Universidad de La Habana
Facultad de Matemática y Computación

Localización de instalaciones

Autor:
Rainel Fernández Abreu.

Tutor:
Dr.Gemayqzel Bouza Allende

Trabajo de Diploma
presentado en opción al título de
Licenciado en Ciencia de la Computación

Enero de 2024

https://github.com/rainel99/Thesis

Agradecimientos

Agradecer a mis padres Ariagna Abreu y Nelson Fernandez por el apoyo constante para lograr este sueño. A mi hermano Robin por su confianza y paciencia. A mi abuela por su cariño infinito. A mi tutora Gema, por su atención, dedicación y profesionalismo en este período de guía, del cual me llevo lindas anécdotas y mucha experiencia académica. A todos mis profesores por sus incansables ganas de enseñar. A esta facultad que cada año es capaz de formar grandes profesionales, y más importantes, excelentes seres humanos humanos.

Por supuesto gracias a mis amigos, a Joaquín, Rodrigo, Jairo, Miguel, Lázaro, Rosen, Claudia a mis amigos de la academia y para siempre amigos de la vida, Karlos, Karel, Elena, Hansel y Sheyla. Gracias a Hiram, por ser amigo y maestro.

A todos, muchas gracias...

Opinión del tutor

Hay temas que son de interés matemático por su complejidad y su importancia práctica. Este estudio teórico va de la mano con la existencia de algoritmos para su solución. El modelo de localización con obstáculos que se trata acá es complejo pues es un modelo de optimización multiobjetivo no suave no convexo. De ahí que para poder abordar el problema sea de interés contar con una descripción inicial del conjunto solución. Se propuso el tema y Rainel aceptó. Realmente el estudiante se entusiasmó con el tema, asistió y evaluó satisfactoriamente cursos de optimización avanzados, estudió la bibliografía en temas nuevos como la optimización multiobjetivo y los problemas de localización y comenzó a trabajar en el tema.

Rainel fue siempre muy proactivo, siempre llegaba a las citas con las tareas resueltas y nuevas ideas. Propuso una heurística para resolver el problema y ejemplos propios para la experimentación.

Fue un gusto trabajar con él, Aplicado, amable, creativo, comprensivo, independiente y organizado, serían los adjetivos con que lo calificaría. Auguro que logrará muchos éxitos en su profesión por sus habilidades, capacidad de trabajo individual y en equipo, amabilidad y su sensibilidad ante los problemas de los demás.

Resumen

El problema de localización con zonas prohibidas es un desafío en la planificación espacial que requiere encontrar la posición ideal para instalaciones, teniendo en cuenta las áreas restringidas donde no se permite la instalación. La planificación espacial es un imperativo en el mundo moderno que exige hallar la ubicación óptima de instalaciones para un uso eficiente de recursos y maximización de ganancias. Esta tesis se enfoca en una instancia del problema, en el cual se quiere encontrar la mejor ubicación a una nueva instalación que esté cerca de algunos lugares mientras está alejada de otros, con la condición de que existe un río con dos puentes que divide la región donde se trabaja, que permiten el acceso de un lado a otro, siendo este río la zona en la cual no es posible colocar instalaciones. Se proponen dos modelos matemáticos para el problema de ubicación utilizando un enfoque multiobjetivo. Se utiliza un algoritmo genético para resolver la problemática planteada y se muestran los resultados del mismo en la experimenación.

Abstract

The problem of location with prohibited zones is a challenge in spatial planning that requires finding the ideal position for facilities, taking into account restricted areas where installation is not allowed. Spatial planning is an imperative in the modern world that demands finding the optimal location of facilities for efficient use of resources and maximization of profits. This thesis focuses on an instance of the problem, in which we want to find the best location for a new facility that is close to some places while it is far from others, with the condition that there is a river with two bridges that divides the region where we work, which allow access from one side to the other, this river being the area in which it is not possible to place facilities. Two mathematical models are proposed for the location problem using a multi-objective approach. A genetic algorithm is used to solve the problem at hand and the results of it are shown in the experimentation.

Índice general

Índice de figuras

Capítulo 1

Introducción

La optimización es un campo fundamental en la ciencia y la ingeniería que se centra en encontrar la mejor solución posible para un problema específico. De forma general, implica maximizar o minimizar una función objetivo sujeta a un conjunto de restricciones. El objetivo es encontrar los valores de las variables de decisión que optimicen el rendimiento o la eficiencia del sistema en cuestión.

Esta esfera de las matemáticas se utiliza en una amplia gama de aplicaciones, desde la planificación de rutas y la programación de tareas hasta el diseño de productos y la toma de decisiones empresariales, ver Drezner y Horst W. Hamacher 2002. A través de técnicas y algoritmos sofisticados, la optimización permite explorar el espacio de soluciones, evaluar diferentes opciones y encontrar la mejor combinación de variables para lograr los objetivos deseados. Esta juega un papel crucial en la mejora de procesos, la reducción de costos, el aumento de la productividad y la toma de decisiones informadas en diversos campos, lo que la convierte en una herramienta fundamental para resolver problemas complejos y mejorar el rendimiento de sistemas en la vida real, referirse a Bertsekas 1999.

El problema de localización se estudia en diversas áreas, algunas de estas son planificación territorial y urbana, ver Garrido 2015, ingeniería y arquitectura, ver Garrido 2019, logística y cadena de suministros,ver Cruz et al. 2021 servicios de emergencia, telecomunicaciones, ver Bonilla 2019, para otros ejemplos de aplicaciones referirise a Daskin 2010. El problema de localización consiste en encontrar la mejor ubicación para un lugar. Un ejemplo práctico aparece cuando se desea encontrar un lugar que esté cerca de los clientes potenciales, lejos de negocios similares y que haya suficiente espacio. Estos criterios ayudan a reducir el espacio de búsqueda y encontrar lo que el decisor define como ¨mejor¨ ubicación. Existen diferentes tipos de problema de localización, para más detalles ver, Cantlebary y L. Li 2020. En varios trabajos se ha considerado el caso de colocar una entidad que se encuentre cerca de un conjunto de puntos dados y alejado de otros en una región. Se busca que el punto a colocar

minimice la distancia a cada punto cercano y maximice la distancia a los alejados. De ahi que se tenga un problema con tantos objetivos a minimizar como puntos cercanos haya y se maximicen tantas funciones como puntos alejados exista. En esta tesis se considera este problema con la novedad de que la región está dividida por un río. La

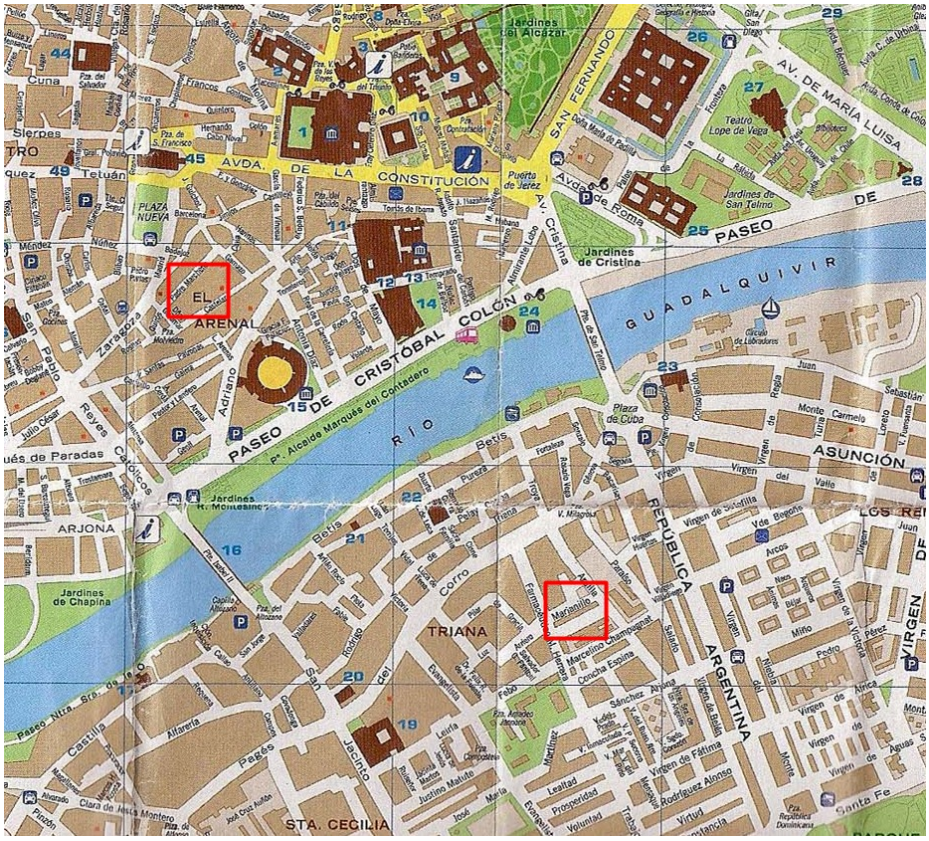

Figura 1.1: Distancia entre dos puntos situados en márgenes diferentes del Guadalquivir

presencia de este accidente natural impone obstáculos físicos que afectan el cálculo de la distancia entre las instalaciones que intervienen en el problema. Si se coloca la entidad en un margen del río, para hallar la distancia a los puntos que están en la otra rivera ese debe decidir que puente utilizar. Por simplicidad se supone que el río tiene forma rectangular, que hay solo dos puentes para cruzarlo, y que son la única forma de pasar de una rivera a la otra. Asumiendo racionalidad, de las dos posibles rutas existentes se toma la de longitud minima. De ahí que la distancia a minimizar si se debe estar cerca de este punto o a maximizar si este debe estar alejado está dada por una función no diferenciable dada por el mínimo entre dos fucniones.

Motivación

La eficiencia en la ubicación es un factor crítico que puede conducir a una optimización de los recursos. Sin embargo, la resolución de problemas de ubicación presenta desafíos computacionales debido a su naturaleza no suave. Esta investigación se centra en abordar estos desafíos computacionales y en desarrollar una solución eficiente para los problemas matemáticos.

Es importante destacar que, aunque los problemas de ubicación pueden ser difíciles de resolver, su importancia en la vida cotidiana y en numerosas aplicaciones prácticas hace que este trabajo sea necesario. Desde la planificación de la ubicación de las instalaciones hasta la optimización de las rutas de entrega, los problemas de ubicación son una parte integral de muchas operaciones y sistemas. Por lo tanto, cualquier mejora en la eficiencia de la resolución de estos problemas puede tener un impacto significativo.

En esta investigación se estará proponiendo una forma de resolver el problema de ubicación con puntos cercanos y alejados en una región donde existe un río con dos puentes para cruzarlo.

Hipótesis

Un algoritmo evolutivo multiobjetivo permitirá encontrar aproximaciones buenas a la solución del problema d ubicar una nueva entidad cercana a ciertos puntos y alejada de otros, ya prefijados, en una región con un río de forma rectangular y dos puentes para atravesarlo.

Objetivos

El objetivo principal de la investigación es desarrollar un algoritmo genético para resolver el problema de optimización multiobjetivo que resulta de un problema de ubicación con puntos cercanos y alejados en una región donde existe un río con dos puentes para cruzarlo. Para ello se cumplirán los siguientes objetivos específicos.

- Investigar el estado del arte de los algoritmos evolutivos, la solución de problemas de localización, los algoritmos para solución de problemas multiobjetivos con énfasis en los algoritmos evolutivos.
- Explorar el impacto de la presencia de un río en la región, como afecta este en el cálculo de distancias entre puntos.
- Implementar un algoritmo evolutivo para resolver el problema de optimización de varios objetivos.

- Evaluar el desempeño de la solución propuesta a partir de experimentos numéricos.

Este resto del documento se estructura de la siguiente manera: El segundo capítulo proporciona una revisión de la literatura, introduciendo definiciones pertinentes y discutiendo trabajos previos en este campo de estudio. El tercer capítulo se dedica a la exposición de los modelos matemáticos propuestos, sus propiedades inherentes y una propuesta algorítmica. El cuarto capítulo profundiza en la implementación de la propuesta algorítmica, describiendo en detalle los experimentos realizados y los resultados obtenidos. Finalmente las conclusiones de la investigación y sugerencias para futuras investigaciones en este ámbito.

Capítulo 2

Preliminares

Este capítulo está dedicado a sentar las bases para el análisis y propuesta de solución computacional del problema que se considera en esta tesis. Primeramente se abordan las definiciones y resultados elementales relacionados con la programación multiobjetivo. Luego de mostrar algunas propuestas algorítmicas para estos modelos, se explica a grandes rasgos en que consiste la metaheurística evolutiva que se adaptará para la solución del problema objeto de estudio. Finalmente se presenta el modelo de localización de entidades: sus distintas formulaciones, haciendo énfasis en el caso multiobjetivo, y algunas de las técnicas que se utilizan para resolverlo.

2.1. Programación Multiobjetivo

La PMO aborda situaciones en las que existen conflictos o compromisos entre diferentes objetivos que deben ser considerados. En lugar de buscar una única solución óptima, se busca un conjunto de soluciones que representen diferentes compromisos entre los objetivos. Estas soluciones se conocen como "soluciones Pareto-óptimas" y se caracterizan por no poder mejorar un objetivo sin empeorar al menos uno de los otros. En este tipo de problemas, se busca encontrar un conjunto de soluciones que representen el mejor compromiso entre los diferentes objetivos, ya que no existe una única solución óptima.

La programación multiobjetivo tiene sus raíces en los trabajos pioneros de los matemáticos Wilfredo Pareto, Britannica 2020, quien introdujo el concepto de eficiencia de Pareto, que establece que una solución es óptima si no se puede mejorar en al menos un objetivo sin empeorar en otro. Abraham Wald, por su parte, desarrolló el enfoque de la programación lineal multiobjetivo, que permitía tratar problemas con múltiples objetivos.

