# PROBLEMAS DE LOCALIZACIÓN MULTIOBJETIVO

JUAN GUILLERMO VILLEGAS RAMÍREZ

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL
MAESTRÍA EN INGENIERÍA INDUSTRIAL
BOGOTÁ D.C.

2003

# PROBLEMAS DE LOCALIZACIÓN MULTIOBJETIVO

JUAN GUILLERMO VILLEGAS RAMÍREZ

**Área de Investigación: Sistemas de Producción**

**Asesor: P.h.D. Fernando Palacios Gómez**

**Co-Asesor: P.h.D. Andrés Medaglia González**

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL
MAESTRÍA EN INGENIERÍA INDUSTRIAL
BOGOTÁ D.C.

2003

# Agradecimientos

Al culminar este trabajo deseo expresar mis agradecimientos a:

A mis padres y hermano por su apoyo en esta etapa tan importante de mi vida.

A Anna por seguir allí a pesar de la distancia que impuso la Maestría.

Al Doctor Fernando Palacios Gómez, asesor de esta tesis, por su invaluable aporte en el desarrollo de la misma, por todo el conocimiento que de él recibí en cada uno de los cursos...

Al Doctor Andrés Medaglia G., quien como coasesor de la tesis me abrió puertas y mostró nuevos caminos...

Al Programa Universidad Empresa de la Universidad de Los Andes por la oportunidad incomparable de participar de él.

A Almacafé S.A. por financiar con sus aportes mis estudios de Maestría y hacerme parte de la familia cafetera. En especial al Doctor Gonzalo Rivera R. y al Doctor Marcos Castillo.

A Ludym, Dilia y Hector por la compañía, la amistad y por compartir la nostalgia de la tierra...

A Lizzy, Mauricio y Gineth por su hospitalidad y por cambiar en mi el paradigma acerca de los bogotanos.

A Angélica por lo mucho que aprendí trabajando a su lado y por brindarme su amistad y hospitalidad.

Al profesor Mario César Vélez G. de la Universidad Eafit por mostrarme el camino.

A María Cañón y Liliana Rubiano, de la sala virtual de la biblioteca, por su valiosa colaboración en el desarrollo de la tesis al conseguir hasta el más difícil de los artículos de la bibliografía.

# Contenido

# Lista de Tablas

# Lista de Figuras

# Lista de Algoritmos

# INTRODUCCIÓN

La localización de las instalaciones donde se prestará el servicio a los clientes, o donde se llevarán a cabo las operaciones, es una de las decisiones estratégicas más importantes que tienen que abordar muchas organizaciones públicas y privadas. Este tipo de decisiones se refieren a la ubicación de una gran diversidad de instalaciones, por ejemplo: escuelas, bancos, bibliotecas, iglesias, estaciones de bomberos, unidades médicas de emergencia, rellenos sanitarios, plantas nucleares, plataformas de explotación petrolera, bodegas, plantas de producción, y muchas otras que se encuentran reportadas en la literatura.

Tradicionalmente, las decisiones de localización se han modelado como situaciones en las que hay un único criterio de decisión. Entre los criterios considerados han primado el costo y el servicio (medido en términos de cobertura o distancia a los clientes), como medida de la calidad de las decisiones. Pero, pocas veces, el costo y la cobertura son considerados simultáneamente, ignorando que estos dos criterios se encuentran en conflicto, y que en general, ofrecer un mejor servicio exige mayores costos de operación.

Como alternativa, en esta investigación se exploran algunos modelos que consideran simultáneamente el costo y la cobertura para abordar las decisiones de localización, integrándolos en un modelo de decisión multiobjetivo. Se desarrollan diferentes metodologías para la obtención de soluciones eficientes, basadas en los algoritmos evolutivos y la programación matemática. Finalmente, se ilustra su aplicación en una situación real

## Objetivos

### Objetivo General

Abordar los problemas de localización multiobjetivo, en los que se consideran como criterios de decisión el costo y la cobertura, y desarrollar técnicas para encontrar soluciones eficientes a dichos problemas.

### Objetivos Específicos

- Recopilar información sobre la aplicación de los algoritmos evolutivos en la solución de problemas de optimización multiobjetivo y en la solución de problemas de localización.
- Implementar algoritmos evolutivos para encontrar soluciones eficientes a los problemas de localización multiobjetivo, utilizar diferentes estrategias para su implementación y

compararlas

- Implementar otras técnicas de solución basadas en programación matemática y compararlas con los algoritmos evolutivos.
- Modelar bajo este enfoque el problema asociado con la configuración de la red de abastecimiento de café colombiano.

Este trabajo se encuentra organizado de la siguiente forma: En el primer capítulo se exponen los conceptos básicos para el desarrollo del trabajo. El capítulo dos está dedicado a la aplicación de dos algoritmos evolutivos para la solución de problemas de localización sin restricciones de capacidad. En el capítulo tres se desarrolla una metodología basada en programación matemática para la solución de problemas de localización multiobjetivo sin restricciones de capacidad. El capítulo cuatro extiende la metodología de programación matemática a situaciones en las que se consideran restricciones de capacidad. En el capítulo cinco se describe una aplicación real de los conceptos desarrollados en la investigación. Por último, el capitulo seis presenta las conclusiones y recomendaciones para futuras investigaciones sobre el tema.

# Capítulo 1

# GENERALIDADES

## 1.1. Problemas de Localización

Las decisiones de ubicación de iglesias en una ciudad [13], de instalaciones de manejo de residuos sólidos [3] o de plantas de embotellado de gas propano [81], para citar unos pocos ejemplos, tienen un elemento en común: todas se enmarcan en lo que se conoce como problemas de localización.

Este tipo de problemas se refiere a la decisión de escoger de un conjunto de lugares, algunos de ellos, para situar varias instalaciones que prestarán su servicio a un conjunto de clientes, con el objetivo de optimizar uno o más criterios, económicos o de otra índole.

La importancia de los problemas de localización se debe en gran medida a:

1. La naturaleza estratégica de las decisiones, lo que las convierte en restricciones para la toma de otras decisiones de nivel táctico y operativo.

2. Su influencia sobre la competitividad de las empresas, ya que son un factor determinante de la calidad y el costo de los servicios prestados.

3. La apertura de nuevas instalaciones, generalmente, exige invertir grandes sumas de dinero con efectos económicos de largo plazo.

4. La multiplicidad de áreas en las que aparecen, algunas de ellas son: planeación urbana ( localización de terminales de buses [64] y estaciones de bomberos [5]), diseño de sistemas de distribución [72], apertura de restaurantes de comidas rápidas [69], localización de instalaciones de manejo de carbón [73], selección de herramientas para sistemas de manufactura flexible [29], ruteo de vehículos [16], entre otras.

Los problemas de localización han recibido bastante atención investigativa, principalmente por su naturaleza práctica y por la dificultad existente para encontrar soluciones óptimas a la mayoría de ellos. El trabajo en esta área se ha dedicado tanto al desarrollo de nuevos modelos [18, 31, 76], como al diseño y aplicación de diferentes técnicas de solución: relajación lagrangiana [11], algoritmos genéticos [4], búsqueda tabú [38], GRASP (*Greedy Randomized Adaptative Search Procedures*)[38], *simulated annealing* [38], *branch and price* [39], *branch*

*and bound* [50], entre otras. Buena parte de este trabajo se encuentra resumido en artículos [18, 31, 77] y libros sobre el tema [26, 30, 70].

## 1.2. Problemas de Localización Multiobjetivo

Minimizar el costo ha sido una de las funciones objetivo más utilizadas al modelar problemas de localización. Sin embargo, según el contexto en el cual se esté resolviendo el problema, existen otras funciones objetivo igual o más importantes. Por ejemplo: minimizar la población afectada (localización de rellenos sanitarios y otras instalaciones de gran impacto ambiental) o maximizar la equidad en el caso de entidades públicas [65].

La existencia de otros objetivos adicionales e influyentes ha hecho necesario modelar algunas decisiones de localización como problemas multiobjetivo. Algunos ejemplos son: localización de estaciones de bomberos [5], terminales de transporte [64], instalaciones de manejo de carbón [73], restaurantes de comida rápida [69] y diseño de sistemas de distribución [47, 72].

Generalmente los objetivos considerados están en conflicto, por ejemplo, en el diseño de redes de distribución se pueden considerar simultáneamente el costo y el servicio como funciones objetivo; en este caso las configuraciones más baratas de la red no ofrecerán el mejor servicio, e igualmente, las configuraciones que ofrezcan un mejor servicio serán bastante costosas. De ahí que en la optimización multiobjetivo no se busque un solo valor óptimo, sino un conjunto de soluciones eficientes de Pareto (o no dominadas) en las cuales no es posible mejorar el valor de alguna de las funciones objetivo sin deteriorar el desempeño de las otras.

Sea (PLM) el problema de localización multiobjetivo escrito como un programa de optimización multiobjetivo:

$$(PLM) \quad \text{mín}\{f_1(\mathbf{x}), ..., f_k(\mathbf{x})\} \tag{1.1}$$

Sujeto a,

$$\mathbf{x} \in \Omega \tag{1.2}$$

Donde $\mathbf{x}$ es un vector de variables de decisión que describe la configuración del sistema, $f_k(\mathbf{x})$ es la función objetivo que evalúa el k-ésimo criterio de decisión (k=1,...,K); y $\Omega$ el conjunto de soluciones (configuraciones) factibles. A continuación se presentan algunas definiciones importantes:

**Definición 1 (Eficiencia de Pareto)** *Sean $\mathbf{x}, \mathbf{y} \in \Omega$. Un par de soluciones factibles de PLM. Se dice que* $\mathbf{y}$ *domina a* $\mathbf{x}$ *(i.e.,$\mathbf{y} \preceq \mathbf{x}$) si $f_k(\mathbf{y}) \leq f_k(\mathbf{x})$ para todo k, y existe algún k tal que la desigualdad se cumple en forma estricta. Una solución factible que no es dominada por ninguna otra solución es llamada eficiente de Pareto o no dominada.*

**Definición 2 (Conjunto óptimo de Pareto, $P^*$)** *El conjunto óptimo de Pareto ($\mathcal{P}^*$) está compuesto por las soluciones factibles que no están dominadas por ninguna otra solución, $P^*$ se define así:*

$$\mathcal{P}^* = \{\mathbf{x} \in \Omega : \textit{no existe} \mathbf{x}' \in \Omega, \mathbf{x}' \preceq \mathbf{x}\}$$

**Definición 3 (Frente de Pareto, $\mathcal{PF}^*$)** *El frente de Pareto es la imagen del conjunto eficiente de Pareto en el espacio de las funciones objetivo:*

$$\mathcal{PF}^* = \{\mathbf{u} = f(\mathbf{x}) = [f_1(\mathbf{x}), ..., f_k(\mathbf{x})] : \mathbf{x} \in \mathcal{P}^*\}$$

Numerosas técnicas se han utilizado para la solución de problemas multiobjetivo: programación dinámica [43], *branch and bound* [66], algoritmos metaheurísticos [53], programación por metas [73], entre otras. En [41] se presenta una amplia reseña de problemas de optimización combinatoria modelados con múltiples objetivos y las técnicas de solución empleadas en cada caso. En el caso particular de los problemas de localización multiobjetivo se han utilizado principalmente: programación por metas [64, 73], programación dinámica [43] y procedimientos especiales [41, 79].

También se han utilizado técnicas de solución que abordan el problema como si fuese de un solo objetivo, pero que para explorar la naturaleza multiobjetivo agregan las funciones objetivo en una sola [52, 72, 75] o que optimizan para una de las funciones objetivo e incluyen las demás como restricciones del problema [71, 75].

En [27] se presenta una reseña de las funciones objetivo utilizadas y las características de los problemas modelados en el análisis multiobjetivo de problemas de localización.

## 1.3. Red de Acopio y Almacenamiento del Café Colombiano

En este trabajo se presentará el caso de la red de acopio y almacenamiento de café colombiano como ejemplo concreto de problemas de localización que pueden modelarse como problemas multiobjetivo .

El proceso de abastecimiento del café colombiano de exportación comienza en las agencias de las cooperativas de caficultores, que se encargan de comprar el café pergamino a los productores del grano en las diferentes zonas cafeteras, distribuidas a lo largo del territorio nacional, desde Nariño hasta la Sierra Nevada de Santa Marta. Una vez adquirido es llevado a las bodegas de Almacafé donde se conserva hasta que se despacha a la trilladora para su transformación y posterior envío al puerto de exportación.

Almacafé es la empresa del gremio cafetero especializada en la operación logística interna del café colombiano. Para cumplir su misión, Almacafé dispone de una amplia red de bodegas utilizada con dos propósitos básicos: adquirir el café colombiano de exportación y garantizar la compra de la cosecha a los caficultores. Al analizar la red de abastecimiento del café colombiano es necesario considerar estos dos fines. El primero de ellos exige que la red sea operada a los menores costos posibles, mientras que el segundo hace necesario que las bodegas estén cerca de las más de 450 agencias de compra. En el pasado los dos objetivos no presentaban un conflicto serio ya que el precio internacional del café permitía tener bodegas por todo el territorio nacional, de alguna manera la operación comercial financiaba la labor social.

Es bien sabido que toda la cadena productiva ligada a la caficultura está pasando por uno de sus más duros momentos. El desmonte del pacto internacional del café ha conducido a precios internacionales muy bajos antes inimaginables. La situación actual exige que la exportación sea un proceso eficiente que permita mantener la rentabilidad de la industria, aún con precios bajos.

Reducir el número de bodegas para el acopio y almacenamiento del café puede ser una alternativa para reducir los costos de operación [1]. Sin embargo, dentro del gremio cafetero, la garantía de compra es un servicio muy apreciado, lo que dificulta la aceptación de propuestas como ésta. El modelaje de la decisión de ubicación de las bodegas de Almacafé como un problema de localización multiobjetivo permitirá presentar diferentes configuraciones de la red, obtenidas considerando simultáneamente el costo y la cobertura. Estas alternativas se podrán comparar con la configuración actual, y además permitirán conocer el *trade off* existente entre los costos y la cobertura.

## 1.4. Modelo de Localización Multiobjetivo

Para modelar el problema de localización multiobjetivo resultante, se utilizan los elementos provenientes del problema de localización de máxima cobertura (MCLP,del inglés Maximum Covering Location Problem) [30] y del problema de localización sin restricciones de capacidad (UFLP, del inglés Uncapacitated Facility Location Problem)[7, 30]

El MCLP [2] considera el servicio ofrecido a los clientes. Su objetivo es localizar un número predeterminado de instalaciones de tal manera que se maximice la demanda atendida dentro del radio de cobertura máxima, previamente fijado. Toda la demanda es asignada a alguna instalación de servicio que haya sido abierta, pero sólo contribuyen en la función objetivo las demandas de los clientes atendidos por instalaciones que se encuentran a una distancia menor o igual al radio de cobertura.

El UFLP [3] busca minimizar el costo total de operación de la red logística (costo fijo y costo variable), con la única restricción de asignar todos los clientes a instalaciones abiertas. A diferencia del MCLP aquí no se considera de antemano el número de instalaciones a abrir, sino que se obtiene como parte de la solución.

Se utilizará una formulación similar a la propuesta en [76] que reúne apropiadamente los conceptos del MCLP y del UFLP.

Sean $\mathcal{I} = \{1, ..., m\}$ el conjunto de lugares donde se pueden localizar las bodegas, $f_i$ el costo fijo de operar una bodega en el lugar $i$, $\mathcal{J} = \{1, ..., n\}$ el conjunto de clientes, $d_j$ la demanda del cliente $j$, $c_{ij}$ el costo de atender toda la demanda del cliente $j$ desde la bodega $i$, $h_{ij}$ la distancia entre la bodega $i$ y el cliente $j$.

Los componentes de cobertura se incorporan en el modelo por intermedio de la distancia maxima de cobertura, $D_{max}$, entendida como la distancia a la cual se considera que un cliente es atendido con un buen servicio. Sea $\mathcal{Q}_j$ el conjunto de bodegas que pueden atender la demanda del cliente $j$ cumpliendo con la distancia máxima de cobertura,$\mathcal{Q}_j = \{i \in \mathcal{I} : h_{ij} \leq D_{max}\}$.

[1] Según la Comisión de Ajuste a la Institucionalidad Cafetera [32]:"La Comisión considera que sin perjuicio de que se conserve la política de comprador de última instancias, es necesario que se evalúe la conveniencia de mantener la estructura logística y de almacenamiento actual, buscando aumentar el nivel de eficiencia. Se debe conservar aquellas instalaciones que minimicen los costos de operación, el resto de la infraestructura debe venderse o entregarse a los municipios o entidades locales para que las utilicen con el compromiso de hacerse cargo de su mantenimiento ".

[2] Ver el Apéndice A para la formulación de este problema

[3] Ver el Apéndice B para la formulación de este problema

Las variables de decision son:

$$y_i = \begin{cases} 1, & \text{si se abre la bodega } i, \\ 0, & \text{de lo contrario.} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{si el cliente } j \text{ es atendido por la bodega } i, \\ 0, & \text{de lo contrario.} \end{cases}$$

La formulación del problema de localización multiobjetivo (PLM) es la siguiente:

$$\text{mín} \quad z_1 = \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} c_{ij}x_{ij} + \sum_{i\in\mathcal{I}} f_i y_i \tag{1.3}$$

$$\text{máx} \quad z_2 = \sum_{j\in\mathcal{J}} d_j \sum_{i\in\mathcal{Q}_j} x_{ij} \tag{1.4}$$

Sujeto a:

$$\sum_{i\in\mathcal{I}} x_{ij} = 1 \quad , \quad j \in \mathcal{J} \tag{1.5}$$

$$x_{ij} \leq y_i \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{1.6}$$

$$x_{ij} \in \{0,1\}, \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{1.7}$$

$$y_i \in \{0,1\}, \quad i \in \mathcal{I} \tag{1.8}$$

La función objetivo (1.3) representa el costo total de operación, el primer término corresponde al costo de satisfacer la demanda de los clientes con las bodegas abiertas y el segundo término al costo fijo de operar dichas bodegas. La función objetivo (1.4) representa la cobertura, medida como la suma de la demanda atendida por bodegas que se encuentran dentro del radio máximo de cobertura. Las restricciones (1.5) junto con la definición de las variables $x_{ij}$ garantizan que cada cliente es atendido por una sola bodega. Las restricciones (1.6) hacen que los clientes sean asignados a bodegas abiertas. Finalmente, en (1.7) y (1.8) se definen las variables de decisión del problema .

# Capítulo 2

# ALGORITMOS EVOLUTIVOS PARA LA SOLUCIÓN DE PROBLEMAS DE LOCALIZACIÓN MULTIOBJETIVO

Los métodos metaheurísticos multiobjetivo, es decir, versiones multiobjetivo de los algoritmos evolutivos, búsqueda tabú, o *simulated annealing*, entre otros; son una de las áreas de investigación más activas en el campo de la optimización multiobjetivo. Entre estos métodos, son precisamente los algoritmos evolutivos la alternativa más utilizada para aproximar la frontera de Pareto de muchos problemas de optimización multiobjetivo en diferentes campos de la ciencia, la ingeniería y la investigación de operaciones [19, 53]. La programación de tareas en una máquina [58] y la solución del problema del árbol cobertor mínimo en su version multiobjetivo [82] son algunos ejemplos de su aplicación. Igualmente, en los últimos años se han reportado implementaciones satisfactorias de los algoritmos evolutivos en la solución de diferentes problemas de localización, entre los que se encuentran el UFLP [4, 59, 60, 61] y el MCLP [4, 51].

## 2.1. Fundamentos de los Algoritmos Evolutivos

Un algoritmo evolutivo (AE) es un proceso de búsqueda aleatoria "inteligente"inspirado en el proceso de evolución de las especies dentro de la naturaleza, en la cual sobreviven y evolucionan los individuos mejor adaptados. Dichos individuos tienen mayores oportunidades de sobrevivir y reproducirse, mientras que los peor adaptados son eliminados. Los mecanismos de reproducción favorecen la combinación de las características de los padres mejor adaptados para producir hijos posiblemente mejores, multiplicando la presencia de dichas características en las siguientes generaciones.

Para simular el proceso de evolución natural, en cada iteración un AE mantiene una población

de cromosomas, que representan diferentes soluciones factibles (y en algunos casos también infactibles) de algún problema particular. La adaptación de las soluciones se mide con una o más funciones objetivo. A las soluciones mejor adaptadas se les permite reproducirse, intercambiando partes de su información genética, en un proceso de cruce para producir nuevas soluciones que comparten algunas características con sus padres.

Para diversificar la búsqueda, después del cruce algunas de las soluciones son mutadas alterando los genes de su representación con una probabilidad muy baja. La población de hijos puede reemplazar los cromosomas peor adaptados de la generación anterior o, si es del caso, reemplazarla totalmente. El ciclo de adaptación, selección y reproducción se repite hasta que se alcanza algún criterio de parada.

La estructura básica de un algoritmo evolutivo se describe así:

**Algoritmo 1** ALGORITMO EVOLUTIVO SIMPLE

1: $t \leftarrow 1$
2: Inicializar $\mathcal{P}(t)$
3: Evaluar $\mathcal{P}(t)$
4: **while** $t \leq T$ **do**
5: Cruzar $\mathcal{P}(t)$ para generar $\mathcal{C}(t)$
6: Mutar $\mathcal{C}(t)$
7: Evaluar $\mathcal{C}(t)$
8: Seleccionar $\mathcal{P}(t+1)$ de $\mathcal{P}(t) \cup \mathcal{C}(t)$
9: $t \leftarrow t + 1$
10: **end while**

Donde, $t$ es el índice de la generación, T el número máximo de generaciones, $\mathcal{P}(t)$ es la población de soluciones en la generación $t$, y $\mathcal{C}(t)$ la población de hijos en la generación $t$.

Ésta es una breve explicación de los algoritmos evolutivos, información introductoria adicional puede encontrase en [8, 9, 12] y con mayor profundidad en [45, 68].

## 2.2. Implementación de los Algoritmos Evolutivos para Problemas de Localización Multiobjetivo

Existen múltiples formas de implementar un Algoritmo Evolutivo, para programarlo es necesario definir la representación de las soluciones, la forma de obtención de la población inicial, los mecanismos utilizados para la selección, cruce y mutación y algunos otros elementos que regulan su funcionamiento

### 2.2.1. Codificación de las Soluciones

El primer paso en la implementación de un algoritmo evolutivo es el desarrollo de un esquema de representación para las soluciones del problema. En este trabajo se han utilizado dos representaciones.

La primera de ellas ha sido utilizada previamente para la solución del UFLP en [4, 59, 60]

y del MCLP en [4]. Es una cadena binaria de longitud $m$, el valor de la $i$-ésima posición indica si la bodega correspondiente está o no abierta. La representación binaria se ilustra en la Figura 2.1.

Como sólo se ha considerado explícitamente la apertura de las bodegas (i.e.,variables $y_i$), para completar una solución se asignan los clientes a las bodegas (variables $x_{ij}$) utilizando un procedimiento que busca minimizar el costo, sin deteriorar la cobertura que se puede obtener con las bodegas abiertas.

Sean $\mathcal{O}^l = \{i : y_i = 1\}$ el conjunto de bodegas abiertas en el $l$-ésimo cromosoma de la población, $\mathcal{Q}_j^l = \{i : y_i = 1, h_{ij} \leq D_{max}\}$ el subconjunto de bodegas abiertas que pueden cubrir al cliente $j$ dentro del radio máximo de cobertura.

Primero, cada uno de los clientes que pueden ser cubiertos, se se asigna a la bodega que lo puede atender al menor costo, así:

$$\text{Si} \quad \mathcal{Q}_j^l \neq \emptyset, \quad \text{entonces } x_{ij} = 1, \text{ para } i = \underset{i \in \mathcal{Q}_j^l}{\operatorname{argmin}} c_{ij} \tag{2.1}$$

Luego, cada uno de los clientes que no pueden ser cubiertos se asigna a la bodega que lo puede atender al menor costo.

$$\text{Si} \quad \mathcal{Q}_j^l = \emptyset, \quad \text{entonces } x_{ij} = 1, \text{ para } i = \underset{i \in \mathcal{O}^l}{\operatorname{argmin}} c_{ij} \tag{2.2}$$

Todas las soluciones generadas con el procedimiento de asignación son factibles, ya que la demanda de cada cliente es satisfecha por una sola de las bodegas abiertas y éstas no tienen límites en su capacidad.

Se utilizó una segunda representación centrada en la asignación de los clientes a las bodegas. El cromosoma es una cadena de $n$ números enteros, en donde cada posición indica a cuál bodega está asignado el $j$-ésimo cliente. Esta vez se han considerado explícitamente las variables $x_{ij}$.

La Figura 2.2 muestra las mismas configuraciones utilizadas en la Figura 2.1 para ejemplificar la representación binaria, pero con los cromosomas correspondientes a la representación entera. Esta segunda representación ha sido utilizada para resolver problemas de asignación generalizada [15], cobertura de conjuntos [14], ruteo de vehículos [6] y localización [25].

Para completar una solución factible de PLM es necesario determinar los valores de las variables $y_i$ garantizando que los clientes son asignados a bodegas abiertas. Esto se logra asumiendo que cuando un cliente se asigna a alguna bodega, dicha bodega ya está abierta.

Sea $b_j^l = \{i : x_{ij} = 1\}$, un conjunto unitario, formado por el índice de la bodega que atiende al $j$-ésimo cliente en el $l$-ésimo cromosoma de la población. El conjunto de bodegas abiertas en el $l$-ésimo cromosoma está definido de la siguiente forma:

$$\mathcal{O}^l = \bigcup_{j \in \mathcal{J}} b_j^l \tag{2.3}$$

Las variables $y_i$ se determinan así:

$$y_i = 1, \text{ para todo } i \in \mathcal{O}^l \tag{2.4}$$

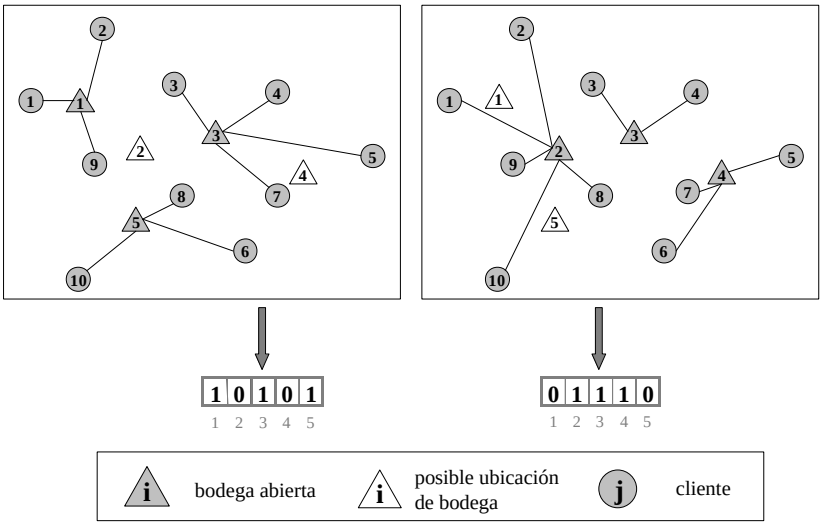


Figura 2.1: Ejemplo de representación binaria para problemas de localización.

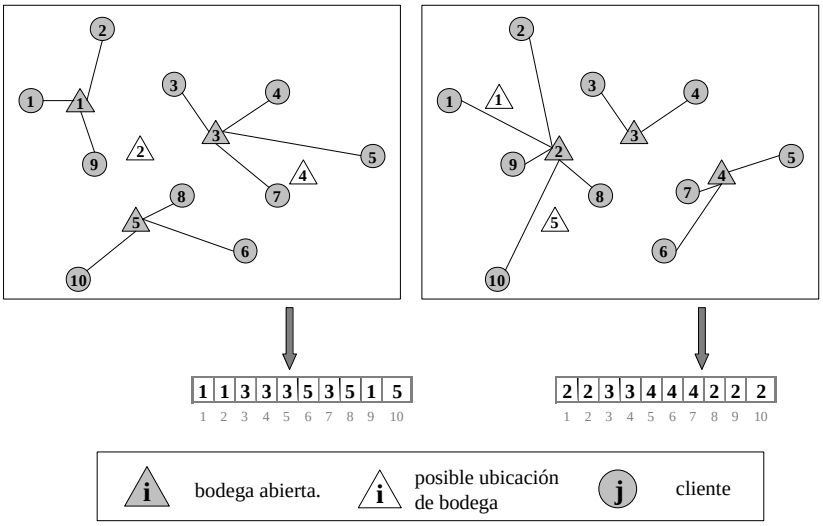


Figura 2.2: Ejemplo de representación entera para problemas de localización.

Con este procedimiento de obtención de las variables $y_i$ se garantiza que todas las soluciones generadas son factibles.

### 2.2.2. Generación de la Población Inicial

Para obtener la población inicial, con la cual comienza el proceso del algoritmo evolutivo, se utiliza un procedimiento bastante simple. En ambas representaciones se hace de forma aleatoria.

En la representación binaria, se generan cromosomas factibles con cadenas aleatorias de ceros y unos de longitud $m$. Para seleccionar el valor de cada posición se genera un número aleatorio uniforme en el intervalo $(0, 1)$, si es menor que 0.5 se toma el valor 0, y si es mayor o igual que 0.5 el valor 1.

Para la representación entera, se genera una cadena de números enteros de longitud $n$. El valor de cada posición se selecciona generando un número aleatorio discreto en el intervalo $[1, m]$.

### 2.2.3. Evaluación y Selección

En un AE multiobjetivo la adaptación de un cromosoma (solución) depende de todas las funciones objetivo y de su comparación con las demás soluciones. En este caso las funciones objetivo son costo (1.3) y cobertura (1.4), con base en sus valores se determina si un cromosoma domina a otro o no.

Por ejemplo, si se tienen tres cromosomas:

| Cromosoma | Costo | Cobertura |
|---|---|---|
| $c1$ | 50 | 75 |
| $c2$ | 65 | 80 |
| $c3$ | 70 | 75 |

Tabla 2.1: Ejemplo de comparación de cromosomas

Se pueden hacer las siguientes afirmaciones:

- $c1 \preceq c3$, tienen la misma cobertura, pero $c1$ tiene un menor costo.
- $c2 \preceq c3$, esta vez tanto el costo como la cobertura de $c2$ son mejores que los de $c3$.
- Cuando se comparan $c1$ y $c2$ no se puede decir que alguno es mejor (i.e., ninguno domina al otro).

Para elegir los padres de las nuevas soluciones se usó el Método de Selección por Torneo Binario: se toman al azar dos cromosomas de la población, se comparan y el que tenga una mejor adaptación es escogido como padre. Los cromosomas mejor adaptados tienen la oportunidad de generar más hijos que los peor adaptados, ya que pueden ganar más torneos [8].

### 2.2.4. Cruce

Los padres seleccionados de $\mathcal{P}(t)$ por torneo binario se cruzan para obtener $\mathcal{C}(t)$. Para llevar a cabo el cruce se probaron dos estrategias: cruce en un punto y cruce uniforme.

En el cruce en un punto se selecciona aleatoriamente una posición del cromosoma y se producen dos hijos: El primer hijo tiene la información del primer padre hasta el punto de cruce y de allí en adelante la del segundo padre. El segundo hijo se genera invirtiendo el papel de los padres. Un ejemplo ilustra mejor este proceso (Figura 2.3).

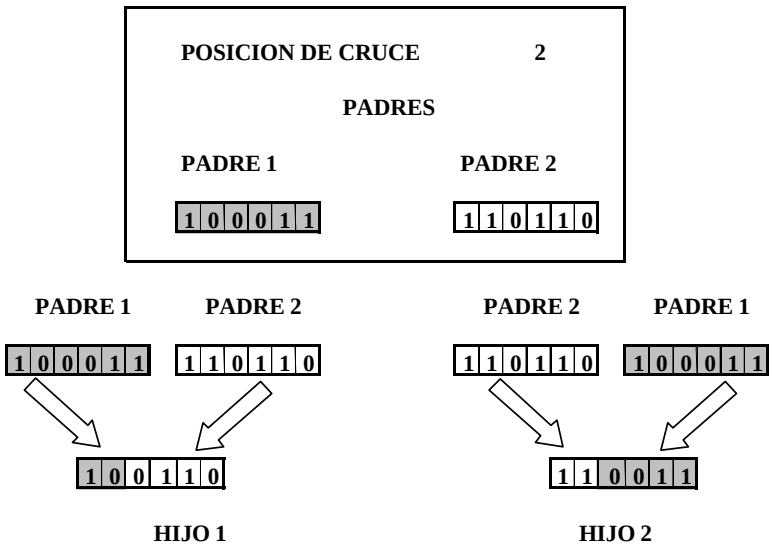


Figura 2.3: Cruce en un punto.

