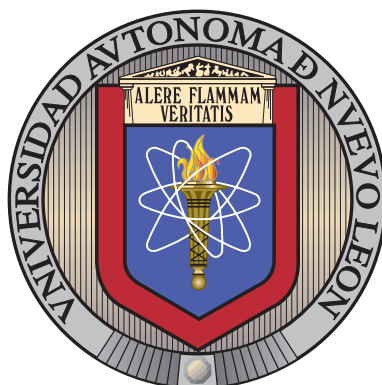


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS

DIVISIÓN DE ESTUDIOS DE POSGRADO



PRICING ON THE TREE OF HUBS LOCATION
PROBLEM

POR

CARLOS EDUARDO CORPUS CARDONA

EN OPCIÓN AL GRADO DE

MAESTRÍA EN CIENCIAS

CON ORIENTACIÓN EN MATEMÁTICAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

FEBRERO 2021

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS

DIVISIÓN DE ESTUDIOS DE POSGRADO



PRICING ON THE TREE OF HUBS LOCATION
PROBLEM

POR

CARLOS EDUARDO CORPUS CARDONA

EN OPCIÓN AL GRADO DE
MAESTRÍA EN CIENCIAS
CON ORIENTACIÓN EN MATEMÁTICAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

FEBRERO 2021

Universidad Autónoma de Nuevo León
Facultad de Ciencias Físico-Matemáticas
División de estudios de posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis “Pricing on the Tree of Hubs Location Problem”, realizada por el alumno Carlos Eduardo Corpus Cardona, con número de matrícula 1585468, sea aceptada para su defensa como opción al grado de Maestría en Ciencias con Orientación en Matemáticas.

El Comité de Tesis

Dr. Jose Fernando Camacho Vallejo
Director

Dr. Víctor Blanco Izquierdo
Co-Director

Dr. Omar Jorge Ibarra Rojas
Revisor

Vo. Bo.

Dr. Persona de visto bueno
División de estudios de posgrado

San Nicolás de los Garza, Nuevo León, Febrero 2021

AGRADECIMIENTOS

Esta tesis fue creada durante unos tiempos muy extraños y no sería posible sin el apoyo no solo de mi coordinador de tesis, sino además de cada una de las personas que de alguna forma me ha brindado su apoyo durante esta travesía, esta tesis es de ustedes.

Gracias Dr. Camacho por sus consejos, no solo en lo académico sino también en lo personal. Por ayudarme con su conocimiento, los comentarios y las críticas que me han ayudado a forjarme de manera profesional.

Al Dr. Víctor Blanco de la Universidad de Granada, el cual, aunque mi estancia fue breve y llena de incertidumbre siempre tuvo el tiempo para apoyarme en mi estancia.

A mi familia, gracias por aguantar todos esos días que me encontraba estresado y ausente, a mi madre, a mis hermanas y mi padre que me apoyaron siempre de manera incondicional. Muchas gracias por todo.

A mis amigos que me han apoyado siempre durante todo este proceso, por sus palabras de aliento, por su preocupación y su ayuda incondicional.

Y agradecer a la Universidad Autónoma de Nuevo León en especial a la Facultad de Ciencias Físico Matemáticas y al Centro de Investigación en Ciencias Físico Matemáticas por todo el apoyo que se me ha sido brindado durante mi tiempo ahí. Gracias al Consejo Nacional de Ciencia y Tecnología por sus apoyos económicos durante esta aventura.

ÍNDICE GENERAL

Agradecimientos	IV
1. Introducción	1
1.1. Objetivo	3
1.2. Estructura de la tesis	3
2. Revisión de Literatura	4
2.1. Antecedentes Matemáticos	11
2.2. Programación Binivel	11
2.3. Teoría de Grafos	12
3. Descripción del problema	18
3.1. Reformulación del modelo de programación binivel a uno MILP	21
4. Un algoritmo coevolutivo para el BTHLPwP	27
4.1. Generación de soluciones	29
4.1.1. Generación de los árboles	30
4.1.2. Resolución nivel inferior	31
4.1.3. Generación de los precios	32
4.2. Función Objetivo del Líder	34
4.3. Evolución de los precios	34

4.4. Selección elitista de un nuevo árbol.	37
4.5. Coevolución	38
5. Experimentación computacional y resultados	39
6. Conclusiones y trabajo a futuro	45
A. Ciudades del CAB dataset	51

CAPÍTULO 1

INTRODUCCIÓN

Cada día se vive en un mundo más conectado. La globalización exige que exista una conexión entre los países en el aspecto comercial y en el aspecto social. Por ejemplo, para el primer aspecto se necesita una buena conexión para enviar y recibir los insumos, bienes, productos, etc. Mientras que el segundo aspecto es importante debido a los fenómenos migratorios, de turismo, entre otros. Ante estos retos, se han tenido que establecer y reforzar las formas que se interrelacionan los diferentes organismos de dichos países.

Sin embargo existen muchos problemas derivados del daño o extravío de bienes, cobro de cuotas, la congestión de usuarios en la red al requerir un servicio. Y es que el hecho de contar con una buena red de transporte y comunicación es fundamental para los retos que surgen en un mundo globalizado. Por ejemplo, la red de vuelos con la que una aerolínea cuenta es crucial para la conectividad que un país tiene con el resto del mundo. O por otro lado, el envío y recepción de bienes en tiempo y forma es necesaria para el funcionamiento de una empresa o negocio.

Para afrontar muchos de estos retos, los organismos encargados de crear estas redes de transporte utilizan centros especializados llamados hubs. Los hubs se encargan de concentrar los bienes desde un nodo origen y redistribuirlos hacia su destino. Se puede pensar en muchas ventajas de la utilización de hubs tales como:

- Permite la creación de servicios de distribución más eficientes permitiendo que una gran cantidad de personas lo utilicen.
- Mucha más frecuencia de los servicios.
- Costos accesibles a los usuarios pero manteniendo infraestructuras de muy buena calidad.

Pero no todo depende solamente de instalar estos hubs, la localización de estos juega un papel muy importante ya que este permitirá el uso óptimo de la red. Aunque la

distancia de las conexiones dentro de la red de hubs pueda ser menor, la distancia total de viaje puede ser mayor ya que no existe garantía de que el movimiento entre la red de hubs sea menor que en una conexión directa [1]. Es por ello que puede ser muy complicado determinar la localización óptima de estos hubs dentro de una red[2].

Aquí entra una de la muchas ramas de la investigación de operaciones, la teoría de localización ha sido ampliamente estudiada por años ([3], [4], [5], [6]), la idea detrás de esta teoría es la ubicación óptima de instalaciones de manera que los costos de transporte sean mínimos, sujeto a ciertos factores que afectan directamente a la ubicación.

A través de los años se han hecho grandes esfuerzos en el área combinando la teoría de localización con hubs, permitiendo a investigadores crear una teoría muy solida que justifica los avances realizados. Véase ([7], [8], [9], [10]).

En esta tesis de maestría nos enfocamos en resolver el problema de localización del árbol de hubs (Tree of Hubs Location Problem, THLP) una variante del problema clásico localización de hubs (Hub Location Problem). El THLP consiste en que dado un número de clientes y bienes que se envían del uno al otro y un número de sitios potenciales para localizar hubs, se tiene que decidir cuales de estos abrir y construir una red en forma de árbol de forma que satisfagan a los clientes a costo mínimo. La principal ventaja de que esta red sea construida en forma de árbol, es que al no existir ciclos, el camino que sigue un bien es único al mismo tiempo que el costo de utilizar la red es mínimo.

En este problema existen dos agentes que están involucrados en la creación y el uso de la red. Por un lado tenemos al agente encargado del diseño de la estructura de la red y por el otro, el usuario que utiliza la red para transportar sus productos o bienes. El detalle es, que por lo general los objetivos de estos dos agentes pueden entrar en conflicto. Mientras el agente que crea la red busca maximizar la ganancia obtenida al utilizarla, el segundo agente busca minimizar el costo al utilizar la red de hubs.

Es claro que la decisión tomada por el agente con mayor jerarquía tiene efecto directo en la decisión que tomará el agente con menor jerarquía, se puede pensar en algunas situaciones donde suceda este sistema de jerarquías por ejemplo, organismos gubernamentales y los ciudadanos, aerolíneas y usuarios, empresas de paquetería y clientes, entre otras.

Para estudiar situaciones con dos tomadores de decisiones con una jerarquía entre ellos, se emplea un área especial de la programación matemática llamada programación binivel, esta área tienen como peculiaridad que una de las restricciones del problema de optimización es otro problema de optimización. Al primer problema se refiere como el

nivel superior (o problema del líder) mientras que el segundo problema se refiere al nivel inferior (o problema del seguidor).

Para esta tesis se propone un enfoque binivel en donde el primer agente (el líder) no solo decide la estructura de la red de hubs (abrir hubs y los arcos conectado esto en forma de árbol) si no que además decide los precios que el usuario tiene que pagar por usar la red. Por otro lado el segundo agente (o seguidor) decide usar o no esta red y como enviar los productos a través de ella.

Es evidente que el uso de la red va a impactar en el objetivo del líder, ya que si los precios que coloca el líder son muy elevados, el seguidor no utilizará la red por lo que el líder no obtendrá ningún beneficio, mientras que si los precios son pequeños y los usuarios si utilizan la red, la ganancia no será tan significativa y podría no cubrir los costos de construcción de la red.

1.1 OBJETIVO

El principal objetivo de la tesis es estudiar los problemas de localización de árboles de concentradores con precios como variables proponiendo una metodología para obtener soluciones de buena calidad con un costo computacional relativamente bajo. Para el problema se reformula el problema binivel a un solo nivel utilizando las condiciones de Karush-Kuhn-Tucker además de analizando las propiedades del problema con la Teoría de Grafos y se resuelve utilizando optimizadores para instancias pequeñas mientras que para instancias más grandes se utiliza un algoritmo metaheurístico co-evolutivo.

1.2 ESTRUCTURA DE LA TESIS

Conforme al problema presentado, los capítulos de la tesis son los siguientes. En el capítulo 2 se presenta una revisión de literatura del tema además se presentan resultados de Teoría de Grafos analizando algunas de las propiedades que se presentan dentro del problema. En el capítulo 3 se presenta el modelo que representa la problemática mostrada, en el capítulo 4 se da una explicación de la metodología utilizada para obtener resultados, mientras que en el capítulo 5 se muestra los resultados de la experimentación computacional y por ultimo, en el capítulo 6 se presentan las conclusiones y posibles direcciones que pueden tomar este tema de investigación.

CAPÍTULO 2

REVISIÓN DE LITERATURA

La teoría de localización es una rama muy importante de la investigación operativa. La necesidad de ubicar en el mejor sitio posible a instalaciones que proveen servicios satisfaciendo las necesidades de usuarios ha atraído la atención de los investigadores. Uno de los primeros investigadores en esta teoría fue Alfred Weber. En su libro "Theory of the Location Industries" menciona algunos factores esenciales al momento de localizar una instalación [3]. Esta teoría tomó gran importancia durante el siglo 19, cuando en plena revolución industrial, la necesidad de abrir fábricas para satisfacer la demanda de productos era cada vez mayor.

En los problemas clásicos de localización de instalaciones se tiene un conjunto de clientes que desean que sus demandas sean satisfechas, ya sea por una o varias instalaciones. El problema consiste en determinar donde localizar las instalaciones y asignar los clientes de manera que los costos de apertura y de asignación sean minimizados [11].

Existen muchas variantes del problema de localización de instalaciones. Por ejemplo, el problema de localización de hubs (HLP, por sus siglas en inglés) es una extensión relativamente nueva para este tipo de problemas. Los hubs son instalaciones especiales cuya función es concentrar el flujo de productos de distintos orígenes y redistribuirlos hacia un destino [7]. Es así como el HLP consiste en localizar estas instalaciones llamadas hubs y asignar los nodos de demanda (creando una red de distribución), para después poder crear las rutas para trasladar los bienes de los distintos pares origen-destino optimizando una función objetivo basada en costos.

Aunque ya se presentaban los primeros esbozos del HLP en [12], no fue si no hasta que en 1986 se publicó el primer artículo del área, llamado "The Location of Interacting Hub Facilities" [13]. En ese artículo, se comienzan a sentar las bases de esta rama, junto con los primeros esbozos para demostrar lemas y teoremas. Después en [14], se formula el

mismo problema pero utilizando programación cuadrática entera para resolver casos más complejos.

Por otra parte, en [15] se brinda la primera formulación del HLP con múltiples asignaciones. A partir de ahí se han generado diversos trabajos que buscan explotar las características del modelo. Por ejemplo, en [16] se busca reducir el número de variables del problema usando algoritmos duales para el problema de localización sin capacidad. Además en [17], se utilizan técnicas heurísticas para resolver el problema de localización de p-hubs. En años recientes se han hecho muchos avances dentro de esta área, tales como [18] en donde se utiliza la descomposición de Benders para un problema de localización hubs semi-conectados, [19] en donde se utiliza un enfoque de optimización robusta para la localización única o múltiple de estos hubs, y [20] en donde se utiliza la relajación Lagrangiana para un problema de localización de hubs sin capacidades con ganancias.

Es claro que los avances del HLP han sido muy grandes lo cual ha llevado a que los investigadores consideren potenciales aplicaciones de este modelo, cada una, con objetivos distintos. Por ejemplo, el transporte tiende a enfocarse al flujo del producto y al tiempo de envío, mientras que en las telecomunicaciones se considera más que los costos de instalación de servidores sean lo menor posible. [21]. Gracias a las características únicas de estas aplicaciones, se ha permitido desarrollar nuevas variantes del HLP.

Una de las más estudiadas es el P-Hub Median Problem, en el cual el número de hubs a localizar se conoce a priori [22]. El P-Hub Center Problem consiste en localizar un número fijo de hubs de forma que el tiempo máximo de traslado entre cualquier par origen-destino sea el mínimo posible [2]. Otra variante es el Tree of Hubs Location Problem (THLP, por sus siglas en inglés). En este último problema se quiere localizar un número fijo de centros (parecido al P-Hub Median Problem) pero se busca que las conexiones entre ellos sean por medio de un grafo en forma de árbol.