2.1.1. Dominancia y optimalidad de Pareto: Caso geométrico

El concepto de que un punto domina a otro está dado por una cierta comparación. En este sentido se tiene

Definición 2.1 *Sea $y_1, y_2 \in R^m$. Se dice que y_1 domina a y_2, denotado por $y_1 \succeq y_2$ si $y_1 - y_2 \in R^m_+$, o sea todas las componentes de $y_1 - y_2$ son no negativas.*

Nótese que si $m = 1$, entonces y_1 domina a y_2 si se cumple la relación clásica de orden $y_1 \geq y_2$ y para cualquier par de puntos y_1, y_2 o y_1 domina a y_2 o y_2 domina a y_1. Si $m > 1$, existen pares de puntos en que ninguno domina al otro como, por ejemplo, si $y_1 = (1,0)$ y $y_2 = (0,1)$

Sea $A \subset R^m$, el concepto de puntos de mínimo en A que se dará a continuación correspondería a generalizaciones del concepto de mínimo en espacios multidimensionales.

Definición 2.2 *Sea $A \subset R^m$. Se dice que y^* es un punto mínimo de A si no existe $y \in A$ tal que $y^* \succeq y$, $y \neq y^*$.*

Por razones teóricas y algorítmicas es complicado hallar este tipo de puntos. De ahí que se haga necesario trabajar con el concepto de dominancia fuerte y mínimo débil.

Definición 2.3 *Sea $y_1, y_2 \in R^m$. Se dice que y_1 domina fuertemente a y_2, denotado por $y_1 \succ y_2$ si $y_1 - y_2 \in R^m_{++}$, o sea que todas las componentes de $y_1 - y_2$ son positivas.*

Resulta claro que si $y_1 \not\succ y_2$ entonces $y_1 \succeq y_2$. Además, si $m = 1$, la relación $\succ$ corresponde al $>$ de la relación clásica de orden. Luego si dos puntos son distintos y $y_1 \geq y_2$ entonces $y_1 > y_2$. Si $m > 1$, existen puntos como $y_1 = (1,0)$ y $y_2 = (1,1)$ que son diferentes $y_1 \succeq y_2$ pero $y_1 \not\succ y_2$.

El concepto de punto de mínimo débil es

Definición 2.4 *Sea $A \subset R^m$. Se dice que y^* un mínimo débil de A si no existe $y \in A$ tal que $y^* \succ y$.*

Nótese que en este caso si y^* es un mínimo es un mínimo débil. Lo contrario no se cumple, considerando $A = \{(1,0),(1,-1)\}$. Como $(1,0) - (1,-1) = (0,1)$, $(1,0)$ es un mínimo débil pero no un mínimo, ya que $(0,1)$ tiene todas sus componentes no negativas, pero la primera no es positiva.

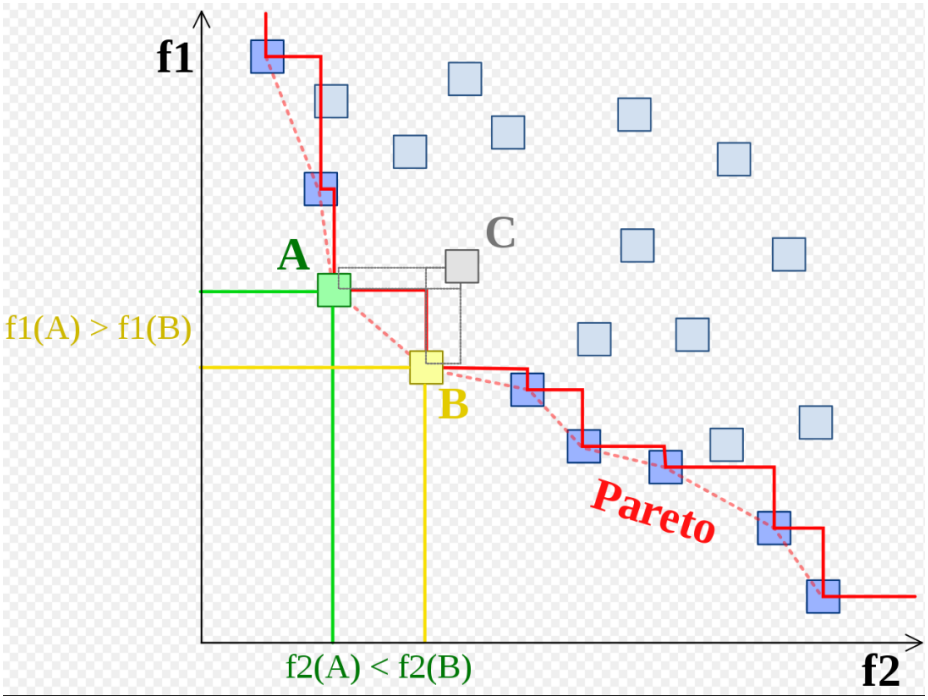

Figura 2.1: Frente Pareto

2.1.2. Dominancia y optimalidad de Pareto: Problema multiobjetivo

Sea $f : R^n \to R^m$, $M \subset R^n$. El problema multiobjetivo se denota por

$$min f(x) \text{ s.a } x \in M. \tag{2.1}$$

En un problema multiobjetivo, se busca encontrar un vector de variables de decisión $x = (x_1, x_2, ..., x_n)$ que optimice estas m funciones

$$f = (f_1(x), f_2(x), ..., f_m(x)),$$

donde cada una representa un criterio que se desea minimizar. Encontrar un punto que las minimice a todas es muy restrictivo. En la mayoría de las aplicaciones hay criterios contradictorios, como maximizar las ganancias y minimizar los costos, de ahí que los conceptos presentados en la subsección anterior sean aplicables en este contexto. Como en la Figura 2.2, extraído de Olvera Pimentel 2019 se buscan puntos solución $x^* \in M$ en el espacio de decisión, tales que $f(x)$ es un punto de mínimo de $f(M)$, o sea en el espacio objetivo. La definición matemática es

Definición 2.5 *Se dice que $x^* \in M$ resuelve el problema (2.1) si no existe $x \in M$ tal que $f(x) \neq f(x^*)$ y $f(x^*) \succeq f(x)$. Estos vectores definen la clase de puntos eficientes*

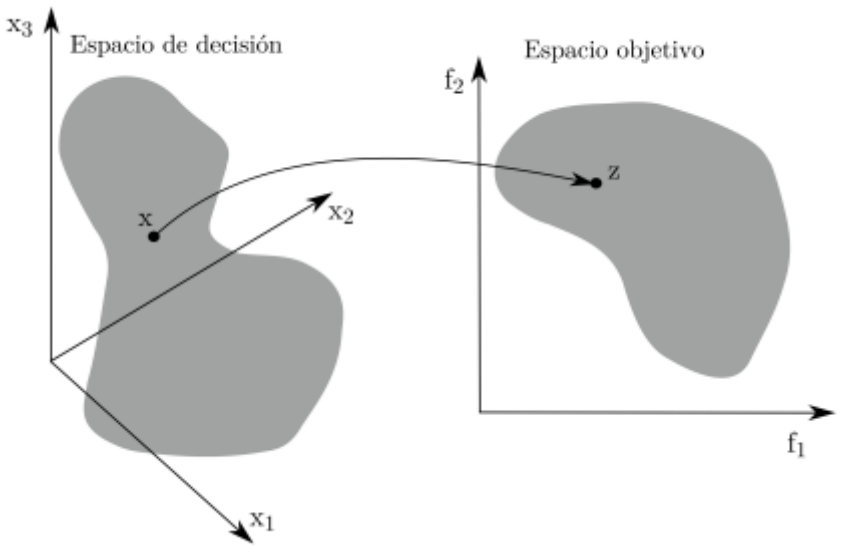

Figura 2.2: Espacio Objetivo

o de óptimos de Pareto de (2.1) *Es* $x^* \in M$ *es un punto débilmente eficiente o un óptimo débil de Pareto si no existe* $x \in M$ *tal que* $f(x^*) \succ f(x)$.

En la Figura 2.1, extraida de Overflow 2024, se ilustran estos conceptos. Con respecto al conjunto imagen se tiene el concepto de frente Pareto

Definición 2.6 *El frente Pareto del problema* (2.1) *está dado por:* $\{y = f(x^*) : x^*$ *es una solución de* (2.1)$\}$. *El frente débil Pareto del problema* (2.1) *está dado por:* $\{y = f(x^*) : x^*$ *es una solución débil de* (2.1)$\}$

La Figura 2.1 muestra elementos del frente de Pareto.

Los problemas multiobjetivo se resuelven teniendo en cuenta varios enfoques. En un caso, la programación por metas, consiste en definir umbrales u_i para las funciones $f_i, i = 2, \ldots, m$ y resolver el problema de optimización clásico mín $f_1(x)$, s. a $f_i(x) \leq u_i$, $i = 2, \ldots m$, $x \in M$. Otro enfoque consiste en ponderar las funciones y para distintos valores de los pesos $\alpha_i \geq 0$, resolver el problema $P(\alpha)$ mín $\sum_i = 1^m \alpha_i f_i(x)$, s.a $x \in M$

En cuanto a condiciones de optimalidad se tiene, ver para más detalles Miettinen 2012

Teorema 2.1 *Si* x^* *resuelve* $P(\alpha)$, *para algún* $\alpha \geq 0$, *entonces* x^* *es un punto débilmente eficiente.*
Si x^* *resuelve* $P(\alpha)$, *para algún* $\alpha > 0$, *entonces* x^* *es un punto eficiente.*

Para la condición suficiente se retoma el concepto de convexidad.

Definición 2.7 *Una función $f : \mathbb{R}^n \to \mathbb{R}$ se dice que es convexa si para todos $x, y \in \mathbb{R}^n$ y $\lambda \in [0,1]$, se tiene que :*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

Un conjunto M es convexo si para todo $x, y \in \mathbb{R}^n$ y $\lambda \in [0,1]$, se cumple que

$$\lambda x + (1-\lambda)y \in M.$$

El problema multiobjetivo (2.1) *es convexo si $f_1, \dots f_m$ son funciones convexas y M es un conjunto convexo.*

Teorema 2.2 *Si el problema multiobjetivo* (2.1) *es convexo entonces para todo punto eficiente existe $\alpha \geq 0$ tal que x^* resuelve $P(\alpha)$.*

En ambos casos se resuelven problemas de optimización clásicos para los cuales existen métodos, ver Luenberger y Ye 2021. La dificultad del primer enfoque es que se debe conocer los valores de los umbrales, lo cual puede ser solicitar mucha información. En el segundo, debe buscarse una suficiente variabilidad en los vectores de peso. Para problemas con muchos objetivos es muy costoso. Un tercer enfoque se basa en buscar direcciones en que disminuyan todas las funciones objetivo, ver Fliege y Svaiter 2000, Graña Drummond et al. 2008 Estos enfoques asumen diferenciabilidad en las funciones. Por las características del problema esta condición es muy fuerte. De ahí que se utilicen métodos que no usan derivadas como el algoritmo de Hooke y Jeeves, ver Harder 2021, el método de Rosenbrock, ver Rosenbrock 1960 y el método de las direcciones conjugadas, ver Powell 1960. Estos métodos no requieren el uso de derivadas y son útiles para resolver problemas de optimización que presentan desafíos debido a la falta de diferenciabilidad.

Cabe destacar que es importante tener una buena representatividad del frente de Pareto. La densidad de Pareto y la dispersión espacial son métricas utilizadas para evaluar la distribución y agrupamiento de los puntos Pareto en problemas de optimización multiobjetivo.

- Densidad de Pareto: La densidad de Pareto se refiere a la concentración de los puntos Pareto dentro de un cluster o conjunto de soluciones no dominadas. Esta métrica indica qué tan cerca están los puntos Pareto entre sí dentro del cluster. Una densidad de Pareto alta sugiere que los puntos están más cerca entre sí, lo que indica un agrupamiento compacto y bien definido. El cálculo de la densidad de Pareto implica calcular la distancia promedio entre los puntos Pareto más cercanos dentro del cluster. Una forma común de calcularlo es tomar

cada punto Pareto y encontrar la distancia mínima hacia otro punto Pareto dentro del mismo cluster. Luego, se promedia todas estas distancias mínimas para obtener la densidad de Pareto.

- Dispersión espacial: La dispersión espacial se refiere a la separación y dispersión de los puntos en el espacio. Esta métrica indica qué tan diseminados están los puntos en términos de distancia. Una dispersión espacial alta indica que los puntos están más alejados entre sí, mientras que una dispersión espacial baja indica que los puntos están más cercanos. El cálculo de la dispersión espacial implica encontrar la distancia mínima entre todos los pares de puntos dentro del conjunto. Se busca la distancia más corta entre todos los puntos para obtener una medida de cuán dispersos están.

Tanto la densidad de Pareto como la dispersión espacial son métricas importantes para comprender la estructura y calidad del agrupamiento de los puntos Pareto. Una buena solución de agrupamiento suele tener una densidad de Pareto alta y una dispersión espacial baja, lo que indica un agrupamiento compacto y bien definido de soluciones no dominadas.

De ahí que es importante para escoger un algoritmo que resuelva un problema de optimización multiobjetivo tener en cuenta no sólo las características del modelo sino también la calidad de la solución, que estará dada por el nivel de dispersión y densidad en el conjunto solución que resulta.

2.2. Algoritmos Genéticos

Los algoritmos genéticos (AG) son una serie de pasos organizados que describen el proceso que se debe seguir para dar solución a un problema específico Baeldung 2023. Se inspiran en la evolución biológica y su base genético-molecular. Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias, semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección. Los AG funcionan entre el conjunto de soluciones de un problema llamado fenotipo, y el conjunto de individuos de una población natural, codificando la información de cada solución en una cadena, generalmente binaria, llamada cromosoma.

Los algoritmos genéticos se distinguen por las siguientes características:

Generación de la población inicial: Se genera una población inicial de individuos (soluciones), usualmente de manera aleatoria.