Para el cruce uniforme se genera un patrón aleatorio que indica cual padre aportará su información en cada una de las posiciones del cromosoma. Al generar el patrón cada padre tiene 5) % de probabilidad de ser seleccionado en cada una de las posiciones del cromosoma. Un segundo hijo se genera invirtiendo el patrón de cruce (i.e., en cada posición del cromosoma se selecciona al padre que no fue seleccionado en el primer hijo). Un ejemplo ilustra mejor este operador de cruce (Figura 2.4).

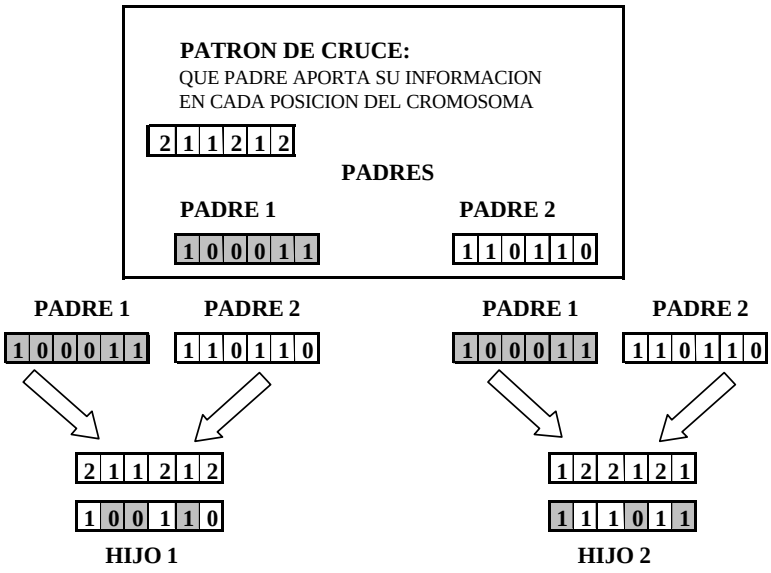


Figura 2.4: Cruce uniforme.

### 2.2.5. Mutación

La mutación modifica el valor de algunas de las posiciones del cromosoma. Es aplicada a cada nuevo cromosoma después del cruce o para generar una nueva solución a partir de una existente. En general, la probabilidad de que un cromosoma mute debe ser baja para evitar que el algoritmo se comporte como una búsqueda aleatoria. Dadas las diferencias de las dos representaciones fue necesario desarrollar dos mecanismos de mutación diferentes.

Para mutar los cromosoma de la representación binaria se genera un patrón aleatorio individual, tomando $m$ numeros aleatorios de una distribución uniforme en el intervalo (0,1), sólo cambian aquellas posiciones del cromosoma donde el valor del patrón es menor que $p_{mb}$ (probabilidad de mutación por posición). La figura 2.5 muestra un ejemplo.

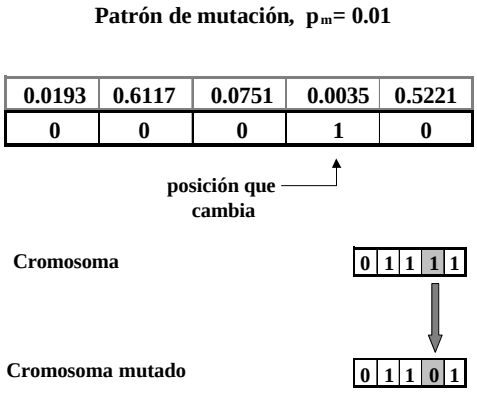


Figura 2.5: Ejemplo de mutación para la representación binaria.

Para la representación entera se tienen dos operados de mutación diferentes. Primero se decide si un cromosoma particular va a mutar o no: generando un número aleatorio uniforme en el intervalo (0,1), el cromosoma muta si el número generado es menor que $p_{mc}$ (probabilidad de mutación por cromosoma). Una vez se sabe que el cromosoma va a mutar se selecciona aleatoriamente, con igual probabilidad, uno de los operadores de mutación.

El primero de los operadores de mutación asigna algunos clientes[1] seleccionados aleatoriamente a otras bodegas que también son seleccionadas aleatoriamente, la Figura 2.7 ilustra el operador de mutación con un ejemplo. El segundo operador de mutación, selecciona aleatoriamente dos clientes e intercambia las bodegas a las que están asignados (Figura 2.6).

[1] menos del 10 % de los clientes

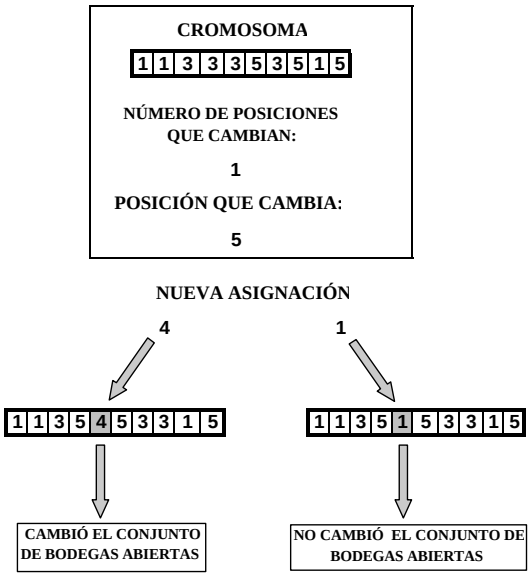


Figura 2.6: Ejemplo de mutación con asignación a otras bodegas.

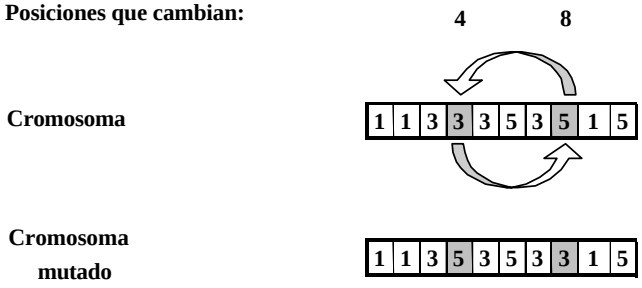


Figura 2.7: Ejemplo de mutación con intercambio de la asignación.

### 2.2.6. Criterio de Parada

El criterio de parada de los algoritmos evolutivos multiobjetivo es una pregunta que se mantiene abierta a la investigación [21]. En este trabajo se optó por fijar un número máximo de generaciones después del cual se detiene la ejecución del algoritmo evolutivo.

## 2.3. Algoritmos Evolutivos Implementados

En esta sección se presenta la implementación de dos algoritmos evolutivos para aproximar la frontera de Pareto del problema de localización multiobjetivo descrito previamente. Los dos algoritmos evolutivos son bastante diferentes. El primero es similar a un algoritmo evolutivo tradicional, mientras que el segundo es una estrategia de búsqueda local que mantiene un archivo de soluciones no dominadas para evaluar las nuevas soluciones .

### 2.3.1. Nondominated Sorting Genetic Algorithm II

Conocido como NSGA, fue introducido en su version inicial en 1994 [35], posteriormente, fue modificado para mejorar su eficiencia computacional dando origen al NSGAII [34].

En NSGAII la selección de los cromosomas se basa en la clasificación de la población en frentes de soluciones no dominadas $(\mathcal{F}_1, ..., \mathcal{F}_c)$. El primer frente $(\mathcal{F}_1)$ lo conforman las soluciones que no son dominadas por ninguna otra solución de la población, el segundo frente $(\mathcal{F}_2)$ las soluciones que serían eficientes si se ignoraran las del primer frente. Así, se van creando frentes hasta clasificar toda la población.

Inicialmente se construye una población aleatoria $(\mathcal{P}_0)$ de tamaño $L$, se evalúa cada uno de los cromosomas de la población inicial, determinando su costo y su cobertura. Evaluados los cromosomas, se aplica clasificación no dominada a la población y a cada cromosoma se le asigna como función de adaptación el frente al que pertenece. Luego, se utilizan torneo binario y operadores de cruce y mutación a $(\mathcal{P}_0)$ para generar la población de hijos $(\mathcal{C}_0)$ de tamaño $L$.

Una vez inicializado el algoritmo, el proceso de selección se repite durante $T$ generaciones: a la población resultante de unir padres e hijos $(\mathcal{R}_t)$ se le aplica clasificación no dominada. Los padres de la generación siguiente $(\mathcal{P}_t)$, se seleccionan tomando los primeros frentes de $(\mathcal{R}_t)$ hasta tener exactamente $L$ soluciones. En todos las operaciones de selección, la densidad de la población en la vecindad de un cromosoma se utiliza como medida secundaria de adaptación (i.e., cuando se comparan dos cromosoma de un mismo frente es mejor aquel que tenga menos soluciones en su vecindad). Aplicando operadores de selección, cruce y mutación a la población de padres $(\mathcal{P}_{t+1})$ se obtiene la población de hijos $(\mathcal{C}_{t+1})$. A continuación se presenta el algoritmo NSGAII:

Para clasificar la población en frentes se comparan todos los cromosomas entre si, determinando el subconjunto de cromosomas dominados por cada cromosoma $(\mathcal{D}^l)$ y el número de cromosomas que dominan a cada cromosoma $(o^l)$. Conocida esta información se determinan los frentes de la población. El procedimiento se presenta en el Algoritmo 3.

**Algoritmo 2** NSGAII

Generar aleatoriamente $\mathcal{P}_0$
Evaluar las funciones objetivo de los cromosomas en $\mathcal{P}_0$
Clasificación no dominada de $\mathcal{P}_0$ (Algoritmo 3)
Generar $\mathcal{C}_0$ aplicando torneo binario, cruce y mutación a $\mathcal{P}_0$
Evaluar las funciones objetivo de los cromosomas en $\mathcal{C}_0$
**while** $t \leq T$ **do**
  $\mathcal{R}_t = \mathcal{P}_t \cup \mathcal{C}_t$
  Clasificación no dominada de $\mathcal{R}_t$ (Algoritmo 3)
  Organizar $\mathcal{R}_t$ utilizando $\leq_{nsga}$
  Formar $\mathcal{P}_{t+1}$ con los primeros $L$ cromosomas de $\mathcal{R}_t$
  Aplicar torneo binario y cruzar $\mathcal{P}_{t+1}$ para obtener $\mathcal{C}_{t+1}$
  Mutar $\mathcal{C}_{t+1}$
  Evaluar las funciones objetivo de los cromosomas en $\mathcal{C}_t$
  $t \leftarrow t + 1$
**end while**

**Algoritmo 3** CLASIFICACIÓN NO DOMINADA

**for all** $\mathbf{x} \in \mathcal{R}_t$ **do**
  **for all** $\mathbf{y} \in \mathcal{R}_t$ **do**
    **if** $\mathbf{x} \prec \mathbf{y}$ **then**
      $\mathcal{D}^x \leftarrow \mathcal{D}^x \cup \{\mathbf{y}\}$
      $o^y \leftarrow o^y + 1$
    **else if** $y \prec x$ **then**
      $\mathcal{D}^y \leftarrow \mathcal{D}^y \cup \{\mathbf{x}\}$
      $o^x \leftarrow o^x + 1$
    **end if**
  **end for**
**end for**
$c \leftarrow 1$ (Contador de frentes)
$l \leftarrow 1$ (Contador de cromosomas)
**while** $l < L$ **do**
  **for all** $\mathbf{x} \in \mathcal{R}_t$ **do**
    **if** $o^{\mathbf{x}} = 0$ **then**
      $\mathcal{F}_c \leftarrow \mathcal{F}_c \cup \{\mathbf{x}\}$
      $l \leftarrow l + 1$
      **for all** $\mathbf{y} \in \mathcal{D}^x$ **do**
        $o^y \leftarrow o^y - 1$
      **end for**
    **end if**
  **end for**
  Densidad Local de $\mathcal{F}_c$ (Algoritmo 4)
  $c \leftarrow c + 1$
**end while**

La distribución uniforme de las soluciones sobre el frente de Pareto es una de las características deseables cuando se busca resolver un problema de optimización multiobjetivo [57]. Es preferible tener soluciones dispersas en el rango de cada una de las funciones objetivo que concentradas en una sola región, en el segundo caso posiblemente no existan grandes diferencias entre una solución y otra.

Al seleccionar los cromosomas que darán origen a una nueva generación se prefieren los cromosomas mejor adaptados y con menos soluciones a su alrededor. En consecuencia, es necesario estimar la densidad de la población en la vecindad de cada solución.

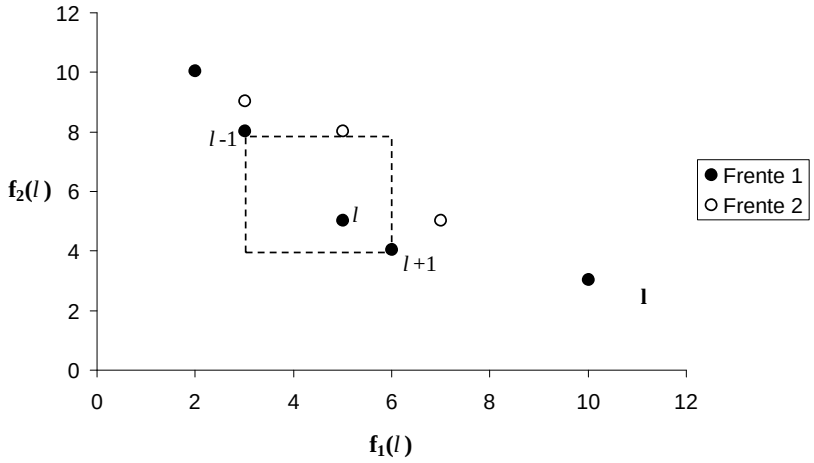


Figura 2.8: Ilustración del método de estimación de la densidad local (Dos funciones objetivo).

Con el método descrito en el algoritmo 4 se encuentra el tamaño promedio de la arista del cuboide que encierra a cada cromosoma sin incluir otro de su mismo frente ($e^l$). $e^l$ es la medida que se utiliza para estimar la densidad de la población en la vecindad de un cromosoma.

**Algoritmo 4** ESTIMACIÓN DE LA DENSIDAD LOCAL

1: $b \leftarrow |\mathcal{F}_c|$ (Número de soluciones en el frente)
2: **for all** $\mathbf{x} \in \mathcal{F}_c$ **do**
3: $e^x \leftarrow 0$ (Inicialice las distancias)
4: **end for**
5: **for all** $k$ **do**
6: Organizar $\mathcal{F}_c$ usando el k-ésimo objetivo
7: $e^1 = e^b = \infty$ (Los puntos extremos siempre se seleccionan)
8: **for all** $l = 2$ to $(b-1)$ **do**
9: $e^l \leftarrow e^l + (f_k^{l+1} - f_k^{l-1})$
10: **end for**
11: **end for**

Cada vez que se comparan dos cromosomas, bien sea en un torneo binario o para organizar la población, se emplea un comparador que permite saber cual está mejor adaptado. En general, se prefiere el cromosomas que pertenece al frente menor, y si ambos pertenecen al mismo frente se prefiere el que se encuentre en una región menos densa. El comparador no dominado ($\leq_{nsga}$ ) se define de la siguiente manera:

**Definición 4** *Sean* $\mathbf{x}, \mathbf{y} \in \mathcal{R}_t$*, dos cromosomas de una población,* $F^x$ *el frente al que pertenece* $\mathbf{x}$*,* $F^y$ *el frente al que pertenece* $\mathbf{y}$*. Se dice que* $\mathbf{x}$ *está mejor adaptado que* $\mathbf{y}$ *(*$\mathbf{x} \leq_{nsga} \mathbf{y}$*), si:*

$$F^x \leq F^y,$$

*o si* $F^x = F^y$ *y* $e^x > e^y$

### 2.3.2. Pareto Archived Evolutionary Strategy

Conocido como PAES, el segundo algoritmo evolutivo implementado fue introducido en 1999 [55]. Se dice que es la forma más simple de aproximar la frontera de Pareto de un problema multiobjetivo [56]. PAES es una estrategia evolutiva (1+1) en la cual se genera un nuevo cromosoma a partir de un solo padre utilizando únicamente operadores de mutación. En un archivo $\mathcal{A}$ de tamaño máximo $L_{\mathcal{A}}$ se almacenan algunas de las soluciones no dominadas encontradas durante la ejecución del algoritmo, para ser utilizadas como referencia en la evaluación de los nuevos cromosomas.

Al algoritmo original propuesto en [55, 56] se le introdujeron algunas variaciones con el fin de utilizar parte de las herramientas implementadas para NSGAII: una vez inicializado el archivo, se escoge aleatoriamente uno de los cromosoma ($\mathbf{x}_0$) para que mutándolo se genere una nueva solución ($mathbfy_0$. En el ciclo principal de PAES se comparan $\mathbf{x_t}$ y $\mathbf{y_t}$ para determinar cuál de ellos domina y a partir de cuál se generarán las nuevas soluciones. Si ninguno de los cromosomas domina al otro, se recurre al archivo ($\mathcal{A}_\sqcup$) para decidir si $\mathbf{y_t}$ se almacena en el archivo o no, y para seleccionar a partir de cuál de los cromosomas ($\mathbf{x_t}$ o $\mathbf{y_t}$) se generarán las nuevas soluciones. El algoritmo 5 presenta los detalles de la implementación modificada de PAES.

Cuando la comparación entre la solución actual $\mathbf{x}_t$ y su mutación $\mathbf{y}_t$ no es concluyente se recurre al archivo para evaluar las dos soluciones. El procedimiento de comparación de $\mathbf{y}_t$, $\mathbf{x}_t$

**Algoritmo 5** PAES

$t \leftarrow 0$
Inicializar $\mathcal{A}$
Seleccionar aleatoriamente $\mathbf{x_0}$ de $\mathcal{A}$
**while** $t < T_{PAES}$ **do**
  Aplicar operadores de mutación a $\mathbf{x}_t$ para producir $\mathbf{y}_t$
  Evaluar $\mathbf{y}_t$
  **if** $\mathbf{x}_t \preceq \mathbf{y}_t$ **then**
    Descartar $\mathbf{y}_t$
    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$
  **else if** $\mathbf{y}_t \preceq \mathbf{x}_t$ **then**
    Descartar $\mathbf{x}_t$
    $\mathbf{x}_{t+1} \leftarrow \mathbf{y}_t$
    Guardar $\mathbf{y}_t$ en $\mathcal{A}$
    Borrar de $\mathcal{A}$ las soluciones dominadas por $\mathbf{y}_t$
  **else if** Existe $\mathbf{l}$ en $\mathcal{A}$ tal que: $\mathbf{l} \preceq \mathbf{y}_t$ **then**
    Descartar $\mathbf{y}_t$
    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$
  **else**
    Comparar $\mathbf{y}_t$, $\mathbf{x}_t$ y $\mathcal{A}$ para seleccionar $\mathbf{x}_{t+1}$ y determinar si $\mathbf{y}_t$ se almacena en $\mathcal{A}$ o no
  **end if**
  $t \leftarrow t + 1$
**end while**

y $\mathcal{A}$ se se describe en el Algoritmo 6. Como mecanismo para mantener la diversidad se utiliza la estimación de la densidad local implementada para NSGAII (Algoritmo 4).

**Algoritmo 6** PROCEDIMIENTO PARA COMPARAR $\mathbf{y}_t$, $\mathbf{x}_t$ y $\mathcal{A}$

1: Densidad Local de $\mathcal{A} \cup \mathbf{y}_t$ (Algoritmo 4)
2: **if** El archivo no está lleno **then**
3: $\mathcal{A} \cup \{\mathbf{y}_t\}$ (Agregar $\mathbf{y}_t$ al archivo)
4: **if** $e^y < e^x$ ($\mathbf{y}_t$ está en una región menos densa que $\mathbf{x}_t$) **then**
5: $\mathbf{x}_{t+1} \leftarrow \mathbf{y}_t$ (Aceptar $\mathbf{y}_t$ como solución para generar los nuevos cromosomas)
6: **else**
7: $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$ (Mantener $\mathbf{x}_t$ como solución para generar los nuevos cromosomas)
8: **end if**
9: **else**
10: **if** $e^{y_t} < e^l, \mathbf{l} \in \mathcal{A}$ ($\mathbf{y}_t$ está en una región menos densa que alguna solución del archivo) **then**
11: $\mathcal{A} \cup \{\mathbf{y}_t\} - \{\mathbf{l}\}$ (Agregar $\mathbf{y}_t$ al archivo y remover $\mathbf{l}$)
12: **if** $e^{y_t} < e^{x_t}$ ($\mathbf{y}_t$ está en una región menos densa que $\mathbf{x}_t$) **then**
13: $\mathbf{x}_{t+1} \leftarrow \mathbf{y}_t$ (Aceptar $\mathbf{y}_t$ como solución para generar los nuevos cromosomas)
14: **else**
15: $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$ (Mantener $\mathbf{x}_t$ como solución para generar los nuevos cromosomas)
16: **end if**
17: **else**
18: **if** $e^{y_t} < e^{x_t}$ ($\mathbf{y}_t$ está en una región menos densa que $\mathbf{x}_t$) **then**
19: $\mathbf{x}_{t+1} \leftarrow \mathbf{y}_t$ (Aceptar $\mathbf{y}_t$ como solución para generar los nuevos cromosomas)
20: $\mathcal{A} \cup \{\mathbf{y}_t\} - \{\mathbf{x}_t\}$ (Agregar $\mathbf{y}_t$al archivo y remover $\mathbf{x}_t$)
21: **else**
22: $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$ (Mantener $\mathbf{x}_t$ como solución para generar los nuevos cromosomas)
23: **end if**
24: **end if**
25: **end if**

## 2.4. Experimento Computacional

Para comparar los algoritmos evolutivos implementados se desarrollaron varios experimentos computacionales, en los que se observo el desempeño de los algoritmos para aproximar la frontera de Pareto de varios problemas de localización multiobjetivo generados aleatoriamente.

### 2.4.1. Medidas de Desempeño para Algoritmos Evolutivos Multiobjetivo

El diseño de medidas para comparar algoritmos de optimización multiobjetivo es un campo activo de investigación. El resultado de un algoritmo evolutivo multiobjetivo no es una sola solución, sino un conjunto de soluciones no dominadas. De ahí que se hayan desarrollado diferentes métricas que permiten evaluar la calidad de la aproximación del Frente de Pareto obtenida por un algoritmo evolutivo. Para un detallado análisis de las métricas existentes se pueden consultar [57, 83, 85].

Siguiendo las recomendaciones de [57], se utilizó la métrica $S$ introducida en [84], que mide la calidad de una aproximación del frente de Pareto con el tamaño del espacio dominado por dicha aproximación. Entre más grande sea el espacio dominado mejor se considera la aproximación.

La calidad de una aproximación del frente de Pareto esta dada por el tamaño del espacio (área, volumen o hipervolumen) comprendido entre dicha aproximación y un punto de referencia. La unión del espacio dominado por todas las soluciones que componen la aproximación del frente de Pareto es el valor de la métrica $S$. Para los problemas de localización abordados el punto de referencia utilizado es el anti–ideal (i.e., una solución que tiene el máximo costo[2] y la mínima cobertura[3]). En la Figura 2.9 se ilustra la métrica $S$ para un problema de localización multiobjetivo, cada solución marcada con ♦ induce un espacio dominado (rectángulo sombreado).

El valor de la métrica $S$ depende de las unidades que se utilicen para medir cada uno de los criterios de decisión. Utilizando el enfoque de [67], se obtiene la métrica modificada $S'$ con valores entre cero y uno. $S'$ se define , como la división del espacio dominado por la aproximación del frente $\widehat{PF}^*$ y el espacio dominado por la solución ideal $v(\mathbf{x}^*)$. La solución ideal es un vector formado por las soluciones óptimas de cada función objetivo, obtenidas optimizando cada criterio sin considerar los demás. La solución ideal es rara vez factible y solo es alcanzable si no existe conflicto entre las funciones objetivo.

Para una aproximación de la frontera de Pareto $\widehat{PF}^*$, la métrica $S'$ está definida por:

$$S' = \frac{S(PF^*)}{S(v(\mathbf{x}^*))}$$

.

Claramente, la solución ideal tendrá $S' = 1$, y cuando no exista ninguna solución factible el valor de $S'$ será cero, puesto que no hay ninguna solución que defina un espacio dominado.

Antes de aplicar la métrica $S'$ es importante definir los limites inferior y superior del espacio

[2]El costo de la solución de maxima cobertura
[3]La cobertura de la solución de mínimo costo

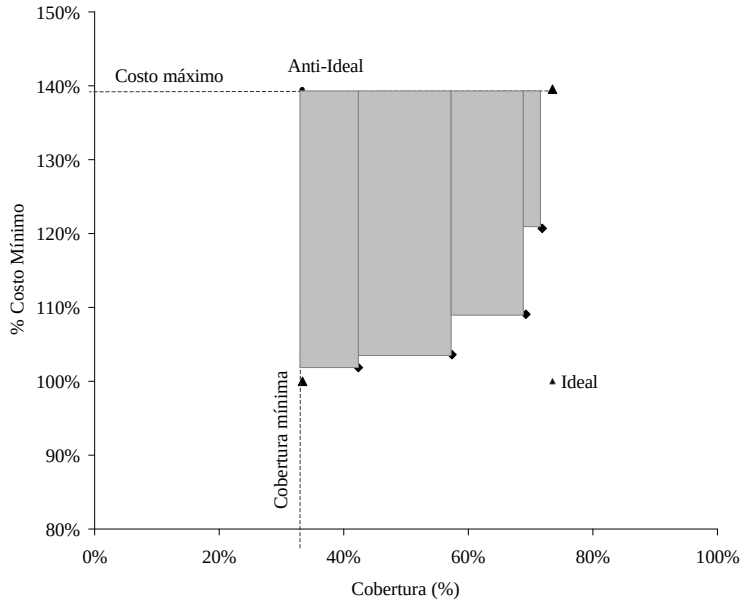


Figura 2.9: Ejemplo de la métrica $S$.

de soluciones no dominadas. Los puntos que limitan el espacio de soluciones no dominadas son precisamente las soluciones de mínimo costo y de máxima cobertura (marcadas con ▲ en las Figuras 2.9 y 2.10). Toda solución que no esté dentro de dichos limites siempre estará dominada por alguna de ellas. Por ejemplo, en la Figura 2.10, el punto **A** está dominado por la solución de mínimo costo ya que tiene un costo superior y una cobertura inferior a las de dicha solución (i.e., un peor servicio a un mayor costo). Igualmente, el punto **B** está dominado por la solución de máxima cobertura, ya que tiene un costo mayor y una cobertura inferior a las de dicha solución.

El resultado anterior exige que siempre que se obtenga una aproximación del Frente de Pareto, utilizando un algoritmo evolutivo, se deban eliminar todas las soluciones que tengan una cobertura inferior a la de la configuración de mínimo costo o un costo superior al de la configuración de máxima cobertura. Para filtrar dichas soluciones es necesario determinar las soluciones extremas de la frontera de Pareto del problema de localización multiobjetivo (configuraciones de costo mínimo y máxima cobertura).

### 2.4.2. Soluciones Extremas de la Frontera de Pareto

Para encontrar las soluciones extremas de la frontera de Pareto se utilizó programación matemática, de este modo se garantiza que las soluciones encontradas pertenecen a la frontera de Pareto y no a una aproximación

La solución de mínimo costo se obtiene en dos fases. Primero se resuelve el UFLP ((2.5)–(2.9)) y el valor óptimo ($z_1^*$) es la cota minima del costo ($\underline{z}_1$) de cualquier configuración de la red.

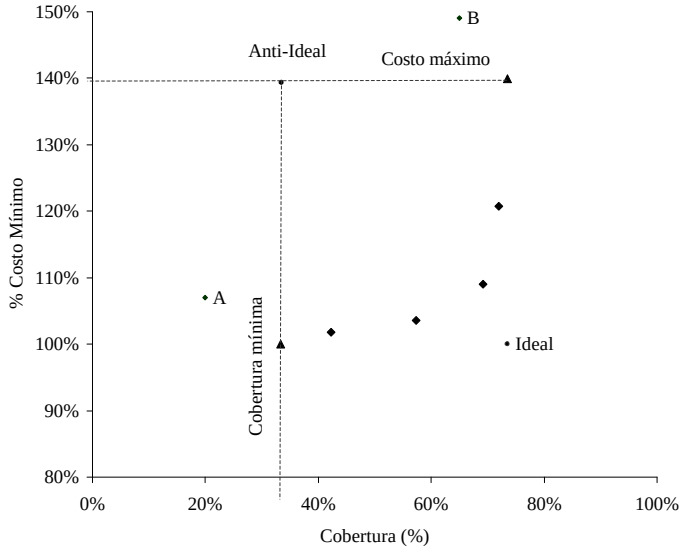


Figura 2.10: Límites para la obtención de la métrica $S'$.

$$\text{mín} \quad z_1 = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} c_{ij}x_{ij} + \sum_{i\in\mathcal{I}} f_i y_i \tag{2.5}$$

Sujeto a,

$$\sum_{i\in\mathcal{I}} x_{ij} = 1 \quad , \quad j \in \mathcal{J} \tag{2.6}$$

$$x_{ij} \leq y_i \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{2.7}$$

$$x_{ij} \in \{0,1\}, \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{2.8}$$

$$y_i \in \{0,1\}, \quad i \in \mathcal{I} \tag{2.9}$$

Una vez hallado el costo mínimo se busca maximizar la cobertura alcanzable con dicho costo. Cualquier solución que tenga una cobertura inferior estará dominada por la solución de mínimo costo, ya que necesariamente será más costosa y además ofrecerá una menor cobertura. La solución de mínimo costo ($\underline{z}_1$) y mínima cobertura ($\underline{z}_2$) queda completamente determinada al resolver el problema de optimización 2.10 –2.15. El valor óptimo ($z_2^*$) es la cota minima de cobertura ($\underline{z}_2$) de cualquier solución no dominada.

$$\text{máx} \quad z_2 = \sum_{j\in\mathcal{J}} d_j \sum_{i\in\mathcal{Q}_j} x_{ij} \tag{2.10}$$

Sujeto a,

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} c_{ij}x_{ij} + \sum_{i\in\mathcal{I}} f_i y_i \leq \quad \underline{z}_1 \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{2.11}$$

$$\sum_{i\in\mathcal{I}} x_{ij} = \quad 1 \quad , \quad j \in \mathcal{J} \tag{2.12}$$

$$x_{ij} \leq \quad y_i \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{2.13}$$

$$x_{ij} \in \quad \{0,1\}, \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{2.14}$$

$$y_i \in \quad \{0,1\}, \quad i \in \mathcal{I} \tag{2.15}$$

Para encontrar la solución de máxima cobertura se realiza un procedimiento similar al que se utilizó para encontrar la solución de mínimo costo.

Primero se encuentra la máxima cobertura alcanzable con el conjunto de bodegas disponible ($\overline{z}_2$). Para obtenerla no es necesario resolver ningún problema de optimización. Basta con examinar que clientes estarían cubiertos si se abrieran todas las bodegas, haciendo $y_i = 1$ para todo $i \in I$ y calculando la expresión:

$$\overline{z}_2 = \sum_{j\in J} d_j \cdot min\{1, \sum_{i\in Q_j} y_i\}$$

La expresión $min\{1, \sum_{i\in Q_j} y_i\}$ garantiza que los clientes son contados una sola vez, aunque puedan ser cubiertos por más de una bodega.