Aunque la primera formulación formal del THLP fue propuesta en [23] y [24]. Existen trabajos previos donde se consideraban estructuras de red con forma de grafo de árbol. Por ejemplo, en [25] se diseña una red de servicios de datos digitales, la cual es necesario que tenga la forma de un árbol debido a la naturaleza de la red de datos. Después, en [26] se utiliza un método de relajación lagrangiana para resolver el problema de diseñar la red de servicios digitales. El principal objetivo de esos dos problemas es minimizar el costo fijo de instalación de hubs, mientras que en el THLP, se busca minimizar el costo

de rutear el flujo del producto a través de la red.

Bajo el mismo esquema donde la red de instalaciones abiertas debe formar un árbol podemos citar a [27], en donde se busca localizar unas instalaciones especiales llamadas "extensivas". Estas instalaciones tienen la peculiaridad de que no son una única instalación. Es decir, se conforman por subinstalaciones, las cuales están conectadas en forma de un árbol y tienen que ser colocadas como un subgrafo dentro de la red.

Por otro lado, en el diseño de las conexiones en las redes se busca conectar los nodos de tal manera que el costo asociado al flujo del producto sea mínimo. En [28] se resuelve un problema de comunicación óptima donde la red es del tipo mínimo árbol recubridor. En ese problema, se construye un árbol recubridor seleccionando los mejores arcos para que todos los nodos estén conectados a mínimo costo. Como se puede ver existen muchos aspectos a tomar a consideración cuando se diseña una red de hubs en forma de árbol, la localización y la conexión de los arcos son algunos de ellos

En los años recientes han habido grandes avances para resolver problemas de localización y diseño de redes en forma de árbol. En [29] se utiliza la descomposición de Benders pero agregando nuevos esquemas de corte para el THLP propuesto en [23]. Por otro lado, en [30] se propone un algoritmo de claves aleatorias sesgadas (BRKGA, por sus siglas en inglés) para resolver el mismo problema que aparece en [23]. Al comparar los resultados se muestra la eficiencia del BRKGA propuesto.

De manera similar, han surgido nuevas variantes del THLP. Por ejemplo, en [31] proponen el problema de Mediana ordenada de localización de un árbol de hubs (OMTHL, por sus siglas en inglés). Este problema consiste en realizar ubicar un número preestablecido de hubs conectados en forma de árbol. El objetivo de este problema es minimizar el promedio ponderado ordenado de las distancias y los costos de transporte.

La tabla 1 muestra las referencias más significativas a los problemas relacionados con la estructura de la red en forma de árbol, en esta tabla se busca conocer los tipos de modelos que utiliza además de los objetivos que quieren conseguir

Revisión de literatura sobre Tree of Hubs Location Problem			
Referencias	Tipo de Modelo	Objetivo	Aporte
Contreras et al. (2010) [23]	Mixta Lineal-Entera	Minimizar	Primera formulación del THLP
Martín de Sá et al. (2012) [29]	Mixta Lineal-Entera	Minimizar	Un algoritmo de descomposición de Bender utilizando nuevos esquemas de cortes
Contreras et al. (2009) [24]	Mixta Lineal-Entera	Minimizar	Relajación Lagrangiana para encontrar cotas inferiores y superiores más ajustadas
Blanco & Marín (2018) [32]	Mixta Lineal-Entera	Minimizar	Proponen una nueva variante del THLP con mejoras, en el se busca añadir nuevos hubs o mejorar los que se tienen
Pessoa et al. (2017) [30]	Mixta Lineal-Entera	Minimizar	Uso de un algoritmo metaheurístico BRKGA para la solución del problema propuesto por Contreras
Pozo et al (2020) [31]	Mixta Lineal-Entera	Minimizar	Proponen una nueva variante del THLP, aquí se busca minimizar la suma total del promedio del costo de los arcos
Lee et al. (1993) [25]	Programación Binaria	Minimizar	Diseño de una red de servicios digitales utilizando una estructura en forma de árbol
Lee et al. (1996) [26]	Programación Binaria	Minimizar	Un enfoque de relajación lagrangiana para el problema de diseñar una red de servicios digitales en forma de árbol
Hakimi et al. (1993) [27]		Minimizar	Sienta las bases de un problema de localización de hubs extensivos en forma de árbol.
Hu (1974) [28]		Minimizar	Resuelve un problema de comunicación óptima del mínimo árbol recubridor

Otro de los aspectos más importantes del problema propuesto en esta investigación, es la decisión de los precios a fijar por el uso de la red de hubs. Dichos precios tienen un rol crucial a la hora de crear dichas redes de distribución. Por ejemplo, una empresa la cual busca entrar a un nuevo mercado de transporte quiere tener precios competitivos en sus redes de distribución al mismo tiempo que sean accesibles para los usuarios. Encontrar estos precios óptimos permitirá posicionar a la empresa en el mercado al cual busca entrar. En general, las empresas están más interesadas en obtener buenas ganancias en lugar de minimizar costos. Sin embargo debido a la naturaleza del HLP, no existen muchos trabajos en los cuales se tenga un objetivo enfocado a maximizar ganancias [33].

Una variante del problema clásico de localización de hubs donde se considera un objetivo de maximización, es el problema de localización de hubs competitivos (CHLP, por sus siglas en inglés). En comparación con el HLP donde se busca el diseño de redes de una sola organización, en el CHLP se considera un mercado donde varias firmas coexisten y cada una busca maximizar sus propias ganancias. En [34] se propone una formulación para el CHLP, donde se busca atraer la atención de los usuarios al instalar hubs de distintas firmas dentro de un mercado, y así maximizar la demanda capturada de las firmas, lo cual implica tener mejor ganancia.

En [35], también trabajaron con un modelo de hubs competitivo, llamado el problema de Stackelberg para la localización de hubs. En este problema se considera una firma grande, la cual compite contra otras firmas pequeñas y cada una busca maximizar su propia ganancia. Para ello, la firma grande ubica sus hubs y después las firmas pequeñas actúan en consecuencia localizando hubs en otros sitios potenciales. Para modelar este problema, se utiliza la programación binivel debido a la naturaleza jerárquica del problema.

Sobre esta misma idea de instalar hubs competitivos, se encuentran los siguientes trabajos. En [36] se propone un problema de localización competitiva en el contexto de una firma de transporte que cuenta con una red de distribución con hubs, y existe una nueva firma que desea entrar al mercado con una red de hubs para maximizar sus propias ganancias. En [37] se propone un modelo de localización de hubs líder-seguidor, en el que dos competidores crean sus redes de transporte con hubs para atraer clientes en un mercado cuyos precios son fijos. En ese trabajo, cada firma busca maximizar su ganancia y la demanda se reparte entre las distintas firmas. En [38] también crean un modelo de localización de hubs con precios en una red que forma por dos firmas. En ese problema, la firma con mayor jerarquía ubica sus hubs, mientras que la firma con menor jerarquía

ubica los suyos en los sitios restantes que no han sido elegidos por la competencia.

Es importante hacer la aclaración de que aunque los problemas recién descritos son modelados con programación binivel, en este proyecto de investigación no existe la competencia entre dos firmas. La diferencia jerárquica entre los tomadores de decisiones se debe a que uno es quien diseña la red y el otro es el usuario.

Debido a los esfuerzos realizados en el área de localización de hubs competitivos, en los años recientes se han hecho muchos avances en el tema de fijación de precios. Por ejemplo, en [39] y [40] se aborda un problema de diseño de redes con hubs donde los objetivos eran maximizar las ganancias. Por otro lado, en [33] se presentan una serie de formulaciones para un problema de localización de hubs para maximizar ganancias considerando distintas asignaciones (única y múltiple). En [41] se introduce un nuevo problema de localización de hubs sin capacidades, en el cual se toman en consideración los precios. El objetivo es maximizar la ganancia escogiendo la mejor red de hubs fijando los precios adecuados de acuerdo a la demanda.

En la tabla 2, trata con problemas relacionados de fijación óptima de precios, de la misma forma que la tabla 1, se busca identificar el tipo de modelo usado en la referencia además del objetivo que se busca obtener.

Revisión de literatura sobre Pricing Hub Location Problem			
Referencias	Tipo de Modelo	Objetivo	Aporte
Cvokic et al (2016) [37]	Programación Bivuel	Maximizar	Un modelo de programación bivuel con competencia entre dos firmas.
Cvokic & Stanimirovic (2018) [41]	Mixta No Lineal-Entera	Maximizar	Un modelo de localización de hubs y precios con asignación única.
Luer Vilagra & Marianov (2013) [36]	Mixta No Lineal-Entera	Maximizar r	Modelo de localización de hubs con dos empresas donde cada una busca maximizar la ganancia
Albeyga et al (2016) [39]	Entero Binario	Maximizar	Un modelo para un problema de diseño en redes con objetivos orientados a ganancias
Taherkhani & Alamur (2018) [33]	Mixto No Lineal-Entero	Maximizar	Una colección de formulaciones para distintos modelos de localización considerando precios
Albeyga et al (2017) [40]	Entero Binario	Maximizar	Proponen una solución exacta para el modelo propuesto por ellos mismos
Esmaeili & Sedehzade (2020) [38]	Entero Mixto-Lineal	Maximizar	Un problema de hubs y pricing entre dos firmas competitivas con contratos
Sasaki & Fukushima (2001) [35]	Programación Bivuel	Maximizar	Proponen un problema en el cual, una empresa tiene que competir contra otras firmas más pequeñas para maximizar su ganancia

2.1 ANTECEDENTES MATEMÁTICOS

En esta sección se muestran distintos conceptos matemáticos de Programación Binivel y de la Teoría de Grafos, los cuales tienen una relación intrínseca con el tema de tesis. Dichos conceptos han sido ampliamente estudiados durante años y son parte de la base fundamental para comprender el problema bajo investigación en esta tesis. Así mismo, forman parte crucial para diseñar un algoritmo para resolver el problema aquí planteado.

2.2 PROGRAMACIÓN BINIVEL

La programación multinivel es una rama de la programación matemática que se utiliza para modelar situaciones donde múltiples decisiones tienen que ser tomadas por distintos agentes, los cuales mantienen una jerarquía entre sí. Esto último conduce a que las decisiones no sean tomadas de forma simultánea, esto es, el tomador de decisiones de mayor jerarquía realiza una acción y después los demás agentes actúan en consecuencia para tomar la mejor decisión ante esa acción. El caso particular de la programación multinivel en el que solo se involucran a dos niveles de decisión se llama Programación Binivel (BP, por sus siglas en inglés).

La programación binivel tiene su base en la Teoría de Juegos, específicamente en el área de los Juegos de Stackelberg, en este contexto, existen jugadores que compiten entre sí, un jugador (llamado líder) realiza un movimiento, luego, el resto de los jugadores (llamados seguidores) realizan un movimiento en consecuencia para amortiguar la afectación en su función objetivo.

Un modelo de programación binivel consiste en un modelo cuyas variables de decisión pueden ser separadas en dos subconjuntos, y uno de estos subconjuntos se determinará con ayuda de otro modelo de programación matemática. Dicho de otra manera, un modelo de programación binivel es un modelo de programación matemática el cual una de las restricciones es otro problema de programación matemática.

Una formulación general de este problema propuesta por [42] se muestra a continuación:

$$\min_{x \in X} F(x, y) = c_1x + d_1y \quad (2 \text{ a.})$$

$$s.t. \quad A_1x + B_1y \leq b_1 \quad (2 \text{ b.})$$

$$\min_{y \in Y} f(x, y) = c_2x + d_2y \quad (2 \text{ c.})$$

$$s.t. \quad A_2x + B_2y \leq b_2 \quad (2 \text{ d.})$$

donde $F : X \times Y \rightarrow \mathbb{R}$ y $f : X \times Y \rightarrow \mathbb{R}$ además $c_1, c_2 \in \mathbb{R}^n, d_1, d_2 \in \mathbb{R}^m, b_1 \in \mathbb{R}^p, b_2 \in \mathbb{R}^q, A_1 \in \mathbb{R}^{p \times n}, B_1 \in \mathbb{R}^{p \times m}, A_2 \in \mathbb{R}^{q \times n}, B_2 \in \mathbb{R}^{q \times m}$.

Las variables x son las variables de decisión del líder, mientras que las variables y son las variables de decisión del seguidor. Para este modelo, el líder busca minimizar la función objetivo $F(x, y)$ sujeto a la restricciones (2 b.) y (2 c.). Por otra parte, el seguidor busca minimizar la función objetivo $f(x, y)$ sujeto a la restricción (2 d.).

Los problemas de programación binivel son np-hard [43] esto implica que no existe un algoritmo en tiempo polinomial que pueda resolver este problema de forma rapida, afortunadamente gracias a los avances que se han realizado con la teoría de dualidad y las condiciones de Karush Kuhn Tucker se puede reformular el problema de programación binivel a un problema de un solo nivel de programación matemática, esto es bueno ya que, a comparación de los problemas binivel, existe software especializado que simplifica muchos las cosas al tratar con ellos.

Sin embargo, el tiempo para encontrar la solución puede ser lento, es por ello que en los últimos años la tendencia al uso de heurísticas y metaheurísticas. Algunos trabajos que implican el uso de metaheurísticas con la programación binivel pueden ser encontradas en [citas]

2.3 TEORÍA DE GRAFOS

La Teoría de Grafos es otra de las áreas que está estrechamente relacionada con el proyecto de investigación que se está realizando. Esta área toma gran importancia dentro de la investigación de operaciones cuando es conveniente representar relaciones entre pares de elementos en un sistema, por ejemplo, en los problemas de localización de instalaciones es común querer representar por medio de un grafo la relación que existe entre los clientes

que exigen una demanda y las instalaciones que pueden cumplirla.

Desde los primeros esbozos de la teoría propuesta por Euler hasta algoritmos que explotan las capacidades de las estructuras en redes, la teoría de grafos ha sido una rama que ha evolucionado mucho con el tiempo. En esta sección se presentan algunos unos conceptos generales de la teoría de grafos, los cuales están relacionados con el THLP además de algunos algoritmos utilizados para facilitar la resolución del problema. Los conceptos se encuentran en [44]. Aunque existan sinónimos a algunos de los conceptos que se presentan, nosotros usaremos la terminología comúnmente utilizada dentro de la comunidad de matemáticas y de la investigación de operaciones.