Fase de evaluación: Se evalúan los individuos de la población con la función fitness. Esta función determina qué tan "aptos" son los individuos, es decir, qué tan buenas son sus soluciones al problema que se está tratando de resolver.

Fase de selección: Se seleccionan los mejores individuos. En esta fase, los individuos más aptos (es decir, aquellos con las mejores soluciones) son seleccionados para la siguiente fase. La idea es que las mejores soluciones tienen una mayor probabilidad de ser seleccionadas, lo que se asemeja a la idea de "supervivencia del más apto" en la evolución natural.
Fase de reproducción: Se cruzan los individuos seleccionados mediante la función de cruce, dando lugar a una nueva generación que va a sustituir a la anterior. Durante este proceso, las soluciones de los individuos seleccionados se combinan para generar nuevas soluciones.
Repetición: Se repiten los pasos 2 a 4 hasta que se cumpla una condición de parada, como alcanzar un número máximo de generaciones o una solución satisfactoria.

2.2.1. Breve reseña histórica

Desde los años 60, se han desarrollado independientemente varias aproximaciones que utilizan la idea de la evolución en un contexto técnico, es decir, para la optimización y simulación con computadoras Bäck y Schwefel 1993. Koza, ver Koza 1992, resume cuatro supuestos centrales para tales algoritmos evolutivos:

- Una entidad tiene la capacidad de reproducirse a sí misma.
- Existe una población de tales entidades autorreproductoras.
- Existe cierta variedad entre las entidades autorreproductoras.
- Alguna diferencia en la capacidad de sobrevivir en el entorno está asociada con la variedad.

Las estrategias de evolución (ver Bäck, Hoffmeister y Schwefel, 1991Schwefel 1981) pertenecen a una clase de conceptos utilizados para realizar una optimización robusta, que puede hacer frente, por ejemplo, a óptimos locales y funciones de optimización no diferenciables o incluso desconocidas, solo accesibles experimentalmente. Este concepto, desarrollado principalmente por Rechenberg (1973) y Schwefel (1977, 1981), se basa en una población de alternativas factibles representadas por puntos de búsqueda en un espacio vectorial $\mathbb{R}^{\varkappa}$.

En contraste con los algoritmos genéticos, un enfoque popular para los algoritmos evolutivos, las estrategias de evolución no dependen de la codificación binaria (cadenas de bits). En cambio, utilizan números de punto flotante para codificar las variables, lo que parece ser una forma más adecuada de codificar variables continuas, como las que se encuentran en los problemas de programación multiobjetivo (MOP). Aunque la recombinación es el operador genético más importante en los algoritmos genéticos, en

las primeras estrategias de evolución, la mutación era el principal operador de cambio evolutivo, ver Sánchez et al. 2022a. Sin embargo, hoy en día se asume generalmente que el concepto evolutivo, especialmente la estructura de datos para las entidades, debe generalizarse y adaptarse a la situación específica del problema, ver Salazar et al. 2019. A estos enfoques más generales se les suele llamar algoritmos evolutivos. Este artículo discute el potencial de un algoritmo evolutivo, principalmente basado en ideas derivadas de las estrategias de evolución, para la programación multiobjetivo, ver de Sardi 2011

En el mundo actual lleno de tecnología y ciencia, los algoritmos evolutivos han encontrado una amplia gama de aplicaciones prácticas. A continuación, se presentan algunos ejemplos de cómo se han aplicado estos algoritmos en diferentes áreas.

En el campo de la reconfiguración óptima de redes de distribución eléctrica, los algoritmos evolutivos han demostrado ser particularmente útiles. Han sido empleados para optimizar la configuración de las redes eléctricas, lo que ha resultado en mejoras en la confiabilidad y los perfiles de tensión, reducción de pérdidas, mantenimiento de balances de cargas, aislamiento de fallas y aceleración de la restauración del servicio,ver Sánchez et al. 2022b. En el ámbito del comercio electrónico, estos algoritmos se han utilizado para predecir qué productos serán más necesarios y, por tanto, más demandados en una situación concreta, ver Lab 2020.

2.3. NSGA-II

Los algoritmos evolutivos también se utilizan para resolver MOP. Algunos de los MOEAs del estado del arte más populares son el NSGA-II (*nondominated sorting Genetic algorithm II*) Chugh et al. 2011; el SPEA2 (*strength Pareto evolutionary algorithm 2*) Villamar y Saltos 2014 ; el IBEA (*indicador-based evolutionary algorithm*), ver Botía 2005; el MODE (*multi-objetive differential evolution*); el SMPSO (*speed-constrained multiobjetive particle swarm opimization*); y el MOEA/D *multi-objetive evolutionary algorithm based on descomposition*, ver Zhang y Li 2007.

El algoritmo NSGA-II es una evolución del original NSGA presentado en 1942. Este algoritmo está basado en el método de ranking de Pareto, en el cual cada individuo de la población es ordenado teniendo en cuenta el número de soluciones que lo dominan. Su funcionamiento se describe a continuación:

- **Inicialización**: Se genera una población inicial de soluciones aleatorias.
- **Evaluación**: Se evalúa la aptitud de cada solución en función de los objetivos que se quieren optimizar.
- **No dominación**: Se realiza una clasificación de las soluciones en distintos niveles de no dominación.

- **Asignación de aptitud**: Se asigna una aptitud a cada solución en función de su nivel de no dominación y de su posición dentro de cada nivel.
- **Selección**: Se seleccionan las soluciones más aptas para reproducirse. Este proceso se realiza teniendo en cuenta la clasificación de no dominación y la asignación de aptitud.
- **Cruce y mutación**: Se aplican operadores genéticos como el cruce y la mutación para generar nuevas soluciones a partir de las seleccionadas.
- **Reemplazo**: Se reemplazan las soluciones menos aptas de la población original con las nuevas soluciones generadas.
- **Repetición**: Se repiten los pasos 2 a 7 hasta que se alcance un criterio de terminación, como un número máximo de generaciones o un nivel de convergencia deseado.
- **Resultados**: Al finalizar el algoritmo, se obtiene un conjunto de soluciones que representan el frente de Pareto, es decir, un conjunto de soluciones no dominadas entre sí y que representan diferentes compromisos entre los objetivos.

La figura 2.3 describe describe la evolución de la población en este algoritmo, la misma fue extraída de pymoo 2021

En particular, los AG han demostrado ser efectivos para resolver problemas de localización, que son problemas de optimización que implican la asignación de recursos a ubicaciones de manera que se optimice uno o varios objetivos dados, referirse a Cantlebary y L. Li 2020. Un estudio reciente utilizó un algoritmo genético para resolver el problema de localización de cobertura máxima (MCLP), que implica encontrar la ubicación óptima de un número dado de instalaciones dentro de un conjunto de clientes, ver Baeldung 2023. El algoritmo propuesto demostró ser eficaz en términos de porcentaje de cobertura y tiempo de cálculo para encontrar las soluciones en casi todos los casos . Otro estudio propuso un nuevo modelo para el problema de localización de terminales de autobús utilizando análisis envolvente de datos con enfoque de programación multiobjetivo. En este modelo, se investigaron las ubicaciones óptimas para desplegar terminales, ver Atta et al. 2018.Otro estudio aplicó el NSGA-II al problema de localización y asignación del centro neurálgico, ver Yin 2017.

Existen varias variantes del algoritmo NSGA-II que han sido propuestas para mejorar su rendimiento. Por ejemplo,en Zheng y Doerr 2023 se proponen dos algoritmos NSGA-II mejorados para encontrar subconjuntos de biomarcadores que exhiben diferentes compensaciones entre precisión y número de características. Otro estudio introdujo una variante de NSGA-II que mejora la complejidad temporal al gestionar la contabilidad de una manera mejor que el algoritmo básico, ver .

En comparación con otros algoritmos, los resultados de las simulaciones en varios problemas difíciles muestran que el NSGA-II es capaz, en la mayoría de los problemas, de encontrar una mejor distribución de soluciones y una mejor convergencia cerca del frente Pareto-optimal verdadero en comparación con la estrategia evolutiva archivada por Pareto y el algoritmo evolutivo fuerza-Pareto, referirse a Fang et al. 2018, Soleimani 2020.

En términos de complejidad temporal, NSGA-II tiene una complejidad de $O(MN^2)$, donde M es el número de objetivos y N es el tamaño de la población. Sin embargo, en los últimos años, varios investigadores han propuesto variantes del NSGA-II que tienen una complejidad temporal de $O(MN\log^{M-1}N)$. Esto significa que el NSGA-II y sus variantes pueden manejar eficientemente problemas con un gran número de objetivos y una gran población. La complejidad $O(MN^2)$ del algoritmo NSGA-II se debe a dos componentes principales del algoritmo: el ordenamiento no dominado y el cálculo de la distancia de aglomeración.

Ordenamiento no dominado: El algoritmo clasifica las soluciones en diferentes frentes no dominados. Para hacer esto, necesita comparar cada solución con todas las demás para verificar si una solución domina a otra o no. Esto da lugar a una complejidad de $O(N^2)$.

Cálculo de la distancia de aglomeración: Después del ordenamiento no dominado, el algoritmo calcula la distancia de aglomeración para cada solución. Este cálculo también implica comparar cada solución con todas las demás, lo que de nuevo da lugar a una complejidad de $O(N^2)$. Estos dos componentes se realizan para cada objetivo, por lo que la complejidad total del algoritmo es $O(MN^2)$.

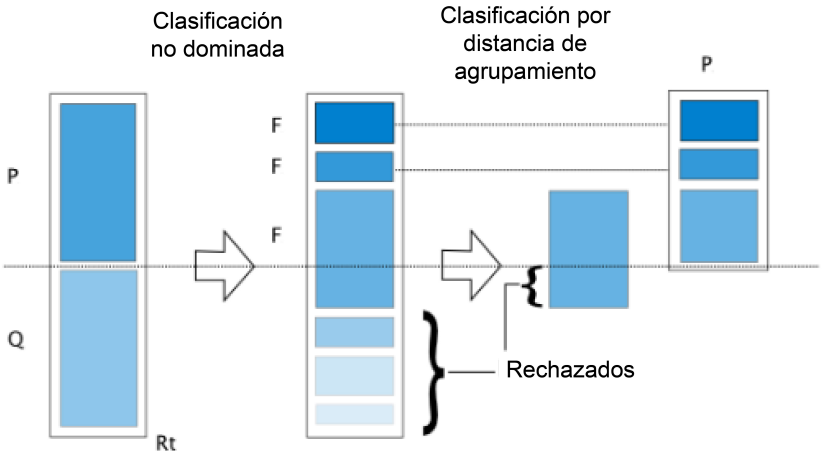

Figura 2.3: NSGA-II

2.4. Pymoo

La biblioteca pymoo es un marco de trabajo de código abierto para la optimización en Python. En esta sección se ofrece un resumen de las funcionalidades de la misma, para más detalles, ver Blank y K. Deb 2020. Esta biblioteca ofrece algoritmos de vanguardia para la optimización de problemas con uno o varios objetivos. En cuanto a algoritmos, pymoo proporciona una variada cantidad. Por ejemplo para problemas de optimización de un solo objetivo se tienen: GA, DE, PSO, Nelder Mead, Pattern Search, BRKGA, ES, SRES, ISRES, CMA-ES, G3PCX, para problemas multiobjetivo: NSGA-II, R-NSGA-II, NSGA-III, R-NSGA-III, U-NSGA-III, MOEA/D, AGE-MOEA, AGE-MOEA2, RVEA, SMS-EMOA.

El funcionamiento de estos algoritmos se puede comprobar en la biblioteca de problemas de prueba de pymoo. En ella se incluyen modelos con un solo objetivo como Ackley, Griewank, Rastrigin, Rosenbrock, Zakharov y multiobjetivo clásicos para comprobar la eficiencia de los algoritmos como BNH, OSY, TNK, Truss2d, Welded Beam, ZDT , destacar que también se incluyen problema dinámicos.

Pymoo proporciona una variedad de operadores de muestreo, selección, cruce y mutación, así como operadores de reparación estos, estos se utilizan para modificar soluciones que violan las restricciones del problema para que sean factibles. Algunos ejemplos de operadores incluyen muestreo aleatorio, LHS para muestreo, selección aleatoria, torneo binario para selección, SBX, UX, HUX, DE Point, Exponential, OX, ERX para cruce, y mutación polinomial, bitflip, mutación inversa para mutación.

Pymoo proporciona varias herramientas de visualización para ayudar a analizar los resultados de la optimización multiobjetivo. Algunos ejemplos de herramientas de visualización incluyen Scatter Plot (2D/3D/ND), Parallel Coordinate Plot (PCP), Radviz, Star Coordinates, Heatmap, Petal Diagram, Spider Web / Radar, Video. En el software de esta tesis se utilizó esta herramienta para observar la población inicial y como la misma iba mejorando con cada generación.

Pymoo permite la personalización de los tipos de variables, incluyendo variables binarias, discretas, de permutación, mixtas y personalizadas. También proporciona ejemplos de inicialización sesgada y del problema del viajante de comercio.

2.5. Problema de localización

Para abordar esta temática de la forma más general, se hace referencia a Drezner y Horst W. Hamacher 2004. Este libro proporciona una introducción básica a las técnicas de análisis de ubicación y cubre tanto la teoría como la práctica. Además, incluye material complementario que amplía el contenido presentado.

Para identificar problemas de ubicación, utilizamos un esquema de clasificación propuesto en la literatura de la teoría de ubicación por Hamacher y Nickel H. W.