En la segunda fase se minimiza el costo necesario para lograr la cobertura $\overline{z}_2$:

$$\text{mín} \quad z_1 = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} c_{ij}x_{ij} + \sum_{i\in\mathcal{I}} f_i y_i \tag{2.16}$$

Sujeto a,

$$\sum_{j\in\mathcal{J}} d_j \sum_{i\in\mathcal{Q}_j} x_{ij} \quad \geq \overline{z}_2 \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{2.17}$$

$$\sum_{i\in\mathcal{I}} x_{ij} = \quad 1 \quad , \quad j \in \mathcal{J} \tag{2.18}$$

$$x_{ij} \leq \quad y_i \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{2.19}$$

$$x_{ij} \in \quad \{0,1\}, \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{2.20}$$

$$y_i \in \quad \{0,1\}, \quad i \in \mathcal{I} \tag{2.21}$$

El valor óptimo $z_1^*$ del problema 2.16 – 2.20 es la cota superior del costo ($\overline{z}_1$) de cualquier solución no dominada. Una solución más costosa aunque tenga la máxima cobertura estará dominada por la solución aquí obtenida, ya que esta es la forma de ofrecer el mejor servicio al menor costo.

### 2.4.3. Afinación de Parámetros

Antes de utilizar un algoritmo evolutivo para la solución de un problema se deben fijar los parámetros que controlan su ejecución (i.e., el tamaño de la población, el número máximo de generaciones y la probabilidad de mutación por posición). Para afinar los parámetros de NSGAII y PAES se utilizó una metodología similar a la empleada en [67], en la cual se prueban todas las combinaciones posibles de los parámetros y se escoge una que obtenga soluciones de buena calidad y no tome demasiado tiempo de ejecución. Para afinar los parámetros se utilizaron cuatro problemas de prueba generados aleatoriamente:

| Problema | m | n |
|---|---|---|
| 1 | 30 | 75 |
| 2 | 30 | 75 |
| 3 | 50 | 150 |
| 4 | 50 | 150 |

Tabla 2.2: Problemas de Prueba.

#### NSGAII

En NSGAII se deben ajustar tres parámetros, tamaño de la población $L$, número máximo de generaciones $T$ y probabilidad de mutación por posición $p_{mb}$ (representación binaria), o por cromosoma $p_{mb}$ (representación entera).

Inicialmente, se ajustaron los parámetros de ejecución de NSGAII con estructura binaria y cruce uniforme. Los niveles de cada uno de los parámetros se muestran en la Tabla 2.3.

| Parámetro | Niveles |
|---|---|
| $L$ | 10, 30, 50 |
| $T$ | 400, 800, 1200 |
| $p_{mb}$ | 0.001, 0.002, ..., 0.005 |

Tabla 2.3: Niveles utilizados para ajustar los parámetros de NSGAII con estructura binaria.

Para cada combinación de parámetros se ejecuto una vez NSGAII en cada problema de prueba. Los resultados obtenidos para el primer problema de prueba se muestran en la Figura 2.11, la Tabla 2.4 presenta los valores de $S'$ obtenidos para cada combinación de parámetros y la Tabla 2.5 se reportan los tiempos de ejecución correspondientes[4].

[4]Los algoritmos evolutivos fueron codificados usando MATLAB 5.3 y los experimentos computacionales se ejecutaron en un AMD Athlon de 1.14 GHz de velocidad y 224 MB de RAM

| $L$ | $T$ | $p_{mb}$ | | | | |
|---|---|---|---|---|---|---|
| | | 0.001 | 0.002 | 0.003 | 0.004 | 0.005 |
| 10 | 400 | 0.671 | 0.696 | 0.686 | 0.674 | 0.747 |
| | 800 | 0.683 | 0.715 | 0.731 | 0.740 | 0.745 |
| | 1200 | 0.712 | 0.726 | 0.744 | 0.752 | 0.750 |
| 30 | 400 | 0.795 | 0.799 | 0.800 | 0.799 | 0.800 |
| | 800 | 0.795 | 0.800 | 0.800 | 0.800 | 0.800 |
| | 1200 | 0.800 | 0.800 | 0.800 | 0.800 | 0.800 |
| 50 | 400 | 0.801 | 0.796 | 0.797 | 0.801 | 0.800 |
| | 800 | 0.801 | 0.797 | 0.797 | 0.801 | 0.801 |
| | 1200 | 0.801 | 0.797 | 0.801 | 0.801 | 0.801 |

Tabla 2.4: Ajuste de parámetros NSGAII con estructura binaria. Valores de $S'$. Problema de prueba 1.

| $L$ | $T$ | $p_{mb}$ | | | | |
|---|---|---|---|---|---|---|
| | | 0.001 | 0.002 | 0.003 | 0.004 | 0.005 |
| 10 | 400 | 26.8 | 26.6 | 26.3 | 25.8 | 26.0 |
| | 800 | 53.6 | 52.4 | 51.3 | 51.6 | 51.6 |
| | 1200 | 80.2 | 78.2 | 76.8 | 76.6 | 77.1 |
| 30 | 400 | 107.5 | 108.7 | 111.7 | 110.7 | 110.5 |
| | 800 | 246.9 | 249.3 | 253.1 | 248.7 | 248.7 |
| | 1200 | 354.1 | 359.0 | 363.3 | 364.2 | 320.1 |
| 50 | 400 | 231.9 | 221.6 | 232.8 | 233.9 | 232.2 |
| | 800 | 520.0 | 499.9 | 521.1 | 458.1 | 458.6 |
| | 1200 | 683.3 | 680.7 | 688.3 | 690.6 | 685.5 |

Tabla 2.5: Ajuste de parámetros NSGAII con estructura binaria. Tiempo de ejecución en segundos. Problema de prueba 1.

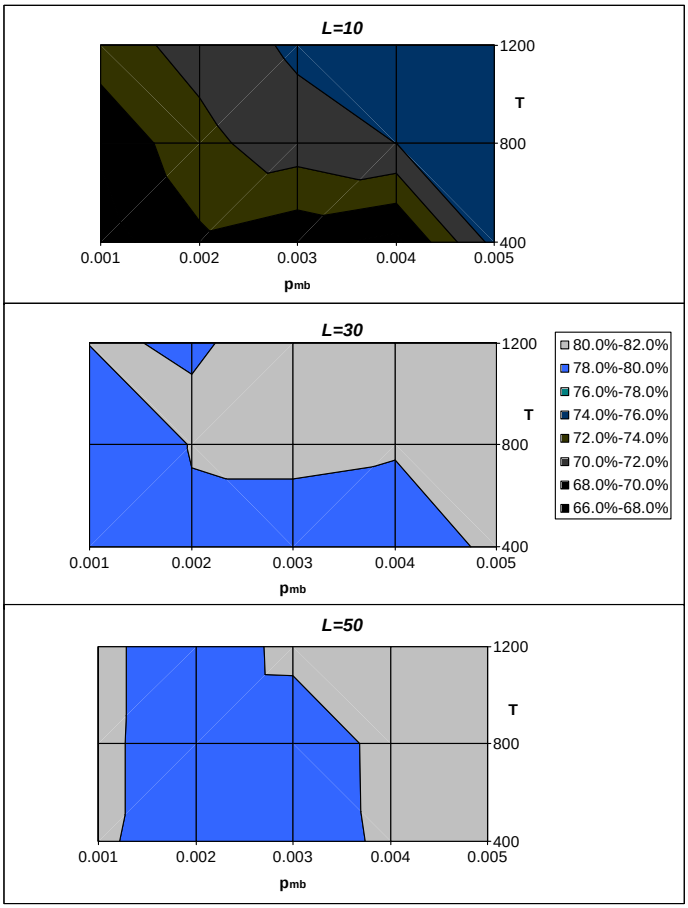


Figura 2.11: Afinación de parámetros. Problema de prueba 1.

Los resultados obtenidos con los problemas de prueba muestran una clara tendencia de NSGAII a necesitar mayor tiempo de ejecución cuando tiene poblaciones grandes (Figura 2.12). Seguramente, debido a la necesidad de comparar muchas soluciones en cada iteración para clasificar la población en frentes. También se observa como los resultados mejoran bastante con poblaciones más grandes.

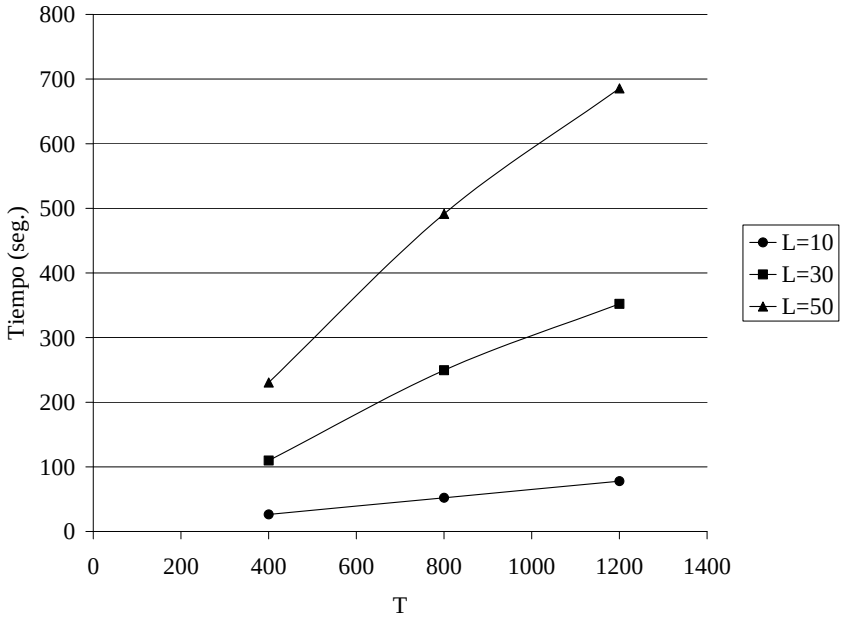


Figura 2.12: Afinación de parámetros. Problema de prueba 1. Tiempos de ejecución promedio

Las combinaciones que obtuvieron resultados prometedores en todos los problemas de prueba se exploraron más profundamente (Tabla 2.6), ejecutando 10 corridas independientes de NSGAII con cada una de las opciones de configuración. Se realizó la prueba no paramétrica de Wilcoxon para comparar las opciones de configuración. En ninguno de los casos se obtuvo diferencia significativa (con un nivel $\alpha = 0{,}05$) entre la opción 1 y la opción 2 ó entre la opción 1 y la opción 3 (vér el Apéndice C para una explicación de la prueba). Como la opción 1 con $T = 800$, $L = 30$ y $p_{mb} = 0{,}003$ tiene menor tiempo de ejecución, se decidió utilizarla para la configuración de NSGAII.

| Opción | $L$ | $T$ | $p_{mb}$ |
|---|---|---|---|
| 1 | 30 | 800 | 0.003 |
| 2 | 30 | 1200 | 0.004 |
| 3 | 50 | 1200 | 0.004 |

Tabla 2.6: Ajuste NSGAII con estructura binaria. Combinaciones prometedoras.

Establecidos los parámetros para NSGAII con estructura binaria y cruce uniforme, se exploró como alternativa utilizar cruce en un punto. Los demás parámetros ($T$,$L$ y $p_{mb}$) se mantuvieron en los valores establecidos en el experimento anterior (opción 1). Los resultados obtenidos en diez corridas de cada una de las opciones se presentan en la Tabla 2.7. Con la

prueba de suma de rangos de Wilcoxon no se obtuvo diferencia significativa (con un nivel $\alpha = 0{,}05$) entre las alternativas de cruce, entonces se optó por utilizar cruce uniforme.

| Semilla | Cruce Uniforme | Cruce en un Punto |
|---|---|---|
| 1 | 0.7424 | 0.7559 |
| 2 | 0.7200 | 0.7505 |
| 3 | 0.7454 | 0.7472 |
| 4 | 0.7413 | 0.7529 |
| 5 | 0.7760 | 0.7563 |
| 6 | 0.7104 | 0.7549 |
| 7 | 0.7507 | 0.7370 |
| 8 | 0.6913 | 0.7612 |
| 9 | 0.7806 | 0.7649 |
| 10 | 0.7927 | 0.7564 |

Tabla 2.7: Ajuste NSGAII con estructura binaria. Comparación de operadores de cruce. Valores de $S'$.

Para terminar, se exploró la utilización de la representación entera. Los parámetros al ajustar NSGAII con representación entera son básicamente los mismos de la representación binaria. La única diferencia es la probabilidad de mutación $p_{mc}$ que se refiere a la probabilidad de que un individuo cambie. mientras que en la estructura binaria la probabilidad de mutación se refería a la probabilidad de que una posición cualquiera del cromosoma cambiara. Para hacer comparables los valores de $p_{mc}$ con los de $p_{mb}$ los niveles considerados de$p_{mc}$ fueron 0.05, 0.10, 0.15 y 0.20. Estos valores hacen que el número de cromosomas que mutan sea similar en ambos algoritmos.

Los resultados obtenidos al calibrar los parámetros de NSGAII con representación entera fueron bastante malos. Los tiempos de ejecución son similares a los de la representación binaria pero los valores de $S'$ son muy inferiores a los obtenidos con la representación entera. Por ejemplo, para el problema de prueba 1 los valores de $S'$ obtenidos no superan 0.20 (después de explorar 24,000 soluciones y tomarse tiempos similares a los de NSGAII con representación binaria), mientras que para el mismo problema hasta la calibración más pobre de la representación binaria obtiene valores superiores al 0.65 en solo 15 segundos.

El desempeño tan pobre de la representación entera se puede explicarse analizando el tamaño de los espacios de búsqueda de cada una de las representaciones (Tabla 2.8). El espacio de búsqueda de la representación binaria es de tamaño $2^m$ mientras que el de la representación entera es de tamaño $m^n$. Por ejemplo, para un problema con 10 bodegas y 25 clientes el tamaño del espacio de búsqueda de la representación entera es 9.7656E+21 veces más grande que el espacio de búsqueda de la representación binaria. En vista del pobre desempeño de la representación entera se decidió no seguir utilizándola en más experimentos.

### PAES

En PAES deben ajustarse el tamaño del archivo ($L_{\mathcal{A}}$) y el número máximo de generaciones. Para garantizar que las comparaciones entre PAES y NSGAII son equitativas se fijó el tamaño del archivo $L_{\mathcal{A}}$ en 30, de manera que ambos algoritmos pueden tener el mismo número de

| m | n | Número de Soluciones | |
|---|---|---|---|
| | | Representación Binaria | Representación Entera |
| 10 | 25 | 1024 | 1.0000E+25 |
| 30 | 75 | 1.0737E+09 | 6.0827E+110 |
| 50 | 150 | 1.1259E+15 | 7.0065E+254 |

Tabla 2.8: Tamaño del espacio de búsqueda.

soluciones no dominadas al final de su ejecución. Para que ambos algoritmos evalúen el mismo número de soluciones se fijo el número máximo de generaciones de PAES $T_{PAES}$ en 24,000 ($T * L = 30 * 800$).

Para no contar como nuevas soluciones que ya están disponibles en la población (o el archivo) y para evitar que la población se llene de cromosomas iguales. En ninguno de los algoritmos se permiten duplicados. Es decir, después de aplicar operadores de mutación a cualquier cromosoma se verifica si el resultado es una solución que ya está en la población, si así es, no se cuenta como nueva solución y se repite el proceso que lo generó (cruce+mutación para NSGAII o mutación para PAES).

La comparación es más eficiente cuando un cromosoma solo se compara con aquellos que tienen el mismo número de bodegas abiertas, esto reduce el número de comparaciones que se deben hacer. Por ejemplo, cuando no existe ningún cromosoma con el mismo número de bodegas se identifica inmediatamente un cromosoma nuevo.

Las condiciones iniciales de ambos algoritmos deben permitir comparaciones equitativas. De ahí que, PAES se inicialice de la misma forma que NSGAII: En el archivo se guardan las soluciones del primer frente de la población inicial de NSGAII $\mathcal{R}_0$ y de allí se escoge la aleatoriamente la solución $\mathbf{x}_0$.

Los parámetros finales escogidos para ejecutar NSGAII y PAES después de la afinación se resumen en la Tabla 2.9.

| NSGAII | | PAES | |
|---|---|---|---|
| $L$ | 30 | $L_{\mathcal{A}}$ | 30 |
| $T$ | 800 | $T_{PAES}$ | 24000 |
| $p_{mb}$ | 0.003 | $p_{mb}$ | 0.05 |

Tabla 2.9: Parámetros de ejecución NSGAII y PAES

### 2.4.4. Generador de Problemas

Como este tipo de problemas de localización multiobjetivo ha sido tratado en muy pocas ocasiones en la literatura, no existe ningún conjunto de problemas de dominio público con los cuales probar el desempeòno de las diferentes técnicas de solución. Se optó entonces por generar aleatoriamente un conjunto de problemas de prueba utilizando las metodologías propuestas en [49, 74, 78] para distintos problemas de localización.

La ubicación de cada clientes se determina aleatoriamente en un cuadrado de lado 190, y su

demanda se genera con una distribución uniforme discreta en el intervalo [10,50].

Los lugares para abrir bodegas se seleccionan según dos criterios: En los problemas tipo A se ubican en el mismo lugar que $m$ clientes seleccionados aleatoriamente y en los problemas tipo B en lugares aleatorios generados en el cuadrado de lado 190.

Para el costo fijo de operación de las bodegas $f_i$, se utilizaron seis opciones diferentes: tres en las que los costos para cada bodega son generados aleatoriamente y tres en las que los costos son iguales para todas las bodegas (Tabla 2.10).

| Opción | Valor |
|---|---|
| C1 | Uniforme Discreta[100,400] |
| C2 | Uniforme Discreta[400,700] |
| C3 | Uniforme Discreta[700,1000] |
| C4 | 400 |
| C5 | 700 |
| C6 | 1000 |

Tabla 2.10: Opciones para generar $f_i$

$h_{ij}$ se calculó con la distancia euclidiana entre la bodega y el cliente[5] $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ donde, $x_i, y_i$ son las coordenadas de la bodega y $x_j, y_j$ las coordenadas del cliente. El costo de asignar un cliente a una bodega se asumió proporcional a la distancia que los separa y a la demanda de cada cliente, la expresión (2.22) se utiliza para calcular $c_{ij}$

$$c_{ij} = (h_{ij} + h_0) \times d_j \times c_{unit} \times FA \tag{2.22}$$

Los componentes del costo de asignación son: cargo fijo de atender un cliente $h_0 = 5$, costo unitario de transporte $c_{unit}$=0.05, por unidad de distancia y por unidad de demanda. $FA$ es un factor aleatorio para perturbar el costo de asignación de cada pareja cliente-bodega, se toma de una distribución uniforme en el intervalo (0,1).

Con todas las opciones disponibles se generaron 36 problemas de diferentes tamaños, todos con distancia máxima de cobertura $D_{max}$ 35. Los detalles de los problemas utilizados se encuentran en la Tabla 2.11

| Número de Problemas | Bodegas | Clientes |
|---|---|---|
| 12 | 10 | 25 |
| 12 | 30 | 75 |
| 12 | 50 | 150 |

Tabla 2.11: Tamaño de los problemas generados

Los problemas generados se identifican con una letra al inicio que puede ser A o B, el número de bodegas disponibles ($m$), un guión y después el número de clientes ($n$) y por último la opción con la cual se genero el costo fijo de las bodegas. Por ejemplo: el problema A10-25C1, fue generado con las bodegas en el mismo lugar que m clientes seleccionados aleatoriamente,

[5] redondeada a cero posiciones decimales

tiene 10 bodegas y 25 clientes, y el costo fijo de las bodegas se genero con una distribución uniforme discreta en el intervalo [100,400].

### 2.4.5. Resultados

Para verificar que los operadores de cruce y mutación si mejoran el proceso de búsqueda se comparó el desempeño de los algoritmos evolutivos en la aproximación de la frontera de Pareto contra el de una búsqueda aleatoria pura. .

La búsqueda aleatoria pura implementada consiste en generar aleatoriamente $l$ cromosomas a la vez y evaluar cuales de ellos son no dominados para guardarlos en un archivo de capacidad ilimitada, cada vez que se generan $l$ cromosomas se comparan entre ellos y con las soluciones del archivo para actualizar el conjunto de soluciones no dominadas. El proceso de generación aleatoria se detiene cuando se han generado 24000 cromosomas. Después de un proceso de afinación rápido, $l$ se fijó en 10

Los dos algoritmos evolutivos y la búsqueda aleatoria se ejecutaron 10 veces en cada problema. Como resultado final se obtiene la aproximación de la frontera de Pareto resultante al unir las soluciones no dominadas de las 10 ejecuciones independientes. En la Tabla 2.12, se presentan el valor de la métrica $S'$ obtenida por cada algoritmo, el tiempo total de ejecución de las 10 corridas independientes (en segundos) y el número de soluciones no dominadas ($|\mathcal{PF}^*|$) de la aproximación de la frontera.

Los resultados de la tabla 2.12 permiten concluir que en los problemas pequeños ($m = 10$, $n = 25$) no existe mucha diferencia entre los diferentes algoritmos utilizados. Incluso, la búsqueda aleatoria pura encuentra buenas aproximaciones de la frontera de Pareto, sin embargo, la búsqueda aleatoria encuentra menos soluciones no dominadas que los Algoritmos evolutivos, por ejemplo, en el problema A10-25C5 la solución que le por encontrar falto a la búsqueda aleatoria, y que los algoritmos evolutivos si encontraron, era precisamente la de mínimo costo.

En problemas más grandes los resultados de la búsqueda aleatoria no son nada buenos, algunas veces (problemas $m = 50$, $n = 150$) no encuentra soluciones que no estén dominadas por las configuraciones de mínimo costo o máxima cobertura. Podría pensarse que el tiempo que se toma la búsqueda aleatoria es un punto a su favor, pero en todos los casos PAES obtiene mejores resultados en un menor tiempo.

Como ejemplo, en la Figura 2.13 se muestran Las fronteras obtenidas con NSGAII, PAES y búsqueda aleatoria para el problema B30-75C1. En este porblema NSGAII es mejor que PAES y PAES a su vez mejor que la búsqueda aleatoria. En la Figura, se ha utilizado el % del costo mínimo y no el costo absoluto para medir $z_1$, ya que cuando se expresan costo y cobertura en términos relativos es más fácil entender el *trade-off* existente.

El menor tiempo de ejecución de PAES se debe al menor número de comparaciones que realiza, ya que no invierte tiempo comparando soluciones poco prometedoras, además siempre compara una solución contra el archivo y no todas las soluciones entre sí como se hace en NSGAII.

En la tabla 2.12 también se pueden comparar los algoritmos evolutivos entre si. Nuevamente, en los problemas pequeños no hay diferencia entre NSGAII y PAES. En los demás problemas NSGAII obtiene mejores valores de $S'$ que PAES. Sin embargo, PAES es más rápido para

| Problema | NSGAII | | | PAES | | | Búsqueda Aleatoria | | |
|---|---|---|---|---|---|---|---|---|---|
| | Tiempo. (seg.) | $S'$ | $\|\mathcal{PF}^*\|$ | Tiempo (seg.) | $S'$ | $\|\mathcal{PF}^*\|$ | Tiempo (seg.) | $S'$ | $\|\mathcal{PF}^*\|$ |
| A10-25C1 | 1747 | 0.7261 | 14 | 440 | 0.7261 | 14 | 790 | 0.7261 | 13 |
| A10-25C2 | 1766 | 0.8117 | 6 | 363 | 0.8117 | 6 | 627 | 0.8117 | 5 |
| A10-25C3 | 1261 | 0.6517 | 17 | 394 | 0.6517 | 17 | 726 | 0.6517 | 17 |
| A10-25C4 | 1439 | 0.7795 | 7 | 397 | 0.7795 | 7 | 615 | 0.7794 | 5 |
| A10-25C5 | 1430 | 0.5856 | 5 | 405 | 0.5856 | 5 | 564 | 0.5856 | 4 |
| A10-25C6 | 1288 | 0.7410 | 13 | 433 | 0.7410 | 13 | 675 | 0.7410 | 12 |
| A30-75C1 | 2562 | 0.5333 | 23 | 502 | 0.5211 | 21 | 990 | 0.2160 | 9 |
| A30-75C2 | 1433 | 0.7631 | 31 | 564 | 0.7360 | 26 | 987 | 0.5500 | 16 |
| A30-75C3 | 1460 | 0.8088 | 47 | 585 | 0.7602 | 32 | 1085 | 0.6404 | 19 |
| A30-75C4 | 1566 | 0.7609 | 30 | 562 | 0.7134 | 27 | 1048 | 0.6231 | 18 |
| A30-75C5 | 1525 | 0.7285 | 49 | 563 | 0.6899 | 34 | 1010 | 0.5750 | 16 |
| A30-75C6 | 1345 | 0.6330 | 51 | 578 | 0.5991 | 53 | 1804 | 1.2684 | 6 |
| A50-150C1 | 2603 | 0.7500 | 10 | 1192 | 0.7253 | 14 | 1889 | 0 | 0 |
| A50-150C2 | 2301 | 0.6387 | 35 | 1212 | 0.5021 | 16 | 1846 | 0 | 0 |
| A50-150C3 | 3178 | 0.8029 | 43 | 1260 | 0.7152 | 32 | 1893 | 0.1427 | 4 |
| A50-150C4 | 2698 | 0.8447 | 23 | 1280 | 0.6925 | 10 | 1839 | 0 | 0 |
| A50-150C5 | 2289 | 0.7515 | 41 | 1241 | 0.4970 | 13 | 1894 | 0.1162 | 7 |
| A50-150C6 | 2284 | 0.7988 | 52 | 1279 | 0.6836 | 26 | 1727 | 0.2256 | 7 |
| B10-25C1 | 3186 | 0.5729 | 5 | 305 | 0.5729 | 5 | 643 | 0.5729 | 4 |
| B10-25C2 | 1748 | 0.6351 | 8 | 378 | 0.6351 | 8 | 638 | 0.6351 | 7 |
| B10-25C3 | 1650 | 0.5982 | 9 | 384 | 0.5982 | 9 | 626 | 0.5982 | 8 |
| B10-25C4 | 1344 | 0.7057 | 13 | 372 | 0.7057 | 13 | 702 | 0.7057 | 12 |
| B10-25C5 | 1511 | 0.6136 | 11 | 394 | 0.6136 | 11 | 763 | 0.6136 | 10 |
| B10-25C6 | 1859 | 0.6993 | 9 | 326 | 0.6993 | 9 | 640 | 0.6993 | 9 |
| B30-75C1 | 1874 | 0.7508 | 20 | 525 | 0.7342 | 20 | 967 | 0.3500 | 6 |
| B30-75C2 | 1413 | 0.7783 | 40 | 595 | 0.7572 | 33 | 1066 | 0.5403 | 20 |
| B30-75C3 | 1474 | 0.6676 | 36 | 511 | 0.5300 | 12 | 923 | 0.4090 | 7 |
| B30-75C4 | 1633 | 0.8071 | 18 | 499 | 0.7795 | 16 | 997 | 0.3949 | 10 |
| B30-75C5 | 1345 | 0.8012 | 43 | 622 | 0.7915 | 41 | 1029 | 0.6285 | 13 |
| B30-75C6 | 1394 | 0.7538 | 43 | 603 | 0.7384 | 29 | 967 | 0.5483 | 14 |
| B50-150C1 | 2575 | 0.6344 | 17 | 1233 | 0.6106 | 15 | 1719 | 0 | 0 |
| B50-150C2 | 2446 | 0.8072 | 40 | 1251 | 0.7590 | 23 | 1851 | 0 | 0 |
| B50-150C3 | 2259 | 0.7862 | 38 | 1218 | 0.6461 | 27 | 1866 | 0.1313 | 1 |
| B50-150C4 | 2385 | 0.7913 | 43 | 1386 | 0.7391 | 25 | 1881 | 0.0050 | 1 |
| B50-150C5 | 2522 | 0.7597 | 39 | 1231 | 0.6729 | 23 | 1746 | 0.0246 | 2 |
| B50-150C6 | 2298 | 0.7665 | 34 | 1269 | 0.6900 | 26 | 1871 | 0.0790 | 3 |

Tabla 2.12: Experimento computacional. Comparacion NSGAII, PAES y ALEATORIO PURO

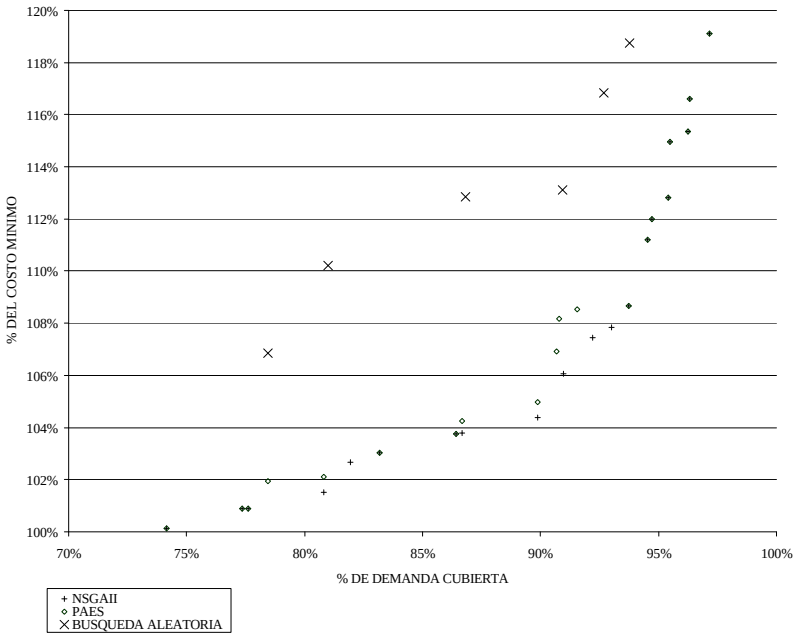


Figura 2.13: Aproximación de la Frontera Eficiente . Problema B30-75C1

explorar las 24,000 soluciones.

Puede pensarse que si se iguala el tiempo de ejecución de los dos algoritmos, PAES sería capaz de encontrar mejores soluciones que NSGAII. Para evaluar esa posibilidad, se efectuó un experimento con el tiempo como criterio de parada. Ambos algoritmos se ejecutaron 10 veces para cada problema. El tiempo fijo de cada ejecución fuer de 180 segundos. Los resultados se resumen en la Tabla 2.13.

Como se observa en la Tabla 2.13, cuando se igualaron los tiempos de ejecución la diferencia entre PAES y NSGAII se disminuye, pero NSGAII sigue siendo mejor para aproximar la frontera de Pareto (i.e., siempre encuentra aproximaciones con igual o mayor espacio dominado).

La razón principal para el desempeño inferior de PAES puede ser su estrategia de búsqueda tan básica. En PAES el comienzo es muy bueno, pero la búsqueda deja de ser eficiente rápidamente. Esta situación se ilustra mejor en la Figura 2.14 donde se muestra el progreso de $S'$ en la ejecución de PAES y NSGAII para un mismo problema.

