nivel. En tercero, se revisan los antecedentes de los problemas multiobjetivo, se presentan sus conceptos básicos y aplicaciones importantes. Todo lo anterior con el fin de comprender las consideraciones teórico-conceptuales que están implicadas en nuestro problema. En cuarto lugar, se realiza la revisión de literatura referente a la logística de problemas relacionados con la vacunación, en general, con especial énfasis en los sistemas para la pandemia del COVID-19.

2.1 LOCALIZACIÓN DE INSTALACIONES

La rama que estudia a los problemas de localización de instalaciones (FLPs, por sus siglas en inglés) es una de las clásicas en el área de investigación de operaciones. Este problema consiste en, dado un conjunto de sitios potenciales y un conjunto de clientes, se deben tomar las siguientes decisiones: primero, en cuales sitios localizar (abrir) las instalaciones y después a cuales clientes asignar en cada instalación abierta, de tal forma que se optimice una función objetivo cumpliendo con las demandas de los clientes sin rebasar la capacidad de las instalaciones abiertas, (Daskin (1997)).

Dentro de los problemas de localización de instalaciones se pueden identificar tres elementos importantes para este tipo de problemas:

- Las *instalaciones* denotan el conjunto de objetos que serán localizados para proporcionar un servicio o un producto.

- Las *locaciones* se refiere al conjunto de posibles sitios para ubicar las instalaciones.
- Los *clientes* que son los usuarios de las instalaciones que demandan el servicio o producto.

Uno de los primeros modelos sobre localización de instalaciones se atribuye a Alfred Weber, dado que el problema de ubicación dentro de la industria fue particularmente relevante a fines del siglo XIX, cuando la revolución industrial estaba bien establecida y el desarrollo del transporte ferroviario, la energía, las telecomunicaciones y el crecimiento urbano brindaron más opciones para distribuir las empresas y los componentes del proceso de fabricación, (Fearon (2006)). Dicho modelo, Weber (1929) menciona algunos de los factores esenciales al momento de localizar instalaciones que minimice los costos de transporte y mano de obra. En primer lugar se desea analizar el transporte óptimo basado en los costos de la distancia, en segundo lugar la distorsión laboral, en la que fuentes más favorables de mano de obra más barata pueden justificar mayores distancias de transporte, y en tercer lugar, la aglomeración y desaglomeración de instalaciones.

Dentro de la literatura, el problema de localización de instalaciones tiene diversas aplicaciones y variantes. Por ejemplo, la consideración de una capacidad limitada en las instalaciones, un radio de cobertura, usuarios con demanda unitaria, instalaciones equilibradas respecto de los usuarios asignados, la ubicación de un número predeterminado de instalaciones, entre otros. En Eiselt and Marianov (2011) y Laporte et al. (2019) se presentan algunas variantes y aplicaciones. Asimismo, se detallan sus modelos matemáticos y los métodos propuestos para la resolución de cada problema específico.

Un modelo general de localización de instalaciones propuesto en Cornuéjols et al. (1983) se presenta a continuación. Sea $I = \{1, \dots, m\}$ el conjunto de los clientes, $J = \{1, \dots, n\}$ el conjunto de los sitios potenciales para ser localizados. Sea f_j los costos fijos de localizar el sitio j , c_{ij} las ganancias de satisfacer la demanda del cliente i de la instalación j . Las variables $x_j = 1$ si la instalación j es localizada (abierta), $x_j = 0$ de otra forma; $y_{ij} = 1$ si la demanda del cliente i es satisfecha por la instalación j , $y_{ij} = 0$ de otra forma.

$$\text{máx} \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij} - \sum_{j \in J} f_j x_j \quad (2.1)$$

$$\text{Sujeto a: } \sum_{j \in J} y_{ij} = 1 \quad \forall i \in I \quad (2.2)$$

$$y_{ij} \leq x_j \quad \forall i \in I, j \in J \quad (2.3)$$

$$x_j, y_{ij} \in \{0, 1\} \quad (2.4)$$

El modelo general de localización de instalaciones está definido por (2.1)-(2.4). Las restricciones respresentativas del problema de localización de instalaciones se expresan en: Ec. (2.2) y Ec. (2.3). Las cuales aseguran que la demanda de cada cliente i es satisfecha y que los clientes son suministrados por una sola instalación abierta respectivamente.

El número de instalaciones que deben abrirse no esta especificado en el modelo anterior, si se desea tener un número fijo de instalaciones abiertas se agrega la siguiente restricción.

$$\sum_{j \in J} x_j = p \quad (2.5)$$

Donde p es un número entero dado, $1 \leq p \leq n$. Este modelo se puede aplicar en diversos contextos como: diseño de redes, enrutamiento de vehículos, y por supuesto en localización de instalaciones.

Uno de los problemas clásicos en el que se considera el número de instalaciones que se deben abrir es el problema de la p -mediana. El cual está motivado por situaciones reales en las cuales se puede aplicar este modelo, como por ejemplo, tener que instalar un número específico de plantas en algunos puntos de un sistema de la industria para minimizar los costos de producción y envío. Otro ejemplo se da cuando se tienen que instalar diferentes hospitales, escuelas, estaciones de bomberos, etc., en una ciudad para que la mayor parte de su población quede atendida en caso de necesitar el servicio. En particular, el problema de la p -mediana trata de localizar p instalaciones dentro de una red de manera que se minimice la distancia ponderada entre los puntos de demanda y la instalación abierta más cercana. A principios de la década de 1960, Hakimi estudio este tipo de problemas Hakimi (1964), Hakimi (1965) de tal manera que el problema de la

p -mediana se ha convertido en uno de los modelos de localización de instalaciones más estudiados en el área.

En problemas más recientes de localización de instalaciones se consideran dos tipos de costos. Por un lado, el costo de instalación que se genera al abrir una instalación y costo de conexión que ocurre cuando un cliente está asignado a una instalación abierta. Por ejemplo, los costos totales pueden incluir los costos fijos asociados con la apertura de los almacenes y los costos de transporte asociados con el envío de productos desde los almacenes a los clientes. Si se puede conectar un número arbitrario de clientes a una instalación, el problema se denomina problema de ubicación de la instalación no capacitada (UFLP). Si cada instalación tiene un límite en el número de clientes que puede atender, se convierte en un problema de ubicación de instalaciones capacitadas (CFLP). Tanto UFLP como CFLP son NP-duros, por tal razón se han estudiado y desarrollado ampliamente algoritmos, exactos y heurísticos para resolver este tipo de problemas, por ejemplo Akinc and Khumawala (1977), Beasley (1988), Erlenkotter (1978) y Galvão and Raggi (1989).

Otro de los modelos más populares entre los modelos de ubicación de instalaciones es el problema de cobertura. Esto se debe a el gran número de aplicaciones en la vida real, especialmente para servicios e instalaciones de emergencia. En algunos problemas de cobertura, un cliente debe ser atendido por al menos una instalación dentro de una distancia crítica dada (no necesariamente la instalación más cercana). En la mayoría de los problemas de cobertura, los clientes reciben servicios por instalaciones dependiendo de la distancia entre el cliente y las instalaciones. El cliente puede recibir servicio de cada instalación cuya distancia del cliente sea igual o menor que un número predefinido. Este número crítico predefinido se denomina distancia de cobertura o radio de cobertura, (Farahani and Hekmatfar (2009)). Muchos de los problemas, como determinar el número y ubicación de escuelas públicas, comisarías, bibliotecas, hospitales, edificios públicos, oficinas de correos, parques, campos militares, instalaciones de radar, sucursales bancarias, centros comerciales e instalaciones de eliminación de desechos, pueden formularse como un problema de cobertura. En Farahani et al. (2012) se realiza un revisión muy extensa de los problemas de cobertura.

En particular, en esta tesis estamos interesados en dos problemas de localización de instalaciones estrechamente relacionados y muy conocidos. El primero de ellos estudia el caso en que se deben ubicar p instalaciones con el objetivo de estar lo más lejos posible. Este problema se denomina problema p -dispersión; Dasarathy and White (1980), Erkut (1990). De forma general, se busca que las instalaciones deben estar lo más lejos

posible de la otra instalación más cercana. Entonces se formula un problema entero que se basa en invertir el valor de las variables de ubicación (1 si la instalación se abre y 0 de otra forma) en las restricciones de distancia para que solo la distancia entre pares de instalaciones abiertas restrinja la maximización. Dado que el objetivo es maximizar la distancia mínima entre las instalaciones ubicadas, la asignación de usuarios es irrelevante. En muchos contextos es importante tener en cuenta la asignación de usuarios. Esa es una de las razones por las que se considera la asignación en el problema de estudio en esta tesis.

El segundo problema de localización de instalaciones de interés específico es el FLP que equilibra la asignación de usuarios en las instalaciones ubicadas. Este problema se plantea en Marín (2011), en el cual el objetivo es minimizar la diferencia entre el número máximo y mínimo de usuarios asignados a las instalaciones. En ese problema, los usuarios se asignan a las instalaciones ubicadas más cercanas.

Como se mencionó anteriormente, una de las principales contribuciones de nuestro estudio es combinar estos dos problemas de localización de instalaciones en un mismo modelo. La naturaleza del problema original de la p -dispersión descuida la asignación de usuarios. Sin embargo, como ya se mencionó, hay situaciones en las que es importante tenerlo en cuenta. Además, se puede buscar el equilibrio de los usuarios asignados a cada instalación ubicada para mejorar la estabilidad del sistema. Es decir, se pretende que la carga de trabajo de las instalaciones sea similar. Por lo cual esta es la primera vez que estos dos problemas se fusionan en uno solo.

Otro factor importante que se debe considerar son las preferencias, que comúnmente se incluyen para considerar el comportamiento de los usuarios en problemas de asignación. Por ejemplo, en el problema de Emparejamiento, los elementos de un conjunto colocan sus preferencias hacia los elementos del otro conjunto y viceversa McVitie and Wilson (1971), Iwama and Miyazaki (2008). Otro enfoque a la hora de considerar las preferencias en el proceso de toma de decisiones corresponde a la asignación de usuarios en FLP. En este caso, los usuarios frecuentan las instalaciones en función de sus propias preferencias, que pueden basarse en el atractivo, la proximidad, la calidad, entre otros. Esto puede verse como la regla de asignación seguida por los usuarios en un esquema de ubicación de instalaciones.

Las preferencias de los usuarios se consideraron por primera vez en FLP en Hanjoul and Peeters (1987), en el que los usuarios ordenaron las instalaciones potenciales en

función de sus propias preferencias. Una vez que se localizan algunas instalaciones, la lista ordenada de preferencias determina la asignación de los usuarios. Desde entonces, diferentes FLP han considerado las preferencias de los usuarios, como en Cánovas et al. (2007), Hansen et al. (2004). Además, la consideración de las preferencias de los usuarios da lugar a una estructura jerárquica del problema, que se puede modelar naturalmente con programación bi-nivel Marić et al. (2012), Maldonado-Pinto et al. (2016), Casas-Ramírez et al. (2020), Calvete et al. (2020).

Recientemente, Camacho-Vallejo et al. (2022) estudia un problema de programación de dos niveles que equilibra el número de usuarios asignados a las instalaciones ubicadas considerando las preferencias de los usuarios. La principal diferencia de nuestro estudio con respecto a Camacho-Vallejo et al. (2022) es que estamos considerando dos funciones objetivo, simultáneamente. Por lo tanto, el algoritmo diseñado en esa investigación no puede extenderse fácilmente a nuestro problema bi-objetivo. La consideración de dos funciones objetivo complica la orientación hacia la mejora de las soluciones. Ese modelo surge del contexto de configuración de una red 5G.

2.2 PROGRAMACIÓN BI-NIVEL

Muchas situaciones reales involucran algún tipo de jerarquía dentro de los tomadores de decisiones. Dichas situaciones se pueden plantear con ayuda de modelos que hagan referencia a la jerarquía por medio de niveles. El área que estudia a ese tipo de problemas se llama programación bi-nivel. En Von Stackelberg (1934) fue la primera vez que se planteó un problema jerárquico en el cual se describen las situaciones reales de un mercado específico. En dicho modelo, diferentes agentes (también llamados tomadores de decisiones) pretenden tomar la mejor decisión en el mercado con respecto a ellos mismos, pero no pueden tomar sus decisiones de manera independiente porque se ven forzados a actuar de acuerdo a la jerarquía existente en el mercado. Si sólo existen dos tomadores de decisiones (caso más simple) entonces esta jerarquía tiene dos niveles, un líder que toma las decisiones de manera independiente y un seguidor que toma sus decisiones considerando la decisión del líder.

Un problema de programación bi-nivel consiste en un problema de optimización que incluye otro problema de optimización dentro de sus restricciones. Por lo tanto, se tiene un problema de dos niveles (nivel superior y nivel inferior). En estos problemas los toma-

dores de decisiones del nivel superior e inferior se les da el nombre de líder y seguidor, respectivamente. El tomador de decisiones del nivel inferior busca optimizar su función objetivo bajo parámetros dados por el tomador de decisiones del nivel superior, quien a su vez con información completa de las posibles reacciones del nivel inferior selecciona los parámetros para optimizar su propio objetivo.

Un modelo general de un problema de programación bi-nivel se muestra a continuación (Bard (2013)):

$$\min_{x \in X} F(x, y) \quad (2.6)$$

$$\text{Sujeto a: } G(x, y) \leq 0 \quad (2.7)$$

$$\min_{y \in Y} f(x, y) \quad (2.8)$$

$$\text{Sujeto a: } g(x, y) \leq 0 \quad (2.9)$$

$$x, y \geq 0 \quad (2.10)$$

El problema de programación bi-nivel está definido por las Ec. (2.6)-(2.10), de tal manera que el problema del líder está definido por las Ec. (2.6)-(2.8), donde la Ec. (2.6) es la función objetivo del líder, dicho problema tiene dos conjuntos de restricciones dados por las Ec. (2.7) y las Ec. (2.8), respectivamente. En particular, la última restricción es otro problema de programación matemática. Este problema es el problema del seguidor que está definido por las Ec. (2.8)-(2.10), la función objetivo del seguidor está dada por la Ec. (2.8) y los conjuntos de restricciones para el problema del seguidor están dados por las Ec. (2.9) y Ec. (2.10), donde ésta última es la restricción de signo de las variables de decisión del problema.

En resumen del modelo anterior, el líder busca minimizar $F(x, y)$, seleccionando una $x \in X$ y considerando la mejor respuesta del seguidor, $y^*(x)$, que minimice $f(x, y)$ en el problema del nivel inferior a su vez también considerando las restricciones funcionales $G(x, y) \leq 0$ y $g(x, y) \leq 0$ en cada uno de los dos niveles del problema.

La región de restricciones del problema de programación bi-nivel se denota de la siguiente manera:

$$S = \{(x, y) : x \in X, y \in Y, G(x, y) \leq 0, g(x, y) \leq 0\} \quad (2.11)$$

El conjunto factible del seguidor, para cada $x \in X$ se denota por:

$$S(x) = \{y \in Y : g(x, y) \leq 0\} \quad (2.12)$$

Dado que el seguidor optimiza $f(x, y)$ después de haber seleccionado x , la reacción racional del seguidor ante un valor de x dado es:

$$\Psi(x) = \arg \min_{y \in Y} \{f(x, y) : g(x, y) \geq 0\}. \quad (2.13)$$

La región factible para el problema binivel se denomina región inducible:

$$IR = \{(x, y) : (x, y) \in S, y \in \Psi\} \quad (2.14)$$

Si la solución óptima del seguidor existe y es única para cada decisión del líder entonces se dice que el problema bi-nivel está bien definido. Sin embargo, si existen múltiples soluciones óptimas para el seguidor, entonces se genera incertidumbre entre estas y no está claro cual de todas ellas va a considerar el líder para su problema.

La posibilidad de que el seguidor pueda tener múltiples respuestas óptimas significa un problema para el líder, ya que no sabe cual solución óptima elegirá el seguidor. Esta elección puede afectar significativamente el valor de la función objetivo del líder y, en consecuencia, su decisión. En muchos estudios, esta dificultad se evita simplemente asumiendo que la solución al problema del nivel inferior es única. En otros casos, se considera un enfoque optimista o pesimista. El enfoque optimista (el más común) asume que el líder es libre de seleccionar la solución que más le convenga entre las decisiones óptimas del seguidor. En este caso, la optimización de nivel superior se toma con respecto a x y y , lo que significa que el líder puede influir en la elección del seguidor, Dempe et al. (2009). Cuando no se permite la cooperación entre el líder y el seguidor, o si el líder tiene aversión al riesgo y desea compensar el daño resultante de una selección indeseable del seguidor, se puede admitir un enfoque pesimista. Es decir el líder se prepara para el peor de los casos, de modo que elige una decisión x que funciona mejor en vista de la respuesta del seguidor. Si no hay garantía de la existencia y unicidad de la solución del nivel inferior,

deberá asumirse alguno de estos dos casos.

Dentro de los problemas bi-nivel se puede presentar el caso donde a pesar de que las funciones puedan ser continuas y acotadas no se pueda garantizar que existe una solución. Por lo tanto, se puede concluir que los problemas de programación bi-nivel no son fáciles de tratar. Además, estos problemas están clasificados como NP-Hard, lo cual implica que no existe un algoritmo en tiempo polinomial que pueda resolver este problema de forma rápida (Bard (2013).)

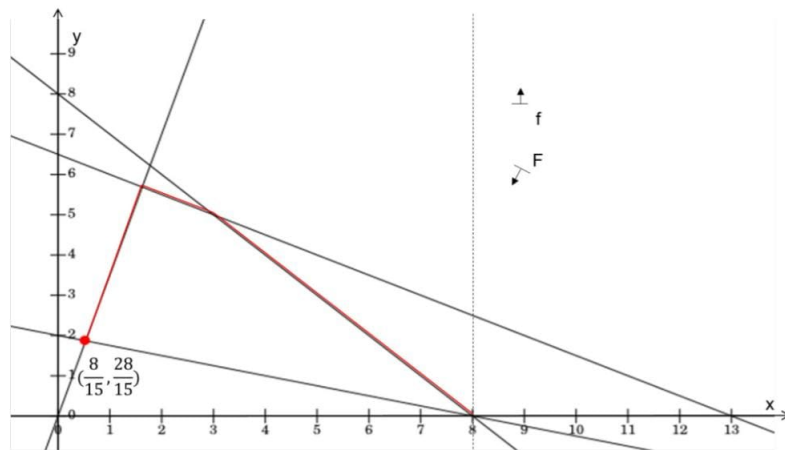
Considerar restricciones en el problema del líder puede ser otro punto de interés, ya que al hacerlo las restricciones pueden provocar que la región inducida se vuelva disconexa, tal como se presenta en la Figura 1 (imagen tomada de Ortega Rosales (2015)). El hecho de dejar la región inducida disconexa complica aún más el problema binivel.

Es de suma importancia encontrar la forma adecuada de resolver un problema bi-nivel, por esta razón dicho problema ha sido de interés en gran número de estudios. Cuando el problema de programación bi-nivel es convexo, es decir, que el problema del seguidor sea convexo; bajo procedimientos apropiados, se puede obtener un problema equivalente de un solo nivel. Existen varias técnicas de reducción de los problemas binivel a un problema de un solo nivel, las técnicas más comunes son: mediante las condiciones de Karush-Kuhn-Tucker, mediante teoremas de funciones implícitas, con la relación primal-dual del nivel inferior, entre otras metodologías.

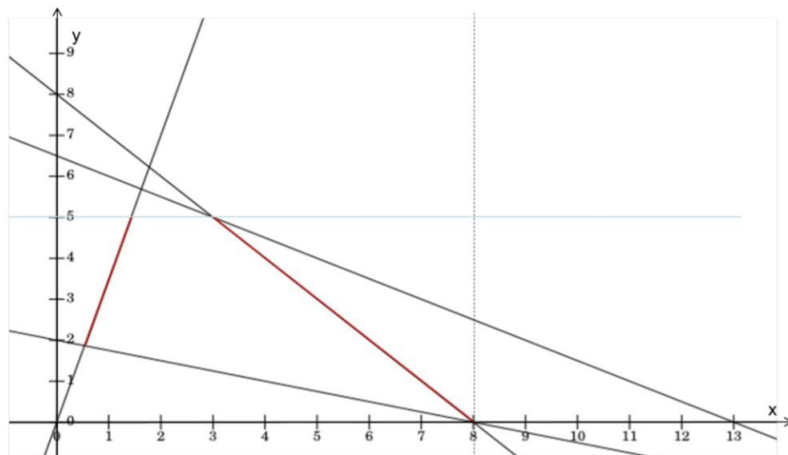
2.3 PROGRAMACIÓN MULTI-OBJETIVO

En la mayoría de los problemas de optimización dentro del mundo real surgen de forma natural escenarios que necesitan optimizar múltiples objetivos de manera simultánea. A menudo, estos objetivos son contradictorios y no existe una única solución que simultáneamente satisfaga a todos. Por lo tanto, la solución que se pretenda obtener debe estar en concordancia con las preferencias o criterios del tomador de decisiones.