Hamacher y Nickel 1998. La clasificación contiene cinco posiciones:
Pos 1. | Pos 2. | Pos 3. | Pos 4. | Pos 5. donde:

- Pos 1: Número de nuevas instalaciones (por ejemplo, 1 para problemas de ubicación de una sola instalación),

- Pos 2: Tipo de problema de ubicación (por ejemplo, problema de ubicación planar (P), discreto (D) o de red (N)),

- Pos 3: Características del problema de ubicación (por ejemplo, pesos positivos, $v > 0$, es decir, $v_i > 0$ para todo $i = 1, \ldots, m$; pesos iguales a uno, $v = 1$, es decir, $v_i = 1$ para todo $i = 1, \ldots, m$; pesos negativos, $w < 0$, es decir, $w_i < 0$ para todo $i = 1, \ldots, m$; solo puntos de atracción, (+); puntos de atracción y repulsión, (+, -); tipo del conjunto factible, por ejemplo $X = \mathbb{R}^2$ o X representa un politopo),

- Pos 4: Definición de las distancias (por ejemplo, métrica de Manhattan, d_1; métrica máxima, d_∞; métrica euclidiana, d_2; métrica euclidiana al cuadrado, d_2^2; métrica l_p, d_p; calibre poliédrico, μ; calibres poliédricos mixtos, μ_i; métrica de atracción d y métrica de repulsión inducida por el calibre μ, (d, μ)),

- Pos 5: Enlace de distancias individuales (por ejemplo, problema de la mediana (mediana), problema del centro (centro), problema del vector (vector sEff, vector Eff o vector wEff)).

2.5.1. Aplicaciones

El problema de localización, o Location Problem, tiene diversas aplicaciones en dominios:

- Ubicación de instalaciones: Este problema implica determinar la ubicación óptima para construir instalaciones como almacenes o fábricas para minimizar los costos de transporte y maximizar la eficiencia de la cadena de suministro Kalyanmoy Deb et al. 2002,Kalyanmoy Deb et al. 2002. Los desafíos específicos incluyen la consideración de las capacidades de las instalaciones y la variabilidad en la demanda y los costos de transporte. Los algoritmos genéticos, como NSGA-II, se han utilizado para abordar estos retos mediante la búsqueda de soluciones óptimas en el espacio de búsqueda Atta et al. 2018.

- Ruteo de vehículos: En este contexto, el problema de localización puede implicar la determinación de rutas óptimas de vehículos para minimizar los costos de transporte y el tiempo de viaje Monika Mangla 2015. Las dificultades pueden incluir la variabilidad en las condiciones del tráfico y las restricciones en los tiempos de entrega.

- Optimización de redes de distribución: El problema de localización también puede aplicarse a la optimización de redes de distribución, como las redes eléctricas o las redes logísticas Atefeh Taghavi 2021, Yin 2017. Los desafíos pueden incluir la necesidad de equilibrar los costos y la eficiencia, así como considerar las restricciones físicas y operativas

- Gestión de residuos: En este contexto, el problema de localización puede implicar determinar las ubicaciones óptimas para las instalaciones de gestión de residuos para minimizar los costos y maximizar la eficiencia del sistema Kalyanmoy Deb et al. 2002. Los retos pueden incluir consideraciones ambientales y regulaciones locales.

Es un problema de gran actualidad pues el mismo se trabaja en varias esferas de la industria, algunos ejemplos son:
El trabajo titulado "A new algorithm for solving planar multiobjective location problems involving the Manhattan norm" presenta un algoritmo efectivo para resolver problemas de ubicación multiobjetivo planos sin restricciones utilizando la norma de Manhattan. El algoritmo genera todas las soluciones eficientes mediante la unión de rectángulos y segmentos de línea., ver Alzorba et al. 2017

El trabajo titulado "Algorithms and Decomposition Methods for Multiobjective Location and Approximation Problems"se enfoca en la optimización multiobjetivo en combinación con el análisis de ubicación. Se estudian problemas de ubicación y aproximación multiobjetivo extendidos, que tienen diversas aplicaciones en economía, ingeniería y física. Se demuestran afirmaciones de dualidad y se descompone el problema para obtener subproblemas de ubicación multiobjetivo., ver Alzorba 2015

"Multiple Tasks for Multiple Objectives: A New Multiobjective Optimization Method via Multitask Optimization", propone un nuevo enfoque para resolver problemas de optimización multiobjetivo a través de la optimización multitarea. Se trata el problema de optimización multiobjetivo como un problema de optimización multitarea y se utiliza un método basado en esta idea, ver «Special Issue 'Latest Advances and Applications of Multi-Objective Optimization Techniques'» 2023 Veamos primeramente el concepto geométrico de no dominancia y optimalidad.

2.5.2. Problema general

De manera general se formula un problema de localización de la siguiente manera: Considere m puntos en el plano,

$$a^1 := (a_1^1, a_2^1), ..., a^m := (a_1^m, a_2^m) \in \mathbb{R}^2$$

que representan algunas localizaciones. El conjunto

$$A := a^1, ..., a^m$$

representa el conjunto de todas las instalaciones existentes. En muchos problemas de análisis de ubicación, quien toma decisiones busca nuevas instalaciones de tal manera que las distancias entre las nuevas instalaciones y las instalaciones existentes sean mínimas en cierto sentido. Una posibilidad es que estas distancias estén descritas por una norma apropiada.

$$\|\cdot\| : \mathbb{R}^2 \to \mathbb{R}$$

Ahora, hay varias posibilidades para definir un problema de ubicación. Por ejemplo, podemos considerar un problema de mediana planar que está definido por

$$\sum_{i=1}^{m} v_i \cdot \|x - a^i\| \to \min_{x \in \mathbb{R}^2} \tag{2.2}$$

donde v_i es una ponderación positiva asociada al punto a^i para todo $i = 1, ..., m$.
En su primera y más simple forma, tal problema (1) fue planteado por el jurista y matemático Fermat en 1629. Él preguntó por el punto que realiza la suma mínima de distancias desde tres puntos dados. En 1909, este problema apareció, en una forma ligeramente generalizada, en el trabajo pionero "Uber den Standort der Industrien" de Weber, ver Weber 1909. Por lo tanto, el problema dado por (1) se llama problema de Fermat-Weber en la literatura de la teoría de la ubicación e involucra, en su formulación original, la norma euclidiana como función de distancia.

Otra clase de problemas de ubicación son los problemas de centro planar. El objetivo es minimizar el máximo de las distancias entre las nuevas instalaciones $x \in \mathbb{R}^2$ y las instalaciones existentes $a_1, \ldots, a_m$, es decir, consideramos el siguiente problema

$$\min_{x \in \mathbb{R}^2} \max_{i=1,...,m} \{v_i \cdot \|x - a_i\|\} \tag{2}$$

con pesos $v_i > 0$ para todo $i = 1, \ldots, m$. Para el caso especial con $v_i = 1$ para todo $i = 1, \ldots, m$ y la norma euclidiana como medida de distancia, el problema de ubicación

(2) se conoce como el problema del círculo más pequeño o problema del círculo de cobertura mínima en la literatura de la teoría de la ubicación. Las aplicaciones de este modelo aparecen, por ejemplo, en la Gestión de Emergencias.

2.5.3. Variantes del problema de localización.

En particular, los AG han demostrado ser efectivos para resolver problemas de localización, que son problemas de optimización que implican la asignación de recursos a ubicaciones de manera que se optimice uno o varios objetivos dados, referirse a Shi y Lin 2022. Un estudio reciente utilizó un algoritmo genético para resolver el problema de localización de cobertura máxima (MCLP), que implica encontrar la ubicación óptima de un número dado de instalaciones dentro de un conjunto de clientes, ver Baeldung 2023. El algoritmo propuesto demostró ser eficaz en términos de porcentaje de cobertura y tiempo de cálculo para encontrar las soluciones en casi todos los casos . Otro estudio propuso un nuevo modelo para el problema de localización de terminales de autobús utilizando análisis envolvente de datos con enfoque de programación multiobjetivo. En este modelo, se investigaron las ubicaciones óptimas para desplegar terminales, ver Atta et al. 2018.Otro estudio aplicó el NSGA-II al problema de localización y asignación del centro neurálgico, ver Yin 2017.

Existen varias variantes del problema de localización, a continuación se presentan algunas de estas variantes:

- Problema de Localización de Instalaciones con Capacidad Ilimitada (UFLP): Este problema tiene como objetivo determinar qué instalaciones se abrirán teniendo en cuenta el balance entre los costos fijos de operación y los costos variables de envío con el fin de minimizarlos Teo y Shumshy 2008.

- Problema de Localización y Ruteo con Restricciones de Capacidad (CLRP): Este problema combina el problema de localización de instalaciones con restricciones de capacidad (CFLP) y el problema de ruteo de vehículos con múltiples depósitos (MDVRP). Se consideran restricciones de capacidad para las plantas a localizar con el objetivo de satisfacer la demanda de los clientes ubicados en determinada zona.Archetti et al. 2010

- Modelo de Weber sobre Zonas Específicas: Este modelo extiende el problema de Weber sobre una superficie esférica con adaptación de restricciones para delimitar zonas para la localización de instalaciones. Las principales restricciones en problemas de localización de instalaciones restringen el establecimiento de las mismas en zonas prohibidas, zonas congestionadas o bien regiones de barrera.Murray et al. 2010

- Problema de Localización de Instalaciones con Capacidad Limitada (CFLP): En este problema, cada instalación tiene una capacidad máxima y se busca minimizar los costos de apertura de las instalaciones y los costos de asignación de los clientes a las instalaciones abiertas.Berman y Krass 2007

- Problema de Localización de Fermat-Weber: Este problema busca minimizar la suma de las distancias ponderadas entre las instalaciones y los clientes. Existen variantes de este problema que consideran diferentes normas para la distancia y restricciones de tipo caja. Yang y S. Deb 2008

- Problema de Localización y Ruteo con Flota Heterogénea (HFVRP): Este problema combina la localización de instalaciones con el ruteo de vehículos, considerando que la flota de vehículos puede ser heterogénea. Dependiendo de la variante, la flota de vehículos puede ser limitada o ilimitada.

- Problema de Localización y Ruteo con Múltiples Objetivos: Este problema considera la optimización simultánea de diversos objetivos, incluyendo perspectivas económicas, sociales y ambientales.

- Problema de Localización con zonas prohibidas: En el mismo se aborda el problema de identificar y delimitar las áreas restringidas o prohibidas. Un ejemplo es el artículo Guntherm 2017 donde se trabaja en la Bahía de La Habana, Cuba. Estas zonas restringidas son áreas donde ciertas actividades, como la navegación o la pesca, están prohibidas o restringidas debido a consideraciones de seguridad, protección ambiental u otras razones.

Capítulo 3

Propuesta

En el presente capítulo se presenta la propuesta de solución al problema planteado en el capítulo anterior. Se definen los modelos matemáticos que se utilizarán para resolver el problema, así como el algoritmo que se empleará. Primeramente se presentan cada uno de los modelos y cómo se calcula la distancia entre puntos para cada caso.Luego se trata la convexidad del problema. Por último se presenta el algoritmo.

3.1. El Modelo matemático

Recordando el problema de la investigación, se tienen un conjunto de puntos A y otro conjunto de puntos B, se quiere encontrar la mejor ubicación de un nuevo punto que cumpla que está lo mas cerca posible a todos los puntos del conjunto A, y lo más lejos a todos los puntos del conjunto B. Se asume que el río está dado por un rectángulo en el centro de una región rectangular y que hay dos puentes que lo atraviesan, siendo estas las únicas vías para ir de una rivera a otra. Para el cálculo de la distancia entre puntos se utiliza distancia Manhattan si el área donde considera el problema es una ciudad. En caso de que la región se encuentre en una zona rural se utiliza la norma euclideana.

Como se planteó en la sección 1.4 los problemas de localización se agrupan usando un vector de 5 componentes. En el contexto de esta tesis, estamos interesados en un problema de ubicación planar (Pos 2) con una sola nueva instalación (Pos 1). El problema tiene características especiales (Pos 3) ya que estamos buscando un punto que esté lo más cerca posible de todos los puntos de un conjunto dado A y lo más lejos posible de todos los puntos de B, dentro de una región que contiene un río con dos puentes, representado de manera rectangular donde no se permite la colocación de puntos.

En cuanto a la definición de las distancias (Pos 4), estamos interesados en las métricas de Manhattan (d_1) y euclidiana (d_2). Finalmente, en términos del enlace

de las distancias individuales (Pos 5), estamos buscando minimizar la suma de las distancias a los puntos en A y maximizar la suma de las distancias a los puntos en B. Esto se puede considerar como un problema de vector (sEff -vector).

Además, se desea evaluar el problema donde se tenga como funciones objetivos la distancia más cercana a los puntos A, en este caso no hay puntos B, y viceversa.

3.2. Cálculo de distancias:

Para cada uno de los modelos tenemos funciones objetivos distintas a minimizar, en cada uno de los casos f representa un vector de distancias de x a cada uno de los puntos del conjunto A y B respectivamente, siendo $x \in \mathbb{R}^2$. Entonces la distancia se calcula:

$$d_l(z,y) = \begin{cases} \|z-y\|_l & \text{si z , y están en la misma rivera} \\ min[\|z-p_1\|_l + \|p_1'-y\|_l + \|p_1-p_1'\|_l, & \\ \|z-p_2\|_l + \|y-p_2'\|_l + \|p_2-p_2'\|_l] & \text{en otro caso} \end{cases} \tag{3.1}$$

donde $l = 1, 2$, p_2, p_1 son las entradas del puente 1 y 2 en la rivera donde está z y p_1' y ,p_2' las salidas de los puentes en la rivera contraria.

En la imagen 3.1 se muestran representados por rectángulos rojos el caso de los puntos cuando están en la misma rivera, y el cálculo de la distancia entre estos no involucra los puentes. Para el segundo caso se consideran los puntos Z y Y en lados opuestos del río y se muestra como calcular la distancia teniendo en cuenta los puentes.