Al comienzo de PAES (menos de 1000 soluciones generadas) las mejoras en $S'$ son muy grandes, pero de allí en adelante las mejoras son muy pocas aunque se exploren muchas más soluciones. Igualmente, la búsqueda de NSGAII parece estancarse después de las 200 generaciones (6000 soluciones generadas).

| Problema | NSGAII | | | PAES | | |
|---|---|---|---|---|---|---|
| | Tiempo. (seg.) | $S'$ | $\lvert\mathcal{PF}^*\rvert$ | Tiempo (seg.) | $S'$ | $\lvert\mathcal{PF}^*\rvert$ |
| A10-25C1 | 1804 | 0.7261 | 14 | 1801 | 0.7261 | 14 |
| A10-25C2 | 1802 | 0.8117 | 6 | 1804 | 0.8117 | 6 |
| A10-25C3 | 1803 | 0.6517 | 17 | 1801 | 0.6517 | 17 |
| A10-25C4 | 1803 | 0.7795 | 7 | 1801 | 0.7795 | 7 |
| A10-25C5 | 1803 | 0.5856 | 5 | 1801 | 0.5856 | 5 |
| A10-25C6 | 1803 | 0.7410 | 13 | 1801 | 0.7410 | 13 |
| A30-75C1 | 1804 | 0.5312 | 23 | 1803 | 0.5250 | 20 |
| A30-75C2 | 1804 | 0.7638 | 35 | 1803 | 0.7411 | 29 |
| A30-75C3 | 1804 | 0.8089 | 46 | 1803 | 0.7956 | 32 |
| A30-75C4 | 1804 | 0.7624 | 34 | 1803 | 0.7590 | 34 |
| A30-75C5 | 1804 | 0.7326 | 52 | 1803 | 0.7062 | 34 |
| A30-75C6 | 1804 | 0.6331 | 54 | 1803 | 0.5994 | 48 |
| A50-150C1 | 1807 | 0.7500 | 10 | 1807 | 0.7320 | 13 |
| A50-150C2 | 1806 | 0.6382 | 36 | 1807 | 0.5022 | 14 |
| A50-150C3 | 1806 | 0.7568 | 45 | 1807 | 0.7164 | 32 |
| A50-150C4 | 1807 | 0.8277 | 20 | 1807 | 0.7045 | 11 |
| A50-150C5 | 1807 | 0.7412 | 37 | 1807 | 0.6135 | 27 |
| A50-150C6 | 1806 | 0.7915 | 37 | 1807 | 0.6843 | 27 |
| B10-25C1 | 1804 | 0.5728 | 5 | 1801 | 0.5729 | 5 |
| B10-25C2 | 1803 | 0.6351 | 8 | 1801 | 0.6351 | 8 |
| B10-25C3 | 1803 | 0.5981 | 9 | 1801 | 0.5982 | 9 |
| B10-25C4 | 1803 | 0.7057 | 13 | 1801 | 0.7057 | 1 |
| B10-25C5 | 1803 | 0.6136 | 11 | 1801 | 0.6136 | 11 |
| B10-25C6 | 1803 | 0.6993 | 9 | 1801 | 0.6993 | 9 |
| B30-75C1 | 1804 | 0.7508 | 20 | 1803 | 0.7348 | 20 |
| B30-75C2 | 1804 | 0.7786 | 42 | 1803 | 0.7654 | 39 |
| B30-75C3 | 1804 | 0.6677 | 38 | 1803 | 0.6049 | 21 |
| B30-75C4 | 1804 | 0.8071 | 18 | 1803 | 0.7857 | 17 |
| B30-75C5 | 1804 | 0.8013 | 43 | 1803 | 0.7931 | 40 |
| B30-75C6 | 1804 | 0.7551 | 44 | 1803 | 0.7407 | 31 |
| B50-150C1 | 1806 | 0.6296 | 14 | 1807 | 0.6141 | 14 |
| B50-150C2 | 1807 | 0.8047 | 38 | 1807 | 0.7669 | 24 |
| B50-150C3 | 1806 | 0.7740 | 36 | 1807 | 0.6520 | 24 |
| B50-150C4 | 1806 | 0.7866 | 32 | 1807 | 0.7425 | 23 |
| B50-150C5 | 1806 | 0.7427 | 35 | 1807 | 0.6756 | 26 |
| B50-150C6 | 1807 | 0.7639 | 26 | 1807 | 0.6903 | 27 |

Tabla 2.13: Experimento computacional. Comparacion NSGAII, PAES. Tiempo de ejecución constante.

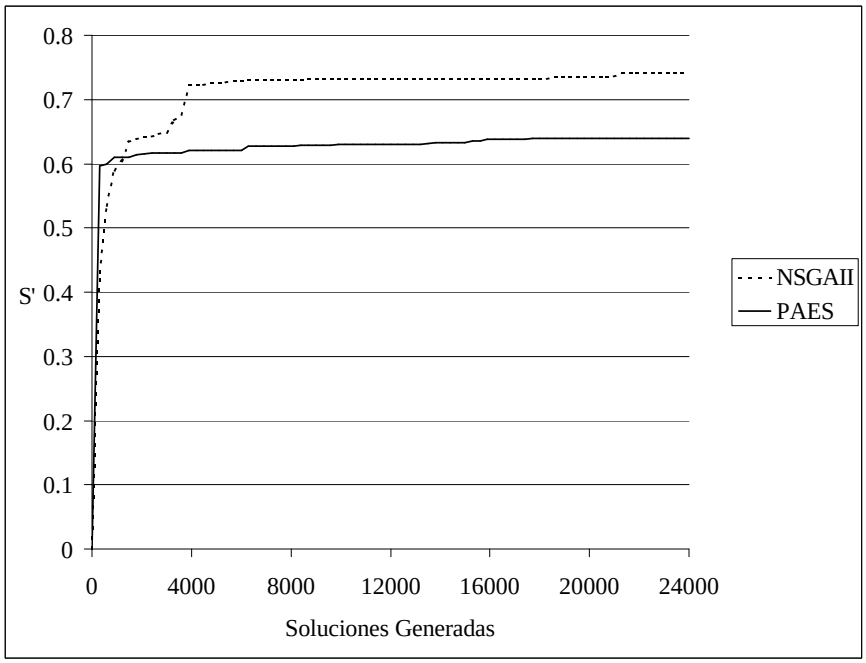


Figura 2.14: Evolución del área dominada. Problema B50-150C6

## 2.5. Conclusiones

En problemas de localización pequeños no importa mucho la estrategia que se utilice, incluso una búsqueda aleatoria permite obtener buenos resultados. Para problemas de localización multiobjetivo (sin restricciones de capacidad), más grandes los algoritmos evolutivos son una alternativa de solución interesante. Su flexibilidad permite abordar problemas en los que los criterios de decisión están en conflicto, como en el caso del costo y la cobertura.

Los dos algoritmos evolutivos implementados (PAES y NSGAII) son superiores a la búsqueda aleatoria pura. PAES permite encontrar soluciones aceptables rápidamente. NSGAII, por su parte, obtiene mejores aproximaciones de la frontera de Pareto, pero con un mayor tiempo de ejecución.

La principal ventaja de PAES: su simplicidad, es también su principal defecto. Como es tan simple, es muy fácil de implementar, y además se ejecuta en poco tiempo. Sin embargo, la búsqueda deja de ser productiva rápidamente. Los operadores de mutación aleatorios no permiten identificar mejores soluciones después de cierto tiempo. Con todo y sus defectos, no debe descalificarse la estrategia evolutiva empleada en PAES, ya que es posible implementar estrategias más sofisticadas con el mismo principio de búsqueda y la utilización de un archivo [54]. Estas mejoras son un camino abierto para futuras investigaciones.

La representación de las soluciones es una característica definitiva en la implementación de los algoritmos evolutivos. En esté caso se comprobó la efectividad de representaciones binarias sobre otras con un mayor mayor número de alternativas. Sin embargo, la representación entera no debería abandonarse del todo, puesto que es la alternativa obvia cuando se consideran restricciones de capacidad. Futuras investigaciones podrían intentar mejorar su desempeño con operadores de cruce y mutación más sofisticados.

Aunque los resultados obtenidos son bastante buenos, los algoritmos evolutivos implementados pueden mejorarse: podrían implementarse operadores de cruce que favorezcan la trasmisión de información de los individuos mejor adaptados (de frentes más cercanos al primero). El criterio de parada es bastante clásico, un criterio de parada que verifique en linea la convergencia del algoritmo hacia el frente de Pareto podría ofrecer mejores resultados.

El comportamiento de los algoritmos evolutivos con respecto al tiempo descubre la necesidad de utilizar operadores de cruce y mutación dinámicos [9]. Por ejemplo, si después de varias iteraciones no se ha obtenido ninguna mejora, un aumento de la probabilidad de mutación o la introducción de nuevos individuos generados aleatoriamente permitiría diversificar la búsqueda.

Una pregunta obvia es la utilización de otras técnicas de solución y su comparación con los algoritmos evolutivos. Ese el tema que se aborda en el siguiente capítulo, esto permitirá ampliar las conclusiones que hasta ahora se han formulado.

# Capítulo 3

# PROGRAMACIÓN MATEMÁTICA PARA LA SOLUCIÓN DE PROBLEMAS DE LOCALIZACIÓN MULTIOBJETIVO

Una alternativa diferente a la utilización de algoritmos evolutivos para aproximar la frontera de Pareto del problema de localización multiobjetivo que se está estudiando, es emplear programación matemática para obtener puntos que pertenezcan a dicha frontera. Concretamente, se formula un problema de localización sin restricciones de capacidad con una sola función objetivo (minimizar el costo o maximizar la cobertura) y se incluye la otra función objetivo como restricción adicional. El compromiso existente entre el costo y la cobertura se explora al observar qué ocurre con la solución óptima cuando se resuelve el problema para diferentes valores de la cota de dicha restricción. Este enfoque ha sido utilizado para abordar otros problemas de localización multiobjetivo en los cuales una de las funciones objetivo es el costo y la otra está relacionada con el servicio prestado a los clientes (minimizar el tiempo de entrega [47] o maximizar la cobertura [71]).

## 3.1. Metodología de Solución

Para generar algunos de los puntos de la frontera de Pareto del problema de localización multiobjetivo se formuló un problema de optimización en el cual se minimiza el costo y se utilizó la cobertura como restricción. Utilizando la notación introducida previamente y modificando ligeramente la formulación presentada en [71], el problema resultante es el siguiente.

### 3.1.1. Problema de Localización con Restricciones de Cobertura (UFLP+C)

$$\text{mín} \quad z_1 = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} c_{ij}x_{ij} + \sum_{i\in\mathcal{I}} f_i y_i \tag{3.1}$$

Sujeto a,

$$\sum_{i\in\mathcal{I}} x_{ij} = 1 \quad , \quad j\in\mathcal{J} \tag{3.2}$$

$$x_{ij} \leq y_i \quad , \quad j\in\mathcal{J}, \quad i\in\mathcal{I} \tag{3.3}$$

$$\sum_{j\in\mathcal{J}} d_j \sum_{i\in\mathcal{Q}_j} x_{ij} \geq v\sum_{j\in\mathcal{J}} d_j \quad , \quad j\in\mathcal{J}, \quad i\in\mathcal{I} \tag{3.4}$$

$$x_{ij} \in \{0,1\} \quad , \quad j\in\mathcal{J}, \quad i\in\mathcal{I} \tag{3.5}$$

$$y_i \in \{0,1\} \quad , \quad i\in\mathcal{I} \tag{3.6}$$

El parámetro $v$ define que porcentaje de la demanda total debe ser atendido dentro del radio máximo de cobertura ($D_{max}$). El compromiso existente entre el costo y la cobertura se explora aumentando sistemáticamente $v$, desde el valor de cobertura ofrecido por la solución de mínimo costo ($v_{min} = \frac{z_2}{\sum_{j\in\mathcal{J}} d_j}$) hasta llegar a la máxima cobertura posible ($v^{max} = \frac{\overline{z}_2}{\sum_{j\in\mathcal{J}} d_j}$).

La función objetivo (3.1) es el costo total de operación, las restricciones (3.2) y (3.3) corresponden a un problema de localización sin restricciones de capacidad y por último se encuentra la restricción adicional de cobertura (3.4).

Resolver esté problema no siempre es suficiente. Para garantizar que la solución obtenida $(z_1^*, \mathbf{y}_i^*, \mathbf{x}_{ij}^*)$ realmente pertenece al conjunto eficiente de Pareto, es necesario que dicha solución sea la que ofrece la mejor cobertura al costo $z_1^*$. Cuando existen óptimos alternos puede existir otra solución $(\mathbf{y}'_i, \mathbf{x}'_{ij})$ de costo $z_1^*$ con una cobertura mayor.

### 3.1.2. Problema de Localización de Máxima Cobertura con Restricciones de Presupuesto(MCLP+B)

La solución que pertenece al conjunto eficiente de Pareto (i.e., la que ofrece la mayor cobertura al costo $z_1^*$), se encuentra resolviendo un segundo problema de optimización, que busca entre los óptimos alternos aquel con mayor cobertura. Este problema maximiza la cobertura garantizando que el costo total es igual al costo obtenido previamente.

$$\text{máx} \sum_{j\in\mathcal{J}} d_j \sum_{i\in\mathcal{Q}_j} x_{ij} \tag{3.7}$$

Sujeto a,

$$\sum_{i\in\mathcal{I}} x_{ij} = 1 \quad , \quad j \in \mathcal{J} \tag{3.8}$$

$$x_{ij} \leq y_i \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{3.9}$$

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} c_{ij}x_{ij} + \sum_{i\in\mathcal{I}} f_i y_i \leq B \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{3.10}$$

$$x_{ij} \in \{0,1\} \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{3.11}$$

$$y_i \in \{0,1\} \quad , \quad i \in \mathcal{I} \tag{3.12}$$

El presupuesto máximo disponible es $B$, su valor es el costo óptimo de la solución obtenida previamente con el problema UFLP+C (i.e.,$B = z_1^*$). La función objetivo (3.7) es la cobertura expresada en términos absolutos. El costo total de operación ha sido incluido como restricción (3.10). Los demás componentes del problema no cambian.

El procedimiento de aproximación de la frontera de Pareto es iterativo. El rango de una de las funciones objetivos se discretiza en $s$ intervalos iguales y se parametriza la restricción correspondiente a la función objetivo cuyo rango se discretizo.

Inicialmente se determinan las soluciones extremas de la frontera eficiente (configuraciones de costo mínimo y cobertura máxima) el procedimiento para hallarlas es igual al explicado en la Sección 2.4.2. Con las cotas $\underline{z}_2$ y $\overline{z}_2$ se determina el incremento ($\Delta v$) que se aplicará a $v$ en cada iteración, $\Delta v = \frac{\overline{z}_2 - \underline{z}_2}{s \cdot \sum\limits_{j\in\mathcal{J}} d_j}$. Conforme crece $v$ la cobertura se hace más exigente para generar puntos eficientes más costosos y con mayor cobertura. Cuando se alcanza el valor máximo ($v^{max} = \frac{\overline{z}_2}{\sum\limits_{j\in\mathcal{J}} d_j}$) se detiene la ejecución del procedimiento.

El procedimiento aquí descrito, se presenta en el Algoritmo 7. Las líneas 16 y 17 sirven para aumentar el valor de $v$, garantizando que no se busca en regiones dominadas por soluciones generadas previamente.

Para generar soluciones mejor distribuidas en la aproximación de la frontera de Pareto, se realiza un procedimiento complementario. Invirtiendo el orden en el cual se resuelven los problemas (primero MCLP+B y después UFLP+C) y parametrizando la cota superior de presupuesto ($B$). El procedimiento es similar al utilizado para aproximar la frontera de Pareto parametrizando la cota de cobertura, los detalles se presentan en el Algoritmo 8.

Primero se obtiene el presupuesto mínimo necesario para alcanzar la máxima cobertura ($\overline{z}_1$), valor desde el cual se empieza a disminuir el presupuesto para que en cada iteración se generen puntos eficientes más baratos pero de menor cobertura. La disminución del presupuesto en cada iteración es $\Delta B = \frac{\overline{z}_1 - \underline{z}_1}{s}$

La generación de nuevas soluciones se detiene cuando el presupuesto es inferior al costo mínimo $\underline{z}_1$, ya que no existe ninguna solución factible que tenga un costo inferior. Las líneas 16 y 17 sirven para disminuir el valor de $B$; cuando el costo de la solución obtenida es inferior al presupuesto máximo de la siguiente iteración no se explora dicha región sino que se disminuye $B$ hasta un valor en el cual se puedan encontrar nuevas soluciones no dominadas.

**Algoritmo 7** PROCEDIMIENTO PARA LA APROXIMACIÓN DE LA FRONTERA DE PARETO PARAMETRIZANDO LA COTA DE COBERTURA, $(v)$

**Input:** Datos del problema: $h_{ij}$, $c_{ij}$, $d_j$ y $f_i$, Distancia máxima de cobertura $D_{max}$, número de intervalos $(s)$, cota máxima de cobertura $(\overline{z}_2)$

**Output:** Aproximación de la frontera de Pareto $(\widehat{\mathcal{PF}}^*)$, Aproximación del conjunto óptimo de Pareto $(\widehat{\mathcal{P}}^*)$

$v^{max} \leftarrow \frac{\overline{z}_2}{\sum_{j\in\mathcal{J}} d_j}$
$\underline{z}_1 \leftarrow$ Resolver(UFLP)
$B \leftarrow \underline{z}_1$
$(\mathbf{x}^*_{ij}, \mathbf{y}^*_i, \underline{z}_2) \leftarrow$ Resolver(MCLP+B)
$\widehat{\mathcal{P}}^* \leftarrow \widehat{\mathcal{P}}^* \cup [\mathbf{x}^*_{ij}, \mathbf{y}^*_i]$
$\widehat{\mathcal{PF}}^* \leftarrow \widehat{\mathcal{PF}}^* \cup [\underline{z}_1, \underline{z}_2]$
$v_{min} \leftarrow \frac{\underline{z}_2}{\sum_{j\in\mathcal{J}} d_j}$
$\Delta v \leftarrow \frac{\overline{z}_2 - \underline{z}_2}{s \cdot \sum_{j\in\mathcal{J}} d_j}$
$v \leftarrow v_{min} + \Delta v$
**while** $v \leq v^{max}$ **do**
  $z^*_1 \leftarrow$ Resolver(UFLP+C)
  $B \leftarrow z^*_1$
  $(\mathbf{x}^*_{ij}, \mathbf{y}^*_i, z^*_2) \leftarrow$ Resolver(MCLP+B)
  $\widehat{\mathcal{P}}^* \leftarrow \widehat{\mathcal{P}}^* \cup [\mathbf{x}^*_{ij}, \mathbf{y}^*_i]$
  $\widehat{\mathcal{PF}}^* \leftarrow \widehat{\mathcal{PF}}^* \cup [z^*_1, z^*_2]$
  $u \leftarrow v_{min} + \Delta v \times \lceil (\frac{z^*_2 - \underline{z}_2}{\Delta v \cdot \sum_{j\in\mathcal{J}} d_j}) \rceil$
  $v \leftarrow$ máx$(v + \Delta v, u)$
**end while**

**Algoritmo 8** PROCEDIMIENTO PARA LA APROXIMACIÓN DE LA FRONTERA DE PARETO PARAMETRIZANDO LA COTA DE PRESUPUESTO ($B$)

**Input:** Datos del problema: $h_{ij}$, $c_{ij}$, $d_j$ y $f_i$, Distancia máxima de cobertura $D_{max}$, número de intervalos ($s$), cota máxima de cobertura ($\overline{z}_2$)

**Output:** Aproximación de la frontera de Pareto ($\widehat{\mathcal{PF}}^*$), Aproximación del conjunto óptimo de Pareto ($\widehat{\mathcal{P}}^*$)

$\underline{z}_1 \leftarrow$ Resolver(UFLP)
$v^{max} \leftarrow \frac{\overline{z}_2}{\sum_{j \in \mathcal{J}} d_j}$
$v \leftarrow v^{max}$
$(\mathbf{x}^*_{ij}, \mathbf{y}^*_i, \overline{z}_1) \leftarrow$ Resolver(UFLP+C)
$\widehat{\mathcal{P}}^* \leftarrow \widehat{\mathcal{P}}^* \cup [\mathbf{x}^*_{ij}, \mathbf{y}^*_i]$
$\widehat{\mathcal{PF}}^* \leftarrow \widehat{\mathcal{PF}}^* \cup [\overline{z}_1, \overline{z}_2]$
$v \leftarrow v^{max}$
$\Delta B \leftarrow \frac{\overline{z}_1 - \underline{z}_1}{s}$
$B \leftarrow \overline{z}_1 - \Delta B$
**while** $B \geq \underline{z}_1$ **do**
    $z^*_2 \leftarrow$ Resolver(MCLP+B)
    $v = \frac{z^*_2}{\sum_{j \in \mathcal{J}} d_j}$
    $(\mathbf{x}^*_{ij}, \mathbf{y}^*_i, z^*_1) \leftarrow$ Resolver(UFLP+C)
    $\widehat{\mathcal{P}}^* \leftarrow \widehat{\mathcal{P}}^* \cup [\mathbf{x}^*_{ij}, \mathbf{y}^*_i]$
    $\widehat{\mathcal{PF}}^* \leftarrow \widehat{\mathcal{PF}}^* \cup [z^*_1, z^*_2]$
    $u \leftarrow \overline{z}_1 - \Delta B \times \lceil (\frac{\overline{z}_1 - z^*_1}{\Delta B}) \rceil$
    $B \leftarrow \text{mín}(P - \Delta P, u)$
**end while**

### 3.1.3. Ejemplo de la Metodología de Generació de la Frontera Eficiente

Con un ejemplo se ilustra la metodología de solución. Los lugares en los cuales se ubican los clientes y las bodegas se muestra en la fígura 3.1. En las tablas 3.2 a 3.3 se encuentra la información de costos, distancias y demandas utilizados para resolver el problema del ejemplo, el costo fijo para todas las bodega es 400. Y la distancia máxima de cobertura ($D_{max}$) es 35.

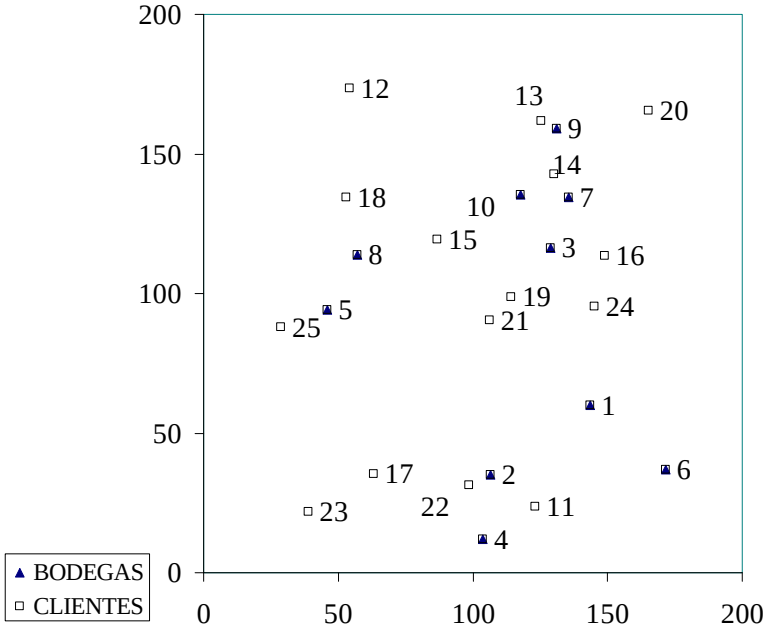


Figura 3.1: Ejemplo de la metodología en Programación Matemática. Ubicación de Bodegas y Clientes.

| Cliente | $d_j$ | Cliente | $d_j$ | Cliente | $d_j$ | Cliente | $d_j$ | Cliente | $d_j$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 11 | 6 | 30 | 11 | 30 | 16 | 10 | 21 | 34 |
| 2 | 45 | 7 | 37 | 12 | 44 | 17 | 34 | 22 | 15 |
| 3 | 43 | 8 | 45 | 13 | 47 | 18 | 21 | 23 | 43 |
| 4 | 15 | 9 | 10 | 14 | 35 | 19 | 13 | 24 | 21 |
| 5 | 23 | 10 | 3 | 15 | 48 | 20 | 24 | 25 | 13 |

Tabla 3.1: Ejemplo de la metodología de Programación Matemática. Demanda de los cliente

| Cliente | Bodega | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | **0** | 44 | 58 | 62 | 103 | 36 | 75 | 101 | 99 | 79 |
| 2 | 44 | **0** | 84 | **23** | 84 | 65 | 103 | 92 | 126 | 100 |
| 3 | 58 | 84 | **0** | 107 | 86 | 89 | **19** | 72 | 42 | **22** |
| 4 | 62 | **23** | 107 | **0** | 100 | 72 | 126 | 112 | 149 | 124 |
| 5 | 103 | 84 | 86 | 100 | **0** | 138 | 98 | **22** | 107 | 82 |
| 6 | 36 | 65 | 89 | 72 | 138 | **0** | 104 | 137 | 128 | 112 |
| 7 | 75 | 103 | **19** | 126 | 98 | 104 | **0** | 81 | **24** | **17** |
| 8 | 101 | 92 | 72 | 112 | **22** | 137 | 81 | **0** | 86 | 64 |
| 9 | 99 | 126 | 42 | 149 | 107 | 128 | **24** | 86 | **0** | **27** |
| 10 | 79 | 100 | **22** | 124 | 82 | 112 | **17** | 64 | **27** | **0** |
| 11 | 41 | **20** | 92 | **22** | 104 | 50 | 111 | 111 | 135 | 111 |
| 12 | 144 | 148 | 94 | 169 | 80 | 180 | 90 | 59 | 78 | 74 |
| 13 | 103 | 128 | 45 | 151 | 104 | 133 | 29 | 83 | **6** | **27** |
| 14 | 83 | 110 | **26** | 133 | 97 | 113 | **9** | 78 | **16** | **14** |
| 15 | 82 | 86 | 42 | 108 | 48 | 118 | 51 | **30** | 59 | **34** |
| 16 | 54 | 89 | **20** | 111 | 105 | 79 | **24** | 92 | 48 | 38 |
| 17 | 83 | 43 | 104 | 46 | 61 | 108 | 122 | 78 | 141 | 113 |
| 18 | 117 | 112 | 78 | 132 | 41 | 153 | 82 | **21** | 81 | 64 |
| 19 | 48 | 64 | **22** | 87 | 68 | 84 | 41 | 59 | 62 | 36 |
| 20 | 107 | 143 | 61 | 165 | 139 | 128 | 42 | 119 | **34** | 56 |
| 21 | 48 | 55 | **34** | 78 | 60 | 84 | 53 | 54 | 73 | 46 |
| 22 | 53 | **8** | 89 | **20** | 81 | 73 | 109 | 92 | 131 | 105 |
| 23 | 111 | 68 | 130 | 65 | 72 | 133 | 148 | 93 | 165 | 138 |
| 24 | **35** | 71 | **26** | 93 | 99 | 64 | 40 | 90 | 65 | 48 |
| 25 | 118 | 93 | 104 | 106 | **18** | 151 | 116 | 38 | 124 | 100 |

Tabla 3.2: Ejemplo de la metodología de Programación Matemática. Distancias entre bodegas y clientes

| Cliente | Bodega | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 2.63 | 25.99 | 34.72 | 40.40 | 56.73 | 22.21 | 46.63 | 62.55 | 59.45 | 46.89 |
| 2 | 117.85 | 10.99 | 208.33 | 61.99 | 199.59 | 145.83 | 225.79 | 227.64 | 267.45 | 237.63 |
| 3 | 143.54 | 176.58 | 11.18 | 245.02 | 186.00 | 221.39 | 53.60 | 153.03 | 98.83 | 57.73 |
| 4 | 49.49 | 22.49 | 91.22 | 3.99 | 73.97 | 54.58 | 92.73 | 82.69 | 118.83 | 104.59 |
| 5 | 130.96 | 108.13 | 109.99 | 117.02 | 5.52 | 149.48 | 114.31 | 30.74 | 132.09 | 105.21 |
| 6 | 67.58 | 109.95 | 131.66 | 122.00 | 199.29 | 6.87 | 160.05 | 229.97 | 194.70 | 190.88 |
| 7 | 154.33 | 184.38 | 48.37 | 237.21 | 184.44 | 219.04 | 9.64 | 174.93 | 50.47 | 41.66 |
| 8 | 254.36 | 237.85 | 172.98 | 288.70 | 66.46 | 300.07 | 179.31 | 11.38 | 222.27 | 158.39 |
| 9 | 50.29 | 65.11 | 23.06 | 75.02 | 51.85 | 71.65 | 15.78 | 47.61 | 2.68 | 16.06 |
| 10 | 145.61 | 175.53 | 49.47 | 235.15 | 158.93 | 218.37 | 42.56 | 123.26 | 61.71 | 8.97 |
| 11 | 72.40 | 37.34 | 149.61 | 40.82 | 158.21 | 87.74 | 171.48 | 161.36 | 206.29 | 175.51 |
| 12 | 306.37 | 334.54 | 197.30 | 386.82 | 190.91 | 443.66 | 191.18 | 129.68 | 198.58 | 177.41 |
| 13 | 240.99 | 301.20 | 110.07 | 395.61 | 260.48 | 332.79 | 73.34 | 204.97 | 24.16 | 74.14 |
| 14 | 166.26 | 201.02 | 51.01 | 237.10 | 169.37 | 208.23 | 25.07 | 140.15 | 40.16 | 30.72 |
| 15 | 209.30 | 206.83 | 110.38 | 253.73 | 131.11 | 288.62 | 121.06 | 90.61 | 147.08 | 102.74 |
| 16 | 28.99 | 50.72 | 11.31 | 53.73 | 57.19 | 44.49 | 15.60 | 51.89 | 27.38 | 20.44 |
| 17 | 139.66 | 76.77 | 192.74 | 94.05 | 110.66 | 198.43 | 208.04 | 139.11 | 231.64 | 211.76 |
| 18 | 127.20 | 125.27 | 85.30 | 137.36 | 48.94 | 181.48 | 86.62 | 24.72 | 93.81 | 71.50 |
| 19 | 31.55 | 48.38 | 18.73 | 65.57 | 46.01 | 60.94 | 29.59 | 38.74 | 41.07 | 25.09 |
| 20 | 125.21 | 169.13 | 85.34 | 214.73 | 185.85 | 173.23 | 60.17 | 139.49 | 49.39 | 71.01 |
| 21 | 96.30 | 98.63 | 70.13 | 131.09 | 107.31 | 158.15 | 105.30 | 96.00 | 130.27 | 86.22 |
| 22 | 47.47 | 10.27 | 68.33 | 18.81 | 66.56 | 60.58 | 90.11 | 67.00 | 101.68 | 84.79 |
| 23 | 239.40 | 171.36 | 294.90 | 159.20 | 172.83 | 279.77 | 350.48 | 199.87 | 366.67 | 337.06 |
| 24 | 43.39 | 81.58 | 29.93 | 92.67 | 99.61 | 66.94 | 43.02 | 94.37 | 67.81 | 51.36 |
| 25 | 72.96 | 68.73 | 64.86 | 75.05 | 13.94 | 107.33 | 76.65 | 27.94 | 91.13 | 63.79 |

Tabla 3.3: Ejemplo de la metodología de Programación Matemática. Costos de Asignación

Para el ejemplo, el número de intervalos en los que se discretizo el rango de cobertura fue $s = 20$. Y la demanda total es 728. Para obtener el valor de $\overline{z}_2$ basta revisar la Tabla 3.2 en la que se han resaltado los valores de distancia menores o iguales a $D_{max}$, con esta información se determinan los clientes que no pueden ser cubiertos: 12, 17 y 23. Considerando la demanda de los demás clientes restantes se obtiene $\overline{z}_2 = 607$.

Completada la información necesaria el proceso de solución sería así:

- Primero con la demanda máxima que puede ser cubierta por las bodegas disponibles $\overline{z}_2 = 607$, se hace $v^{max} = 0.8338$
- Luego se resuelve el UFLP para hallar la cota mínima de costo $\underline{z}_1$. La solución es: costo mínimo $\underline{z}_1 = 2427.60$. Las variables de decisión toman los siguientes valores: bodegas abiertas: $y_2 = y_{10} = 1$. Los clientes asignados a la bodega 2 son: $x_{2,1} = x_{2,2} = x_{2,4} = x_{2,6} = x_{2,11} = x_{2,17} = x_{2,22} = x_{2,23} = 1$, y a la bodega 10: $x_{10,3} = x_{10,5} = x_{10,7} = x_{10,8} = x_{10,9} = x_{10,10} = x_{10,12} = x_{10,13} = x_{10,14} = x_{10,15} = x_{10,16} = x_{10,18} = x_{10,19} = x_{10,20} = x_{10,21} = x_{10,24} = x_{10,25} = 1$.
- Para encontrar la cobertura minima se hace $B$ =2427.60 y se resuelve el MCLP+B. En este caso se obtiene la misma solución del UFLP, cuya cobertura es $\underline{z}_2 = 362$.
- La solución $[\mathbf{x}^*{}_{ij}, \mathbf{y}^*{}_i]$ se guarda en $\widehat{P}^*$ y las funciones objetivo $[\underline{z}_1, \underline{z}_2]$ en $\widehat{\mathcal{PF}}^*$
- Se hace $v_{min} = \frac{362}{728}$=0.49725
- $\Delta v$ =0.01683= $\frac{607-367}{20 \cdot 728}$
- El valor de $v$ en la primera iteración es: $v_{min} + \Delta v$ =0.49725+0.01683=0.51408
- Con $v$ =0.51408 se resuelve el UFLP+C y se obtiene una solución de costo $z_1 = 2444.58$
- Se hace $B = z_1$ =2444.58 y se resuelve el MCLP+B para obtener: cobertura $z_2 = 461$, bodegas abiertas $y_2 = y_7 = y_8 = 1$. A la bodega 2 se asignan los clientes: $x_{2,1} = x_{2,2} = x_{2,4} = x_{2,6} = x_{2,11} = x_{2,17} = x_{2,22} = x_{2,23} = 1$, a la bodega 7: $x_{7,3} = x_{7,7} = x_{7,9} = 1, x_{7,10} = x_{7,13} = x_{7,14} = x_{7,16} = x_{7,19} = x_{7,20} = x_{7,24} = 1$ y a la bodega 8: $x_{8,5} = x_{8,8} = x_{10,12} = x_{10,15} = x_{10,18} = x_{10,21} = x_{10,25} = 1$
- La solución $[\mathbf{x}^*{}_{ij}, \mathbf{y}^*{}_i]$ se guarda en $\widehat{P}^*$ y las funciones objetivo $[z_1^*, z_2^*]$ en $\widehat{\mathcal{PF}}^*$
- Se calcula $u$ con la formula correspondiente y se obtiene $u$ =0.64872. Como $u$ es mayor que el siguiente intervalo de discretización $v + \Delta v$ =0.53091, se hace $v = u$ para no explorar en regiones dominadas por la solución encontrada previamente.
- Después de encontrar una solución no dominada, el ciclo de solución (UFLP+C y luego MCLP+B) se repite para un nuevo valor de $v$.
- El valor de $v$ sigue aumentando y se encuentran otras cuatro soluciones no dominadas, más costosas, pero de mejor cobertura.
- En la ultima iteración, el valor de $v$ es $v^{max}$. Así se encuentran los valores de $\mathbf{x}^*{}_{ij}, \mathbf{y}^*{}_i$ correspondientes a la solución de máxima cobertura que tiene costo $\overline{z}_1^*$ y cobertura $\overline{z}_2^*$.