Un **grafo** $G(V, E)$ es un par ordenado, donde V es un conjunto no vacío de objetos llamados vértices (vertex en inglés) y E es un conjunto de pares no ordenados de vértices. Cada par $(p, q) \in E$ se llama arista (edge en inglés) y se dice que une a los vértices p y q .

La representación visual de un grafo puede ser mediante una unión de puntos y líneas, donde los puntos representan los vértices del grafo y las líneas representan las aristas del grafo. Un ejemplo de un grafo se muestra a continuación, en este caso $V = \{v_1, v_2, v_3\}$

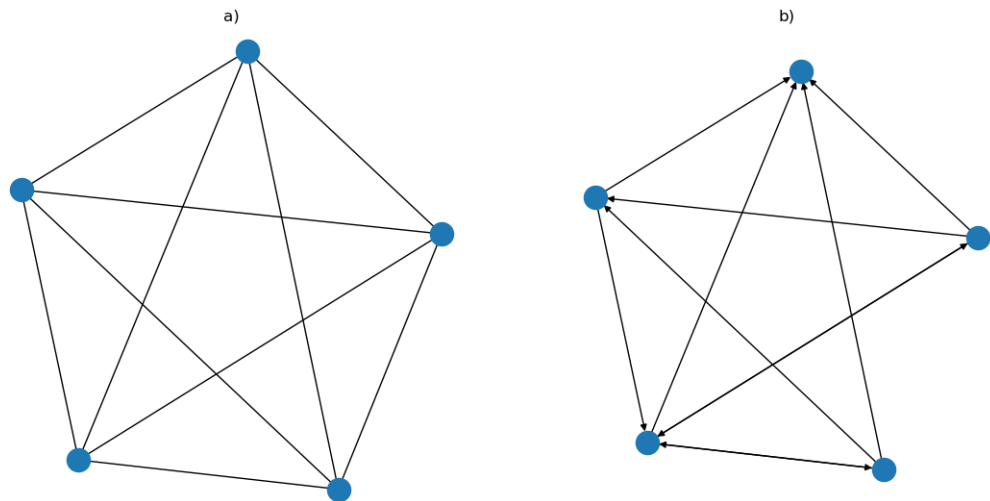


Figura 2.1: a) Ejemplo de un grafo de 5 vértices. b) Ejemplo de un digrafo con 5 vértices

En ocasiones resulta conveniente darle una dirección a una arista, cuando esto sucede, el grafo es llamado grafo dirigido o digrafo. Las aristas de un digrafo suelen llamarse arcos. Un ejemplo de un digrafo se muestra en la figura 1. Un tipo especial de digrafo es llamado red. Una red es un digrafo cuyos vértices son llamados nodos y cuenta con un nodo origen y un nodo destino. Por lo general una red sirve para mostrar el flujo de un producto a través de una red de servicios.

Un grafo o un digrafo G se dice que tiene peso si existe una función que $w : E \rightarrow \mathbb{R}$ que asigna un valor real, llamado peso, a cada arista o arco de los grafos mencionados. Esto resulta beneficioso, ya que en muchos contextos, un grafo con peso nos puede ayudar a representar distintos problemas dentro del área de investigación de operaciones. Por ejemplo, en un problema de localización de instalaciones se puede representar el costo al cubrir la demanda de los clientes utilizando grafos con pesos.

Una de las propiedades más importantes de los grafos es la conectividad, algunos conceptos relacionados con este aspecto de los grafos se muestran a continuación.

Una caminata (walk en inglés) de un grafo es una secuencia de vértices y aristas de G . Se dice que una caminata es cerrada si $v_1 = v_n$ donde v_1 y v_n son los vértices inicial y final del grafo. Una caminata se dice que es un camino (path en inglés) si todos los vértices y todas las aristas son distintas. Por último, una caminata con al menos tres vértices distintos (exceptuando el primero y el último) es llamado un ciclo. Un grafo se dice que está conectado si todo par de vértices de un grafo está unido por un camino.

Ahora introduciremos uno de los conceptos más importantes dentro de la teoría de grafos y dentro del proyecto de investigación, esto es debido a que muchas de las aplicaciones que involucran a la teoría de grafos involucra el uso de este concepto. Un árbol (tree en inglés) es un grafo conectado que no tiene ciclos, esto es importante, ya que al suceder esto, el camino que existe entre los dos nodos es único. Ejemplo de grafos en forma de árbol, se muestran en la figura 2.2.

Como se mencionó, la estructura teórica de un árbol permite un cantidad numerosa de aplicaciones, por ejemplo, en las redes de comunicaciones muchas veces se utilizan este tipo de estructuras ya que al no contener ciclos el envío de información no rebota entre usuarios hasta llegar a su destino o como ya se ha mencionado, en el transporte el uso de redes en forma de árbol permite la accesibilidad de costos en el transporte.

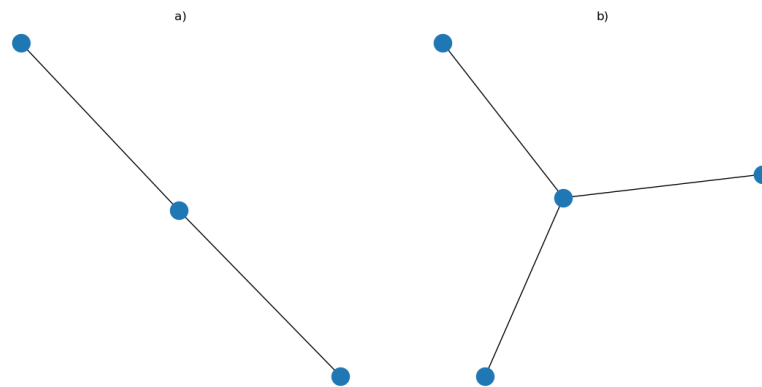


Figura 2.2: a) Ejemplo de un árbol de 3 vértices. b) Ejemplo de un árbol con 4 vértices

Existe una serie de resultados que involucran los árboles, por el momento se encuentran a continuación, pero en capítulos posteriores se harán hincapié de ellos.

Teorema :1

El número de aristas de un árbol con n vértices es $n - 1$.

Teorema2 :

El número de árboles distintos con n vértices es n^{n-2} .

Las demostraciones de estos teoremas se pueden encontrar en [45].

Hay algunos casos especiales cuando se habla de grafos en forma de árbol como subgrafos de grafos más grandes. Sea $G = (V, E)$ un grafo conectado el cual tiene un subgrafo definido como $T = (V', E')$. Si $V = V'$ entonces se dice que T es un árbol recubridor del grafo G . En otras palabras, un árbol recubridor es un subgrafo que tiene todos los vértices de un grafo.

Por último, muchas veces es necesario representar los grafos utilizando números más que representaciones gráficas, esto debido a que las computadoras están más adaptadas a manipular números que imágenes, es por ello que una práctica estándar es representar a un grafo a través de una matriz.

Existe un gran número de matrices que pueden ser asociadas a grafos. La matriz de adyacencia y la matriz de incidencia son algunas de ellas. Estas matrices no solo sirven como herramientas para representar la estructura de un grafo, sino además para estudiar las propiedades de algunos de ellos. En esta tesis se hará hincapié en la matriz de adya-

cencia ya que es la que se considera en la metodología aquí propuesta.

La matriz de adyacencia $A = (a_{ij})_{n \times n}$, de un grafo G con n vértices, es la matriz en el cual la entrada $a_{ij} = 1$ si el vértice v_i es adyacente al vértice v_j y $a_{ij} = 0$ de otra manera, donde v_i y v_j son vértices de G . A continuación se muestra un ejemplo de un grafo con la matriz de adyacencia correspondiente.

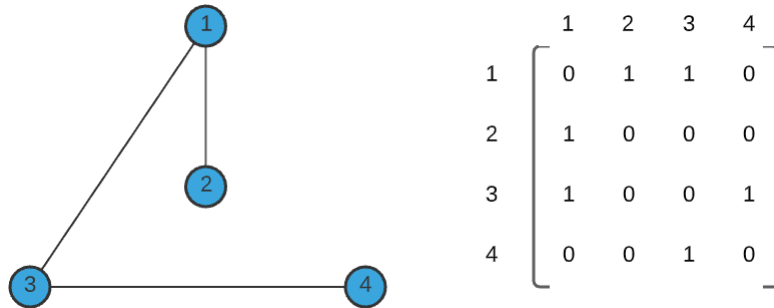


Figura 2.3: Árbol con 4 nodos y su matriz de adyacencia correspondiente

Cabe destacar que estas matrices no tienen que ser simétricas, consideremos un digrafo cuyos arcos tienen diferente peso, una representación de este ejemplo puede verse a continuación:

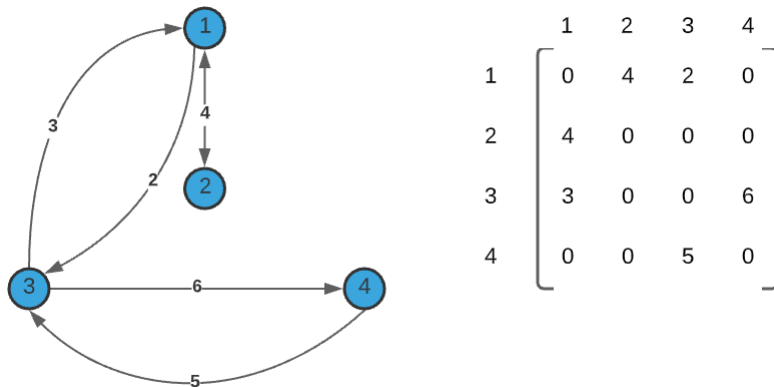


Figura 2.4: Árbol con 4 nodos y su matriz de adyacencia correspondiente

Este último ejemplo está fuertemente relacionado con nuestro problema, ya que los costos asociados a la red de concentradores pueden llegar a ser distintos dependiendo de la dirección en la cual se envían los bienes. Además, las soluciones se representarán en forma matricial.

La teoría de grafos es muy extensa y debido a la gran aplicabilidad a distintos contextos, se pueden pensar en algoritmos que puedan ser utilizados. Uno de los principales y más importantes dentro del área es el algoritmo de Dijkstra [46], este algoritmo es utilizado para encontrar el camino más corto entre un par de nodos, tomando en cuenta el costo peso asociado a las aristas que unen al grafo. Este algoritmo es especialmente importante dentro de nuestro problema ya que cuando el líder decide la estructura de la red y los precios, el seguidor busca encontrar el camino más corto dada dentro de la red ya establecida. Este algoritmo se explicará a más detalle en capítulos posteriores.

CAPÍTULO 3

DESCRIPCIÓN DEL PROBLEMA

En esta sección la descripción y formulación del modelo binivel para el THLP con precios se describen a continuación. Primero, se definen los conjuntos, parámetros y variables de decisión involucrados en la formulación matemática.

Sea I el conjunto de hubs potenciales y sea J el conjunto de clientes. Existe una cantidad de bienes (flujo) que deben ser enviados entre cada par de clientes $k, j \in J$, el cual es denotado por w_{jk} . Con lo anterior, se puede definir el conjunto $K = \{(k, j) : k, j \in J, w_{jk} > 0\}$, el cual indica los bienes que se envían entre cada par de clientes. Para simplificar, sea $a = (k, j) \in K$ un par ordenado, en donde el origen $o(a) = k$ y el destino $d(a) = j$. Entonces se define el flujo w_a para $a \in K$. A continuación, se definen los parámetros de nuestro problema:

- c_a Costo unitario de enviar el bien $a \in K$ de forma directa.
- $d_{o(a)i}$ Costo de acceso del origen al hub $i \in I$
- $d_{id(a)}$ Costo de salida del hub $i \in I$ al destino.
- d_{il} Costo unitario de enviar el bien a través de los hubs $i, l \in I$.
- g_{ail} Costo unitario de uso de la infraestructura de la red por el bien $a \in K$
- f_{il} Costo de construcción de un camino entre hubs $i \in I$ y $l \in I$

Con esta notación, el líder desea construir una red de hubs en forma de árbol, así como decidir los precios para usar los arcos dentro de la red. Para construir un modelo de programación matemática se consideran las siguientes variables de decisión:

- $\pi_{i\ell} \geq 0$: precio unitario por usar el arco $\{i, \ell\}$, para $i, \ell \in I$.
- $y_i = \begin{cases} 1 & \text{si el hub } i \text{ es abierto.} \\ 0 & \text{de otra manera} \end{cases}, \forall i \in I.$

$$\blacksquare z_{i\ell} = \begin{cases} 1 & \text{si el arco } \{i, \ell\} \text{ del hub es parte de la estructura de la red} \\ 0 & \text{de otra manera} \end{cases}, \forall i, \ell \in I.$$

Una vez que el líder ha decidido la estructura de la red, el seguidor decide si utiliza o no la red de distribución, y la manera en que se van a rutear los bienes dentro de la red. Es por esto que se introducen las siguientes variables de decisión del seguidor:

$$\blacksquare x_{a i \ell} = \begin{cases} 1 & \text{Si el bien } a \text{ se envía a través del arco } (i, \ell) \text{ de la red.} \\ 0 & \text{de otra manera} \end{cases}, \forall a \in K, i, \ell \in I.$$

$$\blacksquare x_{a i}^1 = \begin{cases} 1 & \text{Si el bien } a \text{ se envía inicialmente por el hub } i \text{ de la red.} \\ 0 & \text{de alguna otra manera} \end{cases}, \forall a \in K, i \in I.$$

$$\blacksquare x_{a i}^2 = \begin{cases} 1 & \text{Si el bien } a \text{ se envía finalmente por el hub } i \text{ de la red} \\ 0 & \text{de otra manera} \end{cases}, \forall a \in K, i \in I.$$

$$\blacksquare q_a = \begin{cases} 1 & \text{Si el bien } a \text{ se envía de forma directa por los clientes} \\ 0 & \text{de otra manera} \end{cases}, \forall a \in K.$$