Para representar el río y los puentes se reciben las coordenadas de los mismos. Para simplificar el trabajo se representa el río con un rectángulo, de modo que los puentes se representan con pequeños rectángulos sobre este río, de forma tal que se puede ver la representación del rio por tres zonas. La zona 1 abarca donde comienza el río hasta el primer puente, zona 2, el área del río entre los puentes, y la última zona del segundo puente hasta donde finaliza el río. Como trabajo futuro se considerará buscar figuras para representar el río que se ajusten a las peculiaridades de cada problema, así como brindar la posibilidad de rotar el río y trasladar el mismo. La representación de mismo se muestra en la figura 3.2

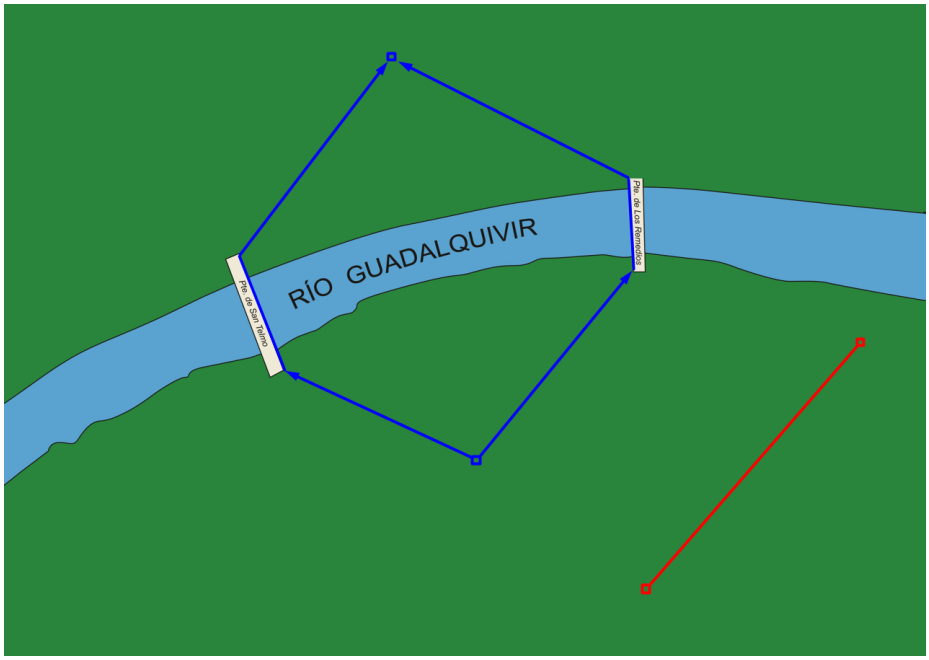

Figura 3.1: Ejemplo del cálculo de las distancias.

3.3. Primer modelo:

variables:
(x, y): Coordenadas del punto que se quiere colocar.

Funciones objetivo:
Minimizar la distancia al conjunto de puntos A. Denotamos los puntos en A como

$$A = \{a_1, a_2, ..., a_p\} \subset R^2$$

Maximizar la distancia al conjunto de puntos B. Denotamos los puntos en B como

$$B = \{b_1, b_2, ..., b_q\} \subset R^2$$

La funcion objetivo se define como:

$$f_l(x,y) = (d_l((x,y),a_1), d_l((x,y),a_2)...(d_l((x,y),a_p), -d_l((x,y),b_1), -d_l((x,y),b_2)... - d_l((x,y),b_q))) \tag{3.2}$$

El punto (x, y) debe estar dentro de la región rectangular $[0, a] \times [0, b]$
Restricciones:
El punto no puede estar dentro del rectángulo que representa el río. Asumimos que

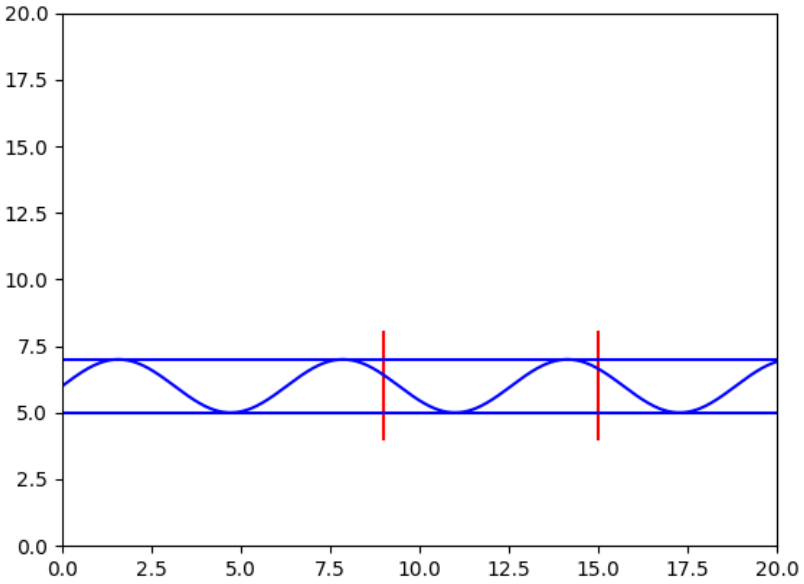

Figura 3.2: Representación del río.

el río está dado por la franja $[0, a] \times [yr_1, yr_2]$ donde $0 \leq yr_1 < yr_2 \leq b$ De ahí que las primeras restricciones son:

$$x \geq 0, x \leq a, y \geq 0, y \leq b$$

$$(x \leq x_{r1}) \text{ o } (x \geq x_{r2}),\ (y \leq y_{r1}) \text{ o } (y \geq y_{r2})$$

Dado que son solo dos casos, se resuelven los problemas:

$$\begin{aligned} &\text{minimizar } f_l(x, y) \\ &\quad \text{s.a } x \geq 0, x \leq a, y \geq 0, y \leq yr_1 \end{aligned}$$

y

$$\begin{aligned} &\text{minimizar } f_l(x, y) \\ &\quad \text{s.a } x \geq 0, x \leq a, y \geq yr_2, y \leq b \end{aligned}$$

Nótese que tienen restricciones de caja y $p+q$ objetivos.

Se propone un segundo modelo en el que solo se consideran dos objetivos para graficar el frente de Pareto. Para ello, se minimiza la media de las distancias de (x, y) a todos los puntos del conjunto A y la media de distancias de (x, y) a todos los puntos del conjunto B.

3.4. Segundo modelo:

variables:
(x, y): Coordenadas del punto que se quiere colocar.

Funciones objetivo:

Minimizar la distancia al conjunto de puntos A

$$A = \{a_1, a_2, ..., a_p\} \subset R^2$$

Maximizar la distancia al conjunto de puntos B

$$B = \{b_1, b_2, ..., b_q\} \subset R^2$$

Las funciones objetivo se define como:

$$\text{mín}(f_1(x, y), f_2(x, y)) \tag{3.3}$$

donde se define f_1, f_2 como:

$$f_1(x, y) = \frac{1}{p} \sum_{i=1}^{p} d((x, y), a_i) \tag{3.4}$$

$$f_2(x) = -\frac{1}{q} \sum_{i=1}^{q} d((x, y), b_i) \tag{3.5}$$

Restricciones:
El punto no puede estar dentro del rectángulo que representa el río. Asumimos que el río está dado por la franja $[0, a] \times [yr_1, yr_2]$ donde $0 \leq yr_1 < yr_2 \leq b$ De ahí que las primeras restricciones son:

$$x \geq 0, x \leq a, y \geq 0, y \leq b$$

$$(x \leq x_{r1}) \text{ o } (x \geq x_{r2}),\ (y \leq y_{r1}) \text{ o } (y \geq y_{r2})$$

Dado que son solo dos casos, se resuelven los problemas:

$$\begin{aligned} &\text{minimizar } (f_1(x,y), f_2(x,y)) \\ &\quad \text{s.a } x \geq 0, x \leq a, y \geq 0, y \leq yr_1 \end{aligned}$$

y

$$\begin{aligned} &\text{minimizar } (f_1(x,y), f_2(x,y)) \\ &\quad \text{s.a } x \geq 0, x \leq a, y \geq yr_2, y \leq b \end{aligned}$$

La elección de trabajar con el problema de dos objetivo en lugar de múltiples objetivos se basa principalmente en la idea de simplificar el problema. Cuando se trabaja con muchos objetivos, el problema puede volverse engorroso. Esto se debe a que cada objetivo adicional introduce una nueva dimensión en el espacio de funciones, lo que puede aumentar exponencialmente el número de soluciones posibles. Además, puede ser difícil equilibrar múltiples objetivos entre sí, ya que mejorar uno puede empeorar otro.

Además, la representación del frente de Pareto, que es el conjunto de todas las soluciones no dominadas, se puede graficar cuando sólo hay dos objetivos. En un gráfico bidimensional, cada punto en el frente de Pareto representa la evaluación de la función objetivo, y es fácil ver cómo cambiar un objetivo afecta al otro. Sin embargo,como se tiene en general muchos puntos y una función objetivo por cada uno de ellos, visualizar el frente de Pareto es imposible para el modelo 1.

3.5. Propiedades del problema

En esta sección se analizan las propiedades de diferenciabilidad y convexidad del problema. Su cumplimiento o no tiene una gran influencia a la hora de escoger el método para su solución. En cuanto a diferenciabilidad, para el cálculo de distancias entre los puntos, en el problema se utilizan las normas 1 y 2. Cuando se utiliza la norma 1 en el cálculo de distancias entre puntos, las funciones objetivo pierden la propiedad de ser diferenciables debido a la aparición de la función modular y el radical respectivamente en el cálculo de la distancia.

Una función se dice que es convexa si el segmento de línea entre cualquier par de puntos en la función se encuentra por encima o en el gráfico. Matemáticamente, una función

$$f : \mathbb{R}^n \to \mathbb{R}$$

es convexa si para todos

$$x, y \in \mathbb{R}^n$$

y para todos

$$\lambda \in [0,1], \text{ tenemos}$$

:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$

En los problemas multiobjetivos también es importante la convexidad. Este concepto conlleva a que que todas las funciones que se quiere minimizar lo sean. Ahora, en este problema, se trata de encontrar un frente de Pareto que esté lo más cerca posible de todos los puntos en el conjunto A y lo más lejos posible de todos los puntos en el conjunto B. La función de distancia, es de hecho convexa. Sin embargo cuando se está maximizando una función convexa (o, de manera equivalente, minimizando el negativo de una función convexa), el problema ya no es convexo.

Para ilustrar este hecho en el caso de la norma euclideana, se considera un ejemplo sencillo en dos dimensiones. En el caso donde se quiere colocar un punto de la forma (x, ay) que esté cerca de puntos de la forma $(ax, ay) \in A$ entonces la función queda $f = (x - ax)^2$, que corresponde al caso convexo 3.3(a), figura extraída de Genérica 2024. Para mostrar el otro caso, suponga que se quiere colocar un punto (x, by) donde los puntos a los que se quiere estar alejados son de la forma (bx, by) la función quedaría $f = -(x - bx)^2$, que corresponde al caso cóncavo 3.3(b),

Este ejemplo muestra cómo el cambio de signo en la distancia afecta la convexidad de la función objetivo. Las funciones de maximización de distancia no son convexas, ya que su forma geométrica es cóncava.

Cuando se utiliza la norma 1, las distancias se calculan utilizando funciones modulares. Considerando los mismo conjuntos A y B del caso anterior, al colocar un punto que se quiera esté cerca de puntos del conjunto A, la función de distancia es $f = |x - bx| + |y - by|$, que es convexa, ver 3.4(a), el caso restante resulta de colocar un punto y se quiera esté lejos de los puntos de B, para este caso la función de distancia queda $f = -|x - bx| - |y - by|$, caso cońcavo, ver 3.4(b)

3.6. Algoritmo

El algoritmo empleado es una caso particular del algoritmo NSGA-II de la biblioteca de python pymoo Blank y K. Deb 2020. Las razones por las cuales se decide usar este algoritmo por encima de otro son las siguientes:

Eficiencia: NSGA-II tiene una velocidad de ejecución rápida, una baja complejidad computacional y es fácil de implementar Q. Zhang y H. Li 2007. Esto lo hace adecuado para problemas que requieren una gran cantidad de cálculos.

Diversidad de soluciones: NSGA-II utiliza un enfoque de clasificación no dominado y operadores de cruce y mutación para mantener la diversidad en la población de

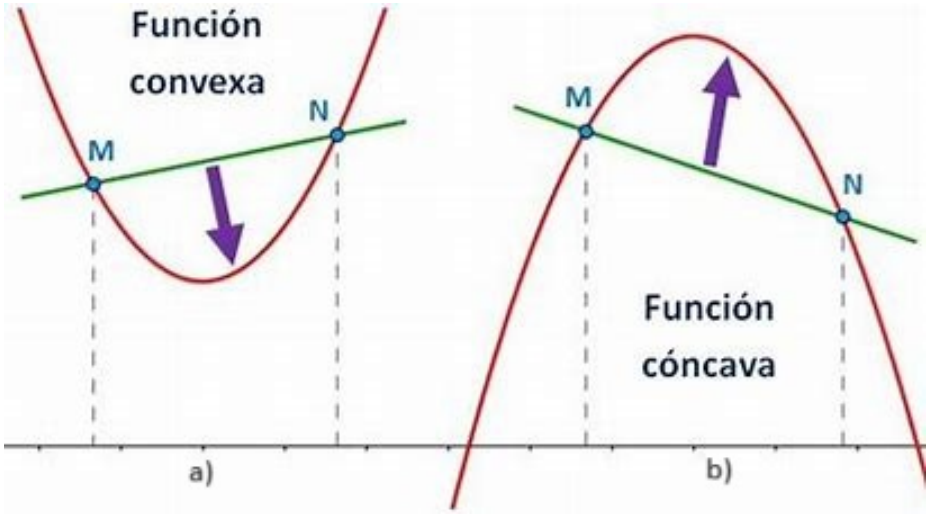

Figura 3.3: Convexidad

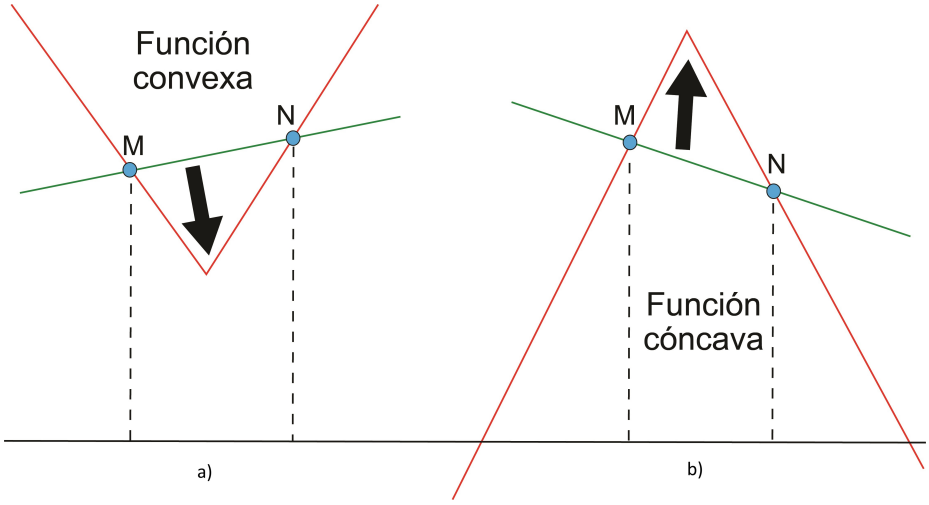

Figura 3.4: Convexidad

soluciones Ma et al. 2023. Esto asegura que se explore un amplio rango de posibles soluciones.