En todos los problemas, la frontera de Pareto también se aproxima discretizando el rango del costo. Para ello se utiliza en procedimiento descrito en el Algoritmo 8 que es muy similar, al ilustrado en el ejemplo. El resultado del procedimiento de aproximación de la frontera eficiente para este ejemplo se resume en la Tabla 3.4 y en la Figura 3.2 se ilustra la aproximación de la frontera eficiente.

| Solución | Costo[1] (%) | Cobertura (%) | Solución obtenida parametrizando costo | Solución obtenida Parametrizando cobertura |
|---|---|---|---|---|
| 1 | 100 | 49.73 | X | X |
| 2 | 101 | 63.32 | X | |
| 3 | 103 | 64.84 | | X |
| 4 | 113 | 75.96 | X | X |
| 5 | 125 | 80.08 | X | X |
| 6 | 140 | 81.87 | X | X |
| 7 | 155 | 83.38 | X | X |

Tabla 3.4: Ejemplo de la metodología de programación matemática. Soluciones generadas.

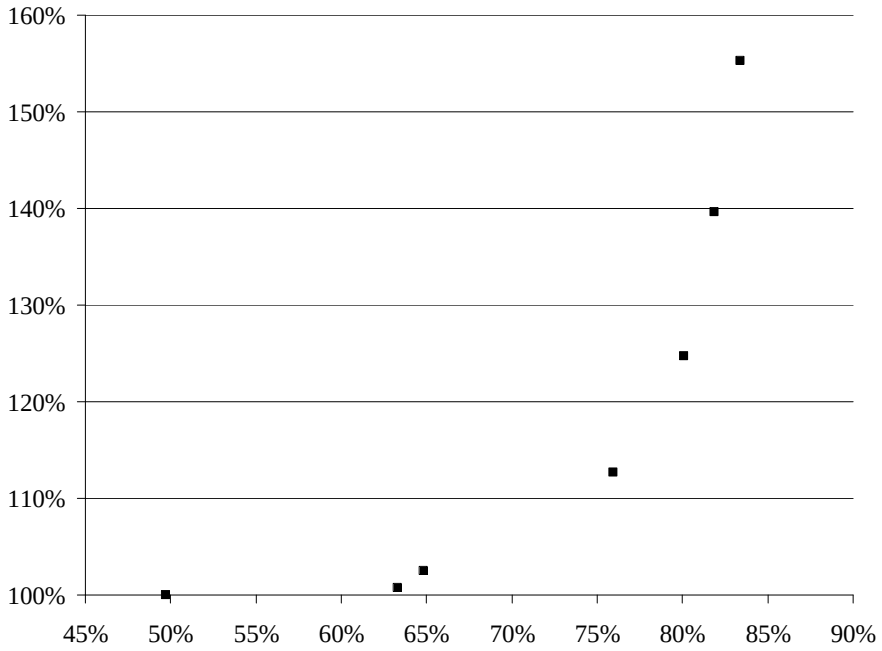


Figura 3.2: Ejemplo de la metodología de programación matemática. Aproximación de la frontera.

## 3.2. Experimentos Computacionales

Para evaluar que tan buena es la metodología propuesta se probó su desempeño en 36 problemas diferentes[1]. Para cada problema se generó la frontera dos veces, una parametrizando la restricción de cobertura y otra parametrizando la restricción de presupuesto. La aproximación de la frontera eficiente es la resultante de unir las aproximaciones obtenidas con los dos procedimientos.

Todos los problemas del experimento se resolvieron con AMPL/Xpress-MP en el servidor de optimización NEOS[2].

A manera de ilustración de la magnitud de los problemas que se resuelven la Tabla 3.5 muestra el número de variables y de restricciones (en la formulación de UFLP+C o MCLP+B) de los diferentes problemas generados.

| Tamaño | Bodegas | Clientes | Variables | Restricciones |
|---|---|---|---|---|
| 10-25 | 10 | 25 | 260 | 275+1 |
| 30-75 | 30 | 75 | 2280 | 2325+1 |
| 50-150 | 50 | 150 | 7550 | 7650+1 |

Tabla 3.5: Problemas generados. Tamaño de la formulación.

### 3.2.1. Número Máximo de Soluciones Generadas

Inicialmente, se probó que tan conveniente podría ser la búsqueda de un mayor número de soluciones pertenecientes a la frontera eficiente. Este número se controla discretizando los rangos de costo $(\overline{z}_1 - \underline{z}_1)$ y de cobertura $(\overline{z}_2 - \underline{z}_2)$ en intervalos más pequeños, que se obtienen haciendo $s$ más grande. En el mejor de los casos se pueden obtener $2s$ soluciones, esto ocurre cuando en cada iteración de ambos procedimientos se obtiene una solución no dominada y además no hay soluciones que se generen con ambos procedimientos. Para hacerse a la idea de lo que ocurre, se tomó como ejemplo el problema B30-75C5. Y se probaron cuatro valores diferentes de $s$: 10, 20, 30 y 40. Los resultados se presentan en la Tabla 3.6 y la Figura 3.3

| | | Número de soluciones | | | | |
|---|---|---|---|---|---|---|
| $s$ | $S'$ | Máximo | Parametrizando $v$ | Parametrizando $B$ | Ambos | Total |
| 10 | 72.49 % | 20 | 6 (33 %) | 9 (50 %) | 3 (17 %) | 18(100 %) |
| 20 | 73.05 % | 40 | 14 (50 %) | 11 (39 %) | 3 (11 %) | 28(100 %) |
| 30 | 73.16 % | 60 | 21 (62 %) | 11 (32 %) | 2 ( 6 %) | 34(100 %) |
| 40 | 73.25 % | 80 | 25 (60 %) | 14 (33 %) | 3 ( 7 %) | 42(100 %) |

Tabla 3.6: Ejemplo del número máximo de soluciones buscadas.

Los valores de $S'$ son muy similares y la fronteras generadas están casi superpuestas. Es claro que ni la calidad ni la forma de la frontera cambian sustancialmente. Buscar demasiadas soluciones tiene además algunos inconvenientes: el tiempo de generación de la frontera puede

[1] Los mismos que se utilizaron para probar los algoritmos evolutivos

[2] Disponible en http://www-neos.mcs.anl.gov/neos/solvers/MILP:XPRESS-AMPL/solver-www.html

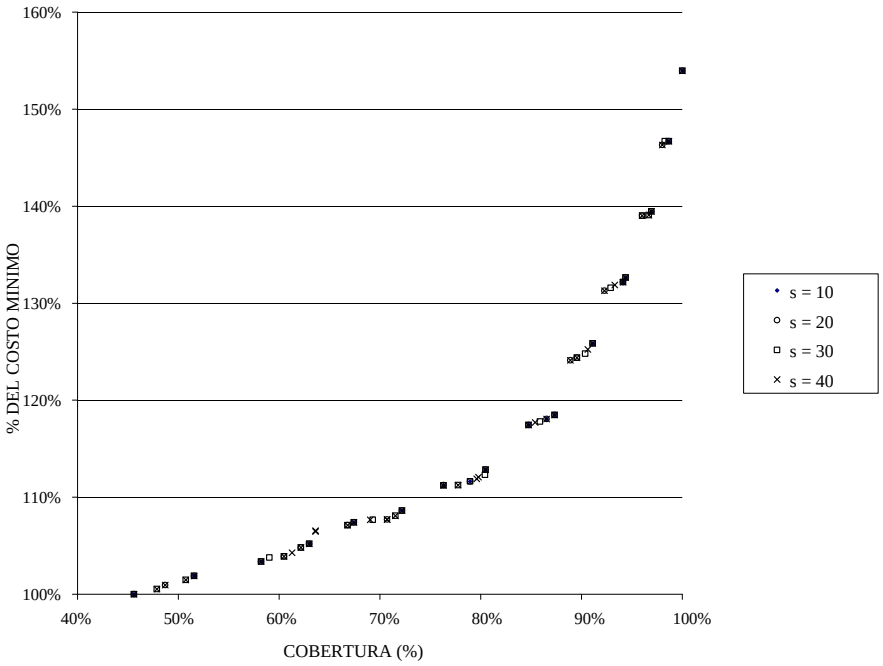


Figura 3.3: Ejemplo del número máximo de soluciones buscadas.

llegar a ser muy grande. Es posible que existan muchas soluciones que no aportan nueva información porque sus funciones objetivo son muy similares y además no tienen grandes diferencias en la configuración de la red. El número de soluciones generadas no crece en la misma proporción que el número máximo de soluciones posibles.

En última instancia muchas de las soluciones no son consideradas y tener demasiadas complica el análisis. Con base en los resultados de este experimento se escogió 20 como un buen valor para $s$. De ahí que, el número máximo de soluciones que se pueden generar en los siguientes experimentos es 40 como .

### 3.2.2. Resultados

En los 36 problemas generados se utilizó una distancia máxima de cobertura de 35. Los resultados obtenidos se resumen en las Tablas 3.7 y 3.8.

Los resultados (figura 3.4) reafirman lo importante que es generar la frontera parametrizaciones tanto $v$ como $B$. Se observa la complementariedad de los procedimientos: del total de soluciones generadas el 38 % se generó en ambos casos, parametrizando la cobertura ($v$) se generó el 31 % y parametrizando el presupuesto ($B$) el otro 31 %. Si únicamente se hubiese utilizado uno de los procedimientos se habría obtenido el 69 % de las soluciones.

Los procedimientos de búsqueda son complementarios no solo en el número de soluciones generadas sino también en las regiones exploradas, es decir, un procedimiento encuentra

| | | Número de Soluciones | | | |
|---|---|---|---|---|---|
| Problema | $S'$ | Parametrizando $v$ | Parametrizando $B$ | Ambos | Total) |
| A10-25C1 | 0.7259 | - | 4 (31 % ) | 9 (69 % ) | 13(100 %) |
| A10-25C2 | 0.8117 | - | - | 6 (100 % ) | 6 (100 %) |
| A10-25C3 | 0.6516 | 4 (25 %) | 2 (13 % ) | 10 (63 % ) | 16(100 %) |
| A10-25C4 | 0.7795 | 1 (14 %) | 1 (14 % ) | 5 (71 % ) | 7 (100 %) |
| A10-25C5 | 0.5856 | - | - | 5 (100 % ) | 5 (100 %) |
| A10-25C6 | 0.7410 | 3 (23 %) | 4 (31 % ) | 6 (46 % ) | 13(100 %) |
| A30-75C1 | 0.5320 | 2 (11 %) | 3 (16 % ) | 14 (74 % ) | 19(100 %) |
| A30-75C2 | 0.7604 | 8 (32 %) | 8 (32 % ) | 9 (36 % ) | 25(100 %) |
| A30-75C3 | 0.8063 | 11 (42 %) | 7 (27 % ) | 8 (31 % ) | 26(100 %) |
| A30-75C4 | 0.7639 | 9 (36 %) | 7 (28 % ) | 9 (36 % ) | 25(100 %) |
| A30-75C5 | 0.7305 | 14 (50 %) | 10 (36 % ) | 4 (14 % ) | 28(100 %) |
| A30-75C6 | 0.6306 | 7 (27 %) | 6 (23 % ) | 13 (50 % ) | 26(100 %) |
| A50-150C1 | 0.7500 | 1 (10 %) | - | 9 (90 % ) | 10(100 %) |
| A50-150C2 | 0.6381 | 7 (27 %) | 10 (38 % ) | 9 (35 % ) | 26(100 %) |
| A50-150C3 | 0.8015 | 7 (25 %) | 13 (46 % ) | 8 (29 % ) | 28(100 %) |
| A50-150C4 | 0.8445 | 11 (48 %) | 9 (39 % ) | 3 (13 % ) | 23(100 %) |
| A50-150C5 | 0.7605 | 13 (45 %) | 10 (34 % ) | 6 (21 % ) | 29(100 %) |
| A50-150C6 | 0.7971 | 10 (42 %) | 8 (33 % ) | 6 (25 % ) | 24(100 %) |

Tabla 3.7: Problemas Tipo A. Aproximación de la Frontera con programacíon matemática.

| | | Número de Soluciones | | | |
|---|---|---|---|---|---|
| Problema | $S'$ | Parametrizando $v$ | Parametrizando $B$ | Ambos | Total) |
| B10-25C1 | 0.5729 | - | - | 5 (100 % ) | 5 (100 %) |
| B10-25C2 | 0.6351 | - | 1 (13 % ) | 7 (88 % ) | 8 (100 %) |
| B10-25C3 | 0.5982 | 2 (22 %) | - | 7 (78 % ) | 9 (100 %) |
| B10-25C4 | 0.7057 | 3 (23 %) | 1 (8 % ) | 9 (69 % ) | 13 (100 %) |
| B10-25C5 | 0.6136 | 3 (27 %) | 1 (9 % ) | 7 (64 % ) | 11 (100 %) |
| B10-25C6 | 0.6993 | 2 (22 %) | 1 (11 % ) | 6 (67 % ) | 9 (100 %) |
| B30-75C1 | 0.7514 | 6 (26 %) | 8 (35 % ) | 9 (39 % ) | 23 (100 %) |
| B30-75C2 | 0.7757 | 8 (29 %) | 11 (39 % ) | 9 (32 % ) | 28 (100 %) |
| B30-75C3 | 0.6680 | 7 (26 %) | 10 (37 % ) | 10(37 % ) | 27 (100 %) |
| B30-75C4 | 0.8069 | 7 (44 %) | 3 (19 % ) | 6 (38 % ) | 16 (100 %) |
| B30-75C5 | 0.7999 | 11 (44 %) | 8 (32 % ) | 6 (24 % ) | 25 (100 %) |
| B30-75C6 | 0.7530 | 7 (33 %) | 8 (38 % ) | 6 (29 % ) | 21 (100 %) |
| B50-150C1 | 0.6350 | 3 (18 %) | 4 (24 % ) | 10(59 % ) | 17 (100 %) |
| B50-150C2 | 0.8067 | 12 (39 %) | 12 (39 % ) | 7 (23 % ) | 31 (100 %) |
| B50-150C3 | 0.7948 | 10 (33 %) | 10 (33 % ) | 10(33 % ) | 30 (100 %) |
| B50-150C4 | 0.7922 | 13 (45 %) | 12 (41 % ) | 4 (14 % ) | 29 (100 %) |
| B50-150C5 | 0.7754 | 7 (29 %) | 9 (38 % ) | 8 (33 % ) | 24 (100 %) |
| B50-150C6 | 0.7704 | 12 (41 %) | 15 (52 % ) | 2 (7 % ) | 29 (100 %) |

Tabla 3.8: Problemas Tipo B. Aproximación de la Frontera con programacíon matemática.

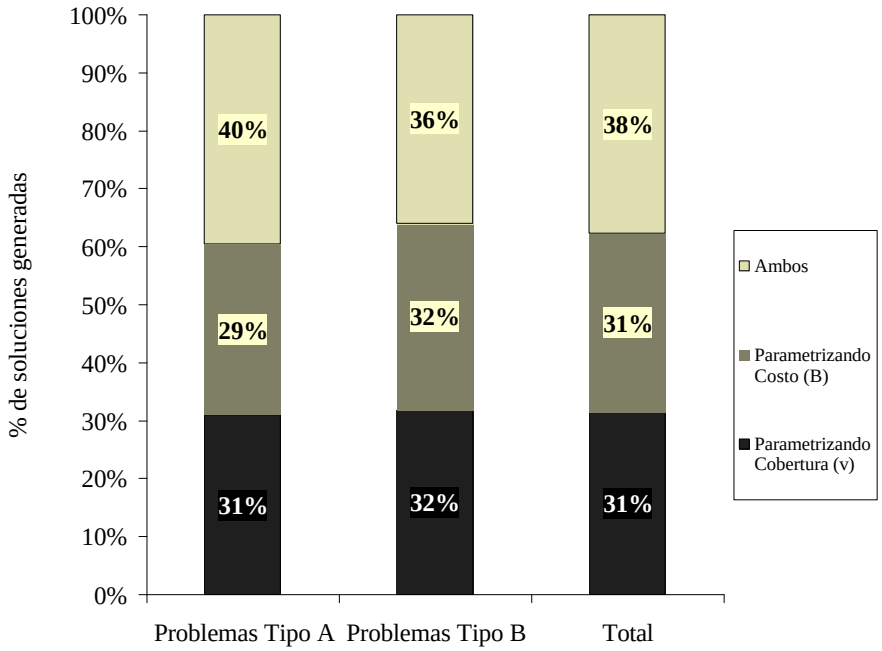


Figura 3.4: Porcentaje de Soluciones generado con cada procedimiento.

soluciones distribuidas en una región de la frontera y el otro en una región diferente. A manera de ejemplo, la Figura 3.5 muestra un problema en el cual la región de baja cobertura fue explorada parametrizando $v$ y la región cercana a la máxima cobertura parametrizando $B$ .

Ejecutar búsquedas parametrizando todas las funciones objetivo tiene algunos inconvenientes: La necesidad de garantizar que las soluciones son eficientes hace que cuando se estén optimizando más de dos funciones objetivo el procedimiento puede llegar a ser bastante costoso computacionalmente. Para garantizar que una solución es eficiente se debe resolver un problema de optimización por cada función objetivo; y considerando todos los ordenes posibles se tendría un total de $k!$ corridas, donde $k$ es el número de funciones objetivo.

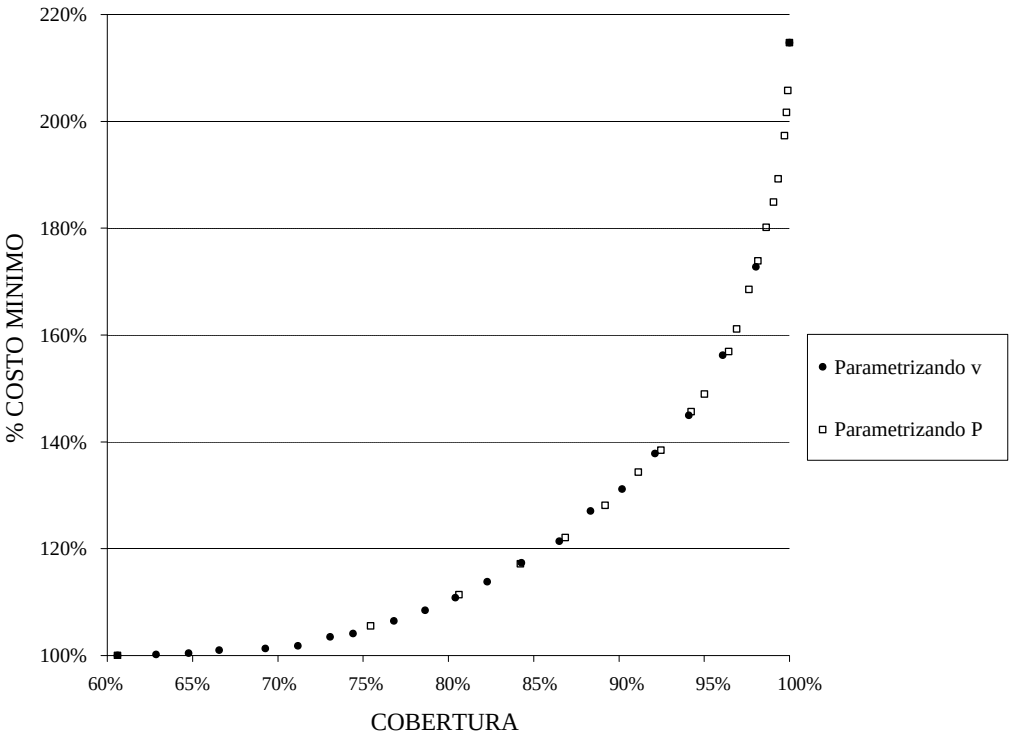


Figura 3.5: Complementariedad de los procedimientos.

### 3.2.3. Comparación entre Programación Matemática y Algoritmos Evolutivos

En el Capítulo 2 se utilizaron los algoritmos evolutivos para aproximar la frontera eficiente del problema de localización multiobjetivo. Aquí se comparan los resultados obtenidos con Programación Matemática y NSGAII (el mejor de los algoritmos evolutivos implementados). Las Tablas 3.9 y 3.10 se resume el desempeño de ambas estrategias en el conjunto de problemas generados para el experimento computacional.

| Problema | Programación Matemática | | NSGAII | | Diferencia $\frac{(2)-(1)}{(1)}$ |
|---|---|---|---|---|---|
| | $S'$ (1) | $\lvert\widehat{\mathcal{PF}}^*\rvert$ | $S'(2)$ | $\lvert\widehat{\mathcal{PF}}^*\rvert$ | |
| A10-25C1 | 0.7259 | 13 | 0.7261 | 14 | 0.03 % |
| A10-25C2 | 0.8117 | 6 | 0.8117 | 6 | 0.00 % |
| A10-25C3 | 0.6516 | 16 | 0.6517 | 17 | 0.02 % |
| A10-25C4 | 0.7795 | 7 | 0.7795 | 7 | 0.00 % |
| A10-25C5 | 0.5856 | 5 | 0.5856 | 5 | 0.00 % |
| A10-25C6 | 0.7410 | 13 | 0.7410 | 13 | 0.00 % |
| A30-75C1 | 0.5320 | 19 | 0.5333 | 23 | 0.24 % |
| A30-75C2 | 0.7604 | 25 | 0.7638 | 31 | 0.45 % |
| A30-75C3 | 0.8063 | 26 | 0.8089 | 47 | 0.32 % |
| A30-75C4 | 0.7639 | 25 | 0.7624 | 30 | -0.20 % |
| A30-75C5 | 0.7305 | 28 | 0.7326 | 49 | 0.29 % |
| A30-75C6 | 0.6306 | 26 | 0.6331 | 51 | 0.40 % |
| A50-150C1 | 0.7500 | 10 | 0.7500 | 10 | 0.00 % |
| A50-150C2 | 0.6381 | 26 | 0.6387 | 35 | 0.09 % |
| A50-150C3 | 0.8015 | 28 | 0.8029 | 43 | 0.17 % |
| A50-150C4 | 0.8445 | 23 | 0.8447 | 23 | 0.02 % |
| A50-150C5 | 0.7605 | 29 | 0.7515 | 41 | -1.18 % |
| A50-150C6 | 0.7971 | 24 | 0.7988 | 52 | 0.21 % |

Tabla 3.9: Comparación Programacíon Matemática y NSGAII. Problemas Tipo A.

En 27 de los problemas NSGAII encuentra soluciones mejores o iguales a las obtenidas con el procedimiento de programación matemática. La desviación entre los valores de $S'$ de ambas aproximaciones es muy baja. Y cuando NSGAII es superado por programación matemática la diferencia máxima es de solo el 2.02 %.

Para ilustrar gráficamente la similitud de las fronteras generadas con ambos procedimientos. En la Figura 3.6 se muestran las aproximaciones de la frontera eficiente encontradas por NSGAII y programación matemática para el problema B30-75C1. Allí, se observa como las dos fronteras están casi superpuestas, la diferencia radica en que el procedimiento de programación matemática encontró más soluciones.

Aunque aquí no se reportan los tiempos de ejecución de los procedimientos de programación matemática [3] debido a que provienen de diferentes servidores a los que no se les conoce la configuración. Para los problemas más grandes (50-150) el tiempo promedio del procedimiento

[3] Vér el Apéndice D para los detalles sobre los tiempos de ejecución

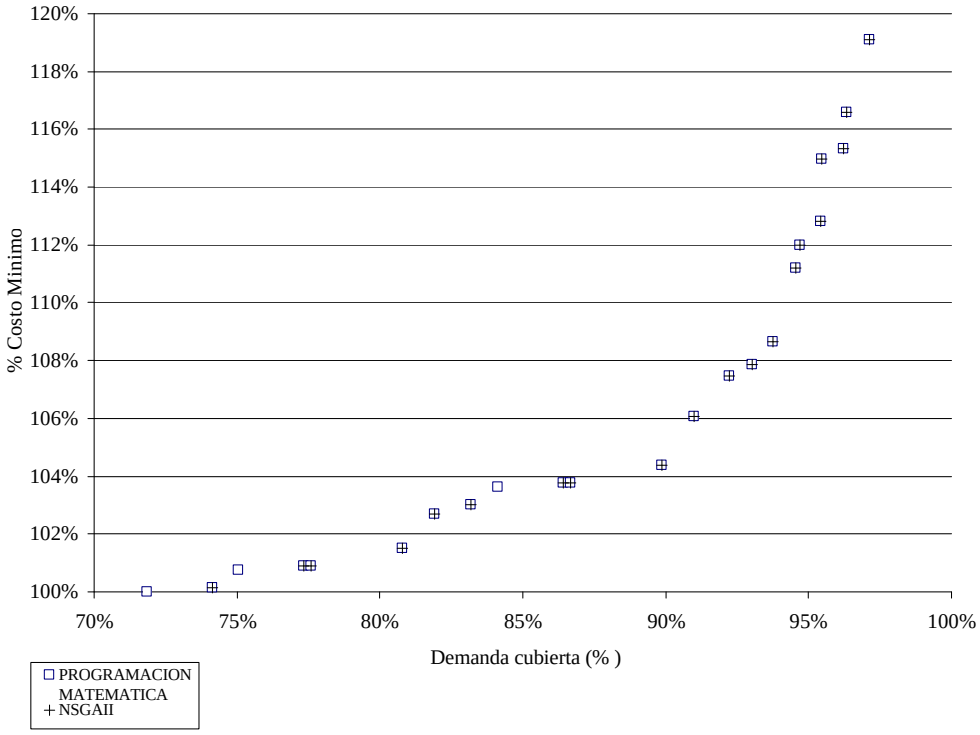


Figura 3.6: Comparación NSGAII y programación matemática.

| Problema | Programación Matemática | | NSGAII | | Diferencia $\frac{(2)-(1)}{(1)}$ |
|---|---|---|---|---|---|
| | $S'$ (1) | $\lvert\widehat{\mathcal{PF}}^*\rvert$ | $S'(2)$ | $\lvert\widehat{\mathcal{PF}}^*\rvert$ | |
| B10-25C1 | 0.5729 | 5 | 0.5729 | 5 | 0.00 % |
| B10-25C2 | 0.6351 | 8 | 0.6351 | 8 | 0.00 % |
| B10-25C3 | 0.5982 | 9 | 0.5982 | 9 | 0.00 % |
| B10-25C4 | 0.7057 | 13 | 0.7057 | 13 | 0.00 % |
| B10-25C5 | 0.6136 | 11 | 0.6136 | 11 | 0.00 % |
| B10-25C6 | 0.6993 | 9 | 0.6993 | 9 | 0.00 % |
| B30-75C1 | 0.7514 | 23 | 0.7508 | 20 | -0.08 % |
| B30-75C2 | 0.7757 | 28 | 0.7786 | 40 | 0.37 % |
| B30-75C3 | 0.6680 | 27 | 0.6677 | 36 | -0.04 % |
| B30-75C4 | 0.8069 | 16 | 0.8071 | 18 | 0.02 % |
| B30-75C5 | 0.7999 | 25 | 0.8014 | 43 | 0.19 % |
| B30-75C6 | 0.7530 | 21 | 0.7551 | 43 | 0.28 % |
| B50-150C1 | 0.6350 | 17 | 0.6344 | 17 | -0.09 % |
| B50-150C2 | 0.8067 | 31 | 0.8072 | 40 | 0.06 % |
| B50-150C3 | 0.7948 | 30 | 0.7862 | 38 | -1.08 % |
| B50-150C4 | 0.7922 | 29 | 0.7913 | 43 | -0.11 % |
| B50-150C5 | 0.7754 | 24 | 0.7597 | 39 | -2.02 % |
| B50-150C6 | 0.7704 | 29 | 0.7665 | 34 | -0.51 % |

Tabla 3.10: Comparación Programacíon Matemática y NSGAII. Problemas Tipo B.

de programación matemática es de casi 2 horas, mientras que las 10 ejecuciones de NSGAII en cada problema se toman en promedio poco más de 41 minutos (2487 segundos).

## 3.3. Conclusiones

Existen varias ventajas asociadas con la utilización de la programación matemática para la aproximación de la frontera eficiente del problema de localización estudiado. La principal es es la certeza que se tiene de que las soluciones obtenidas pertenecen a la frontera eficiente. Otra ventaja clara es la posibilidad de hacer más exhaustiva la búsqueda en algunas regiones de interés, haciendo discretizaciones más finas y limitando los valores de $v$ y $B$ en los rangos deseados, sin la necesidad de explorar toda la frontera eficiente.

Sin embargo, el procedimiento también tiene sus inconvenientes: no es posible controlar la distribución ni la separación de las soluciones, y por consiguiente es posible que se generen soluciones muy similares que no aportan nuevos elementos. No deja de ser una aproximación, ya que en muy pocos casos puede garantizarse que se han generado todas las soluciones pertenecientes a la frontera.

Con sus ventajas e inconvenientes la programación matemática es una alternativa valida e interesante para aproximar la frontera eficiente del problema de localización multiobjetivo (costo - cobertura) considerado. Los resultados obtenidos hacen pensar que en problemas con más funciones objetivos su aplicación puede ser inconveniente. Métodos más interactivos que permitan explorar exhaustivamente algunas regiones de mayor interés o que permitan controlar la distribución de las soluciones a lo largo de la frontera son preguntas abiertas que se deben considerar en futuras investigaciones.

Al comparar el procedimiento basado en programación matemática con los algoritmos evolutivos cada uno ofrece ciertas características superiores al otro. Por ejemplo, el procedimiento de programación matemática garantiza que todas las soluciones que se encuentran pertenecen a la frontera de Pareto, permite hacer exploraciones más exhaustivas en ciertas regiones sin la necesidad de recurrir a mayores modificaciones y cuando la frontera tiene pocas soluciones no desperdicia tiempo buscando en regiones dominadas. Caso contrario al algoritmo evolutivo, que no permite certificar que las soluciones encontradas pertenezcan a la frontera eficiente, es mucho más difícil modificarlo para hacer búsquedas exhaustivas en alguna región; y aunque la frontera eficiente tenga pocas soluciones y el algoritmo evolutivo ya las haya encontrado el proceso de búsqueda sigue aunque no existan más regiones interesantes donde explorar.

Sin embargo, los algoritmos evolutivos también ofrecen sus ventajas: encuentran soluciones tan buenas e incluso mejores que las del método aproximado basado en programación matemática. El tiempo de ejecución es comparable, y en muchos casos menor, que el utilizado por el procedimiento de programación matemática. Y por último, pero no menos importante: su flexibilidad hace que nuevos criterios de decisión u otros problemas con la misma representación puedan abordarse fácilmente.