La programación multi-objetivo es una rama de la programación matemática que se encarga de estudiar los problemas de decisión con múltiples funciones objetivo que deben ser optimizadas sobre un conjunto factible de soluciones; Ehrgott et al. (2005). El



(a) La región factible para un problema binivel lineal



(b) Si el líder tiene restricciones adicionales la región factible puede ser disconexa

Figura 1: Gráficas ilustrativas en programación Bi-nivel lineal

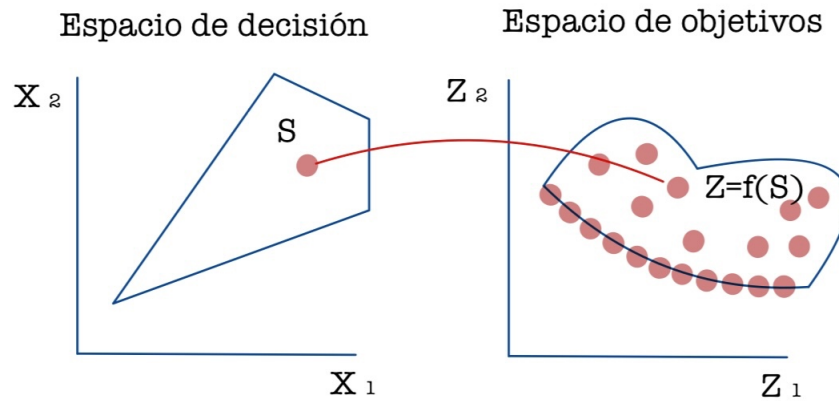


Figura 2: Gráficas ilustrativas de los espacios de decisión y de los objetivos

problema de optimización multi-objetivo general se define a continuación:

$$\text{mín}(F_1(x), F_2(x), \dots, F_n(x)) \quad (2.15)$$

$$\text{Sujeto a: } x \in X \quad (2.16)$$

donde n es el número de funciones objetivo, x es el vector de variables de decisión y X el espacio de decisión factible o delimitado por el conjunto de restricciones. Los problemas multi-objetivo poseen dos espacios de búsqueda: el espacio de decisión conocido como el conjunto de soluciones factibles y el espacio de los objetivos, el cual es el espacio que forman las funciones objetivo (ver Figura 2, tomada de Méndez et al. (2018)).

En la práctica, a menudo observamos un orden parcial de preferencia del tomador de decisiones, por lo cual la noción de optimalidad no existe necesariamente bajo este enfoque. Por tal motivo, cuando se tiene más de una función objetivo la noción de optimalidad cambia, ya que en este tipo de problemas el objetivo es llegar a un buen compromiso considerando todas las funciones objetivo del problema, en lugar de una única solución. El concepto de optimalidad en problemas multi-objetivo fue originalmente propuesto por Francis Y. Edgeworth en 1891 y generalizada por Vilfredo Pareto en 1896, llamándolo Frontera de Pareto. El objetivo de estos problemas es que, en lugar de buscar una única solución óptima, se busca un conjunto de soluciones las cuales pueden ser un número indefinido de soluciones y cuando esto sucede para reducir este número de soluciones, en primer lugar, se procede a desechar aquellas que están dominadas y solo quedarnos con las soluciones eficientes o bien soluciones no-dominadas. Por lo tanto, es importante

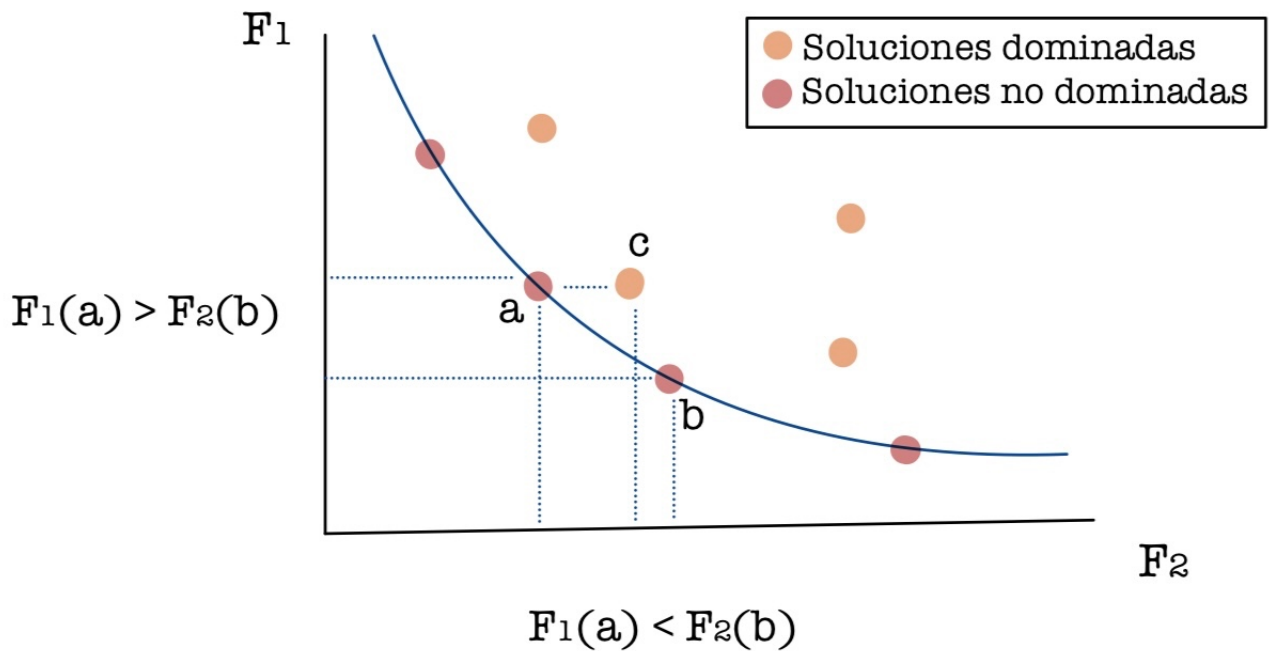


Figura 3: Gráficas ilustrativas de la Frontera de Pareto

introducir la noción de eficiencia en la programación multi-objetivo.

Sea $x^* \in X$ una solución eficiente del problema multi-objetivo, es decir, no existe otro punto $x \in X$ tal que $F(x) \leq F(x^*)$ y $F_i(x) < F_i(x^*)$ en al menos en una función objetivo i .

Esto significa que x^* es una solución eficiente si no existe ninguna otra solución factible x que sea mejor en algún criterio (alguna función objetivo) sin causar un incremento simultáneo en al menos otro criterio. El conjunto de soluciones eficientes se llama **Frontera de Pareto**. Si se representan de forma gráfica, los valores de las funciones objetivo del conjunto de soluciones formarán la frontera del espacio entre las soluciones factibles y no factibles, (ver la Figura 3, Pinilla and Castro (2015))

Dentro de la teoría de los problemas de optimización multi-objetivo es importante introducir varios conceptos de gran utilidad para comprender e interpretar el problema planteado, como son la matriz de pagos y los puntos ideal, utópico y nadir.

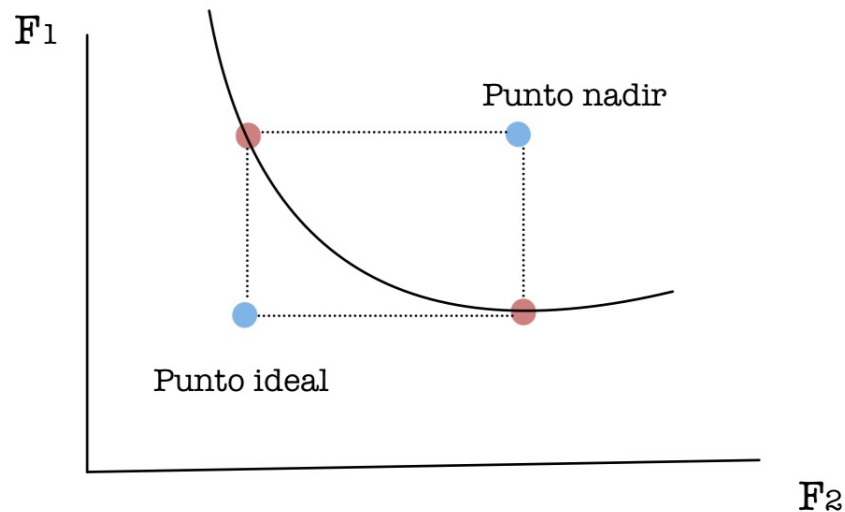


Figura 4: Gráficas ilustrativas del punto ideal, nadir y utópico

La matriz de pagos es una matriz que nos permite cuantificar el nivel de conflicto existente entre los objetivos que se estén considerando. Se construye a partir de las soluciones que optimizan de manera independiente a cada uno de los objetivos. Con cada solución obtenida se construye un vector formado por los valores que toman las funciones objetivo en ese problema y los valores que resultarían para los demás objetivos con esa solución. Se obtiene así una matriz cuadrada en cuya diagonal principal se encuentra el punto *ideal*, formado por los valores óptimos de cada uno de los objetivos. Este punto es en general inalcanzable, a no ser que los objetivos no se encuentren en conflicto en cuyo caso el problema multi-objetivo tendría una única solución. También se puede definir el punto *utópico*, como un punto inalcanzable, cuyas componentes de las diferentes funciones objetivo poseen valores más ambiciosos que el del punto ideal. Por otra parte, el punto *nadir* representa una solución no deseada. Este vector se define por los peores valores de las funciones objetivo en la matriz, (ver la imagen en la Figura 4, Filippi and Vasile (2020)).

Dada la formulación y complejidad de los problemas multi-objetivo existe una gran variedad de técnicas o métodos para encontrar un conjunto de soluciones que son igualmente buenas entre sí para este tipo de problemas. De tal manera que se han desarrollado diferentes enfoques de solución que permiten obtener el conjunto de soluciones; es decir la frontera de Pareto. Entre estos métodos de solución se pueden encontrar los siguientes: método de suma ponderada, método de la ϵ -constraint, método de programación por compromisos y el método de programación por metas.

El método de suma ponderada, se basa en la idea de convertir el problema multi-objetivo en un problema con una sola función objetivo, de forma que se construye la función objetivo por medio de la suma ponderada de las funciones objetivo según un peso relativo que se le asigne a cada una de ellas. De esta forma, para cada ponderación posible, se obtiene un problema escalar consistente en minimizar la función resultante, sujeta a las restricciones del problema original. Este método de solución se puede expresar de la siguiente manera.

$$\text{mín } \sum_{i=1}^n w_i F_i(x) \quad (2.17)$$

$$\text{Sujeto a: } x \in X \quad (2.18)$$

Donde, w_i son los pesos asignados a cada uno de las funciones objetivo del problema y se debe de cumplir lo siguiente: $0 \leq w_i \leq 1$ y $\sum_{i=1}^n w_i = 1$, dicho peso se interpreta como la importancia de la función objetivo i en relación con el resto de las funciones objetivo.

Para llevar a cabo el método de suma ponderada se sugiere considerar en forma sistemática una serie de conjuntos de pesos positivos que multipliquen a cada objetivo del problema. Usualmente se empieza por la optimización individual de cada objetivo, es decir asignar un peso $w_i = 1$ a cada objetivo individualmente y el resto de los pesos igual a cero. Posteriormente se introduce una variación sistemática de estos pesos con una tasa de aumento prefijada, lo cual paso a paso va a representar la importancia que tiene cada objetivo sobre los demás según las preferencias del tomador de decisiones.

En la aplicación de este método, se pueden tener ciertos inconvenientes, tales como: que distintos conjuntos de asignaciones podrían arrojar los mismos resultados; que el tamaño de paso de un conjunto de pesos a otro puede no permitir generar todos los puntos extremos y, por tanto, se obtendría una aproximación al conjunto de soluciones eficientes. Con lo anterior, se hace referencia a que en algunas ocasiones las asignaciones de pesos a cada uno de los objetivos no permiten encontrar todas las soluciones posibles. El método de suma ponderada es posiblemente la técnica multi-objetivo más considerada para la solución de este tipo de problemas. La ventaja que presenta es su fácil aplicación y su desventaja principal es la dificultad para determinar la ponderación correspondiente a cada objetivo, dado que la cantidad de objetivos que maneja un tomador de decisiones

para realizar su elección, y la importancia que tiene cada uno de estos, varía en cada contexto.

El método de ϵ -constraint consiste en optimizar la función objetivo que se considere más importante sobre las otras funciones objetivo del modelo, estas últimas se consideran como restricciones a las cuales se les asigna un valor real que corresponden o sirve como cota inferior de la función objetivo. Las cotas inferiores al igual que la función objetivo seleccionada como la de mayor importancia, representan preferencias subjetivas del tomador de decisiones. De tal manera, el problema multi-objetivo se reduce a un problema con un único objetivo.

$$\text{mín } F_i^*(x) \quad (2.19)$$

$$\text{Sujeto a: } x \in X \quad (2.20)$$

$$F_i(x) \leq \epsilon_i \quad (2.21)$$

donde, $F_i^*(x)$ representa el objetivo seleccionado como el de mayor importancia.

La idea tras el método de la programación por metas es encontrar el conjunto de soluciones que estén lo más cerca posible a las metas definidas para cada uno de los objetivos incluidos en el modelo. Para esto, el tomador de decisiones debe definir el valor objetivo o meta para cada una de las funciones; a continuación resuelve un problema de un solo objetivo que le ayude a minimizar la suma de las desviaciones de los objetivos. Si b_i es la meta correspondiente al objetivo i , el problema de un solo objetivo es:

$$\text{mín } \sum (d_i^- + d_i^+) \quad (2.22)$$

$$\text{Sujeto a: } F_i(x) + d_i^- - d_i^+ = b_i, \quad i = 1, \dots, n \quad (2.23)$$

$$x \in X \quad (2.24)$$

$$d_i^-, d_i^+ \geq 0 \quad i = 1, \dots, n \quad (2.25)$$

donde, las variables d_i^- , d_i^+ son el bajo rendimiento o la superación de la i -ésima meta, respectivamente.

Después de realizar el establecimiento de las metas para cada uno de los objetivos, el tomador de decisiones asigna a estos diferentes niveles de prioridad, es decir, les asigna un peso a cada uno, de modo que le permita ordenarlos atendiendo a las prioridades identificadas. Esta asignación puede ser uno a uno, es decir, en cada nivel un solo objetivo, o pueden aparecer varios compartiendo un mismo nivel; este caso implica una ponderación entre los objetivos que comparten el nivel. De esta forma, la resolución se llevará a cabo atendiendo a las metas impuestas y a los niveles de prioridad establecidos, de forma que se preferirá una solución que mejore al primero, una vez conseguida, se pasa al segundo y así sucesivamente.

Dentro de la literatura existen dos enfoques para generar conjuntos de soluciones de programación multi-objetivo: los métodos de *escalarización* y los métodos de *no escalarización*. Los métodos de escalarización implican la formulación de un modelo mono-objetivo relacionado con el modelo multi-objetivo por medio de una función escalar. Esta única función escalar está en términos de las funciones objetivo del modelo multi-objetivo, escalares auxiliares, un vector de variables o un vector de parámetros.

Los métodos de no escalarización implican un breve tratamiento independiente de cada función objetivo. En contraste con los métodos de escalarización, no utilizan explícitamente una función de escalarización sino que se basan en otros conceptos de optimalidad o conjuntos auxiliares y por lo general se vinculan con conceptos de eficiencia. En seguida se presentan las Tablas 1 y 2, donde se muestran algunos de los métodos usados para resolver problemas multi-objetivo (Pinilla and Castro (2015)).

En la mayoría de los problemas multi-objetivo no es fácil obtener de forma exacta la frontera de Pareto, dado que esto puede ser computacionalmente difícil y costoso. Considerando lo anterior, una forma aproximada de la frontera de Pareto se puede obtener mediante un algoritmo de aproximación y puede resultar conveniente dado que requiere menos esfuerzo computacional.

Los enfoques de aproximación emplean un método iterativo para producir puntos que se aproximen a la frontera de Pareto. Algunos enfoques son exactos y se basan en algoritmos que aseguran la obtención de óptimos de Pareto, mientras que otros enfoques se basan en heurísticas que no necesariamente aseguran que los puntos encontrados sean óptimos de Pareto. Existen varios tipos de heurísticas para la solución de problemas multiobjetivo dentro de los que se encuentran los algoritmos evolutivos (EA), recodido si-

| Método | Nombre en inglés |
|--------------------------------|----------------------------------|
| Programación por metas | Goal programming |
| Suma ponderada | Weighted sum |
| Restricción ϵ | ϵ -constraint |
| Función logro | Achievement scalarizing function |
| Enfoque de Roy | Roy approach |
| Producto ponderado | Weighted product |
| Esquema de arbitraje de Nash | Nash arbitration scheme |
| Función compromiso | Compromise function |
| Media geométrica ponderada | Weighted geometric mean approach |
| Suma objetivo | Objective sum |
| Enfoque de Pascoletti-Serafini | Pascoletti-Serafini approach |
| Método de Benson | Benson's method |
| Norma ponderada | Weighted norm |
| Criterio global ponderado | Weighted global criterion |
| Restricción elástica | Elastic constraint approach |
| Exponencial ponderado | Exponential weighted |
| Producto objetivo | Objective product |
| Potencia media ponderada | Weighted power mean |

Tabla 1: Métodos de escalarización

| Método | Nombre en inglés |
|---------------------------------|--------------------------------------|
| Lexicográfico | Lexicographic method |
| Programación física | Physical programming |
| Intersección de frontera normal | Normal boundary intersection (NBI) |
| Restricción normalizada | Normal constraint (NC) |
| Método Tchebycheff | Wweighted Tchebycheff method |
| Min-max/Enfoque max-ordenado | Min-max method/Max-ordering method |
| Conjunto de nivel | Level set method |
| Lexicográfico max-ordenado | Lexicographic max-ordering |
| Polak | Modified Polak or Polak-based method |

Tabla 2: Métodos de no escalarización

| Método | Nombre en inglés | |
|--|---|------------|
| Algoritmo genético de clasificación no dominada II | Nondominated sorting genetic algorithm II | NSGA II |
| Algoritmo genético multiobjetivo | Multi-objective genetic algorithm | MOGA |
| Algoritmo evolutivo de fuerza de Pareto | Strength Pareto evolutionary algorithm | SPEA |
| Algoritmo genético de clasificación no dominada | Nondominated sorting genetic algorithm | NSGA |
| Estrategia de evolución con archivo de Pareto | Pareto archived evolution strategy | PAES |
| Algoritmo evolutivo de fuerza de Pareto II | Strength Pareto evolutionary algorithm II | SPEA II |
| Algoritmo genético para el vector evaluado | Vector evaluated genetic algorithm | VEGA |
| Algoritmo genético de nicho de Pareto | Niched-Pareto genetig algorithm | NPGA |
| Algoritmo de selección por envoltorio de Pareto II | Pareto envelope-based selection algorithm II | PESA II |
| Algoritmo inmune de vecino no-dominado | Nondominated Neighbor Immune Algorithm | NNIA |
| Algoritmo de selección por envoltorio de Pareto | Pareto envelope-based selection algorithm | PESA |
| Algoritmo de selección clonal de memoria inmune de rango preferencia | Preference rank immune memory clone selection algorithm | PISA |
| Algoritmo genético desordenado multiobjetivo II | Multi-objective messy genetic algorithm II | MOMGA II 6 |
| Algoritmo genético de nicho de Pareto II | Niched-Pareto genetig algorithm II | NPGA II |
| Algoritmo genético desordenado multiobjetivo | Multi-objective messy genetic algorithm | MOMGA |

Tabla 3: Algoritmos evolutivos multiobjetivo

mulado (simulated annealing,SA), búsqueda tabú (tabu search, TS), colonia de hormigas (ant colony, AC), aprendizaje por refuerzo distribuido (distributed reinforcement learning, DRL), algoritmos meméticos (memetic algorithms, MA) y técnicas híbridas. Las técnicas o métodos más trabajados en problemas multi-objetivo han sido los algoritmos evolutivos. En seguida se presenta la Tabla 3, donde se muestran algunos de los algoritmos evolutivos multi-objetivo usados dentro de la literatura Pinilla and Castro (2015).