El modelo resultante de programación binivel del BTHLPwP es como sigue:

$$\text{máx} \quad \sum_{i\ell} \pi_{i\ell} \sum_{a \in K} w_a x_{a i \ell} + \sum_{i, \ell \in I} g_{a i \ell} x_{a i \ell} - \sum_{i, \ell \in I} f_{i\ell} z_{i\ell} \quad (3.1)$$

$$\text{s.t.} \quad \sum_{i \in I} y_i = p, \quad (3.2)$$

$$\sum_{\substack{i, \ell \in I: \\ i > \ell}} z_{i\ell} = p - 1, \quad (3.3)$$

$$z_{i\ell} \leq y_i, \quad (3.4)$$

$$z_{i\ell} \leq y_\ell, \quad (3.5)$$

$$y_i, z_{i\ell} \in \{0, 1\}, \pi_{i\ell} \geq 0 \quad (3.6)$$

$$\text{mín} \quad \sum_{a \in K} w_a \left(\sum_{i \in J} (d_{o(a)i} x_{o(a)i}^1 + d_{id(a)} x_{d(a)i}^2) + \sum_{i, \ell \in I} (d_{i\ell} + \pi_{i\ell}) x_{a i \ell} + c_a q_a \right) \quad (3.7)$$

$$\text{s.t.} \quad \sum_{i \in I} x_{a i}^1 + q_a = 1, \forall a \in K, \quad (3.8)$$

$$\sum_{i \in I} x_{a i}^2 + q_a = 1, \forall a \in K, \quad (3.9)$$

$$\sum_{\ell \in I} x_{ail} + x_{ai}^1 - \sum_{\ell \in I} x_{ali} - x_{ai}^2 = 0, \forall a \in K, i \in I, \quad (3.10)$$

$$x_{ai}^1 \leq y_i, \forall a \in K, i \in I, \quad (3.11)$$

$$x_{ai}^2 \leq y_i, \forall a \in K, i \in I, \quad (3.12)$$

$$x_{ail}^1 \leq z_{i\ell}, \forall a \in K, i, \ell \in I, \quad (3.13)$$

$$x_{ail}, x_{ai}^1, x_{ai}^2, q_a \in \{0, 1\}, \forall a \in K, i, \ell \in I.$$

En la formulación dada por las Ec (3.1)-(3.13), la función objetivo del líder Ec. (3.1) maximiza el ingreso total menos el costo derivado por la construcción de la red de hubs. La restricción (3.2) obliga a que se abran exactamente p hubs; mientras que la restricción (3.3) dice que se van a construir $p - 1$ arcos entre hubs, esto para promover que la estructura de la red sea en forma de árbol. Las restricciones (3.4) y (3.5) permiten que la construcción de arcos sea únicamente entre los hubs abiertos. Por último, la restricción (3.6) indica la naturaleza de las variables de decisión del líder.

Como ya se mencionó anteriormente, basado en la decisiones del líder (las variables π, y, z), ahora el seguidor rutea el flujo del bien minimizando el costo. La función objetivo del seguidor está dada en Ec. (3.7), la cual considera el costo de ruteo del producto a través de la red de hubs. Es decir, para algún bien a hay que obtener el costo de recolectar el flujo (el envío inicial del producto), el costo de viaje entre los hubs y el costo de redistribución del flujo (el envío final del producto), y finalmente el costo de enviar flujos de forma directa ($c_a q_a$).

El resto de las restricciones garantizan la conservación del flujo a través de la red. La restricción (3.8) asegura que cada bien se envíe a un hub o se envíe de forma directa desde el origen hasta el destino. La restricción (3.9) garantiza que el flujo llegue a su destino, ya sea por un hub dentro de la red o de forma directa desde su origen. Por otro lado, la Ec. (3.10) es la restricción de conservación de flujo dentro de la red de hubs, esto es, que el flujo de algún bien a en un hub i esté dado por los bienes que llegan directo del origen o por algún otro hub dentro de la red. Además, que el flujo que sale del hub se envíe directamente al destino o se enruta hacia otro hub dentro de la red. Por último, las restricciones (3.11), (3.12) y (3.13) evitan que el flujo sea enviado a través de hubs que no están abiertos o arcos que no estén en la estructura de red.

Observe que una vez que los precios $\bar{\pi}$, los hubs abiertos ($\bar{H} = i \in I : y_i = 1$) y la red en forma de árbol ($\bar{T} = (i, \ell) : z_{i\ell} = 1$) han sido decididos por el líder, el problema

del seguidor resulta ser un problema de flujo con costo mínimo, el cual puede ser resuelto de manera eficiente utilizando el algoritmo de Dijkstra.

3.1 REFORMULACIÓN DEL MODELO DE PROGRAMACIÓN BINIVEL A UNO MILP

Como se ha hecho mención, se puede reformular un problema de programación binivel a un solo nivel utilizando las condiciones de Karush-Kuhn-Tucker. Esto se logra al considerar las condiciones de optimalidad del problema del seguidor y añadiéndolas al modelo. En este caso, para reformular el modelo binivel, primero se va a considerar el problema resultante del seguidor una vez que el líder ya ha fijado los precios y ha construido la red de hubs $(\bar{\pi}, \bar{H}, \bar{T})$. Esto es como sigue:

$$\min \sum_{a \in K} w_a \left(\sum_{i \in J} (d_{o(a)i} x_{o(a)i}^1 + d_{id(a)} x_{d(a)i}^2) + \sum_{i, \ell \in I} (d_{i\ell} + \pi_{i\ell}) x_{ai\ell} + c_a q_a \right) \quad (3.14)$$

$$\text{s.t.} \quad \sum_{i \in \bar{H}} x_{ai}^1 + q_a = 1, \forall a \in K, \quad (3.15)$$

$$\sum_{i \in \bar{H}} x_{ai}^2 + q_a = 1, \forall a \in K, \quad (3.16)$$

$$\sum_{\substack{\ell \in \bar{H}: \\ \{i, \ell\} \in \bar{T}}} x_{ai\ell} + x_{ai}^1 - \sum_{\substack{\ell \in \bar{H}: \\ \{i, \ell\} \in \bar{T}}} x_{ali} - x_{ai}^2 = 0, \forall a \in K, i \in \bar{H}, \quad (3.17)$$

$$x_{ai}^1, x_{ai}^2, q_a \geq 0, \forall a \in K, i \in I.$$

$$x_{ai\ell} \geq 0, \forall i, \ell \in \bar{H} : \{i, \ell\} \in \bar{T}.$$

Como se puede apreciar en las Ec. (3.14)- (3.17), el problema del seguidor resulta ser un problema de programación lineal. Entonces, es fácil construir el problema dual asociado a este modelo considerando que λ^1 , λ^2 y μ son las variables duales asociadas a las restricciones (3.15), (3.16) y (3.17), respectivamente. Dicho modelo dual se presenta a continuación:

$$\max \sum_{a \in K} (\lambda_a^1 + \lambda_a^2) \quad (3.18)$$

$$\text{s.t.} \quad \lambda_a^1 - \mu_{ai} \leq w_a d_{o(a)i}, \forall a \in K, i \in H \quad (3.19)$$

$$\lambda_a^2 + \mu_{ai} \leq w_a d_{id(a)}, a \in K, i \in H \quad (3.20)$$

$$\lambda_a^1 + \lambda_a^2 \leq w_a c_a, \forall a \in K, \quad (3.21)$$

$$\mu_{ai} - \mu_{al} \leq (d_{i\ell} + \pi_{i\ell})w_a, \forall a \in K, \{i, \ell\} \in T, , \quad (3.22)$$

$$\lambda_a^1, \lambda_a^2 \in \mathbb{R}, \forall a \in K,$$

$$\mu_{ai} \in \mathbb{R}, \forall a \in K, i \in H.$$

Si las restricciones (3.11)–(3.13) se introducen como restricciones de capacidad en el problema de costo mínimo de arriba en lugar de usar \bar{H} y \bar{T} , se obtiene la siguiente formulación dual:

$$\text{máx} \sum_{a \in K} (\lambda_a^1 + \lambda_a^2) - \sum_{a \in K} \sum_{i \in I} y_i (v_{ai}^1 + v_{ai}^2) - \sum_{a \in K} \sum_{i, \ell \in I} u_{ail} z_{il} \quad (3.23)$$

$$\text{s.t. } \lambda_a^1 - \mu_{ai} - v_{ai}^1 \leq w_a d_{o(a)i}, \forall a \in K, i \in H \quad (3.24)$$

$$\lambda_a^2 + \mu_{ai} - v_{ai}^2 \leq w_a d_{id(a)}, \forall a \in K, i \in H \quad (3.25)$$

$$\lambda_a^1 + \lambda_a^2 \leq w_a c_a, \forall a \in K, \quad (3.26)$$

$$\mu_{ai} - \mu_{al} - u_{ail} \leq (d_{i\ell} + \pi_{i\ell})w_a, \forall a \in K, \{i, \ell\} \in T, , \quad (3.27)$$

$$\lambda_a^1, \lambda_a^2 \in \mathbb{R}, \forall a \in K,$$

$$\mu_{ai} \in \mathbb{R}, \forall a \in K, i \in H,$$

$$v_{ai}^1, v_{ai}^2, v_{ail} \geq 0, \forall a \in K, i, \ell \in I.$$

Entonces, el problema binivel puede ser reformulado como el siguiente problema de programación mixta entera de un solo nivel:

$$\text{máx} \sum_{i\ell} \pi_{i\ell} \sum_{a \in K} w_a x_{ail} + \sum_{i, \ell \in I} g_{ail} x_{ail} - \sum_{i, \ell \in I} f_{il} z_{il} \quad (3.28)$$

$$\text{s.t. } \sum_{i \in I} y_i = p, \quad (3.29)$$

$$\sum_{\substack{i, \ell \in I: \\ i > \ell}} z_{il} = p - 1, \quad (3.30)$$

$$z_{il} \leq y_i, \quad (3.31)$$

$$z_{il} \leq y_l, \quad (3.32)$$

$$\sum_{i \in I} x_{ai}^1 + q_a = 1, \forall a \in K, \quad (3.33)$$

$$\sum_{i \in I} x_{ai}^2 + q_a = 1, \forall a \in K, \quad (3.34)$$

$$\sum_{\ell \in I} x_{ail} + x_{ai}^1 - \sum_{\ell \in I} x_{ali} - x_{ai}^2 = 0, \forall a \in K, i \in I, \quad (3.35)$$

$$x_{ai}^1 \leq y_i, \forall a \in K, i \in I, \quad (3.36)$$

$$x_{ai}^2 \leq y_i, \forall a \in K, i \in I, \quad (3.37)$$

$$x_{ail}^1 \leq z_{il}, \forall a \in K, i, \ell \in I, \quad (3.38)$$

$$\lambda_a^1 - \mu_{ai} - v_{ai}^1 \leq w_a d_{o(a)i}, \forall a \in K, i \in H \quad (3.39)$$

$$\lambda_a^2 + \mu_{ai} - v_{ai}^2 \leq w_a d_{id(a)}, \forall a \in K, i \in H \quad (3.40)$$

$$\lambda_a^1 + \lambda_a^2 \leq w_a c_a, \forall a \in K, \quad (3.41)$$

$$\mu_{ai} - \mu_{al} - u_{ail} \leq (d_{il} + \pi_{il})w_a, \forall a \in K, \{i, \ell\} \in T, \quad (3.42)$$

$$\begin{aligned} & \sum_{a \in K} (\lambda_a^1 + \lambda_a^2) - \sum_{a \in K} \sum_{i \in I} y_i (v_{ai}^1 + v_{ai}^2) - \sum_{a \in K} \sum_{i, \ell \in I} u_{ail} z_{il} \\ &= \sum_{a \in K} w_a \left(\sum_{i \in J} (d_{o(a)i} x_{o(a)i}^1 + d_{id(a)} x_{d(a)i}^2) + \sum_{i, \ell \in I} (d_{il} + \pi_{il}) x_{ail} + c_a q_a \right) \end{aligned} \quad (3.43)$$

$$\lambda_a^1, \lambda_a^2 \in \mathbb{R}, \forall a \in K,$$

$$\mu_{ai} \in \mathbb{R}, \forall a \in K, i \in H,$$

$$v_{ai}^1, v_{ai}^2, v_{ail} \geq 0, \forall a \in K, i, \ell \in I.$$

$$x_{ail}, x_{ai}^1, x_{ai}^2, q_a \in \{0, 1\}, \forall a \in K, i, \ell \in I,$$

$$y_i, z_{il} \in \{0, 1\}.$$

En la reformulación [Ecs (3.28)-(3.43)], la Ec. (3.43) hace referencia a que las funciones objetivo del problema dual y el primal sean iguales. Como eso ocurre únicamente en el óptimo, se garantiza que el problema del seguidor sea resuelto a optimalidad. Notemos que en la reformulación recién presentada se encuentran términos no lineales (ver las restricciones (3.28) y (3.43)). En específico, se encuentran los productos de variables $y_i v_{ai}^1$, $y_i v_{ai}^2$, $u_{ail} z_{il}$ y $\pi_{il} x_{ail}$. Sin embargo, estas expresiones se pueden linealizar introduciendo las siguientes variables auxiliares:

$$\xi_{ai}^1 = y_i v_{ai}^1, \quad \xi_{ai}^2 = y_i v_{ai}^2, \quad \rho_{ail} = u_{ail} z_{il}, \quad \text{y} \quad \theta_{ail} = \pi_{il} x_{ail}$$

para todo $a \in L, i, \ell \in I$ usando las envolventes de McCormick:

$$\begin{aligned} v_{ai}^1 - M(1 - y_i) &\leq \xi_{ai}^1 \leq v_{ai}^1 + M(1 - y_i), \\ v_{ai}^2 - M(1 - y_i) &\leq \xi_{ai}^2 \leq v_{ai}^2 + M(1 - y_i), \\ u_{ail} - M(1 - z_{il}) &\leq \rho_{ail} \leq u_{ail} + M(1 - z_{il}), \end{aligned}$$

$$\pi_{i\ell} - M(1 - x_{a_{i\ell}}) \leq \theta_{a_{i\ell}} \leq \pi_{i\ell} + M(1 - x_{a_{i\ell}}).$$