# Capítulo 4

# PROBLEMAS DE LOCALIZACIÓN MULTIOBJETIVO CON RESTRICCIONES DE CAPACIDAD

Existen múltiples razones para considerar que la demanda total que se puede asignar a una instalación es limitada. Muchas de las instalaciones que se busca localizar tienen limitaciones técnicas sobre la demanda máxima que pueden atender, por ejemplo: hospitales, plantas de producción, bodegas y escuelas [30]. Al localizar bodegas es muy común que se tengan restricciones de capacidad y que en los costos no incluya ningún costo de ampliación de la capacidad. Para abordar está situación se deben incluir restricciones de capacidad de las bodegas al formular el problema de localización.

Los problemas de localización con capacidad en los que clientes solo pueden asignarse a una bodega se conocen como SSCPLP (en inglés, Single Source Capacitated Plant Location Problem). Se dice que son mucho más difíciles que los problemas de localización sin restricciones de capacidad, y que los problemas de localización con restricciones de capacidad que no tienen limitaciones sobre el número de instalaciones que pueden atender a un cliente [38].

Para el problema con una sola función objetivo se han propuesto múltiples técnicas de solución: *branch and price* [39], *branch and bound* [50], algoritmos metaheurísticos [25, 36, 38] y muchos heurísticos basados en relajación lagrangiana [1, 10, 48, 50]. En [52] se abordó el problema multiobjetivo con tres criterios de decision: costo fijo, costo variable y distancia promedio a los clientes.

## 4.1. Modelo del Problema de Localización Multiobjetivo con Restricciones de Capacidad

En la sección 1.4 se presentó una formulación del problema de localización sin restricciones de capacidad. La extensión de dicho modelo para incluir capacidad exige considerar dichas restricciones en la formulación e incluir un nuevo parámetro para cada bodega, su capacidad $w_i$. Los demás componentes del modelo permanecen igual.

Sean $\mathcal{I} = \{1, ..., m\}$ el conjunto de lugares donde se pueden localizar las bodegas, $f_i$ el costo fijo y $w_i$ la capacidad máxima de la bodega que se puede operar en el lugar $i$, $\mathcal{J} = \{1, ..., n\}$ el conjunto de clientes, $d_j$ la demanda del cliente $j$, $c_{ij}$ el costo de atender toda la demanda del cliente $j$ desde la bodega $i$, $h_{ij}$ la distancia entre la bodega $i$ y el cliente $j$. Distancia máxima de cobertura $D_{max}$ , $\mathcal{Q}_j$ el conjunto de bodegas que pueden atender la demanda del cliente $j$ cumpliendo con la distancia máxima de cobertura.

Retomando los componentes del problema de localización multiobjetivo ((1.3) – (1.8)) . La formulación del problema de localización multiobjetivo con restricciones de capacidad es la siguiente

$$\text{mín} \quad z_1 = \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} c_{ij}x_{ij} + \sum_{i\in\mathcal{I}} f_i y_i \tag{4.1}$$

$$\text{máx} \quad z_2 = \sum_{j\in\mathcal{J}} d_j \sum_{i\in\mathcal{Q}_j} x_{ij} \tag{4.2}$$

Sujeto a:

$$\sum_{i\in\mathcal{I}} x_{ij} = 1 \quad , \quad j \in \mathcal{J} \tag{4.3}$$

$$\sum_{j\in\mathcal{J}} d_j x_{ij} \leq w_i y_i \quad , \quad i \in \mathcal{I} \tag{4.4}$$

$$x_{ij} \leq y_i \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{4.5}$$

$$x_{ij} \in \{0,1\} \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{4.6}$$

$$y_i \in \{0,1\} \quad , \quad i \in \mathcal{I} \tag{4.7}$$

Las restricciones (4.4) hacen que la demanda total asignada a cada bodegas abierta sea inferior a su capacidad. Las restricciones 4.5 son redundantes, sin embargo la formulación con estas restricciones es mejor [50]. Las demás restricciones siguen siendo las de un problema de localización convencional.

### 4.1.1. Procedimiento de Solución

Para aproximar la frontera eficiente de este problema se utilizó un procedimiento basado en programación matemática, similar al empleado para el problema de localización multiobjetivo sin restricciones de capacidad.

La aproximación de la frontera se genera en dos etapas, en la primera se parametriza el costo y en la segunda la cobertura. Cuando se está parametrizando el costo, se resuelve primero

un SSCPLP con una restricción adicional de cobertura (SSCPLP+C). La formulación del SSCPLP+C es:

$$\text{mín} \quad z_1 = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} c_{ij}x_{ij} + \sum_{i\in\mathcal{I}} f_i y_i \tag{4.8}$$

Sujeto a:

$$\sum_{j\in\mathcal{J}} d_j \sum_{i\in\mathcal{Q}_j} x_{ij} \geq v \sum_{j\in\mathcal{J}} d_j \quad , \quad j\in\mathcal{J}, \quad i\in\mathcal{I} \tag{4.9}$$

(4.10)

$$\sum_{i\in\mathcal{I}} x_{ij} = 1 \quad , \quad j\in\mathcal{J} \tag{4.11}$$

$$\sum_{j\in\mathcal{J}} d_j x_{ij} \leq w_i y_i \quad , \quad i\in\mathcal{I} \tag{4.12}$$

$$x_{ij} \leq y_i \quad , \quad j\in\mathcal{J}, \quad i\in\mathcal{I} \tag{4.13}$$

$$x_{ij} \in \{0,1\} \quad , \quad j\in\mathcal{J}, \quad i\in\mathcal{I} \tag{4.14}$$

$$y_i \in \{0,1\} \quad , \quad i\in\mathcal{I} \tag{4.15}$$

Al igual que con el UFLP+C, con la solución óptima de SSCPLP+C ($z_1^*$) se alimenta un segundo problema de localización (CMCLP+B) que busca alguna solución alterna con mayor cobertura. El CMCLP+B maximiza la cobertura garantizando que el costo no es superior a $z_1^*$.

$$\text{máx} \quad z_2 = \sum_{j\in\mathcal{J}} d_j \sum_{i\in\mathcal{Q}_j} x_{ij} \tag{4.16}$$

Sujeto a:

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} c_{ij}x_{ij} + \sum_{i\in\mathcal{I}} f_i y_i \leq B \quad , j\in\mathcal{J}, \quad i\in\mathcal{I} \tag{4.17}$$

$$\sum_{i\in\mathcal{I}} x_{ij} = 1 \quad , \quad j\in\mathcal{J} \tag{4.18}$$

$$\sum_{j\in\mathcal{J}} d_j x_{ij} \leq w_i y_i \quad , \quad i\in\mathcal{I} \tag{4.19}$$

$$x_{ij} \leq y_i \quad , \quad j\in\mathcal{J}, \quad i\in\mathcal{I} \tag{4.20}$$

$$x_{ij} \in \{0,1\}, \quad j\in\mathcal{J}, \quad i\in\mathcal{I} \tag{4.21}$$

$$y_i \in \{0,1\}, \quad i\in\mathcal{I} \tag{4.22}$$

Para resolver el CMCLP+B el valor de $B$ de la restricción (4.17) se hace igual a $z_1^*$ (valor óptimo obtenido con la solución del SSCPLP+C). La solución de ambos problemas se alterna aumentando sistemáticamente la cota de cobertura $v$ para encontrar soluciones no dominadas de mayor cobertura, hasta llegar a la máxima cobertura posible.

Con un procedimiento complementario se parametriza la cobertura por intermedio del presupuesto máximo disponible $B$, y se invierte el orden de solución de los problemas: primero

se resuelve el CMCLP+B y con su valor óptimo $z_2^*$ se alimenta el valor de $v$ para resolver el SSCPLP+C.

En el Algoritmo 9 se presenta el procedimiento para aproximar la frontera eficiente parametrizando la cota de cobertura $v$. El procedimiento complementario que parametriza la cota de presupuesto $B$ se presenta en el algoritmo 10

**Algoritmo 9** PROBLEMA DE LOCALIZACIÓN CON RESTRICCIONES DE CAPACIDAD. PROCEDIMIENTO PARA LA APROXIMACIÓN DE LA FRONTERA DE PARETO PARAMETRIZANDO LA COTA DE COBERTURA, $(v)$

**Input:** Datos del problema: $h_{ij}$, $c_{ij}$, $d_j$, $w_I$ y $f_i$, distancia máxima de cobertura $D_{max}$, número de intervalos $(s)$.

**Output:** Aproximación de la frontera de Pareto ($\widehat{\mathcal{PF}}^*$), Aproximación del conjunto óptimo de Pareto ($\widehat{\mathcal{P}}^*$)

$\overline{z}_2 \leftarrow$ Resolver(CMCLP)(Sección 4.1.2)
$v^{max} \leftarrow \frac{\overline{z}_2}{\sum_{j\in\mathcal{J}} d_j}$
$\underline{z}_1 \leftarrow$ Resolver(SSCPLP) (Sección 4.1.2)
$B \leftarrow \underline{z}_1$
$(\mathbf{x}_{ij}^*, \mathbf{y}_i^*, \underline{z}_2^*) \leftarrow$ Resolver(CMCLP+B)
$\widehat{\mathcal{P}}^* \leftarrow \widehat{\mathcal{P}}^* \cup [\mathbf{x}_{ij}^*, \mathbf{y}_i^*]$
$\widehat{\mathcal{PF}}^* \leftarrow \widehat{\mathcal{PF}}^* \cup [\underline{z}_1^*, \underline{z}_2^*]$
$v_{min} \leftarrow \frac{\underline{z}_2}{\sum_{j\in\mathcal{J}} d_j}$
$\Delta v \leftarrow \frac{\overline{z}_2 - \underline{z}_2}{s \cdot \sum_{j\in\mathcal{J}} d_j}$
$v \leftarrow v_{min} + \Delta v$
**while** $v \leq v^{max}$ **do**
  $z_1^* \leftarrow$ Resolver(SSCPLP+C)
  $B \leftarrow z_1^*$
  $(\mathbf{x}_{ij}^*, \mathbf{y}_i^*, z_2^*) \leftarrow$ Resolver(CMCLP+B)
  $\widehat{\mathcal{P}}^* \leftarrow \widehat{\mathcal{P}}^* \cup [\mathbf{x}_{ij}^*, \mathbf{y}_i^*]$
  $\widehat{\mathcal{PF}}^* \leftarrow \widehat{\mathcal{PF}}^* \cup [z_1^*, z_2^*]$
  $u \leftarrow v_{min} + \Delta v \times \lceil (\frac{z_2^* - \underline{z}_2}{\Delta v \cdot \sum_{j\in\mathcal{J}} d_j}) \rceil$
  $v \leftarrow \text{máx}(v + \Delta v, u)$
**end while**

**Algoritmo 10** PROBLEMA DE LOCALIZACIÓN CON RESTRICCIONES DE CAPACIDAD. PROCEDIMIENTO PARA LA APROXIMACIÓN DE LA FRONTERA DE PARETO PARAMETRIZANDO LA COTA DE PRESUPUESTO ($B$)

**Input:** Datos del problema: $h_{ij}$, $c_{ij}$, $d_j$ $w_i$ y $f_i$, distancia máxima de cobertura $D_{max}$, número de intervalos ($s$).

**Output:** Aproximación de la frontera de Pareto ($\widehat{\mathcal{PF}}^*$), Aproximación del conjunto óptimo de Pareto ($\widehat{\mathcal{P}}^*$)

1: $\underline{z}_1 \leftarrow$ Resolver(SSCPLP) (Sección 4.1.2)
2: $\overline{z}_2 \leftarrow$ Resolver(CMCLP) (Sección 4.1.2)
3: $v^{max} \leftarrow \frac{\overline{z}_2}{\sum_{j \in \mathcal{J}} d_j}$
4: $v \leftarrow v^{max}$
5: $(\mathbf{x}^*_{ij}, \mathbf{y}^*_i, \overline{z}_1) \leftarrow$ Resolver(SSCPLP+C)
6: $\widehat{\mathcal{P}}^* \leftarrow \widehat{\mathcal{P}}^* \cup [\mathbf{x}^*_{ij}, \mathbf{y}^*_i]$
7: $\widehat{\mathcal{PF}}^* \leftarrow \widehat{\mathcal{PF}}^* \cup [\overline{z}^*_1, \overline{z}^*_2]$
8: $v \leftarrow v^{max}$
9: $\Delta B \leftarrow \frac{\overline{z}_1 - \underline{z}_1}{s}$
10: $B \leftarrow \overline{z}_1 - \Delta B$
11: **while** $B \geq \underline{z}_1$ **do**
12: $z^*_2 \leftarrow$ Resolver(CMCLP+B)
13: $v = \frac{z^*_2}{\sum_{j \in \mathcal{J}} d_j}$
14: $(\mathbf{x}^*_{ij}, \mathbf{y}^*_i, z^*_1) \leftarrow$ Resolver(SSCPLP+C)
15: $\widehat{\mathcal{P}}^* \leftarrow \widehat{\mathcal{P}}^* \cup [\mathbf{x}^*_{ij}, \mathbf{y}^*_i]$
16: $\widehat{\mathcal{PF}}^* \leftarrow \widehat{\mathcal{PF}}^* \cup [z^*_1, z^*_2]$
17: $u \leftarrow \overline{z}_1 - \Delta B \times \lceil (\frac{\overline{z}_1 - z^*_1}{\Delta B}) \rceil$
18: $B \leftarrow \text{mín}(P - \Delta P, u)$
19: **end while**

### 4.1.2. Soluciones Extremas de la Frontera Eficiente

Las soluciones extremas de la frontera eficiente se hallan con un procedimiento similar al utilizado para el problema sin restricciones de capacidad (Sección 2.4.2). La mayor diferencia es la obtención de la cota $\overline{z}_2$ de la solución de máxima cobertura.

A diferencia del problema de localización multiobjetivo sin restricciones de capacidad en el cual la cota máxima de cobertura $\overline{z}_2$ es muy fácil de encontrar, para obtener la cota $\overline{z}_2$ del problema de localización multiobjetivo con restricciones de capacidad se debe resolver el siguiente problema de optimización (CMCLP),:

$$\text{máx} \quad z_2 = \sum_{j\in\mathcal{J}} d_j \sum_{i\in\mathcal{Q}_j} x_{ij} \tag{4.23}$$

Sujeto a:

$$\sum_{i\in\mathcal{I}} x_{ij} = \quad 1 \quad , \quad j \in \mathcal{J} \tag{4.24}$$

$$\sum_{j\in\mathcal{J}} d_j x_{ij} \leq \quad w_i y_i \quad , \quad i \in \mathcal{I} \tag{4.25}$$

$$x_{ij} \leq \quad y_i \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{4.26}$$

$$x_{ij} \in \quad \{0,1\}, \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{4.27}$$

$$y_i \in \quad \{0,1\}, \quad i \in \mathcal{I} \tag{4.28}$$

Para obtener la cota minima de costo ($\underline{z}_1$) se debe resolver un problema similar al UFLP, pero con las restricciones adicionales de capacidad de las bodegas. Este problema es conocido como SSCPLP (del inglés, Single Source Capacitated Plant Location Problem):

$$\text{mín} \quad z_1 = \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} c_{ij} x_{ij} + \sum_{i\in\mathcal{I}} f_i y_i \tag{4.29}$$

Sujeto a:

$$\sum_{i\in\mathcal{I}} x_{ij} = \quad 1 \quad , \quad j \in \mathcal{J} \tag{4.30}$$

$$\sum_{j\in\mathcal{J}} d_j x_{ij} \leq \quad w_i y_i \quad , \quad i \in \mathcal{I} \tag{4.31}$$

$$x_{ij} \leq \quad y_i \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{4.32}$$

$$x_{ij} \in \quad \{0,1\}, \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{4.33}$$

$$y_i \in \quad \{0,1\}, \quad i \in \mathcal{I} \tag{4.34}$$

## 4.2. Experimento Computacional

Para probar que tan buena es la metodología de solución se realizó un experimento computacional. Se aproximó la frontera eficiente de 32 problemas diferentes. Los tamaños de los problemas utilizados se resumen en la Tabla 4.2.

| Tamaño | Bodegas | Clientes | Variables | Restricciones |
|---|---|---|---|---|
| 10-25 | 10 | 25 | 260 | 285+1 |
| 25-50 | 25 | 50 | 1275 | 1325+1 |

Tabla 4.1: Problemas de localización multiobjetivo con restricciones de capacidad. Tamaño de los problemas del experimento computacional.

El experimento computacional se realizo utilizando XpressMP disponible en NEOS. Se exploró la solución de problemas con un mayor número de bodegas y clientes (30-75 y 50-150) pero su solución desborda la memoria del servidor disponible o excede el tiempo total de procesamiento.

### 4.2.1. Generador de Problemas

De este tipo de problemas no hay instancias de dominio público con las cuales se pueda probar el desempeño de la metodología propuesta. Fue necesario desarrollar un generador de problemas aleatorios para probar con ellos la metodología se solución.

El generador de instancias desarrollado es similar al utilizado en [24]: la ubicación de los clientes se genera aleatoriamente en un cuadrado de lado 190. Para la ubicación de las bodegas se utilizaron 2 procedimientos: en los problemas tipo A se ubican en el mismo lugar que $m$ clientes seleccionados aleatoriamente, en los problemas tipo B se ubican aleatoriamente en el cuadrado de lado 190.

La demanda de los clientes $d_j$ se genera con una distribución uniforme discreta en el intervalo [10,50]. La capacidad de las bodegas $w_i$ se genera con una distribución uniforme discreta en el intervalo [20,200]. El valor de $D_{max}$ para todos los problemas fue 35. El costo de asignación de un cliente a una bodega ($c_{ij}$) se asumió proporcional a la demanda del cliente ($d_j$) y a la distancia entre el cliente y la bodega ($h_{ij}$).

$$c_{ij} = 0.05 \cdot d_j \cdot h_{ij} \tag{4.35}$$

.

Para generar el costo fijo de cada bodega $f_i$ se utilizó la expresión 4.36. Para los factores aleatorios $fa_1$ y $fa_2$ se probaron 2 opciones: en los problemas tipo F1, $fa_1$ se toma de una distribución uniforme en el intervalo [100,110) y $fa_2$ de una distribución uniforme en el intervalo [0,100). En los problemas tipo F2, $fa_1$ se toma del intervalo [50,55) y $fa_2$ del intervalo [0,50).

$$f_i = fa_1\sqrt{w_i} + fa_2 \tag{4.36}$$

.

En los problemas de localización con restricciones de capacidad se define el factor $R$, para indicar qué tan fuertes son las restricciones de capacidad, su valor se calcula como la razón de la capacidad total disponible entre la demanda de los clientes.

$$R = \frac{\sum_{i \in \mathcal{I}} w_i}{\sum_{j \in \mathcal{J}} d_j} \tag{4.37}$$

.

Cuanto más grande sea $R$ más capacidad hay y por consiguiente las restricciones de capacidad son menos fuertes. Para $R$ se consideraron 4 valores: 1.5, 2, 3 y 5. Una vez generados los costos fijos de operación, se ajusta la capacidad de todas las bodegas para que el factor $R$ tenga el valor deseado.

Las opciones disponibles para generar los problemas se resumen en la Tabla 4.2 :

| Característica | Opciones |
|---|---|
| Ubicación de las bodegas | A,B |
| Tamaño | 10-25, 25-50 |
| Costo Fijo | F1, F2 |
| R | 1.5, 2, 3, 5 |

Tabla 4.2: Problemas de localización multiobjetivo con restricciones de capacidad. Opciones para generar los problemas del experimento computacional.

Por ejemplo, el problema A10-25F1R3 tiene 10 bodegas y 25 clientes, las bodegas están ubicadas en el mismo lugar que 10 de los clientes, el costo fijo de las bodegas se genero con la opción F1 y la capacidad total de las bodegas es 3 veces la demanda de los clientes.

### 4.2.2. Resultados

Los resultados de aplicar el procedimiento de solución a los problemas con 10 bodegas y 25 clientes se presentan en la Tabla 4.3. Los problemas pequeños no presentan inconvenientes para el procedimiento descrito. La parametrización del costo ya no encuentra tantas soluciones para complementar la frontera eficiente (Figura 4.1). Podría haberse utilizado solo la parametrización de la cobertura para obtener más del 90 % de las soluciones . Aunque no se han reportado los tiempos de ejecución (ver Apéndice D), en general, las restricciones de capacidad exigen mayor tiempo para aproximar la frontera eficiente incluso en instancias pequeñas como estas. En la Figura 4.2 se presenta la aproximación de la frontera para el problema B10-25F1R3.

| | | Número de Soluciones | | | |
|---|---|---|---|---|---|
| Problema | $S'$ | Parametrizando $v$ | Parametrizando $B$ | Ambos | Total) |
| A10-25F1R1.5 | 0.6674 | 1 (33 %) | - | 2(67 %) | 3 (100 %) |
| A10-25F1R2 | 0.5458 | - | - | 5(100 %) | 5 (100 %) |
| A10-25F1R3 | 0.4636 | 2 (18 %) | 1 (9 %) | 8(73 %) | 11 (100 %) |
| A10-25F1R5 | 0.5443 | - | 1 (9 %) | 10(91 % | 11 (100 %) |
| A10-25F2R1.5 | 0.5909 | 2 (20 %) | - | 8(80 %) | 10 (100 %) |
| A10-25F2R2 | 0.4461 | 2 (40 %) | - | 3(60 %) | 5 (100 %) |
| A10-25F2R3 | 0.6311 | 1 (13 %) | - | 7(88 %) | 8 (100 %) |
| A10-25F2R5 | 0.5909 | 2 (25 %) | 1 (13 %) | 5(63 %) | 8 (100 %) |
| B10-25F1R1.5 | 0.5192 | 3 (33 %) | 1 (11 %) | 5(56 %) | 9 (100 %) |
| B10-25F1R2 | 0.7212 | 3 (27 %) | - | 8(73 %) | 11 (100 %) |
| B10-25F1R3 | 0.7529 | 2 (18 %) | - | 9(82 %) | 11 (100 %) |
| B10-25F1R5 | 0.7317 | 3 (21 %) | 1 (7 %) | 10(71 %) | 14 (100 %) |
| B10-25F2R1.5 | 0.6371 | 5 (29 %) | 2 (12 %) | 10(59 %) | 17 (100 %) |
| B10-25F2R2 | 0.6438 | 1 (7 %) | 1 (7 %) | 12(86 %) | 14 (100 %) |
| B10-25F2R3 | 0.7163 | 4 (31 %) | 3 (23 %) | 6(46 %) | 13 (100 %) |
| B10-25F2R5 | 0.7163 | 4 (31 %) | 3 (23 %) | 6(46 %) | 13 (100 %) |

Tabla 4.3: Problemas de localización multiobjetivo con restricciones de capacidad. Resultados experimento computacional (Problemas10-25).

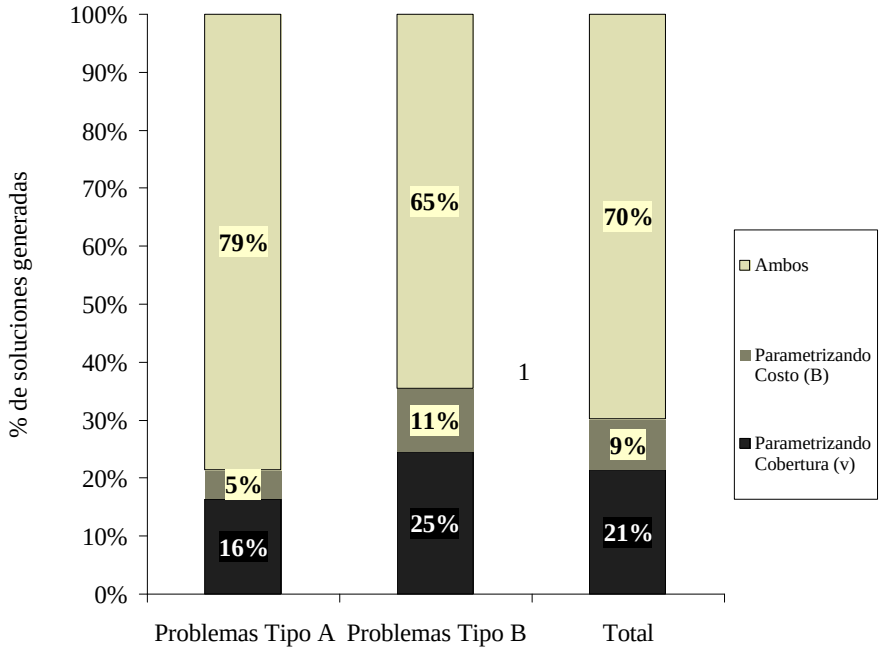


Figura 4.1: Problemas de localización multiobjetivo con restricciones de capacidad. Porcentaje de Soluciones generado con cada procedimiento.

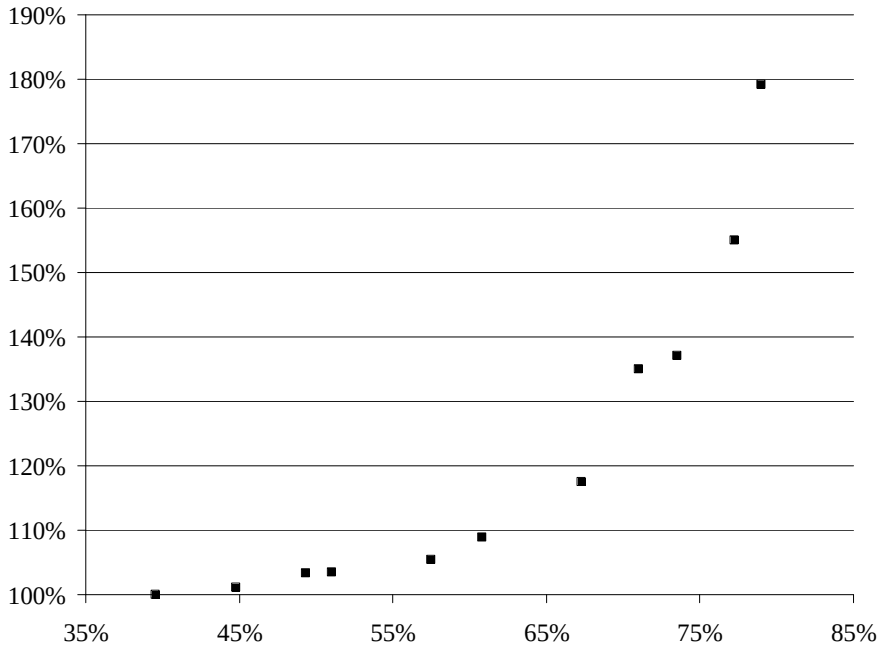


Figura 4.2: Problemas de localización multiobjetivo con restricciones de capacidad. Aproximación de la frontera problema B10-25F1R3 .

Los resultados en problemas un poco más grandes (25 bodegas y 50 clientes) se presentan en la Tabla 4.4). En tres de los problemas (A25-50F2R1.5, B25-50F1R1.5, B25-50F2R1.5) la solución ideal es factible (i.e., la solución de mínimo costo es también la que ofrece mejor cobertura). Como las bodegas son de poca capacidad ($R$ bajo), para poder cumplir la cota de capacidad se deben abrir muchas bodegas y como resultado los clientes son asignados a bodegas cercanas que los pueden atender a bajo costo.

En los problemas A25-50F1R1.5, A25-50F2R2 y B25-50F2R2 se excedió el limite de memoria disponible para su solución. En los demás problemas no se presentó mayor inconveniente, quizás lo ajustado de las restricciones de capacidad haga mucho más difícil la solución de estos problemas.

Para valores mayores de R ($R = 3$ y $R = 5$) se encuentra un mayor número de soluciones en la aproximación de la frontera, y con excepción del problema A25-50F1R5 todos los procedimientos fueron exitosos.

## 4.3. Conclusiones

En algunas situaciones, es conveniente considerar restricciones de capacidad que limitan la demanda que una instalación puede atender. Los modelos y la metodología basada en programación matemática que se desarrollo para el problema sin restricciones de capacidad se pueden extender. Se requieren ligeras modificaciones para considerar las restricciones de capacidad. En general, el procedimiento de aproximación es más complicado y se toma más

| | | Número de Soluciones | | | |
|---|---|---|---|---|---|
| Problema | $S'$ | Parametrizando $v$ | Parametrizando $B$ | Ambos | Total) |
| A25-50F1R1.5 | $ND^a$ | $1^b$ | $ND$ | $ND$ | 1(100 %) |
| A25-50F1R2 | 0.3594 | - | - | 4(100 %) | 4(100 %) |
| A25-50F1R3 | 0.8066 | 3(21 %) | 4(29 %) | 7(50 %) | 14100 %) |
| A25-50F1R5 | 0.7394 | 16(100 %) | $ND^c$ | $ND$ | 16(100 %) |
| A25-50F2R1.5 | 1.0000 | - | - | $1^d$(100 %) | 1(100 %) |
| A25-50F2R2 | $ND^a$ | $ND$ | $ND$ | $2^e$ | 2(100 %) |
| A25-50F2R3 | 0.5700 | 9(43 %) | 2(10 %) | 10(48 %) | 21(100 %) |
| A25-50F2R5 | 0.6549 | 14(56 %) | 9(31 %) | 6(21 %) | 29(100 %) |
| B25-50F1R1.5 | 1.0000 | - | - | $1^d$ (100 %) | 1(100 %) |
| B25-50F1R2 | 0.5887 | - | - | 5 (100 %) | 5(100 %) |
| B25-50F1R3 | 0.6317 | 10(37 %) | 7(26 %) | 10 (37 %) | 27(100 %) |
| B25-50F1R5 | 0.6765 | 10(37 %) | 7(26 %) | 10 (37 %) | 27(100 %) |
| B25-50F2R1.5 | 1.0000 | - | - | $1^d$ (100 %) | 1(100 %) |
| B25-50F2R2 | $ND^a$ | $ND$ | $ND$ | $2^e$ | 2(100 %) |
| B25-50F2R3 | 0.6385 | 13(45 %) | 11(38 %) | 5 (17 %) | 29(100 %) |
| B25-50F2R5 | 0.7507 | 7(32 %) | 7(32 %) | 8 (36 %) | 22(100 %) |

*a. En ambos procedimientos se excedió el limite de memoria, solo se encontraron algunas de las soluciones extremas.*

*b. Solución de mínimo costo.*

*c. El procedimiento excedió el limite de memoria*

*d. Ideal factible*

*e. Soluciones extremas*

Tabla 4.4: Problemas de localización multiobjetivo con restricciones de capacidad. Resultados experimento computacional (Problemas 25-50).

tiempo, ya que los problemas de optimización que se deben resolver en cada iteración son más difíciles.

En problemas pequeños (por ejemplo 10 bodegas y 25 clientes) la metodología sigue siendo efectiva, y en todos los casos permite aproximar apropiadamente la frontera eficiente. Para algunos problemas de un tamaño ligeramente mayor (25 bodegas y 50 clientes), en los que las restricciones de capacidad son muy ajustadas, es posible que sólo se encuentren las soluciones de máxima cobertura o de mínimo costo. Además, es conveniente buscar primero la solución de mínimo costo, y evaluar que cobertura ofrece. En algunos casos, lo restringido de la capacidad puede hacer que esta solución tenga muchas bodegas abiertas y, por consiguiente, buena cobertura.

Lo poco satisfactorio de los resultados obtenidos no descalifican del todo este método de solución, a pesar de sus limitaciones sigue siendo aplicable. Sobre todo si se logra mejorar el procedimiento de solución de los problemas de optimización que se resuelven en cada iteración. El desarrollo de heurísticos lagrangianos y otros métodos basados en programación matemática pueden ser motivo de nuevas investigaciones.

Abordar el problema de localización multiobjetivo con restricciones de capacidad usando algoritmos evolutivos es un camino que se debe explorar. Sobre todo, si se consideran los buenos resultados obtenidos por los algoritmos evolutivos para aproximar la frontera eficiente de problemas de localización multiobjetivo sin restricciones de capacidad. Sin embargo, la incorporación de restricciones de capacidad en un algoritmo evolutivo no es una tarea sencilla. Por ende, podrían explorarse otras técnicas metaheurísticas de optimización multiobjetivo como *Pareto Simulated Annealing* [28] y *Multi-objective Tabu Search* [46]

# Capítulo 5

# RED DE ACOPIO Y ALMACENAMIENTO DE CAFÉ COLOMBIANO

Las áreas de aplicación de los modelos de localización son innumerables, para una amplia recopilación se pueden consultar [26, 31]. Por ejemplo, se han empleado para la ubicación de instalaciones publicas como rellenos sanitarios [3] y estaciones de bomberos [5], y en el sector privado en diversas situaciones como: construcción de plataformas de explotación petrolera [37] y diseño de sistemas logísticos [72], entre otras.