2.4 APLICACIONES RELACIONADAS CON EL COVID-19

Es evidente que la irrupción del COVID-19 ha generado profundos cambios sociales y económicos. En consecuencia, se deben habilitar diferentes enfoques para prevenir y controlar esta enfermedad. Por esta razón, en los últimos años se han dedicado importantes esfuerzos a la pandemia de COVID-19. Por ejemplo, estudios relacionados con la cadena de suministro: Rahman et al. (2021), Burgos and Ivanov (2021), Chowdhury et al. (2021), asignación de pacientes: Sun et al. (2014), Luo et al. (2022), Sarkar et al. (2021), distribución de kits de prueba: Buhat et al. (2021), Thul and Powell (2023), distribución de medicamentos y equipos médicos: Parvin et al. (2018), Abdin et al. (2023) y ubicación de los centros de pruebas de COVID-19: Ozdemir et al. (2022), Manupati et al. (2021), Mitropoulos et al. (2006).

Desde el comienzo de la pandemia, nos hemos enfrentado a una carrera sin precedentes para desarrollar una vacuna. Se han identificado esfuerzos exitosos y hay vacunas disponibles. Sin embargo, la tasa de producción no es suficiente para vacunar a toda la población mundial. Por lo tanto, la cantidad limitada de vacunas disponibles crea otro gran desafío: ¿Cómo distribuir vacunas para mitigar el impacto cerca del final de la pan-

demia?

En los últimos años, ha habido un interés creciente en la logística de vacunas (Duijzer et al. (2018a)). Por ejemplo, los estudios que consideran la distribución equitativa de las vacunas se pueden encontrar en: Enayati and Özaltın (2020), Medlock and Galvani (2009), Mohammadi et al. (2022), Rastegar et al. (2021). En particular, Bertsimas et al. (2022) propone un modelo que busca optimizar la estrategia de distribución de vacunas. La decisión principal del problema es ubicar los centros de vacunación dentro de una nación. Concluyen que determinar el número y la ubicación de los centros de vacunación también es fundamental para garantizar que las vacunas estén disponibles para toda la población. Existen algunos trabajos donde se estudia como ubicar y distribuir vacunas, por ejemplo Rastegar et al. (2021), los autores presentan un modelo de programación lineal entero mixto para un inventario y problemas de localización. Sin embargo, se enfocan en una distribución equitativa de las vacunas contra la influenza en los países en desarrollo durante la pandemia. Su modelo propuesto utiliza una función objetivo que apunta a distribuir vacunas de manera equitativa. Además, presentan un estudio de caso de un país en desarrollo para mostrar la eficacia y demostrar la aplicabilidad de su modelo de optimización propuesto.

Vale la pena mencionar que debido a la rápida propagación del COVID-19, la demanda de instalaciones médicas de emergencia ha crecido enormemente. Con respecto a este tema, Wang et al. (2022) aborda una situación en la que se integran problemas de localización y asignación con el objetivo de ayudar en el manejo de las principales emergencias de salud pública. Tiene como objetivos la minimización del tiempo de permanencia en los lugares y la asignación de instalaciones médicas de emergencia en caso de ocurrencia de una emergencia de salud pública. Además, se propone un algoritmo genético para dar solución al problema.

La asignación de vacunas y los factores asociados a ella, como la región geográfica, las limitaciones operativas, la capacidad de los centros médicos y la disponibilidad de vacunas se consideran en Yarmand et al. (2014) y Abbasi et al. (2020). En el primero, Yarmand et al. (2014), considera un proceso de decisión en el que se realiza la vacunación en dos fases para contener el brote de una enfermedad infecciosa en un conjunto de regiones geográficas. En la primera fase, se asigna un número limitado de dosis de vacuna a cada región; mientras que en la segunda fase se pueden asignar dosis adicionales a las regiones donde la epidemia no ha sido contenida. Formulan el problema de asignación de vacunas como un programa lineal estocástico de dos pasos y lo reducen a un programa

lineal para obtener soluciones aceptables. En el segundo, los autores proponen un problema de programación lineal entero mixto con el objetivo de desarrollar un modelo efectivo de asignación de vacunas basado en los parámetros de la población total susceptible al COVID-19 y el riesgo de exposición (ver Abbasi et al. (2020)). Analizan un caso de estudio en el que se aplica el modelo propuesto para probar diferentes escenarios de asignación y distribución de vacunas.

Un aspecto importante a considerar en cuanto a la ubicación y asignación de vacunas se refiere a la equidad. En Enayati and Özaltın (2020), el problema a resolver es la minimización del número de dosis de vacuna distribuidas para extinguir de manera efectiva un brote emergente en sus primeras etapas. Los autores proponen una restricción de equidad para ayudar a las autoridades públicas de salud a distribuir vacunas de manera más justa al tomar decisiones sobre la distribución de vacunas.

Un enfoque importante en el contexto de la pandemia es considerar múltiples objetivos simultáneamente, para el mismo problema. Este último ayuda a proporcionar más información gerencial relacionada con la ubicación y asignación de vacunas. La razón es que se puede identificar un compromiso entre dos funciones objetivo diferentes para evaluar diferentes escenarios de toma de decisiones. En Tang et al. (2022), se aborda un problema de planificación de vacunación de varios períodos. Ese problema minimiza la distancia total de viaje de los habitantes vacunados y minimiza el costo operativo del programa de salud. Un plan óptimo determina, para cada período, que puntos de vacunación abrir, cuantas estaciones de vacunación habilitar en cada sitio, como asignar habitantes de diferentes regiones para abrir los puntos de vacunación y la cantidad de reabastecimiento en cada punto de vacunación. Se formulan el problema como un programa bi-objetivo lineal entero mixto. Inicialmente, proponen dos métodos de solución: la suma ponderada y el método de ϵ -constraint. Además, desarrollan un algoritmo genético donde se incluye una estrategia de asignación y un nuevo método de programación dinámica en el algoritmo para obtener buenas soluciones factibles. Además, se presenta un caso de estudio que indica que los métodos reducen el costo operativo y la distancia total de viaje. En Li et al. (2021), se propone un modelo de programación no lineal entero mixto de múltiples objetivos para ayudar a los centros de control y prevención de enfermedades a determinar las ubicaciones de las estaciones de vacunación y, al mismo tiempo, considerar la distancia de viaje, el costo operativo y horas de trabajo.

Otro estudio bi-objetivo relacionado aparece en Mohammadi et al. (2022), que busca optimizar el diseño de una red de distribución de vacunas a través de un modelo de

programación no lineal entero mixto, pero considerando dos objetivos: la minimización del número total esperado de muertes en la población y la minimización de los costos totales de distribución asociados a la campaña de vacunación. El modelo propuesto se linealiza y valida a través de un caso de estudio real de la campaña de vacunación contra el COVID-19 en Francia. Además, los autores muestran que una estrategia de vacunación que combina la priorización de la población y las restricciones de cuarentena conduce a una disminución de 8,5 % en el número total de muertes.

Partiendo de la idea anterior de priorizar a la población en el proceso de vacunación contra el COVID-19, consideramos que los habitantes son libres de decidir a cuales puntos de vacunación acudir y vacunarse. Por lo tanto, consideramos las preferencias de los habitantes con respecto a cada uno de los posibles puntos de vacunación. Vale la pena resaltar que las preferencias de los habitantes que recibirán una dosis de vacunación contra el COVID-19 no han sido consideradas antes en la literatura. Además, en este problema, las autoridades gubernamentales no imponen a los habitantes a cuales puntos de vacunación acudir, pero la ubicación de los puntos de vacunación claramente afecta las decisiones de los habitantes. Este tipo de jerarquía en el proceso de toma de decisiones tampoco se ha considerado antes en la literatura sobre un problema de planificación de la vacunación contra el COVID-19.

2.5 RESUMEN DE LA REVISIÓN DE LA LITERATURA

En resumen, en este estudio se propone un problema de programación bi-nivel bi-objetivo. El tomador de decisiones asociado al nivel superior tiene como objetivo equilibrar los habitantes que asisten a cada punto de vacunación, y también tiene como objetivo ubicar los puntos de vacunación más alejados entre sí para motivar a los habitantes a ir a recibir una dosis de vacuna. Por otro lado, el tomador de decisiones asociado al nivel inferior pretende maximizar las preferencias de los habitantes en cuanto a los puntos de vacunación a los que se acercarán.

Para ilustrar mejor el vacío en la literatura que nuestra investigación está llenando, las principales características de los artículos estrechamente relacionados y mencionados anteriormente se resumen en la Tabla 4.

| Referencia | Objetivo | Bi-objetivo | Bi-nivel | Preferencias | Metaheurística | COVID-19 |
|-------------------------------|------------------------|-------------|----------|--------------|----------------|----------|
| Dasarathy and White (1980) | Max min distancia | | | | | |
| Erkut (1990) | Max min distancia | | | | | |
| Marín (2011) | Min desbalanceo | | | | | |
| Hanjoul and Peeters (1987) | Min costos | | | X | | |
| Cánovas et al. (2007) | Min costos | | | X | | |
| Hansen et al. (2004) | Min costos | | X | X | | |
| Marić et al. (2012) | Min costos | | X | X | PSO, SA, VNS | |
| Maldonado-Pinto et al. (2016) | Min costos | | X | X | EA+PR | |
| Casas-Ramírez et al. (2020) | Min costos | | X | X | EA | |
| Calvete et al. (2020) | Min costos | | X | X | EA | |
| Camacho-Vallejo et al. (2022) | Min max desbalanceo | | X | X | EA | |
| Wang et al. (2022) | Min tiempo de alivio | | | | GA | X |
| Enayati and Özaltın (2020) | Min vacunas utilizadas | | | | | X |
| Tang et al. (2022) | Min costos | X | | | NSGA-II | X |
| | Min distancia de viaje | | | | | |
| Mohammadi et al. (2022) | Min costos | X | | | | X |
| | Min muertes | | | | | |
| Este trabajo | Max min distancia | X | X | X | BONXEA | X |
| | Min desbalanceo | | | | | |

Tabla 4: Resumen de las características consideradas en los problemas de localización de instalaciones estrechamente relacionados

CAPÍTULO 3

DESCRIPCIÓN DEL PROBLEMA

La descripción y formulación del problema de programación bi-objetivo bi-nivel se presentan en esta sección.

Considere la siguiente situación, el Gobierno necesita ubicar un número predefinido de puntos de vacunación para vacunar a los habitantes contra el COVID-19. Los habitantes son libres de asistir al punto de vacunación que prefieran. Esta última decisión afecta directamente al hacinamiento en los puntos de vacunación, lo cual es indeseable. Esta situación se puede modelar bajo un enfoque de programación bi-objetivo bi-nivel. El nivel superior está asociado con el Gobierno y el nivel inferior con los habitantes.

Primero, se definen los conjuntos, parámetros, variables de decisión y auxiliares que intervienen en el modelo. Sea I el conjunto de sitios potenciales de puntos de vacunación que deben ser localizados, y sea J el conjunto de todos los habitantes que necesitan recibir la vacuna. El número predefinido de puntos de vacunación que deben ubicarse se indica con p . La distancia entre un punto de vacunación $i \in I$ y el habitante $j \in J$ está representada por d_{ij} . Además, g_{ij} representa la preferencia que tiene un habitante $j \in J$ por un punto de vacunación $i \in I$.

Las variables de decisión binarias asociadas con el nivel superior son y_i , las cuales toman el valor de 1 cuando un punto de vacunación está ubicado en la ubicación $i \in I$, y 0 si no. Por otro lado, las variables de decisión binarias de nivel inferior se denotan por x_{ij} , si es 1 significa que el habitante $j \in J$ recibe la vacuna en el punto de vacunación $i \in I$, y 0 en otro caso

Las variables continuas auxiliares U y L calculan el número máximo y mínimo de habitantes asignados a cualquier punto de vacunación, respectivamente.

Como se mencionó anteriormente, el Gobierno tiene como objetivo mantener la *sana distancia* y evitar aglomeraciones. Por lo tanto, se consideran dos objetivos en este problema. El primero es equilibrar el número de habitantes asignados a cada punto de vacunación, es decir, que la población se distribuya equitativamente (lo más posible) en los puntos de vacunación abiertos. Esta función objetivo se formula de la siguiente manera:

$$\text{mín } U - L \quad (3.1)$$

La segunda función objetivo apunta a que los puntos de vacunación estén lo más alejadas posible entre sí, es decir, maximizar la distancia mínima entre los puntos de vacunación abiertos. Esta función objetivo se representa de la siguiente manera:

$$\text{máx}_{i,k \in I; i < k} \text{mín} \{d_{ik}y_iy_k\} \quad (3.2)$$

Esta última ecuación se puede linealizar introduciendo una variable continua auxiliar r , que indica la distancia mínima entre cada par de puntos de vacunación abiertos. Por lo tanto, la segunda función objetivo linealizada es:

$$\text{máx } r \quad (3.3)$$

Se deben incluir un conjunto adicional de restricciones para lograr la linealización de la ecuación (3.2). Sea M una constante positiva suficientemente grande, generalmente denominada M grande. Se calcula la distancia de cada par de puntos de vacunación ubicados, y r tomará el valor mínimo entre todos ellos. Esto es lo siguiente:

$$r \leq M(2 - y_i - y_k) + d_{ik} \quad \forall i, k \in I, i < k \quad (3.4)$$

La forma en que funciona la linealización se explica a continuación. La ecuación (3.4) establece límites superiores para el valor de r . Dado que M es una constante grande, se debe tener en cuenta que esta restricción se relaja cuando no se encuentran los puntos de vacunación i -ésimo ni k -ésimo. Cuando se ubican dos puntos de vacunación específicos, entonces el primer término del lado derecho es cero. En este caso, r está acotado por la distancia entre estos dos puntos de vacunación. Debido al signo de la desigualdad dada en la Ec. (3.4) y por la orientación de la función objetivo definida en la Eq. (3.3), el menor valor de las distancias involucradas en las restricciones de límite se asigna a r (que es el máximo posible).

Se considera que hay un número predefinido de puntos de vacunación a abrir. Esto se asegura por la siguiente restricción:

$$\sum_{i \in I} y_i = p \quad (3.5)$$

Como se indicó anteriormente, las variables de decisión de nivel superior son binarias:

$$y_i \in \{0, 1\} \quad \forall i \in I \quad (3.6)$$

Una vez que el Gobierno ha elegido los puntos de vacunación que va a ubicar (x), los habitantes deben ser asignados a dichos puntos de vacunación para vacunarse (y). Dado que estamos suponiendo que la asignación de habitantes a los puntos de vacunación la realiza cada habitante y no la impone el Gobierno, sus preferencias g_{ij} hacia los puntos de vacunación son consideradas. Las preferencias se dan como un ordenamiento de los puntos de vacunación, en la que 1 indica al punto de vacunación menos preferido y $|I|$ al más preferido. Por lo tanto, se persigue la maximización de estas preferencias. La función objetivo asociada con el nivel inferior es la siguiente:

$$\text{máx} \sum_{i \in I} \sum_{j \in J} x_{ij} g_{ij} \quad (3.7)$$

En el nivel inferior hay algunas restricciones asociadas. La primera indica que los habitantes pueden asignarse solo a un punto de vacunación ubicado.

$$x_{ij} \leq y_i \quad \forall i \in I, \forall j \in J \quad (3.8)$$

Otra restricción impuesta es que se debe garantizar que cada habitante reciba una sola vacuna. Es decir, se permite una asignación única de habitantes a puntos de vacunación.

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in J \quad (3.9)$$

Sea n el número total de habitantes que recibirán una vacuna, es decir, $n = |J|$. Por lo tanto, el número máximo y mínimo de habitantes asignados a los puntos de vacunación ubicados se calculan mediante las siguientes restricciones, respectivamente:

$$U \geq \sum_{j \in J} x_{ij} \quad \forall i \in I \quad (3.10)$$

$$L \leq \sum_{j \in J} x_{ij} + n(1 - y_i) \quad \forall i \in I \quad (3.11)$$

Observe que la Ec. (3.10) cuenta el número de habitantes asignados a los puntos de vacunación. Al considerar la desigualdad del tipo *mayor o igual que*, el valor de U que satisface la Ec. (3.10) está limitado por el número máximo de asignaciones a un punto de vacunación específico. Por otra parte, la Ec. (3.11) diferencia entre puntos de vacunación abiertos y cerrados y ayuda a calcular el número mínimo de habitantes asignados a un punto de vacunación específico. En detalle, si un punto de vacunación está ubicado en un sitio potencial específico, entonces el segundo término del lado derecho es cero. Debido al signo de la desigualdad, el valor de L que satisface la Ec. (3.11) es el número mínimo de habitantes asignados. Si no se encuentra un punto de vacunación, los habitantes no se pueden asignar mediante la ecuación (3.8). Entonces, la restricción (3.11) se relaja ya que está limitada por n .

Finalmente, las variables de decisión de nivel inferior son binarias.

$$x_{i,j} \in \{0, 1\} \quad \forall i \in I, \forall j \in J \quad (3.12)$$

El modelo de programación bi-objetivo bi-nivel resultante se define por:

$$(\text{mín } U - L, \quad \text{máx } r)$$

Sujeto a:

$$r \leq M(2 - y_i - y_k) + d_{ik} \quad \forall i, k \in I, i < k$$

$$\sum_{i \in I} y_i = p$$

$$y_i \in \{0, 1\} \quad \forall i \in I$$

Donde se fija y , x resuelve a optimalidad:

$$\text{máx} \sum_{i \in I} \sum_{j \in J} x_{ij} g_{ij}$$

Sujeto a:

$$x_{ij} \leq y_i \quad \forall i \in I, \forall j \in J$$

$$\sum_{i \in I} x_{ij} = 1 \quad \forall j \in J$$

$$U \geq \sum_{j \in J} x_{ij} \quad \forall i \in I$$

$$\begin{aligned}
L &\leq \sum_{j \in J} x_{ij} + n(1 - y_i) && \forall i \in I \\
x_{i,j} &\in \{0, 1\} && \forall i \in I, \forall j \in J
\end{aligned}$$

Vale la pena señalar que si cada uno de los habitantes tiene un orden diferente de preferencias para los puntos de vacunación ubicados, entonces existe una única solución óptima del nivel inferior (Hansen et al. (2004)). Sin embargo, esto puede ser poco realista en nuestro problema debido a la gran cantidad de habitantes en comparación con los puntos de vacunación ubicados. Así, se espera que más de un habitante tenga el mismo ordenamiento de preferencias, lo que lleva a la existencia de múltiples soluciones óptimas para el nivel inferior. En ese caso, diferentes soluciones óptimas de nivel inferior, para una decisión fija de nivel superior, podrían conducir a diferentes valores de la función objetivo de nivel superior. Se puede asumir que los enfoques optimistas o pesimistas manejan ese problema (Dempe (2002), Colson (2007)). En el primer caso, el tomador de decisiones del nivel inferior selecciona la solución óptima que es más conveniente para el nivel superior (enfoque cooperativo), mientras que en el segundo, la solución que afecta más negativamente al nivel superior es seleccionado por el tomador de decisiones de nivel inferior. En este estudio, se sigue el enfoque optimista.

La última suposición es necesaria para tener un problema de programación bi-nivel bien definido. Sin embargo, en el problema bajo estudio, no es sencillo identificar el enfoque optimista ya que estamos optimizando simultáneamente dos funciones objetivo. Se debe tener en cuenta que ambas funciones objetivo se pueden ver como *separable*, es decir, Eq. (3.1) está determinada por las variables de nivel inferior, y la ecuación. (3.3) solo se define mediante variables de nivel superior. Por lo tanto, el enfoque optimista se basa en elegir la solución óptima del nivel inferior que sea más conveniente para equilibrar a los habitantes que asisten a los puntos de vacunación.