para todo $a \in K$, $i, \ell \in I$

Finalmente, agregando estas restricciones, el modelo del BTHLPwP puede ser reformulado como el siguiente problema de programación lineal mixta entera de un solo nivel:

$$\text{máx} \sum_{i\ell} \sum_{a \in K} w_a \theta_{ail} + \sum_{i, \ell \in I} g_{ail} x_{ail} - \sum_{i, \ell \in I} f_{i\ell} z_{i\ell} \quad (3.44)$$

$$\text{s.t.} \sum_{i \in I} y_i = p, \quad (3.45)$$

$$\sum_{\substack{i, \ell \in I: \\ i > \ell}} z_{i\ell} = p - 1, \quad (3.46)$$

$$z_{i\ell} \leq y_i, \quad (3.47)$$

$$z_{i\ell} \leq y_\ell, \quad (3.48)$$

$$\sum_{i \in I} x_{ai}^1 + q_a = 1, \forall a \in K, \quad (3.49)$$

$$\sum_{i \in I} x_{ai}^2 + q_a = 1, \forall a \in K, \quad (3.50)$$

$$\sum_{\ell \in I} x_{ail} + x_{ai}^1 - \sum_{\ell \in I} x_{ali} - x_{ai}^2 = 0, \forall a \in K, i \in I, \quad (3.51)$$

$$x_{ai}^1 \leq y_i, \forall a \in K, i \in I, \quad (3.52)$$

$$x_{ai}^2 \leq y_i, \forall a \in K, i \in I, \quad (3.53)$$

$$x_{ail}^1 \leq z_{i\ell}, \forall a \in K, i, \ell \in I, \quad (3.54)$$

$$\lambda_a^1 - \mu_{ai} - v_{ai}^1 \leq w_a d_{o(a)i}, \forall a \in K, i \in H \quad (3.55)$$

$$\lambda_a^2 + \mu_{ai} - v_{ai}^2 \leq w_a d_{id(a)}, \forall a \in K, i \in H \quad (3.56)$$

$$\lambda_a^1 + \lambda_a^2 \leq w_a c_a, \forall a \in K, \quad (3.57)$$

$$\mu_{ai} - \mu_{al} - u_{ail} \leq (d_{i\ell} + \pi_{i\ell}) w_a, \forall a \in K, \{i, \ell\} \in T, \quad (3.58)$$

$$\begin{aligned} & \sum_{a \in K} (\lambda_a^1 + \lambda_a^2) - \sum_{a \in K} \sum_{i \in I} (\xi_{ai}^1 + \xi_{ai}^2) - \sum_{a \in K} \sum_{i, \ell \in I} \rho_{ail} \\ & = \sum_{a \in K} w_a \left(\sum_{i \in J} (d_{o(a)i} x_{o(a)i}^1 + d_{id(a)} x_{d(a)i}^2) + \sum_{i, \ell \in I} (d_{i\ell} x_{ail} + \theta_{ail}) + c_a q_a \right) \end{aligned} \quad (3.59)$$

$$v_{ai}^1 - M(1 - y_i) \leq \xi_{ai}^1 \leq v_{ai}^1 + M(1 - y_i), \forall a \in K, i \in I, \quad (3.60)$$

$$\xi_{ai}^1 \leq M y_i, \forall a \in K, i \in I, \quad (3.61)$$

$$v_{ai}^2 - M(1 - y_i) \leq \xi_{ai}^2 \leq v_{ai}^2 + M(1 - y_i), \quad (3.62)$$

$$\xi_{ai}^2 \leq M y_i, \forall a \in K, i \in I, \quad (3.63)$$

$$u_{ail} - M(1 - z_{i\ell}) \leq \rho_{ail} \leq u_{ail} + M(1 - z_{i\ell}), \forall a \in K, i, \ell \in N \quad (3.64)$$

$$u_{ail} \leq M z_{i\ell}, \forall a \in K, i, \ell \in N \quad (3.65)$$

$$\pi_{i\ell} - M(1 - x_{ail}) \leq \theta_{ail} \leq \pi_{i\ell} + M(1 - x_{ail}), \forall a \in K, i, \ell \in N \quad (3.66)$$

$$\theta_{ail} \leq M x_{ail}, \forall a \in K, i, \ell \in N \quad (3.67)$$

$$\lambda_a^1, \lambda_a^2 \in \mathbb{R}, \forall a \in K,$$

$$\mu_{ai} \in \mathbb{R}, \forall a \in K, i \in H,$$

$$v_{ai}^1, v_{ai}^2, v_{ail} \geq 0, \forall a \in K, i, \ell \in I.$$

$$x_{ail}, x_{ai}^1, x_{ai}^2, q_a \in \{0, 1\}, \forall a \in K, i, \ell \in I,$$

$$y_i, z_{i\ell} \in \{0, 1\}.$$

A pesar de haber reformulado el problema de programación binivel como uno de un solo nivel, la estructura de la reformulación resultante no tiene un buen comportamiento computacional. Es decir, solo se pueden resolver a optimalidad instancias de tamaño chico mediante optimizadores comerciales. Es por esto que se propone un algoritmo metaheurístico para resolver el problema aquí estudiado.

CAPÍTULO 4

UN ALGORITMO COEVOLUTIVO PARA EL BTHLPWP

En este capítulo, se describe el algoritmo coevolutivo propuesto y sus componentes. En los algoritmos coevolutivos, (a diferencia de los algoritmos evolutivos), la aptitud de los individuos (soluciones) no depende solamente de ellos, sino que también está dada por las interacciones que se tienen con otros individuos. Existen dos tipos de algoritmos coevolutivos de acuerdo al tipo de interacción que exista entre ellos: el cooperativo y el competitivo. En el primero, las dos poblaciones de individuos evolucionan tomando las características que les permita mejorar; mientras que en los competitivos, las poblaciones de individuos evolucionan a expensas de los otros individuos. Literatura relacionada con los algoritmos coevolutivos puede encontrarse en [47], [48], [49].

Los algoritmos coevolutivos cooperativos tienen tres características principales. La primera consiste en descomponer un problema en subcomponentes, donde cada uno es representado como una población en la cual, sus soluciones interactúan para resolver el problema original. La segunda es la forma del proceso de mejora de las soluciones; por lo general, este proceso se determina por la evolución de cada una de las poblaciones a través de un esquema de algoritmo genético estándar utilizando los operadores genéticos conocidos (selección, cruza, mutación). Y por último, la representación de una solución para el problema original, en este caso una solución completa del problema es creada a partir de los mejores representantes de cada una de las poblaciones.

Como ya se ha dicho, las decisiones que debe tomar el líder en el problema del nivel superior son: decidir la estructura del árbol y fijar los precios para el uso de la red. Dado que ambas decisiones están relacionadas de manera muy estrecha, ya que los precios del uso de la red están basados en qué tanto se utiliza la red de distribución, se propone un algoritmo coevolutivo cooperativo para el problema del nivel superior. Es decir, las

decisiones que tiene que tomar el líder evolucionan de manera simultánea.

Una de las ventajas de los algoritmos evolutivos es que al tratar con poblaciones de soluciones permiten que el espacio de búsqueda sea diverso. Además que se ha demostrado que tienen alta efectividad en problemas difíciles de resolver.

El algoritmo coevolutivo que se propone se conforma de las siguientes partes:

- La construcción de una población de soluciones aleatorias para la red en forma de árbol.
- Basados en las soluciones de la red en forma de árbol, se generan unas cotas para los precios. Estas cotas se encuentran al considerar las redes con precios gratuitos, se calcula el camino más corto y se encuentra cuando el flujo de los commodities cambia, se calcula el valor de cambio y el arco donde cambió y se fija este como la cota del precio.
- Dadas las cotas, se generan precios aleatorios por arco, después se hace una mejora de estos precios utilizando un algoritmo genético.
- Se vuelve a evaluar los precios en la función objetivo del líder, es decir con los árboles que ya han sido creados, y después se hace una selección de los mejores árboles creando así las interacciones entre las variables de decisión del líder.
- Con los mejores árboles seleccionados se vuelven a calcular cotas y se repite el proceso.

Un esquema general del algoritmo propuesto es el siguiente:

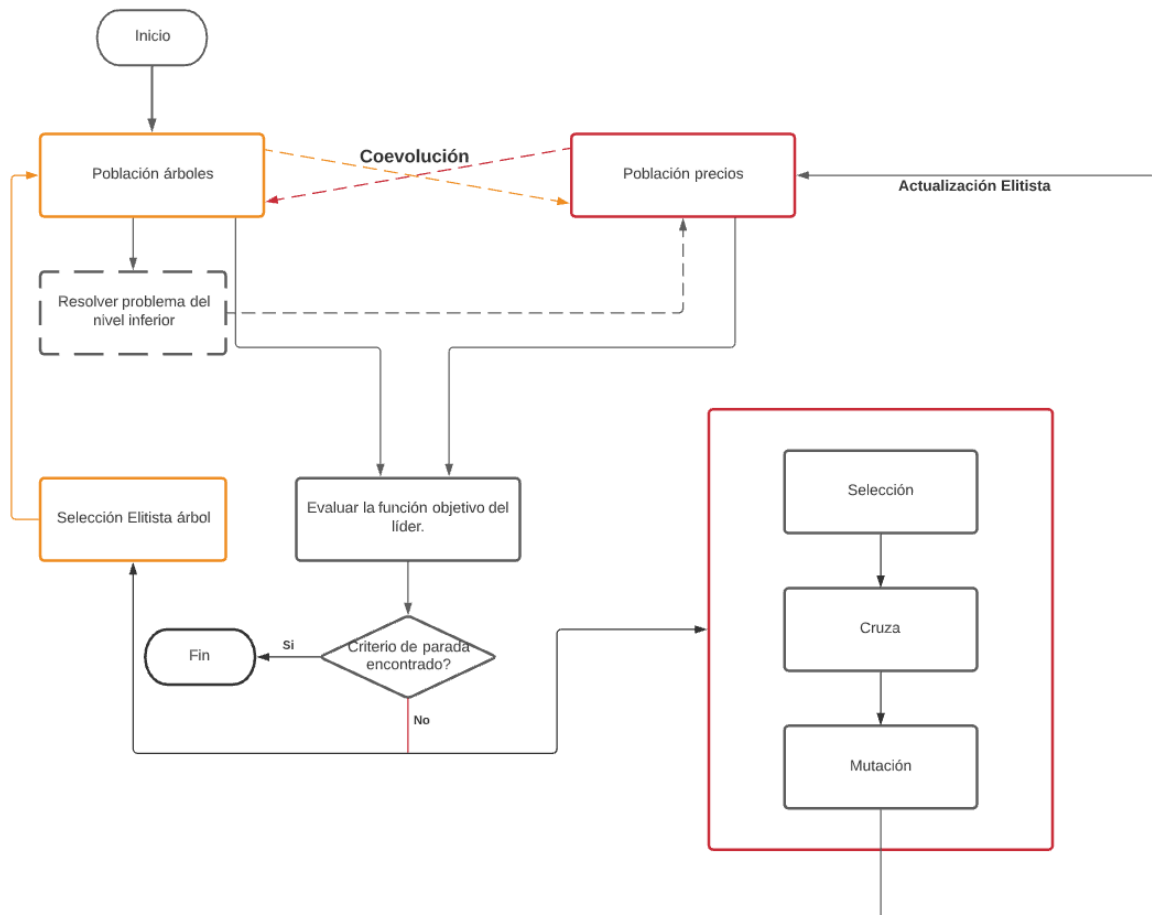


Figura 4.1: Draft del diagrama de flujo

A continuación, se detallan cada uno de los componentes del algoritmo propuesto.

4.1 GENERACIÓN DE SOLUCIONES

Esta sección se centra en la creación de las poblaciones de soluciones involucradas en el problema del líder. Como ya se ha mencionado, el líder una vez que ha definido la red en forma de árbol y los precios a utilizar, el problema del seguidor es un problema de flujo mínimo, el cual se resuelve con un algoritmo convencional de ruta más corta. Por lo que esta sección se enfoca en las soluciones del líder y la forma en que se obtienen.

4.1.1 GENERACIÓN DE LOS ÁRBOLES

Primero, se tienen que crear los árboles para la red de hubs. Para esto, dado un número de sitios potenciales se busca abrir un número p de hubs. Para esto, se busca utilizar un vector binario, donde un 0 indica que el hub no ha sido abierto y un 1 indica que si se abrió ese hub. Un ejemplo de una solución es la siguiente.



Figura 4.2: Ejemplo de una solución de árboles. 4 hubs con 10 sitios potenciales

Una vez que se tiene el vector binario que indica la selección de hubs, se crean árboles aleatorios con los hubs abiertos. Cabe recordar, que de acuerdo al Teorema 2 visto en la sección de Teoría de Grafos, el número total de árboles que se pueden crear con n nodos es de n^{n-2} . Finalmente, una representación de estos árboles puede ser lograda utilizando matrices de adyacencia.

$$\begin{array}{c}
 \begin{array}{c} H2 \\ H4 \\ H5 \\ H8 \end{array}
 \begin{array}{c}
 H2 \quad H4 \quad H5 \quad H8 \\
 \left[\begin{array}{cccc}
 0 & 1 & 1 & 0 \\
 1 & 0 & 0 & 0 \\
 1 & 0 & 0 & 1 \\
 0 & 0 & 1 & 0
 \end{array} \right]
 \end{array}
 \end{array}
 \qquad
 \begin{array}{c}
 \begin{array}{c} H2 \\ H4 \\ H5 \\ H8 \end{array}
 \begin{array}{c}
 H2 \quad H4 \quad H5 \quad H8 \\
 \left[\begin{array}{cccc}
 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 1 \\
 1 & 1 & 1 & 0
 \end{array} \right]
 \end{array}
 \end{array}$$

Figura 4.3: Ejemplo de dos matrices de adyacencia que representan árboles con 4 nodos.