Calidad de las soluciones: Los resultados de simulación en problemas de prueba difíciles muestran que NSGA-II es capaz de encontrar una mejor distribución de soluciones y una mejor convergencia cerca del frente Pareto-optimal verdadero en comparación con otros algoritmos.

Aplicabilidad: NSGA-II ha demostrado ser efectivo en una variedad de problemas de optimización multiobjetivo, incluyendo problemas combinatorios Verma et al. 2021.

Además la no diferenciabilidad del problema hace que no se puedan aplicar algoritmos como los propuestos en Fliege y Grana Drumound.

En esta tesis se utilizó NSGA-II: Non-dominated Sorting Genetic Algorithm como algoritmo genético de entre toda la familia de algoritmos NSGA. Se utilizaron algunos tipos de cruzamientos de los brindados por la biblioteca como son: SBX, PointCrossover, TwoPointCrossover, SinglePointCrossover, ExponentialCrossover, UniformCrossover. Se usó solo un tipo de mutación: PolynomialMutation. Como método de selección se empleó TournamentSelection, o selección basada en torneo. Para más detalles de la coniguración de los algoritmos puede referirse a Blank y K. Deb 2020.

Capítulo 4

Detalles de Implementación y Experimentos

4.1. Implementación

Un primer enfoque para construir el algoritmo fue convertirlo a un problema de localización más sencillo, en este sólo se contemplaba un conjunto de puntos A de los cuales se quería que el punto a colocar estuviera lo más cerca posible. De esta forma el problema que se quiere resolver es encontrar el centroide del conjunto A.

Cuando se busca el centroide de un conjunto de puntos, se está resolviendo un problema de optimización en el que se minimiza la suma de las distancias euclidianas entre el centroide y cada punto en el conjunto. Sea el conjunto

$$A = a_1, a_2, ..., a_n,$$

en un espacio euclidiano el centroide c se calcula como:

$$c = \frac{1}{n} \sum_{i=1}^{n} a_i$$

Para este caso se está minimizando la función objetivo:

$$f(c) = \sum_{i=1}^{n} \|c - a_i\|_p$$

Es importante mencionar que el centroide minimiza la suma de las distancias al cuadrado a todos los puntos en el conjunto A. Esto se conoce como la distancia euclidiana cuadrada. Si se quisiera minimizar la suma de las distancias (no al cuadrado) a todos los puntos en A, el punto que haría esto sería la mediana geométrica, no el centroide.

A partir de resolver este problema se escaló al problema de tener un segundo conjunto de puntos B y encontrar el punto que minimizara la media de las distancias hacia los puntos de A y maximizara la media de las distancias a los puntos B.

El problema de optimización multiobjetivo que se plantea, que implica minimizar la media de las distancias a los puntos en el conjunto A y maximizar la media de las distancias a los puntos en el conjunto B, es una extensión natural del problema más simple de encontrar el centroide de un conjunto de puntos.

La solución de dos objetivos es valiosa por varias razones. Proporciona una forma simplificada de abordar el problema que puede ser más fácil de entender e implementar. Esto fue especialmente útil en las primeras etapas de la investigación, cuando todavía se están explorando diferentes enfoques y se está tratando de obtener una comprensión intuitiva del problema.

La solución de dos objetivos permite además una visualización más sencilla de los resultados. Visualizar el espacio objetivo para un problema de optimización con n objetivos requeriría trazar en $\mathbb{R}^n$, lo cual no posible para $n > 3$. Con solo dos objetivos, los resultados se pueden trazar fácilmente en un gráfico bidimensional, lo que facilita la interpretación y el análisis de los resultados.

La implementación de ambas variantes solo difiere en la inicialización de la clase LocationProblemWithRiver y en como se calculan los objetivos para cada variante en particular. Se muestran fragmentos de código de la implementación que reflejan estas diferencias.

```
    class LocationProblemWithRiver(Problem):
    def __init__(self, a, b, norm, river, bridges):
        super().__init__(
            n_obj=2, n_var=2, n_constr=3,
            xl = settings.XL,
            xu = settings.XU)
            .
            .
            .
    def _evaluate(self, x, out, *args, **kwargs):
        #some code
        f1[i] = np.mean(distances_a)
        f2[i] = -np.mean(distances_b)
```

```
    class LocationProblemWithRiver(Problem):
    def __init__(self, a, b, norm, river, bridges):
        super().__init__(
            n_obj=len(a) + len(b),n_var=2, n_constr=3,
            xl = settings.XL,
            xu = settings.XU)
            .
            .
            .
```

```
10      def _evaluate(self, x, out, *args, **kwargs):
11          #some code
12          f[i, j] = distance
13          f[i, len(self.a) + j] = -distance
```

4.2. Experimentos

En esta sección se muestran los resultados de los experimentos realizados para comprobar el rendimiento de los algoritmos correspondientes a cada uno de los modelos presentados en el capítulo anterior. Los mismos se corrieron en, ACER Aspire A515-54, con Procesador: Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz, 8.00 GB de RAM, en Python 3.11

De cada experimento se corrieron 30 simulaciones para comprobar que los resultados del mismo fueran similares en cada caso. También se cambió para cada corrida el tipo de cruzamiento con que se ejecutaba el algoritmo.

En cada experimento se grafican las soluciones obtenidas en el plano y para los ejemplos del modelo 2, además el frente de Pareto.

Los experimentos toman como entrada un conjunto de puntos A y B, de los que se quiere estar cerca y lejos respectivamente, el número de individuos de la población, el método de cruzamiento a emplear y el número de generaciones. La salida del algoritmo es una gráfica que muestra como queda el espacio de diseño luego de 100 generaciones.

En las imágenes que representan el espacio de diseño y el espacio objetivo, las cruces azules son los puntos del conjunto B, los puntos del conjunto A se representan con estrellas verdes, el río esta representado por un rectángulo azul y las líneas de color rojo en este rectángulo son una representación de los puentes.
Para los experimentos, se trabajó en una región delimitada por un cuadrado de 20x20 unidades en ambos ejes, tanto en el de las abscisas como en el de las ordenadas.

A continuación, en sendas secciones se muestran los experimentos realizados.

Experimento 1

En este experimento se prueban los resultados de colocar una nueva instalación en una región con solo 4 puntos a los que se quiere estar lo más cerca posible, dichos puntos en la misma rivera. El conjunto de puntos de entrada utilizado fue:
A = {[1. 10.],[10. 10.],[13. 14.],[2. 18.]}
B = {None}
Como se muestra en las gráficas 4.1 y 4.2, se obtuvieron 82 soluciones Pareto usando el modelo 1 y la norma 1, en el caso de la norma 2 se obtuvieron 89 soluciones Pareto.

Como se aprecia en al figura existe un área poblada por los puntos del conjunto solución la cuál está bien definida.

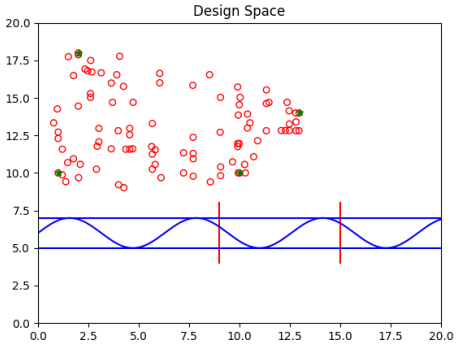

Figura 4.1: Experimento 1, modelo 1, norma 1

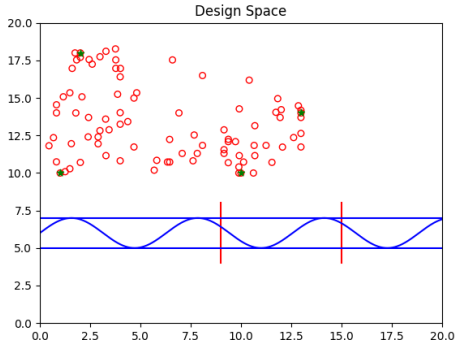

Figura 4.2: Experimento 1, modelo 1, norma 2

Experimento 2

En este experimento se prueban los resultados de colocar una nueva instalación en una región con varios puntos, donde no hay instalaciones de las que se quiere estar lo más alejado posible, se pretende en este experimento ver el desempeño del algoritmo cuando los puntos están en lados distintos del río y todos son puntos a los que se quiere estar cerca. El conjunto de puntos de entrada utilizado fue:
A = {[2. 3.],[7. 4.],[12. 1.],[10. 11.],[7. 8.]}
B = {None}
Como se muestra en las gráficas 4.3 y 4.4, se obtuvieron 87 soluciones Pareto usando el modelo 1 y la norma 1 y la norma 2.

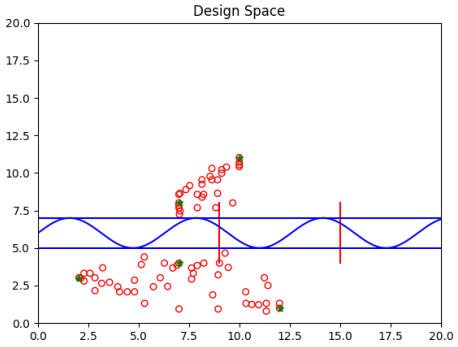

Figura 4.3: Experimento 2, modelo 1, norma 1

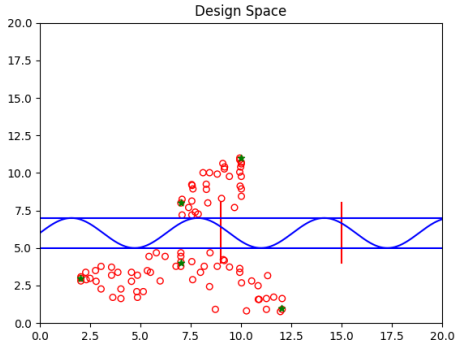

Figura 4.4: Experimento 2, modelo 1, norma 2

En este experimento se observa que el algoritmo halla soluciones en ambas rive-

ras. Esto se corresponde con la configuración de este ejemplo en que hay puntos del conjunto A a ambos lados del río. Si bien el conjunto de Pareto tiene dos componentes conexas, se hallan puntos en ambas, lo cual ilustra el buen comportamiento del algoritmo. No existe ninguno sobre el río por tanto no se violan las restricciones del problema y por último se ve como el algoritmo encuentra el acceso más corto en el puente 1, que está mas cerca a todos los puntos dados.

Experimento 3

En este experimento se prueban los resultados de colocar una nueva instalación en una región con 5 puntos, un punto al que se quiere estar lo más cerca posible, y otros 4 de los que se quiere estar lo más alejado. El conjunto de puntos de entrada utilizado fue:
A = {[10. 10.]}
B = {[1. 1.],[3. 3.],[9. 4.],[17. 2.]}
Como se muestra en las gráficas 4.5 y 4.6, se obtuvieron 89 soluciones Pareto usando el modelo 1 y norma 1, en el caso de la norma 2 se calculan 84 soluciones Pareto.

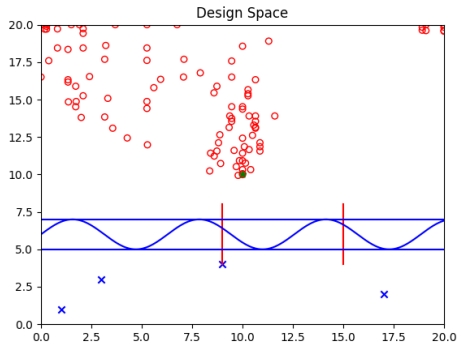

Figura 4.5: Experimento 3, modelo 1, norma 1

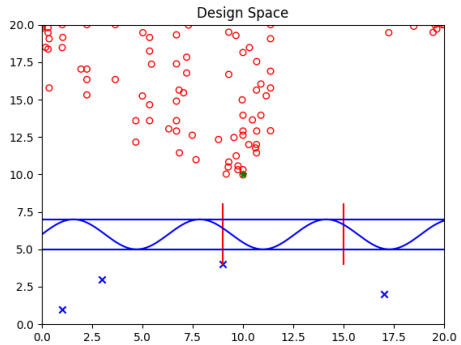

Figura 4.6: Experimento 3, modelo 1, norma 2

Al tener 2 funciones objetivos, se utiliza el modelo 2. Como se observa en la figura 4.8 se obtiene un frente de Pareto bien definido. Se obtuvieron 84 soluciones Pareto de aplicar el modelo 2 y la norma 1 e igual cantidad con la norma 2 y modelo 1, ver figura 4.10

Para ambos casos del modelo 2, se observa un frente de Pareto que muestra la relación negativa entre las dos variables. A medida que una variable aumenta, la otra disminuye, lo cual se indica por la tendencia descendente de los puntos azules en el gráfico. El frente de Pareto muestra un trade-off entre las dos variables. No puedes mejorar una variable sin empeorar la otra.