Para obtener todas las soluciones óptimas múltiples posibles del nivel inferior, se identifica el problema de separación correspondiente (Plastria (2012)). Es decir, se identifican todas las asignaciones de habitantes a los puntos de vacunación ubicados y se agrega la restricción adecuada. Es decir, está prohibido que todas las asignaciones identificadas se repitan en el problema aumentado. Además, también se incluye una restricción que garantiza que se mantenga el valor de la función objetivo de nivel inferior. Entonces, se resuelve el problema de nivel inferior aumentado. Si no es factible, implica que no existen más soluciones óptimas de nivel inferior. En el otro caso, se registran las soluciones óptimas y al final se selecciona la que ofrece mayor equilibrio para el hacinamiento en los

puntos de vacunación.

ALGORITMO BI-OBJETIVO ANIDADO DE ENTROPÍA CRUZADA

En este capítulo se aborda el algoritmo propuesto para resolver el problema planteado secciones anteriores. También se presenta una breve introducción al concepto de metaheurísticas y un análisis de Entropía Cruzada que nos ayuda a comprender más a fondo el objetivo y funcionamiento de nuestro algoritmo.

4.1 METAHEURÍSTICAS

En optimización existen muchos problemas difíciles de resolver en la práctica, ya que muchos de ellos pertenecen a la clase de problemas NP-Hard, lo que significa que no existe un algoritmo que los resuelva en un tiempo polinomial. Es decir son problemas difíciles y complejos, que no se pueden resolver mediante el uso de métodos tradicionales o exactos. Por ejemplo los problemas de optimización combinatoria. Sin embargo, existen diversas técnicas que nos ayudan a resolver dichos problemas y llegar a una solución. Es el caso de las heurísticas y metaheurísticas. Los métodos tradicionales o exactos aseguran que se encuentre la solución óptima de un problema, mientras que las heurísticas y metaheurísticas no garantizan llegar a la solución óptima, pero pueden obtener una solución casi óptima en un tiempo computacional más corto.

Las heurísticas son estrategias dependientes del problema, lo que significa que aprovechan las particularidades del problema de optimización en el proceso de solución. Por lo tanto, una heurística puede adaptarse a un problema específico, pero la misma heurística no es necesariamente aplicable a otros problemas. Por otro lado, las metaheurísticas son estrategias independientes del problema, que pueden aplicarse a una amplia gama de problemas. En consecuencia, existe una variedad de técnicas heurísticas y metaheurísticas

para resolver problemas de optimización (Toaza and Esztergár-Kiss (2023)).

El concepto de Metaheurística se atribuye a F. Glover en el año de 1986, el concepto de Metaheurística se deriva de dos palabras de origen griego: Meta y Heurística. La primera se puede entender como *más allá de o en un nivel superior*. Mientras que la segunda proviene del vocablo griego *heuriskein* que significa *encontrar, descubrir o hallar*. De tal forma que F. Glover define las Metaheurísticas como un procedimiento de alto nivel que guía y modifica otras heurísticas para explorar soluciones más allá de la simple optimalidad local, Muñoz (2007).

Las metaheurísticas proporcionan una amplia y sólida familia de técnicas o métodos de resolución de problemas, los cuales en muchos casos son creados imitando fenómenos naturales. Aunque es posible que estas técnicas o métodos no encuentren una solución óptima, pueden encontrar una solución casi óptima en un período moderado.

Las metaheurísticas se pueden clasificar en función del tipo de procedimiento que utiliza para llegar a una solución (Pérez (2004)). A continuación se presentan algunas de las clasificaciones de metaheurísticas y algunos de los ejemplos más relevantes.

4.1.1 METAHEURÍSTICAS CONSTRUCTIVAS

Las Metaheurísticas Constructivas son el tipo de metaheurísticas que construyen paso a paso una solución, es decir utilizan procedimientos que tratan de obtener una solución a partir del análisis y selección paulatina de los componentes (atributos o características) que forman dicha solución. Un ejemplo de este tipo de metaheurísticas es la estrategia voraz o greedy, la cual implica la elección del atributo o característica que da mejores resultados inmediatos, sin tener en cuenta una perspectiva más amplia.

Esto se puede ver con el ejemplo del agente viajero que deberá visitar tres ciudades, pero deberá hacerlo recorriendo la menor distancia posible empezando en la ciudad C. Las distancias entre las ciudades A,B,C y D se pueden observar en la Figura 5.

Aplicando un algoritmo greedy al problema del agente viajero planteado, se viajará entre las ciudades eligiendo el siguiente destino la ciudad más cercana a la ciudad en

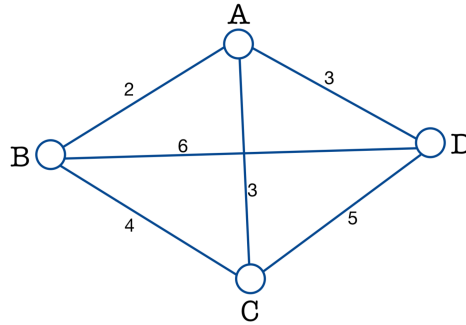


Figura 5: Ejemplo Agente Viajero

donde se encuentra actualmente. De tal modo que, el recorrido del agente viajero es el siguiente: Ciudad C-Ciudad A, Ciudad A- Ciudad B, Ciudad B- Ciudad D sumando en total una distancia de 11. Como se puede observar el algoritmo greedy va seleccionando a cual ciudad moverse sin considerar una visión global del problema, simplemente se elige la ciudad que este más cerca. Sin embargo, el recorrido que se plantea no parece ser el mejor recorrido a pesar de que se ha elegido siempre la mejor opción. Si se escoge por ejemplo el siguiente recorrido: Ciudad C - Ciudad B - Ciudad A - Ciudad D se obtiene una distancia de 9, mejorando la solución obtenida por el algoritmo greedy.

Este tipo de algoritmo suele realizar una búsqueda miope, ya que solo tiene en cuenta la selección del mejor atributo o característica en el momento, y no mira un poco más allá, si las elecciones siguientes serán mejores o peores. Se puede concluir que los algoritmos *greedy* son rápidos pero en general no dan una buena solución.

Una metaheurística constructiva que se ha estado implementando desde hace algunos años y proporciona mejores resultados que un algoritmo tipo *greedy* es el algoritmo GRASP.

El algoritmo *Greedy Randomized Adaptive Search Procedure* (GRASP) es una metaheurística iterativa multiarranque que en cada iteración realiza dos fases perfectamente definidas Figura 6. La primera de ellas construye una solución factible al problema, mediante un procedimiento aleatorio codicioso, que luego en la segunda fase se mejora mediante un procedimiento de búsqueda local. El algoritmo termina cuando se cumple el criterio de paro.

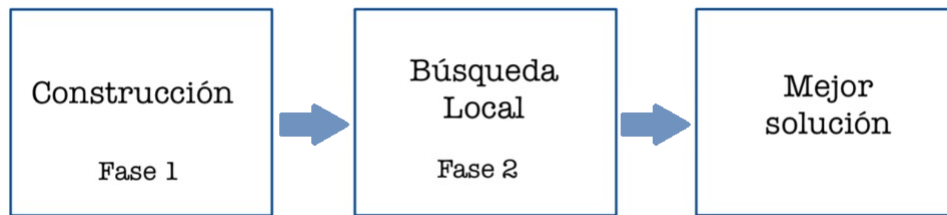


Figura 6: Fases Metaheurística GRASP

4.1.2 METAHEURÍSTICAS DE BÚSQUEDA

Este tipo de metaheurísticas parten de una solución factible y tratan de mejorarla. Usan transformaciones o movimientos para recorrer el espacio de soluciones y explorar las estructuras de otros vecindarios o entornos. Un ejemplo muy importante de este tipo de metaheurísticas son las metaheurísticas de búsqueda local que basa su estrategia en el análisis de soluciones del vecindario o entorno de la solución que realiza el recorrido.

De manera general el algoritmo de Búsqueda Local inicia con una solución factible para el problema, posteriormente se realiza una búsqueda en la vecindad o entorno con el objetivo de encontrar una mejor solución. Si se encuentra una mejor solución, se reemplaza la solución actual por la nueva solución encontrada y continua nuevamente con el proceso, hasta que no se pueda mejorar la solución actual. Sin embargo, al utilizar un algoritmo de Búsqueda Local puede ocurrir que quede atrapado en un óptimo local es decir una solución que ya no puede ser mejorada. Por tanto, un aspecto fundamental es el diseño de los vecindarios o entornos y los procedimientos de búsqueda local que se pretenden utilizar. Algunos ejemplos de estrategias que se pueden aplicar en la Búsqueda Local son:

- Seleccionar el mejor vecino de todos.
(best-improvement rule)
- Seleccionar el primer vecino que mejora la solución.
(first-improvement rule)
- Seleccionar aleatoriamente un vecino como mejora de la solución actual.

Dado que en muchos casos que se aplica Búsqueda Local y solamente se llega a un óptimo local se debe de extender la búsqueda para continuar más allá de los óptimos locales, lo que da lugar a la Búsqueda Global. Las metaheurísticas de búsqueda global incorporan alguna de las estrategias de escapar de los óptimos locales como:

- Volver a iniciar la búsqueda desde otra solución de arranque.
- Permitir movimientos o transformaciones de la solución de búsqueda que no sean de mejora.
- Modificar la estructura de los vecindarios o entornos.

Estas tres estrategias dan lugar, respectivamente, a la metaheurística multi-arranque, metaheurística de entorno variable y metaheurística de búsqueda no monótona.

4.1.3 METAHEURÍSTICAS EVOLUTIVAS

Bajo este enfoque las metaheurísticas usan procedimientos basados en conjuntos de soluciones que evolucionan sobre el espacio de soluciones. Los algoritmos evolutivos toman como base la teoría de la evolución de las especies, la teoría de selección natural de Darwin y la teoría genética de Mendel. Por lo cual se establecen estrategias para realizar la evolución de las soluciones en el espacio de búsqueda llamados poblaciones con el objetivo de encontrar una solución óptima mediante la interacción entre los miembros de la población para crear nuevas soluciones.

Un ejemplo muy representativo de este tipo de metaheurísticas es el Algoritmo Genético. El mecanismo general del algoritmo genético, inicia con un conjunto de individuos es decir un conjunto de soluciones factibles para el problema que se busca resolver, codificadas de una forma específica para el problema y una función de aptitud que permite evaluar a cada individuo. Los individuos se pueden generar aleatoriamente y el objetivo del algoritmo genético es analizar estos individuos de tal manera que se van descartando los menos aceptables hasta elegir al mejor. Cada uno de los individuos es evaluado por la función de aptitud que ayuda a clasificar a los individuos para saber que tan bueno es con respecto a los demás individuos. Después de que se evaluaron los individuos, se conservan los mejores y se continua con la reproducción. En este paso se pueden realizar la cruce y mutación, es decir cambios aleatorios en los individuos. Después de esto la descendencia resultante continua con la siguiente generación, formando una nueva población de individuos, que se somete a una ronda de evaluación de aptitud. Los individuos que han empeorado, o no han mejorado, con los cambios en su código son eliminadas; de nuevo conservando a los mejores. Nuevamente, se seleccionan y se reproducen estos individuos hacia la siguiente generación, y el proceso se repite. Las expectativas son que la aptitud media de la población se incrementará en cada ronda y, por tanto, repitiendo

este proceso, pueden descubrirse soluciones muy buenas del problema. Se muestra en la Figura 7 un diagrama del algoritmo genético.

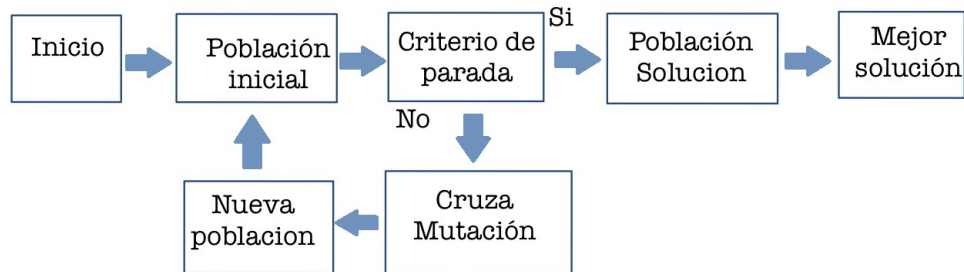


Figura 7: Algoritmo Genético

En la Figura 8 se puede ver un ejemplo de la cruza de dos individuos (Padres) de la población generando dos nuevos individuos (Hijos) para la nueva población. Se selecciona generalmente un punto de cruza que divide en dos a cada uno de los Padres y se combinan dichas partes creando nuevos individuos. Por otra parte en la Figura 9 se observa un ejemplo de mutación a un individuo, se selecciona un gen que será mutado y dará como resultado un nuevo individuo para la población.

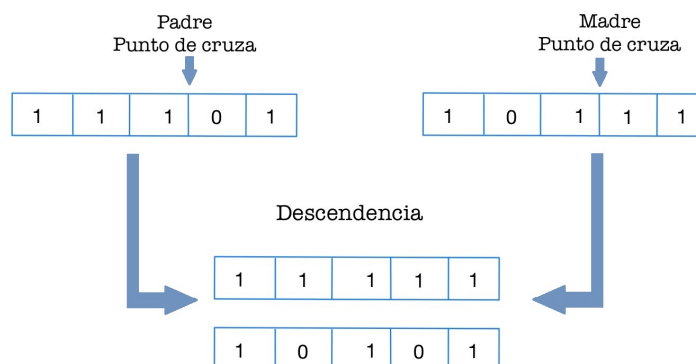


Figura 8: Cruza

Existen diversas formas de combinar la información proporcionada por los individuos de la población para hacerla evolucionar, por lo tanto, existe una gran diversidad de metaheurísticas evolutivas.

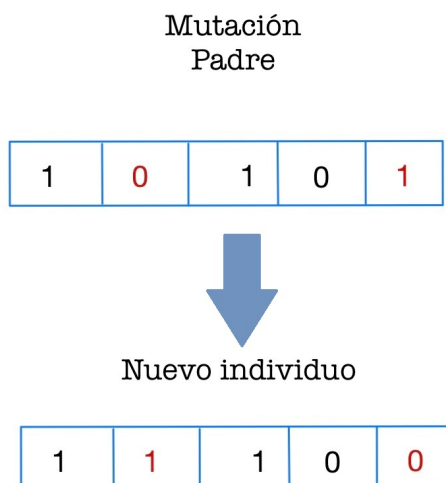


Figura 9: Mutación

4.1.4 METAHEURÍSTICAS DE RELAJACIÓN

Las metaheurísticas de relajación utilizan procedimientos que relajan el problema original. Es decir, modificaciones en el modelo que hacen al problema más fácil de resolver. Por ejemplo, una relajación muy común de un problema es eliminar, debilitar o modificar restricciones u objetivos del problema original. Un caso muy común donde se recurre a la opción de relajación de problema es en programación lineal entera, donde la relajación lineal consiste en ignorar la restricción de que las variables sean enteras. Después de que se aplica la relajación del problema se puede usar métodos conocidos para resolverlo, en este caso por ejemplo se puede utilizar el método Simplex para encontrar una solución entera muy próxima a la solución del problema relajado. Dentro de las metaheurísticas que recurren a la relajación de restricciones se encuentran la Relajación Lagrangiana, que en términos generales busca remover la restricción más difícil de trabajar y colocarla en la función objetivo con la intención de resolver el problema de una forma más sencilla.

En conclusión, se puede decir que no existe una clasificación formal respecto a la gran diversidad de metaheurísticas desarrolladas hasta el momento, en esta pequeña sección se expusieron de manera general algunas de las metaheurísticas más comunes y utilizadas en la literatura.

Es importante destacar que en los últimos años la aplicación de heurísticas y metaheurísticas ha crecido de forma exponencial, con la ayuda de estos métodos de resolución de problemas hoy en día problemas que antes se pensaba que eran imposibles de resolver

con la aplicación de este tipo de herramientas se puede obtener una buena solución.

En Camacho-Vallejo et al. (2023) se realiza un análisis bibliométrico sobre las diferentes metaheurísticas que se han desarrollado y aplicado a lo largo de los últimos años para resolver problemas de programación bi-nivel, que actualmente este tipo de problemas se ha vuelto muy popular para modelar situaciones reales. En nuestro caso se plantea un problema bi-nivel bi-objetivo, el cual lo resolvemos por medio de una metaheurística ya muy conocida: Entropía Cruzada. Dicha metaheurística y el algoritmo que utilizamos para resolver nuestro problema se explica a continuación.

4.2 ALGORITMO DE ENTROPÍA CRUZADA

El algoritmo de Entropía Cruzada (XE) es un método iterativo con dos fases principales. Primero, crea una muestra aleatoria de soluciones mediante un mecanismo específico. Luego, actualiza los parámetros involucrados en el generador de soluciones. Esto último se logra considerando las características de la muestra aleatoria anterior para crear soluciones mejoradas en la siguiente iteración.

El algoritmo de entropía cruzada tiene una historia relativamente corta en problemas de optimización. Rubinstein desarrolló el algoritmo de entropía cruzada como un método para estimar la probabilidad de eventos raros en redes estocásticas complejas (Rubinstein (1997)). Unos años más tarde, el algoritmo de entropía cruzada se aplicó por primera vez en el contexto de la optimización combinatoria, lo que inspiró su aplicación en diversas aplicaciones en investigación operativa: localización de instalaciones capacitadas; Caserta and Rico (2009), Hosseini-zhad et al. (2015), Casas-Ramírez et al. (2018), cadena de suministro de circuito cerrado; Wang et al. (2016), cadena de suministro sostenible; Kalantari and Hosseini-zhad (2022), programación de taller; Santosa et al. (2011), distribución de las instalaciones; Ning and Li (2018), sistemas de distribución de agua; Perelman and Ostfeld (2007), entre otros.

Descrito a detalle, el algoritmo de entropía cruzada consiste en la ejecución repetida de los siguientes pasos (De Boer et al. (2005)):

1. Genera una muestra aleatoria de soluciones en función de una función de distribución de probabilidad preespecificada.

2. Se utiliza la muestra aleatoria para seleccionar un conjunto élite, que se utiliza para modificar los parámetros de la función de distribución de probabilidad buscando producir una muestra con mejores valores de la función objetivo en la siguiente iteración.

La idea general es obtener una secuencia de valores de parámetros que se actualicen de manera adecuada de una iteración a la siguiente. Si tiene éxito, la búsqueda converge al óptimo global, o al menos, a un óptimo local.

El algoritmo de entropía cruzada se ha aplicado con éxito en problemas discretos relacionados con el aquí estudiado. Por ejemplo, en Caserta and Rico (2009) se estudia un problema de ubicación-asignación. En el se consideran la capacidad y los costos fijos de las instalaciones, y se persigue la minimización del costo total. Se propone una formulación de un problema entero mixto no lineal y se implementa el método de entropía cruzada para resolver el problema. Ocurren tres etapas diferentes, cada una con diferentes funciones de densidad. En la primera etapa, se resuelve un problema de ubicación con cobertura utilizando una función normal multivariada. En la segunda etapa, la asignación se realiza siguiendo una función de densidad multinomial. Luego, en la tercera etapa, se resuelve un problema continuo de ubicación de una sola instalación. El enfoque propuesto obtiene buenos resultados en comparación con las soluciones obtenidas por GAMS. Además, en Hosseininezhad et al. (2015), se presenta un algoritmo híbrido para el problema de dimensionamiento de lotes de múltiples productos y períodos múltiples capacitados. En ese problema, muchas mercancías compiten por el espacio y los recursos limitados en cada período. El enfoque de solución propuesto considera una relajación lagrangiana y aplica una metaheurística basada en entropía cruzada para la versión no capacitada del problema.

Además, el algoritmo de entropía cruzada se ha aplicado para resolver problemas de ubicación de instalaciones modelados como programas bi-nivel. Por ejemplo, en Casas-Ramírez et al. (2018) se estudia un problema bi-nivel. En este trabajo, el nivel superior está asociado a una empresa que ubica instalaciones capacitadas con el fin de minimizar los costos de ubicación y distribución. En el nivel inferior, los clientes buscan maximizar sus preferencias de ser asignados a sus instalaciones preferidas. Un algoritmo de entropía cruzada está diseñado para obtener soluciones de nivel superior; mientras que el nivel inferior se resuelve a través de tres enfoques diferentes: mediante un procedimiento aleatorio adaptativo, mediante un procedimiento basado en un costo de arrepentimiento y mediante un método exacto. El algoritmo de entropía cruzada construye un conjunto de

soluciones de nivel superior y para cada una de ellas, se resuelve el problema de nivel inferior resultante. En otras palabras, se utiliza el enfoque anidado clásico, de algoritmos diseñados para resolver problemas de dos niveles.