En la Figura 4.3 se puede ver las conexiones que existen entre los 4 hubs, si algún par de hubs está conectado, se representa con el valor de 1. Por otro lado, si los hubs no están conectados, entonces se representa con un cero. Dado que las matrices de adyacencia son simétricas, el envío de bienes entre hubs es bilateral (se pueden enviar en cualquier dirección).

Una vez que se tiene la selección de los hubs y su matriz de adyacencia, se define la estructura de la red de hubs. Cuando se cuenta con una estructura del árbol definida, ahora hace falta representar la red de distribución que seguirán los bienes. Esto se hace

por medio de una matriz de adyacencia aumentada conformada por submatrices las cuales representan distintas partes de la red general. Un ejemplo de esta matriz aumentada se muestra a continuación.

	<i>i</i>	<i>j</i>	H2	H4	H5	H8
<i>i</i>	0	1	1	1	1	1
<i>j</i>	0	0	0	0	0	0
H2	0	1	0	1	1	0
H4	0	1	1	0	0	0
H5	0	1	1	0	0	1
H8	0	1	0	0	1	0

Figura 4.4: Red general que siguen los bienes

La Figura 4.4 es una matriz aumentada, la cual representa el envío de bienes del cliente i al cliente j . La submatriz de color verde representa la matriz de adyacencia de los clientes, esta matriz no es simétrica ya que se considera que el envío es exclusivamente del cliente i al cliente j . La submatriz naranja representa la conexión entre el cliente i (el origen) y los hubs que han sido abiertos (para este ejemplo se abrieron 4 hubs). Mientras que la submatriz amarilla representa la conexión de los hubs al cliente j (el destino). Por último, en la submatriz de color rojo se representa la estructura del árbol, la cual ya ha sido explicada anteriormente.

4.1.2 RESOLUCIÓN NIVEL INFERIOR

El problema del nivel inferior es parte fundamental dentro del algoritmo, ya que al resolverlo nos permite encontrar precios de manera equilibrada permitiendo el uso óptimo de los arcos de la red de hubs.

Cabe recordar que una vez que se fijan los precios y la red de hubs, el problema del nivel inferior es un problema de flujo mínimo, el cual, se resuelve utilizando el algoritmo clásico de Dijkstra.

Para lograr resolver el problema de nivel inferior, se parte de que se tiene la matriz aumentada de adyacencia y se llena de información de los costos asociados a los bienes. Vale la pena recordar que la matriz de adyacencia sirve para representar un grafo. En-

tonces, se calcula el camino más corto considerando unos precios iniciales iguales a cero, de esta forma se identifican los arcos que utilizan los clientes para transportar los bienes. Esto último permite calcular precios de manera aleatoria para cada uno de los arcos de la red de hubs.

El problema de nivel inferior deberá ser resuelto en múltiples ocasiones durante el proceso de calcular los precios de los arcos, el cual será explicado en la siguiente sección.

4.1.3 GENERACIÓN DE LOS PRECIOS

Finalmente, una de las partes más importantes dentro del algoritmo coevolutivo es la creación inicial de la población aleatoria de los precios. Para esto, una vez que ya se crearon los árboles, y se han encontrado los arcos utilizados al resolver el problema del nivel inferior, se pretende encontrar precios equilibrados para permitir que los usuarios sigan utilizando la red en lugar de mandar de forma directa los bienes. Para ello, se selecciona un arco de la estructura de la red de hubs y se verifica si alguno de los bienes enviados lo ha utilizado, si el arco se ha utilizado, se aumenta el costo en los arcos en una unidad hasta que el camino más corto de alguno de los bienes cambie (se resuelve de nuevo el problema de nivel inferior). Cuando sucede este cambio, se fija ese valor como una cota superior para el arco. Después, se pasa al siguiente arco hasta terminar con todos los arcos usados en la red. Finalmente, se genera un valor aleatorio entre cero y el valor de la cota encontrado para cada uno de los arcos. Con esto, se crea una serie de precios aleatorios para cada arco de cada árbol

Un esquema de este proceso se encuentra a continuación:

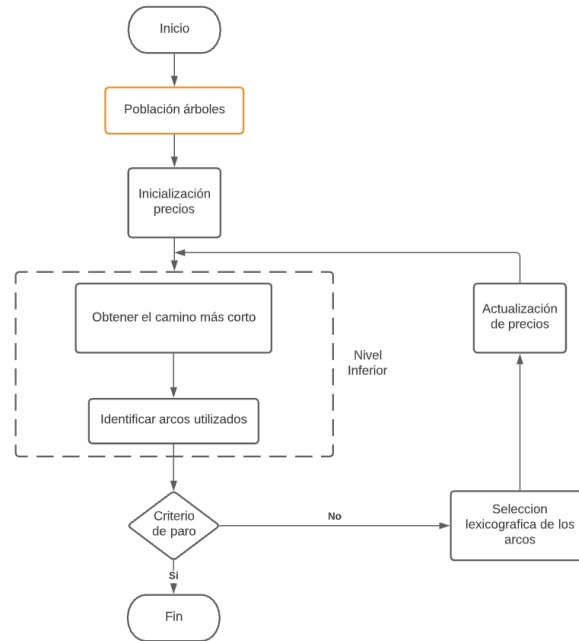


Figura 4.5: Draft del diagrama de flujo [NO LO HAS CAMBIADO]

De acuerdo al Teorema 1 enunciado en la sección de Teoría de grafos, el número de aristas de un árbol con n vértices es $n - 1$. Además, recuerde que asumimos que los costos de envío de bienes entre hubs son distintos dependiendo de la dirección de la cual se envía. Un ejemplo de una solución parcial asociada a los precios se muestra a continuación.

12	20	2	15	21	12
----	----	---	----	----	----

Figura 4.6: Representación de los precios

La Figura 4.6 representa la codificación de los precios obtenidos por el procedimiento ilustrado en la Figura 4.5. Cada recuadro representa un arco del árbol. En este ejemplo, se tienen 3 arcos ya que se considera un árbol con 4 nodos. Sin embargo, al tener 3 arcos dirigidos nos lleva a considerar 6 arcos, esto es, 3 de ida y 3 de vuelta.

Hay que recalcar que los precios son aleatorios. Por lo que no se puede garantizar que estos son los mejores precios que se pueden encontrar. Entonces, las soluciones pasan a una etapa de mejora mediante un proceso evolutivo.

4.2 FUNCIÓN OBJETIVO DEL LÍDER

Una vez se cuenta con la población inicial de árboles y la población de precios para cada árbol, se puede calcular la función objetivo del líder. Para ello, se prueba cada una de las soluciones aleatorias de precios en los árboles correspondientes. Luego, se vuelve a resolver el problema de nivel inferior para obtener los nuevos arcos utilizados por los clientes. Se cuenta el número de veces que un arco apareció, y se multiplica por el precio del arco para obtener la ganancia que genera cada uno de los bienes. Al final se suma y se obtiene la ganancia para cada solución de árbol.

Recordando que cada árbol tiene una ganancia de costo fijo por el uso de la red, se suma dicha ganancia a la ganancia obtenida con los precios. Además, cada uno de los árboles tiene un costo de construcción de la red de hubs, por lo que a la ganancia obtenida por los precios y el costo fijo por el uso de la red se le debe restar el costo de construcción.

Finalmente, a la ganancia total obtenida de lo anterior se le debe multiplicar el flujo o la cantidad de bienes que envía cada uno de los clientes, para finalmente obtener la ganancia total que obtiene el líder en el problema del nivel superior.

Es claro que entre más arcos utilicen los bienes enviados, mayor será la ganancia obtenida. Note que el hecho de utilizar los precios que sigan manteniendo el flujo de la red es muy importante. Además que nos permitirá identificar las mejores redes construidas de hubs.

Es importante entender que las soluciones generadas tanto de árboles como de precios no necesariamente son las mejores desde un principio. Esto se debe a que la construcción de estas soluciones están basadas en la aleatoriedad, por lo que muy probablemente aún se pueda mejorar la estructura de la red y los precios.

4.3 EVOLUCIÓN DE LOS PRECIOS

En este problema, intuitivamente se puede suponer que al fijar precios altos se obtiene una mayor ganancia para el problema del líder. Sin embargo, los precios altos pueden desmotivar al uso de la red. Entonces, debido a que los precios se generaron de forma aleatoria, esta parte es fundamental para el problema. Es obvio considerar que al fijar

precios de forma aleatoria, no garantiza que sean los mejores para el problema. Para la etapa de mejora de precios se propone un algoritmo genético el cual ha demostrado tener muchas ventajas en la generación de redes de hubs [50].

Los algoritmos genéticos, al igual que los evolutivos, están basados en el proceso de evolución de las especies, es decir, los individuos que tengan una mejor aptitud tienen más oportunidades de sobrevivir y tener descendencia. La descendencia se conforma de los rasgos de la ascendencia, además de una serie de mutaciones que afectan las características de los nuevos individuos. Dentro del contexto de la optimización, los individuos representan a las soluciones, de donde aquellas que tengan una mejor evaluación dentro de la función objetivo tienen más oportunidades de ser escogidas para generar nuevas soluciones, las cuales se derivan de las antiguas soluciones pero modificando de manera aleatoria algunos de los atributos para las nuevas soluciones. Un diagrama de flujo del algoritmo genético propuesto e muestra a continuación:

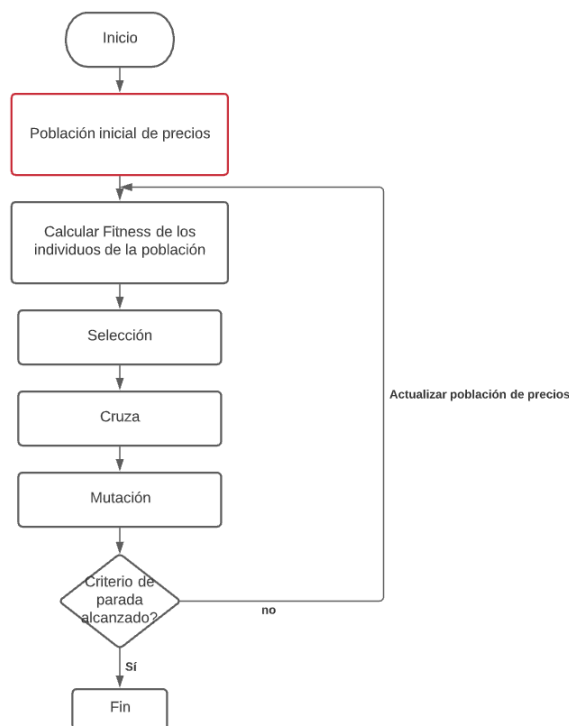


Figura 4.7: Un diagrama de flujo para el algoritmo de evolución de los precios

El algoritmo comienza con una población inicial de precios y a cada una se le evalúa su aptitud en la función objetivo del líder. Se seleccionan las mejores soluciones para entrar a los operadores genéticos. Es decir, a la cruce, la cual consiste en combinar la información de dos soluciones para generar nuevas soluciones. Después, se entra a la muta

para modificar atributos de las soluciones y con esto diversificar el espacio de búsqueda. Este proceso continúa hasta que el criterio de parada se cumpla. A continuación se explica cada una de las funciones del algoritmo dentro del contexto del problema:

- **Población Inicial** Las soluciones (precios) se crean de forma aleatoria considerando cotas superiores para los arcos que están en la red de árbol.
- **Selección** Para la selección de las mejores soluciones, se calcula el valor de su función de aptitud utilizando la función objetivo del líder. Luego, se seleccionan un número fijo del total de las soluciones, la selección se hace de manera elitista, seleccionando aquellas que brinden un mejor valor a la función objetivo.
- **Cruza** En este componente, se crean todas las posibles combinaciones de parejas (los padres) entre las soluciones de precios seleccionadas. Cada una de estas parejas tiene la posibilidad de crear dos nuevas soluciones (los hijos). Para la primera solución se selecciona un punto aleatorio de corte, la información genética de los padres se corta en ese punto, de esta forma el primer hijo hereda primero las características del primer padre y luego del segundo padre. Por otro lado, el segundo hijo (o la segunda solución) hereda primero las características del segundo padre y luego las del primero.
- **Mutación** Se toman todas las soluciones generadas en la cruce y se busca en cada una de ellas el arco cuyo precio sea el menor de todos los arcos. El precio de este arco se modifica al sumarle un porcentaje de los precios originales para aumentar el valor de ese arco.

Este proceso de evolución de soluciones continúa hasta completar un número fijo de soluciones

Un ejemplo gráfico de este proceso se encuentra a continuación.

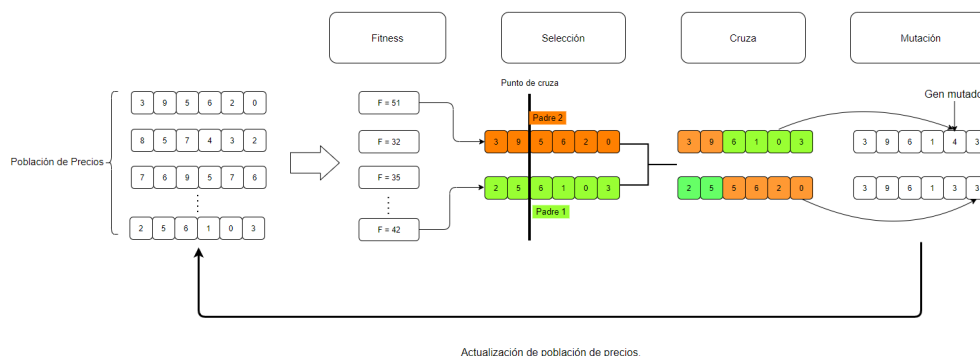


Figura 4.8: Ejemplo gráfico del algoritmo evolutivo

En la Figura 4.8 se comienza con la población generada de la forma descrita en la Sección 4.1 [REF], utilizando estas soluciones generadas, se calcula la función objetivo del líder. Una vez que se ha calculado el valor de todas las funciones objetivo, se selecciona de manera elitista a un subconjunto de ellas. Una vez que se tiene la selección de las mejores soluciones, se hace una recombinación de la información del vector de soluciones de los precios seleccionados. Para ello, se genera un punto de cruce de manera aleatoria (en este caso, el punto de cruce es a partir de la segunda posición del vector solución). Este proceso de cruce, genera un par de nuevas soluciones donde el precio de los dos primeros arcos de la nueva solución generada corresponde a los precios del padre 1, mientras que el resto corresponde a los precios del padre 2, mientras que en el segundo hijo sucede lo contrario. Finalmente, para diversificar las soluciones, estas son sometidas a un proceso de mutación. Para ello, se selecciona el arco cuyo precio es menor y se modifica sumándole una cantidad al precio de ese arco. Finalmente, con estas nuevas soluciones, se actualiza la lista inicial de la población de precios, para ver si mejora o no.