Los modelos de localización multiobjetivo estudiados en este trabajo no son una excepción. A continuación, se describe la aplicación de dichos modelos para el análisis de la red de acopio y almacenamiento de café colombiano.

## 5.1. Descripción del Problema

El proceso de exportación de cafe colombiano comienza por las agencias de compra de las cooperativas de caficultores, que se encuentran en los diferentes de departamentos del territorio nacional. Actualmente se cuenta con mas de 450 agencias de compra que se encuentran en los departamentos de Antioquia, Boyacá, Caldas, Cauca, Caquetá, Casanare, Cesar, Cundinamarca, Huila, Guajira , Magdalena, Meta, Nariño, Norte de Santander, Quindío, Risaralda, Santander, Tolima y Valle del Cauca.

La cooperativa de caficultores, envía a Almacafé el café adquirido para la Federación Nacional de Cafeteros, donde se conserva para enviarlo posteriormente a la trilladora. Una vez trillado el café colombiano se lleva al puerto para su exportación. Para recibir el café que adquieren las cooperativas de caficultores, Almacafé cuenta con una red con más de 30 bodegas ubicadas en las principales ciudades de nuestro país, y en poblaciones intermedias cercanas a las zonas de gran producción cafetera.

La situación actual de los precios internacionales exige que el proceso de exportación sea lo más eficiente posible. Como solución para disminuir los costos de operación se ha planteado la necesidad de reducir la infraestructura de operación [32]. Sin embargo, desde el punto de

vista de los caficultores esta alternativa no es del todo atractiva. La garantía de compra de la cosecha, (en un lugar cercano al lugar de producción) es de los servicios más apreciados, de los que presta la Federación Nacional de Cafeteros por intermedio de Almacafé.

Claramente, el costo de operación y el servicio al caficultor son dos objetivos que se encuentran en conflicto. Operar la red eficientemente exige reducir el número de bodegas, mientras que garantizar la compra de la cosecha exige estar cerca de los caficultores. Esta situación motivó la utilización de los modelos de localización multiobjetivo como herramienta para el análisis de la red de acopio y almacenamiento del café colombiano.

## 5.2. Datos

Para construir el modelo de optimización multiobjetivo es necesario contar con los datos de ubicación de las bodegas y clientes, los costos de operación de las bodegas y en general todos los componentes del problema de localización multiobjetivo descrito en (1.3) – (1.8). Por razones de confidencialidad los valores de dichos datos no se reportan en este documento. Sin embargo, se incluye una breve descripción de los diferentes componentes.

Siguiendo las recomendaciones de [17] los 450 puntos de compra se agregaron en 47 nodos. Para agregarlos se consideró la cercanía y el volumen de café comprado. En general, cada cooperativa está representada por sus agencias principales (entre uno y tres agencias), igualmente, el café adquirido por cada cooperativa ($d_j$) se consolido en dichas agencias de compra. Para la oferta de café de cada cooperativa se utilizó la información correspondiente al año 2001[33]. La ubicación de todas las agencias de compra y su distancia a las diferentes bodegas de Almacafé ($h_{ij}$) es información que se conoce plenamente. Como distancia máxima de cobertura se escogió $D_{max} = 150$km. Después del proceso de consolidación, la distribución geográfica de los diferentes puntos de compra se resume en la Tabla 5.1.

Para el análisis se han considerado 25 ubicaciones para las bodegas (Tabla 5.2), que resultan de consolidar en una sola bodega toda la capacidad de almacenamiento disponible en una misma ciudad o población. En los costos fijos de operación de las bodegas ($f_i$) se reúnen los costos de mantenimiento, seguridad, impuestos, servicios públicos y personal mínimo requerido para su operación. También se cuenta con la información sobre las capacidades máximas de cada bodega ($w_i$). Como es de esperarse, el costo de operación y la capacidad de una bodega se encuentran altamente correlacionados.

La Figura 5.1 ilustra la ubicación geográfica de las puntos de compra ("▪"marcados en verde) y las bodegas ("◊"marcadas en rojos)

Por último, se recogió la información sobre los costos de asignación de los puntos de compra a las bodega ($c_{ij}$). En estos costos se incluye el costo del transporte del café desde la agencia de compra hasta la bodega de Almacafé y su descargue, el costo de adquirir el café en la ubicación correspondiente (el precio de compra en las bodegas es diferente, las bodegas más cercanas a los puertos pagan un mayor precio), los costos de trillar el cafe en dicha plaza (dependen de la disponibilidad y propiedad de las trilladoras de la región, por ejemplo, si la trilladora es del Fondo Nacional del Café no se incurre en el costo del desplazamiento urbano) y por ultimo los costos de enviar el café excelso al puerto de exportación.

| Departamento | Puntos de Compra |
|---|---|
| Antioquia | 6 |
| Boyacá | 2 |
| Caldas | 3 |
| Caquetá | 1 |
| Casanare | 1 |
| Cauca | 3 |
| Cesar Guajira | 2 |
| Cundinamarca | 3 |
| Huila | 3 |
| Magdalena | 1 |
| Meta | 1 |
| Nariño | 2 |
| Norte Santander | 2 |
| Quindío | 3 |
| Risaralda | 2 |
| Santander | 3 |
| Tolima | 5 |
| Valle del Cauca | 4 |

Tabla 5.1: Ubicación de las agencias de compra por departamento

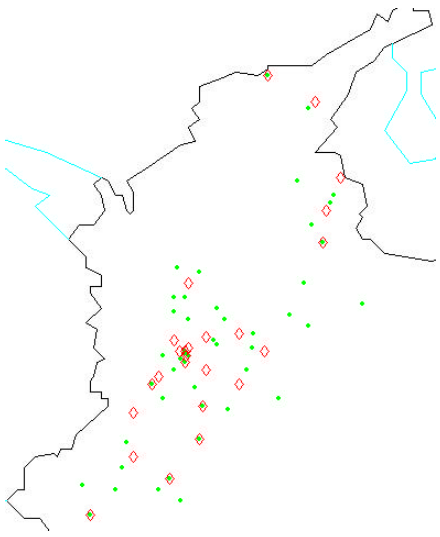

Figura 5.1: Ubicación de los puntos de compra y las bodegas.

| Bodega | % de la capacidad total |
|---|---|
| Medellín | 22.0 % |
| Buga | 16.2 % |
| Ibague | 7 % |
| Chinchina | 6.5 % |
| Bogota | 5.1 % |
| Letras | 4.6 % |
| Pereira | 4 % |
| Armenia | 3.6 % |
| Arroyohondo | 3.5 % |
| Bucaramanga | 2.9 % |
| Manizales | 2.9 % |
| Popayan | 2.9 % |
| Pasto | 2.8 % |
| Neiva | 2.4 % |
| Tulua | 2.2 % |
| Santa Marta | 2.2 % |
| Honda | 1.8 % |
| Girardot | 1.6 % |
| Pamplona | 1.4 % |
| Chaparral | 0.9 % |
| Garzon | 0.9 % |
| Santa Rosa | 0.8 % |
| Valledupar | 0.8 % |
| La Virginia | 0.6 % |
| Cucuta | 0.6 % |

Tabla 5.2: Conjunto de bodegas

## 5.3. Resultados

### 5.3.1. Análisis Preliminar

Inicialmente se analizó la distribución regional de la oferta de café y la capacidad de almacenamiento disponible. Se identificaron varias zonas del país: En el eje cafetero, norte del Valle y Antioquia se concentra el mayor número de bodegas (Tulua, Santa Rosa, Pereira, Medellín, Manizales, Letras, La Virginia, Chinchina, Buga y Armenia), el 63.4 % de la capacidad de almacenamiento y el 56 % de la oferta de café. La región del Tolima y Huila, concentra el 20 % del café, allí se encuentran 6 bodegas (Ibagué, Neiva, Garzón, Chaparral, Honda y Girardot) y el 12.8 % de la capacidad de almacenamiento.

El 7.4 % de la oferta se encuentra en el departamento del Cauca que tiene una Bodega en Popayan (2.9 % de la capacidad de almacenamiento). Santander aporta el 5.8 % del café y cuenta con la bodega de Bucaramanga (2.9 % de la capacidad de almacenamiento). El restante 10.8 % de la oferta de café lo aportan las otras regiones, donde se tienen 7 bodegas que representan el 17.9 % de la capacidad de Almacenamiento.

Analizando las distancias entre las bodegas y los puntos de compra se encontró que algunos puntos solo pueden ser cubiertos por una sola bodega. La Tabla 5.3 muestra las bodegas, los puntos de compra que solo pueden ser cubiertos por cada bodega y su aporte a la oferta total de café.

| Bodega | Puntos de Compra | % Oferta |
|---|---|---|
| Bogotá | 2 | 0.7 % |
| Cucutá | 2 | 0.8 % |
| Garzón | 2 | 4.5 % |
| Medellín | 4 | 11.0 % |
| Pasto | 2 | 1.2 % |
| Popayan | 1 | 0.9 % |
| Santa Marta | 1 | 1.4 % |
| Valledupar | 1 | 1.9 % |

Tabla 5.3: Puntos de Compra que solo pueden ser cubiertos por una Bodega

También hay cinco puntos de compra que no pueden ser cubiertos por ninguna bodega, porque la distancia que lo separa de la bodega más cercana es mayor que la distancia máxima de cobertura. Su aporte a la oferta de café se muestran en la Tabla 5.4. De ahí que, la configuración de máxima cobertura puede tener, a lo sumo, el 95.6 % de la demanda cubierta.

Por ultimo se analizó el potencial de cobertura de cada bodega. Es decir, para cada bodega que porcentaje de la demanda total se encuentra a una distancia inferior a $D_{max}$.

Los resultados de la Tabla 5.5 analizados en conjunto con los de la Tabla 5.3, indican que posiblemente las soluciones de cobertura alta van a tener muchas bodegas abiertas. Debido a que las bodegas de Valledupar, Santa Marta, Pasto, Cucutá y Bogotá que son las de menor potencial de cobertura también están incluidas en la Tabla 5.3. Es decir, estas bodegas, aunque no aportan mucho a la cobertura total, se deben abrir para lograr coberturas cercanas a la máxima ya que algunos puntos de compra solo pueden ser cubiertos por ellas.

| Puntos de Compra | Distancia a la bodega más cercana | % Oferta |
|---|---|---|
| Boyacá (Bog) | 194 | 0.1 % |
| Casanare | 404 | 0.2 % |
| Catatumbo | 261 | 2.1 % |
| Sur del Cesar | 159 | 0.6 % |
| Oriente Tolima | 152 | 1.4 % |

Tabla 5.4: Puntos de compra que no pueden ser cubiertos

| Bodega | Potencial de Bodega Cobertura |
|---|---|
| Manizales | 40.5 % |
| La Virginia | 37.5 % |
| Chinchina | 36.6 % |
| Pereira | 35.8 % |
| Armenia | 32.0 % |
| Santa Rosa | 30.7 % |
| Tulua | 29.6 % |
| Letras | 27.7 % |
| Ibagué | 23.8 % |
| Medellín | 23.5 % |
| Buga | 18.0 % |
| Arroyohondo | 13.2 % |
| Girardot | 11.6 % |
| Garzón | 9.6 % |
| Pamplona | 9.5 % |
| Popayan | 7.4 % |
| Chaparral | 7.0 % |
| Honda | 6.6 % |
| Bucaramenga | 5.8 % |
| Neiva | 5.0 % |
| Bogotá | 2.4 % |
| Valledupar | 1.9 % |
| Santa Marta | 1.4 % |
| Pasto | 1.2 % |
| Cucutá | 0.8 % |

Tabla 5.5: Potencial de cobertura de las bodegas

### 5.3.2. Configuración Actual

Una descripción breve de la configuración actual permitirá conocer algunos detalles de la operación de la red. Se opera con un total de 15 bodegas (Tabla 5.8). Algunas son utilizadas en el limite de su capacidad y para liberar capacidad se recurre a envíos de café a otras bodegas, lo que aumenta los costos de operación de la red. La cobertura que ofrece la configuración es del 86.3 %. La Figura 5.2 ilustra la ubicación de las bodegas y la asignación de los puntos de compra a ellas .

| Bodega | Oferta asignada | Puntos de compra asignados |
|---|---|---|
| Medellin | 18.7 % | 6 |
| Chinchina | 14.1 % | 2 |
| Ibague | 11.2 % | 5 |
| Neiva | 9.4 % | 3 |
| Bucaramanga | 8.1 % | 4 |
| Pereira | 7.9 % | 3 |
| Armenia | 7.5 % | 4 |
| Popayan | 7.3 % | 3 |
| Buga | 3.8 % | 2 |
| Manizales | 3.7 % | 2 |
| Valledupar | 2.5 % | 2 |
| Bogota | 2.2 % | 6 |
| Santa Marta | 1.4 % | 1 |
| Pasto | 1.2 % | 2 |
| Cucuta | 0.8 % | 2 |

Tabla 5.6: Configuración actual. Bodegas abiertas

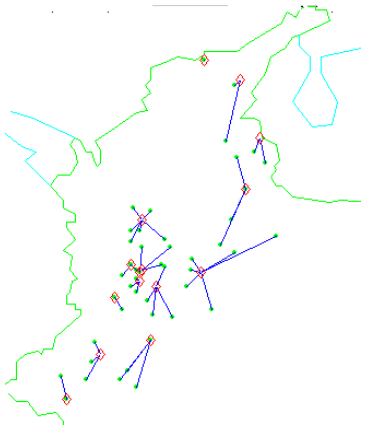

Figura 5.2: Configuración actual.

## 5.4. Modelos de localización multiobjetivo aplicados al diseño de la red de acopio y almacenamiento de café

Utilizando el modelo de localización multiobjetivo formulado en r. Se analizó la red de abastecimiento de café colombiano. Inicialmente, no se consideraron restricciones de capacidad. Para la aproximación de la frontera eficiente se utilizó la metodología de programación matemática descrita en el Capítulo 3. El modelo se construyo usando AMPL/XpresMP disponible en el servidor de optimización NEOS.

Se obtuvo una aproximación de la frontera eficiente con 28 soluciones diferentes. Todas son de un costo muy inferior al de la configuración actual. Sin embargo, haber ignorado las restricciones de capacidad, hace que en todas las soluciones se abran las bodegas más pequeñas, ya que son más baratas. En consecuencia, si se consideraran las restricciones de capacidad ninguna de las soluciones obtenidas sería factible. La Figura 5.3 ilustra la aproximación de la frontera.

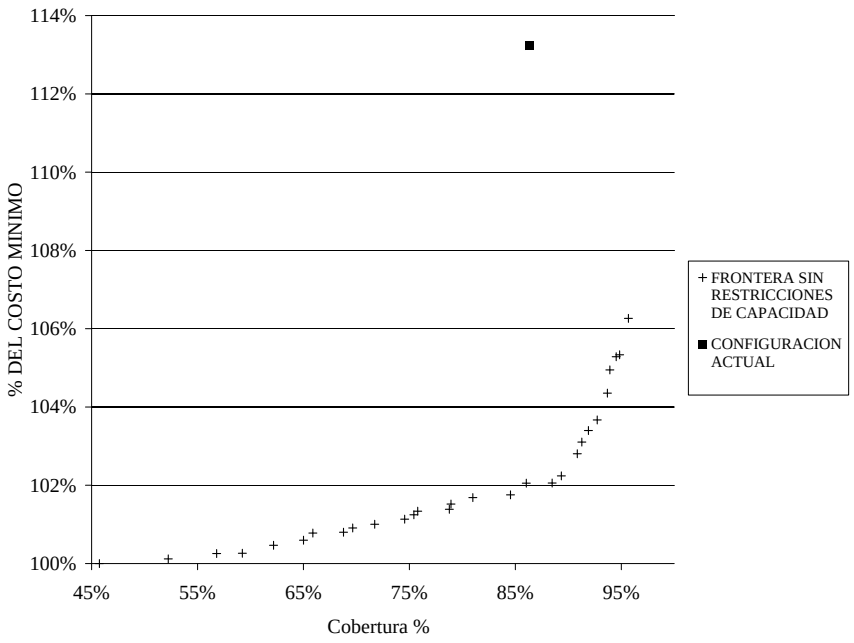


Figura 5.3: Abastecimiento del café colombiano. Frontera sin restricciones de capacidad

En la mayoría de soluciones se abren las bodegas más pequeñas, que son precisamente las más baratas, esto hace que el análisis de costo no tenga mucha validez. Los resultados principales se derivan de analizar el componente de cobertura del modelo. Por ejemplo, en la configuración de máxima cobertura, que se ilustra en la Figura 5.4, se seleccionaron apenas 12 bodegas.

La ubicación de las bodegas muestra una clara regionalización del país, en ninguna región se seleccionaron más de 3 bodegas a la vez. En el eje cafetero y el norte del Valle se abrieron las bodegas de Tulua y Manizales, en el gran Tolima las bodegas de Chaparral y Garzón, en el nororiente las de Cucutá, Bucaramanga y Valledupar, en Antioquia la bodega de Medellín. En el sur del país Pasto y Popayan, en el Norte la bodega de Santa Marta. Y por ultimo la

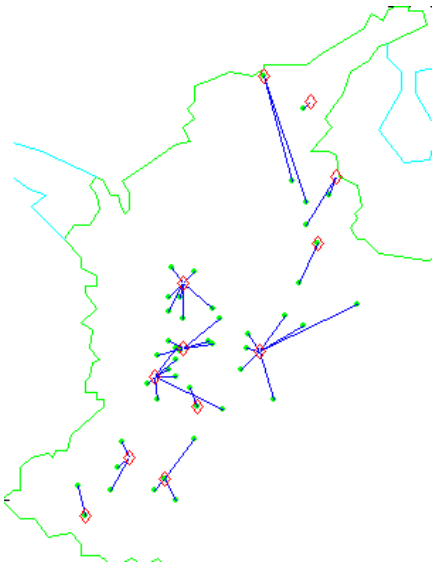

Figura 5.4: Configuración de máxima cobertura. Modelo sin restricciones de capacidad

bodega de Bogotá que atiende el centro del pais.

La cobertura obtenida con esta configuración es del 95.6 %, superior a la de la configuración actual. El 9.3 % de diferencia se deriva de la posibilidad de atender algunos puntos de compra desde bodegas más cercanas, algo que no es posible dadas las restricciones de capacidad existentes. Los resultados sugieren que aumentando la capacidad de algunas bodegas como Valledupar y Neiva se puede mejorar la cobertura, sin embargo, como no se ha considerado el costo de ampliación de la capacidad, no es posible determinar la inversión necesaria para aumentar el servicio.

Para mejorar el análisis se recurrió al modelo de localización con restricciones de capacidad desarrollado en el Capítulo 4. Nuevamente, se utilizo NEOS para resolver los problemas de optimización necesarios para aproximar la frontera eficiente del problema.

Como resultado, se obtuvo una aproximación de la frontera con 26 soluciones no dominadas, que se resumen en la Tabla 5.7 y la Figura 5.5

| Configuración | Bodegas | % del costo mínimo | Cobertura ( %) |
|---|---|---|---|
| Mínimo costo | 4 | 100.0 % | 49.8 % |
| 1 | 5 | 100.0 % | 55.7 % |
| 2 | 5 | 100.3 % | 59.4 % |
| 3 | 5 | 100.3 % | 60.7 % |
| 4 | 6 | 100.4 % | 61.5 % |
| 5 | 6 | 100.6 % | 65.1 % |
| 6 | 6 | 100.7 % | 66.5 % |
| 7 | 7 | 101.1 % | 68.9 % |
| 8 | 7 | 101.2 % | 70.2 % |
| 9 | 7 | 101.3 % | 71.0 % |
| 10 | 7 | 101.7 % | 72.3 % |
| 11 | 7 | 101.8 % | 74.9 % |
| 12 | 8 | 102.4 % | 76.0 % |
| 13 | 8 | 102.4 % | 77.3 % |
| 14 | 8 | 102.6 % | 79.5 % |
| 15 | 9 | 103.4 % | 81.3 % |
| 16 | 9 | 103.5 % | 83.1 % |
| 17 | 10 | 104.4 % | 84.0 % |
| 18 | 10 | 104.7 % | 84.4 % |
| 19 | 11 | 105.4 % | 85.1 % |
| 20 | 11 | 105.6 % | 85.2 % |
| 21 | 11 | 105.7 % | 85.5 % |
| 22 | 12 | 106.2 % | 85.6 % |
| 23 | 12 | 106.5 % | 86.3 % |
| Máxima Cobertura | 13 | 107.5 % | 86.5 % |

Tabla 5.7: Aproximación de la Frontera. Configuraciones de la red

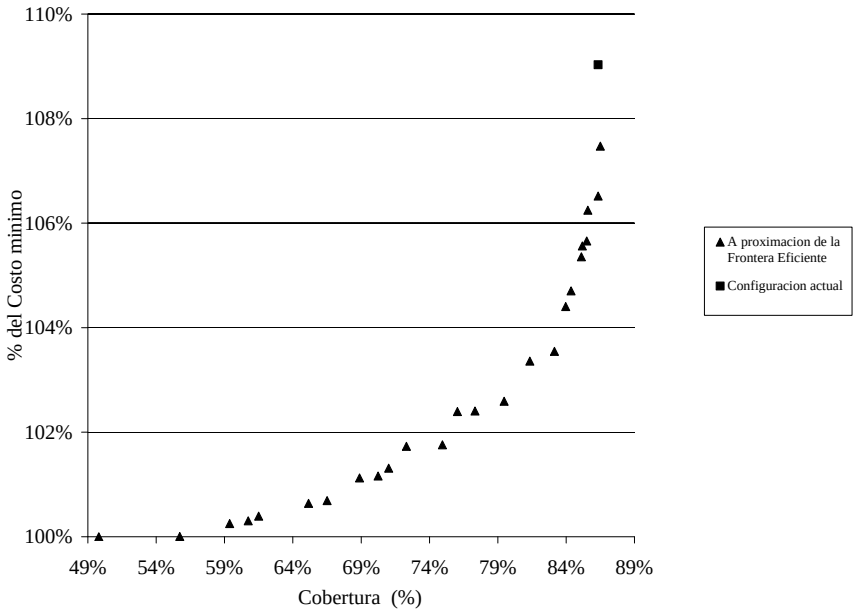


Figura 5.5: Modelo con restricciones de capacidad. Aproximación de la frontera

La configuración actual no es eficiente en el sentido de Pareto, está dominada por dos de las configuraciones de la aproximación de la frontera: una que configuración ofrece la misma cobertura a un menor costo y con tres bodegas menos, y la configuración de maxima cobertura que ofrece mayor cobertura a un costo menor y con dos bodegas menos.

La cobertura y el costo crecen de forma diferente, por ejemplo: la diferencia en costo entre la configuración de costo mínimo y la de máxima cobertura es del 7 %, pero esta diferencia se traduce en una cobertura superior en 37 % de la configuración de máxima cobertura con respecto a la cobertura de la configuración de mínimo costo.

La Figura 5.6 muestra la relación entre el costo, la cobertura y el número de bodegas. En un comienzo mejorar la cobertura ofrecida no es muy costoso, por ejemplo, en la configuración 7, con aumentar en 1.1 % el costo de operación se logra una cobertura mejor en un 19.1 % con respecto a la configuración de mínimo costo.

Lo contrario ocurre con las bodegas abiertas para llegar a las configuraciones de mayor cobertura, el costo sigue creciendo en la misma misma proporción, mientras que el aporte marginal que las nuevas bodegas hacen a la cobertura es muy pequeño. Esto ocurre porque las ultimas bodegas se abren con el objeto de cubrir puntos de compra pequeños y lejanos de las demás bodegas , como en el caso de Pasto y Cucutá.

Como medida de la importancia de las bodegas en el diseño de la red, se ha tomado el número de veces que fueron abiertas en las soluciones de la aproximación de la frontera eficiente. Figura 5.7. Las bodegas de Buga, Chinchina, Santa Marta, Medellín, Neiva, Bucaramanga, Popayan, Ibague y Manizales aparecen repetidamente en las soluciones (más de 10 veces). Lo que indica su importancia para el abastecimiento de café. Otras bodegas como Pasto, Cucutá y Bogotá, a pesar del poco aporte que sus regiones tiene en el total de la oferta de

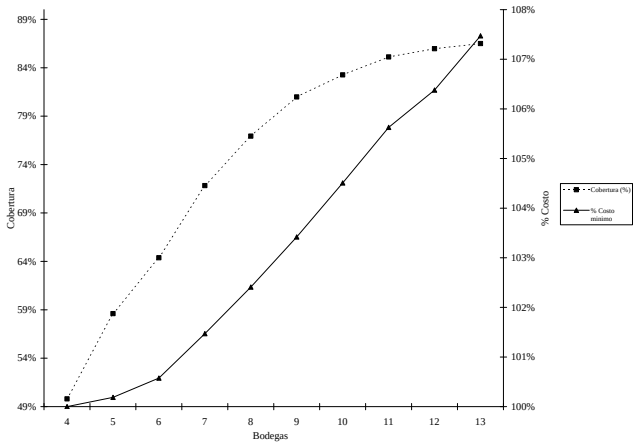


Figura 5.6: Crecimiento del costo y la cobertura. Modelo con restricciones de capacidad

café, son importantes si se piensa en ofrecer un buen servicio. Por el contrario Bodegas como Armenia (0 veces)y Pereira (2 veces), que se pensaría son muy importantes no aparecen o aparecen en muy pocas soluciones, debido a que sus áreas de influencia se traslapan con las de bodegas más importantes para la red como Buga y Chinchina.

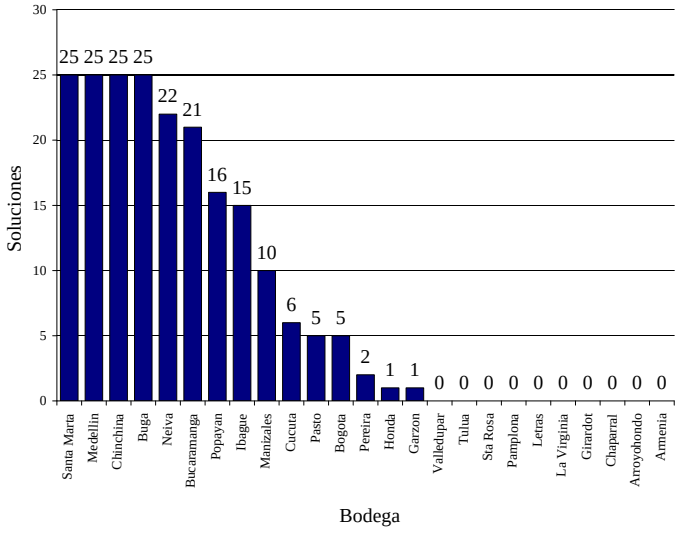


Figura 5.7: Modelo con restricciones de capacidad. Importancia de las bodegas

Para terminar se describirá brevemente la configuración de la frontera que ofrece la misma cobertura que la configuración actual a un menor costo. Existen otras 23 soluciones en la frontera eficiente que también son interesantes para el análisis, en Figura 5.8 se muestran algunas de ellas.

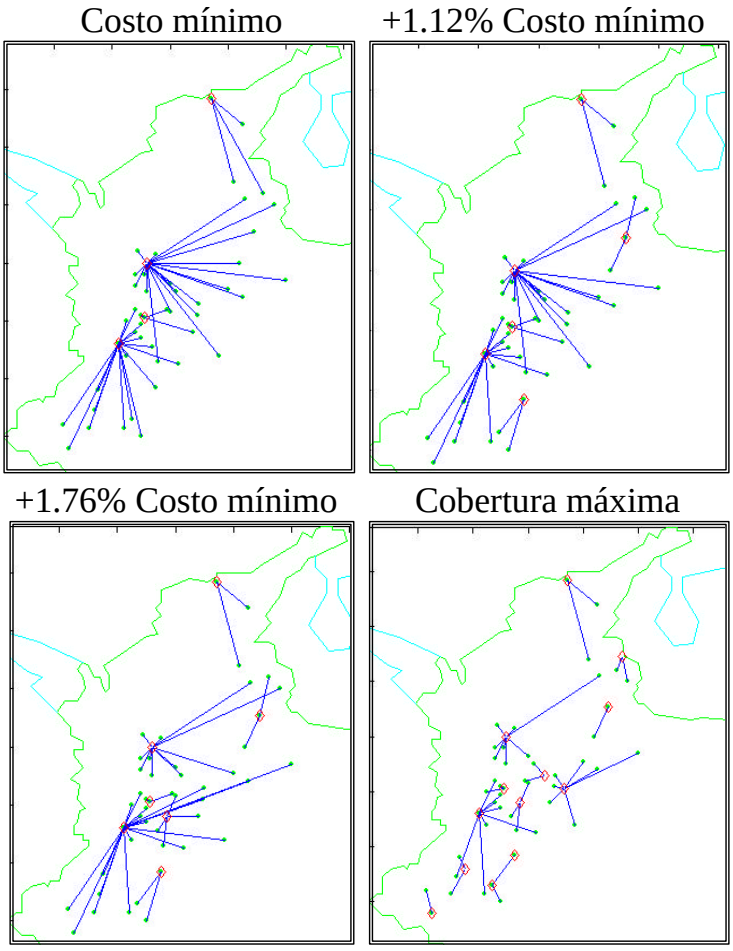


Figura 5.8: Modelo con restricciones de capacidad. Configuraciones de la frontera eficiente

La configuración mostrada en la Figura 5.9 , tiene la misma cobertura que la configuración actual. Pero, el costo necesario para operarla es inferior al de la configuración actual en un 2.35 %. Opera sin las bodegas de Armenia, Pereira y Valledupar. Los puntos de compra que son atendidos actualmente por Valledupar pasan a ser atendidos por Santa Marta. En el eje cafetero y norte del Valle se hace una reasignación de varios puntos de compra. Los puntos de compra atendidos por la bodega de Armenia pasan a ser atendidos por la bodega de Buga. Los de Risaralda pasan a ser atendidos por Buga y Chinchina, que a su vez libera capacidad

trasfiriéndole un punto de compra a Medellín.

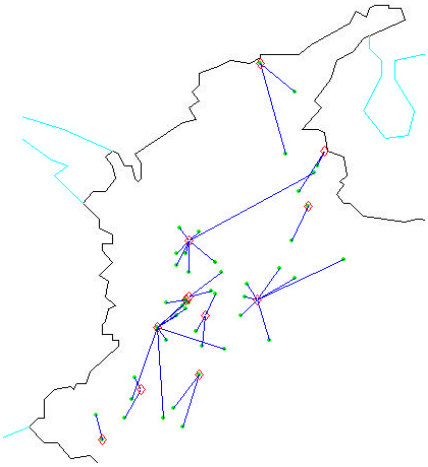

Figura 5.9: Modelo con restricciones de capacidad. Mejora a la configuración actual

En general, los puntos de compra que no pueden ser cubiertos porque las limitaciones de capacidad lo impiden o porque están muy alejados de cualquier bodega son asignados a bodegas algo retiradas. Por ejemplo tres puntos de compra, uno en el Cauca, uno en el Huila y otro en el Tolima que no pueden ser atendidos por las bodegas cercanas debido a las limitaciones de capacidad son asignados a Buga. El punto de compra del Catatumbo que no puede ser cubierto por ninguna bodega es asignado a Medellin. Y un punto de compra de Bucaramnaga pasa a ser atendido por Bogotá.