Existen otras aplicaciones de entropía cruzada en el contexto de la programación multiobjetivo. En Unveren and Acan (2007) se usa para encontrar fronteras eficientes para problemas con múltiples funciones objetivo multimodales de variables continuas. Para obtener los frentes de Pareto, el conjunto de soluciones no dominadas se divide en grupos y se aplica un procedimiento de entropía cruzada a cada una de ellas. Este método es muy similar al propuesto por Kroese et al. (2006). En Caballero et al. (2015) se propone un método de entropía cruzada adaptado para aproximar el frente de Pareto. Para lograr esto último, se forma una cuadrícula uniforme en el espacio de los objetivos. Cada celda de la cuadrícula contiene su propia población de élite (con soluciones no dominadas). Como consecuencia, los vectores de probabilidad asociados se actualizan, respectivamente. Al final del proceso, la aproximación del frente de Pareto consiste en la unión del conjunto de soluciones no dominadas en cada celda. Recientemente, en Kalantari and Hosseini-nezhad (2022) se diseñó un método de entropía cruzada multiobjetivo para resolver una cadena de suministro de alimentos sostenible. Cuando se genera el conjunto de soluciones, se considera una matriz dominante para identificar la mejor solución.

4.3 ALGORITMO PROPUESTO

Como se mencionó anteriormente, el problema estudiado en esta investigación se aborda mediante un algoritmo bi-objetivo anidado de Entropía Cruzada (BONXEA, por las siglas en inglés de: Bi-objective Nested Cross Entropy Algorithm). Para el diseño de BONXEA se consideran los métodos de entropía cruzada propuestos en Casas-Ramírez et al. (2018) y Caballero et al. (2015). En el primero, se estudia un problema de ubicación de instalaciones discretas de dos niveles con capacidades; mientras que, en el segundo, se analizan diferentes problemas multiobjetivo. Para inicializar nuestro BONXEA propuesto, se debe definir una cuadrícula en el espacio de objetivos. Dado que estamos considerando dos funciones objetivo, se crea una cuadrícula de 3×3 . El número de celdas en la grilla corresponde al número de objetivos considerados más uno. La lógica es encontrar las mejores soluciones para cada objetivo y para la región central de la frontera de Pareto. Para construir la cuadrícula, se necesitan los valores mínimo y máximo para cada función objetivo. Sean F_1 , F_2 y F_3 las funciones objetivo consideradas en BONXEA, donde F_1 corresponde a la función objetivo asociada a la distancia entre los puntos de vacunación,

F_2 está asociada al número de habitantes en cada punto de vacunación, y F_3 es la suma ponderada de la estandarización de F_1 y F_2 .

Después de eso, se crea una población de N soluciones de nivel superior (y). Para cada una de estas soluciones, el problema de nivel inferior se resuelve de manera óptima (atributo anidado de BONXEA). Ahora, se pueden evaluar las funciones objetivo de nivel superior. Las soluciones en la población inicial se agrupan en tres subpoblaciones, cada una de ellas asociada a F_1 , F_2 y F_3 , respectivamente. Se forman tres muestras de tamaño d que contienen las mejores soluciones, en términos de la calidad de la función objetivo correspondiente. Para cada muestra, las frecuencias de ubicación de un punto de vacunación se actualizan en la fase de construcción. Por lo tanto, se crean nuevas subpoblaciones de soluciones basadas en los vectores de frecuencia actualizados para cada una de las tres muestras. El procedimiento se repite hasta alcanzar el número máximo de iteraciones. Las soluciones no dominadas y sus respectivos valores de función objetivo se almacenan y se dan como resultado de nuestro algoritmo.

A continuación se presenta el pseudocódigo del BONXEA:

Algoritmo 1: BONXEA

Input: $N, d, MaxIter$

Output: P

$Iter \leftarrow 0$

$P \leftarrow \emptyset$ (Frontera Pareto)

$\gamma^k \leftarrow 0,5$ (Frecuencias iniciales)

Definir una cuadrícula 3×3 que limita F_1 y F_2

mientras $Iter < MaxIter$ **hacer**

para $i : 1..N$ **hacer**

$\bar{y}_i \leftarrow$ Crear una solución del nivel superior basada en γ^k

$y_i \leftarrow$ Reparar la solución de nivel superior (\bar{y}_i)

$x_i \leftarrow$ Resolver de forma óptima el nivel inferior (y_i)

 Evaluar $F_1(x_i), F_2(y_i)$ y $F_3(x_i, y_i)$

fin

para cada celda no vacía c de la cuadrícula 3×3 **hacer**

 Identificar las d soluciones élite Actualizar el vector de frecuencias γ^k

 Obtener P_c de acuerdo con F_1 y F_2

$P \leftarrow P \cup P_c$

fin

 Actualizar P con los puntos no dominados

$Iter \leftarrow Iter + 1$

fin

4.3.1 DESCRIPCIÓN DEL BONXEA

A continuación se describe de forma más detallada la descripción de cada una de las etapas del algoritmo propuesto para resolver el problema bajo estudio.

Codificación de la solución

El algoritmo propuesto explora soluciones del nivel superior y para cada una de ellas se obtiene la reacción óptima del nivel inferior. En este caso, una solución de nivel superior en la iteración k está representada por el vector binario $y^k = (y_1^k, \dots, y_{|I|}^k)$, en el que un 1 indica que un punto de vacunación está ubicado en el sitio $i \in I$, y 0 en caso contrario.

Creación de soluciones

En la iteración k , se crean N soluciones aleatorias pero basadas en un vector de frecuencia

$\gamma^k = (p_1^k, \dots, p_{|I|}^k)$, que indica la probabilidad de ubicar una instalación en cada sitio. Al comienzo de la BONXEA, cuando $k = 1$, las probabilidades se establecen en 0,5, es decir, $\gamma^1 = (0,5, \dots, 0,5)$. A medida que el algoritmo continúa, los vectores de frecuencia γ^k se actualizarán de una iteración a la siguiente. Eso nos llevará a privilegiar las instalaciones más convenientes para ubicar a los puntos de vacunación.

Reparación de soluciones

En el caso de que una solución de nivel superior $y^k = (y_1^k, \dots, y_{|I|}^k)$ en la iteración k no haya localizado los p puntos de vacunación requeridos, se realiza una rutina para reparar la solución, es decir, para lograr la factibilidad. En concreto, se localiza el punto de vacunación no localizado asociado a la frecuencia más alta en el vector $k = (p_1^k, \dots, p_{|I|}^k)$. Si es necesario ubicar más de un punto de vacunación para lograr el p requerido, el procedimiento de reparación continúa de manera análoga. El procedimiento de reparación se detiene cuando se localizan los p puntos de vacunación.

Solución óptima del problema del nivel inferior

Recordemos que una vez que se conocen los puntos de vacunación ubicados, los habitantes se asignan de manera óptima a estas instalaciones. La asignación debe tener en cuenta las preferencias de los habitantes con respecto a los puntos de vacunación. Dado que no se considera capacidad alguna en los puntos de vacunación, todos los habitantes pueden asignarse a las instalaciones ubicadas de su preferencia.

Por lo tanto, el problema de nivel inferior se puede resolver de manera óptima utilizando una matriz auxiliar de preferencias ordenadas, en la que los índices de las instalaciones se ordenan de manera decreciente en función de las preferencias de cada habitante. Es decir, cada columna de la matriz auxiliar contiene los índices ordenados de los puntos de vacunación. Una vez que se construye la matriz auxiliar, el problema de nivel inferior se puede resolver de manera óptima utilizando un procedimiento sencillo. Para cada habitante se identifica su punto de vacunación ubicado de preferencia y se realiza la asignación. Siguiendo el procedimiento descrito, se logra la asignación óptima de habitantes a los puntos de vacunación (ver Marić et al. (2012), Calvete et al. (2020)). Por lo tanto, se obtiene una solución óptima de nivel inferior para una solución fija dada de nivel superior, es decir, $x^{*k}(y^k)$. En aras de la simplicidad, se denota por x^k .

Evaluación de las funciones objetivo

Una vez obtenida una solución de nivel superior y_i^k y su correspondiente respuesta óptima

de nivel inferior x^k , se pueden evaluar las funciones objetivo consideradas en el BONXEA, es decir, F_1 , F_2 y F_3 . Recuerde que F_1 viene dado por la Ec. (1), F_2 viene dada por la Ec. (3), y F_3 viene dado por $\alpha NF_1 + (1 - \alpha)NF_2$, donde NF_1 y NF_2 son la estandarización de F_1 y F_2 , respectivamente.

Creación de las subpoblaciones en base a la dominancia

En primer lugar, tomamos las poblaciones de élite para cada una de las tres celdas consideradas en la cuadrícula de 3×3 . Cada población élite está asociada a cada una de las tres funciones objetivo consideradas, esto es, F_1 , F_2 y F_3 . Por lo tanto, se seleccionan las mejores d soluciones con respecto al objetivo F_1 y se evalúa el objetivo F_2 para tales soluciones. Se almacenan las soluciones no dominadas en esa celda, que se indican con y^{1k} , y^{2k} , ..., y^{dk} . Posteriormente, se repite el mismo procedimiento para identificar las soluciones no dominadas en las dos celdas restantes, es decir, respecto a F_2 y F_3 , respectivamente. Los autores destacan que para identificar las soluciones no dominadas se utilizan únicamente los objetivos F_1 y F_2 por ser los objetivos reales en el problema en estudio. Recuerde que el objetivo F_3 es una función objetivo auxiliar para identificar la parte central del frente de Pareto.

Bajo este enfoque se obtienen tres frentes de Pareto, uno para cada una de las tres celdas consideradas dentro de la grilla. De tal manera que, cada subpoblación evoluciona independientemente de las otras dos subpoblaciones restantes.

Actualizando el vector de probabilidades γ

En la iteración k , los vectores γ_l^k , con $l = 1, 2, 3$, son vectores que contienen la frecuencia con la que se encuentra un punto de vacunación específico en las soluciones de la subpoblación élite correspondiente. El uso de vectores γ_l^k es para almacenar información de los componentes de las soluciones élite, es decir, para identificar la repetición de puntos de vacunación en las soluciones de buena calidad. Esto se hace para favorecer la ubicación de estos puntos de vacunación en las próximas soluciones aleatorias. Para cada subpoblación l las probabilidades se obtienen de la siguiente manera:

$$\gamma_l^k = \left(\frac{\sum_{s=1}^d y_1^{sk}}{d}, \frac{\sum_{s=1}^d y_2^{sk}}{d}, \dots, \frac{\sum_{s=1}^d y_{|I|}^{sk}}{d} \right)$$

Creación de nuevas soluciones

Para generar nuevas soluciones de cada subpoblación l en la iteración k , se utilizan vectores γ_i^k para determinar si se ubica un punto de vacunación o no. Como se mencionó anteriormente, el procedimiento busca privilegiar los puntos de vacunación más convenientes en cuanto a la información proporcionada por las soluciones elite de cada subpoblación. Para las nuevas soluciones creadas, se genera un vector aleatorio de tamaño $|I|$, si el número aleatorio i es menor que su correspondiente probabilidad γ_i^{lk} , entonces, se ubica ese punto de vacunación.

Si aparece un 0 en una posición dentro de los vectores γ_l , significa que en las nuevas soluciones creadas, un punto de vacunación no debe estar ubicado en ese sitio específico. De lo contrario, cuando aparece una probabilidad positiva, existe la posibilidad de ubicar un punto de vacunación en ese sitio potencial. En caso de que la probabilidad sea de 1, se debe ubicar un punto de vacunación en ese sitio específico para todas las nuevas soluciones creadas. Siguiendo este criterio, el algoritmo converge iterativamente a soluciones de buena calidad.

Criterio de parada

Los criterios de parada se establecen como un número máximo de iteraciones y un número predeterminado de iteraciones sin actualizar el conjunto de soluciones no dominado en cada subpoblación.

Frente de Pareto final

En cada iteración del BONXEA se obtiene un frente de Pareto para cada una de las tres subpoblaciones. Además, las fronteras obtenidas para cada celda de la cuadrícula se actualizan en cada iteración tomando los frentes anteriores.

Al final del BONXEA, el conjunto de puntos no dominados obtenidos para cada subpoblación de elite se fusionan para formar el frente de Pareto final. Algunas soluciones no dominadas en una celda están dominadas por algunas soluciones del frente de Pareto de otra celda. Por lo tanto, al final, solo se informan las soluciones no dominadas de las tres celdas.

CAPÍTULO 5

EXPERIMENTACIÓN COMPUTACIONAL

En este capítulo se presenta la experimentación computacional del algoritmo propuesto y se presentan los resultados obtenidos.

Para medir el rendimiento del algoritmo propuesto y su capacidad para aproximar el frente de Pareto, se llevó a cabo una intensa experimentación computacional sobre un conjunto de instancias adaptadas de la literatura. Todos los experimentos computacionales se realizaron en una computadora con un procesador Intel Core i7-4790 de 3,60 GHz con 32,00 GB de RAM con Windows 8.1. y el código BONXEA se codificó con el software FICO Xpress 8.11.

El conjunto de instancias se adaptó de las instancias utilizadas para un problema de ubicación de instalaciones discretas con asignación equilibrada de clientes, que se propuso en Marín (2011). En dicho problema se incluyen los parámetros referentes al tamaño de instancia, es decir, número de instalaciones posibles y número de clientes (puntos de vacunación y habitantes en nuestro caso, respectivamente). Además, se incluyeron las distancias entre puntos de vacunación y habitantes. Los parámetros faltantes que son necesarios para nuestro problema se han completado como se describe a continuación. En particular, faltaban datos relacionados con las preferencias de los habitantes hacia los puntos de vacunación. Estas preferencias se generaron a través del proceso descrito en Casas-Ramírez and Camacho-Vallejo (2017). Básicamente, este proceso tiene en cuenta los costos existentes y genera costos ficticios modificados para clasificarlos en orden ascendente. En base a esto se generan preferencias, logrando que el punto de vacunación más cercano no sea siempre el más preferido por un habitante.

Para el problema estudiado en esta investigación, las instancias deben contener los datos sobre el número de puntos de vacunación, el número de habitantes, el número

predefinido de puntos de vacunación que se deben ubicar, las preferencias y las distancias de los habitantes hacia los puntos de vacunación.

En la experimentación computacional denotamos una instancia de la siguiente manera: $m - n - p - v$, como se hace en Marín (2011). Por ejemplo, la instancia 30-50-3-10 significa que hay 30 puntos de vacunación potenciales, 50 habitantes, 3 puntos de vacunación para abrir y solo 10 pares de distancia que se han intercambiado. Este último se refiere a un parámetro que agrega complejidad a la instancia.

En total, se generaron 45 instancias diferentes combinando diferentes números de los siguientes parámetros:

- I) Los puntos de vacunación a abrir ($|I|$), se establecieron a partir del conjunto $\{20, 30, 50, 100\}$;
- II) El número de habitantes, varían entre $\{20, 30, 50, 100\}$;
- III) El número de puntos de vacunación para abrir p , que fueron entre $\{3, 4, 6, 10\}$; y
- IV) El nivel de perturbación v , que se utilizó para modificar los valores de una instancia sin perder la estructura, dichos valores se eligieron entre $\{10, 30, 50, 100\}$.

Debido a la estocasticidad inherente del BONXEA, se realizaron 100 corridas para cada una de las instancias. A continuación se muestran los resultados obtenidos de la experimentación en la Tabla 5. En la Tabla 5 la primera columna indica la etiqueta de la instancia a la que nos referimos. Las columnas restantes están asociadas con las soluciones no dominadas. La segunda y cuarta columnas muestran el número mínimo y máximo de soluciones no dominadas obtenidas por el BONXEA, respectivamente. La tercera columna muestra el promedio de las soluciones no dominadas obtenidas entre las 100 corridas del algoritmo. Las columnas quinta y sexta muestran los valores mínimo y máximo para la función objetivo F_1 , respectivamente. Recordando que la función objetivo F_1 se refiere al saldo de los habitantes en los Puntos de vacunación abiertos. Finalmente, en la séptima y octava columna se muestran los valores mínimo y máximo obtenidos para la segunda función objetivo F_2 . Cabe señalar que la segunda función objetivo F_2 busca maximizar la distancia mínima entre dos puntos de vacunación abiertos. Con estas últimas cuatro columnas, podemos identificar los extremos del frente de Pareto aproximado que se obtuvo.

De la Tabla 5 se puede ver que el BONXEA siempre obtuvo soluciones no dominadas en cada una de las instancias que se trabajaron. El mayor número de soluciones

| Instancia | Min | Promedio | Max | Min F_1 | Max F_1 | Min F_2 | Max F_2 |
|---------------|-----|----------|-----|-----------|-----------|-----------|-----------|
| 20-20-3-1 | 2 | 3.5 | 4 | 1 | 4 | 29 | 37 |
| 20-20-3-10 | 1 | 2.6 | 4 | 1 | 9 | 16 | 18 |
| 20-20-3-50 | 2 | 2 | 2 | 1 | 17 | 7 | 18 |
| 20-20-6-1 | 2 | 2.5 | 4 | 1 | 3 | 9 | 13 |
| 20-20-6-10 | 3 | 3.5 | 5 | 1 | 5 | 7 | 14 |
| 20-20-6-50 | 2 | 2.9 | 4 | 1 | 4 | 5 | 10 |
| 30-30-3-1 | 2 | 4.1 | 7 | 0 | 9 | 48 | 71 |
| 30-30-3-10 | 2 | 2.9 | 4 | 8 | 17 | 26 | 28 |
| 30-30-3-100 | 2 | 3.6 | 5 | 0 | 8 | 23 | 26 |
| 30-30-4-10 | 2 | 3 | 4 | 10 | 13 | 10 | 27 |
| 30-30-4-100 | 2 | 3.2 | 6 | 1 | 4 | 20 | 26 |
| 30-50-3-10 | 1 | 2.1 | 4 | 1 | 17 | 4 | 28 |
| 30-50-3-50 | 2 | 2.9 | 4 | 1 | 6 | 25 | 27 |
| 30-50-3-100 | 1 | 2.9 | 4 | 4 | 14 | 1 | 26 |
| 30-50-6-10 | 3 | 3.5 | 5 | 1 | 8 | 11 | 23 |
| 30-50-6-50 | 3 | 4.1 | 6 | 2 | 14 | 15 | 22 |
| 30-50-6-100 | 2 | 4.7 | 6 | 1 | 13 | 6 | 22 |
| 30-50-10-10 | 3 | 3.7 | 5 | 5 | 6 | 14 | 19 |
| 30-50-10-50 | 2 | 3.4 | 4 | 2 | 7 | 7 | 14 |
| 30-50-10-100 | 2 | 3.5 | 5 | 4 | 8 | 7 | 13 |
| 30-100-3-50 | 2 | 4.1 | 6 | 4 | 10 | 16 | 28 |
| 30-100-3-100 | 2 | 3.4 | 6 | 4 | 12 | 25 | 28 |
| 30-100-6-10 | 3 | 4 | 6 | 4 | 12 | 10 | 24 |
| 30-100-6-50 | 2 | 4.6 | 9 | 3 | 21 | 9 | 23 |
| 30-100-6-100 | 4 | 5.2 | 7 | 3 | 13 | 8 | 22 |
| 30-100-10-10 | 2 | 3.5 | 5 | 9 | 16 | 9 | 19 |
| 30-100-10-50 | 2 | 3.7 | 5 | 7 | 15 | 9 | 18 |
| 30-100-10-100 | 3 | 4.7 | 7 | 6 | 15 | 4 | 18 |
| 50-50-3-10 | 1 | 2.5 | 6 | 2 | 18 | 1 | 41 |
| 50-50-3-50 | 3 | 3.7 | 5 | 1 | 5 | 36 | 48 |
| 50-50-3-100 | 1 | 2.8 | 5 | 1 | 1 | 47 | 47 |
| 50-50-6-10 | 3 | 4.6 | 6 | 1 | 11 | 26 | 39 |
| 50-50-10-10 | 2 | 3.3 | 5 | 2 | 7 | 21 | 39 |
| 50-50-10-50 | 2 | 3.7 | 6 | 2 | 8 | 26 | 33 |
| 50-50-10-100 | 2 | 3.3 | 6 | 4 | 12 | 2 | 34 |
| 50-100-3-10 | 3 | 3.5 | 6 | 1 | 14 | 44 | 48 |
| 50-100-3-50 | 2 | 3.9 | 7 | 1 | 12 | 37 | 47 |
| 50-100-3-100 | 2 | 4.6 | 7 | 7 | 23 | 46 | 47 |
| 50-100-6-50 | 3 | 4.2 | 8 | 3 | 16 | 31 | 39 |
| 50-100-6-100 | 2 | 4 | 5 | 3 | 12 | 33 | 42 |
| 50-100-10-50 | 2 | 3.3 | 4 | 2 | 6 | 23 | 32 |
| 50-100-10-100 | 3 | 4 | 5 | 3 | 12 | 27 | 35 |
| 100-100-3-1 | 1 | 2.6 | 4 | 1 | 5 | 102 | 105 |
| 100-100-3-10 | 1 | 3.5 | 8 | 1 | 49 | 3 | 95 |
| 100-100-3-50 | 2 | 3.8 | 5 | 1 | 20 | 93 | 97 |