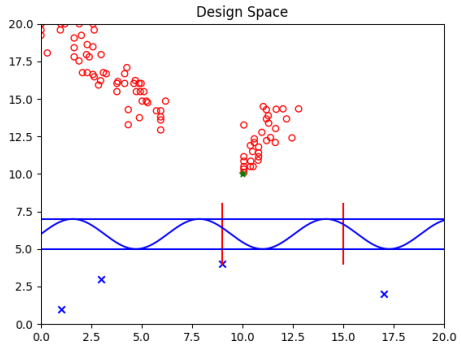

Figura 4.7: Experimento 3.1, modelo 2, norma 1

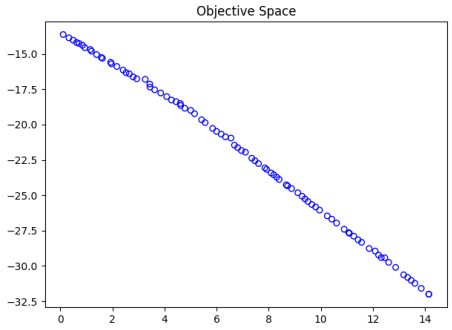

Figura 4.8: Experimento 3.1, Frente de Pareto

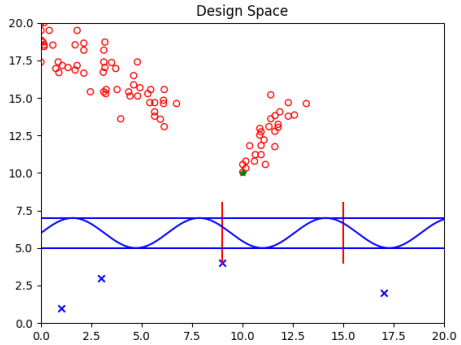

Figura 4.9: Experimento 3.2, modelo 2, norma 2

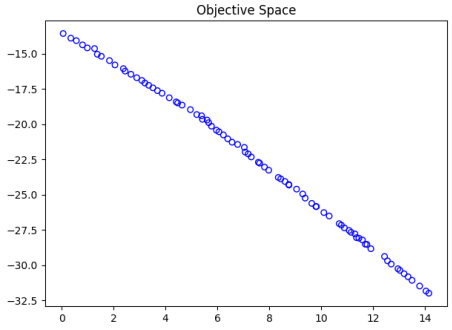

Figura 4.10: Experimento 3.2, Frente Pareto

En este experimento se verifica el buen desempeño del algoritmo con la presencia del río, se observa la región bien definidas. En este caso en la rivera superior se aprecia la existencia de un área de puntos donde todos son soluciones que se acercan a (10,10) y luego varían sus coordenadas para cumplir con los otros objetivos.

Experimento 4

En este experimento se prueban los resultados de colocar una nueva instalación en una región con varios puntos, tanto puntos a los que se quiere estar cerca como puntos de los que se quiere estar alejado, o sea $A, B \neq \emptyset$ con más de un punto cada uno. Además se considera que ambos conjuntos tienen elementos en ambos lados del río pero que la mayoría de los puntos de A se encuentran a la izquierda y lo de B a la derecha de la región. En este experimento en especial se agrandó el tamaño de la población, en 2 de los 3 casos, para lograr un conjunto solución más amplio y fuera más fácil observar el espacio de las soluciones. Se muestran los 4 casos que corresponden a los dos modelos y las dos normas.

Como se muestra en las gráficas 4.11 y 4.12, se obtuvieron 299 soluciones Pareto usando el modelo 1 y norma 1, en el caso de la norma 2 se calculan 296 soluciones Pareto. El conjunto de puntos de entrada utilizado fue:
A = {[2. 8.],[3. 14.],[4. 8.],[1. 1.],[2. 4.],[3. 4.]}
B = {[12. 12.],[17. 16.],[15. 9.],[10. 2.],[13. 5.],[19. 1.]}

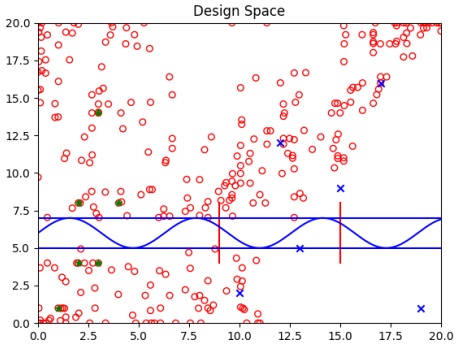

Figura 4.11: Experimento 4, modelo 1, norma 1

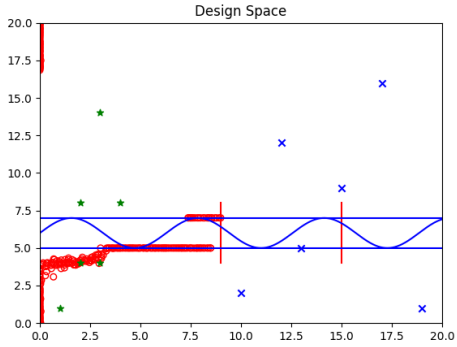

Figura 4.12: Experimento 4, modelo 1, norma 2

En este experimentos se comprueba el segundo modelo desarrollado, con las mismas entradas que en el experimento 4. Como se muestra en las gráficas 4.13 y 4.16, se obtuvieron 91 soluciones Pareto usando el modelo 2 y norma 1, en el caso de la norma 2 se calculan 296 soluciones Pareto, ver figuras 4.15 y 4.16

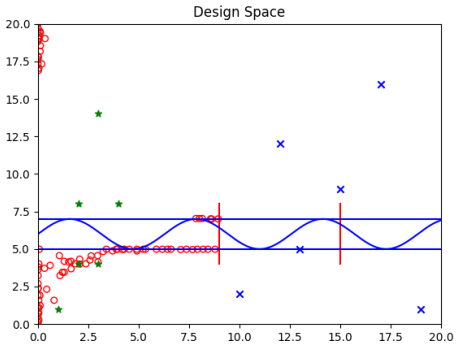

Figura 4.13: Experimento 4.1, modelo 2, norma 1

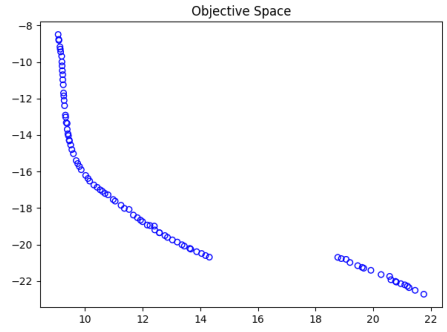

Figura 4.14: Experimento 4.1, Frente Pareto

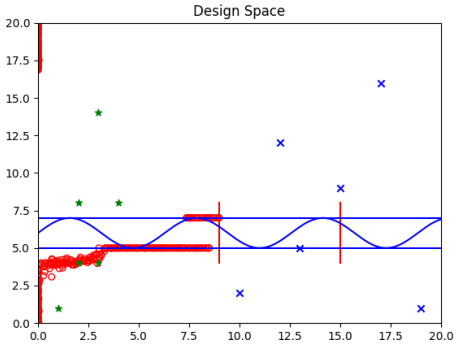

Figura 4.15: Experimento 4.2, modelo 2, norma 2

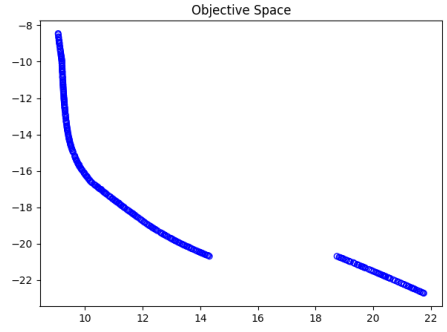

Figura 4.16: Experimento 4.2, Frente Pareto

Para este caso la región del conjunto solución es bastante amplia debido a la cantidad de puntos dados y la forma en que están repartidos sobre el mapa. El valor de la dispersión es mayor que la mayoría de los experimentos debido al área que ocupan los puntos del conjunto solución. Para casos como estos se trabaja con una población más grande como se menciona anteriormente, pues al existir tal dispersión se necesita que haya máss puntos Pareto en el conjunto solución para poder visualizar mejor donde se encuentra la región que determina las coordenadas del conjunto solución.

Los frente de Pareto de los experimentos del modelo 2, a diferencia del experimento 3, hay una brecha notable en los puntos de datos entre los valores x de aproximadamente 15 y 18. Esto podría indicar que hay un rango de valores para una de las variables que no produce soluciones óptimas.

Experimento 5

Para este experimento se deseaba ver el resultado del algoritmo bajo condiciones no muy cómodas, donde ambos conjuntos A y B de puntos existiera y estuvieran de una manera bastante aleatoria en el mapa y en ambos regiones respecto al río, o sea $A, B \neq \emptyset$. Se muestran los 4 casos que corresponden a los dos modelos y las dos normas.

Como se muestra en las gráficas 4.17 y 4.18, se obtuvieron 85 soluciones Pareto usando el modelo 1 y las normas 1 y 2. El conjunto de puntos de entrada utilizado fue:
A = {[2. 8.],[3. 14.],[4. 8.],[1. 1.],[2. 4.],[3. 4.],[19. 2.],[14. 14.],[13. 1.]}
B = {[12. 12.],[17. 16.],[15. 9.],[10. 2.],[13. 5.],[19. 1.],[4. 4.],[3. 16.],[2. 17.]}

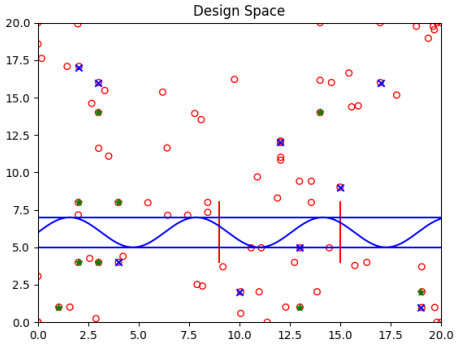

Figura 4.17: Experimento 5, modelo 1, norma 1

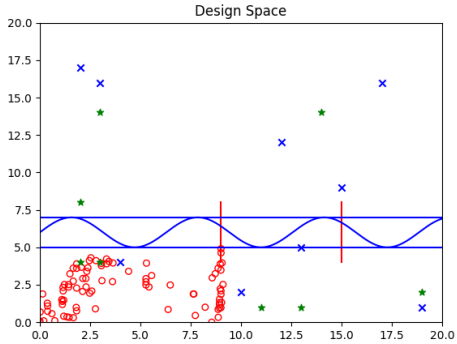

Figura 4.18: Experimento 5, modelo 1, norma 2

Se comprueba el segundo modelo desarrollado, con las mismas entradas que en el experimento 5. Como se muestra en las gráficas 4.19 y 4.20, se obtuvieron 88

soluciones Pareto usando el modelo 2 y norma 1, en el caso de la norma 2 se calculan 86 soluciones Pareto, ver figuras 4.21 y 4.22.

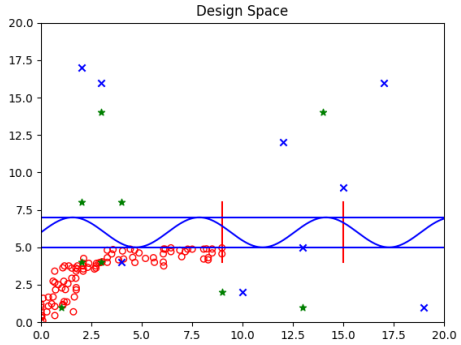

Figura 4.19: Experimento 5.1, modelo 2, norma 1

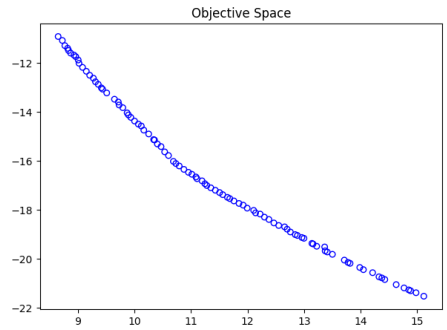

Figura 4.20: Experimento 5.1, Frente de Pareto

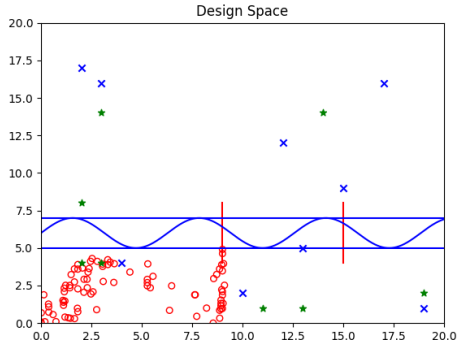

Figura 4.21: Experimento 5.2, modelo 2, norma 2

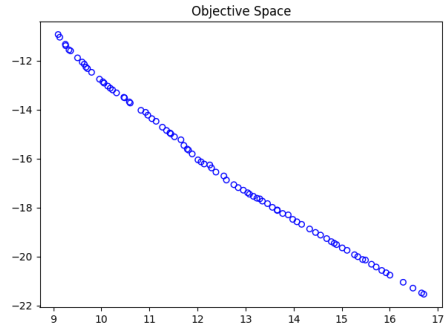

Figura 4.22: Experimento 5.2, Frente de Pareto

Este experimento reveló como con un conjunto de puntos un poco aleatorios, el algoritmo no tiene buen desempeño, pues nunca se ve una región definida del conjunto solución, los puntos están por cualquier lugar del mapa. Si bien son puntos no dominados entre sí, no queda claro que san soluciones del problema, de modo que como trabajo futuro se planea mejorar estos casos.