Es importante mencionar que aunque las soluciones de precios han evolucionado a través de este proceso, la población de árboles permanece estática. Sin embargo, más adelante se detalla el componente referente a la coevolución de las poblaciones. El hecho de que la población de árboles permanezca estática va a evolucionar de manera independiente, ya que puede existir otra selección de hubs y arcos que mejoren la experiencia de los clientes, así como la ganancia del creador de esta red de distribución

En la siguiente sección, se muestra un proceso en el cual se van creando nuevas soluciones de manera que impacten positivamente en la función objetivo del líder.

4.4 SELECCIÓN ELITISTA DE UN NUEVO ÁRBOL.

Una vez que ha evolucionado la población de los precios, se vuelve a resolver el nivel inferior identificando y contando el número de veces que cada hub ha sido usado por los bienes enviados. De esta forma, cuando se crea una nueva solución, lo hará de manera que se seleccionen los hubs que más han sido utilizados. Si alguno de los hubs no ha sido utilizado en la población inicial de árboles, se le asigna una probabilidad menor de ser escogido. De esta manera, la nueva solución es creada de una manera voraz-aleatoria.

Finalmente, se actualiza la población de árboles con las nuevas soluciones agregadas. Esto permite mantener o diversificar aún más a la población.

Por último, el proceso de coevolución es detallado.

4.5 COEVOLUCIÓN

Una vez que la población de árboles es actualizada, se vuelve a realizar el proceso para encontrar los precios asociados a la población actualizada de árboles y entrar a la fase de mejora y posteriormente actualizar de nuevo las poblaciones, de esta manera nuevas soluciones van creando nuevas poblaciones para tratar de encontrar la estructura y precios óptimos. Este proceso de coevolución se continúa realizando hasta un número fijo de iteraciones sin mejora.

Explicado el algoritmo, en el siguiente capítulo se hablará de la generación de las instancias utilizadas, de los parámetros utilizados y finalmente la experimentación realizada así como los resultados serán mostrados.

CAPÍTULO 5

EXPERIMENTACIÓN COMPUTACIONAL Y RESULTADOS

La experimentación fue realizada utilizando un procesador Intel Core i5-10400f de 2.9 GHz con 16 Gb de RAM, corriendo en un sistema operativo Windows 10. Los algoritmos fueron implementados utilizando el lenguaje de programación Python. El conjunto de instancias se encuentran en la literatura del CAB dataset introducida por O Kelly [14] modificando algunos de los parámetros del problema. Se analizaron 110 instancias divididas en 4 tipos, dependiendo del número de clientes utilizados.

Ya se han descrito los parámetros del problema, igual se hace un recordatorio de estos en la siguiente tabla

Parametro	Valor
$ I $	25, 20, 15, 10
$ J $	25, 20, 15, 10
$ K $	4, 8
p	3, 5, 10
c_a	Costo directo entre clientes
$d_{o(a)i}$	Costo fijo de acceso a la red
$d_{id(a)}$	Costo fijo de salida de la red
d_{il}	Costo fijo dentro de la red
g_{ail}	Costo fijo por uso de la red
f_{il}	Costo de construcción de los arcos de la red
w_{jk}	Flujo del bien

Tabla 5.1: Tabla de parámetros

Los distintos costos que se mencionan en la tabla anterior se obtienen calculando la distancia de haversine dado las coordenadas de los hubs y clientes. Para las instancias de los clientes de tamaño 20, 15, 10 se toman submatrices de la instancia de tamaño 25. Para los bienes escogidos, se toman de manera aleatoria pares de clientes origen destino los cuales envían. A continuación se muestran los resultados de la experimentación

I	J	K	p	Instancia	F best	Promedio	F worst	tiempo(seg)	GAP (%)
25	25	4	3	25-25-4-3-a	19300091.2	18807701.8	18058193.7	607.45	2.5(%)
				25-25-4-3-b	17532418.2	16709083.4	15465059.2	541.6	4.6(%)
				25-25-4-3-c	40589917.1	38415687.9	36050503.1	560.92	5.3(%)
				25-25-4-3-d	25877966.9	24151115.7	22139298.1	973.09	6.6(%)
				25-25-4-3-e	11270057.0	10784182.7	10083725.9	571.08	4.3(%)
			5	25-25-4-5-a	36859621.1	34090994.5	31471115.6	454.17	5.5(%)
				25-25-4-5-b	18585539.7	17307997.5	15249499.0	418.9	6.8(%)
				25-25-4-5-c	49679034.5	47524195.3	45144547.2	780.72	4.3(%)
				25-25-4-5-d	30675985.6	28309626.2	27088819.0	1081.67	6.4(%)
				25-25-4-5-e	11427378.0	10839370.0	10203185.32	529.87	5.1(%)
			10	25-25-4-10-a	18427918.1	17575907.3	16936001.91	1081.14	4.6(%)
				25-25-4-10-b	25454982.9	24762511.0	23260499.7	2082.06	2.7(%)
				25-25-4-10-c	46964733.3	44731158.7	42509284.8	1651.55	4.7(%)
				25-25-4-10-d	28632491.1	27111327.6	25356631.2	2630.57	5.3(%)
				25-25-4-10-e	11891281.0	11366798.4	11039729.3	1492.03	4.4(%)

Tabla 5.2: Tabla de resultados: 25 hubs, 25 clientes, 4 bienes

Para el primer conjunto de instancias para los 4 bienes, se divide en tres valores para la función objetivo, el primero es el mejor valor de la función objetivo del líder obtenido del número total de corridas realizadas del algoritmo, el segundo es el promedio obtenido del total de las corridas y el último es el peor valor obtenido de estas corridas. Algo que se puede notar a simple vista es que el tiempo generado por el algoritmo es relativamente pequeño con un tiempo promedio menor a una hora. Por último se puede ver que el GAP es pequeño con un promedio general menor al 5 %, sin embargo, creemos que estos tiempos se pueden mejorar.

Dentro del contexto del problema se puede ver que en la mayoría de instancias, abrir 5 hubs resulta mejor que abrir 3 y 10, esto es importante notarlo ya que los clientes prefieren utilizar más hubs contar de ahorrar dinero o tiempo, pero no lo suficiente como para estar utilizando una red de 10 hubs.

I	J	K	p	Instancia	F best	Promedio	F worst	(seg)	GAP (%)
20	20	4	3	20-20-4-3-a	47585094.2	45339154.74	42748295.28	530.56	4.7%
				20-20-4-3-b	51224948.8	48943941.1	46713159.2	417.96	4.4%
				20-20-4-3-c	90612070.1	88233536.3	85639570.6	796.11	2.6%
				20-20-4-3-d	26113372.8	25133703.6	23709559.8	1046.29	3.7%
				20-20-4-3-e	15736738.5	14764196.3	10557154.4	397.08	6.1%
			5	20-20-4-5-a	48601833.1	45949875.2	42617063.6	937.06	5.4%
				20-20-4-5-b	63106734.7	58865556.5	55784126.0	790.95	6.7%
				20-20-4-5-c	90519730.2	84641445.0	80014636.5	1179.49	6.4%
				20-20-4-5-d	30243353.4	29487945.9	29019329.5	1382.59	2.4%
				20-20-4-5-e	15062656.2	14488037.6	14046450.3	866.04	3.8%
			10	20-20-4-10-a	45870782.6	43544766.3	40081966.5	2314.52	5%
				20-20-4-10-b	64887054	62121668.3	60185608.4	2141.15	4.2%
				20-20-4-10-c	83032236.0	78279179.2	73895471.5	2814.21	5.7%
				20-20-4-10-d	25799182.1	25029696.3	24052268.3	3149.08	2.9%
				20-20-4-10-e	16936833.5	15823815.3	15038583.6	2137.06	6.5%

Tabla 5.3: Tabla de resultados: 20 hubs, 20 clientes, 4 bienes

Para el resto de conjuntos de instancias se obtuvieron resultados similares, el valor objetivo obtenido por las distintas tablas, en general es mayor que las obtenidas en el primer conjunto de instancias, lo cual es logico pensarlo ya que se envían más bienes entre los clientes. Podemos observar como de nuevo, en general los clientes prefieren utilizar 5 hubs que 3 y 10, aunque hay algunos casos excepcionales, sin embargo el gap mostrado cuando se abren 5 hubs es en general más grande que con los otros hubs, por lo que creemos que en estos casos se pueden aun seguir mejorando el algoritmo, además que el tiempo de computación es más grande debido a un número mayor de commodities sin superar de manera general las 2 horas.

I	J	K	p	Instancia	F best	Promedio	F worst	(seg)	GAP (%)
15	15	4	3	15-15-4-3-a	26825250.62	25362100.1	24343703.92	474.41	5.4%
				15-15-4-3-b	9706133.2	9407742.8	9092652.5	345.79	3.07%
				15-15-4-3-c	29518761.2	27573970.2	26194037.6	448.11	6.5%
				15-15-4-3-d	2983227.5	2878728.5	2638658.2	465.79	3.5%
				15-15-4-3-e	4995283.9	4782707.4	4339223.1	510.43	3.6%
			5	15-15-4-5-a	24845600.8	23653699.2	21943882.7	756.44	4.7%
				15-15-4-5-b	12090464.6	11569032.0	11018288.1	748.93	4.3%
				15-15-4-5-c	19725257.2	18712873.3	18176737.9	802.37	5.1%
				15-15-4-5-d	2583227.5	2449872.7	2332359.5	674.02	5.05%
				15-15-4-5-e	6498412.8	6137761.3	5757109.4	926.22	5.5%
			10	15-15-4-10-a	24377969.6	22749861.3	21203585.4	1914.2	6.6%
				15-15-4-10-b	12756000.7	11963570.5	11026948.8	2119.48	6.2%
				15-15-4-10-c	13690513.0	13057317.9	12339844.1	1512.34	4.6%
				15-15-4-10-d	1941863.2	1795784.4	1579932.6	2584.8	7.5%
				15-15-4-10-e	7060183.5	6747359.7	6290357.6	1997.2	4.4%

Tabla 5.4: Tabla de resultados: 15 hubs, 15 clientes, 4 bienes

I	J	K	p	Instancia	F best	F promedio	F worst	(seg)	GAP (%)
10	10	4	3	10-10-4-3-a	5935745.6	5631810.0	5318541.3	271.97	5.1%
				10-10-4-3-b	8856594.5	8533255.8	7991357	410.08	3.6%
				10-10-4-3-c	17885784.3	17555486.4	17083885.9	469.77	1.8%
				10-10-4-3-d	20942805.7	19564658.0	18045257.3	511.03	6.5%
				10-10-4-3-e	42568051.5	40704702.3	38279686.7	413.5	4.3%
			5	10-10-4-5-a	7058226.8	6608751.9	6145537.5	465.41	6.3%
				10-10-4-5-b	9328688.5	8826289.4	8167103.5	885.31	5.3%
				10-10-4-5-c	15473546.9	14572848.4	13090254.3	673.01	5.8%
				10-10-4-5-d	17011733.5	15804882.5	14086681.4	869.19	7%
				10-10-4-5-e	36836337.3	34431288.9	31561182.0	625.26	6.5%

Tabla 5.5: Tabla de resultados: 10 hubs, 10 clientes, 4 bienes

I	J	K	p	Instancia	F best	Promedio	F worst	tiempo(seg)	GAP (%)
25	25	8	3	25-25-8-3-a	48726398.27	44622948.65	40877498.4	1426.3	8.8(%)
				25-25-8-3-b	68716866.14	64023451.42	59237377.24	1165.26	6.8(%)
				25-25-8-3-c	97897575.18	91262430.44	87157725.1	1209.85	6.7(%)
				25-25-8-3-d	56631715.03	53942767.22	49077819.9	1511.68	4.7(%)
				25-25-8-3-e	146721924.3	141524952.4	137630595.9	1568.12	4.3(%)
			5	25-25-8-5-a	54637964.8	52078756.09	48199360.5	2454.17	4.6(%)
				25-25-8-5-b	78303372.33	73817594.58	69934620.4	2072.8	6.8(%)
				25-25-8-5-c	118492755.8	112793992.4	109929668.7	2209.85	4.8(%)
				25-25-8-5-d	57874974.58	52932600.8	49644953.6	2526.37	8.5(%)
				25-25-8-5-e	124684859.7	118958338.9	114289838.7	2445.15	4.5(%)
			10	25-25-8-10-a	39186701.77	37306507.39	35258217.6	3149.45	4.7(%)
				25-25-8-10-b	75984864.57	72111465.12	68595517.8	3417.76	5(%)
				25-25-8-10-c	92595203.62	87479959.84	82738388.8	2578.89	5.5(%)
				25-25-8-10-d	57685142.61	54296351.99	50361552.4	2630.57	5.8(%)
				25-25-8-10-e	151956359.5	143822129.9	135112119.7	2009	5.3(%)