Tabla 5: Resumen de las soluciones obtenidas por el BONXEA

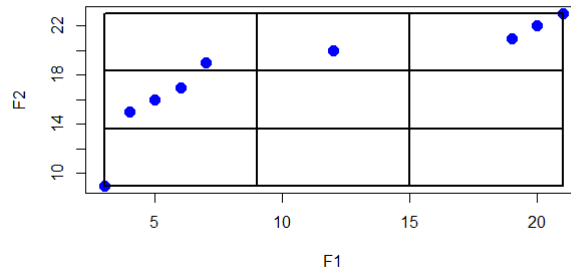


Figura 10: Instancia 30-100-6-50

no dominadas que se obtuvo es de 9 soluciones, las cuales se obtuvieron en la instancia 30-100-6-50. También se puede observar que en promedio, hay al menos dos soluciones no dominadas para cada instancia. Recordando que F_1 se refiere al equilibrio de habitantes. Por lo tanto el caso de equilibrio perfecto entre habitantes y los Puntos de vacunación se podría tener $F_1 = 0$. Si esto no es posible, el rango en el que varía F_1 es muy limitado. Es decir, no hay muchas soluciones no dominadas con valores grandes de F_1 . Por ejemplo, para las instancias 30-30-3-1 y 30-30-3-100, se encuentra un equilibrio perfecto, es decir, hay exactamente 10 habitantes que van a cada Punto de vacunación abierto. Sin embargo, la instancia 100-100-3-10 muestra un resultado sorprendente. Se consideran 100 clientes y el peor saldo reportado es 49. Esto implica que hay mucha disparidad entre la aglomeración en los puntos de vacunación abiertos. Sin embargo, después de revisar esta solución, se puede observar que el saldo de 49 corresponde al mejor valor para F_2 . Es decir, los puntos de vacunación muy distantes hacen que los habitantes se concentren mayoritariamente en un punto de vacunación central. Eso ocurre solo en ese caso en particular, y debido a la distribución espacial de los habitantes y los puntos de vacunación potenciales. También es importante aclarar que los valores obtenidos de F_2 son números enteros porque así están definidos en los datos, no se están calculando distancias euclidianas entre habitantes y los puntos de vacunación.

A partir de los resultados obtenidos, también se puede identificar que el algoritmo propuesto encuentra soluciones que nos permiten aproximar correctamente los frentes de Pareto de este problema jerarquizado. Para ilustrar lo anterior, se muestran las gráficas de los frentes de Pareto obtenidos de cada una de las instancias mencionadas anteriormente.