Como resultado de los experimentos se pudo observar un buen rendimiento del algoritmo en la mayoría de los casos de prueba. Para los casos menos aleatorios donde los conjuntos de puntos A y B están en clúster separados, el algoritmo encuentra sin dificultad el Frente Pareto, si los puntos están de manera más aleatoria en el mapa

se cree que con un algoritmo enfocado a algunos objetivos se pueden obtener mejores resultados. La combinación de este algoritmo con otro exacto es una idea a desarrollar.

Conclusiones y trabajo futuro

En este trabajo, se ha desarrollado un algoritmo para resolver el problema de localización, que consiste en determinar la ubicación óptima de un nuevo punto con respecto a dos conjuntos de puntos, A y B. El objetivo es que el nuevo punto esté lo más cerca posible de todos los puntos en A y lo más lejos posible de todos los puntos en B. Este problema se convierte en un reto por la presencia de un río que dificulta el cálculo de las distancias entre puntos que están en lados opuestos del río.

Para resolver este problema, se utilizó el algoritmo genético NSGA-II, que demostró ser efectivo en la mayoría de los escenarios. Sin embargo, este estudio se limitó a una representación simplificada del río y a coordenadas cartesianas.

Este trabajo representa un primer paso importante hacia la resolución del problema de localización en presencia de obstáculos geográficos como ríos. Pero quedan problemas abiertos.

En trabajos futuros se planea abordar estas limitaciones. En particular, se pretende trabajar con coordenadas longitudinales, lo que permitiría una representación más realista del problema. Además, se buscará representar el río de una manera menos simple, lo que podría mejorar la precisión del algoritmo.

Se desea también integrar otras consideraciones a este problema. Si bien la protección ambiental de la zona esta implícita al por ejemplo colocar una fábrica alejada de playas y ríos para evitar su contaminación se pueden tomar en cuenta otras restricciones. La sostenibilidad de la construcción de la nueva entidad también puede depender del lugar en que se coloca. Incorporar esta consideración sería también interesante en la aplicación.

Si bien se ha investigado la influencia de restricciones geográficas como la presencia de ríos, montañas o áreas inaccesibles, aún existen desafíos en la modelización y resolución de problemas con restricciones geográficas complejas.

El cálculo de la distancia es más complejo en zonas con montañas, cuevas, u otros accidentes geográficos que afectan la accesibilidad, De ahí que estos problemas son también de interés en el turismo de naturaleza.

Esta tesis brinda un primer enfoque para la solución de este problema. El estudio teórico de este problema permitirá hallar condiciones necesarias de optimalidad útiles para refinar este algoritmo.

Bibliografía

Alzorba, S. (2015). *Algorithms and Decomposition Methods for Multiobjective Location and Approximation Problems* [Tesis doctoral, Martin-Luther-Universität Halle-Wittenberg]. (Vid. pág. 17).

Alzorba, S., Günther, C., Popovici, N., & Tammer, C. (2017). A new algorithm for solving planar multiobjective location problems involving the Manhattan norm. *European Journal of Operational Research*, *258*(1), 35-46 (vid. pág. 17).

Archetti, C., Speranza, M., & Hertz, A. (2010). The Capacitated Location Routing Problem with Single Assignment. *Journal of Heuristics*, *16*(2), 235-257 (vid. pág. 19).

Atefeh Taghavi, R. G. (2021). A genetic algorithm for solving bus terminal location problem using data envelopment analysis with multi-objective programming approach. *SpringerLink Home Soft Computing Article Methodologies and Application*. https://link.springer.com/article/10.1007/s10479-021-04244-4 (vid. pág. 17).

Atta, S., Mahapatra, P. R. S., & Mukhopadhyay, A. (2018). Solving maximal covering location problem using genetic algorithm with local refinement. *Soft Computing*, *22*, 3891-3906. https://link.springer.com/article/10.1007/s00500-017-2598-3 (vid. págs. 13, 16, 19).

Bäck, T., & Schwefel, H.-P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, *1*(1), 1-23 (vid. pág. 11).

Baeldung. (2023). Real-World Uses for Genetic Algorithms. *Baeldung on Computer Science*. https://www.baeldung.com/cs/genetic-algorithms-applications (vid. págs. 10, 13, 19).

Berman, O., & Krass, D. (2007). A multi-objective approach for capacitated facility location problems with safety stocks. *European Journal of Operational Research*, *182*(2), 670-683 (vid. pág. 20).

Bertsekas, D. P. (1999). *Nonlinear Programming: Theory and Algorithms*. Athena Scientific. (Vid. pág. 1).

Blank, J., & Deb, K. [K.]. (2020). pymoo: Multi-Objective Optimization in Python. *IEEE Access*, *8*, 89497-89509 (vid. págs. 15, 27, 29).

Bonilla, M. V. (2019). *Modelos de localización*. %5E2%5E (vid. pág. 1).

Botía, W. (2005). Funcionamiento del algoritmo IBEA [Accedido el 2 de Enero de 2024]. %5E1%5E (vid. pág. 12).

Britannica. (2020). Vilfredo Pareto, Italian Economist, Sociologist and Philosopher. %5E2%5E (vid. pág. 5).

Cantlebary, L., & Li, L. (2020). Facility location problem. *Cornell University Computational Optimization Open Textbook - Optimization Wiki*. %5E1%5E (vid. págs. 1, 13).

Chugh, T., Sindhya, K., Hakanen, J., & Miettinen, K. (2011). A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Swarm and Evolutionary Computation*, *1*, 32-49 (vid. pág. 12).

Cruz, R. A. C., Bernal, J. M., & Castro, J. A. O. (2021). Modelos logísticos estocásticos aplicados a la cadena de suministro: una revisión de la literatura. *Ingeniería*, *26*(3). %5E2%5E (vid. pág. 1).

Daskin, M. S. (2010). *Network and Discrete Location: Models, Algorithms, and Applications*. Springer. (Vid. pág. 1).

Deb, K. [Kalyanmoy], Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*, 182-197. https://ieeexplore.ieee.org/document/996017 (vid. págs. 16, 17).

de Sardi, J. L. (2011). Optimización Multiobjetivo Evolutiva a través un modelo de programación [Accedido el 2 de Enero de 2024]. *Redalyc*. %5E4%5E (vid. pág. 12).

Drezner, Z., & Hamacher, H. W. [Horst W.]. (2002). *Facility Location: Applications and Theory*. Springer Berlin, Heidelberg. (Vid. pág. 1).

Drezner, Z., & Hamacher, H. W. [Horst W.]. (2004). *Facility location: applications and theory*. Springer Science & Business Media. (Vid. pág. 15).

Fang, X., Wang, W., He, L., Huang, Z., Liu, Y., & Zhang, L. (2018). Research on Improved NSGA-II Algorithm and Its Application in Emergency Management. *Mathematical Problems in Engineering*, *2018*. https://doi.org/10.1155/2018/1306341 (vid. pág. 14).

Fliege, J., & Svaiter, B. (2000). Steepest descent methods for multicriteria optimization. *Mathematical Methods of OR*. %5E3%5E (vid. pág. 9).

Garrido, L. F. (2015). Problemas de localización de instalaciones [Accedido el 22 de Diciembre de 2023]. %5E5%5E (vid. pág. 1).

Garrido, L. F. (2019). Problemas DE Ubicación Y Localización DE Plantas. *Revista de Ingeniería y Arquitectura*, *20*, 45-55 (vid. pág. 1).

Genérica, P. W. (2024). Funciones Convexas y Cóncavas [Accedido: 2024-01-04]. P%C3%A1gina%20Web%20Gen%C3%A9rica (vid. pág. 27).

Graña Drummond, L., Maculan, N., & Svaiter, B. (2008). On the choice of parameters for the weighting method in vector optimization. *Math. Program.*, *111*, 201-216 (vid. pág. 9).

Guntherm, C. (2017). Zonas prohibidas en la bahía de La Habana. *Investigación Operacional*. %5E1%5E (vid. pág. 20).

Hamacher, H. W. [H. W.], & Nickel, S. (1998). Classification of Location Models. *Location Science*, *6*, 229-242 (vid. pág. 15).

Harder, D. W. (2021). El algoritmo de Hooke y Jeeves [Accedido el 2 de Enero de 2024]. %5E1%5E (vid. pág. 9).

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press. (Vid. pág. 11).

Lab, T. B. B. (2020). Qué son los algoritmos evolutivos y para qué se usan [Accedido el 2 de Enero de 2024]. %5E3%5E (vid. pág. 12).

Luenberger, D. G., & Ye, Y. (2021). *Linear and Nonlinear Programming*. Springer. %5E1%5E (vid. pág. 9).

Ma, H., Zhang, Y., Sun, S., Liu, T., & Shan, Y. (2023). A comprehensive survey on NSGA-II for multi-objective optimization and applications. *Artificial Intelligence Review*, *56*, 15217-15270. https://link.springer.com/article/10.1007/s10462-023-10526-z (vid. pág. 28).

Miettinen, K. (2012). *Nonlinear Multiobjective Optimization*. Springer. (Vid. pág. 8).

Monika Mangla, R. A. (2015). Implementing Genetic Algorithm to solve Facility Location Problem. *IRJET*. https://www.irjet.net/archives/V2/i5/IRJET-V2I5134.pdf (vid. pág. 17).

Murray, A., Tong, D., & Kim, T. (2010). A multi-zone Weber model for location of undesirable facilities. *Environment and Planning B: Planning and Design*, *37*(1), 119-133 (vid. pág. 19).

Olvera Pimentel, C. V. (2019). *Optimización multiobjetivo mediante un algoritmo evolutivo cuántico* [Tesis de maestría, Instituto Politécnico Nacional, Centro de Investigación y Desarrollo de Tecnología Digital, México]. https://tesis.ipn.mx/jspui/handle/123456789/27356 (vid. pág. 7).

Overflow, S. (2024). Fitness function with multiple weights in DEAP [Accessed: 2024-01-04]. Stack%20Overflow (vid. pág. 8).

Powell, M. J. D. (1960). El método de las direcciones conjugadas [Accedido el 2 de Enero de 2024]. %5E1%5E (vid. pág. 9).

pymoo. (2021). *NSGA evolution*. https://pymoo.org/algorithms/moo/nsga2.html# (vid. pág. 13).

Rosenbrock, H. H. (1960). Rosenbrock methods - Wikipedia [Accedido el 2 de Enero de 2024]. https://en.wikipedia.org/wiki/Rosenbrock_methods (vid. pág. 9).

Salazar, A. F., López, J. F., Tavizón, A., & Araiza-Vázquez, M. J. (2019). Estudio de un Algoritmo Genético para la Administración Académica. *Formación universitaria*. %5E2%5E (vid. pág. 12).

Sánchez, G. C., Abril, I. P., & Sánchez, Z. G. (2022a). Exploración científica de los algoritmos evolutivos en la reconfiguración óptima de redes de distribución eléctrica. *Revista Universidad y Sociedad*. %5E1%5E (vid. pág. 12).

Sánchez, G. C., Abril, I. P., & Sánchez, Z. G. (2022b). Exploración científica de los algoritmos evolutivos en la reconfiguración óptima de redes de distribución eléctrica [Accedido el 2 de Enero de 2024]. *Universidad y Sociedad*. %5E1%5E (vid. pág. 12).

Schwefel, H.-P. (1981). *Numerical optimization of computer models*. Wiley. (Vid. pág. 11).

Shi, F., & Lin, J. (2022). Virtual Machine Resource Allocation Optimization in Cloud Computing Based on Multiobjective Genetic Algorithm. *Computational Intelligence and Neuroscience*. %5E1%5E (vid. pág. 19).

Soleimani, M. (2020). NSGA-II algorithm for hub location-allocation problem considering hub... *World Journal of Engineering*. https://www.sciencegate.app/document/10.1108/wje-12-2020-0658 (vid. pág. 14).

Special Issue 'Latest Advances and Applications of Multi-Objective Optimization Techniques'. (2023). *Applied Sciences* (vid. pág. 17).

Teo, C., & Shumshy, F. (2008). Uncapacitated Facility Location Problem. *European Journal of Operational Research*, *190*(2), 231-245 (vid. pág. 19).

Verma, S., Pant, M., & Snasel, V. (2021). A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems. *IEEE Access*, *9*, 57757-57791. https://ieeexplore.ieee.org/document/9393947 (vid. pág. 28).

Villamar, C. J., & Saltos, I. V. (2014). Aplicación de SPEA2 al cálculo de esquemas de dosificación para el tratamiento quimioterapéutico del cáncer [Accedido el 2 de Enero de 2024]. *Maskana*. %5E1%5E (vid. pág. 12).

Weber, A. (1909). *Über den Standort der Industrien*. Tübingen, J.C.B. Mohr. (Vid. pág. 18).

Yang, X., & Deb, S. (2008). Solving Fermat-Weber problems efficiently with differential evolution. *Computers and Operations Research*, *35*(9), 2780-2795 (vid. pág. 20).

Yin, X. Z. Y. (2017). Research on the application of genetic algorithm in logistics location problem. *IEEE Xplore Digital Library*. https://ieeexplore.ieee.org/document/8091454/ (vid. págs. 13, 17, 19).

Zhang & Li. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition [Accedido el 2 de Enero de 2024]. %5E1%5E (vid. pág. 12).

Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, *11*(6), 712-731 (vid. pág. 27).

Zheng, W., & Doerr, B. (2023). Mathematical Runtime Analysis for the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). *arXiv preprint arXiv:2112.08581v5*. https://arxiv.org/abs/2112.08581 (vid. pág. 13).