Tabla 5.6: Tabla de resultados: 25 hubs, 25 clientes, 8 bienes

I	J	K	p	Instancia	F best	Promedio	F worst	(seg)	GAP (%)
20	20	8	3	20-20-8-3-a	6954698.31	6488606.702	5894367.74	450.32	6.7%
				20-20-8-3-b	10579206.42	10083539.67	8560193.61	417.96	4.6%
				20-20-8-3-c	31946219.271	29097714.21	24886260.28	573.431	8.9%
				20-20-8-3-d	115375064.8	109452133.8	104176399.3	1046.29	5.1%
				20-20-8-3-e	120118768.6	114073646.5	108459156.8	421.37	5.03%
			5	20-20-8-5-a	7789699.15	7573121.868	7181909.77	565.77	2.7%
				20-20-8-5-b	10847918.1	10441686.19	10169396.7	890.95	3.7%
				20-20-8-5-c	33546217.5	31563261.88	28826256.32	995.89	5.9%
				20-20-8-5-d	119232936.8	115328629.5	110941760.8	936.02	3.2%
				20-20-8-5-e	53338481.7	51759113.79	50176248.42	906.52	2.9%
			10	20-20-8-10-a	7192837.51	6826463.093	6147216.98	2209.92	5.09%
				20-20-8-10-b	9073021.97	8668904.617	8206787.9	2141.15	4.4%
				20-20-8-10-c	83032236.0	78279179.2	73895471.5	2814.21	5.7%
				20-20-8-10-d	97910732.3	96174756.51	92726885.7	4776.21	1.7%
				20-20-8-10-e	45695371.82	43880172.97	41114597.7	5064.28	3.9%

Tabla 5.7: Tabla de resultados: 20 hubs, 20 clientes, 8 bienes

I	J	K	p	Instancia	F best	Promedio	F worst	(seg)	GAP (%)
15	15	8	3	15-15-8-3-a	30475952.55	29588914.98	28189157.85	767.12	2.9%
				15-15-8-3-b	15828378.33	14991386.88	14552795.56	345.79	5.2%
				15-15-8-3-c	55817028.33	54287722.7	52937283.33	575.2	2.7%
				15-15-8-3-d	60881982.44	58140758.7	54763037.82	465.79	4.5%
				15-15-8-3-e	79862148.49	76968378.0	74000375.65	606.55	3.6%
			5	15-15-8-5-a	36770335.51	38675075.33	34611773.98	1188.82	4.9%
				15-15-8-5-b	21923406.32	21074598.16	20086842.82	1513.48	4.3%
				15-15-8-5-c	54791864.01	51313463.02	47274858.56	1258.36	6.3%
				15-15-8-5-d	68482658.17	65377631.06	62111395.16	1449.57	4.5%
				15-15-8-5-e	115914858.9	113445076.6	110387871.1	1906.55	2.1%
			10	15-15-8-10-a	40917296	39543953.04	38355928.62	2672.49	3.3%
				15-15-8-10-b	20103820.87	19576635.27	19052210.72	2877.38	2.6%
				15-15-8-10-c	59805440.51	58203533.77	54403937.87	1573.12	2.6%
				15-15-8-10-d	60045524.06	57411975.38	53237283.14	2584.8	4.3%
				15-15-8-10-e	79872941.7	77384182.8	72797198.56	1997.2	3.1%

Tabla 5.8: Tabla de resultados: 15 hubs, 15 clientes, 4 bienes

I	J	K	p	Instancia	F best	F promedio	F worst	(seg)	GAP (%)
10	10	8	3	10-10-4-3-a	5935745.6	5631810.0	5318541.3	271.97	5.1%
				10-10-4-3-b	8856594.5	8533255.8	7991357	410.08	3.6%
				10-10-4-3-c	17885784.3	17555486.4	17083885.9	469.77	1.8%
				10-10-4-3-d	20942805.7	19564658.0	18045257.3	511.03	6.5%
				10-10-4-3-e	42568051.5	40704702.3	38279686.7	413.5	4.3%
			5	10-10-4-5-a	7058226.8	6608751.9	6145537.5	465.41	6.3%
				10-10-4-5-b	9328688.5	8826289.4	8167103.5	885.31	5.3%
				10-10-4-5-c	15473546.9	14572848.4	13090254.3	673.01	5.8%
				10-10-4-5-d	17011733.5	15804882.5	14086681.4	869.19	7%
				10-10-4-5-e	36836337.3	34431288.9	31561182.0	625.26	6.5%

Tabla 5.9: Tabla de resultados: 10 hubs, 10 clientes, 4 bienes

CAPÍTULO 6

CONCLUSIONES Y TRABAJO A FUTURO

En esta tesis, se presenta el problema binivel de localización de hubs en forma de árbol. Observamos que en los problemas clásicos de hubs se suelen asumir que la red de hubs es completa o que hay un solo tomador de decisiones, sin embargo esto no siempre es válido, además de las decisiones de localización de hubs y la asignación de clientes a estos, decidir la estructura de la red además de asignar precios por el uso de esta es necesario. Una de las muchas estructuras mencionadas en la literatura es una estructura en forma de grafo de árbol, la cual presenta muchas ventajas a las estructuras clásicas tales como envío y recibo de bienes sin la preocupación de realizar ciclos, además de que el camino que sigue uno de estos bienes es único, por lo que el costo siempre será el mínimo posible. Otro de los aspectos más importantes del problema es la introducción de una variable que asigna precios a los arcos de la red, en la literatura relacionado a problemas de pricing se suelen considerar a los precios ya fijos, por lo que el problema al final resulta en un problema de flujo mínimo, al considerar los precios variables, el problema se complica mucho más. En este problema existen dos tomadores de decisiones, el primero busca abrir hubs, crear la red y asignar precios de manera de maximizar la ganancia por usar la red, mientras que el segundo busca minimizar el costo por usar la red.

Para ello definimos un modelo de programación binivel que busca modelar esta situación, el cual debido al contexto del problema del problema se puede utilizar de manera eficiente. En la literatura estudiada se puede ver que no existe ninguna mención a un problema similar, por lo que creemos que la introducción de este problema al campo de localización de instalaciones puede ayudar a encontrar modelos más robustos que ayuden a modelar estas o nuevas situaciones.

Para ello, adaptamos instancias ya establecidas en la literatura para resolver el problema, sin embargo ante la reformulación plantada nos dimos cuenta que solo podía resolver instancias muy limitadas por lo que la realización de un algoritmo metaheurístico era necesario para poder resolver el problema. Debido a la naturaleza de las soluciones que tenía que tener el problema, y viendo que las soluciones de precios y la red de hubs

evolucionaban de manera independiente, decidimos crear un algoritmo coevolutivo en donde las dos soluciones mejoraban utilizando características de la otra.

En los resultados de nuestra experimentación mostraron la gran ventaja del algoritmo, permitiendo obtener soluciones aceptables en un tiempo computacional relativamente rapido, además que las diferencia entre las soluciones generadas por el algoritmo es relativamente pequeña por lo que se podría intuir que cada vez se iban encontrando soluciones que podrían ser óptimas.

La importancia de los problemas de localización de hubs es cada vez más grande además de las variantes de este problema, es importante mostrar las ventajas y desventajas que tiene el modelo propuesto con otros modelos introducidos en la literatura.

Algunos enfoques que se podría realizar en trabajo a futuro es la realización de un diseño de experimentos que permita un mejor ajuste de los parametros ya que el problema depende mucho de ellos, además de verificar la efectividad del algoritmo corriendo instancias más grandes para el problema.

BIBLIOGRAFÍA

- [1] Reza Zanjirani Farahani, Masoud Hekmatfar, Alireza Boloori Arabani, and Ehsan Nikbakhsh. Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4):1096–1109, 2013.
- [2] Ann Campbell, Timothy Lowe, and Li Zhang. The p-hub center allocation problem. *European Journal of Operational Research*, 176:819–835, 02 2007.
- [3] Alfred Weber. *Theory of the Location of Industries*. University of Chicago Press, 1929.
- [4] The Cornuejols. The uncapacitated facility location problem. 02 1999.
- [5] Richard Church and Charles ReVelle. The maximal covering location problem. In *Papers of the regional science association*, volume 32, pages 101–118. Springer-Verlag, 1974.
- [6] S Louis Hakimi. On locating new facilities in a competitive environment. *European Journal of Operational Research*, 12(1):29–35, 1983.
- [7] Sibel Alumur Alev and Bahar Kara. Network hub location problems: The state of the art. *European Journal of Operational Research*, 190:1–21, 10 2008.
- [8] Shahin Gelareh and Stefan Nickel. Hub location problems in transportation networks. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):1092–1111, 2011.
- [9] Stefan Nickel, Anita Schöbel, and Tim Sonneborn. Hub location problems in urban traffic networks. In *Mathematical methods on optimization in transportation systems*, pages 95–107. Springer, 2001.
- [10] Masoud Hekmatfar and Mirsaman Pishvaei. Hub location problem. In *Facility Location*, pages 243–270. Springer, 2009.

-
- [11] Ling-Yun Wu and Xiang Zhang. Capacitated facility location problem with general setup cost. 33:1226–1241, 05 2006.
- [12] AJ Goldman. Optimal locations for centers in a network. *Transportation Science*, 3(4):352–360, 1969.
- [13] Morton O’Kelly. The location of interacting hub facilities. *Transportation Science*, 20:92–106, 05 1986.
- [14] Morton O’Kelly. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32:393–404, 12 1987.
- [15] James Campbell. Integer programming formulations of discrete hub location problem. *European Journal of Operational Research*, 72:387–405, 01 1994.
- [16] J. G. Klincewicz. A dual algorithm for the uncapacitated hub location problem. 1996.
- [17] John G Klincewicz. Heuristics for the p-hub location problem. *European Journal of Operational Research*, 53(1):25–37, 1991.
- [18] Elisangela Martins de Sá, Reinaldo Morabito, and Ricardo Saraiva de Camargo. Benders decomposition applied to a robust multiple allocation incomplete hub location problem. *Computers & Operations Research*, 89:31–50, 2018.
- [19] Fereidoon Habibzadeh Boukani, Babak Farhang Moghaddam, and Mir Saman Pishvae. Robust optimization approach to capacitated single and multiple allocation hub location problems. *Computational and Applied Mathematics*, 35(1):45–60, 2016.
- [20] Armaghan Alibeyg, Ivan Contreras, and Elena Fernández. A lagrangean relaxation for uncapacitated hub location problems with profits.
- [21] Ivan Contreras. *Hub Location Problems*, pages 311–344. 01 2015.
- [22] James Campbell. Hub location and the p-hub median problem. *Operations Research*, 44:923–935, 12 1996.
- [23] Ivan Contreras, Elena Fernández, and Alfredo Mart n. The tree of hubs location problem. *European Journal of Operational Research*, 202(2):390–400, 2010.
- [24] Ivan Contreras, Elena Fernández, and Alfredo Mart n. Tight bounds from a path based formulation for the tree of hub location problem. *Computers and Operations Research*, 36(12):3117–3127, 2009.

-
- [25] Youngho Lee, Lu Lu, Yuping Qiu, and Fred Glover. Strong formulations and cutting planes for designing digital data service networks. *Telecommunication Systems*, 2:261–274, 12 1993.
- [26] Youngho Lee, Byung Ha Lim, and June S Park. A hub location problem in designing digital data service networks: Lagrangian relaxation approach. *Location Science*, 4(3):185–194, 1996.
- [27] S. Hakimi, Edward Schmeichel, and Martine Labbé. On locating path- or tree-shaped facilities on networks. *Networks*, 23:543 – 555, 09 1993.
- [28] Te C Hu. Optimum communication spanning trees. *SIAM Journal on Computing*, 3(3):188–195, 1974.
- [29] Elisangela Martins de Sá, Ricardo Camargo, and G. Miranda. An improved benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research*, 226:185-202, 04 2013.
- [30] Luciana Pessoa, Andréa Santos, and Mauricio Resende. A biased random-key genetic algorithm for the tree of hubs location problem. *Optimization Letters*, 11:1371-1384, 10 2017.
- [31] Miguel Pozo, Justo Puerto, and Antonio Chía. The ordered median tree of hubs location problem. *TOP*, 06 2020.
- [32] Víctor Blanco and Alfredo Marín. Upgrading nodes in tree-shaped hub location. *Computers and Operations Research*, 102, 02 2018.
- [33] Gita Taherkhani and Sibel Alumur Alev. Profit maximizing hub location problems. *Omega*, 86, 02 2018.
- [34] Vladimir Marianov, Daniel Serra, and Charles ReVelle. Location of hubs in a competitive environment. *European Journal of Operational Research*, 114(2):363–371, 1999.
- [35] Mihiro Sasaki and Masao Fukushima. Stackelberg hub location problem. *Journal of the Operations Research Society of Japan*, 44, 12 2001.
- [36] Armin Luer-Villagra and Vladimir Marianov. A competitive hub location and pricing problem. *European Journal of Operational Research*, 231, 06 2013.
- [37] Dimitrije Cvokic, Yury Kochetov, and Plyasunov Aleksandr. A leader-follower hub location problem under fixed markups. volume 9869, pages 350-363, 09 2016.

-
- [38] Maryam Esmaeili and Samane Sedehzade. Designing a hub location and pricing network in a competitive environment. *Journal of Industrial and Management Optimization*, 13:1–15, 01 2017.
- [39] Armaghan Alibeyg, Ivan Contreras, and Elena Fernández. Hub network design problems with profits. *Transportation Research Part E Logistics and Transportation Review*, 96:40–59, 2016.
- [40] Armaghan Alibeyg, Ivan Contreras, and Elena Fernández. Exact solution of hub network design problems with profits. *European Journal of Operational Research*, 266(1):57–71, 2018.
- [41] Dimitrije Cvokic and Zorica Stanimirovic. A single allocation hub location and pricing problem. *Computational and Applied Mathematics*, 39, 03 2020.
- [42] Jonathan F Bard. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media, 2013.
- [43] Jonathan F Bard. Some properties of the bilevel programming problem. *Journal of optimization theory and applications*, 68(2):371–378, 1991.
- [44] Leslie R Foulds. *Graph theory applications*. Springer Science & Business Media, 2012.
- [45] John W Moon. Various proofs of cayley formula for counting trees. In *A seminar on Graph Theory*. Holt, Rinehart and Winston, 1967.
- [46] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [47] Mitchell A Potter and Kenneth A De Jong. A cooperative coevolutionary approach to function optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 249–257. Springer, 1994.
- [48] Elena Popovici, Anthony Bucci, R Paul Wiegand, and Edwin D De Jong. *Coevolutionary principles.*, 2012.
- [49] Jan Paredis. *Coevolutionary algorithms*, pages 224–238. 11 2000.
- [50] Xin-She Yang. *Nature-inspired optimization algorithms*. Academic Press, 2020.

APÉNDICE A

CIUDADES DEL CAB DATASET