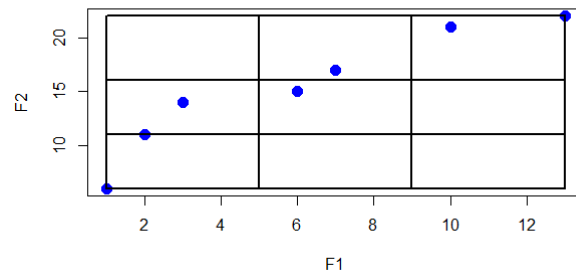


Figura 11: Instancia 30-50-6-100

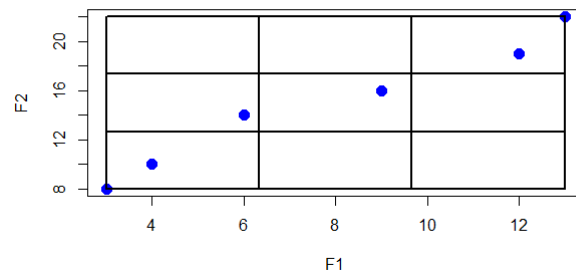


Figura 12: Instancia 30-100-6-100

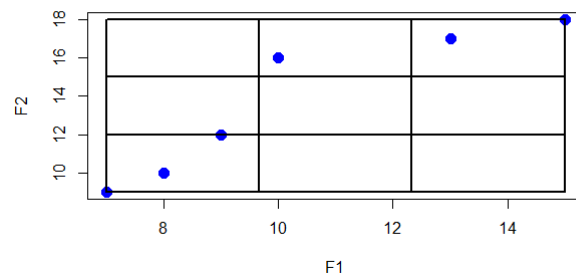


Figura 13: Instancia 30-100-10-50

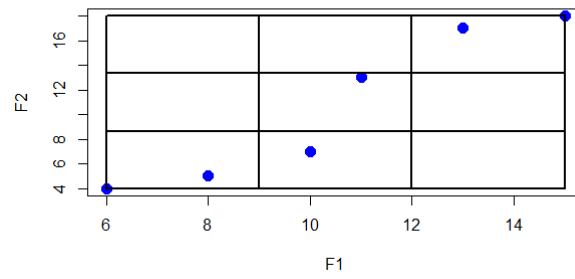


Figura 14: Instancia 30-100-10-100

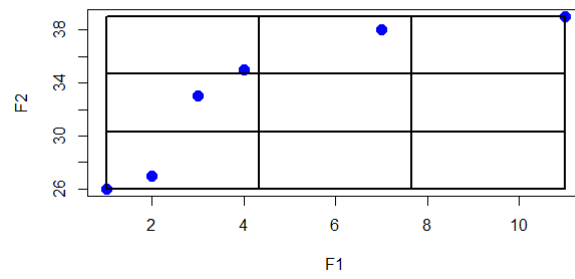


Figura 15: Instancia 50-50-6-10

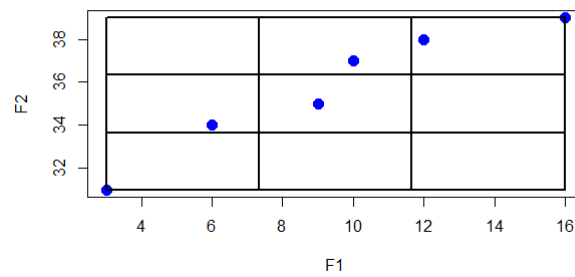


Figura 16: Instancia 50-100-6-5

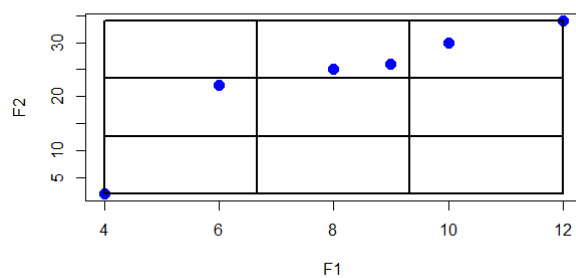


Figura 17: Instancia 50-100-10-100

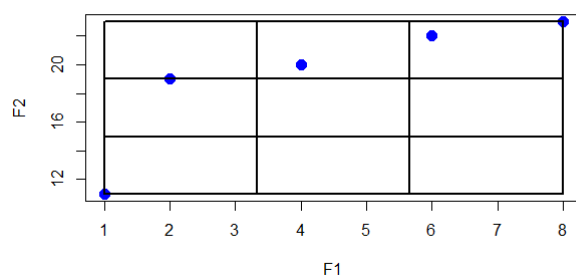


Figura 18: Instancia 30-50-6-10

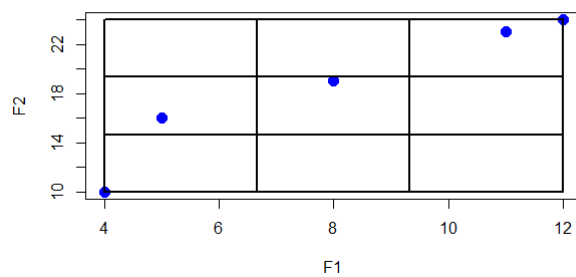


Figura 19: Instancia 30-100-6-10

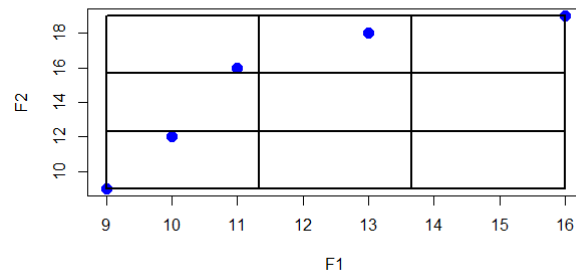


Figura 20: Instancia 30-100-10-10

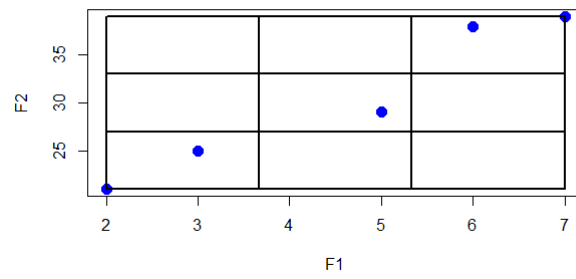


Figura 21: Instancia 50-50-10-10

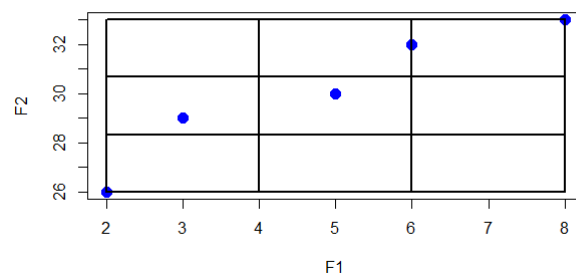


Figura 22: Instancia 50-50-10-50

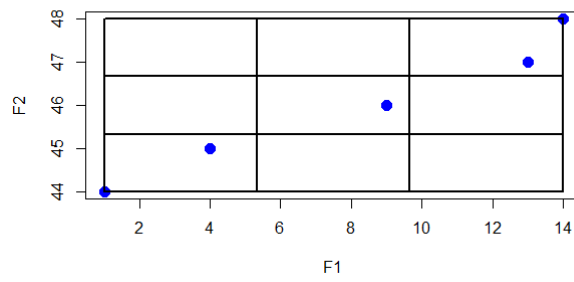


Figura 23: Instancia 50-100-3-10

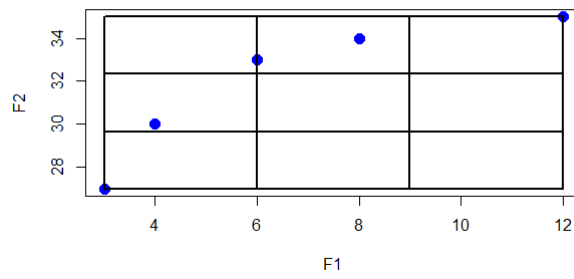


Figura 24: Instancia 50-100-10-50

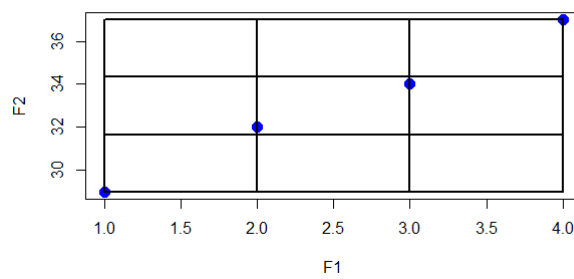


Figura 25: Instancia 20-20-3-1

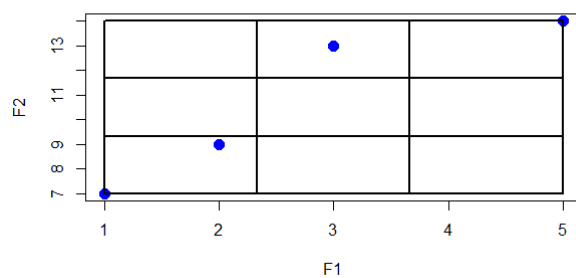


Figura 26: Instancia 20-20-6-10

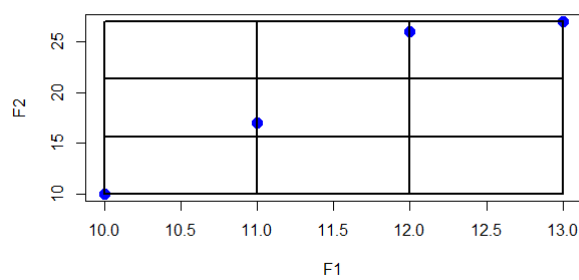


Figura 27: Instancia 30-30-4-10

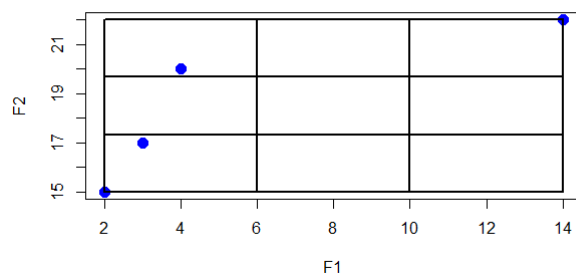


Figura 28: Instancia 30-50-6-50

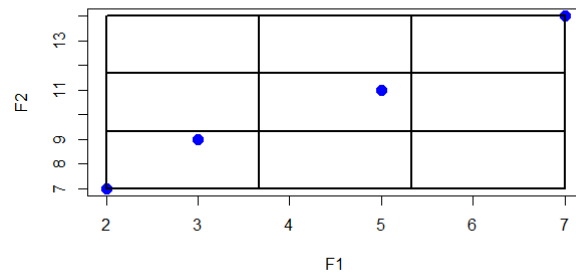


Figura 29: Instancia 30-50-10-50

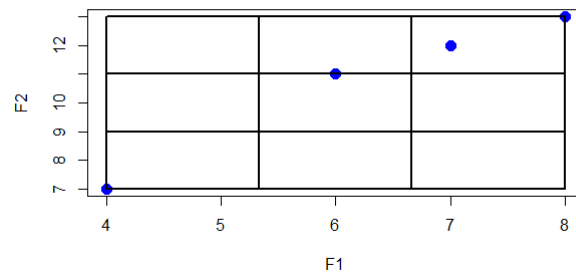


Figura 30: Instancia 30-50-10-100

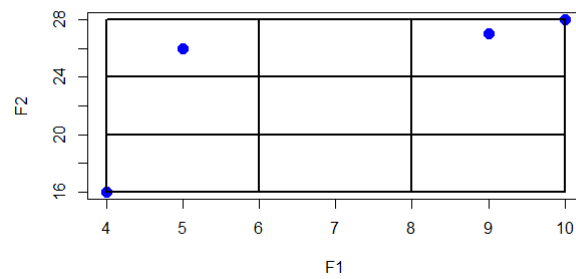


Figura 31: Instancia 30-100-3-50

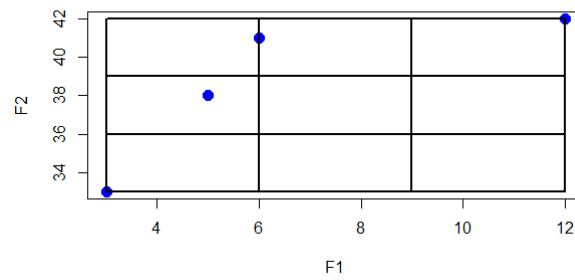


Figura 32: Instancia 50-100-6-100

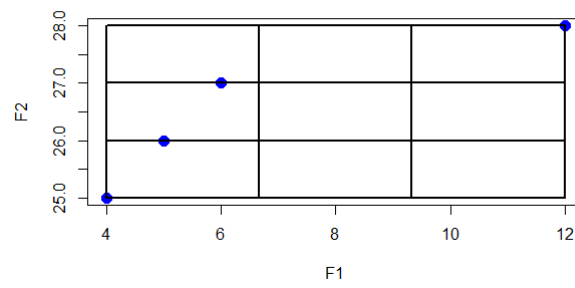


Figura 33: Instancia 30-100-3-100

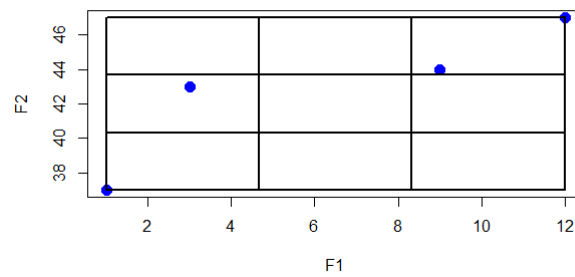


Figura 34: Instancia 50-100-3-50

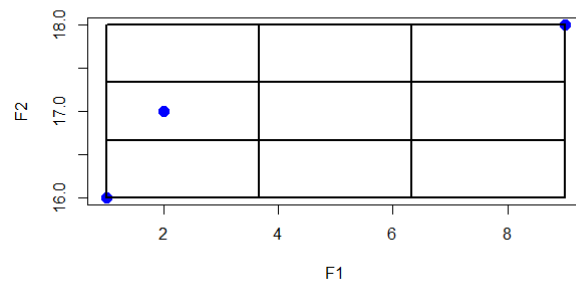


Figura 35: Instancia 20-20-3-10

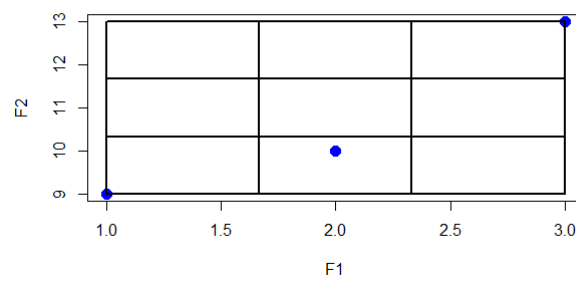


Figura 36: Instancia 20-20-6-1

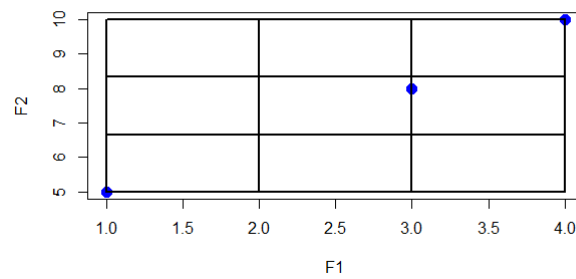


Figura 37: Instancia 20-20-6-50

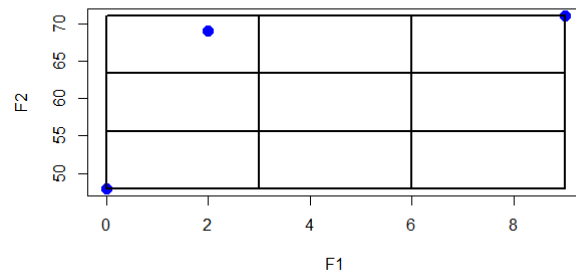


Figura 38: Instancia 30-30-3-1

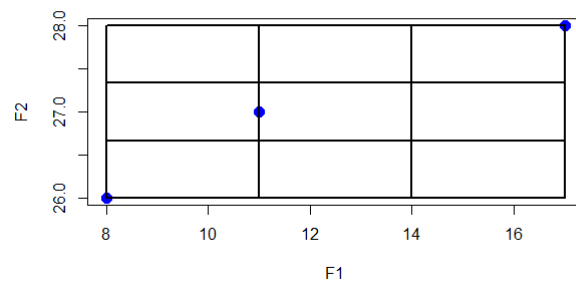


Figura 39: Instancia 30-30-3-10

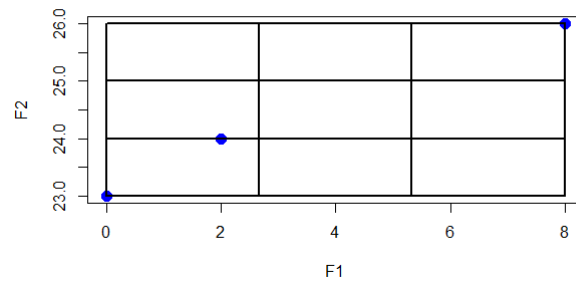


Figura 40: Instancia 30-30-3-100

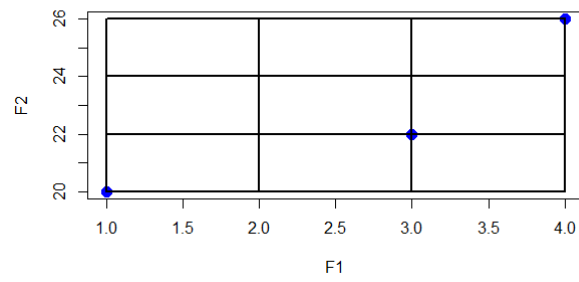


Figura 41: Instancia 30-30-4-100

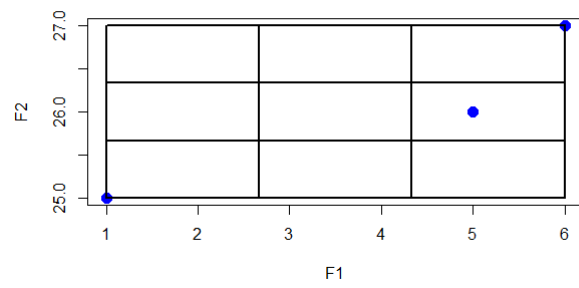


Figura 42: Instancia 30-50-3-50

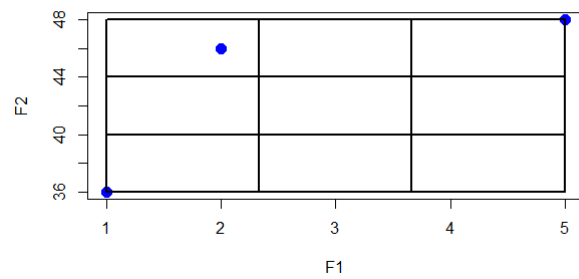


Figura 43: Instancia 50-50-3-50

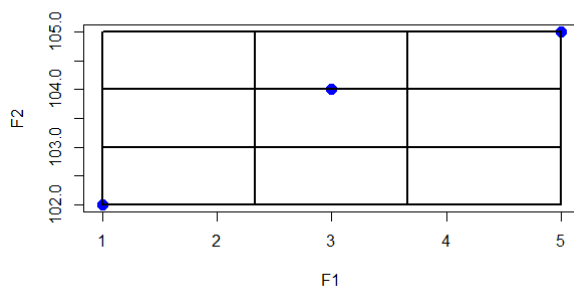


Figura 44: Instancia 100-100-3-1

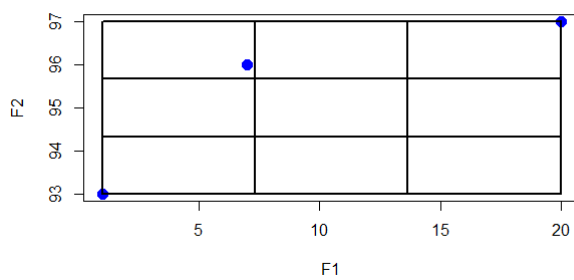


Figura 45: Instancia 100-100-3-50

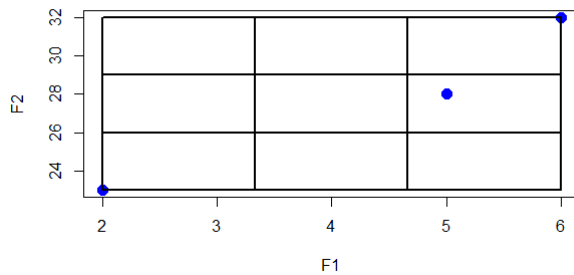


Figura 46: Instancia 50-50-10-100

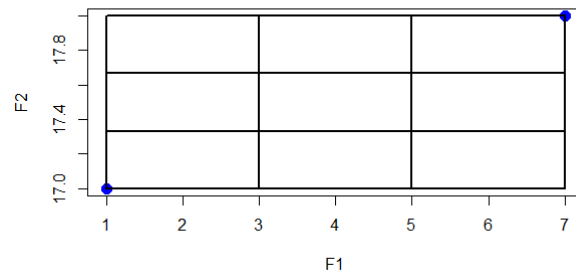


Figura 47: Instancia 20-20-3-50

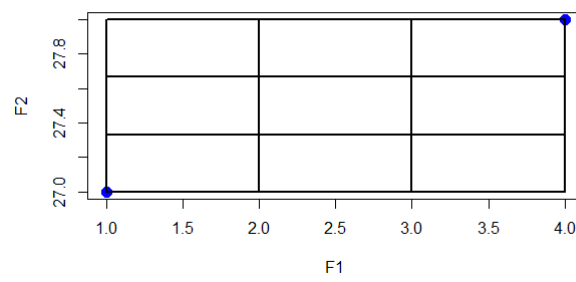


Figura 48: Instancia 30-50-3-10

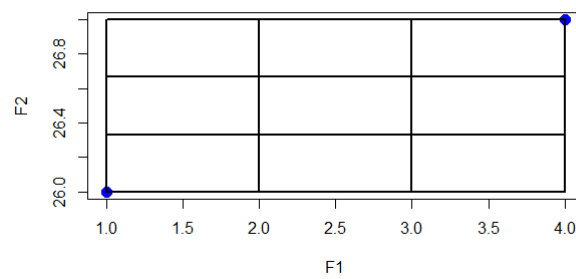


Figura 49: Instancia 30-50-3-100

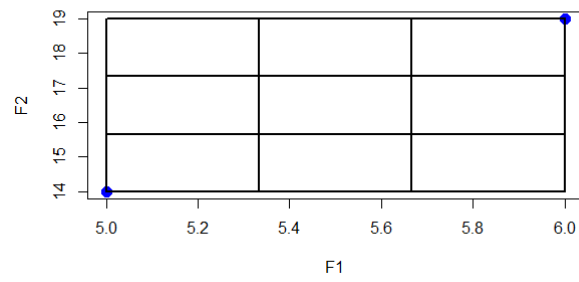


Figura 50: Instancia 30-50-10-10

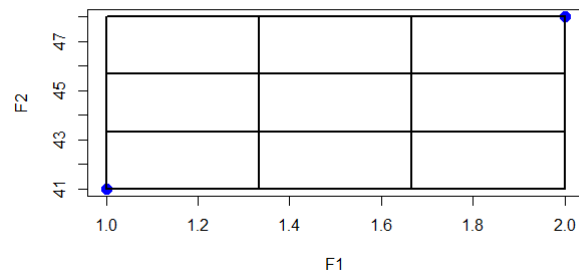


Figura 51: Instancia 50-50-3-10

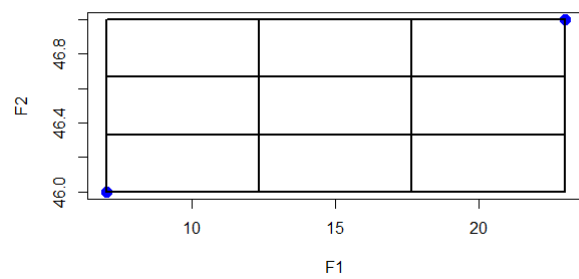


Figura 52: Instancia 50-100-3-100

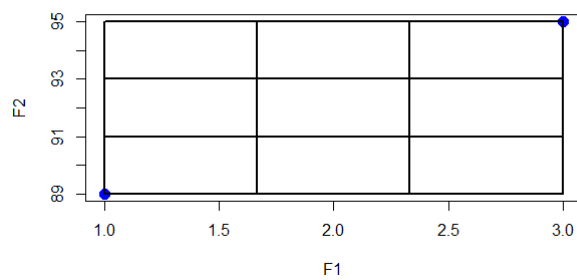


Figura 53: Instancia 100-100-3-10

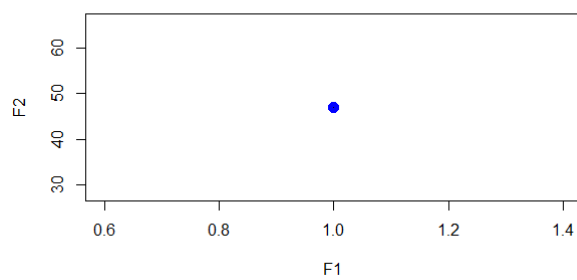


Figura 54: Instancia 50-50-3-100

Se puede ver que en las Figura 10-54 hay un número adecuado de soluciones graficadas. En los gráficos presentados, el número máximo de soluciones obtenidas en el frente de Pareto es 9 y el mínimo 1. Pero, como ya se mencionó anteriormente, es muy importante tener en cuenta que la primer función objetivo del líder F_1 busca minimizar el balance entre los habitantes que se atienden a un punto de vacunación abierto específico, que es un valor entero pequeño, por lo que no existen valores excesivos posibles para ese objetivo.

Para mejor visualización de las gráficas se acomodaron por la cantidad de soluciones encontradas, comenzando con la instancia que tiene mayor cantidad de soluciones en la Frontera Pareto y terminando con la instancia con la menor cantidad de soluciones.

En las Figuras 10 y 11 se muestran las dos instancias con la mayor cantidad de soluciones. Por ejemplo, la instancia 30-100-6-50 con 9 soluciones en la frontera Pareto, la instancia 30-50-6-100 con 7 soluciones. Las gráficas con 6 soluciones se muestran en las Figuras 12-17. En las Figuras 18-24 se presentan las instancias con cinco soluciones en la frontera Pareto. Las instancias con cuatro soluciones se muestran en las Figura 25-34. En las Figuras 35-46 las fronteras Pareto de las instancias con tres soluciones. Por último, en las Figuras 47- 53 se muestran las instancias con dos soluciones en la frontera Pareto, y el la Figuras 54 la instancia que solo obtuvo una única solución. Es importante destacar que la instancia 50-50-3-100 es la única instancia con una sola solución en la frontera Pareto, es decir esta solución domina a todas las demás soluciones encontradas

Además, en las gráficas muestran la cuadrícula de 3x3 que se realiza en cada iteración para considerar las mejores soluciones para la creación de las nueve subpoblaciones de soluciones. Dentro de la cuadrícula se observa que en la cuadrícula específica que corresponde a los mejores valores para F_1 y F_2 , el número de soluciones contenidas es de una a tres. Esto valida la idea de considerar la cuadrícula para encontrar siempre la mejor solución para cada función objetivo. En la gran mayoría de las 45 instancias consideradas en esta experimentación computacional, este comportamiento se mantiene. Así mismo, se puede observar que existe diversidad dentro de las soluciones, es decir, algunas son mejores para F_1 , otras mejores para F_2 , y algunas otras que están en el centro, según la suma ponderada considerada como función objetivo.

Para ilustrar el espacio limitado en el que se desenvuelve la función objetivo del líder (en relación con el equilibrio), se realiza una ejecución independiente del BONXEA y se establecen 100 ejecuciones como criterio de parada. Por lo tanto, se registró el mejor

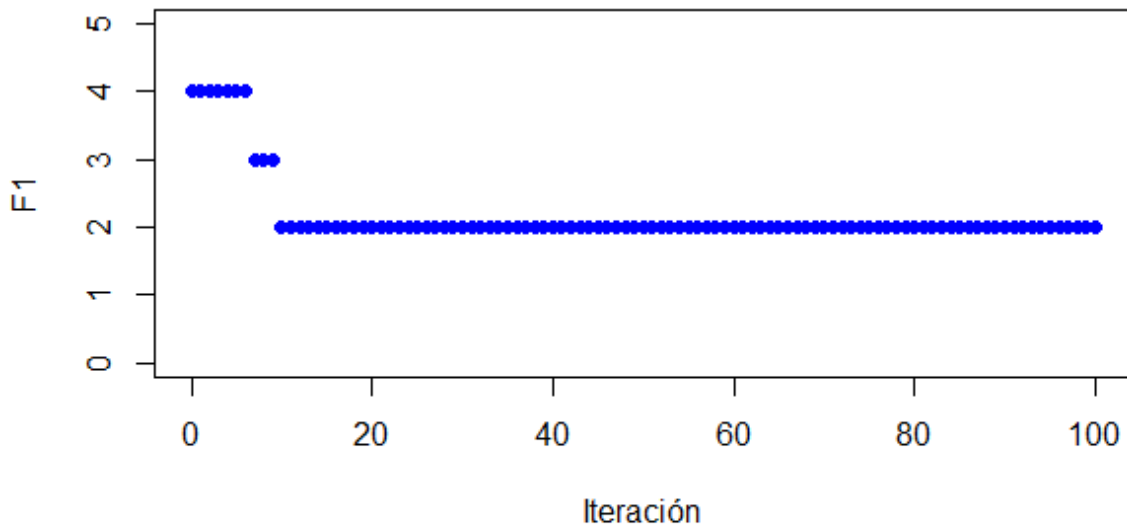


Figura 55: Convergencia de BONXEA con respecto a F_1 .

valor obtenido para F_1 en cada iteración. Esto puede darnos una idea de la capacidad de convergencia del BONXEA. Para llevar a cabo este análisis se utilizó la instancia que arrojó la mayor cantidad de soluciones dentro de la frontera Pareto, la instancia 30-100-6-50. Se puede ver en la Figura 55 que de un valor de balance de 4 mejora a 2, y no se reportan alguna otra mejora en las 90 iteraciones restantes. Además, en cada iteración de BONXEA pueden surgir otras soluciones no dominadas, pero este análisis particular se realiza para demostrar la convergencia de BONXEA con respecto a la función objetivo del líder en particular. En la Figura 56 se realiza el mismo análisis, pero con respecto a F_2 . También se puede notar que el BONXEA obtiene el mejor valor de F_2 en las primeras 40 iteraciones. Este es un buen indicador de que la BONXEA contiene un componente de intensificación bien diseñado. Esta misma situación se presenta para el resto de las instancias. Por lo que se puede concluir que la convergencia del algoritmo que proponemos, BONXEA, se llega en un número relativamente pequeño de iteraciones.

Podemos resumir que el algoritmo propuesto genera un conjunto de soluciones eficientes de tamaño razonable para nuestro problema. Por tanto, el decisor tendrá que elegir la solución que le resulte más conveniente en el momento en que se está resolviendo un problema real. Es decir, de todas las soluciones no dominadas obtenidas por el BONXEA, solo una puede implementarse en la práctica. Sin embargo, esto es común para los pro-

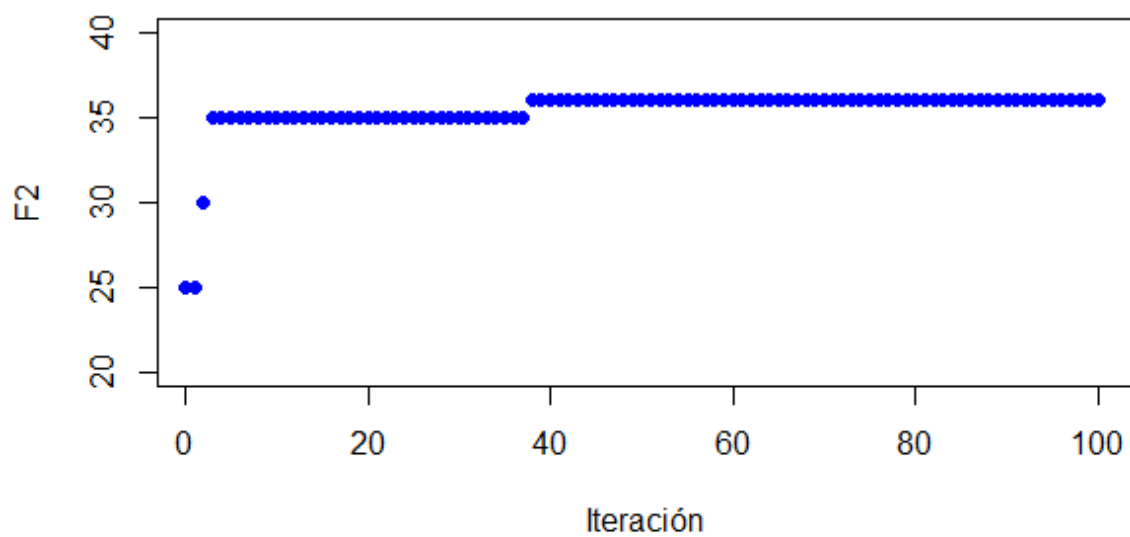


Figura 56: Convergencia de BONXEA con respecto a F_2 .

blemas prácticos de toma de decisiones bi-objetivo.

CAPÍTULO 6

CONCLUSIONES

En este último capítulo se comparten las conclusiones generales sobre el trabajo descrito en los capítulos anteriores.

Recapitulando un poco nuestro modelo planteado surge de la necesidad de contar con las herramientas necesarias para tomar decisiones rápidas y eficientes frente a una situación de logística humanitaria, en este caso nos enfocamos en la pandemia de COVID-19 que tomó por sorpresa a todo el mundo en diciembre del 2019. Frente a esta situación, que se puede decir que nadie estaba preparado para enfrentar, en este trabajo de investigación se propone un modelo que ayuda a ubicar diferentes puntos de vacunación para los habitantes de una determinada región, de tal manera que exista el acceso más equitativo a las vacunas, ya que la vacunación de los habitantes hoy en día es la forma más segura y eficaz de controlar la pandemia de COVID-19.

Para realizar esta investigación se propuso un modelo bi-objetivo bi-nivel. Donde el nivel superior está asociado con el Gobierno y el nivel inferior con los habitantes. El gobierno tiene como objetivo mantener la sana distancia y evitar aglomeraciones. Por lo tanto, las funciones objetivo son equilibrar el número de habitantes asignados a cada punto de vacunación, es decir, que la población se distribuya equitativamente en los puntos de vacunación abiertos. La segunda función objetivo apunta a que los puntos de vacunación estén lo más alejadas posible entre sí, es decir, maximizar la distancia mínima entre los puntos de vacunación abiertos. Por otro lado, los habitantes eligen a que punto de vacunación asistir, por lo tanto sus preferencias hacia los puntos de vacunación son consideradas, de tal manera que la función objetivo asociada con el nivel inferior es maximizar esas preferencias.

En el Capítulo 1 se planteó el objetivo general y los objetivos específicos de este trabajo de investigación. Dichos objetivos se han trabajado a lo largo de la investigación

aquí presentada. Por lo tanto, se puede concluir de forma satisfactoria que se cumplió con cada uno de ellos. En seguida se describe cada uno de ellos.

El objetivo general que se planteó al inicio de esta investigación fue proponer una formulación de programación matemática bi-objetivo bi-nivel para modelar un sistema de vacunación que a mantuviera la *sana distancia* entre los habitantes sin imponer restricciones en cuanto al punto de vacunación que debe asistir. Como se mencionó anteriormente este modelo surgió de la motivación de la presente situación de la pandemia de COVID-19 y tener una herramienta que nos ayude a tomar decisiones de una forma rápida y eficiente cuando se presente una situación como esta o cualquier otro tipo de situación de desastre natural en la cual interviene la logística humanitaria. Dicho modelo se presenta de forma más detallada en el Capítulo 3.

Bajo este enfoque se puede concluir que el modelo propuesto se puede aplicar a cualquier situación en la cual se deba realizar asignaciones de personas, materiales, alimentos, objetos, ayuda en especie u otras cosas a determinado conjunto de centros de vacunación, centros de acopio u otro. Simplemente considerando los parámetros necesarios para el problema de interés. De tal manera que ya se cuenta con una herramienta para tomar decisiones en una situación de este tipo y poder implementarla en futuras situaciones de logística humanitaria. Además se puede concluir que los resultados obtenidos son confiables y de buena calidad.

Dentro de los objetivos específicos que se plantearon y el cumplimiento de ellos; destaca lo siguiente:

- Se propuso una metaheurística de entropía cruzada que considera simultáneamente dos objetivos y maneja la estructura bi-nivel del problema planteado. Es decir, que resuelve el problema del nivel inferior mientras considera las decisiones de nivel superior.