En resumen, esta configuración de la red, que puede mejorar el desempeño de la configuración actual, obtiene los ahorros reportados, en gran medida, gracias a la consolidación de la oferta de café del eje cafetero en 3 bodegas (Chinchina, Manizales y Buga), para aprovechar mejor la capacidad de la bodega de Buga, que es una de las mas grandes. Así, se mejora la utilización de la infraestructura disponible en la zona, y se elimina el traslape de las zonas de influencia de algunas bodegas.

| Bodega | Oferta asignada | Puntos de compra asignados |
|---|---|---|
| Medellin | 25.6 % | 8 |
| Buga | 24.6 % | 10 |
| Chinchina | 13.7 % | 3 |
| Ibague | 8.3 % | 3 |
| Bucaramanga | 5.8 % | 2 |
| Neiva | 5.2 % | 3 |
| Popayan | 4.5 % | 2 |
| Santa Marta | 3.9 % | 3 |
| Manizales | 3.8 % | 2 |
| Bogota | 2.7 % | 7 |
| Pasto | 1.2 % | 2 |
| Cucutá | 0.8 % | 2 |

Tabla 5.8: Mejora a la Configuración actual. Bodegas abiertas

## 5.5. Conclusiones

Los modelos de localización multiobjetivo desarrollados en este trabajo han mostrado su utilidad para abordar una situación real, relacionada con el diseño de la red de acopio y almacenamiento del café colombiano. La metodología de programación matemática propuesta sirvió como herramienta para obtener una aproximación de la frontera eficiente de soluciones no dominadas que representan configuraciones de la red.

Con dicha frontera se determinó que la configuración actual de la red no es eficiente y puede mejorarse de múltiples formas. Una alternativa que parece apropiada, es operar con tres bodegas menos y reasignar algunos puntos de compra a otras bodegas.

También se analizó el *trade off* existente entre el costo y la garantía de compra a los caficultores. Se encontró que si se opera con configuraciones de muy bajo costo se puede estar desperdiciando la oportunidad de ofrecer un mejor servicio a un costo marginal muy bajo. Ya que existen configuraciones de la red con costos cercanos al mínimo, pero con coberturas mucho mejores. Por otro lado, si se quiere ofrecer un buen servicio, con configuraciones de cobertura alta se debe incurrir en mayores costos, que marginalmente aportan poco a la cobertura, porque las últimas bodegas se abren con el objetivo de cubrir unos pocos puntos de compra pequeños y aislados.

Desarrollar modelos multiobjetivo que consideren criterios adicionales de decisión como la distancia promedio, la utilización o la equidad enriquecerían el análisis de la situación abordada. También son susceptibles de ser aplicados modelos que consideren los sistemas de distribución con todos sus niveles De este modo se podría involucrar en el análisis otras decisiones concernientes a las trilladoras y los puntos de compra de las cooperativas.

# Capítulo 6

# CONCLUSIONES

Los modelos multiobjetivo desarrollados permiten analizar problemas de localización, en los cuales tanto el costo como la cobertura son importantes. De este modo, se obtienen soluciones eficientes mucho mejores que las que se obtendrían si se ignorase la naturaleza multiobjetivo del problema.

Para la obtención de soluciones eficientes se implementaron dos algoritmos evolutivos, NSGAII y PAES. Ambos algoritmos evolutivos permiten obtener buenas aproximaciones de la frontera eficiente de los problemas de localización multiobjetivo sin restricciones de capacidad. Cuando se compara su desempeño, NSGAII obtiene mejores resultados que PAES, pero con tiempos de ejecución más largos.

Se desarrolló una segunda metodología de solución, basada en programación matemática. Esta metodología es igualmente efectiva para aproximar la frontera eficiente de los problemas de localización multiobjetivo sin restricciones de capacidad. Con la ventaja adicional de saber que las soluciones obtenidas pertenecen a la frontera eficiente. Al comparar NSGAII y la metodología de programación matemática, se observó que los resultados de los algoritmos evolutivos son similares y en algunos casos mejores que los obtenidos con está última.

Cuando se extienden los modelos de localización para considerar restricciones de capacidad, los problemas que se pueden abordar sin tener complicación son de tamaño reducido. En general, cuando las restricciones de capacidad son demasiado exigentes, es probable que la metodología de programación matemática no pueda obtener una buena aproximación de la frontera eficiente.

Utilizando la metodología de programación matemática, se analizó la red de acopio y almacenamiento de café colombiano. Con la aproximación de frontera eficiente, se determinó que la configuración actual no es eficiente, y que puede ser mejorada para reducir sus costos de operación sin deteriorar la cobertura que ofrece, o que puede mejorarse la cobertura que ofrece a uno costo inferior al actual. Además, se determinó el *trade off* existente entre el costo y el servicio ofrecido a los caficultores. Inicialmente, mejorar el servicio ofrecido por las soluciones de bajo costo no impone mayores sobrecostos en la operación del sistema. Pero, cuando la cobertura que se busca es cercana al máximo posible cualquier mejora es bastante costosa.

Se identificaron distintas formas de continuar con en el trabajo aquí presentado, básicamente con la mejora de las herramientas de solución desarrolladas, con la aplicación de nuevos

métodos de solución y con el desarrollo de modelos que permitan abordar situaciones más complejas y con otros criterios de decisión.

# Bibliografía


[1] M.C. Agar and S. Salhi. Lagrangean heuristics applied to a variety of large capaicrated plant locations problems. *Journal of the Operational Research Society*, 49(1072–1084), 1998.

[2] C.H. Aikens. Facility location models for distribution planning. *European Journal of Operational Research*, 22(263–279), 1985.

[3] A.P. Antunes. Location analysis helps manage solid waste in central Portugal. *Interfaces*, 29(4):32–43, 1999.

[4] J. Badhury, R. Batta, and J.H. Jaramillo. On the use of genetic algorithms to solve location problems. *Computers & Operations Research*, 29:761–779, 2002.

[5] M.A. Badri, A.K. Mortagy, and C.A. Alsayed. A multi-objective model for locating fire stations. *European Journal of Operational Research*, 110:243–260, 1998.

[6] B.M. Baker and M.A. Ayechew. A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 30:787–800, 2003.

[7] M.L. Balinski. Integer programming: Methods, uses and computation. *Management Science*, 12:253–313, 1965.

[8] D. Beasley, D. R. Bull, and R.R. Martin. An overview of genetic algorithms: Part 1, fundamental. *University Computing*, 15(2):58–69, 1993. Recuperado de `http://citeseer.nj.nec.com/16527.html`.

[9] D. Beasley, D. R. Bull, and R.R. Martin. An overview of genetic algorithms: Part 2, research topics. *University Computing*, 15(4):170–181, 1993. Recuperado de `http://citeseer.nj.nec.com/beasley93overview.html`.

[10] J.E. Beasley. Lagrangean heuristics for location problems. *European Journal of Operational Research*, 65:383–399, 1993.

[11] J.E. Beasley. Lagrangian heuristics for location problems. *European Journal of Operational Research*, 65:383–399, 1993.

[12] J.E. Beasley. Population heuristics. Handbook of applied optimization, P.M. Pardalos y M.G.C. Resende (eds). Oxford University Press, 2000. Recuperado de `http://citeseer.nj.nec.com/beasley99population.html`.

[13] J.E. Beasley. OR-notes: Facility location, 2003. Recuperado de `http://mscmga.ms.ic.ac.uk/jeb/or/facloc.html`.

[14] J.E. Beasley and P.C. Chu. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94:392–404, 1996.

[15] J.E. Beasley and P.C. Chu. A genetic algorithm for the generalised assignement problem. *Computers & Operations Research*, 24(1):17–23, 1997.

[16] J. Bramel and D. Simchi-Levi. A location based heuristic for general routing problems. *Operations Research*, 43(4):649–660, 1995.

[17] J. Bramel and D. Simchi-Leviy. *The Logic of Logistics*. Springer, 1997.

[18] M.L. Brandeau and S.S. Chiu. An overview of representative problems in location research. *Management Science*, 35:645–674, 1989.

[19] C.A. Coello C., G.A. Lamont, and D.A. Van Veldhuizen. *Evolutionary Algorithms for solving multi-objective Problems*. Kluwer, New York, 2002.

[20] C.H. Chung. Recent applications of the maximal covering location problem (M.C.L.P.). *Journal of the Operational Research Society*, 37(8):735–746, 1986.

[21] C. A. Coello Coello. An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends. In P.J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 3–13, Washington D.C., 1999. IEEE Press. Recuperado de `http://citeseer.nj.nec.com/coellocoello99updated.html`.

[22] W.W. Cooper, L.M. Seiford, and K. Tone. *Data Envelopment Analysis*. Kluwer, Boston, 2000.

[23] G. Cornuejols, M.L. Fisher, and G.L. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23:789–810., 1977.

[24] G. Cornuejols, R. Sridharan, and J.M. Thizy. A comparison of heuristics and relaxations for the capacitated plan location problem. *European Journal of Operational Research*, 50:280–297, 1991.

[25] M.J. Cortinhal and M.E. Captivo. Genetic algorithms for the single source capacitated location problem: a computacional study. In *MIC'2001 -4th Metaheuristics International Conference*, pages 355–359, Porto, Portugal, 2001. Recuperado de `http://143.129.203.3/eume/MIC2001/MIC2001_355_360.pdf`.

[26] J. Current, M. Daskin, and D. Schilling. *Facility Location: a survey of applications and methods*, chapter 3, Discrete Network Location Models, pages 86–122. Springer Verlag, New York, 1995.

[27] J. Current, H. Min, and D . Schilling. Multiobjective analysis of facility location decisions. *European Journal of Operational Research*, 49:295–307, 1990.

[28] P. Czyzak and A. Jaszkiewicz. Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7:34–47, 1998.

[29] M.S. Daskin. Rationalizing tool selection in a flexible manufacturing system for sheet metal products. *Operations Research*, 38:1104–1115, 1990.

[30] M.S. Daskin. *Network and Discrete Location. Models, Algorithms and Applications.* Wiley, New York, 1995.

[31] M.S. Daskin and S.H. Owen. Strategic facility location: a review. *European Journal of Operational Research*, 111:423–447, 1998.

[32] Comision de Ajuste a la Institucionalidad Cafetera. *El Café capital social y estratégico.* Comision de Ajuste a la Institucionalidad Cafetera, Bogota, 2002.

[33] FederaciónÑacional de Cafeteros de Colombia. Participación de las cooperativas en el mercado cafetero durante el año civil 2001. Bogotá, 2001.

[34] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India., 2000.

[35] K. Deb and N. Srinivas. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994. Recuperado de `http://citeseer.nj.nec.com/srinivas94multiobjective.html`.

[36] H Delmaire, J.A. Díaz, E. Fernández, and M. Ortega. Reactive GRASP and tabu search heuristics for the single source capacitated plant location problem. *INFOR*, 37(3):194–225, 1999.

[37] M.D. Devine and W.G. Lesso. Models for the minimun cost development of offshore oil fields. *Management Science*, 18:378–387, 1972.

[38] J.A. Díaz. *Algorithmic approaches for the single source capacitated plant location problem.* Tesis doctoral, Universidad Politécnica de Cataluña, 2001.

[39] J.A. Díaz and E. Fernández. A branch and price algorithm for the single source capacitated plant location problem. *Journal of the Operational Research Society*, 53:728–740, 2002.

[40] D.J. Eaton, M.S. Daskin, D. Simmons, B. Bulloch, and G. Jansma. Determining medical service vehicle deployment in Austin, Texas. *Interfaces*, 15:96–108, 1985.

[41] M. Ehrgott and X. Gandibleux. An annotated bibliography of multiobjective combinatorial optimization. Reporte, universidad de kaiserslautern, 2000. Recuperado de `http://citeseer.nj.nec.com/ehrgott00annotated.html`.

[42] D. Erlenkotter. A dual-based procedure for uncapacitated facility location. *Operations Research*, 26:992–1009, 1978.

[43] E. Fernandez and J. Puerto. Multiobjetive solution of uncapacitated plant location problem. *European Journal of Operational Research*, 145:509–529, 2003.

[44] R. Galvaõ, L.G. Acosta-Espejo, and B. Boffey. A comparison of lagrangean and surrogate relaxations for the maximal covering location problem. *European Journal of Operational Research*, 124:377–389, 200.

[45] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Adisson-Wesley, Reading M.A., 1989.

[46] M.P. Hansen. Tabu Search in Multiobjective Optimisation : MOTS. In *Proceedings of the 13th International Conference on Multiple Criteria Decision Making (MCDM'97)*, 1997. Recuperado de Recuperado de `http://citeseer.nj.nec.com/hansen97tabu.html`.

[47] M. Hapke, A. Jaszkiewicz, and J. Zak. The design of the physical distribution system with the application of the multiple objective mathematical programming. case study. In T. Traskalik and J. Michnik, editors, *Multi-objective Programming and Goal Programming. Recent Developments*, pages 297–309. Physica-Verlag, Heidelberg, 2002. Recuperado de `http://www-idss.cs.put.poznan.pl/~jaszkiewicz/`.

[48] K.S. Hindi and K Pienkons. Efficient solution of large scale, single source, capacitated plant location problems. *Journal of the Operational Research Society*, 50:268–274, 1999.

[49] M. Hoefer. Ufllib, benchmark instances for the uncapacitated facility location problem, Enero 2003. Recuperado de `http://www.mpi-sb.mpg.de/units/ag1/projects/benchmarks/UflLib/`.

[50] K. Holmberg, M. Ronnqvist, and D. Yuan. An exact algorithm for the capacitated facility location problems with single sourcing. *European Journal of Operational Research*, 113:544–559, 1999.

[51] R.G. Ichiro and L.A. Lorena. A constructive genetic algorithm for the maximal covering location problem. In *MIC2001 4th Metaheuristics International Conference*, Porto, Portugal, 2001.

[52] V. Jayaraman. A multi-objective logistics model for a capacitated service facility problem. *International Journal of Physical Distribution & Logistics*, 29(1):65–81, 1999.

[53] D.F. Jones, S.K. Mirrazavi, and M. Tamiz. Multi–objective meta–heuristics: An overview of the current state–of–the–art. *European Journal of Operational Research*, 137:1–9, 2002.

[54] J. Knowles and D. Corne. M-PAES: A memetic algorithm for multiobjective optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, pages 325–332, IEEE Press. Recuperado de `http://citeseer.nj.nec.com/324901.html`.

[55] J.D. Knowles and D. Corne. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation. In P.J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 98–105, Washington D.C., 1999. IEEE Press. Recuperado de `http://citeseer.nj.nec.com/knowles99pareto.html`.

[56] J.D. Knowles and D. Corne. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000. Recuperado de `http://citeseer.nj.nec.com/knowles00approximating.html`.

[57] J.D. Knowles and D. Corne. On metrics for comparing non-dominated sets, 2002. Recuperado de `http://citeseer.nj.nec.com/knowles01metrics.html`.

[58] M. Koksalan and A.B. Keha. Using genetic algorithms for single machine bicriteria scheduling problems. *European Journal of Operational Research*, 145:543–556, 2003.

[59] J. Kratica, V. Filipovic, and D. Tosic. Solving the uncapacitated warehouse location problem by SGA with add-heuristic, 1998. Recuperado de `http://citeseer.nj.nec.com/436990.html`.

[60] J. Kratica, D. Tosic, V. Filipovic, and I. Liubic. Solving the simple plant location problem by genetic algorithm. *RAIRO Operations Research*, 35:127–142, 2001.

[61] J. Kratica, D. Tosic, V. Filipovic, and I. Ljubic. Solving the simple plant location problem by genetic algorithm, 2001. Recuperado de `http//citeseer.nj.nec.com/kratica01solving.html`.

[62] A.A. Kuehn and M.J. Hamburger. A heuristic program for locating warehouses. *Management Science*, 9(4):643–666, 1963.

[63] B.M. Kuhmwala. An effciente branch and bound algorithm for the warehouse location problem. *Management Science*, 18:718–731, 1972.

[64] N.K. Kwak, M.C. Hemer, and M.J. Schiniederjans. An application of goal programming to resolve a site location problem. *Interfaces*, pages 65–72, 1982.

[65] M. Marsh and D. Schilling. Equity measurement in facility location analysis. a review and framework. *European Journal of Operational Research*, 74:1–17, 1994.

[66] G. Mavrotas and D. Diakoulaki. A branch and bound algorithm for mixed zero–one multiple objective linear programming. *European Journal of Operational Research*, 107:530–541, 1998.

[67] A.L. Medaglia and S.C.Fang. A genetic-based framework for solving (multicriteria) weighted matching problems. *European Journal of Operational Research*, 149(1):77–101, 2003.

[68] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin, 1996.

[69] H. Min. A multiobjetive retail service location model for fast food restaurants. *Omega*, 15:429–441, 1987.

[70] P.B. Mirchandani and R.L. Francis. *Discrete Location Theory*. Wiley, New York, 1990.

[71] L.K. Nozick. The fixed charge facility location problem with coverage restrictions. *Transportation Research Part E*, 37:281–296, 2001.

[72] L.K. Nozick and M.A. Turnquist. Inventory, transportation, service quality and the location of distribution centers. *European Journal of Operational Research*, 129:362–371, 2001.

[73] J.P. Osleeb and S.J. Ratick. A mixed integer and multiple objective programming model to analyze coal handling in New England. *European Journal of Operational Research*, 83:302–313, 1983.

[74] M.G.C. Resende. Computing approximate solutions of the maximum covering problem with GRASP. *Journal of Heuristics*, 4:161–171, 1998. Recuperado de `http://www.research.att.com/~mgcr/doc/gmcov.ps.Z`.

[75] C Revelle. Facility siting and integer–friendly programming. *European Journal of Operational Research*, 65:147–158, 1993.

[76] C. Revelle and G. Laporte. The plant location problem: new models and research prospects. *Operations Research*, 44:864–874, 1996.

[77] C. Revelle, D. Marks, and J.C. Liebman. An analysis of private and public sector location models. *Management Science*, 16:692–707, 1970.

[78] M. Ronnqvist, S. Tragantalerngsak, and J.Holt. A repeated matching heuristic for the single source capacitated facility location problem. *European Journal of Operational Research*, 116:51–68, 1999.

[79] G.T. Ross and R.M. Soland. A multicriteria approach to the location of public facilities. *European Journal of Operational Research*, 4:307–321, 1980.

[80] S.M. Ross. *Probabilidad y Estadistica para Ingenieros*. Mc Graw Hill, México, 2 edition, 2000.

[81] J.K. Sankaran and N.R. Srinivasa-Raghavan. Locating and sizing plants for bottling propane in south India. *Interfaces*, 27(6):1–15, 1997.

[82] G. Zhou and M. Gen. Genetic algorithm approach on multi-criteria minimum spanning tree problem. *European Journal of Operational Research*, 114:141–152, 1999.

[83] E. Zitzler, M. Laumanns, L. Thiele, C.M. Fonseca, and V.G. Fonseca. Why quality assessment of multiobjective optimizers is difficult, 2002. Recuperado de `http://citeseer.nj.nec.com/525706.html`.

[84] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms — A comparative case study. *Parallel Problem Solving from Nature–PPSN–V*, pages 292–303, 1998. Recuperado de `http://citeseer.nj.nec.com/zitzler98multiobjective.html`.

[85] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review, Junio 2002. Recuperado de `http://citeseer.nj.nec.com/zitzler02performance.html`.

# Apéndice A

# Problema de Localización de Máxima Cobertura

Conocido como MCLP (del inglés, Maximum Covering Location Problem), es uno de los modelos de localización más importantes. El objetivo del MCLP es localizar un número predeterminado de instalaciones, de manera que se maximice la demanda cubierta.

Ha sido aplicado en diferentes áreas, tales como: localización de vehículos de emergencia [40], selección de herramientas para sistemas de manufactura flexible [29], diseño de redes de telecomunicaciones [74], entre otras. En [20] se presenta una amplia reseña de sus aplicaciones. Para su solución se han utilizado diferentes técnicas, entre las que se encuentran: heurísticos lagrangianos [44], heurísticos miopes [30] y técnicas metaheurísticas como: algoritmos genéticos [4, 51] y GRASP [74].

Sean $\mathcal{I} = \{1, ..., m\}$ el conjunto de lugares donde se pueden localizar las bodegas, $\mathcal{J} = \{1, ..., n\}$ el conjunto de clientes, $d_j$ la demanda del cliente $j$, $h_{ij}$ la distancia entre la bodega $i$ y el cliente $j$, $p$ el número de instalaciones a abrir.

Para evaluar la cobertura, se definen la distancia maxima de cobertura $D_{max}$ (i.e., la distancia a la cual se considera que un cliente esta cubierto por una instalación), y el conjunto de bodegas que pueden cubrir al cliente $j$, $\mathcal{Q}_j = \{i \in \mathcal{I} : h_{ij} \leq D_{max}\}$. .

Las variables de decision son:

$$y_i = \begin{cases} 1, & \text{si se abre la instalación } i, \\ 0, & \text{de lo contrario.} \end{cases}$$

Además, se define la variable $z_j$ que indica si un cliente está cubierto o no.

$$z_j = \begin{cases} 1, & \text{si el cliente j está cubierto,} \\ 0, & \text{de lo contrario.} \end{cases}$$

Utilizando la formulación presentada en [26], el MCLP se define así:

$$\text{máx} \sum_{j \in \mathcal{J}} d_j z_j \tag{A.1}$$

Sujeto a:

$$\sum_{i \in \mathcal{I}} x_{ij} \geq z_j \quad , \quad j \in \mathcal{J} \tag{A.2}$$

$$\sum_{i \in \mathcal{I}} y_i = p \quad , \quad j \in \mathcal{J} \tag{A.3}$$

$$y_i \in \{0,1\}, \quad i \in \mathcal{I} \tag{A.4}$$

$$z_j \in \{0,1\}, \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{A.5}$$

La función objetivo (A.1) busca maximizar la demanda total cubierta. Las restricciones (A.2) garantizan que la demanda de cada cliente se cuenta sólo cuando se ha abierto una instalación que lo puede cubrir. La restricción (A.3) hace que se abran exactamente $p$ instalaciones. Finalmente, en (A.4) y (A.5) se definen las variables de decisión del problema y las variables de cobertura $z_i$.

# Apéndice B

# Problema de Localización sin Restricciones de Capacidad

El problema de localización sin restricciones de capacidad, conocido como UFLP (Uncapacitated Facility Location Problem), busca minimizar el costo total de satisfacer la demanda de los clientes, mediante la apertura de algunas instalaciones escogidas de un conjunto de lugares potenciales de ubicación. Típicamente, existen dos componentes de costo, uno fijo asoiciado con la operación y apertura de las instalaciones y otro variable asociado con la asignación de los clientes a las instalciones abiertas. Este problema también es conocido como el SPLP (del inglés, Simple Plant Location Problem) o UFCLP (del inglés, Uncapacitated Fixed Charge Location Problem).

Su aplicación en la toma de decisiones es bastante amplia, ha sido utilizado para diseñar sistemas de distribución [2], diseñar métodos heurísticos para el ruteo de vehículos [16], localizar cuentas bancarias [23], localizar de bodegas [62], entre otras.

Para su solución se han propuesto técnicas muy diversas, tales como: heurísticos simples [62], heurísticos lagrangianos [10], heurísticos duales [42], algoritmos de *Branch and Bound* [63] y algoritmos genéticos [4, 61]. Para mayor profundidad se puede recurrir a [70], donde se presenta un análisis de las principales técnicas utilizadas para resolver el UFLP.

Sean $\mathcal{I} = \{1, ..., m\}$ el conjunto de lugares donde se pueden localizar las bodegas, $f_i$ el costo fijo de abrir u operar una bodega en el lugar $i$, $\mathcal{J} = \{1, ..., n\}$ el conjunto de clientes, $d_j$ la demanda del cliente $j$, $c_{ij}$ el costo de asignar el cliente $j$ a la bodega $i$.

Las variables de decision son:

$$y_i = \begin{cases} 1, & \text{si se abre la bodega } i, \\ 0, & \text{de lo contrario.} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{si el cliente } j \text{ es atendido por la bodega } i, \\ 0, & \text{de lo contrario.} \end{cases}$$

$$\text{mín} \quad z_1 = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} x_{ij} + \sum_{i \in \mathcal{I}} f_i y_i \tag{B.1}$$

Sujeto a:

$$\sum_{i\in\mathcal{I}} x_{ij} = 1 \quad , \quad j \in \mathcal{J} \tag{B.2}$$

$$x_{ij} \leq y_i \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{B.3}$$

$$x_{ij} \in \{0,1\}, \quad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{B.4}$$

$$y_i \in \{0,1\}, \quad i \in \mathcal{I} \tag{B.5}$$

La función objetivo (B.1) representa el costo total . Las restricciones (B.2) garantizan que se satisfaga la demanda de cada cliente. Las restricciones (B.3) garantizan que los clientes sean asignados a bodegas abiertas. En las restricciones (B.4) y (B.5) se definen las variables de decisión del problema .

# Apéndice C

# Prueba de Wilcoxon

Esta prueba no parametrica, también conocida como prueba de rango o Mann-Wihtney, sirve para probar la hipótesis de que dos muestras independientes pertenecen a la misma población.

Sean $A = \{a_1, a_2, \ldots, a_m\}$ y $B = \{b_1, b_2, \ldots, b_n\}$ las dos muestras independientes a las que se le quiere hacer la prueba. Entonces se organizan $A$ y $B$ en una secuencia $C$, tal que los datos estén en orden descendente. Por ejemplo, si

$$A = \{0{,}7424, 0{,}7200, 0{,}7454, 0{,}7413, 0{,}7760, 0{,}7104, 0{,}7507, 0{,}6913, 0{,}7806, 0{,}7927\}(m = 10),$$

$$B = \{0{,}7559, 0{,}7505, 0{,}7472, 0{,}7529, 0{,}7563, 0{,}7549, 0{,}7370, 0{,}7612, 0{,}7649, 0{,}7564\}(n = 10)$$

, se obtiene

$$C = \{0{,}7927, 0{,}7806, 0{,}7760, \underline{0{,}7649}, \underline{0{,}7612}, \underline{0{,}7564}, \underline{0{,}7563}, \underline{0{,}7559}, \underline{0{,}7549}, \underline{0{,}7529}, 0{,}7507,$$

$$\underline{0{,}7505}, \underline{0{,}7472}, 0{,}7454, 0{,}7424, 0{,}7413, \underline{0{,}7370}, 0{,}7200, 0{,}7104, 0{,}6913, \}$$

(Los numeros subrayados pertenecen a $A$)

Luego se asigna un rango entre 1 y $N(= n + m)$ a cada uno de los elementos de C. En caso de un empate se le asigna el rango medio a las observaciones empatadas. Para el caso del ejemplo anterior se obtiene el siguiente rango,

$$R = \{1, 2, 3, \underline{4}, \underline{5}, \underline{6}, \underline{7}, \underline{8}, \underline{9}, \underline{10}, 11, \underline{12}, \underline{13}, 14, 15, 16, \underline{17}, 18, 19, 20\}$$

Luego se suman los rangos de los datos proveniente de la muestra $A$,

$$S = 91$$

Este suma distribuye aproximadamente normal con media $m(m+n+1)/2$ y varianza $mn(m+n+1)/12$. Mediante la estandarización de $S$ se obtiene el estadístico $T$. De este modo,

$$T = \frac{S - m(m+n+1)/2}{mn(m+n+1)/12}$$

Es claro que $T$ sigue aproximadamente una distribución normal estándar. $T$ puede ser utilizado para probar la hipótesis nula de que los dos muestras provienen de la misma población

con un nivel de significancia $\alpha$. La hipótesis nula se rechazará si $T \leq -T_{\alpha/2}$ o $T \geq T_{\alpha/2}$, siendo $T_{\alpha/2}$ el percentil $1 - \alpha/2$ de una distribución normal estándar.

En el ejemplo anterior, se tiene que $T = -1{,}06$. Si tomamos $\alpha = 0{,}05$, entonces $T_{0{,}025} = 1{,}96$. Puesto que $T = 1{,}06 \geq -1{,}96 = -T_{0{,}025}$ y $T = 1{,}06 \leq 1{,}96 = -T_{0{,}025}$, se acepta, a un nivel de significancia del 5 %, la hipótesis de que las dos muestras provienen de la misma población.

Para mayores detalles sobre la prueba se pueden consultar [22, 80].

# Apéndice D

# Tiempos de Ejecución Procedimientos de Programación Matemática

Los tiempos de ejecución que se reportan a continuación, corresponden a diferentes servidores de los cuales no se conoce la configuración. Todos los problemas se resolvieron usando AMPL/XpressMP 12.13. Disponible en el servidor de optimización NEOS `http://www-neos.mcs.anl.gov/neos/solvers/MILP:XPRESS-AMPL/solver-www.html`

| Problema | Tiempo de Ejecución (HH:MM:SS) |
|---|---|
| A10-25C1 | 0:00:31 |
| A10-25C2 | 0:00:25 |
| A10-25C3 | 0:00:47 |
| A10-25C4 | 0:00:27 |
| A10-25C5 | 0:00:25 |
| A10-25C6 | 0:00:27 |
| A30-75C1 | 0:07:49 |
| A30-75C2 | 0:07:50 |
| A30-75C3 | 0:08:30 |
| A30-75C4 | 0:11:12 |
| A30-75C5 | 0:12:56 |
| A30-75C6 | 1:07:38 |
| A50-150C1 | 0:32:11 |
| A50-150C2 | 0:55:13 |
| A50-150C3 | 1:12:24 |
| A50-150C4 | 1:24:39 |
| A50-150C5 | 6:21:17 |
| A50-150C6 | 4:39:00 |

Tabla D.1: Problemas de localización sin restricciones de capacidad. Tiempos de ejecución. Problemas Tipo A.

| Problema | Tiempo de Ejecución (HH:MM:SS) |
|---|---|
| B10-25C1 | 0:00:20 |
| B10-25C2 | 0:00:26 |
| B10-25C3 | 0:00:26 |
| B10-25C4 | 0:00:36 |
| B10-25C5 | 0:00:30 |
| B10-25C6 | 0:00:25 |
| B30-75C1 | 0:04:05 |
| B30-75C2 | 0:09:16 |
| B30-75C3 | 0:09:01 |
| B30-75C4 | 0:03:41 |
| B30-75C5 | 0:25:35 |
| B30-75C6 | 0:09:55 |
| B50-150C1 | 0:17:25 |
| B50-150C2 | 0:54:36 |
| B50-150C3 | 1:46:23 |
| B50-150C4 | 1:29:02 |
| B50-150C5 | 1:35:01 |
| B50-150C6 | 3:35:15 |

Tabla D.2: Problemas de localización sin restricciones de capacidad. Tiempos de ejecución. Problemas Tipo B.

| Problema | Tiempo de Ejecución (HH:MM:SS) |
|---|---|
| A10-25F1R1.5 | 1:13:27 |
| A10-25F1R2 | 4:19:14 |
| A10-25F1R3 | 2:51:30 |
| A10-25F1R5 | 1:22:26 |
| A10-25F2R1.5 | 24:55:1 |
| A10-25F2R2 | 5:12:42 |
| A10-25F2R3 | 3:08:21 |
| A10-25F2R5 | 3:08:27 |
| B10-25F1R1.5 | 3:48:09 |
| B10-25F1R2 | 2:30:38 |
| B10-25F1R3 | 0:17:36 |
| B10-25F1R5 | 0:35:15 |
| B10-25F2R1.5 | 0:09:03 |
| B10-25F2R2 | 0:03:17 |
| B10-25F2R3 | 0:03:48 |
| B10-25F2R5 | 0:03:45 |

Tabla D.3: Problemas de localización con restricciones de capacidad. Tiempos de ejecución, Problemas de Tamaño 10-25.

| Problema | Tiempo de Ejecución (HH:MM:SS) |
|---|---|
| A25-50F1R1.5 | 35:13:19 |
| A25-50F1R2 | 36:54:55 |
| A25-50F1R3 | 57:54:52 |
| A25-50F1R5 | 43:14:02 |
| A25-50F2R1.5 | 22:41:26 |
| A25-50F2R2 | 22:47:07 |
| A25-50F2R3 | 20:39:36 |
| A25-50F2R5 | 17:14:27 |
| B25-50F1R1.5 | 34:54:48 |
| B25-50F1R2 | 36:49:13 |
| B25-50F1R3 | 8:33:42 |
| B25-50F1R5 | 30:06:21 |
| B25-50F2R1.5 | 37:18:48 |
| B25-50F2R2 | 37:58:03 |
| B25-50F2R3 | 17:00:27 |
| B25-50F2R5 | 20:10:56 |

Tabla D.4: Problemas de localización con restricciones de capacidad. Tiempos de ejecución, Problemas de Tamaño 25-50.