Es importante destacar que esta es la primera vez que se aplicó la entropía cruzada para resolver un problema bi-objetivo bi-nivel. Al igual que el modelo propuesto, la metaheurística que se propone se puede aplicar a cualquier otra situación de desastre natural que se pueda asemejar a la nuestra considerando los parámetros correspondientes. De tal forma que si nos encontramos en una situación como la descrita aquí o cualquier otra parecida se tiene el modelo y la metaheurística para encontrar una solución rápida y eficiente, que en la mayoría de las ocasiones es de vital importancia para la seguridad y bienestar de las personas afectadas.

El algoritmo propuesto que se describe de forma detallada en el Capítulo 4 arrojó buenos resultados. En todas las instancias de prueba se obtuvo una buena solución en un tiempo razonable.

- Se llevó a cabo una extensa experimentación computacional y se validó la metaheurística propuesta y se obtuvo información gerencial importante con respecto a la ubicación de los puntos de vacunación y el distanciamiento social. Nuestros experimentos computacionales son alentadores teniendo en cuenta el potencial de expandir el algoritmo BONXEA. La experimentación computacional muestra que el algoritmo propuesto es capaz de obtener soluciones de alta calidad de la frontera de Pareto en un tiempo razonable.
- De los resultados obtenidos sobre el modelo propuesto y de la metaheurística de entropía cruzada utilizada para resolver el problema bajo estudio siempre se obtuvieron soluciones no dominadas en cada una de las instancias que se trabajaron, que van desde tener una única solución no dominada hasta tener 9 soluciones no dominadas.

También dentro de los resultados obtenidos esta presente el equilibrio perfecto entre los habitantes y los puntos de vacunación. Es decir cuando la función objetivo F_1 toma el valor de 0. Sin embargo el otro lado extremo también está presente, cuando existe disparidad entre la aglomeración en los puntos de vacunación que están abiertos. Esto se da por que los puntos de vacunación estan muy distantes y provoca que los habitantes se concentren en el punto de vacunación central. En conclusión esto ocurre solo en casos particulares debido a la distribución espacial de los habitantes y los puntos de vacunación potenciales.

A partir de los resultados analizados se puede observar que existe diversidad dentro de las soluciones.

Se puede decir que para instancias pequeñas el tiempo que le tomaba a nuestro algoritmo encontrar la frontera de Pareto es muy rápido. Para instancias más grandes el algoritmo aumentaba un poco el tiempo para llegar a la frontera de Pareto. Sobre la convergencia del algoritmo que proponemos, se puede analizar que se llega en un número relativamente pequeño de iteraciones.

En resumen podemos concluir que el algoritmo propuesto genera un conjunto de soluciones eficientes de tamaño razonable para nuestro problema.

Para terminar y como conclusión final; una de las principales aportaciones de este trabajo es el hecho de considerar un problema bi-objetivo bi-nivel bajo un enfoque de logística humanitaria y poder resolverlo por medio de nuestro algoritmo propuesto.

Teniendo en mente que la estructura de nuestro modelo es difícil de trabajar se pudo diseñar un algoritmo que implementa la Entropía Cruzada para resolverlo y obtener buenos resultados.

6.1 LÍNEAS DE INVESTIGACIÓN FUTURAS

Después de haber realizado la investigación para esta tesis, nos dimos cuenta de varias posibles extensiones o algunas variantes que pudieran abordarse para contribuir al conocimiento. Dentro de las líneas de investigación futuras se destaca lo siguiente:

Debido a que uno de los objetivos aquí planteados era el de generar una herramienta para resolver eficientemente este tipo de modelos bi-nivel bi-objetivo de localización de instalaciones donde los usuarios toman sus propias decisiones hacia cual instalación acudir. La primer idea es buscar más aplicaciones en diferentes contextos en donde se pueda aplicar (con pequeñas adaptaciones) el modelo aquí propuesto. También sería de interés encontrar otros contextos en los cuales sea de utilidad contar con una herramienta de soporte de decisiones.

Otro punto importante que nos gustaría trabajar sería el de implementar nuestro algoritmo propuesto con datos reales sobre ubicaciones de puntos de vacunación, número de habitantes y ubicación de ellos para la asignación a cada punto de vacunación. Es decir, poder construir una instancia real, pero con la información que se obtiene de páginas gubernamentales hoy en día, no es posible. Sin embargo, creemos que sería de gran ayuda poder utilizar datos reales y no solo sintéticos para ver el impacto real en la sociedad.

También como línea de investigación futura se puede plantear la idea de agregar restricciones para que nuestro algoritmo sea un poco más cercano a la realidad por ejemplo: considerar capacidad en cada uno de los puntos de vacunación, horarios específicos de atención, etc. Esto se puede observar en situaciones reales, en donde en algunos puntos de vacunación no se tenía la cantidad de vacunas necesaria para todos los habitantes que asistieron a determinado punto de vacunación. Las ventanas de tiempo serían para simular los diferentes horarios de atención para la vacunación.

Otra posible modificación que se podría implementar en nuestro algoritmo para ofrecer la ayuda necesaria para los habitantes es no restringir el número de puntos de va-

cunación que se deben ubicar, de tal manera que se busca ubicar los puntos de vacunación necesarios para que todos los habitantes tengan la oportunidad del acceso a la vacuna. Obteniendo resultados se podrían comparar con el número de puntos de vacunación que propone el algoritmo y la cantidad de centros que el gobierno propone, y tomar la mejor decisión a esta situación.

Para terminar, otra idea sería considerar dos objetivos en el problema del seguidor. Actualmente el líder tiene dos objetivos y el seguidor solo uno. En el otro caso, no queda claro como resolver el problema del nivel inferior, ya que se obtendría un frente de Pareto como solución de ese problema. Entonces, no queda claro cual de todas las soluciones no-dominadas sería dada como respuesta óptima del seguidor. En ese tipo de problemas hay que asumir enfoques y definir algunos aspectos muy finos (dependiendo el problema en cuestión) para poder obtener una solución factible binivel en el sentido estricto.

BIBLIOGRAFÍA

- Abbasi, B., Fadaki, M., Kokshagina, O., Saeed, N., and Chhetri, P. (2020). Modeling vaccine allocations in the covid-19 pandemic: A case study in australia. *Available at SSRN 3744520*.
- Abdin, A. F., Fang, Y.-P., Caunhye, A., Alem, D., Barros, A., and Zio, E. (2023). An optimization model for planning testing and control strategies to limit the spread of a pandemic—the case of covid-19. *European journal of operational research*, 304(1):308–324.
- Akinc, U. and Khumawala, B. M. (1977). An efficient branch and bound algorithm for the capacitated warehouse location problem. *Management Science*, 23(6):585–594.
- Bard, J. F. (2013). *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media.
- Beasley, J. E. (1988). An algorithm for solving large capacitated warehouse location problems. *European Journal of Operational Research*, 33(3):314–325.
- Bertsimas, D., Digalakis Jr, V., Jacquillat, A., Li, M. L., and Previero, A. (2022). Where to locate covid-19 mass vaccination facilities? *Naval Research Logistics (NRL)*, 69(2):179–200.
- Buhat, C. A. H., Duero, J. C. C., Felix, E. F. O., Rabajante, J. F., and Mamplata, J. B. (2021). Optimal allocation of covid-19 test kits among accredited testing centers in the philippines. *Journal of healthcare informatics research*, 5(1):54–69.
- Burgos, D. and Ivanov, D. (2021). Food retail supply chain resilience and the covid-19 pandemic: A digital twin-based impact analysis and improvement directions. *Transportation Research Part E: Logistics and Transportation Review*, 152:102412.
- Caballero, R., Hernández-Díaz, A. G., Laguna, M., and Molina, J. (2015). Cross entropy for multiobjective combinatorial optimization problems with linear relaxations. *European Journal of Operational Research*, 243(2):362–368.

- Calvete, H. I., Galé, C., Iranzo, J. A., Camacho-Vallejo, J.-F., and Casas-Ramírez, M.-S. (2020). A matheuristic for solving the bilevel approach of the facility location problem with cardinality constraints and preferences. *Computers & Operations Research*, 124:105066.
- Camacho-Vallejo, J.-F., Corpus, C., and Villegas, J. G. (2023). Metaheuristics for bilevel optimization: A comprehensive review. *Computers & Operations Research*, page 106410.
- Camacho-Vallejo, J.-F., García-Vélez, J.-C., and Corpus, C. (2022). A 5g hubs location hierarchized problem that balances the connection of the users. *Mobile Networks and Applications*, pages 1–11.
- Cánovas, L., García, S., Labbé, M., and Marín, A. (2007). A strengthened formulation for the simple plant location problem with order. *Operations Research Letters*, 35(2):141–150.
- Casas-Ramírez, M.-S. and Camacho-Vallejo, J.-F. (2017). Solving the p-median bilevel problem with order through a hybrid heuristic. *Applied Soft Computing*, 60:73–86.
- Casas-Ramírez, M.-S., Camacho-Vallejo, J.-F., Díaz, J. A., and Luna, D. E. (2020). A bi-level maximal covering location problem. *Operational Research*, 20:827–855.
- Casas-Ramírez, M.-S., Camacho-Vallejo, J.-F., and Martínez-Salazar, I.-A. (2018). Approximating solutions to a bilevel capacitated facility location problem with customer’s patronization toward a list of preferences. *Applied Mathematics and Computation*, 319:369–386.
- Caserta, M. and Rico, E. Q. (2009). A cross entropy-lagrangian hybrid algorithm for the multi-item capacitated lot-sizing problem with setup times. *Computers & Operations Research*, 36(2):530–548.
- Chowdhury, P., Paul, S. K., Kaisar, S., and Moktadir, M. A. (2021). Covid-19 pandemic related supply chain studies: A systematic review. *Transportation Research Part E: Logistics and Transportation Review*, 148:102271.
- Cornuéjols, G., Nemhauser, G., and Wolsey, L. (1983). The uncapacitated facility location problem. Technical report, Cornell University Operations Research and Industrial Engineering.
- Dasarathy, B. and White, L. J. (1980). A maxmin location problem. *Operations Research*, 28(6):1385–1401.
- Daskin, M. (1997). Network and discrete location: models, algorithms and applications. *Journal of the Operational Research Society*, 48(7):763–764.

- De Boer, P.-T., Kroese, D. P., Mannor, S., and Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67.
- Dempe, S. et al. (2009). Bilevel programming: Implicit function approach. *Encyclopedia of optimization*, 2:260–266.
- Duijzer, L. E., Van Jaarsveld, W., and Dekker, R. (2018a). Literature review: The vaccine supply chain. *European Journal of Operational Research*, 268(1):174–192.
- Duijzer, L. E., van Jaarsveld, W. L., Wallinga, J., and Dekker, R. (2018b). Dose-optimal vaccine allocation over multiple populations. *Production and Operations Management*, 27(1):143–159.
- Ehrgott, M., Wiecek, M. M., et al. (2005). Multiobjective programming. *Multiple criteria decision analysis: State of the art surveys*, 78:667–708.
- Eiselt, H. A. and Marianov, V. (2011). Foundations of location analysis.
- Enayati, S. and Özaltın, O. Y. (2020). Optimal influenza vaccine distribution with equity. *European Journal of Operational Research*, 283(2):714–725.
- Erkut, E. (1990). The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48–60.
- Erlenkotter, D. (1978). A dual-based procedure for uncapacitated facility location. *Operations Research*, 26(6):992–1009.
- Farahani, R. Z., Asgari, N., Heidari, N., Hosseini, M., and Goh, M. (2012). Covering problems in facility location: A review. *Computers & Industrial Engineering*, 62(1):368–407.
- Farahani, R. Z. and Hekmatfar, M. (2009). *Facility location: concepts, models, algorithms and case studies*. Springer Science & Business Media.
- Fearon, D. (2006). Alfred weber: theory of the location of industries, 1909. *ICIS Center for Spatially Integrated Social Science*.
- Filippi, G. and Vasile, M. (2020). Inflationary differential evolution for constrained multi-objective optimisation problems. In *Bioinspired Optimization Methods and Their Applications: 9th International Conference, BIOMA 2020, Brussels, Belgium, November 19–20, 2020, Proceedings 9*, pages 29–42. Springer.
- Galvão, R. D. and Raggi, L. A. (1989). A method for solving to optimality uncapacitated location problems. *Annals of Operations Research*, 18.

- Hakimi, S. L. (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations research*, 12(3):450–459.
- Hakimi, S. L. (1965). Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations research*, 13(3):462–475.
- Hanjoul, P. and Peeters, D. (1987). A facility location problem with clients' preference orderings. *Regional Science and Urban Economics*, 17(3):451–473.
- Hansen, P., Kochetov, Y., and Mladenovi, N. (2004). *Lower bounds for the uncapacitated facility location problem with user preferences*. Groupe d'études et de recherche en analyse des décisions, HEC Montréal.
- Hosseininezhad, S. J., Salhi, S., and Jabalameli, M. S. (2015). A cross entropy-based heuristic for the capacitated multi-source weber problem with facility fixed cost. *Computers & Industrial Engineering*, 83:151–158.
- Iwama, K. and Miyazaki, S. (2008). A survey of the stable marriage problem and its variants. In *International conference on informatics education and research for knowledge-circulating society (ICKS 2008)*, pages 131–136. IEEE.
- Kalantari, F. and Hosseininezhad, S. J. (2022). A multi-objective cross entropy-based algorithm for sustainable global food supply chain with risk considerations: A case study. *Computers & Industrial Engineering*, 164:107766.
- Kroese, D. P., Porotsky, S., and Rubinstein, R. Y. (2006). The cross-entropy method for continuous multi-extremal optimization. *Methodology and Computing in Applied Probability*, 8(3):383–407.
- Laporte, G., Nickel, S., and Saldanha-da Gama, F. (2019). *Introduction to location science*. Springer.
- Li, X., Pan, Y., Jiang, S., Huang, Q., Chen, Z., Zhang, M., and Zhang, Z. (2021). Locate vaccination stations considering travel distance, operational cost, and work schedule. *Omega*, 101:102236.
- Luo, L., Wan, X., and Wang, Q. (2022). A multi-period location–allocation model for integrated management of emergency medical supplies and infected patients during epidemics. *Computers & Industrial Engineering*, 173:108640.
- Maldonado-Pinto, S., Casas-Ramírez, M.-S., and Camacho-Vallejo, J.-F. (2016). Analyzing the performance of a hybrid heuristic for solving a bilevel location problem under

- different approaches to tackle the lower level. *Mathematical Problems in Engineering*, 2016.
- Manupati, V. K., Schoenherr, T., Wagner, S. M., Soni, B., Panigrahi, S., and Ramkumar, M. (2021). Convalescent plasma bank facility location-allocation problem for covid-19. *Transportation Research Part E: Logistics and Transportation Review*, 156:102517.
- Marić, M., Stanimirović, Z., and Milenković, N. (2012). Metaheuristic methods for solving the bilevel uncapacitated facility location problem with clients' preferences. *Electronic Notes in Discrete Mathematics*, 39:43–50.
- Marín, A. (2011). The discrete facility location problem with balanced allocation of customers. *European Journal of Operational Research*, 210(1):27–38.
- McVitie, D. G. and Wilson, L. B. (1971). The stable marriage problem. *Communications of the ACM*, 14(7):486–490.
- Medlock, J. and Galvani, A. P. (2009). Optimizing influenza vaccine distribution. *Science*, 325(5948):1705–1708.
- Méndez, M., Frutos, M., Pulido, A., Fabio, M., Rossit, D. A., and Alamo, A. L. (2018). Manejo de conocimiento en algoritmos evolutivos multiobjetivo para el diseño óptimo de un sistema de engranajes.
- Mitropoulos, P., Mitropoulos, I., Giannikos, I., and Sissouras, A. (2006). A biobjective model for the locational planning of hospitals and health centers. *Health Care Management Science*, 9(2):171–179.
- Mohammadi, M., Dehghan, M., Pirayesh, A., and Dolgui, A. (2022). Bi-objective optimization of a stochastic resilient vaccine distribution network in the context of the covid-19 pandemic. *Omega*, 113:102725.
- Muñoz, A. D. (2007). *Metaheurísticas*, volume 22. Librería-Editorial Dykinson.
- Ning, X. and Li, P. (2018). A cross-entropy approach to the single row facility layout problem. *International Journal of Production Research*, 56(11):3781–3794.
- Ortega Rosales, A. (2015). *Sobre los problemas binivel lineales con coeficientes intervalos en los lados derechos de las restricciones del nivel inferior*. PhD thesis, Universidad Autónoma de Nuevo León.
- Ozdemir, I., Dursunoglu, C. F., Kara, B. Y., and Dora, M. (2022). Logistics of temporary testing centers for coronavirus disease. *Transportation Research Part C: Emerging Technologies*, page 103954.

- Parvin, H., Beygi, S., Helm, J. E., Larson, P. S., and Van Oyen, M. P. (2018). Distribution of medication considering information, transshipment, and clustering: Malaria in malawi. *Production and Operations Management*, 27(4):774–797.
- Perelman, L. and Ostfeld, A. (2007). An adaptive heuristic cross-entropy algorithm for optimal design of water distribution systems. *Engineering Optimization*, 39(4):413–428.
- Pérez, J. A. M. (2004). Metaheurísticas: Concepto y propiedades. *Departamento de estadística, IO y Computación. Universidad de La Laguna. Tenerife, España.*
- Pinilla, J. A. A. and Castro, J. A. O. (2015). Optimización multiobjetivo en la gestión de cadenas de suministro de biocombustibles. una revisión de la literatura. *Ingeniería*, 20(1):37–63.
- Rahman, T., Taghikhah, F., Paul, S. K., Shukla, N., and Agarwal, R. (2021). An agent-based model for supply chain recovery in the wake of the covid-19 pandemic. *Computers & Industrial Engineering*, 158:107401.
- Rastegar, M., Tavana, M., Meraj, A., and Mina, H. (2021). An inventory-location optimization model for equitable influenza vaccine distribution in developing countries during the covid-19 pandemic. *Vaccine*, 39(3):495–504.
- Rubinstein, R. Y. (1997). Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112.
- Santosa, B., Budiman, M. A., Wiratno, S. E., et al. (2011). A cross entropy-genetic algorithm for m-machines no-wait job-shopscheduling problem. *Journal of Intelligent Learning Systems and Applications*, 3(03):171.
- Sarkar, S., Pramanik, A., Maiti, J., and Reniers, G. (2021). Covid-19 outbreak: A data-driven optimization model for allocation of patients. *Computers & Industrial Engineering*, 161:107675.
- Sun, L., DePuy, G. W., and Evans, G. W. (2014). Multi-objective optimization models for patient allocation during a pandemic influenza outbreak. *Computers & Operations Research*, 51:350–359.
- Tang, L., Li, Y., Bai, D., Liu, T., and Coelho, L. C. (2022). Bi-objective optimization for a multi-period covid-19 vaccination planning problem. *Omega*, 110:102617.
- Thul, L. and Powell, W. (2023). Stochastic optimization for vaccine and testing kit allocation for the covid-19 pandemic. *European journal of operational research*, 304(1):325–338.

- Toaza, B. and Esztergár-Kiss, D. (2023). A review of metaheuristic algorithms for solving tsp-based scheduling optimization problems. *Applied Soft Computing*, page 110908.
- Unveren, A. and Acan, A. (2007). Multi-objective optimization with cross entropy method: Stochastic learning with clustered pareto fronts. In *2007 IEEE Congress on Evolutionary Computation*, pages 3065–3071. IEEE.
- Von Stackelberg, H. (1934). *Marktform und gleichgewicht*. J. springer.
- Wang, S., Wu, Y. J., and Li, R. (2022). An improved genetic algorithm for location allocation problem with grey theory in public health emergencies. *International Journal of Environmental Research and Public Health*, 19(15):9752.
- Wang, Z., Soleimani, H., Kannan, D., and Xu, L. (2016). Advanced cross-entropy in closed-loop supply chain planning. *Journal of Cleaner Production*, 135:201–213.
- Weber, A. (1929). *Alfred Weber's theory of the location of industries*. University of Chicago Press.
- Yarmand, H., Ivy, J. S., Denton, B., and Lloyd, A. L. (2014). Optimal two-phase vaccine allocation to geographically different regions under uncertainty. *European Journal of Operational Research*, 233(1):208–219.